A Semi-Automated HTTP Traffic Analysis for Online Payments for Empowering Security, Forensics and Privacy Analysis

Salatiel Ezennaya-Gomez
salatiel.ezennaya@ovgu.de
Otto-von-Guericke University Magdeburg
Magdeburg, Sachsen-Anhalt, Germany

Stefan Kiltz
kiltz@iti.cs.uni-magdeburg.de
Otto-von-Guericke University Magdeburg
Magdeburg, Sachsen-Anhalt, Germany

Christian Kraetzer
kraetzer@iti.cs.uni-magdeburg.de
Otto-von-Guericke University Magdeburg
Magdeburg, Sachsen-Anhalt, Germany

Jana Dittmann
jana.dittmann@ovgu.de
Otto-von-Guericke University Magdeburg
Magdeburg, Sachsen-Anhalt, Germany

ABSTRACT

The paper discusses means to identify potential impacts of data flows on customers' security, and privacy during online payments. The main objectives of our research are looking into the evolution of cybercrime new trends of online payments and detection, more precisely the usage of mobile phones, and describing methodologies for digital trace identification in data flows for potential online payment fraud. The paper aims to identify potential actions for identity theft while conducting the Reconnaissance step of the kill chain, and documenting a forensic methodology for guidance and further data collection for law enforcement bodies. Moreover, a secondary objective of the paper is to identify, from a user's perspective, transparency issues of data sharing among involved parties for online payments. We thus declare the transparency analysis as the incident triggering a forensic examination. Hence, we devise a semi-automated traffic analysis approach, based on previous work, to examine data flows, and data exchanged among parties in online payments. For this, the main steps are segmenting traffic generated by the process payment, and other sources, subsequently, identifying data streams in the process. We conduct three tests which include three different payment gateways: PayPal, Klarna-sofort, and Amazon Pay. The experiment setup requires circumventing TLS encryption for the correct identification of forensic data types in TCP/IP traffic, and potential data leaks. However, it requires no extensive expertise in mobile security for its installation. In the results, we identified some important security vulnerabilities from some payment APIs that pose financial and privacy risks to the marketplace's customers.

CCS CONCEPTS

• **Applied computing** → **Evidence collection, storage and analysis**; *Network forensics*; • **Security and privacy** → Economics of security and privacy.

ARES 2021, August 17–20, 2021, Vienna, Austria

ACM ISBN 978-1-4503-9051-4/21/08.
https://doi.org/10.1145/3465481.3470114

KEYWORDS

Fintechs, online payments, online privacy, digital forensics, data streams

ACM Reference Format:
Salatiel Ezennaya-Gomez, Stefan Kiltz, Christian Kraetzer, and Jana Dittmann. 2021. A Semi-Automated HTTP Traffic Analysis for Online Payments for Empowering Security, Forensics and Privacy Analysis. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3465481.3470114

1 INTRODUCTION

The payment landscape is complex, and constantly changing, since more financial technology (Fintech) companies are taking part of the online payment ecosystem offering versatile and interoperable solutions, while traditional banks operate mainly in mobile banking. The diversity of Fintech companies makes it difficult to identify what actors are involved in a payment process, what data are exchanged, and in what security and privacy conditions. Moreover, online payment fraud is becoming more sophisticated over time with JSON attacks [3]; bypassing content security policy for credit card data theft [1]; installing malware, by SMS phishing, which imitates credit card verification of banking apps and payment wallets, such as Google Pay [5]; using application reverse engineering [11], including social engineering. In many occasions, payment frauds has been perpetrated due to naïve security errors of software applications, as it is shown in [11]. When conducting an investigation over an alleged online payment fraud or privacy violation, computer forensics helps to structure and present the results of such investigation.

The paper aims to identify potential actions for identity theft while conducting the Reconnaissance step of the kill chain [6], and document the results applying a computer forensic approach for guidance and further data collection for law enforcement bodies. We aim to answer the following research questions: (*Q1*) how many entities are involved in an online payment process for a simple donation or item purchase? (*Q2*) Are other entities, non-Fintech specific, also involved in a payment process? (*Q3*) Do these entities also gather information from payment processes? (*Q4*) What are the security and privacy measures employed in these data sharing process?

To answer the research questions, we draw the hypothesis of the existence of transparency issues in the data sharing practices among entities in online payments. For the study, we employ a forensic methodology, based on previous work on a data-centric examination approach introduced by Kiltz in [7], providing a classification of data streams and types gathered in an online payment traffic. Our topic of interest is smartphone-based online payments. Thus, we conduct a series of online payment tests, gathering network traffic from a customer perspective and his mobile device. For this, we exploit a set of methods that supports the forensic analysis, which are: explicit means of intrusion and detections (EMID) (i.e., a man-in-the-middle proxy), an operating system (OS) (i.e., an Android rooted phone), and data processing and evaluation forensic tools (DPE), i.e., functionalities offered by the Wireshark tool suite, and MITMproxy visualization tools.

Along with these methods, we apply a semi-automated traffic segmentation that highlights payment related and non-related traffic, also based on previous work by Libert in [13]. We aim to assist law enforcement bodies in the identification of data traces in police investigations involving online payment frauds, while identifying potential transparency issues helping data protection officers to investigate potential privacy issues.

The paper is structured as follows: section 2 gives an overview on the two methods used for the traffic analysis. Sections 3 and 4 describe the methodology, experimental setup, and analysis tools used for the intended purpose. Sections 5 and 6 present the results of the analysis, and the discussion, respectively. Finally, conclusions are exposed in section 7.

2 BACKGROUND

The approach we use in the paper combines previous work on a data-centric examination approach introduced by Kiltz in [7] for digital forensics, and the *Webxray* tool for auditing disclosure of third-party data collection by Libert in [13].

The former approach is a forensic methodology that provides semantics to IT systems, layers of data and their processing for every step of a forensic investigation. Firstly, it formally defines a set of investigation steps as described in [8] to group forensic examination activities with similar characteristics into mutually exclusive steps (see section 3).

When conducting the aforementioned steps, there is a set of data streams including data types that are identified in the process. The approach defines forensic data types for data of the investigation described in table 1 denoted as DT_i.

Those data types serve as source and result of the data processing by forensic methods applied by the examiners. The data types can be found in data streams denoted along the paper as DS_i. The author defines up to three types of data streams: *mass storage* DS_T, *main memory* DS_M, and *network* DS_N. For our analysis, we are interested in the network data stream, DS_N. For the more detail description of these methods and semantics, see [7].

One part of our analysis approach relies on a software platform called *Webxray*, implementing a forensic tool suite and thus providing forensic methods. It is developed by Libert introduced in [12], and expanded with a module called *privacyxray* in [13]. The software monitors third-party network traffic generated when a given website is loaded. The software attributes the traffic to entities (i.e., pre-registered most common companies and subsidiaries) which receive data from the website. It is complemented with *privacyxray*, which audits data disclosure of third-parties data flows from the privacy policies of the given website with multi-lingual support.

3 METHODOLOGY

This section describes how the payment data were collected and analyzed, and what methods were used for this study. We performed a set of steps according to the description of the first of the kill chain phases, i.e., Reconnaissance, which consists of the identification and selection of targets through literature research, among other sources [6]. The forensic methodology incorporated into our approach consists of the following six steps:

(1) **Strategic preparation (SP)** *is a set of measures taken in order to support a forensic investigation prior to an incident* [7]: For our particular objective, our incident hypothesis is the existence of transparency issues while conducting an online payment regarding data sharing practices. For this step, we read the conditions and data sharing practices written in policies statements of target marketplaces, and their payment options. The results and findings described in this article also act as strategic preparation for the next examination;

(2) **Operational preparation (OP)** *is a set of measures of preparation for a forensic investigation after a suspected incident* [7]: We executed OP by forensic analysis tools to gather, investigate and analyze case-specific data, and the devices for the experiment for our incident of *transparency analysis*, thus *empowering security, forensics and privacy analysis*. Moreover, we search for API documentation used in the selected marketplace for a better understanding of the information, and HTTP messages exchanged for further analysis (e.g., identification of authorization/authentication tokens, transaction ID, among other attributes). In parallel, we collect links of each website where the data sharing conditions and third-party companies are described for further steps;

(3) **Data gathering (DG)** *is the step were data are acquired and digital evidence secured* [7]: Once the marketplace is selected, we conduct the payment capturing the traffic with the experimental setup running successfully (detailed in 4). In this step, we gather a network data stream denoted as DS_{N_1} originated from the captured traffic. In parallel, all sites visited collected in OP involving the payment are scrapped following the points further described in *segmentation* step. This action gathers a second network data stream denoted as DS_{N_2}. Concretely, we target session data DT_7 (e.g., HTML source, JavaScript). From DT_7, we can extract names of third-party companies with whom the Fintech companies share data;

(4) **Data investigation (DI)** *is a set of measures for the extraction of data* [7]: in this step, a semi-automated traffic segmentation is performed for DS_{N_1} marking traffic originated by the actions of the payment process, and substantiating the results from traffic generated by cookies-trackers with the original *Webxray*'s output.On the other hand, regarding

Table 1: Forensic data types defined in [7] (updated from [9])

DT_i	Forensic data type	Description (according to [7])
DT_1	Raw data	A sequence of bits or data streams of system components not (yet) classified
DT_2	Hardware data	Data not or only in a limited way influenced by the OS and application
DT_3	Details about data	Meta data describing other data
DT_4	Configuration data	Modify the behaviour of the system and applications
DT_5	Communication protocol data	Modify the communication behaviour of the system
DT_6	Process data	Data about a running process
DT_7	Session data	Data collected by a system during a session
DT_8	User data	Contents created, edited or consumed by the user

the traffic generated from the payment gateway, we identify companies listed in their terms and conditions or third-party data sharing practices. Subsequently, these companies are searched in the traffic. This last step, may result in false positives, that is, a company listed in this process, may be already listed in the merchant's website. Thus, those detected in both comparisons are marked twice (i.e., one originating from the merchant, and two originating from the payment gateway company) for a thorough inspection in step DA;

(5) **Data analysis (DA)** *is a set of actions where data traces are separated and visualized* [7]: subsequently, packets previously marked in conversations are studied in detail. It is possible to inspect headers and payload of the packets since the TLS encryption is circumvented.
Particularly, we analyze forensic traces originated from the DS_{N_1} and DS_{N_2}. Potential traces that could be relevant for an investigation are categorized into *Data types*. Table 2 summarizes the notation;

(6) **Documentation (DO)**: in this step we describe the proceedings, also could be included non-technical descriptions. In our case, this paper is the output of this step as described in [7].

Table 2 shows the aforementioned investigation steps, the network data streams, data types, chosen forensic and exploited methods in the analysis, and its tools, which are further described in section 4. Particularly in the *Segmentation* step of DI, the semi-automated analysis of captured traffic based on an adaptation of Libert's methodology in [13] is executed as follows:

(1) TCP conversations captured in step DG are saved in a *pcap* file (output from *Wireshark* tool suite, see table 2). Company server owners are automatically identified (employing python libraries) along with Fintech companies which appear in server's domain for each captured conversation. Then, we filter and identify TCP conversations generated by possible third-party trackers in the merchant's site substantiating the result with the original *Webxray*'s output [1]. Thus, traffic generated from ad-trackers not related to the payments are marked as "advertising". These packets are out of the scope of our analysis. However, they are not discarded, since they could be interesting for subsequent investigations. It may occur that traffic originated from apps installed in the device is also captured in the *pcap* file. To discard this traffic, we previously filtered it out manually identifying the source in the domain. However, we ensure in the experimental setup that the device has only the necessary applications to run the tests, thus reducing traffic noise. This information is reserved for a further comparison with the output of the following point;

(2) Previously in step OP, we collected the privacy policies links of the merchant and payment gateway companies. We store those links into a predefined list of HTML websites. Note that, in general in data protection statements, a company should provide a list of third-party companies classified by purpose (e.g., for payments, advertising, analytics). Further, third-party company names and links (if they are included) are extracted by web-scrapping. This information is stored into a database to compare it to the output of the previous step. This step is inspired by a module in *Webxray* tool, *policyxray*, which is a more complete version of our task (see [13]);

(3) For each domain (extracted from links) and company names stored in the previous step are compared to domains/companies contained in the network recording. For those conversations not identified, their packets are manually marked for a future inspection. These conversations can be traffic noise or an external non-expected connection. As a result of the segmentation process, we obtain a table with TCP conversations (ordered by time) which are segmented into four non-mutually exclusive sets: *advertising*, *web content*, *payment API*, and *not identified*; name of companies (server owners or Fintechs), including the source of those conversations, that is, merchant and/or payment gateway.

4 EXPERIMENTAL SETUP

This section describes the creation of the experimental setup, the material and tools used, and the protocol of actions performed for each test. The global scenario for analysis is based on a browser running in a mobile phone (see figure 1). We divided the experiment into two tests. The first test $T1$ is a donation in *unicef.de*, in which we study PayPal gateway ($T1.a$), and Klarna-sofort gateway ($T1.b$). In test $T2$, we conduct a payment in an online book store testing Amazon Pay gateway.

For the client setup, we used a *Client* (rooted phone Nexus 5 with Lineage OS (Android 6.0.1) [2]); a laptop (Ubuntu 18.04-Intel i7-7500U

[1] *Webxray* default setup identifies companies listed as advertising companies. Thus, it helps to filter our captured traffic.

[2] Lineage OS https://lineageosroms.com/hammerhead/

Table 2: Investigation step (except DO) map to network data streams, data types, exploit methods, and tools

Investigation Step	Accessed Data Streams	Data Types	Forensic exploit methods [3]	Tools implementing the forensic method
SP	–	Non-case specific	–	–
OP	–	DT_1	Operating system (OS)	Lineage OS - Android 6.0.1
DG	DS_{N_1}, DS_{N_2}	$DT_1\|_{DS_{N_1}}$, $DT_7\|_{DS_{N_2}}$	Explicit means of intrusion detection (EMID)	Man-in-the-middle proxy (MITMproxy - capture option), Wireshark Tool Suite (capture)
DI	DS_{N_1}, DS_{N_2}	$DT_{3-8}\|_{DS_{N_1}}$	Data processing and evaluation (DPE)	Wireshark Tool Suite, Python libraries (segmentation, and web-scraping)
DA	DS_{N_1}, DS_{N_2}	$DT_{3,5-8}\|_{DS_{N_1}}$	–	MITMproxy (visualization)

CPU 2.70GHz x2) as an access point, and man-in-the-middle proxy (*AP-mitm*), which is an interactive HTTP proxy called *MITMproxy*. The setup guidelines for the MITMproxy and certificate installation in Android for the TLS circumvention are described in [2]. The software and tools for DG, DI, DA, and to conduct the payment: Wireshark tool suite (network protocol analyzer[3]); Python 3 libraries; *Webxray* libraries[4]; and personal PayPal, Amazon credential accounts to conduct the payments, and personal online banking credentials for Klarna-sofort option. Figure 1 shows an exemplary diagram of the experiment setup, and further external connections to domains (S_1, S_2,..,S_n e.g., merchant, payment gateways, contacted in the tests). All material and analysis tools used in the experiment are open source, and online available.

4.1 Test 1 (T1): Donation in unicef.de

The payment process is performed in the following action protocol: (1) enter UNICEF website, read, and accept cookies conditions (default conditions); (2) click Donate; (3) input personal data, and choose payment option; (4) login into chosen payment option, and click "continue'; (5) client is redirected to UNICEF website.

4.2 Test 2 (T2): Purchase in book store

Particularly, for Amazon Pay option steps are similar to $T1$: (1) enter medimops.de, read, and accept cookies conditions (default conditions); (2) choose an item; (3) click Pay; (4) input personal data (or login), and choose payment option; (5) login into chosen payment option, and click "accept the payment"; (6) client is redirected to medimops.de.

5 RESULTS

This section presents the results of the analysis conducted in $T1$ and $T2$ for each step described in section 3.

5.1 OP and DG: API documentation and mobile traffic acquisition

5.1.1 T1: PayPal & Klarna gateways. We searched online for SDK documentation used for the payment website of *unicef.de*. Particularly for *unicef.de*, the website used a payment API from a software provider (i.e., FundraisingBox[5]) that connects to several payment gateways through another payment gateway called Stripe, which bridges to other payment gateways, such as PayPal and Amazon Pay. Commonly, these APIs embed trackers into the website, which generate traffic. The list of third-party trackers was available at the payment policies options, as well as privacy policy statement of UNICEF. Moreover, third-party data processors were listed in the privacy policies as cookie providers or payment providers. All listed companies along with privacy policy website links were documented. This information served as input of the semi-automated module for network traffic detection.

5.1.2 T2: Amazon Pay gateway. The payment gateways listed in the data sharing statement of the merchant's website[6] were Amazon Pay, Klarna-sofort, PayPal, and PayONE. Therefore, we expected them to appear during the traffic analysis. Regarding ad-tracking cookies, we did not accept the cookies, only those essential, as the merchant claims. However, this action does not assure that connections related to ad-tracking will not be present in the network traffic.

After gathering all public available documentation, we conducted step DG capturing all traffic between our device for both tests, and the corresponding servers according to the experiment setup and tools used.

5.2 DI: Semi-automated segmentation of traffic conversations

This step performed the segmentation of all TCP conversations established captured with Wireshark. The overall conversations related to the payment process for test $T1$, and $T2$ are summarized in table 3. It shows domains invoked by scripts embedded in the merchant's website, and in payment gateways sites. The companies

[3] https://www.wireshark.org/
[4] https://github.com/timlib/webxray

[5] https://support.fundraisingbox.com/article/combining-items-and-suggested-amounts-in-the-form-462.html
[6] Terms and conditions: https://www.medimops.de/Datenschutz/

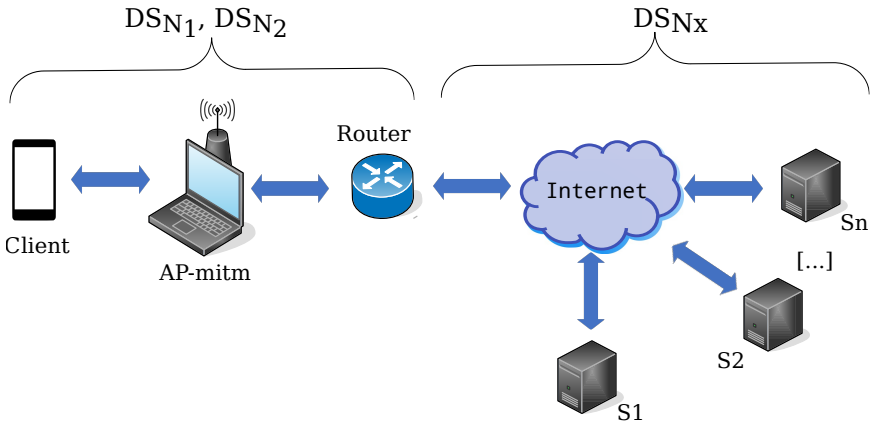

Figure 1: Experiment setup scenario. Data stream $DS_{N_{1,2}}$ are available for our analysis as described in section 3. Data streams DS_{N_x} are unknown for the current investigation.

listed were the owners of the domains and servers. In addition, column *Found in protocol action* shows the steps in which each conversation appeared. Sources and destination IP addresses are mapped onto DT_5. The overall conversations were expected according to merchants' data protection statement, and third-party data sharing policies. However, conversations with third parties originated from payment gateways, were highlighted as further inspection. In $T1.a$, the highlighted conversations were *pop-edc2.mix.linkedin.com*. For $T1.b$, and $T2$ there was no third-party conversation.

5.3 DA: Traffic content analysis

For this step, we examined data types DT_3, DT_5,DT_6, DT_7, and DT_8 which are found in packet content of HTTP traffic in decrypted domain. We searched for attributes categorized in each DTs. The table 4 shows the data types, some attributes found in the analysis from in each category, the location or field in the packet analysis, and an example of notation where DT_i occurs regarding DS_{N_i} and S_n in the landscape (figure 1). For this, we accessed to HTTP headers information using the MITMproxy interface, and examined firstly, the conversation highlighted as third-party from the payment gateway. Secondly, traffic generated from the payment API which may contain DT_8 retrieved by e.g., JavaScript, queries, and lastly, HTTP traffic with POST methods.

During the process of $T1$, we introduced our personal data in the payment form provided by *elasticbeanstalk.com* called by *secure.fundraisingbox.com*. Once the payment option is selected, we clicked Donate to forward our client to PayPal or Klarna servers. In this process, we established a conversation with *unicef.de* for email validation, and with servers *secure.fundraisingbox.com*, and *stripe.com*. We have found that personal identifiable information (PII) is sent to servers*unicef.de*, and *secure.fundraisingbox.com*. Concretely, for UNICEF's connection, the email is sent in clear text in decrypted domain. For the API connection, there was a GET method to *secure.fundraisingbox.com*, prompted by a query, which included the content of the payment form, that were our name, surname, address, and email in the URL. The form included fields such as credit card number, and expiry date. On the other had, no PII has been found in Stripe server's connection neither in the headers nor body.

5.3.1 Test 1.a - PayPal gateway. According to Paypal's developers documentation (obtained in OP), when our client is redirected to PayPal services, a requested script generates a time-limited alphanumeric token (i.e., *EC-12345678*) which shares with the merchant and tracks the whole transaction. This token is appended to the URL according to the API information[7]. Searching for the number of occurrences of the token, we find it in a conversation with the domain *pop-edc2.mix.linkedin.com*. Investigating further these HTTP headers of these packets, we find that in the *referer* field of the HTTP header there is the PayPal URL with the ID token. At this point, we have observed in steps OP and DI outputs, that the company is approved in PayPal's list of third-party data sharing practices for marketing purposes employing web beacons. Several cookies are installed during these conversations with Linkedin servers. The nature of these cookies are security purposes according to their description[8]. The cookie names are *bcookie* and *bscookie*, in which the former is a browser identifier cookie to uniquely identify devices addressing Linkedin servers, and the latter saves the state of 2factor user authentication, in case this specific devices authenticates to Linkedin servers. These type of cookies last up to two years. With this information, we revised the purposes of data sharing for Linkedin in PayPal's third-parties data sharing practices. According to the statement, the data disclosure was for Marketing and public relation purposes. In the details, the description stated the data shared were "Company name, encrypted e-mail address of PayPal users (without specifying the bank account details)".

Regarding only PayPal's connections, we observe HTTP POST and GET methods for cookies installation in the client. Concretely, the POST method includes the login email in the cookie field of the header. Moreover, since we are at decrypted domain, we check if the user's login credentials for PayPal are encrypted within TLS encryption. However, login name (i.e., email) and password are in plain text in the body of the packet.

[7] More information about PayPal API in https://developer.paypal.com/docs/payflow/express-checkout/

[8] https://www.linkedin.com/legal/l/cookie-table#thirdparty

Table 3: Domains captured while conducting payment test $T1$. Data are mapped to DT_5

S_n	Domain S_n	Purpose	Fintech(s)/Server owners	Found in protocol action
1	www.unicef.de	web content	UNICEF, OpenIT GmbH	1,2,3,5
2	(A) stripecdn.map.fastly.net, (B) js.stripe.com, (C) m.stripe.com, (D) stripe.com	payment API	Stripe, Fastly	2, 3
3	fb-env-1.eu-central-1.elasticbeanstalk.com	web content	Amazon	2, 3
4	secure.fundraisingbox.com	payment API	FundraisingBox	2, 3
	Test T1.a			
5	(A) fastly.glb.paypal.com, (B) dualstack.paypal-dynamic-2.map.fastly.net, (C) paypal.map.fastly.net, (D) dub.stats.paypal.com	payment API	PayPal, Fastly	4, 5
6	pop-edc2.mix.linkedin.com	advertising	LinkedIn Corporation	4
	Test T1.b			
7	production.eu2.sofort.klarna.net	payment API	Klarna, Amazon	4
	Test 2			
8	(A) www.medimops.de, (B) images2.medimops.eu	web content	medimops.de, Cloudflare, Inc.	1,2,3,5
9	events.production.eu1.fer.klarna.net	payment API	Klarna, Amazon	5
10	pay1.de	payment API	PayONE, Network of PayONE FRA	5
11	payments-de.amazon.com, static-na.payments-amazon.com, c.media-amazon.com	Payment API	Amazon.com, Inc.	4, 5, 6

Table 4: Attributes found in the analysis, their data categorization, location or field within the packet, and an example of notation where DT_i has been located in the landscape

DTs	Attributes in data	Location	Example
DT_3	Country, language	HTTP requests, URL address	$DS_{N_{1,2}}$ $\forall$conversations $Client \leftrightarrow S_n$
DT_5	IP Address, Domain	IP info in packet (see table 3)	DS_{N_1} $\forall$conversations $Client \leftrightarrow S_n$
DT_6	Referrers, User Agents, Do-Not-Track policies	HTTP Header (GET methods)	DS_{N_1} for $Client \leftrightarrow S_6$
DT_7	authentication tokens, transaction identifiers, cookie types	HTTP headers (GET method)	DS_{N_1} for $Client \leftrightarrow S_6, S_9$
DT_8	PII: email and password, address, name	HTTP packet payload, HTTP Request (GET/POST method)	DS_{N_1} for $Client \leftrightarrow S_1, S_4, S_{8A}$

5.3.2 Test 1.b - Klarna gateway. Klarna-sofort payment worked with the user's online banking credentials, that is, we introduced our online banking credential to accept the payment. Following the same procedure as $T1.a$, we searched for DTs in HTTP packets. Concretely, in a POST method we found our login credentials, along with ID tokens in the body of the HTTP request. Unlike the PayPal case, Klarna's backend encrypted the user's password in the body.

5.3.3 Test 2 - AmazonPay gateway. For $T2$, traffic generated during protocol steps 4 and 5 conversation from *production.eu1.fer.klarna.net* were segmented for further inspection. During the inspection we observed many POST methods from this domain, however, we could not find any indication of DT_8 in clear text. We could observe that the name of form fields were in plain, but their values were encrypted.

6 DISCUSSION

The presented results showcase the interaction among Fintech companies, and their data sharing practices in detail. However, the analysis is taken only from the client's perspective, i.e., connections created among client-servers for a single payment process. Thus, we lose tracks of how information is exchanged among parties. With these results, we can successfully answer our research questions, since we could identify the number of companies ($Q1$), including what companies are involved in the payment and data sharing

(*Q2, Q3*), and the utilized security mechanisms (*Q4*) by merchant's checkout website and payment gateways.

For $T1$ we have up to eight companies involved in the payment beforehand, UNICEF, PayPal, and Klarna, and including payment software providers and web content providers, i.e., FundraisingBox, Amazon, OpenIT. For $T2$, only companies involved in the payment, and web content were presented in the traffic, which are medimops.de, Klarna, Amazon, PayONE and Cloudflare. Particularly for $T1.a$, third-party companies cannot be out of the equation from getting information about the payment, as it is demonstrated with the tandem PayPal-Linkedin.

Data sharing statements aim to explain this complex interaction bringing more transparency in their business processes. Nevertheless, policies are complex to understand and often ignored by the users, but usability evaluation of online privacy policies is out of the scope of this paper.

Following a structured forensic investigation, we could identify some important security issues involving DT_8 in DS_{N_1}. Once TLS protection is circumvented, the results from $T1$ have shown that there are no additional protection measures taken for PII. As in the case of FundraisingBox API and PayPal gateway SDK, PII in plain text is available to any attacker. Specifically for the European Union region, the EU's data protection regulation (i.e., GDPR) in its Article 32, states: *"the processor shall implement appropriate technical and organisational measures to ensure a level of security appropriate to the risk including inter alia as appropriate: the pseudonymisation and encryption of personal data;*[…]" [4]. We understand that this statement is not fully implemented in the $T1.a$ from our analysis. In contrast, in tests $T1.b$, and $T2$ a second encryption of login data was employed. When sensitive information is involved, a defense-in-depth approach should be applied in the communication as suggested by King et al.(2021), in [10]. Regarding how transaction data could be shared among Fintechs for marketing purposes, we found an example in $T1.a$. With the setting of cookies, the sharing of a token indicating that a specific device has made a purchase, and the data shared between companies, it is likely to identify an individual. As users conducting the payment, this information was not displayed during the checkout process. It should be noted that we selected the default setup for cookies at the time we made the payment at PayPal.

Our approach has limitations. Firstly, one of the branches of our methodology to obtain DS_{N_2} is semi-automated. This is due to web-scraping of privacy policies, which turns to be complex across sites. One reason is the variety and constantly changing of websites' structures. This challenge is also considered by Libert [13] for *Webxray*. We believe that an improvement for this particular task of our approach would be to capture user interaction across websites, and gather all data protection statements involving the payment process. However, we do not rule out that this proposal may present other challenges. A second limitation towards a fully automated traffic analysis is relying on human supervision to evaluate transparency in the data protection statements according to what has been found in the network traffic. Moreover, the set of attributes for further analysis could be not limited to those presented in our case. Other identifiable attributes can be considered such as, phone's IMEI, and Google Ads ID number. Notwithstanding these limitations, we obtained satisfactory results and usefulness applying our structured forensic approach with the semi-automated traffic segmentation.

7 CONCLUSIONS

Online payments are a complex case of data exchange among Fintech companies. When accessing to a merchant's website to purchase a good, multiple connections (client-servers) are generated by scripts, or requests to other third parties which may store forms, images, and other type of data. Some connections are for personalized advertising and analytic purposes, others necessary to conduct the payment. In such scenario, data and entities traceability is a challenge. This paper introduces the application of a computer forensic method enhanced with a semi-automated traffic analysis for the identification of potential security and privacy issues in online payments. Moreover, the paper yields a structured procedure for the identification of data traces. For such purpose, we set an examination scenario which circumvent TLS encryption for a detailed data inspections and their classification. We tested three different payment gateways offered by three Fintech companies popularly known: PayPal, Klarna-sofort (in Germany), and AmazonPay. We documented what companies are expected and confirmed to obtain our data from our chosen marketplaces, what are the security measures taken for the data sharing, and what information about these data sharing practices is available publicly. In the analysis process, we could identify our personal and login data in plain text in different parts of the packet structure, contrasting with data sharing statements and term and conditions of the entities involved in the payment process. Such security weaknesses in e-commerce websites (often affecting medium and small size merchants with low security detection capabilities) as observed in the paper, facilitate web-skimming attacks by injecting malicious scripts seeking to collect user's payment information into merchant's checkout website, for further selling it in darkweb marketplaces. Moreover, digital fingerprinting is another threat to secure online payments, since attackers may profile a device and user's purchases in a compromised merchant's website. Note that both attacks are increasing lately, becoming the first major threat for financial institutions. The experimental setup and analysis approach will be used in future work to study other types of smartphone-based online payments, such as in-app payments. The presented approach may help in the identification and documentation of security and potential transparency issues in police investigation involving online payment frauds, and data protection officers on investigating privacy violations, respectively.

ACKNOWLEDGMENTS

The work presented has been supported by the Bundesministerium fuer Bildung und Forschung (BMBF), under the project FINANzkriminalitaeT: MethodIsche Analyse von Bedrohungsszenarien fuer moderne Karten- und App-basierte Zahlungssysteme (FINANTIA; FKZ:13N15297). We would like to thank Mario Hildebrandt, and Jose Ramon Gascon-Manzano for their suggestions in the process of the experimental setup, and traffic analysis.

REFERENCES

[1] 2020. Hackers Using Google Analytics to Bypass Web Security and Steal Credit Cards. https://thehackernews.com/2020/06/google-analytics-hacking.html Accessed 7 May. 2021.

[2] Aldo Cortesi, Maximilian Hils, Thomas Kriechbaumer, and contributors. 2010–. mitmproxy: A free and open source interactive HTTPS proxy. https://mitmproxy.org/ [Version 6.0].

[3] Kevin Dennis, Maxat Alibayev, Sean J. Barbeau, and Jay Ligatti. 2020. Cybersecurity Vulnerabilities in Mobile Fare Payment Applications: A Case Study. *Transportation Research Record: Journal of the Transportation Research Board* 2674, 11 (sep 2020), 616–624. https://doi.org/10.1177/0361198120945982

[4] EuropeanParlamient and of the Council. 2016. Directive 95/46/EC (General data protection regulation). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679 Accessed 7 May. 2021.

[5] Crista Giering, Fnaves, Andrew Conway, and Adam McNeil. 2021. FluBot Android Malware Spreading Rapidly Through Europe, May Hit U.S. Soon. https://www.proofpoint.com/us/blog/threat-insight/flubot-android-malware-spreading-rapidly-through-europe-may-hit-us-soon Accessed 7 May. 2021.

[6] Eric Hutchins, Michael Cloppert, and Rohan Amin. 2011. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. In *The Proceedings of the 6th International Conference on Information Warfare and Security*. Citeseer, 113.

[7] Stefan Kiltz. 2020. *Data-Centric Examination Approach (DCEA) for a qualitative determination of error, loss and uncertainty in digital and digitised forensics.* Ph.D. Dissertation. Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik. http://dx.doi.org/10.25673/34647

[8] S. Kiltz, J. Dittmann, and C. Vielhauer. 2015. Supporting Forensic Design - A Course Profile to Teach Forensics. In *2015 Ninth International Conference on IT Security Incident Management & IT Forensics (IMF)*. IEEE Computer Society, Los Alamitos, CA, USA, 85–95. https://doi.org/10.1109/IMF.2015.16

[9] Stefan Kiltz, Tobias Hoppe, and Jana Dittmann. 2009. A New Forensic Model and Its Application to the Collection, Extraction and Long Term Storage of Screen Content off a Memory Dump. In *Proceedings of the 16th International Conference on Digital Signal Processing* (Santorini, Greece) *(DSP'09)*. IEEE Press, 1135 –1140.

[10] S. T. King, N. Scaife, P. Traynor, Z. Abi Din, C. Peeters, and H. Venugopala. 2021. Credit Card Fraud Is a Computer Security Problem. *IEEE Security & Privacy* 19, 02 (mar 2021), 65–69. https://doi.org/10.1109/MSEC.2021.3050247

[11] Alissa Knight. 2019. Alissa Knight: How I hacked 30 mobile banking Apps & the future of API Security. https://www.youtube.com/watch?v=hoZ3YY-2B2E. https://www.youtube.com/watch?v=hoZ3YY-2B2E

[12] Timothy Libert. 2015. Exposing the hidden web: An analysis of third-party HTTP requests on 1 million websites. *arXiv preprint arXiv:1511.00619* (2015).

[13] Timothy Libert. 2018. An Automated Approach to Auditing Disclosure of Third-Party Data Collection in Website Privacy Policies. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 207–216. https://doi.org/10.1145/3178876.3186087