Analyzing Leakage of Personal Information by Malware

Tobias Urban [a,b,*], Dennis Tatang [b], Thorsten Holz [b], and Norbert Pohlmann [a]
[a] *Institute for Internet Security, Westphalian University of Applied Sciences, NRW, Germany*
E-mails: urban@internet-sicherheit.de, pohlmann@intenret-sicherheit.de
[b] *Horst Görtz Institute for IT Security , Ruhr-University Bochum, NRW, Germany*
E-mails: dennis.tatang@rub.de, thorsten.holz@rub.de

Abstract. Advertisements are the fuel that runs many online services such as websites or mobile apps, but also adversaries started to abuse ads for financial gains. Nowadays, online advertising companies track users all over the web in order to create successful online ads campaigns specifically tailored for a target audience. A popular phenomenon on the Internet, so-called *adware*, abuses online advertisements by maliciously injecting or replacing ads on websites. As many consider ads to be quite privacy intrusive, much work has gone into studying the effects of online advertisements on users' privacy. However, only little work has been done so far into analyzing the privacy implications of adware.

In this work, we shed light on the capabilities, mainly concerning tracking and personal data exfiltrating, of adware and potentially unwanted programs (PUPs), at scale. To this end, we capture the communication of adware/PUPs in the Firefox browser on the application level to circumvent lower-level encryption (e. g., TLS). Using this framework for capturing the network traffic, we dynamically analyze the communication of over 16,000 adware or potentially unwanted program samples. We find that around 37% of requests issued by the analyzed samples contain some kind of personal information. Furthermore, we identify the services used by adversaries and provide insights on the used tracking techniques.

Keywords: Adware, Potentially Unwanted Programs, Privacy, Online tracking, Data leakage

1. Introduction

Web browsers became one of the most important types of application and they are especially relevant for providing us access to modern web applications for specific tasks such as e-mail, spreadsheets, or image editing. Furthermore, browsers enable us to interact with each other, share ideas, or even give access to a broad variety of multimedia content such as music or video streaming services. Due to this broad usage of browsers, they mediate a massive amount of personal data which also increases over time. Consequently, adversaries started to target the browser ecosystem with novel attack vectors (e. g., banking fraud or man-in-the-browser attacks). Especially malicious software that tampers with the browser session, such as potentially unwanted programs (PUPs), adware, and malicious browser extensions, pose an important threat for users today. These new threats include injection or replacing ads on websites which is an easy way for adversaries to make a financial profit [1].

First and third party user tracking is a commonly known part of the business model of most websites and other online applications (e. g., mobile apps) [2–8]. Personal data (especially clickstream data) is

[*] Corresponding author. E-mail: urban@internet-sicherheit.de.

collected and analyzed in order to create behavioral user profiles that can be used for targeted advertising [9]. As malware starts to use ads for personal gains, questions about the privacy implication of this kind of malicious software arise.

While similarities in these topics exist there are technical and motivational differences why users are being tracked. On a technical level, users do not give consent or know about the installation—and therefore the tracking—of adware and PUPs, in contrast to websites where tracking is commonly known. As adware and PUPs have rich access to the users' device and can therefore access data inside and outside the browser, websites and extensions are limited to the browser. Once an adversary infected a system, she can easily track every step of a user e. g., by injecting a tracking object into every visited website. Thus, the adversary can silently create comprehensive user profiles that might contain highly sensitive personal data which might be of high value for some ad companies.

On the contrary, on a motivational level websites want to enhance the users' experience while browsing their site (e. g., by suggesting products the users might like). To do so, they monitor the users' behavior on the website and try to interfere the users' interests. Browser extensions might collect personal data in order to provide their service to the user (e. g., an extension might collect the users' passwords to store them in a protected vault which can be used on different devices). However, the motivation of malware to exfiltrate personal data is purely malicious. For adversaries, classical Internet-related fraud (e. g., credit-card fraud) is increasingly difficult due to new security measures. Thus, adversaries look for new ways to make a profit (e. g., ransomware or ad injection). Behavioral profiles of users or data to build such profiles (e. g., clickstreams) can be sold to third parties [10]. Selling such profiles might be another way for adversaries to make a profit.

In this work, we examine this phenomenon on a larger scale and report on privacy leakage and user tracking of PUPs and adware. To the best of our knowledge, we are the first to study this topic on a larger scale for both adware and PUPs, and we focus on privacy implications of malware in general. More specifically, we address unnoticed privacy implication of adware and PUPs in this paper. Our results show that adware and PUPs mainly exfiltrate clickstream data of users which provide great insight into the personal, digital life of their victims. More than a quarter of the analyzed malware samples (27% of the analyzed adware and 30% of the analyzed PUPs) leak the full URL visited by the users. Additionally, we show that the leakage of personal data is a significant part of the malicious behavior of the analyzed malware. We also identified popular data sinks used by the analyzed adware and PUPs samples which are often located in Asia. Previous work found the high prevalence of adware and PUPs [11], which shows that this leakage of personal data is a considerable threat to all Internet users.

In summary, we make the following four contributions in this paper:

- We present a framework that is capable of capturing the network traffic emitted by a given browser on the application level (Section 4), which allows us to analyze traffic of any software that tampers with the users' browser session.
- We provide detailed insights into the impact of PUPs and adware on users' privacy. In our measurements, over 45% of all analyzed adware and PUPs samples leaked personal data or tracked users (Section 4.3). To the best of our knowledge, we are the first to report on data leakage and profiling by adware and PUPs on a large scale.
- We identified (1) the services used to track users, (2) the websites most commonly tracked, (3) and data predominantly exfiltrated by adware or PUPs (see Section 5.1).
- Finally, we present an analysis on objects not used to track the user or to leak personal data (e. g., images and style sheets) (Section 5.2).

This paper is an extended version of the paper: "*Towards Understanding Privacy Implications of Adware and Potentially Unwanted Programs*" presented at the European Symposium on Research in Computer Security (ESORICS) 2018 [12].

2. Background

In this section, we explain the terms *adware*, *potentially unwanted programs*, and browser extensions. Further, we give a brief overview of the adware ecosystem and describe several tracking mechanisms.

2.1. Adware, Potentially Unwanted Programs, and Browser Extensions

In this work, we analyze two different types of software, namely adware and potentially unwanted programs (PUPs). We further analyze browser extensions to assess our results and to make them more comparable to other related work (e. g., [1, 2, 13, 14]). In the following, we explain these types of software and discuss how we understand them in the scope of this work:

(1) **Adware** is (malicious) software that generates revenue by displaying ads to users (e. g., by injecting or replacing ads on websites). Aside from the ad injection, adware often redirects search requests to advertising websites or collects private data of the users (e. g., clickstream data). Commonly, adware is considered to be malicious if the collection of data or ad-injection happens without adequately notifying the user and if it is installed like other malware (e. g., drive-by-downloads).
(2) Potentially unwanted programs (**PUPs**), is a type of software that users might perceive undesirable, as it is installed along with software the user intends to install. The PUPs are bundled with popular benign software and are distributed by so-called pay-per-install services (PPI). PPI services get paid for installations of software (the installer bundle) on target hosts. PUPs could be software with any capability, malicious or benign. However, in the wild, this kind of software often shows similar behavior as adware [11] (e. g., ad-injection or user-tracking).
(3) **Browser extensions** are programs that extend the functionality of a web browser (e. g., block advertisements). Extensions have generous access to many functions provided by the browser.

In this work, we examine the negative privacy implications of adware and PUPs and compare these findings to extension downloaded from the Firefox Add-On repository [15]. In the past, adware or PUPs could come in form of an extension but due to policy changes of Firefox one can only install extensions present in their repository. This is probably why none of the analyzed samples successfully installed an extension. We focus on the negative privacy impact of adware and PUPs but also give hints regarding the "ad injection" and "search query redirection" capabilities of the analyzed samples (see Section 5).

As just defined, adware and PUPs have similar capabilities, and therefore it is reasonable to analyze both and compare them to each other. In order to make our results more comparable to previous work, we additionally analyzed browser extensions which are well explored regarding their (malicious) behavior. Of course, adware has more access to the operating system and could, therefore, come along with many other malicious capabilities than browser extensions. Therefore, we analyze the outbound network traffic that is not emerging from the browser ("second channel") to examine privacy breaches on that channel, too.

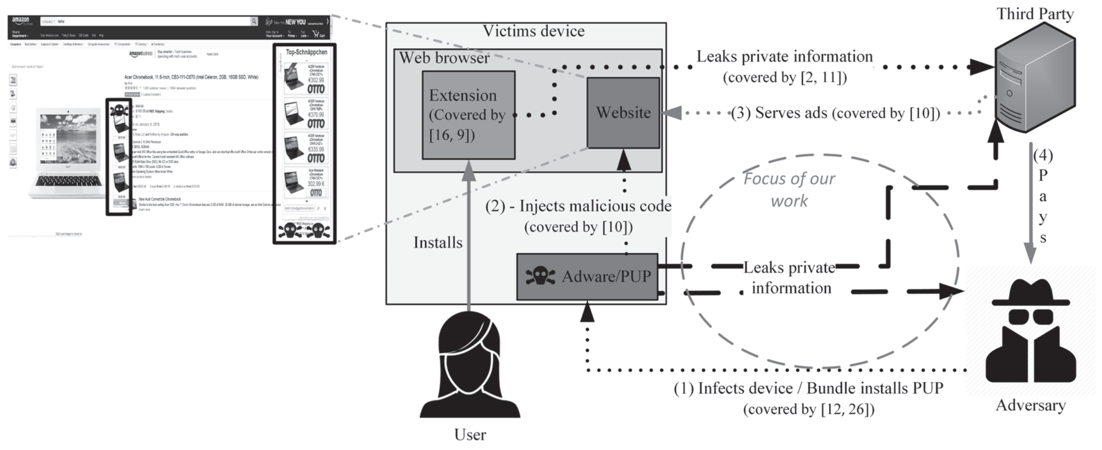

Third Party
Victims device
Web browser
Extension (Covered by [16, 9])
Website
Leaks private information (covered by [2, 11])
(3) Serves ads (covered by [10])
(4) Pays
Focus of our work
(2) - Injects malicious code (covered by [10])
Installs
Adware/PUP
Leaks private information
(1) Infects device / Bundle installs PUP (covered by [12, 26])
Adversary
User

Figure 1. Overview of the adware ecosystem. The adversary infects the victim's device with malicious software which insert ads into a visited website. After displaying the ads, or a click on the ad by the user, the adversary gets paid typically by a an ad network.

2.2. *Adware ecosystem*

The focus of this work lies in the analysis of privacy implications of adware and PUPs. The adware ecosystem is presented in Figure 1: (1) The user's system is infected with software (i. e., adware, PUPs, or extensions) that tampers with the browser session. (2) The extensions, PUPs, and adware inject their (malicious) objects (e. g., JavaScript code, or images) into the visited website. These objects might be used to load some content from a third party (e. g., ads), or might exfiltrate private information about the user. Many parts of the ecosystems are already well explored (dotted lines). In this work, we analyze the privacy implications of adware an PUPs for users (dashed lines). To the best of our knowledge, there has been no research analyzing this part systematically on a large scale.

The main monetization technique of adware (as the name hints) is injecting ads into websites and getting paid based on the payment model of the ad-network (e. g., pay-per-view) (3). Nevertheless, authors of adware, PUPs, or malicious extensions might also sell private data they exfiltrate from their victims [16] (4).

2.3. *Tracking Mechanisms*

Tracking mechanisms can be subdivided into *stateful* and *stateless* tracking methods. Stateful tracking identifies users through a unique identifier chosen by the tracker. On the contrary, stateless tracking tries to determine users through properties of the users' device or browser (e. g., installed fonts or drivers).

Two exemplary *stateful* tracking techniques are explained in the following:

- A *web beacon* (sometimes called *tracking pixel* or *web bug*) is often not larger than 1x1 pixel and usually a transparent graphic image, which is placed on a website for monitoring the user behavior [17]. It is often used with cookies as an additional tracking mechanism. Software that tampers with the user's browser session, like browser extensions, can insert such web beacons on every visited website.
- *Third party cookies* are a popular way to track users across different servers. In contrast to first-party cookies, which are set by the currently visited website, third party cookies are set, e. g., by content

loaded from the third party by the visited website. However, third-party cookies are set for the same reason than standard first-party cookies so that a visited website can identify a user later on.

Two examples of *stateless* tracking are browser and canvas fingerprints:

- *Browser fingerprinting* enables website providers to recognize and identify a user's system by unique properties of each browser. Eckersley demonstrates that a combination of browser and device features can almost uniquely identify most users on the web [18]. Web-based browser fingerprinting is, therefore, a conventional technique that has been investigated by several other researchers [18–21]. This technique can further be abused for customization of displayed products, e. g., recently Hupperich et al. showed that the location plays a role in the price offered for hotel bookings [22].
- *Canvas fingerprinting* is possible by abusing the HTML canvas element, that was introduced in HTML5, to draw graphics onto websites. Mowery and Shacham demonstrate that it is feasible to use for user tracking [7].

3. Related Work

In this section, we discuss work closely related to ours and explain how our approach relates to previous work on this topic.

Adware & Malicious Add-Ons

Jagpal et al. [23] present WEBEVAL, a system that identifies malicious extensions for the Google Chrome web browser. The authors identify different types of malicious extensions. The two most common types are Facebook session hijackers and ad-injectors (adware). Similar to our work, they perform a dynamic analysis of each extension and log how it interacts with the browser and operating system. Jagpal et al. do that by performing everyday tasks like querying search engines, visiting social media, and browsing favorite news sites. Aside from their dynamic approach they also conduct a static code analysis to decide if an extension is malicious or not.

HULK [13] is another framework that is used to identify malicious browser extensions. Hulk employs so-called *HoneyPages* and a technique called "event handler fuzzing". HoneyPages are empty HTML pages. If an extension queries for a tag on a website (e. g., *getElementById ("foo")*) this tag is automatically inserted into the HoneyPage. Thus, the extension assumes the element is present on the website and interacts with it. Using *event handler fuzzing*, *Hulk* pretends to visit all websites on the Alexa Top 1M [24] but just presents a HoneyPage to the extension.

Thomas et al. [1] combine *Hulk* and *WebEval* to measure the effect of malicious extensions on the websites google.com, amazon.com, and walmart.com. They report that 5% of the daily unique IP addresses visiting *google.com* are infected with malware that injects ads into websites.

ORIGINTRACER [8] is a tool developed by Arshad et al. , which allows tracking the provenance of web content injected into websites by web extensions. They evaluate the usability and performance of the introduced tool and show that such a tool is of great value for users to identify content that was injected into websites by third parties.

Neither HULK, WEBEVAL nor ORIGINTRACER target privacy implications but focus on identifying malicious browser extensions. We measured and analyzed the negative privacy impact for users that are infected by adware or PUPs.

Analysis About Fingerprinting On The Web

In a large-scale study, Acar et al. examine three advanced web tracking mechanisms (canvas fingerprinting, evercookies, and cookie syncing) [3]. According to their study, 5% of the top 100k websites use canvas fingerprints to identify users.

In 2010, Ashkan et al. conducted a study on the use of Flash cookies [25]. 50% of the websites in their set (Alexa top 100 sites [24]) use this kind of cookie mostly without disclosing this in their privacy policies. Note that since May 2011, all EU countries adopted a directive which says amongst others that websites have to display a "warning" to users if they use cookies [26].

FPDETECTIVE, a framework to analyze and detect web-based fingerprints, is introduced by Acar et al. [27]. They used their framework to crawl the most popular websites and analyze if the JavaScript code that is transmitted is used to create fingerprints. In their work, the authors show that fingerprinting is a growing problem and significantly more attractive than previous work suggested.

Englehardt and Narayanan [5] present the most recent study on online tracking. They introduce the open-source measurement tool OPENWPM, which they used to crawl and analyze the top one million websites on the internet. They measure several stateful and stateless tracking techniques and discover some methods that have not been noticed in the wild before (e. g., audio fingerprinting).

The introduced work measures the tracking capabilities and other privacy implications of modern websites. In this work, we analyze the exfiltration of private data and user tracking by malware, i. e., adware and PUPs.

Prevalence of Potentially Unwanted Programs

The prevalence and distribution of PUPs are examined by Kotziaset al. [11]. By analyzing AV telemetry, Kotzias et al. show that around 54% of 3.9 million analyzed hosts have PUPs installed. Furthermore, they found that the top PUP publisher ranks 15 among all software publisher (benign or not). They analyze the PUP-malware relationship and conclude that PUP and malware distribution is independent from another.

The pay-per-install (PPI) ecosystem is analyzed by Thomas et al. [28]. The authors show that PPIs sell access to the users' systems for prices ranging from 0.10$ to 1.50$ per installation. Furthermore, they show that PPI services take a considerable part in distributing PUPs. Based on Google Safe Browsing telemetry, they show that PUPs are downloaded three times more often than classical malware. Both works show the massive prevalence of PUPs but do not investigate the influence this type of software has on the users' privacy.

Privacy Implications of Browser Extensions

The privacy diffusion enabled by browser extensions is examined by Starov and Nikiforakis [2]. They dynamically analyze the privacy leakage of extensions available for the Google Chrome browser. They find that a non-negligible amount (6.3%) of the top 10,000 extensions leak privacy-sensitive data. To counter the leakage, they design BROWSINGFOG a tool to conceal the user's actual interest on the web. The tool pretends to visit different websites on the internet ("fog") which makes it arguably harder to distinguish between intended and non-intended page visits.

The most recent work in this field of research is written by Weissbacher et al. [14]. The authors present a prototype implementation called EX-RAY that can identify the privacy-violating behavior of browser extensions. In their work, they use an unsupervised learning approach to identify those extensions. The proposed experimental setup is comparable to our setup but only captures traffic on the network level. Thus, they cannot access and analyze the data, if they are transferred over a TLS secured channel.

The work of Starov and Nikiforakis is to some extent comparable to our work but, due to the nature of their analysis framework, does not cover tracking capabilities of extensions and does not look for exfiltrated metadata (e. g., user-agents or passwords). In [2] the software is analyzed that might need some personal information to successfully run their service (e. g., to identify malicious URLs). In contrast, we focus on malware that exfiltrates data in a purely malicious manner which foreshadows that there is a clear distinction between these two types of software. On a technical level we extend the findings of [2] by (1) identifying all exfiltrated data, (2) showing that there is a significant difference in type and amount of exfiltrated data, (3) identify websites to which visits are primary tracked, (4) analyzing the tracking behavior of malware, (5) determining the tracking services used by different malware families, and (6) identifying the used tracking techniques.

4. Approach

In this section, we introduce our framework, describe its working principles, inform about our analyzed data set, and give an overview of the investigated samples. Note that in contrast to most related work, due to the application-level monitoring, our system can even inspect HTTPS traffic, can find private data in encoded and deflated content, and allows a stateful analysis.

4.1. Framework

We developed a framework (see Figure 2) that allows us to (1) perform a *stateful* analysis of each sample, (2) *capture*, if needed *decrypt*, *decode* and *analyze* HTTP(s) communication on application level, and further (3) collect and analyze all network traffic not emerging from the browser.

The general workflow of a single analysis run goes as follows. The analysis slave pulls and installs an adware sample, PUP sample or extension from the server (1). Afterward, the slave visits a predefined set of websites (2a) and logs the resulting communication. To do so, we developed a browser extension that captures all network traffic on the application level. Since we save the traffic on the application level, we can inspect all requests and responses before or after they are encrypted or decrypted, by the TLS layer. After visiting a website, we wait for 30 seconds so it can finish loading and the analyzed software has time to inject content into the site. Additionally, we record all traffic on network level that is originated from aside the browser (2b). We cannot decrypt the traffic apart from the browser. Thus in our analysis, we are limited to the unencrypted traffic. At the end of the analysis run, the plain HTTP(s) traffic and the further communication is sent to the server for review. Before the analysis we—if needed and possible—inflate (e. g., *gzip*) and decode (e. g., *BASE64*) all data (see also Section 4.3).

In this work, we perform a *stateful* analysis which means that the used browser has properties that a mock browser or a default state would typically not show (e. g., a browsing history or cookies). If one wants to analyze the tracking capabilities of the software, it is inevitable to perform a *stateful* analysis because resetting the state of the browser during the investigation of a sample might disable some mechanisms that are used for tracking (e. g., cookies). The clean installation state of our slaves—that is recovered after each restart—has a browsing history, several cookies set, passwords in the browser's password vault, and other properties that are usually set when using a browser. Note that most prior work performs a stateless analysis of ad-injectors or browser extensions [1, 13, 27]. Only OpenWPM performs a stateful analysis [5].

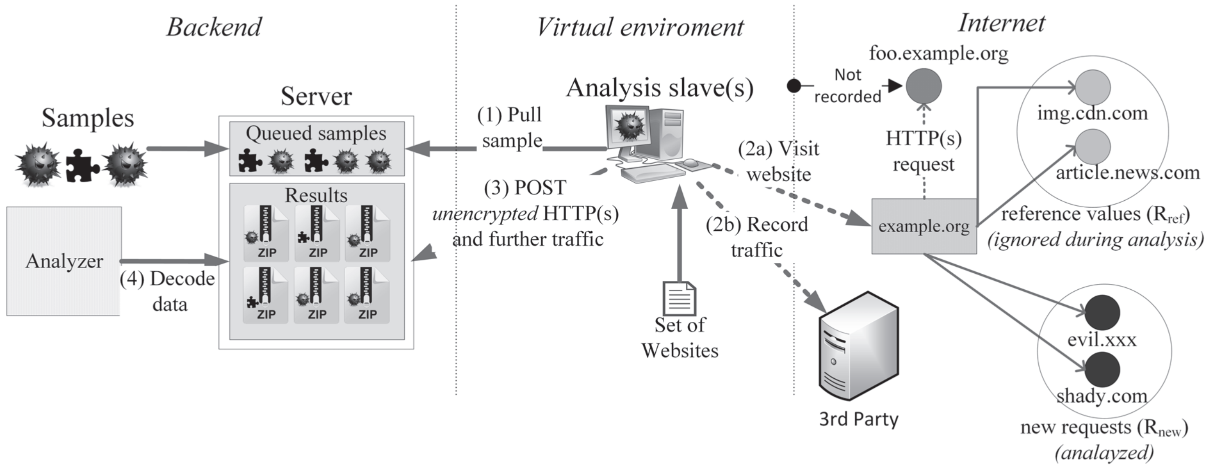


Figure 2. Overview of our developed framework for the dynamic traffic analysis of adware, PUPs and browser extensions.

To conduct a representative analysis, we need to learn the regular communication of a website to distinguish between requests regularly issued by the site and requests issued by an object injected by the adware, PUP, or extension. We collect the non-malicious regular communication of a website for our analysis by visiting all sites with an analysis slave— but without installed sample or browser extension.

Since websites tend to load dynamic content from various and often changing sources, each slave collects new reference values after analyzing two samples.All collected reverence values are combined to one reference set R_{ref}. In our analysis, we consider requests that target domains (TLD+1) that are not part of R_{ref} for a given site. We call that set R_{new} Example (see also the right side of Figure 2): Let's assume that R_{ref} for *example.org* contains requests to *cdn.com* and *news.com*. However, if an infected client visits *example.com* the websites issues requests to *evil.xxx*, and *shady.com*. In our study, we only consider requests *evil.xxx*, and *shady.com* because they are not in R_{new}.

4.2. Dataset

We used the global Alexa Top 100 [24] (as of 01/15/2017) as the basis for our set of websites which are visited by the analysis slaves. We restricted our analysis to unique hostnames from this list (e. g., we only analyze *google.com* even if *google.co.uk* is on the list as well) because we assume that the communication would be similar.

After filtering the sets consists of 57 domains. We added five popular e-commerce domains (e. g., *bestbuy.com*) because we expect the adware or PUPs to be more active on e-commerce websites, which turned out to be true for PUPs but not necessarily for adware (see Section 5. For each of those domains, we chose two subsites either randomly by visiting the domain and selecting two links, or if possible by selecting the most popular subsites for this site (e. g., products).

A more detailed overview of the set can be found in Appendix A. In total, the analysis of each sample takes around 70 minutes (including booting, infection, visiting the 128 websites, waiting 30 secs., etc.). Previous work either visited a broad set of websites once to conduct their analysis (e. g., [5]), used some mock pages to analyze the injected content (e. g., [13]), or did not disclose how many sites they visit (e. g., [23]).

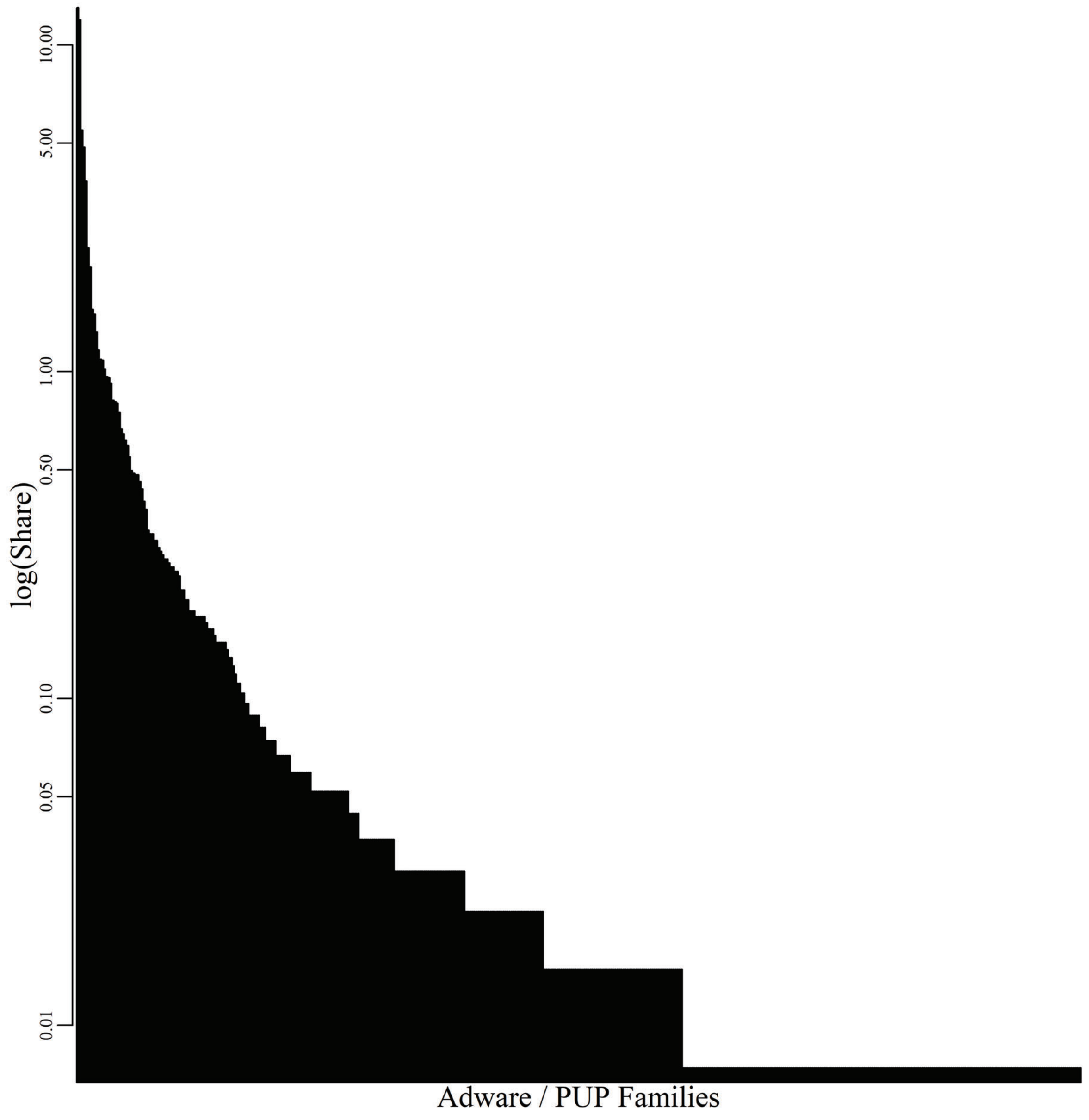


Figure 3. Distribution, on a logarithmic scale, of the analyzed malware sample families. One adware family (*Dealply*) is dominant in our set while the rest is more or less balanced - which allows us to generalize our results.

For our analysis, we used 8,536 distinct adware samples (referred to as S_{AD}) and 8,109 distinct PUP samples (S_{PUP}) (different regarding SHA256 hashes). The samples in $S_{AD} \cup S_{PUP}$ come from 484 different malware families (AV labels). Less than 12% of the samples belong to the most common adware family (*DealPly*), and 5% belong to the most common PUP family (*InstallCore*). The full distribution—on a logarithmic scale—of malware families is displayed in Figure 3. The distribution of samples across malware families shows that the data set is balanced and allows to generalize our results.

We used samples that were submitted to VirusTotal [29] between 01/01/2017 and 12/20/2017. Virus-

Total shut down their API in August and ever since then provides a data set for researchers on Google drive that is updated monthly. The used samples are either identified to be a potentially unwanted program (PUP) or adware by the anti-malware engines used by VirusTotal. We used samples with these labels because we expect that those samples will primarily exfiltrate private data and inject content into websites. To better assess our findings regarding adware and PUPs and to make our work more comparable with previous work, we analyzed the top 5,500 Firefox extensions (S_{ext}) available in the Firefox add-on repository [15]. According to the number of users, we took from the add-on repository, the top 5,500 extension cover 97.2% off all Firefox extension installations. Previous work focused on Chrome extensions, and therefore our analysis also complements these results.

4.3. Analysis

In the following, we focus on analyzing the communication of adware and PUPs. More specifically, we analyze the used tracking services, exfiltrated information, and tracked websites. Additionally, we compare these findings to privacy leakage of the browser extensions we analyzed and with results of previous work. A website can implement a Content Security Policy (CSP) as a defense mechanism to mitigate certain types of attacks like cross-site scripting or data injection attacks. During our analysis, we found that only 17 subsites use CSPs.

Exfiltrated Personal Information

In this work, we consider information to be private if it holds: (1) data that can be used to identify the client (e. g., IP-addresses), (2) can be used to create a user profile (e. g., visited URLs), or (3) contain sensitive data stored on the computer (e. g., passwords). We consider a website to be a tracker (or tracking service) if it gathers data that can be used to identify users or create profiles about them.

We identified the exfiltrated data by analyzing the transferred cookie, or data sent via the HTTP body. Individual headers can be used to gather personal information about the user (e. g., the user agent or user's preferred language), but these headers are commonly set by default. Hence, we cannot measure if the analyzed sample utilizes these fields. Before analyzing the fields we, if possible, deflate (e. g., *gzip/deflate*) and decode (e. g., *BSAE64*) them. If possible, we repeat this process in case fields are encoded or inflated multiple times, as observed by Starov et al. [2] (e. g., *base64_enc(base64_enc(url_enc(<data>))))*.

After the inflating and decoding, we perform a keyword matching to determine whether a request is used to leak private information. We identified the keywords by manual inspection of several requests issued by the different analyzed samples. We used 13 keyword categories that on the one hand are commonly used to identify or track users (e. g., screen resolution or installed fonts) and on the other hand information that is specific for our analysis setup (e. g., IP addresses or passwords). Some categories are identified by multiple keywords others just by one (e. g., the password is equal for all machines all the time while the user agent varies from sample to sample). We found 15,462 keywords in the analyzed requests. A manual inspection of a sample of the requests we identified a small (less than ten requests) to be false negatives (e. g., a keyword in a seemingly random string - *AR5**WIN7SP1**UFB2RI3*). A list of the most relevant keywords (based on their occurrence) is given in Table 2. Furthermore, we check if script code that is sent to the client within the response might be used to track users. If possible, we implemented several metrics provided in [5] and [27] to identify JavaScript that is used to track users.

To summarize, we consider a request to have negative privacy implication if and only if (1) it is part of R_{new}, and (2) it is used for tracking or contains private information.

5. Results

In this section, we provide an overview of the results of our analysis. Throughout this section, if not stated otherwise, we only consider requests used to track users or leak personal data to third parties.

In total, we analyzed 16,645 malicious software samples (8,536 adware samples and 8,109 PUPs) and 5,500 Firefox extensions. We analyzed about 850GB (compressed JSON data) of generated adware/PUP traffic. 45% of the adware samples, 40% of the PUP samples, and 45% of the Firefox extensions inject content into a website that issued requests to domains not present in R_{ref}. Our results, if not stated otherwise, only take these samples into account.

We found that the adware and PUP samples issued 21,429 requests to domains not present in our reference dataset, an increase of 10%. 61 of the adware samples changed the home page of the browser, and 221 changed the browser's standard search engine or redirected search queries. In contrast, only 6 PUPs changed the home page, but still, 180 replaced the default search engine. Due to Firefox policies, Firefox extensions cannot change these attributes.

5.1. Privacy Aspects

In this subsection, we present the results of the analysis of the HTTP(s) traffic emerging from the browser. Remember that our framework allows to (1) analyze all traffic in plain text—no matter if HTTPs was used or not—and (2) tries to deflate and decode all data before the analysis (e. g., *HTTP GET* parameters).

Tracked websites

Table 1 displays the top websites to which visits were actively tracked by the analyzed samples. We consider a website to be tracked if the analyzed sample injects content that can be used for tracking (e. g., a web beacon), or if an observed outgoing request contains any personal information. In our set of websites, each site is tracked by at least 1.5% of the adware and PUP samples. These samples circumvent the CSPs used by websites.

It is notable that the extensions and adware focus on popular websites (e. g., *Youtube* or *Instagram*) from different categories while PUPs predominantly focuses on shopping sites. This indicates that PUPs try to understand what a user plans to buy while adware is gathering information that gives a broader overview of the users habits since they track more general websites as well as shopping sites. Accordingly, this allows providing targeted ads for individual persons, making these kinds of information valuable for ad-companies. Overall, way fewer extensions exfiltrate personal information (31.64%) compared to adware and PUPs (46.41%).

Our results show that user tracking is a significant part of the malicious behavior of adware and PUPs. Almost 40% of the request issued by the adware samples, and 35% of the requests issued by PUPs contain personal information or may be used to track users (e. g., they include the visited URL: shady.com/?*url=google.com%2Fiphone%2B6*). In contrast, only 28% of the requests are used by the extensions for those purposes.

Leaked personal information

To measure the privacy impact, we first identify the transferred personal information triggered by the tested samples. We analyze the transferred cookie, and data sent in the HTTP body requests. Furthermore, we inspect if a response contains JavaScript that is used for stateless tracking or if the answer includes a web beacon.

Table 1
Websites that were actively tracked by the analyzed samples (Alexa Ranks as off 11/30/2017).

Adware				PUPs				Extensions			
%-Sam.	Website	Cat.	Rank	%-Sam.	Website	Cat.	Rank	%-Sam.	Website	Cat.	Rank
15.94	tmall.com	shopping	14	17.02	tmall.com	shopping	14	19.74	tmall.com	shopping	14
6.54	msn.com	misc	49	6.65	cnn.com	news	106	10.05	instagram.com	image	17
5.40	cnn.com	news	106	6.07	asos.com	shopping	360	9.40	youtube.com	video	2
5.28	youtube.com	video	2	5.96	ebay.com	shopping	38	7.11	microsoft.com	shopping	50
4.93	asos.com	shopping	360	5.90	target.com	shopping	283	6.13	cnn.com	news	106
4.06	youku.com	video	283	5.87	walmart.com	shopping	148	5.99	rakuten.co.jp	misc.	110
3.97	groupon.com	shopping	266	5.81	xinhuanet.com	adult	85	5.81	reddit.com	social	17
3.86	mail.ru	misc.	51	5.54	groupon.com	shopping	266	5.42	mail.ru	misc.	51
3.61	sogou.com	search	156	5.39	naver.com	misc.	49	5.12	bing.com	search	41
3.34	reddit.com	social	17	5.22	amazon.com	shopping	10	4.85	facebook.com	social	3

As described in Section 4.3, after deflating and decoding, we perform a keyword matching to determine whether a request leaks personal information usable for tracking mechanisms or not. Table 2 shows the results of that matching. Table 5 displays the third parties receiving the personal information. Note, if a request contains multiple keywords, we count the request numerous times.

In general, compared to PUPs, extensions and adware focus on meta information (e. g., language, time, IP address, etc.). The visited domain is exfiltrated by all analyzed software types alike (32%) while PUPs and adware predominately exfiltrate the full request URL (domain and GET parameters). However, one can argue that some extensions transfer this information as part of their service (e. g., an extension that checks if the users visit a malicious website will naturally send the current URL to a third party). In contrast, adware or PUPs leak personal data in a malicious manner or because the used ad services requires the current URL. In either way, the user's privacy is undermined unnoticed and without the user's consent. Table 2 shows that PUPs and adware, in contrast to extensions, focuses on the user's clickstream (i.e., browsing history). This is a more significant threat to the user privacy due to the detailed information leaked users' personal life (e. g., habits).

We can *not* identify any privacy-related information in about 6.9% of the requests issued by adware and PUPs (e. g., cdn.gigya.com/JS/gigya.js?apiKey=3_GL3L[...]) and 56% of the requests did *not* contain any data we analyzed (e. g., code.jquery.com/jquery-2.2.4.min.js).

To the best of our knowledge, there has not been any report on privacy breaches of adware and PUPs. Our measurements show that a significant part, more than ⅓, of the adware's and PUPs communication leaks personal information of users or tracks them. If one takes into account that the majority of the leaked data is the user's browsing history (Domain and URL in Table 2) this kind of leakage is way more severe than the extension leaks. Starov and Nikiforakis observed that several Chrome extensions, 6.3% of the top 10k, 'unintentionally' leak the HTTP referrer header to third parties (e. g., by embedding objects on every website) [2]. We observed a comparable leakage by 6.55% of the analyzed Firefox extensions and by 6.91% of the analyzed adware. We did not further investigate this unintentional leakage because the header provides only little utility for the adversary and there are several other ways for her to access this information (e. g., by merely reading the visited URL) and furthermore we cannot measure if the header is utilized. Naturally, the third party receiving the referrer header could use this information. Thus, this kind of leakage still poses a threat to the user's privacy.

Figure 4 shows which personal data is predominately collected by the most prominent malware families in our data set, along with the scaled amounts of samples leaking such data. To increase readability,

Table 2
Most commonly leaked personal information

	ADWARE			PUP			EXTENSIONS		
Information	%-S.	median	max	%-S.	median	max	%-S.	median	max
IP address	0.92	3	3	0.69	2	3	0.85	6	30
Operating sys.	5.49	2	5	5.54	2	5	6.21	2	30
User-Agent	5.41	2	2	4.77	2	3	5.35	14	60
Desktop res.	7.35	3	20	6.32	2	7	7.19	2	9
Domain	32.16	2	27	35.12	2	26	32.77	2	126
Full URL	27.18	2	13	29.52	2	10	15.56	2	66
Referrer leak	6.91	0	19	3.31	3	23	6.55	0	20

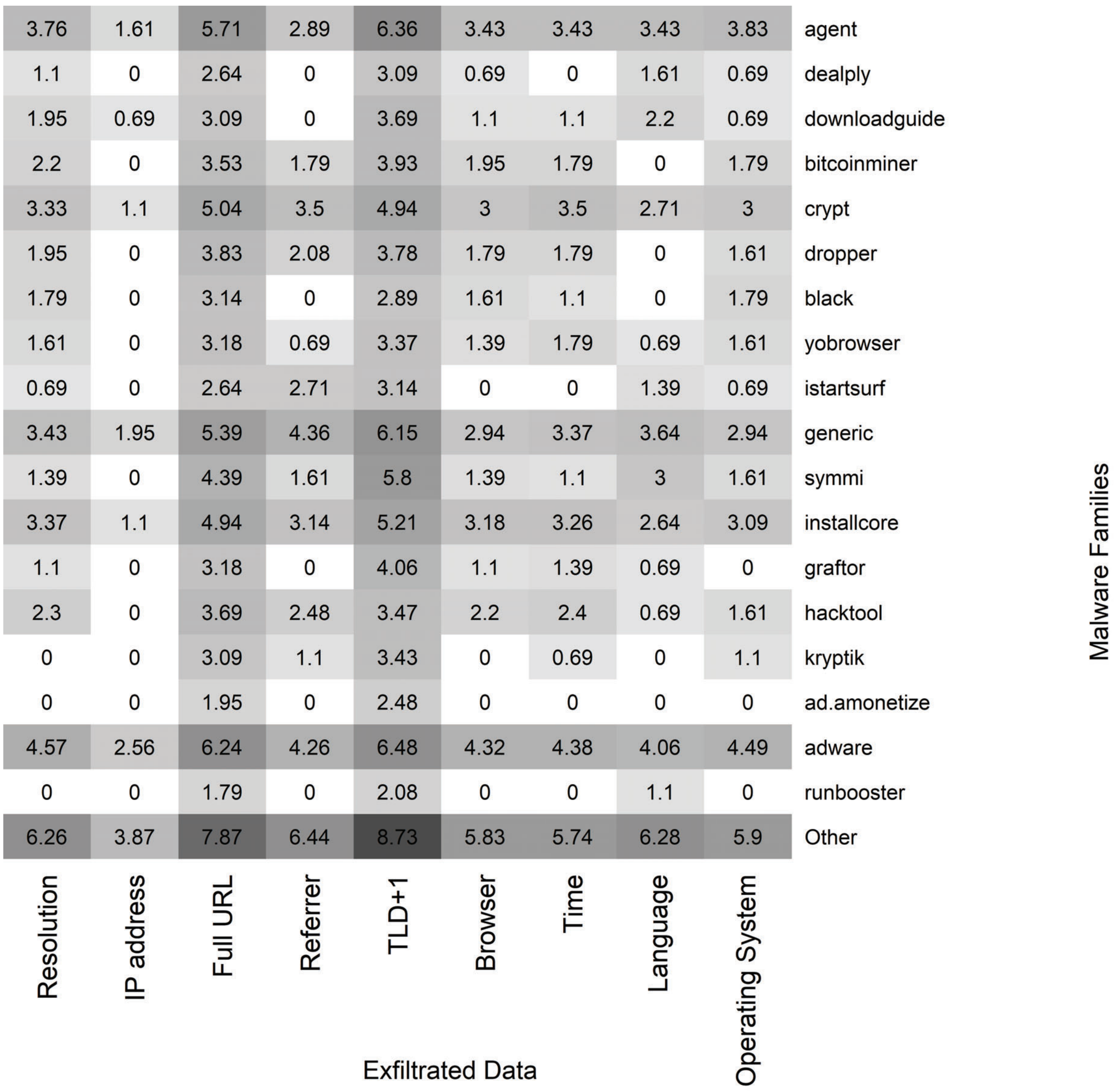
we only listed services used at least seven times by any family and the top 16 malware families individually and combined all other families to *Other*. All families exfiltrate clickstream data (i. e., TLD+1 and Full URL)—which is exfiltrated mostly (see as also Table 2). The families *Runbooster* and *Amonetize* only exfiltrate this kind of data while other families collect all data categories (e. g., *Adware* or *Agent*). Only seven families actively exfiltrate the IP address. However, they could extract the IP addresses from the requests they use to exfiltrate the data. We only consider an IP address to be leaked if it part of the HTTP GET or POST data.

Tracking services

Figure 5 displays the tracking services used by the different malware families, along with the scaled numbers of appearance. To increase readability, we only listed services used at least nine times by any family and again only the top 16 malware families. *Agent*, *Dealply*, the most common adware families in our dataset, and *InstallCore*, the most common PUP family in our dataset, are using a broad variety of tracking services One can see that *TaboTabo* and *MMStat* are overall the most common services used to track users. *taobao.com* is operated by *Zhejiang Taobao Network Ltd.*, while *mmstat.com* is operated by *Alibaba Co., Ltd.*. Both two big Chinese players in the Internet landscape. The third most common observed tracker, *GoogleVideo*, is a content delivery network—which is also a known tracker—used to host video or sound files.

Table 3 displays the most common services to which privacy-related information is leaked or which provide tracking tools (e. g., web beacons). Only one service gathers additional information about the client's system aside from the domain. All, but one, tracking services are operated by "big players" based in China. The analyzed extensions tend to use tracking services operated by American companies (e. g., *Google* or *Facebook*). Our results show that the services used by Firefox extensions are comparable to Google Chrome extensions [2].

In total, only 151 different trackers were used while 60 trackers where used by only three or fewer samples. This hints that adware and PUP authors tend to rely on existing infrastructure rather than setting up their own (in contrast to C&C communication structures of botnets). Among the observed tracking services, there is no indication for any preferred service. The top 20 services are used on average by 7.48% (± 0.78%) of the adware and PUP samples. This result indicates that the used services do not differentiate among each other regarding the utility for the adware or PUP.



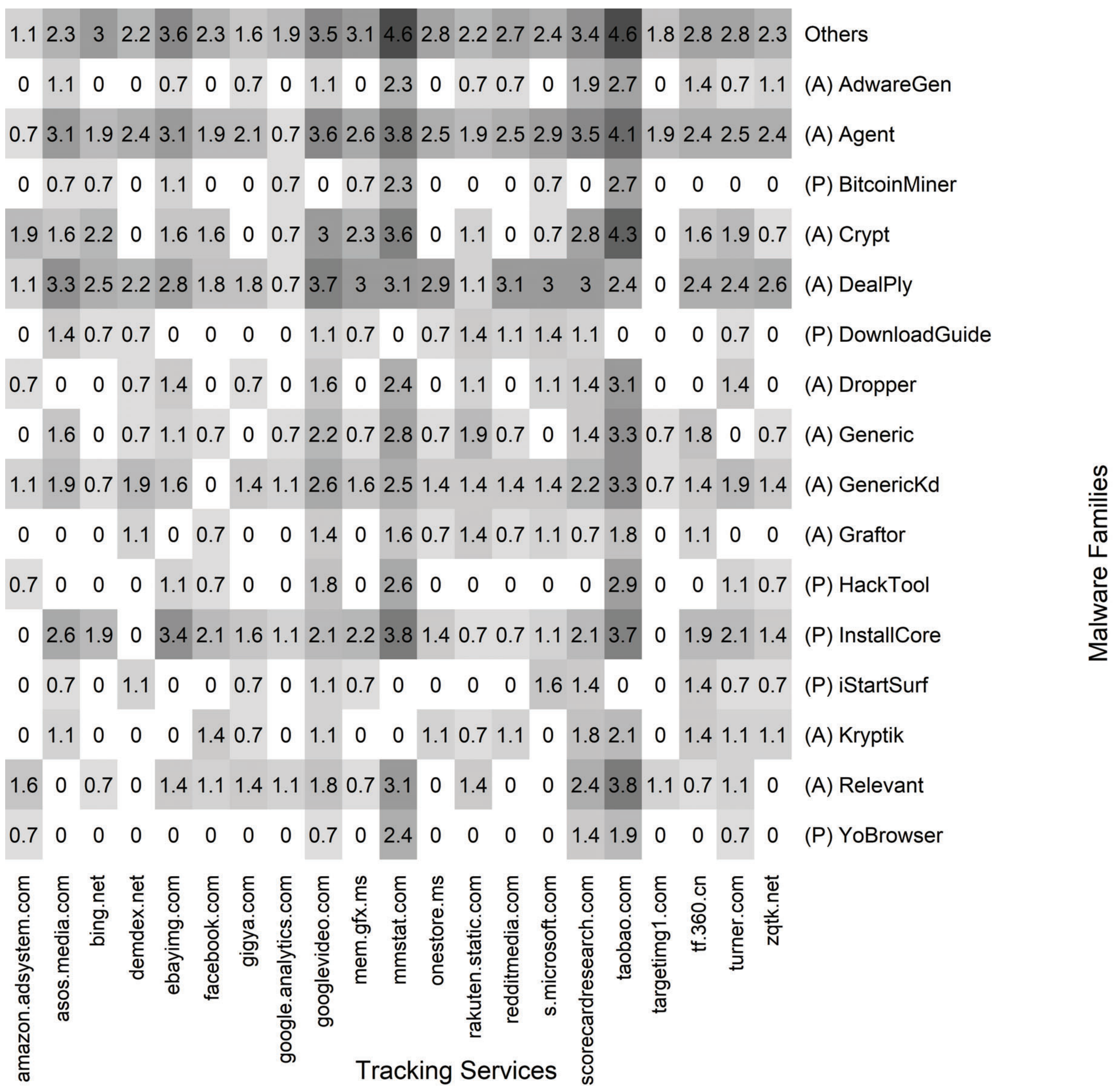
	Resolution	IP address	Full URL	Referrer	TLD+1	Browser	Time	Language	Operating System
agent	3.76	1.61	5.71	2.89	6.36	3.43	3.43	3.43	3.83
dealply	1.1	0	2.64	0	3.09	0.69	0	1.61	0.69
downloadguide	1.95	0.69	3.09	0	3.69	1.1	1.1	2.2	0.69
bitcoinminer	2.2	0	3.53	1.79	3.93	1.95	1.79	0	1.79
crypt	3.33	1.1	5.04	3.5	4.94	3	3.5	2.71	3
dropper	1.95	0	3.83	2.08	3.78	1.79	1.79	0	1.61
black	1.79	0	3.14	0	2.89	1.61	1.1	0	1.79
yobrowser	1.61	0	3.18	0.69	3.37	1.39	1.79	0.69	1.61
istartsurf	0.69	0	2.64	2.71	3.14	0	0	1.39	0.69
generic	3.43	1.95	5.39	4.36	6.15	2.94	3.37	3.64	2.94
symmi	1.39	0	4.39	1.61	5.8	1.39	1.1	3	1.61
installcore	3.37	1.1	4.94	3.14	5.21	3.18	3.26	2.64	3.09
graftor	1.1	0	3.18	0	4.06	1.1	1.39	0.69	0
hacktool	2.3	0	3.69	2.48	3.47	2.2	2.4	0.69	1.61
kryptik	0	0	3.09	1.1	3.43	0	0.69	0	1.1
ad.amonetize	0	0	1.95	0	2.48	0	0	0	0
adware	4.57	2.56	6.24	4.26	6.48	4.32	4.38	4.06	4.49
runbooster	0	0	1.79	0	2.08	0	0	1.1	0
Other	6.26	3.87	7.87	6.44	8.73	5.83	5.74	6.28	5.9

Figure 4. Stolen Data by adware / PUP family.

Along with the findings that ad-injection targets users in South Asia, and South East Asia [1] our results show that adware also uses services based in Asia. The usage of these services is understandable because access to big American tracking services (e. g., Facebook or Google) is not possible since they are blocked in China and other Asian countries [30].

Tracking techniques

Table 4 presents the tracking techniques utilized by the analyzed samples—only requests are listed that are used for a specific tracking technique. Previous work shows that stateless tracking is becoming



	amazon.adsystem.com	asos.media.com	bing.net	demdex.net	ebayimg.com	facebook.com	gigya.com	google.analytics.com	googlevideo.com	mem.gfx.ms	mmstat.com	onestore.ms	rakuten.static.com	redditmedia.com	s.microsoft.com	scorecardresearch.com	taobao.com	targetimg1.com	tf.360.cn	turner.com	zqtk.net
Others	1.1	2.3	3	2.2	3.6	2.3	1.6	1.9	3.5	3.1	4.6	2.8	2.2	2.7	2.4	3.4	4.6	1.8	2.8	2.8	2.3
(A) AdwareGen	0	1.1	0	0	0.7	0	0.7	0	1.1	0	2.3	0	0.7	0.7	0	1.9	2.7	0	1.4	0.7	1.1
(A) Agent	0.7	3.1	1.9	2.4	3.1	1.9	2.1	0.7	3.6	2.6	3.8	2.5	1.9	2.5	2.9	3.5	4.1	1.9	2.4	2.5	2.4
(P) BitcoinMiner	0	0.7	0.7	0	1.1	0	0	0.7	0	0.7	2.3	0	0	0	0.7	0	2.7	0	0	0	0
(A) Crypt	1.9	1.6	2.2	0	1.6	1.6	0	0.7	3	2.3	3.6	0	1.1	0	0.7	2.8	4.3	0	1.6	1.9	0.7
(A) DealPly	1.1	3.3	2.5	2.2	2.8	1.8	1.8	0.7	3.7	3	3.1	2.9	1.1	3.1	3	3	2.4	0	2.4	2.4	2.6
(P) DownloadGuide	0	1.4	0.7	0.7	0	0	0	0	1.1	0.7	0	0.7	1.4	1.1	1.4	1.1	0	0	0	0.7	0
(A) Dropper	0.7	0	0	0.7	1.4	0	0.7	0	1.6	0	2.4	0	1.1	0	1.1	1.4	3.1	0	0	1.4	0
(A) Generic	0	1.6	0	0.7	1.1	0.7	0	0.7	2.2	0.7	2.8	0.7	1.9	0.7	0	1.4	3.3	0.7	1.8	0	0.7
(A) GenericKd	1.1	1.9	0.7	1.9	1.6	0	1.4	1.1	2.6	1.6	2.5	1.4	1.4	1.4	1.4	2.2	3.3	0.7	1.4	1.9	1.4
(A) Graftor	0	0	0	1.1	0	0.7	0	0	1.4	0	1.6	0.7	1.4	0.7	1.1	0.7	1.8	0	1.1	0	0
(P) HackTool	0.7	0	0	0	1.1	0.7	0	0	1.8	0	2.6	0	0	0	0	0	2.9	0	0	1.1	0.7
(P) InstallCore	0	2.6	1.9	0	3.4	2.1	1.6	1.1	2.1	2.2	3.8	1.4	0.7	0.7	1.1	2.1	3.7	0	1.9	2.1	1.4
(P) iStartSurf	0	0.7	0	1.1	0	0	0.7	0	1.1	0.7	0	0	0	0	1.6	1.4	0	0	1.4	0.7	0.7
(A) Kryptik	0	1.1	0	0	0	1.4	0.7	0	1.1	0	0	1.1	0.7	1.1	0	1.8	2.1	0	1.4	1.1	1.1
(A) Relevant	1.6	0	0.7	0	1.4	1.1	1.4	1.1	1.8	0.7	3.1	0	1.4	0	0	2.4	3.8	1.1	0.7	1.1	0
(P) YoBrowser	0.7	0	0	0	0	0	0	0	0.7	0	2.4	0	0	0	0	1.4	1.9	0	0	0.7	0

Figure 5. Top tracking services used by the analyzed adware (A) and PUP (P) families.

more common on popular websites [5]. However, the analyzed adware samples and PUPs do not utilize stateless tracking techniques. This behavior is comprehensible since the samples can manipulate every website the user visits and therefore can inject a stateful tracking object into each site. Thus, they do not have to rely on more complex and error-prone stateless tracking techniques.

Our analysis shows that web beacons are the most common tracking method among all analyzed samples (adware, PUPs and browser extensions). This result is reasonable since they are easy to implement and are not as easy to block as third-party cookies. It is notable that extensions do not as often use web

Table 3
Top tracking services used by the analyzed adware and PUP samples, leaked information, and domain owners

Service	%-S.	Information	Company
taobao.com	10.04	URL, time, language, operating sys.	Zhejiang Taobao Network Ltd.
mmstat.com	8.47	URL, time, language, operating sys., browser, screen res.	Alibaba Co., Ltd.
sogoucdn.com	8.36	domain	Sogou Info. Service Co., Ltd
ebaystatic.com	8.28	domain	eBay Inc.
ykimg.com	8.23	domain	Nexperian Holding Ltd.

Table 4
Tracking techniques used by the analyzed adware and extensions. The vast majority tracks the users in a different way (e. g., by leaking the URL to a third party).

		Cookies	Web beacon	Stateless	Data leakage
Adware	(%-Sam.)	0.03 %	17.36 %	0.02 %	88.55 %
PUPs	(%-Sam.)	0.02 %	16.93 %	1.07 %	87.42 %
Extensions	(%-Sam.)	4.47 %	9.83 %	0.09 %	89.32 %

beacons but utilize 3rd party cookies more commonly.

The results indicate that user tracking is less critical to adware and PUP authors than exfiltrating personal data. But one can argue that exfiltrating the visited URL or domain is also a form of tracking. Requests that contain personal information but do not follow a specific tracking scheme are not considered (e. g., A request contains personal information and loads a picture bigger than a typical web beacon is not counted). The vast majority (around 88%) of requests that impact the users' privacy leak personal information.

Non-browser emitted communication

Since the full communication of the extensions is captured on the browser level; this section only considers the adware and PUP samples. The analysis in this section includes all (adware/PUP) samples even if they did not insert any object into a website.

Similar to the analysis of the traffic emitted by the browser, we used the communication of R_{ref} as reference values for non-malicious communication (e. g., connections issued by the operating system). The analysis in this chapter *excludes* all local traffic and traffic on the browser level. We found that 37% of the samples established connections by non-browser processes and communicated with over 200,000 IP addresses. 166 of these IP addresses were tagged as malicious according to a self-implemented blacklist web application that is crawling multiple blacklists, i. e., 30 different online provided blacklists. For usability reasons, our implemented blacklist service offers a REST API, to request IP addresses and domain names comfortable. We used our identified keywords to check if any private information is sent to any of these IP addresses (malicious or non-malicious). To do that we match the identified keywords against the payload of each packet. Encrypted traffic is not considered. Less than 0.5% of the packets contain meta information (e. g., operation system, or used language), and no packet contained clickstream data. This indicates that adware either only communicates from within the browser, gathers information and sends them in a later packet all at once, or that the communication is encrypted and was therefore not inspectable by our system.

Domain analysis

In this section, we present the results of our domain analysis for the domains in S_{new}. Our analysis focuses on two parts: (1) we checked if any of the domains are blacklisted, and (2) we perform *WHOIS* requests to get the registrars of these domains.

Only one domain is flagged to be malicious by the Google Safe Browsing API [31]. The domain was used by a pop-up window and is flagged to be used for social engineering attacks. Additionally, we used the Web-Of-Trust (WOT) API [10] to assess these domains. Five domains were blacklisted and six rated to be malicious by the WOT community. In total, 11% of the domains received a "negative" or "questionable" rating. Only slightly more than 8% of all domains are flagged to be used for tracking. This indicates that the services used by the adware are not commonly known for their tracking capabilities. Our findings show that the used services are not outright malicious and mostly serve legit purposes. However, the adware uses these services for different purposes (e. g., data exfiltration) than the desired purpose. Thus, these domains are not flagged (e. g., an Amazon bucket) to be used for tracking. Therefore, it is not always unambiguously decidable—on the domain level—if a domain is used for tracking.

Furthermore, we analyzed the registrars and organizations for these domains. *GoDaddy* is the most prominent registrar for the inspected domains and is used by both malware and extensions. In terms of registered domains, *GoDaddy* is the world leading registrar [32]. *MarkMonitor Inc.* is the second largest domain registrar in our data set. *MarkMonitor* focuses on enterprises who are interested in protecting their brand online. As for the organizations, most companies did not state their name or used a proxy company for the registration (e. g., *Domains By Proxy* or, *Perfect Privacy*). This applies to domains used by extensions as well as domains used by adware and PUPs.

5.2. Further communication

After analyzing the requests used to track users or to leak private information, we now analyze the requests used for other purposes (e. g., ad-injection).

Attacked websites

Table 5 lists the top websites into which objects (that issued requests) were injected that had no direct privacy implications. Examples for objects that might not issue requests are inline JavaScript or images embedded in BASE64 encoding. A distribution of the responses' content types can be found in Section 5.3. Our results indicate that the adware and PUPs circumvents the CSPs used by some websites.

In contrast to the tracked websites, the top attacked websites cover a broader field of categories. More than half (60%) of the websites into which the adware or PUP injected content are hosted in Asia indicating that malware authors tend to target that market. In contrast only 30% of the top websites into which extensions injected content are hosted in Asia. Previous work also observed that users affected by ad-injecting often live in South America, South Asia, and South East Asia [1].

Different adware samples and extensions seem to target different websites (the amount of samples injecting objects into specific websites is quite low), while PUPs samples seem to target similar websites (note that the amount of samples targeting a website is quite high). In contrast to adware, the extensions focus on American/ European websites. This is probably due to the low popularity of the Firefox browser in Asia [33]. *mall.360.com* was superseded by *i360mall.com* and thus the ranking dropped significantly during the course of our analysis.

Table 5
Top websites into which objects—that were not used to track the user or used to leak data—were injected.

Adware				PUPs				Extensions			
%-Sam.	Website	Cat.	Rank	%-Sam.	Website	Cat.	Rank	%-Sam.	Website	Cat.	Rank
11.89	cnn.com	news	106	17.43	sohu.com	misc.	18	13.87	asos.com	shopping	360
9.16	sohu.com	misc.	18	17.00	taobao.com	shopping	10	13.15	facebook.com	social	3
7.31	asos.com	shopping	360	16.81	cnn.com	news	106	7.45	sohu.com	misc.	18
6.81	blogspot.com	social	61	16.08	coccoc.com	misc.	??	7.18	xvideo.com	adult	45
6.37	baidu.com	search	4	16.00	i360mall.com	shopping	2,631	6.33	mail.ru	misc.	51
5.82	news.gmw.cn	news	136	15.52	pornhub.com	adult	14	6.06	baidu.com	search	4
5.49	coccoc.com	misc.	??	15.25	hao123.com	misc.	86	5.04	craigslist.org	misc.	128
5.08	xhamster.com	adult	62	14.91	alibaba.com	shopping	126	4.99	youtube.com	video	2
4.99	youku.com	video	283	14.57	tmall.com	shopping	14	4.73	news.gmw.com	news	136
4.99	i360mall.com	shopping	2,631	14.35	blogspot.com	social	61	4.59	cnn.com	news	106

5.3. Response mime types

The distribution of the observed MIME-Types of adware and PUP communication—that did *not* contain privacy related information—is shown in Figure 6 (bar chart) along with the sizes of the responses (according to the *Content-Length* HTTP header field) and the share how often these sizes were observed (violin plot). One can see that adware predominately loads JavaScript code (e. g., third party libraries), HTML code(e. g., websites displayed in an <*iframe*> tag), or other textual content (e. g., JSON objects). We used simple heuristics (e. g., we checked if the text starts with a <*html*> tag) to determine if the textual content contains script or HTML code and counted it towards the respective category, if necessary. If it comes to HTML code, we measured that the content is either (almost) zero (e. g., an empty frame) or between 1kb and 100kb big.

In contrast, extensions and PUPs only load very little textual content, but excessively load new style sheets or fonts. Furthermore, it seems that PUPS and extensions inject less visible content into websites (i. e., images and HTML objects). This indicates that PUPs prioritizes user tracking over content injection (e. g., ad-injection)

6. Discussion

In the following, we discuss ethical considerations and limitations of our work.

6.1. Ethical Considerations

Running live malware samples always comes with some ethical issues. On the one hand, one wants to understand how malware works in a realistic environment but on the other hand, running malware might result in harming individuals not involved in the analysis process (e. g., via credit card fraud). Since we run malware that generates revenue by displaying ads and stealing private information we eventually created some income for the malware authors during our analysis. We implemented measures to decrease the potential harm a sample can cause (e. g., by limiting the upload bandwidth to minimize their participation in a possible DDoS attack).

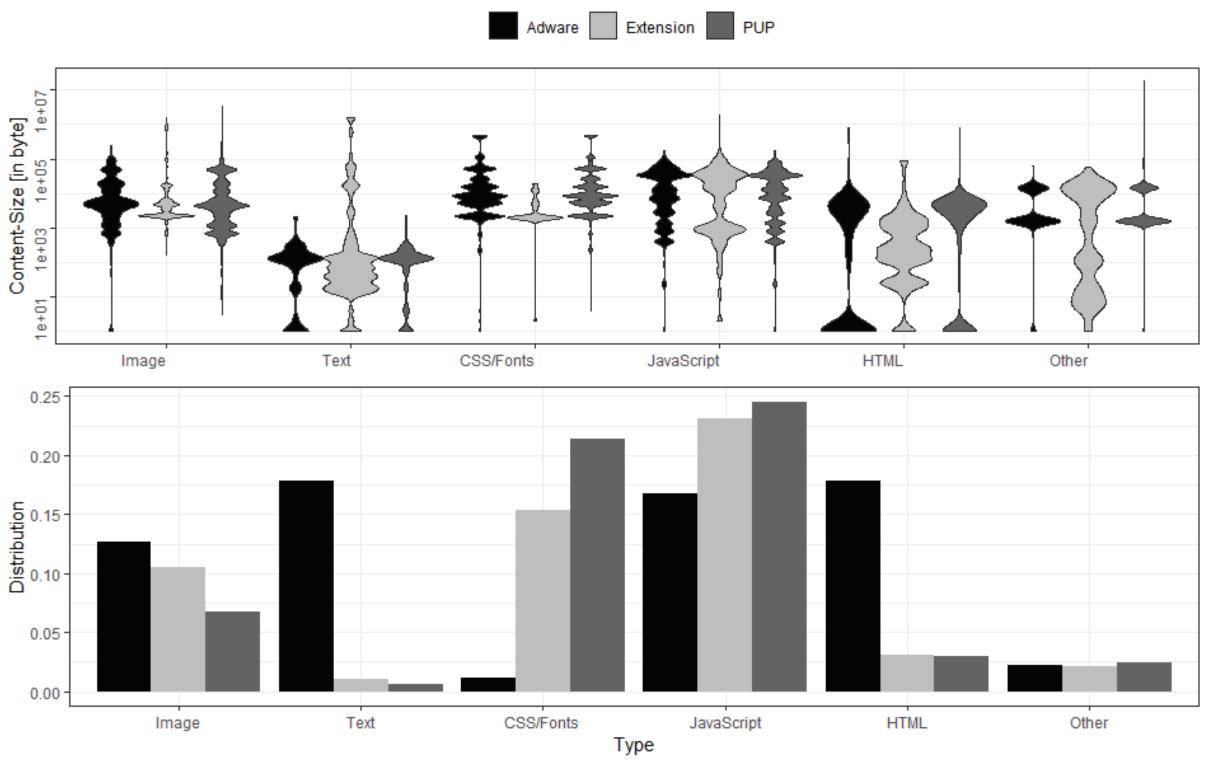


Figure 6. Amount of observed response MIME types of requests (bar charts) that did not contain private information and sizes of the responses with the corresponding distribution (violin plot).

To do so, we block some well-known ports which are not necessary for our analysis (e. g., the IMAP port 143). Furthermore, we limit the upload bandwidth of each analysis slaves which will decrease their participation in a possible DDoS attack. Of course, these measures will not prevent all possible attacks, which is probably impossible. We did not see any indications, based on network traffic statistics, that our analysis bots took part in a DDoS attack. During the course of our analysis, we only received one security alert when one malware sample scanned our internal firewall.

6.2. Limitations

Our developed framework allows the dynamic analysis of software that tampers with the users' browser session. However, it comes, like most dynamic approaches, with some limitations. Using a predefined set of websites leaves the risk that the analyzed software does not get active on the visited websites (e. g., banking-malware might only get active on specific subsites of a particular banking site). However, previous work has shown that the top-ranked pages trigger a lot of malware samples and extensions [1, 2, 13, 27]. Also, some samples might only inject content into websites only if certain search words appear, as shown in [1]. Since we use a predefined set of websites and therefore predefined keywords, we will not see injections related to other keywords.

Currently, our analysis slaves do not interact with the websites in a way a real user might (e. g., scrolling, or clicking on links). Some malware samples might only trigger if an event occurs, if the user interacts with a website we missed this kind of behaviour.

Since we are using a virtual environment to execute the malware, some samples might recognize that they are being analyzed. We took several measures to hide that the malware is executed on a virtual machine (e. g., changing CPU information and some registry keys). However, a malware sample might still detect that it is being analyzed and show a different behavior.

6.3. Future Work

In this work we only focused on two types of malicious software, adware and PUPs. Future work should measure how other types of malware collect and use personal data. For example, there are reports on ransomware that threatens users to publish personal data unless they pay the ransom [34].

Our work only considers desktop applications (i. e., the browser). However, as more and more users use mobile and IoT devices, the privacy implications of malware tailored for such devices should be addressed in future work.

7. Conclusion

Our results show that not only websites and browser extensions but also—on a massive scale—adware and PUPs negatively impact the user's privacy. We analyzed over 16,000 adware and PUP samples towards their privacy implications to the user. Our results illustrate that these kinds of software excessively leak private data (e. g., IP addresses or clickstream data). More than 37% of all requests issued by malware or PUPs is used for one of these two purposes. Adware and PUPs mainly focus on the user's clickstream which holds sensitive personal information and may give great detail of the user's life ranging from e. g., habits, personal preferences to political views. Thus, adware is a not negligible threat to the user's privacy especially because the leakage happens without consent or knowledge of the user. Regarding the tracking behavior PUPs and adware are quite similar and, since they heavily focus on the users' clickstream, pose a far worse threat to the users' privacy than extensions do.

We could show that while there are—regarding the privacy influence—similarities between extensions and adware/PUPs there are also apparent differences. Adware and PUPs mainly focus on the users' clickstream and can, therefore, create comprehensive profiles of users' which are valuable to different companies (e. g., ad-networks). Furthermore, our results show that adware and PUPs do not adopt state of the art tracking techniques.

Acknowledgments

This work was partially supported by the Ministry of Culture and Science of the German State of North Rhine-Westphalia (MKW grant 005-1703-0021 "MEwM") and partially supported by the German Federal Ministry of Education and Research (BMBF grants 16KIS0395 "secUnity" and 01IS14009B "BD-Sec"). We would like to thank the anonymous reviewers for their valuable feedback.

Appendix A. Set of Websites

The websites used in our analysis are listed in Table 6. We used the Alexa top 100 as the basis for the set which is described in detail in Section 4.2.

The set consists of ten search engines, 20 social media sites, 11 online-shops, 5 domains hosting adult content, and 16 domains that do not fit in any of these categories (e. g., *github.com* or *cnn.com*). 34 of the domains are hosted in the USA, 14 are hosted in the China, four in Russia, three in the Netherlands, two in Ireland, and five sites are hosted in different countries in Asia (ROK, SVR, JPN, HKG, and TWN).

search.rakuten.co.jp	instagram.com	coccocqc.com
movie.youku.com	search.naver.com	forums.craigslist.org
www.baidu.com	everysinglewordspoken.tumblr.com	www.ebay.com
health.china.com.cn	foodwishes.blogspot.com	coccoc.com
www.flipkart.com	edition.cnn.com	news.xinhuanet.com
www.zalando.de	www.google.com	hyperboleandahalf.blogspot.com
channel.pixnet.net	www.youtube.com	history.gmw.cn
vk.com	www.bing.com	sd.360.cn
marketplace.asos.com	stock.sohu.com	2kindsofpeople.tumblr.com
imgur.com	github.com	www.xvideos.com
zy.youku.com	xhamster.com	www.pixnet.net
military.china.com.cn	news.gmw.cn	www.alibaba.com
finance.qq.com	en.bongacams.com	world.taobao.com
mall.360.com	stackoverflow.com	www.microsoftstore.com
www.reddit.com	www.asos.com	bbs.tianya.cn
www.twitch.tv	www.so.com	www.apple.com
world.tmall.com	en.wikipedia.org	news.mail.ru
www.quora.com	www.aliexpress.com	news.youth.cn
ok.ru	news.naver.com	www.xinhuanet.com
www.groupon.com	www.sogou.com	auto.mail.ru
www.pornhub.com	www.facebook.com	twitter.com
yandex.ru	cbachina.sports.sohu.com	www.msn.com
www.linkedin.com	www.amazon.com	de.pinterest.com
newyork.craigslist.org	intl.target.com	www.imdb.com
ent.qq.com	www.hao123.com	www.microsoft.com
www.walmart.com	v.youth.cn	

Table 6
Set of websites used in our analysis.

Appendix B. Recorded Communication

Listing 1 is an example of a request and response pair captured by our framework. The information is saved in JSON format to simplify the evaluation. We record the HTTP headers, HTTP method, HTTP body, response status, cookies, measure the size of an image (if possible), etc. If the response contained text (in the shown example it includes an image), it would be recorded as well.

Listing 1: Example of a captured request and response

```
{
   "method":"GET",
   "status":"200 - OK",
   "RequestHeader":[
      "Host=log.mmstat.com",
      "User-Agent=Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko",
      "Accept=*/*",
      "Accept-Language=de,en-US;q=0.7,en;q=0.3",
      "Accept-Encoding=gzip, deflate, br",
      "Referer={https://login.tmall.com/?from=sm&redirectURL=https\%3A\%2F\%2Fsec.taobao.com\%2Fquery.htm\%3Faction\%3
           DQueryAction\%26event_submit_do_login\%3Dok\%26smApp\%3Dmalldetail\%26smPolicy\%3Dmalldetail-DetailForPc-
           anti_Spider-html-checklogin\%26smCharset\%3DGBK\%26smTag\%3DMTk0Ljk0LjEyNy43LCxlMTc2ZDkyNTgwMDI0NTdiYjQ2MzRhOGM5M2
           RkZDMwZg\%253D\%253D\%26smReturn\%3Dhttps\%253A\%252F\%252Fworld.tmall.com\%252Fworld\%252Fitem.htm\%253FtoSite\%2
           53Dmain\%2526id\%253D44202283370\%26smSign\%3DdtyXfYNAVfLWRNo1l8PHKQ\%253D\%253D}"',
      "Cookie=cna=LjjmEUd/iyICAcJefwdWYbAZ",
      "Connection=keep-alive"
   ],
   "ResponseHeader":[
      "Server=nginx",
      "Date=Mon, 11 Sep 2017 09:28:43 GMT",
      "Content-Type=image/gif",
      "Content-Length=43",
      "P3P=CP=\"NOI DSP COR CURa ADMa DEVa PSAa PSDa OUR IND UNI PUR NAV\"",
      "Set-Cookie=sca=fc61cff6; path=/; domain=.mmstat.com\natpsida=cc742f3dd5117b728a78efcc_1505122123_1; path=/; domain=.
           mmstat.com",
      "Expires=Thu, 01 Jan 1970 00:00:01 GMT",
      "Cache-Control=no-cache",
      "Pragma=no-cache",
      "X-Firefox-Spdy=h2"
   ],
   "imgSize":1,
   "timestamp":"2017-09-11T09:28:43.222Z",
   "origin":"{https://login.tmall.com/?from=sm&redirectURL=https\%3A\%2F\%2Fsec.taobao.com\%2Fquery.htm\%3Faction\%3
        DQueryAction\%26event_submit_do_login\%3Dok\%26smApp\%3Dmalldetail\%26smPolicy\%3Dmalldetail-DetailForPc-anti_Spider-
        html-checklogin\%26smCharset\%3DGBK\%26smTag\%3DMTk0Ljk0LjEyNy43LCxlMTc2ZDkyNTgwMDI0NTdiYjQ2MzRhOGM5M2RkZDMwZg\%253D
        \%253D\%26smReturn\%3Dhttps\%253A\%252F\%252Fworld.tmall.com\%252Fworld\%252Fitem.htm\%253FtoSite\%253Dmain\%2526id\%
        253D44202283370\%26smSign\%3DdtyXfYNAVfLWRNo1l8PHKQ\%253D\%253D}"',
   "url":"{https://log.mmstat.com/v.gif?logtype=1&title=\%u7406\%u60F3\%u751F\%u6D3B\%u4E0A\%u5929\%u732B&pre=https\%3A\%2F\%2
        Flogin.tmall.com\%2F\%3Ffrom\%3Dsm\%26redirect_url\%3Dhttps\%253A\%252F\%252Fsec.taobao.com\%252Fquery.htm\%253
        Faction\%253DQueryAction\%2526event_submit_do_login\%253Dok\%2526smApp\%253Dmalldetail\%2526smPolicy\%253Dmalldetail-
        DetailForPc-anti_Spider-html-checklogin\%2526smCharset\%253DGBK\%2526smTag\%253DMTk0Ljk0LjEyNy43LCxlMTc2ZDkyNTgwMDI0
        NTdiYjQ2MzRhOGM5M2RkZDMwZg\%25253D\%25253D\%2526smReturn\%253Dhttps\%25253A\%25252F\%25252Fworld.tmall.com\%25252
        Fworld\%25252Fitem.htm\%25253FtoSite\%25253Dmain\%252526id\%25253D44202283370\%2526smSign\%253DdtyXfYNAVfLWRNo1l8PHKQ
        \%25253D\%25253D&cache=ea81a81&scr=1024x768&cna=LjjmEUd/iyICAcJefwdWYbAZ&spm-cnt=a2240.7829288.0.0.4ea31a57jWYhdi&
        category=&uidaplus=&aplus&yunid=&&asid=AQAAAABJV7ZZ+ZwOegAAAADT/QfErCW0qg==&p=1&o=win10&b=ie11&s=1024x768&w=trident&
        ism=pc&lver=7.6.7&jsver=aplus_std&tag=1&stag=-1&ltag=-1}",
   "httpGetData":[
      "logtype=1",
      "title=\%u7406\%u60F3\%u751F\%u6D3B\%u4E0A\%u5929\%u732B",
      "pre={https\%3A\%2F\%2Flogin.tmall.com\%2F\%3Ffrom\%3Dsm\%26redirect_url\%3Dhttps\%253A\%252F\%252Fsec.taobao.com\%252
           Fquery.htm\%253Faction\%253DQueryAction\%2526event_submit_do_login\%253Dok\%2526smApp\%253Dmalldetail\%2526
           smPolicy\%253Dmalldetail-DetailForPc-anti_Spider-html-checklogin\%2526smCharset\%253DGBK\%2526smTag\%253DMTk0Ljk0
           LjEyNy43LCxlMTc2ZDkyNTgwMDI0NTdiYjQ2MzRhOGM5M2RkZDMwZg\%25253D\%25253D\%2526smReturn\%253Dhttps\%25253A\%25252F\%2
           5252Fworld.tmall.com\%25252Fworld\%25252Fitem.htm\%25253FtoSite\%25253Dmain\%252526id\%25253D44202283370\%2526
           smSign\%253DdtyXfYNAVfLWRNo1l8PHKQ\%25253D\%25253D}",
      "cache=ea81a81",
      "scr=1024x768",
      "cna=LjjmEUd/iyICAcJefwdWYbAZ",
      "spm-cnt=a2240.7829288.0.0.4ea31a57jWYhdi",
      "category=",
      "uidaplus=",
      "aplus",
      "yunid=",
      "",
      "asid=AQAAAABJV7ZZ+ZwOegAAAADT/QfErCW0qg==",
      "p=1",
      "o=win10",
      "b=ie11",
      "s=1024x768",
      "w=trident",
      "ism=pc",
      "lver=7.6.7",
      "jsver=aplus_std",
      "tag=1",
      "stag=-1",
      "ltag=-1"
   ]
}
```

References

[1] K. Thomas, E. Bursztein, C. Grier, G. Ho, N. Jagpal, A. Kapravelos, D. Mccoy, A. Nappa, V. Paxson, P. Pearce, N. Provos and M.A. Rajab, Ad Injection at Scale: Assessing Deceptive Advertisement Modifications, in: *Proceedings of the 2015 IEEE Symposium on Security and Privacy*, SP '15, IEEE Computer Society, Washington, DC, USA, 2015, pp. 151–167. ISBN 978-1-4673-6949-7. http://dx.doi.org/10.1109/SP.2015.17.

[2] O. Starov and N. Nikiforakis, Extended Tracking Powers: Measuring the Privacy Diffusion Enabled by Browser Extensions, in: *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2017, pp. 1481–1490. ISBN 978-1-4503-4913-0. https://doi.org/10.1145/3038912.3052596.

[3] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan and C. Diaz, The Web Never Forgets: Persistent Tracking Mechanisms in the Wild, in: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, ACM, New York, NY, USA, 2014, pp. 674–689. ISBN 978-1-4503-2957-6. http://doi.acm.org/10.1145/2660267.2660347.

[4] K. Boda, A.M. Földes, G.G. Gulyás and S. Imre, User Tracking on the Web via Cross-browser Fingerprinting, in: *Proceedings of the 16th Nordic Conference on Information Security Technology for Applications*, NordSec'11, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 31–46. ISBN 978-3-642-29614-7. http://dx.doi.org/10.1007/978-3-642-29615-4_4.

[5] S. Englehardt and A. Narayanan, Online Tracking: A 1-million-site Measurement and Analysis, in: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, ACM, New York, NY, USA, 2016, pp. 1388–1401. ISBN 978-1-4503-4139-4. http://doi.acm.org/10.1145/2976749.2978313.

[6] L. Olejnik, G. Acar, C. Castelluccia and C. Diaz, The Leaking Battery, in: *Revised Selected Papers of the 10th International Workshop on Data Privacy Management, and Security Assurance - Volume 9481*, Springer-Verlag New York, Inc., New York, NY, USA, 2016, pp. 254–263. ISBN 978-3-319-29882-5. http://dx.doi.org/10.1007/978-3-319-29883-2_18.

[7] K. Mowery and H. Shacham, Pixel perfect: Fingerprinting canvas in HTML5, in: *Proceedings of the Web 2.0 Security & Privacy Workshop (W2SP)*, M. Fredriksonn, ed., IEEE Computer Society, New York, NY, USA, 2012, pp. 1–12.

[8] S. Arshad, A. Kharraz and W. Robertson, Identifying Extension-Based Ad Injection via Fine-Grained Web Content Provenance, in: *Research in Attacks, Intrusions, and Defenses: 19th International Symposium, RAID 2016, Paris, France, September 19-21, 2016, Proceedings*, F. Monrose, M. Dacier, G. Blanc and J. Garcia-Alfaro, eds, Springer International Publishing, Cham, 2016, pp. 415–436. ISBN 978-3-319-45719-2. https://doi.org/10.1007/978-3-319-45719-2_19.

[9] R.E. Bucklin and C. Sismeiro, A model of web site browsing behavior estimated on clickstream data, *Journal of marketing research* **40**(3) (2003), 249–267.

[10] W.S. LLC, WOT API | WOT (Web of Trust), 2017, Accessed: 2017-10-31.

[11] P. Kotzias, L. Bilge and J. Caballero, Measuring PUP Prevalence and PUP Distribution through Pay-Per-Install Services, in: *25th USENIX Security Symposium (USENIX Security 16)*, USENIX Association, Austin, TX, 2016, pp. 739–756. ISBN 978-1-931971-32-4. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kotzias.

[12] T. Urban, D. Tatang, T. Holz and N. Pohlmann, Towards Understanding Privacy Implications of Adware and Potentially Unwanted Programs, in: *Computer Security*, J. Lopez, J. Zhou and M. Soriano, eds, Springer International Publishing, Cham, 2018, pp. 449–469. ISBN 978-3-319-99073-6.

[13] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna and V. Paxson, Hulk: Eliciting Malicious Behavior in Browser Extensions, in: *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, USENIX Association, Berkeley, CA, USA, 2014, pp. 641–654. ISBN 978-1-931971-15-7. http://dl.acm.org/citation.cfm?id=2671225.2671266.

[14] M. Weissbacher, E. Mariconti, G. Suarez-Tangil, G. Stringhini, W. Robertson and E. Kirda, Ex-Ray: Detection of History-Leaking Browser Extensions, in: *Proceedings of the 33st Annual Computer Security Applications Conference*, ACM, New York, NY, USA, 2017, pp. 1–13, ACM publishing.

[15] M. Foundation, Add-ons for Firefox, 2017, Accessed: 2017-07-05.

[16] D. Bonderud, WoT Privacy Breach: Trust Tanks as Browser Add-On Caught Selling User Data, 2017, Accessed: 2017-10-31.

[17] R.M. Smith, The web bug faq, *Nov* **11** (1999), 4.

[18] P. Eckersley, How Unique is Your Web Browser?, in: *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 1–18. ISBN 3-642-14526-4, 978-3-642-14526-1. http://dl.acm.org/citation.cfm?id=1881151.1881152.

[19] T. Hupperich, D. Maiorca, M. Kührer, T. Holz and G. Giacinto, On the Robustness of Mobile Device Fingerprinting: Can Mobile Users Escape Modern Web-Tracking Mechanisms?, in: *Proceedings of the 31st Annual Computer Security Applications Conference*, ACSAC 2015, ACM, New York, NY, USA, 2015, pp. 191–200. ISBN 978-1-4503-3682-6. http://doi.acm.org/10.1145/2818000.2818032.

[20] A. Kurtz, H. Gascon, T. Becker, K. Rieck and F.C. Freiling, Fingerprinting Mobile Devices Using Personalized Configurations, *Proceedings on Privacy Enhancing Technologies (PoPETs)* **2016**(1) (2016), 4–19. http://www.degruyter.com/view/j/popets.2016.2016.issue-1/popets-2015-0027/popets-2015-0027.xml.

[21] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens and G. Vigna, Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting, in: *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, IEEE Computer Society, Washington, DC, USA, 2013, pp. 541–555. ISBN 978-0-7695-4977-4. http://dx.doi.org/10.1109/SP.2013.43.

[22] T. Hupperich, D. Tatang, N. Wilkop and T. Holz, An Empirical Study on Online Price Differentiation, in: *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, CODASPY '18, ACM, New York, NY, USA, 2018, pp. 76–83. ISBN 978-1-4503-5632-9. http://doi.acm.org/10.1145/3176258.3176338.

[23] N. Jagpal, E. Dingle, J.-P. Gravel, P. Mavrommatis, N. Provos, M.A. Rajab and K. Thomas, Trends and Lessons from Three Years Fighting Malicious Extensions, in: *Proceedings of the 24th USENIX Conference on Security Symposium*, SEC'15, USENIX Association, Berkeley, CA, USA, 2015, pp. 579–593. ISBN 978-1-931971-232. http://dl.acm.org/citation.cfm?id=2831143.2831180.

[24] I. Alexa Internet, Top 500 Global Sites, 2017.

[25] A. Soltani, S. Canty, Q. Mayo, L. Thomas and C.J. Hoofnagle, Flash Cookies and Privacy., in: *AAAI Spring Symposium: Intelligent Information Privacy Management*, Association for the Advancement of Artificial Intelligence, Palo Alto, CA, USA, 2010, pp. 1–6. http://dblp.uni-trier.de/db/conf/aaaiss/aaaiss2010-5.html#SoltaniCMTH10.

[26] E. Parliament and the Council, Directive 2009/136/EC, 2009.

[27] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens and B. Preneel, FPDetective: Dusting the Web for Fingerprinters, in: *Proceedings of the 2013 ACM SIGSAC Conference on Computer &#38; Communications Security*, CCS '13, ACM, New York, NY, USA, 2013, pp. 1129–1140. ISBN 978-1-4503-2477-9. http://doi.acm.org/10.1145/2508859.2516674.

[28] K. Thomas, J.A.E. Crespo, R. Rasti, J.-M. Picod, C. Phillips, M.-A. Decoste, C. Sharp, F. Tirelo, A. Tofigh, M.-A. Courteau, L. Ballard, R. Shield, N. Jagpal, M.A. Rajab, P. Mavrommatis, N. Provos, E. Bursztein and D. McCoy, Investigating Commercial Pay-Per-Install and the Distribution of Unwanted Software, in: *25th USENIX Security Symposium (USENIX Security 16)*, USENIX Association, Austin, TX, 2016, pp. 721–739. ISBN 978-1-931971-32-4. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/thomas.

[29] VirusTotal, Free Online Virus, Malware and URL Scanner, 2017, Accessed: 2017-07-24.

[30] GreatFire, Blocked sites in China - Bringing Transparency To The Great Firewall Of China, 2017.

[31] G. Inc., Google Safe Browsing APIs (v4), 2017, Accessed: 2017-08-04.

[32] DomainState, Domain Tools, Stats, News, Forum and Directory, 2017, Accessed: 2017-08-09.

[33] StatCounter, GlobalStats Browser Market Share, 2017, Accessed: 2017-08-09.

[34] Check Point, Charger Malware Calls and Raises the Risk on Google Play, 2017. https://blog.checkpoint.com/2017/01/24/charger-malware/.