OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

Computer Forensics in Cyber-Physical Systems
Applying Existing Forensic Knowledge and Procedures from Classical IT to Automation and Automotive

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von Dipl. Inf. Robert Altschaffel

geb. am 15.07.1984 in Halle

Gutachterinnen/Gutachter

Prof. Dr-Ing. Jana Dittmann
Prof. Dr-Ing. Felix Freiling
Prof Dr. Stefan Katzenbeisser

Magdeburg, den 12.10.2020

University of Magdeburg

School of Computer Science

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

Dissertation

Computer Forensics in Cyber-Physical Systems - Applying Existing Forensic Knowledge and Procedures from Classical IT to Automation and Automotive

Author:

Robert Altschaffel

2020

Advisors:

Prof. Dr.-Ing. Jana Dittmann

Institute of Technical and Business Information Systems

Altschaffel, Robert:
Computer Forensics in Cyber-Physical Systems - Applying Existing Forensic Knowledge and Procedures from Classical IT to Automation and Automotive
Dissertation, University of Magdeburg, 2020.

Abstract

English

This thesis contributes to the topic of computer forensics in the domains of Industrial Control Systems (ICS) and Automotive IT.

Computer forensics in classical IT systems, consisting of potentially networked Desktop Computers (from here on referred to as *Desktop IT*), is a well-researched topic. However, computer forensics in the Industrial Control Systems (ICS) and Automotive IT is still an emerging field. This thesis investigates the potential application of methods, procedures and processes from the field of computer forensics in the Desktop IT domain to the ICS and automotive domains (in short referred to as *investigated domains*). This is necessary due to an increasing amount of known attacks against ICS and Automotive IT systems.

This thesis describes and completes the adaptation of concepts, methods, procedures and processes from the field of computer forensics in Desktop IT to these domains. These concepts, methods, procedures and processes are based on a forensic process model to which the author of this thesis contributed. The iteration of this forensic process model published in [KDV15] provides the foundation for the considerations during the scope of this thesis.

In order to apply and adapt methods, procedures and processes from the field of computer forensics in Desktop IT to these domains, an understanding of the goals of the computer forensic process is necessary. As forensic science can be applied to achieve different aims in varying contexts, which carry different implications for the forensic process (for example in terms of admissibility of evidence to court, adherence to restrictions or simply in the nature of the evidence useful to investigate a given suspicion) these *Investigative Contexts* are formalized within this thesis. A survey on forensic procedures and methods in the classical IT domain identifies the various aspects required for a process model to perform forensic investigations into computer systems. This survey is used as a foundation to establish eight *Forensic Process Model Criteria* for a comprehensive forensic process model during this thesis.

These criteria are used during the exploration of the [KDV15] forensic process model and serve as a backdrop for a discussion of various aspects of this model. After a discussion on how this forensic process model addresses the various criteria, the model is then enhanced and adapted to handle the specifics of Desktop IT as well as the investigated domains.

The main approach is a thorough and systematic analysis of the three domains (Desktop IT, Industrial Control Systems and Automotive IT) in order to identify which properties influence the computer forensic process in light of the given aspects. This is achieved by analyzing the employed components, the employed communication architectures and the scenarios in which these are employed. Analyzing known attack scenarios in the respective domains provides additional input. In order to improve the understanding of potential forensic traces in the ICS and Automotive IT domains a forensic-driven view on the employed components is established.

The analysis of these domains leads to the identification of 29 *Impact Factors* from the ICS domain and 25 *Impact Factors* from the Automotive IT domain which influence the forensic process in these domains. These *Impact Factors* are used to describe the similarity and differences between the domains discussed in this thesis. The ICS and Automotive IT share many of these influences on the forensic process while they differ greatly from the Desktop IT domain.

While specific tools and methods differ, the aspects of the [KDV15]-model for the forensic process can, in general, be applied to the ICS and Automotive IT domains. The [KDV15]-model consists of *Investigation Steps*, *Data Types* and *Classes of Methods*. The investigation and alteration of these aspects in the light of the investigated domains is presented within this thesis.

These alterations amount to five major adaptations of the forensic process as described in [KDV15]. Firstly, for the *Investigation Steps*, the emphasis of various phases of the forensic process shifts. Since the components employed in the investigated domains have very limited resources, the *Strategic preparation* (*SP*) phase is essential to enable a meaningful amount of forensic traces which can be used to further a forensic investigation. Secondly, the overall structure of the forensic process is slightly altered. Thirdly, most of the *Data Types* require a clearer definition in order to fulfill their role in supporting the forensic process by attaching methods for gathering or analyzing various types of data. Fourthly, one new data type is introduced. Finally, the *Classes of Methods* are redefined in order to improve the forensic process within the investigated domains.

This altered process model is put to the test in a complex case study in the ICS domain.

The main contribution of this thesis is the transfer of knowledge from the classical IT domain to the ICS and Automotive IT domains. This increases the maturity of computer forensics in these two distinct domains. This thesis discusses the forensic traces available in these different domains based on scientific and comprehensive analysis of the given domains. In addition, the methods available to acquire and investigate these traces are shown. Some of the methods and tools currently missing to increase the maturity of computer forensics in these domains are identified.

The research is supported by a notable number of relevant publications and industrial research.

German

Diese Arbeit beschäftigt sich mit dem Gebiet der Computerforensik in den Bereichen Industrieller Steuernetzwerke (ICS) und Automotiver IT. Computerforensik in klassischen IT-Systemen, bestehend aus potenziell vernetzten Desktop-Computern (von nun an als *Desktop IT* bezeichnet), ist ein umfangreich erforschtes Feld. Die Computerforensik in den Bereichen Industrieller Steuernetzwerke (ICS) und Automotive IT ist jedoch noch ein erst entstehendes Forschungsfeld. Diese Arbeit untersucht die mögliche Übertragung von Methoden, Verfahren und Prozessen aus dem Bereich der Computerforensik im Anwendungsgebiet der Desktop-IT auf die Anwendungsgebiete Industrieller Steuernetzwerke (ICS) und Automotive IT (kurz als *untersuchte Domänen* bezeichnet). Dies ist aufgrund einer zunehmenden Anzahl bekannter Angriffe auf Industrielle Steuernetzwerke (ICS) und Automotive IT erforderlich. Diese Arbeit beschreibt und vervollständigt die Anpassung von Konzepten, Methoden, Verfahren und Prozessen aus dem Anwendungsgebiet der Computerforensik in der Desktop-IT an die untersuchten Domänen. Diese Konzepte, Methoden, Verfahren und Prozesse basieren auf einem forensischen Prozessmodell, zu dem der Autor dieser Arbeit beigetragen hat. Die in [KDV15] veröffentlichte Iteration dieses forensischen Prozessmodells bildet die Grundlage für die Überlegungen im Rahmen dieser Arbeit.

Um Methoden, Verfahren und Prozesse aus der Computerforensik aus dem Anwendungsgebiet der Desktop-IT auf diese Bereiche anwenden und anpassen zu können, ist ein Verständnis der Ziele der computerforensischen Untersuchung erforderlich. Forensische Untersuchungen können angewendet werden, um unterschiedliche Ziele in unterschiedlichen Kontexten zu erreichen. Dies hat unterschiedliche Auswirkungen auf den forensischen Prozess, beispielsweise in Bezug auf die Zulässigkeit von Beweismitteln vor Gericht oder die Einhaltung von Beschränkungen bei der Erhebung und dem Umgang mit Beweismitteln oder einfach nur in Bezug auf die Art der Beweismittel welche für einen bestimmten Sachverhalt nützlich erscheinen. Diese Arbeit formalisiert diesen Umstand in Form von zwei *Investigativen Kontexten* (*Investigative Contexts*).

Eine Studie verschiedener forensische Prozessmodelle aus dem Bereich der Desktop-IT wird verwendet um, gemeinsam mit dem den verschiedenen zuvor erwähnten Aspekten, zu ergründen welche Eigenschaften für ein umfassendes forensisches Prozessmodel wünschenswert sind. Diese Eigenschaften werden in Form von acht *Kriterien für forensische Prozessmodelle* (textit Forensic Process Model Criteria) formalisiert. Diese Kriterien werden bei der Untersuchung des forensischen Prozessmodells aus [KDV15] verwendet und dienen als Hintergrund für die Diskussion verschiedener Aspekte dieses Modells. Nach einer Diskussion darüber, wie dieses forensische Prozessmodell die verschiedenen Kriterien berücksichtigt, wird das Modell erweitert und an die Besonderheiten der untersuchten Domänen angepasst. Das Ziel ist dabei, dass das Modell sowohl der Computerforensik in der Desktop-IT als auch in den untersuchten Domänen dienlich ist.

Der Hauptansatz hierfür ist eine gründliche und systematische Analyse der drei Anwendungsgebiete (Desktop-IT, Industrielle Steuernetzwerke und Automotive IT), um festzustellen, welche Eigenschaften dieser Domänen Einfluss auf den computerforensischen Prozess haben. Dies wird erreicht, indem die verwendeten Komponenten, die verwendeten Kommunikationsarchitekturen und die Szenarien, in denen diese verwendet werden, analysiert werden. Die Analyse bekannter Angriffsszenarien in den jeweiligen Domänen rundet diese Untersuchung ab. Um das Verständnis potenzieller forensischer Spuren in den untersuchten Domänen zu verbessern, wird eine forensisch gesteuerte Sicht auf die verwendeten Komponenten eingeführt und verwendet.

Diese Domänenanalyse führt zur Identifizierung von 29 *Einflussfaktoren* (*Impact Factors*) aus der Domäne Industrieller Steuernetze sowie 25 *Einflussfaktoren* aus der Domäne Automotive IT. Diese *Einflussfaktoren* sind Eigenschaften, die den forensischen Prozess in diesen Domänen beeinflussen. Weiterhin werden sie verwendet um die Ähnlichkeiten respektive Unterschiede zwischen den in dieser Arbeit diskutierten Domänen zu beschreiben. Industrielle Steuernetzwerke und Automotive IT teilen viele dieser *Einflussfaktoren.* Die Überschneidungen mit der Desktop-IT Domäne sind geringer, was einen starken Unterschied zeigt.

Während sich bestimmte Werkzeuge und Methoden unterscheiden, können die Aspekte des [KDV15]-Modells für den forensischen Prozess im Allgemeinen auf die untersuchten Domänen angewendet werden. Das [KDV15] -Modell besteht aus *Untersuchungsschritten* (*Investigation Steps*), *Datenarten* (*Data Types* und *Methodenklassen* (*Classes of Methods*). Die Überarbeitung dieser Aspekte im Licht der untersuchten Domänen wird in dieser Arbeit betrachtet.

Bei den *Untersuchungsschritten* verschiebt sich der Schwerpunkt zwischen den Phasen. Da die in den untersuchten Domänen verwendeten Komponenten nur über sehr begrenzte Ressourcen verfügen, ist die Phase *Strategische Vorbereitung* (*Strategic Preparation*) unerlässlich, um eine sinnvolle Menge forensischer Spuren zu erlangen. Darüber hinaus ist die Gesamtstruktur des forensischen Prozesses geringfügig verändert. Für die *Datentypen* ist eine klare Neudefinition der meisten von ihnen notwendig um ihre Rolle im forensischen Prozess zu erfüllen. Zusätzlich wird ein neuer *Datentyp* eingeführt. Zusätzlich werden die *Methodenklassen* neu definiert, um den forensischen Prozess innerhalb der untersuchten Domänen zu verbessern.

Dieses veränderte Prozessmodell wird in einem komplexen Szenario aus der Domäne industrieller Steuernetze angewendet und somit evaluiert.

Der Hauptbeitrag dieser Arbeit ist der Wissenstransfer vor der Desktop IT zu den untersuchten Domänen. Dies erhöht den Reifegrad der Computerforensik in diesen beiden Anwendungsfeldern. In dieser Arbeit werden die in diesen verschiedenen Bereichen verfügbaren forensischen Spuren auf der Grundlage einer wissenschaftlichen und umfassenden Analyse der jeweiligen Bereiche erörtert. Darüber hinaus werden die verfügbaren Methoden zum Erfassen und Untersuchen dieser Spuren aufgezeigt. Auf fehlende Methoden und Werkzeuge wird hingewiesen, um die Schritte zu identifizieren, die erforderlich sind, um die Reife der Computerforensik in diesen Bereichen anzuheben. Dafür werden einige unterstützende Konzepte eingeführt.

Die Forschung wird durch eine Anzahl relevanter Veröffentlichungen und industrielle Forschung unterstützt.

Contents

List of Figures

List of Tables

1. Introduction

Computing units form the backbone of many systems humanity relies on. While the impact of computing units on business processes is obvious, an even greater amount of computing units perform their work invisible to the casual observer. These computer systems embedded into devices are referred to as embedded systems.

The development of using computing units to improve, simplify or even enable certain processes or functions is referred to, depending on domain, as Digitization, Digitalization, Digital Transformation or the use of Information Technology (IT).

While there are certain prime examples how Digitalization transformed the work within countless offices, this PhD thesis discusses computing units usually invisible in their function to the common user of technology. Examples are the various computing units deployed within the automobile industry. Here, these computing units mimic the function of beforehand purely mechanical systems. Calculations substitute for mechanical engineering. Would these computing units only implement well-known basic functions, the user - in this case the driver - might not even be aware of the sophisticated hardware of his vehicle. However, manufacturers used the opportunities provided by digital systems and included additional functionality like diagnosis functions which display warnings locally to the driver if something with one of the locally installed vehicle components seems erroneous. The localized solutions evolved and the car is becoming a part of a network of interconnected cars, which share and receive data to inform each other of potential problems on the road ahead in order to make life of every participant easier.

These systems are nowadays highly complex and hard to engineer. Faults and errors or just unspecified states might cause problems - and in a scenario involving heavy, fast moving objects this entails risk for limb and life.

Besides unintentional disasters, there are those with means and motivation to cause intentional disasters. The rapidly interconnecting world of cars includes new interfaces and attack surfaces for potential attackers as well as a sprawling environment of different components, protocols and responsibilities.

With crashes caused solely by misbehaving computing units becoming a potential occurrence, the need to understand what really caused the behavior of a given computing unit arises.

Forensic science aims at establishing mechanisms to reconstruct the events leading to a specific symptom. This reconstruction has to be in a comprehensible, repeatable manner in order to carry the burden of potential proof. While forensic science has been researched, discussed and used for classical IT (the 'office' side of IT - sometimes referred to as Desktop IT, but relying on extensive networking between various systems today) quite broadly, the application of forensic science to the computing units within automobiles is not yet well-researched and carries some implications and limitations due to the nature of Automotive IT.

Another domain similar to Automotive IT is the domain of Industrial Control Systems (ICS). These automation systems nowadays control physical processes, like manufacturing processes. While these systems started out as mere replacement for mechanical process control, they evolved. This development mimics the development in the Automotive IT domain. As shown in later chapters of this work, the structure of ICS also carries implications and limitations similar to Automotive IT and faulty control logic might cause the same danger for limb and live. Hence, the need for forensic investigation on ICS is the same as for Automotive IT.

The aim of this work is to quench this need for mechanisms, procedures and potential tools to perform forensic investigations within non-traditional IT environments and to show and overcome the implications and limitations in order to show what is needed to bring Automotive IT and ICS to a forensic ready-state.

In order to achieve this goal, various methods are used. The central approach is to adapt methods from the well-established field of (potentially networked) Desktop IT. The forensic model from [KDV15], to which the author of this thesis has contributed, is selected and used as a foundation for the adaptation of the forensic process to the ICS and Automotive IT domains. A methodical investigation into the concepts of this forensic process model and how these concepts can be applied to the domains of Industrial Control Systems and Automotive IT follows.

The principal aspects of the [KDV15]-model are *Investigation Steps*, *Data Types* and *Classes of Methods*. This thesis further expands on the design principles of *Structural Impact* and *Data Streams* used during the creation of this forensic process model. After establishing an understanding of these aspects, this thesis investigates how the ICS and Automotive IT domain impact each of these aspects and how these principal components of the [KDV15]-model have to be altered in order to serve as foundation for a comprehensive forensic process within these domains. What constitutes a comprehensive forensic process is determined by taking benefits and shortfalls of other forensic models, criteria for evidence usability in and outside of court settings, and data protection related factors into consideration and establishing a set of eight *Forensic Process Model Criteria*.

1.1 Aim of This Thesis

The goal of this thesis is to adapt knowledge from the domain of (potentially networked) Desktop IT to the ICS and Automotive IT domains. This knowledge about forensic investigations consists of concepts, procedures, methods and tools. In order to apply this knowledge into these different domains alterations and adaptations are necessary.

The knowledge about forensic investigations is exemplified by a revised version of the forensic process model developed for [BSI11] in the iteration presented in [KDV15]. This foundation is enriched by additional insight into the creation process of this forensic process model which includes the concepts of *Structural Impact* and *Data Streams*. If and how these established procedures from the field of Desktop IT can be adapted to the ICS and Automotive IT domains is a complex task which can be subdivided into a range of different research questions:

- **Research Question 1**: What technical or organizational properties do the domains of (potentially networked) Desktop IT, ICS and Automotive IT possess that impact forensics?
- **Research Question 2**: How do these properties impact forensic procedures and methods discussed in the [KDV15]-model?
- **Research Question 3**: How do methods and procedures from the [KDV15]-model (representing computer forensics in the Desktop IT domain) have to be adapted or altered to be used in the ICS and Automotive IT domains?

These research questions cover the adaptation of methods, procedures and the [KDV15]-model for the forensic process from the classical IT domain to the ICS and Automotive IT domains.

They do not cover how accurately the sensors used in the three domains capture the properties of their environment. The impact of error, loss and uncertainty associated with the processing of inputs is described in Section 2.1.5. For further detailed considerations on this topic, see the dissertation thesis of Stefan Kiltz which explores this topic in great detail ([Kil20]).

1.2 Structure of This Thesis

The **first chapter** gives an introduction into the aims of this thesis and identifies the various research questions required to reach these aims.

The **second chapter** introduces the fundamentals for this thesis. An overview on the field of forensic science in general is given. This is followed by a discussion on computer forensics defining the scope and aim of the computer forensic process in order to identify the various aspects of importance during a computer forensic investigation. This is followed by an introduction into the three computational domains (Desktop IT, Industrial Control Systems and Automotive IT) in order to define the application field covered by this thesis. After this, a discussion of what constitutes a good forensic process model follows. This discussion is based on the various aspects of importance during a computer investigation which have been identified earlier in this chapter as well as the discussion of a set of selected forensic process models designed to address the topic of computer forensics in the Desktop IT domain. This discussion leads to criteria for a comprehensive forensic process model (*Process Model Criteria*). The chapter is finished by a review of the current state of computer forensics in the ICS and Automotive IT domains.

The **third chapter** discusses the development of the [KDV15]-model for the forensic process and how it addresses the various aspects of Computer forensics with a focus towards the (potentially networked) Desktop IT domain. By discussing the design process of this forensic model, the concepts of *Structural Impact* and *Data Streams* are explored. This forensic process model is then used as a foundation for the remainder of this thesis.

The second part of this chapter discusses the steps necessary to answer the research questions. Here, a concept for a systematic examination into the nature of forensic investigations in the three covered domains is formulated. This approach compromises domain analysis, revisiting of current procedures and evaluation of the altered process.

The **fourth chapter** covers the domain analysis as discussed in the preceding chapter. Here, specific properties of the various domains are identified under the light of the criteria and procedures introduced in the second chapter. Their influence on forensic investigations is explored.

The **fifth chapter** discusses the potential alterations to the forensic process as known from the field of classical Desktop-IT. This also includes the introduction of novel concepts to better address the specifics of the different domains.

The **sixth chapter** evaluates the alterations to forensic concepts as well as the new concepts by putting them up against a complex case study.

The **seventh chapter** concludes this work by summarizing the findings of this thesis and the alterations to the forensic process. In addition, remaining open questions and demands are identified and discussed.

1.3 Contributions Made to the Field of Computer Forensics During the Course of the Thesis

This thesis discusses the various aspects which influenced the creation of the forensic process model as described in [KDV15] and then adapts and enhances this model in order to increase the maturity of computer forensics in the ICS and Automotive IT domains.

The creation of the [KDV15]-model and the various adaptations and enhancements applied to the model during the course of this thesis require a deep understanding of the field of Forensics in general and computer forensics in particular. This understanding is provided by Section 2.1.1. This leads to the establishment of two different *Investigative Contexts* in order to distinguish between the different aims a forensic process can carry. This is due to the fact that these aims carry different implications for the forensic process (for example in terms of admissibility of evidence to court, adherence to restrictions or simply in the nature of the evidence useful to investigate a given suspicion). Additional input for the creation and enhancement of the [KDV15]-model is derived from various forensic process models and the desire to unify their advantages.

This thesis extends the considerations during the creation of the [KDV15]-model by including recently published forensic process models, updating the list of these advantages in Section 2.3.2. Additionally, these advantages are formalized by establishing four *Process Challenges* and defining eight *Forensic Process Model Criteria* for a comprehensive forensic process model (see Section 2.3.2.4). Furthermore, the development of the [KDV15]-model for the forensic process is explored in detail. This leads to an understanding how the desire to unify advantages of the reviewed forensic process models influenced the creation of the [KDV15]-model and how these advantages are addressed. The formalization of the eight *Forensic Process Model Criteria* allows for a discussion on how well the [KDV15]-model aligns to these criteria.

The core concepts of the forensic process model as described in [KDV15] are expanded by the inclusion of additional publications (a notable portion of them authored by the author of this thesis) and the inclusion of the concepts of *Structural Impact* and *Data Streams* which are integral to addressing the *Forensic Process Model Criteria* and provide some useful concepts for the exploration of the embedded domains considered in the scope of this thesis.

A comparative, scientific analysis of the ICS and Automotive IT domains is performed in order to identify potential sources of forensic evidence. This is achieved by analyzing employed components, communication architectures and hierarchies while considering the concepts of the [KDV15]-model. This systematic analysis establishes a forensic-driven view on the employed components, architectures and hierarchies. This view serves as a foundation for the identification or proposal of various methods in order to gather and investigate these pieces of potential evidence. Analyzing the scenarios in which these computer systems are employed while taking known attack scenarios into consideration shows additional aspects of these domains. The analysis of these domains leads to the identification of 29 *Influence Factors* from the ICS domain and 25 *Influence Factors* from the Automotive IT domain which impact the forensic process in these domains. These *Influence Factors* present the answer to **Research Question 1** and hence are the foundation for the identification of six *Forensic Process Consequences* from the ICS domain and six *Forensic Process Consequences* which address **Research Question 2** and form the foundation of the adaptation of the forensic process to these domains.

These *IFs* are also used to evaluate how similar the three domains discussed in this thesis actually are. The ICS and Automotive IT domain share 22 *IFs*, slightly differ in five and have five respective 3 unique *IFs*. Of the totally of all *IFs* only seven are completely and two partly shared with the Desktop IT domain. This denotes a strong similarity between the ICS and Automotive IT domain and a notable difference to the Desktop IT domain.

Combining the deep understanding of the [KDV15]-model achieved by investigating and discussing its development history and the specifics of the ICS and Automotive IT domains allows for the adaptation and enhancement of the [KDV15]-model to better represent these two emergent domains without sacrificing its usage when addressing Desktop IT.

A visual reference for this conduct can be found in Figure 7.1, which can be used as a guide on how to read this thesis.

While specific tools and methods differ, the aspects of the [KDV15]-model for the forensic process can, in general, be applied to the ICS and Automotive IT domains. The [KDV15]-model consists of *Investigation Steps*, *Data Types* and *Classes of Methods*. The investigation and alteration of these aspects in the light of the investigated domains is presented in this thesis. The specific results answer **Research Question 3**.

For the *Investigation Steps*, the emphasis of various phases shifts. Since the components employed in the investigated domains have very limited resources, the strategic preparation (SP) phase is essential in order to achieve a meaningful amount of forensic traces. For the *Data Types*, most of them need a clearer definition in order to fulfill their role in supporting the forensic process by attaching methods for gathering or analyzing various types of data. In addition, one new data type is introduced. Additionally, the *Classes of Methods* are redefined in order to improve the forensic process within the investigated domains. Additionally, the concepts of *Structural Impact* and *Data Streams* are expanded. These concepts where not explicitly mentioned in [KDV15] but influenced the design process of the forensic process model. In the course of this thesis, the relevance of these two concepts becomes clear.

This altered process model is put to test in one complex test case in the ICS domain. Here, domain-specific solutions for the various phases of the forensic process are used. Additionally, considerations on how achieve the steps necessary to achieve forensic readiness in these two domains are presented.

The transfer of knowledge from the classical IT domain to the ICS and Automotive IT domains increases the maturity of computer forensics in these two distinct domains. This thesis discusses the forensic traces available in these different domains based on scientific and comprehensive analysis of the given domains. In addition, the methods available to acquire and investigate these traces are shown. Missing methods and tools are pointed out in order to identify the steps necessary to increase the maturity of computer forensics in these domains even further.

The research is supported by a great number of relevant publications (some of them as primary author, others as co-author) and industrial research discussing various aspects of computer forensics in these three domains. These include the development and refinement of the [KDV15]-model ([**KHA^{+}09**], [**AKD09**], [**ACKD12**] and [**ADKK14**]), the identification of specific characteristics of the ICS and Automotive IT domains which impact the forensic process as well as a forensic-driven view on the employed components, architectures and hierarchies ([**HAK^{+}17**], [**ALKD17**], [**ALK^{+}18**], [**AHKD19**], [**eSMP^{+}20**], [**GGAW20**] and [**AHNH20**]), the identification and establishment of methods for gathering and investigating forensic traces in these domains ([**HAK^{+}17**], [**ALKD17**], [**ALK^{+}18**], [**AHKD19**], [**AHNH20**] and [**AH20**]) as well as adjacent topics (like identifying the possibility to perform forensic investigations on encrypted communication streams as presented in [**KAD16**]).

1.4 Relevant Publications of the Author Published During the Course of This Thesis

This research is conducted while working on relevant research activities with various collaborators and has spawned a broad range of relevant, peer-reviewed publications which show some of the problems and answers presented in this thesis. Please take note that any publication of the author of this thesis (either as primary author or as co-author) will be **highlighted in bold** during the course of this thesis. A short overview on these publications - and their role within the research done during the course of this thesis - is presented here:

2020

- **A Simulated Steam Turbine Generator subsystem for Research and Training** ([**AH20**])
  R. Altschaffel and M. Hildebrandt
  This paper discusses the creation of a fully virtualized steam turbine governing system in order to perform training and research on incident detection and ICS forensics.

- **Nuclear Power Plant in a Box** ([**AHeS+20**])
  R. Altschaffel and T. Holczer and R. A. Busquim e Silva and P. Gyorgy and M. Hildebrandt and M. Hewes
  This paper discusses the creation of a fully virtualized nuclear power plant in order to perform training and research on incident response, incident recovery and ICS forensics. A power plant is chosen as an example of a complex ICS architecture and is used in this thesis during the case study.

- **The Nuclear SIEM** ([**AHNH20**])
  R. Altschaffel and T. Holczer and C. Neal and M. Hildebrandt
  This paper discusses the challenges of making ICS in nuclear power plants forensic-ready with regards to cyber attacks. It discusses the potential forensic traces, the means to gather them, the requirements for placing the sensors and the storage for these traces. It also discusses potential security-problems caused by the inclusion of such a system.

- **Threat Analysis of Steganographic and Covert Communication in Nuclear I&C Systems** ([HAL+20])
  M. Hildebrandt and **R. Altschaffel** and K. Lamshöft and M. Lange and M. Szemkus and T. Neubert and C. Vielhauer and Y. Ding and J. Dittmann
  This paper discusses the potential use of steganographic communication during an IT attack and the detection and investigation of such a communication. Besides this, this paper provides a summary on the network architecture and its implication on network traffic based on the exemplary implementation for a security-aware network architecture in [NSS11].

- **Blue team support for EPS related cybersecurity readiness** ([**GGAW20**])
  D. Gupta and D. Govindaraj and **R. Altschaffel** and K. Waedt
  This paper discusses the need of the pre-incident preparation in cyber security in order to be able to detect, investigate and counter potential cyber security incidents. Here, computer forensics is seen as a potential tool used during incident response. An approach seen in various process models (see Section 2.3.1.3 and Section 2.3.1.5).

- **Understanding Nuclear Cyber Security Measures, Risks and Consequences: from Tank Levels to Plant Processes** ([**eSMP+20**])
  R. A. Busquim e Silva and R. P. Marques and J. R. C. Piqueira and P. Smith and M. Hewes and S. Purvis and J. Li and **R. Altschaffel**
  This paper discusses the need to have an understanding of the overall cyber-physical system in order to perform a meaningful incident response or a forensic investigation. Only by having the complete picture of the control system and the controlled physical process in mind, structural impact can be judged. This is especially important for the decision process within the operational preparation (see Section 5.3.1).

2019

- **Digital Forensics in Industrial Control Systems ([AHKD19])**
  A. Altschaffel and M. Hildebrandt and S. Kiltz and J. Dittmann
  This paper discusses the properties of ICS relevant for computer forensics. It proposes a more finely graduated understanding of the various levels of components in the ICS context based on their properties influencing the forensic process. In addition, the necessity for alterations to the [KDV15] model is discussed. After establishing that the current data types do not represent all the data present in ICS environments in a manner useful for forensic processes, the data types are reorganized and an additional data type is added. In addition, the concept of a criticality map is introduced (see Section 5.3.1).

2018

- **A Survey on Open Forensics in Embedded Systems of Systems ([ALK+18])**
  R. Altschaffel and K. Lamshöft and S. Kiltz and M. Hildebrandt and J. Dittmann
  This journal paper describes the computer forensic process in ICS environments. It shows different attack patterns and the potential forensics traces caused by these patterns in the different data streams. Furthermore, it discusses how these environments impact the overall forensic process and which alterations to the process are necessary. A survey on tools usable during the various phases of the computer forensic process is performed.

- **Exploring the processing of personal data in modern vehicles - A Proposal of a testbed for explorative research to achieve transparency for privacy and security ([KAH+18])**
  A. Koch and **R. Altschaffel** and M. Hildebrandt and S. Kiltz and J. Dittmann
  This paper describes a test setup used to identify potential data sources within Automotive IT environments in manufacturer-independent manner. This approach allows for a better understanding and increased transparency of Automotive IT systems for independent forensic investigators.

2017

- **A Survey on Open Automotive Forensics ([ALKD17])**
  R. Altschaffel and K. Lamshöft and S. Kiltz and J. Dittmann
  This paper describes the components of Automotive IT from the viewpoint of computer forensics. Here, a survey on the various available open tools usable during forensic investigations into the components of Automotive IT is performed. These tools are arranged to fit the computer forensic process described in [KDV15] in order to identify gaps in the current process.

- **Exploring the possibility of forensic investigations on steam turbine governing systems ([HAK+17])**
  R. Altschaffel and K. Lamshöft and S. Kiltz and J. Dittmann
  This paper discusses the possibility of conducting computer forensic investigations in steam turbine control systems. To achieve this, a generalized model of a steam turbine control system from the viewpoint of computer forensics is created. Various forensic traces are identified and considerations on how these traces can be gathered and analyzed during the computer forensic process are discussed.

- **Adapting Organic Computing Architectures to an Automotive Environment to Increase Safety & Security** ([LAD17])
  K. Lamshöft and **R. Altschaffel** and J. Dittmann
  This paper discusses various attack scenarios and adversarial models in the context of Automotive IT. This serves as a foundation to discuss how adaptive Automotive IT has to be designed in order to offer resilience against these attacks.

2016

- *(German)* **Tendenzen zum Profiling von verschlüsselten Netzwerkverkehren - Möglichkeiten und Grenzen** ([KAD16])
  Tendencies for profiling of encrypted network traffic - possibilities and limits
  C. Krätzer and **R. Altschaffel** and J. Dittmann
  This paper discusses the possibility to perform forensic investigation on encrypted network traffic by analyzing heuristic data in order to identify the use case performed within the encrypted session. It also discusses limits to this approach.

2015

- **Evaluation of impacts of IT-incidents on automotive safety with regard to supporting reaction strategies for the driver** ([KAH+15])
  S. Kuhlmann and **R. Altschaffel** and T. Hoppe and J. Dittmann and C. Neubüser
  This paper discusses strategies to inform the driver of a vehicle of a suspected security incident in order to perform incident response or forensic investigations. Hence, it deals with one of the specific aspects present when dealing with Automotive IT.

2014

- **A Hierarchical Model for the Description of Internet-Based Communication** ([ADKK14])
  R. Altschaffel and J. Dittmann and C. Krätzer and S. Kiltz
  This work deals with the description of internet-based communication and hence serves as a foundation for the discussion and adaptation of the various data types (see Section 5.3.2).

2013

- **Statistical Pattern Recognition Based Content Analysis on Encrypted Network: Traffic for the TeamViewer Application** ([ACK+13])
  R. Altschaffel and R. Clausing and C. Krätzer and T. Hoppe and S. Kiltz and J. Dittmann
  This work discusses the possibility for forensic investigation on encrypted network streams by showing the possibility to still extract useful forensic evidence which goes beyond the raw meta data. It is demonstrated that observation of the use case performed within the encrypted communication is still possible.

2012

- *(German)* **Erste Betrachtung einer Metrik für Methoden der IT-Forensik ([ACKD12])**
  First considerations for a metric to judge methods usable in Computer Forensics
  R. Altschaffel and R. Clausing and S. Kiltz and J. Dittmann
  This paper describes the first considerations for the establishment of a metric to judge the usefulness of certain forensic methods and tools during a given investigation. It establishes criteria to determine the cost of employing an using a certain forensic methods as well as to judge the quantity and quality of the additional traces made available by employing this method. This work describes an earlier version of the process model published in [KDV15] and uses the various aspects of this process model for the establishment of criteria. It contributes a discussion on the concept of Structural Impact during the forensic process.

2010

- **A transparent Bridge for forensic sound network traffic data acquisition ([KHAD10])**
  S. Kilz and M. Hildebrandt and **R. Altschaffel** and J. Dittmann
  This paper utilizes an early version of the forensic process model published in [KDV15] in order to establish a very basic software design for a generic forensic application. This software design is then used to create the Linux Forensic Transparent Bridge. This tool allows the capture of raw network communication data while maintaining authenticity and integrity of the recorded data and of the recording protocol. This paper therefore describes how a forensic process model can translate into requirements for dedicated forensic tools which maximize the evidential value of the forensic traces gathered or processed using the respective tool.

2009

- **From the Computer Incident Taxonomy to a Computer Forensic Examination Taxonomy ([AKD09])**
  R. Altschaffel and S. Kiltz and J. Dittmann
  This work describes an earlier version of the forensic process model published in [KDV15] and uses it as a basis for a Forensic Examination Taxonomy (FET). This FET is a taxonomy designed to describe the results of a forensic investigation in a formalized manner. This formalization is similar to the CERT-taxonomy presented in [HL98] which is usually used to describe security incidents. The main contribution to the development of the [KDV15]-model from this publication is the discussion of the scope of the forensic process.

- *(German)* **Sicherstellung von gelöschtem Schadcode anhand von RAM-Analysen und Filecarving mit Hilfe eines forensischen Datenmodells ([KHA$^+$09])**
  Gathering of deleted malware code by using RAM analyzes and file carving and applying a model for forensic data
  S. Kiltz and M. Hildebrandt and **R. Altschaffel** and J. Dittmann and C. Vielhauer and C. Schulz
  This work uses an earlier version of the forensic process model published in [KDV15]. This model already contains the three major elements of the forensic process model. It is used in this publication to ensure the integrity and authenticity of evidence extracted from a memory dump of a Windows-based system. Furthermore, approaches for the long term preservation of forensic data are discussed.

2. Fundamentals and Derived Methodology

This chapter presents the fundamentals necessary for understanding the challenges and approaches in adapting the [KDV15]-model to the ICS and Automotive IT domains while enhancing certain aspects of the model. As the primary aim of this thesis is the establishment of good and comprehensive forensic procedures, an understanding of what good and comprehensive forensic procedures entail is necessary.

Such an understanding can only be achieved by providing an overview on the nature and aims of forensics in general. This is done in Section 2.1.1. This overview is followed by a discussion of the term *digital forensics* in Section 2.1. This establishes the scope of digital forensics as discussed in this thesis. After this definition is provided, the various aspects relevant for forensics are introduced. These include the activities performed during forensic investigations (see Section 2.1.3), Evidence dynamics in digital forensics (see Section 2.1.4) and the topic of Error, loss and uncertainty (see Section 2.1.5). Legal aspects - constraints due to privacy protection (see Section 2.1.6) and legal requirements for the admissibility of evidence (see Section 2.1.7) - follow. A discussion of these legal requirements leads to the establishment of eight *Admissibility Factors*).

All these aspects are necessary for understanding and evaluating the scope of the forensic process models discussed later in this chapter in Section 2.3.1. This review of forensic process models is preceded by an introduction to the relevant domains discussed in this thesis (see Section 2.2). The definitions established in this section describe the computer systems taken into account in this thesis. This introduction aims at establishing an understanding of these domains as they are seen during the scope of this work. A more detailed view on these domains will be necessary later in this work (see Chapter 4), after the following chapter defines the criteria under which this closer look has to take place (see Section 3.2).

This chapter continues with a discussion of selected computer forensic process models (see Section 2.3.1) in order to gain an understanding of what makes a good forensic process model. These process models originate from the domain of classic Desktop IT and are selected based on historic significance or the specific aspects they address. This review aims at understanding the scope of the computer forensic process and its evolution over the last decades. This section gives an overview on the aspects and scopes addressed by these process models. These models are discussed and their advantages and core concepts taken into account for the establishment of eight *Forensic Process Model Criteria* for a comprehensive forensic process model in Section 2.3.2.4.

This chapter ends with a survey on the current state of forensic investigations in the ICS and Automotive IT domains (see Section 2.3.3 and Section 2.3.4 respectively).

2.1 Digital Forensics

The following section deals with the topic of digital forensics. At first, the term of forensics in general is introduced (see Section 2.1.1). After this, the term and scope of forensics as used and discussed during this thesis is defined (see Section 2.1.2). Afterwards, a short introduction into the activities during the forensic process is given (see Section 2.1.3) before some of the problems and constraints for digital forensics are discussed (**Evidence Dynamics**, **Error, Uncertainty and Loss** and **legal constraints** - see Section 2.1.4, Section 2.1.5 and Section 2.1.6 respectively). Lastly, the legal guidelines for the admissibility of forensic evidence are discussed (see Section 2.1.7). This all prepares for the discussion of forensic process models in the various domains (see Section 2.3.1).

2.1.1 Forensics

It is often of interest to understand why an event occurs. Various disciplines of science construct and verify models and theorems to describe, to name only a few examples, physical processes or human behavior. This enables the prediction of certain events as well as an understanding of why certain events occurred.

Forensics also aims at understanding why certain events occurred. The term was in its original sense used to describe the judicial processes in the ancient roman *forum* where accuser and defender brought forth arguments and evidence for guilt or innocence. It is reasonable to conclude, that this process aims at performing a reconstruction of a crime in a reliable fashion. The definition of Crime reconstruction, as found in [Arn17], supports this conclusion:

> „Crime reconstruction is the determination of the actions and events surrounding the commission of a crime.“

Such reconstruction would usually entail the answering of the five basic W-questions (Who, Where, What, When, Why). Often, an additional question (How) is added. This combination is often referred to as **5WH** (or **5W1H**). The meaning of these questions in the context of a criminal investigation is discussed in [Ste09]:

> *„Who: Persons involved in the investigation, including suspects, witnesses, and victims Where: The location of the crime and other relevant locations What: Description of the facts of the crime in question When: The time of the crime and other related events Why: The motivation for the crime and why it happened at a given time How: How the crime was committed.“*

As shown before, answering these questions has to rely on certain pieces of **evidence**. Evidence is defined, according to [NN11], as:

> *„Evidence is any of the material items or assertions of fact that may be submitted to a competent tribunal as a means of ascertaining the truth of any alleged matter of fact under investigation before it [..]“*

In the case of material items, the evidence could take the form of documentary evidence (e.g. a contract if the investigation is concerned with the breach of said contract) or real evidence (e.g. a knife left at the scene of a crime). Assertions of fact include testimonial evidence (e.g. the testimony of a witness) or demonstrative evidence (e.g. a map visualizing the testimony of a witness).

Obviously, the quality of the answers to the investigative questions corresponds to the quality of the evidence. Hence, a competent tribunal will aim at being able to judge the quality of the evidence in question. This quality includes the provenance, the explanatory power and the credibility of the given evidence.

A major factor of evidence quality is the question of how the evidence is handled and preserved. Should the evidence not be handled with the specific care, its value degrades. This is quite obvious for evidence which decays over time - foot prints in fresh snow are an illustrative example. Here, the evidence will fade if not preserved correctly. While this fading would negate the foot prints usefulness as evidence, mishandling the trace would also have a negative impact. The trace could be destroyed or altered in a way which allows no further use for event reconstruction. In this example, someone could step on the foot print and destroy it.

This aspect is a main consideration during forensic investigation and [Arn17] refers to it as **Evidence Dynamics** using the following definition of this term:

> *„Evidence dynamics refers to any influence that adds, changes, relocates, obscures, contaminates, or obliterates evidence, regardless of intent.“*

Evidence dynamic has a negative influence on the explanatory power of evidence. Since a competent tribunal will require knowledge on the question if the evidence before them is to be trusted and why it should be trusted, not being able to determine if there is evidence dynamics affecting this piece of evidence also has a negative effect on the explanatory power of evidence.

Considerations on how to achieve optimal explanatory power from given evidence has fueled research. This usually aims at understanding the provenance of the evidence in question as well as reducing evidence dynamics.

Since ancient times, the procedures of acquiring, weighting and bringing forth evidence to courts has become more and more scientific. Hence, the process of collecting evidence, handling it and weighing its conclusiveness has become scientific in nature - and is referred to as **forensic science** in modern times. Forensic science is able to give estimations of the conclusiveness of certain traces. This estimations takes into account factors like the provenance of a piece of evidence, its handling and the accuracy of conclusions or predictions delivered by the procedures applied to it (see [IR01] for more information on this topic). Forensic science is defined by [Arn17] as:

> *„The application of scientific methods to establish factual answers to legal problems.“*

Today, forensic science encompasses a broad range of various disciplines, including such varied topics as Forensic Accounting, Forensic Pathology, Forensic Seismology or Forensic DNA Analysis.

While all these forensic sciences are of interest to court proceedings, this thesis focuses on the field of **digital forensics**. Hence, an introduction to this field is given in the following sections. This introduction defines the field of digital forensics and its borders and the factors which dictate the conclusiveness of evidence gathered during investigations in the digital domain.

2.1.2 Definition of Digital Forensics

Digital forensics is a forensic science that deals with evidence derived from digital sources. A descriptive definition of digital forensics was published by the Digital Forensics Research Conference in 2001 ([DFR01]):

> *„The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.“*

This definition does not state explicitly what kind of digital sources are considered. Hence, the definition covers all kinds of digital sources including, e.g. hard drives in Desktop IT Systems, working memory in any kind of computing unit or a digital communication network inside an Industrial Control System (ICS). Obviously, available procedures and research is focused on the types of digital sources in high demand - the ones employed in Desktop IT systems.

The given definition includes unauthorized actions and criminal behavior. Hence, it aims at supporting court proceedings and the investigation of unauthorized access. This is in line with the various considerations detailed during the preceding section (see Section 2.1.1).

After discussing what is included in this definition of forensic investigations, it is prudent to discuss what is not included by this definition of forensic investigations. For example, it is notable, that many legal codes differentiate between criminal law and private law. Criminal law handles crimes while private law handles contract or policy violations ([Arn17], page 5). This would, in fact, not be criminal behavior and hence not be covered by the definition of digital forensics from [DFR01].

Additionally, unauthorized actions could also be caused by a variety of accidents, malfunctions or user errors. Often it is not clear if a given unauthorized action is caused intentionally by an attacker or unintentionally by an error, mistake, accident or malfunction. In the first case, criminal (or private) law might apply to the attack in question. In the second case, criminal (or private) law usually does not apply, with the potential exception of warranty or defects liability.

However, even if there is no legal necessity for an event reconstruction, such a reconstruction might still be useful for the systems administrators, operators or owners in order to fix problems with the systems or to stop mistakes from being repeated. Again, the interested parties would obviously prefer a reconstruction based on solid evidence - such evidence would have limited (and known) evidence dynamic, a clearly identifiable origin and would have been handled in a scientific manner. Hence, it seems useful to apply the principles of forensic science to these cases.

This also helps to cover all investigations in which an error was suspected as the cause of unwanted behavior and in which an attack on the system was finally identified as the perpetrator. This holds especially true if one considers the case where an unauthorized action seems like a honest mistake at first but seems to be caused by a deliberate action after an investigation is concluded. Here, lost care in handling the evidence cannot be regained.

This viewpoint is supported by the definition of digital forensics given by [Bis18]:

> *„Digital forensics is the science of identifying and analyzing entities, states, and state transitions of events that have occurred or are occurring.“*

The author further clarifies the meaning of this definition:

> *„The distinction between this and the legal notion of forensics is important. Digital forensics, also called* computer forensics*, may in fact be a component of a legal forensic analysis, in which case the computer analysts must acquire the information and perform the analysis in such a way that they meet the appropriate legal requirements. More commonly, digital forensics is used to figure out what caused an anomaly or to understand the nature of an attack, including how the attackers entered the system, what they did there, and how the defenses failed without following legal strictures. This complicates matters should the analysis uncover information requiring legal intervention.“*

While this definition includes any unwanted behavior introduced not by an attacker but unintentionally by accident, error or malfunction, the level of detail of the initial definition is missing.

Hence, a redefinition of the term **digital forensics** to better reflect the scope of this thesis seems appropriate. This redefinition should not only include any unwanted behavior introduced by an attacker but also those introduced unintentionally. Private law should be included as well as criminal law since they do not differ in their need for reliable evidence. Since private and criminal law are both judged in courts, the neutral wording of *subject to court cases* is chosen. In addition, the wording is slightly altered in order to increase the clarity of the used terms. Hence, during the course of this thesis, the following definition is introduced and henceforth used:

> **Digital Forensics (DF)**:
> *The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be subject to potential court cases, or disruptive to intended systems operations.*

From the viewpoint of court proceedings, it is noteworthy, that the field of digital forensics investigates two very different types of events. These two types of events differ in nature and affect the way an investigation takes place. For the scope of this thesis, these two types of events will be referred to as **Investigative Contexts** and the following definitions are introduced:

> **Investigative Context 1 (IC1)**:
> *Investigations into unwanted behavior of the system.*
>
> **Investigative Context 2 (IC2)**:
> *Investigations into behavior which has been carried out with the use of computer systems and is subject to a potential court case.*

In the case of **IC1**, the system itself is affected by an incident, while in the case of **IC2**, the system is merely a tool used for unwanted conduct.

For **IC1**, this encompasses manipulation or siphoning of data stored or processed within the system or the degradation of the systems capabilities in any way (for example by disabling components or by blocking resources). It is important to note that this not only includes targeted malicious attacks but also honest error, accidents and malfunctions. On one hand, it is not always clear if any unwanted behavior of the system is the consequence of an attack or an error and on the other hand a forensic investigation might also be used to identify technical problems in order to mitigate them.

In **IC2**, the system is merely used in its original function - to process data. Here the system works as intended but the intention of the user is malicious and potentially criminal. This investigated behavior might entail the bookkeeping of criminal enterprises, communications to foster the trade of illicit goods or simply writing ransom notes. An interesting example would be the creation of malicious software (malware). If the fact whether a certain system was used to create a given malware is investigated this would also fall under **IC2**.

The fundamental difference between **IC1** and **IC2** is the aim of the forensic process. In **IC1** this is always to figure out why a computer system behaved in a certain way, in the case of **IC2** this is always to figure out how some person (or group of persons) used a computer system towards fostering illicit means.

A practical example can be found in the forensic work done by law enforcement agencies. Discussions with German law enforcement personal showed that the forensic investigators differentiate between two different aspects in their work. These are *cybercrime in a narrow meaning* (*German: Cybercrime im engeren Sinne* - akin to **IC1**) and *crime performed using the internet* (German: 'Tatmittel Internet' - akin to **IC2**). In the first case, computer systems are attacked. In the second case, computer systems are used as a tool for criminal behavior. This difference is also visible in the German crime statistics from 2018 [BKA18]. In this statistic, *Computer fraud* would be an example for a case in **IC2** as it represents fraud conducted using a computer. *Data manipulation*, *computer sabotage* and *theft of data* in contrast are examples for **IC1**.

These definitions can be used as a foundation to discuss the nature of attacks performed in the scope of computer systems as described by [Sch04] (page 15):

> *„**THE UNCHANGING NATURE OF ATTACKS***
> *If you strip away the technological buzzwords and graphical user interfaces, cyberspace isn't all that different from its flesh-and-blood, bricks-and-mortar, atoms-not-bits, real-world counterpart. Like the physical world, people populate it. These people interact with others, form complex social and business relationships, live and die. Cyberspace has communities, large and small. Cyberspace is filled with commerce. There are agreements and contracts, disagreements and torts. And the threats in the digital world mirror the threats in the physical world. If embezzlement is a threat, then digital embezzlement is also a threat. If physical banks are robbed, then digital banks will be robbed. Invasion of privacy is the same problem whether the invasion takes the form of photographer with a telephoto lens or a hacker who can eavesdrop on private chat sessions. Cyberspace crime includes everything you'd expect from the physical world: theft, racketeering, vandalism, voyeurism, exploitation, extortion, con games, fraud. [...]"*

Here, various scenarios which could be examined during a forensic investigation are mentioned. These scenarios describe situations targeting computer systems (**IC1**) or those using computer systems for illicit means (**IC2**).

In the case of a (digital) bank robbing, one would be interested in how the (digital) bank was robbed and one would perform the forensic investigation in **IC1**. In the case of (digital) embezzlement, however, the working of the bank itself is of secondary interest - here the (digital) financial records are of importance. This would represent **IC2**, especially if there is no reason to believe that these records were falsified. Hence, these different investigative scenarios put a focus on different kinds of traces usable as evidence for a forensic investigation (as discussed later on in Section 2.3.1).

While all forensic investigations will aim at achieving a reliable event reconstruction (and hence have to address the problems of **Evidence Dynamics** and **Error, Uncertainty and Loss** - see Section 2.1.4 and Section 2.1.5 respectively) and must follow legal constraints (see Section 2.1.6), some also entail stricter, legal guidelines for the forensic process. In **IC2**, some behavior worthy of judicial attention triggering a forensic investigation is inherently assumed. Hence, the investigation should follow forensic principles in each case.

Hence, the legal guidelines for the admissibility of forensic evidence must be adhered to (see Section 2.1.7). In **IC1**, this is not necessarily the case. Here an investigation into unwanted system behavior does not necessarily start with certainty in regards to if this behavior was in fact caused by an attack on the security of the system - it might also be an accident, a malfunction or a user error. Also, the investigation might not aim at filling legal charges but at only identifying what caused the problem and how to fix it. Here, strict adherence to the legal guidelines for the admissibility of forensic evidence is not strictly required but might still be useful in order to improve the explanatory power of the available evidence.

2.1.3 Activities in Digital Forensics

The definitions of digital forensics give an overview on the different activities during the forensic process. These activities are described in [DFR01]) as:

> *„[...] preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence [...]"*

Various models exist to describe these activities and formulate a process in which these activities take place. These models aim at maximizing the credibility and explanatory power of the available evidence. This can be done by adhering to scientific standards. An investigation performed in such a way is a forensically sound investigation, as defined by [Arn17]:

> *„An investigation is forensically sound if it adheres to established digital forensics principles, standards, and processes.“*

Forensic process models are used to ensure that the investigation follows these established principles, standards and processes. These specific process models are discussed later in this chapter (see Section 2.3.1).

Generally, these models are focused on the types of digital sources in high demand - Desktop systems.

2.1.4 Evidence Dynamics in Digital Forensics

Before going into the details of the various potential digital sources, focus is given to the factors which dictate the conclusiveness of evidence for all potential digital sources. One factor mentioned before is evidence dynamics. In digital forensics evidence dynamics can be understood as relating to **integrity**. Integrity is defined by [Bis18] as:

> *„Integrity refers to the trustworthiness of data or resources, and it is usually phrased in terms of preventing improper or unauthorized changes. Integrity includes data integrity (the content of the information) and origin integrity (the source of the data, often called authentication). The source of the information may bear on its accuracy and credibility and on the trust that people place in the information.“*

The term **authenticity** can be understood as the *integrity of information about the origin* (or **origin integrity**). For a clearer distinction, the term **authenticity** will be used when referring to **origin integrity** in the scope of this thesis. However, the question of what constitutes the origin of a given piece of data remains.

An illustrative example can be seen in Figure 2.1. In this example, a communication between *Participant A* and *Participant B* is observed by *Participant C*. The forensic trace in question would be a transcript of the communication between *Participant A* and *Participant B* provided by *Participant C*. Here, authenticity could refer to different aspects. At first, authenticity could be used to describe that *Participant C* is indeed *Participant C*. In this case, the authenticity of *Participant C* would be given. Since *Participant C* is the origin of the transcript, one could argue that **origin authenticity** is established in regards to the transcript.

While this is of interest during a forensic investigation, it is also of interest to know if the entities in the observed communication are indeed *Participant A* and *Participant B*. This aspect can be described as **entity authenticity**. This **entity authenticity** for *Participant A* would be given if it could be assured that *Participant A* in the transcript of the communication corresponds to *Participant A*. The same relationship exists with regard to *Participant B*.

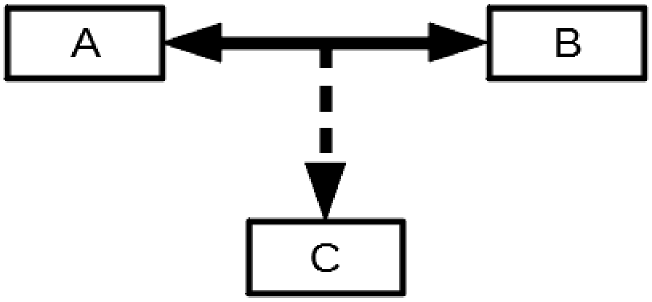


Figure 2.1: First example to illustrate *Origin Authenticity* and *Entity Authenticity*

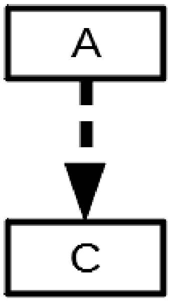


Figure 2.2: Second example to illustrate *Origin Authenticity* and *Entity Authenticity*

A second example can be seen in Figure 2.2. In this example, *Participant C* observes *Participant A* performing a certain action. In this case, *Participant C* could be a witness and *Participant A* a potential perpetrator. If *Participant C* gives testimony about the perceived actions of *Participant A*, one question would be if the testimony comes from *Participant C* or was created by another party claiming *Participant C* to be the origin of this testimony. This question refers to **origin authenticity**. The question if the perpetrator identified by *Participant C* as *Participant A* is indeed *Participant A* would refer to **entity authenticity**.

These two basic examples illustrate that the distinction between these two aspects of authenticity is important in the context of forensic investigations. Hence, a working definition of the terms touched in this section is necessary. During this thesis, the following definitions are used for integrity and authenticity:

> **Integrity**:
> *Integrity refers to a certain piece of evidence. It is achieved when the this piece of evidence is not altered.*
>
> **Origin Authenticity**:
> *Origin Authenticity refers to a certain piece of evidence. It is achieved when the origin of this piece of evidence is established beyond doubt.*
>
> **Entity Authenticity**:
> *Entity Authenticity refers to an entity or entities considered in a piece of evidence. It is achieved when the identity of these entities are established beyond doubt.*

These different factors are aspects of data security. In the context of a forensic investigation, authenticity describes information about the origin of a trace and the entities considered in this trace. Integrity describes if a forensic trace is not altered. From a process point of view, both register at the moment the data is gathered and then need to be preserved in the course of the forensic investigation. In the case of origin authenticity this is information on where, when and how the trace is gathered. This information allows for an estimate on how authentic the trace is. This record itself must be preserved during the forensic process. In case of integrity, the data gathering itself might have an impact on the data by changing it. This is especially true if the data is gathered by querying various components for information. These queries inevitably alter the system state.

Again, even if the mechanism does not alter the data, mechanisms to ensure that gathered data is not altered during the forensic process are still required. Hence, authenticity and integrity must be preserved during the scope of a forensic investigation. This is summarized by [Bis18]:

> *„[...] integrity includes both the correctness and the trustworthiness of the data. The origin of the data (how and from whom it was obtained), how well the data was protected, before it arrived at the current machine, and how well the data is protected on the current machine all affect the integrity of the data.“*

It is notable that the term integrity in [Bis18] refers to data integrity and origin integrity (authenticity).

2.1.5 Error, Uncertainty and Loss in Digital Forensics

Additionally, the gathering mechanisms might be unable to perfectly gather the data. There might be some measurement errors in the sensors (or methods) used to gather the evidence. This effect is part of the problems for the evidence gathering as described by [Cas02]:

> *„When criminal activities on a computer network are being reconstructed using digital records, every aspect has some degree of error. The origin and time of the events, as well as who was responsible for the events can be uncertain. Lost data can give an incomplete picture of the crime.“*

Here, three different aspects are described as having a negative impact on the trustworthiness of evidence. These are **error**, **uncertainty** and **loss**. While [Cas02] contains no specific definition of these three aspects, the examples given can be used to construct working definitions for these three terms.

To describe error, [Cas02] describes examples for error in reconstruction and interpretation during the analysis of already gathered data. Error might occur if a given piece of evidence is not interpreted correctly, for example if entries within a file are mistaken for financial numbers when they are in fact telephone numbers. Another example would be using timestamps without considering that these timestamps might refer to another time zone.

The example given by [Cas02] describes errors during the analysis. Beyond this, there might also be errors during the gathering of the evidence. In this case, the evidence does not exactly match a perfect representation of the evidence. This might be the case if a bit flip occurs during the gathering process.

This example stands in contrast to loss in which the evidence is incomplete. In this case wrong evidence is added to the analysis.

Hence, it seems necessary to include those cases into the working definition for error:

> **Error**:
> *Error occurs if evidence is incorrectly interpreted or altered before it is interpreted.*

Uncertainty is illustrated by [Cas02] as uncertainty about the origin of a given action. The example describes how NAT (Network Address Translation) can alter an IP-address and hence make it unreliable to determine the origin of a given transmission.

Another example given by [Cas02] is the deviation between various local system clocks. Hence, the investigator cannot be certain that the time stamp associated with a given event is really accurate. There is uncertainty about this time stamp.

Beyond the examples given by [Cas02], uncertainty could be considered to occur when evidence cannot be collaborated. If there is, for example, a log of events and corresponding time stamps, one cannot confirm that the log itself is not tempered with or how accurate the time stamps are. If there are three logs of these events, taken by independent parties with respective time stamps, the evidence can be collaborated and hence a greater degree of certainty is reached. This is also the case when the time stamps might slightly differ (due to slightly different local system times) but the time offset between them is still constant.

The more general description of a lack of collaboration as the root for uncertainty opens up the path for a more general working definition of uncertainty:

> **Uncertainty**:
> *Uncertainty exists if it cannot be determined whether evidence is integer and authentic.*

As for loss, sometimes sensors are simply not capable of capturing all evidence. An easy example (given by [Cas02]) is network traffic capturing. Here a slow network interface or a slow mass memory might be overloaded with constant heavy traffic to the point that individual pieces of the network traffic are dropped. As this is quite clear and already addresses the potential reasons for loss of evidence during gathering, the following definition is straightforward:

> **Loss**:
> *Loss occurs if the entirety of evidence is not collected.*

Given these examples and definitions, there is an obvious overlap between these three aspects. This overlap boils down to the interplay between error, loss and uncertainty. If the investigator is unable to quantify the loss or error of a given method of data gathering or analysis, the uncertainty of any evidence gathered by this method will rise as rightly pointed out by [Cas02].

Forensic methods should aim at reducing error, uncertainty and loss and should at least be able to quantify these factors in order to allow for an appropriate weighting of evidence.

2.1.6 Data Protection in Digital Forensics

Another data security aspect intertwined with the forensic process is **confidentiality**. Confidentiality is defined by [Bis18] as:

> *„Confidentiality is the concealment of information or resources.“*

This concealment is with regards to parties not authorized to have access to the given information. However, during a forensic investigation the investigator might gain access to a broad range of evidence. Evidence recorded during the forensic process might be of interest to unauthorized parties for different reasons. This data might include business data required to continue the operation of a business or of interest to competitors. Forensic data might also allow a rival or potential attacker to gain a deep understanding of the investigated system. This is an unwanted effect, and hence the confidentiality of the evidence has to be preserved. Furthermore, it is possible, that this business data includes personal data.

Hence, the field of data protection and data privacy might be factors during a computer forensic investigation. There are several domestic laws which apply to questions of personal data. Hence, it is necessary to understand how these laws apply to digital forensics and what influence they might have on forensic process. Clarity on this can be achieved by looking at the respective regulations. Article 4 of the GDPR ([GDP16]) gives the following definitions for personal data and the understanding of processing:

> *„‘personal data‘ means any information relating to an identified or identifiable natural person (’data subject’); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person;*
>
> *’processing’ means any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means, such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction [..]“*

This leads to the question if personal data is located within computer systems under forensic investigation and if a forensic investigation processes this data.

There is a high chance of personal data to be found within a computer system, since many business systems store data of customers or clients. This data would fit the definition of the GDPR. Outside of business systems, personal computers will most likely also include instances of personal data. This means, that the forensic investigator should be prepared to encounter personal data while investigating a given computer system. As [Jor17] puts it:

> *„Digital forensics and incident response is fundamentally about digital evidence, and some of that evidence will be data that is affected by the European Union General Data Protection Regulation.“*

As for the processing of personal data, a computer forensic investigation in **IC2** will basically always aim at personal data. In this *Investigative Context*, the computer system is used as a means to edit data by the perpetrator. This data is most likely personal data. This data is obviously relevant to the forensic process and hence will be retrieved and disseminated - hence processed in the sense of the GDPR. However, in **IC2** a digital forensic investigation is started to support a legal case. Hence, article 9, paragraph 1 of the GDPR ([GDP16]) might give some additional insight. This article defines special categories of personal data with a higher need of protection:

> *„Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited.“*

However, Paragraph 2f of the same article ([GDP16]) states when paragraph 1 shall not apply:

> *„processing is necessary for the establishment, exercise or defence of legal claims or whenever courts are acting in their judicial capacity[..]“*

Hence, one could argue, that in the case of a forensic investigation in **IC2**, this exemption might hold true. In this case, it is up to legal minds if a digital forensic investigation into the system in question is warranted or not and beyond the scope of this thesis.

In **IC1**, personal data is not the focus of the investigation. In this context the computer system itself is the target of the attack. While an unauthorized access or alternation of personal data might be part of the attack, the investigation will focus on the breaking into the system itself. An analogy would be the case of a break-in into a bank. While it is of interest what the burglars stole, it is not the central question. Here the investigation would focus on security footage, picked locks and the like. An investigation could be performed with no detailed knowledge on the exact contents stolen. The same holds true for an investigation in **IC1** - here not the personal data itself is of central interest. Other pieces of evidence are. However, if the stolen personal data (or in the analogous example, the stolen financial documents) appear somewhere else, a detailed look at the personal data (or the stolen financial documents) is required.

This leads to two conclusions. First, in **IC1**, personal data might not play a role at all and it could be avoided to collect such data in the first place. The second is, that there are obviously sets of different data which needs to be handled differently and carry differing weights in the different *Investigative Contexts*. This has an influence on the various forensic process models (see Section 3.1.2 for more details).

2.1.7 Legal Guidelines for the Admissibility of Forensic Evidence

As discussed in Section 2.1.2, evidence gathered in a forensic investigation might be used during legal cases (see Section 2.1.2). In this case the evidence has to follow the legal guidelines for the admissibility of forensic evidence.

While this thesis does not discuss these legal guidelines in any depth these guidelines offer some insight into the properties courts require in the evidence presented before them. As a forensic investigation should be conducted in a manner in which the evidence could be used in a court case, such an analysis gives valuable input for understanding what constitutes a good forensic process model.

The legal guidelines vary as part of the different legal systems. Hence, it is beyond the scope of this thesis to cover all these different sets of rules. Instead, this work selects the legal codes of the United States and Germany respectively as examples.

2.1.7.1 United States of America

In the United States, admissibility of forensic evidence is governed by the Federal Rules of Evidence (see [Fed15]). Rule 702 handles the admissibility of testimony by expert witnesses. A forensic investigator would be considered as such an expert witness. Hence these rules pertain to the testimony of forensic investigators having performed forensic investigations:

> „***Rule 702. Testimony by Expert Witnesses*** *A witness who is qualified as an expert by knowledge, skill, experience, training, or education may testify in the form of an opinion or otherwise if:*
> *(a) the expert's scientific, technical, or other specialized knowledge will help the trier of fact to understand the evidence or to determine a fact in issue;*
> *(b) the testimony is based on sufficient facts or data; (c) the testimony is the product of reliable principles and methods; and*
> *(d) the expert has reliably applied the principles and methods to the facts of the case.“*

While these rules set the basic guideline, they have been amended in 2000 to follow the rulings made in the supreme court between Daubert and Merell Dow Pharmaceuticals, Inc. (see [Dau93]). This ruling determined the standards for the admissibility of expert testimony. It overrode the older **Frye Rules**. The notes to the amendment prove highly useful to understand the standards for admissibility of evidence to the United States Supreme Court (see [Fed15]):

> „***Committee Notes on Rules-2000 Amendment*** *Daubert set forth a non-exclusive checklist for trial courts to use in assessing the reliability of scientific expert testimony. The specific factors explicated by the Daubert Court are (1) whether the expert's technique or theory can be or has been tested-that is, whether the expert's theory can be challenged in some objective sense, or whether it is instead simply a subjective, conclusory approach that cannot reasonably be assessed for reliability; (2) whether the technique or theory has been subject to peer review and publication; (3) the known or potential rate of error of the technique or theory when applied; (4) the existence and maintenance of standards and controls; and (5) whether the technique or theory has been generally accepted in the scientific community. [...]*
>
> *No attempt has been made to "codify" these specific factors. The ruling itself emphasized that the factors were neither exclusive or dispositive. Other cases have recognized that not all of the specific Daubert factors can apply to every type of expert testimony.“*

This ruling ([Dau93]) is often referred to as the **Daubert-Ruling** and hence the resulting criteria are often referred to as the **Daubert-Criteria** or the **Daubert-Factors**. While the Daubert-Factors are not used by all state courts of the United States, they are highly influential by virtue of being the standard for the admissibility of evidence to the United States Supreme Court. The use of the Daubert-Factors has also increased over the years. In 2004 a survey on the implementation of the Daubert-Rulings in various state courts (see [BJ04]) stated:

> *„As the analysis in this article shows, only a minority of state courts have wholeheartedly adopted the Daubert trilogy. Many states adopted Daubert when they thought that it clearly liberalized admissibility standards for scientific evidence relative to the old Frye general acceptance test.“*

While in 2004, only a minority of states had adopted the Daubert-Criteria, roughly half of the United State state courts applied the Daubert Criteria in 2008 (see [Cal08] for a discussion on the use of either Daubert or Frye in the United States justice system in the year 2008). In 2019, 35 state courts of the United States use the Daubert Criteria, 10 the Frye Standard and 5 other rulings for the admissibility of evidence (see [Mat19] for a survey on the use of various standards for the admissibility of evidence in the United States justice system in the year 2019). In addition, even when Daubert is not adapted, it has influence on the question of admissibility of evidence. An example from [Cal08] states:

> *„South Carolina's test, like Daubert, imposes a "gate-keeping" requirement for scientific evidence. However, South Carolina's test uses different factors to assess reliability: "(1) the publications and peer review of the technique; (2) prior application of the method to the type of evidence involved in the case; (3) the quality control procedures used to ensure reliability; and (4) the consistency of the method with recognized scientific laws and procedure."“*

This shows the relevance of the Daubert-Criteria. Hence, they can serve as a guideline to judge the admissibility of any evidence found during a forensic process. As such, any forensic method should aim to fulfill the criteria as stated in the Committee Notes on Rules-2000 Amendment (see [Fed15]).

2.1.7.2 Germany

The admissibility of digital evidence is also a relevant topic for German courts. An overview on how this topic is handled can be found in [Kno09]. This article describes the use of digital evidence in the German legal system in theory and praxis. Additionally, it gives an overview on how the evidentiary value of evidence is determined. Since this fundamental source is written in German, all quotes taken from [Kno09] will be given as translation by the author of this thesis.

According to [Kno09], the digital domain comes with some serious challenges for the admissibility as evidence. A major factor is the fact that digital sources can be easily manipulated without specific traces when compared to physical evidence. This is especially true for electronic text documents when compared to hand-written script. Additionally, the ability to create artificial photo, sound or video recordings is bringing new challenges for the consideration of evidence.

The article defines two different perspectives for digital evidence.

The first perspective is the *ex post* perspective, where a piece of digital evidence is presented in front of a court and the court has to decide on the evidentiary value of this evidence.

The second perspective is the *ex ante* perspective. This perspective aims at creating evidence with a high evidentiary value. The creation of a treaty between parties in way that guarantees high evidentiary value is an example. This could be done by using digital signatures. Another example would be the documentation of various events, including the actions performed during a forensic investigation.

According to German Process law, various types of evidence are common during court proceedings. According to [Kno09], *legal inspection* and *expert testimony* are relevant in the case of digital evidence.

In the case of *legal inspection*, the judge reviews the evidence as it appears. This is the case if the judge reviews digital text documents, video recordings and the like. Being able to perform this type of inspection relies on the availability of hardware and software able to interpret the given file format. This might be problematic in the case of obscure proprietary file formats. As [Kno09] points out, there is also a problem with trust towards the hardware and software used in the inspection of the material. Evidence could potentially contain hidden fragments which exploit properties or errors of common playback software in order to alter the reproduced content. This lack of trust is an even greater concern if proprietary systems not operated by the court itself have to be used to review evidence. In order for the court to be able to trust their inspection of the evidence, an additional review of the hardware and software used for reproducing the content is necessary to ensure the functionality and freedom from tampering.

If using *expert testimony* the expert reviewing the evidence in question has the same problems concerning the trust in the used hardware and software. However, an expert witness is in general better equipped to evaluate the trustworthiness of the used hardware and software than a judge. Additionally, the evaluation of many pieces of digital evidence might require expert knowledge to extract the evidence from a computer system in the first place and then to interpret it later.

According to [Kno09] German courts prefer using physical evidence or witnesses instead of relying on digital evidence when possible.

However, [Kno09] gives noteworthy criteria for establishing the evidentiary value of admitted evidence. While the elaboration offered in this article focuses on digital evidence, these criteria also hold true for physical evidence. The article discusses three primary reference points:

- Integrity and Authenticity
  The article identifies this as the main reference point concerning digital evidence. Only if a piece of digital evidence is with predominant probability not manipulated it can be used during a legal trial. While integrity and authenticity are discussed in more detail in Section 2.1.4, [Kno09] points out that integrity also includes the completeness of digital data and that authenticity includes the attribution of a given file to a person. This

is a major factor in legal proceedings, especially when using text files or spreadsheets potentially created by a suspect in **IC2**.

- Creation time, documentation and file history
  Further reference points are the creation date, a documentation of the creation and the file history. The file history describes when and how the file was edited in which way. Again, these reference points are only usable if they are reliable. For an example, [Kno09] uses the digital representation of a photo showing a certain event. Here the exact date of the file creation, the circumstances and the history of how this photo was edited are of importance. Again, this example addresses **IC2**.

- Information about the data source
  This describes the freedom of interference of the recording system. It also takes into account how uninfluenceable the specific recording system in regards to a potential attacker. These aspects aim at the traceability of the creation process of the evidence. To determine the evidentiary value of a certain piece of evidence, it must be known if the system recording this evidence was working correctly and without external or hidden interference in the creation process. If the creation process is secured against such interference and the correct operation is documented, the evidence has a higher evidentiary value. Again, the explanation offered by [Kno09] is mostly focused on digital evidence used in **IC2**. The aforementioned recording systems could be digital cameras while the evidence in questions would be photos taken of a certain event. Here, these photos will carry a higher evidentiary value if it is confirmed that the camera in question works correctly and adds the correct time stamps to the pictures taken.

These reference points offer some advice on the properties of useful forensic evidence. However, the examples given by [Kno09] focus strongly on **IC2**. This is understandable when reviewing the German crime statistics from 2018 [BKA18]. In this statistic, *Computer fraud* alone accounts for 81.4% of the crimes that are grouped under the field of *computer crime*. *Computer fraud* describes frauds executed by using a computer system. It is a typical example of a case situated in **IC2**. *Data manipulation*, *computer sabotage* and *theft of data* as examples for **IC1** amount to around 10% of the registered cases.

2.1.7.3 Conclusions from Legal Guidelines for the Admissibility of Forensic Evidence

The reviews of the US-American and German guidelines for the admissibility of forensic evidence show that determining the evidentiary value is the primary factor for the use of digital evidence. In the US-American legal system, the Daubert-Factors play an important role to determine if a given method produces evidence that is admissible and the evidentiary value assigned to this evidence. Factors that German courts will consider when evaluating the evidence before them have been proposed by [Kno09]. Combining these factors gives some insight on the properties evidence should have in order to be admissible and reliable. Hence, these factors shall be introduced as *Admissibility Factors* (*AF*):

- **Admissibility Factor 0** (**AF0**):
  The evidence helps the trier of fact to determine a fact in issue

- **Admissibility Factor 1 (AF1)**:
  The technique or theory producing (or drawing a conclusion from) the evidence has been peer reviewed

- **Admissibility Factor 2 (AF2)**:
  The technique or theory producing (or drawing a conclusion from) the evidence is generally accepted in the scientific community

- **Admissibility Factor 3 (AF3)**:
  The technique or theory producing (or drawing a conclusion from) the evidence has been tested for reliability

- **Admissibility Factor 4 (AF4)**:
  The technique or theory producing (or drawing a conclusion from) the evidence has a known rate of error

- **Admissibility Factor 5 (AF5)**:
  The technique or theory producing (or drawing a conclusion from) the evidence follows standards and controls

- **Admissibility Factor 6 (AF6)**:
  The technique producing (or drawing a conclusion from) the evidence ensures the integrity and authenticity of the evidence

- **Admissibility Factor 7 (AF7)**:
  The technique producing (or drawing a conclusion from) the evidence documents the process of recording or storing a certain event or piece of information

These factors are arranged in this specific order to underline their different nature.

AF0 describes that the given evidence has any use for the court in that it could help determine a fact in issue. In essence, it describes if the evidence is relevant to the specific case at all. This is an important factor from the viewpoint of the court. For the forensic investigator, this is not usually the case. The forensic investigator is not tasked with judging if a given piece of evidence is relevant to the court case in question. This tasks falls into the responsibility of the court. Hence, from the viewpoint of the forensic investigator, this factor is not relevant. Therefor the numeral *0* was assigned to it in order to visualize this clear difference from the other factors. **AF0** is motivated by the inclusion of item *(a)* in [Fed15].

AF1 and **AF2** describe whether the technique or procedure leading to the given piece of evidence is working at all. **AF3** and **AF4** describes how well and reliable the technique or procedure is working by focusing on reliability tests and known error rates. These four factors are based on the Daubert-Rulings from [Fed15]. These factors address a specific technique or procedure, while **AF5**, **AF6** and **AF7** address a specific implementation of that technique.

For example, the method of taking video recordings of events in order to prove to a court that the event took place could be, as a general procedure, peer reviewed and accepted (**AF1** and **AF2**). In this case, factors concerning the human visual perception like the video being taken during twilight (**AF3**) or the human capacity to wrongly recognize persons (**AF4**) could be known. **AF5** would consider if the camera in question is built to industry standards producing known video formats. It would also address if this camera is known to not include mechanisms an attacker could use to alter the video in question in a simple and hidden manner. **AF6** and **AF7** would address the inclusion of a reliable time stamp or even digital signatures for the videos taken by this specific camera.

Another interesting fact is the prevalence of **IC2**. The German crime statistics from 2018 (see [BKA18]) provide an overview on the relative occurrence of **IC1** against **IC2**.

2.2 Computational Domains

In the scope of this thesis, the application of forensic science to investigate a range of events in various different IT domains is explored. Hence, an understanding of the various IT domains is required. This understanding focuses on similarities and differences which might impact the forensic process. This section also serves to define the scope and understanding of these domains for the remainder of this thesis.

2.2.1 Classical IT

As with most things that almost everyone knows something about, Classical IT is hard to define. A definition is, however, necessary in order to compare it with other computational domains. Since this thesis discusses the topic of forensics in the digital domain, it is worth taking a look on the computing devices covered by most of the literature on the field of computer forensics. While a specific discussion on the various models to conduct forensic investigations in this domain is covered in later chapters (see Section 2.3.1 for the discussion of the current state of the art of forensic in this computational domains), it suffices to say these usually cover computing systems designed for personal or business use. These systems consist of input devices (keyboard and mouse), computing unit, memory (volatile working memory and non-volatile mass storage) and output devices (monitor, speakers). Also, these systems are often interconnected to each other. During the course of the early twenty-first century this interconnectivity has become the norm.

This domain covers classical computer systems, often referred to as Desktop Computers or Personal Computers. They originate from the concept of Home Computers as relatively affordable and accessible computers for personal or business use. In short, the domain of Desktop IT covers everything most users would immediately associate with the term ’computer’.

As these systems are used in business environments, the domain of business informatics offers some advice. For example, a useful definition can be found in [LL11]:

> *„Information technology (IT) consists of all the hardware and software that a firm needs to use in order to achieve its business objectives. This includes not only computer machines, storage devices, and handheld mobile devices, but also software, such as the Windows or Linux operating systems, the Microsoft Office Desktop productivity suite, and the many thousands of computer programs that can be found in a typical large firm.“*

This definition clarifies, that IT consists of hardware and software components. Furthermore, these software components encompass operating systems as well as applications.

This opens up the way to a more formal definition which is used for this thesis. Since all computational domains rely on the use of computers to achieve certain goals, they are quite similar in their structure and the technologies used. Hence, the main difference is in the aim of use of these technologies. Therefore, the domain of Desktop IT shall be defined as follows for the scope of this thesis:

> **Desktop IT**:
> *The domain of Desktop IT covers computer systems designed to receive, store, manipulate and transmit data (or information)*

This definition focuses on the fact, that computers in this domain are used to manipulate data - in contrast to embedded systems which mostly aim at interfacing with physical processes. It also addresses the abbreviation *IT* which stands for *Information Technology.* For a better understanding, the term *Desktop Computers* might be used when referring to specific computer systems. Furthermore, when it comes to emphasizing that a statement refers primarily to networked systems, the term *networked Desktop IT* is used.

2.2.2 Industrial Control Systems

In simply terms, Industrial Control Systems (ICS) are systems used to control industrial processes. A conclusive definition can be found in the glossary of [NIS15]:

> *„General term that encompasses several types of control systems, including supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC) often found in the industrial sectors and critical infrastructures. An ICS consists of combinations of control components (e.g., electrical, mechanical, hydraulic, pneumatic) that act together to achieve an industrial objective (e.g. manufacturing, transportation of matter or energy).“*

Another, yet similar, definition can be found in [KL15]:

> *„An industrial control system (ICS) is a broad class of automation systems used to provide control and monitoring functionality in manufacturing and industrial facilities. An ICS actually is the aggregate of a variety of system types including process control systems (PCS), distributed control systems (DCS), supervisory control and data acquisition (SCADA) systems, safety instrumented systems (SIS), and many others.“*

These definitions cover what kind of systems are used for which kind of tasks. Both point towards the use of such ICS in industrial and manufacturing facilities. This also leads to this domain sometimes being referred to as *Operational Technology (OT).* Both definitions also point out an inherent diversity of the computational components within this domain. The several types of control systems mentioned in this definition cover different aspects of the 'industrial objective'. As such, these components work together in order to perform this objective under the guidance of operators (or engineers). This obviously requires some form of hierarchy. Before going into more detail on the various components, discussing these hierarchies will provide more clarity about the usage of these specific types of control systems.

2.2.2.1 Purdue Enterprise Reference Architecture (PERA)

These different aspects of the ’industrial objective’ form hierarchies found in most industrial control systems. The Purdue Enterprise Reference Architecture (PERA, see [Wil92]) is often used to describe enterprise hierarchies including industrial control systems. Since an understanding of this hierarchy offers some more insight into the understanding of Industrial Control Systems, this reference architecture is presented here. It describes five different levels of components inside an enterprise hierarchy and assigns them to different zones. A handy summary of these levels is taken from [RW211]:

- *„**Level 0 - Process** sensors and actuators involved in the basic manufacturing process, performing basic functions of the ICS, e.g. driving a motor, measuring variables, setting an output, and performing physical actions e.g. painting, welding or bending,*
- ***Level 1 - Basic Control** controllers (typically a PLC) that direct and manipulate the manufacturing process, interfacing with the Level 0 devices (e.g., I/O, sensors, and actuators),*
- ***Level 2 - Area Supervisory Control** applications and functions associated with the Cell/Area zone runtime supervision and operation (incl. operator interfaces or alarms),*
- ***Level 3 - Site Level** plant-wide ICS functions,*
- ***Level 4 - Site Business Planning and Logistics** functions and systems (incl. basic business administration tasks) that need standard access to services provided by the enterprise network,*
- ***Level 5 - Enterprise** centralized IT systems and functions (e.g. Enterprise resource management, business-to-business, and business-to-customer services).“*

The levels are grouped into three zones. (see Figure 2.3). Levels 0 to 2 form the **Cell/Area-Zone**. Level 0 to Level 4 constitute the **Manufacturing Zone**. Level 4 and Level 5 form the **Enterprise Zone**.

In this hierarchy, the Manufacturing Zone compromises everything that falls into the definition of ICS according to [NIS15]. The Enterprise Zone represents the Information Technology required to run an enterprise. PERA points out how closely intertwined these two domains are in a practical context. Although, they differ in terms of tasks, technologies and employed components. In essence, this describes the relationship between ICS and Desktop IT (see Section 2.2.1).

2.2.2.2 Industrial Control System Components and Working Definition of Industrial Control System Terms

The hierarchy presented in PERA (see Section 2.2.2.1) provides a guideline on where the different components listed within the definitions given in Section 2.2.2 are located within an Industrial Control Systems.

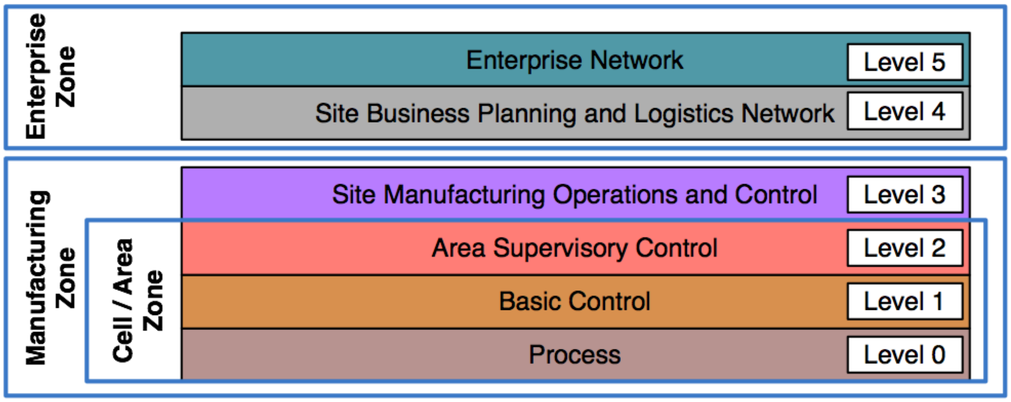


Figure 2.3: Levels in Industrial Control Systems according to [RW211]

Programmable Logic Controllers (PLC) are directly attached to the physical process. They perform basic control functions like reading sensor outputs and sending control signals to various actors. Hence, they cover Level 1 of the PERA model. On higher levels, Supervisory control and data acquisition (SCADA) collect information about these PLCs and present them to an operator or engineer. These SCADA systems are geared towards presenting an interface between the operators (or engineers) and the physical process. This takes place on either Level 2 or Level 3 of the PERA model - depending on whether the supervised system in question is barely a subsystem or the collection of subsystems in itself.

In this domain - especially in the case of PLCs - the computing units are not necessarily obvious to the casual observer - they are embedded in the process. The computer systems employed here do not necessarily require all the components computer systems in the Desktop domain employ - input devices might be reduced to simple on/off switches and output devices might be reduced to simple status lights - if present at all. The same goes for the SCADA systems, which constitute a Human-Machine-Interface (HMI). These HMIs will often use displays as output devices but input devices which often might be switches or touch-screens, while keyboards are a rarity.

After establishing an understanding of what an ICS is supposed to do and what components it contains in which specific functions, it is necessary to formulate a definition of ICS as understood within the scope of this thesis. In addition, different terms mentioned in passing beforehand need to be defined. The best way to achieve this, is to define the various components present in ICS. Based on preparatory work (see [**AHKD19**]), these components are defined as follows for the scope of this thesis:

Sensor:
Collects information about the environment, e.g. a physical process.
Actuator (or **Actor**):
Manipulates the environment, e.g. a physical process.
Processing Unit (or **Programmable Logic Controller**):
Evaluates the data gathered by sensors and/or gives control signals to actors.
Communication Wiring:
The physical and logical carrier that facilitates communication between sensors,

actors and processing units.

Note that this definition allows for processing units which only gather data from sensors and/or give control signals to actors and perform no computation beyond those necessary for this relay of data and/or commands.

In this case, these processing units might forward the gathered data to another processing unit which in turn evaluates the data and gives control signals to an actor. Here the tasks of reading from sensors and passing control signals to actors are distributed. Hence, in this case a distributed control system (DCS) is used in contrast to a monolithic control unit:

Distributed Control System (**DCS**):
A group of processing units which distribute the task of reading sensors, processing data and sending control signals to actors between the various processing units.

With almost all components mentioned in the definition from [NIS15] defined, only the supervisory control and data acquisition (SCADA) is missing out. For the course of this work, a straightforward definition is used. This definition is based on the tasks such a system is supposed to provide. During the course of this work the following working definition will be used:

Supervisory control and data acquisition (**SCADA**):
A system which collects data about a physical process and presents those to an operator, granting the operator the ability to supervise and influence the process. The data is gathered from the processing units and the ability to influence the process is granted by sending signals to the respective processing units. Dedicated overrides might gather the data directly from the sensors or might control actors directly (e.g. emergency stop switch).

With all these terms cleared up, the definition from [NIS15] serves as a foundation for the working definition of ICS used within the scope of this work:

Industrial Control System (**ICS**):
A communication network of Actors, Sensors and Processing units geared towards controlling a physical process.

This definition is not geared towards manufacturing processes and includes systems used to control physical processes outside of manufacturing contexts. Examples would include control systems which regulate sluices in order to keep water levels even. Such a system would employ the same kind of components as discussed in this section. It also implies increased risk for life and limb and could be classified as a critical infrastructure. Other systems classified as critical infrastructure, like power generation, are also included by this definition.

2.2.2.3 Industrial Control System in the Wild

Industrial Control Systems come in many different shapes and forms controlling a broad range of different functions while still adhering the definition provided in Section 2.2.2.2. This section gives a brief overview on where and how Industrial Control Systems are used to facilitate a better understanding of the term. The overview provided here is high-level and mainly used to show the broad range and abundance of ICS. The scenarios in which these systems are employed are thoroughly analyzed in Section 4.3.1.

Industrial Control System are used for a variety of tasks. These include:

- Controlling manufacturing processes
  Processes which transform raw material (or parts) into a final product (or different parts).

  This includes conveyor belts and industrial robots which assemble vehicles as well as chemical plants where chemicals are mixed under specific conditions.

- Controlling various physical processes
  Physical processed controlled in a way that is beneficial to a given task. This would include controlling the pumps and sluice gates of a pumped hydroelectric energy storage or the process of generating energy in a power plant. In fact, it could also include the sluice gate controls used to keep the water level in a given range to prevent flooding or areas.

- Building Management Systems (BMS)
  A system that controls a buildings mechanical and electrical systems, including lighting, heating, ventilation, air conditioning (often grouped together as HVAC) and room access.

2.2.3 Automotive Systems

Automotive systems describe systems that are able to move by themselves using some form of propulsion. Usually, this refers to civilian land vehicles, especially cars and trucks. According to the Cambridge Dictionary [1], a vehicle is defined as:

> *„Vehicle:*
> *a machine, usually with wheels and an engine, used for transporting people or goods on land, especially on roads“*

This definition describes vehicles along the lines of the task they are used for. Furthermore, it mentions some of its components. The components mentioned here are required to perform the task of transporting people or goods.

This definition does not put many limits on what constitutes an automotive system and hence covers a broad range of very different vehicles. Not all the vehicles that befit this definition are discussed in the scope of this thesis. In this thesis, automotive systems are mostly represented by cars. According to the Cambridge Dictionary [2] a car is defined as:

[1] https://dictionary.cambridge.org/dictionary/english/vehicle, 19/02/2020

[2] https://dictionary.cambridge.org/dictionary/english/car, 19/02/2020

> *„Car:*
> *a road vehicle with an engine, four wheels, and seats for a small number of people“*

While this thesis discusses the influence of emerging technologies, like the increasing automation of driving task, on the forensic process, this thesis views road vehicles for a small number of people controlled by a driver as the baseline scenario for all considerations on computer forensics in Automotive IT systems. Since cars are indeed very numerous road vehicles and they share various technologies and components with other road vehicles (like trucks, for example), cars provide a representative basis.

2.2.3.1 Vehicle Components and Working Definition of Automotive Terms

Formerly, the vehicle was controlled using mechanical components. Modern vehicles utilize a broad range of electronic components instead. These include computing units which control the driving process. The electronic components controlling this driving process are referred to as **Automotive IT** during this thesis. Hence, the following discussion establishes the scope of the **Automotive IT domain**.

From the viewpoint of computer science and computer forensics, the components of vehicles can be divided into dumb and smart components. Classical, mechanical components are considered dumb since they do not contain any program logic or are attached to electronic systems which contain program logic in order to control their function. Smart components in contrast either contain program logic themselves or are attached to such a component containing program logic able to control or influence its functions.

These smart components can be broken down into various sets of components as has been done by the preparatory work [**ALKD17**]. There, five groups of smart components are defined.

A preliminary definition for these components in [**ALKD17**] is as follows:

> *„Sensors measure the conditions of the vehicle's systems and environment (e.g. pressure, speed, rain intensity etc.) but can also capture user input requests.*
>
> *Actuators are units that perform a mechanic actuation.*
>
> *Electronic Control Units (ECUs) perform the electronic processing of input signals, which are acquired via different types of sensors and relay commands to the actuators. [..]*
>
> *Direct analogue cable connections are used to carry measured signals (from sensors to ECU) or actuation impulses (from ECU to actuators).*
>
> *Shared Digital Bus Systems are used for communication among ECUs. Beyond the direct connections between ECUs and sensors/actuators, ECUs are additionally connected amongst each other via digital field bus systems [..]. This shared medium is used to exchange required information, like forwarding digitized sensor signals, exchanging current operating parameters, remote actuation requests or diagnostic requests for maintenance purposes. [..]“*

These types of smart components seem quite similar to those present in Industrial control systems (see Section 2.2.2.2). Both domains contain sensors which gather information, processing units which process this information (Programmable Logic Controllers in the ICS domain and Electronic Control Units in the Automotive IT domain respectively) and actuators which act based on these computations. While the underlying physical processes are different, the smart components used to control these physical processes are very similar.

In order to improve clarity of terms and ease understanding, it seems useful to match the terms used within the Automotive IT domain as best as possible to these used in the ICS domains. As mentioned before, sensors, actors and computing units perform the same basic tasks in both domains.

The only difference is a division between various types of communication technology based on their task. In the current definition, *Direct analogue cable connections* are used for communication between sensors and the respective computing unit performing a computation based on this sensor reading. They might also be used for the communication between a computing unit and the respective actor implementing an action based on a signal from this computing unit. *Shared Digital Bus Systems* are used for the communication between computing units and are in general more complex.

As will be shown during the course of this work (see Section 4.2.3), such a division carries meaningful impact on the forensic process.

The *Direct analogue cable connections* are point-to-point (unicast) and serve as inputs or outputs for the respective computing units. Communication on these channels is usually unidirectional. Hence, if communication takes place on such a connection, sender and receiver are well-defined. The communication structure is very easy. For *Shared Digital Bus Systems*, more complex communication scenarios are possible. The amount of potential communication partners alone confirms this fact. This is not specific to the Automotive IT domain but also holds true for the Industrial Control System domain.

While these two different types of communication channels carry different implications for the forensic process, the definition given in Section 2.2.2.2 is broad enough to encompass both types at this point of time. While analyzing the impact of these different types of communication channel is left open to later sections of this thesis (see Section 4.2.3 for the section concerning Automotive IT architectures and Section 4.2.2 for the section concerning ICS architectures respectively), the following definition for components of Automotive IT will be used during the course of this thesis:

Sensor:
Collects information about the environment, e.g. a physical process
Actuator (or **Actor**):
Manipulates the environment during the driving task
Processing Unit (or **Electronic Control Unit**):
Evaluates the data gathered by sensors and/or gives control signals to actors
Communication Wiring:
The physical and logical carrier that facilitates communication between sensors, actors and processing units

These definitions are in line with the ones discussed is Section 2.2.2.2, already demonstrating the kinship of these domains.

In both domains, physical processes are controlled using actors which implement the control systems provided by processing units based on the input of sensors about the current state of the physical process and its surroundings. From this point of view, the main difference between these two domains is the task which the respective systems are designed to perform.

2.2.3.2 Automotive Systems in the Wild

Although this thesis focuses on cars, it is notable that other vehicles use similar technology. The obviously related field of trucks and buses uses the same basic computing units to control the vehicle. In the case of trucks additional actors might be added in order to support loading procedures. An example would be the hydraulic rams used to empty the open box-bed attached to a dump truck. In the case of buses these additional actors would be automated opening mechanisms for the doors and information systems for the passengers.

In essence, the same technologies are also used inside trains. Here additional actors for the opening and closing doors would be installed as well as information systems for the passengers. The same goes for trams. In the case of (public) transportation, these systems are integrated into an overall system which is tasked with controlling the schedule of the various vehicles.

As a more extreme example, planes use the same basic building blocks for their computer systems as the cars discussed in the scope of this thesis.

2.3 Forensics in the Scope of the Different Computational Domains

As discussed in detail in Section 2.1, forensic science is used to reconstruct the events which are subject to potential court cases or disrupt the intended use of the systems. Hence, forensics is a process which contains various activities (see Section 2.1.3) and faces various challenges:

- integrity and authenticity of traces (see Section 2.1.4),
- error, uncertainty and loss (see Section 2.1.4),
- legal constraints for the collection and use of evidence (see Section 2.1.6), and
- legal requirements for the admissibility of evidence in court (see Section 2.1.7).

These challenges are not unique to computer forensics within and have been addressed by practitioners and theorists for various domains. Experience is usually imparted in the form of guidelines and best practices. As forensic investigations aiming at computer systems have been a relevant topic for quite some years, various guidelines for performing computer forensic investigations, especially in the domain of Desktop systems, exist. This section presents an overview into these guidelines tailored towards the use in the Desktop IT domain (see Section 2.3.1). The resulting forensic process models will be compared in Section 2.3.2 in order to identify factors which constitute a good forensic process model. Common themes and differences will be addressed to serve as a background for the selection and introduction of the forensic process model used in the scope of this thesis. Furthermore, an overview into publications on the topic of forensic investigation into computer systems aside from classic IT environments is given in Section 2.3.3 and Section 2.3.4.

2.3.1 State of the Art in Desktop IT Forensics

Forensics in the Desktop IT domain is a rather well-researched topic, as the broad number of publications on this topic confirms. This section gives an overview on the development of process models and guidelines on how to perform forensic investigations on classical Desktop IT systems. This exploration of forensic process models aims at understanding what entails a useful forensic process model. Hence, this review will look into the specific models in more detail than the survey performed in [Pol07] by taking into account the models respective understanding of the computer forensic process, how the forensic process is started and which notable concepts are embedded within the specific models.

The selection of discussed models represents some prominent (historic) examples published during the development of forensic science in the Desktop IT domain as well as some recent examples. These examples are presented in chronological order. Those selected examples are:

- **1995** M. Pollit ([Pol95])

  Computer Forensics - an approach to evidence in cyberspace

- **2001** M. Braid ([Bra01])

  Collecting Electronic Evidence After a System Compromise

- **2003** K. Mandia, Chris Proise, Matt Pepe ([MPP03])

  Incident Response & Computer Forensics (2nd Edition)

- **2004** E. Casey ([Cas04])

  Digital Evidence and Computer Crime (2nd Edition)

- **2007** F. Freiling, B. Schwittay ([FS07])

  A Common Process Model for Incident Response and Computer Forensics

- **2017** A. Arnes ([Arn17])

  Digital Forensics

Please note, that some sources are used in outdated (or even multiple editions) in order to provide a historic overview on the development of computer forensic models used in the domain of Desktop IT.

2.3.1.1 Computer Forensics - an Approach to Evidence in Cyberspace ([Pol95])

This work (see [Pol95]) is a very early example of a process model to handle computer forensic investigation. This early publication discusses the fundamental differences between 'classical' evidence and 'digital' evidence. This is done by introducing a 'Document Paradigm', which states the differences between documents as physical evidence and documents as digital evidence. The work identifies differences in four distinct phases. These phases and the differences between physical and digital evidence are, in the words of the original author (Pollit) as follows:

> *„First, a document is acquired. How it is acquired (via consent, search warrant, a public record, business record) is subject to a set of rules that have a long and well-documented history. Even so, there are often cases where there will be room for disagreement which will then result in litigation. Rarely is determining that the document physically exists or where it came from, a problem. With digital evidence, this is often a problem. What does this binary string represent? Where did it come from? While these questions, to the computer literate, may seem obvious at first glance, they are neither obvious nor understandable to the layman. These problems then require a substantial foundation being laid prior to their admission into evidence at trial.*
>
> *If the document is in English, then anyone who can read English can probably determine what the document says. It's format and content define its purpose. A binary file requires conversion, in the form of a program, which will transform the data into a form which is humanly readable. Only then, can a human determine what the document is.*
>
> *Evaluation of the document follows. This is the time when the reader determines if the information contained in the document is relevant and determines who could testify concerning this document. When our digital data is in human readable form, we can also make these determinations. However, the electronic context of a file is arguably still significant. This will impact on how the evidence is introduced and by whom.*
>
> *Ultimately, the document may be offered for evidence. This must be done by a warm, breathing, human being who has legal standing to explain it's origin, it's meaning, or both. In the case of paper evidence, the judge and jury may physically inspect the paper and will hear someone who is personally aware of the document describe it and it's significance. It is not necessary to explain the three prior steps to the court, as these are generally accepted by all participants. At this stage of legal history, such is not the case for digital evidence. As a result, it is often necessary to have the testimony of someone who can explain the process of acquisition, identification, and evaluation.“*

Arranged more clearly, the four distinct phases as proposed in this paradigm are:

- Identification
- Acquisition
- Evaluation
- Admission as Evidence

It is noteworthy, that the forensic process includes the admission of evidence in front of a court. This emphasizes that this early work is strictly aimed at law enforcement and court filings. This work does not take other scenarios where an event reconstruction is useful into consideration.

Based on the differences between physical and digital evidence, Pollit proposes a *Digital Paradigm* to describe how something can become digital evidence. Although not explicitly stated, Figure 2.4 gives a good overview of this process.

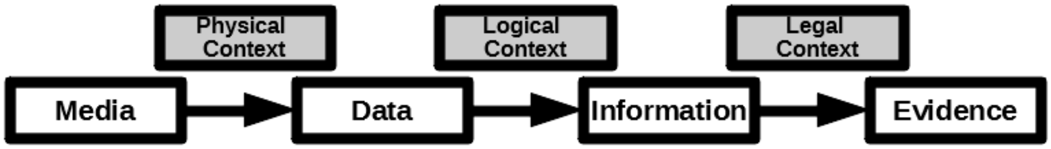


Figure 2.4: The Digital Paradigm, according to ([Pol95])

In order to extract data from a physical medium, a physical context is required. This could be an understanding of how to extract this data from the physical medium or a device being capable of doing so. For example, a laser scanning the indentations of an optical disc would be such a device. Then a logical context is applied to this stream of data. In this case, this logical context would be a file system and/or a file format to interpret the data on this optical disc. After the information contained on this media is made available, a legal context is applied to ascertain if this piece of information might be of any value for a judicial process and hence could be considered evidence.

This exemplifies the problems with the legal inspection of evidence as discussed in the paragraph about the admissibility of evidence in German Courts in Section 2.1.7. As [Kno09] pointed out, trust towards the hardware and software used in order to put the evidence in different context is a major factor in determining the evidentary value of the given evidence.

2.3.1.2 Collecting Electronic Evidence After a System Compromise ([Bra01])

This work (see [Bra01]) adds a new angle to the field of computer forensics. The author states that collecting electronic evidence serves two distinct purposes. The first one (coined as *Responsibility*) is akin to the purpose presented by [Pol95] in aiming to identify a culprit and find evidence against the respective culprit. The second one (coined as *Future Prevention*) aims at investigating how an attack occurred in order to prevent future attacks.

The author lays out some challenges and complexities which are specific to digital forensics. These complexities can be summarized as follows:

- „*Generally, electronic evidence has none of the permanence that conventional evidence has* “
- „*[electronic evidence] is more difficult to present in a way that can be readily understood* “
- „*it is very difficult to tie the transaction to a person* “

The complexities identified by the author of [Bra01] show that the main focus of computer forensics remains enabling court proceedings. In the context of this source, future prevention is important, but an afterthought. This work also presents five general rules of evidence which align closely to the forensic aspects as discussed before (see Section 2.1). As provided in [Bra01] these five rules are:

> *„**1. Admissible***
>
> *This is the most basic rule - the evidence must be able to be used in court or elsewhere. Failure to comply with this rule is equivalent to not collecting the evidence in the first place, except the cost is higher.*
>
> ***2. Authentic***
>
> *If you can't tie the evidence positively to the incident, you can't use it to prove anything. You must be able to show that the evidence relates to the incident in a relevant way.*
>
> ***3. Complete***
>
> *It's not enough to collect evidence that just shows one perspective of the incident. Not only should you collect evidence that can help prove the attacker's actions but for completeness it is also necessary to consider and evaluate all evidence available to the investigators and retain that which may contradict or otherwise diminish the reliability of other potentially incriminating evidence held about the suspect. Similarly, it is vital to collect evidence that eliminates alternative suspects. For instance, if you can show the attacker was logged in at the time of the incident, you also need to show who else was logged in and demonstrate why you think they didn't do it. This is called Exculpatory Evidence and is an important part of proving a case.*
>
> ***4. Reliable***
>
> *Your evidence collection and analysis procedures must not cast doubt on the evidence's authenticity and veracity.*
>
> ***5. Believable***
>
> *The evidence you present should be clear, easy to understand and believable by a jury. There's no point presenting a binary dump of process memory if the jury has no idea what it all means. Similarly, if you present them with a formatted version that can be readily understood by a jury, you must be able to show the relationship to the original binary, otherwise there's no way for the jury to know whether you've faked it.“*

These rules still hold true today. Rule 2, 3 and 4 form the core of any forensic investigation since they basically dictate the conclusiveness of the evidence. Rule 1 and 5 are especially important if the forensic investigation is done to support a court proceeding. Here, admissibility and presentation are more important topics. If an investigation is done in order to close a vulnerability to prevent future damage, it is not of such a great importance that this evidence can be presented in a way a judge or a jury could follow it. Also, the potential admissibility of the evidence to a court is not of relevance in this case.

The procedure presented in this work is also aimed at supporting a judicial proceeding, as can be seen by the specifics of the forensic process in this model.

The author gives practical advice on how to perform an investigation into forensic traces within computer systems. Therefore, specific advice on certain aspects of the investigation is given. This includes some rather generic points and some more specific advice like keeping detailed logs of any action performed during the investigation. Here, the concept of *Order of*

Volatility is motivated. This concept states that some evidence is located in volatile memory and not available after a system is powered down. Furthermore, some evidence might be altered constantly.

The forensic process itself is defined in this work as a four-step procedure which gives a high level outline of how to perform a forensic investigation. The procedure is shown in Figure 2.5.

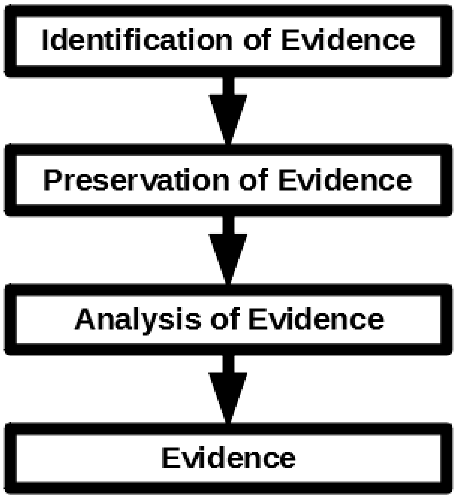


Figure 2.5: General procedure for a forensic investigation according to ([Bra01])

According to this procedure, a forensic investigation starts with the *Identification of Evidence* where the identification of potential sources of evidence is performed. Here, an understanding of which data might be of interest as well as how and where this data is stored. In this phase, the volatile nature of some forensic traces comes into effect. The author proposes to draw up a list of an order of volatility in order to determine which traces have to be preserved first.

The next step is the *Preservation of Evidence* in which said data is collected. A major challenge here is evidence dynamics (see Section 2.1.4). The author states:

> *„The evidence found must be preserved as close as possible to its original state. Any changes made during this phase must be documented and justified.“*

Which clarifies the importance of integrity and authenticity when handling evidence.

This is followed by the *Analysis of Evidence* which includes the extraction of relevant information and the reconstruction of events. Again, this procedure is aimed at supporting a judicial proceeding as it includes the step of *Presentation of Evidence*. Here, the evidence is presented in a comprehensible manner to people with non-technical backgrounds. This could include judge or jury. But while not explicitly stated, this could also include managers which might or might not make additional resources available in order to prevent future incidents.

2.3.1.3 Incident Response & Computer Forensics - 2nd Edition ([MPP03])

This work (see [MPP03]) focuses on the topic of incident response. It uses computer forensics as a tool used to support incident response. Incident response describes the processes handling

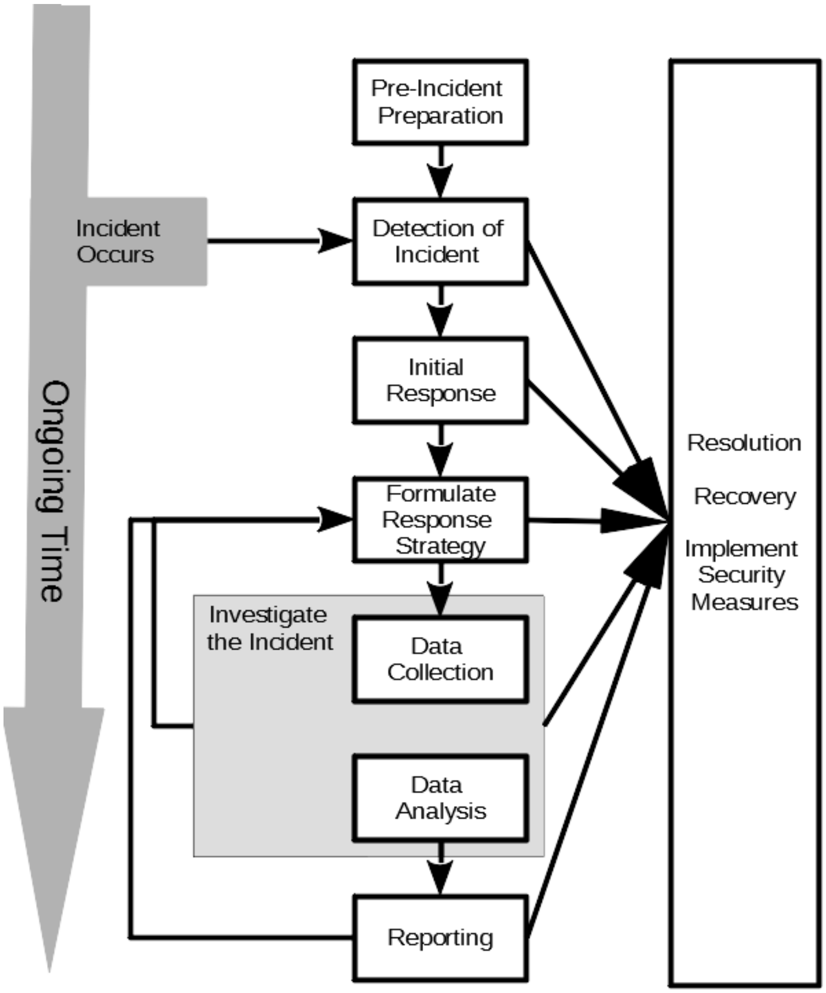


Figure 2.6: Incident Response Process according to ([MPP03])

a computer security incident. In order to provide clarity on this term, the definition from [CMGS12] for security incident is used:

> *„A computer security incident is a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices.“*

This work shows the importance of incident response and computer forensics going hand in hand (at least in **IC1** - see Section 2.1.2). Here, the forensic investigation is started due to some misbehavior of a computer system. Such misbehavior could be a security incident (or a system failure). The incident response process is also started after a security incident is noticed (or due to a system failure which is mistaken for a security incident). Additionally, computer forensics is used for the prevention of future incidents (just like the second usage of Computer forensics presented in [Bra01]). Hence, computer forensics can be considered a tool usable during incident response.

The author of [MPP03] proposes a rather complex incident response process as seen in Figure 2.6. The process consists of eight phases. It begins with a *Pre-Incident Preparation*, which includes the steps necessary to detect an incident in the first place. When an incident occurs, it is hopefully detected thanks to this preparation. This step is covered by the phase *Detection of Incident*. After the incident is detected, an *Initial Response* takes place. In this phase, the severity of the incident is rated. This includes achieving a preliminary understanding of the dangers posed by the incident. This is done, for example, in terms of financial damage. Such an assessment has an influence on the amount of resources allocated to the following phases, as will be established during the *Formulate Response Strategy* phase. In this phase, a decision is made if a forensic investigation is used to acquire further details about the security incident. If a forensic investigation takes place, the investigation consists of *Data Collection* and *Data Analysis*. This phase is summarized as *Investigate the Incident*. The process also includes a *Reporting* phase, but it is noteworthy that this process is more aimed towards keeping businesses running and secure than to enable court proceedings. In this sense, the process is aligned with Investigate Context 1. The process ends with *Resolution, Recovery, Implement Security Measures*. This phase is concurrent with the other phases taking place after the security incident occurs. It describes the measures taken to resolve and recover from the security incident and the implementation of security measures in order to prevent future exposure.

This work is a prime example on how forensic investigations can be used outside of court proceedings. Here, a corporate background is addressed. A focus on business continuity not present in the models geared towards court proceedings is embedded in this model for incident response.

2.3.1.4 Digital Evidence and Computer Crime - 2nd Edition ([Cas04])

The forensic process model presented in [Cas04] encompasses twelve steps to be performed during a forensic investigation.

In this model, a forensic process starts with *incident alerts or accusation*. Hence, this forensic model does not include a pre-incident step.

The next three steps (*Assessment of worth*, *Incident/crime scene protocol* and *Identification or seizure*) cover the initial response to the beginning of an investigation. Here potential sources for forensic traces are identified and the physical access to them is ensured.

The following three steps (*Preservation*, *Recovery*, *Harvesting*) describe the steps necessary to gather and preserve this data. These steps also cover the reconstruction of deleted data. During these three steps, the amount of data under consideration rises.

The succeeding three steps (*Reduction*, *Organization and search* and *Analysis*) cover the extraction of meaningful information from the data gathered and preserved in the preceding steps. One could conclude that during these three steps, the amount of data under consideration is reduced to the amount relevant for the investigated incident or accusation.

As in many other process models the forensics process is concluded by a presentation of the results to interested parties. This presentation consists of two distinct steps (*Reporting* and *Persuasion and testimony*).

The twelve steps of the forensic process can roughly be grouped into four categories (and one additional for the trigger of the forensic investigation in form of *incident alerts or accusation*. While [Cas04] does not conduct such a grouping, it seems useful for the sake of clear arrangement.

Concurrent to these twelve steps the model also includes a step handling *Case management*. This step is concurrent with all other activities and runs in parallel to them. In essence, this step forms a framework around the entire forensic investigation. This step is concerned with documenting the performed actions and providing a chain of custody for the evidence gathered.

Additional comments on this forensic model by the same author (see [Cas11]) clarify that it might be necessary to perform certain steps more than once and at different times during the forensic investigation. Hence, the sequence seen in Figure 2.7 can only be seen as an approximation of an ideal and simple investigation. In contrast to grouping the twelve steps into four categories, [Cas11] suggests to simplify the model by treating *Recovery*, *Harvesting*, *Reduction* and *Organization and Search* as subcomponents of an *examination* step.

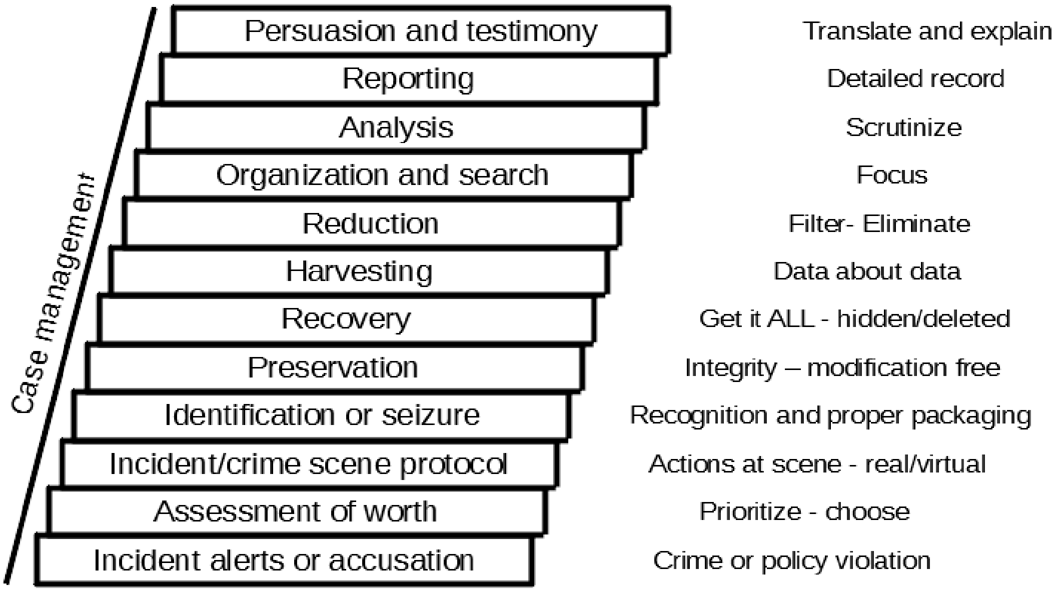


Figure 2.7: Investigative process model according to ([Cas04])

This forensic process model has a focus on judicial proceedings. However, the terms used point towards a dual use. For example, the forensic process in this model starts either with an accusation (related to legal proceedings) or with an incident alarm. The same goes for the third step which is described as *Incident/crime scene protocol*.

2.3.1.5 A Common Process Model for Incident Response and Computer Forensics ([FS07])

This work (see ([FS07]) aims at unifying the fields of incident response and computer forensics. It shows the relationship between these two fields and questions whether the separation between these two topics makes any senses at all. According to [FS07] the main difference between these two fields is the setting of different priorities. Hence, this model proposes a unifying view on incident response and computer forensics and suggests a common model (in the following referred to as *The Common Model*).

For the respective fields, the model relies on input from [Cas04] (see Section 2.3.1.4) and [MPP03] (see Section 2.3.1.3) as fundamentals for the Common Model. The Common Model is shown in Figure 2.8.

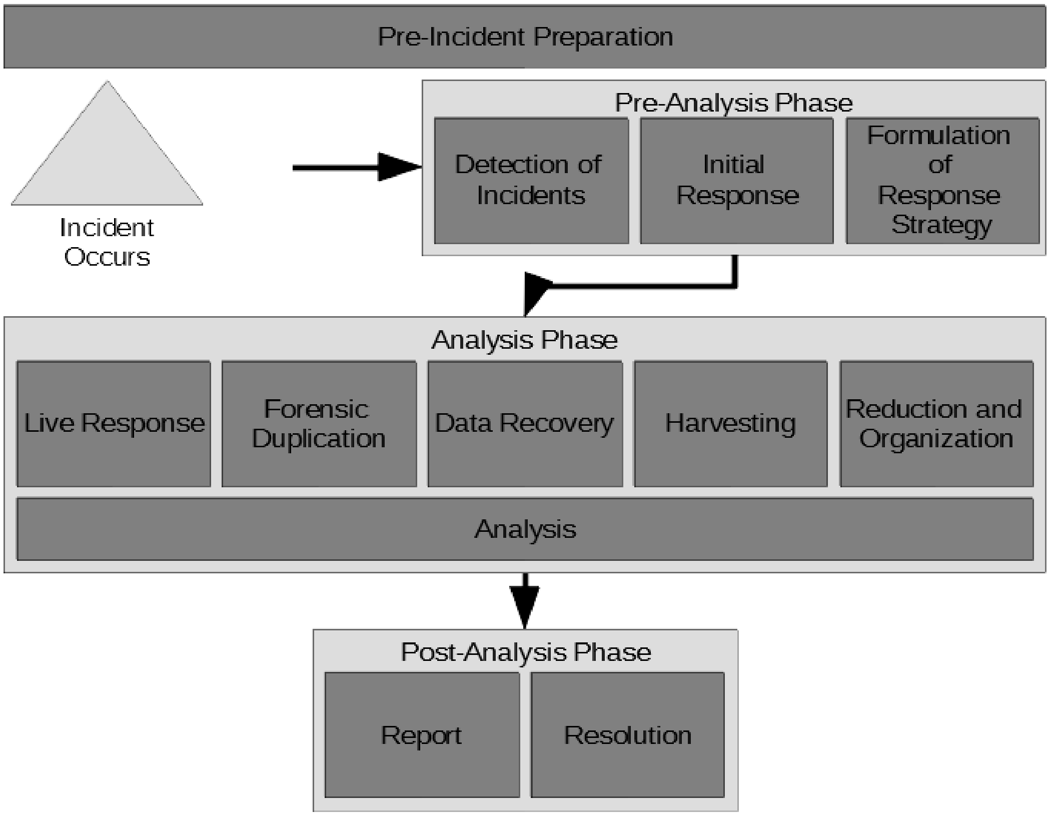


Figure 2.8: A Common Process Model for Incident Response and Computer Forensics according to ([FS07])

The model consists of three principal main phases: *Pre-Analysis Phase*, *Analysis Phase* and *Post-Analysis Phase*.

In this model, the process actually begins before an incident occurs with a *Pre-Incident Preparation*. This preparation aims at making the organization and personnel ready to respond to an occurring incident. It aims at establishing incident response procedures in order to facilitate an efficient response. This preparation does not refer to a specific incident but is done without a specific cause.

Activities specific to a given incident begin after the incident occurs. The *Pre-Analysis Phase* covers the *Detection of Incidents*, the *Initial Response* and the *Formulation of Response Strategy*.

The *Detection of Incidents* triggers the entire process. It deals with the detection of a symptom of an unauthorized or unlawful action. This can either be detected by a person or a security mechanism. This detection then leads to an *Initial Response* which includes the collection of information in order to confirm or discard the suspected incident. This step is clearly geared towards providing incident response by focusing on keeping harm from the organization. While the description of this step states that every action taken has to be documented, compromises in terms of the forensic process are accepted here. During the *Formulation of Response Strategy* the impact of the incident at hand is judged and a decision on how to react is made accordingly. This decision is based on factors like the criticality of the impacted systems to the business process, potential perpetrators, the apparent skill of the attacker, the downtime caused by the incident or monetary loss. Aside from these factors already presented by [FS07], the apparent resources of the attacker to perform the attack and the resources of the defenders to perform a meaningful recovery also plays a role. The decision whether to perform a full-scale forensic investigation is also made at this step.

The *Analysis Phase* covers the forensic analysis of the compromised computer systems. According to the decisions made during the *Formulation of Response Strategy*-step, this analysis can vary in scope and extent. The *Live Response*-step covers the collection of volatile memory which would be lost after the system is switched off. The Common Model states, that the data collection activities during this step should be performed in a way to alter the system as little as possible in order to reduce the impact on the collected evidence to a bare minimum.

Copies of all non-volatile memory are created during the *Forensic Duplication*-step while maintaining a chain of custody.

Hidden, deleted, damaged or data which is otherwise not available is reconstructed during the *Data Recovery*-step.

The *Harvesting*-step describes the gathering of metadata concerning the preserved material. Metadata is data about data, like the file names, file types or time stamps for file access.

These three steps align with the (*Preservation*, *Recovery* and *Harvesting*-step from the [Cas04]-model for the forensic process (see Section 2.3.1.4) and increase the amount of data available for analysis.

The amount of data is reduced during the *Reduction and Organization*-step where data which is identified to be irrelevant to the case is dropped. This identification can only be based on the metadata extracted during the *Harvesting*-step and the hypothesis formulated during the *Initial Response*-step. The model states the importance of documenting every performed action precisely and immediately.

The *Analysis Phase* is followed up by the *Post-Analysis Phase*. This phase covers the activities after an analysis took place. These contain the creation of a written report during the *Report*-step. According to [FS07] this report aims at presenting the evidence in a manner that meets the legal standards of admissibility in courts. Hence, it has to rely on the documentation of the analysis and the performed actions.

The model further contains a *Resolution*-step which describes steps to solve the problem which caused the incident. This could include taking measures in order to close potential vulnerabilities.

This step is something rarely included in forensic process models and really demonstrates the dual use of the Common Model as a model to integrate incident response and Forensic Investigations.

The activities required for incident response frame the forensic process and embed forensics as a tool for detailed analysis of the incident in question. This has some impact on the quality of the forensic analysis of course. With the priority being incident response during the first steps of the model, data and potential evidence will be altered by the first response.

2.3.1.6 Digital Forensics ([Arn17])

A forensic process with five distinct phases is proposed in [Arn17]. This work also gives a summary on the characteristics of forensic process models. This is noteworthy and hence is repeated here as useful criteria to measure all forensic process models against it. According to the summary in [Arn17], a forensic process model should:

> *) *„Adhere to the leading practice and theory from traditional forensic investigations.*
>
> *) *Be easily adaptable and practically oriented, and support traditional investigation steps.*
>
> *) *Remain independent of technology, product, and procedure.*
>
> *) *Applicable to law enforcement, corporate and incident investigations.*
>
> *) *Support forensic reconstructions.*
>
> *) *Improve proper and efficient identification of facts.*
>
> *) *Limit unnecessary generation of new artifacts.*
>
> *) *Use and support forensic science and technology, and the results must be usable in a court of law.*
>
> *) *Must be capable of being scrutinized by the judiciary if the case is presented in court.“*

This summary states that any forensic investigation should be performed in a manner in which the results could be presented before a court of law. However, it also states that forensic investigations can be performed in other settings, namely corporate or incident investigation. These corporate and incident investigations are similar to the scenario presented in Section 2.3.1.3.

The forensic process is shown in Figure 2.9 and constitutes five phases. Formal definitions for these phases are found in [Arn17] and presented here as a foundation for a detailed discussion:

> - *„The Identification Phase: The task of detecting, recognizing, and determining the incident or crime to investigate*
> - *The Collection Phase: Collection of data from digital devices to make a digital copy using forensically sound methods and techniques.*
> - *The Examination Phase: Preparation and extraction of potential digital evidence from collected data sources.*
> - *The Analysis Phase: The processing of information that addresses the objective of the investigation with the purpose of determining the facts about an event, the significance of the evidence, and the person(s) responsible.*

- *The Presentation Phase: The process by which the examiner shares results from the analysis phase in the form of reports to the interested party of parties.“*

This understanding of the forensic process begins with the detection of an incident and the decision whether to investigate made in the *Identification Phase*. This includes the potential decision to not investigate at all, if the incident responders hypothesis concludes that no criminal laws are violated and an investigation would not be economically feasible. This phase also covers the identification of digital devices which might offer digital evidence.

The *Collection Phase* covers the collection of traces from the digital devices while taking the concept of the *order of volatility* into account. This concept was already introduced in [Bra01] (see Section 2.3.1.2) but is used here in more detail. Furthermore, [Arn17], gives a definition for the term:

> *„Definition [...] Order of Volatility: Prioritization of the potential evidence source to be collected according to the volatility of the data.“*

Hence, the concept states, that, generally speaking, the most volatile data should be acquired before less volatile data. In general, data which is only present within the volatile part of the system memory is more volatile than those present in non-volatile memory.

In the *Examination Phase*, a further analysis is prepared by extracting potential forensic evidence from the collected data sources. This could be the extraction of files from storage media or the reconstruction of images from raw data streams. This enables the possibility to put all these potential traces into context during the *Analysis Phase*. In this phase, the various extracted elements are reviewed in order to determine if they carry some value for the event reconstruction and as potential evidence. As a curious fact, this model includes the tying of a specific person (or a group of persons) to the events investigated during the forensic process. This is surprising, since the problems with linking digital evidence to actual people have not been reduced since [Bra01] pointed out the difficulty with tying digital transactions to actual persons (see Section 2.3.1.2). The process ends with the *Presentation Phase* which encompasses the presentation of the investigation results. In this case, this does not necessarily mean a presentation in a court setting. The complete process aims at maintaining a chain of custody to increase the value of the evidence.

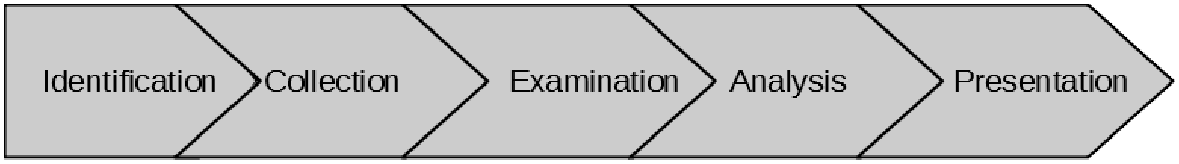


Figure 2.9: The digital forensics process according to ([Arn17])

2.3.2 Comparison of Forensic Process Models for Desktop IT

After the preceding section gave an overview on various forensic process models, comparing and understanding the development taking place inside the field of computer forensics is possible.

The emergence of computer forensics began with understanding if digital objects can constitute forensic evidence at all. This notation has been discussed in [Pol95] (see Section 2.3.1.1). Here, the various challenges unique to computer forensics are discussed in a concept referred to as *The Digital Paradigm*. These stem from the fact that digital evidence is encoded by file formats and the physical representation of the various bits on the specific carrier media. Here, **IC2** is addressed. Computer systems are simple tools used by suspects to perform actions that could also take part in the analogue world, like writing documents or transferring money.

The idea that the system itself might be the target of an attack arose soon ([Bra01] - see Section 2.3.1.2). Here, computer forensics serves a dual purpose - supporting investigations in **IC2** but also supporting the security of a computer system by investigating potential system compromise (addressed by **IC1**).

While [Pol95] discusses how evidence gathered from computer systems could be used in legal proceedings, [Bra01] gives a more practical guide on how to gather this evidence. This includes advice on some specific factors. A prime example is the suggestion to keep detailed logs of any action performed during the investigation. Another factor is the introduction of the *order of volatility* which is concerned with the differing volatility of certain pieces of evidence and the resulting need to secure some evidence at the earliest possible moment before it is unavailable or outdated by the systems operation.

However, these two early works ([Pol95] and [Bra01]) are focused on supporting judicial proceeding in court settings.

A slight shift in focus occurred when the relationship between computer forensics and incident response became a relevant topic. In [MPP03] (see Section 2.3.1.3) the main focus is on handling computer security incidents. Here computer forensics is only a tool in order to support incident response by analyzing the actions of an attacker with the ultimate aim of preventing future attacks. In this case **IC1** is addressed. A major difference is the fact that a forensic investigation is not necessarily seen as supporting a legal proceeding into some criminal action in this case. Rather a forensic investigation might also be triggered by a system misbehavior caused by a system failure.

Unifying computer forensics and incident response is the goal of the Common Model from [FS07] (see Section 2.3.1.5). This model has a stronger focus on computer forensics and tries to embed forensic investigations into the incident response process. The model clearly states that performing a comprehensive incident response process affects a forensic investigation by altering the system state and potential traces. Here, a compromise between the needs of effective incident response (the ability to alter the system in order to prevent further damage or compromise) and the forensics (unaltered traces) has to be found and the model clearly tends to focus more on incident response.

The process models which include computer forensics into incident response address **IC1**.

More recent examples of computer forensic process models like [Cas04], [Cas11] and [Arn17] address both **IC1** and **IC2** (see Section 2.3.1.6 and Section 2.3.1.4 respectively). These models postulate that the evidence gathered during the forensic process has to adhere to standards for court admissibility but that the forensic process itself might also be used in corporate settings where a judicial proceeding is not the ultimate aim. These models also take a closer and more detailed look on how the evidence has to be gathered and subsequently analyzed.

Aspects like the *order of volatility* are discussed in a more detailed and systematic manner in [Arn17].

Although these process models have a different focus and see different scenarios for the use of forensic investigations into computer systems, they carry many similarities. An overview of these factors is given in Table 2.1. Here the focus of the various models, the number of phases or steps, the inclusion and type of preparatory activities and the presence of a presentation phase are shown.

2.3.2.1 Phases of the Forensic Investigation

All these forensic process models split the forensic process into various phases or steps of activities that fulfill a given purpose. Most models have four or five phases, with the exception of the more granular model in [Cas04].

All these models, except the Common Model (see Section 2.3.1.5), have a phase in which potentially interesting evidence is identified. This phase is referred to as *Identification (of Evidence)*. In this phase the potential sources of evidence are listed. Some models (namely [Bra01], [FS07] and [Arn17]) differentiate between the evidence present in volatile memory and those present in non-volatile memory. This differentiation is implicit if a model explicitly addresses the *order of volatility* or includes different steps to address volatile and non-volatile memory respectively during the forensic process (see the next paragraph for a more detailed view on the gathering of evidence from volatile in contrast to non-volatile memory). If the model is aware of an *order of volatility* the decisions in which order these traces are gathered is determined during this step.

The next phase is referred to as *Acquisition* or *Preservation of Evidence*. This phase covers the actual gathering of the forensic traces identified in the preceding step. The term *Preservation of Evidence* implies more than just the actual gathering of the forensic traces. It implies activities necessary to ensure the authenticity and integrity of the gathered evidence, or as [Bra01] describes it, the *evidences authenticity and veracity* (see Section 2.1.4). Maintaining a *chain of custody* is proposed in [FS07] and [Arn17] in order to maintain the integrity and authenticity of this evidence. Such a chain of custody documents all actions performed on a given piece of evidence in order to enable the potential reconstruction of the performed actions. This is also included in [Cas04] in which a concurrent phase of *case management* is performed in parallel to all other phases. This phase covers the documentation of the performed actions and hence serves to ensure authenticity and integrity of the gathered and processed evidence.

After the evidence is gathered and preserved, it is analyzed. The discussed models describe this step as *Evaluation* or *Analysis (of Evidence)*. The most detailed look on this step is given by the detailed model presented in [Cas04] where the steps of *Recovery*, *Harvesting*, *Reduction*, *Organization and search* and *analysis* roughly correspond to the activities described to happen within the scope of the other forensic process models. Here, *Recovery*, *Harvesting*, *Reduction* describe activities necessary to perform an analysis on the full data included in reconstructed or converted files. These steps can be described as preparation for the actual analysis. This can also be seen in the process model presented by [Arn17] where this step is divided into *Examination* and *Analysis*. In this case, *Examination* corresponds to *Recovery* and *Harvesting*. A more detailed division of this phase can also be found in [FS07]. The outline of steps here closely follows those postulated by [Cas04].

Table 2.1: Comparison of Forensic Process Models

Name			
Focus	Number of phases or steps	preparatory activities	Presentation Phase
Noteworthy concepts			
Computer Forensics - an approach to evidence in cyberspace ([Pol95])			
IC2 Legal proceedings	4 phases	none	yes
The Digital Paradigm			
Collecting Electronic Evidence After a System Compromise ([Bra01])			
IC2 Legal proceedings **IC1** Future prevention	4 phases	none	yes
Order of Volatility motivated			
Incident Response & Computer Forensics (2nd Edition) ([MPP03])			
IC1 Incident response	8 phases 2 of them cover the core forensic investigation	Establish methods to detect an Incident	yes
Digital Evidence and Computer Crime (2nd Edition) ([Cas04])			
IC2 Legal proceedings **IC1** Incidents	12 steps Concurrent *case management*	no	yes
Concurrent *case management*			
A Common Process Model for Incident Response and Computer Forensics ([FS07])			
IC1 Incident response	12 phases 6 of them cover the core forensic investigation	Establish procedures and train personnel for Incident Response	yes
Live Forensics and *Forensic Duplication* as separate steps			
Digital Forensics ([Arn17])			
IC2 Legal proceedings **IC1** Incidents	5 phases	no	yes
Order of Volatility discussed in detail			

The presented forensic process models also include the *Presentation of Evidence* as the last step of the forensic process. This step encompasses the presentation of the collected evidence and the conclusions made during the analysis in front of a specific audience. For the process models aiming towards the use in legal proceedings this audience is a judge or a jury. Models geared solely towards the use in legal settings refer to this step as *Admission as Evidence* (see Section 2.3.1.1 - [Pol95]). For those models with a corporate focus - or those geared more towards incident response - this audience might be a team of technical experts tasked with preventing further security incidents. Including such a phase seems necessary as the guidelines for the admissibility of evidence place a high value on expert testimony in order to judge the evidentiary value of the evidence (see Section 2.1.7).

2.3.2.2 Starting Point of the Forensic Investigation

The question when a forensic investigation is triggered is important since it impacts and defines the earliest stages of a forensic investigation. In those models geared towards the use in legal proceedings, the forensic process begins with a legal accusation. This implies a strong suspicion which is then investigated. Such instances mostly pertain to **IC2**.

Another angle can be found in those process models which mostly focus on incident response (specifically [MPP03] and [FS07]). These models embed forensic investigations into the incident response process as tools in order to reconstruct the event which led to the incident in question. Here, primarily **IC1** is addressed. In order to trigger a reaction to an incident (and a potential forensic investigation), an incident must be detected at first. The incident-specific process starts with this detection of an incident (a security breach or an unauthorized access). However, especially [MPP03] includes a *Pre-Incident Preparation Phase* which covers the activities undertaken in order to be able to detect such incidents. This includes the installation of security measures which monitor the system and are able to produce alerts. This process is not incident-specific but rather more general. It might be caused by the results of a preceding investigation which led to the implementation of this new detection mechanism.

More recent models (namely [Cas04] and [Arn17]) are open to both paths. Here an investigation can be triggered by a specific accusation (primarily in **IC2**) or by the detection of a potential security incident (primarily in **IC1**).

How an investigation is triggered has some implications for the forensic process. Especially for the *Identification* phase where the potentially relevant pieces of evidence are identified. Depending on the *Investigative Context*, certain pieces of evidence are of higher interest than others. For example, a specific accusation in **IC2** could trigger an investigation into financial wrong-doing. Here documents like text files or spreadsheets are of interest. If the investigation is triggered by a suspected security breach, access log files or information about network connections would be of a higher relevance.

In those investigations triggered by a suspected security breach, the *Pre-Incident Preparation Phase* has an impact on the amount of available traces. Security measures installed to detect certain security breaches also provide data which might be of relevance to a forensic investigation. This is especially true for the understanding of the *Pre-Incident Preparation Phase* as used in [MPP03]. Here this phase includes the implementation of detection mechanisms. While the fact that the data provided by these mechanism is useful for an investigation is not explicitly mentioned by [MPP03] it is a rather obvious conclusion.

2.3.2.3 *Live Forensics* and *Post-Mortem Forensics*

Some of the presented models (specifically [Bra01] and [Arn17] include the concept of an *order of volatility*. This concept describes the fact that some potential evidence might only be available for a limited amount of time or might change quickly. This is especially true for potential evidence which is only available within the *Volatile Memory* of a computer system (see Section 3.1.2.5). Such evidence can only be gathered before the system is shut down and the contents of the volatile memory are lost.

When the data collection is performed while the system is still active this is referred to as *Live Forensics*. Performing *Live Forensics* enables the possibility to gather evidence which would otherwise be lost. However, it causes a negative impact of Evidence Dynamics. In the simplest case, the system state will change during the data collection and might not be entirely consistent. Some data is also only available by starting processes in order to query them. Doing so changes the system state inevitably. When an Investigation is performed with **IC1** in mind, an attacker might still be active in the system and use the time needed to perform *Live Forensics* to cause further damage or to cover potential tracks, thereby reducing the amount of potential evidence.

A step to address the potential use of *Live Forensics* in the forensic process is included in the Common Model from [FS07]. The question on whether to perform *Live Forensics* is a central question of the forensic investigation and greatly depends on why the forensic investigation was started. If the investigation was started from a legal accusation in **IC2**, *Live Forensics* would generally speaking not offer any added benefits with the potential exception of the relevant text documents or spreadsheets being only available in a volatile portion of the memory, like a memory disk. If the investigation is started from a potential security incident (**IC1**), information like the current state of network connections or active processes are only available by performing *Live Forensics* on the contents of the *Volatile Memory*.

Performing only *Live Forensics* is also a possibility if the system in question is not supposed to be shut down for a forensic analysis. This might be especially true for potential security incidents where business continuity is deemed a priority. In this case, Live Forensic might be the only tool available for the forensic investigator in order to gather data.

The forensic models addressing court cases (primarily in **IC2**) assume that the system in question will be shut down for data gathering however. Here, the non-volatile memory (mass storage) can be duplicated and used for further investigation. This process is known as *Post-Mortem Forensics*. A forensic sound copy of the mass storage can ensure authenticity, integrity and a chain of custody. Working on the duplicate in a read-only manner enables the repeatability of the performed steps not possible in *Live Forensics*. Here, evidence dynamics is a minor problem. Additionally, some of the problems mentioned for *Live Forensics* are not relevant here. If the system is immediately disconnected from the power supply an active attacker might not have any additional time to cover potential tracks.

As these two means of gathering potential evidence differ wildly in their impact on the forensic process the Common Model from [FS07] uses two different steps to describe these processes. These are referred to as *Live Forensics* and *Forensic Duplication*.

2.3.2.4 What Makes a Good Forensic Process Model

The preceding paragraphs summarized the properties of selected forensic process models used to perform forensic investigations in (potentially networked) Desktop IT systems. These properties serve as a foundation to establish an understanding of what a good forensic process model should include. This can be broken down into two major aspects. One aspect is the aim of the process described by the model and the second is the process itself. Since the forensic process is aimed towards achieving this specific aim, both aspects have an impact on each other. At first the aim of the forensic process will be discussed before the process to achieve this aim is explored.

Aim of a Forensic Process

The general aim of a forensic process is discussed in Section 2.1.1 as **event reconstruction**. Additional clarity is provided by Section 2.1. Here, the term **Digital Forensics** is defined specifically as:

> *The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.*

This definition gives a brief overview on how the forensic process has to be performed and identifies the aims of the forensic process in greater clarity. According to this definition, the aim of digital forensics is *the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.* The concept of *Investigative Contexts* is introduced in Section 2.1 in order to describe these two distinct aims a forensic process can have. The discussion in the previous sections showed that different forensic process models are indeed designed with one of these two aims in mind. They address either **IC1** (*Investigations into unwanted behavior of the system*) or **IC2** (*Investigations into behavior which has been carried out with the use of computer systems and is subject to a potential court case*). As shown in Section 2.3.2, investigations in **IC1** are often performed to support incident response.

In each case, both contexts need to be addressed by a forensic process model. The fact that the practitioner-driven forensic process models address **IC1** ([MPP03] in Section 2.3.1.3), **IC2** ([Pol95] in Section 2.3.1.1) or both of these contexts ([Bra01] in Section 2.3.1.2, [Cas04] in Section 2.3.1.4) shows that both of them are relevant for the field of digital forensics. Further support for this fact comes from the German crime statistics from 2018 [BKA18]. While most of the cases from the category of *computer crime* fall into the category of *computer fraud* (around eighty percent) which is covered by **IC2**, a relevant amount of cases pertains to **IC1**. *Data manipulation*, *computer sabotage* and *theft of data* are covered by **IC1** and amount to a total of around ten percent of the registered cases in these crime statistics.

Challenges for the Forensic Process

As discussed before, a forensic process model has to serve as a guideline on how to achieve the aim(s) of the forensic process while taking various challenges into consideration. These challenges are discussed in the preceding sections of this chapter and a short overview is provided by the introductory paragraph of Section 2.3. These challenges are repeated here for better readability. Furthermore, they are defined as *Process Challenges* as they represent general challenges for the forensic process. These four *Process Challenges* are:

- **Process Challenge 1 (PC1)**:
  integrity and *authenticity* of traces (see Section 2.1.4)
- **Process Challenge 2 (PC2)**:
  error, uncertainty and loss (see Section 2.1.4)
- **Process Challenge 3 (PC3)**:
  constraints for the collection and use of evidence (see Section 2.1.6)
- **Process Challenge 4 (PC4)**:
  legal requirements for the admissibility of evidence in court (see Section 2.1.7)

PC1 and **PC2** correspond to the fact that the forensic investigation should aim to make a conclusion based on reliable evidence with a high evidentiary value. This is true whether the investigation takes place in **IC1** or **IC2**. Such evidentiary value is increased when the integrity and authenticity of the traces are beyond doubt. A forensic process can undermine the integrity and authenticity of the forensic traces or maintain it, addressing **PC1**. Knowledge about error, uncertainty and loss associated with a given forensic method are necessary in order to determine the evidentiary value of the given evidence provided by this method with any certainty. This is also emphasized by the inclusion of *Admissibility Factor 4* in the considerations on the factors required for forensic evidence to be admissible performed in Section 2.1.7. This factor states that the rate of the error for a technique has to be known before any evidence resulting from the use of this technique is admissible before court.

PC3 addresses the constraints on the use and collection of evidence during a forensic investigation. This expands on the discussion on legal constraints performed in Section 2.1.7). A primary concern is personal data affected by the forensic investigation. This is usually the case for a forensic investigation in **IC2**. Although special privileges might be granted here by merit of these investigations being performed by law enforcement personal, constraints still remain. This includes the handling of evidence in a way that restricts access to it. In the case of personal data, this is mandated by the legal regulations (as discussed in Section 2.1.7). The same need applies for business secrets which might be present within the collected evidence. Such a restriction includes limiting the amount of persons with access to said data and could additionally include ensuring the confidentiality by employing encryption.

While every forensic investigation should come to a conclusion based on reliable evidence the standards for a criminal investigation are even higher. Such a criminal investigation might take place in either **IC1** or **IC2** and some investigations might even combine these cases (for example if an attacker gains unauthorized access to a privileged account in order to defraud customers). These standards are represented by **PC4** and are discussed in Section 2.1.7 leading to the establishment of eight distinct **Admissibility Factors**. While these factors are not strictly required for investigations outside the domain of criminal law and court, they still prove useful to increase the evidentiary value of the evidence in question and hence should at least be used as guidelines in each consideration.

The Forensic Process Itself

The forensic process consists of several activities (as discussed in Section 2.1.3). These activities follow a logical and chronological sequence. Various activities require that other activities were performed before them. For example, a piece of evidence must be collected before it can be analyzed and it must be analyzed before it can be presented as evidence in front of a jury or a court. All forensic models discussed in Section 2.3.1 use phases or steps to describe this sequence. A closer look on how these models describe the sequence and the differences in including various activities is given in Section 2.3.2.1. However, it can be concluded, that structuring the forensic process into different phases and steps is both proven and practical. These phases should be structured along the logical sequence in which the activities of a forensic investigation are performed.

The starting point of a forensic process has an influence on which activities are included within this process. This aspect is discussed in detail in Section 2.3.2.2. This discussion concludes that some process models (mostly those geared towards the use of forensics within **IC1** and incident response) include a *Pre-Incident Preparation Phase* which covers activities performed to increase the ability to detect a security incident. These activities have to be performed by the system operators (or more likely by system administrators) regardless of specific incidents. As investigations in **IC1** are often performed by the system operators (or again, system administrators), a forensic process model should address the *Pre-Incident Preparation*. This is supported by the fact that the measures installed in order to detect security incident themselves might also create evidence of value for a forensic investigation. This includes, for example, the log files created by an IDS (*Intrusion Detection System*) during the detection of a system intrusion (which would qualify as an incident).

The various considerations on *Live Forensics* and *Post-Mortem Forensics* have been presented in Section 2.3.2.3. These considerations made clear that both these concepts have a potential use during a forensic investigation. Hence, a comprehensive forensic process model should be able to address these two concepts. These two concepts carry the ability to address certain pieces of forensic evidence. For example, *Volatile Memory* can only be obtained during *Live Forensics*. However, the use of each of these concepts comes with an associated cost the forensic process. In this example, the gathering of evidence from volatile memory would keep the investigated computer system active while an attacker could alter the non-volatile memory. Hence, a comprehensive forensic process model should provide guidance on which of these two concepts should be used in a specific case.

Summary

These considerations provide some clarity on what a good, comprehensive forensic process model should address. The points discussed in Section 2.3 provide the foundation to identify eight different *Forensic Process Model Criteria* which a forensic process model should ideally fulfill:

- **Forensic Process Model Criterion 1 (PMC1)**:
  Addresses **PC1** (integrity and authenticity of traces)

- **Forensic Process Model Criterion 2 (PMC2)**:
  Addresses **PC2** (error, uncertainty and loss)

- **Forensic Process Model Criterion 3 (PMC3)**:
  Addresses **PC3** (constraints for the collection and use of evidence)

- **Forensic Process Model Criterion 4 (PMC4)**:
  Addresses **PC4** (*Admissibility Factors*: legal requirements for the admissibility of evidence in court)
- **Forensic Process Model Criterion 5 (PMC5)**:
  Addresses **IC1** and **IC2**
- **Forensic Process Model Criterion 6 (PMC6)**:
  Includes the possibility of a *Pre-Incident Preparation*
- **Forensic Process Model Criterion 7 (PMC7)**:
  Addresses *Live Forensics* and *Post-Mortem Forensics* and includes some guidance to decide when to use which approach
- **Forensic Process Model Criterion 8 (PMC8)**:
  Structures the process based on the logical sequence of the investigation

These criteria are the result of the discussion in this chapter (specifically Section 2.1 and Section 2.3). **PMC1** to **PMC4** align with the *Process Challenges* discussed earlier in this section. **PMC5** to **PMC7** represent the entire diversity of the forensic process models compared in Section 2.3.2 and the different potential aspects of a forensic investigation these models cover. These include investigations into unwanted behavior of the system (either by malicious intent, accident or fault - **IC1**) and unwanted behavior conducted using computer systems (**IC2**). **PMC8** takes a special role in these criteria since it represents the primary aim of a forensic process model - to structure the forensic process in a manner which is supportive to the forensic investigator as discussed in Section 2.1.3 and Section 2.3.2.4.

Hence, **PMC1** to **PMC4** address the challenges associated with the use of specific methods while **PMC5** to **PMC8** address the overall structure of the forensic process.

2.3.3 State of the Art in Industrial Control Systems Forensics

The need for forensics in the ICS domain was explored in [FC08]. This publication gives some basic insights into the specific challenges of the ICS domain and proposes some mitigation strategies in order to increase forensic-readiness in the ICS domain.

Over the years, a few publications described case studies and exploration into the domain of ICS forensics. A selection of these publications and their contributions are presented here.

Incident Analysis & Digital Forensics in SCADA and Industrial Control Systems - [STM13]

[STM13] describes the basic structure of ICS and its components. This publication gives an overview on the steps which can be performed by a system operator before a security incident happens (a *Pre-Incident Preparation* as requested to be included in the forensic process as **PMC6** in Section 2.3.2.2). It further describes the nature of data available in common ICS:

> *„Log files constitute a critical source of evidence in a digital forensic investigation. In the traditional IT domain, keeping log files is usually enforced by security policies. However, in the field of control systems logging focuses mostly on the*

> *production monitoring and to support troubleshooting. Thus, even though control systems are engineered so that transactions and activity are closely monitored, resulting log files may lack valuable information required by an investigation [..]“*

Logging mechanisms in ICS focus on data about the underlying physical process (production monitoring in this quote), while other information is scarce. This is best exemplified by regarding the Historian, which is part of many employed ICS (see Section 4.2.2). Such a Historian is described in [ENI17] as:

> *„Historian: it is a high-capacity system designed to collect and store the logs generated by the readings and operations of the sensors, assets, alarms and other events generated by plant devices, part of the network.“*

It is noteworthy that the alarms mentioned in this description refer to the physical process. These alarms are triggered by certain properties of the physical process reaching certain points. For example, if a temperature sensor reports a value close to a certain limit in regards to the physical process such an alarm could be triggered. In short, the Historian is tasked with documenting the physical process. This is necessary in order to identify problems with the physical process or to monitor the performance indicators which could be necessary for the operators of the ICS in question.

[STM13] suggests establishing additional logging mechanisms in order to record more data which might be better suited to investigate a potential IT-security incident within the ICS.

It is a relevant question whether this would fall into **IC1** or **IC2** (see Section 2.1.2). Considering the usual operations with the ICS domain, these two contexts seem to be closely tied to classical IT. However, the reason to establish this additional logging mechanism is to provide additional data in order to identify and investigate misbehavior of the system by identifying wrong or faulty control messages. This aligns to **IC1**. This topic is explored in greater detail in Section 5.3.5.1.

An Architecture for SCADA Network Forensics - [KGC+06]

One such approach to include additional logging mechanisms was suggested in [KGC+06]. This publication discusses the introduction of Forensic Agents into the ICS architecture. These Forensic Agents access the communication within the ICS architecture. The Forensic Agents are supposed to be attached to Level 0, Level 1 and Level 2 of the PERA (see Section 2.2.2.1). Hence, the Forensic Agents would have access to the network traffic within the ICS-specific part of the overall network. The inclusion of such Forensic Agents aims at identifying wrong or faulty control messages and hence addresses **IC1**.

Towards a SCADA Forensics Architecture - [WDJC13]

[WDJC13] provides an useful overview on the topic of forensics in ICS environments. A forensic process for ICS is introduced and quoted here:

- *„Phase 1- Identification and Preparation*
- *Phase 2- Identifying data sources*
- *Phase 3- Preservation, Prioritising and Collection*
- *Phase 4- Examination*
- *Phase 5- Analysis*
- *Phase 6- Reporting and Presentation*
- *Phase 7- Reviewing results“*

This process is in line with the discussion on forensic process models designed for the use in Desktop IT (see Section 2.3.1). It does not include a *Pre-Incident Preparation Phase* and hence does not comply with **PMC6** as introduced in Section 2.3.2.4. Of high value is the identification of general challenges for the computer forensic process in ICS. These challenges are relevant for the course of a forensic investigation and hence are discussed. According to [WDJC13], these challenges are:

- *Live Forensics* and integrity of data
  ICS are not supposed to be shut off to perform a forensic operation. This is discussed in Section 4.3.1.

- Lack of compatible forensic tools for field devices
  This lack of tools can be seen in Section 4.1.2 and Section 4.2.2.

- Lack of forensically sound storage
  [WDJC13] states that the data storages usually available in ICS systems (Historians) are not forensically sound.

- Identifying data sources on a SCADA system
  According to [WDJC13] the identification of data sources in ICS is difficult due to the complex architectures. This thesis comes to a different conclusion however seeing a benefit in the complex, but quite similar architectures of ICS (see Section 4.2.2).

- Increase of sophisticated attacks
  The complexity of attacks in the ICS domain is in general high. This is confirmed by the discussion of attacks on ICS in Section 4.4.1.

These challenges closely align to investigations performed within **IC1**.

Developing Cyber Forensics for SCADA Industrial Control Systems - [SJaTW14]

An overview on various developments necessitating an increase of forensic ability in ICS is provided in [SJaTW14]. After an introduction on the topic of ICS it identifies typical attacks on ICS. These typical attacks include implementation-specific vulnerabilities but could be generalized. Indeed, Section 4.4.1 provides a more general discussion on the nature of current attacks performed against ICS. [SJaTW14] describes a forensic process based on the considerations proposed in [WDJC13]. The authors further argue for the creation of a forensic toolkit in order to perform investigations into ICS. The collection of tools proposed here is not tailored towards the use in ICS but consists of general purpose forensic tools. However, some of them are useful for the forensic process in the ICS domain. Overall, [SJaTW14] provides a good overview on the field of forensics in ICS.

Forensics in Industrial Control System: A Case Study - [VKL16]

Another introduction on the field of ICS is provided by [VKL16]. This publication further discusses differences in the understanding of security in the various different domains. For the forensic process itself, [VKL16] differentiates between *Network data acquisition* and *Device data acquisition*. A similar distinction is made in the [KDV15]-model for the forensic process with the introduction of the three *Data Streams* which cover *Network data acquisition* (*Communication*) and *Device data acquisition* (*Non-Volatile Memory* and *Volatile Memory*) (see Section 3.1.2.5). A case study on the use of forensics in a realistic ICS scenario completes this work.

Introduction to Network Forensics - [ENI19]

[ENI19] contains a training exercise which represents an attack on an ICS. This exercise contains baselining and an investigation into the attack (see Section 4.2.2.4). The exercise includes an introduction into ICS and confers a basic understanding of ICS architecture and processes. Furthermore, it gives an introduction into some of the protocols used in ICS.

2.3.4 State of the Art in Automotive Systems Forensics

This section gives an overview about the current applications of forensic science in automotive environments. This is not limited to computer forensics but the use of forensics in general. This is necessary since two topics which are strongly linked with the use of forensics in the context of vehicles do not touch the field of computer forensics directly. These two topics are crash reconstruction and stolen vehicle recovery. A recent addition is the usage of data collected and stored within vehicles to support forensic investigations. This section will take a look on these three topics and then concludes with a summary.

2.3.4.1 Crash Reconstruction

Crash reconstruction has a notable history in the automotive domain. Investigations into the cause of traffic accidents are an important tool used by insurance companies and police investigators. Since the 1990s, so called *Event Data Recorders (EDRs)* are common inside passenger vehicles (see [NHT02]). These EDRs record vehicle specific data in case of an accident. The specific implementations are vendor-specific and often include additional functionality or certain data sets. [Smi16] names *SDM* (Sensing and Diagnostics Mode) from GM[3] or the *RCM* (Restraint Control Module) of Ford[4] as historic examples. A recent survey on these vendor-specific EDRs was performed in [BBCS16]. The study states, that the most relevant data sets recorded by these EDRs are the *Vehicle velocity change* (ΔV) and the *Pre-Crash vehicle speed*.

While the inclusion of an EDR is not required by law in the United States, the majority of new vehicles already included them in 2000 according to [Gyo00]. Technical documentation ([Bos17] of the *Bosch CDR* EDR retrieval software[5] points to a significantly lower penetration of EDRs in the European market.

[3] https://www.gm.com/, 11/05/2020

[4] https://www.ford.com/, 11/05/2020

[5] https://www.boschdiagnostics.com/cdr/, 11/05/2020

Table 2.2: Required data elements for all vehicles equipped with an EDR from [NHT12]

Data Element	Recording Interval	Samples per Second
ΔV, longitudinal	0 to 250 ms or 0 to End of Event + 30 ms	100
Maximum ΔV, longitudinal	0 to 300 ms or 0 to End of Event + 30 ms	NA
Time, maximum ΔV, longitudinal	0 to 300 ms or 0 to End of Event + 30 ms	NA
Speed, vehicle indicated	-5.0 to 0 sec	2
Engine throttle, % full (or accelerator pedal, % full)	-5.0 to 0 sec	2
Service brake, on/off	-5.0 to 0 sec	2
Ignition cycle, crash	-1.0 sec	NA
Ignition cycle, download	time of download	NA
Safety belt status, driver	-1.0 sec	NA
Frontal air bag warning, lamp, on/off	-1.0 sec	NA
Frontal air bag deployment, time to deploy, driver	Event	NA
Frontal air bag deployment, time to deploy, passenger	Event	NA
Multi-event, number of event	Event	NA
Time from event 1 to 2	as needed	NA
Complete file recorded (y/n)	Following other data	NA

The US-American NHTSA (National Highway Traffic Safety Administration)[6] aimed towards a standardization of these EDRs and the interfaces used to access these EDRs. In 2012, the standardization effort of the NHTSA came into effect. This guideline (see [NHT12]) standardized the data sets recorded by these EDRs. Furthermore, the guideline states that data retrieval tools and/or methods have to be made commercially available by the specific manufacturers to crash investigators and researchers. In addition, the guidelines give requirements for the survivability of the data recorders in the case of crashes.

A listing of the required data elements for all vehicles equipped with an EDR is shown in Table 2.2.

This listing of the bare minimal data recorded by an EDR includes specific information about the state of the vehicle in the time span before and after a crash (the crash being referred to as *the event* in the context of [NHT12]). For example, in the case of the indicated speed, the value is recorded from 5 seconds before the accident onwards. This is the longest amount of time recorded before an incident. The longitudinal ΔV is recorded for 300 ms after the event or for 30 ms after the end of the event, whichever is shorter. Hence, the EDR is recording information about the physical state of the car for a very limited amount of time.

[6] https://www.nhtsa.gov/, 11/05/2020

Noteworthy in this set of recorded data is *Ignition cycle, download* since it does not refer to a data set recorded at the time of the crash but rather to a diagnostic value queried at the time the data is recovered from the EDR. This has an obvious impact on the reconstruction since a certain time span elapses between the crash and the gathering of this data.

A set of data elements required under specific conditions is also included in [NHT12]. This set includes information about the safety belt status, the deployment of specific airbags and the occupant size classification. It also includes the engine rpm and further details about the physical state of the vehicle like the lateral acceleration or the longitudinal acceleration at the time of the accident.

The specification ([NHT12]) puts great emphasis on ensuring that the recorded data will be available after the accident. This data survivability relies on resilience against the brute mechanical force of the impact. Requirements for crash tests are given. However, the specification is unconcerned with malicious manipulation of the recorded data. There are no required mechanisms to ensure integrity and authenticity. Basic mechanisms to increase integrity, like checksums, are not required to protect from accidental data manipulation caused by the force of the crash.

This clearly shows the focus of EDRs in supporting crash reconstruction. Indeed, the use of EDR data in reconstructing crash events is well researched (see [SDM08]) and the specification itself refers to the usefulness of EDRs in this aspect (see [NHT12] - section II D. in particular discusses NHTSA's reliance on EDR data for crash reconstruction and crash studies). The recorded data is helpful to deduce the physical state of the car right before and during the accident.

However, this set of recorded data is insufficient to deduce any information about the state of the vehicles electronic systems. There is no data which describes the state of the computing units or the signals communicated between them. While it is recorded if gas pedal or brake pedal were triggered right before an accident, it is not specified if this date is recorded at the pedals themselves (detecting the fact that the pedals have been triggered), the control signal generated by this actions (detecting the fact that a signal is transmitted on the communication network which conveys that one of the pedals is pushed to the attached ECUs) or at the actor performing the physical action (detecting the fact that the brake or the gas are indeed breaking or accelerating).

In normal operations there should be no distinction between these three potential approaches to record this single date. If the pedal (a sensor) is pushed, the communication network transmits a signal communicating this fact to the corresponding ECUs which then relay a signal to the brake or gas (an actor) which then performs the actions. However, in the case of an attack on the vehicles computer systems, this might not be the case. An attacker could break this causality, for example by inserting the signal of a triggered pedal directly into the communication. In order to be able to detect such an incident, additional data would be required. Hence, while the EDR greatly supports forensic investigations into vehicle crashes, its use for computer forensic investigations into vehicle misbehavior is limited.

Regardless of EDRs further attempts to acquire data relevant to the investigation of vehicle crashes is documented. [RA02] performed selective examinations of the readability of data storage in ECU (Electronic Control Units). In general, this approach is quite difficult and can only be achieved by using debugging interfaces if these interfaces are present and activated in the first place. More details on the use of these debugging interfaces are discussed in Section 4.1.3.3.

Considering the proceedings and aims of crash reconstruction, it is difficult to fit such a procedure into **IC1** or **IC2** (see Section 2.1.2). This complex topic is discussed in Section 5.3.5.1.

2.3.4.2 Stolen Vehicle Recovery

Various methods address the recovery of stolen vehicles. Different techniques are used in order to recover stolen vehicles. One technique is the inclusion of technology used to track the vehicles physical location inside the vehicle itself. These systems can be standalone solutions or integrated into technology packages.

Standalone solutions include the installation of dedicated GPS trackers. Various commercial manufacturers for such systems exist. Some of these solutions include complex background services for the automated notification of law enforcement personal (e.g. *LoJack*®)[7] while others only consist of the physical device which makes the current position of the vehicle available via web interface or application. Usually, the position of the vehicle for a given time span will be stored. This data might be useful for a forensic investigation, if the vehicle is somehow involved in a criminal endeavor. In the case of a stolen vehicle, the usefulness of this information is obvious (and a primary reason why such systems are included in the first place). But this information might also be useful during a trial to determine the whereabouts of a suspect at a given time. A more explicit example would be in the case where the vehicle was used as a getaway car. In this case, the parallels for **IC1** are obvious. Some tracking solutions are not primarily installed in order to track down theft but to monitor employees like delivery drivers. Here, the data might still be useful in order to investigate potential crimes performed using the vehicle or crimes which led to the theft of the vehicle. In each case, privacy concerns are a relevant factor in the use of this data (see Section 2.1.6). This data would provide little use in the case of the vehicle being the target of an attack on the Automotive IT (which would equate to **IC2**). However, it might be possible to notice a gap between transmissions of the current position if an attack on the Automotive IT would block or delay such transmissions for a short amount of time.

Other techniques to recover stolen vehicles rely on classical police work like interviewing potential witnesses or reviewing the data of Automatic number-plate recognition (APNR) systems employed in CCTV (closed-circuit television) or road-rule enforcement cameras.

2.3.4.3 Forensic Use of Data Collected and Stored Within Vehicles

A recent addition to the fields of forensics is the usage of data collected and stored within vehicles.

An overview of the potential value of this data and its sources is provided by [LEA16]. This paper identifies the embedded infotainment systems and mobile devices connected to these systems inside vehicles as the source of potential data. These embedded infotainment systems are referred to as In-Vehicle-Infotainment (IVI) in other sources and might be connected to an online system (like, for example, in the case of *BMW ConnectedDrive*[8]). These online

[7] https://www.lojack.com/, 08/05/2020

[8] https://www.bmw-connecteddrive.de/app/index.html#/portal, 08/05/2020

connected infotainment systems can also double as a tracking device for stolen vehicle recovery (see Section 2.3.4.2).

An overview on what kind of data can be expected to be found in these infotainment systems is also provided by [LEA16]. These findings are summarized here:

- phone book
- WiFi SSID
- logs of USB device connections
- mailbox
- GPS related information

Some of these findings could support a forensic investigation in **IC1**, where the vehicle infotainment system (and potentially attached mobile devices) performs the same task a Desktop computer would usually perform by providing communications used in an act which might lead to a forensic investigation. Here, the *phone book* or the *mailbox* might provide useful evidence. Of course, actions where the car is used for mobility during such an action would also fall under **IC1** and could profit from the availability of *GPS related information* or a list of *WiFI SSIDs* in order to collaborate the position of the vehicle.

However, the *WiFI SSIDs* or the *logs of USB device connections* could also help to identify connected networks or devices when investigating an attack on the Automotive IT. This aligns with **IC2**.

Further considerations on this topic can be found in [SCDLK19] which focuses on the communication of IVIs with the online systems of specific manufacturers. It does not reveal additional available data in relation to [LEA16] but shows that some of this data can also be found within the communication rather than only in the physical devices.

A test setup used to identify data sources as discussed in this section was proposed in [**KAH^{+}18**]. This publication describes the problems with the proprietary and closed source nature of these data sources. It suggests a manufacturer-independent setup to achieve a better understanding and increased transparency of Automotive IT systems for independent forensic investigators.

2.3.4.4 Summary of Current Automotive Systems Forensics

Forensics in the automotive environment is a relevant topic and some of the advances in this field can be of use to investigate incidents involving Automotive IT. The data currently collected in the automotive environment mostly supports investigations into **IC1** but some might be of use to investigate **IC2** as well. In general, three different forensic scenarios in the automotive environment can be identified.

Crash Reconstruction (see Section 2.3.4.1) is well-established and has reached a notable degree of maturity. Electronic Data Recorders (EDR) record information about the state of the vehicle during a crash in order to support this reconstruction. However, the data provided by these EDRs is not useful in order to investigate incidents using (**IC1**) or aiming at (**IC2**) Automotive IT.

In terms of the *Forensic Process Model Criteria* introduced in Section 2.3.2.4, crash reconstruction aims at addressing **PMC1**, **PMC2** and **PMC4** to enable the use of the investigation results in litigation. This use in legal proceedings also covers **PMC3**. **PMC5** might not be applicable here since the nature of the investigated incident is hard to fit in either **IC1** or **IC2** (as discussed in more detail in Section 5.3.5.1). Crash reconstruction relies on methods installed before an investigation becomes necessary and hence rely on a *Pre-Incident Preparation* as addressed by **PMC6**. **PMC7** and **PMC8** are not applicable here since the structure would involve traces not found within a computer system. While a well-defined process to gather, process and analyze these traces is necessary, the use of EDRs is merely a single step within this chain dedicated to gather, process and analyze a specific set of traces and does by no means cover the entire process of crash reconstruction.

The techniques used in Stolen Vehicle Recovery (see Section 2.3.4.2) address obtaining the physical location of a vehicle. This information could be used if a vehicle (including its Automotive IT) is used in an action leading to a forensic investigation. Such scenarios would pertain to **IC1**. In this case the use of a technical method to identify the location of a stolen vehicle would represent the use of a specific method during a forensic investigation. As such, only **PMC1** to **PMC4** are applicable. **PMC1** and **PMC2** are of relevance in order for the findings to be of any value. **PMC4** might be of relevance if, for example, a car theft is litigated. **PMC3** is either covered by the use in legal scenario or of no primary concern for the investigators.

The most comprehensive use for forensic investigations in Automotive IT can be found in research on using data from the In-Vehicle-Infotainment (IVI) during forensic investigations (see Section 2.3.4.3). This data is situated in the Automotive IT and might be relevant to scenarios in **IC1** in which the Automotive IT is used in the same way as Desktop IT during a criminal activity (that is for communicating threats or criminal conspiracy). However, the data collected in these systems might also include technical data like *logs of USB device connections* which could be used during an investigation into an attack directly aimed at the Automotive IT (**IC2**).

3. An Approach to Adapt Computer Forensics from Desktop IT to Automation and Automotive

This chapter consists of two major parts.

The first part (Section 3.1) discusses the development of the [KDV15]-model for the forensic process and how it addresses the various aspects of computer forensics discussed in the previous chapter. The [KDV15]-model is designed with the (potentially networked) Desktop IT domain in mind and is then used as a foundation for transferring knowledge from this domain to the ICS and Automotive IT domains for the remainder of this thesis.

This section describes the state of the forensic process model at the time of the publication of [KDV15] as the starting point of the adaptation performed in this thesis. Since then, various additions have been published. These include [**ALKD17**], [**ALK^{+}18**], [**AHKD19**] and most notable [Kil20] which explores the topics of error, loss and uncertainty (see Section 2.1.5) in great detail.

The second part (Section 3.2) of this chapter discusses the steps necessary to facilitate this transfer of knowledge. Here, a concept for the systematic examination of the nature of forensic investigations in the three covered domains is formulated. This approach compromises domain analysis, revisiting of current procedures and evaluation of the altered process.

3.1 The Creation of the [KDV15]-Model for Computer Forensics

This section discusses the creation and development of the forensic process model described in [KDV15] and how this model relates to the requirements for forensic process models supporting the forensic process as discussed in Section 2.3.2.

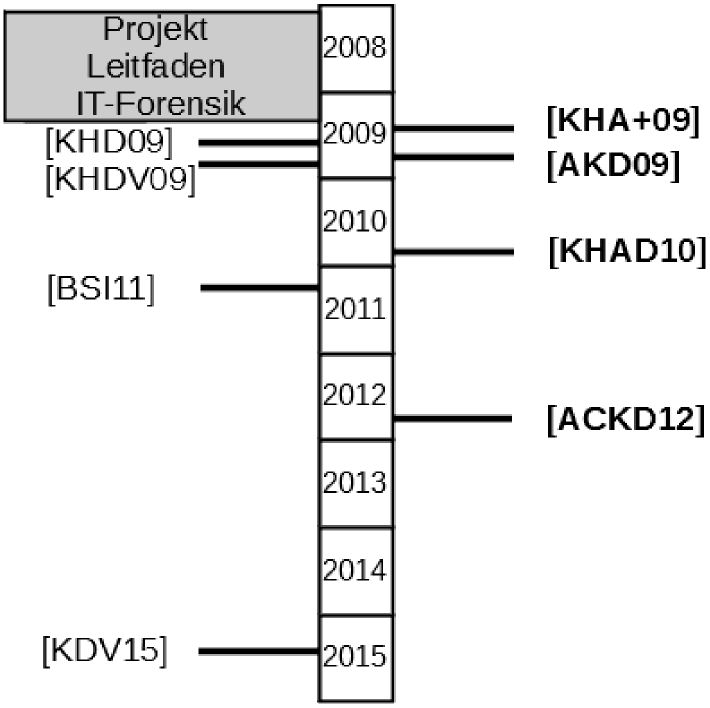


Figure 3.1: Milestones in the creation of the [KDV15]-Model up until the stage of the model used as the foundation for the adaptation performed during this thesis

The [KDV15]-model for the forensic process in (potentially networked) Desktop IT systems is based on the work published in [BSI11]. The *BSI Leitfaden IT-Forensik* (German for *„Guidelines for IT Forensics“*; The BSI is the *Bundesamt für Sicherheit in der Informationstechnik*, the German *Federal Office for Information Security*[1]) was published in its most recent version in the year 2011 (this is the version referred to as [BSI11]). However, the history of these guidelines and the forensic process model described therein began earlier and the model has been adapted and improved during later work.

During the research project *Leitfaden IT-Forensik*[2] (German for *„Guidelines for IT Forensics“*) substantial work towards the forensic process model was performed involving the author of this thesis. This work was accompanied by various publications ([**AKD09**], [**KHA$^+$09**], [KHDV09], [KHD09] and [**KHAD10**]) in order to develop and publish various aspects of the forensic process model. These publications were released during the course of the work on the research project *Leitfaden IT-Forensik*.

[1] https://www.bsi.bund.de/EN/TheBSI/thebsi_node.html, 05/05/2020

[2] https://forschung-sachsen-anhalt.de/project/leitfaden-forensik-11414, 19/01/2020

Afterwards, additional publications were released in order to refine this forensic process model. First considerations to establish a metric in order to judge the usefulness of certain forensic methods and tools during a given investigation were made in [**ACKD12**].

A major milestone was the publication of [KDV15] which summarizes research done on this forensic process model. It represents a fairly mature development stage of this forensic process model in relation to the use in Desktop IT. Hence, the state of the forensic process model as presented in [KDV15] is the foundation for the adaptation to the ICS and Automotive IT domains and the enhancements to the forensics process model performed during the course of this thesis.

This stage of the forensic process model as presented in [KDV15] and the influence of the various publications on this model is described later in this chapter (see Section 3.1.2). The next section gives a more comprehensive overview on the publication history of this forensic process model.

3.1.1 Publications and Work Done Leading to the [KDV15]-Model

The chronological relation between the various publications and the work performed during the research project *Leitfaden IT-Forensik* which led to the creation of the [KDV15]-model is shown in Figure 3.1.

Publications and Work Done in 2008

In February 2008, the research project *Leitfaden IT-Forensik*[3] (German for *„Guidelines for IT Forensics“*) started as a joint work with various contributors including the Research Group Multimedia and Security[4] at the Otto-von-Guericke-University of Magdeburg[5]. This research project provided substantial input to the forensic process model and led to the creation of the *BSI Leitfaden IT-Forensik* (see [BSI11] for these German guidelines).

The development of the forensic process model during said joint research project underwent several stages. The knowledge about this development process provides some insights into the underlying considerations and concepts during the establishment of the [KDV15]-model for the forensic process which would otherwise be not accessible.

Publications and Work Done in 2009

The joint work on project *Leitfaden IT-Forensik* was finalized in 2009. In addition, a number of supporting publications discussing various aspects of the forensic model were released.

The first publication of the year was [**KHA$^+$09**]. This work is written in German and the title *Sicherstellung von gelöschtem Schadcode anhand von RAM-Analysen und Filecarving mit Hilfe eines forensischen Datenmodells* roughly translates to *Gathering of deleted malware code by using RAM analyzes and file carving and applying a model for forensic data*. This publication uses the forensic model discussed in this chapter. The version of this model used here already contains three major elements of the [KDV15]-model. It covers six distinct *Investigation Steps*, eight *Data Types* and six *Methods* using definitions very close to the ones

[3] https://forschung-sachsen-anhalt.de/project/leitfaden-forensik-11414, 19/01/2020
[4] https://omen.cs.uni-magdeburg.de/itiamsl/english/home/home.html
[5] https://www.ovgu.de/en/

used in the [KDV15]-model. The forensic process model is used to describe the selection of methods and tools usable during the forensic process. This is shown with an example in which various pieces of data are gathered and then combined during the forensic process in order to achieve some valuable evidence. This example is revised and explained in detail in Section D.3. In addition, the model is used to ensure the integrity and authenticity of evidence extracted from a memory dump of a Windows-based system. Furthermore, approaches for the long term preservation of forensic data are discussed.

[KHD09] follows a similar outline. It describes the forensic process model which became the [KDV15]-model. Therefore it is the earliest publication on this model in English. The description of the forensic process model is close to the one offered in [**KHA+09**] although it states that the various *Investigation Steps* do not necessarily follow each other in a strict order. It is possible to switch back to certain steps during the forensic process as is visualized in Figure 3.2.

The application case used in this publication shows the need for the inclusion of a *Strategic Preparation*-phase into the forensic process. The scenario gives an example of *Live Forensics* where screen contents are extracted from the volatile memory of a windows-based system and then reconstructed. The *Strategic Preparation* includes the deployment of methods required to gather these memory contents with as little impact on the running system as possible and hence increases the evidentiary value of the collected evidence.

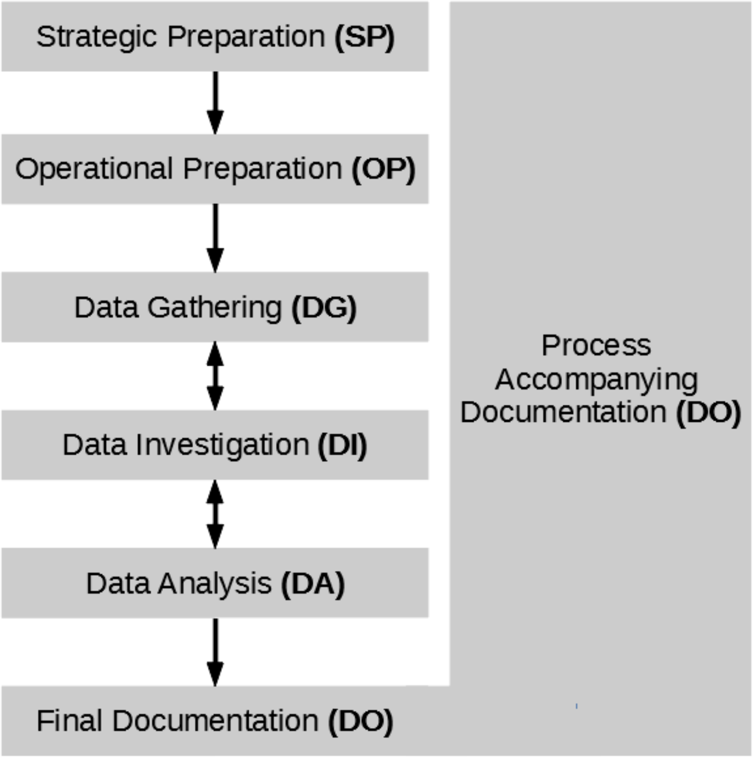

Strategic Preparation (SP)
Operational Preparation (OP)
Data Gathering (DG)
Data Investigation (DI)
Data Analysis (DA)
Final Documentation (DO)
Process Accompanying Documentation (DO)

Figure 3.2: Order of the *Investigation Steps* as described in [KHD09]

[**AKD09**] was published in the same year. It uses a forensic process model to support the creation of a Forensic Examination Taxonomy. This taxonomy is geared towards describing forensic examinations and their results in a common language. This taxonomy is based on the CERT-Taxonomy used for describing computer security incidents (see [HL98]) and includes elements required to describe a forensic process. In order to achieve this, it uses an underlying forensic process model which is based on the one presented in [KHD09]. The main contribution to the development of the [KDV15]-model from this publication is the discussion of the scope of the forensic process. The elements chosen for the taxonomy present a broad view on the field of forensic investigations. This includes malicious attacks on a computer system, the theft of data from a computer system as well as error and malfunction. Hence, it covers the entire width of **IC1**.

A successor to [KHD09] is provided by [KHDV09]. Here the same forensic process model is used for a similar task. Additional consideration is put on ensuring authenticity and integrity in the example case used in this publication.

Publications and Work Done in 2010

[**KHAD10**] uses the forensic process model from the earlier publications in order to infer requirements for the creation of a dedicated forensic tool. This publication covers the design considerations of the Linux Forensic Transparent Bridge (LFTB). The LFTB is designed to capture network traffic in a manner which supports the forensic process. This includes performing functions aligned to the phase of *Data Gathering* and *Passive documentation*. The tool captures the network traffic and documents this process in a dedicated log file. The evidentiary value of this network capture is further increased by ensuring integrity and authenticity of the capture. For the integrity, a specific capturing method with a lower rate of loss is selected (addressing **PMC2**). To maintain integrity during the entire forensic process and to ensure authenticity, all reports and the evidence are cryptographically signed. This supports **PMC1**. Confidentiality of the gathered data can be ensured by using encryption (addressing **PMC3**). Additionally, the tool ensures its own integrity by performing an analysis of the runtime environment on startup.

Publications and Work Done in 2012

The work done in [**ACKD12**] is published in German. The title *Erste Betrachtung einer Metrik für Methoden der IT-Forensik* translates to *First considerations for a metric to judge methods usable in computer forensics*. This work relies on the forensic process model from earlier publications and is the last publication on the forensic process model in German language. The paper establishes criteria to determine the cost of employing and using certain forensic methods as well as judging the quantity and quality of the additional traces made available by employing this method. A major contribution of this paper is the discussion on what this cost for the forensic process actually compromises. This discussion is based on the forensic process model from earlier publications since this cost also includes factors in regard to the overall forensic process. Besides the monetary cost to deploy a given method and the work time required to use the method, this cost includes a potential negative impact on the entire computer forensic process, by making various other traces unavailable or less reliable (see the discussion on *Live Forensics* and *Post-Mortem Forensics* in Section 2.3.2.3). This factor is described as *Strukturwirkung* (German for *„Structural Impact“*) and takes on the concepts established during the joint work on the project *Leitfaden IT-Forensik*.

Publications and Work Done in 2015

The final installment of the forensic process model geared towards the use in Desktop IT was published in 2015. This work was published in [KDV15]. Since this model is the foundation for the enhancements and adaptations made to this forensic process model during the course of this thesis, the state of the model in this publication is presented in full detail in Section 3.1.2.

3.1.2 The [KDV15]-Model

A set of selected forensic process models geared towards the use in Desktop IT has been discussed in Section 2.3.1. This survey on the field of process models for conducting forensic investigations in networked computer systems aimed at identifying desired properties for such process models. Additional input from various sections in Chapter 2 was used to compile such a list. This list of *Forensic Process Model Criteria* is presented in Section 2.3.2.4. It includes eight distinct *Forensic Process Model Criteria.*

Although such a formal list was not compiled for the initial creation of the forensic process model which became the [KDV15]-model, some of the criteria were addressed. These criteria are based on the properties of the forensic process models surveyed before and during the creation of the initial model for the *Leitfaden IT-Forensik* ([BSI11]). Among these desired properties are:

- The division of the process into phases of activities (**PMC8**)
- The possibility to maintain a progressive chain of custody or documentation (supporting (**PMC1** and **PMC4**)
- The consideration of *Live Forensics* against *Post Mortem-Forensics* (or the order of volatility in general - this corresponds to **PMC7**)
- The fact that forensic investigations can be triggered by either a legal accusation or a potential security incident which has an impact on which evidence is of interest for investigating the given hypothesis (corresponding to **PMC5**)

The forensic process models presented in Section 2.3.1 usually address some of these aspects with varying degrees of detail. Some models address various aspects more comprehensive than others. A prominent example here is the fact that some of these models focus on one of the two different *Investigative Contexts*. In this case, the processes and methods proposed by them are tailored towards handling this specific kind of investigation. As these investigations require different kinds of evidence and as these kinds of evidence are handled in a different manner, these models do not adhere to **PMC5**.

The forensic process model from [KDV15] is aimed at addressing all these desired properties in order to present a unified process model able to address the different handling of specific types of evidence. As the following section will show, the model fits these requirements. This is helpful, since the use of the [KDV15]-model is mandated by the task description and a logical consequence of the amount of publications of the author using this specific forensic process model (see Section 1.4). Hence, the [KDV15]-model serves as the foundation for the research into adapting it to the ICS and Automotive IT domains performed in this thesis. Hence, a detailed introduction in the models aspects and how these address the various *Forensic Process Model Criteria* identified in Section 2.3.2.4 is warranted.

This forensic process model consists of three major elements. It divides the forensic process into six distinct *Investigation Steps* (see Section 3.1.2.1). Furthermore, it defines eight *Data Types* (see Section 3.1.2.2) and six *Methods* (actually representing *Classes of Methods*; see Section 3.1.2.3). Various other aspects have been discussed during the publication history on this forensic process model (as discussed in Section 3.1). These aspects are not necessarily made explicit in these publications but inform underlying principles and hence are explained in the course of this section. These include the concepts of *Data Streams* and *Structural Impact*. *Data Streams* group the forensic evidence addressed by this model in three distinct categories with different characteristics. *Structural Impact* describes how a given forensic method affects the entire forensic process.

The following sections describe the three major elements of the [KDV15]-model and the additional concepts of *Data Streams* (see Section 3.1.2.5), *Structural Impact* (Section 3.1.2.4) and the *Classification Scheme for forensic tools and methods* (see Section 3.1.2.6). These concepts are not mentioned in [KDV15] itself but still represent underlying principles of the forensic process as understood by the discussed model.

3.1.2.1 *Investigation Steps* in the [KDV15]-Model

In order to achieve a division of the forensic process into phases of activities, this forensic process model divides the forensic process into a set of six distinct *Investigation Steps*. These *Investigation Steps* appear in [KDV15] with the following definitions:

- *„Strategic Preparation **SP***
  measures taken by the operator of an IT-system in order to support a forensic investigation prior to an incident
- *Operational Preparation **OP***
  measures of preparation for a forensic investigation after a suspected incident
- *Data Gathering **DG***
  measures to acquire and secure digital evidence
- *Data Investigation **DI***
  measures to evaluate and extract data for further investigation
- *Data Analysis **DA***
  measures for detailed analysis and correlation between digital evidence from various sources
- *Documentation **DO***
  measures for the detailed documentation of the proceedings, also for the transformation into a different form of description for the report of the incident (e.g. non-technical)“

These *Investigation Steps* include the four most common steps shown in Section 2.3.2.1. The *Operational Preparation* covers the steps performed during the *Identification (of Evidence)*.

Here, the specific sources for potential evidence are identified and the decision which data is gathered is made. This influences if the *Data gathering* includes *Live Forensics* or not. The step itself aligns with the *Acquisition* or *Preservation of Evidence* found in every other process model. While this *Investigation Step* is not split between *Live Forensics* and *Post-Mortem Forensics* this task is done within the [KDV15]-model by the inclusion of *Data types*, *Methods* and *Data Streams*. The *Data Investigation* describes the activities performed between the gathering of data and the analysis of said data. These steps are present in the [FS07] and [Cas04]-models as *Recovery* or *Harvesting*. A similar phase can be found in [Arn17] as *Examination*. Other process models do not make this fine distinction. The last of the four usual steps is the *Documentation* which covers the generation of a report. This report might also be in the form of testimony in front of a court. This activity covers the *Presentation of Evidence* phase commonly found in forensic process model.

This model also includes a *Strategic Preparation*. This is an extension of the *Pre-Incident Preparation* present in those models aiming at primarily supporting incident response ([MPP03] and [FS07]). It goes beyond establishing measures to detect a potential security incident before said incident takes place and also includes the implementation of measures which aim at supporting a potential forensic process. Such measures could be taken by system operators in order to gain more reliable forensic data in case a forensic investigation becomes necessary. Here, additional data could be stored or the authenticity and integrity of data already present could be enhanced by including mechanisms. In addition, mechanisms to gather additional data could be installed beforehand if the installation during an ongoing investigation would alter the system state. Such an example would include the installation of network taps which would usually cause a short disconnect while the tap is added to the network topography or which could cause a noticeable delay.

Performing such a *Strategic Preparation* is beyond the scope of the forensic process models which start with an accusation (like [Pol95]) - here the process begins after an accusation has been made. Hence, the [KDV15]-model puts the system operator (or rather persons able to configure the system, like e.g. the system administrator) into the focus by enabling such a *Strategic Preparation* and by addressing incidents in **IC1** which are triggered by the detection of certain indicators of compromise.

Another special feature of this model is the *Documentation* phase. According to [**AKD09**] this phase is split between a *Process Accompanying Documentation* and a *Final Documentation*. The *Process Accompanying Documentation* is similar to the *case management* proposed by [Cas04]. It takes place concurrent to the entire computer forensic process and documents the actions performed and decisions made. This helps maintaining the authenticity and integrity of the forensic evidence by maintaining a chain of custody. In the end of the forensic process the *Process Accompanying Documentation* is used to create the *Final Documentation*.

Statements about the overall structure of the forensic process are not contained in [KDV15]. While [BSI11] includes the possibility to branch back from **DA** and **DI** to **DI** or **DG** respectively, such a possibility is not explicitly included in [KDV15]. Hence, this points to a flat structure with the *Investigation Steps* following in a strict order. These two different overall structures can be seen in Figure 3.3. The difference between these two overall structures is the omission of the possibility to repeat certain *Investigation Steps* if need arises. This difference is most likely not the result of a deliberate design decision but rather the result of the need for brevity in some publications following the work on project *Leitfaden IT-Forensik*. The overall

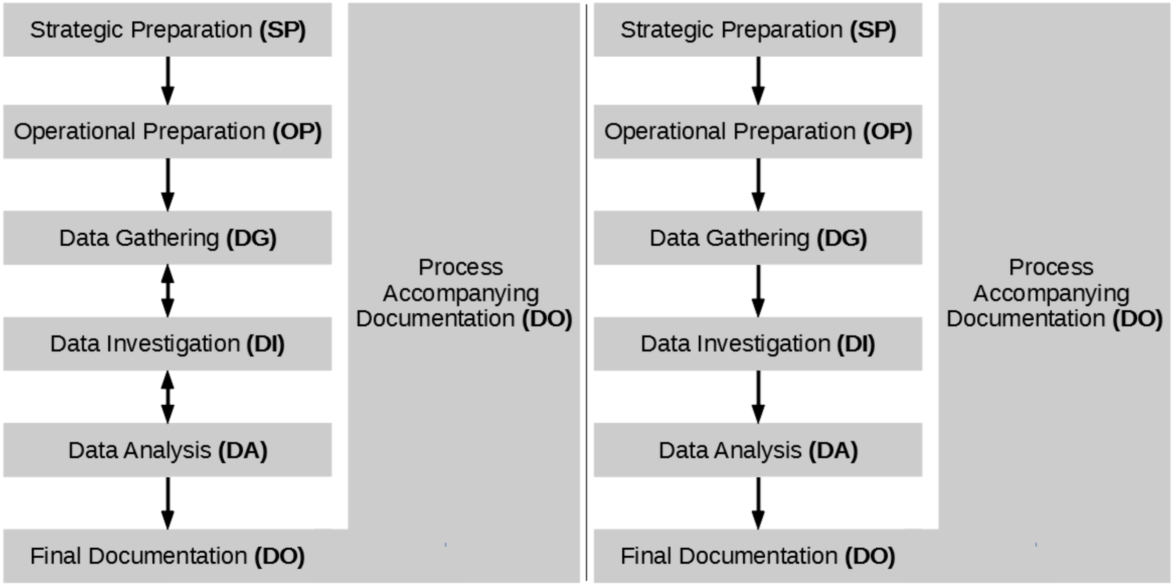

Strategic Preparation (SP)
Operational Preparation (OP)
Data Gathering (DG)
Data Investigation (DI)
Data Analysis (DA)
Final Documentation (DO)
Process Accompanying Documentation (DO)
Strategic Preparation (SP)
Operational Preparation (OP)
Data Gathering (DG)
Data Investigation (DI)
Data Analysis (DA)
Final Documentation (DO)
Process Accompanying Documentation (DO)

Figure 3.3: Order of the *Investigation Steps* in the [KDV15]-Model with [BSI11] on the left side and [KDV15] on the right side

structure of the forensic process as well as the certain benefits and drawbacks of different structures is addressed in Section 5.3.1.5.

By including a *Strategic Preparation*-phase this process model allows for a potential *Pre-Incident Preparation*. This possibility is especially important when taking **IC1** into account where investigations are often triggered by the detection of an incident. As the inclusion of a *Strategic Preparation*-phase covers the system operators (or administrators) possibilities to increase detection capabilities and to prepare for the gathering of additional evidence, this forensic process model addresses both **IC1** and **IC2**. Hence, it contributes to fulfill **PMC5**. The presence of a *Strategic Preparation*-phase however fulfills **PMC6**.

By dividing the forensic process into six distinct *Investigation Steps* which follow the logical sequence of a forensic investigation, the [KDV15]-model adheres to **PMC8**.

The *Documentation*-phase supports the maintaining of the authenticity and integrity of the evidence during the forensic process. The *Documentation*-phase therefore addresses **PMC1**.

3.1.2.2 *Data Types* in the [KDV15]-Model

Data Types are a feature introduced in the [KDV15]-model to group data which is handled in a specific way or of interest for investigations into specific hypothesis. Although not well motivated in [KDV15] they can offer great benefit to the computer forensic process. This is due to the fact that the *Data Types* provide a guideline on what methods and tools can be used on certain pieces of potential forensic evidence.

The discussion of the various forensic models has shown that **IC1** and **IC2** usually rely on different types of data as evidence. These types of data are handled in a different manner and are often taken from different sources within the system. There are dedicated methods and tools to gather, investigate or analyze these different types of data. For example, in an investigation in **IC2**, spreadsheets of financial data might be of interest. These can be analyzed by dedicated software designed for forensic accounting. These spreadsheets will usually be stored on non-volatile mass storage and can be extracted directly from the specific directories. An investigation in **IC1** might however require a list of the currently active network connections. This can only be extracted from the volatile memory of a system with specialized tools or reconstructed from a recording of the network traffic. In these cases, gathering and analyzing these two types of evidence differs widely.

Hence, a division of all the potentially available forensic evidence into different sets of data types can support the forensic process by grouping tools and procedures to address specific types of data according to their nature. Eight *Data Types* are proposed in [KDV15]. Please note that some missing words have been filled in from [**KHAD10**]. Those are noted with []. These data types are, according to [KDV15]:

- ***„DT1***
  hardware data [is data] in a system, which is not or only in a limited way influenced by the operating system and application. Examples are the real time clock, interrupts, hardware serial numbers or the code of firmware of hardware devices
- ***DT2***
  raw data as a sequence of bits (or data streams) of components of the system not (yet) [interpreted], can contain data of all the other data types. Examples are all sorts of dumps, primarily memory- and mass-storage dumps and network packets
- ***DT3***
  details about data as meta data added to user data, stored within user data or externally, can be persistent or volatile. Examples are MAC-times of files or sequence numbers of network packets
- ***DT4***
  configuration data that can be changed by the OS or applications, which modify the behaviour of the system, including the configuration of hardware, of the OS and applications, but not its behaviour with regards to communication
- ***DT5***
  network communication data, which modifies the behaviour of the system with regards to communication, including, amongst network configuration data, also inter process communication (e.g. pipes and RPC) of IT applications
- ***DT6***
  process data about a running process including the status of the process, the owner of the process, its priority, memory usage or the related application (regarding IT-applications these can be single threads or data about them)
- ***DT7***
  session data collected by a system during a session, regardless of whether the

session was initiated by a user, an application or the OS including started programs, visited websites or documents within a user session

- ***DT8***
  user data contents created, edited or consumed by the user including media data e.g. pictures, texts, audio or video“

These categories are not mutually exclusive. In fact, the same physical representation of data could belong to different data types, depending on the context the data is interpreted in. For example, the same bits of data can be considered as **DT2** (*raw data*) while creating a forensically sound image of a section of non-volatile mass storage and then be considered as **DT8** (*user data*) while the contents of the file extracted from this raw data are reviewed. This transformation could be described as adding a logical context to this piece of data when using the similar concept of the *Digital Paradigm* proposed in [Pol95] (see Section 2.3.1.1). Hence, one could argue that these data types form a layered approach to categorize the potentially relevant data according to how they are interpreted during a forensic investigation.

A good example for the usefulness of different *Data Types* can be found in Section 2.3.4.3. This section discusses the forensic use of data stored within Automotive IT. This data contains various sets of data which can be useful for different types of investigations. For example, a *phone book* or a *mailbox* - useful for investigations in **IC1** - would be part of **DT8**. *WiFI SSIDs* could be useful to investigate a potential attack on Automotive IT. This would pertain to **IC2**. This type of data would belong to **DT5**. This shows quite clearly that different forensic investigations require different *Types of Data*.

Being implicitly motivated by the need to investigate very different questions within a forensic investigation, the *Data Types* address the need for both of the *Investigative Contexts*. Hence, the *Data Types* address **PMC5**.

A comprehensive example for the close relationship between *Investigation Steps* and *Data Types* in the [KDV15]-model is provided in Section D.3.

3.1.2.3 *Classes of Methods* in the [KDV15]-Model

Another concept used in the [KDV15]-model is originally referred to as *Methods*. However, *Classes of Methods* is a more distinct description and will be used during the course of this thesis. This concept describes categories of tools or methods usable during a forensic investigation based on which software component employed in Desktop Computers provides this method. [KDV15] gives the following definitions for these *classes of Methods*:

- *„**Operating System (OS)** methods provided by the operating system such as maintaining process lists, log file creation*
- ***File System (FS)** methods provided by the file system such as updating of file access times, permission management, storage space management*
- ***Explicit means of intrusion detection (EMID)** methods provided by additional software with the characteristic of being executed autonomous on a routine basis and without a suspicion of an incident*

- ***IT application (ITA)*** *methods provided by IT-Applications that are operated by the user, in addition to their main functionality they also provide forensic methods such as log keeping*
- ***Scaling of methods for evidence gathering (SMG)*** *methods to further collect evidence to be used if a suspicion is raised but unsuited for routine usage in a production environment such as high false positives, CPU demands etc.*
- ***Data processing and evaluation (DPE)*** *methods which support a detailed forensic investigation, display, processing and documentation including dedicated forensic suites and tool collections“*

Again, [KDV15] does not provide a lot of insight on why these categories are chosen. The categorization still offers some benefits to the forensic process by giving information about certain prerequisites to use certain forensic methods or tools during the investigation. In addition, certain *Data Types* are generally addressed by specific *Classes of Methods* as can be deducted from the definitions of the various *Classes of Methods*.

For example, methods belonging to the class *Operating System* are already present within a given system environment and can be used directly. Here, no special requirements exist before such a method can be used. These methods could provide **DT1** (*hardware data*), **DT4** (*configuration data*), **DT6** (*process data*) or **DT7** (*session data*).

The same goes for the *File System* which is also usually a part of a running computer system. Methods from this class could provide **DT2** (*raw data*) in the form of raw access to file contents. They could also provide **DT3** (*details about data*) by providing access to the various time stamps associated with files in most complex file systems.

Explicit means of intrusion detection however describes those methods installed during a *Strategic Preparation* (or during a *Pre-Incident Preparation* like proposed by ([MPP03] and [FS07]). Methods from this category can provide additional evidence or cause a forensic investigation in the first place by providing the ability to detect potential incidents. These methods must be implemented before a specific investigation starts in order to have an impact. Data provided by these methods could include **DT5** (*network communication data*), **DT6** (*process data*) or **DT7** (*session data*). This represents the data these*Explicit means of intrusion detection* use in order to detect potential incidents.

IT application describes all the functionality included in common IT-applications that could be used to support forensic investigations. A trivial example is the list of recently opened documents available in almost any kind of text processing software or the history in a web browser application. These methods are available if the given application is installed and the respective traces have not been deactivated or purged by the system operator. These methods can provide **DT7** (*session data*) or **DT8** (*user data*).

Methods which are unusable in the normal productive operation of the computer system in question are grouped together in the *Scaling of methods for evidence gathering* class of methods. These methods are only used if there is a valid reason to do so, since they either restrict productive operations or produce vast amounts of data which would usually be unnecessary or impractical to store. This impracticality can originate from the required storage capacity or from data protection concerns (see Section 2.1.6 for a discussion on the role and impact of data protection in computer forensics). Depending on the specific method various types of data could be addressed here. If the method relies on storing the entire network traffic, **DT2** (*raw data*) and **DT3** (*details about data*) would be processed here. Other examples might include an extended monitoring and logging of running processes (addressing **DT6** - *process data*) or sessions (addressing **DT7** - *session data*).

Dedicated methods and tools to document the forensic process are designated as *Data processing and evaluation*. This describes methods which are able to support or even provide the *Process Accompanying Documentation* or the *Case management* as described by [Cas04]. Also included in this class of methods are dedicated forensic methods able to support the forensic process by organizing data or visualizing relationships between various sets of data. Tools which create a time line of certain events are a prime example of methods in this category. Such methods are usually not installed on a computer system and are instead part of dedicated forensic workstations or tool collections. Depending on the exact kind of method, various types of data can be addressed.

The inclusion of these *Classes of Methods* touches the selection of usable tools used during either **IC1** or **IC2** (supporting **PMC5**) and the requirements for the use of specific tools (by, for example implying that these tools are installed on a system before an investigation is triggered and hence require a *Strategic Preparation*, supporting **PMC6**).

3.1.2.4 Structural Impact in the [KDV15]-Model

The concept of *Structural Impact* (German for *„Structural Impact“*) is integral to many considerations in the [KDV15]-model for the forensic process. It is briefly mentioned in [BSI11] while [**ACKD12**] expands on this concept. An early working definition from the joint work on the research project *Leitfaden IT-Forensik* is provided here in an English translation conducted by the author of this thesis:

> *„Structural Impact discusses how the application of a forensic methods alters the data of the investigated system. A distinction is made between the cases where this alteration is local or network-wide. Furthermore a distinction if only data or also non-volatile data is affected. [...] During Live Forensics the investigator must be aware that every used forensic method alters at least the volatile data on the investigated system and potentially also the non-volatile data.“*

The detailed description of the *Methodencharakterisierung* and its elements from [BSI11] (German for *„characteristics of methods“* - see Section 3.1.2.6) gives some information on what aspects of the forensic process can be influenced by *Structural Impact*. This overview is provided in Table 3.1 in German as well as in the English translation. These terms form a tree where *stw1,1,1,1* is an expansion of *stw1,1,1* which is an expansion of *stw1,1* which is an expansion of *stw1*. In this case, the meaning is that there is an impact on the forensic target (in form of a system) of the forensic investigation by locally altering data in *Volatile Memory*. Based on Table 3.1 a *Structural Impact* might alter data in *Volatile Memory* and/or data in *Non-Volatile Memory* locally and/or network wide.

Based on these considerations on the effect of the *Structural Impact*, [**ACKD12**] discusses a metric to judge the value of the use of certain forensic methods during a forensic investigation.

Table 3.1: Aspects of *Structural Impact* in [BSI11]

Descriptor	German	Translation to English
	Strukturwirkung (STW)	*Structural Impact*
stw1	Wirkung auf den forensischen Prozess	Impact on the forensic process
stw1,1	Wirkung auf das Untersuchungsziel (UZ)	Impact on the target of the investigation
stw1,1,1	wirkt informationsverändernd lokal (la)	local alteration of data
stw1,1,1,1	flüchtig	volatile
stw1,1,1,2	nichtflüchtig	non-volatile
stw1,1,2	wirkt informationsverändernd netzwerkweit (osi 1,...,7)	network-wide alteration of data
stw1,1,2,1	flüchtig	volatile
stw1,1,2,2	nichtflüchtig	non-volatile
stw1,2	keine (Informationsveränderung)	no alteration of data

On one hand, the benefit of using the given method is considered while on the other hand the cost for said method is taken into account. The benefit is usually in the form of additional data made available by the use of a given method. The cost comes in terms of work time or monetary cost but also in terms of *Structural Impact*. This would be the case if the use of a certain method makes the use of different methods impossible or ineffective by altering data gathered or investigated by different method. Hence, it describes the impact on the entire computer forensic process by employing said method.

This entire concept expands on the discussion on the use of *Live Forensics* and *Post-Mortem Forensics* in Section 2.3.2.3. It gives a metric to judge if keeping a system active to gather certain pieces of evidence only available in *Live Forensics* is worth reducing the evidentiary value of the evidence which could be gathered during *Post-Mortem Forensics*. Hence, this concept aligns with **PMC7**. Also, the *Structural Impact* addresses the integrity of potential traces (**PMC1**) since *Structural Impact* leads to a decrease in integrity. One could argue that a greater *Structural Impact* also entails the potential loss of data (**PMC2**).

3.1.2.5 Data Streams in the [KDV15]-Model

The [KDV15]-model implies the presence of three different sources for forensic data inside computer systems. The *Network data stream* is mentioned in [**KHAD10**] and the definition of **DT2** (*raw data*) refers to *memory- and massstorage dumps*.

There are three different sources for potential forensic data within the networked Desktop IT domain. *Mass Storage* and *Main Memory* are frequently mentioned and mostly used during *Post-Mortem* or *Live Forensics* respectively. Some methods belonging to the classes *Explicit means of intrusion detection* or *Scaling of methods for evidence gathering* use the network communication for detection or gather information directly from the (network) *Communication*.

These three potential sources for data carry different implications for the forensic process in terms of accessibility and *Structural Impact* (see Section 3.1.2.4). Therefore it is appropriate to treat these sources differently. These sources are referred to as *Data Streams*.

A closer look on and a formalization of these *Data Streams* is missing during all publications leading up to [KDV15]. A formal definition was only presented in [**ALK$^+$18**]. This paper discusses the particulars of these three different data streams in the context of Industrial Control Systems. Although published long after [KDV15] this paper gives an exhaustive explanation of the *Data Streams* presented here to provide a better understanding of this implicit core concept of the forensic process model. This brief overview on the specific possibilities and challenges for the forensic process when dealing with these *Data Streams* is given here. Although [**ALK$^+$18**] is geared towards the use in Industrial Control Systems, these *Data Streams* are also applicable to classical Desktop IT. After some adjustments for an improved clarity of the terms, these *Data Streams* are defined as follows:

- Non-volatile Memory
  Referred to in [**ALK$^+$18**] as *Mass storage*, this *Data Stream* represents the non-volatile storage of a computer system. This is often synonymous with bulk storage. It can take the form of internal storage or external storage. Internal storage includes every type of non-volatile storage that is not removed from the computer system during normal operation, like hard drives and flash drives. External storage is storage which is regularly attached and detached from a computer system. Examples include optical discs or thumb drives. They have in common that data is retained after the computer system in question is deactivated.

  Due to the specifics of various file systems, mass storage can indeed contain very old data. Most file systems do not outright remove contents when the user marks them for deletion. Rather, the specified locations are released for rewriting. As long as nothing is rewritten in the specific location, the data is still available. Hence, this data stream allows for a look into the, relatively speaking, remote past of the computer system in question.

  Generally speaking, these devices can be removed from the computer system during *Post-Mortem Forensics* and attached to a forensic work station in order to create a forensic copy of the storage in question which can then be used during the computer forensic process. If the process of creating a copy is well-documented and a solid chain of custody is kept for the forensic copy, integrity and authenticity are generally high.

  The non-volatile storage can also be accessed during *Live Forensics*. In this case, the access is managed by methods of the *Operating System* and the *File System*. Hence, every access made to the *Non-volatile Memory* alters the state of the operating system (by influencing the performed operations) and the file system (by setting file access time stamps). In addition, the system might perform actions during the time it takes to gather the results of these access actions. Here, a *Structural Impact* on the following actions cannot be avoided. The alteration of the system state lowers the integrity and authenticity of every piece of evidence gathered from the system after such an access on the live system. The drawbacks in this case are the same as faced during the gathering of evidence from the *Volatile Memory*.

- Volatile Memory
  [**ALK$^+$18**] uses the term *Main Memory* to describe this *Data Stream*. It represents the volatile memory of a computer system. This is usually working memory, but might also include virtual memory disks which will lose all data once power to the memory is cut

off. This is usually the case when the system in question is shut off. Hence, this data stream is unavailable in *Post-Mortem Forensics* and can only be accessed during *Live Forensics*.

The allocation of main memory carries some similarities with the file systems from mass storage. If some portion of working memory is no longer required, it is released. In general, the effort to explicitly delete memory locations that are released is not made. Data relevant for already terminated processes can be found in these sections. Due to the specific implementations of memory chips used for the working memory, this data could however fade without being explicitly overwritten by another process using this section of memory. Since the *Volatile Memory* is reset every time the computer system is powered off, the potential to look into the past of the system is less pronounced than in the case of *Non-volatile Memory*.

Main memory is managed by *Methods* of the *Operating System* and therefore every access to this data stream has to be performed by sending access requests to the operating system. These requests have to rely on the fact that the OS is answering them 'honestly'. Hence, a manipulation of the OS will falsify this data. In addition, the request itself will alter the system state. Furthermore, the fact that the system is performing operations (and possibly executing malicious activities, including the alteration of potential evidence in the *Mass Storage*) causes a *Structural Impact* on the entire computer forensic process. The integrity and authenticity of any piece of evidence gathered from the corresponding system is negatively impacted.

- Communication
  Communication encompasses all the data transmitted via various communication networks. This includes complex communication networks like BUS networks (e.g. Ethernet) or direct serial connection. No matter the medium or employed technology, they all have in common that the communication can only be observed at the moment it occurs. This is usually the case during *Live Forensics*. A communication medium will, in contrast to *Non-volatile Memory* and *Volatile Memory* never contain any artifacts of past processes or past communications.

  While the local network interface (e.g. the network card inside a Desktop computer) might store certain pieces of information in its *Volatile Memory* (for example to implement buffering or assemble the payload from different network communication packets together), practical access to *Communication* is only possible by accessing the carrier medium or one of the network interfaces involved and recording the communication.

  Here, direct access to the carrier medium is preferred since it allows for a more thorough control of the integrity and authenticity of the captured data. If a network tap is installed before the capturing of *Communication* begins, this process is a purely passive affair and has no *Structural Impact* on the system in question.

As discussed before, the same data can exist in different forms and different *Data Streams*. For example, the network communication would be present in the *Communication* but could also be recorded to *Non-volatile Memory* in form of some mass storage. However, the methods to gather these two different representations of the same data would differ. In the first case methods addressing communication data would be used. This would entail the installation of a network tap or the logging of the network communication on one of the systems involved in the communication. In the second case, methods to gather data from *Non-volatile Memory* would be used. This could be as simple as producing a forensic image of the mass storage medium storing the given data. The same data could also be present in the *Volatile Memory* of the system. For example, this data could be cached in the main memory for some reason. The methods to access this data in the *Volatile Memory* would still differ.

However, these three representations would be analyzed in the same way once they have been gathered since they belong to the same data type. It is generally advantageous to gather the same data from different data streams when possible in order to increase authenticity and integrity by crosschecking between all available representations.

A similar understanding of the different nature of various data sources was presented in [VKL16]. This source differentiates between *Network data acquisition* and *Device data acquisition* which has to be handled in different ways. The reason to include a third *Data Stream* in the [KDV15]-model arises from the differing volatility of the specific sources.

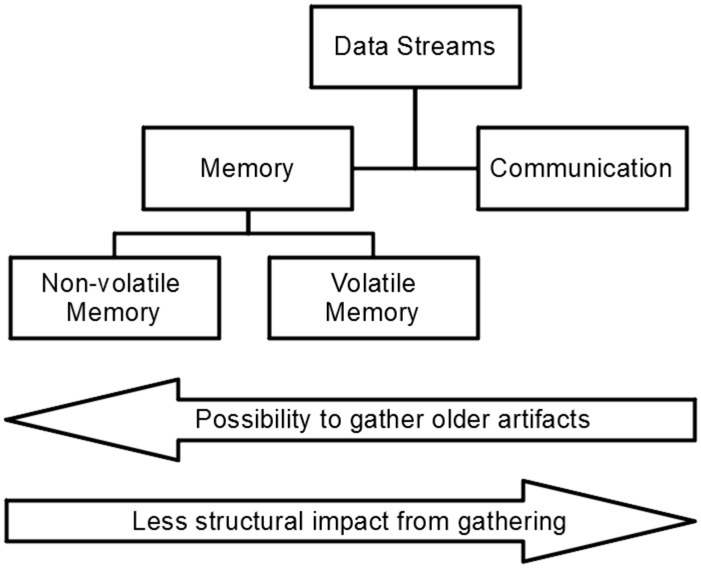

Data Streams
Memory
Communication
Non-volatile Memory
Volatile Memory
Possibility to gather older artifacts
Less structural impact from gathering

Figure 3.4: Possibility to gather data artifacts in different *Data Streams*

As mentioned when describing the specific *Data Streams* some of them carry the possibility to explore the past of a given medium. This results in the relationship shown in Figure 3.4. The further to the left the data stream is arranged in Figure 3.4, the greater the potential to explore the past of the given medium is. *Non-volatile Memory* might contain artifacts or files which have been deleted a long time ago. *Volatile Memory*, especially main memory, might contain memory segments of already terminated processes or data which is no longer used. *Communication* however contains only the communication which is currently active on the respective network.

However, the further to the right a *Data Stream* in Figure 3.4 is arranged, the lesser the *Structural Impact* when gathering evidence belonging to this *Data Stream*. *Communication* between two computer systems can be observed with no *Structural Impact* at all as long as a network tap is present. The gathering of *Non-volatile Memory* involves querying the system for memory contents at least in the case of working memory - which would be the usual case during *Live Forensics*. This implies a greater *Structural Impact*. Access to *Non-volatile Memory* implies either querying for the storage contents or shutting down the system - the ultimate *Structural Impact*.

The concept of *Data Streams* hence helps to identify the origins of potential forensic traces. They imply requirements for the gathering of these traces from the process point of view. The *Data Streams* of *Communication* and *Volatile Memory* are unavailable once a Desktop Computer is switched off and only the *Non-volatile Memory* remains available during *Post-Mortem Forensics*. Therefore, the concept of *Data Streams* supports **PMC7**.

3.1.2.6 *Classification Scheme for Forensic Tools and Methods* in [KDV15]

Another concept found in [BSI11] is referred to as *Methodencharakterisierung* (German for *„characteristics of methods“*). During the course of this thesis, this scheme is referred to as *Classification Scheme for forensic tools and methods* (or short: *Classification Scheme*). This scheme is intended to support a forensic investigator by providing a short overview on the properties of forensic methods usable during the forensic process and supporting decision making. The methods are assigned to specific *Investigation Steps*, *(Classes of) Methods* and *Data Types*. Furthermore, the scheme describes various additional properties of these methods which influence the forensic process as can be seen in Table 3.2 where an English translation of these terms is provided by the author of this thesis.

Some of these properties describe technical terms which have a practical input on the forensic process. This includes the property *DV* (*Datenvolumen*, German for *„Data Volume“*) which describes the amount of output data to be expected by the use of this forensic method. This carries some implications (storage for this data must be available and the monetary cost to acquire this storage might make the usage of this method impractical). However, these implications describe practical aspects and not so much the structure of the forensic process.

More fundamental aspects can be found in the elements which touch on the entire forensic process. These properties describe important aspects of methods and tools usable during forensic investigations. The most relevant of these are the *Untersuchungsvorraussetzung* (German for *„requirements for the (promising) usage of a forensic tool“*), the *Datenschutzrelevanz* (German for *„relevancy of data protection concerns“*) and the *Schutzmaßnahmen* (German for *„Protection measures for the integrity of a forensic tool, its input and its output “*). Since the forensic process aims at proving a certain sequence of events (see Section 2.3.2.4), the element of *Beweiskrafttendenz* (German for *„Tendency for evidentiary value“*) also requires some attention in detail. The following paragraphs provide this detail.

Requirements for the Promising Usage of a Forensic Tool

The *Untersuchungsvorraussetzung* describes conditions which must be fulfilled in order for the forensic method to be usable in a way to achieve meaningful evidence. These conditions have to be understood from a process point of view. Simple hardware requirements are handled by different aspects of this scheme. Investigation into the specific conditions envisioned in [BSI11] provide a better understanding of this fact. Such a list of conditions can be seen in Table 3.3 based on the list provided in [BSI11]. Here, the terms in German are provided with a translation to English performed by the author of this thesis.

Table 3.2: Characteristics of forensic methods and tools in [BSI11]

Descriptor	Written out in German	Description in English
HW/SW	Hardware/Software	Describes if the method is implemented in hardware or software
AB	Allgemeine Beschreibung	General description which might contain version number of the tool, a description of input or output, configuration, or where to get it
UO	Untersuchungsort	Location of the investigation If the investigation happens locally or remotely and if it involves the entire system or subsystems
AE	Aktivierung erforderlich	If the method requires an installation or even an activation
UA	Untersuchungsaktion	Describes the action performed by the method
UZ	Untersuchungsziel	Describes what type of data is processed by the method
UV	Untersuchungs-vorraussetzungen	Describes requirements for the promising usage of the method
UE	Untersuchungsergebniss	Describes the output of the method
DSR	Datenschutzrelevanz	If data with privacy implications is gathered/processed by this method
OSI		On which OSI network layer this method works (if applicable)
STW	Strukturwirkung	*Structural Impact* caused by the use of this method
DV	Datenvolumen	The volume of the outputs created by the use of this method
BK	Beweiskraft	Judges the evidentiary value
SM	Schutzmaßnahmen	If the method protects its own integrity, input and output

Table 3.3: Potential requirements for the promising use of forensic methods in the scheme for *characteristics of methods* from [BSI11]

Descriptor	German	Translation to English
	Untersuchungsvorraussetzung (UV)	Requirements for usage
uv1	Logging ist eingeschaltet [la] [osi1,...,7]	logging mechanisms are activated
uv2	Netzwerkverbindung(en) wurden nicht getrennt [osi1,...,7]	network connections were not disconnected
uv3	Spannungsversorgung wurde nicht unterbrochen [la]	power connection was not disconnected
uv4	Caching aktiviert	Caching activated
uv5	Computersystem ist technisch funktionsfähig	Computer system is functional
uv6	Systemzugang (Administrator)	Access to the system with Administrator privileges
uv7	Systemzugang (Nutzer)	Access to the system with User privileges

These elements cover some different topics. *uv1* is concerned with the activation of additional logging mechanisms in advance of an incident. Such an action could only be performed during a *Strategic Preparation*. A similar case can be found in *uv4*. The activation of cache, which then might contain artifacts of evidence, is only useful if performed before an incident takes place.

uv2 and *uv3* expand on the concept of *Live Forensics* and differentiate between those cases where the computer system is still active and connected to the network and those where the computer system is still active but disconnected. The later might be the case if some user was afraid of malicious activity still being performed across the network. Here the use might have plugged out the network connection but kept the system active. Obviously, these two possibilities have an impact on the forensic process. As long as the system was not disconnected from the power grid, *Live Forensics* can be performed and the main memory (belonging to the *Data Stream* of *Volatile Memory*) investigated. If the network connection is plugged out, *Communication* might become unavailable during the investigation. If the power and the network connection are both disconnected only the *Non-volatile Memory* in form of the mass storage remains as a source of forensic evidence. This is equivalent to *Post-Mortem Forensics*. Hence, this concept could lead to an extended understanding of the term *Live Forensics* by differentiating between *Live Forensics* and *Live Forensics with disconnected network*.

Another interesting aspect mostly of relevancy during *Live Forensics* is represented by *uv6* and *uv7*. These elements describe the need of certain privileges on the investigated system. An example is the requirement of *root* (administrator) privileges on a *Linux* system in order to access the *procfs* [6] during *Live Forensics*. This virtual file system offers information about the running processes (**DT6**). Access to some portions of the *Non-volatile Memory* might also be linked to certain user privileges if the access should is performed using the *Operating System*

[6] http://man7.org/linux/man-pages/man5/proc.5.html, 26/04/2020

Table 3.4: Relevancy of data protection concerns during the use of forensic methods in the scheme for *characteristics of methods* from [BSI11]

Descriptor	German	Translation to English
	Datenschutzrelevanz (DSR)	Relevancy of data protection concerns
dsr1	nicht relevant	not relevant
dsr2	relevant	relevant
dsr2,1	Pseudonymisierung erforderlich	Pseudonymization required
dsr2,2	Anonymisierung erforderlich	Anonymization required
dsr2,3	Verschlüsselung erforderlich	Encryption required

and *File System* like this is the case during *Live Forensics*. When performing *Post-Mortem Forensics* these privileges are usually not required since access to the *Mass Storage* is realized circumventing the *Operating System* if possible. In this case, a forensic copy of the storage device is created. However, one additional aspect overlooked in [BSI11] touches on this topic. The knowledge of certain cryptographic keys in order to access encrypted storage media is an obvious extension to these user privileges.

While not a core concept of the forensic process as described in [KDV15], the impact of these requirements for the meaningful use of forensic tools is obvious. Actions can be done to fulfill these requirements during the *Strategic Preparation*. The *Operational Preparation* needs to take note which of these requirements are met. Hence, knowledge of the process-based requirements for forensic methods potentially used is necessary for an efficient *Operational Preparation*.

Relevancy of Data Protection Concerns

There might be constraints for the collection and usage of evidence during a forensic process. This challenge is described by **PMC3** and expanded in detail on in Section 2.1.6. The element of *Datenschutzrelevanz* (German for *„relevancy of data protection concerns“*) covers this aspect in the scope of the *Classification Scheme for forensic tools and methods*. The possible values for this element, based on the list provided in [BSI11], are provided in Table 3.4. This table includes the terms in German and a translation provided by the author of this thesis.

Table 3.4 differentiates between no relevant concerns for privacy during the use of a specific tool (*dsr1*) and a given relevancy (*dsr2*).

Furthermore, it suggests methods to address these concerns. *dsr2.1* describes pseudonymization as an approach to achieve de-identification. Here, personally identifiable information is replaced by pseudonyms. These pseudonyms are artificial and each entry referring to a specific person should use the same artificial identifier. For example, user names in a log file could be replaced by randomly generated sequences. However, each instance of the same user name would be replaced by the same sequence. Anonymization (*dsr2.2*) is a stronger version where this personally identifiable information is entirely removed from the data. These two approaches are usable to address privacy concerns.

Encryption (*dsr2.3*) also addresses other concerns for the collection and use of evidence. Encryption can also help to mitigate concerns over the potential collection of business secrets.

Table 3.5: Tendency for evidentiary value in the scheme for *characteristics of methods* from [BSI11]

Descriptor	German	Translation to English
	Beweiskrafttendenz (BK)	Tendency for evidentiary value
bk1	ja	yes
bk2	nein	no
bk3	eher schwierig	rather difficult
bk4	eher nicht	rather not
bk5	keine Aussage möglich	no statement possible

The usefulness of encryption is determined by various factors, like the quality of the encryption and the management of the encryption keys. While the use of pseudonymization and anonymization at the same time provides no benefit and is even a contradicting method, both of these can be combined with encryption in order to reach a higher degree of protection for personal data.

Tendency for Evidentiary Value

The element of *Beweiskrafttendenz* (German for *„Tendency for evidentiary value“*) describes the potential evidentiary value provided by the output of a given forensic method. [KDV15] gives a short description of *Beweiskraftendenz* which is provided here as a translation provided by the author of this thesis:

> *„An assessment of the tendency to prove evidence in relation to the guilt or innocence thesis [...].“*

This statement summarizes a range of different factors which have an impact on the ability of a given piece of evidence to prove or disprove a certain thesis of guilt or innocence. As discussed in Section 2.1.7 and Section 2.3.2.4, this relies on various *Admissibility Factors* (**AF**). Some of these factors are aligned to different aspects of *Classification Scheme for forensic tools and methods*. For example, **AF6** (*The technique producing (or drawing a conclusion from) the evidence ensures the integrity and authenticity of the evidence*) is addressed by the *Protection measures for the integrity of a forensic tool, its input and its output* discussed in the following paragraph. Another *Admissibility Factor*, **AF7**, is defined as *The technique producing (or drawing a conclusion from) the evidence documents the process of recording or storing a certain event or piece of information* is handled by the *Process Accompanying Documentation* and potentially supported by some of the *Protection measures for the integrity of a forensic tool, its input and its output* discussed in the following paragraph. Hence, this element touches on topics aligned with the remaining six *Admissibility Factors*. **AF1** to **AF5** touch on the **Daubert-Factors** as presented in Section 2.1.7) and are concerned with the method itself. As long as these *Admissibility Factors* are fulfilled, **AF0** (*The evidence helps the trier of fact to determine a fact in issue*) remains as the most relevant factor for determining the *Tendency for evidentiary value*.

Table 3.6: Protection measures for the integrity of a forensic tool, its input and its output in the scheme for *characteristics of methods* from [BSI11]

Descriptor	German	Translation to English
	Schutzmaßnahmen (SM)	Protection measures
sm1	(Eigen-)Schutz des forensischen Werkzeugs	Self-protection of the forensic tool
sm1,1	(Eigen-)Schutz (durch HW/SW)	Self-protection (by hardware/software)
sm1,2	externe Schutzmaßnahmen notwendig	External protection required
sm2	Schutz von UZ während der Verarbeitung durch das Werkzeug	Protection of investigation target during processing by the tool
sm2,1	Schutz gegen Veränderung durch das Werkzeug (durch HW/SW)	Protection against Alteration by the tool (by hardware/software)
sm2,2	externe Schutzmaßnahmen notwendig	External protection required
sm3	Schutz von UE durch das Werkzeug	Protection of investigation result by the tool
sm3,1	Schutz vor späterer Manipulation (durch HW/SW)	Protection against subsequent manipulation (by hardware/software)
sm3,2	externe Schutzmaßnahmen notwendig	External protection required

If a given piece of evidence is useful to prove a given fact depends on the case at hand and might wildly differ depending on the *Investigative Context* (see Section 2.1.2 for considerations on the different *Investigative Contexts*). As discussed in Section 3.1.2.2, different *Data Types* are useful for different *Investigative Contexts*. Hence, a generalized answer for the usefulness of the output of a given forensic tool for the determination of guilt and innocence is difficult. Even before considering the varying interpretation a court might have. Hence, the elements from [BSI11] as seen (and translated by the author of this thesis) in Table 3.5 are vague.

It is doubtful if a generalized determination of **AF0** is possible without taking the *Investigative Context* into consideration. This would prevent a generalized statement of the *Tendency for evidentiary value*. One could argue that this element could be used in a slightly different manner. In this case, **AF0** would always be assumed to be true. Since **AF6** and **AF7** are handled by different parts of the scheme for *characteristics of methods*, such considerations would be focused on **AF1** to **AF5** which align with the **Daubert-Factors**.

Protection Measures for the Integrity of a Forensic tool, its Input and its Output

The *Schutzmaßnahmen* (German for *„Protection measures“* mentioned in the *Classification Scheme for forensic tools and methods* refers to protection measures employed to ensure the integrity of the forensic tool itself, its input and its output. The specific aspects can be seen in Table 3.6 which contains the German description and an English translation provided by the author of this thesis.

The elements in Table 3.6 align with the forensic tool itself, the input used and the output created by this tool.

sm1 describes the self-protection of the forensic tool. Such a self-protection (*sm1,1* exists if the forensic tool has some mechanisms to ensure the integrity and authenticity of the forensic tool. This could be achieved by the ability of the forensic tool to check its integrity against pre-generated hash values (or against signatures which would also ensure authenticity). If this is not the case, external methods could be used to ensure the protection of the integrity and authenticity of the forensic tool, like using external mechanisms to check if a given hash value or signature is generated for the tool in question. In each case, generating these hash values during the *Strategic Preparation* vastly increases the usefulness of these measures by providing a known-good foundation to test against.

Ensuring the integrity and the authenticity increases the evidentiary value of the evidence collected or processed with this forensic method. As long as it cannot be assured that a tool is not altered or tampered with all the considerations during the design of the tool itself (for example the **Daubert-Factors** as presented in Section 2.1.7) might be of no use.

sm2 discusses an alteration of the data source during operations performed on the source. A data source can be an input file, a computer system or a component in this case. Such an alteration is a *Structural Impact* (see Section 3.1.2.4). While such an alteration (and therefore a *Structural Impact*) cannot be avoided during *Live Forensics*, a *Post-Mortem Investigation* has better options in order to limit or entirely circumvent additional *Structural Impact*. With the investigated system already deactivated during a *Post-Mortem Investigation*, *Non-volatile Memory* can be obtained without further alteration to the system state (see Section 3.1.2.5). Here, mechanisms that prevent the alteration of the mass storage media in question would protect the investigation target. In this case, the mass storage media would be the investigation target. Such methods could be built-in mechanisms against such an alteration (*sm2,1*) or the use of external methods like a hardware-based write blocker (*sm2,2*).

This concept is not only relevant during *Data Gathering* but also during *Data Investigation* and *Data Analysis*. In this case, input files are processed. Some tools will readily alter the input files while some include measures to ensure that the input files cannot be altered (*sm3,1*). Providing protection against such a manipulation is readily achievable in these *Investigation Steps* by working on digital copies of the gathered data. This external mechanisms provide a required integrity of the input data, especially when combined with some methods to externally calculate hash values of the input data and comparing these during different steps of the forensic process (*sm3,2*).

The protection of the output of a forensic tool is addressed by *sm3*. In the case of *sm3,1*, the forensic tool itself offers methods in order to provide protection of the integrity and authenticity of its output. For example, this would be the case if the tool documents the actions performed and attaches a signature to the output. If the tool does not provide any mechanisms to support this, external methods can be used (*sm3,2*) as a part of the *Process Accompanying Documentation*. Such a protection increases the evidentiary value of the evidence in question. Even if an output is generated by the most reliable method, it is still important to discern if the output at hand is the same as the one generated by said method.

Forensic Process Model Criteria* Addressed by the *Classification Scheme for Forensic Tools and Methods

These three elements found in the *Classification Scheme for forensic tools and methods* in [BSI11] touch on important aspects of the forensic process. Hence, elements address some of the *Forensic Process Model Criteria* postulated in Section 2.3.2.4.

The *Untersuchungsvorraussetzung* (German for *„requirements for the (promising) usage of a forensic tool“*) touches on the necessary requirements for the use of certain methods and tools from a process viewpoint. It discusses if a given method requires certain actions to be taken beforehand in order to achieve meaningful success with the application of said method. Furthermore it states which actions must not be taken before. This touches the *Forensic Process Model Criteria* addressing the structure of the forensic process. These requirements can state if a method is only usable during *Live Forensics* or can also be used during a *Post-Mortem Investigation*. Hence, it supports the consideration if a given method offers enough benefit to be performed during *Live Forensics* to justify a potential *Structural Impact*. This addresses **PMC7**. Some of the requirements rely on actions performed during a *Strategic Preparation*. Hence, this element supports the inclusion of a pre-incident preparation in the forensic process (and addresses **PMC6**). These requirements might also dictate a logical sequence of tools to be used. Hence they help structure the forensic process as required by **PMC8**.

The element of *Datenschutzrelevanz* (German for *„relevancy of data protection concerns“*) covers constraints for the collection and use of evidence during a forensic process. This challenge is described by **PMC3**.

The *Schutzmaßnahmen* (German for *„Protection measures“*) aim to increase the integrity and authenticity of the forensic traces (**PMC1**). This is done by supporting integrity and authenticity of the forensic tools used, the sources processed and the output generated by these tools. Protection measures for the integrity and authenticity of the generated outputs support the *Process Accompanying Documentation*.

3.1.2.7 Comparison Between the [KDV15]-Model and the *Forensic Process Model Criteria*

Although only established during the course of this thesis (specifically in Section 2.3.2.4), various concepts present in the [KDV15]-model address a number of the *Forensic Process Model Criteria*. A short overview on the creation process of this forensic process model is given in Section 3.1 and Section 3.1.2 and shows why these aspects were addressed.

This section gives an overview on how the presented forensic process model addresses the *Forensic Process Model Criteria* postulated in Section 2.3.2.4. A summary overview can be found in Table 3.7. The table shows that most of the *Forensic Process Model Criteria* are addressed by the [KDV15]-model at least to some extent.

PMC1 (*Integrity and Authenticity of Traces*)

This *Forensic Process Model Criterion* is addressed by the inclusion of a *Documentation Phase* (Section 3.1.2.1) in the computer forensic process.

Especially the *Process Accompanying Documentation* is concerned with ensuring the integrity and authenticity of forensic evidence by documenting the forensic process in a reliable manner.

Table 3.7: Overlap between aspects of the [KDV15]-model and the *Forensic Process Model Criteria*

Forensic Process Model Criteria	Addressed by
PMC1 *integrity and authenticity of traces*	*Documentation Phase* (Section 3.1.2.1) *Structural Impact* (Section 3.1.2.4) *Classification Scheme* (Section 3.1.2.6)
PMC2 *error, uncertainty and loss*	*Structural Impact* (Section 3.1.2.4)
PMC3 *constraints for the collection and use of evidence*	*Classification Scheme* (Section 3.1.2.6)
PMC4 *Admissibility Factors*	*Classification Scheme* (Section 3.1.2.6)
PMC5 *Investigative Contexts*	*Strategic Preparation Phase* (Section 3.1.2.1) *Data Types* (Section 3.1.2.2)
PMC6 *Pre-Incident Preparation*	*Strategic Preparation Phase* (Section 3.1.2.1) *Classes of Methods* (Section 3.1.2.3) *Classification Scheme* (Section 3.1.2.6)
PMC7 *Live Forensics and Post-Mortem Forensics*	*Data Streams* (Section 3.1.2.5) *Structural Impact* (Section 3.1.2.4) *Classification Scheme* (Section 3.1.2.6)
PMC8 *Structures the process*	*Investigation Steps* (Section 3.1.2.1)

This can be achieved by, for example, documenting each step of the forensic process and including hash values or signatures for the outputs obtained by the use of forensic methods.

If such measures need to be handled by external tools or are already included by the tools used during the forensic process is described in the *Classification Scheme* (Section 3.1.2.6) which categorizes forensic tools based on their specific properties. A property described in this *Classification Scheme* identifies the presence of internal measures to ensure the integrity and authenticity of the forensic tool itself, the data source addressed by the given tool or the outputs of this tool.

Integrity and authenticity of traces can also be impacted if forensic methods alter the system state and therefore the evidence. This is described as *Structural Impact* (Section 3.1.2.4). The concept of *Structural Impact* describes how a forensic method alters the system state or the forensic process.

PMC2 (*Error, Uncertainty and Loss*)

PMC2 is not addressed in detail. The core concepts of the [KDV15]-model (*Investigation Steps*, *Data Types* and *Methods*) are describing the forensic process. The tools used during this process are not the specific focus of the model, with the exception of the *Classification Scheme for forensic tools and methods*.

The property *Beweiskrafttendenz* (German for *„Tendency for evidentiary value“*) in the *Classification Scheme for forensic tools and methods* takes **PMC2** into account. The property *Beweiskrafttendenz* (German for *„Tendency for evidentiary value“*) in the *Classification Scheme for forensic tools and methods* takes **PMC2** into account.

In addition, the discussion of *Structural Impact* addresses some of the aspects of **PMC2**. *Structural Impact* might cause the loss of data which could otherwise be gathered during the forensic process. However, addressing **PMC2** is not the focus of these two concepts.

PMC3 (*Constraints for the Collection and Use of Evidence*)

This aspect is addressed by the *Classification Scheme for forensic tools and methods*. The property of *Datenschutzrelevanz* (German for *„Relevancy of data protection concerns“*) covers constraints for the collection and use of evidence during a forensic process by stating the necessity of certain methods to address these concerns. These methods (pseudonymization and anonymization) are mostly motivated by privacy concerns (see Section 2.1.6). However, encryption could also be used to cover constraints in the case were business secrets are affected.

PMC4 (*Admissibility Factors*)

PMC4 concerns the *Admissibility Factors* and is partly addressed by the *Classification Scheme for forensic tools and methods*. This *Classification Scheme for forensic tools and methods* contains an element referred to as *Beweiskrafttendenz* (German for *„ Tendency for evidentiary value“*) which describes the potential evidentiary value provided by the output of a given forensic method. This element takes the *Admissibility Factors* into account but can only provide a superficial, generalized statement.

PMC5 (*Investigative Contexts*)

This *Forensic Process Model Criterion* is addressed by two major concepts of the [KDV15]-model. These concepts are the inclusion of a *Strategic Preparation* (Section 3.1.2.1) and *Data Types* (Section 3.1.2.2).

The *Strategic Preparation* is necessary in order to address the different starting points of forensic investigations as discussed in Section 2.3.2.2. Investigations in **IC1** usually require some form of incident detection. Means for an incident detection can only be prepared during a *Pre-Incident Preparation* which is covered by the *Strategic Preparation*. Furthermore, the incident detection itself could be considered a part of the *Strategic Preparation* as only the detection of an incident triggers an investigation.

While the *Strategic Preparation* is mostly relevant for **IC1** the *Data Types* support both *Investigative Contexts*. The *Data Types* classify potential forensic evidence based on how this evidence is processed and analyzed during the forensic process. **IC1** and **IC2** respectively usually require different types of data as evidence. For example, an investigation in **IC2** might rely on spreadsheets of financial data as evidence. This data can be analyzed by dedicated software designed for forensic accounting. An investigation in **IC1** might however require a list of the currently active network connections. In such a case, gathering and analyzing these two types of evidence differ widely.

PMC6 (*Pre-Incident Preparation*)

The *Pre-Incident Preparation* is covered by various concepts of the [KDV15]-model. The most pronounced of these concepts is the inclusion of a *Strategic Preparation Phase*, but the concepts of *Classes of Methods* and the *Classification Scheme for forensic tools and methods* also touch on this topic.

The *Strategic Preparation Phase* fulfills the function of a *Pre-Incident Preparation.* This *Investigation Step* allows for the general preparation of forensic readiness before a specific incident occurs. Hence, it covers the deployment and operation of measures which enable the detection of incidents. Furthermore, it includes methods deployed in order to provide additional data sources for potential evidence in the case of a forensic investigation.

The *Methods* (Section 3.1.2.3) from the [KDV15]-model also support the *Pre-Incident Preparation* by classifying certain forensic methods into *Classes of Methods.* If a certain method belongs to a given class, this might imply that these tools have to be installed on a system before an investigation is prepared.

A more detailed look on this topic is provided by the *Classification Scheme for forensic tools and methods.* This scheme directly includes a property referred to as *Untersuchungsvorraussetzung* (German for *„requirements for the (promising) usage of a forensic tool"*) which touches the necessary requirements for the use of certain methods and tools. This view is taken from a process viewpoint. It discusses if a given method requires certain actions to be performed before the forensic method can be used. Some of these requirements rely on actions performed during a *Strategic Preparation.*

PMC7 (*Live Forensics and Post-Mortem Forensics*)

The two concepts of *Live Forensics* and *Post-Mortem Forensics* (see Section 2.3.2.3) are taken into account by three underlying concepts of the [KDV15]-model not directly included in [KDV15]. These three concepts are the *Data Streams* (Section 3.1.2.5), the *Structural Impact* and the *Classification Scheme for forensic tools and methods.*

The *Data Streams* identify the origins of potential forensic traces. Three different *Data Streams* in (networked) Desktop Computers are identified and they imply different requirements for the gathering of these traces. The *Data Streams* of *Communication* and *Volatile Memory* are unavailable once a Desktop Computer is switched off and hence are only available during *Live Forensics.* Only *Non-volatile Memory* is available during *Post-Mortem Forensics.* Hence, identifying which methods and tools work on which *Data Streams* helps to identify which tools or methods are most suitable during either *Live Forensics* or *Post-Mortem Forensics.*

The *Structural Impact* describes how a certain action (including the use of a forensic tool or method) alters the system state and therefore affects the entire forensic process. *Structural Impact* might alter data in *Volatile Memory* and/or *Non-volatile Memory* locally and/or network wide. This would also include the unavailability of *Communication* and *Volatile Memory* after a Desktop Computer is switched off. This concept addresses the fundamental question if keeping a system active to gather certain pieces of evidence only available in *Live Forensics* is worth reducing the evidentiary value of the evidence which could be gathered during *Post-Mortem Forensics.*

The selection of specific tools is further supported by the *Classification Scheme for forensic tools and methods.* As mentioned before, this scheme includes a property referred to as *Untersuchungsvorraussetzung* (German for *„requirements for the (promising) usage of a forensic tool"*). These requirements include those that essentially require that a certain method or tool is used during *Live Forensics.* The scheme differs between those cases in which a Desktop System is still active and connected to the network and those in which a Desktop System is still active but disconnected from the network. One set of potential requirements aligns to *Live Forensics* while the other aligns to *Live Forensics with disconnected network*.

PMC8 (*Structures the Process*)

The six *Investigation Steps* in the [KDV15]-model structure the forensic process and hence address **PMC8**. These steps follow a logical and chronological order of the forensic process and hence support its execution. Tools and methods are assigned to specific *Investigation Steps* in order to support tool selection and decision making.

Summary

The [KDV15]-model for the forensic process touches on all *Forensic Process Model Criteria* to varying extent. In general, the *Forensic Process Model Criteria* which address the structure of the forensic process in general are well addressed while those regarding specific forensic tools and methods are addressed to a lesser extent.

PMC5, **PMC6**, **PMC7** and **PMC8** are focused on the structure of the forensic process and are addressed in a comprehensive manner by the *Investigation Steps* (see Section 3.1.2.1), including the *Strategic Preparation Phase*. The six *Investigation Steps* structure the forensic process and allow for a *Pre-Incident Preparation*. The decision on whether to perform *Live Forensics* or to rely on *Post-Mortem Forensics* is supported by the concept of *Structural Impact* (see Section 3.1.2.4). *Structural Impact* describes the impact of certain actions (like the deployment of a specific tool) on the forensic process. This includes local or network-wide alteration of data or the loss of access to entire *Data Streams* (like the *Main Memory* when switching of the Desktop Computer). The concepts of *Data Streams* and the *Untersuchungsvorraussetzung* (German for *„requirements for the (promising) usage of a forensic tool“*) from the *Classification Scheme for forensic tools and methods* (see Section 3.1.2.6) can be used to describe which forensic methods or tools can only be performed during *Live Forensics*. Furthermore, the *Untersuchungsvorraussetzung* (German for *„requirements for the (promising) usage of a forensic tool“*) from the *Classification Scheme for forensic tools and methods* describes the potential requirement of a *Strategic Preparation* (see Section 3.1.2.1).

PMC1 has a dual focus. At first, integrity and authenticity are determined during the gathering of a piece of evidence (during the *Data Gathering*). Then the integrity and authenticity must be maintained during the forensic process as established in Section 2.1.4. The degree of integrity and authenticity achievable during *Data Gathering* depends on the method or tool used to perform this gathering. Here the *Structural Impact* describes how previous actions could have impacted integrity and authenticity. Furthermore, the *Classification Scheme for forensic tools and methods* includes classifications for mechanisms of said methods and tools to protect the integrity of the data source. The *Documentation Phase*, especially in form of the *Process Accompanying Documentation*, supports the maintaining of integrity and authenticity of the gathered evidence during the forensic process. Again, the *Classification Scheme for forensic tools and methods* provides information if given forensic methods provide mechanisms to ensure the integrity and authenticity of the gathered evidence.

PMC2, **PMC3** and **PMC4** are focused on specific tools. This is understandable since these *Forensic Process Model Criteria* are closely aligned to *Process Challenges* (see Section 2.3.2.4). The selection of specific tools is not the focus of the [KDV15]-model. However, the core concepts of *Investigation Steps*, *Data Types* (see Section 3.1.2.2) and *Classes of Methods* (see Section 3.1.2.3) help to identify where a specific tool can be used effectively during the forensic process. Besides this, the *Classification Scheme for forensic tools and methods* partly addresses various aspects of the specific tools to a certain degree.

In general, it can be concluded that the [KDV15]-model covers the process side of the *Forensic Process Model Criteria* comprehensively. The *Forensic Process Model Criteria* which align to the *Process Challenges* and hence refer to specific tools are not addressed as comprehensively. However, the [KDV15]-model can be considered a comprehensive computer forensic process model.

3.2 An Approach to Adapt the [KDV15]-Model from Computer Forensics to Forensics in Cyber-Physical Systems

The preceding chapter defines the scope of this work by defining the relevant domains and the field of computer forensics. It gives an overview on the embedded systems used in Industrial Control Systems and Automotive IT (see Chapter 2). Also, the forensic process and the broad range of factors influencing a forensic investigation in general are introduced leading up to the establishment of the eight *Forensic Process Model Criteria* (**PMC**). As demonstrated, computer forensics is well-established in the domain of classical Desktop IT.

The first major part of this chapter described the [KDV15]-model for the forensic process (see Section 3.1) in detail. Most importantly the discussion on this forensic process model established that it is a comprehensive forensic process, although it addresses the eight *Forensic Process Model Criteria* to a varying degree.

This second major part of this chapter maps the path towards establishing a comprehensive computer forensic process model for cyber-physical systems based on the forensic process model laid out in Section 3.1.2. It discusses the conceptual work required in order to map a path towards transferring the computer forensic process from the [KDV15]-model for computer forensics in the Desktop IT domain to the ICS and Automotive IT domains.

A primary question is how similar these domains actually are. Answering this question requires a thorough analysis of the respective domains in order to identify similarities and differences. These similarities and differences form the foundation to investigate how the principles known from the classical IT domain (as detailed in section Section 2.1.1) can be applied to the ICS and Autmotive IT domains (in short: *cyber-physical domains*).

To serve as a solid foundation, this domain analysis needs to investigate all relevant aspects of the cyber-physical domain. These aspects are the employed hardware, the architectures in which this hardware is used and the scenarios in which these components are employed. These aspects cover the various sources for potential evidence - the *Data Streams* discussed in Section 3.1.2.5. This aims at understanding how a forensic investigation can be performed in these domains with a focus on **DG** (*Data Gathering*) and **DI** *Data Investigation*. While these aspects cover what components are used in which manner for what purpose, additional attention is warranted to also analyze how these systems are attacked in order to identify the traces various attacks could provoke within these systems in the different *Data Streams*.

An analysis of the specific hardware of given components covers which components are used within cyber-physical systems. It aims at identifying properties which influence the forensic process. For example a component which does not employ any programmable memory would be handled differently in a forensic process in contrast to one which does employ programmable memory. This analysis identifies the potentially available *Data Streams*, mostly with regard to *Volatile Memory* and *Non-volatile Memory* by identifying said memory and the interfaces to access this memory. In addition, the various communication interfaces are also of interest for the identification of a potential *Communication Data Stream*, since interfaces are the foundation for any communication.

The *Communication Data Stream* can be explored further by investigating in which manner these components are employed. This is done by analyzing the employed communication architectures. The communication architectures dictate the way data flows between the various employed components. Hence, the availability and accessibility can of potential *Communication Data Streams* can be judged.

Analyzing the purpose in which these components are used provides additional knowledge about the data flows between various components. This includes the analysis of certain communication strategies in terms of both, timing and content, and hence, the availability of potential traces for the forensic process. This availability might have an impact on the forensic process.

The last major aspect of these domains is an analysis of the attacks on respective ICS and Automotive IT systems. This analysis gives information on what attack vectors attackers use and which actions they perform within the compromised systems. Therefore, the identification of potential forensic traces caused by the attack is possible. In addition, this aspect serves as anchor to validate the conclusions made in the other steps of the analysis and the potential alterations to a model for cyber-physical system forensics.

Analysis of these four central aspects should answer the questions which forensic traces might be available in ICS and Automotive IT systems, how they can be accessed and which specific implications for accessing this data are linked to these domains. This addresses **Research Question 1** by identifying 29 (ICS domain) plus 25 (Automotive IT domain) *Influence Factors* (*IF*) which impact the conduct of forensics in these domains. These *IFs* are the foundation to identify challenges not (or insufficiently) covered by the current [KDV15]-model for forensic investigation in light of these domains. Hence, it offers valuable insight for revisiting the forensic process model in the light of these cyber-physical domains.

After this fundamental analysis, the current model for a forensic process (as detailed in section Section 3.1.2) is reviewed under the scope of the peculiarities of the ICS and Automotive IT domains as identified beforehand. This addresses **Research Question 2** by identifying 6 (ICS domain) plus 6 (Automotive IT domain) *Forensic Process Consequences* (*FPC*) which impact the overall conduct of the forensic process and these domains..

The survey of potential forensic traces available in cyber-physical environments enables a review of the established *Data Types*. The survey on the availability and accessibility of forensic traces fuels a discussion on how this impacts the various *Investigation Steps*. Furthermore, the *Classes of Methods* are discussed and revisited according to the findings of the analysis. This discussion leads to necessary alterations to the forensic process model when employing the [KDV15] forensic process model in cyber-physical systems in order to answer **Research Question 3**.

The altered model is evaluated by employing it in a complex case study. To evaluate the usefulness of the altered model, this case study has to take place in a controlled environment, where the quality of the results obtained using the altered forensic model during an investigation can be compared to the achievable results without using the any forensic process model at all or the unaltered [KDV15]-model.

Based on these considerations, the fundamental concept of this work in order to bring insight from the field of computer forensics into cyber-physical systems forensics follows these steps:

- *Step 1* **Domain Analysis**
  This analysis aims at identifying similarities and differences between the classical Desk-

top IT on one hand and the Automotive IT and ICS domains on the other hand. This analysis breaks down into distinct parts:

- *Step 1.1* Analysis of employed components
  This includes the analysis of computing units and attached communication interfaces and/or input/output devices in order to identify constraints inherent to the employed hardware and their potential impact on the forensic process.
  This step is necessary in order to judge the possibility to gather *Volatile Memory* and *Non-volatile Memory* since this memory is usually internal to these components (or at least directly attached to them).
  As this step analyzes the basic building blocks of the systems in question this step has to be performed first.
  This analysis is conducted as a hands-on-investigation into a selected range of components as available to the author during his research work, supplemented by a literature research in order to cover a broader range of different components (especially with regard to cover different vendors).

- *Step 1.2* Analysis of system architectures and communication
  This includes the analysis of the employed system architectures to gain an understanding of the communication hierarchies in question. This aims at identifying the characteristics of communication flows as a foundation to discuss potential impact on the forensic process.
  This step is necessary in order to judge the possibility to gather *Communication* since this *Data Stream* is situated within the communication architecture.
  As this step analysis how the basic building blocks of the systems in question are put together this step has to succeed the analysis of the basic building blocks.
  This analysis is conducted by employing a literature research as well as a hands-on-investigation into a selected range of specific implementations accessible to the author during the course of his research work.

- *Step 1.3* Analysis of scenarios
  This includes the analysis of the processes in which these systems are usually used (these use-cases) and the potential impact of these processes on the behavior of the employed systems and the potential impact on the forensic process. In addition, the nature of employment of these systems itself might have an impact on the forensic process.
  As this step analyzes for what purpose the system architectures of the systems in question are used, this step has to succeed the analysis of these architectures.
  This analysis is conducted by employing a literature research as well as a hands-on-investigation into a selected range of specific use-case scenarios as accessible to the author during the course of this research work.

- *Step 1.4* Analysis of attacks
  This includes the analysis of known attacks in order to identify attack patterns and potential forensic traces these attacks could have created. This serves as a foundation for the validation of the previous findings and the potential necessary adaptation to the forensic process.

As attacks aims at gaining information about or disrupting an use-case a meaningful analysis of these attacks has to rely on an understanding of the use-cases. Hence, this step has to succeed the analysis of the various use-cases.
This analysis has to rely on literature research and the analysis of known proof-of-concept implementations of various attacks, as well as a reflection of the possibility to perform the various basic attacks within the architectures and scenarios employed in cyber-physical systems.

- *Step 2* **Revisiting the forensic model**
  This discusses the impact of the findings from the analysis of the various aspects on the constituting parts of the forensic model (as detailed in section Section 2.1.1):

  - *Step 2.1* Revisiting *Investigation Steps*
    This includes an analysis whether the *Investigation Steps* (see Section 3.1.2.1) befit an investigation into cyber-physical systems. Potential alterations to these investigation steps in order to better reflect the specifics of cyber-physical systems also take place in this step. This might include the investigation steps on a macro-level (order, overall aim or task of the *Investigation Step*) or the micro-level (given procedures or mechanisms within the given step).
  - *Step 2.2* Revisiting *Data Types*
    This includes an analysis whether the *Data Types* (see Section 3.1.2.2) are able to cover data in cyber-physical systems. Potential alterations to the *Data Types* in order to better reflect the specifics of cyber-physical systems also belong to this step.
  - *Step 2.3* Revisiting *Classes of Methods*
    This includes an analysis whether the *Classes of Methods* (see Section 3.1.2.3) are useful for an investigation into cyber-physical systems. Potential alterations to the *Classes of Methods* in order to better reflect the specifics of cyber-physical systems also take place in this step.
  - *Step 2.4* Revisiting other aspects
    This includes an analysis whether the other aspects included within the [KDV15]-model (*Structural Impact*, *Data Streams* and the *Classification Scheme for forensic tools and methods* - see Section 3.1.2.4, Section 3.1.2.5 and Section 3.1.2.6 respectively) are useful for an investigation into cyber-physical systems. Potential alterations to these aspects in order to better reflect the specifics of cyber-physical systems also take place in this step.

- *Step 3* **Evaluation of the revised forensic model**
  This evaluates the usefulness of the altered forensic model in investigating incidents in cyber-physical systems.

  - *Step 3.1* Evaluation based on case studies
    This includes the application of the altered model for the forensic process to a case study. The case study will take place in a controlled environment in order to compare the results after the application of the forensic process to the achievable results.

4. Step 1: Domain Analysis

The preceding chapter (see Section 3.2) mapped the road towards bringing lessons from the desktop IT domain to the ICS and Automotive IT domains. This chapter describes *Step 1* of this road.

Analyzing the ICS and Automotive IT domains is necessary in order to investigate the respective impact of their properties on the forensic process. This analysis is structured along the path laid out before. Hence, this chapter starts with analyzing the employed components, moves on to the employed system architectures and then takes a look at the scenarios in which these components and architectures are employed. Finally, known potential attack patterns are analyzed in order to understand the nature of the attacks on these systems.

The analysis starts with the Desktop IT domain, then regards the ICS domain and then moves to the Automotive IT domain. Summaries conclude the respective sections. This analysis answers **Research Question 1**.

During the course of this section, various examples of components and tools are used. These components and tools are exemplary but chosen to represent the properties of the given domains to the greatest extent. For the respective components the market shares of various vendors were used to select examples from the market leaders which should be more common than niche product. However, these common examples are supplemented with some more specialized niche products. Once the components are analyzed and the contents of their respective *Data Streams* are identified, access to this data is discussed. The tools here represent a selection. Whenever possible open source solutions were chosen. When this was not possible free solutions were chosen. If none of these were available commercial but freely available tools were taken into considerations. The aim of this analysis is to identify the specific properties that impact the forensic process in these domains. When possible a range of tools will be presented in order to find common characteristics of these tools. Obviously, this approach cannot cover any and all tools. And while there is a general lack of freely available tools tailored to the forensic use in the ICS and Automotive IT domain such tools might exist in vendor-internal working groups. Although, practical experience in the field convinces the author of this thesis to assume that this is only the case in a very limited capacity.

4.1 *Step 1.1* Analysis of Employed Components

The domain analysis starts with the computing units and the attached periphery. This periphery includes attached communication interfaces and/or input/output devices. This process aims at identifying properties and constraints inherent to the employed hardware and their potential impact on the forensic process.

This analysis covers the *Data Streams* (see Section 3.1.2.5) of *Non-volatile Memory* (Mass Storage) and *Volatile Memory* (Main Memory) since these two types of memory are either contained within the employed components (in the case of working memory or internal storage) or directly attached to it (in the case of removable storage media). After the identification of the potentially available forensic traces within these two *Data Streams* using the *Data Types* (see Section 3.1.2.2), the accessibility of these traces is discussed. The discussion of the specific domains is concluded by a summary of the findings in regards to the use of *Non-volatile Memory* and *Volatile Memory* for forensic purposes.

4.1.1 Desktop IT Components

According to the definition established in Section 2.2.1, the domain of Desktop IT covers computer systems designed to receive, store, manipulate and transmit data. This category describes anything that comes to mind when thinking about classical computer systems found within countless offices around the world, used to receive and send correspondence or to create and edit documents. This is the domain of general purpose computers and their assorted periphery. The overall systems consist of a computer system and its periphery, covering input and output devices.

The computer system consists of computing units and memory. Modern general purpose computers usually have a main computing unit (Central Processing Unit - CPU) and other supporting computing units which are optimized towards certain tanks (a prime example being the Graphics Processing Unit - GPU). These computing units usually have a specifically assigned working memory required for their computational tasks. This memory represents the *Volatile Memory Data Stream* in Desktop IT systems. In addition, the overall system has a general purpose working memory which supports the computing units by allowing (relatively) fast access to a given set of data for a limited amount of time. In contrast, a larger portion of mass (and slower to access) memory is also installed within a computer system, usually in form of a hard drive. This storage represents the *Non-volatile Memory Data Stream.* Internal communication is performed by using cable connections or being directly attached to a shared main circuit board (motherboard or mainboard). A computer system also has an array of interfaces to communicate with its periphery, external memory or other computer systems.

Typical input devices for general purpose computers are keyboards and mouse. Although, the use of touch screens (which double as input and output devices) is not unknown of, it is a far from common occurrence. Typical output devices are monitors or loudspeakers. Printers are also quite common and could be considered as output devices.

In general, components for general purpose computers are cheap and readily available. This is due to the abundance of general purpose computing systems. These systems are usually employed inside offices and homes which negates the need to resist any adverse conditions (like humidity, temperature or concussion). This also reduces the cost of the specific components.

Since all the components are relatively cheap, these systems usually have an abundance of memory (both volatile and non-volatile - but especially the later) and computing power.

The modification of a general purpose computer system is a task performed with relative ease and regularity. Should a component within such a general purpose system fail or not be able to fulfill increasing requirements for its performance anymore it can be easily replaced and/or upgraded.

A better overview on capabilities and properties of these general purpose computers is achieved by examining some examples for contemporaneous computer systems. Such examples can be found in Table 4.1. These systems have been selected to represent the six Personal Computer vendors with the highest market share in the fourth quarter of 2018 (see [Gar19]). The list includes desk-based Personal Computers as well as notebook Personal computers (as does the underlying data for the market shares).

The overview presented in Table 4.1 shows that these *Personal Computer Examples* share some characteristics. The *Personal Computer Examples* have abundant computing power and an extensive amount of main memory (*Volatile Memory*) and storage (*Non-volatile Memory*). Even though these examples represent the latest generation of computer systems, these observation even holds true for slightly older models. In comparison to the components used in ICS and Automotive IT (see Section 4.1.2 and Section 4.1.3 respectively), classical computer systems have abundant resources. For example, the cache included in many of the computing units is more extensive than the entire working memory of computing units in the other investigated domains. The same goes for graphical co-processors which have more computational power than some computing systems in those other domains. Even for mobile laptops, as shown in *Personal Computer Examples 4, 5 and 6*, the computing power and memory capacity is still higher - usually in the order of magnitudes.

Another interesting factor is that these systems are highly customizable. Reconfiguring the components to include even more storage, a more powerful processor or additional devices is common. Hence, these systems usually rely on common hardware interfaces to put these components together.

Some of these properties influence the topic of forensic investigations and potential attacks and can already be deducted from taking these basic characteristics into consideration. While this domain forms the baseline, some aspects are still worth being pointed out as they stand in a strong contrast to the other domains:

- *Cheap memory*
  The fact that components for general purpose computers are comparatively cheap leads to an abundance of memory and computing power. Such systems are able to store extensive log files due to having little constraints in terms of mass storage. These log

[1] https://www8.hp.com/h20195/v2/GetDocument.aspx?docname=4AA7-3215ENUC/, 19/01/2020

[2] https://www.dell.com/en-uk/work/shop/xps-desktops/xps-8930-desktop/spd/xps-8930-desktop, 19/01/2020

[3] https://www.acer.com/ac/en/US/content/model/DT.BAPAA.011, 19/01/2020

[4] https://www.asus.com/uk/Laptops/ASUS-Laptop-15-X509FA/specifications/, 19/01/2020

[5] https://www.lenovo.com/gb/en/laptops/thinkpad/x-series/X1-Carbon-Gen-7/p/22TP2TXX17G, 19/01/2020

[6] https://www.apple.com/uk/macbook-pro-16/specs/, 19/01/2020

Table 4.1: Computing power, memory and interfaces of exemplary selected contemporaneous Personal Computers

Identifier			
CPU	RAM	Memory	Interfaces
Personal Computer Example 1: HP Z2 Tower G4 Workstation[1]			
Intel® Xeon® E-2286G 4.0 GHz base frequency 12 MB cache 6 cores	128 GB 2DDR4-2666 ECC SDRAM	6 TB SATA7200HDD	dual-port GbE NIC
Personal Computer 2 Example: Dell XPS 8930[2]			
9th Gen Intel® Core^{TM} i5 9400 6-Core 9MB Cache up to 4.1GHz	8GB DDR4 2666MHz	1TB 7200RPM SATA HDD	802.11ac Bluetooth 4.1
Personal Computer 3 Example: Acer Aspire TC-885-UR18[3]			
Intel Core^{TM}i3-8100 Quad-core 3.60 GHz	8 GB up to 32 GB	256 GB SSD	Gigabit Ethernet IEEE 802.11ac USB
Personal Computer 4 Example: ASUS Laptop 15 X509FA[4]			
Intel® Core^{TM}i7 8565U	4 GB Onboard up to 16 GB SDRAM	500GB 5400 rpm SATA HDD 256GB PCIe® x2 SSD	Wi-Fi (802.11 ac) Bluetooth® 4.2 USB
Personal Computer 5 Example: Lenovo ThinkPad X1 Carbon (Gen 7)[5]			
Intel® Core i5-8365U 1.60GHz up to 4.10GHz with Turbo Boost 4 Cores, 6MB Cache	8GB LPDDR3	256GB Solid State Drive	Intel® Wireless AC 9560 Bluetooth Version 5.0 vPro Onboard
Personal Computer 6 Example: Apple MacBook Pro 16[6]			
2.6GHz 6core Intel Core i7 Turbo Boost up to 4.5GHz 12MB shared L3 cache	16GB 2666MHz DDR4 onboard	512GB SSD	802.11ac

files would contain forensic traces and could be used during a forensic investigation. Usually, applications run on these computer systems will create log files.

- *Cheap computing power*
  Cheap computing power favors the use of various security measures. Generally speaking, running dedicated security software does not degrade the systems performance in a dramatic manner. Hence, it is common to find at least some basic security software running on most general purpose computer systems. While this software can enable the detection of attacks it might also provide additional forensic traces during an investigation.

  Having abundant computing power also allows for the generous use of encryption for communication or file storage.

- *Easy to access and customize*
  Interfaces between the various components are standardized in order to allow for easy customization and upgrading of the computer systems. These standards are readily available.
  In addition, the components are relatively easy to access and remove, especially in the case of more spacious Desktop computer. In the case of some specific laptop vendors, this might not be correct.

  In each case, removing the non-volatile storage (e.g. hard-drive) from such a computer system is a relatively minor task. Interfaces in order to access the storage medium are standardized and commonplace.

A Desktop Computer system also includes an operating system. According to [Tan07] the operating system is tasked with managing the access to the hardware components. Different operating systems are in use with Desktop Computers. An overview of contemporaneous market shares of various operating systems used in Desktop Computers can be found in Figure 4.1 which is based on the statistics provided by [Sta20][7]. The most common is *Microsoft Windows*[8] with more than 75% market share across the entire time frame presented in Figure 4.1. A major share is held by Apple[9] OS X with around 15% market share. Among the remaining market share, GNU/Linux [10] is of note with around 1.5% of market share. In general, it can be assumed that a Desktop Computer employs a commonly available operating system with standardized and widely available components. It is common for the different operating systems used in the Desktop IT domain to share file formats, especially in the case of **DT8**.

Each operating system provides access to *Non-volatile* and *Volatile* memory. This access falls under the *Methods* of **OS** (*Operating System* - see Section 3.1.2.3 for the introduction on the *Classes of Methods* within the [KDV15]-model for the forensic process). The exact details on how the access is performed vary.

The following sections discuss which *Data Streams* available in Desktop IT systems are of potential use during a forensic investigation and how the respective *Data Types* (see Section 3.1.2.2) can be gathered (during *Data Gathering*) and investigated (during *Data Investigation*).

[7] https://gs.statcounter.com/os-market-share/desktop/worldwide, 11/05/2020
[8] https://www.microsoft.com/en-gb/windows/, 11/05/2020
[9] https://www.apple.com/, 11/05/2020
[10] https://www.gnu.org/gnu/linux-and-gnu.en.html/, 11/05/2020

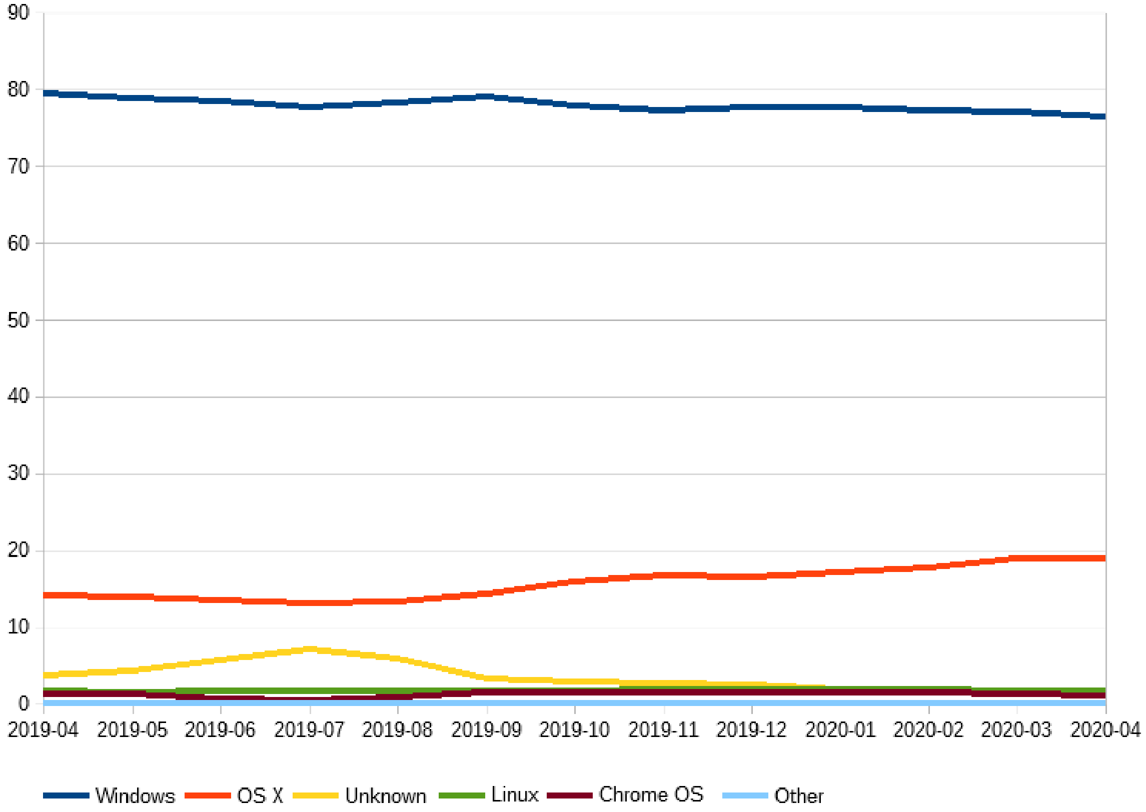


Figure 4.1: Market share of various operating systems used in Desktop Computers between April 2019 and April 2020 based on [Sta20]

4.1.1.1 Availability of *Data Types* in *Non-volatile Memory* in Desktop IT

The general abundance of storage, especially mass storage, leads to a potential abundance of forensic traces in the *Data Stream* of *Non-volatile Memory.* Since mass storage is readily available there is little need to save storage space. In turn, many applications create log files which might be of potential interest for forensic investigations - especially in the context of **IC2**. This includes the operating system itself. Its representation in the mass storage contains *hardware data* (**DT1**), *configuration data* (**DT4**), *network communication data* (**DT5**), *process data* (**DT6**) and *session data* (**DT7**). The mass storage is, in general, managed by a file system. The file system itself contains *details about data* (**DT3**). In addition, the file system might provide raw access to the *raw data* (**DT2**). Since Desktop IT systems are used to create, edit or consume media files, the storage medium might also include *user data* (**DT8**) which are especially useful for forensic investigations into **IC2**. Hence, all *Data Types* are potentially present within the *Non-volatile Data Stream* of a Desktop Computer system.

4.1.1.2 Gathering and Investigating the *Non-volatile Memory* in Desktop IT

This data can either be accessed during *Live Forensics* or *Post-Mortem Forensics.*

During *Live Forensics*, access is only possible by using the methods provided by the **OS** (*Operating System)* in conjunction with the **FS** (*File System).* Using these methods to access the mass storage alters the forensic traces by influencing the operations performed by the operating system and potentially by changing the contents of the file system (file access time stamps or a potential file system journal). Hence, these imply a *Structural Impact* on the forensic process by lowering the authenticity of every piece of evidence gathered thereafter (see Section 3.1.2.4 for the definition and discussion on *Structural Impact* and Section 2.1.4 for the discussion on *Evidence Dynamics* which makes this concept necessary in the first place). Hence, accessing *Mass Storage* during *Live Forensics* can only be recommended in the case specific information would be of more use during *Live Forensics* than during *Post-Mortem Forensics*. For example, the file access timestamps of executable files could be used to determine if they have been recently executed in order to inform a search for artifacts of these executables in the *Volatile Memory* which is only available during *Live Forensics*.

During *Post-Mortem Forensics*, the *Non-volatile Memory* can be easily accessed due to the standardization of the components and the accessibility of the components. Removing *Non-volatile Memory*, like hard drives, from these computer systems is a comparatively easy task. Hardware able to interface with this storage is readily available since these components are standardized and commonplace. This enables the creation of a forensic copy of the storage medium which can then be further investigated during the Data gathering (**DG**). Reliable tools geared towards forensic use are available for this process. An example is *dcfldd*[11] which offers additional functionality to ensure that the created copy is indeed a bit-for-bit match of the non-volatile storage medium in question.

Methods of **FS** are required in order to extract files from the mass storage (or a forensic copy of the mass storage) during the Data investigation (**DI**). The file systems used in Desktop computers are, in general, standardized and implementations are readily available. Beside the use of general purpose tools for file access implementations geared towards forensic use are available. An example is *Autopsy*®[12]. It offers additional functionality to browse the file system, to review certain files and to document the actions performed.

Hence, the access to the *Non-volatile Memory Data Stream* is comparatively easy and offers access to a comparatively high amount of forensic traces.

Both approaches are well-researched and a large number of publications address the **DG** and **DI** of *Non-volatile Memory* in the form of mass storage in Desktop computers. See [Arn17], [Cas04] or [BSI11] for examples on publications concerned with performing investigations into the *Non-volatile Memory* in classical Desktop IT systems.

4.1.1.3 Availability of *Data Types* in *Volatile Memory* in Desktop IT

Main memory is also abundant and offers a high amount of potentially relevant forensic traces. The *Volatile Memory* has the same potential contents as the *Non-volatile Memory*. However, the contents in the main memory will belong to active or recently active processes. For example, if a word processing application is currently running within the computer system **DT8** in form of the currently edited document will be available. The same holds true for any other type of application. Usually, the operating system will be loaded and will have

[11] http://dcfldd.sourceforge.net/, (18/03/2020)
[12] https://www.sleuthkit.org/, (18/03/2020)

need to store some **DT1**, **DT4**, **DT5**, **DT6** and **DT7** in the main memory for fast access. Since the operating system usually has various drivers for file systems loaded **DT3** within the file system might also be present. However, **DT3** can also be available as meta data about open files or buffered network communication. Since raw access to the main memory is also possible, **DT2** is also available. The *Volatile Memory Data Stream* of a Desktop Computer system might contain all *Data Types*.

4.1.1.4 Gathering and Investigating the *Volatile Memory* in Desktop IT

Even with standardized interfaces, the access to the main memory is still a complex task.

Beyond the trivial solution of a virtualized system in which the entire main memory can simply be dumped by the hypervisor gathering data from the main memory will always involve the operating system currently managing this memory. Obviously, this is only possible during *Live Forensics*. Hence, all methods used during **DG** to gather *Volatile Memory* must rely on requests to the operating system (and hence employ the *Method* **OS**). If the respective operating system is outside the direct control of the investigator the integrity of the given answers might be in doubt due to the fact that a manipulation of the underlying operating system would also falsify the data returned. This challenging aspect is discussed in more detail in [Car06]. In addition, every request for main memory to the operating system inevitably alters the system state. This *Structural Impact* is a concept introduced in Section 3.1.2.4 while various forensic process models elaborate on the usage of *Live Forensics* (see Section 2.3.2.1).

Data can be gathered in two forms. It can either be captured as **DT2** or as any other form of data (specifically **DT1**, **DT4**, **DT5**, **DT6** and **DT7**). The difference here is the abstraction level. In the case of **DT2**, a chunk of not yet interpreted data is taken from the main memory. In the second case, a certain piece of information (represented by any of the other *Data Types*) is requested from the system.

In this case, the request not only consists of the *Data Gathering* (**DG**) of the requested data but also of the *Data Investigation* (**DI**) in the form of the interpretation of this data. This is often the case when using methods provided by the **OS** since these represent internal functions usually used for system diagnostics. An example includes accessing the *procfs*[13] under a GNU/Linux operating system. In this case, **DG** and **DG** cannot be separated as only the interpretation of **DT2** provided by the *Method* of **OS** is accessible to the investigator. Another example for such an approach provided by a *Method* of the **OS** would be the *netstat*[14] command under a Windows operating system. This process is close to what [LCLW14] describes as *Runtime Interrogation*:

> *„**Runtime interrogation** enables you to quickly sweep across an entire enterprise and check for specific indicators in physical memory (instead of capturing a full memory dump from each system).“*

There are methods besides those provided by the **OS** which can produce a similar output with a potential forensic value. Malware scanners which access the working memory in order to

[13] http://man7.org/linux/man-pages/man5/proc.5.html, 17/03/2020

[14] https://docs.microsoft.com/de-de/windows-server/administration/windows-commands/netstat, 17/03/2020

scan for malware signatures are an example for *Explicit means of intrusion detection* (**EMID**) which also uses methods of the **OS** and then presents an interpretation of this **DT2** which carries a potential value for a forensic investigation. This is generally only possible because of the extensive access rights granted to such Malware scanners, but it is possible that other software performs similar functions.

If the data is gathered in the form of **DT2** tools that are able to interpret this raw data during the **DI** are necessary.

Gathering this **DT2** during the **DG** from the main memory (*Volatile Memory*) in the first place is a complex task. This process is described in detail by [LCLW14], which differentiates between *Hardware-based Acquisition* and *Software-based Acquisition*. In the first case, a special hardware-interface is used in order to gather the *Volatile Memory*.

One such hardware-interface present in some computer systems is an IEEE 1394 interface, often referred to as FireWire. Its usefulness to gather *Volatile Memory* has been examined in various publications (with an overview provided by [ZWZ$^+$10]). However, it is not a very common interface in contemporaneous Computer systems. None of the *Personal Computer Examples* presented in Table 4.1 does include a FireWire interface. In addition, according to [LCLW14] this method only works if no more than 4 GBytes of main memory are present. This greatly reduces usability since only *Personal Computer Example 4* comes with 4 GBytes of main memory - and even that only in certain configurations.

In addition, such a hardware-interface has to be present in the computer system in the first place since installing it would require a system restart and the loss of the current contents of the main memory. Hence, the *Volatile Memory Data Stream* would be lost in this case. Here, a *Strategic Preparation* (**SP**) is necessary in order to include such a system.

If a greater amount of main memory is installed, dedicated hardware is required. Currently, there exists only one commercially available piece of hardware. This hardware is called *CaptureGUARD*[15] and its various iterations have been cited in research about memory forensics for some time (for example in [LCLW14]). This piece of hardware comes at an extensive cost which makes its widespread deployment highly unlikely. This is especially problematic since such a device would have to be installed during a **SP** in order to access the current state of the main memory.

Apart from this, research implementations for such hardware devices exist. An approach using Direct Memory Access (DMA) and the PCI bus is suggested in [CG04]. This work also contains the *Tribble-Card* as a proof-of-concept device implementing this approach. The *Tribble-Card* was used to retrieve about 256 MByte of working memory from a test system. While the PCI bus is fast enough to transfer 256 MByte, it would most likely not suffice to transfer the massive amounts of working memory used in contemporaneous Desktop Computers (see Table 4.1). Using different interfaces for the transfer of the data once obtained seems possible, however. The usage of DMA to access the *Volatile Memory* is also used by other approaches. For example, [RS12] uses it to circumvent security mechanisms. This work could be used as a foundation to implement another device to gather *Volatile Memory* from contemporaneous Desktop Computers.

There exist various tools in order to acquire portions or the entirety of the main memory by *Software-based Acquisition*. A comprehensive list of these tools can be found in [LCLW14].

[15] https://www.bluerisc.com/captureguard/, 18/03/2020

However, since they all share some characteristics which influence the forensic process they can be considered without examining the given tools in detail.

All these tools require access rights to the main memory. This usually requires system administrator privileges. Also this software alters the main memory. At the very least **DT6** is influenced by the starting of a new process. Sections of *Volatile Memory* which have been unassigned before and could contain artifacts of potential evidence will be used by the tool and overwritten. Also, creating such a memory dump takes time in which the system performs additional actions and hence alters its state. A potential attacker (or a malicious piece of software) could use the time to cover potential tracks or cause more damage to the system. Hence, in order to reduce the time between the incident and the finalization of a memory dump, such a tool capable of dumping the memory could be installed during a *Strategic Preparation* (**SP**). Else, potential installation routines would also cause more alteration to the system state and delay.

An example of the use of *Software-based Acquisition* during a forensic process is provided in Section D.3 which is based on the scenario used in [**KHA$^+$09**].

When the **DT2** is finally collected **DI** can be performed after the system itself has been deactivated. This offers the additional benefit of being able to ensure that the investigated data is not altered by the investigation itself. Here, tools for the interpretation of this data are necessary. One tool of potential use for this purpose is *Volatility*[16]. This tool allows for the extraction of **DT1**,**DT4**, **DT5**, **DT6**, **DT7** and **DT8** from **DT2**. *Volatility* is well-established in the forensic community as various publications using this framework show (for example, [RC14], [GLB13], [Rob13] and [Mac13]).

No matter whether performing *Runtime Interrogation* to query for specific sets of information or if obtaining **DT2** using *Software-based Acquisition*, all memory access is handled by the **OS** and hence has to be performed by requesting access from the **OS**. All these requests have to rely on the fact that the underlying system is not compromised and able to fulfill these requests correctly and completely. This might not be given in the case of the presence of sophisticated malicious software which could alter or redirect such requests (see [ELS17] for details on how malicious software can prevent diagnostic requests from obtaining information about the processes triggered by this software).

4.1.1.5 Availability and Access to the Forensic *Data Streams* in Desktop IT

Memory is abundant in Desktop IT systems. The extensive amount of mass storage incites software vendors to include the creation of log files in their applications. These log files can support a forensic investigation, especially for investigations performed in **IC1**. All *Data Types*, including **DT8** (which is mostly useful for investigations carried out in **IC2**) can be found in the *Non-volatile Memory* and in the *Volatile Memory*.

Hence, the *Non-volatile Memory* is able to provide forensic evidence for investigations into Desktop IT systems. **DG** for *Non-volatile Memory* in Desktop Computer Systems is well-researched and can be performed in *Post-Mortem Forensics* by reliable tools geared towards forensic use. Dedicated software to interpret these forensic copies during **DI** are available. This is a standard approach for many Computer forensic investigations. Access during *Live Forensics* is also possible, albeit restricted to using methods of *Operating System* and *File*

[16] https://www.volatilityfoundation.org/, 18/03/2020

System. Integrity and authenticity of the gathered, investigated and analyzed traces is generally high due to the availability of dedicated tools for forensic use. Hence, the evidentiary value is high.

Access to *Volatile Memory* is a complex topic. There exist some software solutions designed to capture portions of the *Volatile Memory* during **DG**. These tools require extensive system access privileges and cause a high *Structural Impact*. The interpretation of this *Raw data* (**DT3**) is possible with the help of dedicated software used during **DI**. Another approach is the *Runtime Interrogation* in which certain areas of the *Volatile Memory* are queried using methods which might be part of the **OS** or an **EMID**. This could be understood as performing diagnostic requests to the operating system.

All methods to gather information from the *Volatile Memory* not using specific hardware-interfaces have to rely on the operating system to provide them access to the *Volatile Memory*. Hence, they all have to rely on the operating system. If it cannot be confirmed that the operating systems is not compromised, authenticity suffers and the evidentiary value is reduced.

Specific hardware-interfaces to capture *Volatile Memory* can circumvent this by giving direct access to the *Volatile Memory*. Such devices would have to be installed in the **SP** before an incident happens. However, these devices are rare, expensive or academic proof-of-concept. It is unlikely that one of these devices is available.

While integrity and authenticity of traces gathered during *Live Forensics* from the *Volatile Memory* is never optimal, the evidence gathered in this manner might still prove useful for the forensic investigation.

4.1.2 Industrial Control Systems Components

An introduction into the domain of Industrial Control Systems (ICS) has been provided in Section 2.2.2. The following section establishes a forensic understanding of these ICS components. After discussing the contents and accessibility of *Non-Volatile Memory* and *Volatile Memory* in ICS a summary is provided in Section 4.1.2.5.

ICS perform functions required in order to achieve industrial objectives. At a fundamental level they consist of four basic types of components. These components consist of actuators, sensors, processing units and the communication wiring between them. Sensors collect environment information whilst actuators manipulate the environment. The processing units in ICS are Programmable Logic Controllers (PLCs) reading the sensors and driving the actuators. The definitions established in Section 2.2.2 can be found in Section B.6.

The processing units are the most interesting of these components from the viewpoint of computer forensics. These processing units compute data about the physical process provided by the sensors and generate control signals for the actuators. Hence, they are computer systems. As such, they contain a processor, memory and interfaces in order to facilitate communication with sensors, actuators and other computing units.

An overview on the characteristics of some selected ICS processing units is presented in Table 4.2. This selection aims at including a diverse and representative range of ICS computing units. These computing units are generally referred to as *Programmable Logic Controllers* (PLCs) or *Speicherprogrammierbare Steuerung* (SPS, German for *„Programmable*

Table 4.2: Computing power, memory and interfaces of exemplary selected ICS Processing Units

Name			
CPU	Program Memory	Data Memory	Interfaces
ICS Processing Unit Example 1: Siemens Simantic S7-1500 CPU 1515-2 PN[17]			
	500 KB	3 MB	2x PROFINET IO IRT (2 Port) 1x PROFINET IO RT
ICS Processing Unit Example 2: Siemens Simantic S7-1500 CPU 1518F-4 PN/DP MFP[18]			
	6 MB	20 MB	2x PROFINET IO IRT (2 Port) 1x PROFINET IO RT 1x PPROFINET (Gbit) 1x PROFIBUS DP
ICS Processing Unit Example 3: Allen-Bradley 5580 Controllers[19]			
	40 MB		Embedded Ethernet (1 GBps) USB (Programming)
ICS Processing Unit Example 4: Allen-Bradley MicroLogix 1100[20]			
	4 KB	4 KB	Embedded Ethernet (100 MBps)
ICS Processing Unit Example 5: Fuji Electric SPF[21], 12/03/2020			
16-bit OS /Executing Processor	8000 x 32 bits = 250 KB	20000 x 16 bits = 312.5 KB	
ICS Processing Unit Example 6: ABB PM573-ETH[22]			
Speed for 1000 instructions ... of Logic = 0.05 0.06 ms of Word = 0.05 0.09 ms of Floating-point = 0.5 0.70 ms	512 kB	512 kB	Serial FBP Ethernet

Controller“). Another term used to describe these components is *Field Devices* since these components are used within production fields.

The common use of the German term SPS already points to the market share of the German automation manufacturer Siemens. According to [Daw18], Siemens and Rockwell Automation have an added market share of over 50 %. *ICS Processing Unit Example 1* and *ICS Processing Unit Example 2* cover contemporaneous PLC offered by Siemens, while *ICS Processing Unit Example 3* and *ICS Processing Unit Example 4* cover different solutions of Rockwell Automations product lines. The entries *ICS Processing Unit Example 5* and *ICS Processing Unit Example 6* represent smaller, more specialized vendors.

[17] https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500/cpus.html, 12/03/2020

[18] https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500/cpus.html, 12/03/2020

[19] https://ab.rockwellautomation.com/Programmable-Controllers/ControlLogix/5580-Controllers#overview, 12/03/2020

[20] https://ab.rockwellautomation.com/Programmable-Controllers/MicroLogix-1100, 12/03/2020

[21] https://www.fujielectric.com/products/plc/spf/specification/

[22] https://new.abb.com/products/1SAP130300R0271/pm573-ethac500-prog-logic-contr-512kb, 12/03/2020

The *ICS Processing Unit Examples* use a different terminology than used in classical Desktop IT systems. Here, the terms *Program memory* and *Data memory* are used. *Program memory* is the section of memory in which the executable user program is stored[23]. *Data Memory* contains information about the various inputs and outputs of the processing unit. These inputs and outputs are connected to the various sensors and actuators respectively. *Data memory* is always realized as *Volatile Memory*. The *ICS Processing Unit Examples* contain no example of a processing unit which does not implement *Program memory* as *Volatile Memory*. However, it would not be entirely technical impossible. This would be similar to having static program routines. The difference between programmable, re-programmable and hard-wired in embedded components is a major topic in ICS components and will be discussed in Section 4.1.2.6.

Usually, the executable program is loaded from some form of *Non-volatile Memory*. In the case of *ICS Processing Unit Example 6*, the specifications for from where the executable program is loaded from into the *Program memory* is as follows:

> *„Memory Type User Program: Flash EPROM, non-volatile RAM, SD Card“*

This is a common occurrence for ICS processing units. The *Non-volatile Memory* is either implemented in the form of an internal storage or through the use of memory cards which can be removed from the processing units. Removing these memory cards will usually terminate the execution of the user program.

Beside the potential split of the *Volatile Memory* into *Program memory* and *Data memory* and the potential use of removable media to implement mass storage *Non-volatile Memory*, some more conclusions can be deducted from Table 4.2.

Compared to the *Personal Computer Examples* presented in Section 4.1.1 the amount of available memory is orders of magnitude lower. *ICS Processing Unit Example 2*, which is a current top-line PLC offered by Siemens contains 6MB of *Program memory* and 20 MB of *Data memory*. This leads to a shortage of memory with little place for extensive log files. This affects the availability of forensic evidence since there is a smaller amount of potential data available.

Another factor is the comparatively low computation power of ECUs. Manufacturers often do not even give detailed information about the respective processing power as can be seen in Section 4.1.1. Here only *ICS Processing Unit Example 5* and *ICS Processing Unit Example 6* contain some information about the processing power at all. *ICS Processing Unit Example 5* uses a *16-bit Executing Processor* which stands in contrast to the 64-bit processors commonly used in Desktop IT. *ICS Processing Unit Example 6* gives some detailed information about the amount of instructions performable by the CPU. This lack of information on the processing power of the used CPUs is understandable when considering that the CPU only needs to perform a limited set of operations per cycle. These instructions cover the reading of inputs from the sensors, the processing of said inputs and the setting of outputs to control the actuators. Since the amount of inputs and outputs are limited and pre-defined, computing power is not really a major factor as long as the computing power is sufficient to perform

[23] https://automationprimer.com/2016/08/28/plc-memory/, 09/05/2020

these operations with a given cycle rate. However, this means, that excess computing power is usually rare.

The operating systems used within PLC are of widely varying complexity. Some are bare execution environments for a user program which controls the physical process. Other examples include far more complex operating systems. If additional data is available depends on the complexity of the operating system. This might be a rather complex system or a bare execution environment for the user program. *ICS Processing Unit Example 2* for example employs a GNU/Linux subsystem[24]. The use of GNU/Linux in SIMATIC PLCs was already shown in 2011 by [Ber11]. While other PLCs use less complex operating systems it can be assumed that a PLC will generally contain an operating system able to manage the user program and the communication with the periphery (potentially including different PLCs or an engineers workstation) at the same time.

The following sections discuss which data in ICS components are of potential use during a forensic investigation. Furthermore, *Data Gathering* and *Data Investigation* of data available in ICS components are discussed in general terms.

4.1.2.1 Availability of *Data Types* in *Non-volatile Memory* in Industrial Control Systems Components

In general, ICS components do not enjoy the vast amount of storage available in Desktop IT systems. This leads to a reduced amount of potential available forensic traces by excluding the creation of extensive log files and the storing of non-necessary data on the mass storage used in ICS components. However, some *Data Types* need to be present within the *Non-volatile Memory* in order for the component to be functional at all. This includes **DT1** since it contains basic information about the hardware configuration, including inputs and outputs. This data is absolutely necessary for the PLC in order to perform the assigned functions. Some of this configuration might be changeable by the system which implies **DT4**. Furthermore, **DT5** is usually available since it defines the communication behavior of the PLC with the attached sensors, actuators, other PLCs and potential supervisory systems. This data has a representation in **DT2**. Depending on the complexity of the format of the storage medium, a complex file system might also include **DT3**.

These theoretical considerations are confirmed in [Kei18]. This Bachelors thesis was supervised by the author of this thesis and investigated the forensic data traces available in the *Siemens SIMATIC S7 1516F-3 PN/DP*[25] PLC. This PLC is the predecessor of *ICS Processing Unit Example 2*. The *Non-volatile Memory* is realized by a *SIMATIC Memory Card*. These memory cards are removable Flash Drives. Currently, Siemens offers these memory cards with a size of up to 32 GB[26]. This is orders of magnitude less memory than usual in Desktop IT systems (see Table 4.1 for a listing of memory on contemporaneous Personal Computers).

A map of the *Data Types* contained within the memory of the *Siemens SIMATIC S7 1516F-3 PN/DP* PLC is provided in [Kei18]. This map makes no distinction between *Non-volatile Memory* and *Volatile Memory*. Table 4.3 shows an overview of the *Data Types* available within the *Non-volatile Memory* of the *Siemens SIMATIC S7 1516F-3 PN/DP* based on the

[24] https://cert-portal.siemens.com/productcert/pdf/ssb-439005.pdf, 09/05/2020

[25] https://mall.industry.siemens.com/mall/en/de/Catalog/Product/6ES7516-3FN02-0AB0, 09/05/2020

[26] https://support.industry.siemens.com/cs/products/6es7954-8lt03-0aa0/simatic-s7-memory-card-32-gb?pid=915083&mlfb=6ES7954-8LT03-0AA0&mfn=ps&lc=de-WW, 09/05/2020

Table 4.3: *Data Types* available in the *Non-volatile Memory* of a *Siemens SIMATIC S7 1516F-3 PN/DP* based on [Kei18]

Data Type	Comment
DT1	contains interrupts triggered by physical inputs contains information about device type contains information about device id contains license number
DT2	contains raw data
DT3	contains file access times
DT5	depends on used communication protocols example PLC used SNMP

work of [Kei18] after a distinction between *Non-volatile Memory* and *Volatile Memory* and a translation to English is performed by the author of this thesis.

In order to increase the validity of these observations, this listing of potential available *Data Types* in the *Non-volatile Memory* is actualized based on updated documentation provided in [SIE19]. An updated version of the contents of the SIMATIC memory card based on [SIE19] can be seen in Figure 4.2.

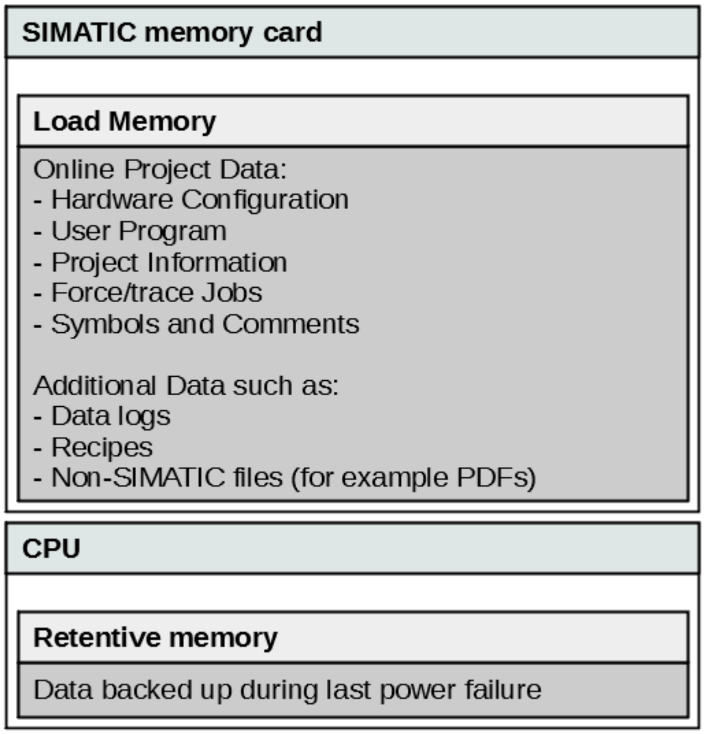


Figure 4.2: Contents of *Non-volatile Memory* in various Siemens SIMATIC PLCs based on [SIE19]

The overview provided in Figure 4.2 aligns with the considerations on the *Data Types* available within the *Non-volatile Memory* of PLCs. The memory card includes *Hardware Configuration* which aligns to **DT1** and **DT5** in this case, since this configuration also includes the configuration of the communication with attached components. Furthermore it contains the *User Program* which represents the functions the PLC should perform in the context of the physical process it controls. This executable does not easily fit within the *Data Types* as established in the [KDV15]-model (see Section 3.1.2.2 for a discussion on the various *Data Types*). The *user program* could be interpreted as **DT2** which would be sufficient to describe the physical representation of the program. However, this is not an optimal match to describe the *user program* on a higher abstraction level. This problem cannot be resolved in a satisfactory manner using the current *Data Types* and is discussed in more detail in Section 5.3.2. The memory card might also contain *Data logs* which represents logged data about the physical process. This is not to be confused with **DT6** which entails data about processes. However, in the case of **DT6** this refers to computing processes. In the case of the contents of the memory card these *Data logs* refer to the physical processes controlled by the PLC. The contents of these *Data logs* most likely fall into **DT8**. A deeper discussion on this topic is performed in Section 5.3.2, which also includes a redefinition of **DT6** in order to provide more clarity of the used terms. The memory card contains a file system which furthermore leads to a presence of **DT3**. This file systems also allows for the presence of *Non-SIMATIC files* on the memory card. These files would usually be manuals or accompanying documentation for the *user program*. These *Non-SIMATIC files* hence fall under **DT8**. Only the relative abundance of memory available in the case of these memory cards when compared to other storage technologies used in PLCs makes the inclusions of such data which is not entirely necessary for the performing of the core functionality of the PLC even possible. All these files have a raw physical representation which necessitates the inclusion of **DT2** to the list of potential available *Data Types* in the *Non-volatile Memory* of PLCs.

An additional section of *Non-volatile Memory* shown in Figure 4.2 is located within the CPU of the PLC. This section is referred to as *Retentive memory*. According to [Ehr15] this section is implemented as *NVRAM*. Some information about this memory can be found in [SIE17]:

> *„NVRAM stands for Non-volatile Memory with optional read and write access. This usually means a magneto-resistive RAM (MRAM) or a battery-buffered static RAM (SRAM). Just like conventional RAM, NVRAM can be used in read and write mode. In contrast, the data stored in an NVRAM is kept after a power failure. Due to costs, NVRAMs in SIMATIC IPCs only have a size between 128 kByte and 512 kByte. On the NVRAM, important process data of a SIMATIC IPC software control can be stored, e.g. the state of controller tags, diagnostic messages or parts of the control program.“*

This memory is used to store data about the physical process in the case of a power failure. This would enable the PLC to resume the physical process once the power is restored. In accordance with the previous discussion, this data about the physical process would fall under **DT8**. The NVRAM is part of the protection measures against power failures employed in this type of PLCs. It is notable how small the size of the NVRAM actually is.

A further exploration on the forensic data available in ICS components was performed in [**HAK$^+$17**]. Here, a Steam Turbine Governing System (STG) and its control components were used as an exemplary infrastructure in order to identify *Data Streams*. This publication comes to the conclusion, that the information about the physical process constitutes **DT8**. Depending on the configuration of the system, some parts of the information about the physical process could be stored in *Non-volatile Memory*. The fact that this can be configured leads to the conclusion that **DT4** is also present in *Non-volatile Memory*. [**HAK$^+$17**] argues that this information about the physical process could be aligned to various sessions and hence could, in another representation and context, be interpreted as **DT7**.

4.1.2.2 Gathering and Investigating the *Non-volatile Memory* in Industrial Control Systems Components

How difficult this potentially useful data is to extract from the PLC depends on the technology used to implement the *Non-volatile Memory*.

In the case of the *ICS Processing Unit Example 1* and *ICS Processing Unit Example 2* memory cards are used for the main portion of *Non-volatile Memory*. These memory cards can be removed from the PLC once the execution is stopped. This leads to a *Post-Mortem Forensics* scenario in which the data can be gathered with relative ease while maintaining integrity and authenticity of the gathered data. The main difficulty here might be the need for specialized software in order to access these memory cards in the first place. This could, for example, include the use of non-standard file systems.

Other PLCs (like *ICS Processing Unit Example 3*, *ICS Processing Unit Example 4* and *ICS Processing Unit Example 6*) use different technologies for the implementation of *Non-volatile Memory*. *ICS Processing Unit Example 4* uses *non-volatile battery backed RAM*[27] while *ICS Processing Unit Example 6* uses *Flash EPROM*. These forms of memory are internal and comparatively difficult to access. Especially since PLCs are designed to withstand adverse environmental conditions and are therefore of a more robust construction. Also, easy customization of the internals of a given PLC is not a design priority as it is in the Desktop IT domain (see Section 4.1.1). Here the physical access to the storage medium alone is difficult. Especially in the case of *non-volatile battery backed RAM* which might lose its contents on being disconnected from said battery. The use of such storage technologies also makes the use of more-standard interfaces less likely when compared to memory cards. Here, the access to the physical representation of the data is more difficult.

If the physical representation of the data (effectively providing **DT2**) is gathered in some manner, the interpretation of this data might require specialized software. The formats used in the ICS domain are usually proprietary and closed source. This might start with the employed memory structure which might not be a standard file system. Knowledge of the memory structure is necessary in order to identify the different files (or memory sections). This is a necessary foundation for a specific interpretation of the gathered data. This interpretation of the data might be, in the best case, conducted without the use of specialized software.

This is easier in the case of memory cards. Here, standard file systems might be used and the potential inclusions of accompanying media (**DT8**) makes the inclusion of standard file types less unlikely. As an example, the SIMATIC memory cards might include *PDF files* among the *Non-SIMATIC files* as seen in [SIE19]. Such a *PDF file* could be interpreted by standard software. In addition, storing *Data logs* with recorded data of the physical process in standard file formats is not entirely implausible.

[27] https://literature.rockwellautomation.com/idc/groups/literature/documents/pp/1763-pp001_-en-p.pdf, 09/05/2020

However, in many cases the use of specialized software provided by the vendors is necessary. This software is, in general, not tailored for the use within the forensic process and hence does not address any of the *Admissibility Factors* (see Section 2.1.7). Hence, the evidentiary value of evidence gathered and investigated by employing closed sourced is reduced due to a lack of traceability.

The same holds true when employing *Live Forensics* to gather and investigate contents of the *Non-volatile Memory* of a PLC. During *Live Forensics*, access is only possible by using the access provided by the operating system in conjunction with the file system (implicating the use of the *Methods* of **OS** and **FS**). This carries the same *Structural Impact* as when procedures of the same nature are used in the context of Desktop IT (see Section 4.1.1.2). Any request made to access the mass storage alters the forensic traces by influencing the operations performed by the operating system and potentially by changing the contents of the file system (file access time stamps). As discussed before, a PLC might employ a very simple or non-standard file system to manage access to the mass storage. Such a rudimentary file system might not include any file access time stamps at all. Also, PLCs employ operating systems of varying complexity. For the less complex access to specific sections of the mass storage by the use of the operating system might not be possible at all.

If however, a complex operating system is present, it would theoretically offer the possibility to use all the tools provided by the operating system during the forensic process. Even if such a complex operating system is present it is hidden from the user and not accessible during normal operations (as is the case for *ICS Processing Unit Example 1* and *ICS Processing Unit Example 2*). However, [Ber11] has shown how to gain access to the underlying operating system within a PLC and use *Methods* provided by the **OS** to gather various pieces of data. Although [Ber11] only discusses the gathering of *Volatile Memory* this could also be applied to *Non-volatile Memory*. However, performing such an action causes a massive *Structural Impact*. This goes as far that gaining the (from the manufacturers viewpoint) unwanted access to the underlying operating system could be considered a security incident in its own right. However, such access could be established during a **SP** and then used in the case that a **DG** of the *Non-volatile Memory* is deemed necessary during a forensic investigation.

The access might also rely on the use of specialized software designed for the communication of an external (engineering) workstation with the operating system of the given PLC using one of the various communication interfaces usually included within the PLC. This software is usually proprietary and closed source. This software might be able to either request the entire raw content of the *Non-volatile Memory* (**DT2**) or to query for certain sections of the memory contents. This is akin to *Runtime Interrogation* (as described in Section 4.1.1.4). The software used to perform the request is usually also able to interpret the response to any query. However, these requests cause a *Structural Impact* by altering the operation of the operating system and the potential contents of the file system. In addition, this software is not designed for the use during a forensic process. Hence, the *Admissibility Factors* are not addressed and the evidentiary value of evidence gathered and investigated by employing such software is reduced.

4.1.2.3 Availability of *Data Types* in *Volatile Memory* in Industrial Control Systems Components

Information about the *Data Types* present in the *Volatile Memory* of ICS components can be found in [SIE19]. A summary of these findings is presented in Figure 4.3.

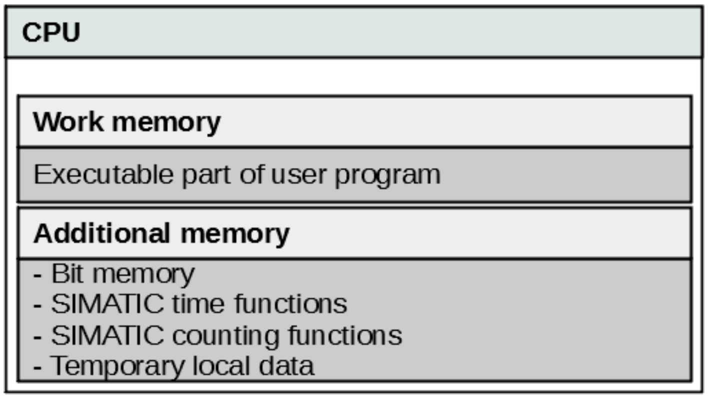


Figure 4.3: Contents of *Volatile Memory* in various Siemens SIMATIC PLCs based on [SIE19]

Based on [SIE19], an *executable part of user program* is identified within the working memory of the PLC. This program code represents **DT2**. In addition, some memory is assigned to store data about the physical process. Here, the same problems arise as with the *Data logs* in the *Non-volatile Memory* (see Section 4.1.2.3). Again, this data refers to the physical process and not to a computing process. Based on the current definition of the *Data Types* **DT8** seems like the best fit to represent this kind of data.

Although no further data is mentioned in [SIE19] it can be assumed that the *Volatile Memory* will contain some additional data in order to function efficiently. Information about the hardware configuration, including the attached sensors and actuators is required in order to perform the basic function (**DT1**). Depending on the exact nature of this configuration data it might also include **DT4** and **DT5**.

If additional data is available depends on the complexity of the operating system. As shown in Section 4.1.2, the operating system might be a rather complex environment or a bare execution environment for the user program. It can be assumed that a PLC will generally contain an operating system able to manage different (computing) processes at the same time which would necessitate the presence of **DT6**. More complex operating systems, like those employed in *ICS Processing Unit Example 1* and *ICS Processing Unit Example 2* employ complex services like SNMP (as [Kei18] discusses). This also implies the presence of **DT5**.

The exploration of the availability of forensic traces in *Volatile Memory* in Steam Turbine Governing Systems (STG) performed in [**HAK+17**] concurs with these findings.

4.1.2.4 Gathering and Investigating the *Volatile Memory* in Industrial Control Systems Components

Gathering and investigating the *Volatile Memory* in ICS components faces the same challenges as with Desktop IT (see Section 4.1.1.4). However, some of these challenges are amplified in the ICS domain.

In general, *Volatile Memory* is only available during *Live Forensics*. A loss of power will destroy any memory contents. Every method to access the contents of the main memory has to rely on operations provided by the operating system (and therefore uses the *Method Operating System*). Hence these requests to the operating system alter the system state and cause *Structural Impact*.

If the operating system is not in the direct control of the investigator, the integrity of the results of these memory requests is not ensured. This is amplified in an ICS environment, where most operating systems are proprietary and closed source. Here, the inner functions of the operating system are less likely to be known for the forensic investigator. Especially when considering that a greater market fragmentation and a widespread use of legacy components exist in the ICS domain than in the Desktop IT domain.

The varying complexity of the different ICS components not only has an impact on the availability of different forensic traces (see Section 4.1.2.3) but also on the gathering of these traces.

In general, every PLC offers some possibility to gain access to sections of the *Volatile Memory* by performing *Runtime Interrogation* (as described in Section 4.1.1.4). Usually this would include the use of some specialized management software installed on a Desktop Computer which queries the PLC for the requested information using some network interface. Examples for such software include the Siemens *Totally Integrated Automation Portal*[28] (in the case of *ICS Processing Unit Example 1* and *ICS Processing Unit Example 2*) or the Rockwell Automation Studio 5000 Logix Designer®[29] (in the case of *ICS Processing Unit Example 3*). These proprietary and closed source applications offered by the PLC vendors also perform the interpretation of this data. This interpretation of the gathered data would have to rely on the employed operating system and the vendor software not being manipulated and working in the way as claimed by the vendor. Hence, integrity and authenticity is affected.

There is also a limited selection of open source libraries able to perform requests to the respective PLCs akin to *Runtime Interrogation*. Two of these libraries are described in [**ALK$^+$18**]. *NodesS7*[30] uses *Node.js*[31] in order to query a SIMATIC PLC for information about the physical process. Variables concerning the physical process can be read and write from a PLC of the SIMATIC S7 1200 or 1500 Series (like those in *ICS Processing Unit Example 1* and *ICS Processing Unit Example 2*). Hence, this tool can be used to acquire **DT8**. In order for this method to work, the option *Enable GET/PUT Access* must have been set in the respective PLCs during a **SP**. *Snap7*[32] is an open source C++ suite with a similar use. This suite can be used to read and write various sections of data corresponding to the physical process. In addition, the suite can also be used to obtain information about the state of the PLC, including **DT1** and **DT4**. The requests to the PLCs performed by these two tools cause a *Structural Impact*. However, since these two tools are open source, a forensic investigator can obtain an increased knowledge on the inner workings of these tools. Also, an adaptation of these tools for forensic use is possible. In this case, the functionality to write the memory would have to be removed, every operation logged and the integrity and authenticity of the gathered data improved by using cryptographic means. Such required adaptations are described in more detail in Section D.4.

If a complex operating system is present an approach as discussed for the same conditions in Section 4.1.2.2 is possible. [Ber11] has shown that gaining access to the underlying operating

[28] https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html, 09/05/2020

[29] https://www.rockwellautomation.com/en_NA/products/factorytalk/overview.page?pagetitle=Studio-5000-Design-Software&docid=5ab2b75609a270c180a32208b634cac7, 09/05/2020

[30] https://github.com/plcpeople/nodeS7, 09/05/2020

[31] https://nodejs.org/en/, 09/05/2020

[32] http://snap7.sourceforge.net/, 09/05/2020

system inside a PLC and using this access to gather data from the *Volatile Memory* is possible. This could be used during a forensic scenario but carries the same implication as when used to gather *Non-volatile Memory*. Gaining this access alters the system state and hence causes a *Structural Impact*. Once this access is established, any action performed still causes *Structural Impact*. However, the overall *Structural Impact* could be reduced if such an access is established during **SP**.

If access to a complex operating system is possible methods described as *Software-based Acquisition* in Section 4.1.1.4 could be used to gather **DT2**. In the case of an underlying GNU/Linux, as in the examples used by [Ber11], even the same software as used under a Desktop Linux can be used for the acquisition of *Volatile Memory* as **DT2**. This chunk of raw data would then require interpretation during **DI**. However, this interpretation could be conduced independent of the employed operating system. Hence, some potential manipulation of the operating systems which could affect *Runtime Interrogation* would have a lesser impact since at least the interpretation could be performed with manipulation-free software on a forensic workstation.

Another possibility to gather data from ICS components was discovered during the research work performed in [Kei18], a Bachelors thesis supervised by the author of this thesis. This work focuses on the *Siemens SIMATIC S7 1516F-3 PN/DP*[33] PLC, a predecessor of *ICS Processing Unit Example 2*. It was discovered that this PLC maintains the variables which describe the current state of the user program during reprogramming. Hence, these variables could be gathered by a purpose built program which is loaded unto the PLC. These variables contain information about the underlying physical process controlled by the PLC which corresponds to **DT8**. In order for this approach to work, the PLC may not be put into *Stop Mode* by either manual action or actions of the programming interface. Should the *Stop Mode* be engaged, the variables are zeroed out and these memory contents are lost.

A good concluding summary on the use of *Volatile Memory* during forensic investigations in the ICS domain can be found in [**ALK+18**]:

> *„Access to main memory in general is only possible by sending requests to the respective PLCs. The accessible data is limited by the diagnostic functions of those PLCs. These diagnostic functions might be extensive in theory but are usually very limited or not available at all. This type of data gathering carries the same implications as in Desktop IT - sending these requests alters the state of the system under investigation (structural impact). Hence, it alters the communication on the field bus system transferring the requests to (and the answer from) the PLC and the specific PLC. While these implications seem grave, it might still be worth acquiring this data when the investigators take these implications into account during the discussion of the conclusiveness of the traces. Hence, the investigator should have an idea of what specific data should be requested in order to keep these implications low and predictable.“*

[33] https://mall.industry.siemens.com/mall/en/de/Catalog/Product/6ES7516-3FN02-0AB0, 09/05/2020

Table 4.4: *Data Types* available in the various *Data Streams* of Industrial Control System components

Data Type	*Non-volatile Memory*	*Volatile Memory*
DT1	✓	✓
DT2	✓	✓
DT3	✓	(✓)
DT4	✓	✓
DT5	✓	✓
DT6		(✓)
DT7		(✓)
DT8	✓	✓

4.1.2.5 Availability and Access to the Forensic *Data Streams* in Industrial Control System Components

This section provides an overview on the considerations on the use of the data from the *Non-volatile Memory* and *Volatile Memory* for forensic use in the ICS domain.

In general, ICS components do not enjoy the vast amount of storage available in Desktop IT systems. This leads to a reduced amount of potential available forensic traces. This excludes the creation of extensive log files and the storing of non-necessary data on the mass storage used in ICS components. Also ICS components vary widely in regards of the complexity of the used operating systems. Some employ only bare execution environments for the user program while others even employ GNU/Linux ([Ber11]). This reduces the generality of methods and statements.

In addition, ICS computing units are less complex than Desktop Computers. They are also used for more narrow tasks. However, as shown in Table 4.4 all *Data Types* are available in ICS components, albeit in a slightly different role. **DT1**, **DT3**, **DT4**, **DT5** and **DT6** are present in ICS components. **DT2** as the respective physical raw representation of the other *Data Types* is also present within ICS components. **DT8** is also present. Although the current definition of **DT8** presented in Section 3.1.2.2 does not directly describe the data about the physical process controlled by the specific PLCs **DT8** still seems the best fit to describe this data. Following this consideration, the information about the physical process could be aligned to various sessions and hence could, in another representation and context, be interpreted as **DT7**. This data about the physical process controlled by the PLC is without doubt useful for a forensic investigation which has to investigate the misbehavior of the physical process controlled by the PLC. Notable is the lack of any privacy related data which reduces privacy related constraints for the gathering of evidence. However, there still might be concerns due to the presence of information about the controlled physical process within the PLCs memory which could be considered a business secret (see **PC3** in Section 2.3.2.4).

To describe such misbehavior of the physical process controlled by the PLC a redefinition of the *Investigative Contexts* (see Section 2.1.2) is necessary since neither **IC1** nor **IC2** offers a perfect fit for such an investigation. This circumstance is discussed in Section 5.3.5.1.

The different nature of ICS leads to a notable difference in available *Data Types* and shows some problems with the current definition of these available *Data Types* in the [KDV15]-model. This problem is addressed in Section 5.3.2.

Evidence from the *Non-volatile Memory* can be gathered either using *Post-Mortem Forensics* or *Live Forensics*.

During *Live Forensics*, access to the *Non-volatile Memory* is managed by the operating system. These *Methods* provided by the **OS** have to rely on the fact that the operating system has not been tampered with. If this cannot be proven, integrity and authenticity of the gathered evidence suffers. In essence, various sections of the *Non-volatile Memory* are requested from the operating system. In general, the operating system will only provide data that is already interpreted. Depending on the complexity of the operating system, the accessible information might contain a subset or all of the contents of the *Non-volatile Memory*. In the example provided by [Ber11], the underlying GNU/Linux operating system could be used to employ methods known from the Desktop IT domain to achieve a complete copy of the entire *Non-volatile Memory* of the PLC (see Section 4.1.1.2 for the respective approaches used in the Desktop IT domain). This would provide the data in a raw, uninterpreted form (**DT2**).

Post-Mortem Forensics relies on physical access to the respective memory. ICS use different technologies to implement *Non-volatile Memory*. These include Memory cards and various internal storage technologies. In the case of memory cards, the medium can simple be removed. Since these memory cards also usually employed standard interfaces, their contents can be gathered with proven methods maintaining a high degree of integrity and authenticity. For other technologies, physical access is more difficult. In general, ICS components are of sturdy construction and not built for access to their inner workings. This complicates physical access. Some memory technologies (for example *non-volatile battery backed RAM*) rely on battery power to store the respective data. If the power source is disconnected during the physical access, this data is lost. Also, the logical access to these devices is not as standardized and might require specialized hardware and software. This would also provide the evidence in form of **DT2**.

If the data is obtained as **DT2** interpretation is necessary. This might require specialized software. Formats used in the ICS domain are usually proprietary and closed source. Hence the forensic investigator has little knowledge of the inner workings of this software. In addition, this software is usually not built with the requirements for a forensic process in mind (the *Admissibility Factors* as discussed in Section 2.1.7). This reduces the integrity and the authenticity of the forensic evidence to a lesser degree than in those cases in which it is necessary to entirely rely on the operating system of the PLC. Also, as usual during **DI** in *Post-Mortem Forensics*, the interpretation can be repeated and external methods to maintain the integrity of the **DT2** (like simply using specific copies of the data for the interpretation) can be used.

Access to the *Volatile Memory* is only possible during *Live Forensics* and has to rely on the operating system.

Usually, the gathering of evidence from the *Volatile Memory* of an ICS component will be performed using *Runtime Interrogation*. This returns already interpreted data. The fact that **DG** and **DI** have to rely on the operating system which might be tampered with and whose inner workings are usually hidden to the forensic investigator reduces integrity and authenticity of the gathered evidence.

If the PLC contains a complex operating system, access to **DT2** might also be possible. This is usually not intended by the manufacturer and security measures have to be circumvented as in the case shown by [Ber11]. It will reduce the *Structural Impact* of such an approach when the access to the underlying operating system is established during **SP**. If the data is obtained in the form of **DT2**, interpretation is necessary. This comes with the same benefits and problems as for **DT2** gathered from the *Non-volatile Memory*.

In general, ICS contain various *Data Types* which might be useful during a forensic investigation. In general, the wide variety of used operating systems, memory technologies and interfaces makes generalized assumptions more difficult than in the Desktop IT domain. Accessibility and knowledge about the inner workings of the respective systems and tools employed to access these systems also suffer from this diversity of these proprietary and closed source systems. This lack of transparency reduces the trust in the integrity and authenticity of the evidence, since most tools will only fulfill **AF0** (see the *Admissibility Factors* as discussed in Section 2.1.7). However, there exist some open source solutions for querying data from the PLC. These open source solutions could be adapted in order to provide forensic tools.

Physical access to *Non-volatile Memory* as employed during *Post-Mortem Forensics* is either simple (in case of memory cards) or complex (in case of *non-volatile battery backed RAM*). Logical access to *Non-volatile Memory* and *Volatile Memory* can be performed by using *Runtime Interrogation*. This type of access carries the usual *Structural Impact*.

4.1.2.6 The Varying Complexity of Industrial Control Systems Components

It is noteworthy that the sensors and actuators attached to PLCs actually cover components of varying complexity. As [**AHKD19**] discusses, some of these sensors and actuators might be simple hard-wired mechanisms without any control logic while others contain (re-)programmable logic. From a computer forensics point of view there is a difference whether a device is hard-wired or (re-)programmable. [**AHKD19**] states that a component which is hard-wired or does not contain any program logic offers less attack vectors since an alteration of the performed operations would require a physical alteration (or defect) of the component. If a forensic investigation (conducted with methods outside the scope of computer forensics but no less supporting a computer forensic investigation) could prove that no physical tampering has happened the logic of said component is not of primary interest for the computer forensic process. If such a component would misbehave, it would point to a systematic failure in the wiring of the component or at least a supply-chain attack. Beyond this statement, it is noteworthy that those components which are (re-)programmable contain some *Non-volatile Memory* which might be of relevance during the computer forensic process.

Hence, it is necessary to distinguish between those sensors and actuators which are hard-wired and those which are (re-)programmable. A suggestion based on the Purdue enterprise reference architecture ([Wil92]; see Section 2.2.2.1 for an introduction) is proposed by [**AHKD19**]. Here, *Level 0* is split between hard-wired and (re-) programmable components into *Level 0* and *Level 0.5*. This leads to the following redefinition:

- "**Level 0 - hard-wired Process** any components directly involved in the basic manufacturing process that does not contain any program logic.
- **Level 0.5 - programmable Process** components with program logic directly involved in the basic manufacturing process.„

4.1.3 Automotive IT Components

This section covers the basic components of Automotive IT. This establishes an understanding of Automotive IT and allows for an analysis of the specific components with regard to the computer forensic process. After exploring the availability and accessibility of forensic traces in *Non-Volatile Memory* and *Volatile Memory* in Automotive IT a summary is provided in Section 4.1.3.6.

As already discussed in Section 2.2.3.1, a vehicle consists of different components which form subsystems responsible for various tasks. On the most fundamental level a set of basic components can be identified. These basic components can be divided into passive and active components as shown in [**ALKD17**]. Passive (or dumb) components do not have any electronic functionality. Examples include the car body or the car seats. Active components do have an electronic functionality and contain programmable logic. Hence, they are of greater interest for this thesis.

Expanding on [**ALKD17**], these principal active components are:

- **Sensors** measure the conditions of the vehicles systems and its environment (e.g., pressure, speed, light levels, rain intensity etc.) as well as user input.
- **Actuators** are electrically operated and manipulate their environment in non-electric aspects (e.g., mechanics, temperature, pressure, etc.).
- **Electronic Control Units (ECUs)** electronically process input signals acquired via sensors and relay commands to actuators.
- **Direct analogue cable connections** connect sensors and actuators directly to a specific ECU.
- **Shared Digital Bus Systems** are used for communication among ECUs ([Hil12])

This listing shows notable similarities between the basic components of Automotive IT and ICS (see Section 4.1.2). Again, sensors measure environmental conditions and user input, processing units process (compute) these inputs and actuators act on the results of these computing processes. In Automotive IT, these processing units are referred to as Electronic Control Units (ECUs). Various communication media are used to tie the processing units and their attached sensors and actuators together (Direct analogue cable connections) or to facilitate communication between the various processing units (Shared Digital Bus Systems).

There is a trend towards the inclusion of more and more ECUs and a reliance on these ECUs to take over additional functionality within the vehicle. In general, the number of ECUs in cars has increased over the last years. An estimate in [SST14] gives the number of ECUs in luxury cars to less than 10 for the year 1985 with an increase to more than 100 in the 2010s.

Some ECUs designed for the use in contemporary cars are listed in Table 4.5 to provide an overview on the capabilities of the ECUs used within vehicles. The components shown in Table 4.5 are selected to be representative. Reviewing the technical data, some notable properties which impact the computer forensic process can be observed.

Table 4.5: Computing power, memory and interfaces of exemplary selected ECUs in Automotive IT

Name			
CPU	RAM	Memory	Interfaces
ECU Model Example 1: HY-TTC 580[34]			
32 bit TI TMS570 Dual-Core 180 Mhz	256 kB int. 2 MB ext.	3 MB int. flash 8 MB ext. flash	7 x CAN (1 Mbit/s) 1 x Ethernet (10 Mbit/s) 1 x RS-232 1 x Ethernet 1 x (10 Mbit/s)
ECU Model Example 2: Transtron Diesel Engine[35]			
32-bit RISC chip			CAN
ECU Model Example 3: Embitel BCM[36]			
MPC5674F [37] 2x32-bit CPU	256 KB on-chip	4 MB on-chip flash	
ECU Model Example 4: Continental GVCU[38]			
32-bit	96 kByte	1 MByte flash	3x CAN 1x LIN 16x Digital In 12x Analog In 19x Digital Out
ECU Model Example 5: Bosch Steering System ECU[39]			
32-bit		flash	CAN CAN FD FlexRay
ECU Model Example 6: Schaeffler Engineering PROtroniC TopLINE[40]			
Dual Processor NXP MPC8544 (@1 GHz) Co-Processor IBM PPC440 (@400 MHz)	256 MByte 32 MByte	64 MByte flash 32 MByte flash	4x CAN 1x FlexRay 2x LIN 2x Ethernet 24 analog inputs 6 analog outputs
ECU Model Example 7: NXP S12XS[41]			
16-bit	4, 6 or 16 KByte	64, 128 or 256 KByte flash	16 analog inputs CAN BDM
ECU Model Example 8: KEA128[42]			
up to 48MHz Cortex-M0+ core	up to 16 KB	up to 128 KB flash	up to 16 analog channels 2x SPI up to 3x UART 2x 12C 1x MSCAN

At first, the components employed here have very limited resources when compared to systems in Desktop IT (see Section 4.1.1).

Even *ECU Model Example 5* which is a high-powered development ECU for testing has very low processing power compared to systems in Desktop IT. In general, the processing power is quite low. Various examples use 16-bit (*ECU Model Example 7*) or 32-bit processors (almost any other *ECU Model Example*). In general, the clock rates of the given ECUs are orders of magnitude lower than in Desktop IT processors. While the clock rate alone is not a reliable indicator for processing performance, such a large mismatch indicates large differences in performance.

If the amount of memory is taken into consideration these limitations become even more prominent. Even the largest amount of memory is still given in low Megabyte ranges. Additionally, in those cases where information about the memory technology is provided usually flash memory is employed. The memory wear implicit to flash memory and the long life cycles of cars (which includes the internal Automotive IT) forces the developers of Automotive IT to rely on operations in which writing information to the *Non-volatile Memory* is a rare occurrence. This makes writing extensive log files impractical and has major implications for the computer forensic process. When compared to ICS components (see Section 4.1.2) the memory size is in similar orders of magnitude.

Another notable factor is that the specifications given are vague. This has two primary reasons. For once, some automotive manufacturers see advantage in protecting their systems by a security-by-obscurity approach. Hence, only very limited information on the components is made public. More important however is the fact that different parts might be used interchangeable. In the case of *ECU Model Example 2*, every time a new batch of this ECU is produced another set of processors might be used as long as they fulfill the given specifications and are cost-effective. Hence, having two ECUs of the same model and make does not guarantee that the same internal components might be used. Since different ECUs might offer different debug interfaces, this might have some impact on the computer forensic process.

Basically all ECUs include analog or digital in/output to directly attach sensors or actuators to these units. These represent the direct analogue cable connections established earlier in this section. LIN (Local Interconnect Network) is a common interface for ECUs and used for the same purpose. CAN (Controller Area Network - see [ISO15a]) is ubiquitous in the domain of Automotive IT. Beside LIN and CAN, FlexRay is another shared digital bus system used in

[34] https://www.aradex.de/en/products/ecu-electronic-control-unit/electronic-control-unit-hy-ttc-500/, 16/01/2020

[35] https://www.transtron.com/en/products/control/dieselengine.html, 16/01/2020

[36] https://www.embitel.com/body-control-module-ecu-in-automotive, 16/01/2020

[37] https://www.nxp.com/docs/en/data-sheet/MPC5674F.pdf, 16/01/2020

[38] https://www.continental-automotive.com/getattachment/4619d545-2d3a-43f4-a0ab-8a50e09130c2/PI_Overview_Conti-gVCU_Conti_CD_08-2015.pdf.pdf, 16/01/2020

[39] https://www.bosch-mobility-solutions.com/en/products%2Dand%2Dservices/passenger%2Dcars%2Dand%2Dlight%2Dcommercial%2Dvehicles/steering%2Dsystems/electric%2Dpower%2Dsteering%2Dsystems/electronic%2Dcontrol%2Dunit/, 16/01/2020

[40] https://www.schaeffler-engineering.com/remotemedien/media/_shared_media_rwd/03_worldwide_1/websites_worldwide/germany_3/schaeffler_engineering/documents_2/products_documents/Schaeffler_Engineering_PROtroniC_TopLINE_UCU_TB_E.pdf, 16/01/2020

[41] https://www.nxp.com/docs/en/data-sheet/MC9S12XS256RMV1.pdf, 16/01/2020

[42] https://www.nxp.com/docs/en/data-sheet/S9KEA128P80M48SF0.pdf, 16/01/2020

Automotive IT, as seen with *ECU Model Example 5* and *ECU Model Example 6*. Some high-end examples geared towards the development of new vehicles (*ECU Model Example 4* and *ECU Model Example 5*) include Ethernet interfaces to follow current trends in Automotive IT. These technologies and their impact on the forensic process will be discussed in Section 4.2.3.

Additionally, all the components presented here are built to last. Several of the system descriptions stress the ECUs resistance to environmental effects. As an example, the description of *ECU Model Example 2* states:

> *„[..]enhanced heat- and vibration-resistant as well as fully waterproof design[..]"*

The specification of *ECU Model Example 7* gives specific information about the temperatures it is designed to withstand:

> *„Ambient temperature range -40 °C to 125 °C"*

Hence, the ECUs are sturdily built. Components are configured directly at the vendor and everything is put together in a heat-resistant, shock-proof and water-proof casing. Opening up this casing is significantly harder than in most computer systems in classical IT environments. Putting everything back together inside the case without diminishing the robustness to environmental effects is even harder and requires specialist knowledge and tools. In short, the hurdle to open up the employed components in order to perform a forensic investigation is higher.

These properties already implicate an impact on the forensic process. This is due to the availability of certain traces and the accessibility of these traces. The following sections look into more detail on which traces are available in the *Data Streams* of *Non-volatile Memory* and *Volatile Memory* in Automotive IT and how these traces can be accessed. The *Communication Data Stream* which takes place between different components in Automotive IT is discussed in Section 4.2.3.

Like PLCs in the ICS domain, ECUs in Automotive domain do not, in general, employ complex operating systems. Usually the operating system only manages the access to the attached hardware and executes the user program. Notable exceptions are the infotainment systems included in modern cars as discussed in Section 2.3.4.3. These In-Vehicle-Infotainment systems (IVI) employ complex operating systems. As discussed in Section 2.3.4.3, these IVIs are more akin to Desktop Computers plugged directly into Automotive IT and interacting with the Automotive IT. Hence, they are of no primary concern for the following sections, since their use for the forensic process is covered in Section 4.1.1 and Section 2.3.4.3 respectively.

4.1.3.1 Relationship Between *Live Forensics* and *Post-Mortem Forensics* in Automotive IT

A peculiarity of Automotive IT deserves special attention since it has a great influence on the forensic process. This peculiarity is caused by the way components in Automotive IT are attached to the power supply. These attachments are performed using so called terminals[43].

[43] http://www.bosch-classic.com/media/en/bosch_classic/teile_1/switches/downloads_3/klemmenbezeichnungen.pdf, 11/05/2020

Automotive components are usually powered by the vehicle battery. Components that are connected directly via battery positive (terminal 30) therefore only lose power supply in exceptional cases (for example, in the event of an accident, if there is an emergency disconnect of the battery, or an empty battery). Accordingly, these ECUs are normally never 'switched off'. Other ECUs are connected to the voltage supply via ignition plus (terminal 15). This also leads to the fact that some information which would usually be stored in *Non-volatile Memory* in systems from different domains is stored in the *Volatile Memory* in Automotive IT.

This circumstance shifts the lines between *Live Forensics* and *Post-Mortem Forensics* in Automotive IT. On one hand, the voltage is generally held by the battery. The *Volatile Memory* is retained accordingly and only lost in specific circumstances. It cannot be assumed that every incident in Automotive IT environments ends in an accident which would cause such a circumstance by triggering an emergency disconnect. Hence, a significant amount of forensic investigations could take advantage of this and be performed in a *Live Forensics* setting. However, with the respective components still receiving power after the engine is switched off the informative value of the *Volatile Memory* decreases over time.

If a component is actually disconnected from the power supply, the *Volatile Memory* is lost. In this case, only procedures of *Post-Mortem Forensics* remain. This is often the case in crash reconstruction (see Section 2.3.4.1). Here, the battery should disconnect due to safety reasons.

How Long Does *Live Forensics* Stay *Live Forensics*?

The possibility of the investigated system being supplied with power all the time during a very long timespan before an investigation starts and the **DG** is performed leads to the question on the when performing an investigation into a still active system can still be considered *Live Forensics* at all. This question has already emerged in Section 3.1.2.6. During the discussion of the *Untersuchungsvorraussetzung* (German for *„requirements for the (promising) usage of a forensic tool"*) which is part of the *Classification Scheme for forensic tools and methods*, different degrees of *Live Forensics* were identified. A difference between those cases in which a computer system is still active and connected to a given network and those cases in which a computer system is still active but disconnected was identified. These two different aspects can be described as *Live Forensics* and *Live Forensics with disconnected network*. While this construct might prove helpful in certain cases, it does not cover the question for how long *Live Forensics* can be considered *Live Forensics* at all.

In order to answer this question it is necessary to understand the purposes of *Live Forensics*. As discusses in Section 2.3.2.3, *Live Forensics* aims at gathering the contents of *Volatile Memory* which would be lost and impossible to obtain if the system is powered off. As shown at various points of this thesis, the system state is also altered by ongoing normal system operations. This *Structural Impact* reduces the value of the obtainable data. This reduction is greater when the alteration to the system state is growing bigger. At one point, the data might be considered worthless for the investigation at all if one has to assume that the data has shifted too much from the state during the incident which is investigated.

If it is possible to understand which data has changed between the incident and **DG** this data could be considered as *Loss* and simply disregarded (see Section 2.1.5 for a discussion on

Error, Uncertainty and Loss in Digital Forensics). Data altered by the operation without the forensic investigator being able to determine this alteration is more difficult for the forensic process. This data would be considered as an *Error* since it is simply incorrect. The biggest problem however is the *Uncertainty*. Here it is impossible for the investigator to know if the data obtained has changed since the investigated incident. This disrupts the trust in the data and reduces the evidentiary value.

It is hard to establish a general rule on when the data losses its evidentiary value. The elapsed time between the investigated incident and the **DG** is an obvious factor. However, the regularity and frequency with which the data being viewed are overwritten is significantly more important. For example, a section of *Volatile Memory* which contains a counter which stores the amount of instances in which a temperature sensor provided values outside the operating parameters will be updated with a lower regularity and frequency than a section which contains a variable for the current speed of the vehicle. The speed of the vehicle can be considered as an extreme example of losing its usefulness for an investigation if not gathered when the investigated incident and the **DG** happen at the same time.

From this point of view, it can be argued that *Live Forensics* stays *Live Forensics* as long as some contents from the *Volatile Memory* are still deemed to have enough evidentiary value for the forensic process. Such a consideration requires in-depth knowledge into when and how often specific data fields within the *Volatile Memory* are updated is essential to reduce *Uncertainty* and to prevent *Error* during **DG** and to maintain some evidentiary value.

In the Loop Forensics

It is also possible to reactivate the vehicle (or individual components thereof - especially ECUs) for **DG**. In the context of this thesis, this special form of *Live Forensics* is referred to as *in the loop Forensics*. The terminology is motivated by the so-called *hardware in the loop* tests, in which individual components are tested in a simulated system environment. [**ALKD17**]) discussed the concept of using *in the loop Forensics* in Automotive environments and elaborated on the challenges faced when using this approach. The publication notes that the process of disconnecting the ECU and then switching it on again obviously alters the state of the ECU under investigation. The environment of the test itself might also have some *Structural Impact* on the gathered forensic data. The self-diagnose routines implemented in most of the ECUs will cause additional errors if the outside behavior (sensors, actuators and communication interfaces) is not simulated in a befitting manner. These errors might be stored as DTCs (Diagnostic Trouble Codes) causing a *Structural Impact* on *Non-volatile Memory* and *Volatile Memory*.

This approach is also used by [RA02] in order to analyze ECUs extracted from crash vehicles. Here these components are recovered and then powered up again to be scrutinized using diagnostic requests. Some suggestions on how to build such a *hardware in the loop* can be found in [Smi16] where this concept is referred to as *ECU Test Benches*. This book gives some basic overview on how sensor inputs can be simulated and the setup used to gain vendor-specific insight into the ECU and the retained data.

This concept is extended on by [**KAH^{+}18**]. This work also discusses different approaches on how the sensor input can be simulated in order to prevent the presence of additional DTCs only caused by the investigation itself. [**KAH^{+}18**] presents three different strategies for performing such a simulation:

- *Physical simulation of the input signal*
  This involves creating the physical conditions detected by the sensor. Examples would be using heaters for temperature sensors. This is obviously impractical for some sensors present in modern cars, but does not require any specialized knowledge about the inner workings of the sensor and hence can be done independently of vendor-specific knowledge.

- *Electrical simulation of the input signal*
  This replaces an analogue sensor with an electrical signal. This requires a deep understanding of the specific value range of the sensor. Analogue sensors include (according to [**KAH$^+$18**]) temperature or pressure dependent resistors, magnetic sensors or proximity sensors.

- *Electronic simulation of the input signal*
  This involves sending the sensor reading directly via digital communication bus to the specific ECUs. Hence, it requires knowledge about the specific protocols, the value ranges the timing of transmissions and the architecture of the Automotive IT. Section 4.2.3 provides a discussion on the architecture of Automotive IT and the impact of sensor readings being transmitted via digital communication bus systems for forensic investigations.

This short summary shows that the ability to perform *in the loop Forensics* is greatly impacted by the presence of specific knowledge about the sensors. This includes value ranges, specific encodings and communication timings.

Of course, a tiered approach can be used. Some tests with a physical simulation of the input can be performed in order to obtain this specific knowledge. This can then be used to simplify further simulation by using the simpler (implementation-wise) approach of electronic simulation of the input signal.

4.1.3.2 Availability of *Data Types* in *Non-volatile Memory* in Automotive IT

Automotive IT does not contain great amounts of memory. This applies for *Non-volatile Memory* as well as for *Volatile Memory*. This leads to a reduced amount and variety of traces with forensic relevance.

Since *Non-volatile Memory* in ECUs is mostly implemented as flash memory and has to survive the long live cycles usual with cars, ECU tend to limit writing operations to *Non-volatile Memory*. This excludes data which changes often.

Some *Data Types* need to be present in ECUs in order to perform the basic functions of the component. This includes **DT1** since it contains basic information about the hardware configuration, including inputs and outputs. This data is a basic requirement for the ECU in order to function. **DT5** is necessary as it controls the ECUs communication with other components including other ECUs. Furthermore some configurations might be alterable which implies the presence of **DT4**. All this data has a physical representation within the *Non-volatile Memory* which is covered by **DT2**. The *Non-volatile Memory* also contains the user program, which might be of interest during a forensic investigation if a potential manipulation of said user program has to be investigated. This user program implements the functionality the ECU has for the driving process. In its physical representation this user program would also fall under **DT2**. However, this is not an optimal solution and hence is discussed in more detail in Section 5.3.2.

ECUs usually contains DTCs (Diagnostic Trouble Codes). These codes are set when certain conditions are detected. This could include information about non-responsive attached components, unexpected sensor readings (like e.g. conflicting readings from different sensors, or sensor readings outside of the operating parameters) or fault states. These DTCs offer some major use for forensic investigations. Information about the physical defect of a sensor is **DT1** while information about sensor readings outside of the operating parameters could be considered as **DT8**.

4.1.3.3 Gathering and Investigating the *Non-volatile Memory* in Automotive IT

Forensic data can be extracted from the *Non-volatile Memory* in Automotive IT during the course of *Post-Mortem Forensics* by gaining physical access to the storage medium.

Physical access to the *Non-volatile Memory* in Automotive IT is a difficult undertaking. In general, the ECUs used in Automotive IT are sturdily built to survive automotive environments. Opening up the casing is difficult even, especially when the continued usage of the ECU after the disassembly is aimed for. Intellectual property and copyright protection measures might complicate the physical access to the respective physical memory further. Accessing the memory requires dedicated instruments, especially in comparison to the easily customizable and accessible devices of the Desktop IT domain (see Section 4.1.1). Parts of the mass storage often are part of the MCU silicone itself which further complicates access (see [**ALKD17**]).

Research on the practicality of physically accessing *Non-volatile Memory* in Automotive IT was performed by [RA02]. The research work covers the recovery of forensic data from an EEPROM/flash memory taken from crash vehicles. This example includes the **DG** of the forensic data and a hypothetical **DI** based on this data. This interpretation is done in the context of a crash reconstruction (see Section 2.3.4.1). However, [RA02] shows the possibilities and challenges of using physical access to investigative *Non-volatile Memory* in Automotive IT.

Access is easier achievable by employing *Live Forensics* or *in the loop Forensics*.

In general, ECUs offer various diagnostic interfaces, protocols and tools usable during **DG**. While the specific diagnostic protocols are vendor-specific, this section discusses two notable ones and their use during the forensic process. Some tools interacting with these diagnostic protocols are also presented.

In general, the diagnostic protocols are dependent on the underlying transport protocols. The diagnostic protocols used by European automobile manufacturers communicate using the CAN bus ([ISO15a] - Section 4.2.3 discusses the CAN bus in detail). The most common of these protocols is the *on-board diagnostic protocol* (OBD - [ISO15b]) which is legally required for exhaust gas diagnosis. Other common protocols are the *keyword 2000 protocol* (KWP 2000 - [ISO99]) and *Unified Diagnostic Services* (UDS - [ISO13a]). These protocols rely on the standardized *ISO TP* ([ISO11]) protocol as a transport protocol.

Before looking at the *OBD-II* diagnostic protocol in detail, it should be noted that the use of any diagnostic protocol causes a *Structural Impact* by altering the operation of the ECU and potentially the contents of the *Non-volatile Memory.* The *Volatile Memory* is invariably altered. Since these requests also use the CAN bus, which is the usual communication medium employed between ECUs, the *Communication* is also altered. Furthermore, only the areas of the *Non-volatile Memory* that are accessible via these diagnostic interfaces can be obtained in this manner.

OBD-II

This paragraph gives some general information on OBD before the following paragraphs introduce some examples for tools which can be used to utilize the diagnostic functions provided by OBD for **DG** of *Non-volatile Memory* in Automotive IT.

OBD was originally intended as a protocol for accessing emission-relevant monitoring functions and has been mandatory for all newly registered vehicles on the US market since the early 1990s according to [CAR15]. Over the years OBD has been standardized to OBD-II and numerous manufacturers have introduced additional diagnostic functions in their respective implementations. In general, OBD is used in the majority of the current cars and offers the ability to access various diagnostic functions. Widely used is the possibility of reading out error codes (DTC - Diagnostic Trouble Codes) using OBD-II. There is a wide collection of tools that, based on the OBD-II interface, use various diagnostic functions and can hence be used during **DG**. The OBD-II interface is usually connected directly to the gateway, which then forwards the diagnostic requests to the corresponding bus systems and also forwards the answers to the diagnostic requests to the respective diagnostic device.

Various signal protocols are used to implement OBD. These form the foundations for the communication of higher protocols. Examples here are SAE J1850 ([SAE06]) and ISO 9141-2 ([ISO94]).

An exemplary selection of tools able to perform diagnostic requests to ECUs using OBD-II is introduced here:

- O2OO Data Logger[44]
  The *O2OO Data Logger* is a tool to gather diagnostic data using OBD2. A definition on which data should be collected cyclically is required. This set of data is then saved into a database.

  This tool is not designed for the use in a forensic investigation and hence does not include any mechanisms to ensure integrity or authenticity of the gathered data. However, since this tool is open source it could be adapted for the use in forensic scenarios by adding some of the features discussed in Section D.4.

  For communication with the vehicle, this tool relies on the use of a *ELM237 OBD to RS232 Interpreter*[45].

- FREEDIAG[46][47]
  FREEDIAG is a complete open source diagnostic environment with a wide range of manufacturer-specific diagnostic protocol implementations. According to the developers, these protocols include:

[44] https://www.vanheusden.com/O2OO/, 11/05/2020

[45] https://www.elmelectronics.com/wp-content/uploads/2017/01/ELM327DS.pdf, 11/05/2020

[46] http://freediag.sourceforge.net/, 11/05/2020

[47] https://github.com/fenugrec/freediag/, 11/05/2020

  - ISO 9141
  - ISO 9141-2
  - ISO 14230
  - SAE J1850 (PWM and VPW)
  - SAE J2818 / KWP1281 / VAG (over K line)
  - Mercedes Gearbox Protocol

  The software contains extensive functionality for querying all OBD2-specific parameters. The software is also able to periodically query values and read (or delete) DTCs. The option to use this tool in batch mode is helpful for use in a forensic scenario. This is due to the fact that the operations can be performed in a command line which makes the *Process Accompanying Documentation* by logging the inputs and outputs in the console possible. This is easier to achieve than during the use of a complex GUI.

  However, during laboratory tests, the software sometimes displayed a curious communication behavior. Sometimes diagnostic requests were delayed and not immediately send. This lack of traceability of the operation reduces confidence in this tool within forensic scenarios. Also, the tool contains no measures to ensure integrity or authenticity of the gathered data. This would require external measures or an adaptation to forensic use which is possible due to this software being open source (see Section D.4).

- Caring Caribou[48]
  Caring Caribou is an open source tool for testing the security of Automotive IT environments. This tool is written in Python and has the ability to read DTCs. This read-process can be coupled with a brute force search for components if the specific logical addresses of the respective ECUs are not known. This tool is also able to reset and hence delete DTCs. Using this tool in a forensic scenario would benefit from a modification which removes this feature and instead adds features which can be used to protect the integrity and the authenticity of the results of the diagnostic queries.

- VCDS[49]
  VCDS is a proprietary and commercial software used for diagnostics in automotive systems employing Automotive IT belonging to the Volkswagen AG (which includes Audi, Seat and Škoda). In order to perform diagnostic requests, the software uses the generic OBD-II functionality as well as vendor-specific functionality (which is tailored towards the use with Volkswagen AG vehicles). The more generic functionality allows for the reading and writing of DTCs. The more vendor-specific functionality includes the reading of specific sensor inputs, the triggering of specific functions as well as the configuration of the specific ECUs. This software is also able to read the configuration of the specific ECUs.

All the tools in this exemplary selection share some similarities. At first, all the tools offer access to the diagnostic functions of ECUs available via OBD2 connection. This requires physical access to the OBD2 socket. This includes the presence of a specific connector (with several commercial solutions available with similar characteristics). The ODB2 socket itself

[48] https://github.com/CaringCaribou/caringcaribou, 12/05/2020
[49] https://www.ross-tech.com/vcds/download/current.php, 12/05/2020

is accessible with relative ease in any modern car. Usually it is located in front of the driver slightly left below the steering wheel. This complicates access by passengers during normal operations.

The amount of data accessible depends on the vendor and the specific software in use. While general diagnostic functions are common place some vendors implement far more detailed diagnostic functions. Accessing this vendor-specific diagnostic functions require vendor-specific knowledge like identifiers of the specific ECUs or their internal data fields or the necessary understanding to interpret this data. Vehicle vendors usually do not publish this information, so the diagnostic solutions not supported by a specific vendor have to rely on deduction or reverse engineering in order to gain this vendor-specific knowledge before using it.

Obtaining this vendor-specific knowledge is a complex task. [Smi16] is partly concerned with deducing such vendor-specific knowledge while [**KAH$^+$18**] discusses the use of a testbed for *in the loop Forensics* (see Section 4.1.3.1) in order to obtain such vendor-specific knowledge for the use of an independent forensic investigator. The inclusion of three open source tools created without manufacturer support shows that there is some notable research into understanding the inner workings of modern vehicles and sharing this information.

Furthermore, using these diagnostic functions carries a notable *Structural Impact*. All the tools presented here perform diagnostic requests to the respective ECU and thus alter the state of the ECUs accordingly. This involves at least the *Volatile Memory*. The *Communication* is also affected since these requests use the normal digital communication bus system employed within Automotive IT. This also leads to the problems present when performing **DG** on the *Communication* in Automotive IT (see Section 4.2.3.4 for a discussion on the problems faced during forensic investigations into the *Communication* in Automotive IT).

In addition, the diagnostic software can only request information included by the diagnostic functions of the specific ECUs and has to rely on the reply to the diagnostic requests. If the ECU is manipulated to give an incorrect response to the diagnostic queries, this fact might be undetectable by the means of these diagnostic requests. This has a negative impact on the traceability of forensic evidence obtained in this manner.

This fact is reinforced by the circumstance that none of the tools presented here are designed for use in a forensic process. Most tools have not only the ability to read DTCs but also to delete these potential forensic traces. Depending on the vendor-specific functions accessible using the ODB2 connection, even an alteration of the configuration of the ECUs is possible. In addition, these tools to not produce a protocol of the actions performed or protect the integrity and the authenticity of the evidence gathered using these diagnostic requests in any manner. However, Automotive IT is an interesting field for researchers and enthusiasts which lead to a notable number of open source tools which can be adapted for the use in the forensic process. Such an adaptation would include ensuring that no write operations can be performed, that the state of the ECU is altered at least as possible (in short: to reduce the *Structural Impact* as much as possible), that a protocol of the performed operations exist and that the integrity and authenticity of the preserved evidence is protected by cryptographic means. How to create (or adapt) a tool for it to be of most use during a forensic investigation is discussed in Section D.4.

Calibration Protocols

A number of calibration protocols for the use in Automotive IT exist. These protocols are intended for remote adjustment of ECUs. These calibration protocols commonly provide read and write access to the variables and memory areas of the processors (*Volatile Memory*) and their flash memories (*Non-volatile Memory*) at runtime during *Live Forensics.* These protocols utilize very precise time stamps, since these timestamps are measured and transmitted by the target system itself.

Currently *XCP* (eXtendedCalibrationProtocol)[50] is a common calibration protocol. This calibration protocol can use different bus systems (CAN, CAN FD, SxI, etc.) for transmission. Its predecessor, CCP [51] was only able to use CAN for transmission. The application areas of XCP are listed as the following by the authors of the standard ([ASA17]):

- Calibration of ECU parameters
- Measurement of ECU variables
- Stimulation of ECU variables
- ECU programming

The system is designed in such a way that the behavior of the ECU is only altered if the user performs an operation which explicitly alters parameters or variables. A purely passive readout of the ECUs data is thus possible. However, while the aim of calibration protocols is to avoid any interference with the usual operation of the ECU in order to enable the fine-tuning abilities these protocols are designed for a *Structural Impact* on the state of the ECU cannot be excluded in principle. The resulting data traffic has a *Structural Impact* on the *Communication* within the Automotive IT. The use of calibration protocols causes a notable load on the communication bus.

In general, the use of calibration protocols seems reasonable for certain forensic investigations. If, for example, an individual control unit is assessed as particularly interesting for a forensic investigation during **OP**, calibration protocols can be used, for example, to collect the *Volatile Memory* of this control unit or to check its functionality. The resulting load on the communication bus and the resulting *Structural Impact* on the *Communication* make the wide use of calibration protocols less attractive.

Another difficulty is the availability of tools utilizing calibration protocols and their specific implementations within the ECUs of various vendors. In general, calibration protocols have in common that they are protected against unauthorized use with the help of cryptographic procedures. Along with the reliance on vendor-specific information this makes the use of calibration protocols for the independent forensic investigator highly problematic. One freely available open source tool exists for accessing XCP:

- Caring Caribou[52]
  Beside the use already presented during usual diagnostic sessions, *Caring Caribou* implements XCP and can be used to accessing XCP-compatible ECUs. This open source

[50] https://www.asam.net/standards/detail/mcd-1-xcp/, 12/05/2020
[51] https://www.asam.net/standards/detail/mcd-1-ccp, 12/05/2020
[52] https://github.com/CaringCaribou/caringcaribou, 12/05/2020

> tool can employ a brute force search in order to discover XCP-compatible ECUs. This can then be used to retrieve information about the specific ECUs using XCP, potentially including the entire contents of *Non-volatile Memory* and *Volatile Memory*.

While these possibilities seem impressive indeed it is important to note that they rely on vendor-specific information which is generally not available to the independent forensic investigator. Again, this tool is not build for the use within a forensic process. An adaptation would be possible due to the open source nature of this tool. The features required for such an adaptation are discussed in Section D.4.

4.1.3.4 Availability of *Data Types* in *Volatile Memory* in Automotive IT

The *Volatile Memory* of ECUs in Automotive ITs contains all the values which have to change quite regularly and those required for the basic operation of the ECU.

This includes **DT1** since it contains basic information about the hardware configuration, including inputs and outputs. This data is a basic requirement for the ECU in order to function. **DT5** is necessary as it controls the ECUs communication with other components including other ECUs. Furthermore some configurations might be alterable which implies the presence of **DT4**. All this data has a physical representation within the *Volatile Memory* which is covered by **DT2**. The *Volatile Memory* also contains data required to perform the basic task of the ECU. This includes the input from the various sensors (or other ECUs) about the state of the vehicle. This could be understood as **DT8**. However, this seems unsatisfactory and is discussed in more detail in Section 5.3.2.

4.1.3.5 Gathering and Investigating the *Volatile Memory* in Automotive IT

The relationship between *Live Forensics* and *Post-Mortem Forensics* is unusual in the scope of Automotive IT. The same can be said for the relationship between *Non-volatile Memory* and *Volatile Memory*.

For once, the *Volatile Memory* is maintained over a long period of time since the ECUs are only disconnected from power supply in unusual circumstances. As discussed in Section 4.1.3.1 this does not mean that every usable date within *Volatile Memory* is available for an extended amount of time. Instead, the data falls victim to *Structural Impact* and *Evidence Dynamics* and as such becomes unavailable or at least less reliable over time.

The other factor is that physical access to *Non-volatile Memory* is much more complicated than in the Desktop IT and ICS domains. The usual approach to access *Non-volatile Memory* in Automotive IT is presented as using diagnostic requests to the specific ECUs (see Section 4.1.3.3).

This approach is shared when accessing *Volatile Memory* in Automotive IT. This incurs the same possibilities and constraints as when using this approach in the scope of *Non-volatile Memory*. In essence, it is irrelevant for the specific diagnostic tool if the content that is queried is provided by *Non-volatile Memory* and *Volatile Memory*. However, it is still relevant for the forensic investigator to understand if a requested date is stored in *Non-volatile Memory* or *Volatile Memory*. While in both cases the date could be altered, the date would definitely be lost if it is stored in *Volatile Memory* and one of the cases in which the power supply to the ECU is lost occurs.

Debug Protocols

One additional approach usable to obtain data from the *Volatile Memory* in Automotive IT is the use of debug protocols. An example for such a debug protocol is the Background Debug Mode (BDM - see [Fre06]). BDM is implemented in various chips of the now defunct manufacturer Freescale Semiconductor. The use of BDM is continued after Freescale Semiconductor was merged with NXP[53].

The BDM relies on the presences of an additional, separate microprocessor (MCU) within the ECU. The only task of this MCU is to ensure interference-free access to the *Volatile Memory* of the ECU. Diagnostic requests via DBM are then transmitted directly to this separate MCU. This separate MCU then answers the inquiries with the help of access to the respective memory locations. This access is independent of the executing MCU. This has the advantage that a read access to a specific memory area does not cause any *Structural Impact* on the operation of the executing MCU.

A difference to the usual calibration protocols is the type of connection to a BDM interface. While XCP can be used via the central communication bus, a separate interface (a so-called BDM Pod) must be connected directly to the chip for BDM. This implies a physical access to the respective ECU, and indeed, often to the physical interior of the ECU. This physical access is, in general, difficult in Automotive IT. The separate connection used for the communication between the BDM-MCU and the BDM-pod however circumvents the need to use the digital bus system within the Automotive IT for communication. In contrast to the Calibration Protocols described in Section 4.1.3.3 this additional communication connection prevents a *Structural Impact* on the *Communication*.

Similar implementations exist for other microcontroller architectures, for example JTAG (Joint Action Test Group). JTAG is standardized in [IEE13]. It also uses a debug interface situated directly in the specific ECU. Hence, it also requires physical access in order to be used but does not alter the *Communication* on the shared digital bus system employed within Automotive IT.

Another set of debug protocols use the shared digital bus system for communication to the respective ECU. One example is DXCPL (DAP over CAN Physical Layer - [DXC14]). This protocol is used with ECUs employing chips manufactured by Infineon[54]. Due to the use of the shared digital bus system this protocol has a notable *Structural Impact* on the *Communication* on this medium.

All these protocols are designed aiming for a low *Structural Impact* on the state of the executing ECU. This is essential for performing debug operations and helps the forensic process. Especially in the case of a physical separation between the executing MCU and the MCU which responds to diagnostic queries, this *Structural Impact* is low. If it can be ensured that these MCUs are physically separated and no write operation caused by the diagnostic occurs on the storage occurs, the *Structural Impact* on the MCU is non-existent. In general, debugging software also creates log files which can be used to document the actions performed. However, the software used for such access is not geared towards the use in forensic scenarios. In addition, it is highly vendor-specific.

[53] https://www.nxp.com/, 12/05/2020

[54] https://www.infineon.com/, 12/05/2020

Table 4.6: *Data Types* available in the various *Data Streams* of Automotive IT components

Data Type	*Non-volatile Memory*	*Volatile Memory*
DT1	✓	✓
DT2	✓	✓
DT3		
DT4	✓	✓
DT5	✓	✓
DT6		
DT7		
DT8	✓	✓

4.1.3.6 Availability and Access to the Forensic *Data Streams* in Automotive IT

This section provides an overview on the availability of forensic traces in *Non-Volatile Memory* and *Volatile Memory* in Automotive IT and the means to access these forensic *Data Streams.*

Components in Automotive IT share the lack of memory with the components employed in the ICS domain. The size of *Non-volatile Memory* and *Volatile Memory* are in the same order of magnitude in both domains (see Section 4.1.2.5).

A distinct difference to the other domains discussed in this thesis is the fact that computing unit in Automotive IT (ECUs) are usually not shut down or disconnected from power supply when the vehicle is stopped or turned off. Usually these ECUs are powered by the vehicle battery and only exceptional cases lead to a loss of power. This has an impact on the relationship between *Live Forensics* and *Post-Mortem Forensics* as discussed in Section 4.1.3.1.

ECUs contain a number of *Data Types* which could be useful for the forensic process as can be seen in Table 4.6. This includes **DT1**, **DT2**, **DT4** and **DT5**. Information required for the driving task, like the current speed of the vehicle, is also present in *Non-volatile Memory.* Such data would be considered as **DT8** even though the current definition in the [KDV15]-model does not describe this data in a satisfactory manner. A mitigation for this is discussed in Section 5.3.2.

Especially useful for the forensic use are DTCs (Diagnostic Trouble Codes). These DTCs describe certain conditions. This includes the detected failure of attached components or values outside the normal operation parameters.

In general, classical *Post-Mortem Forensics* is difficult in Automotive IT. *Non-volatile Memory* is only accessible when opening up the ECUs used in Automotive IT. These ECUs are designed to survive adverse physical conditions and sturdily built. Opening up these components is made more difficult by the inclusion of vendor-specific intellectual property protection mechanisms. The *Non-volatile Memory* is also often integrated directly with the processing unit. An example for employing physical access to perform **DG** on *Non-volatile Memory* in Automotive IT can be found in [RA02].

Another option to obtain *Non-volatile Memory* in Automotive IT is to use *Live Forensics* or *in the loop Forensics* (as discussed in Section 4.1.3.1). In this case, diagnostic requests are used to query for the contents of *Non-volatile Memory* and *Volatile Memory* within an ECU. In *Live Forensics*, this is done with an ECU which has not been disconnected from the power supply. In *in the loop Forensics* a specific ECU which has been disconnected from the power supply is reconnected to a simulated environment including a power supply. Ideally, this environment is created in a way to prevent the alteration of data by triggering new alarms or unusual conditions which would cause further DTCs.

Several protocols and tools exist to perform a **DG** in such a manner. A comprehensive discussion on the forensic use of such protocols is provided by Section 4.1.3.3. All these protocols share the need for vendor-specific information in order to work efficiently and the fact that they use the digital communication bus used for the usual operation of Automotive IT for the diagnostic requests. The shared use of the bus for normal operation and diagnostic request caused a *Structural Impact* on the communication. The diagnostic requests itself cause a *Structural Impact* in the ECU and have to rely on the ECU providing correct responses. These two factors limit the evidentary value and hence the usefulness for the forensic process. This usefulness is further limited by the fact that the tools used to perform these diagnostic requests are not designed for the use within forensic scenarios. A survey on open source tools usable for this approach is presented in [**ALKD17**]. These tools can form the foundation for the creation of forensic tools (following the guidelines provided in [**ALKD17**]).

A set of calibration protocols used for the fine-tuning of ECUs exists. These protocols are highly vendor-specific and use the regular communication bus for communication with the ECUs. Hence, these protocols behave in the same way as the diagnostic protocols with regard to forensic usefulness.

Volatile Memory in Automotive IT is obtained in a similar fashion to *Non-volatile Memory*. Using either *Live Forensics* or *in the loop Forensics* diagnostic requests performed by dedicated software is employed. Basically there is no difference for the diagnostic tool whether data from the *Non-volatile Memory* or *Volatile Memory* is queried. The implications for the forensic process in terms of *Structural Impact* by altering the state of the executing ECU and the *Communication* within the communication bus are the same. However, the knowledge if a given date is stored in *Non-volatile Memory* or *Volatile Memory* is relevant for the forensic investigator if the investigation is performed in *in the loop Forensics* and the date stored within *Volatile Memory* would therefore be lost.

An additional possibility to obtain *Volatile Memory* is provided by debug protocols. These protocols aim towards a low *Structural Impact* on the state of the executing ECU and use a dedicated communication line to the specific ECU which prevents a *Structural Impact* on the *Communication*. These protocols rely on the specific ECU possessing an additional diagnostic MCU and a dedicated communication interface like BDM (Background Debug Mode). Employing these protocols requires physical access to the interfaces within the ECU. These protocols are highly vendor-specific.

In general, all the methods to query the ECU for information have to rely on the fact that the ECU is not tampered with and hence able to provide truthful answers to these requests. Not being able to ensure this reduces the evidentiary value of all traces gathered by these requests. This is further impacted by the fact that the software designed for these processes is not designed for the forensic process.

Another major factor is the necessary knowledge of vendor-specific protocols, implementations, commands and knowledge to send these diagnostic requests. This knowledge is inherent to the proprietary and closed source applications used by vendors. Hence, these tools are also able to interpret the responses to said request. For an independent forensic investigator without access to these vendor-specific solutions or the bare knowledge, the domain is far less accessible. However, there is a selection of usable open source tools (as presented in [**ALKD17**]) and approaches to deduce such vendor-specific knowledge ([**KAH^{+}18**] and [Smi16]).

4.2 *Step 1.2* Analysis of System Architectures and Communication

This section analysis the system architectures used in the various domains discussed in this thesis. These system architectures form the basis for the *Communication* present when various components of the discussed domains work together in order to perform a specific task. These tasks also have an impact on the specific data which is communicated between the various components and are discussed in Section 4.3.

The analysis is performed in a similar fashion than the one performed on the specific components in Section 4.1. At first, some general considerations on the architectures are given. Then the *Data Types* available within the *Communication* in these architectures are identified before the possibilities to obtain and interpret this data are discussed. A summary follows each of the specific domains. Again, the Desktop IT domain (Section 4.2.1) is discussed as a baseline before the ICS (Section 4.2.2) and the Automotive IT domain (Section 4.2.3) are then discussed.

4.2.1 Desktop IT Architectures

The state purpose of Desktop IT established in Section 2.2.1 is to *receive, store, manipulate and transmit data*. Receiving and transmitting data requires communication with other systems able to also receive and transmit data. This communication relies on a common shared medium over which this transmitting of data can take place. The consequence of this is described in [LL11] for a business context:

> *„The resulting IT infrastructure links different pieces of computer hardware and smaller networks into an enterprise-wide network so that information can flow freely across the organization and between the firm and other organizations. It can link different types of computer hardware, including mainframes, servers, PCs, mobile phones, and other handheld devices, and it includes public infrastructures such as the telephone system, the Internet, and public network services. The enterprise infrastructure also requires software to link disparate applications and enable data to flow freely among different parts of the business, such as enterprise applications [..] and Web services [..].“*

This diverse range of different computer systems relies on a surprisingly narrow and standardized set of technologies in order to provide common shared media and the protocols required to facilitate communication.

An overview on the communication interfaces included in contemporaneous computer systems from the domain of Desktop IT was presented in Table 4.1 in Section 4.1.1. The *Personal Computer Examples* each contains either Ethernet or 802.11ac interfaces. Ethernet[55] describes a family of standardized, cable-bound networking technologies. 802.11ac$^{\text{TM}}$[56] is better known as WLAN (Wireless Local Area Network). Both protocols can rightly be considered the standard media for communication in the domain of Desktop IT. This is further exemplified

[55] http://grouper.ieee.org/groups/802/3/, 13/05/2020
[56] http://grouper.ieee.org/groups/802/11/, 13/05/2020

by the fact that publications on network forensics like [ENI19] do not consider any other media at all.

The architectures employed in the Desktop IT domain are less standardized. Since Desktop IT is used in different scenarios, including home networks and business networks, it is difficult to find a generalized architecture for networks in Desktop IT. However, for a better understanding of network architectures, two exemplary business networks are presented here before the discussion shifts to the topic of availability and accessibility of forensic traces within the *Communication* in the Desktop IT domain.

Typical Computing Configuration for the era of Enterprise Computing (1992-present) based on [LL11]

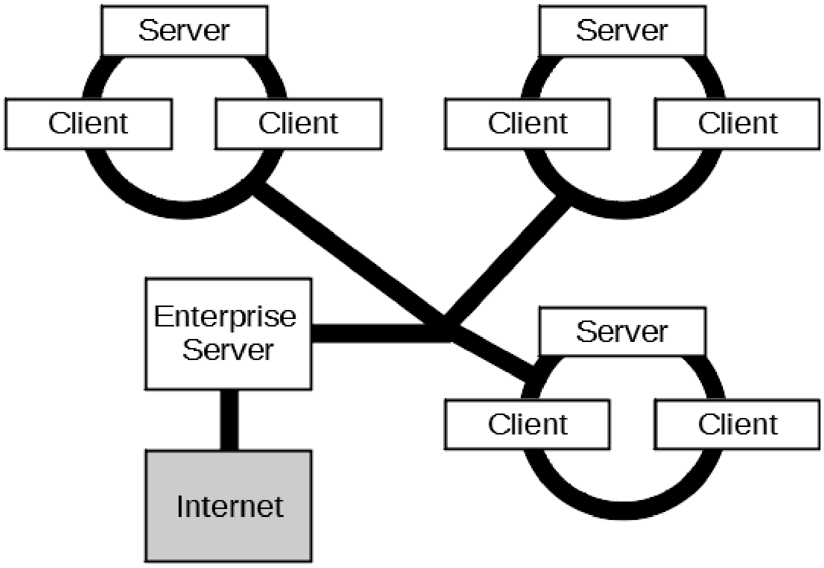


Figure 4.4: Typical computing configuration for the era of Enterprise Computing (1992-present) based on [LL11]

A typical network configuration as presented in [LL11] as an example for an IT infrastructure is shown in Figure 4.4. This example includes three subnetworks with some clients or workstation, one server for localized functionality. These subnetworks are connected to each other and to an Enterprise server which provides access to the Internet and usually also provides some additional functionality like serving as a Web-Server or EMail-Server. Note that Figure 4.4 does not include any of the elements necessary to maintain such a network infrastructure.

Model Infrastructure for a Small Business Based on [BSI11]

A more detailed and complex example for an IT infrastructure can be seen in Figure 4.5. This infrastructure is based what on what [BSI11] designates as a *Musterlandschaft* (German for *„Model landscape“*, but more accurately translated as *„Model infrastructure“* in this context). This setup reflects the IT of a fictive small business. This business manufactures a product at a secondary location, markets and sells it.

The setup includes the various clients (marked with a **C**), services (marked with a **S**) and also the necessary infrastructure elements (marked with a **N**) to connect them together. The

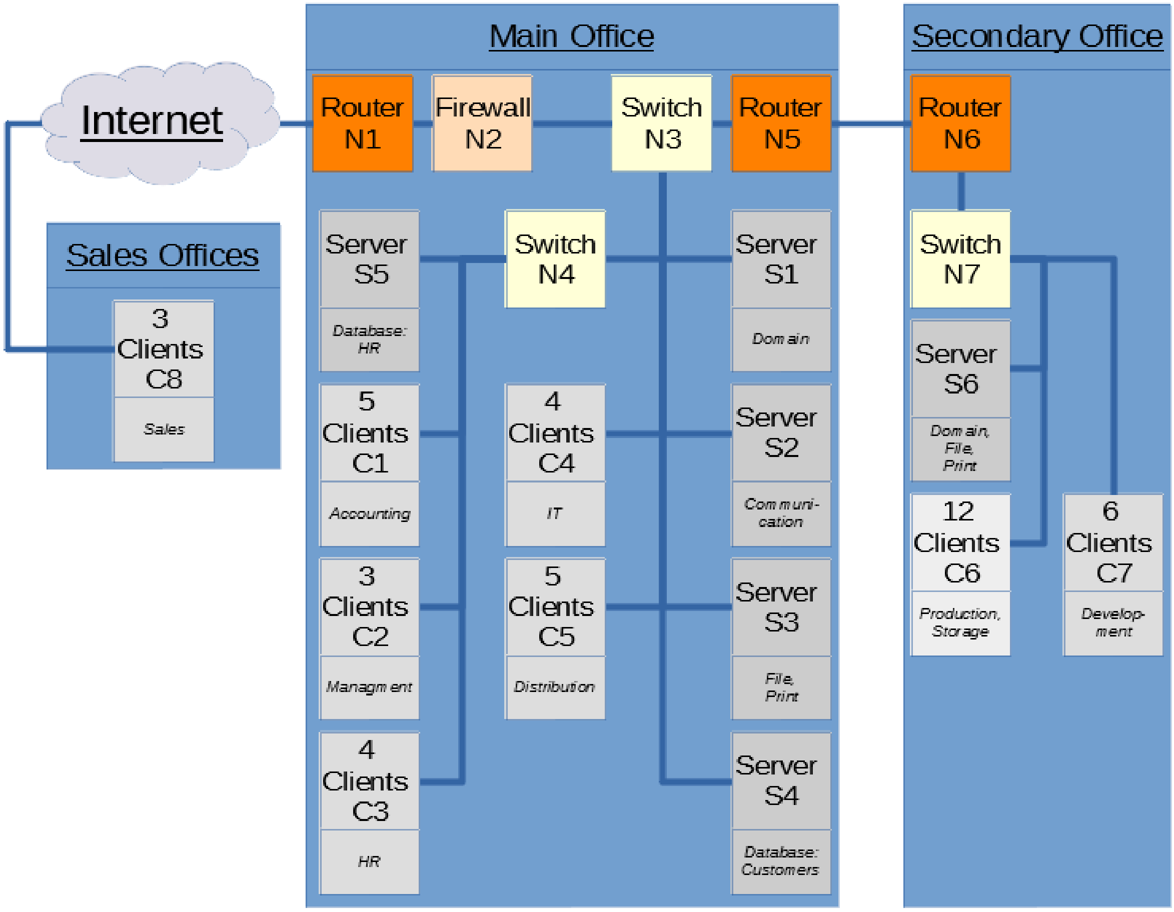


Figure 4.5: Model infrastructure for a small business based on [BSI11]

exemplary infrastructure is split between two physical locations. One is representing the Main Office, while the other represents a Secondary Office (and the place in which the manufacturing is handled).

The Main Office contains a subnetwork. The systems there are connected to a switch *N4* which handles the communication between the systems within this subnetwork and the communication to and from the main network of the Main Office. This subnetwork is tasked with accounting, management and human resources (HR). The main network is connected via the switch *N3*. It contains various servers providing services and various clients. The communication to the subnetwork, other location and the Internet is handled by switch *N3*.

Router *N5* and Router *N6* connect the Main Office and the Secondary Office. The Secondary Office contains a server (*S6*) and a couple of clients. Some of them are responsible for manufacturing processes which shows the connection between Desktop IT and ICS as discussed in Section 2.2.2. The switch *N7* is used to facilitate the communication within the Secondary Office.

The Main Office also has a connection to the Internet. This is realized using the firewall *N2* and the router *N1*. A number of external Sales Offices are connected to the business network using the Internet.

4.2.1.1 Availability of *Data Types* in *Communication* in Desktop IT

The contents communicated in the Desktop IT domain are as diverse as the employed architectures. The *Data Types* introduced in the [KDV15]-model were chosen to represent this width of data (see Section 3.1.2.2).

In general, every *Data Type* can occur in the *Communication* between systems in the Desktop IT domain. Notable is **DT2** which represents the raw, uninterpreted data transferred between these systems. **DT3** includes the meta data required to facilitate this transfer. In the case of communication networks it includes network addresses. Sharing **DT8** is in essence the main purpose of using Desktop Computers in the first place. Hence, **DT8** usually has a notable share on the traces available in *Communication* in this domain.

Types of Communication

Further considerations on how to describe the communication performed in the Desktop IT domain was performed in [**ADKK14**]. This consideration is not really motivated from a technical point of view but more from a point of view of the purpose of the communication. Hence, the classification scheme discussed in [**ADKK14**] describes what an user is intend to achieve by a certain form of communication. This categorization based on the intent during the communication is, in essence, a finer subdivision of **DT8**. However, it also includes communication performed by the system in order to provide the user with the possibility to communicate.

Since the intent of a given communication is of interest when performing a forensic investigation in **IC2**, a brief overview on the categories which describe elements belonging to **DT8** is given. This exemplifies the entire diversity of data summarized in **DT8**. In order to better fit within the [KDV15]-model for the forensic process, some alterations and additions are made. The categories presented in [**ADKK14**] are:

- **Information Research**
  Covers all instances of Internet-based communication where information on the Internet is accessed or searched for. This would be a subcategory of **DT8**.

- **Placement of Information**
  Covers all instances of Internet-based communication where information is placed on the Internet. This would be a subcategory of **DT8**.

- **Direct Communication**
  Covers the direct communication between a limited number of participants or a non-persistent communication to an unlimited number of participants (broadcast). This would be a subcategory of **DT8**.

- **Location-based data and Navigation data**
  Covers all data transferred during the use of map or navigation software. This would be a subcategory of **DT8**.

- **Trade**
  Covers all communication in which a financial transaction or a trade takes place. This would be a subcategory of **DT8**.

4.2.1.2 Gathering and Investigating the *Communication* in Desktop IT

As discussed in Section 3.1.2.5, *Communication* can be captured only at the moment it occurs. If it is not captured during this moment, the potential evidence is lost. This does not mean that there will be no hint at the fact that *Communication* took place, since the *Communication* will usually cause some alteration to the state of the participating systems. The *Communication* happened for a reason in the first place which usually includes the transfer of information which should then be present in the recipient of the communicated message.

In order to capture *Communication*, physical access to the medium used for transmitting this *Communication* is required. This can be done by directly accessing the medium or by accessing a device which is already attached to the medium. After this access is gained, the traffic can be recorded (**DG**). However, the question at which location within the network architecture this recording takes place is relevant as it dictates which *Communication* is captured. After the capturing, the *Communication* can be subsequently investigated (**DI**). This also lays out the outline for this section.

Gaining physical access to a medium usually involves inserting a dedicated device to perform the **DG**. If such a device is already attached to the medium, this step can be skipped. How the physical access to the medium is possible differs depending on the medium. In Section 4.2.1 two different media were introduced. Ethernet, which is cable-based, and WLAN (Wireless Local Area Network) which is using radio. These two differ in terms of how they are accessed.

Accessing the Carrier Medium in Cable-Based Networks

Various possibilities to gain physical access to cable-bound carrier medium are discussed in [ENI19]. These possibilities form a solid foundation for an overview:

- *Inline network taps*
  These devices can be inserted between two physically connected devices. They commonly possess four network interfaces. Two of these are used to connect the two formerly directly connected devices. The other two are used to mirror the network traffic. This implies some devices connected to these two network interfaces which record the data. The insertion of the network tap into the network causes a disruption (while the two physically connected devices are disconnected and the connections are replugged to include the network tap). Network taps are easy to obtain and comparatively affordable. An example for a network tap is the *Throwingstar*[57].

- *Vampire taps*
  Vampire taps do not cause a disruption to the network when applied. They are directly put on the network cable between two physically connected devices and then penetrate the shielding of the cable in order to insert a probe which is then able to duplicate the data to one network interface. It is also possible that two network interfaces are used -

[57] https://greatscottgadgets.com/throwingstar/, 16/06/2020

one for each direction of communication. Devices have to be attached to these network interfaces in order to record the data. The process of penetrating the shielding of the cable carries some risk of physically destroying the cable. Vampire Taps have to be specifically built for a certain type of cable. These devices are considerably harder to obtain. In fact, no mass-produced version could be found by the author of this thesis.

- *Fiber-optic network taps*
  These are used when a Fiber-optic cable is used as a carrier medium. They might require splicing the fiber-optic cable - a process which is not reversible and requires dedicated tools. The process of inserting such a network tap disrupts the network communication for a brief period while reattaching the connection. Due to technical limitations the inclusion of such a tap into a carrier will reduce the signal strength which might prove problematic. Fiber-optic network taps can be bought from various manufacturers (e.g. *Profitap*[58]). They are significantly more expensive than network taps for copper-based cables.

- *Hubs*
  A hub is a type of network infrastructure element which connects all the stations on local subnet together. Every piece of communication send from any system within the subnet to any other is forwarded to all systems since the hub does not maintain any information on which system in the subnet is connected to which port. Hence, all communication on the subnet can simply be recorded by dedicated devices attached to one of the ports. A hub is usually already part of the network infrastructure before an incident happens. Hence, starting to record does not disrupt the network. Hubs are hard to obtain in these times since switches (see the next point) offer a better networking performance. Currently, most of the devices sold as hubs are actually switches which do not share this useful characteristic. Some leftover hubs might still be for sale and would most likely not imply great acquisition costs.

- *Switches*
  Switches replaced hubs and offer a greater networking performing by keeping track of which system is connected to which port. Hence, the switch is able to send any given message only to the designated recipient of the message. This saves bandwidth but does mean that not every attached device is able to monitor all communication within the given network. However, many switches contain a *Monitor port*. This port can be configured to mirror the communication send out on any given other port. A practical problem would be one of bandwidth since the combined traffic on the ports mirrored might excess the bandwidth of the monitor port. A device which captures the data most be attached to the monitor port in order to record the data. Since switches are part of basically any network infrastructure, as can be seen in the two examples provided in Section 4.2.1), switches are commonplace and present within the network anyways. Hence, configuring the monitor port should usually not disrupt the network as long as the switch does not need to perform a restart. Depending on the network layer the switch operates, it might also be referred to as a router.

[58] https://www.profitap.com/fiber-taps/, 16/06/2020

Accessing the Carrier Medium in Wireless Networks

Gaining physical access to a wireless network is easier than in cable-bound networks. This is due to the fact that these wireless networks use air as a carrier medium.

The most relevant points about accessing the carrier medium in wireless networks are summarized by [ENI19]:

> *„To capture WLAN traffic, investigators need an 802.11 wireless card capable of running in Monitor mode; a mode that many WLAN cards do not support. There is a difference between Monitor mode and Promiscuous mode that can be summed up as follows:*
>
> - *Monitor mode: Packets are captured without associating to an access point. Traffic from (and to) all access points and stations in radio range will be captured, independent of SSID. This can be thought of an analogue to standing in a room and listening to people's conversations.*
> - *Promiscuous mode: Packets are captured after associating with an access point and the system will be listening to all packets, even those not addressed to it. 'All packets' in promiscuous mode means packets from all stations being associated with the AP. This can be thought of an analogue to joining a group of people in a conversation, and hearing sentences not related to you.*
>
> *An important difference between monitor mode and promiscuous mode is that in monitor mode the packets are captured in 802.11 format while in promiscuous mode they are presented in Ethernet (802.3) format. From a forensic standpoint, monitor mode is preferable as it is completely passive and conveys more information. "*

In addition to the possibility of using a wireless card in monitor mode some possibilities from cable-bound networks can be used. WLANs might include infrastructure elements. Common examples are WLAN-routers which connect cable-bound and wireless networks. These network infrastructure elements might also contain Monitor ports which could be used to acquire the network traffic in wireless networks.

A general benefit in wireless networks is that the medium will not be disrupted by a recording of the network traffic.

After Gaining Physical Access (DG)

After physical access to the specific carrier is gained, the network traffic has to be captured. In the case of network taps or Monitor ports the traffic is delayed to dedicated ports where it can be recorded. There exist a wide variety of tools in order to perform packet capturing. This section will give a brief overview on some common ones:

- tcpdump[59]
  tcpdump is an open source tool designed to dump traffic on a network. This traffic can be saved to a file for later analysis.

[59] https://www.tcpdump.org/, 16/05/2020

The tool itself is a command line tool which eases the use within a forensic scenarios since the inputs and output of the command line can easily recorded by external tools.

- netsniff-ng toolkit[60]
  netsniff-ng toolkit is another open source tool able to capture network traffic and save it to a file for subsequent analysis.

  Like *tcpdump*, this is a command line tool with the same benefits to perform a *Process Accompanying Documentation*.

- Wireshark[61]
  Wireshark is an open source network protocol analyzer. It offers a broad range of possibilities during **DI**. It can also be used during **DG** to capture network traffic. *Wireshark* is able to analyze the captures created by the other tools presented in this section.

- Linux Forensic Transparent Bridge (LFTB) ([**KHAD10**])
  The *Linux Forensic Transparent Bridge (LFTB)* is a system designed to perform traffic capturing in forensic scenarios.

  All the tools presented in this section so far are not designed for the use in a forensic investigation and hence do not include any mechanisms to ensure integrity or authenticity of the gathered data. While the command line tool allows to perform a *Process Accompanying Documentation* with relative ease by recording the inputs and outputs of the command line documenting the actions performed is not included in their functionality. Section D.4 discusses some considerations for the properties tools designed for the use in forensic scenarios should have. How the *Linux Forensic Transparent Bridge (LFTB)* meets these criteria is discussed in Section D.4.

  The *Linux Forensic Transparent Bridge (LFTB)* documents the actions performed. The integrity and authenticity of the captured data and the documentation of the performed actions are ensured using cryptographic means. The *LFTB* is designed to have no *Structural Impact* on the *Communication* on the network by not performing any operations in which data is transmitted to the network.

This overview provides some usable tools to perform **DG** on *Communication* in Desktop IT. However, all the tools presented here have to rely on the physical access provided by other means. Ideally, these tools would be installed on a dedicated system which does not interact with the monitored network at all and is connected to a network port used for recording (Monitor Port in case of a switch). However, these tools could also be run on any of the systems already present in the network. This would, of course, alter the state of these systems and would cause some *Structural Impact* on *Volatile Memory*. This is also the case for *Non-volatile Memory* if the recordings are to be stored on mass storage.

It is of note that configuring a Monitor port during an incident will also alter the state of the switch. A *Structural Impact* on its *Non-volatile Memory* and *Volatile Memory* is implied.

It is worth considering, that if the installation of a network tap disrupts the communication there is a *Structural Impact* on the *Communication* none of the tools can prevent. Installing

[60] http://netsniff-ng.org/, 16/05/2020
[61] https://www.wireshark.org/, 16/05/2020

network taps before an incident occurs (during **SP**) prevents this problem. Since network taps are cheap and easy to obtain, this is possible approach. Of course, these network taps need to inaccessible for unauthorized persons since they could be used to eavesdrop on *Communication* - which is essentially their function in the first place anyways.

Placement of Means to Obtain Access to the Carrier Medium in Desktop IT Architectures

A general consideration for performing **DG** on *Communication* is where to place the means to gain access to the carrier medium. This has to be based on the network architectures. Two examples were shown in Section 4.2.1. One has to consider which network and subnetwork is of interest for the forensic investigation. This dictates the placement of the specific points to perform the **DG**.

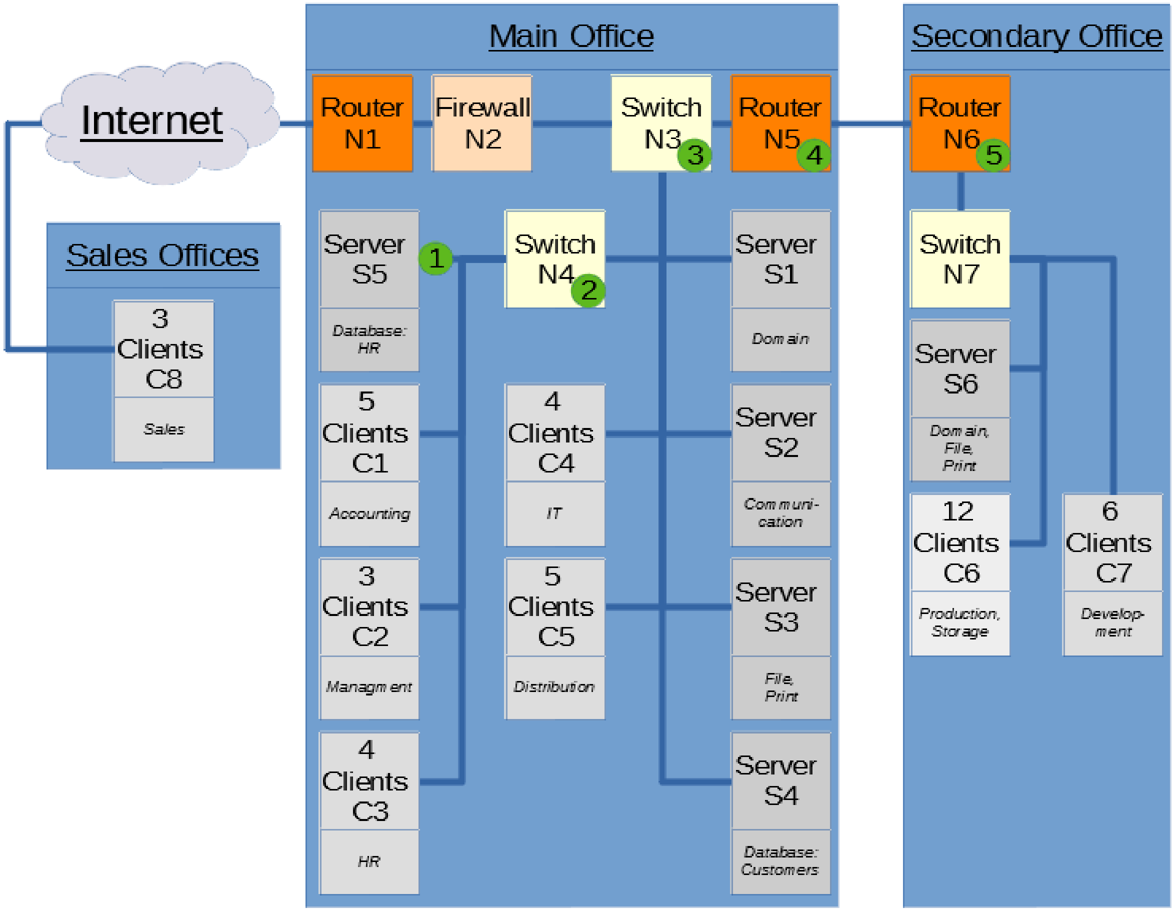


Figure 4.6: Model infrastructure for a small business based on [BSI11] including different potential locations to perform **DG**

Some representative examples for possible locations where to gain physical access to the carrier, which is cable-bound in this case, are presented in Figure 4.6. An overview on the method needed to gain access at these locations and the potential *Communication* available at these locations is presented in Table 4.7.

Table 4.7: Comparison of the potential access to *Communication* in case of different access points to the carrier medium in Figure 4.6

Descriptor	Type of access
Accessible *Communication*	
Capture Location 1	Network tap between *N4* and *S5*
incoming to *S5* outgoing from *S5*	
Capture Location 2	Monitor port of *N4*
entire subnetwork: between *S5*, *C1*, *C2* and *C3* **subnetwork to other networks:** incoming for *S5*, *C1*, *C2* and *C3* from all others outgoing from *S5*, *C1*, *C2* and *C3* to all others	
Capture Location 3	Monitor port of *N3*
entire subnetwork: between *S1*, *S2*, *S3*, *S4*, *C4* and *C5* **subnetwork to other networks, including subnetwork at *N4*** incoming to *S1*, *S2*, *S3*, *S4*, *C4* and *C5* from all others outgoing from *S1*, *S2*, *S3*, *S4*, *C4* and *C5* to all others	
Capture Location 4	Monitor port of *N5*
Main Office to Secondary Office between *N5* and *N6*	
Capture Location 5	Monitor port of *N6*
Main Office to Secondary Office between *N5* and *N6*	

The potential locations in Table 4.7 generally tend to use the Monitor Port of switches since it allows for the recording of the network traffic belonging to an entire network. In the cases of **Capture Location 2** and **Capture Location 3** this might lead to problems with the bandwidth in some cases. The switch *N4* (**Capture Location 2**) is responsible for coordinating the network traffic between *S5*, *C1*, *C2* and *C3*. Without any connections to other networks, this covers already twelve clients and one server. Hence, in theory, the bandwidth of any single port times 13 could be present as network traffic - this is too much for a Monitor Port. For *N3* (**Capture Location 3**) this effect is even more extreme.

The Monitor Port can however be configured to only mirror a selected subset of *Communication*. Examples would include the traffic to and from other networks. Since the other networks are attached by using a single port, this single port could be mirrored without concerns for bandwidth.

This is also the case for **Capture Location 4** and **Capture Location 5**. It can be assumed that the respective Monitor Ports of *N5* and *N6* are able to mirror the entire traffic in terms of bandwidth. In essence this traffic covers all the *Communication* between the two physical locations. At the first glance, **Capture Location 4** and **Capture Location 5** do not seem to differ. In normal operation, the network traffic captured at both locations will be identical. In the case of a potential security incident (**IC1**), this might not be the case. If an attacker is able to manipulate the traffic between these two locations, **Capture Location 4** and **Capture Location 5** would show differing network traffic.

Capture Location 1 represents a network tap between *S5* and *N4*. It would only provide access to the *Communication* between these two components. This includes traffic directed to *S5* from outside sources and traffic directed from *S5* to outside sources. Installing such a network tap while an incident occurs would cause a disruption of the network traffic. This would lead to a *Structural Impact*. However, a network tap at the specific location could be installed beforehand. In case the need arises a tool to gather the traffic could be attached without causing a disruption.

In general, recording network traffic can be considered a *Method* of **SMG**, since usually, capturing the entire data is constraint by issues or privacy (see Section 2.1.6) and practicality (the massive amount of data stored). However, the identification of places where the installation of network taps or the gathering of data from Monitor ports could be useful is a part of the *Strategic Preparation*. Here, the administrator of the infrastructure in question can deploy those switches with Monitor ports or install network taps at positions of interest.

Investigating the Captured Network Traffic (DI)

Investigating captured *Communication* is an important point of many forensic investigations in Desktop IT. Hence, there exist a broad range of publications on this topic. The aforementioned [ENI19] provides a practical introduction into the field of *network forensics*. The same topic is referred to as *internet forensics* in [Arn17]. [LPM04] simply describes it as *network evidence*.

Despite the great variety of terms all **DI** performed on *Communication* boils down to the analysis of the captured network traffic. In order to perform such an analysis, knowledge of the specific protocols used during the *Communication* is necessary. While a broad range of proprietary and closed source protocols is in existence a significant portion of protocols is standardized and generally known to the public.

This is exemplified by the range of protocols supported by *Wireshark*[62]. This open source tool is widely used to analyze network traffic in detail. As long as the protocol specifications are known additional modules to interpret and analyze these protocols can be added. As discussed for its use during **DG**, *Wireshark* is not designed with forensic use in mind. It does not ensure the integrity and authenticity of the capture analyzed and does not document the actions performed. However, the former can be solved by working on copies of the captures and using external cryptographic means to ensure integrity and authenticity. The later can be solved by using the command line interface *tshark*[63] which enables the easy use of external measures to produce a protocol of the performed operations.

In general, everything can be investigated using protocol analyzers. However, depending on the amount of data present within the system, some approaches are necessary in order to reduce the work load. All protocol analyzers contain powerful filters to exclude certain pieces of network traffic of no relevance for the current investigation. Understanding which pieces of network communication are important for the investigation is a non-trivial task. Knowledge about the usual traffic within a network is highly useful.

Obtaining such knowledge can be referred to as *Baselining*. This includes getting samples of known-good communication in order to understand which traffic on the network could

[62] https://www.wireshark.org/, 16/05/2020

[63] https://www.wireshark.org/docs/man-pages/tshark.html, 16/05/2020

be considered as normal. Such an undertaking would have to take place during **SP** since a known-good is not available anymore after an incident is happening. The reasoning on why to obtain such a baseline can be deduced the underlying principles of intrusion detection from [Bis18]:

> *„The actions of the users and processes generally conform to a statistically predictable pattern. An user who does only word processing when using the computer is unlikely to perform a system maintenance function.“*

In essence, the aims of a forensic investigation and intrusion detection in regard to *Communication* are similar - to identify unusual behavior. Hence, the tools used for intrusion detection can also be used in the scope of a forensic investigation, especially in **IC1**. This goes back to the close relationship between computer forensics and incident response as shown in Section 2.3.1. Intrusion Detection Systems (IDS) can help the forensic investigator to identify interesting aspects of the network traffic under investigation. There exist various solutions which analyze the network traffic using statistical methods, heuristics (all feed from a baseline that has to occur during **SP**) and known attack signatures. An example is the open source solution *snort*[64].

The increasing use of encryption in network communication increases the difficulty of performing investigations into network traffic. An approach to lessen the impact of encryption on the forensic investigation was discussed in [**ACK$^+$13**] and [**KAD16**]. Here, statistical pattern recognition is used to extract some information about the actions taking place within the encrypted network traffic without attacking the encryption directly. [**ACK$^+$13**] shows that the identification of the use case performed with the investigated software is still possible. However, it relies on models based training data which would have to be collected before a potential incident happens.

4.2.1.3 Availability and Access to the Forensic *Data Streams* of *Communication* in Desktop IT

The forensic process for the use of *Communication* in Desktop IT is well-established. Various works discuss these topics to varying extent. Some notable examples are [ENI19], [Arn17] and [LPM04]. *Communication* in Desktop IT contains all *Data Types*.

DG relies on obtaining access to the carrier medium. In the case of cable-bound media this can be done using network taps or by using components already attached to the medium. If the network taps are not installed during **SP** the installation of these interfaces will cause a disruption of the *Communication*. Various tools exist to capture network traffic. Since the capturing has a major *Structural Impact* on the capturing system this ideally is a dedicated system only attached to the network tap and with no relation to the monitored network at all. However, a computer system with in the network can also be used. If the system on the network is only used to forward the data to such a system dedicated to capture the network traffic it would behave in the same way as a switch relaying network traffic via Monitor port to such a dedicated system.

In the case of a wireless medium, physical access is easier. A dedicated computer with a suitable network interface can record the network traffic without any interruption.

[64] https://www.snort.org/, 16/05/2020

Placement of the physical access to the network is a relevant topic. The investigator has to understand which *Communication* is available when the **DG** is performed at a certain location within the network infrastructure. This consideration is best made before an incident occurs (during **SP**) or at very least before the **DG** starts (during **OP**).

The protocols used in the Desktop IT domain contain a broad range of proprietary and closed source protocols as well as a significant portion of protocols which are standardized and generally known to the public. As long as the protocol specifications are known, powerful tools to perform **DI** and **DA** are available. Although these tools are generally not designed for the use within forensic scenarios, their drawbacks can be circumvented.

In general, *Communication* in the Desktop IT domain can be accessed with relative ease and investigated in a satisfactory manner, showing the maturity of this research field.

4.2.2 Industrial Control Systems Architectures

Industrial Control Systems rely on the cooperation of various components in order to achieve an 'industrial objective' as discussed in Section 2.2.2. This cooperation requires the creation of control networks. These networks follow specific patterns and hierarchies. Understanding these hierarchies allows an understanding of what constitutes *Communication* in the ICS domain. The following section will discuss these hierarchies and the protocols used to implement them, before the available *Data Types* within this *Communication* are identified. A discussion on how to obtain these traces from the *Communication* follows before the entire section is summarized.

4.2.2.1 Industrial Control Systems Hierarchy

In general, ICS implement a far more standardized hierarchy than networks in the Desktop IT domain. As discussed in Section 2.2.2.1, the Purdue Enterprise Reference Architecture (PERA) - [Wil92]) is often used to describe enterprise hierarchies including ICS. PERA describes six different levels in ICS hierarchies. These six levels are mapped on three different zones (see Figure 2.3 for a visual reference). As already discussed in Section 2.2.2.1, the **Enterprise Zone** represents business functions which are performed by systems from the classical Desktop IT domain. This domain already has been discussed in Section 4.2.1. The **Manufacturing Zone** represents the ICS-specific network which is of interest during this section.

The same distinction between the different levels is used in [KL15] which describes the segmentation of network within ICS. This network segmentation implements the network architecture. Such a network segmentation is visualized in Figure 4.7. Although the levels do not share the same terminology as PERA, they share the same function. For example, the lowest level describes the various process networks in [KL15] and in [Wil92]. This shows the hierarchy of ICS. Various *Process networks* (*Level 0 - Process* in PERA) are connected to a *Control network* (*Level 1 - Basic Control* in PERA). Various *control networks* are then connected to a *Supervisory network* (*Level 2 - Area Supervisory Control* in PERA). This network is then connected to a *Plant control network* (*Level 3 - Site Level* in PERA). This is then attached to an *Operations planning network* (*Level 4 - Site Business Planning and Logistics* in PERA). A *Business network* (*Level 5 - Enterprise* in PERA) forms the top of the hierarchy. This relationship is visualized in Figure 4.8.

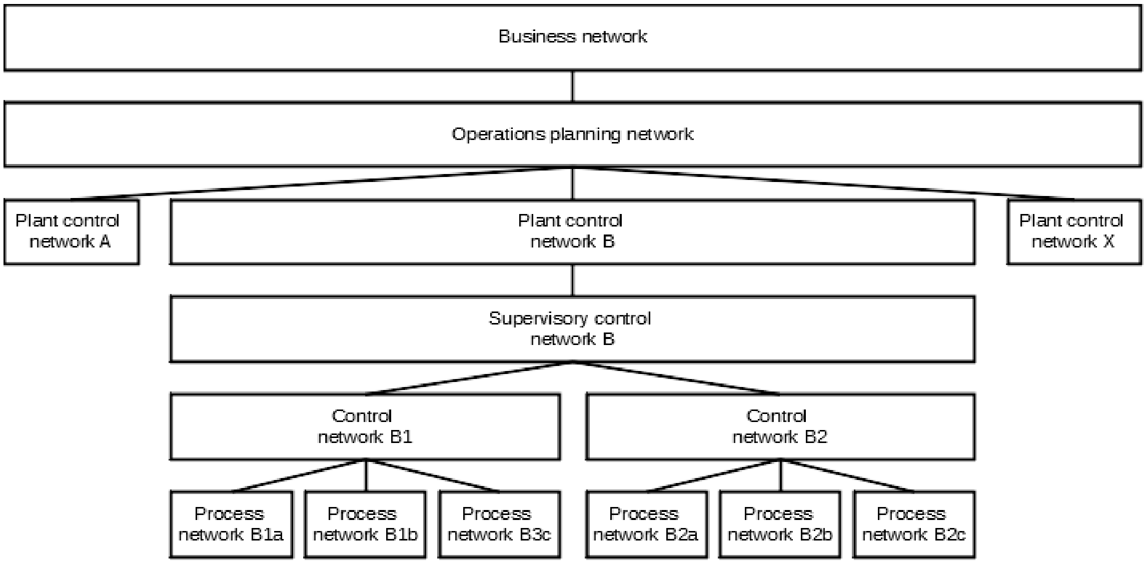


Figure 4.7: Network segmentation in Industrial Control Systems based on [KL15]

All these networks are segmented. Hence, the communication between the given networks has to be conducted using infrastructure elements. These infrastructure elements could be switches as discussed in Section 4.2.1. Also, the communication has to occur along the lines of a hierarchy. Hence a communication in the infrastructure shown in Figure 4.7 between *Process network B1a* and *Process network B2b* would have to traverse along the lines of *Control network B1*, *Supervisory network B* and *Control network B2*.

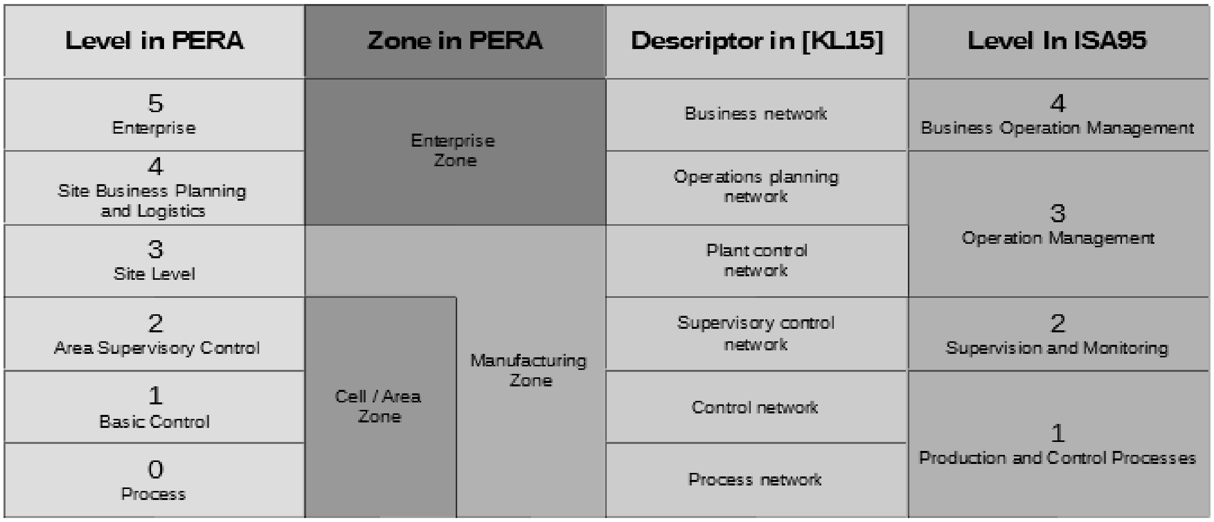
The collection of standards provided as *ISA95-Standard* by the International Society of Automation[65] uses a slightly different model to describe the communication hierarchy in ICS. Since this communication hierarchy contains a description of the specific communication flows. This standard consists of four different levels ([ISA10]):

- production and control process
- supervision and monitoring
- operation management
- business operation management

The mapping of these levels to the levels of PERA is provided in Figure 4.8. In essence, the *Operation Management* (*Level 3*) of ISA95 performs the same functions as *Level 2 - Area Supervisory Control* and *Level 3 - Site Level* in PERA. *Production and Control Processes* (*Level 1*) in ISA95 covers *Level 0 - Process* and *Level 1 - Basic Control* in PERA.

Of higher relevance for exploring the *Communication* within ICS architectures than these specific definitions of various levels is how the components from these levels communicate with other components within the hierarchy. This communication flow is shown in Figure 4.9

[65] https://www.isa.org/isa95/, 17/05/2020



Level in PERA	Zone in PERA	Descriptor in [KL15]	Level In ISA95
5 Enterprise	Enterprise Zone	Business network	4 Business Operation Management
4 Site Business Planning and Logistics		Operations planning network	3 Operation Management
3 Site Level	Manufacturing Zone	Plant control network	
2 Area Supervisory Control	Cell / Area Zone	Supervisory control network	2 Supervision and Monitoring
1 Basic Control		Control network	1 Production and Control Processes
0 Process		Process network	

Figure 4.8: Mapping between levels of ICS hierarchy in the Purdue Reference Enterprise Architecture ([Wil92]), the terminology used in [KL15] and *ISA95* ([ISA10])

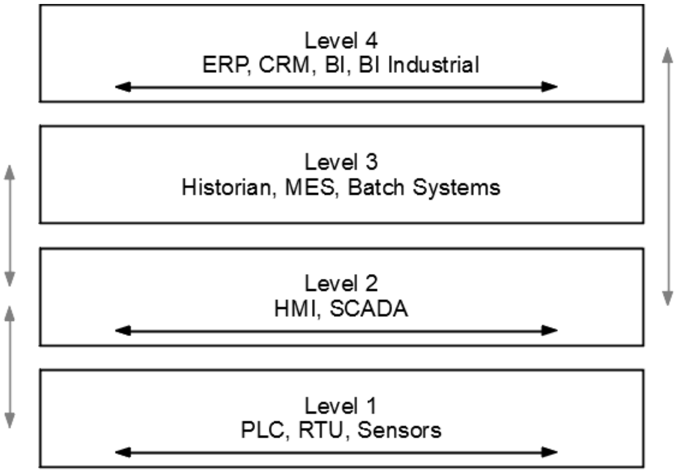


Figure 4.9: Flows of *Communication* within the ICS hierarchy according to the *ISA95-Standard* ([ISA10])

which presents the flow of *Communication* within the ICS hierarchy according to the ISA95-Standard ([ISA10]).

These flows consist of two distinct classes of *Communication* flows. One class denotes a *Communication* within a given level (intra-level). The other class represents *Communication* between various levels (inter-level). These two classes are described in [ENI17] as:

> - *„**Horizontal communications**: all the data exchanges that occur between devices and systems located within the same level.*
> - ***Vertical Communications**: all the data exchanges that occur between devices and systems that are located in different levels.“*

Such a behavior is common for the system hierarchies used in ICS. While the hierarchies presented here are widely-used but can still be considered exemplary. A similar systems hierarchy is described in [NIS15]. Aside from the specific hierarchy used, *Communication* in ICS follows a hierarchy consisting of segregated networks.

4.2.2.2 Industrial Control Systems Protocols

After the standardized nature of communication flows in ICS is demonstrated it is necessary to analyze what protocols and media are used to enable this *Communication*. These considerations are focused on the **Manufacturing Zone** in PERA which represents the ICS-specific network.

These protocols are referred to as *fieldbus protocols* during the course of this thesis, employing the definition provided by [KL15]:

> *„Fieldbus protocols [..] are commonly deployed to connect process-connected devices (e.g. sensors) to basic control devices (e.g. programmable logic controller or PLC), and control devices to supervisory systems (e.g. ICS server, human-machine interface or HMI, historian).“*

This definition aligns well with the tasks performed within the **Manufacturing Zone** in PERA.

Since PLCs are the basic controller devices, considering their interfaces is a valid step to obtain an overview on the protocols used in the ICS domains. This overview on the interfaces included in PLCs is provided in Table 4.2 which is included in Section 4.1.2. The **ICS Processing Unit Examples** utilize PROFINET ([Prob]), PROFIBUS ([Proa]) or Ethernet (as discussed in Section 4.2.1). These protocols are indeed amongst the most common ones, but there exists a long list of protocols used as fieldbus protocols. The following list provides an overview on these protocols by listing the *Communication Profile Families* from [ICE19] (the industrial standard for fieldbus in ICS):

- FOUNDATION Fieldbus (FF)

- CIP (Common Industrial Protocol)
- PROFIBUS and PROFINET
- P-NET
- WorldFIP
- INTERBUS
- SwiftNet
- CC-Link
- HART
- VNET/IP
- TCnet
- EtherCAT
- Ethernet POWERLINK
- EPA (Ethernet for Plant Automation)
- Modbus
- SERCOS
- RAPIEnet
- SafetyNet p
- MECHATROLINK

This list can further be expanded by the protocol examples provided in [ENI17] for *level one* and *level two* of the ISA95-Standard ([ISA10]). However, the list is truncated to only include the fieldbus protocols relevant for this thesis:

- CANopen
- IEEE 802.15.4 + ZigBEE (ECC)
- S7 Communication (Siemens)
- OPC

These listings exemplify the wide variety of protocols used within the ICS domain. Some of the protocols listed are highly vendor-specific and are only implemented in the components of the specific vendor. Other protocols are legacy and rarely used today. The selection of protocols taken into consideration during this thesis is dictated by the availability of components able to use the specific protocols and the relevance of the specific protocols. The **ICS Processing Unit Examples** presented in Section 4.1.2 present vendors with an extensive market share. So it is reasonable to assume that the protocols used by these vendors are also the most relevant.

4.2.2.3 Availability of *Data Types* in *Communication* in Industrial Control Systems

An understanding of the availability of the specific *Data Types* present in the *Communication* in ICS environments can be obtained by reviewing the description of the horizontal and vertical *Communication* in the ISA95-Standard ([ISA10]) as provided by [ENI17]:

- *„4.* ***Within level one****: numerical values are commonly exchanged between field devices as sensors, PLCs, or RTUs, among others. The information is exchanged among devices within the same level, but only the one that acts as the master can command the others.*
- ***5. Within level two****: the interchanged information or actions acquired or sent by SCADA systems are notified to the HMI system for a more understandable visualization on the part of the user. These communications must be realized in real time.*
- ***6. Within level four****: standard IT communications between the CRM and ERP systems (among others) in order to exchange needed information for customer-related, invoice and billing processes. If BI systems are in place, additional interactions may be created to fulfill the needs of the information.“*

This short description of the data flows within the specific levels shows that the *Communication* within *level one* and *level two* of an ICS hierarchy implementing the ISA95-Standard is specific to ICS. *level four* is concerned with processes taking place in the domain of Desktop IT and already discussed there (see Section 4.2.1). *level one* and *level two* consists mainly of *Communication* required to perform the physical process enacted by the ICS. In the current definition of the *Data Types*, this would represent **DT8**. The physical representation of this data would be considered **DT2**. Protocols are employed in order to facilitate this transfer. Hence, the inclusion of **DT3** is also necessary.

[ENI17] adds the following description of the specific vertical *Communication*:

- *„****1. Between level one and two (bi-directional)****: exchange of information between the sensors, or field devices, and the systems in charge of interpreting and processing the readings of these devices. Output is usually numerical, controlling operational functions (e.g. closing valves).*
- ***2. Between level two and three (bi-directional)****: information exchange between supervisory and operation management systems; the interpreted information (originated from level one and processed in level 2) is communicated to the higher level systems to register (Data Historian), verify (MES) and transfer to other processes (Batch) if needed.*
- ***3. Between level two and four****: information exchange between operation management systems and the ERP or BI systems, regarding mostly the operational status, progress and evolution as to aid on manufacture planning or resource needs.“*

Again, these data flows are centered on the underlying physical process and shows how the information regarding this process flows between the different levels. The *Communication* between *level two* and *level four* once again shows the close relationship between the ICS and the Desktop IT domain. The 'industrial objective' has to rely on support provided by tasks performed within the Desktop IT domain. Examples would include the ordering of raw materials or the scheduling of maintenance. In order to perform these tasks, the systems in the Desktop IT domain need some information about the state of the physical process like the usage of raw material or the productive output.

These *Data Types* are the ones relevant for the normal operation of the ICS. However, the specific components are also potentially programmed or configured via network interfaces. This implies the presence of **DT1** to provide information to the configuration interface in the *Communication*. **DT4**, **DT5** are needed in order to perform a meaningful operation of the ICS. Supervisory systems might also collect information about the state of the PLC. This would include **DT6**. In essence, the majority of data found in the *Communication* of ICS is **DT8** as it represents information about the physical process.

A further exploration on the forensic data available in ICS components performed in [**HAK+17**] confirms this prevalence of **DT8** in the *Communication* within the ICS domains.

Since **DT8** is of reduced use during a forensic investigation performed in **IC1**, this lack of other available data is detrimental to any investigation performed in **DT8**. This fact can also be found in [FC08]:

> *„Due to the unique and uncommon nature of control systems technologies, there is often inadequate information collected from these countermeasures following a cyber attack or incident."*

4.2.2.4 Gathering and Investigating the *Communication* in Industrial Control Systems

This section discusses how the *Communication* in ICS can be accessed and interpreted during a forensic investigation. It follows the same steps as used in the Section 4.2.1.2 to obtain access and investigate the *Communication* in the Desktop IT domain.

At first, physical access to the carrier is necessary. Then, the *Communication* can be recorded. Again, the placement of the recording locations dictates which network traffic is available for recording and is hence discussed in this section. After the network traffic is recorded (**DG**), it can be investigated (**DI**).

Accessing the Carrier Medium in Cable-Based Ethernet Networks and Performing DG)

The ability to access the *Communication* in the ICS domain is highly dependent on which protocols and interfaces are used.

A number of protocols can use (or generally use) Ethernet for transport. If this is the case, **DG** can be performed exactly the way as in the Desktop IT domain (see Section 4.2.1.2). It also carries the same implications in regards of the selection of capturing locations, physical access to the medium and the tools available to record the data. In this case, well-tested approaches from this more mature domain can be applied.

There are different approaches that aim to establish a permanent physical access to the carrier medium in Ethernet-based ICS infrastructures in order to support forensic investigations:

- As discussed in Section 2.3.3, [KGC+06] suggests establishing additional logging mechanisms, referred to as Forensic Agents, into an Ethernet-based ICS infrastructure in order to record various parts of the *Communication* better suited to investigate a potential IT-security incident within the ICS. The Forensic Agents are supposed to be attached to Level 0, Level 1, and Level 2 of PERA (see Section 2.2.2.1). Hence, the Forensic Agents would have access to the network traffic within the ICS-specific part of the overall network. In essence, these Forensic Agents represent network taps. [KGC+06] also suggests an architecture to perform the necessary storing of the gathered data. The storage mechanisms proposed however do not address integrity and authenticity of the gathered data. This could be added by implementing the guidance provided in Section D.4.

- Indeed data from the *Communication* is already recorded in a notable amount of ICS architectures. As discussed in Section 2.3.3, Historians are tasked with archiving information about the physical process. Hence, they aim to preserve **DT8** (data documenting the physical process). This data is useful to review the performance of the process controlled by the ICS and to optimize this process. Usually, a Process Historian is tasked with storing this data for relatively short amount of time while a Plant Historian stores this data for the entire lifetime of a plant. These Historians might, depending on implementation, use certain methods to ensure the integrity and authenticity of the data. This might even go so far as to satisfy forensic requirements though this is generally not the case (see [WDJC13]). In any case, they do not address anything besides the physical process. Hence all the other *Data Types* are not recorded by these Historians which might prove problematic during a forensic investigation.

- The inclusion of another system designed to gather data from the *Communication* in order to benefit forensic investigations and incident response was proposed in [**AHNH20**]. This approach addresses the need to gather data from varying *Data Types* in order to support a forensic investigation and how to support the forensic process by maintaining authenticity and integrity of said data. As a foundation, it discusses the different *Data Types* available in the *Communication* of ICS. It relies on a simpler model of only two different categories of data. These two categories are described as *Data about the physical process* and *data about the control components.* [**AHNH20**] provides the following definition for these:

  > *„Data about the physical process contains set points, sensor readings and the state of actors.*
  >
  > *Data about the control components contains data which is similarly found in IT (such as log messages from components, firewall and switch logs) and new traces including industrial communication protocols (Industrial Ethernet-Based: Modbus/TCP, OPC UA, OPC DA, S7comm, ..; Field Bus-Based: Modbus, Profibus, CAN, ..)“*

> *Data about the physical process* hence describes **DT8**, while *data about the control components* covers any other *Data Types*. Since the *Data Types* are designed to categorize data on how it is handled in the forensic process, this simplified model is entirely reasonable for this reduced scope. *Data about the physical process* is interpreted differently from *data about the control components*.
>
> This paper discusses the requirements to store the gathered data by using cryptographic means in order to ensure integrity, authenticity and confidentiality.

Accessing the Carrier Medium in Other Cable-Based Networks and Performing DG

The topic of gaining physical access to the carrier medium is a lot more problematic when protocols do not rely on Ethernet but instead use other media for transport. Here, vendor-specific solutions to access these media are required. This is, in fact, often unwanted or even strictly prohibited by the specific vendors. One such example is *PBMaster* ([TPS09]) which is discussed in [**ALK+18**]. This open software implementation of Profibus DB ([Proa]) was designed for the use with a hardware-converter to make the Profibus network traffic accessible to other devices. Due to patent violations, distribution of the software is prohibited and limited to members of the Profibus International Organization and therefore not usable for an independent forensic investigator.

Accessing the Carrier Medium in Wireless Networks and Performing DG

Using wireless communication inside ICS is becoming a topic of increasing relevance as shown by publications like [LLW+15] and [SPH20]. What is coined as *Industry 4.0* (see [WEG19]) might include Industrial Wireless Sensor and Actuator Networks (IWSAN). However, such developments are far from standard and the diversity of scenarios in which ICS are used makes it doubtful that wireless networks will be used in all ICS environments (see Section 4.3.1). However, accessing wireless networks might be necessary during a forensic investigation into ICS.

An overview of different common wireless communication technologies used in ICS is provided by [LLW+15]. This study identifies the following wireless communication technologies as relevant for ICS environments:

- Wi-Fi
- ZigBee
- Bluetooth
- RFID

In general, physical access to the medium is easier when compared to wired connections since the media used for transmission is air. However, a receiver able to access the specific frequency is necessary.

Wi-Fi (WLAN) is also used within the Desktop IT. Hence, the use of the same tools as when applied to access WLANs in Desktop IT is possible (see Section 4.2.1.2). The other technologies require dedicated interfaces in order to access them.

ZigBee (see [ZB15]) is an open standard designed to facilitate *Communication* between components of IoT (Internet-of-Things). The standard can also be used in industrial environments, for example to attach sensor networks. Since this standard is open, solutions to access a ZigBee network are available. One example is the use of a *CC2531 ZigBee USB-Transceiver* with the *zigbee2mqtt*[66]-firmware.

Bluetooth is another standardized protocol (see [BT20]). It is designed to facilitate transmissions of data over comparatively short distances. As a protocol that is widely used there is a range of various receivers available. There is a distinction between various editions of BlueTooth in terms of the required transceivers. There are also dedicated receives available which have no ability to send BlueTooth signal which prevents a *Structural Impact* on the *Communication*. One such example is the *Adafruit Bluefruit LE Sniffer*[67] which can be used to passively access BlueTooth Low Energy *Communication*[68].

RFID is a technology mostly used to enable a transceiver to identify physical objects in their proximity based on specific senders (see [RFI08]). These senders are usually referred to as RFID tags. In industrial applications this might be used when a system is tasked with sorting or transporting physical objects. Here, this ICS could employ a RFID transceiver as a sensor which identifies the various physical objects to be sorted. RFID would generally not be used for *Communication* within an ICS. However, RFID sniffers are readily available. An example is *RFIDuino*[69].

In general, all these standardized protocols not entirely specific to the ICS domain can be accessed quite easily with existing solutions. Further technical possibilities to access various transport media used in the ICS and Automotive IT domain are discussed in [Lam19], which is only available in German and deals with identifying vehicles based on their wireless interfaces.

Two protocols not mentioned in [LLW$^+$15] deserve some attention. These are WirelessHART and IWLAN.

WirelessHART is the wireless version of HART (see [HAR13]). WirelessHART uses a mesh-network in which every attached device also acts as router. This makes the architecture less predictable than in other ICS scenarios which follow a very strict architecture to facilitate *Communication* between the various components. [MMW$^+$14] identifies the use of 15 different channels as another problem for recording WirelessHART network traffic. However, [MMW$^+$14] provides an architecture for a device able to capture this traffic. A device to capture WirelessHART is commercially available as *open-sniffer*[70].

IWLAN (Industrial WLAN)[71] is employed by Siemens SIMATIC. The technical documentation of the devices using this IWLAN[72] reveal that IWLAN is in fact employing regular WLAN (specifically *IEEE 802.11ac Wave 2*). Hence, this *Communication* can be accessed and recorded in the same manner as regular WLAN (see Section 4.2.1.2).

[66] https://www.zigbee2mqtt.io/getting_started/flashing_the_cc2531.html, 21/05/2020

[67] https://www.adafruit.com/product/2269, 21/05/2020

[68] https://learn.adafruit.com/introducing-the-adafruit-bluefruit-le-sniffer, 21/05/2020

[69] https://github.com/marcboon/RFIDuino, 21/05/2020

[70] https://www.sewio.net/open-sniffer/, 21/05/2020

[71] https://new.siemens.com/global/en/products/automation/industrial-communication/industrial-wireless-lan.html, 21/05/2020

[72] https://new.siemens.com/global/en/products/automation/industrial-communication/industrial-wireless-lan/scalance-w1750d-direct-access-point.html, 21/05/2020

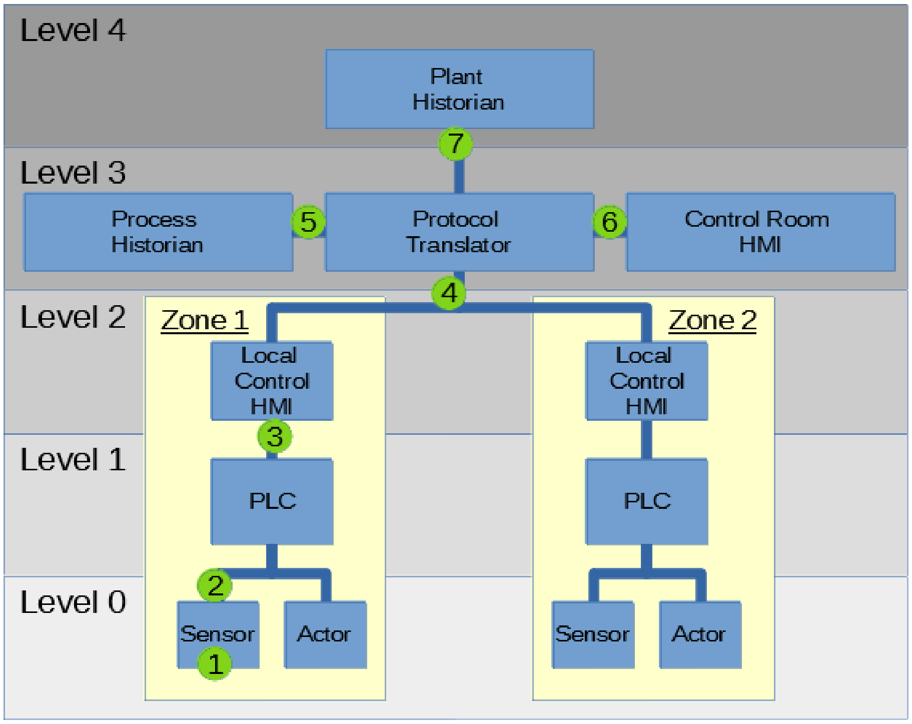


Figure 4.10: Different representations of the same physical value inside the hierarchy of an Industrial Control System

Placement of Means to Obtain Access to the Carrier Medium in ICS Architectures

The placement of the capturing locations is an important topic where segregated networks are used. This is the case in ICS as with the more complex networks discussed during the excerpt on Desktop IT (see Section 4.2.1). This placement is obviously also a critical factor for approaches proposed to enable access to the *Communication* within Ethernet-based ICS architectures in [KGC+06] and [**AHNH20**].

Indeed, the placement of means to get physical access might be easier, since ICS network hierarchies are usually more standardized and simple. However, it has to be noted that the hierarchical nature of ICS communication can lead to a propagation of a given piece of data over various infrastructure elements.

A simple infrastructure to visualize this fact and its consequences is presented in Figure 4.10. This simple infrastructure follows the PERA hierarchy. It contains two Zones each with a Local Control HMIs (Level 2), PLC (Level 1), a sensor (Level 0) and an actor (Level 0). The entire site is controlled from a Control Room HMI (Level 3). Other side-wide systems (Level 3) consist of a Process Historian and a Protocol Translator. The data about the physical process (**DT8**) delivered by multiple sites would then be aggregated and stored in a Plant Historian (Level 4).

In this infrastructure, a digital sensor is measuring a physical property at **Position 1**. This physical property is translated into a digital value represented by a specific representation based on the data format used. This physical representation is then send to the attached PLC using the medium marked as **Position 2**. The PLC then processes the sensor input and transmit a resulting control signal to the attached actor. The sensor input is also forwarded to the Local Control HMI via **Position 3**. The Local Control HMI performs local (zone-wide) control functions. A physical representation of the physical measurement performed at the sensor is then send forward to the Protocol Translator via **Position 4**. Such a Protocol Translator is included here since ICS could contain various technologies to implement the different zones. Some of these technologies could be legacy or incompatible. Hence, the Protocol Translator is tasked with converting between different protocols and formats. From the Protocol Translator a representation of the physical measurement is send to the Process Historian (via **Position 5**), the Control Room HMI (via **Position 6**) and the Plant Historian (via **Position 7**).

There is no guarantee that the physical representation of the measure value does represent the value correctly. Some of this is even the case in a scenario devoid of any malicious attacker. There will be some loss during the conversion of the analog physical measurement to a digital value in any case. This happens at **Position 1** and can be mitigated by employing multiple sensors and the PLC performing some majority void or any other correcting method on these values. Another alteration of the value representing the physical measurement might happen in the Protocol Translator due to conversion processes. This can also happen because, for example, the PLC and the Local Control HMI use different formats to represent numbers and a conversion is necessary. In general, all this can be attributed to a loss due to conversion inaccuracies. However, if an attacker manipulates the transmitted representation at **Position 5** for example, the manipulation would not be noticed at the other parts of the architecture. Notably it would have no direct effect on Zone 1. Since there are multiple representations of the same physical measurements present within this architecture, it is necessary to add information about the exact recording location to any recorded value. While this is also the case in any other domain, it is more important to stress this point in the ICS domain since exactly the same data is supposed to be available in different parts of the network. The truth is, that a manipulation performed at **Position 4** will be visible in recordings made at different locations than one performed at **Position 2**.

It seems helpful to record said **DT8** at multiple locations in the infrastructure. If the data about the physical values is differing across those different locations, there is a transmission error or an attack.

Investigating the Captured Network Traffic (DI)

The success of a **DI** into ICS *Communication* is highly affected by the protocols employed within the specific ICS. Based on the nature of the protocol, tools able to investigate already captured network traffic might be available. One such tool is *Wireshark*[73]. This open source tool can also be used to analyze ICS-specific network traffic. It supports a range of protocols out of the box or using specific modules which are openly available. These protocols include Profinet (see [Prob]), OPC UA[74] and s7comm[75] (the SIEMENS SIMATIC-specific protocol - requires *s7commwireshark*[76]).

[73] https://www.wireshark.org/, 16/05/2020
[74] https://opcfoundation.org/, 16/05/2020
[75] http://gmiru.com/article/s7comm/, 16/05/2020
[76] https://sourceforge.net/projects/s7commwireshark/, 16/05/2020

Table 4.8: *Data Types* available in the *Communication* in ICS

Data Type	*Communication*
DT1	
DT2	✓
DT3	✓
DT4	
DT5	
DT6	
DT7	
DT8	✓

As discussed during other occurrences of *Wireshark* during this thesis the software is not designed with forensic use in mind. See Section 4.2.1.2 for further discussion and advice for mitigation.

Once the network traffic can be interpreted it can be understood. The lessons from the Desktop IT domain as discussed in Section 4.2.1.2 apply here as well. An understanding of the usual communication is highly useful for the analysis. In ICS environments obtaining such knowledge has some advantage when compared to the Desktop IT domain. While the behavior of the people using the systems in Desktop IT domain is rather unpredictable this is not the case in the ICS domain. Many tasks are predictable and follow a cyclic pattern. Especially since the majority of *Communication* in ICS pertains to **DT8** (as discussed in Section 4.2.2.3). In this domain, this pertains to data about the physical process. Sensor readings are performed periodically and the physical process will usually take place in a well-defined manner. This makes the resulting *Communication* better predictable and benefits the detection of unusual behavior. However, this is greatly benefited by obtaining the base line behavior of the network during **SP** in order to gain a better understanding of which *Communication* behavior can be considered normal. Also, some knowledge about the underlying physical process is of use to better understand what the transmitted **DT8** implies about the state of the physical process. This might especially be of interest if an attacker is using the ICS to disturb the physical process. Such an occurrence would currently not be covered by either of the *Investigative Contexts* but is entirely possible. This circumstance is discussed in Section 5.3.5.1.

4.2.2.5 Availability and Access to the Forensic *Data Streams* of *Communication* in Industrial Control Systems

The *Communication* in ICS is organized along the lines of hierarchies. These hierarchies and the resulting flow of *Communication* follow established patterns as described by PERA (Purdue Enterprise Reference Architecture - see [Wil92]) or the ISA95-Standard ([ISA10]).

In essence, the *Communication* can contain all *Data Types* - due to the fact that the PLC can be programmed and configured using network interfaces. However, in the usual operation, the bulk of data transmitted belongs to **DT8** since this data describes the underlying physical process. This data has a raw representation (**DT2**) and uses protocols to be transmitted (implying **DT3**). An overview is provided in Table 4.8.

The physical access to the carrier medium can be handled in the same way as in the Desktop IT domain (see Section 4.2.1.2). Some vendor-specific carriers might be used which require specialized hardware.

Once the network traffic is captured, it can be analyzed in the same manner as as in the Desktop IT domain (see Section 4.2.1.2). This depends however on the specific protocols used. Dissectors, which interpret these protocols, are available for a range of various protocols. However, there exist a broad variety of protocols, some of them legacy, which are far more difficult to access.

In general, analyzing the *Communication* between ICS components benefits from the advantage of a more orderly communication behavior of the respective components when compared to the Desktop IT domain. However, an understanding of the underlying physical processes might be helpful in order to interpret **DT8** during a forensic investigation.

4.2.3 Automotive IT Architectures

The overview provided on ECUs (Electronic Control Units) in Table 4.5 showed that CAN is the dominant communication interface in the Automotive IT domain. However, different other technologies are used for different functions of the network.

At first, a general overview on the overall topology of Automotive IT is given, before a more detailed look on the overall hierarchy is provided before specific protocols are introduced. Then the availability of the various *Data Types* within these networks is discussed before the access to these specific carriers is analyzed. Afterwards, the possibility to perform **DI** on the gathered *Communication* is given before this section is concluded with a summary.

4.2.3.1 Automotive IT Hierarchy

In general, Automotive IT consists of a communication network which ties the various ECUs together. This network is usually split into various subnetworks. This is made necessary by the amount of ECUs in contemporaneous in cars. If the bandwidth of a given carrier is not high enough to ensure soft real-time functionality it is necessary to split the network into subnetworks. This split usually occurs along the line of functionality and various manufacturers have varying naming conventions for the resulting subnetworks.

For example, one subnetwork might connect all the ECUs responsible for comfort functionality or to control the powertrain and the engine. The exact amount of these subnetworks might vary but three to five is a representative amount ([Cur20]). For example, the car investigated in [**KAH+18**][77] contains five of these subnetworks.

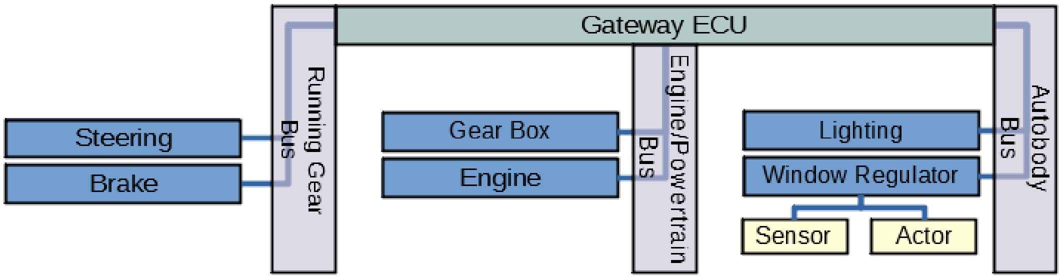


Figure 4.11: Example for a network in Automotive IT including three subnetworks

[77] VW Golf Mk7 1.2 TSI

These subnetworks are connected together via a Gateway ECU. For all practical purposes, this Gateway ECU can be considered as a router. Figure 4.11 shows an example for a network in Automotive IT. This example includes three subnetworks which are connected by the Gateway ECU. These three subnetworks handle various tasks during the operation of the vehicle. Usually, these networks would be realized using CAN (Controller Area Network). An infotainment network (with other requirements in terms of bandwidth and robustness) could use another network technology like MOST (Multimedia Oriented Systems Transport).

The visualization also shows a sensor and an actor directly attached to the Window Regulator ECU. This could be considered a subsubnetwork and would usually employ yet different network technologies, like LIN-Bus (Local Interconnect Network).

These three network technologies will be discussed during the course of this section.

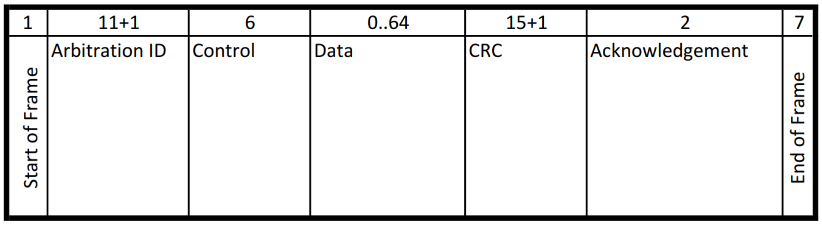
4.2.3.2 Automotive IT Protocols

This section gives a short overview on the carrier technologies and protocols used in Automotive IT in order to provide a foundation for the discussion on how these various technologies can be accessed in the scope of a forensic investigation.

CAN

The CAN bus (Controller Area Network) is a widespread bus system within Automotive IT and hence forms a foundation for understanding *Communication* within Automotive IT. This is even more relevant since the CAN bus has a set of properties which influence the forensic process. The CAN bus is standardized in [CAN91] and [ISO15a].

According to [Wal06] (which is only available in German but gives a comprehensive introduction to Automotive IT), the CAN bus is a network with a linear topology. Here, the control units are connected to a common transmission line via a stub.



1	11+1	6	0..64	15+1	2	7
Start of Frame	Arbitration ID	Control	Data	CRC	Acknowledgement	End of Frame

Figure 4.12: CAN Frame based on [CAN91]

The basic structure of a CAN frame is shown in Figure 4.12. In addition to the transmitted data (*Data*) the exchanged messages also contain an arbitration field (*Arbitration ID*) which contains the CAN ID. The CAN ID primarily provides information about the type of message. Since certain messages should only be sent by certain control units the CAN ID also implies a given sender for specific messages. The same applies to the recipient of the message - in principle every ECU connected to the data bus receives all messages and then uses the CAN ID to decide whether the message is relevant for the specific ECU. With complete knowledge of all the CAN IDs possible in a system, it is therefore possible, without containing explicit sender or receiver addresses, to implicitly derive the sender and (desired) recipient of all messages.

Since the control units of the CAN bus have equal access privileges this is called a multi-master architecture. In such architecture there is no central control device that regulates the time access to the bus. This task is distributed among the ECUs. An arbitration procedure is used in the CAN bus for this purpose. This process is based on the CAN IDs and organizes the sending access to the shared medium. The arbitration procedure ensures that messages with a low CAN ID have priority. If two control units start to transmit at the same time, the control unit that sends the message with the higher CAN ID will not be able to successfully carry out the transmission process. The ECU determines that a message with CAN ID is being transmitted in parallel and that this message has a higher priority and stops the process.

This entire concept works reasonably well if all devices attached to the BUS are honest and follow the specific rules of arbitration. The most important considerations following on these specifications are summarized in [Smi16]:

> *„Because CAN bus packets are broadcast, all controllers on the same network see every packet, kind of like UDP on Ethernet networks. The packets don't carry information about which controller (or attacker) sent what. Because any device can see and transmit packets, it's trivial for any device on the bus to simulate any other device.“*

Depending on the specific implementation, the CAN bus offers a bandwidth of 250K/500K or 1MBit per Second. However, there is a certain overhead due to the used protocols[78]. Also, the non-deterministic arbitration used requires a certain amount of spare bandwidth to fulfill real-time requirements associated with the driving task.

MOST

MOST (Multimedia Oriented Systems Transport - see [Grz11]) is usually used for multimedia applications. Usually, the MOST network forms a ring of various connected devices ([Wal06]). One of these devices acts as a MOST master responsible for giving a timing signal. MOST consists of a control channel and various data channels. The control channel is used to setup the data channels. The data channels are used to stream multimedia content. It is notable that MOST uses fiber-optic cables as a physical carrier.

LIN

According to [Wal06], the LIN bus (Local Interconnect Network - see [LIN10]) is a linear bus system that is primarily used to connect control units, sensors and actuators in a limited space. These spatially restricted LIN clusters are then connected to higher-level CAN networks (as presented in Figure 4.11). The primary difference to the CAN bus is the use of a single-master-multiple-slave architecture. This means that the master requests the attached actuators and sensors connected to the network to send data.

[78] https://copperhilltech.com/blog/sae-j1939-bandwidth-busload-and-message-frame-frequency/, 25/05/2020

FlexRay

Although not mentioned before, FlexRay (see [ISO19] and [ISO13b]) has some interesting properties which would impact a forensic process. An example of an infrastructure using FlexRay is provided in Figure 4.13.

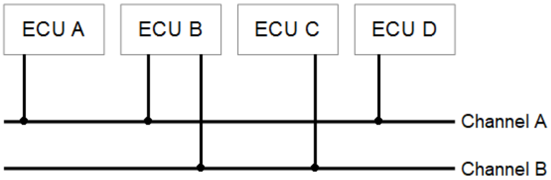


Figure 4.13: Architecture used in FlexRay based on [Wal06]

It is notable that FlexRay uses two channels (*Channel A* and *Channel B*), whereby not every component is connected to every channel. It is possible that a component is only connected on one of the two channels or on both. The two channels can be used to transmit the identical data twice in order to provide protection against transmission errors. Physical access to both channels is necessary to ensure access to all *Communication* within a FlexRay Bus.

Direct Cable Connections

Inside individual functional units there are sometimes direct cable connections between the individual ECUs, actuators and sensors. Although the LIN bus is intended to take over such wiring, these very specialized solutions still occur. Such wiring has in common that the data present in these networks is not transferred directly to other networks - an ECU may transmit post-processed data that it receives via direct cable connection from connected sensors to a bus system, however.

4.2.3.3 Availability of *Data Types* in *Communication* in Automotive IT

Networks in Automotive IT are used to transfer data necessary for the driving task. This can be considered **DT8**. Furthermore, the MOST bus is specifically designed to transfer multimedia contents like maps or music. This is also considered as **DT8**. Diagnostic data can be transferred using the various communication bus systems within Automotive IT (see Section 4.1.3.3 and Section 4.1.3.5). This diagnostic data contains **DT1**, **DT4** and **DT5**. One could consider the potential presence of a diagnostic session as **DT7**. In addition, all this data has a raw physical representation (**DT2**).

4.2.3.4 Gathering and Investigating the *Communication* on the CAN bus in Automotive IT

Again, performing **DG** and **DI** relies on gaining physical access to the medium first. Gaining this physical access relies on the employed carrier technology. Due to the abundance of CAN in Automotive IT, the abundance of available tools to access and interpret CAN traffic and the implications of CAN for the gathering of *Non-volatile Memory* and *Volatile Memory* by employing diagnostic requests (see Section 4.1.3.3 and Section 4.1.3.5), CAN represents the major part of this section. Hence, CAN will be discussed in its entirety before other carriers and protocols are discussed.

Obtaining Physical Access to CAN

Obtaining physical access to a CAN bus might be considered as the best researched topic in the security of Automotive IT. Various approaches are used in different publications, including [**KAH$^+$18**], [Smi16] and [Cor16]. Due to the simplicity of the CAN protocol and the used carrier there are various solutions in order to gain physical access to the CAN bus:

- CANtact[79]
  CANtact is a simple USB-to-CAN converter. It consists of open hardware and open source software. It is possible to acquire pre-produced CANTact boards but the open specifications (including circuit diagrams) can also be used to build the hardware independently. The open source nature of this solution makes it easy for the examiner to comprehend the operations performed.
- Beaglebone RS485 CAN CAPE[80]
  Beaglebone RS485 CAN CAPE is an additional module for the Beaglebone microprocessor. It thus allows physical access to the CAN bus.
- PiCAN2
  This hardware module is an extension for the Rasperry Pi microcomputer. It creates a physical interface to the CAN bus. During the tests with this interface it was notable that after sending data with this device receiving data did not work reliably anymore. A loss of data is to be expected here and accordingly the suitability for forensic use must be regarded as very low.

Beside these simple, partly open source solutions there exist different dedicated commercial solutions. A solution which is not able to send any data to the CAN bus would support the forensic investigation by ensuring no *Structural Impact* on the *Communication*. In essence, all these solutions could be considered as network taps.

After Obtaining Physical Access (DG)

After physical access to the specific carrier is obtained the *Communication* has to be captured. There exist a wide variety of tools in order to perform packet capturing. The following list will give a brief overview on some potential tools usable during **DG** and extends the survey performed in [**ALKD17**]:

- can-utils[81]
  can-utils is a collection of software tools for low-level access to the CAN bus. This simple implementation is executed from the command line. *can-utils* can be used to record or generate CAN traffic. The tool *cansniffer* included in this collection is able to passively record the CAN traffic and offers various filter options. The *can-utils* can use different means to gain access to the carrier medium, like *CANtact*, *Beaglebone RS485 CAN CAPE*, and *PiCAN2*. This was tested during the extensive research work done by

[79] http://linklayer.github.io/cantact/, 21/05/2020
[80] http://www.waveshare.com/rs485-can-cape.htm/, 21/05/2020
[81] https://github.com/linux-can/can-utils, 22/05/2020

the author of this thesis, including the research performed in [**LAD17**], [**ALKD17**], [**KAH$^+$18**] and [**ALK$^+$18**].

The *can-utils* are open source, which enables the investigator to easily understand the operations performed. Also, the command line interface of this collection allows for an easier *Process Accompanying Documentation* since it requires only the recording of inputs and outputs to the specific command line.

- Octane CAN bus sniffer[82]
  The *Octane CAN bus sniffer* ([OCT14]) can be used during **DG** to capture network traffic in a CAN network (see [BEM13]). *Octane CAN bus sniffer* is able to use various CAN interfaces like *CANtact*, *Beaglebone RS485 CAN CAPE*, and *PiCAN2*. This tool is open source but uses a GUI, which makes a *Process Accompanying Documentation* of the performed actions more difficult.

  Currently, the main website of the project[83] is not available and *Octane CAN bus sniffer* is potentially not maintained anymore.

- Python CAN[84]
  Python-CAN is a Python library for conveniently sending and receiving CAN data. The library can be used on any system capable of running Python. The library allows the simple implementation of a basic CAN sniffer. A hardware interface is also required for this software, but the hardware interfaces presented during this thesis in the paragraph about gaining physical access to a CAN carrier work well. This library is open source.

- Kayak[85]
  Kayak uses TCP/IP over SocketCANd as an additional abstraction layer for the access to the CAN bus. This abstraction layer enables multiple users to access the bus system at the same time. *Kayak* has several options for logging, saving or generating CAN frames. Kayak is GUI-based and open source.

- SavvyCAN[86]
  SaavyCAN is another open source tool which is able to collect network traffic on the CAN bus. This tool utilizes a GUI.

- BUSMASTER[87]
  BUSMASTER is an open source tool for simulating, testing and analyzing traffic on communication bus systems within Automotive IT. This also includes the functionality of capturing and storing network traffic on the corresponding media. Accordingly, this tool can be used during **DG**.

- Wireshark[88]
  Wireshark is also usable for recording network traffic in Automotive IT. *Wireshark* is open source and GUI-based. *tshark*[89] is the included command line version of *Wireshark*.

[82] https://web.archive.org/web/20180324085447/http://octane.gmu.edu/, 22/05/2020
[83] http://octane.gmu.edu/, 22/05/2020
[84] https://github.com/rberkow/python-can, 22/05/2020
[85] https://dschanoeh.github.io/Kayak/, 22/05/2020
[86] https://www.savvycan.com/, 22/05/2020
[87] https://rbei-etas.github.io/busmaster/, 22/05/2020
[88] https://www.wireshark.org/, 22/05/2020
[89] https://www.wireshark.org/docs/man-pages/tshark.html, 16/05/2020

This short list of potential tools to record network traffic in CAN networks exemplify that the logical access to the CAN bus can be implemented quite easily. The fundamental requirement is physical access to the carrier.

The suitability of the tools presented here for the forensic process is limited, however. Although during tests with the specific tools in a laboratory setup (see [**KAH+18**]) no data loss was detected none of the tools offer mechanisms to protect the authenticity and integrity of the collected network traffic. Correspondingly, external mechanisms must be used here to, for example, ensure the integrity through hash values.

Also, many of the tools presented here have not only the ability to capture network traffic but also to create network traffic. During the tests, no instances have been observed where a tool send messages to the CAN bus not requested by the user (and hence having a *Structural Impact* on the *Communication*). However, removal of the send features would be useful to adapt said tools to forensic use in order to exclude such effects entirely.

Furthermore none of the tools provided a protocol of the actions performed which makes a *Process Accompanying Documentation* of the performed actions more difficult. For the command line based tools (*can-utils* and *tshark*) this can be alleviated by the use of external methods to capture the command line.

In general, the presented tools are all open source which makes an adaptation to dedicated use in forensic scenario by implementing some or all of the suggestions provided in Section D.4 possible.

Placement of Means to Obtain Access to the Carrier Medium in Automotive IT Architectures

Again, the placement of the device used to access the medium is an important question. However, since a CAN communication bus utilizes broadcast for all messages, it is sufficient to attach the device anywhere at the physical carrier. Ideally, a stub can be used. If that is not the case the shielding of the CAN cable has to be opened up and the device directly attached there (see [**KAH+18**]). Safety concerns and the volatility of *Communication* would dictate to create such a physical access during a **SP**.

Investigating the Captured Network Traffic (DI)

Investigating captured *Communication* in the Automotive IT domain relies heavily on vendor-specific knowledge. There is no lack of tools in order to interpret the traffic on a CAN bus as the following list of available tools usable during **DI** will clearly demonstrate. However, without any knowledge of the specific CAN IDs the interpretation of said traffic is difficult. The CAN IDs are part of the Arbitration ID (see Figure 4.12 and Section 4.2.3.2). These CAN IDs represents the type of the message. Since only certain ECUs are supposed to send messages of specific types, the CAN ID also implies a specific sender of a given message.

These CAN IDs are vendor-specific and sometimes model-specific. The same goes for the data block transmitted within the respective CAN Frames. Here a forensic investigator who has access to this vendor-specific knowledge has an immense advantage over an independent forensic investigator. However, the topic of reverse engineering specific CAN networks in order to obtain this model-specific knowledge is a relevant topic in the Automotive IT domain. Examples for such approaches can be found in [**KAH+18**] or [Smi16].

The following list provides a brief overview on some potential tools usable during **DG** and extends the survey performed in [**ALKD17**]. It is notable that some of the tools presented there can also be used to record data. In this case, their properties which influence the forensic process are not repeated:

- Octane CAN bus sniffer[90]
  The *Octane CAN bus sniffer* ([OCT14]) is not only a pure network sniffer but an extensive tool collection to investigate CAN data traffic (see [BEM13]).
- Kayak[91]
  Kayak offers several features to identify and visualize signals transmitted via CAN.
- SavvyCAN[92]
  SaavyCAN also offers the possibility to visualize CAN frames.
- BUSMASTER[93]
  BUSMASTER can be used to investigate captured CAN traffic.
- Wireshark[94]
  Wireshark is also usable for recording network traffic in Automotive IT. *Wireshark* is open source and GUI-based. *tshark*[95] is the included command line version of *Wireshark*.
- c0f Fingerprinting Tool[96]
  This tool performs a statistical analysis of the most frequently cyclically sent CAN messages. This static analysis leads to specific fingerprints, which are stored in JSON format in a database. This analysis can be used to quickly and easily identify the vehicle type. In laboratory tests performed in the scope of this thesis this tool could also be used to efficiently check whether the ECUs connected to the given CAN bus communicate in an expected manner. In the event that an ECU fails the observable communication behavior changes. This happens very rapidly in some cases. Hence, this tool can be used to quickly check whether there were any changes or irregularities in the internal vehicle *Communication*. In this case, a fingerprint of normal operation must be generated before an incident occurs (as part of an **SP**).

An overview on the most important properties of the tools usable during **DG** and **DI** in CAN bus systems is provided in Table 4.9. These properties are focused on the evidentiary value provided by evidence gathered and processed using these tools. When no *send functionality* is included, the tool is not able to have a *Structural Impact* on the *Communication* on the respective CAN network. *Protects integrity* regards if the given tool protects the integrity of the gathered evidence in any way. *Protocols action* describes if the tool is producing a log file of the actions performed which would greatly support the *Process Accompanying Documentation*. The same concern is addressed by identifying the used user *interface*. This

[90] https://web.archive.org/web/20180324085447/http://octane.gmu.edu/, 22/05/2020
[91] https://dschanoeh.github.io/Kayak/, 22/05/2020
[92] https://www.savvycan.com/, 22/05/2020
[93] https://rbei-etas.github.io/busmaster/, 22/05/2020
[94] https://www.wireshark.org/, 22/05/2020
[95] https://www.wireshark.org/docs/man-pages/tshark.html, 16/05/2020
[96] https://github.com/zombieCraig/c0f/, 24/05/2020

Table 4.9: Overview on tools usable during **DG** and **DI** in CAN-based networks in Automotive IT

Tool						
usable in...		open source	interface	send functionality	protects integrity	protocols actions
DG	**DI**					
can-utils: cansniffer						
y	n	y	CLI	n	n	n
Octane CAN bus sniffer						
y	y	y	GUI	n	n	n
Python-CAN						
y	y	y	CLI	y	n	n
Kayak						
y	y	y	GUI	y	n	n
Octane CAN bus sniffer						
y	y	y	GUI	n	n	n
SavvyCAN						
y	y	y	GUI	n	n	n
BUSMASTER						
y	y	y	GUI	y	n	n
Wireshark						
y	y	y	GUI	n	n	n
tshark						
y	y	y	CLI	n	n	n

can be either GUI (Graphical User Interface) or CLI (Command Line Interface). To document the actions performed is easier when using a CLI through external means. Hence, CLI is generally advantageous for performing a *Process Accompanying Documentation*.

These criteria are indeed usable for any tools considered to be used during **DG** or **DI** of *Communication*.

The overview provided in Table 4.9 already shows that none of the tools is designed with forensic use in mind. However, the open source nature of the provided tools allows for the adaptation of these tools following the suggestions provided in Section D.4.

One major problem when interpreting any evidence gathered using the CAN bus is the architecture of the CAN bus. This also affects the evidence from *Non-volatile Memory* and *Volatile Memory* obtained by using diagnostic and/or calibration protocols which rely on the CAN bus for transport (see Section 4.1.3.3 and Section 4.1.3.5). Every device connected to a given CAN bus receives every message. The device can discern if the message is addressed to itself by interpreting the CAN ID. Also, every device can send on the CAN bus using any CAN ID. Hence, a malicious device could answer to any message designated to another device. This is also possible for diagnostic requests. An example for such an attack is described in the German language PHD thesis [Hop14]. Here, the author inserts a malicious device into Automotive IT which responds to diagnostic requests in order to simulate the presence of removed components. Indeed the same circumstance allows for the possibility to perform an electronic simulation of the input signal in case of *In the loop Forensics* in the first place (see Section 4.1.3.1).

In essence, it can be summarized that on the CAN bus there exists no origin authenticity in regard to specific ECUs. Any message received on any given CAN bus could have been transmitted by any device connected to the CAN bus. This has dire consequences for the evidentiary value of anything obtained from or using the CAN bus.

4.2.3.5 Gathering and Investigating the *Communication* on Other Bus Systems in Automotive IT

While the prominent CAN bus receives a notable amount of attention from security researchers and Automotive IT enthusiasts, the same is not true for other bus systems. However, an overview about their use during forensic investigations is provided here.

MOST

Indeed, no non-commercial solution could be identified to access the MOST bus. Here, vendor-specific, commercial hardware and software is required to access the MOST-bus and to investigate the recorded traffic. However, commercial adapters exists and obtainable on the regular commercial market:

- VN2640[97]
  The *VN2640* is an USB-interface for MOST150

A tool in order to gain logical access the MOST network and to analyze the corresponding traffic is provided by the same vendor:

- CANalyzer[98]
  CANalyzer is a software suite designed for the analysis of various networks in Automotive IT. The software is designed to work with interfaces provided by the same vendor. The software is closed source and uses a GUI.

The use of this software for forensic scenarios is only recommended due to the lack of alternatives. Currently, no way for an independent forensic investigator to perform **DG** and **DI** with completely traceable and disclosed methods is known to the author of this thesis.

Since the MOST bus uses fiber-optic cable this physical access would have to rely on already present spare stubs, the removal of a currently attached device or a splicing of the cable which again requires specialized equipment.

LIN

The LIN bus can be accessed by a greater selection of available tools. Various solutions to gain physical access to the carrier are available for the use during a forensic investigation:

- LIN bus adapter for CAN-Hacker 3.2[99]

[97] https://www.vector.com/int/en/products/products-a-z/hardware/network-interfaces/vn2640/, 22/05/2020

[98] https://www.vector.com/int/en/products/products-a-z/software/canalyzer/#c9463, 22/05/2020

[99] http://canhacker.com/projects/lin-bus-analyzing-tool/, 22/05/2020

- Various LIN interfaces by VECTOR[100]

- Kvaser Leaf Professional LIN[101]

As with the necessary adapters there is greater range of software usable to gain logical access to the LIN bus. Some of these tools are presented here:

- CARBUS Analyzer[102]
  This tool is designed to be used together with the *LIN bus adapter for CAN-Hacker 3.2* by the same vendor. The tool is closed source and uses a GUI.
- CANalyzer[103]
  CANalyzer is also able to record and interpret LIN network traffic. The software is designed to work with interfaces provided by the same vendor (*Various LIN interfaces by VECTOR*). The software is closed source and uses a GUI.
- BUSMASTER[104]
  BUSMASTER is an open source tool for simulating, testing and analyzing various bus technologies (CAN, CAN FD, LIN, FlexRay). This also includes the functionality of capturing and storing network traffic in these networks. Accordingly, this tool can be used during **DG** and **DI**. *BUSMASTER* uses a GUI for user interaction. The versatility of the different bus systems implemented by BUSMASTER should be emphasized.

All these tools are not designed for forensic use. The use of GUI complicates the *Process Accompanying Documentation*. No functionality to ensure the integrity and authenticity of the captured data is provided. In addition, the inner workings of the closed source applications (*CARBUS Analyzer* and *CANalyzer*) cannot be entirely relied on by an independent investigator.

Direct Cable Connections

DG is only possible directly on the respective cable connection. The specialized nature of this wiring results in a high difficulty for **DG** (and also for **DI**). The physical access to the cable connection can differ greatly depending on the carrier medium and is too obscure to be considered here in great detail.

4.2.3.6 Availability and Access to the Forensic *Data Streams* of *Communication* in Automotive IT

Networks in Automotive IT are used to transfer data necessary for the driving task. This can be considered **DT8**. Furthermore, multimedia contents like maps or music are also transferred

[100] https://www.vector.com/int/en/products/products-a-z/hardware/network-interfaces/, 22/05/2020
[101] https://www.kvaser.com/product/kvaser-leaf-professional-lin/, 22/05/2020
[102] http://canhacker.com/projects/carbus-analyzer/, 22/05/2020
[103] https://www.vector.com/int/en/products/products-a-z/software/canalyzer/#c9463, 22/05/2020
[104] https://rbei-etas.github.io/busmaster/, 22/05/2020

Table 4.10: *Data Types* available in the *Communication* in Automotive IT

Data Type	*Communication*
DT1	✓
DT2	✓
DT3	✓
DT4	✓
DT5	✓
DT6	
DT7	
DT8	✓

within Automotive IT. These are also considered as **DT8**. In addition, Diagnostic data can be transferred using the various communication bus systems within Automotive IT (see Section 4.1.3.3 and Section 4.1.3.5). This diagnostic data contains **DT1**, **DT4** and **DT5**. One could consider the potential presence of a diagnostic session as **DT7**. In addition, all this data has a raw physical representation (**DT2**) and communication protocols are used to facilitate this data transfer leading to the presence of **DT3**. An overview on this is provided in Table 4.10.

The most important bus system in Automotive IT is the CAN bus. The CAN bus covers the overall high level communication between the various ECUs. Accessing the CAN bus is well-researched. Various physical interfaces to obtain this physical access are available. There exist a great collection of tools usable during **DG** and **DI**. None of these tools is designed with forensic investigations in mind but a notable number of open source tools could be adapted for forensic use.

Due to the nature of the CAN bus origin authenticity in regard to specific ECUs cannot be assumed. Any message received on any given CAN bus could have been send by any device connected to the CAN bus. This reduces the evidentiary value of anything obtained from or using the CAN bus. This is especially dire since the diagnostic requests and the corresponding response used to query contents of *Non-volatile Memory* and *Volatile Memory* rely on the CAN bus for transfer (see Section 4.1.3.3 and Section 4.1.3.5).

For the other bus systems, the availability of interfaces and tools to perform **DG** and **DI** is far lower. Some commercial solutions exist to access the LIN Bus. None of the tools currently available for accessing the LIN Bus is designed for forensic use. However, one open source tool (*BUSMASTER*) is available for LIN bus and could be adapted to fit forensic needs.

Interfaces to capture *Communication* should be installed during **SP**. This would include the preparation of physical access with the respective devices to capture the network traffic only being attached during **DG** once it has been deemed necessary during **OP**.

A major obstacle when dealing with *Communication* in Automotive IT is the high degree of vendor-specific knowledge required in order to interpret the various messages. Here an independent forensic investigator has to rely on reverse engineering to obtain this knowledge. Examples for such approaches can be found in [**KAH+18**] or [Smi16].

4.3 *Step 1.3* Analysis of Scenarios

This section describes the contexts in which ICS and Automotive IT are used. These contexts might impact the forensic process in one way or another. This section starts with discussing ICS and then moves unto Automotive IT. A discussion of the Desktop IT domain is omitted here since it is the baseline everything else is compared to.

The way systems in the ICS and Automotive IT system are employed has some impact on forensic investigations.

Some of these factors can be found in the listing of the most important differences between the ICS and Desktop IT domain in terms of security provided in [NIS15]. While some of these differences are already covered by the specific employed components (see Section 4.1) or architectures (see Section 4.2) some originate directly form the way ICS are used. These factors have been adapted to focus on forensic investigations instead of security against malicious attacks. The factors mentioned in [NIS15] which affect forensic investigations are:

- Component Accessibility
  Where the physical components are located has an influence on the forensic process. This mostly affects the accessibility of the various components for a direct, physical access which might prove a requirement for certain means to perform a **DG** as explored in Section 4.1 and Section 4.2.
- Component Lifetime
  The lifetime of components has an influence on the forensic process since a longer lifetime implicates the use of older and more legacy components which might use outdated protocols or outdated interfaces.
- Availability (Reliability) Requirements
  Shutting down systems to perform a forensic investigation might not be possible due to the high cost of shutting down a system or the problems related with the physical process when shutting down a system. This might prevent the possibility to perform a *Post-Mortem Investigation*. On the other hand, the risk to keep a potentially compromised system active might be higher if this could lead to a catastrophic result with regard to the controlled physical process.
- Performance Requirements
  If the processing task performed by the computing units is time critical, this might impact the ability to perform a forensic investigation during *Live Forensics*. The lack of resources (computation power and bandwidth) could interfere with these real time requirements.

How the ICS and Automotive IT domain affect forensic investigations in terms of these factors is discussed in the course of this section. This discussion is structured along the lines of these three identified major factors.

4.3.1 Industrial Control Systems Scenarios

Industrial Control Systems are diverse constructs which control physical processes in order to achieve an industrial objective (see Section 2.2.2). Section 2.2.2.3 already introduced how ICS are deployed. This section discusses how the deployment of ICS affects the forensic process.

Component Accessibility

As discussed in Section 4.1.2, ICS components are deployed in adverse environments. This includes environments with high or low temperatures, dust or high amounts of humidity. Hence ICS components are generally built more robust than components in Desktop IT. This makes physically accessing the components more difficult.

In addition, [NIS15] states:

> *„Components can be isolated, remote, and require extensive physical effort to gain access to them“*

This might especially be the case when an ICS spans a large physical space or when certain components are installed in locations which are hard to reach. This complicates the physical access to the specific components which might complicate certain means of **DG** which rely on obtaining physical access to the components. Also, a lack of physical access makes *Post-Mortem Investigation* impossible. If a component is remote and cannot be reached physically and logically any more this component cannot be investigated.

Component Lifetime

The length of the lifetime of ICS is a topic of discussion and depending on the exact use of the system in question. [NIS15] assumes a lifetime of a specific ICS component in the order of 10 to 15 years. In the case of most manufacturing process this might be true. A certain model of car is manufactured on a production line for around 10 years before the line is closed and replaced by a new one. In this case, the various components might be replaced by newer ones. There are however also industrial manufacturing process which take place over a longer timespan. For example, an automated system within a saw mill might perform the same task over the course of 20 or 30 years. This observation is supported by [VKL16] which states:

> *„It is common for an ICS system to run for 20 or 30 years without update or upgrade“*

There are also those systems which require an extensive certification. This is the case with critical infrastructures. Here, new components, architectures, protocols, etc. would have to be certified. This makes the replacement of these difficult and unlikely.

Of course, legacy systems also occur in the Desktop IT domain but they are more common in the ICS domain.

Legacy systems complicate the access since they require yet different interfaces and use yet different protocols. Also, the documentation for these components might be unavailable and hence the required knowledge might be unavailable even for a forensic investigator with extensive vendor-specific knowledge.

Availability (Reliability) Requirements

The ICS domain offers various reasons to not shut down an ICS in order to perform a forensic investigation and also some reasons to shut it down in order to perform a comprehensive investigation before the ICS resumes its operation.

On one hand, there is a high monetary cost associated with shutting down ICS since this would prevent the achieving of the industrial objective (see Section 2.2.2). Not achieving this objective might even be more critical than mere financial loss. For example, electrical power generation water supply are widely considered as critical infrastructures (see [EU008] for the European guidelines and [KRI16] for the German guidelines on the identification of critical infrastructures). Shutting down such a critical infrastructure has dire consequences for public safety. Another example is provided in [KL15] where the disruption of the production of a pharmaceutical plant producing a specific vaccine is used as an example how the shutting down of certain key ICS might not be an option.

On the other hand, these ICS control processes which might be dangerous if performed in a faulty (or maliciously manipulated) manner or which are dangerous in itself (if left uncontrolled). The first example is the case when an attacker is able to manipulate the production process in the pharmaceutical plant used as an example by [KL15]. Or if an attacker would be able to control the processes in a chemical plant in order to cause catastrophic damage. This would letting an ICS continue its operation after a suspected attack seems risky.

Other physical processes are dangerous when not controlled by an ICS. This applies to some chemical processes or those present in nuclear power generation. Here, the ICS cannot be simply shut off from one moment to another without risking severe consequences. In this case the physical process has to be shut down in order to prevent these severe consequences.

In general, *Post-Mortem Investigations* in ICS will only rarely be employed when compared to *Live Forensics*, which also offer an easier accessibility to *Non-volatile Memory* and *Volatile Memory* (see Section 4.1.2.2 and Section 4.1.2.4 respectively).

Performance Requirements

ICS generally involve the need for time-critical responses from PLCs. This might lead to conflicts between performing *Live Forensics* and fulfilling these real time requirements, especially when considering the low computing power of the components involved (see Section 4.1.2) and the bandwidth of some network technologies used (see Section 4.2.2).

Additional Aspects

There are aspects of the scenarios in which ICS are deployed beyond these three factors which affect the forensic process in the ICS domain.

The objective of ICS is to achieve an industrial objective (see Section 2.2.2). How such an industrial objective is achieved can in some cases be considered a business secret or intellectual property. This restricts the handling of forensic evidence which could disclose these business secrets (**PC3** - see Section 2.3.2.4).

4.3.2 Automotive Scenarios

This section discusses how the identified three factors of deployment affect computer forensics in the Automotive IT domain.

Component Accessibility

Physical access to components in Automotive IT is, in general, only possible if the vehicle is not moving. In this case, some components can easily be accessed by opening up the engine hood. Other components are less accessible because they are situated less accessible within the vehicle. Here, the disassembly of certain parts might be necessary. Such a disassembly is in general more complex than in the Desktop IT domain. It might also require specialized equipment to be performed. Also the components are designed to withstand extreme environmental conditions including temperatures more extreme than those faced by Desktop IT or ICS components and constant vibrations (see Section 4.1.3). This leads to the individual casing of the specific components being sturdier.

An advantage over ICS for the forensic investigator is however the fact that all components of Automotive IT are usually together in one place and not distributed across a larger area.

This more complicated access and the problems with accessing *Non-volatile Memory* and *Volatile Memory* during a *Post-Mortem Investigation* (see Section 4.1.3.3 and Section 4.1.3.5 respectively) lead to a general reliance on *Runtime Interrogation* during *Live Forensics*.

Component Lifetime

The life cycle of cars is long when compared to Desktop IT. Depending on the quality and use of the car, it can be assumed that the current life cycle of a car is around 10 to 12 years[105][106].

Since the protocols and interfaces used in Automotive IT do not change often (and their more standardized nature in the first place - see Section 4.2.3) this is does not lead to the same problems with legacy components and interfaces as in the ICS domain. However, the requirement of a compatibility with those legacy components prevents the inclusion of more secure communication protocols which would also benefit the forensic process by ensuring origin authenticity in contrast to the currently used CAN bus - see Section 4.2.3.2.

Availability (Reliability) Requirements

Shutting down a car is usually no major problem as it does not incur massive financial cost and is not required to maintain a critical infrastructure.

This being said, a car has to be moved into a safe position before shutting down its Automotive IT to prevent the risk of injury of the driver. Directly shutting down the Automotive IT after a potential attack is identified carries the risk of the driver not being able to handle the vehicle safely.

On the other hand, a potentially manipulated car that is not brought to a standstill immediately can cause an accident, as presented by [MV15]. In each case, a security problem in Automotive IT implies a safety problem.

Performance Requirements

Automotive IT involves time-critical responses. This might lead to conflicts between performing *Live Forensics* and fulfilling these real time requirements, especially when considering the low computing power of the components involved (see Section 4.1.3) and the bandwidth of the CAN bus (see Section 4.2.3).

[105] https://www.acea.be/statistics/tag/category/average-vehicle-age, 23/05/2020

[106] https://www.aarp.org/auto/trends-lifestyle/info-2018/how-long-do-cars-last.html, 23/05/2020

4.4 *Step 1.4* Analysis of Attacks

Knowledge of the nature of attacks in given domain can be used in order to understand what peculiarities the forensic process in this domain has to address. This section discusses known attacks in the ICS and Automotive IT domains in order to understand which *Data Types* are relevant for forensic investigations in order to address **AF0**.

This section discusses known attacks on ICS and then discusses attacks on Automotive IT. Desktop IT is omitted due to this topic being well-researched and already considered in detail during the creation of the [KDV15]-model for the forensic process.

4.4.1 Industrial Control Systems Attacks

The topic of various attacks on ICS and how the various attacks affect the forensic process within ICS is comprehensively discussed in [**ALK^{+}18**]. This publication discusses four different real attacks on ICS in order to identify the potential forensic traces caused by these specific attacks.

Another approach was chosen in [SJaTW14] which identified three different types of typical attacks against SCADA systems. These typical patters are focused on typical vulnerabilities present in ICS and the consequences of said attacks. These three types of typical attacks as described in [SJaTW14] are:

- Communication stack
  Many of the protocols used in ICS do not employ any security mechanics (see Section 4.2.2.2)
- Hardware
  Lack of authentication at PLCs allows attacker to alter the set points
- Software
  ICS are plagued by security problems in software implementations

While this provides an useful overview, it does not directly address the forensic evidence available after such methods have been used by an attack (with the notable exception of a manipulation of set points which would alter *Volatile Memory* of the PLC as well as the behavior of the physical process). Hence, the results of [**ALK^{+}18**] are updated to serve as a foundation to discuss the presence of evidence in the various *Data Streams* after certain attacks.

This section provides an overview on the results of [**ALK^{+}18**] and follows the same structure, although the contents were actualized. At first, an overview on the four attacks is provided, including the role of the ICS in the given attack. Then, the potential forensic traces caused by these attacks are identified.

4.4.1.1 Four Different Examples for Attacks Targeting Industrial Control Systems

These specific examples for attacks on ICS are taken from [**ALK^{+}18**]. These examples were chosen to represents a wide range of different implications for the forensic process.

Dragonfly (aka Energetic Bear)

Dragonfly attacked energy various suppliers (see [Sym14]). Dragonfly gathered information by infecting the targeted systems with the Dragonfly Remote Access Trojan horse (RAT). This remote access enabled attackers to use the specialized Havex malware on the systems. The specific functionality is described in [Nel16]:

> *„Havex used an OPC malware scanning module to gather information about ICS devices and send that data back to Command and Control (C&C) servers used by the Dragonfly group. The malware used an industrial protocol scanner to find networked devices on TCP ports 44818, 102 and 502. Automation companies such as Siemens and Rockwell Automation use these ports for ICS system communication. The industrial processes using the protocols are found in consumer goods manufacturing and packaging applications“*

While this attack was aimed towards ICS it did not directly attack the ICS in question. This attack targeted the Desktop IT domain but could reach the upper levels of the ICS since some of these components are connected directly to the systems within Desktop IT. For further reference defer to the PERA hierarchy (see Section 2.2.2.1) or the discussions on ICS hierarchies (see Section 4.2.2.1). Hence, this attack aims at the connection between Desktop IT and ICS. It is included here to show the close relationship between these both domains and to shows a common course for many attacks which are related to ICS.

BlackEnergy

BlackEnergy started out as a popular crimeware (malware designed to automate criminal activities) (see [FSE14]). However, this malware is mostly known for its use during the cyber attacks on Ukrainian power grids in late 2015. These attacks were performed in a complex manner aligned with the Cyber Kill Chain®, a model to describe complex attacks ([CCK15]). According to [DvHvH17], the attackers used an approach with different steps. **[ALK+18]** provides a summary of these steps based on [LAC16]:

> *„In a preparation stage, access to the network was obtained by using spear-fishing attacks deploying the BlackEnergy 3 malware. An extensive reconnaissance stage followed, allowing the tailoring and planning of further actions. When the final stage of the attack started, the attackers sent the signal to open the power breakers, overwrote the firmware of serial-to-Ethernet-converters (thereby destroying the link between SCADA and the PLC), disabled the uninterruptable power supply and, finally, wiped the hard drives of the SCADA systems. The last three steps aimed at making recovery from this cyber attack more difficult. However, the attack did not aim at 'hacking' the ICS in question - the attackers essentially took over the SCADA system, sending perfectly legitimate commands to the ICS in question.“*

This attack targeted the ICS but did not compromise any of the components belonging to the ICS. The attack led the Desktop IT components to send malicious (but legitimate) commands to the ICS.

Stuxnet

Stuxnet is a prime example for a complex attack scenario. This attack also followed a multi-step-approach which consisted of compromising systems in the targeted network (level 4 and 5 of PERA), then infecting the programming network for the ICS (level 4 of PERA) and then finally the ICS itself (level 0-2 in PERA) as described in [FMC].

In essence, the final stage of the Stuxnet attack was the injection of malicious logic into the ICS code while using rootkit techniques to hide this logic from the programmer. The modified ICS code (containing the malicious logic, still hidden from the programmers view) was then loaded on the ICS and executed - leading to the breakdown of actuators (permanent physical destruction).

This complex attack included a manipulating of components along the entire scope of the PERA hierarchy. It compromised systems in the connected Desktop IT as well als ICS.

PLC-Blaster

PLC-Blaster is a proof of concept for a malicious software executed directly in the *Volatile-Memory* of a PLC presented in [SBS16]. This malicious software resides within a PLC and scans the attached network for potential targets. After the identification of such targets, the malicious software tries to infect the PLC in question using the network connection. As a proof of concept, [SBS16] shows various possible malicious actions the malicious software could then perform one the PLCs are infected.

This proof of concept does not include any means how the malicious software got into the ICS network in the first place. Hence, this scenario only regards the ICS components of the entire PERA hierarchy.

4.4.1.2 Traces Caused by Different Attacks Targeting Industrial Control Systems

The four attacks discussed in Section 4.4.1.1 cause forensic traces within the various *Data Streams* and levels of the PERA Hierarchy. This points is already discussed in [**ALK+18**] but is summarized here in order to provide a better understanding of forensic traces in the ICS domain.

An overview on these traces, based on the respective table provided in [**ALK+18**], can be seen in Table 4.11. It is notable that this overview just shows in which *Data Streams* and levels of the PERA hierarchy the discussed attacks could have left forensic traces. It does not give any notion on how accessible and comprehensive these traces are. In essence, this overview describes which *Data Streams* on which levels of the PERA hierarchy were affected by the attack.

Already the overview provided in Table 4.11 shows that depending on the specific attacks all *Data Streams* and levels of the PERA hierarchy might include traces of value for the forensic investigation. The following paragraphs will describe the traces caused by these different attacks in more detail and provide a conclusion.

Table 4.11: Availability of forensic traces in various *Data Streams* and levels of the PERA hierarchy caused by various attacks targeting ICS based on [**ALK+18**] with the addition of a hypothetical supply chain attack

Attack				
Possible traces in ..				
Data Stream	Level 1	Level 2	Level 3	Level 4
Dragonfly				
Non-volatile Memory				✓
Volatile Memory				✓
Communication			✓	✓
BlackEnergy				
Non-volatile Memory		✓	✓	(✓)
Volatile Memory		✓	✓	(✓)
Communication	✓	✓	✓	✓
Stuxnet				
Non-volatile Memory	✓	✓	✓	✓
Volatile Memory	✓	✓	✓	✓
Communication	✓	✓	✓	✓
PLC-Blaster				
Non-volatile Memory				
Volatile Memory	✓			
Communication	✓	✓		
Hypothetical supply-chain attack				
Non-volatile Memory	✓			
Volatile Memory	(✓)			
Communication	(✓)	(✓)	(✓)	(✓)

Dragonfly (aka Energetic Bear)

Dragonfly performs network scans in the Business IT. This causes some traces in *Communication* on level 4 of PERA. However, it could also be propagated to level 3 since some of the systems located on level 4 would have access to level 3. No ICS systems on level 1 or level 2 in itself are altered (neither *Non-volatile Memory* nor *Volatile Memory*). However, as some systems in Desktop IT might be compromised, it is necessary to point out, that the components on level 4 might contain traces in *Non-volatile Memory* and *Volatile Memory* (in contrast to the conclusion made in [**ALK+18**]).

BlackEnergy

BlackEnergy aims at altering the behavior of components situated on level 2. Hence, this attack affects and leaves traces in all *Data Streams* on this level. It also compromises systems on level 3 as a necessary stepping stone leaving traces in all *Data Streams*. From [LAC16] it is not clear if level 4 is also affected in this manner, but level 5 (entirely inside the Desktop IT domain) will surely have traces of the spearfishing campaign.

Components on level 1, namely PLCs, are not directly compromised. However, capturing the *Communication* from level 2 to these components would show the malicious command signals.

Stuxnet

Stuxnet is a highly complex and advanced attack that manipulates and compromises various systems on the way to the PLC on level 1. It leaves well-hidden traces on anything it touches. Most dramatically, it alters all *Data Streams* on level 1 and 2. level 3 and 4 are to be considered as stepping stones but also are affected and might provide traces in all *Data Streams*.

PLC-Blaster

PLC-Blaster is only resident in the *Volatile Memory* of the PLCs on level 1. It uses (and hence alters) the *Communication* to other components on level 1 and level 2 to spread itself. Since the level 2 components differ from the ones employed on level 1, this proof of concept cannot compromise any system on level 2. Also, the proof of concept does not include any function to alter the *Non-volatile Memory* of PLCs. While such functionality could be added, it is not included and hence the *Non-volatile Memory* is removed from Table 4.11 in contrast to [**ALK+18**].

Other possible Attack Scenarios

Aside from the four specific attacks explored in this section additional attacks on ICS environments are possible. All of these attacks include the transmission of malicious code or programs via network interfaces into pre-installed components. It is also possible that malicious code might already be included within a piece of hardware at the time of installation. Such a supply chain attack would then include a section of malicious logic either triggered by some command and control signal (C2) introduced via the *Communication* or some other logical condition. Such a logical condition would also most likely be supplied via one of the attached sensors (relying on *Communication*) but might be far less noticeable since e.g. a specific time or state of the physical process might be the trigger and the transmission of said data totally inoffensive, regular and in itself not noteworthy.

In each case, traces of the malicious logic might be found within the *Non-volatile Memory* of the compromised component. Such traces might also be found in the *Volatile Memory*. Depending on the exact nature of the trigger for the malicious behavior, a command and control signal might be propagated through the entire network or not required at all.

Conclusion

Attacks targeting ICS are very diverse. Based on the systems directly affected by the attacks traces can be found on all levels of the PERA hierarchy and in all *Data Streams*. However, since a notable portion of attacks depends on the transmission of legitimate but malicious commands the *Communication* seems to be of great importance. This is especially true if the attacker has no means to manipulate the components before they are installed within the ICS.

Also, the close relationship between ICS and Desktop IT is visible. This is due to the fact that Desktop IT is often the stepping stone for accessing ICS networks. Furthermore, the Desktop IT portion of PERA is often used to perform a reconnaissance on the ICS (see [CCK15] and [DvHvH17]).

The four examples of specific attacks cover a broad range of known specific attacks on ICS while the hypothetical supply chain attack covers a range of different but similar attack scenarios. To the best knowledge of the author all known attack on ICS do correspond to one of these five given examples.

4.4.2 Automotive IT Attacks

This section discusses attacks on Automotive IT and what potential forensic traces these attacks cause.

During the last years some attacks on Automotive IT became public. Since these attacks are less diverse in nature than those encountered in the ICS domain (see Section 4.4.1), the discussion will describe these attacks in a generalized manner.

The most famous example is the attack performed by [MV15] in which the researchers gained access to the CAN bus (see Section 4.2.3 by compromising a device with access to the CAN bus and external interfaces. In this case, this device was the radio which contained various external interfaces. The attack was possible since every device on the CAN bus has the possibility to receive and send any message. In this case, the compromised system could send specific commands over the CAN bus in order to achieve the desired effect.

In essence, this attacks boils down to three steps which are characteristic for attacks on Automotive IT:

- Step 1: Gain access to the CAN bus
  This can be either logically (as in [MV15]) or physically (when directly connecting to the CAN bus)

- Step 2: Gain model-specific knowledge on the commands available
  This can be skipped if such knowledge is already available

- Step 3: Send the commands
  In order to achieve the desired effects

It is notable that the first two steps are the same a forensic investigator has to perform in order to access the *Communication* within Automotive IT and to gain the possibility to perform diagnostic requests, which are a backbone of investigations into the *Non-volatile Memory* and *Volatile Memory* in Automotive IT. Indeed, the attacks might also employ diagnostic commands (as is the case in [MV15]).

This is the case since obtaining physical access to the PLCs is comparatively difficult (as discussed in Section 4.3.2). Gaining physical access to the CAN bus is easier.

Hence, the biggest group of attacks against Automotive IT are those were command and diagnostic requests are used to achieve a malicious intent. The commands are in itself legitimate. Depending on these commands, the *Non-Volatile Memory* and/or *Volatile Memory* of various PLCs could be altered. This is especially the case when the commands intend to alter parameters (like used in Chip-Tuning) or to reset DTCs (see Section 4.1.3.2). Such attacks directly alter the *Communication* in each case.

If an attacker gains physical access to an ECU and disassembles the shielding in order to gain direct access to the *Non-Volatile Memory* an alteration is possible. This is also the case when an attacker gains access to the debug interfaces potentially included in Automotive IT (see Section 4.1.3.5) and alters the *Volatile Memory*. Depending on the exact behavior of the manipulated component, this might or might not cause an alteration of the components behavior and hence might or might not incur evidence in the *Communication*. Such an attack will alter *Volatile Memory* and *Non-Volatile Memory*. However, the barriers to perform such an attack are high are require physical access to the interior of the specific component. Hence, this is an unlikely attack pattern outside the field of supply-chain attacks.

It can be concluded, that the *Communication* is the most useful source of forensic evidence in the Automotive IT domain, since in this domain basically everything happens based on the *Communication*.

5. Step 2: Revisiting the Forensic model

This chapter discusses how the findings of the analysis of the ICS and Automotive IT domain affect the conduct of forensic investigations and how the forensic process model described in Section 3.1.2 has to be altered in order to better reflect these two domains. In essence, this chapter aims at answering **Research Question 3**. In order to achieve this, the findings of Chapter 4 which aimed towards **Research Question 1** and **Research Question 2** are summarized to provide a clear foundation for the necessary adaptation.

This chapter starts with a summary on the differences between the discussed domains and their impact on the forensic process (Section 5.1), then discusses how these differences affect the forensic process model discussed in Section 3.1.2 (Section 5.2). The adaptation of the forensic process model concludes this chapter in Section 5.3.

5.1 Summary of Differences Between the Specific Domains

Chapter 4 provided a comprehensive analysis of the ICS and Automotive IT domain with a focus on how the properties of these domains affect the forensic process. The analysis of components identified how the specific domains impact **DG** and **DI** of *Non-volatile Memory* and *Volatile Memory* (see Section 4.1). The analysis of the employed architectures identified how these domains impact **DG** and **DI** of *Communication* (see Section 4.2). Further influence on the forensic process results from the scenarios common in these domains (see Section 4.3). A discussion of typical attacks and the forensic traces they leave within systems belonging to these domains provides additional input (see Section 4.4).

The following sections provide a summary on how the technical or organizational properties impact the conduct of forensics. This summary stands as the answer to **Research Question 1** and is split between a section discussing ICS and a section discussing Automotive IT respectively. This summary is organized along the lines of Chapter 4. At first, the impact of the properties of the specific components on forensic is discussed, followed by the impact of the architectures, the scenarios and then the respective attacks.

5.1.1 Industrial Control System

This section discusses the impact of the properties of the ICS domain on the conduct of forensic investigations.

5.1.1.1 Impact of the Properties of Components in Industrial Control System on the Conduct of Forensic Investigations

The impact of the properties of the components used in ICS is discussed in Section 4.1.2. The following nine *Influence Factors* (*IF*) are identified:

- Low memory capacity ($\mathbf{IF}_{ICS.Comp.1}$)
  ICS components have less memory and storage available than Desktop IT system. This leads to a reduced amount of potential available forensic traces since extensive log files are usually not created.

- Low computing power ($\mathbf{IF}_{ICS.Comp.2}$)
  ICS components have lower computing power available than Desktop IT system. Some of these computing units are able to fulfill their computing task but have no capacity to fulfill additional diagnostic requests without impact for the controlled physical process.

- High diversity of the domain in terms of software($\mathbf{IF}_{ICS.Comp.3}$)
  ICS components vary widely in regards of the complexity of the used operating systems. Some employ only bare execution environments for the user program while others even employ GNU/Linux ([Ber11]). This reduces the generality of methods and statements.

- Difficult physical access to *Non-volatile Memory* ($\mathbf{IF}_{ICS.Comp.4}$)
  Physical access to *Non-volatile Memory* as employed during *Post-Mortem Forensics* is either simple (in case of memory cards) or complex (in case of *non-volatile battery backed RAM*). In the later case, the memory might be part of the respective silicone of the computing unit. Also, the casings of PLCs are usually not designed to be opened up easily. Special equipment might be necessary. Depending on the employed technology, specialized interfaces are necessary to access the *Non-volatile Memory* contents.

- Logical access to *Non-volatile Memory* and *Volatile Memory* possible(**IF**$_{ICS.Comp.5}$)
  Logical access to *Non-volatile Memory* and *Volatile Memory* by performing *Runtime Interrogation* during *Live Forensics* is possible. The extent of evidence achievable by this method is highly dependent on the specific system used (see $IF_{ICS.Comp.2}$).

- Availability of *Data Types* highly dependent on the specific systems complexity (**IF**$_{ICS.Comp.6}$)
  DT1, **DT3**, **DT4**, **DT5** and **DT6** are present in ICS components since they are required for the PLC to perform its function. Data which represents the physical process can be considered as **DT8**. Each of this *Data Types* has a physical representation in **DT2**. However, in the case of a PLC running a complex Desktop operating system the same *Data Types* as in Desktop IT will be present (see $IF_{ICS.Comp.3}$).

- Reliance on vendor-specific knowledge and interfaces (**IF**$_{ICS.Comp.7}$)
  Formats used in the ICS domain are usually proprietary and closed source. Hence the forensic investigator has limited means to interpret **DT2** in case data can be acquired in this form. *Runtime Interrogation* is more common but the forensic investigator has little knowledge of the inner workings of the software used to obtain logical access to *Non-volatile Memory* and *Volatile Memory* (see $IF_{ICS.Comp.5}$). This lack of traceability of the performed operations violates the *Admissibility Factors* (see Section 2.1.7) and has a negative impact on the evidentiary value.

- Lack of tools geared towards forensic use (**IF**$_{ICS.Comp.8}$)
  If a tool is available to access either *Non-volatile Memory* and *Volatile Memory* by performing *Runtime Interrogation* or interpreting **DT2** gathered from an ICS, this tool will most likely not be geared towards forensic use. Hence, the *Admissibility Factors* as discussed in Section 2.1.7 are not fully addressed and the evidentiary value is negatively impacted.

- Varying complexity of ICS components (**IF**$_{ICS.Comp.9}$)
  As discussed in Section 4.1.2.6, sensors and actuators attached to PLC actually can have varying degrees of complexity. Some are hard-wired, others are (re-)programmable. Those that are more complex might contain traces in *Non-Volatile Memory* and *Volatile Memory*. For those components which are not re-programmable, the contents of *Non-Volatile Memory* are of no interest outside of attacks involving supply-chain attacks.

5.1.1.2 Impact of the Properties of System Architectures in Industrial Control System on the Conduct of Forensic Investigations

The impact of the properties of the system architectures used in ICS is discussed in Section 4.2.2. The following eleven **IFs** are identified:

- Established network hierarchies (**IF**$_{ICS.Arch.1}$)
  The *Communication* in ICS is organized alongside of hierarchies. These hierarchies and the resulting flow of *Communication* follow established patterns as described by the PERA (Purdue enterprise reference architecture - [Wil92]) or the ISA95-Standard ([ISA10]). This can benefit the placement of network taps or similar means to access the *Communication*.

- Low bandwidth ($\mathbf{IF}_{ICS.Arch.2}$)
  Depending on the employed network technology ICS networks might not have an abundance of bandwidth as usual in Desktop IT. In some cases the bandwidth will be enough to fulfill the computing task but have no capacity to fulfill additional diagnostic requests without impact for the controlled physical process.

- High diversity of deployed (vendor-specific and proprietary) cable-based carrier mediums ($\mathbf{IF}_{ICS.Arch.3}$)
  A broad range of different, incompatible technologies are used to facilitate networking within ICS. Some of these technologies are highly vendor-specific, proprietary and closed source. These require specialized hardware to access. This makes this domain less accessible for an independent forensic investigator.

- High diversity of deployed (vendor-specific and proprietary) communication protocols ($\mathbf{IF}_{ICS.Arch.4}$)
  A broad range of different, incompatible technologies are used to facilitate networking within ICS. Some of these technologies are highly vendor-specific, proprietary and closed source and require specialized software to interpret. This makes this domain less accessible for an independent forensic investigator.

- Ethernet-based communication protocols are easy to access ($\mathbf{IF}_{ICS.Arch.5}$)
  In the case of Ethernet-based networks, methods known from the Desktop IT domain (see Section 4.2.1.2) can be used to access the carriers and perform **DG**. Here, tools created for forensic use are available and provide a high degree of evidentiary value.

- Wireless communication protocols are easy to access ($\mathbf{IF}_{ICS.Arch.6}$)
  In the case of wireless networks, standardized and open technologies are used which can be accessed with generally available hard- and software. Depending on the exact technology used, tools geared towards forensic use might be available.

- Ethernet-based communication protocols are easier to interpret ($\mathbf{IF}_{ICS.Arch.7}$)
  Some of the Ethernet-based communication protocols of the newer generation follow open standards which allows for an investigation using tried and tested software, including open-source solutions. While no dedicated software to perform forensic investigations into ICS *Communication* exists, the evidentiary value is positively impacted by the use of tried and tested open source software in contrast to highly vendor specific solutions from the viewpoint of an independent forensic investigator.

- Lack of tools geared towards forensic use ($\mathbf{IF}_{ICS.Arch.8}$)
  There is a general lack of tools geared towards forensic use in this Domain in regards to *Communication*.

- Presence of Historians within ICS ($\mathbf{IF}_{ICS.Arch.9}$)
  Some ICS might contain a Historian tasked with preserving data about the physical process (**DT8**). These historians might, depending on implementation, use certain methods to ensure the integrity and authenticity of the data. This might even go so far as to satisfy forensic requirements, though this is generally not the case (see [WDJC13]). In any case, they do not address anything besides the physical process but the **DT8** provided by them might still support a forensic investigation.

- Dominance of **DT8** in *Communication* in ICS (**IF**$_{ICS.Arch.10}$)
  In essence, the *Communication* can contain all *Data Types* - due to the fact that the PLC can be programmed and configured using network interfaces. However, in the usual operation, the bulk of data transmitted belongs to **DT8** since this data describes the underlying physical process. This data has a raw representation (**DT2**) and uses protocols to be transmitted (implying **DT3**).

  An important note to avoid confusion: this *Influence Factor* will be redefined during the revision of the core concepts of the forensic process model since the *Data Types* will be revisited during this process (see Section 5.3). Hence, the entry of this *IF* in Section A.1 is altered to reflect this change.

- Low and predictable traffic (**IF**$_{ICS.Arch.11}$)
  While Desktop IT networks contain mainly user-generated traffic, traffic in ICS is routine and predictable. In addition, the amount of traffic is - by modern standards - relatively low. This can simplify forensic investigations into the *Communication* in ICS considerable.

5.1.1.3 Impact of Scenarios in Which Industrial Control System are Used on the Conduct of Forensic Investigations

The impact of the scenarios in which ICS are deployed on the forensic process is discussed in Section 4.3.1. The following six **IFs** are identified:

- Reduced component accessibility (**IF**$_{ICS.Scen.1}$)
  Physical access to ICS components might be difficult since ICS can span large areas. Components might be installed in locations which are difficult to access. This complicates **DG** especially when considering *Post-Mortem Investigation* impossible. If a component is remote and cannot be reached physically and logically any more, this component cannot be investigated.

- High component lifetime (**IF**$_{ICS.Scen.2}$)
  Depending on the ICS in question, the respective lifetime might be between 10 and 30 years. Legacy systems complicate the access since they require yet different interfaces and use yet different protocols. Also, the documentation for these components might be unavailable and hence the required knowledge might not be available even for a forensic investigator with extensive vendor-specific knowledge.

- High availability requirements (**IF**$_{ICS.Scen.3}$)
  There is a high monetary cost associated with shutting down ICS in order to perform forensic investigations. Some ICS might provide critical infrastructures and shutting down these systems has an impact on public safety. Some physical processes controlled by an ICS might be hazardous and unstable when the ICS controlling the process is shut down. This applies to some chemical processes or those present in nuclear power generation. Here, the ICS cannot be simply shut off from one moment to another without risking severe consequences. In this case the physical process have to be shut down in an orderly fashion to prevent these severe consequences. This favors the use of *Live Forensics*.

- ICS control physical processes which might end in catastrophic events should the ICS be compromised ($\mathbf{IF}_{ICS.Scen.4}$)
  ICS can be deployed to control processes which might be dangerous when performed in a faulty (or maliciously manipulated) manner or which are dangerous in itself. An attacker who gains control about the ICS of a chemical plant might be able to cause catastrophic damage. This would letting an ICS continue its operation after a suspected attack seem risky and argues for shutting down the respective ICS and perform *Post-Mortem Forensics*.

- ICS requires time-critical responses from PLCs ($\mathbf{IF}_{ICS.Scen.5}$)
  ICS generally involve the need for time-critical responses from PLCs. This might led to conflicts between performing *Live Forensics* and fulfilling these real time requirements, especially when considering the low computing power of the components involved (see $IF_{ICS.Comp.2}$) and the bandwidth of some network technologies used in ICS (see $IF_{ICS.Arch.2}$).

- Forensic evidence might contain intellectual property ($\mathbf{IF}_{ICS.Scen.6}$)
  The way an ICS achieves an industrial objectives can be considered a business secret. This restricts the handling of forensic evidence which could disclose such business secrets (**PC3** - see Section 2.3.2.4). Since ICS are deployed for specific industrial objectives and contain highly task-specific (and usually far less accessible) **DT8**, this is more relevant than in the Automotive IT domain.

5.1.1.4 Impact of Attacks Targeting Industrial Control System on the Conduct of Forensic Investigations

The impact of the attacks targeting ICS on the forensic process, including the potential traces caused by these attacks, is discussed in Section 4.4.1. The following three **IFs** are identified:

- Diverse attacks targeting ICS ($\mathbf{IF}_{ICS.Atk.1}$)
  Attacks targeting ICS are very diverse. Hence, specific attacks can leave forensic traces in all *Data Streams* and components within an ICS.

- Focus on legitimate but malicious commands ($\mathbf{IF}_{ICS.Atk.2}$)
  A notable amount of attacks on ICS focus on transmitting legitimate but malicious commands. This implies a great importance of *Communication*.

- Close relationship between ICS and Desktop IT ($\mathbf{IF}_{ICS.Atk.3}$)
  Desktop IT is often used as stepping stone to gain access to any ICS. This is also the case when the attack consists of transmitting legitimate but malicious commands from the Desktop IT to the ICS (see $IF_{ICS.Atk.2}$). A forensic investigation into ICS might also include a forensic investigation into the connected Desktop IT in order to identify the path chosen by the attacker to gain access to the ICS.

5.1.2 Automotive IT

This section discusses the impact of the properties of the Automotive IT domain on the conduct of forensic investigations.

5.1.2.1 Impact of the Properties of Components in Automotive IT on the Conduct of Forensic Investigations

The impact of the properties of the components in Automotive IT is discussed in Section 4.1.3. The following nine **IFs** are identified:

- Low memory capacity ($\mathbf{IF}_{AMI.Comp.1}$)
  Components in Automotive IT have less memory and storage available than Desktop IT system (and even less than those in the ICS domain). This leads to a reduced amount of potential available forensic traces since extensive log files are usually not created.

- Low computing power ($\mathbf{IF}_{AMI.Comp.2}$)
  Components in Automotive IT have lower computing power available than Desktop IT system. Some of these computing units are able to fulfill their computing task but have no capacity to fulfill additional diagnostic requests without impact to the driving task.

- Memory wear reduces reliance on *Non-volatile Memory* ($\mathbf{IF}_{AMI.Comp.3}$)
  Non-volatile Memory in Automotive IT is often flash memory and the long life cycles of Automotive IT forces the system to only write necessary data to the *Non-volatile Memory.* This reduces the amount of forensic traces available from *Non-volatile Memory* further.

- Difficult physical access to *Non-volatile Memory* ($\mathbf{IF}_{AMI.Comp.4}$)
  Physical access to *Non-volatile Memory* as employed during *Post-Mortem Forensics* is difficult. The memory might be part of the respective silicone of the computing unit. Also, the casings of ECUs are designed not to be opened up easily. Special equipment might be necessary. Specialized interfaces are necessary to access the *Non-volatile Memory* contents.

- Logical access to *Non-volatile Memory* and *Volatile Memory* possible ($\mathbf{IF}_{AMI.Comp.5}$)
  Logical access to *Non-volatile Memory* and *Volatile Memory* by performing *Runtime Interrogation* during *Live Forensics* is possible. The extend of evidence achievable by this method is generally high since ECUs usually have diagnostic functionality which can be accessed with standardized interfaces and applications. However, most of the logical access performed in this manner relies on the CAN bus for transport. The authenticity of any message received using the CAN bus is at least doubtful (see $IF_{AMI.Arch.7}$).

- Reduced availability of *Data Types* ($\mathbf{IF}_{AMI.Comp.6}$)
  Data within ECUs includes **DT1**, **DT2** and **DT5**. Information required for the driving task, like the current speed of the vehicle, is also present in *Non-volatile Memory.* Such data would be considered as **DT8**.

- Reliance on vendor-specific knowledge and interfaces ($\mathbf{IF}_{AMI.Comp.7}$)
  Interpreting **DT2** gathered in the Automotive IT domain relies on vendor-specific knowledge. This is also the case for tools performing diagnostic requests to the specific ECUs (see $IF_{ICS.Comp.5}$). The forensic investigator has little knowledge of the inner workings of the software used to obtain logical access to *Non-volatile Memory* and *Volatile Memory.* This lack of traceability of the performed operations violates the *Admissibility Factors*

(see Section 2.1.7) and has a negative impact on the evidentiary value. For an independent forensic investigator without access to vendor-specific solutions or knowledge the domain is far less accessible.

However, there is a selection of usable open source tools (as presented in [**ALKD17**]) and approaches to deduce such vendor-specific knowledge ([**KAH+18**] and [Smi16]).

- Lack of tools geared towards forensic use (**IF**$_{AMI.Comp.8}$)
  If a tool is available to access either *Non-volatile Memory* and *Volatile Memory* by performing *Runtime Interrogation* or interpreting **DT2** gathered from Automotive IT, this tool will most likely not be geared towards forensic use. Hence, the *Admissibility Factors* as discussed in Section 2.1.7 are not fully addressed and the evidentiary value is negatively impacted.

- ECUs are usually not shut down(**IF**$_{AMI.Comp.9}$)
  A distinct peculiarity of Automotive IT is the fact that ECUs are usually not shut down or disconnected from power when the vehicle is stopped or turned off (see Section 4.1.3.1). Usually these ECUs are powered by the vehicle battery and only exceptional cases (like an accident) result in a loss of power. Hence, the *Volatile Memory* is available in most cases.

5.1.2.2 Impact of the Properties of System Architectures in Automotive IT on the Conduct of Forensic Investigations

The impact of the properties of the system architectures used in Automotive IT is discussed in Section 4.2.3. The following ten **IFs** are identified:

- Established and simple network hierarchies (**IF**$_{AMI.Arch.1}$)
  The *Communication* in Automotive IT usually follows a very clear hierarchy with a CAN gateway connecting various CAN buses and other buses together. This clear and simple structure can benefit the placement of network taps or similar means to access the *Communication*.

- Low bandwidth (**IF**$_{AMI.Arch.2}$)
  Networks in Automotive generally do not have an abundance of bandwidth as usual in Desktop IT. In some cases the bandwidth will be enough to fulfill the computing task but have no capacity to fulfill additional diagnostic requests without impact for the driving task.

- High diversity of deployed (vendor-specific and proprietary) cable-based carrier mediums (**IF**$_{AMI.Arch.3}$)
  While the CAN bus is the dominant bus system within Automotive IT, other bus system still occur. Some of these technologies are highly vendor-specific, proprietary and closed source. These require specialized hardware to access. This makes this domain less accessible for an independent forensic investigator.

- High diversity of deployed (vendor-specific and proprietary) communication protocols (**IF**$_{AMI.Arch.4}$)
  While the CAN protocol is the dominant communication protocol in Automotive IT

other protocols are used with several of other bus technologies. Some of these protocols are highly vendor-specific, proprietary and closed source and require specialized software to interpret. This makes this domain less accessible for an independent forensic investigator.

- CAN-based communication is easy to physically access ($\mathbf{IF}_{AMI.Arch.5}$)
  In the case of CAN-based networks well-established methods can be used to access the carriers and perform **DG** (see Section 4.2.3.4). Although no tools directly created for a forensic use are available some of the available tools can be used in a forensic context while retaining decent evidentiary value. The presence of many different open sources tool allows for the adaptation of said tools for forensic use.

- Reliance on vendor-specific knowledge ($\mathbf{IF}_{AMI.Arch.6}$)
  Interpreting **DT2** gathered in the Automotive IT domain relies on vendor-specific knowledge. While the widely used CAN protocol is interpretable by generally available software, the meaning of the various CAN IDs and data fields (the CAN Matrix) is not known to an independent investigator. This makes the domain far less accessible for an independent forensic investigator.

  Approaches to deduce such vendor-specific knowledge ([**KAH+18**] and [Smi16]) are known but time-consuming.

- Lack of tools geared towards forensic use ($\mathbf{IF}_{AMI.Arch.7}$)
  There is a general lack of tools geared towards for forensic use in this Domain in regards to *Communication*.

- Doubtful Origin Authenticity in CAN networks ($\mathbf{IF}_{AMI.Arch.8}$)
  The most important internal vehicle network, the CAN bus, does not use explicit addressing. Any device on the CAN bus can receive and send any message. There is therefore no form of origin authenticity at all. Accordingly, the authorship of individual messages is at least questionable for any communication that runs via the CAN bus.

- Dominance of **DT8** in *Communication* in Automotive IT ($\mathbf{IF}_{AMI.Arch.9}$)
  Networks in Automotive IT are used to transfer data necessary for the driving task. This can be considered **DT8**. This data represents the vast majority of data transferred in Automotive IT. Furthermore, multimedia contents like maps or music are also transferred within Automotive IT. In addition, Diagnostic data can be transferred using the various communication bus systems within Automotive IT (see Section 4.1.3.3 and Section 4.1.3.5). This diagnostic data contains **DT1**, **DT4** and **DT5**. One could consider the potential presence of a diagnostic session as **DT7**. In addition, all this data has a raw physical representation (**DT2**).

  An important note to avoid confusion: this *Influence Factor* will be redefined during the revision of the core concepts of the forensic process model since the *Data Types* will be revisited during this process (see Section 5.3). Hence, the entry of this *IF* in Section A.1 is altered to reflect this change.

- Low and predictable traffic ($\mathbf{IF}_{AMI.Arch.10}$)
  While Desktop IT networks contain mainly user-generated traffic, traffic in Automotive IT is routine and predictable. In addition, the amount of traffic is - by modern standards

- relatively low. This can simplify forensic investigations into the *Communication* in Automotive IT considerable.

5.1.2.3 Impact of Scenarios in which Automotive IT is used on the Conduct of Forensic Investigations

The impact of the scenarios in which Automotive IT is used on the forensic process is discussed in Section 4.3.2. The following five **IFs** are identified:

- Reduced component accessibility ($\mathbf{IF}_{AMI.Scen.1}$)
  Physical access to components in Automotive IT is, in general, only possible if the vehicle is not moving. In this case, some components can easily be accessed by opening up the engine hood. Other components are less accessible because they are situated less accessible within the vehicle. Here, the disassembly of certain parts might be necessary. Such a disassembly is in general more complex than in the Desktop IT domain. It might also require specialized equipment to be performed. Also the components are designed to withstand extreme environmental conditions including temperatures more extreme than those faced by Desktop IT or ICS components and constant vibrations (see Section 4.1.3). This leads to the individual casing of the specific components being more sturdy.

  An advantage over ICS for the forensic investigator is however the fact that all components of Automotive IT are usually together in one place and not distributed over a large area.

- High component lifetime ($\mathbf{IF}_{AMI.Scen.2}$)
  The average lifetime of Automotive IT is around 10 to 15 years. Since the protocols and interfaces used in Automotive IT do not change often, legacy protocols are not so much a problem as with ICS. However, some legacy components with unusual interfaces to directly access them might be encountered.

- Medium availability requirements ($\mathbf{IF}_{AMI.Scen.3}$)
  Shutting down a car usually does not incur massive financial cost. However, it requires a safe position before shutting down the Automotive IT is possible without endangering driver or passengers. There is less reason to avoid *Post-Mortem Forensics* due to the need to constantly operate the system.

- Automotive IT control physical processes which might cause risk to limb and life should the Automotive IT be compromised ($\mathbf{IF}_{AMI.Scen.4}$)
  Automotive IT controls heavy objects which move at a high speed. If the Automotive IT is compromised it implies risk for live and limb. This is a good reason to perform an investigation after the vehicle is shut down. Please note, that this would imply *Post-Mortem Forensics* in other domains, but in the domain of Automotive IT this is not necessarily the case (see Section 4.1.3.1 and $IF_{AMI.Comp.9}$).

- Automotive IT must fulfill (soft) real-time requirements ($\mathbf{IF}_{AMI.Scen.5}$)
  Automotive IT involves time-critical responses. This might led to conflicts between performing *Live Forensics* and fulfilling these real time requirements, especially when considering the low computing power of the components involved (see $IF_{ICS.Comp.2}$) and the bandwidth of the CAN bus (see $IF_{AMI.Arch.2}$).

5.1.2.4 Impact of Attacks Targeting Automotive IT on the Conduct of Forensic Investigations

The impact of the attacks targeting Automotive IT on the forensic process, including the potential traces caused by these attacks, is discussed in Section 4.4.1. The following *IF* is identified:

- Focus on legitimate but malicious commands ($\mathbf{IF}_{AMI.Atk.1}$)
  A notable amount of attacks on Automotive IT focus on transmitting legitimate but malicious commands. This implies a great importance of *Communication*. Indeed, all other attacks require physical access to the ECUs. This access is difficult (see $IF_{AMI.Comp.3}$) or implies a supply chain attack.

5.1.3 Comparison of the *Influence Factors* in the Different Domains

Before this adaptation starts, it is important to point out that the investigated domains are indeed quite similar. Indeed, they share 22 of the *IFs* (see Section A.1.3). Five *IFs* slightly differ between these domains (see Section A.1.4). Five are unique to the ICS domain and three to the Automotive IT domain (see Section A.1.5).

The slight differences between the investigated domains are in the available *Data Types* ($IF_{ICS.Comp.6}$ and $IF_{AMI.Comp.6}$), the employed cable-bound network technologies ($IF_{ICS.Arch.5}$ and $IF_{AMI.Arch.5}$), the degree of reliance on vendor-specific knowledge and protocols ($IF_{ICS.Arch.7}$ and $IF_{AMI.Arch.6}$), the availability requirements ($IF_{ICS.Scen.3}$ and $IF_{AMI.Scen.3}$) and the dangers associated with the controlled physical process ($IF_{ICS.Scen.4}$ and $IF_{AMI.Scen.4}$).

IFs unique to the ICS domain are the high diversity in the complexity of the software and components ($IF_{ICS.Comp.3}$ and $IF_{ICS.Comp.9}$), the presence of certain technologies (like wireless networks - $IF_{ICS.Arch.6}$) and components (Historians - $IF_{ICS.Arch.9}$), the common problem of intellectual property in forensic traces collected in this domain ($IF_{ICS.Scen.6}$), the complexity of the employed attacks ($IF_{ICS.Atk.1}$) and the close connection to Desktop IT ($IF_{ICS.Atk.3}$).

IFs unique to the Automotive IT domain are memory wear of the commonly employed *Non-volatile Memory* ($IF_{AMI.Comp.3}$), the fact that ECUs usually retain power supply ($IF_{AMI.Comp.9}$) and doubtful origin authenticity on the most commonly used communication network (CAN bus - $IF_{AMI.Arch.8}$).

They have more in common than they share with the Desktop IT domain as Table 5.1 summarizes. This figure shows which of the *IFs* are also relevant in the Desktop IT domain. This includes the varying complexity of software ($IF_{ICS.Comp.3}$), the possibility to obtain logical access to *Non-volatile Memory* and *Volatile Memory* ($IF_{ICS.Comp.5}$ and $IF_{AMI.Comp.5}$) and the comparatively easy access to common cable-based and wireless networks ($IF_{ICS.Arch.5}$, $IF_{ICS.Arch.6}$ and $IF_{AMI.Arch.5}$). Forensic investigations in Desktop IT might also touch the topic of intellectual property ($IF_{ICS.Scen.6}$) and investigate very complex multi-stage attacks ($IF_{ICS.Atk.1}$). These will obviously involve Desktop IT ($IF_{ICS.Atk.3}$).

Vendor-specific protocols also have an impact on Desktop IT, but due to the spread of these systems reserve-engineered analysis methods are more commonly available ($IF_{ICS.Comp.7}$, $IF_{ICS.Arch.7}$, $IF_{AMI.Comp.7}$ and $IF_{AMI.Arch.6}$).

Table 5.1: *Influence Factors* is the different domains

IF in		shared	different	relevant in
ICS	Automotive IT	... in both domains		Desktop IT
$IF_{ICS.Comp.1}$	$IF_{AMI.Comp.1}$	✓		
$IF_{ICS.Comp.2}$	$IF_{AMI.Comp.2}$	✓		
$IF_{ICS.Comp.3}$				✓
	$IF_{AMI.Comp.3}$			
$IF_{ICS.Comp.4}$	$IF_{AMI.Comp.4}$	✓		
$IF_{ICS.Comp.5}$	$IF_{AMI.Comp.5}$	✓		✓
$IF_{ICS.Comp.6}$	$IF_{AMI.Comp.6}$		✓	
$IF_{ICS.Comp.7}$	$IF_{AMI.Comp.7}$	✓		(✓)
$IF_{ICS.Comp.8}$	$IF_{AMI.Comp.8}$	✓		
$IF_{ICS.Comp.9}$				
	$IF_{AMI.Comp.9}$			
$IF_{ICS.Arch.1}$	$IF_{AMI.Arch.1}$	✓		
$IF_{ICS.Arch.2}$	$IF_{AMI.Arch.2}$	✓		
$IF_{ICS.Arch.3}$	$IF_{AMI.Arch.3}$	✓		
$IF_{ICS.Arch.4}$	$IF_{AMI.Arch.4}$	✓		
$IF_{ICS.Arch.5}$	$IF_{AMI.Arch.5}$		✓	✓
$IF_{ICS.Arch.6}$				✓
$IF_{ICS.Arch.7}$	$IF_{AMI.Arch.6}$		✓	(✓)
$IF_{ICS.Arch.8}$	$IF_{AMI.Arch.7}$	✓		
	$IF_{AMI.Arch.8}$			
$IF_{ICS.Arch.9}$				
$IF_{ICS.Arch.10}$	$IF_{AMI.Arch.9}$	✓		
$IF_{ICS.Arch.11}$	$IF_{AMI.Arch.10}$	✓		
$IF_{ICS.Scen.1}$	$IF_{AMI.Scen.1}$	✓		
$IF_{ICS.Scen.2}$	$IF_{AMI.Scen.2}$	✓		
$IF_{ICS.Scen.3}$	$IF_{AMI.Scen.3}$		✓	
$IF_{ICS.Scen.4}$	$IF_{AMI.Scen.4}$		✓	
$IF_{ICS.Scen.5}$	$IF_{AMI.Scen.5}$	✓		
$IF_{ICS.Scen.6}$				✓
$IF_{ICS.Atk.1}$				✓
$IF_{ICS.Atk.2}$	$IF_{AMI.Atk.1}$	✓		
$IF_{ICS.Atk.3}$				✓

5.2 Impact of the Differences Between the Specific Domains on the [KDV15]-Model for the Forensic Process

This section discusses how the various *Influence Factors* presented in Section 5.1 affect the forensic as described in Section 3.1.2. Hence, this section provides the answer to **Research Question 2**.

The impact of these *Influence Factors* on the respective components of the [KDV15]-model for the forensic process is summarized in Table 5.2 and Table 5.3.

Please note, that an impact does not simply mean that performing a certain action is more difficult in a given domain in contrast to Desktop IT. This is the case with all the vendor-specific and non-standard interfaces and protocols which can be found in both investigated domains. These only make obtaining access and interpreting the specific traces more difficult. But this circumstance does not fundamentally change the forensic process. If however, accessing a certain *Data Stream* during *Post-Mortem Forensics* is practically impossible it would incur a substantial impact on the dynamics of a forensic process. Also, when the relationship between *Post-Mortem Forensics* and *Live Forensics* shifts due to some property of a given domain this would incur a fundamental impact.

These *Influence Factors* can be grouped together to describe particular consequences for the forensic process in the given domain. These consequences are referred to as *Forensic Process Consequences* (*FPC*). These *FPCs* are discussed in the following section and represent the answer to **Research Question 2**.

5.2.1 Impact of the Differences Between the Industrial Control System Domain and the Desktop IT domain on the [KDV15]-Model for the Forensic Process

Those *Influence Factors* which have a direct impact on the dynamics of the forensic process are shown in Table 5.2. The table does not include those factors which barely lead to an increased difficulty to perform **DG** or **DI** but those that fundamentally alter the dynamics of the forensic process.

These *Influence Factors* form clusters of factors which imply consequences for the forensic process. This results in six **FPCs** for the ICS domain:

- Reduced usefulness of *Non-Volatile Memory* ($\mathbf{FPC}_{ICS.1}$)
  The *Data Stream* of *Non-Volatile Memory* usually contains a limited number of forensic traces ($IF_{ICS.Comp.1}$). These traces are also harder to access during a *Post-Mortem Investigation* ($IF_{ICS.Comp.4}$ and $IF_{ICS.Scen.1}$).

- Increased importance of *Communication* ($\mathbf{FPC}_{ICS.2}$)
  PLCs rely on various inputs in order to perform their tasks. Attacks in the ICS domain often rely on inserting legitimate but malicious commands ($IF_{ICS.Atk.2}$). In general, the attacker also has to rely on logical access to a PLC in order to perform an attack ($IF_{ICS.Comp.3}$ and $IF_{ICS.Scen.1}$). In addition, the orderly nature of *Communication* in ICS supports the forensic process by greatly reducing the difficulty of **DI** and **DA** ($IF_{ICS.Arch.1}$, $IF_{ICS.Arch.10}$ and $IF_{ICS.Arch.11}$).

Table 5.2: Influence of the *Influence Factors* from ICS to the components of the [KDV15]-model for the forensic process

IF	*Data Streams*	*Structural Impact*	*Investigation Steps*	*Data Types*	*Methods*
Descriptor					
$\mathbf{IF}_{ICS.Comp.1}$	✓				
Low memory capacity					
$\mathbf{IF}_{ICS.Comp.2}$	✓				
Low computing power					
$\mathbf{IF}_{ICS.Comp.4}$	✓	✓			✓
Difficult physical access to Non-volatile Memory					
$\mathbf{IF}_{ICS.Comp.5}$	✓	✓			✓
Logical access to Non-volatile Memory and Volatile Memory possible					
$\mathbf{IF}_{ICS.Comp.6}$				✓	
Availability of Data Types highly dependent on the specific systems complexity					
$\mathbf{IF}_{ICS.Arch.1}$	✓		✓		
Established network hierarchies					
$\mathbf{IF}_{ICS.Arch.2}$		✓			
Low bandwidth					
$\mathbf{IF}_{ICS.Arch.9}$					✓
Presence of Historians within ICS					
$\mathbf{IF}_{ICS.Arch.10}$	✓			✓	
Dominance of ***DT8*** *in Communication in ICS*					
$\mathbf{IF}_{ICS.Arch.11}$	✓		✓		
Low and predictable traffic					
$\mathbf{IF}_{ICS.Scen.1}$	✓		✓		
Reduced component accessibility					
$\mathbf{IF}_{ICS.Scen.3}$			✓		
High availability requirements					
$\mathbf{IF}_{ICS.Scen.4}$			✓		
ICS control physical processes which might end in catastrophic events should the ICS be compromised					
$\mathbf{IF}_{ICS.Scen.5}$	✓	✓			
ICS requires time-critical responses from PLCs					
$\mathbf{IF}_{ICS.Scen.6}$	✓	✓			
Forensic evidence might contain intellectual property					
$\mathbf{IF}_{ICS.Atk.1}$	✓				
Diverse Attacks targeting ICS					
$\mathbf{IF}_{ICS.Atk.2}$	✓				
Focus on legitimate but malicious commands					
$\mathbf{IF}_{ICS.Atk.3}$			✓		
Close relationship between ICS and Desktop IT					

- Access to *Non-Volatile Memory* and *Volatile Memory* is usually performed relying on *Communication* ($\mathbf{FPC}_{ICS.3}$)
  The difficulties in accessing *Non-Volatile Memory* physically lead to a reliance on logical access to obtain forensic evidence ($IF_{ICS.Comp.4}$, $IF_{ICS.Comp.5}$ and $IF_{ICS.Scen.1}$). This logical access has to rely on *Communication* and hence causes a *Structural Impact* on *Communication.*

- Focus on *Live Forensics* in ICS ($\mathbf{FPC}_{ICS.4}$)
  The ICS domain carries various reasons not to shut down the ICS in order to perform a forensic investigation. This includes the cost to shut down an ICS ($IF_{ICS.Scen.3}$), the loss the logical access to *Non-Volatile Memory* and *Volatile Memory* ($FPC_{ICS.3}$) and the increased importance of *Communication* ($FPC_{ICS.2}$)

- *Live Forensics* might interfere with industrial objective ($\mathbf{FPC}_{ICS.5}$)
  While *Live Forensics* always causes some *Structural Impact* on the various *Data Streams* it might also interfere with achieving the industrial objective of the ICS since the employed components and communication technologies have little resources to spare ($IF_{ICS.Comp.2}$ and $IF_{ICS.Arch.2}$) and the need to fulfill at least soft real-time requirements ($IF_{ICS.Scen.5}$).

- Need to interpret **DT8** in ICS ($\mathbf{FPC}_{ICS.6}$)
  ICS control physical processes with a potential impact to health and safety. Hence, an understanding of these processes is necessary to understand whether a system should still be shut down ($IF_{ICS.Scen.4}$) or if legitimate but malicious control commands cause unwanted behavior in the physical process ($IF_{ICS.Atk.2}$). This is also relevant since **DT8** is widely available in ICS ($IF_{ICS.Arch.9}$ and $IF_{ICS.Arch.10}$).

  An important note to avoid confusion: this *Forensic Process Consequence* will be redefined during the revision of the core concepts of the forensic process model since the *Data Types* will be revisited during this process (see Section 5.3). Hence, the entry of this *FPC* in Section A.2 is altered to reflect this change.

$FPC_{ICS.1}$ and $FPC_{ICS.2}$ shift the relative importance between the various *Data Streams. Non-Volatile Memory* is less relevant in the ICS domain when compared to the Desktop IT domain. However, *Communication* has an increased relevance.

Live Forensics also has an increased relevance when compared to Desktop IT due to the shift in relevance of the respective *Data Streams* and the focus on *Live Forensics* ($FPC_{ICS.4}$, also influenced by $FPC_{ICS.3}$).

$FPC_{ICS.5}$ is an unique problem of the ICS domain and has to be addressed in some manner in the course of the forensic process as not to affect the industrial objective during the forensic process. A possible mitigation is the inclusion of mechanisms which does not rely on the resources of the PLCs and the network technologies but provide additional data in case of a forensic investigation. Such an inclusion of mechanisms has to be performed during **SP**.

$FPC_{ICS.6}$ shifts the relative importance of the various *Data Types* for the forensic investigation and highlights again, that the *Investigative Contexts* do not address the case where a physical process controlled by an ICs misbehaves in a satisfactory manner.

5.2.2 Impact of the Differences Between the Automotive IT Domain and the Desktop IT Domain on the [KDV15]-Model for the Forensic Process

Again, the **IFs** which have a direct impact on the dynamics of the forensic process are shown in Table 5.3.

These **IFs** form clusters of factors which imply consequences for the forensic process. This results in six **FPCs** for the Automotive IT domain:

- Reduced usefulness of *Non-Volatile Memory* ($\mathbf{FPC}_{AMI.1}$)
  The *Data Stream* of *Non-Volatile Memory* usually contains a limited number of forensic traces ($IF_{AMI.Comp.1}$ and $IF_{AMI.Comp.3}$). These traces are also harder to access in a *Post-Mortem Investigation* ($IF_{AMI.Comp.4}$ and $IF_{ICS.Scen.1}$).

- Increased importance of *Communication* ($\mathbf{FPC}_{AMI.2}$)
  ECUs rely on various inputs in order to perform their tasks. Attacks in the Automotive IT domain often rely on inserting legitimate but malicious commands ($IF_{AMI.Atk.1}$). In general, the attacker also has to rely on logical access to an ECU in order to perform an attack ($IF_{AMI.Comp.4}$, $IF_{AMI.Comp.5}$ and $IF_{AMI.Scen.1}$). In addition, the orderly nature of *Communication* in Automotive IT supports the forensic process by greatly reducing the difficulty of **DI** and **DA** ($IF_{AMI.Arch.1}$, $IF_{AMI.Arch.9}$ and $IF_{AMI.Arch.10}$). This is however problematic due to the lack of Origin Authenticity in Automotive IT networks ($IF_{AMI.Arch.8}$).

- Access to *Non-Volatile Memory* and *Volatile Memory* is usually performed relying on *Communication* $\mathbf{FPC}_{AMI.3}$)
  The difficulties in accessing *Non-Volatile Memory* physically lead to a reliance on logical access to obtain forensic evidence ($IF_{AMI.Comp.5}$ and $IF_{AMI.Scen.1}$). This logical access has to rely on *Communication* and hence causes a *Structural Impact* on *Communication.* However, ECUs usually support diagnostic requests. This is however problematic due to the lack of Origin Authenticity in Automotive IT networks ($IF_{AMI.Arch.8}$).

- Focus on *Live Forensics* in Automotive IT ($\mathbf{FPC}_{AMI.4}$)
  A vehicle cannot simply be stopped during the driving task ($IF_{AMI.Scen.3}$). However, it can be stopped without incurring great financial loss.

  Due to the reliance on logical access to *Non-Volatile Memory* and *Volatile Memory* ($FPC_{AMI.3}$) and the increased importance of *Communication* ($FPC_{AMI.2}$), performing *Live Forensics* seems a good choice to perform forensic investigations in Automotive IT. This is supported by the fact that shutting down a vehicle does not shut down the respective ECUs ($IF_{AMI.Comp.9}$).

- *Live Forensics* might interfere with the driving task ($\mathbf{FPC}_{AMI.5}$)
  While *Live Forensics* always causes some *Structural Impact* on the various *Data Streams* it might also interfere with performing the driving task since the employed components and communication technologies have little resources to spare ($IF_{AMI.Comp.2}$ and $IF_{AMI.Arch.2}$) and the need to fulfill at least soft real-time requirements ($IF_{AMI.Scen.5}$).

Table 5.3: Influence of the *Influence Factors* from Automotive IT to the components of the [KDV15]-model for the forensic process

IF	*Data Streams*	*Structural Impact*	*Investigation Steps*	*Data Types*	*Methods*
Descriptor					
$\mathbf{IF}_{AMI.Comp.1}$	✓				
Low memory capacity					
$\mathbf{IF}_{AMI.Comp.2}$	✓				
Low computing power					
$\mathbf{IF}_{AMI.Comp.3}$	✓				
Memory wear reduces reliance on Non-volatile Memory					
$\mathbf{IF}_{AMI.Comp.4}$	✓	✓			✓
Difficult Physical Access to Non-volatile Memory					
$\mathbf{IF}_{AMI.Comp.5}$	✓	✓			✓
Logical Access to Non-volatile Memory and Volatile Memory possible					
$\mathbf{IF}_{AMI.Comp.6}$				✓	
Reduced availability of Data Types					
$\mathbf{IF}_{AMI.Comp.9}$	✓		✓		
ECUs are usually not shut down					
$\mathbf{IF}_{AMI.Arch.1}$	✓		✓		
Established and simple network hierarchies					
$\mathbf{IF}_{AMI.Arch.2}$		✓			
Low bandwidth					
$\mathbf{IF}_{AMI.Arch.8}$					
Doubtful Origin Authenticity in CAN networks					
$\mathbf{IF}_{AMI.Arch.9}$	✓			✓	
*Dominance of **DT8** in Communication in Automotive IT*					
$\mathbf{IF}_{AMI.Arch.10}$	✓		✓		
Low and predictable traffic					
$\mathbf{IF}_{AMI.Scen.1}$	✓		✓		
Reduced component accessibility					
$\mathbf{IF}_{AMI.Scen.3}$			✓		
Medium availability requirements					
$\mathbf{IF}_{AMI.Scen.4}$			✓		
Automotive IT control physical processes which might cause risk to limb and life should the Automotive IT be compromised					
$\mathbf{IF}_{AMI.Scen.5}$	✓				
Automotive IT must fulfill (soft) real-time requirements					
$\mathbf{IF}_{AMI.Atk.1}$	✓				
Focus on legitimate but malicious commands					

- Need to interpret **DT8** in Automotive IT ($\mathbf{FPC}_{AMI.6}$)
  Automotive IT controls driving tasks with a potential impact to health and safety. Hence, an understanding of the driving task is important to understand if the vehicle is behaving in a way that might cause an accident ($IF_{AMI.Scen.4}$). This is also relevant since **DT8** is widely available in Automotive IT ($IF_{AMI.Arch.9}$).

  An important note to avoid confusion: this *Forensic Process Consequence* will be redefined during the revision of the core concepts of the forensic process model since the *Data Types* will be revisited during this process (see Section 5.3). Hence, the entry of this *FPC* in Section A.2 is altered to reflect this change.

$FPC_{AMI.1}$ and $FPC_{AMI.2}$ shift the relative importance between the various *Data Streams*. *Non-Volatile Memory* is less relevant in the Automotive IT domain when compared to the Desktop IT domain. However, *Communication* has an increased relevance.

Live Forensics also has a greater relevance when compared to Desktop IT due to this shift in relevance of the respective *Data Streams* and the focus on *Live Forensics* ($FPC_{AMI.4}$, also influenced by $FPC_{AMI.3}$). *Live Forensics* in the Automotive IT domain can be considered as a kind of *Delayed Live Forensics* due to $IF_{AMI.Comp.9}$.

$FPC_{AMI.5}$ is an unique problem of the Automotive IT domain and has to be addressed in some manner in the course of the forensic process as not to affect the industrial objective during the forensic process. A possible mitigation is the inclusion of mechanisms which does not rely on the resources of the PLCs and the network technologies but provide additional data in case of a forensic investigation. This inclusion of mechanisms has to be performed during **SP**.

$FPC_{AMI.6}$ shifts the relative importance of the various *Data Types* for the forensic investigation and highlights again, that the *Investigative Contexts* do not address the case where a driving task controlled by Automotive IT misbehaves in a satisfactory manner.

5.3 Revisiting the [KDV15]-Model

Which *Influence Factors* imply which *Forensic Process Consequences* has been discussed in Section 5.2. This discussion answered **Research Question 2**.

The following section discusses the resulting necessary adaptations to the [KDV15]-model of the forensic process (as introduced in Section 3.1.2) to the ICS and Automotive IT domains (in short referred to as *investigated domains*). In doing so it provides the answer to **Research Question 3**.

Before this adaptation starts, it is important to point out that the investigated domains are indeed quite similar, as the comparison in Section 5.1.3 shows. They invoke similar *Forensic Process Consequences*, as shown in Table 5.4. Indeed, the *Forensic Process Consequences* for the investigated domains seems to be alike. But this is true only on a high abstraction level. While the properties of both domains touch on the complex relationship between *Live Forensics* and *Post-Mortem Forensics* ($FPC_{ICS.4}$ or $FPC_{AMI.4}$) respectively the result on how this behavior shifts is not the same in both domains.

If the explanation of a certain adaptation relies on a *Forensic Process Consequence* related to both investigated domains in the same manner, the notation of $FPC_{numeral}$ will be used. If

Table 5.4: *Forensic Process Consequences* from ICS and Automotive IT in regard to the [KDV15]-model

Forensic Process Consequences	
in ICS	*in Automotive IT*
FPC$_{ICS.1}$	**FPC**$_{AMI.1}$
Reduced usefulness of *Non-Volatile Memory*	
FPC$_{ICS.2}$	**FPC**$_{AMI.2}$
Increased importance of *Communication*	
FPC$_{ICS.3}$	**FPC**$_{AMI.3}$
Access to *Non-Volatile Memory* and *Volatile Memory* is usually performed relying on *Communication*	
FPC$_{ICS.4}$	**FPC**$_{AMI.4}$
Focus on *Live Forensics*	
FPC$_{ICS.5}$	**FPC**$_{AMI.5}$
Live Forensics might interfere with the physical process	
FPC$_{ICS.6}$	**FPC**$_{AMI.6}$
Need to interpret **DT8**	

the explanation relies on the specifics of an *Forensic Process Consequence* from one of these two domains the original notation is used.

This section is structured along the lines of the components of the [KDV15]-model for the forensic process. Hence, *Investigation Steps*, *Data Types* and *Classes of Methods* will be discussed. A review on other concepts included in the [KDV15]-model (namely *Data Streams* and *Structural Impact*) follows this discussion. This section is concluded by a discussion of concepts not formerly part of the [KDV15]-model.

5.3.1 *Step 2.1* Revisiting *Investigation Steps*

A discussion of the impact of the properties of the ICS domain on the *Investigation Steps* was already provided in [**AHKD19**]. This thesis continues this discussion by providing a comprehensive analysis of how the ICS and Automotive IT domain influence the forensic process (*Influence Factors*) and which consequences this has for the forensic process (*Forensic Process Consequences*).

There are some *Forensic Process Consequences* which directly impact the *Investigation Steps*.

5.3.1.1 *Forensic Process Consequences* Impacting *Strategic Preparation*

Strategic Preparation is of a increased importance in the investigated domains. There are two main reasons for this. The first one is the lack of usable data for the forensic investigation in *Non-volatile Memory* and *Volatile Memory* (FPC_1). The second one is interlinked with the first one and is the increased importance of *Communication* (FPC_2).

The lack of forensic evidence in the *Non-volatile Memory* is caused by the low memory available in the specific components. This increases the importance of a **SP** in order to install methods which can provide forensic evidence in the case a forensic investigation is required (FPC_1).

Communication is of increased importance in these two investigated domains (FPC_1). As shown in Section 3.1.2.5, *Communication* can only be observed (and hence gathered) at the moment it is happening. The installation of means to access *Communication* efficiently in case of an incident reduces the time required to obtain access to the *Communication* during **DG** - and hence reduces the loss of valuable forensic evidence.

In addition, *Communication* in the investigated domains can be analyzed more easily due to a more orderly behavior. While this is of obvious use for anomaly detection in order to detect incidents, it could also benefit forensic investigations by implementing measures which collect information about notable events in forensic sound manner during **SP**.

Collecting baseline data during an **SP** also supports the investigation into *Communication* behavior.

While these factors increase the importance of **SP** in the investigated domains, they do not necessitate any change to the definition of **SP**. This *Investigation Step* is highly relevant in the investigated domains.

5.3.1.2 *Forensic Process Consequences* Impacting *Operational Preparation*

Operational Preparation in the investigated domains includes the decision whether to perform *Live Forensics* or *Post-Mortem Forensics.* The specifics of the investigated domains cause some *Forensic Process Consequences* which affect this decision.

For once, there is a notable focus towards *Live Forensics* in both of these domains (FPC_4). The reasons for this differ slightly in both investigated domains. However, they share the problematic access to data during *Post-Mortem Investigations* and have to rely on logical access to the *Non-Volatile* and *Volatile Memory* (FPC_3). In the case of ICS, deactivating the ICS in question incurs high monetary costs or comes with other constraints related to the physical process ($FPC_{ICS.4}$). In the case of Automotive IT ($FPC_{AMI.4}$), *Live Forensics* can also be interpreted as *Delayed Live Forensics* (see Section 4.1.3.1). In this case, the investigation does not share the characteristics of a *Post-Mortem Investigation.*

There is also the risk that the forensic investigation might interfere with the physical process the respective control systems are controlling (FPC_5). This is due to the resource limitations and the (soft) real-time requirements of the investigated domains. This aspect is addressed by the *Criticality Map* (see Section 5.3.5.2).

The close relationship between ICS and Desktop IT might lead to the decision to perform an additional forensic investigation into a Desktop IT system since it might have been used to gain access to the ICS in the first place ($IF_{ICS.Atk.3}$).

A factor in the decision on whether to shut down a system (which would implicate solely relying on *Post-Mortem Forensics* in any domain but the Automotive IT domain) should factor in the risk the system (or rather the physical processes controlled by the system) poses to its environment. In order to ascertain this risk, an understanding of the physical process in general and the current state of the physical process is necessary (FPC_6). Hence, an investigator who performs the **OP** in one of the investigated domains needs domain-specific knowledge of the physical processes controlled by these systems.

This deep knowledge of the physical process is also necessary to identify the data traces which are relevant during **DG**.

While these factors affect the decisions made during **OP** and pose additional requirements to the forensic investigator they do not necessitate any change to the definition of **OP**. This *Investigation Step* is relevant in the investigated domains.

5.3.1.3 *Forensic Process Consequences* Impacting *Data Gathering*, *Data Investigation* and *Data Analysis*

The domain analysis in Chapter 4 mainly focused on the question of the availability, accessibility and interpretability of forensics traces in the three *Data Streams* in three respective domains. As shown in the discussions of the respective domains these forensic traces are still available, accessible and interpretable, although with some attached difficulties. The various *Influence Factors* (see Section 5.1) describe these difficulties and shifted focus.

These challenges do not fundamentally change anything about the tasks performed during *Data Gathering*, *Data Investigation* and *Data Analysis*. Although different tools are used than in the Desktop IT domain the respective tools still serve the same overall purpose - to access or interpret forensic traces.

Hence, these *Investigation Steps* are relevant in the investigated domains and do not necessitate any alteration.

5.3.1.4 *Forensic Process Consequences* Impacting *Documentation*

Documentation is not well-served in the investigated domains. With barely any tool geared towards the use in forensic scenarios and a high amount of interfaces and protocols which are proprietary and closed source, *Documentation* is problematic.

This does not make *Documentation* in the investigated domains obsolete. Indeed it shows the demand for additional tools geared towards the use in forensic scenarios. How such tools should be designed to reach forensic demands is discussed in Section D.4.

However, the revision of the forensic process model shall be used to clarify the definition used for **DO** to include *Process Accompanying Documentation* and *Final Documentation*. Hence, the following alteration to the definition for **DO** is made:

> **Documentation (DO)**:
> measures for the detailed documentation of the proceedings (*Process Accompanying Documentation*) and for the compilation of a report on the incident (*Final Documentation*)

5.3.1.5 *Forensic Process Consequences* Impacting the Overall Structure of the Forensic Process

Communication plays a major role in security incidents in the investigated domains and the investigations into these incidents (FPC_2). Attacks in the investigated domains often include legitimate but malicious commands ($IF_{ICS.Atk.2}$ and $IF_{AMI.Atk.1}$). Hence, an investigation will usually discern whether a certain unwanted behavior is caused by such malicious commands. When this is the case, an investigation into the origin of these commands might be started.

In the current version of the [KDV15]-model this is problematic due to the overall structure of the forensic process.

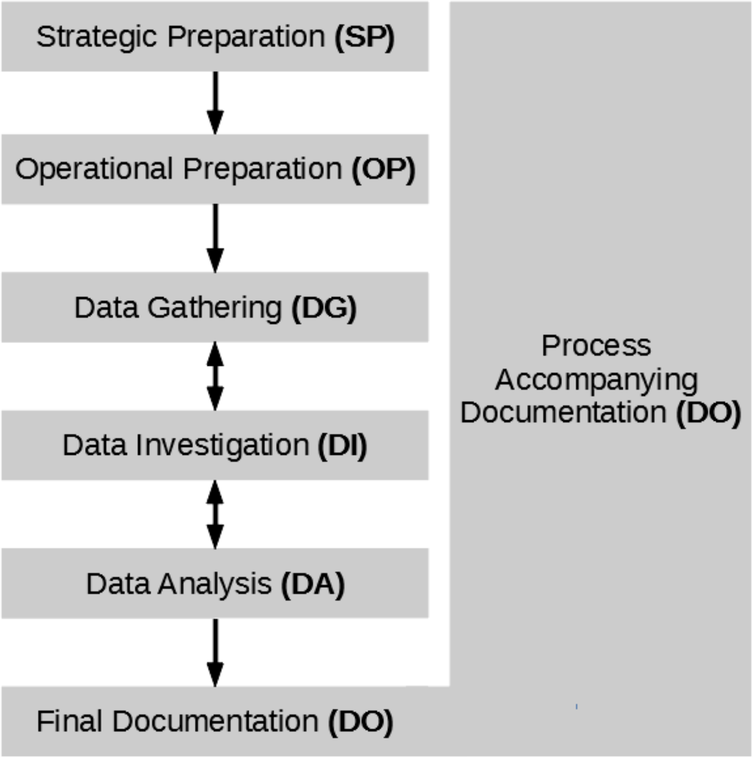

Strategic Preparation **(SP)**
Operational Preparation **(OP)**
Data Gathering **(DG)**
Data Investigation **(DI)**
Data Analysis **(DA)**
Final Documentation **(DO)**
Process Accompanying Documentation **(DO)**

Figure 5.1: Order of the *Investigation Steps* in [BSI11]

[KDV15] does not contain any information about the overall structure of the forensic process and the relationship between the various phases. Hence, a flat structure with each step following the next has to be assumed. This structure is illustrated in Figure 3.3. A slightly different structure is used in [BSI11]. The definition of the *Investigation Steps* in [BSI11] states that **DA** and **DI** can lead back to the preceding *Investigation Step* (**DI** and **DG** respectively). The resulting structure is shown in Figure 5.1. However, it does not include the possibility to go back towards **OP** once the investigation has passed this *Investigation Step*.

None of these structures is optimal for the investigated domains. The decision which data sources are investigated is made during **OP**. Also, the decision whether the system needs to be shut down is made during **OP**. Both decisions might be revised once the presences of malicious but legitimate commands within the investigated *Communication* is confirmed. Hence, the forensic process should be able to return to the **OP**. This might be necessary when the malicious commands originate from another network within the overall system hierarchy. In the case of an ICS, this might be the attached Desktop IT ($IF_{ICS.Atk.2}$). These different networks might imply different factors for the consideration of *Live Forensics* or *Post-Mortem Forensics* which makes a new decision process in another **OP** necessary.

It is necessary to retain the possibility to go from **DA** to **DI** in case the analysis of a certain piece of evidence identifies the potential importance of another piece of evidence which is then in turn investigated in more detail. The same applies for **DI** and **DG**. These loops are performed on a micro-level of the overall structure. The need to perform another **OP** happens on the macro-level of the overall process. Here the decision whether to investigate additional

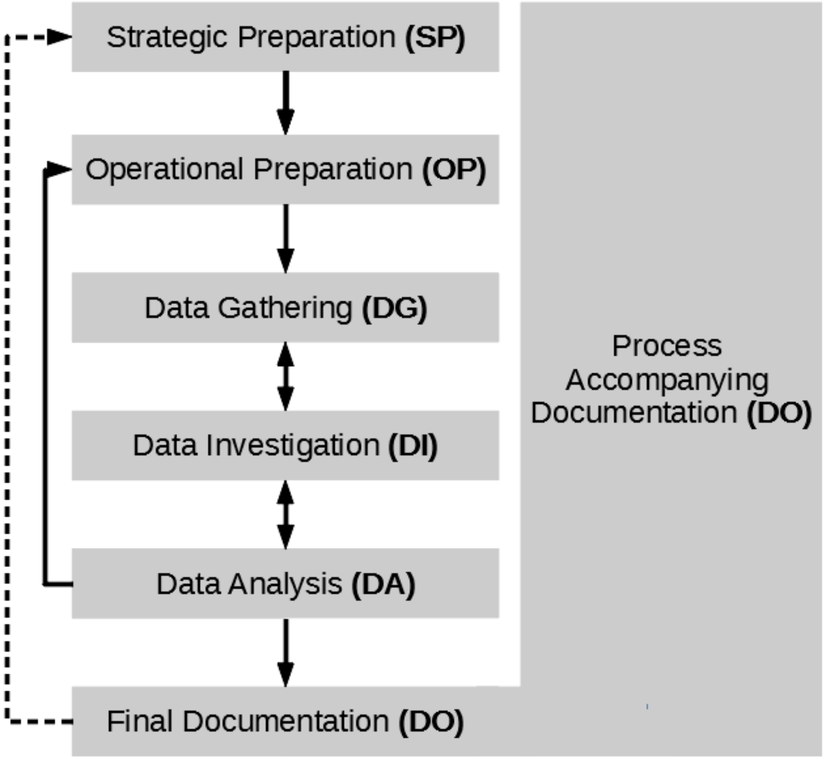

Strategic Preparation (SP)
Operational Preparation (OP)
Data Gathering (DG)
Data Investigation (DI)
Data Analysis (DA)
Final Documentation (DO)
Process Accompanying Documentation (DO)

Figure 5.2: Revised order of the *Investigation Steps*

networks and components is made. This is only possible after the **DA**. Hence, the loop back to **OP** originates from **DA**.

In addition, once the overall investigation is done and the **DO** finished, measures how to improve **SP** can be identified. Hence a feedback loop from **DO** to **SP** is added. Note that improving any capability to perform a given *Investigation Step* before an incident triggers and investigation is part of the **SP**.

The revised overall structure of the forensic structure is shown in Figure 5.2.

5.3.1.6 Summary of Required Changes to *Investigations Steps*

The *Investigations Steps* fulfill the same tasks in the investigated domains as they do in Desktop IT. There is only a shift on the relative importance of the respective *Investigations Steps*.

SP has an increased importance since the lack of an **SP** seriously limits the amount of available forensic traces during a potential investigation or at least delays the beginning of any meaningful **DG** performed on the *Communication* which is an important *Data Stream* in the investigated domains.

OP is affected by considerations not necessary in the Desktop IT domain and requires specialized knowledge about the underlying physical processes in order to be performed in a satisfactory manner.

DG, **DI** and **DA** use different tools in each domain but retain the same tasks.

DO is badly addressed in the investigated domains. However, its tasks are of no less importance in the investigated domains. The definition has been revised to include *Process Accompanying Documentation* and *Final Documentation*.

The overall structure and relationship between these *Investigation Steps* has been revised to include a loop back from **DA** to **OP** in order to denote that a forensic investigation might be extended to include additional networks after it has been confirmed that an attack is occurring. The revised structure for the forensic process can be seen in Figure 5.2.

5.3.2 *Step 2.2* Revisiting *Data Types*

How the *Data Types* can be interpreted in the ICS domain is discussed in [**AHKD19**]. Some of the *Data Types* received clarifications due to vagueness or ambiguity of terms in regards to the ICS domain. These alterations are updated here to also include the Automotive IT domain. In addition, **DT8** received a major overhaul in [**AHKD19**]. This is supported by the increased importance of **DT8** and the slightly different notion this data has in the investigated domains (FPC_3).

This section discusses alterations to the definitions for the various *Data Types* as presented in Section 3.1.2.2.

5.3.2.1 *hardware data* (DT1)

In the context of the investigated domains, **DT1** describes specific hardware information, like model-numbers or information about the computing power, memory size and attached interfaces. This *Data Type* is relevant in the investigated domains.

The wording of this definition can be cleaned up. The terms of 'operating system' and 'application' should be avoided since they are effectively a surrogate for 'software of any kind'. Furthermore, following the example set by [Kil20] this *Data Type* changes the ordering with **DT2**. Hence, the new definition for *hardware data* is suggested as:

> **hardware data (DT2)**:
> *Data in a computing unit which is not, or only in a limited way, influenced by software.*

5.3.2.2 *raw data* (DT2)

DT2 describes the physical representation of any data in any *Data Stream*. As such this *Data Type* is relevant in the investigated domains. The redefinition of this *Data Type* only includes a minor rewording and a reordering. Following the consideration presented in [Kil20], all other *Data Types* can be extracted from this *Data Type*. Hence, it assumes the descriptor **DT1**:

> **raw data (DT1)**:
> *A sequence of bits within the Data Streams of a computing systems not (yet) interpreted.*

5.3.2.3 *details about data* (DT3)

DT3 is data which provides information about other data. This is the case with protocol overhead in *Communication* or file systems in *Non-volatile Memory*. Both exist in the investigated domains. Hence, this *Data Type* is relevant for the investigated domains.

[**AHKD19**] suggests a rewording in order to clarify the relationship between data and its meta date. This suggestion is adopted here, with a minor fix for spelling:

> **details about data (DT3)**:
> *Data added to other data, stored within the annotated chunk of data or externally*

5.3.2.4 *configuration data* (DT4)

DT4 describes the configuration of a computing unit in all regards besides its communication behavior. Such a configuration is also relevant in the investigated domains.

The terms of 'operating system' and 'application' should be avoided since they are effectively a surrogate for 'software of any kind'. The alteration proposed in [**AHKD19**] to clean up the structure of the definition is adopted. Hence, the following definition for *hardware data* is suggested:

> **configuration data (DT4)**:
> *Data which can be changed by software and which modifies the behavior of software and hardware, excluding the communication behavior*

5.3.2.5 *network configuration data* (DT5)

DT5 describes the configuration of a computing unit in regards to its communication behavior. In the investigated domains this would included network interfaces as well as the connections to attached actors and sensors. Hence, this *Data Type* is relevant for the investigated domains.

Here, no major adaptations are required:

> **network configuration data (DT5)**:
> *Data that modifies system behavior with regards to communication*

5.3.2.6 *process data* (DT6)

DT6 describes data about running computing process. This data might be relevant during a forensic investigation within the investigated domains.

As [**AHKD19**] discusses, the term of 'process' in the ICS domain usually describes the physical process controlled by the ICS. This ambiguity is detrimental, and hence the suggestion for a redefinition as proposed in [**AHKD19**] is adopted:

> **process data (DT6)**:
> *data about a running software process within a computing unit*

5.3.2.7 *session data* (DT7)

The original definition of *session data* refers to the data collected during a session. However, it is not entirely clear what a session constitutes, especially in the context of the investigated domains. [**AHKD19**] discusses this topic and provides the following considerations:

> *„It relies on an unclear definition of a session. From the perspective of a forensic investigation, a session should include all processes and their communication within the same scope and time frame. In an ICS context, this describes a snapshot of the sensor readings and actor controls stored by the plant historian within a specific time frame. For production processes session data can also relate to the data gathered during the production of one specific item. Hence, we propose: data collected by a system during a session, which consist of a number of processes with the same scope and time frame“*

In the sense of these considerations, a session in Automotive IT would constitute a driving task. From the point of view of a forensic investigator such data might be useful in **IC2**. If the investigator is tasked with investigating the use of a vehicle in a crime (for example as a getaway car), data about a specific driving task would be useful. The same would be applicable if an ICS is used in a manufacturing process producing illegal goods.

Hence, the suggestion provided by [**AHKD19**] is adopted:

> **session data** (**DT7**):
> *data collected by a system during a session, which consist of a number of processes with the same scope and time frame*

5.3.2.8 *user data* (DT8)

DT8 describes contents created, edited or consumed by the user. This is not applicable to the investigated domains (with limited exceptions like maps or audio files presented by In-Vehicle-Infotainment systems). However, the investigated domains still have an abundance of data about the controlled physical processes which was coined as **DT8** for lack of a better fitting *Data Type*.

[**AHKD19**] identifies additional issues with the current definition of **DT8** when used in the ICS domain:

> *„user data is defined as content created, edited or consumed by the user including media. This data type represents the data linked to the key functionality (or purpose) of a system in question. In Desktop IT, this might be handling of office files or the creation of images. In an ICS context, this would be the physical process itself. However, from an forensic point of view, the means of such file handling (or processing) are different. An executable will be analyzed in a different way and with a different scope than a media file. An executable file might also be a media file in another context, e.g. an executable be executed on a PLC and edited on a Desktop PC might be seen (and analyzed) as 'media file' in one context and an*

> *'executable' in another case. Hence, we propose the creation of two distinct data types. These two data types are linked to performing the key functionality of the system in question. One of these data types contains the media, which is created, edited, consumed or processed by the user (the ends). The other contains the applications used to perform this creating, editing or processing (the means). The media might be anything fulfilling the purpose of the system - editing office files, developing software by editing source code, displaying video files. In an ICS, this media would represent the programming of physical processes performed on a workstation (with the corresponding means being the environment used to perform this programming and the execution). The means to process this media might change over time but the nature of this media does not. We propose for the investigated system:*
>
> - *DT8* ***application data*** *is data representing functions needed to create, edit, consume or process content relied to the key functionality of the system.*
> - *DT9* ***functional data*** *is data content created, edited, consumed or processed as the key functionality of the system.“*

The definition suggested for **DT9** in [**AHKD19**] describes the data about the physical process so widely available and relevant in the investigated domains well ($IF_{ICS.Arch.9}$, $IF_{ICS.Arch.10}$ and $IF_{AMI.Arch.9}$). The definition also retains the use in the Desktop IT domain since it is not so specific to only cover the investigated domains. Hence, this proposed definition is adopted, including the split of **DT8** into two new *Data Types*.

application data (DT8):
data representing functions needed to create, edit, consume or process content relied to the key functionality of the system

functional data (DT9):
data content created, edited, consumed or processed as the key functionality of the system

5.3.2.9 Summary of Required Changes to *Data Types*

In general, all *Data Types* are relevant in the investigated domains ($IF_{ICS.Comp.6}$, $IF_{ICS.Arch.9}$, $IF_{ICS.Arch.10}$, $IF_{AMI.Comp.6}$ and $IF_{AMI.Arch.9}$). Although, some changes to the specific definitions and the inclusion of a new *Data Type* are necessary.

A reordering of the *Data Types* is motivated by the considerations presented in [Kil20]. Since all other *Data Types* can be extracted from *raw data* the designation **DT1** is assigned to it in the following with *hardware data* being designated as **DT2**.

DT2 (now *hardware data*), **DT3** and **DT4** require some notable adaptation to fit the investigated domains. **DT6** requires a new definition due to some ambiguity of the terms used in its definition in the ICS domain.

DT7 faces a major revision due to the need to understand what a sessions constitutes in the investigate domains. The new definition is geared towards the use in **IC2**.

DT8 is altered and split into two *Data Types*. The definition is altered to take the key functionality of a system into consideration in order to represent the investigated domains and Desktop IT. The two new *Data Types* describe *Application Data* and *Functional Data*, which includes data about the physical process.

5.3.3 *Step 2.3* Revisiting *Classes of Methods*

The *Classes of Methods* describe categories of tools or methods usable during the forensic investigation. The six categories represents methods available in Desktop IT and are described in Section 3.1.2.3. This section discusses whether these *Classes of Methods* are applicable to the investigated domains and whether they suffice to describe the domain.

This section is structured along the lines of the various *Classes of Methods*.

5.3.3.1 *Operating system* (OS)

In the investigated domains, operating systems of varying complexity are used in the specific computing units ($IF_{ICS.Comp.3}$). Some are bare execution environments for an user program which controls the physical process. Other examples include far more complex operating systems, like GNU/Linux in SIMATIC PLCs ([Ber11]; see Section 4.1.2.5). Sometimes, the separation between **OS** and **ITA** is not entirely clear, if the execution environment is only able to execute a hardwired program. In this case, it could be said that **OS** and **ITA** are merged into one. Hence, a discussion on the **OS** has to include a discussion on **ITA**.

The definition of (**OS**) and **ITA** provided in Section 3.1.2.3 (from [KDV15]) is:

- ***„Operating system (OS)***
  methods provided by the operating system such as maintaining process lists, log file creation [..]
- ***IT application (ITA)***
  methods provided by IT-Applications that are operated by the user, in addition to their main functionality they also provide forensic methods such as log keeping

In essence, the primary difference between those terms is that **ITA** is operated by the user. The addendum that **ITA** provides this forensic evidence in addition to its main functionality is irrelevant since the main function of an **OS** is not to provide forensic traces either - it does so in addition to its main function. In the investigated domains there is not necessarily an user in the sense of an user of Desktop IT. However, there is an industrial objective (in case of ICS) or a driving task (in case of Automotive IT). One could argue, that the dividing element is, that the functionality of the **ITA** is to implement this industrial objective (or driving task). This is different from the task of an **OS**. Based on the definition provided by [Tan07], the **OS** is tasked with managing the access to the hardware components.

In practical terms it must be considered that the components in the investigated domains generally have diagnostic functions accessible via various interfaces (FPC_3). These diagnostic functions provide access to various *Data Types*. It is not necessarily transparent whether the responses to these queries originate from **OS** or **ITA**. Here, taking the use of the specific data within the queried system into account might be a useful suggestions.

DT1 to **DT6** describe *Data Types* with some relevance to the **OS** in order to perform its function. **DT7**, **DT8** and **DT9** refer to the use case of the system - the media content, industrial objective or driving task. These three *Data Types* refer to what the **ITA** within the component is tasked to control.

Even if the **OS** and **ITA** are too deeply intertwined separating the categories in this manner would still provide benefits for the forensic process since the Data Types provided by **OS** are, generally speaking, useful in **IC1** and those provided by **ITA** are, generally speaking, relevant to **IC2**.

However, this does not open the way to a better definition of the *Class of Method* **OS**, besides removing the examples. It just clarifies what this entails in the investigated domain. From the sheer reliance on diagnostic requests, it should be clear that **OS** is highly relevant in the investigated domains. The revised definition for **OS** is:

> **Operating System** (**OS**):
> methods provided by the operating system

5.3.3.2 *File system* (FS)

The investigated domains do not necessarily include file systems (see Section 4.1.2.1 and Section 4.1.3.2). In addition, the usefulness of *Non-volatile Memory* is reduced in the investigated domains (FPC_1). However, where file systems are present, they still provide useful methods to the forensic process. This lets **FS** retain some relevance for the investigated domains. Also, **FS** must be maintained in the revised forensic process model to also address the Desktop IT domain.

The only change to the definition is the dropping of the examples:

> **File System** (**FS**):
> methods provided by the file system

5.3.3.3 *Explicit means of intrusion detection* (EMID)

The discussion in Chapter 4 does not include many examples of **EMID**. However, the reduced usefulness of *Non-volatile Memory* (FPC_1) and the increased importance of *Communication* (FPC_2) in the investigated domains lead to an increased relevance of **EMID**. As shown in Section 3.1.2.5, *Communication* can only be observed (and hence gathered) at the moment it occurs. Using **EMID** to discern which *Communication* is worth gathering supports this process. Obviously, **EMID** have to be implemented during **SP**. This is also beneficial due to the increased importance of **SP** in the investigated domains (see Section 5.3.1.1).

The monitoring of networks (*Communication*) is described as an important tool in ICS security by [KL15]. Such Monitoring could also provide use for the forensic process if it includes logging mechanics adhering to the requirements of forensic use. [NIS15] also demands the inclusion of such monitoring mechanisms in ICS:

> *„The security architecture of an ICS must also incorporate mechanisms to monitor, log, and audit activities occurring on various systems and networks. Monitoring, logging and auditing activities are imperative to understanding the current state of the ICS, validating the system is operating as intended, and that no policy violations or cyber incidents have hindered the operation of the system. “*

This conclusion is further supported by [BSI19].

The practical side of this has been explored in [**AHNH20**] and [Hie19]. [**AHNH20**] explores the potential benefits of including a SIEM (Security Information and Event Management; see [BMZ14]) into the ICS part of the overall infrastructure of a nuclear power plant (NPP). This aims at improving detection, incident response on and forensic event reconstruction of cyber attacks in this environment. Such a SIEM would monitor *Communication*. The proposal aims at transferring knowledge from the Desktop IT domain (in which SIEMs are more common) to the ICS domain. In [Hie19] the usability of the ELK stack[1] to support forensic investigations into ICS is explored. This bachelor thesis in German Language attests some forensic usefulness to this solution but notes the lack of some necessary data converters.

In the Automotive domain, this topic was discussed by [Hop14] who suggested the inclusion of an IDS (Intrusion Detection System) in Automotive IT which monitors the *Communication* to identify incidents. An example for another tool adaptable for such a purpose was presented in Section 4.2.3.4. The tool *c0f Fingerprinting Tool*[2] can use fingerprints to identify irregularities in the *Communication*. With some adaptation, it could also log these irregularities and provide evidence for the forensic process.

This discussion shows a disparity between ideal and the current implementation. However, it does not necessitate any alteration to the current definition of **EMID**:

> **Explicit Means of Intrusion Detection (EMID)**:
> methods provided by additional software with the characteristic of being executed autonomous on a routine basis and without a suspicion of an incident

5.3.3.4 *IT application* (ITA)

The relationship between **OS** and **ITA** has already been discussed in Section 5.3.3.1. This discussion also entails that the investigated domain do not necessary have an user in the same sense in Desktop IT. However, the applications processed on the computing units in these domains still fulfill a certain task to achieve an objective. In the case of ICS, this is the industrial objective. In the case of Automotive IT, this is the driving task. Hence, the definition requires a replacement of the term user with some term that encompasses these three possibilities.

The redefinition of **DT8** and **DT9** can serve as an inspiration here (see Section 5.3.2.8). Here, the 'key functionality' of a system is mentioned. This covers the industrial objective, the driving task or whatever action a user wants to perform in Desktop IT.

In addition, the reference to the fact that forensics is only a secondary function of the **ITA** is dropped since it is not necessary. Providing forensic use is not the primary function of **OS**, **FS** or **EMID** either. Hence, the following definition is proposed:

[1] https://www.elastic.co/, 26/05/2020
[2] https://github.com/zombieCraig/c0f/, 24/05/2020

> **IT Application (ITA):**
> methods provided by the functions that fulfill the key functionality of the system

Based on this definition, **ITA** is relevant for the investigated domains and represents the user programs. One clear example for an **ITA** in the ICS domain is a Historian which provides **DT9** which is in line with the observations made in Section 5.3.3.1 ($IF_{ICS.Arch.9}$).

5.3.3.5 *Scaling of methods for evidence gathering* (SMG)

Many of the measures discussed in Chapter 4 fall under the category of **SMG**. Hence, the relevance of this *Class of Methods* for the investigated domains is obvious. The definition of **SMG** only requires some minor fixes to the wording and a generalization of 'CPU demands'. Hence, the new definition stands as:

> **Scaling of Methods for Evidence Gathering (SMG):**
> methods to collect additional evidence if a suspicion is raised but unsuited for routine usage in a production environment (due to reasons such as high false positives or high resource demand)

5.3.3.6 *Data processing and evaluation* (DPE)

This *Class of Methods* combines two types of tools. One type are tools supporting *Process Accompanying Documentation* and *Final Documentation* (see Section 3.1.2.1). The other type are those tools which are designed for a forensic use and hence address the requirements for a forensics tool (see Section D.4).

There is a lack of both types of tools in the investigated domains ($IF_{ICS.Arch.8}$ and $IF_{AMI.Comp.8}$).

Especially tailored solutions that integrate forensic case management with analytic functions and organize evidence automatically are missing. While it is possible to use toolkits from the Desktop IT domain, these toolkits miss domain-specific knowledge necessary for an efficient use.

For the general forensic process, some tools from the Desktop IT domain can be used since many of the methods used to perform **DG**, **DI** and **DA** in the investigated domains rely on the use of diagnostic request to the specific computing units performed by Desktop IT applications (FPC_3 and FPC_4). The external measures usually used to provide integrity and authenticity for the results of **DG**, **DI** and **DA** can be used in this case. The same applies for the documentation of action in tools which rely on a CLI (Command Line Interface) for user interaction.

Although **DPE** lacks dedicated tools for the use in the investigated domains, it is nonetheless relevant. There is no reason to redefine **DPE**, hence, the original definition is in continued use:

> **Data Processing and Evaluation (DPE):**
> methods which support a detailed forensic investigation, display, processing and documentation)

5.3.3.7 Summary of Required Changes to *Classes of Methods*

The relationship between **OS** and **ITA** has been discussed and led to a redefinition of both *Classes of Methods*.

None of the other *Classes of Methods* requires a major redefinition.

FS is of reduced relevance while **EMID** is of more relevance in the investigated domain. **SMG** is slightly redefined for increased generality. **DPE** is important in the investigated domain but suffers from a lack of tools designed for forensic use.

5.3.4 *Step 2.4* Revisiting Other Aspects of the [KDV15]-Model

This includes an analysis whether the other aspects included within the [KDV15]-model (*Structural Impact*, *Data Streams* and the *Classification Scheme for forensic tools and methods*) are useful for an investigation into cyber-physical systems. Potential alterations to these aspects in order to better reflect the specifics of cyber-physical systems also take place in this step.

5.3.4.1 *Structural Impact*

The concept of *Structural Impact* describes how a certain action alters the system state of the investigated system and in consequence the available forensic traces (see Section 3.1.2.4). The discussion of the term in Section 3.1.2.4 includes the various potential forms of *Structural Impact* in (potentially networked) Desktop IT systems. Table 3.1 shows, that *Structural Impact* might alter data in *Volatile Memory* and/or data in *Non-Volatile Memory* locally and/or across the network.

This observations are confirmed for the investigated domains as can be seen in numerous examples in Chapter 4.

The fact that two of the combined *Forensic Process Consequences* (FPC_3 and FPC_5) are at least partly based on *Structural Impact* validates the usefulness of this concept to describe the alteration of data and state during the forensic process by the forensic process.

However, there is a need provide a solid definition for *Structural Impact* devoid of any unnecessary elements. Based on the considerations in Section 3.1.2.4, the following definition is proposed:

> **Structural Impact (SI)**:
> *describes the alteration of the system state caused by the application of forensic methods. This alteration might propagate to connected systems.*

5.3.4.2 *Data Streams*

The *Data Streams* (see Section 3.1.2.5) provided a great benefit during Chapter 4 by grouping the various methods to gather forensic evidence based on the location of said evidence within the investigated domains. They provided a clear distinction on how certain sources for forensics traces where accessed and which implications from the point of view of the forensic process this access has (including a *Structural Impact*).

However, it has to be noted that the generally reduced usefulness of *Non-volatile Memory* (FPC_1), the reliance on diagnostic requests to gain access to *Non-volatile Memory* and *Volatile Memory* (FPC_3) and the increased importance of investigating *Communication* (FPC_2) implies that most forms of **DG** in the investigated domains domains rely on *Communication*.

Even under these circumstances, there are still relevant differences between *Data Streams*. For once, *Communication* takes places in the investigated domains without the interference of the investigator. *Communication* is required to fulfill an industrial objective, a driving task or whatever the user of a networked Desktop System is intent to do. It can be observed without interfering with it and hence causes no *Structural Impact*, cases aside where a network tap must be plugged into a physical connection and requires a temporary interruption of the *Communication*. On the other hand, the diagnostic requests performed by an investigator to gather contents of *Non-volatile Memory* and *Volatile Memory* (FPC_3) alter the system behavior and hence cause a *Structural Impact*.

The difference between *Non-volatile Memory* and *Volatile Memory* lies in the conditions in which the evidence in these *Data Streams* remains accessible. If the respective computing units were powered off *Volatile Memory* of these computing units is unavailable. It is important to point out that the Automotive IT domain has some specific properties with regards to the power supply of individual ECUs. In the case of Automotive IT ECUs are not usually powered off when the vehicle containing the ECUs is switched off ($IF_{AMI.Comp.9}$). Hence, in normal operation these ECUs are active for long time spans and the question how long after the incident *Live Forensics* can still be considered *Live Forensics* arises. This question is discussed in Section 4.1.3.1 and the essential answer to this is, that it depends on the *Data Types* and circumstances. For example, some data about hardware defects (**DT1**) will usually not change too often and might be retained over a longer period of time. The current data about the driving task (**DT9**) however changes often and is less useful after a certain time span elapsed.

Section 5.1.2 uses the term *Delayed Live Forensics* to describe this circumstance, but a formal definition how long after an incident the data within the *Volatile Memory* cannot be given. It must always be considered for the specific case, taking the respective *Data Type*, how often said *Data Type* is supposed to change and the time elapsed since the incident into account.

Overall, the *Data Streams* offer a notable benefit when describing the various tools used during the forensic process. They provide some inherent implications of using a tool addressing a given *Data Stream*.

However, Section 3.1.2.5 shows the need for a general definition of the various *Data Streams* devoid of any unnecessary elements and agnostic towards the various domains discussed during this thesis. Hence, the following definitions are proposed:

> **Non-volatile Memory**:
> *Memory inside a computing unit which maintains its content after the unit is disconnected from its respective power supply.*
>
> **Volatile Memory**:
> *Memory inside a computing unit which loses its content after the unit is disconnected from its respective power supply.*
>
> **Communication**:
> *All the data transmitted to other computing units via communication interfaces.*

5.3.4.3 *Classification Scheme for Forensic Tools and Methods*

The discussion of the *Classification Scheme for forensic tools and methods* in Section 3.1.2.6 focuses on some selected aspects with a fundamental aspect on the forensic process. Although the formal *Classification Scheme* is not used during the course of this thesis, some of the properties assigned to forensic tools motivated by this scheme were explored.

The most important of these properties is the *Structural Impact* which is highly relevant and hence discussed separately in Section 5.3.4.1.

Requirements for the promising usage of a forensic tool

These requirements are concerned with process specific conditions which must be fulfilled in order for a forensic tool to provide any useful evidence. The various examples of such conditions are provided in Table 3.3.

Some of these requirements are also applicable in the investigated domains. As shown in the discussion on *Data Streams* and *Live Forensics* in Section 5.3.4.2, *uv2* and *uv3*, are relevant the investigated domains. The lack of logging mechanisms in the investigated domains is one of the major challenges in the investigated domains (FPC_1) but does not make the inclusion of *uv1* unnecessary. *uv5* is a requirement to perform a range of diagnostic tests or *In the loop Forensics* (see Section 4.1.3.1), although in this case it may only refer to a single component and not an overall system. *uv6* and *uv7* describe certain user privileges. While some of the components used in the investigated domains (especially in the Automotive IT) do not contain any user management, this might still be the case in the case of more complex environments.

A consideration would be to add *vendor-specific knowledge* to the list of potential requirements ($IF_{ICS.Comp.7}$, $IF_{AMI.Comp.7}$ and $IF_{AMI.Arch.6}$). A prominent example would be the CAN bus in the Automotive IT domain which is easy to access and uses a well-specified communication protocol with a broad range of software able to dissect the network traffic. But which cannot be analyzed in full without vendor-specific knowledge about the specific CAN IDs (see Section 4.2.3.4). This knowledge would be necessary to assign various messages to specific ECUs which most likely would have send them (necessary due to $IF_{AMI.Arch.8}$).

On the next level, analyzing **DT9** would not only require the vendor-specific knowledge to interpret the specific network traffic but also knowledge in regards to the underlying physical process. Hence, it could be prudent to also include *process-specific knowledge* in such a list of potential requirements for the promising use of a forensic tool.

Relevancy of data protection concerns

The investigated domains do not contain personal data to the same extent as the Desktop IT domain. A notable exception are In-Vehicle-Infotainment (IVI) systems in Automotive IT (see Section 2.3.4.3). These contain personal data to a high extent but are not the focus of this thesis.

More relevant for the ICS domain might be the need to maintain business secrets since the information about the physical process could be considered a business secret ($IF_{ICS.Scen.6}$). If not protected, this information might fall into the hands of a competitor which then uses it to copy the manufacturing process. Although this is not a direct concern of data protection (Section 2.1.6), this property of the *Classification Scheme* offers the best fit within this scheme. Hence, the inclusion of some elements to describe the need for protection based on intellectual property would be helpful to address the ICS domain.

Tendency for evidentiary value

The evidentiary value is based on various different factors to judge the usefulness of evidence obtained by a certain tool during the forensic process. As discussed in Section 3.1.2.6, this mostly touches on the *Admissibility Factors* aligned with the **Daubert-Factors** (see Section 2.1.7 and Section 2.3.2.4).

Since **AF0** is assumed to always be true, this tendency boils down to the question whether the forensic evidence is integer and authentic and what the impact of error, loss, uncertainty and *Structural Impact* is. These questions were discussed in Chapter 4.

However, the construct of *Tendency for evidentiary value* combines too many factors in itself and hence is to unwieldy for practical use. Breaking down this property into a group of different properties, aligned with the *Admissibility Factors* seems to be a good choice for improving the *Classification Scheme*. It is worth noting that this adaptation does not originate from the investigated domains but from the complexity of this property in itself.

Protection measures for the integrity of a forensic tool, its input and its output

This concept is widely used in Chapter 4 to discuss whether and how a tool provides any of these protection measures. Barely any of the mentioned tools provide such mechanisms ($IF_{ICS.Comp.8}$, $IF_{ICS.Arch.8}$, $IF_{AMI.Comp.8}$ and $IF_{AMI.Arch.7}$).

However, this property still remains an important factor to determine whether a tool is suited for forensic use (see Section D.4). Hence, this properties does not necessitate any changes.

5.3.5 *Step 2.5* Revisiting Other Aspects

Some additional concepts not part of the [KDV15]-model require some additional discussion. This includes entirely new concepts established during the course of this thesis and some introduced during the various publications preceding this thesis (see Section 1.4).

5.3.5.1 *Investigative Contexts*

The concept of *Investigative Contexts* is introduced in this thesis based on the conclusion that Digital forensics investigates two different types of events (see Section 2.1.2). These two different types of events can be observed at various points during this thesis. However, the question arises on whether the two *Investigative Contexts* cover the entirely width of possible forensic investigations or if they require a redefinition after the analysis of the investigated domains.

There a two cases originating from the discussion in Section 4.1.2.5:

- Case 1:
  An attacker causes the physical process controlled by a computing unit to misbehave.
- Case 2:
  The physical process controlled a computing unit misbehaves.

Obviously, both cases could trigger a forensic investigation. Starting the discussion with *Case 1* there are various reasons why this case should belong to **IC1**. At first, such an attack could be described as *Data Manipulation* ([BKA18]). Also, this would cause an unwanted behavior of the system (which is the original definition of **IC1**).

However, controlling a physical process is the objective of the IT in the investigated domains just as creating text documents or spreadsheets can be considered the objective of Desktop IT. If this objective is used for illicit means, this would be considered **IC2**.

These different arguments suggest that *Case 1* could be either **IC1** or **IC2**. However, the meaning of the term *system* within the definition of **IC1** is also important, since when one would consider the physical process to be part of the system, *Case 1* would clearly fall into **IC1**. But the definition for ICS provided in Section 2.2.2 do not provide such an easy solution.

From the point of view of the forensic process it might be most useful to look at the *Investigative Contexts* in the way they are investigated during the forensic process. **IC2** usually relies on **DT8** and **DT9** (originally on **DT8** but this *Data Type* has been redefined and split into two in Section 5.3.2), while **IC1** relies on the other *Data Types*.

In *Case 1*, **DT9** would be predominately used to investigate if the physical process behaves in an unwanted manner. If this is the case, further *Data Types* would be investigated in order to identify how this unwanted control commands came to pass. This goes beyond an investigation in **IC2**. Manipulated **DT9** could also be a reason to start an investigation in **IC1**. Here, the unwanted behavior of the physical process would be the starting point of the forensic investigation (see Section 2.3.2.2). Following this argumentation, *Case 1* would be **IC1**.

Case 2 is a simpler version of *Case 1*. It carries the same trigger for a forensic investigation. The fact whether this trigger was due to a malfunction or a malicious attack can only be detected after a forensic investigation. Hence, *Case 2* also represents **IC1**.

5.3.5.2 Criticality Map

A concept to address the challenge that a forensic investigation could impact the control of the physical process FPC_5 was introduced in [**AHKD19**] implementing joint considerations on this topic with other members of the research group. The fundamental problem addressed by this concept are the resource limitations of the investigated domain and the (soft) real-time requirements. Hence, a *Criticality Map* identifies where the forensic process and the real-time requirements might conflict by identifying shared resources.

Such a *Criticality Map* can be seen in Figure 5.3. The example shows an ICS environment but is also relevant for the Automotive IT domain. In this example, three potential forensic tools and the respective shared resources within the ICS are shown. All forensic tools aim to obtain **DT9**.

- *Forensic Tool 1* is directly connected to the *SCADA server* to obtain **DT9** from it. It uses requests to the *Volatile Memory* of the *SCADA server*. Here, only the *SCADA server* is a shared resource. As long as this *SCADA server* has enough resources to maintain the requirements of the physical process and provide the forensic tool with the data, there is no problem for controlling the physical process. However, there is a *Structural Impact* in the *SCADA server* due to the requests of the Tool, which alter its *Volatile Memory*.

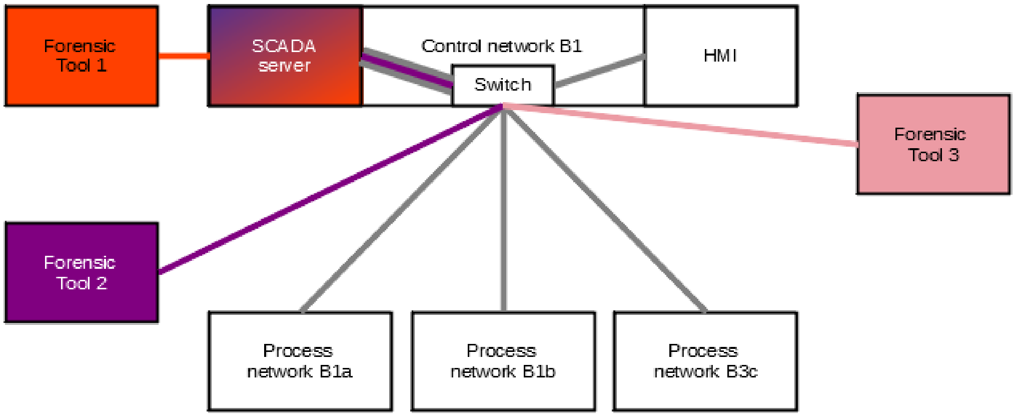


Figure 5.3: Example of a *Criticality Map* in an ICS network segment based on [KL15]

- *Forensic Tool 2* monitors the network connection and extracts **DT9** from the network connection. The physical access to the network requires a network tap or a monitor port (see Section 4.2.1.2). If this interface is installed before the monitoring occurs, no disruption to the network traffic occurs. Furthermore, passive monitoring of the traffic does not incur any *Structural Impact*. Also, there are no shared resources. Hence, such an approach has no effect on the ability of the ICS to fulfill the requirements of the physical process.

- *Forensic Tool 3* connects to the *SCADA server* using the *Control network B1*. This carries all the implications of *Forensic Tool 1* and adds the network connection as shared resource. This alters the *Communication* in this network and might also lead to a lack of bandwidth necessary to fulfill the requirements for controlled physical process.

Judging from this exploration, *Forensic Tool 2* seems like the best choice. However, some restrictions - like encrypted network traffic - might prevent the use of such an approach. In this case, *Forensic Tool 1* would be the best choice since it has less downsides than *Forensic Tool 3*.

It would be prudent to create such a *Criticality Map* during **SP** for use during **OP**.

5.3.6 Summary of the Adaptation

This section revisited various aspects to the forensic process as described in Section 3.1.2. The necessary adaptations are summarized here to provide the answer to **Research Question 3**.

The *Investigations Steps* fulfill the same tasks in the investigated domains as they do in Desktop IT. There is only a shift on the relative importance of the respective *Investigations Steps*. The importance of **SP** is increased to provide the potential for a meaningful **DG**. The definition of **DO** has been revised to include *Process Accompanying Documentation* and *Final Documentation*. The overall structure of the forensics process has been revised to include a loop from **DA** to **OP**. The revised structure for the forensic process can be seen in Figure 5.2. The revised definitions for the *Investigation Steps* are:

- **Strategic Preparation (SP)**:
  measures taken by the operator of an IT-system in order to support a forensic investigation prior to an incident
- **Operational Preparation (OP)**:
  measures of preparation for a forensic investigation after a suspected incident
- **Data Gathering (DG)**:
  measures to acquire and secure digital evidence
- **Data Investigation (DI)**:
  measures to evaluate and extract data for further investigation
- **Data Analysis (DA)**:
  measures for detailed analysis and correlation between digital evidence from various sources
- **Documentation (DO)**:
  measures for the detailed documentation of the proceedings (*Process Accompanying Documentation*) and for the compilation of a report on the incident (*Final Documentation*)

The *Data Types* require some major revisions. Motivated by the considerations presented in [Kil20], a reordering of elements takes place. In addition, the new **DT2**, **DT3** and **DT4** require some notable adaptation to fit the investigated domains. **DT6** requires a new definition due to some ambiguity of the terms used in its definition in the ICS domain. **DT7** is revised to include a definition of session that is geared towards the use in the investigated domains. **DT8** is altered and split into two *Data Types.* The definition is altered to take the key functionality of a system into consideration in order to represent the investigated domains and Desktop IT. The two new *Data Types* describe *Application Data* and *Functional Data*, which includes data about the physical process. The revised definitions for the *Data Types* are:

- **raw data (DT1)**:
  A sequence of bits within the Data Streams of a computing systems not (yet) interpreted.
- **hardware data (DT2)**:
  Data in a computing unit which is not, or only in a limited way, influenced by software.
- **details about data (DT3)**:
  Data added to other data, stored within the annotated chunk of data or externally
- **configuration data (DT4)**:
  Data which can be changed by software and which modifies the behavior of software and hardware, excluding the communication behavior
- **network configuration data (DT5)**:
  Data that modifies system behavior with regards to communication
- **process data (DT6)**:
  data about a running software process within a computing unit

- **session data (DT7)**:
  data collected by a system during a session, which consist of a number of processes with the same scope and time frame
- **application data (DT8)**:
  data representing functions needed to create, edit, consume or process content relied to the key functionality of the system
- **functional data (DT9)**:
  data content created, edited, consumed or processed as the key functionality of the system

The *Classes of Methods* are also affected by a shift of relevance between the various classes. Due to the simplicity of certain systems in the investigated domains, the distinction between **OS** and **ITA** requires some alteration, although none of the *Classes of Methods* does require a major redefinition. The current definition for the *Classes of Methods* is:

- **Operating System (OS)**:
  methods provided by the operating system
- **File System (FS)**:
  methods provided by the file system
- **Explicit Means of Intrusion Detection (EMID)**:
  methods provided by additional software with the characteristic of being executed autonomous on a routine basis and without a suspicion of an incident
- **IT Application (ITA)**:
  methods provided by the functions that fulfill the key functionality of the system
- **Scaling of Methods for Evidence Gathering (SMG)**:
  methods to collect additional evidence if a suspicion is raised but unsuited for routine usage in a production environment (due to reasons such as high false positives or high resource demand)
- **Data Processing and Evaluation (DPE)**:
  methods which support a detailed forensic investigation, display, processing and documentation)

The concept of *Structural Impact* was of great use during the domain analysis and received a cleaned up definition:

Structural Impact (SI):
describes the alteration of the system state caused by the application of forensic methods. This alteration might propagate to connected systems.

The *Data Streams* provided great benefit during the domain analysis. The fact that ECUs are not usually powered off when the vehicle containing the ECUs is switched off, necessitated an discussion on how long after an incident *Live Forensics* can be considered *Live Forensics*. This depends on the specific *Data Types* since some data will be retained unchanged in the *Non-volatile Memory* over longer time than others. No alteration to the concept of *Data Streams* in itself is necessary, however the *Data Streams* themselves received cleaned up definitions:

> **Non-volatile Memory**:
> *Memory inside a computing unit which maintains its content after the unit is disconnected from its respective power supply.*
> **Volatile Memory**:
> *Memory inside a computing unit which loses its content after the unit is disconnected from its respective power supply.*
> **Communication**:
> *All the data transmitted to other computing units via communication interfaces.*

Some properties from within the *Classification Scheme for forensic tools and methods* where used during the analysis. The properties from the *Classification Scheme* with an impact on the overall forensic process were used and some suggestions for alterations in the scope of the investigated domains have been made. Due to the selected use the *Classification Scheme* found during this thesis, no exhaustive alteration seems necessary at this point of time.

5.3.7 Adaptations in Light of the *Forensic Process Model Criteria*

Section 3.1.2.7 discusses how the [KDV15]-model fulfills the *Forensic Process Model Criteria* (see Section 2.3.2.4). Since these **PMCs** describe what a comprehensive forensic process model should address, it is worthwhile to discuss whether the adaptation of the [KDV15]-model has some impact on the fulfillment of these **PMCs**.

The [KDV15]-model is well-equipped to describe the structure of the forensic process (**PMC5**, **PMC6**, **PMC7** and **PMC8**). This strength is increased through the adaptation.

PMC5 received additional attention and clarification due to the discussion on the various *Investigative Contents* in Section 5.3.5.1. **PMC6** is validated by the the increased importance of **SP** in the investigated domains. In addition, conducting such an **SP** is well-addressed by the concepts of the revised model (including the *Criticality Map* - see Section 5.3.5.2). **PMC7** is addressed by discussing the various considerations for and against performing *Live Forensics* in the investigated domains (see Section 5.3.1.2).

The **PMCs** concerned with the **PCs** are not addressed any more or less than in the [KDV15]-model. However, a revision and comprehensive overhaul of the *Classification Scheme for forensic tools and methods* could increase the coverage of these factors (see Section 5.3.4.3).

6. Step 3: Evaluation of the Revised Forensic Model

This chapter evaluates the altered forensic process model presented in Section 5.1. This evaluation discerns whether the altered forensic process model can support forensic investigations in the ICS domain. It also evaluates whether the *Influence Factors* (see Section 5.1) and the *Forensic Process Consequences* (see Section 5.2) have the expected influence on the forensic process in this domain.

This chapter describes a complex case study performed in the domain of ICS. During this case study, a forensic investigation into a complex ICS was conducted. This section reflects how this case study aligns with the revisited [KDV15]-model for the forensic process.

This case study was conducted during the *International Training Course for Protecting Computer Based Systems in Nuclear Security*[1] conducted in 2019 by the IAEA (International Atomic Energy Agency)[2] in Daejon, Republic of Korea. This training course included a capstone exercise, which [Pur19] describes as:

> *„A mock scenario that included adversaries taking control of a nuclear research institute's physical protection system and implanting malware at a nuclear power plant, to compromise security and cause sabotage [..]"*

The author of this thesis was among the team responsible for setting up this exercise and was responsible for identifying the forensic traces the trainees could be able to find during their exercise while conducting incident response and computer forensic during said scenario. As such, the author of this thesis was able to apply forensic methods to a realistic attack scenario implemented by experts on the field of ICS in Nuclear Power Plants (NPP) and ICS security in a controlled environment.

Of course, this setup was simplified to only include a limited set of components in order to be implemented using a mock up instead of a real NPP. The reasons for this are due to the

[1] https://www.iaea.org/events/evt1703440, 30/03/2020

[2] https://www.iaea.org/, 30/03/2020

reduced risk of catastrophic damage, reduced cost and repeatability. [**AH20**], [**AHeS$^+$20**] and [**eSMP$^+$20**] expand on the topic on how simulators and mock ups can be used during training for incident response and computer forensics in NPP scenarios.

This case study starts with an overview on the employed scenario including the employed system architecture. It is discussed how this scenario and system architecture aligns with the *IFs* (see Section 5.1) and the *FPCs* (see Section 5.2). Then the forensic process starts. Following the revisited [KDV15]-model as discussed in Section 5.3, this includes a **SP** before the specific incident begins. Then, after a symptom triggers the investigation, the forensic investigations starts in earnest. Again, it will be discussed how the conduct of this forensic process aligns with the *IFs* and the *FPCs*. Afterwards a summary on how the adapted forensic process addresses this scenario and whether the *IFs* and the *FPCs* are confirmed in this scenario is provided. It is evaluated how the use of the revised forensic process model benefited the forensic investigation in this case study.

6.1 Setup of the Industrial Control System in the Scenario

The basic setup of the ICS in the scenario follows the IAEAs exemplary implementation for a secure infrastructure of IT in NPPs (see [NSS11]). This exemplary implementation includes five Security Levels and various Security Zones which are used to define network traffic flows. These Security Levels form a graded approach. The Security Levels are defined based on the impact of a potential compromise of these systems. These five Security Levels are summarized in [**HAL$^+$20**]:

- „Security Level 1 (SL1): systems vital to the facility (e.g. physical emergency protection)
- Security Level 2 (SL2): operational control systems which require high security
- Security Level 3 (SL3): supervision systems not required for operations
- Security Level 4 (SL4): technical data management systems (e.g. used for maintenance)
- Security Level 3 (SL3): business systems“

These Security Levels are compared to PERA (Purdue Enterprise Reference Architecture - Section 2.2.2.1) - in [**AHeS$^+$20**] providing the following conclusion:

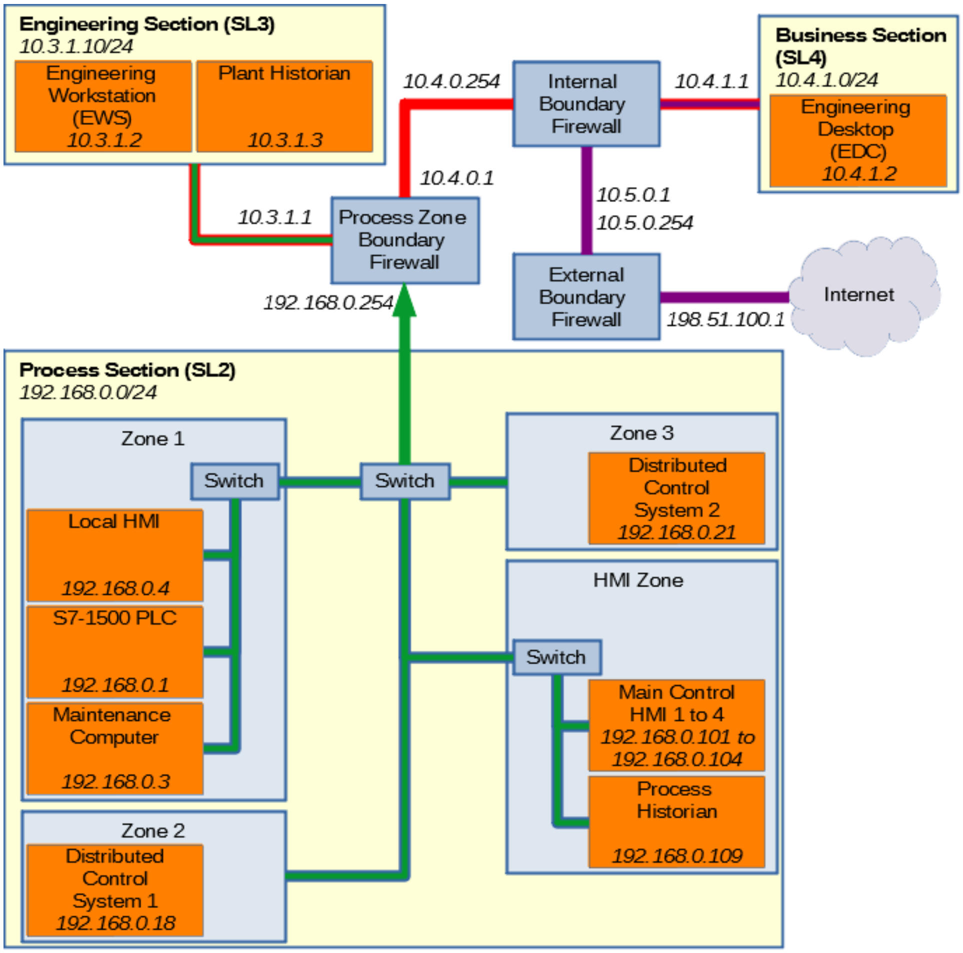
> *„These Security Levels roughly corresponds to the Purdue levels. Security Level 1-3 roughly aligns to the Cell/Area-Zone while Security Levels 4 and 3 align with the Enterprise Zone.“*

A main difference however is, that these Security Levels and Security Zones are used to restrict the traffic in these networks in order to provide increased security. An overview on these restrictions is provided by [**HAL$^+$20**]:

> *„Each security level is divided from the adjacent security levels by routers, firewalls and/or data diodes. In general, information flow from a lower security level*

> *to a higher security level is allowed, while communication from higher to lower security levels is highly restricted. From SL2 to SL1 no communication is allowed, while SL3 to SL2 allows for necessary acknowledgments or control packets to pass. SL4 to SL3 allows for specific and limited activity, while SL3 to SL4 carries little restriction. SL3 might be connected to the internet. In addition, systems on the same security level are grouped into various zones representing logical or physical groupings decoupled from other systems by using computer security measures (like firewalls). "*

This overview is necessary to understand the overall architecture of the network used within this scenario and to understand some implications from some forensic evidence encountered during the forensic investigation.



Engineering Section (SL3)
10.3.1.10/24
Engineering Workstation (EWS) 10.3.1.2
Plant Historian 10.3.1.3
10.4.0.254
Internal Boundary Firewall
10.4.1.1
Business Section (SL4)
10.4.1.0/24
Engineering Desktop (EDC) 10.4.1.2
10.4.0.1
10.5.0.1
10.5.0.254
10.3.1.1
Process Zone Boundary Firewall
External Boundary Firewall
Internet
192.168.0.254
198.51.100.1
Process Section (SL2)
192.168.0.0/24
Zone 1
Switch
Switch
Zone 3
Distributed Control System 2 192.168.0.21
Local HMI 192.168.0.4
S7-1500 PLC 192.168.0.1
Maintenance Computer 192.168.0.3
HMI Zone
Switch
Main Control HMI 1 to 4 192.168.0.101 to 192.168.0.104
Process Historian 192.168.0.109
Zone 2
Distributed Control System 1 192.168.0.18

Figure 6.1: Architecture of the overall system used during the case study

The architecture of the overall ICS is visualized in Figure 6.1. This architecture is a simplified example and does not include all systems potentially present in such an environment due to resource constraints. The creation of such an environment is discussed in [**AHeS+20**] and [**eSMP+20**]. However, the architecture includes a *Business Section* on Security Level 4, an *Engineering Section* on Security Level 3 and a *Process Section* on Security Level 2.

The *Business Section* contains systems that are used for technical management. Examples include maintenance schedulers, work permit and work order systems or Desktop Computers for the engineers and system analysts. One such *Engineering Desktop Computer* (EDC) is included in this exemplary architecture. This Security Level might have access to the Internet.

The *Engineering Section* contains systems not required for the operation of the overall ICS. This includes systems used by engineers to configure, maintain and diagnose control systems. These systems are referred to as *Engineering Workstations* (EWS). One of these systems is included in this example. This example also includes a *Plant Historian* which is tasked with recording information about the physical process and storing it in order to support engineers and process analysts. [NSS11] states that this Security Level has no access to the Internet and that only specific and limited communication activity to Security Level 4 is allowed.

The focus of this scenario is the *Process Section*. This section consists of four distinct zones.

Zone 1 is the zone most relevant for this scenario and hence shown in greatest detail. It contains a *SIMATIC S7-1300 PLC*[3] which controls the condenser subsystem. Furthermore, a *Local HMI* to monitor and control the operations of the condenser subsystem and a *Maintenance Computer* which is only powered on to perform maintenance tasks.

Zone 2 and *Zone 3* represent other zones which control different subsystems of the overall NPP. In the case of the training scenario, these were implemented using the *Asherah NPP simulator* (ANS) as presented in [**AHeS+20**]. In essence, these zones provide realistic communication with the other parts of the overall ICS and realistic feedback to the actions performed and triggered in *Zone 1*. A *HMI Zone* represents the main control room. Here four *Main Control HMIs* provide an overview and control of the physical process. It is noteworthy, that usually these *Main Control HMIs* should be situated on Security Level 3. [NSS11] states that this Security Level has no access to the Internet. Only one way network traffic to Security Level 3 is allowed. Remote maintenance from Security Level 3 may be allowed for a defined period on a case by case basis.

The overall architecture used in the scenario already confirms some of the *IFs* established in Section 5.1. The close relationship between ICS and Desktop IT is obvious. Security Level 4 and Security Level 3 are basically Desktop IT ($IF_{ICS.Atk.3}$). Also, the architecture contains Historians ($IF_{ICS.Arch.9}$). Overall, the architecture follows a clear and established network hierarchy ($IF_{ICS.Arch.1}$).

6.2 Forensic Process

This section discusses the entire forensic process as performed during this case study.

6.2.1 SP before the Incident

The forensic process as described in Chapter 5 includes a **SP**. This **SP** was used to perform various steps in preparation before an incident occurred.

At first, a network diagram as shown in Figure 6.1 was obtained in order to plan the additional steps necessary during the **SP**. This network diagram allows for the identification of communication flows.

[3] https://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1300/documents/EN/software_complete_en.pdf, 30/03/2020

Since the network traffic between various Security Levels is restricted storing this network traffic does not produce excessive amounts of data. This is the case with the network traffic between Security Levels 3 and 4 and also between Security Level 2 and Security Level 3. Hence, methods to monitor the boundaries between these Security Levels were installed. Since the network is Ethernet-based the installation of such mechanisms is quite easy. The low volume of network traffic allows for the use of the Monitor Ports of the infrastructure elements connecting the various Security Levels (see Section 4.2.1.2). Hence, the *Internal Boundary Firewall* (at the 10.4.1.1 interface) and the *Process Zone Boundary Firewall* (at the 103.1.1 interface) were monitored. Monitoring of the network traffic was also prepared in the *Process Section* which contains four zones within Security Level 2. Here, all traffic is routed across the central switch. This switch was used as the monitoring point. Due to privacy concerns and the problem of intellectual property ($IF_{ICS.Scen.6}$) the respective data should have been recorded with a tool like *LFTB* ([**KHAD10**]) which provides integrity, authenticity and confidentiality of the gathered data (see Section 4.2.1.2). However, in this case a basic network sniffer running on a dedicated machine was used. As can be seen in Section C.2, enough additional information of the captures was provided to ensure a decent level of evidentiary value associated with the network captures.

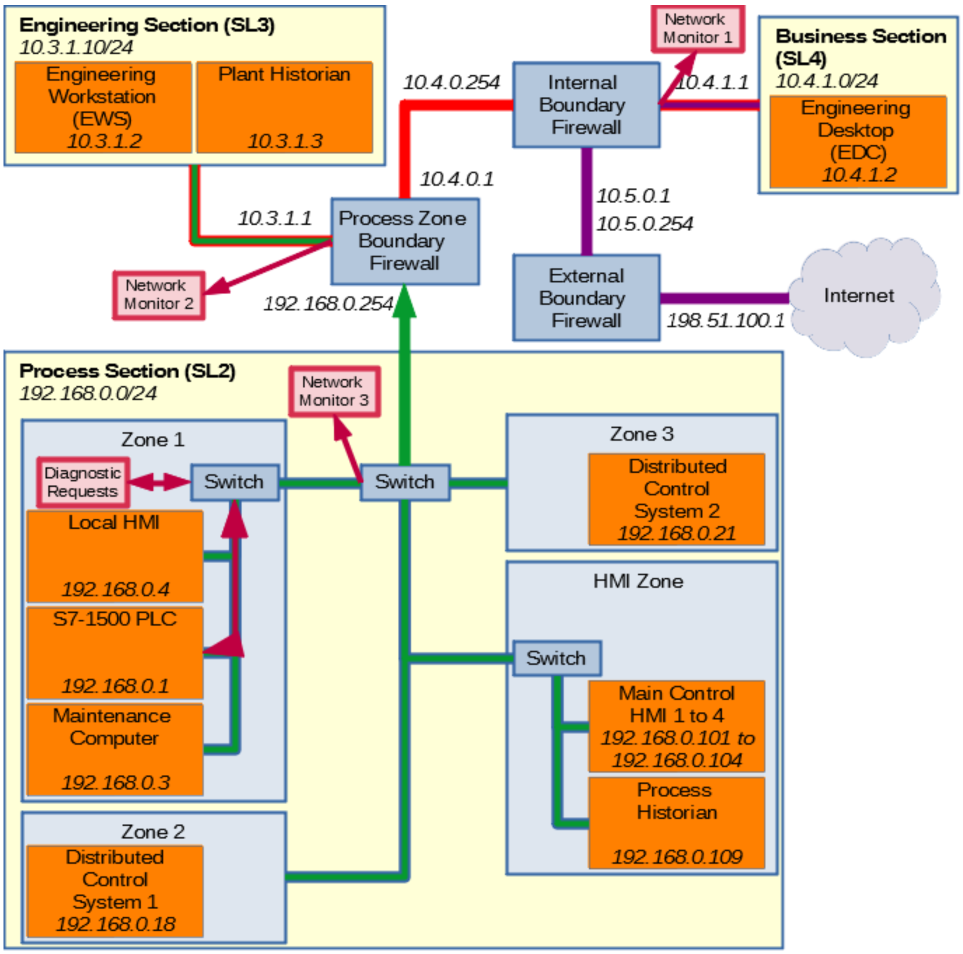

Engineering Section (SL3)
10.3.1.10/24
Engineering Workstation (EWS) 10.3.1.2
Plant Historian 10.3.1.3
Network Monitor 1
Business Section (SL4)
10.4.1.0/24
Engineering Desktop (EDC) 10.4.1.2
10.4.0.254
Internal Boundary Firewall
10.4.1.1
10.4.0.1
10.5.0.1
10.5.0.254
10.3.1.1
Process Zone Boundary Firewall
Network Monitor 2
192.168.0.254
External Boundary Firewall
198.51.100.1
Internet
Process Section (SL2)
192.168.0.0/24
Network Monitor 3
Zone 1
Diagnostic Requests
Switch
Switch
Local HMI 192.168.0.4
S7-1500 PLC 192.168.0.1
Maintenance Computer 192.168.0.3
Zone 2
Distributed Control System 1 192.168.0.18
Zone 3
Distributed Control System 2 192.168.0.21
HMI Zone
Switch
Main Control HMI 1 to 4 192.168.0.101 to 192.168.0.104
Process Historian 192.168.0.109

Figure 6.2: *Criticality Map* for the system used during the case study

This strategic planning on how to perform **DG** is also supported by the *Criticality Map* (see Section 5.3.5.2) for the overall system. This criticality map is shown in a simplified form in Figure 6.2. Green denotes resources used for a specific type of **DG**. Unidirectional arrows indicate unidirectional network traffic while bidirectional arrows indicate bidirectional traffic. The three *Network Monitors* each only use a dedicated, unidirectional connection from the network infrastructure elements. This does not incur any interference with the physical process. *Diagnostic Request 1* is added as an example for additional means to perform **DG**. In this case, bidirectional network traffic between the component performing the request and the PLC would take place using the switch. The PLC and the switch would be shared resources in this case potentially interfering with the control process.

The possibility to monitor network traffic was used to gain an understanding of the baseline behavior of the network traffic in the various networks. Due to the usual monotony of traffic in Security Level 2, common patterns can be easily identified. But this also applies to the other monitored network. Such a baseline provides an understandings on the actions usually performed and the protocols usually employed in these networks. It helps to identify unusual behavior in case of a forensic investigation.

In addition, a basic understanding of the physical processes was achieved in order to be able to give a reasonable assessment of the potential consequences in case of an incident. This understanding of the underlying physical process illustrates the critical nature of the PLC responsible for the condenser and the critical events that might result from a condenser failure based on [PNN04]. This analysis consists of potential consequences of abnormal behavior, potential consequences of digital compromise and an analysis of the financial cost incurred for a specific downtime. These list are not included in this thesis due to security concerns although Section C.1 provides an example on what such lists would entail. In a real world scenario, this analysis would be done for all components. This understanding is also necessary in order to correlate various events within the physical process with potential actions performed by an attacker during system compromise. In addition, such an understanding is necessary in order to interpret the network traffic in Security Level 2 (which is mostly **DT9** and forms the majority of network traffic in the overall system).

The basic understanding of the network behavior and the used protocols also enabled the preparation of tools able to interpret these protocols. In this case, the *Wireshark*[4] was identified at being able to interpret the specific protocols with the additional *s7commwireshark*[5] extension. This made understanding the basic communication behavior in the specific networks easier.

Organizational processes are also relevant during the **SP**. These organizational processes establish which personnel has access to which systems. Even more important, these processes establish the circumstances for exceptions in the restrictions in the flow of network traffic. During a forensic investigation, knowledge of these exceptions is useful when reviewing *Communication*. This does not necessarily mean that the forensic investigator is aware of all scheduled system accesses or exceptions to network traffic restrictions at all times, but that the investigator is aware on how to obtain these records in case of need.

This also confirms some of the *IFs* established in Section 5.1. The generally low amount of network traffic makes the monitoring of network traffic at selected points in the architecture

[4] https://www.wireshark.org/, 30/03/2020

[5] https://sourceforge.net/projects/s7commwireshark/, 30/03/2020

practical ($IF_{ICS.Arch.11}$). Since the traffic is Ethernet-based, the physical access to the network can be performed easily ($IF_{ICS.Arch.3}$).

In the *Process Section* **DT9** dominates the traffic ($IF_{ICS.Arch.10}$). Also, this network traffic is regular and of a relatively low volume ($IF_{ICS.Arch.11}$).

Since the controlled system is of high importance to the overall process, the system has high availability requirements ($IF_{ICS.Scen.3}$). In addition, this functionality is critical and a loss of it might incur severe damage ($IF_{ICS.Scen.4}$). Hence, an understanding of the physical processes is required in order establish the ability to make informed decisions during the forensic process (especially during **OP**). This also requires an understanding of **DT9** ($FPC_{ICS.6}$).

6.2.2 Condenser Subsystem

Although technical details on the function of the condenser are not necessary to follow the forensic process as discussed in this case study, the discussion on the **SP** highlighted the importance for some understanding of the underlying physical process. This understanding might be helpful to follow the reasoning used during the forensic investigation, especially in the case of the **OP**. Hence, a brief introduction into the operation of the condenser shall be provided here. A more conclusive overview can be found in [ARG17].

The condenser condenses hot turbine exhaust steam into water, so it can be reused in the steam generator as feed water. In essence, the condenser is a heat exchanger. In addition, the condenser is tasked with maintaining a vacuum to maximize the efficiency of the turbine. Hence, it consists of two main functions - CD Level Control and CD Pressure Control. In order to achieve these functions sensors for temperature, pressure and water level are included. Actors are pumps (for water and vacuum). A condenser storage tank is present within the system. On demand, water can be pumped from the condenser to the condenser storage tank or vice versa in order to maintain the CD level. All these actors are controlled digitally in this scenario.

If the condenser entirely fails, the turbine must be stopped. This ends the power generation.

6.2.3 The Incident and OP

The incident started with the operators at Main Control HMI 1 and 2 noticing a shift in the water level in the condenser. Notification of the engineer responsible for the condenser and a check on the Local HMI attached to the condenser (as well as visual inspection) confirmed this observation. This triggered the investigation into the misbehavior of the condenser.

At first, a consideration on the criticality of the observed behavior was made in order to decide whether the system should be shut down or not. These considerations benefited from the analysis performed during **SP**. Here, the analysis of potential consequences of abnormal behavior, potential consequences of digital compromise and an analysis of the financial cost (see Section C.1) provided input for an informed discussion. Since the potential impacts were not catastrophic and the financial cost very high, it was judged that shutting down the system was not necessary at this point of time. But the option to shut the system down later due to shifting circumstances was discussed and kept open. Note that in the actual *International Training Course for Protecting Computer Based Systems in Nuclear Security* this decision was made by the chief engineer after a discussion with the investigator after a first review of the indicators of compromise.

Due to the monitoring installed in the **SP** it was decided to not cause any further *Structural Impact* and to rely on the network monitoring already in place (see the *Criticality Map* for this ICS as shown in Figure 6.2). It was decided to gather the *Communication* within the *Process Section* and to compare it to the baseline *Communication* within the *Process Section* (which was also prepared during the **SP**). In addition, **DT9** as available in the Main Control Room HMI 1 and 2 was chosen for continued monitoring.

The discussion was fueled by various *IFs* and *FPCs* including $IF_{ICS.Scen.3}$, $IF_{ICS.Scen.4}$, $FPC_{ICS.2}$,$FPC_{ICS.4}$ and $FPC_{ICS.5}$.

6.2.4 DG and DI in the *Process Section*

The *Communication* in the *Process Section* could be physically accessed by using *Network Monitor 3* for physical access. The monitor port of this device was used to capture the specific network traffic, as already discussed in Section 6.2.1. The network traffic was acquired as **DT1** and its integrity ensured by employing a cryptographic hash function on the network traffic. This network capture is referred to as **Process-Section-Abnormal.pcap**.

A forensic copy of **Process-Section-Abnormal.pcap** was used to perform **DI**. Due to the identification of the used protocols during **SP**, *Wireshark*[6] with all the necessary dissectors was available on the forensic workstation used to perform the **DI**.

Statistic information about the network capture is provided in Table C.4. The same information is provided for the network capture used for establishing the base line behavior the *Process Section* (**Process-Section-Baseline.pcap** - see Table C.3).

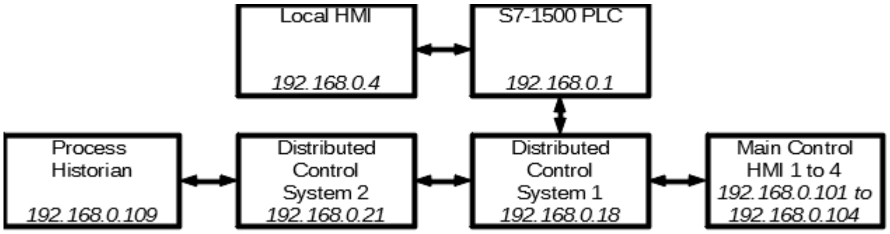


Figure 6.3: Baseline for network connection behavior in the *Process Section* used during the case study - *Process-Section-Baseline.pcap*

A listing of the connections available in both captures is provided in Section C.2.1 and Section C.2.2 respectively. The creation of a network map during **SP** allows for a mapping of the respective network addresses on physical systems which is highly useful to understand the network behavior. In this case, the baseline capture shows some forms of regular traffic, which is also visualized in Figure 6.3:

- **PLC to DCS 1**:
  192.168.0.1 <-> 192.168.0.18
- **PLC to Local HMI**:
  192.168.0.1 <-> 192.168.0.4

[6] https://www.wireshark.org/, 30/03/2020

- **DCS1 to DCS2**:
  192.168.0.18 -> 192.168.0.21

- **DCS 1 to Main Control Room HMI**:
  192.168.0.21 <-> 192.168.0.101
  192.168.0.21 <-> 192.168.0.104
  192.168.0.21 <-> 192.168.0.103
  192.168.0.21 <-> 192.168.0.102

- **DCS 2 to Process Historian**:
  192.168.0.18 <-> 192.168.0.109

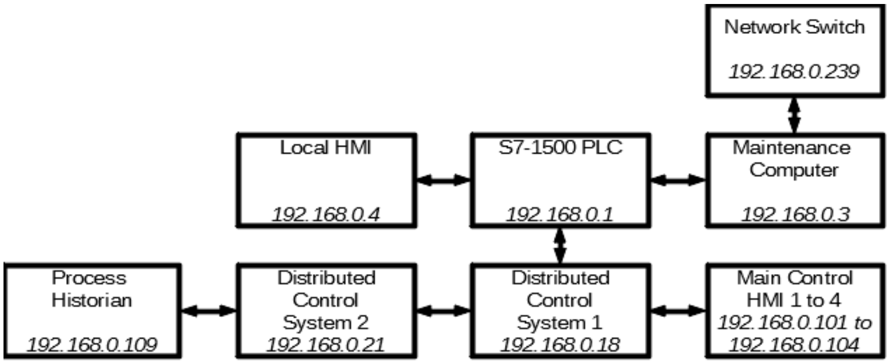


Figure 6.4: Network connection behavior during after the incident was noticed in the *Process Section* used during the case study - *Process-Section-Abnormal.pcap*

A comparison to the network connections in **Process-Section-Abnormal.pcap** shows some differences (see Section C.2.2). At first, all the connections present in **Process-Section-Baseline.pcap** are also present in **Process-Section-Abnormal.pcap**. However, it includes additional connections, some of them with notable amounts of data transmitted. These connections are shown in Figure 6.4:

- **PLC to Maintenance Computer**:
  192.168.0.1 <-> 192.168.0.3

- **Maintenance Computer to Network Switch**:
  192.168.0.3 <-> 192.168.0.239

The connection between PLC to Maintenance Computer was deemed the most interesting for this time being. Hence, the network traffic was filtered for this connection. The listing of protocols present in this filtered capture is provided in Table C.5 and shows a notable amount of s7comm plus traffic. This protocol is used for communication with SIMATIC PLCs. This communication can investigated by filtering the network traffic for s7comm plus traffic and reviewing the respective packets. Two of these packets are shown in Section C.2.2.2. They contain the *Request SetMultiVariables* command send from the Maintenance Computer to the PLC (Packet 34839) and the response which acknowledges this requests (Packet 34862). These requests are roughly repeated twice per second. The value this variable is set to decreases over time. At the beginning of the capture, 80 is observed as target. After a while this is reduced to 70 (Packet 99973). These requests explain the abnormal behavior of the physical process and corresponds to the **DT9** monitored at Main Control Room HMI 1 and 2 (at the GUI accessing the respective *Volatile Memory* of these components).

Please note that a full list of the filters used for the analysis of these network captures is provided in Section C.2.2.1. This also includes a detailed excerpt of the mentioned network packets.

The second connection between Maintenance Computer to the Network Switch was also subject to investigation. The listing of protocols implies the presence of HTTP traffic (see Table C.6). Investigating the HTTP stream yields inconclusive results. There are login attempts to the web interface of the Network switch. This alone is suspicious.

However, it was unclear why the Maintenance Computer is performing these actions. Further investigation to understand the nature of this compromise was necessary. In the restricted environment of this scenario, maintenance actions are scheduled and recorded. There was no record for a planned maintenance at this point in time.

This analysis exemplifies the need to be able to interpret the **DT9** in ICS ($FPC_{ICS.6}$). Furthermore, it already seemed obvious that the incident was caused by legitimate but malicious commands ($IF_{ICS.Atk.2}$).

6.2.5 DA in the *Process Section*

The identified commands in the *Communication* between PLC and Maintenance Computer align with the abnormal behavior of the condenser. The altered values requested by the commands send from the Maintenance Computer correspond to the altered behavior of the PLC and the attached actors. Hence, these commands can be identified as causing this altered behavior.

Every physical maintenance access to the Maintenance Computer is logged (as a hard copy list) as required by organizational methods. Due to preparation (organizational procedures during **SP**) these records could be obtained. Additional information from the CCTV (Closed Circuit Television) confirmed that no physical to the Maintenance Computer occurred during the time span of the incident.

Logical access to the Maintenance Computer from Security Level 3 and above is restricted (see Section 6.1). In addition, no such logical access has been identified during the analysis of *Process-Section-Abnormal.pcap*.

At this point of the forensic investigation, legitimate but malicious control signals are send from the Maintenance Computer to the attached control network. Why the Maintenance Computer is transmitting these signals is not known.

6.2.6 Repeated OP

With this knowledge, the decision whether to shut down the ICS in question can be reiterated and the next steps for the forensic investigation can be identified.

The understanding that the abnormal behavior is indeed caused by a malicious commands allows for a reconsideration of the threat to the control physical process. Since the malicious commands are transmitted by the Maintenance Computer these commands can be disrupted by disconnecting the Maintenance Computer from the network. During the **SP** knowledge about the general architecture of the ICS and the physical process as such was obtained. According to this knowledge, the Maintenance Computer does not perform any function necessary for the continuous operation of the network system. In addition, the disruption of these malicious commands will allow for the physical process to be restored to normal operation parameters without any adverse effect.

From a point of view of forensics, disconnecting the network connection from the Maintenance Computer without shutting the system down would allow for an analysis of the *Non-volatile Memory* during *Live Forensics*. It might be possible that an attacker is still active by using a malicious software installed on the Maintenance Computer running a preconfigured attack routine. Such a routine could be used to disrupt forensic evidence in *Non-volatile Memory* and *Volatile Memory* once the disconnect is identified. Since the usual behavior of the network protocols used to transmit these commands includes an acknowledgment, this disconnect could be identified certainly. However, **DT6** in the *Volatile Memory* of the Maintenance Computer might be highly relevant to identify the origin of the commands within the Maintenance Computer.

Considering these factors, the Maintenance Computer is removed from the network.

Additional forensic investigation is deemed necessary since it is unclear on how an attacker could misuse the Maintenance Computer to send malicious commands and how the attacker achieved such a degree of knowledge about the underlying physical process to perform such an attack.

To understand how the Maintenance Computer was misused, an investigation into the Maintenance Computer is necessary. This investigation can be performed using *Live Forensics* at first to obtain **DT6** from the *Volatile Memory*. Following this, a *Post-Mortem Investigation* will follow. Since the Maintenance Computer is a Desktop Computer (with some ICS specific software suites installed), this investigation can rely on well-tested methods from the Desktop IT domain.

The knowledge the attacker used to perform the attack is available at various positions within the overall architecture. Reviewing Figure 6.1 and using specific knowledge about the organizational procedures within the given ICS, some potential sources for the knowledge used by the attacker are identified:

- *Potential Source 1*: *Zone 1* in the *Process Section*
  Since this zone is controlling the physical process information about this physical process is available here. An attacker could capture and analyze the *Communication* in this zone to identify functional parameters (not unlike the review of **DT9** performed during this forensic investigation). This requires beforehand knowledge of protocols used within this zone. In addition, the attacker will obtain only the normal operating parameters and have no additional technical specifications about the employed components.

- *Potential Source 2*: Plant Historian in the *Engineering Section*
  The Historian obtains some information about the physical process from the *Process Section*. This information could help the attacker to understand the normal operating

parameters but lacks additional technical specifications about the employed components. The protocols used might not be known to the attacker in this case since the protocols might be converted at the Process Zone Boundary Firewall. In this case, the attacker would have to actively infiltrate the Plant Historian.

- *Potential Source 3*: Along the line of network traffic between *Zone 1* in the *Process Section* and Plant Historian in the *Engineering Section*
  The attacker could passively capture the network traffic between *Zone 1* and the Plant Historian. Examples include the switches, the Process Zone Boundary Firewall or any of the physical carriers. The same information about the physical process as in the other two cases would be obtainable to an attacker. Depending on the exact physical location of the access, information about the network protocols used in *Zone 1* in the *Process Section* might be obtainable.

- *Potential Source 4*: Engineering Workstation in the *Engineering Section*
  The EWS is used to configure and maintain the systems in the *Process Section*. Programming the PLCs is considered as a part of configuration in the ICS domain. Hence, the EWS contains what would be considered program code for some of the components in *Process Section* (**DT8**). Hence, the EWS contains the most conclusive information about the physical process and the used protocols.

Potential Source 1 was already investigated and no exfiltration of data about the physical process was detected. However, the investigation of the Maintenance Computer was still in progress at this point of time. *Potential Source 3* requires an investigation into every of the network infrastructure elements and the attached carriers to obtain clarity. Such an investigation could interfere with the underlying physical process since all these network infrastructure elements are shared resources by definition. In addition, the effort to investigate all these components is comparatively high. *Potential Source 2* and *Potential Source 4* can be investigated with less effort. The knowledge about the physical process and employed protocols of use for the attack as performed against *Zone 1* in the *Process Section* would have to be exfiltrated. Hence, the *Communication* from the *Engineering Section* might contain evidence. Since the traffic from Security Level 3 is sparse the traffic is monitored in full (as prepared during **SP**).

Hence, investigating the *Communication* from the *Engineering Section* does not require great effort. Since *Potential Source 4* offers the specific knowledge deployed by the attacker during the attack, the EWS seemed the most likely source.

6.2.7 DG and DI in the *Engineering Section*

At this point, a review of the monitored network traffic in the *Engineering Section* was conducted. The capture was already available due to the procedures conducted during **SP**. The generally low volume of traffic (and the specific laws in this restricted domain) allowed for the capture of the entire traffic.

Again, baseline knowledge about the protocols used and activities performed in this network was acquired during the **SP**. The components in the *Engineering Section* are Desktop Computers with specific software installed to perform their functions in this ICS environment. This shows the close relationship between ICS and Desktop IT once more ($IF_{ICS.Atk.3}$).

A forensic copy of **Engineering-Section.pcap** was used to perform **DI**. Due to the identification of the used protocols during **SP**, *Wireshark*[7] with all the necessary dissectors was available on the forensic workstation used to perform the **DI**.

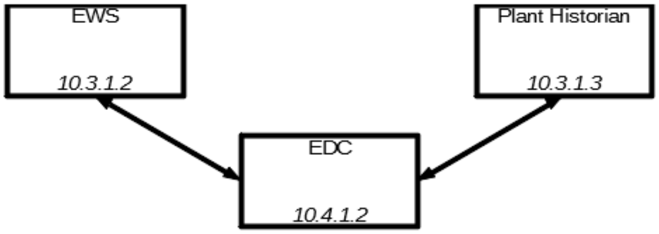


Figure 6.5: Network connection behavior in the *Engineering Section* used during the case study - *Engineering-Section.pcap*

A listing of the connections identified in **Engineering-Section.pcap** is provided in Section C.2.3. The principal connections found in this trace are visualized in Figure 6.5. Both connections are to the Engineering Desktop Computer (EDC) in Security Level 4. A great amount of traffic was exchanged between EWS and EDC while the traffic between Plant Historian and EDC was low.

Hence, the *Communication* between EWS and EDC was investigated first. Again, a full list of the filters used for the analysis of this network capture is provided in Section C.2.3.1. A protocol overview of the network traffic between EWS and EDC (provided in Table C.8) shows a notable amount of Telnet traffic within this network. During the **SP** it was established that it is not unusual that some engineers prefer using this legacy protocol since some of the legacy components employed within the ICS are unable to use more sophisticated protocols due to resource constraints ($IF_{ICS.Scen.2}$). It also simplifies the **DI** since the unencrypted protocol is easy to interpret.

Packet 143, *Packet 145*, *Packet 147* of **Engineering-Section.pcap** already show some unusual behavior. These packets are provided in Section C.2.3.2 as reference. Each of these packets request a new connection to the Telnet server on the EWS from the EDC. This can be observed due to the set *SYN* flags and the different ports in every request. This behavior is repeated throughout the capture with incrementing source ports. These requests happen with a short delay. *Packet 204* shows the response from the Telnet server on the EWS after the server is momentarily flooded with connections. In this case, investigating the last packets communicated between EWS and EDC indicates if the Telnet connection was finally successful.

Packet 33030 confirms this. The used port numbers can be used for more specific filter for this successful connection. *Wireshark* allows for the reconstruction of the Telnet session using the 'follow' operation. The reconstructed session can be seen in Figure C.1. It includes a login, a browsing of the file system and the exfiltration of some construction documents. This led to the conclusion that required knowledge to perform the attack was transferred from the EWS to the EDC after a bruteforce attack to obtain a Telnet login.

However, **Engineering-Section.pcap** also contains a connection between the EDC and the Plant Historian. This connection also had to be reviewed. Table C.9 shows the protocols used

[7] https://www.wireshark.org/, 30/03/2020

in this connection. Here, FTP is used. Due to knowledge obtained during the **SP** it is known that FTP is also used within this network due to constraints of some of the legacy components. Engineer employ this protocol to retrieve historic data about the physical process from the Plant Historian. This data supports engineering tasks.

FTP uses two connections. One for the command and one for data transfer. The connection used for commands was more interesting since it contains information about the actions performed. The reconstructed FTP section is provided in Section C.2.3.3. The login was performed using a valid password. Some historic data concerning the condenser was requested.

6.2.8 DA in the *Engineering Section*

DI identified a brute force attack on the Telnet server on the EWS which led to the exfiltration of construction data. In addition, a FTP sessions to the Plant Historian was identified. This FTP session was used to transfer some historical data about the operation of the condenser. Both these connections were initiated by the Engineering Desktop Computer (EDC) in Security Level 4.

The FTP session happened before the Telnet session. In itself, the FTP sessions seems entirely innocuous since valid credentials were used and this action is not unusual (as established during **SP**). However, the temporal proximity to the brute force attack makes it likely that this action belongs to the overall attack.

Specific knowledge about the domain confirms that the data exfiltrated from the EWS could supply the attacker with the necessary technical details to perform the attack against the condenser.

6.2.9 Conclusion of the Investigation

The investigation into the *Process Section* could identify the cause of the unwanted system behavior. However, it could provide no information on how the Maintenance Computer was compromised.

However, it became clear the attacker possesses very specific knowledge in order to implement an attack causing the unwanted system behavior. An investigation into the *Engineering Section* identified the exfiltration of technical data from the EWS to the EDC.

This is time for another return to the **OP** in order to identify the next steps during the investigation. In this case, this was an investigation into the EWS while continuing the investigation into the Maintenance Computer.

The investigation of the EWS is conducted in the scope of the Desktop IT domain. The investigation of the Maintenance Computer also is a classical investigation as performed in the Desktop IT domain. Hence, these investigations are not helpful to illustrate the forensic process in ICS besides the point that ICS and Desktop IT are closely intertwined ($IF_{ICS.Atk.3}$).

However, closure to the results of the overall investigation should be provided before the case study is review in light of the applied forensic process model.

The Maintenance Computer was removed from the network and investigated. A hardware implant was identified. This implant provided a communication interface to a wireless network used by an attacker.

Further investigation into the *Business Section* confirmed the presence of malicious software on the Engineering Desktop Computer (EDC). This malicious software was disguised as a genuine analysis software which was installed within the system after a social engineering attack on the chief analyst. This also included altering the firewall rules to allow communication to a license server which actually served as a Command & Control Server for the malicious software.

Tracing the network traffic caused by the EDC it was identified that the Engineering Workstation (EWS) and the Plant Historian in the *Engineering Section* were accessed. There, documents specifying the physical processes were accessed and exfiltrated. Also information about upcoming maintenance work and hardware replacements was stolen from the *Business Section* using credentials acquired on the EWS.

This specific information helped the adversary to tailor an attack to the physical process present in the *Process Section* and to perform a supply chain attack, using the replacement of hardware components to deploy the hardware implant. This attack is complex and points to an adversary with high resources and expertise. However, such an adversary is not unlikely in the NPP domain, as the examples provided in Section 4.4.1.1 clearly show.

This case study confirmed the assumptions on a range of *IFs* and *FPCs*. $IF_{ICS.Arch.1}$, $IF_{ICS.Arch.5}$, $IF_{ICS.Arch.7}$, $IF_{ICS.Arch.8}$, $IF_{ICS.Arch.10}$, $IF_{ICS.Arch.11}$, $IF_{ICS.Scen.2}$, $IF_{ICS.Scen.3}$, $IF_{ICS.Scen.4}$, $IF_{ICS.Scen.5}$, $IF_{ICS.Atk.2}$, $IF_{ICS.Atk.3}$, $FPC_{ICS.2}$, $FPC_{ICS.4}$, $FPC_{ICS.5}$ and $FPC_{ICS.6}$ were already identified.

The conclusion adds $IF_{ICS.Atk.1}$ due to the complexity of the attack.

6.3 The Use of the Revised Forensic Process Model in This Case Study

This section discusses whether and how the revised forensic process model as presented in Chapter 5 helped the conduct of this case study when compared to the [KDV15]-model as presented in Section 3.1.2.

6.3.1 Advantages of Using a Forensic Process Model During This Case Study

While the *Forensic Process Model Criteria* (*PMC* - see Section 2.3.2.4) describe what constitutes a good forensic process model the main task of a forensic process model is always to support the forensic investigator. Hence, the primary question is, if the use of a forensic process model helped the forensic investigator in this case.

In this case study, not all *PMCs* were relevant to the same degree. Constraints for the collection and use of evidence (**PMC3**) and the admissibility of the collected evidence in courts (**PMC4**) were no primary concern in this scenario.

However, the evidence gathered, investigated and analyzed during the investigation still needed to be handled in a specific way to maintain the reliability of the conclusion (the evidentiary value). Here, a forensic process model supports the investigator by providing the forensic process with a clear structure. Documenting the actions performed along the lines of this structure provides a chain of evidence which supports and maintains the reliability of the conclusion (**PMC1**).

This also includes all the actions performed before the incident started (**PMC6**). The understanding of a clearly structured process helps identifying the steps necessary to prepare a potential investigation before an incident happens. Following a structured process also includes documenting all the steps made during this preparation in order to increase the reliability of the conclusions made from the respective evidence.

The structured approach also leads to the identification and gathering of evidence at the beginning of the investigation in contrast to a piecemeal approach. In such an approach, a piece of evidence is gathered and investigated leading to the identification of another piece of evidence which then needs to be gathered and investigated. This piece of evidence might not be available anymore at this point of time. The structured approach however aims at identifying all sources of potential evidence in the beginning of the investigation and then gathers the respective evidence before it is investigated. This does not exclude the extension of an investigation as has been shown during this case study. In fact, new investigations were launched into different networks and systems contributing to an overall documentation of these investigate actions.

In addition, a structure process allows for the structured documentation of the actions performed and the decisions made during the forensic process. This is especially useful if the conclusions of the investigation come under scrutiny - either in court or in an audit. Here, a structured process helps describe the entire forensic investigation in a plausible and more importantly reproducible manner.

6.3.2 Advantages of Using the Revised Forensic Process Model During This Case Study

In order to discuss the advantages of using the revised forensic process model as presented in Chapter 5 in this case study, the advantages of using the [KDV15]-model as presented in Section 3.1.2 in this case study are discussed first.

The [KDV15]-model has some notable advantages over other forensic process models when applied to this case study.

At first, the forensic process model aims at identifying the potential sources of forensic evidence through the use of the *Data Streams*. *Communication* takes the primary role in this case study but *Non-volatile Memory* and *Volatile Memory* were also relevant during the investigation into the Maintenance Computer. These *Data Streams* have some structural implications and requirements attached to them which help to conduct the forensic process in a manner in which the greatest possible amount of forensic evidence with the greatest possible quality (by avoiding extensive *Structural Impact*) is available for the investigator.

The **SP** is of great importance to achieve this aim. By identifying the potential sources of evidence beforehand, measures to gather the evidence from this sources with a reduced *Structural Impact* were installed beforehand. This applies to the monitoring of the *Communication* during this case study. The identification of the system baseline behavior was crucial to perform the **DI** and **DA** in an efficient manner. By understanding which network traffic and behavior is innocuous, the identification of malicious behavior was simplified and accelerated.

The great impact of **SP** on this forensic investigation can be evaluated by taking into consideration every piece of evidence found during the forensic investigation.

- Malicious commands to the PLC - *Communication* in *Zone 1* in *Process Section*
Gathering this trace was reduced to simply obtaining the capture from the monitoring device installed in the respective network segment during **SP**. This trace could also have been gathered if physical access to the respective network segment was obtained after the abnormal behavior was noticed. In this case, a part of the evidence would have been lost since it was not recorded. Without any beforehand knowledge about the underlying physical process and the network architecture (which was also obtained during **SP**) **DI** of this evidence would have been delayed.

- Contents of the Maintenance Computer - *Non-volatile Memory* and *Volatile Memory* of Maintenance Computer in *Process Section*
These traces could be gathered after the **DI** and **DA** of the *Communication* in *Zone 1* in *Process Section* were performed. A second **OP** identified the Maintenance Computer as an object of interest for the forensic investigation. Here, only a delay due to a slower **DI** and **DA** of the *Communication* in *Zone 1* in *Process Section* might have lead to a loss of some of the *Volatile Memory* due to continued operation of the Maintenance Computer.

- Exfiltrated technical data - *Communication* in *Engineering Section*
Without **SP** this evidence would have been lost since the *Communication* took place before the incident.

- Hints of activity on EWS - *Non-volatile Memory* of the EWS in *Engineering Section*
However, some traces of the brute force attack might be available in the *Non-volatile Memory* of the EWS without the use of any **SP**. Here, some log entries of the repeated login attempts might be available. In this case study this evidence was also available and used to collaborate the results achieved by investigating the *Communication*.

- Hints of activity on Plant Historian - *Non-volatile Memory* of the Plant Historian in *Engineering Section*
The Plant Historian might retain some log entries of the requested file. However, since this request was innocuous and adheres to the usual behavior in the network this is not necessarily the case.

Without performing an **SP** only a limited amount of evidence would have been available and the investigation would have been more difficult. The possibility to collaborate certain pieces of evidence (the involvement of the EWS) would not have been given. This would have reduced the evidentiary value of the evidence in question.

However, by using this forensic process model the **DO** increased the evidentiary value during the entire process by documenting the actions performed.

During this forensic investigation, different data was handled in a different way. **DT1** required dedicated software to interpret it. In the case of *Communication*, this was the network dissector *Wireshark* which includes the functionality to interpret different protocols, like the FTP session (**DT7**) or the Telnet connection attempts (also **DT7**). Even without being able to interpret the Telnet protocol, the brute force attack could be identified by reviewing the meta data of the packets (**DT3**). Interpreting data about the physical process required additional modules. The **SP** identified these modules before the incident and hence allowed for an installation of these modules on the forensic workstation in order to accelerate the forensic process.

Here, the revised forensic process model includes **DT9** to describe this data about the physical process. Tools attached to this *Data Type* can be selected to interpret **DT9**. In this case study this tool was *s7commwireshark*. During the investigation, **DT8** seemed relevant when the EWS was considered as the source for the attackers specific knowledge (*Potential Source 4*). Here, documentation was exfiltrated. However, the attacker could have also exfiltrated the program code of the PLC which would have required specialized tools to investigate. Documentation and program code are both **DT8**.

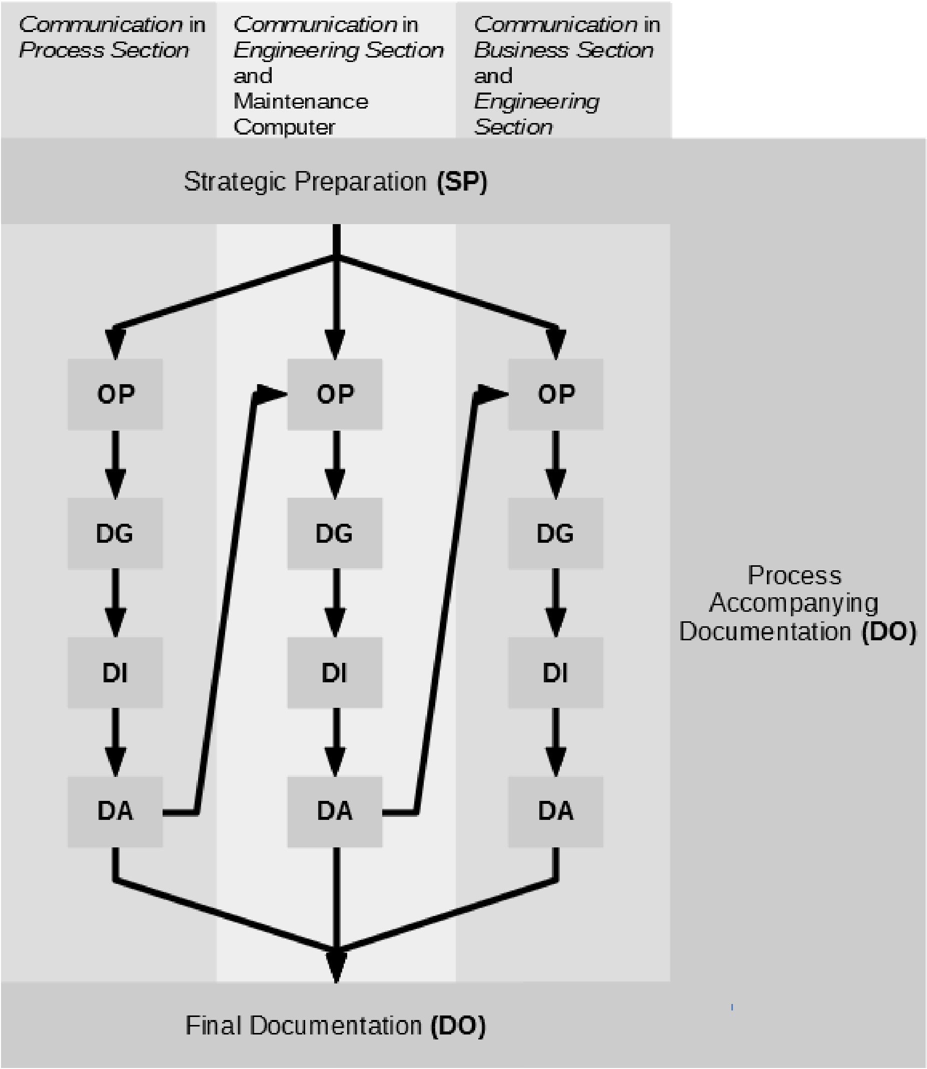

Communication in Process Section
Communication in Engineering Section and Maintenance Computer
Communication in Business Section and Engineering Section
Strategic Preparation (SP)
OP
DG
DI
DA
OP
DG
DI
DA
OP
DG
DI
DA
Process Accompanying Documentation (DO)
Final Documentation (DO)

Figure 6.6: Overall structure of the forensic investigation during the case study

This is an advantage of the revised model. A far greater advantage comes from the altered overall structure of the forensic process. The inclusion of a loop from **DA** to **OP** allowed for a new evaluation on whether the ICS should be shut down once new facts about the incident was known. In addition, it allowed for an extension of the forensic investigation to other

Table 6.1: *Data Types* from different *Data Streams* used in the various *Investigation Steps* during the case study - *N* denotes *Non-volatile Memory*, *V* denotes *Volatile Memory* and *C* denotes *Communication*

Data Type	**SP**			**OP**			**DG**			**DI**			**DA**			**DO**		
Data Stream	N	V	C	N	V	C	N	V	C	N	V	C	N	V	C	N	V	C
DT1			✓						✓									✓
DT2	✓															✓		
DT3			✓									✓			✓			✓
DT4																		
DT5	✓		✓									✓			✓	✓		✓
DT6																		
DT7																		
DT8																		
DT9	✓		✓		✓			✓			✓	✓		✓	✓	✓	✓	✓

networks and systems. This is especially important since components in ICS work in complex networks which are usually attached to networks of Desktop IT (Security Level 4 in the case study).

During the case study, the need to return to the **OP** arose two times. During the first return to the **OP** new information on the cause of the abnormal behavior showed that it was not necessary to shut down the ICS. In addition, the investigation was extended to different networks and systems upon both returns to **OP**. The overall structure of the forensic process as performed during the case study is shown in Figure 6.6. During each return of **OP** knowledge and methods prepared during the **SP** were taken into consideration to judge if the ICS has to be shut down and to decide which investigative actions to take next. After the malicious commands transmitted from the Maintenance Computer in *Zone 1* of the *Process Section*, the *Communication* of the *Engineering Section* and the *Volatile Memory* (and after obtaining the **DT6** also the *Non-volatile Memory*) of the Maintenance Computer were identified for further investigation. Shutting down the system was no longer under consideration since a fix to the abnormal behavior presented itself. After the identification of the data exfiltration during the investigation of the *Communication* in the *Engineering Section* the EDC in the *Business Section* was identified as the next target for further investigation. Additional investigation into the *Non-Volatile Memory* of the systems in the *Engineering Section* was considered to collaborate the gathered evidence. Since the systems in the *Engineering Section* are not constantly required for the full operation of the ICS a brief unavailability of the systems in order to perform **DG** on their *Non-Volatile Memory* is possible. All the actions performed during this investigation are documented and the results compiled into a final report.

This would not be possible in the [KDV15]-model since it does not support such a return to the **OP**.

The structured nature of the forensic process in the [KDV15]-model also allows for a structured description of the forensic process. During the case study, various *Data Types* were used. These can be mapped to the *Investigation Steps*:

- **SP**
  During the **SP** various data was collected in order to prepare for a potential incident. Some of this data was necessary to understand the network infrastructure (by creating a network map as shown in Figure 6.1) and the organizational processes. Other data was necessary in order to baseline the physical processes and the network behavior.

  To achieve an understanding of the network infrastructure, the collection of **DT1** and **DT5** was necessary. This data was obtained from the *Non-volatile Memory* of the involved system (or from the documentation of these systems which informed the configuration of these systems in the first place).

  The baseline behavior of the overall network was obtained by gathering the *Communication* within the network in the form of **DT1**. This **DT1** was then interpreted to obtain information about the used network protocols (**DT3**). This also confirmed the **DT5** identified early.

  For the baseline behavior of the physical process, reviewing **DT9** was necessary. This was interpreted from the **DT2** of the *Communication*. However, the *Non-Volatile Memory* of the Plant Historian would also be able to supply this **DT9**. Both could be used in conjunction.

  The understanding of the control processes and the overall architecture enabled the creation of a *Criticality Map* as provided in Figure 6.2.

- **OP**
  The **OP** was driven by $IF_{ICS.Scen.4}$. The prevention of catastrophic damage was a high priority. In order to understand if catastrophic damage was imminent **DT9** was the source of information. Indeed, the investigation was triggered by some abnormal behavior noticeable in **DT9**. In this case, the **DT9** originated from the *Non-Volatile Memory* of the Main Control HMIs.

  Based on the preparation performed in the **SP**, the decision was made to access the *Communication* in the *Process Section* since this has no interference with the physical process when using the already installed monitoring devices (as identified by the *Criticality Map*). In addition, **DT9** as available in the Main Control Room HMIs was selected for continued monitoring.

- **DG**
  The **DG** saw the gathering of **DT1** from the *Communication* in the *Process Section* and the monitoring of **DT9** from the *Volatile Memory* of Main Control Room HMI 1 and 2. Capturing the *Communication* from the *Process Section* was realized using the monitoring devices installed during **SP** and extracting the data from these. **DT9** from the *Volatile Memory* of Main Control Room HMI 1 and 2 could be accessed by using the GUI of these components.

- **DI**
  During the **DI**, the gathered **DT1** was dissected in order to obtain **DT3**, **DT5** and **DT9**. **DT3** was used in order investigate the communication behavior between the various systems. **DT9**. The investigation into **DT9** identified the malicious commands influencing the physical process.

- **DA**

  During the **DA** information about the network configuration (**DT5**) and the meta data of the network packets containing the malicious commands (**DT3**) was used to identify the Maintenance Computer as the origin of these commands. The effect of the malicious commands (**DT9**) was confirmed by a comparison with the **DT9** gathered from Main Control Room HMI 1 and 2.

- **DO**

  Although the case study continued with a new **OP** after the altered circumstances at this point of the investigation, here the contents of the **DO** at this point of the investigation should be explored. In essence, the **DO** protocoled all the data collected during the forensic investigation. Hence, it contains all *Data Types* which have been collected during the entire investigation.

  Hence the **DO** contains at this point in time:

  - **DT1** from the *Communication* of the various networks obtained during **SP** (for the baseline)
  - **DT1** from the *Communication* of the *Process Section* obtained during **DG**
  - **DT2** gathered from the *Non-volatile Memory* of the employed components during **SP**
  - **DT3** from the *Communication* of the various networks obtained during **SP** (for the baseline)
  - **DT3** from the *Communication* of the *Process Section* extracted from **DT2** during **DI**
  - **DT5** from the *Communication* of the *Process Section* extracted from **DT2** during **DI**
  - **DT9** from the *Communication* of the various networks obtained during **SP** (for the baseline)
  - **DT9** from the *Non-volatile Memory* of the Plant Historian obtained during **SP** (for the baseline)
  - **DT9** from the *Volatile Memory* of the Control HMI 1 and 2 obtained during **OP** (triggering the investigation)
  - **DT9** from the *Volatile Memory* of the Control HMI 1 and 2 monitored during **DG**
  - **DT9** from the *Communication* of the *Process Section* extracted from **DT2** during **DI**
  - Conclusions based on the **DT3**, **DT5** and **DT9** during **DA**

  This is visualized in Table 6.1.

Such a structured description of the forensic process provides a structured list of the evidence in question and greatly supports documenting the entire forensic process. This is necessary to provide a conclusive picture of the actions performed and the decisions taken. Especially in the case of a judicial proceeding or an audit such a chain of custody is of great importance.

Also, this structured description relies on the fact that the *Data Types* are able to describe the domain in question in full detail. This is only possible due to the adaptation of **DT8** from the old forensic process model to the redefined versions of **DT8** and **DT9**.

Hence, the advantages of using the revised forensic process model are the overall structure of the forensic process which befits the use in the ICS domain and the ability to describe all aspects of the forensic investigation in the ICS domain.

6.4 *Influence Factors* and *Forensic Process Consequences* in This Case Study

This section discusses the relevance of the *Influence Factors* and *Forensic Process Consequences* during the case study. Both represent assumptions about the forensic process in the ICS domains which are either confirmed or denied during the case study. This section starts with the *Influence Factors* and then discusses the *Forensic Process Consequences*.

6.4.1 *Influence Factors* in This Case Study

This case study is also useful to confirm the *Influence Factors* and their impact on the forensic process. An overview on the relevance of the *IFs* during this case study is provided in Table 6.2.

Although various of these *IFs* were pointed out during the conduct of the case study, how they affected the forensic process warrants specific attention which is provided here:

- Low computing power ($IF_{ICS.Comp.2}$)
  The low computing power of some employed components lead to the use of legacy communication protocols like Telnet and FTP which do not provide encryption. This simplified the attack and the forensic investigation.

- Logical access to *Non-volatile Memory* and *Volatile Memory* possible($IF_{ICS.Comp.5}$)
  The *Volatile Memory* of the Main Control Room HMI 1 and 2 was accessed using logical access via the GUI to obtain **DT9**.

- Reliance on vendor-specific knowledge and interfaces ($IF_{ICS.Comp.7}$)
  The vendor-specific s7comm-plus protocol was used to communication **DT9**.

- Lack of tools geared towards forensic use ($IF_{ICS.Comp.8}$)
  No tools for forensic use were available to obtain the **DT9** from the Main Control Room HMI 1 and 2.

- Established network hierarchies ($IF_{ICS.Arch.1}$)
  The case study followed the network hierarchy as described in [NSS11].

- Low bandwidth ($IF_{ICS.Arch.2}$)
  The possibility to use to much of the bandwidth led to the creation of a *Criticality Map* and made the use of methods for **DG** which do use the bandwidth preferable.

- High diversity of deployed (vendor-specific and proprietary) communication protocols ($IF_{ICS.Arch.4}$)
  The vendor-specific s7comm-plus protocol was used to communication **DT9**.

Table 6.2: *Influence Factors* relevant for this case study

IF	relevant in this case study
$IF_{ICS.Comp.1}$	
$IF_{ICS.Comp.2}$	✓
$IF_{ICS.Comp.3}$	
$IF_{ICS.Comp.4}$	
$IF_{ICS.Comp.5}$	✓
$IF_{ICS.Comp.6}$	
$IF_{ICS.Comp.7}$	✓
$IF_{ICS.Comp.8}$	✓
$IF_{ICS.Comp.9}$	
$IF_{ICS.Arch.1}$	✓
$IF_{ICS.Arch.2}$	✓
$IF_{ICS.Arch.3}$	
$IF_{ICS.Arch.4}$	✓
$IF_{ICS.Arch.5}$	✓
$IF_{ICS.Arch.6}$	
$IF_{ICS.Arch.7}$	✓
$IF_{ICS.Arch.8}$	✓
$IF_{ICS.Arch.9}$	✓
$IF_{ICS.Arch.10}$	✓
$IF_{ICS.Arch.11}$	✓
$IF_{ICS.Scen.1}$	
$IF_{ICS.Scen.2}$	✓
$IF_{ICS.Scen.3}$	✓
$IF_{ICS.Scen.4}$	✓
$IF_{ICS.Scen.5}$	✓
$IF_{ICS.Scen.6}$	✓
$IF_{ICS.Atk.1}$	✓
$IF_{ICS.Atk.2}$	✓
$IF_{ICS.Atk.3}$	✓

- Ethernet-based communication protocols are easy to access ($IF_{ICS.Arch.5}$)
  The access to the Ethernet-based communication was easy.

- Ethernet-based communication protocols are easier to interpret ($IF_{ICS.Arch.7}$)
  Ethernet-based communication protocols are vendor-specific but common enough that protocol dissectors are available as shown in this case study.

- Lack of tools geared towards forensic use ($IF_{ICS.Arch.8}$)
  No forensic toolkit to investigate the gathered *Communication* was available.

- Presence of Historians within ICS ($IF_{ICS.Arch.9}$)
  The Plant Historian provided **DT9** to understand the baseline function of the physical process during **SP**.

- Dominance of **DT9** in *Communication* in ICS ($IF_{ICS.Arch.10}$)
  DT9 made of the bulk of *Communication* in the *Process Section*.

- Low and predictable traffic ($IF_{ICS.Arch.11}$)
  The traffic in the *Process Section* and *Engineering Section* was low and predictable simplifying the forensic investigation.

- High component lifetime ($IF_{ICS.Scen.2}$)
  The inclusion of older hardware led to a lack of resources and the reliance on legacy protocols (see $IF_{ICS.Comp.2}$).

- High availability requirements ($IF_{ICS.Scen.3}$)
  The ICS in question was required to continue the physical process and stopping it to perform a *Post-Mortem Investigation* would have led to a shut down of the physical process.

- ICS control physical processes which might end in catastrophic events should the ICS be compromised ($IF_{ICS.Scen.4}$)
  There was the risk of catastrophic damage in this case study as explored during **SP**.

- ICS requires time-critical responses from PLCs ($IF_{ICS.Scen.5}$)
  A *Criticality Map* was created in order to prevent any **DG** which could interfere with the real-time requirements in this scenario.

- Forensic evidence might contain intellectual property ($IF_{ICS.Scen.6}$)
  DT9 and **DT1** was used by the attacker to perform the attack and these *Data Types* are also contained in the **DO**.

- Diverse attacks targeting ICS ($IF_{ICS.Atk.1}$)
  The attack performed in this case study was complex.

- Focus on legitimate but malicious commands ($IF_{ICS.Atk.2}$)
  The attack performed in this case included the insertion of legitimate but malicious commands.

- Close relationship between ICS and Desktop IT ($IF_{ICS.Atk.3}$)
  The investigation included an investigation into the Desktop IT section of the overall network.

Table 6.3: *Forensic Process Consequences* relevant for this case study

FPC	relevant in this case study
$FPC_{ICS.1}$	✓
$FPC_{ICS.2}$	✓
$FPC_{ICS.3}$	✓
$FPC_{ICS.4}$	✓
$FPC_{ICS.5}$	✓
$FPC_{ICS.6}$	✓

Hence, 21 of the 28 *IFs* from ICS were relevant during this case study. Of those that were not relevant, four describe properties of the specific components. Two describe the employed architecture and one describes the scenario.

Of the four *IFs* not relevant in this case study which aligned to the specific components $IF_{ICS.Comp.1}$ was not relevant due to the reliance on *Communication* during the investigation. The only components in which the *Non-volatile Memory* or the *Volatile Memory* were gathered were the Plant Historian and the Maintenance Computer which both have sufficient memory. A reliance on the traces available in the *Non-volatile Memory* or the *Volatile Memory* of the employed PLC was not necessary in this scenario. This also made the difficult physical access to the *Non-volatile Memory* ($IF_{ICS.Comp.4}$) irrelevant. Also, the use of only one PLC prevented $IF_{ICS.Comp.3}$ and $\mathbf{IF}_{ICS.Comp.6}$ from having any relevance for this scenario.

The two*IFs* not relevant in this case study which aligned to the architecture all were irrelevant due to the use of Ethernet-based cable-bound communication. $IF_{ICS.Arch.3}$ and $IF_{ICS.Arch.6}$ describe the potential use of other networking technologies.

$IF_{ICS.Scen.1}$ was irrelevant since the access to *Non-volatile Memory* during this forensic investigation were irrelevant.

Although these seven *IFs* were irrelevant during this investigation, it is easy to consider scenarios where they are of relevance. Consider using different network technologies or investigating an attack which relies on persistent malicious code on a PLC in which an investigation into the *Non-volatile Memory* of the PLC is necessary (see Section 4.4.1).

Hence, the *IFs* for the ICS domain can be considered as confirmed by this case study.

6.4.2 *Forensic Process Consequences* in This Case Study

The case study confirmed all the *Forensic Process Consequences* identified in Chapter 5. An overview on this is provided in Table 6.3. However, a detailed description on how the *FPCs* could be observed during the case study offers more insight:

- Reduced usefulness of *Non-Volatile Memory* ($FPC_{ICS.1}$)
  Non-Volatile Memory was used in a supporting role during **SP** in this case study. The difficulties in accessing the *Non-Volatile Memory* were circumvented by relying on the *Communication* instead.
- Increased importance of *Communication* ($FPC_{ICS.2}$)
  Communication provided the bulk of the forensic evidence used during the investigation in the case study.

- Access to *Non-Volatile Memory* and *Volatile Memory* is usually performed relying on *Communication* ($FPC_{ICS.3}$)
  Requesting historical **DT9** from the Plant Historians *Non-volatile Memory* was done using *Communication*.

- Focus on *Live Forensics* in ICS ($FPC_{ICS.4}$)
  The investigation was performed live to the specifics of the scenario which include the cost associated with shutting down the ICS.

- *Live Forensics* might interfere with industrial objective ($FPC_{ICS.5}$)
  A *Criticality Map* was created to identify means to perform a forensic investigation without interfering with the physical process.

- Need to interpret **DT9** in ICS ($FPC_{ICS.6}$)
  DT9 was of high relevance during the forensic investigation. Without the ability to interpret **DT9** the forensic investigation would have faced great difficulties in not being able to confirm the impact of the malicious commands transmitted by the Maintenance Computer.

Due to the relevance of all the *Forensic Process Consequences*, it can be concluded that the *Forensic Process Consequences* which correspond to the ICS domain are confirmed.

7. Summary

This chapter summarizes and concludes this thesis. Section 7.1 provides an overview on the results of this thesis. Section 7.2 discusses the limitations of the scope of this thesis and feeds into open research questions for future work which is discussed in Section 7.3.

7.1 Results of This Thesis

The overall structure of this thesis is summarized in Figure 7.1. Constructs created during the course of this thesis are marked in **bold** and highlighted in light yellow, the answers to the **Research Questions** are marked in **bold** and highlighted in red. In addition, the answers to the **Research Questions** are summarized in Table 7.1.

Section 2.1 provides a discussion on what digital forensics entails. A working definition for computer forensics is provided. The potential aims of a forensic investigation in the Desktop IT domain as two *Investigative Contexts* (*IC*).

Following this, various topics that represent challenges for the forensic process are discussed. Those are the *Evidence Dynamics in Digital Forensics* (Section 2.1.4), *Error, Uncertainty and Loss in Digital Forensics* (Section 2.1.5) and *Data Protection in Digital Forensics* (Section 2.1.6).

Consideration on the *Legal Guidelines for the Admissibility of Forensic Evidence* (Section 2.1.7) allow for the identification of eight *Admissibility Factors* (*AF*). These *AFs* describe requirements evidence must fulfill in order to be admissible.

These four aspects form challenges for the forensic process and are the foundation for the establishment of four *Process Challenges* (*PC* - see Section 2.3.2.4).

Necessary background is provided by the discussion of the various computational domains relevant for this thesis (Section 2.2). This includes the establishment of *working definitions for the computational domains.*

The discussion on *Activities in Digital Forensics* (Section 2.1.3) anticipates the discussion of various forensic process models in the Desktop IT domain (Section 2.3.1). This review

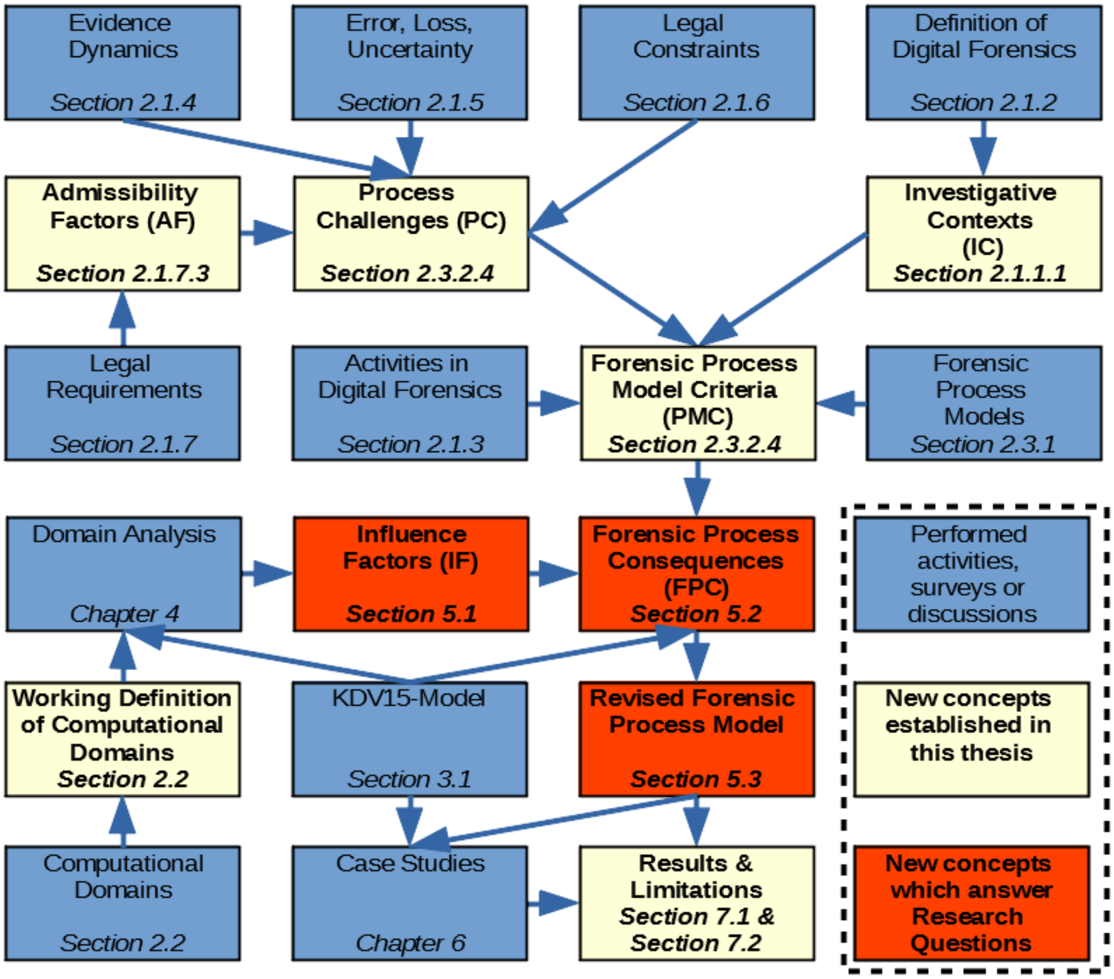


Figure 7.1: Summary of the results of this thesis

of forensic process models identifies characteristics comprehensive models for the forensic process should share. Combined with the considerations on which challenges a forensic process faces (the four *AFs*) and the aims of forensic investigations (the two *ICs*) eight *Forensic Process Model Criteria* (*PMC* - Section 2.3.2.4) are established. These describe criteria a comprehensive forensic process model should address.

The [KDV15]-model for the forensic process is introduced in Section 3.1. This introduction includes the history of the forensic process model. This history is necessary to understand the model and its driving concepts in entirety. The [KDV15]-model consists of six *Investigation Steps*, eight *Data Types* and six *Classes of Methods*. The central concepts of *Data Streams* and *Structural Impact* (which are not included in [KDV15]) are introduced and explored in the scope of various publications on this model - most of them by the author of this thesis. The exploration of the [KDV15]-model ends with a comparison between the model and the *PMC* in Section 3.1.2.7. This discussion concludes, that the [KDV15]-model covers the process side of the *PMC* comprehensively. The *PMC* which align to the *PC* and hence refer to specific tools are not addressed as comprehensively. However, the [KDV15]-model can be considered a comprehensive computer forensic process model.

Concepts from the [KDV15]-model are then used to perform the domain analysis on the three computational domains in Chapter 4. This domain analysis is based on the *working definitions for the computational domains*. It is explored which *Data Streams* are available in the specific domains and how the various *Investigation Steps* of the forensic process can be performed in these domains.

This analysis leads to the identification of 29 (ICS domain) plus 25 (Automotive IT domain) *Influence Factors* (*IF*) which impact the conduct of forensics in these domains (see Section 5.1). These 54 *Influence Factors* represent the answer to **Research Question 1**. A compiled list of *Influence Factors* is available in Section A.1. While some of these *Influence Factors* seem obvious even obvious factors have to be stated when they influence the forensic process and differ from the specifics of the Desktop IT.

These *IFs* are used to describe how similar the investigated domains are. They share 22 of the *IFs* (see Section A.1.3). Five *IFs* slightly differ between these domains (see Section A.1.4). Five are unique to the ICS domain and three to the Automotive IT domain (see Section A.1.5). The IT domain only shares seven (and an additional two to a limited extent) of the *IFs* with any of the investigated domains, showing the difference in properties which influence the forensic process in these domains.

The *IFs* are then used to explore how they impact the forensic process as described in the [KDV15]-model (Section 5.2). This exploration leads to the identification of 6 (ICS domain) plus 6 (Automotive IT domain) *Forensic Process Consequences* (*FPC*). These *FPCs* influence fundamental dynamics of the forensic process. In addition, they answer **Research Question 2**. The compiled list of *Forensic Process Consequences* is provided in Section A.2.

The *FPCs* lead to the adaptation of the [KDV15]-model as performed in Section 5.3. This includes alterations on the macro-level (e.g. introducing new *Data Types* or altering the definition of *Investigation Steps*) as well as on a micro-level (e.g. introducing or altering specific concepts to handle a certain *Investigation Step* in a way that befits the cyber-physical domain better). These alterations are aimed to be performed in a way, that the model does not lose its use for classical Desktop IT. Hence, the changes are in the form of alterations and additions and not in form of omissions.

This adaptation requires no changes to the *Investigation Steps*, besides a clarification to include *Process Accompanying Documentation* and *Final Documentation* within the *Documentation*. However, the overall structure of the forensic process has been revised and altered to account for the possibility to extend the investigation to additional networks if required. The *Data Types* require some notable adaptation to the investigated domains. The most exhaustive of these is the splitting up of **DT8** into two different *Data Types* and the redefinition of **DT7**. The *Classes of Methods* require some clarification but no redefinition. The concepts of *Structural Impact* and *Data Streams* have proven valuable concepts during the course of this thesis and need no fundamental alteration. They have received condensed definitions. However, the *Data Streams* require some additional discussion on the scope of *Live Forensics*, which has been provided in Section 5.3.4.2. Selected aspects of the *Classification Scheme for forensic tools and methods* have been used during the domain analysis. Due to this selection use, no exhaustive alteration seems necessary at this point of time. The revisited state of the [KDV15]-model as presented in Section 5.3.6 provides the answer to **Research Question 3**. The entirety of the revised core components of the [KDV15]-model is provided in Section A.3.

Table 7.1: Influence of the *Influence Factors* from Automotive IT to the components of the [KDV15]-model for the forensic process

Identifier	
Research Question	
Answer	answered
Research Question 1:	
What technical or organizational properties do the domains of (potentially networked) Desktop IT, ICS and Automotive IT have that impact forensics?	
29+25 *Influence Factors* (Section 5.1)	✓
Research Question 2:	
How do these properties impact forensic procedures and methods discussed in the [KDV15]-model?	
6+6 **Forensic Process Consequences** (Section 5.3)	✓
Research Question 3:	
How do methods and procedures from the [KDV15]-model (representing computer forensics in the Desktop IT domain) be adapted or altered to be used in the ICS and Automotive IT domains?	
Adapted [KDV15]-model (Section 5.3.6)	✓

An overview on the state of the **Research Questions** is provided in Table 7.1. All the **Research Questions** were addressed and answered during the course of this thesis.

Finally, a case study was performed in Chapter 6 with the aim of identifying the limitations of the work done during the course of this thesis. This case study confirmed the *IFs* and *FPCs* during a complex investigation into the ICS domain in an attack scenario compiled by ICS security experts. The forensic process as adapted during the course of this thesis is applicable to the investigated domains.

The conduct of the case study identified the advantages of using the revised forensic process model. The main advantages are the altered overall structure of the forensic process which allows for a repeat of the **OP** and the ability to describe all aspects of the forensic investigation in the ICS domain.

21 out of 28 *Influence Factors* and 6 out of 6 *Forensic Process Consequences* for the ICS domain were relevant during the case study. The irrelevant *IFs* were mostly due to the reliance on *Communication* in this scenario, which circumvented some challenges with *Non-volatile Memory* in the ICS domain, and the reliance on an Ethernet-based cable-bound network which makes any *IFs* describing the influence of different network technologies irrelevant. Hence, the overall process to identify these *Influence Factors* and *Forensic Process Consequences* was proven useful by its success in the ICS domain.

7.2 Limitations

The results achieve in this thesis have some limitations. Some of these limitations have to be addressed in future research work in order to maintain the usefulness of the forensic process model as presented and adapted during this thesis. Such aspects will be discussed in Section 7.3.

The limitations can roughly be grouped into four categories: components or aspects of the investigated domains not included in this thesis, future trends in the specific domains, generalization of observations and results as well as additional aspects. This section is structured along those four categories.

7.2.1 Components or Aspects of the Investigated Domains not Included in This Thesis

Some components available in the investigated domains were not discussed in detail during the course of this thesis. The aim of this thesis was to investigate and analyze the basic building blocks and the most common components in each of the investigated domains in lieu of various special case. Namely, the components that warrant additional investigation in the scope of other research work are the Human-Machine-Interface (HMI), commercial grade Historians, the In-Vehicle-Infotainment (IVI) and smart sensors.

Human-Machine-Interface (HMI)

The HMI is a component of ICS tasked with providing the operator with information about the physical process and allowing the operator to control the process. While HMIs and their *Communication* with other ICS components is included in the analysis of ICS architectures (Section 4.2.2), no dedicated HMI is considered during the exploration of *Non-volatile Memory* and *Volatile Memory* performed during the analysis of ICS components (see Section 4.1.2). Such an investigation would have to cover various different HMIs, since these come in various grades of complexity. Some of them are hardware components with dedicated software, while others are software solutions. A review on the current state of the market shows, that the trend points towards the use of software solutions on general purpose operation systems like *2711P PanelView^{TM}Plus 6 Graphic Terminals*[1] or *SIMATIC WinCC V7*[2]. In the case of software running on general purpose operation system, a dedicated review of access to *Non-volatile Memory* and *Volatile Memory* would not be necessary since methods from the domain of Desktop IT could be applied (although general *IFs* from the investigated domains, like a more rugged shielding, would still apply and complicate some means to access). Of higher academic interest are the HMIs employing specialized hardware, operating environments and software.

Commercial Grade Historians

As with the HMIs, Historians were only covered in passing during the analysis. While they played a major role during the discussion of ICS architectures in Section 4.2.2 no dedicated Historian was investigated in detail. Again, there exist various forms of Historians, like software which runs on general purpose operating system (like *Prosys OPC UA Historian*[3]) as well as dedicated hardware Historians employing specialized hardware, operating environments and software. Again, the later are of most interest from the point of view of research.

[1] https://ab.rockwellautomation.com/Graphic-Terminals, 29/05/2020

[2] https://new.siemens.com/global/en/products/automation/industry-software/automation-software/scada/simatic-wincc-v7.html, 29/05/2020

[3] https://www.prosysopc.com/products/opc-ua-historian/, 29/05/2020

In-Vehicle-Infotainment (IVI)

IVIs were briefly discussed in Section 2.3.4.3. These systems provide passengers of vehicles with information and infotainment in providing multimedia functionality. As such, they appear as a kind of Desktop IT inside the Automotive IT environment. In terms of characteristics they stand between these two domains. Since these IVIs are not a typical specimen, they were not discussed during the analysis of Automotive IT (see Section 4.1.3). However, as Section 2.3.4.3 shows, these IVIs might be useful during the forensic process. This special case has been explored by [SCDLK19] and [LEA16] but might still warrant further research. Associated with this aspect is the inclusion of potential online systems with permanent connection to the IVI.

Smart Sensors

Smart sensors are sensors with a comparatively high level of complexity and resources in terms of memory and processor performance. The *Non-volatile Memory* and *Volatile Memory* of these smart sensors might contain forensic evidence. Section 4.1.2.6 discusses these smart sensors, but none of these smart sensors was investigated in detail during the course of this thesis. Smart sensors could be available in both investigated domains.

7.2.2 Future Trends in the Investigated Domains

The IT in the respective domains is evolving. However, future trends are hard to predict.

The rise of *Industry 4.0* seems to point towards the usage of more wireless networks and smart sensors in ICS ([WEG19]). Although the diversity of scenarios in which ICS are used makes it doubtful that wireless networks will be used in all ICS environments (see Section 4.3.1), a profound change in architecture could affect the forensic process in ICS. Also, an increasing amount of smart sensors would warrant additional attention to these components. However, currently, these trends do not seem to undermine any of the principles for the conduct of forensic investigation into ICS discussed in this thesis.

Extensive and numerous discussions with Automotive IT designers and developers indicate the goal of fundamentally simplifying automotive IT. A recurring theme is introducing a central control unit, which is supposed to combine the functions of different ECUs. Another topic of relevance is the shift to different network technologies. However, it is unclear if and when this change happens. Most likely it will be a gradual change, considering the long life cycles of vehicles and the associated Automotive IT. It is clear however, that such a fundamental shift would have a massive impact on the conduct of forensics within Automotive IT.

7.2.3 Generalization of Observations and Results

There are numerous vendors for components in the investigated domains. While this thesis identified the most relevant vendors (in terms of market share) and discussed a range of various components, it could by no means address all components. A focus was undeniably on components from *SIMATIC* due to the availability of these in the research facilities. Additionally, a detailed analysis of a broad range of components from other vendors would improve the results of this thesis by providing a better generalizability of the results performed.

This does not only include computing units and network technologies, but various attached components like dedicated HMIs or commercial grade Historians.

7.2.4 Other Aspects

This thesis focused on the guidelines for admissibility from the point of view of US-American and German law (see Section 2.1.7). Hence, the *Admissibility Factors* (see Section 2.1.7) are designed with these two law systems in mind. Universality was a consideration during the establishment of these *Admissibility Factors* but other juristic systems might pose different or additional requirements on the admissibility of evidence. For the discussion on the requirements for data protection in forensics (see Section 2.1.6) mostly European and German law was referred to. Other juristic systems might be less strict or stricter. This would impact the requirements and constraints for the gathering of evidence during a forensic investigations.

7.3 Future Work

This section discusses future research work and open questions. Some of these questions result from the limitations of this thesis as discussed in Section 7.2. Other objectives for future research which could improve the results of this thesis are included.

7.3.1 Additional Case Studies

From the authors point of view, case studies work best when the scenario, attacks and infrastructure is not designed by the investigator. The case study presented in Chapter 6 had this advantage and provided fundamental insight into the forensic process in the ICS domain in a complex scenario. Such case studies should be performed in order to test the results of this thesis.

This is especially necessary for the Automotive IT domain, since this thesis lacks such a case study. The case study performed for the ICS domain was able to confirm the *Influence Factors* and *Forensic Process Consequences* for the ICS domain. This is also necessary to the Automotive IT. Although the overall process to identify these *Influence Factors* and *Forensic Process Consequences* was proven useful by its success in the ICS domain, it lacks confirmation in the Automotive IT domain.

However, conducting such a case study requires an Automotive IT setup different from the ones used to explore the domain in the first place (during Chapter 4). It also requires a complex attack pattern as even the complex case study performed in ICS did not touch on all potential *IFs* due to not including all the technologies describe by these *IFs*.

7.3.2 Increasing the Generalization of the Observations and Results

Future research will focus on a broader range of components from the investigated domains covering additional vendors. The aim is to also include specialized components, like the ones not investigated in detail during this thesis (see Section 7.2.1). Especially Human-Machine-Interface (HMI), commercial grade Historians and smart sensors are research fields so far not entirely explored by the community.

7.3.3 Future Trends in the Investigated Domains

As the investigated domains change, forensics have to adapt. New technologies require new tools to access forensic evidence. However, a fundamental shift in architecture as would be caused by the use of a central control unit in Automotive IT would necessitate a revision of the forensic process. It is hard to judge when such a shift occurs, but this development would not only bring new challenges - but also the possibility to embed forensic functionality into this new architecture right from the start. This could create a forensic-ready new Automotive IT and negate some of the challenges forensics currently face in the Automotive IT domain.

The increasing importance of smart sensors makes them a more interesting objective for research into their forensic capabilities.

7.3.4 Other Aspects

This section discusses future research work not directly related to the limitations of the research work done in this thesis (see Section 7.2).

7.3.4.1 Improvement of the Concepts of the [KDV15]-Model

The *Classification Scheme for forensic tools and methods* has been discussed in the scope of the investigated domains are some suggestions how to adapt this scheme to the investigated domains have been discussed (see Section 5.3.4.3). However, a comprehensive adaptation of this *Classification Scheme* is open work. If such an adaptation of the entire *Classification Scheme* would support the forensic process in the investigated model is open to debate. During the course of this thesis, some properties (*Structural Impact*, Requirements for the promising usage of a forensic tool, Relevancy of data protection concerns, Tendency for evidentiary value and Protection measures for the integrity of a forensic tool, its input and its output) were used to varying degrees. This selection represents the authors view on the most relevant properties in terms of influence on the forensic process.

Such a revision could help to address those *PMCs* concerned with the *PCs* more comprehensively.

7.3.4.2 Further Research into *Investigate Contexts*

The *Investigative Contexts* which have been introduced in this thesis require some additional discussion and review from other scientist and practitioners. The concept proved useful during this thesis, but is untested outside the scope of theory so far. Some of the potential problems where explored in Section 5.3.5.1.

7.3.4.3 Research Into the Inclusion of SIEMs Into Industrial Control Systems and Automotive IT

The discussion in Section 5.3.3.3 showed that there is currently a lack of mechanisms in order to detect intrusions into the investigated domains. These systems could double as sources for forensic evidence during a forensic investigations. There are various efforts to bring this functionality into both domains.

Research Into the Inclusion of SIEMs into ICS

The discussion in Section 5.3.3.3 presented the research done in [**AHNH20**]. This work explores the potential benefits of including a SIEM (Security Information and Event Management; see [BMZ14]) into the ICS part of the overall infrastructure of a nuclear power plant (NPP). This concept aims at improving detection, incident response on and forensic event reconstruction of cyber attacks in this environment.

Such an ICS-SIEM has to be designed and a resulting demonstrator has to be tested in a fitting environment. In order to provide such a fitting environment additional research into the simulation of complex ICS (on the example for NPP) is conducted. The current state of research is presented in [**AHeS$^+$20**]. This paper discusses the creation of a fully virtualized nuclear power plant in order to perform training and research on incident response, incident recovery and ICS forensics.

A necessary requirement for a meaningful correlation of events relies on an understanding of the influence between the physical process and the respective control signals. This process-specific knowledge has also been cited to perform a forensic investigation into **DT9**. In order to improve the understanding of this relationship, various research is currently conducted. [**GGAW20**], [**eSMP$^+$20**] and [**AH20**] describe some of progress on this topic. [**GGAW20**] focuses on the need of process-specific knowledge during incident response and **SP**. [**eSMP$^+$20**] focuses on the need to have an understanding of the overall physical system to perform incident response and forensic investigation. [**AH20**] focuses on the Steam Turbine Governing System - a subsystem of such an ICS - in order to explore the subsystem and its forensic capabilities in detail.

Extended Event Logging in the Automotive IT Domain

A topic of many considerations is the inclusion of Extended Event Logging in the Automotive IT. Such an approach is currently in the state of design considerations and has so far not been implemented in any kind of demonstrator.

Basically, this Extended Event Logging is supposed to log certain events that can be observed on the communication networks within Automotive IT. As discussed during the course of this thesis, *Communication* has a very high relevance in Automotive IT ($FPC_{ICS.2}$).

Three basic questions have to be answered to design such an Extended Event Logging:

- *Question 1:* How are the potentially relevant events observed?
- *Question 2:* Which events are relevant?
- *Question 3:* How are the relevant events stored?

Question 1 can be answered fairly easily. In the current automotive IT, the only place where all communication networks converge is the Gateway ECU (see Section 4.2.3.1). Hence, at this location all events observable on the communication networks can be observed. The Extended Event Logging could be either implemented in the Gateway ECU or be attached as a separate piece of hardware to a monitor port provided by the Gateway ECU.

Question 2 is more difficult. Basically, there are two potential approaches to identify interesting events. One is to monitor for certain type of events based on signatures or identifiers. The second one is the use of heuristics. Section 5.3.3.3 already discussed the use of heuristics to identify shifts in the behavior of the system. So far, this approach seems useful, especially due to the orderly nature of network traffic in Automotive IT ($IF_{AMI.Arch.10}$). However, additional research has to be performed in order to assess the potential of this approach with certainty.

A fundamental question is, if these signatures and heuristic models should be predefined, self-learning or updated externally.

Question 3 requires the same level of robustness against physical destruction as is the case with Event Data Recorders (EDRs) (see Section 2.3.4.1). Since the events would include more data volume than those stored by an EDR, it would require substantial larger amounts of data.

The component itself must be protected against manipulation from outside in order to ensure the forensic usability of the data collected. This includes both the manipulation of the recorded data, the interface to a reader for said data and the component itself.

As regards the protection against manipulation of the component itself, reference is made here to existing protection concepts for protecting components in Automotive against manipulation.

The protection of the inputs could be achieved by combining the device with the gateway control device. In this case, it would not be technically possible for a signal that passes through the gateway to be unknown to the Extended Event Logging mechanism. If the Extended Event Logging mechanism is attached separately it might be disconnected or the communication connection manipulated.

The Extended Event Logging mechanism would also require an interface for a reader to access the stored data. Due to reasons of resource limitations and the general lack of security on the CAN bus would dictate the use of a specialized interface which could provide integrity and authenticity during the transfer ($IF_{AMI.Arch.8}$ and Section 5.3.5.2).

Furthermore, it could be considered to include IPS functionality in such a system by blocking malicious messages if the Extended Event Logging is implemented in the Gateway ECU.

A. Appendix: Answers to the Research Questions

A.1 Research Question 1: List of Influence Factors

- **Research Question 1**: What technical or organizational properties do the domains of (potentially networked) Desktop IT, ICS and Automotive IT possess that impact forensics?

A.1.1 List of Influence Factors from the Industrial Control System Domain

- Low memory capacity ($\mathbf{IF}_{ICS.Comp.1}$)

- Low computing power ($\mathbf{IF}_{ICS.Comp.2}$)

- High diversity of the domain in terms of software ($\mathbf{IF}_{ICS.Comp.3}$)

- Difficult physical access to *Non-volatile Memory* ($\mathbf{IF}_{ICS.Comp.4}$)

- Logical access to *Non-volatile Memory* and *Volatile Memory* possible($\mathbf{IF}_{ICS.Comp.5}$)

- Availability of *Data Types* highly dependent on the specific systems complexity ($\mathbf{IF}_{ICS.Comp.6}$)

- Reliance on vendor-specific knowledge and interfaces ($\mathbf{IF}_{ICS.Comp.7}$)

- Lack of tools geared towards forensic use ($\mathbf{IF}_{ICS.Comp.8}$)

- Varying complexity of ICS components ($\mathbf{IF}_{ICS.Comp.9}$)
- Established network hierarchies ($\mathbf{IF}_{ICS.Arch.1}$)
- Low bandwidth ($\mathbf{IF}_{ICS.Arch.2}$)
- High diversity of deployed (vendor-specific and proprietary) cable-based carrier mediums ($\mathbf{IF}_{ICS.Arch.3}$)
- High diversity of deployed (vendor-specific and proprietary) communication protocols ($\mathbf{IF}_{ICS.Arch.4}$)
- Ethernet-based communication protocols are easy to access ($\mathbf{IF}_{ICS.Arch.5}$)
- Wireless communication protocols are easy to access ($\mathbf{IF}_{ICS.Arch.6}$)
- Ethernet-based communication protocols are easier to interpret ($\mathbf{IF}_{ICS.Arch.7}$)
- Lack of tools geared towards forensic use ($\mathbf{IF}_{ICS.Arch.8}$)
- Presence of Historians within ICS ($\mathbf{IF}_{ICS.Arch.9}$)
- Dominance of **DT9** in *Communication* in ICS ($\mathbf{IF}_{ICS.Arch.10}$)
- Low and predictable traffic ($\mathbf{IF}_{ICS.Arch.11}$)
- Reduced component accessibility ($\mathbf{IF}_{ICS.Scen.1}$)
- High component lifetime ($\mathbf{IF}_{ICS.Scen.2}$)
- High availability requirements ($\mathbf{IF}_{ICS.Scen.3}$)
- ICS control physical processes which might end in catastrophic events should the ICS be compromised ($\mathbf{IF}_{ICS.Scen.4}$)
- ICS requires time-critical responses from PLCs ($\mathbf{IF}_{ICS.Scen.5}$)

- Forensic evidence might contain intellectual property ($\mathbf{IF}_{ICS.Scen.6}$)

- Diverse attacks targeting ICS ($\mathbf{IF}_{ICS.Atk.1}$)

- Focus on legitimate but malicious commands ($\mathbf{IF}_{ICS.Atk.2}$)

- Close relationship between ICS and Desktop IT ($\mathbf{IF}_{ICS.Atk.3}$)

A.1.2 List of Influence Factors from Automotive IT

- Low memory capacity ($\mathbf{IF}_{AMI.Comp.1}$)

- Low computing power ($\mathbf{IF}_{AMI.Comp.2}$)

- Memory wear reduces reliance on *Non-volatile Memory* ($\mathbf{IF}_{AMI.Comp.3}$)

- Difficult physical access to *Non-volatile Memory* ($\mathbf{IF}_{AMI.Comp.4}$)

- Logical access to *Non-volatile Memory* and *Volatile Memory* possible ($\mathbf{IF}_{AMI.Comp.5}$)

- Reduced availability of *Data Types* ($\mathbf{IF}_{AMI.Comp.6}$)

- Reliance on vendor-specific knowledge and interfaces ($\mathbf{IF}_{AMI.Comp.7}$)

- Lack of tools geared towards forensic use ($\mathbf{IF}_{AMI.Comp.8}$)

- ECUs are usually not shut down($\mathbf{IF}_{AMI.Comp.9}$)

- Established and simple network hierarchies ($\mathbf{IF}_{AMI.Arch.1}$)

- Low bandwidth ($\mathbf{IF}_{AMI.Arch.2}$)

- High diversity of deployed (vendor-specific and proprietary) cable-based carrier mediums ($\mathbf{IF}_{AMI.Arch.3}$)

- High diversity of deployed (vendor-specific and proprietary) communication protocols ($\mathbf{IF}_{AMI.Arch.4}$)
- CAN-based communication is easy to physically access ($\mathbf{IF}_{AMI.Arch.5}$)
- Reliance on vendor-specific knowledge ($\mathbf{IF}_{AMI.Arch.6}$)
- Lack of tools geared towards forensic use ($\mathbf{IF}_{AMI.Arch.7}$)
- Doubtful Origin Authenticity in CAN networks ($\mathbf{IF}_{AMI.Arch.8}$)
- Dominance of **DT9** in *Communication* in Automotive IT ($\mathbf{IF}_{AMI.Arch.9}$)
- Low and predictable traffic ($\mathbf{IF}_{AMI.Arch.10}$)
- Reduced component accessibility ($\mathbf{IF}_{AMI.Scen.1}$)
- High component lifetime ($\mathbf{IF}_{AMI.Scen.2}$)
- Medium availability requirements ($\mathbf{IF}_{AMI.Scen.3}$)
- Automotive IT control physical processes which might cause risk to limb and life should the Automotive IT be compromised ($\mathbf{IF}_{AMI.Scen.4}$)
- Automotive IT must fulfill (soft) real-time requirements ($\mathbf{IF}_{AMI.Scen.5}$)
- Focus on legitimate but malicious commands ($\mathbf{IF}_{AMI.Atk.1}$)

A.1.3 Influence Factors Shared in Industrial Control Systems and Automotive IT

- $IF_{ICS.Comp.1}$ and $IF_{AMI.Comp.1}$
- $IF_{ICS.Comp.2}$ and $IF_{AMI.Comp.2}$
- $IF_{ICS.Comp.4}$ and $IF_{AMI.Comp.4}$
- $IF_{ICS.Comp.5}$ and $IF_{AMI.Comp.5}$
- $IF_{ICS.Comp.7}$ and $IF_{AMI.Comp.7}$

- $IF_{ICS.Comp.8}$ and $IF_{AMI.Comp.8}$
- $IF_{ICS.Arch.1}$ and $IF_{AMI.Arch.1}$
- $IF_{ICS.Arch.2}$ and $IF_{AMI.Arch.2}$
- $IF_{ICS.Arch.3}$ and $IF_{AMI.Arch.3}$
- $IF_{ICS.Arch.4}$ and $IF_{AMI.Arch.4}$
- $IF_{ICS.Arch.8}$ and $IF_{AMI.Arch.7}$
- $IF_{ICS.Arch.10}$ and $IF_{AMI.Arch.9}$
- $IF_{ICS.Arch.11}$ and $IF_{AMI.Arch.10}$
- $IF_{ICS.Scen.1}$ and $IF_{AMI.Scen.1}$
- $IF_{ICS.Scen.2}$ and $IF_{AMI.Scen.2}$
- $IF_{ICS.Scen.5}$ and $IF_{AMI.Scen.5}$
- $IF_{ICS.Atk.2}$ and $IF_{AMI.Atk.1}$

A.1.4 Slightly Different Influence Factors in Industrial Control Systems and Automotive IT

- $IF_{ICS.Comp.6}$ and $IF_{AMI.Comp.6}$
- $IF_{ICS.Arch.5}$ and $IF_{AMI.Arch.5}$
- $IF_{ICS.Arch.7}$ and $IF_{AMI.Arch.6}$
- $IF_{ICS.Scen.3}$ and $IF_{AMI.Scen.3}$
- $IF_{ICS.Scen.4}$ and $IF_{AMI.Scen.4}$

A.1.5 Influence Factors Unique to Either Industrial Control Systems or Automotive IT

- $IF_{ICS.Comp.3}$
- $IF_{ICS.Comp.9}$
- $IF_{ICS.Arch.6}$
- $IF_{ICS.Arch.9}$
- $IF_{ICS.Scen.6}$
- $IF_{ICS.Atk.1}$
- $IF_{ICS.Atk.3}$
- $IF_{AMI.Comp.3}$
- $IF_{AMI.Comp.9}$
- $IF_{AMI.Arch.8}$

A.2 Research Question 2: List of Forensic Process Consequences

- **Research Question 2**: How do these properties impact forensic procedures and methods discussed in the [KDV15]-model?

A.2.1 List of Forensic Process Consequences From the Industrial Control System Domain

- Reduced usefulness of *Non-Volatile Memory* ($\mathbf{FPC}_{ICS.1}$)
- Increased importance of *Communication* ($\mathbf{FPC}_{ICS.2}$)
- Access to *Non-Volatile Memory* and *Volatile Memory* is usually performed relying on *Communication* ($\mathbf{FPC}_{ICS.3}$)
- Focus on *Live Forensics* in ICS ($\mathbf{FPC}_{ICS.4}$)
- *Live Forensics* might interfere with industrial objective ($\mathbf{FPC}_{ICS.5}$)
- Need to interpret **DT9** in ICS ($\mathbf{FPC}_{ICS.6}$)

A.2.2 List of Forensic Process Consequences From Automotive IT

- Reduced usefulness of *Non-Volatile Memory* ($\mathbf{FPC}_{AMI.1}$)
- Increased importance of *Communication* ($\mathbf{FPC}_{AMI.2}$)
- Access to *Non-Volatile Memory* and *Volatile Memory* is usually performed relying on *Communication* $\mathbf{FPC}_{AMI.3}$)
- Focus on *Live Forensics* in Automotive IT ($\mathbf{FPC}_{AMI.4}$)
- *Live Forensics* might interfere with the driving task ($\mathbf{FPC}_{AMI.5}$)
- Need to interpret **DT9** in Automotive IT ($\mathbf{FPC}_{AMI.6}$)

A.3 Research Question 3: Revised Concepts of the [KDV15]-Model for the Forensic Process

- **Research Question 3**: How do methods and procedures from the [KDV15]-model (representing computer forensics in the Desktop IT domain) have to be adapted or altered to be used in the ICS and Automotive IT domains?

A.3.1 *Investigation Steps*

A.3.1.1 Definition of the *Investigation Steps*

- **Strategic Preparation (SP)**:
  measures taken by the operator of an IT-system in order to support a forensic investigation prior to an incident

- **Operational Preparation (OP)**:
  measures of preparation for a forensic investigation after a suspected incident

- **Data Gathering (DG)**:
  measures to acquire and secure digital evidence

- **Data Investigation (DI)**:
  measures to evaluate and extract data for further investigation

- **Data Analysis (DA)**:
  measures for detailed analysis and correlation between digital evidence from various sources

- **Documentation (DO)**:
  measures for the detailed documentation of the proceedings (*Process Accompanying Documentation*) and for the compilation of a report on the incident (*Final Documentation*)

A.3.1.2 Structure of the *Investigation Steps*

- **SP** leads to **OP**
- **OP** leads to **DG**
- **DG** leads to **DI**
- **DI** leads to **DA** and **DG**
- **DA** leads to **DO** and **DI** and **OP**
- **DO** leads to **SP** as a feedback loop

See Figure A.1.

A.3.2 *Data Types*

- **raw data (DT1)**:
  A sequence of bits within the Data Streams of a computing systems not (yet) interpreted.

- **hardware data (DT2)**:
  Data in a computing unit which is not, or only in a limited way, influenced by software.

- **details about data (DT3)**:
  Data added to other data, stored within the annotated chunk of data or externally

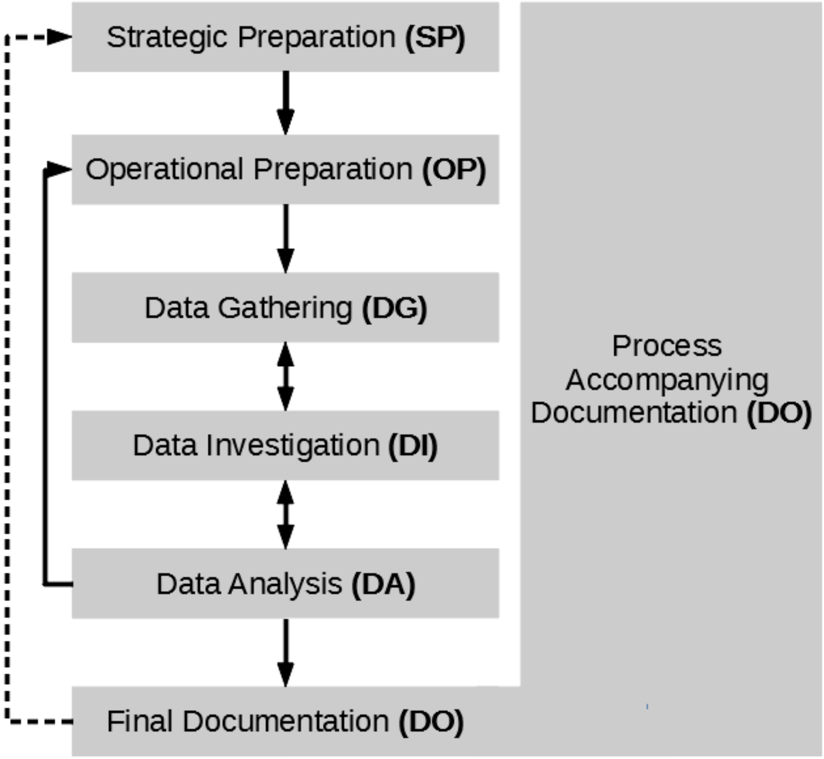

Strategic Preparation **(SP)**
Operational Preparation **(OP)**
Data Gathering **(DG)**
Data Investigation **(DI)**
Data Analysis **(DA)**
Final Documentation **(DO)**
Process Accompanying Documentation **(DO)**

Figure A.1: Revised *Investigation Steps*

- **configuration data** (**DT4**):
  Data which can be changed by software and which modifies the behavior of software and hardware, excluding the communication behavior
- **network configuration data** (**DT5**):
  Data that modifies system behavior with regards to communication
- **process data** (**DT6**):
  data about a running software process within a computing unit
- **session data** (**DT7**):
  data collected by a system during a session, which consist of a number of processes with the same scope and time frame
- **application data** (**DT8**):
  data representing functions needed to create, edit, consume or process content relied to the key functionality of the system
- **functional data** (**DT9**):
  data content created, edited, consumed or processed as the key functionality of the system

A.3.3 *Classes of Methods*

- **Operating System** (**OS**):
  methods provided by the operating system

- **File System (FS)**:
  methods provided by the file system

- **Explicit Means of Intrusion Detection (EMID)**:
  methods provided by additional software with the characteristic of being executed autonomous on a routine basis and without a suspicion of an incident

- **IT Application (ITA)**:
  methods provided by the functions that fulfill the key functionality of the system

- **Scaling of Methods for Evidence Gathering (SMG)**:
  methods to collect additional evidence if a suspicion is raised but unsuited for routine usage in a production environment (due to reasons such as high false positives or high resource demand)

- **Data Processing and Evaluation (DPE)**:
  methods which support a detailed forensic investigation, display, processing and documentation)

A.3.4 *Structural Impact*

Structural Impact (SI):
describes the alteration of the system state caused by the application of forensic methods. This alteration might propagate to connected systems.

A.3.5 *Data Streams*

Non-volatile Memory:
Memory inside a computing unit which maintains its content after the unit is disconnected from its respective power supply.
Volatile Memory:
Memory inside a computing unit which loses its content after the unit is disconnected from its respective power supply.
Communication:
All the data transmitted to other computing units via communication interfaces.

B. Appendix: Definitions

B.1 Digital Forensics

Defined in Section 2.1.2:

Digital Forensics (*DF*):
The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be subject to potential court cases, or disruptive to intended systems operations.

B.2 *Investigative Contexts* (*IC*)

Defined in Section 2.1.2:

Investigative Context 1 (**IC1**):
Investigations into unwanted behavior of the system.
Investigative Context 2 (**IC2**):
Investigations into behavior which has been carried out with the use of computer systems and is subject to a potential court case.

B.3 Aspects of Information Security

Defined in Section 2.1.4:

Integrity:
Integrity refers to a certain piece of evidence. It is achieved when the this piece of evidence is not altered. **Origin Authenticity**:

Origin Authenticity refers to a certain piece of evidence. It is achieved when the origin of this piece of evidence is established beyond doubt. **Entity Authenticity**: *Entity Authenticity refers to an entity or entities considered in a piece of evidence. It is achieved when the identity of these entities are established beyond doubt.*

B.4 Error, Uncertainty and Loss

Defines in Section 2.1.5:

Error:
Error occurs if evidence is incorrectly interpreted or altered before it is interpreted.
Uncertainty:
Uncertainty exists if it cannot be determined whether evidence is integer and authentic.
Loss:
Loss occurs if the entirety of evidence is not collected.

B.5 Desktop IT

Defines in Section 2.2.1:

Desktop IT:
The domain of Desktop IT covers computer systems designed to receive, store, manipulate and transmit data (or information)

B.6 Components and Terms in the Industrial Control System Domain

Defined in Section 2.2.2.2:

Sensor:
Collects information about the environment, e.g. a physical process.
Actuator (or **Actor**):
Manipulates the environment, e.g. a physical process.
Processing Unit (or **Programmable Logic Controller**):
Evaluates the data gathered by sensors and/or gives control signals to actors.
Communication Wiring:
The physical and logical carrier that facilitates communication between sensors, actors and processing units.
Distributed Control System (**DCS**):
A group of processing units which distribute the task of reading sensors, processing data and sending control signals to actors between the various processing units.
Supervisory control and data acquisition (**SCADA**):
A system which collects data about a physical process and presents those to an operator, granting the operator the ability to supervise and influence the process. The

data is gathered from the processing units and the ability to influence the process is granted by sending signals to the respective processing units. Dedicated overrides might gather the data directly from the sensors or might control actors directly (e.g. emergency stop switch).

Industrial Control System (ICS):
A communication network of Actors, Sensors and Processing units geared towards controlling a physical process.

B.7 Components in Automotive IT

Defined in Section 2.2.3.1:

Sensor:
Collects information about the environment, e.g. a physical process

Actuator (or **Actor**):
Manipulates the environment, e.g. a physical process

Processing Unit (or **Electronic Control Unit**):
Evaluates the data gathered by sensors and/or gives control signals to actors

Communication Wiring:
The physical and logical carrier that facilitates communication between sensors, actors and processing units

B.8 *Admissibility Factors (AF)*

Defined in Section 2.1.7.3:

- **Admissibility Factor 0 (AF0)**:
  The evidence helps the trier of fact to determine a fact in issue

- **Admissibility Factor 1 (AF1)**:
  The technique or theory producing (or drawing a conclusion from) the evidence has been peer reviewed

- **Admissibility Factor 2 (AF2)**:
  The technique or theory producing (or drawing a conclusion from) the evidence is generally accepted in the scientific community

- **Admissibility Factor 3 (AF3)**:
  The technique or theory producing (or drawing a conclusion from) the evidence has been tested for reliability

- **Admissibility Factor 4 (AF4)**:
  The technique or theory producing (or drawing a conclusion from) the evidence has a known rate of error

- **Admissibility Factor 5 (AF5)**:
  The technique or theory producing (or drawing a conclusion from) the evidence follows standards and controls
- **Admissibility Factor 6 (AF6)**:
  The technique producing (or drawing a conclusion from) the evidence ensures the integrity and authenticity of the evidence
- **Admissibility Factor 7 (AF7)**:
  The technique producing (or drawing a conclusion from) the evidence documents the process of recording or storing a certain event or piece of information

B.9 *Process Challenges (PC)*

Defined in Section 2.3.2.4:

- **Process Challenge 1 (PC1)**:
  integrity and *authenticity* of traces (see Section 2.1.4)
- **Process Challenge 2 (PC2)**:
  error, uncertainty and loss (see Section 2.1.4)
- **Process Challenge 3 (PC3)**:
  constraints for the collection and use of evidence (see Section 2.1.6)
- **Process Challenge 4 (PC4)**:
  legal requirements for the admissibility of evidence in court (see Section 2.1.7)

B.10 *Forensic Process Model Criteria (PMC)*

Defined in Section 2.3.2.4:

- **Forensic Process Model Criterion 1 (PMC1)**:
  Addresses **PC1** (integrity and authenticity of traces)
- **Forensic Process Model Criterion 2 (PMC2)**:
  Addresses **PC2** (error, uncertainty and loss)
- **Forensic Process Model Criterion 3 (PMC3)**:
  Addresses **PC3** (constraints for the collection and use of evidence)
- **Forensic Process Model Criterion 4 (PMC4)**:
  Addresses **PC4** (*Admissibility Factors*: legal requirements for the admissibility of evidence in court)
- **Forensic Process Model Criterion 5 (PMC5)**:
  Addresses **IC1** and **IC2**
- **Forensic Process Model Criterion 6 (PMC6)**:
  Includes the possibility of a *Pre-Incident Preparation*

- **Forensic Process Model Criterion 7 (PMC7)**:
  Addresses *Live Forensics* and *Post-Mortem Forensics* and includes some guidance to decide when to use which approach

- **Forensic Process Model Criterion 8 (PMC8)**:
  Structures the process based on the logical sequence of the investigation

C. Appendix: Material for the Industrial Control System case study

C.1 Abnormal Operations Analysis and Computer Security Consequence Analysis for the ICS Case Study

C.1.1 Abnormal Operations Analysis

Table C.1 presents the analysis of abnormal behavior of the Condenser in the ICS case study. This analysis was performed during **SP**. Please note that only a redacted version can be presented here. However, this overview should provide an idea on how such an analysis would look like and which elements it should contain. Performing such a abnormal operation analysis requires process-specific knowledge but is necessary to make informed decisions during incident response or a forensic investigation - especially when considering if the physical process or the ICS should be shut down.

C.1.2 Computer Security Consequence Analysis

The computer security consequence analysis has been conducted using the guidance provided in US NUREG/CR-6847 ([PNN04]). To perform a consequence analysis following this guidance involves several steps:

- Step 1: Identify and briefly describe the functionality supported by the asset and outline the way it interacts with the system being analyzed.
- Step 2: Identify and describe the different types of compromise that could have a negative effect on the critical system. This should consider the loss of confidentiality, integrity and availability of the asset.
- Step 3: Examine the consequences to both the critical system being analyzed and the plant. Consequences to the critical system can be rated as none, degraded (able to perform its function but with less reliability), or failed (unable to perform its function). Moreover, impact to the plant is assessed using a three-level scale: low, medium and

Table C.1: Abnormal Operations Analysis for Condenser in ICS case study

Identifier	
Failure Description	Consequences
Failure Scenario 1: Condenser Outlet Pump Failure	
When the three CD outlet pumps fail, in sequence, the CD level increases. [redacted]	This failure has the following consequences: 1. [redacted] 2. [redacted] 3. [redacted] 4. Finally, [redacted]
Failure Scenario 2: Condenser Level Measurement Failure	
[redacted]	[redacted]
Failure Scenario 3: Condenser Cooling Pump Failure	
[redacted]	[redacted]
Failure Scenario 4: Leakage in CD system	
[redacted]	[redacted]

high impact. These impacts should be determined in relation to safety systems, safety support systems, plant security, emergency preparedness, and continuity of power.

Please note that only a redacted version can be presented in Table C.2. However, this overview should provide an idea on how such an analysis would look like and which elements it should include. It lists all digital assets within the given zone, describes their function and the consequences of a potential loss of confidentiality, integrity and availability.

Performing such a abnormal operation analysis requires process-specific knowledge but is necessary to make informed decisions during incident response or a forensic investigation - especially when considering if the physical process or the ICS should be shut down.

C.1.3 Financial Cost Analysis

The following calculation was provided by Dr. Rodney Aparecido Busquim e Silva:

> *„ ANP is a 2,772 MWt (33% efficiency). Therefore, it produces 2772*0,33 * 24 = 21,954 MWh of electricity per day.(Instructors Guidelines for ANP trip due to condenser being compromised. Assuming an average cost of electricity of US$ 110/MWh (March 2019, RPK price - global pretrolprices.com) and an average nuclear generation cost of about US$ 34/MWh, one day off would cause a reduction of about US$ 1.67 million of profit (revenue minus cost - assuming capital cost are already paid off).“*
>
> *[..]*
>
> *„ The 2018 AP1000 base plant capital cost estimation for the mechanical equipment supply, which includes the condenser and other mechanical items, is about US$ 3.7 billion dollars - around 16% of the AP1000 capital cost. Only for this exercise, we have assumed that the condenser cost is assuming to be 20% of the mechanical*

Table C.2: Computer Security Consequence Analysis for Condenser in ICS case study

Digital Asset			
Functional Description			
Step 1	Step 2	Step 3	
Type of Interaction	Digital Compromise	Potential Consequence to Critical System	Consequence to Plant
Siemens PLC			
Provides control of components, such as pumps and valves. *Provide operational information such as alarms and setpoint values.*			
Controls the function of CD based on user program	**Confidentiality**: Digital Information can be read	[redacted]	[redacted] **IMPACT** [redacted]
	Integrity: Digital Signals or set points can be corrupted	[redacted]	[redacted] **IMPACT** [redacted]
	Availability: Could result in deny of control of equipment	[redacted]	[redacted] **IMPACT** [redacted]
Network Switch			
[..]			
Maintenance Computer			
[..]			
Local HMI			
[..]			

equipment supply cost. Therefore, the condenser replacement due to structural failure could represent a US$ 740 million dollars cost. However, structural failures may appear over a long time of operation due, for example, more temperature cycling than designed. “

C.2 Network Captures Used in the ICS Case Study

This section provides information about the network captures during the ICS case study. Please note, that time and date have been intentionally removed.

C.2.1 *Process Section* Baseline Capture

- Name:
  Process-Section-Baseline.pcap
- Capture Location:
  Network Monitor 3
- Hash (MD5):
  D29617D87CF8C4DE39B9BC42E9F1870E
- Hash (SHA256):
  cfc07995dfb45a522f8b1583ca460b72600d119d8a29751cb4e18043799421e9

Table C.3: Statistics of *Process Section* baseline capture - Section C.2.1

Address A			**Address B**		
Overall		**A -> B**		**B -> A**	
Packets	Bytes	Packets	Bytes	Packets	Bytes
192.168.0.1			**192.168.0.18**		
21814	4018988	10907	1681588	10907	2337400
192.168.0.21			**192.168.0.101**		
12658	3742476	6874	2818969	5784	923507
192.168.0.21			**192.168.0.104**		
10033	3418891	5621	2719162	4412	699729
192.168.0.21			**192.168.0.103**		
9912	3383619	5543	2693862	4369	689757
192.168.0.18			**192.168.0.109**		
8559	1148315	3891	608383	4668	539932
192.168.0.18			**192.168.0.21**		
7552	2399656	7552	2399656	0	0
192.168.0.21			**192.168.0.102**		
7265	3096836	4275	2635717	2990	461119
192.168.0.1			**192.168.0.4**		
1929	191520	956	108330	973	83190

- Hash (RIPEMD160):
  4b2896f913bb0040e920ec88b47cd298a6ca2efa

- Hash (SHA1):
  f10a39bf5f4f4b5df90316b18dd9916785e0c7f9

- Format:
  pcapng

- Hardware:
  Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz (with SSE4.2)

- OS:
  32-bit Windows 7 Service Pack 1, build 7601

- Application:
  Dumpcap (Wireshark) 2.6.5 (v2.6.5-0-gf766965a)

- Dropped Packets:
  0 (0.0%)

- Packets:
  79865(100.0%)

- Timespan:
  65.104 s

- Bytes:
  21417519 (100.0%)

- Average Packets per Second:
  1226.7

- Average Packet size:
  268 B

- Average Byte/s:
  328k

- Connection Analysis:
  Section C.2.1

C.2.2 *Process Section* Capture

- Name:
  Process-Section-Abnormal.pcap

- Capture Location:
  Network Monitor 3

- Hash (MD5):
  C97F5B74DC0CEA126B2C6140BDE4D0B4

- Hash (SHA256):
  027851a315a1e4c16044f80ee83008ac4a7f15e84668bf4c810f9b0f2392ef0a

- Hash (RIPEMD160):
  39101e60ee7c64d7d7f42444d05a1592854dbc27

- Hash (SHA1):
  8d92ede4eae8d4911b5c3ff86235ce9577037963

- Format:
  pcapng

- Hardware:
  Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz (with SSE4.2)

- OS:
  32-bit Windows 7 Service Pack 1, build 7601

- Application:
  Dumpcap (Wireshark) 2.6.5 (v2.6.5-0-gf766965a)

- Dropped Packets:
  0 (0.0%)

- Packets:
  157790 (100.0%)

- Timespan:
  124.490

- Bytes:
  41673993 (100.0%)

- Average Packets per Second:
  1267.5

- Average Packet size:
  264 B

- Average Byte/s:
  334k

- Connection Analysis:
  Table C.4

C.2.2.1 Wireshark Filters for Investigating the *Process Section* Capture

- **PLC to Maintenance Computer**:
  ip.addr == 192.168.0.1 && ip.addr == 192.168.0.3

- **PLC to Maintenance Computer** - s7comm plus traffic
  ip.addr == 192.168.0.1 && ip.addr == 192.168.0.3 && s7comm-plus

- **Maintenance Computer to Network Switch**
  ip.addr == 192.168.0.3 && ip.addr == 192.168.0.239

- **Maintenance Computer to Network Switch** - HTTP traffic
  ip.addr == 192.168.0.3 && ip.addr == 192.168.0.239 && http

C.2.2.2 Results of Investigating the Network Traffic Between PLC and Maintenance Computer

Detail view of Packet 34859:

```
Internet Protocol Version 4, Src: 192.168.0.3, Dst: 192.168.0.1
[..]
S7 Communication Plus
    Header: Protocol version=V3
    Integrity part
    Data: Request SetMultiVariables
        Opcode: Request (0x31)
        Reserved: 0x0000
        Function: SetMultiVariables (0x0542)
        Reserved: 0x0000
        Sequence number: 1933
        Session Id: 0x0000038e
        Transport flags: 0x34, Bit2-AlwaysSet?, Bit4-AlwaysSet?, Bit5-AlwaysSet?
        Request Set
            Unknown: 0x00000000
            Item Count: 1
            Number of fields in complete Item-Dataset: 5
            AddressList
```

```
                Item Address [1]: NativeObjects.theMArea_Rid, SYM-CRC=73dd3adf,
ControllerArea.ValueActual, LID=261
            ValueList
                Item Value [1]: (Int) = 80
            Filling byte: 0x00
        ObjectQualifier
        Integrity Id: 1884
        Data unknown: 00000000
    Trailer: Protocol version=V3
```

Detail view of Packet 34862:

```
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.3
[..]
S7 Communication Plus
    Header: Protocol version=V3
    Integrity part
    Data: Response SetMultiVariables
        Opcode: Response (0x32)
        Reserved: 0x0000
        Function: SetMultiVariables (0x0542)
        Reserved: 0x0000
        Sequence number: 1933
        Transport flags: 0x34, Bit2-AlwaysSet?, Bit4-AlwaysSet?, Bit5-AlwaysSet?
        Response Set
        Integrity Id: 3817
        Data unknown: 00000000
    Trailer: Protocol version=V3
```

Detail view of Packet 99975:

```
Internet Protocol Version 4, Src: 192.168.0.3, Dst: 192.168.0.1
[..]
S7 Communication Plus
    Header: Protocol version=V3
    Integrity part
    Data: Request SetMultiVariables
        Opcode: Request (0x31)
        Reserved: 0x0000
        Function: SetMultiVariables (0x0542)
        Reserved: 0x0000
        Sequence number: 1933
        Session Id: 0x0000038e
        Transport flags: 0x34, Bit2-AlwaysSet?, Bit4-AlwaysSet?, Bit5-AlwaysSet?
        Request Set
            Unknown: 0x00000000
            Item Count: 1
            Number of fields in complete Item-Dataset: 5
            AddressList
            ValueList
                Item Value [1]: (Int) = 70
            Filling byte: 0x00
        ObjectQualifier
        Integrity Id: 1884
        Data unknown: 00000000
    Trailer: Protocol version=V3
```

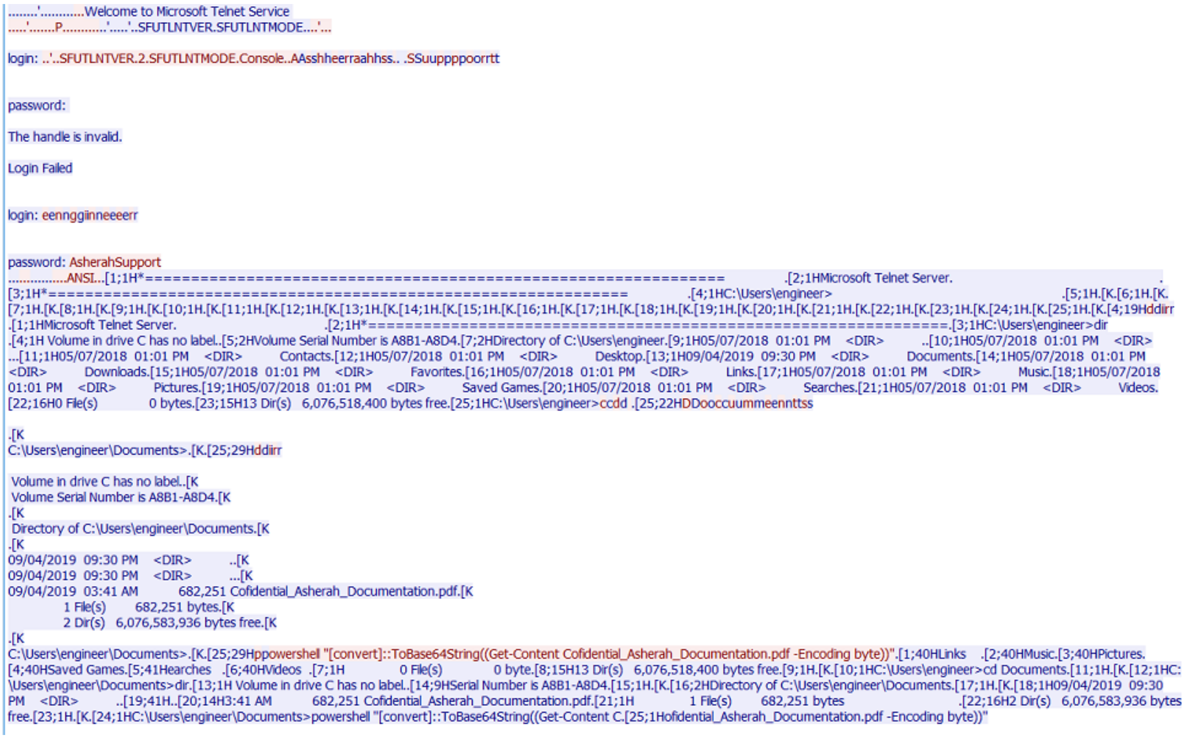
C.2.3 *Engineering Section* Capture

- Name:
  Engineering-Section.pcap
- Capture Location:
  Network Monitor 2
- Hash (MD5):
  E20C919EFDA30F02462A160EB3048E87
- Hash (SHA256):
  52bef469bd809896aaa762182ea17c1261e2729cbeac58cf431844270d85c544
- Hash (RIPEMD160):
  d0d2870fcaa2389b4606333d3fe0e8bfe2ad0011
- Hash (SHA1):
  75015f50649c9cb831b04fa2ad858e17b331b25f
- Format:
  pcapng
- Hardware:
  unknown
- OS:
  unknown
- Application:
  unknown
- Dropped Packets:
  unknown
- Packets:
  33051 (unknown)
- Timespan:
  1532
- Bytes:
  2562580 (unknown)
- Average Packets per Second:
  21.6
- Average Packet size:
  78 B
- Average Byte/s:
  1672k
- Connection Analysis:
  Table C.7

C.2.3.1 Wireshark Filters for Investigating the *Engineering Section* Capture

- **EWS to EDC**:
  ip.addr == 10.3.1.2 && ip.addr == 10.4.1.2
- **EWS to EDC** - successful Telnet connection:
  ip.addr == 10.3.1.2 && ip.addr == 10.4.1.2 && tcp.port == 53578
- **Plant Historian to EDC**:
  ip.addr == 10.3.1.3 && ip.addr == 10.4.1.2
- **Plant Historian to EDC** - FTP command:
  ip.addr == 10.3.1.3 && ip.addr == 10.4.1.2 && ftp

C.2.3.2 Results of Investigating the Network Traffic Between EWS and EDC



```
........'...........Welcome to Microsoft Telnet Service
.....'.......P............'.....'..SFUTLNTVER.SFUTLNTMODE....'...

login: ..'..SFUTLNTVER.2.SFUTLNTMODE.Console..AAsshheerraahhss.. .SSuuppppoorrtt


password:

The handle is invalid.

Login Failed


login: eennggiinneeeerr


password: AsherahSupport
...............ANSI...[1;1H*===============================================================          .[2;1HMicrosoft Telnet Server.                                    .
[3;1H*=============================================================          .[4;1HC:\Users\engineer>                                              .[5;1H.[K.[6;1H.[K.
[7;1H.[K.[8;1H.[K.[9;1H.[K.[10;1H.[K.[11;1H.[K.[12;1H.[K.[13;1H.[K.[14;1H.[K.[15;1H.[K.[16;1H.[K.[17;1H.[K.[18;1H.[K.[19;1H.[K.[20;1H.[K.[21;1H.[K.[22;1H.[K.[23;1H.[K.[24;1H.[K.[25;1H.[K.[4;19Hddirr
.[1;1HMicrosoft Telnet Server.                    .[2;1H*===============================================================.[3;1HC:\Users\engineer>dir
.[4;1H Volume in drive C has no label..[5;2HVolume Serial Number is A8B1-A8D4.[7;2HDirectory of C:\Users\engineer.[9;1H05/07/2018  01:01 PM    <DIR>          ..[10;1H05/07/2018  01:01 PM    <DIR>
...[11;1H05/07/2018  01:01 PM    <DIR>          Contacts.[12;1H05/07/2018  01:01 PM    <DIR>          Desktop.[13;1H09/04/2019  09:30 PM    <DIR>          Documents.[14;1H05/07/2018  01:01 PM
<DIR>          Downloads.[15;1H05/07/2018  01:01 PM    <DIR>          Favorites.[16;1H05/07/2018  01:01 PM    <DIR>          Links.[17;1H05/07/2018  01:01 PM    <DIR>          Music.[18;1H05/07/2018
01:01 PM    <DIR>          Pictures.[19;1H05/07/2018  01:01 PM    <DIR>          Saved Games.[20;1H05/07/2018  01:01 PM    <DIR>          Searches.[21;1H05/07/2018  01:01 PM    <DIR>          Videos.
[22;16H0 File(s)              0 bytes.[23;15H13 Dir(s)   6,076,518,400 bytes free.[25;1HC:\Users\engineer>ccdd .[25;22HDDooccuummeennttss

.[K
C:\Users\engineer\Documents>.[K.[25;29Hddiirr

 Volume in drive C has no label..[K
 Volume Serial Number is A8B1-A8D4.[K
.[K
 Directory of C:\Users\engineer\Documents.[K
.[K
09/04/2019  09:30 PM    <DIR>          ..[K
09/04/2019  09:30 PM    <DIR>          ...[K
09/04/2019  03:41 AM           682,251 Cofidential_Asherah_Documentation.pdf.[K
               1 File(s)        682,251 bytes.[K
               2 Dir(s)   6,076,583,936 bytes free.[K
.[K
C:\Users\engineer\Documents>.[K.[25;29Hppowershell "[convert]::ToBase64String((Get-Content Cofidential_Asherah_Documentation.pdf -Encoding byte))".[1;40HLinks    .[2;40HMusic.[3;40HPictures.
[4;40HSaved Games.[5;41Hearches   .[6;40HVideos  .[7;1H              0 File(s)              0 byte.[8;15H13 Dir(s)   6,076,518,400 bytes free.[9;1H.[K.[10;1HC:\Users\engineer>cd Documents.[11;1H.[K.[12;1HC:
\Users\engineer\Documents>dir.[13;1H Volume in drive C has no label..[14;9HSerial Number is A8B1-A8D4.[15;1H.[K.[16;2HDirectory of C:\Users\engineer\Documents.[17;1H.[K.[18;1H09/04/2019  09:30
PM    <DIR>          ..[19;41H..[20;14H3:41 AM           682,251 Cofidential_Asherah_Documentation.pdf.[21;1H              1 File(s)        682,251 bytes                        .[22;16H2 Dir(s)   6,076,583,936 bytes
free.[23;1H.[K.[24;1HC:\Users\engineer\Documents>powershell "[convert]::ToBase64String((Get-Content C.[25;1Hofidential_Asherah_Documentation.pdf -Encoding byte))"
```

Figure C.1: Reconstructed Telnet session between EDC and EWS

Detail view of Packet 143:

```
[..]
Transmission Control Protocol, Src Port: 49614, Dst Port: 23, Seq: 0, Len: 0
    Source Port: 49614
    Destination Port: 23
    [Stream index: 5]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 4103579759
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
    Flags: 0x002 (SYN)
[..]
```

Detail view of Packet 145:

```
[..]
Transmission Control Protocol, Src Port: 49615, Dst Port: 23, Seq: 0, Len: 0
    Source Port: 49615
    Destination Port: 23
    [Stream index: 6]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 3470665438
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
    Flags: 0x002 (SYN)
[..]
```

Detail view of Packet 147:

```
[..]
Transmission Control Protocol, Src Port: 49616, Dst Port: 23, Seq: 0, Len: 0
    Source Port: 49616
    Destination Port: 23
    [Stream index: 7]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 345169189
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
    Flags: 0x002 (SYN)
[..]
```

Detail view of Packet 204:

```
[..]
Frame 204: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits)
Ethernet II, Src: Dell_04:b1:42 (00:16:f0:04:b1:42), Dst: Cisco_44:a1:bd (00:01:64:44:a1:bd)
Internet Protocol Version 4, Src: 10.3.1.2, Dst: 10.4.1.2
Transmission Control Protocol, Src Port: 23, Dst Port: 49623, Seq: 1, Ack: 1, Len: 76
    Source Port: 23
    Destination Port: 49623
    [Stream index: 14]
    [TCP Segment Len: 76]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 4233934838
    [Next sequence number: 77    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 614361652
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window size value: 256
    [Calculated window size: 65536]
    [Window size scaling factor: 256]
    Checksum: 0x2bd1 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    [SEQ/ACK analysis]
    [Timestamps]
    TCP payload (76 bytes)
Telnet
    Data:
r
n
    Data: No more connections are allowed to telnet server. Please try again later.
```

Detail view of Packet 33050:

```
[..]
Transmission Control Protocol, Src Port: 23, Dst Port: 53578, Seq: 240609, Ack: 264, Len: 2012
    Source Port: 23
    Destination Port: 53578
    [Stream index: 3903]
    [TCP Segment Len: 2012]
    Sequence number: 240609    (relative sequence number)
    Sequence number (raw): 483666056
    [Next sequence number: 242621    (relative sequence number)]
    Acknowledgment number: 264    (relative ack number)
    Acknowledgment number (raw): 2349536535
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window size value: 63977
    [Calculated window size: 63977]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x702e [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
    [SEQ/ACK analysis]
    [Timestamps]
    TCP payload (2012 bytes)
```

C.2.3.3 Results of Investigating the Network Traffic Between Plant Historian and EDC

FTP session:
italic is traffic from the FTP server (Plant Historian) to the EDC
normal is traffic from the EDC to the FTP Server (Plant Historian)

220 (vsFTPd 3.0.3)
AUTH TLS

```
530 Please login with USER and PASS.
AUTH SSL
530 Please login with USER and PASS.
USER lbois
331 Please specify the password.
PASS I<3engineering
230 Login successful.
CWD /home/lbois
250 Directory successfully changed.
TYPE I
200 Switching to Binary mode.
PASV
227 Entering Passive Mode (10,3,1,3,39,120).
RETR asherah-pressurizer-2019-09-03.csv
150 Opening BINARY mode data connection for asherah-pressurizer-2019-09-03.csv (45328 bytes).
226 Transfer complete.
```

Table C.4: Statistics of *Process Section* baseline capture - Section C.2.2

Address A			**Address B**		
Overall		**A -> B**		**B -> A**	
Packets	Bytes	Packets	Bytes	Packets	Bytes
192.168.0.1			**192.168.0.18**		
42018	7727208	21010	3232305	21008	4494903
192.168.0.21			**192.168.0.102**		
20847	6744042	11532	5266005	9315	1478037
192.168.0.21			**192.168.0.101**		
19992	6640959	11114	5236143	8878	1404816
192.168.0.21			**192.168.0.103**		
19181	6517135	10720	5182152	8461	1334983
192.168.0.21			**192.168.0.104**		
19113	6457270	10696	5122892	8417	1334378
192.168.0.18			**192.168.0.109**		
16391	2197006	7455	1164188	8936	1032818
192.168.0.18			**192.168.0.21**		
15026	4789340	15026	4789340	0	0
192.168.0.1			**192.168.0.4**		
3665	370221	1808	211399	1857	158822
192.168.0.1			**192.168.0.3**		
777	117733	259	27603	518	90130
192.168.0.3			**192.168.0.239**		
449	73609	286	34714	163	38895
192.168.0.3			**192.168.0.255**		
24	2208	24	2208	0	0
192.168.0.209			**239.255.255.250**		
6	1074	6	1074	0	0
192.168.0.209			**224.0.0.251**		
2	200	2	200	0	0
192.168.0.21			**192.168.0.255**		
1	243	1	243	0	0
192.168.0.104			**255.255.255.255**		
1	243	1	243	0	0
192.168.0.209			**224.0.0.252 1**		
1	75	1	75	0	0

Table C.5: Protocols present in network traffic between PLC and Maintenance Computer

Protocol	Packets	% of Packets
TCP	777	100
TPKT - ISO on TCP - RFC1006	296	38
ISO 8073/X.224 COTP	296	38
S7Comm Plus	296	222

Table C.6: Protocols present in network traffic between Maintenance Computer and Network Switch

Protocol	Packets	% of Packets
TCP	449	100
Hypertext Transfer Protocol	53	12
Line-based text data	9	2
HTML Form URL Encoded	2	0.5

Table C.7: Statistics of *Engineering Section* capture - Section C.2.3

Address A			**Address B**		
Overall		**A -> B**		**B -> A**	
Packets	Bytes	Packets	Bytes	Packets	Bytes
10.3.1.2			**10.4.1.2**		
32906	2505056	16475	1479617	16431	1025439
10.3.1.3			**10.4.1.2**		
115	53026	67	50003	48	3023
10.3.1.2			**192.168.0.101**		
8	2262	8	2262	0	0
10.3.1.2			**192.168.0.103**		
4	998	4	998	0	0
10.3.1.3			**192.168.0.104**		
1	183	1	183	0	0

Table C.8: Protocols present in network traffic between EWS and EDC

Protocol	Packets	% of Packets
TCP	32840	99.8
Telnet	8206	24.9

Table C.9: Protocols present in network traffic between Plant Historian and EDC

Protocol	Packets	% of Packets
TCP	115	92.3
FTP Data	33	28.7

D. Appendix: Additional Material

D.1 Compiled Table of *Data Types* available in the various *Data Streams* in the ICS Domain

An overview on the *Data Types* present in the ICS domain is presented in Table D.1. It combines the overview provided in Table 4.4 and Table 4.8 and uses the revised definitions of the *Data Types* as provided in Section A.3.2

Table D.1: *Data Types* available in the various *Data Streams* in the ICS Domain based on the revised definitions provided in Section A.3.2

Data Type	*Non-volatile Memory*	*Volatile Memory*	*Communication*
DT1	✓	✓	✓
DT2	✓	✓	
DT3	✓	(✓)	✓
DT4	✓	✓	
DT5	✓	✓	
DT6		(✓)	
DT7		(✓)	
DT8	✓	✓	
DT9	✓	✓	✓

D.2 Compiled Table of *Data Types* available in the various *Data Streams*in the Automotive IT Domain

An overview on the *Data Types* present in the Automotive IT domain is presented in Table D.2. It combines the overview provided in Table 4.6 and Table 4.10 and uses the revised definitions of the *Data Types* as provided in Section A.3.2

Table D.2: *Data Types* available in the various *Data Streams* in the Automotive IT domain based on the revised definitions provided in Section A.3.2

Data Type	*Non-volatile Memory*	*Volatile Memory*	*Communication*
DT1	✓	✓	✓
DT2	✓	✓	(✓)
DT3			✓
DT4	✓	✓	✓
DT5	✓	✓	✓
DT6			
DT7			(✓)
DT8			
DT9	✓	✓	✓

D.3 Example for the Relationship Between *Investigation Steps* and *Data Types* in the [KDV15]-Model

An illustrative example for the relationship between *Investigation Steps* and *Data Types* is provided by [**KHA$^+$09**]. For illustrative purposes, this example has been revised taking the altered definitions of the *Data Types* as provided in Section A.3.2 into account.

In this thesis, this example is used as a foundation for the exemplary forensic process shown in Figure D.1. In this example, an investigated computer system was infected by malicious software. This malicious software deletes its own representation from the hard drive after being loaded into the working memory and executed. Due to the specifics of the computer system, a usual recovery of the deleted malicious software from the HDD is not possible. In [**KHA$^+$09**], *ssheater*[1] was used a malicious software that implements this behavior. This malicious software attacks the SSH daemon[2] of a Linux operating system. The software is only present in the working memory after a successful installation.

The example has been updated for the use in this thesis and hence uses an adapted terminology. The forensic process is started after unusual behavior is noticed on the affected computer system. In this case *Data Gathering* is performed using both *Live Forensics* and *Post-Mortem Forensics*. The inclusion of *Live Forensics* allows for the gathering of contents of the working memory. This includes information about the active processes (**DT3** and **DT6**) and the creation of a raw image of the working memory (**DT1**). During *Post-Mortem Forensics*, a forensic image of the Non-volatile Memory (the mass storage) is created (**DT1**).

During the *Data Investigation* the collected data is investigated. Based on a database with signatures of malicious software which was created before the specific incident (during the *Strategic Preparation*), the raw image of the working memory (**DT1**) can be investigated and the presence of malicious software within the memory confirmed.

Furthermore, the raw image of the HDD is analyzed (**DT1**). The HDD uses an Ext3 file system. In contrast to other file systems, this file system zeros out pointers to the specific location of the raw data within the file systems instead of just marking these pointers as

[1] https://securiteam.com/tools/5NP0A00IBW/, 19/04/2020
[2] https://www.openssh.com/manual.html, 20/04/2020

unused[3] [4]. This prevents classical methods for the recovery of deleted files where these pointers are still present and can be used to identify the location of the contents of the deleted files. Hence, in this step the malicious software cannot be found on in this HDD image due to it being deleted.

However, the presence of the malicious software detected in the RAM allows for additional methods in order to investigate if the malicious code can be detected on the HDD. Namely, the known signature of the enables the use of *Data Carving* techniques (see [Dic06]). These methods are also referred to as *File Carving*. They work by browsing the entire raw data for known signatures. These signatures represent a start pattern which usually marks the start of a file. This could commonly be a file header (for example, the file header of a graphic file in the Portable Network Graphics (PNG) format[5] could be used to find the start of those specific graphic files). The following data is assumed to belong to the file in question. The end of the assumed file is either marked by a footer or just by the assumed maximal length. The use of *Data Carving* leads to great amounts of data. Usually, there is a large amount of false positives which do not represent an actual file. Sometimes fragmentation and missing footers lead to unusable files which only contain a fraction of the initial file. Hence, this process is usually only used with some starting assumption to find something of value from the deleted files. In this case, the assumption is, that the malicious software was present on the investigated computer system. Additionally, the signature was extracted from the working memory and can be used as a signature. This allows for the recovery of chucks from the deleted file and the confirmation that the malicious code was indeed present on the hard drive.

During the *Data Analysis* these chucks or information are correlated. Here the representation of the malicious code within the RAM Image (**DT1**), the representation of the malicious codes within the HDD Image (**DT1**), the metadata from the working memory (**DT3**), data about the processes from the working memory (**DT6**) and the metadata from the investigation of the HDD (**DT3**) can be correlated.

This correlation feeds the *Documentation*. Additional input for the *Documentation* is provided by the *Process Accompanying Documentation* which documented the steps taken during the investigation leading to the current conclusions found during the *Data Analysis*.

D.4 Tool Design for the Forensic Process

The lack of tools designed to be used in a forensic process is a challenge of the investigated domains ($IF_{ICS.Comp.8}$, $IF_{ICS.Arch.8}$, $IF_{AMI.Comp.8}$ and $IF_{AMI.Arch.7}$). This has already been noted in [**ALKD17**] and [**ALK$^+$18**]. Dedicated research on the design process of tools geared towards forensic use was performed in [**KHAD10**] and [HKD11].

These requirements are not affected by the specific properties of the investigated domains. However, in the light of the lack of dedicated forensic software in these domains, a summary of these requirements seems necessary. [**ALK$^+$18**] defines the following criteria for the design of tools usable during the forensic process:

[3] https://www.slashroot.in/how-does-file-deletion-work-linux, 20/04/2020
[4] http://batleth.sapienti-sat.org/projects/FAQs/ext3-faq.html, 20/04/2020
[5] https://www.w3.org/TR/2003/REC-PNG-20031110/, 20/04/2020

- *„the collected/processed data should be useful for the forensic process*
- *ensure the integrity, authenticity and confidentiality of the collected/processed data*
- *have a minimized and well-known structural impact, ensuring the integrity of the source data as best as possible*
- *document the actions performed*
- *the frequency of possible errors during processing should be known. “*

Table 4.9 in Section 4.2.3.4 also implies some properties beneficial for tools used during the forensic process. These can be summarized as these two additional criteria:

- *does not include any unnecessary send functionality*
- *is open source (and hence supports the comprehensibility of the actions performed))*

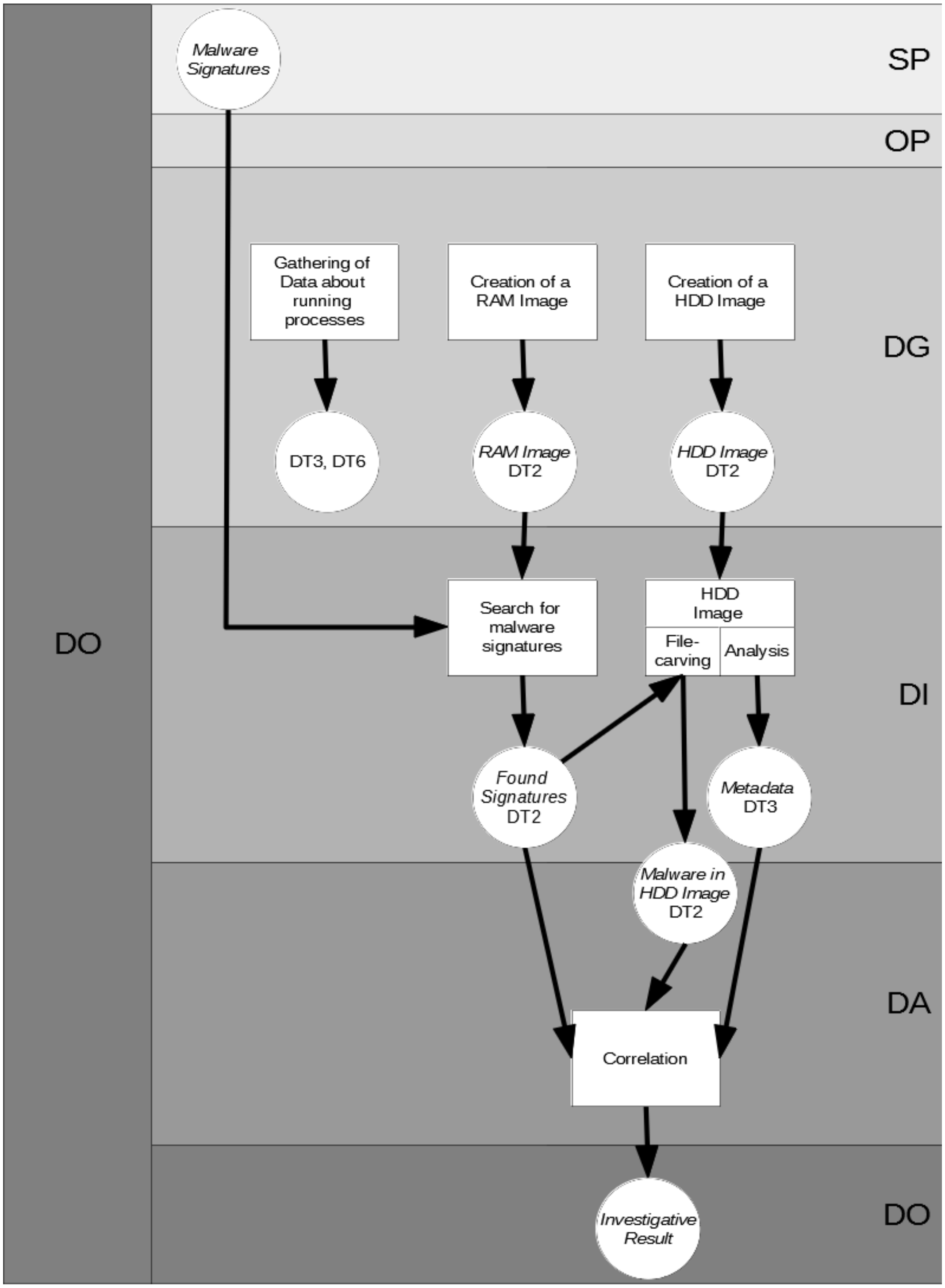


Figure D.1: Example for the relationship between *Investigation Steps* and *Data Types* based on [**KHA+09**]

Bibliography

[ACK+13] R. Altschaffel, R. Clausing, C. Krätzer, T. Hoppe, S. Kiltz, and J. Dittmann. Statistical Pattern Recognition Based Content Analysis on Encrypted Network: Traffic for the TeamViewer Application. IEEE, 2013. (cited on Page 9 and 155)

[ACKD12] R. Altschaffel, R. Clausing, S. Kiltz, and J. Dittmann. Erste Betrachtung einer Metrik für Methoden der IT-Forensik. *D-A-CH Security 2012 : Bestandsaufnahme - Konzepte - Anwendungen - Perspektiven*, pages 266–277, 2012. (cited on Page 6, 10, 71, 73, and 81)

[ADKK14] R. Altschaffel, J. Dittmann, C. Krätzer, and S. Kiltz. A Hierarchical Model for the Description of Internet-Based Communication. 2014. (cited on Page 6, 9, and 147)

[AH20] R. Altschaffel and M. Hildebrandt. A Simulated Steam Turbine Generator subsystem for Research and Training. In *Third International Conference on Nuclear Security: Sustaining and Strengthening Efforts (ICONS 2020)*, 2020. (cited on Page 6, 7, 234, and 267)

[AHeS+20] R. Altschaffel, T. Holczer, R. A. Busquim e Silva, P. Gyorgy, M. Hildebrandt, and M. Hewes. Nuclear Power Plant in a Box. In *Third International Conference on Nuclear Security: Sustaining and Strengthening Efforts (ICONS 2020)*, 2020. (cited on Page 7, 234, 235, 236, and 267)

[AHKD19] R. Altschaffel, M. Hildebrandt, S. Kiltz, and J. Dittmann. Digital Forensics in Industrial Control Systems. In *Proceedings of 38th International Conference of Computer Safety, Reliability, and Security (Safecomp 2019)*, pages 128–136. Springer Nature Switzerland, 2019. (cited on Page 6, 8, 34, 69, 126, 211, 216, 217, 218, 219, and 228)

[AHNH20] R. Altschaffel, T. Holczer, C. Neal, and M. Hildebrandt. The Nuclear SIEM. In *Third International Conference on Nuclear Security: Sustaining and Strengthening Efforts (ICONS 2020)*, 2020. (cited on Page 6, 7, 163, 166, 222, and 266)

[AKD09] R. Altschaffel, S. Kiltz, and J. Dittmann. From the Computer Incident Taxonomy to a Computer Forensic Examination. In *Fifth International Conference on IT Security Incident Management and IT Forensics*. IEEE, 2009. (cited on Page 6, 10, 70, 72, and 76)

[ALK+18] R. Altschaffel, K. Lamshöft, S. Kiltz, M. Hildebrandt, and J. Dittmann. A Survey on Open Forensics in Embedded Systems of Systems. *International Journal on*

Advances in Security, 11:104–117, 2018. (cited on Page xviii, 6, 8, 69, 83, 122, 123, 164, 174, 185, 186, 187, 188, 189, and 301)

[ALKD17] R. Altschaffel, K. Lamshöft, S. Kiltz, and J. Dittmann. A Survey on Open Automotive Forensics. *Unknown Journal*, pages 65–70, 2017. (cited on Page 6, 8, 37, 69, 127, 132, 134, 142, 143, 173, 174, 176, 200, and 301)

[ARG17] S. Anandan, R.Rajesh, and K. Ganesh. A Study of the Condenser in Nuclear Power Plants. *International Journal of ChemTech Research*, 2017. (cited on Page 239)

[Arn17] A. Arnes. *Digital Forensics*. 2017. (cited on Page viii, xv, 14, 15, 16, 20, 40, 50, 51, 52, 53, 54, 55, 56, 76, 109, 154, and 155)

[ASA17] ASAM MCD-1 XCP v 1.5.0. Technical report, Association for Standardization of Automation and Measuring Systems, 2017. (cited on Page 138)

[BBCS16] W. Bortles, W. Biever, N. Carter, and C. Smith. A Compendium of Passenger Vehicle Event Data Recorder Literature and Analysis of Validation Studies. Technical report, Kineticorp LLC, 2016. (cited on Page 63)

[BEM13] P.N. Borazjani, C.E. Everett, and D. McCoy. OCTANE: An Extensible Open Source Car Security Testbed. Technical report, 2013. (cited on Page 174 and 176)

[Ber11] D. Beresford. Siemens Simatic S7 PLC Exploitation. *Unknown Journal*, 2011. (cited on Page 116, 120, 122, 123, 124, 125, 194, and 220)

[Bis18] M. Bishop. *Computer Security - Art and Science*. Addison-Wesley, second edition, 2018. (cited on Page 17, 20, 22, 24, and 155)

[BJ04] D. E. Beckstein and J. D. Jackson. The Daubert Triology in the States. *Jurimetrics*, (44), 2004. (cited on Page 27)

[BKA18] Polizeiliche Kriminalstatistik (PKS). Technical report, BKA, 2018. (cited on Page 18, 29, 31, 57, and 228)

[BMZ14] S. Bhatt, P. Manadhata, and L. Zomlot. The Operational Role of Security Information and Event Management Systems. *Security & Privacy, IEEE*, 12, 2014. (cited on Page 222 and 266)

[Bos17] CDR Vehicle List. Technical report, Bosch, 2017. (cited on Page 63)

[Bra01] M. Braid. Collecting Electronic Evidence After a System Compromise. *Unknown Journal*, 2001. (cited on Page viii, xv, 40, 42, 44, 45, 51, 52, 53, 54, 56, and 57)

[BSI11] BSI Leitfaden IT-Forensik. Technical report, Bundesamt für Sicherheit in der Informationstechnik, 2011. (cited on Page xv, xvi, xvii, 2, 70, 71, 74, 76, 77, 81, 82, 86, 87, 88, 89, 90, 91, 93, 109, 145, 146, 152, and 214)

[BSI19] BSI-CS 134E : RECOMMENDATION: IT IN PRODUCTION - Monitoring and Anomaly Detection in Production Networks. Technical report, Bundesamt für Sicherheit in der Informationstechnik, 2019. (cited on Page 222)

[BT20] Bluetooth Core Specification Version 5.2 Feature Overview. Technical report, Bluetooth SIG, 2020. (cited on Page 165)

[Cal08] M. C. Calhoun. Scientific Evidence in Court: Daubert or Frye, 15 Years Later. *Legal Background*, 23(37), 2008. (cited on Page 27)

[CAN91] CAN Specification Version 2.0. Technical report, BOSCH, 1991. (cited on Page xvi and 170)

[Car06] B. D. Carrier. Risks of live digital forensic analysis. *Communications of the ACM*, pages 56–61, 2006. (cited on Page 110)

[CAR15] On-Board Diagnostic II (OBD II) Systems - Fact Sheet / FAQs. Technical report, California Enviormental Protection Agency, 2015. (cited on Page 135)

[Cas02] E. Casey. Error, Uncertainty, and Loss in Digital Evidence. *International Journal of Digital Evidence*, 2002. (cited on Page 22 and 23)

[Cas04] E. Casey. *Digital Evidence and Computer Crime*. Second edition, 2004. (cited on Page viii, xv, 40, 46, 47, 48, 49, 52, 53, 54, 55, 57, 76, 81, and 109)

[Cas11] E. Casey. *Digital Evidence and Computer Crime*. Third edition, 2011. (cited on Page 47 and 52)

[CCK15] The Cyber Kill Chain®. Technical report, LOCKHEED MARTIN, 2015. (cited on Page 186 and 190)

[CG04] B. D. Carrier and J. Grant. A hardware-based memory acquisition procedure for digital investigations. *The International Journal of Digital Forensics & Incident Response*, 2004. (cited on Page 111)

[CMGS12] P. Cichonski, T. Millar, T. Grance, and K. Scarfone. Computer Security Incident Handling Guide, 2012. (cited on Page 45)

[Cor16] S. Corrigan. Introduction to the Controller Area Network (CAN). Technical report, Texas Instruments, 2016. (cited on Page 173)

[Cur20] R. Currie. Hacking the CAN Bus: Basic Manipulation of a Modern Automobile Through CAN Bus Reverse Engineering. Technical report, 2020. (cited on Page 169)

[Dau93] Daubert v. Merrell Dow Pharmaceuticals, Inc., 509 U.S. 579 (1993). Technical report, United States Supreme Court, 1993. (cited on Page 26 and 27)

[Daw18] T. Dawson. Who Were the Leading Vendors of Industrial Controls in 2017? 2018. (cited on Page 114)

[DFR01] A Road Map for Digital Forensic Research. Technical report, DFRWS, 2001. (cited on Page 16 and 19)

[Dic06] D. Dickerman. Advanced Data Carving. Submission for DFRWS 2006 Data Carving Challenge, 2006. (cited on Page 301)

[DvHvH17] P. A.L. Ducheine, J. van Haaster, and R. van Harskamp. Manoeuvering and Generating Effects in the information Environment. *Netherlands Annual Review of Military Studies 2017: Winning Without Killing*, 2017. (cited on Page 186 and 190)

[DXC14] DXCPL DAP over CAN Physical Layer. Technical report, Infineon, 2014. (cited on Page 140)

[Ehr15] S. Ehrampoosh. Memory Allocation in Siemens PLC-S7 Programming, 2015. (cited on Page 118)

[ELS17] S. Eresheim, R. Luh, and S. Schrittwieser. The Evolution of Process Hiding Techniques in Malware - Current Threats and Possible Countermeasures. *Journal of Information Processing*, 25:866–874, 2017. (cited on Page 112)

[ENI17] Communication network dependencies for ICS/SCADA Systems. Technical report, ENISA, 2017. (cited on Page 61, 159, 160, and 161)

[ENI19] Introduction to Network Forensics. Technical report, ENISA, 2019. (cited on Page 63, 145, 148, 150, 154, and 155)

[eSMP$^+$20] R. A. Busquim e Silva, R. P. Marques, J. R. C. Piqueira, P. Smith, M. Hewes, S. Purvis, J. Li, and R. Altschaffel. Understanding Nuclear Cyber Security Measures, Risks and Consequences: from Tank Levels to Plant Processes. In *Third International Conference on Nuclear Security: Sustaining and Strengthening Efforts (ICONS 2020)*, 2020. (cited on Page 6, 7, 234, 235, and 267)

[EU008] COUNCIL DIRECTIVE 2008/114/EC of 8 December 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection. Technical report, THE COUNCIL OF THE EUROPEAN UNION, 2008. (cited on Page 183)

[FC08] M. Fabro and E. Cornelius. Recommended Practice: Creating Cyber Forensics Plans for Control Systems. Technical report, DHS National Cyber Security Division, 2008. (cited on Page 60 and 162)

[Fed15] Federal Rules for Evidence. Technical report, United States of America, 2015. (cited on Page 26, 27, and 30)

[FMC] N. Falliere, L. O Murchu, and E. Chien. [w32.stuxnet dossier. Technical report, Symantic. (cited on Page 187)

[Fre06] Introduction to HCS08 Background Debug Mode. Technical report, Freescale Semiconductor, 2006. (cited on Page 140)

[FS07] F. C. Freiling and B. Schwittay. A Common Process Model for Incident Response and Computer Forensics. *Unknown Journal*, 2007. (cited on Page viii, xv, 40, 47, 48, 49, 52, 53, 54, 55, 56, 76, and 80)

[FSE14] BLACKENERGY & QUEDAGH - The convergence of crimeware and APT attacks. Technical report, F-SECURE LABS, 2014. (cited on Page 186)

[Gar19] Gartner Says Worldwide PC Shipments Declined 4.3 Percent in 4Q18 and 1.3 Percent for the Year. Technical report, Gartner, 2019. (cited on Page 105)

[GDP16] General Data Protection Regulation. Technical report, European Union, 2016. (cited on Page 24 and 25)

[GGAW20] D. Gupta, D. Govindaraj, R. Altschaffel, and K. Waedt. Blue team support for EPS related cybersecurity readiness. In *Third International Conference on Nuclear Security: Sustaining and Strengthening Efforts (ICONS 2020)*, 2020. (cited on Page 6, 7, and 267)

[GLB13] M. Graziano, A. Lanzi, and D. Balzarotti. Hypervisor Memory Forensics. *International Workshop on Recent Advances in Intrusion Detection*, 2013. (cited on Page 112)

[Grz11] A. Grzemba. *MOST - The Automotive Multimedia Network*. Franzis Verlag GmBH, 2011. (cited on Page 171)

[Gyo00] S. Gyorke. Ten Common Questions about Event Data Recorders, 2000. (cited on Page 63)

[HAK+17] M. Hildebrandt, R. Altschaffel, S. Kiltz, M. Hildebrandt, and J. Dittmann. Exploring the Possibility of Forensic Investigations on Steam Turbine Governing Systems. *Unknown Journal*, 2017. (cited on Page 6, 8, 118, 119, 121, and 162)

[HAL+20] M. Hildebrandt, R. Altschaffel, K. Lamshöft, M. Lange, M. Szemkus, T. Neubert, C. Vielhauer, Y. Ding, and J. Dittmann. Threat Analysis of Steganographic and Covert Communication in Nuclear I&C Systems. In *Third International Conference on Nuclear Security: Sustaining and Strengthening Efforts (ICONS 2020)*, 2020. (cited on Page 7 and 234)

[HAR13] HART Communication Application Guide. Technical report, HART Communication Foundation, 2013. (cited on Page 165)

[Hie19] J. Hielscher. Elchtest in Industrienetzen: Der ELK-Stack als Netzwerk-Forensik-Werkzeug. Master's thesis, Otto von Guericke University Magdeburg, 2019. (cited on Page 222)

[Hil12] A. Hillier. *Hillier's Fundamentals of Automotive Electronics Book 2*. Oxford University Press, sixth edition, 2012. (cited on Page 127)

[HKD11] M. Hildebrandt, S. Kiltz, and J. Dittmann. A Common Scheme for Evaluation of Forensic Software. 2011. (cited on Page 301)

[HL98] J. D. Howard and T. A. Longstaff. A common language for computer security incidents (sand98-8667). Technical report, Sandia National Laboratories, 1998. (cited on Page 10 and 73)

[Hop14] T. Hoppe. *Prävention, Detektion und Reaktion gegen drei Ausprägungsformen automotiver Malware - Eine methodische Analyse im Spektrum von Manipulationen und Schutzkonzepten*. PhD thesis, Otto von Guericke University Magdeburg, 2014. (cited on Page 177 and 222)

[ICE19] IEC 61784-1:2019 Industrial communication networks - Profiles Part 1: Fieldbus profiles. Technical report, International Electrotechnical Comission, 2019. (cited on Page 159)

[IEE13] IEEE STANDARD 1149.1-2013 - IEEE Standard for Test Access Port and Boundary-Scan Architecture. Technical report, IEEE, 2013. (cited on Page 140)

[IR01] K. Inman and N. Rudin. *Principles and Practises of Criminalistics: The Profession of Forensic Science*. CRC Press LLC Boca Raton Florida, USA, 2001. (cited on Page 15)

[ISA10] ANSI/ISA-95.00.01-2010: Enterprise Control System Integration 1: Models and terminology. Technical report, International Society of Automation, 2010. (cited on Page xvi, 157, 158, 159, 160, 161, 168, and 195)

[ISO94] ISO 9141-2:1994(en) Road vehicles - Diagnostic systems - Part 2: CARB requirements for interchange of digital information. Technical report, International Organization for Standardization, 1994. (cited on Page 135)

[ISO99] ISO 14230-3: Road vehicles – Diagnostic systems – Keyword Protocol 2000. Technical report, International Organization for Standardization, 1999. (cited on Page 134)

[ISO11] ISO 15765-1:2011: Road vehicles – Diagnostic communication over Controller Area Network (DoCAN). Technical report, International Organization for Standardization, 2011. (cited on Page 134)

[ISO13a] ISO 14229-1:2013: Road vehicles – Unified diagnostic services (UDS). Technical report, International Organization for Standardization, 2013. (cited on Page 134)

[ISO13b] ISO 17458-5:2013 Road vehicles - FlexRay communications system - Part 5: Electrical physical layer conformance test specification. Technical report, International Organization for Standardization, 2013. (cited on Page 172)

[ISO15a] ISO 11898-1:2015 Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling. Technical report, International Organization for Standardization, 2015. (cited on Page 129, 134, and 170)

[ISO15b] ISO 15031: Road vehicles – Communication between vehicle and external equipment for emissions-related diagnostics. Technical report, International Organization for Standardization, 2015. (cited on Page 134)

[ISO19] ISO 17458-1:2013 Road vehicles - FlexRay communications system - Part 1: General information and use case definition. Technical report, International Organization for Standardization, 2019. (cited on Page 172)

[Jor17] J. Jordaan. The GDPR and DFIR: The Impact of the EU General Data Protection Regulation on Digital Forensics and Incident Response, 2017. (cited on Page 24)

[KAD16] C. Krätzer, R. Altschaffel, and J. Dittmann. Tendenzen zum Profiling von verschlüsselten Netzwerkverkehren - Möglichkeiten und Grenzen. In *6th International Symposium "New Technologies"*, 2016. (cited on Page 6, 9, and 155)

[KAH$^+$15] S. Kuhlmann, R. Altschaffel, T. Hoppe, J. Dittmann, and C. Neubüser. Evaluation of impacts of IT-incidents on automotive safety with regard to supporting reaction strategies for the driver. In *Traffic safety through integrated technologies: 24th Enhanced Safety of Vehicle Conference*, 2015. (cited on Page 9)

[KAH$^+$18] A. Koch, R. Altschaffel, M. Hildebrandt, S. Kiltz, and J. Dittmann. Exploring the Processing of Personal Data in Modern Vehicles - A Proposal of a Testbed for Explorative Research to Achieve Transparency for Privacy and Security. pages 15–26, 05 2018. (cited on Page 8, 67, 132, 133, 137, 143, 169, 173, 174, 175, 180, 200, and 201)

[KDV15] S. Kiltz, J. Dittmann, and C. Vielhauer. Supporting Forensic Design - A Course Profile to Teach Forensics. In *IMF '15: Proceedings of the 2015 Ninth International Conference on IT Security Incident Management & IT Forensics (imf 2015)*. IEEE, 2015. (cited on Page iii, iv, v, vi, viii, x, xi, xii, xiii, xv, xvii, xviii, 2, 3, 4, 5, 6, 8, 10, 11, 13, 63, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 85, 86, 87, 89, 90, 91, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 107, 118, 124, 141, 147, 185, 205, 206, 207, 208, 209, 210, 211, 213, 214, 215, 217, 219, 220, 221, 223, 224, 225, 227, 229, 231, 232, 233, 234, 247, 248, 251, 260, 261, 262, 266, 274, 275, 277, and 300)

[Kei18] O. Keil. Forensik in Automatisierungssystemen - Konzept zur Identifikation und Erhebung verschiedener Datenquellen. Master's thesis, Otto von Guericke University Magdeburg, 2018. (cited on Page xvii, 116, 117, 121, and 123)

[KGC$^+$06] T. Kilpatrick, J. Gonzalez, R. Chandia, M. Papa, and S. Shenoi. An Architecture for SCADA Network Forensics. *IFIP Advances in Information and Communication*, 22, 2006. (cited on Page 61, 163, and 166)

[KHA$^+$09] S. Kiltz, M. Hildebrandt, R. Altschaffel, J. Dittmann, C. Vielhauer, and C. Schulz. Sicherstellung von gelöschtem Schadcode anhand von RAM-Analysen und Filecarving mit Hilfe eines forensischen Datenmodells. In *Sichere Wege in der vernetzten Welt: Tagungsband zum 11. Deutschen IT-Sicherheitskongress*, 2009. (cited on Page xvi, 6, 11, 70, 71, 72, 112, 300, and 303)

[KHAD10] S. Kilz, M. Hildebrandt, R. Altschaffel, and J. Dittmann. A transparent Bridge for forensic sound network traffic data acquisition. In *Sicherheit 2010. Sicherheit, Schutz und Zuverlässigkeit*, 2010. (cited on Page 10, 70, 73, 78, 82, 151, 237, and 301)

[KHD09] S. Kiltz, T. Hoppe, and J. Dittmann. A New Forensic Model And Its Application To The Collection, Extraction And Long Term Storage Of Screen Content Off A Memory Dump. In *16th International Conference on Digital Signal Processing (DSP2009)*, pages 1–6, 2009. (cited on Page xv, 70, 72, and 73)

[KHDV09] S. Kiltz, T. Hoppe, J. Dittmann, and C. Vielhauer. Video surveillance: A new forensic model for the forensically sound retrival of picture content off a memory dump. In *Proceedings of Informatik2009 - Digitale Multimedia-Forensik, GI Informatik*, pages 1619–1633, 2009. (cited on Page 70 and 73)

[Kil20] S. Kiltz. *Data-Centric Examination Approach (DCEA) for a qualitative determination of error, loss and uncertainty in digital and digitised forensics.* PhD thesis, Otto von Guericke University Magdeburg, 2020. (cited on Page 3, 69, 216, 219, and 230)

[KL15] E. D. Knapp and J. T. Langill. *Industrial Network Security.* Syngress, second edition, 2015. (cited on Page xvi, 32, 156, 157, 158, 159, 183, 221, and 229)

[Kno09] M. Knopp. Rechtliche Perspektiven zur digitalen Beweisführung. In *GI Jahrestagung 2009*, pages 1552–1566, 2009. (cited on Page 27, 28, 29, and 42)

[KRI16] Verordnung zur Bestimmung Kritischer Infrastrukturen nach dem BSI-Gesetz (BSI-Kritisverordnung - BSI-KritisV). Technical report, Bundesministerium des Innern, 2016. (cited on Page 183)

[LAC16] R. M. Lee, M. J. Assante, and T. Conway. Analysis of the Cyber Attack on the Ukrainian Power Grid. Technical report, SANS Institute, 2016. (cited on Page 186 and 189)

[LAD17] K. Lamshöft, R. Altschaffel, and J. Dittmann. Adapting Organic Computing Architectures to an Automotive Environment to Increase Safety & Security. *Automotive - Safety & Security 2017 - Sicherheit und Zuverlässigkeit für automobile Informationstechnik*, pages 103–119, 2017. (cited on Page 9 and 174)

[Lam19] K. Lamshöft. Von 100 kHz bis 6 GHz - der Einfluss von Software-Defined Radio auf die Verfolgbarkeit von vernetzten Fahrzeugen: Nutzung von identifizierenden Informationen in Funkprotokollen zur Verfolgung von Fahrzeugen. Master's thesis, Otto von Guericke University Magdeburg, 2019. (cited on Page 165)

[LCLW14] M. Ligh, A. Case, J. Levy, and A. Walters. *The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory.* Wiley, 2014. (cited on Page 110 and 111)

[LEA16] *Vehicular Digital Forensics: What Does My Vehicle Know About Me?* Association for Computing Machinery, 2016. (cited on Page 66, 67, and 264)

[LIN10] LIN Specification Package. Technical report, LIN Consortium, 2010. (cited on Page 171)

[LL11] K. C. Laudon and J.P. Laudon. *Management Information Systems.* Prentice Hall, twelfth edition, 2011. (cited on Page xv, 31, 144, and 145)

[LLW$^+$15] X. Li, D. Li, J. Wan, A. Vasilakos, C. Lai, and S. Wang. A review of industrial wireless networks in the context of Industry 4.0. In *Wireless Networks*, 2015. (cited on Page 164 and 165)

[LPM04] J. T. Luttgents, M. Pepe, and K. Mandia. *Incident Response & Computer Forensics.* McGraw-Hill, third edition, 2004. (cited on Page 154 and 155)

[Mac13] H. Macht. *Live Memory Forensics on Android with Volatility.* PhD thesis, Friedrich-Alexander University Erlangen-Nuremberg, 2013. (cited on Page 112)

[Mat19] S.C. Matthiesen, Wickert & Lehrer. Admissibility of Expert Testimony in all 50 States. *Unknown Journal*, 2019. (cited on Page 27)

[MMW$^+$14] T. Machado, I. Muller, J. Winter, V. Dickow, C. E. Pereira, and J. C. Netto. WirelessHART network analyzer with coexistence detection. 2014. (cited on Page 165)

[MPP03] K. Mandia, C. Proise, and M. Pepe. *Incident Response and Computer Forensics*. McGraw-Hill/Osborne, second edition, 2003. (cited on Page viii, xv, 40, 44, 45, 48, 52, 54, 55, 57, 76, and 80)

[MV15] C. Miller and C. Valasek. Remote Exploitation of an Unaltered Passenger Vehicle, 2015. (cited on Page 184, 190, and 191)

[Nel16] M. Nelson. The Impact of Dragonfly Malware on Industrial Control Systems. Technical report, SANS Institute, 2016. (cited on Page 186)

[NHT02] Event Data Recorders. Technical report, NHTSA EDR Working Group, 2002. (cited on Page 63)

[NHT12] Federal Motor Vehicle Safety Standards; Event Data Recorders. Technical report, NHTSA, 2012. (cited on Page xvii, 64, and 65)

[NIS15] NIST. NIST Special Publication 800-82 Revision 2: Guide to Industrial Control Systems (ICS) Security. 2015. (cited on Page 32, 33, 35, 159, 181, 182, and 221)

[NN11] H. Nagel and J. Norton. Evidence. *Encyclopaedia Britannica*, 2011. (cited on Page 14)

[NSS11] Nuclear Security Series No. 17 Computer Security at Nuclear Facilities. Technical report, International Atomic Energy Agency (IAEA), 2011. (cited on Page 7, 234, 236, and 254)

[OCT14] *OCTANE Quick-Start Guide*, 2014. (cited on Page 174 and 176)

[PNN04] Cyber Security Self-Assessment Method for U.S. Nuclear Power Plants. Technical report, Pacific Northwest National Laboratory, 2004. (cited on Page 238 and 285)

[Pol95] M. Pollitt. Computer Forensics: an approach to evidence in cyberspace. 1995. (cited on Page viii, xv, 40, 42, 52, 54, 55, 57, 76, and 79)

[Pol07] M. Pollitt. An Ad Hoc Review of Digital Forensic Models. *Proceedings of the Second International Workshop*, 2007. (cited on Page 40)

[Proa] Profibus. Technical report, PROFIBUS and PROFINET International. Available online at https://www.profibus.com/technology/profibus/. (cited on Page 159 and 164)

[Prob] Profinet. Technical report, PROFIBUS and PROFINET International. Available online at https://www.profibus.com/technology/profinet/. (cited on Page 159 and 167)

[Pur19] S. Purvis. IAEA Conducts Training Course on Protecting Nuclear Facilities from Cyber-Attacks, 2019. (cited on Page 233)

[RA02] W. Rosenbluth and H. A. Adams. Retrieval and Interpretation of Crash-Related Data from Nonresponsive Electronic Control Units in Land Vehicle Systems. *Journal of Testing and Evaluation*, 30:350–361, 2002. (cited on Page 65, 132, 134, and 141)

[RC14] G. G. Richard and A. Case. In lieu of swap: Analyzing compressed RAM in Mac OS X and Linux. *Digital Investigation*, 11, 2014. (cited on Page 112)

[RFI08] EPC^{TM}Radio-Frequency Identify Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz Version 1.2.0. Technical report, EPCglobal Inc., 2008. (cited on Page 165)

[Rob13] C. Robertston. Indicators of Compromise in Memory Forensics. 2013. (cited on Page 112)

[RS12] R.Breuk and A. Spruyt. Integrating DMA attacks in exploitation frameworks. Technical report, 2012. (cited on Page 111)

[RW211] Converged Plantwide Ethernet (CPwE) Design and Implementation Guide. Technical report, Rockwell Automation, 2011. (cited on Page xv, 33, and 34)

[SAE06] SAE J1850_200606 Class B Data Communications Network Interface. Technical report, SAE International, 2006. (cited on Page 135)

[SBS16] R. Spenneberg, M. Brüggemann, and H. Schwartke. PLC-Blaster: A Worm Living Solely in the PLC. 2016. (cited on Page 187)

[SCDLK19] D. Steiner, L. Chen, H. Darren, and N. Le-Khac. Vehicle Communication within Networks -Investigation and Analysis approach: A Case Study. *Unknown Journal*, 2019. (cited on Page 67 and 264)

[Sch04] B. Schneier. *Secrets & Lies*. Wiley Publishing, 2004. (cited on Page 18)

[SDM08] N. Singleton, J. Daily, and G. Manes. Automobile Event Data Recorder Forensics. *Advances in Digital Forensics IV*, 2008. (cited on Page 65)

[SIE17] SIMATIC IPC - Protection against power failures. Technical report, Siemens, 2017. (cited on Page 118)

[SIE19] S7-1500, ET 200SP, ET 200Pro, Simantic Drive Controller - Structure and Use of CPU Memory. Technical report, Siemens, 2019. (cited on Page xv, 117, 119, 120, and 121)

[SJaTW14] J. Stirland, K. Jones, and H. Janicke an T. Wu. Developing Cyber Forensics for SCADA Industrial Control Systems. *Unknown Journal*, 2014. (cited on Page 62 and 185)

[Smi16] C. Smith. *The Car Hackers' Handbook*. No Starch Press, 2016. (cited on Page 63, 132, 137, 143, 171, 173, 175, 180, 200, and 201)

[SPH20] A. Seferagic, E. De Poorter, and J. Hoebeke. Survey on Wireless Technology Trade-Offs for the Industrial Internet of Things. *Sensors*, 2020. (cited on Page 164)

[SST14] T. Sugimura, K. Sugimoto, and M. Tsuyuki. *Junction Blocks Simplify and Decrease Networks When Matched to ECU and Wire Harness*. Wiley, 2014. (cited on Page 127)

[Sta20] Desktop Operating System Market Share Worldwide Apr 2019 - Apr 2020. Technical report, Statcounter Global Stats, 2020. (cited on Page xv, 107, and 108)

[Ste09] P. Stelfox. *Criminal Investigation: An Introduction to Principles and Practice*. 2009. (cited on Page 14)

[STM13] *Incident Analysis & Digital Forensics in SCADA and Industrial Control Systems*, 2013. (cited on Page 60 and 61)

[Sym14] Dragonfly: Cyberespionage Attacks Against Energy Suppliers. Technical report, Symatic, 2014. (cited on Page 186)

[Tan07] A. Tanenbaum. *Modern Operating Systems*. Prentice Hall, third edition, 2007. (cited on Page 107 and 220)

[TPS09] D. K. Tran, P. Pisa, and P. Smolik. An Open Implementation of Profibus DP. 2009. (cited on Page 164)

[VKL16] P. Van Vliet, M. T. Kechadi, and N. Le-Khac. Forensics in Industrial Control System: A Case Study. 2016. (cited on Page 63, 85, and 182)

[Wal06] H. Wallentowitz. *Handbuch Fahrzeugelektronik*. Viehweg-Teubner, 2006. (cited on Page xvi, 170, 171, and 172)

[WDJC13] T. Wu, J. F. P. Disso, K. Jones, and A. Campos. Towards a SCADA Forensics Architecture. In *ICS-CSR 2013: Proceedings of the 1st International Symposium on ICS & SCADA Cyber Security Research 2013*, 2013. (cited on Page 61, 62, 163, and 196)

[WEG19] R. L. Wichmann, B. Eisenbart, and K. Gericke. The Direction of Industry: A Literature Review on Industry 4.0. *Proceedings of the Design Society: International Conference on Engineering Design*, 1, 2019. (cited on Page 164 and 264)

[Wil92] T. J. Williams. The Purdue enterprise reference architecture: a technical guide for CIM planning and implementation. *Unknown Journal*, 1992. (cited on Page xvi, 33, 126, 156, 158, 168, and 195)

[ZB15] ZigBee Specification. Technical report, ZigBee Alliance, 2015. (cited on Page 165)

[ZWZ$^+$10] L. Zhang, L. Wang, R. Zhang, S. Zhang, and Y. Zhou. Live Memory Acquisition through FireWire. *Forensics in Telecommunications, Information, and Multimedia. e-Forensics 2010. Lecture Notes of the Institute for Computer Sciences,* *Social Informatics and Telecommunications Engineering, vol 56.*, pages 159–167, 2010. (cited on Page 111)