Rods with Laser Beams: Understanding Browser Fingerprinting on Phishing Pages

Iskander Sanchez-Rola and Leyla Bilge, Norton Research Group; Davide Balzarotti, EURECOM; Armin Buescher, Crosspoint Labs; Petros Efstathopoulos, Norton Research Group

https://www.usenix.org/conference/usenixsecurity23/presentation/sanchez-rola

This paper is included in the Proceedings of the 32nd USENIX Security Symposiument.
August 9–11, 2023 • Anaheim, CA, USA
978-1-939133-37-3

Open access to the Proceedings of the 32nd USENIX Security Symposium is sponsored by USENIX.
Rods with Laser Beams: Understanding Browser Fingerprinting on Phishing Pages

Iskander Sanchez-Rola*, Leyla Bilge*, Davide Balzarotti†, Armin Buescher§, Petros Efstatopoulos*
Norton Research Group*, EURECOM†, Crosspoint Labs§

Abstract

Phishing is one of the most common forms of social engineering attacks and is regularly used by criminals to compromise millions of accounts every year. Numerous solutions have been proposed to detect or prevent identity thefts, but phishers have responded by improving their methods and adopting more sophisticated techniques. One of the most recent advancements is the use of browser fingerprinting. In particular, fingerprinting techniques can be used as an additional piece of information that complements the stolen credentials. This is confirmed by the fact that credentials with fingerprint data are sold for higher prices in underground markets.

To understand the real extent of this phenomenon, we conducted the largest study of the phishing ecosystem in the topic by analyzing more than 1.7M recent phishing pages that emerged over the course of 21 months. In our systematic study, we performed detailed measurements to estimate the prevalence of fingerprinting techniques in phishing pages.

We found that more than one in four phishing pages adopt some form of fingerprinting. This seems an ever growing trend as the percentage of pages using these techniques steadily increased during the analysis period (last month doubling what detected in the first month).

1 Introduction

Tracking users’ online activity is a ubiquitous practice with different goals. At the core of online tracking is the desire to learn about a user’s habits, preferences, identity, and other information capable of creating a profile which can then be used to customize the user experience [61, 65]. This includes advertising and marketing, but also web site personalization, analytics services, social media sharing, and others. The effectiveness of online tracking has fueled very lucrative business models, often leading to situations where trading profiles of oblivious users—and, therefore, the potential of capturing their attention—becomes one the primary transaction instrument of the Internet [14, 36].

The intensity and pervasiveness of such online tracking practices has captured the attention of not only marketers, engineers, researchers and journalists, but also that of regulators [24, 25]. As a result, efforts are being made to contain the effects of tracking, and provide users with the ability to better control their online footprint [11, 13, 27, 66]. The increasing complexity of modern Internet-facing applications, however, presents endless opportunities for tracking methods that may be less invasive, (temporarily) more compliant, and equally effective. Browser fingerprinting is one such tracking practice, capable of uniquely identifying users with relatively high accuracy.

Calculating a browser fingerprint (henceforth referred to simply as a fingerprint) is not a new idea [3, 22, 32, 35]: for many years websites had access to information exposed by the browser, such as the browser type and version, IP address, plugins, and the list of available fonts. By combining these elements, websites can generate fingerprints for groups of users with similar configurations. With the increase in the number of available APIs [74] and in their sophistication [16, 33, 34, 35, 46, 63], these groups may be reduced to a single person, thus identifying users uniquely. This presents an increased attack surface for those using fingerprints for privacy-invasive or malicious purposes, but it has also been used for benign purposes—such as providing additional elements of (“zero-trust”-style) authentication.

Ironically, the use of fingerprinting as an additional authentication instrument has further increased the value of capturing (or stealing) users fingerprints for malicious purposes. For example, phishing attacks aim at deceiving users and stealing credentials that would provide access to monetizable services. Such credentials are sold on the dark web, for fraudulent use. Complementing a user’s stolen credentials with the corresponding browser fingerprint significantly increases the value of the stolen assets, as the additional information can allow to bypass authentication checks. This is confirmed by the existence of various marketplaces (e.g., Richlogs [10] and Genesis [45]) which specialize in selling user fingerprints. The price of this information can even reach $200 per user [28]. If we compare it with the average price of $15 for stolen credentials reported by a recent analysis [55], we can clearly see the additional cost of fingerprint informa-
The prevalence of fingerprinting, performed by legitimate websites (for authentication and tracking) resulted in a densely populated research space. Some researchers have performed large-scale measurements to understand the scale of the problem [22, 32, 35, 60, 61, 64], some devised new fingerprinting techniques to increase the precision of the unique identifiers created [16, 33, 34, 46, 63], while others have explored new areas of application such as authentication [8, 20] and automated detection of crawlers [4, 7, 72].

Only very recently, researchers looked at the adoption of fingerprinting practices on phishing websites [75, 77]. In 2021, Zhang et al. demonstrated the use of fingerprinting for cloaking purposes—i.e., the ability of phishing sites to conceal their malicious content when a fingerprint reveals that the visitor is not a real user—by performing a study on 100K phishing web-pages [77]. In 2022, Lin et al. showed instead that it would be possible for the phishers to match the multifactor authentication requirements of particular legitimate websites that incorporate fingerprinting authentication elements [75]. Motivated by these recent findings and by the discovery of stolen fingerprints in underground markets [10, 28, 45], we decided to conduct a large-scale study of the use of fingerprinting in the phishing ecosystem.

In particular, in this paper we investigate the fingerprinting practices of 1.7M phishing websites that were active between December 2020 and August 2022. To be able to capture the run-time behavior of the phishing websites, we actively queried a large phishing feed and accessed the phishing websites immediately after they appeared in the feed. Our automated analysis framework instrumented 109 fingerprinting functions, the highest number used to date in this type of study. Following the definition of the most recent survey on the topic [35], a browser fingerprint is a set of information related to a user’s device extracted from a variety of sources, from the hardware to the operating system to the browser and its configuration. More concretely, fingerprinting refers to the process of collecting such information through a web browser, with the objective of building a fingerprint of a device. In this work we will focus on scripts that invoke fingerprinting-related API calls and later harvest the data (e.g., through GET request). Furthermore, we will focus our analysis on new code included by the phishing page and not present in the original target website.

We start by conducting detailed measurements and longitudinal analysis on our data to obtain high-level statistics about the fingerprinting adoption on phishing websites. While in our global data 18.7% of the phishing sites perform fingerprinting, when we focus our analysis to phishing websites that were successful at attracting real users, the percentage increases to 24%. We then investigate the real-world impact of the phishing websites that use fingerprinting, incorporating user telemetry collected from millions of users. We found that, similar to legitimate websites, phishers often include third-party scripts, some of which belong to well-known trackers. However, it is interesting to note that the most common trackers observed on phishing websites are different from the top trackers encountered on legitimate websites. When we compared the code snippets that include fingerprinting functions used by phishing sites with the ones employed by their targets, we discovered that 91% of the phishing sites include completely new additional scripts. We perform a thorough large-scale comparison of the two and report the most and least prevalent fingerprinting functions that are used by phishers, also breaking down statistics for basic and advanced fingerprinting methods. Our results show that over time there is a clear increase in the fraction of phishing websites that adopt advanced fingerprinting techniques compared to previous studies.

We finalize our study by looking at how the set of fingerprinting functions can be used to cluster phishing sites that are part of the same campaign or that rely on the same kit. We also present two case studies, in which a post-mortem manual analysis of phishing sites that carry the patterns we selected, revealed very sophisticated phishing campaigns that target multiple institutions from multiple categories. The attackers not only perform highly advanced fingerprinting techniques, but also implement very obfuscated and sophisticated code for staying under the radar as long as possible.

2 Background and Related Work

2.1 Fingerprinting

A browser fingerprint is a unique user identifier that can be computed by using a combination of hardware and software information (more information about the computation of fingerprints in Appendix A). It is mainly used by third-party trackers in order to track users without the need to rely on traditional cookies. Previous studies [39, 71] analyzed how fingerprints evolved over time, and showed that certain dynamics could bring inaccuracies that need to be taken into account. Researchers indicated that there could be many reasons behind these changes, like user actions or environment updates. The last decade has produced a large corpus of research aimed at understanding the details of the web tracking phenomenon by measuring its size and by estimating its impact for the Internet users [22, 60, 61, 64]. Similar to the attackers-vs-defenders arms-race that exists in many areas of security, we are witnessing a complex cat-and-mouse game between trackers and privacy advocates. In fact, to be able to continue their tracking activities even when an anti-tracking solution is present on the user’s browser [11, 13, 27, 66], trackers started to leverage browsing fingerprinting techniques [3, 32, 35].

A recent work by Iqbal et al. [32] reported that more than 10% of the top 100K popular websites include trackers that
employ fingerprinting techniques. While web tracking is typically deployed to enable targeted advertisement, previous studies have shown that the high prevalence of web tracking could pose a serious threat to the users’ privacy [19]. For instance, trackers could exploit their knowledge of the users browsing history to learn more about their habits, their political preferences, and their religious beliefs. Besides tracking, browser fingerprinting techniques can also be used as part of multi-factor authentication and to enhance other aspects of web security [8, 20]. A recent study by Durey et al. [20] discovered that first parties regularly collect fingerprints during sign-up, sign-in, and payment processes and showed that fingerprinting can be very powerful in protecting users against cookie hijacking and account hijacking attacks. On the other hand, fingerprinting can also be abused by cybercriminals to uniquely identify crawlers [4, 7, 72].

2.2 Phishing

A phishing attack is a type of social engineering attack in which the attacker impersonates a trusted party to deceive a victim to reveal sensitive information, such as account details, authentication credentials, and financial information. Phishing attacks are one of the most prevalent cyber attacks that cause large financial and reputation risks [26]. Phishers target a wide range of victims – ranging from consumers to professionals, from mobile to traditional PC users – by leveraging various means, which include Web pages, emails, mobile applications, and SMS messages.

In response to the increasing number of phishing attacks, a large corpus of anti-phishing solutions have been proposed over the past two decades [17, 37, 41, 59, 69]. These solutions can be categorized into two main categories: (i) blocklist-based [29, 47, 48, 50], and (ii) allowlist-based [1, 42]. While earlier solutions [9, 12, 41] proposed phishing detection based on the analysis of URLs and domain names, more recent solutions [1, 29, 42, 47, 48, 50] automatically analyze phishing pages to identify features that are effective at distinguishing them from legitimate websites.

Regardless of the approach taken, all existing solutions need to analyze known phishing websites to build their ground truth datasets. Typically this step involves visiting known phishing websites in an automated fashion [29, 47, 48, 50], or analyzing existing phishing kits [30, 49]. To evade being detected by anti-phishing solutions, sophisticated phishing attacks incorporate evasive techniques to distinguish the visits of regular users from automated bots. If the visitor is suspected to be a security crawler, the website replace its malicious content with a benign page [47].

Phishers first started this arms-race by deploying server-side techniques [31, 38]. However, when security crawlers adopted techniques to thwart server-side approaches [31] phishers quickly responded by devising client-side techniques. These include browser fingerprinting for content manipulation, which was much more effective [4, 77]. CrawlPhish analyzed a little over 100K phishing websites collected over a period of 14 months to identify client-side evasion techniques adopted by phishers [77]. By looking at the JavaScript code, the authors identified 1,128 different implementations that might be sourced by different actors. Acharya and Vadrevu explored the effectiveness of fingerprinting and showed that the 23 most popular security crawlers could be easily circumvented by using fingerprinting techniques [4]. In this work, we complement this information by analyzing how fingerprints calculated by phishers are used, based the actions performed after the data collection (i.e., harvesting data or manipulating content). This allows us to comprehend how malicious groups operate with respect to the usage of browser fingerprinting techniques on their websites.

2.3 Phishing & Fingerprinting

We can divide the use of fingerprinting techniques in two broad classes. In the first group we put all cases in which the website extracts information from the browser to customize the page (e.g., according to the language, the browser, or the screen resolution) or to redirect users to different targets (e.g., to better serve users from different countries). We refer to this first category as Client-side Content manipulation and Endpoint Selection (or content manipulation for short). These techniques can be used for perfectly legitimate purposes, or as a way to hide their content from certain crawlers.

The second category includes instead the cases in which the website uses fingerprinting techniques to build a unique user (or device) identifier (UUID). As discussed above, user tracking and authentication are the most common uses in this category for benign websites. In addition, phishing pages can also compute and steal user identifiers, with the goal of reusing or re-selling them along with the user credentials to bypass security checks [10, 28, 45].

In 2022, Lin et al. [75] proposed a technique to identify the set of fingerprinting functions used for two-factor authentication by a few legitimate popular websites. They also looked at the adoption of fingerprinting by around 300K phishing sites. They concluded the study by stating that phishing attacks from May 2018 to April 2021 did not collect the accurate fingerprints required for effective multi-factor authentication but that, however, this phenomenon could get more prevalent in the future. In this paper, we perform a finer-grained analysis with a focus on fingerprinting adoption at larger-scale on phishing sites. We also investigate various types of fingerprinting intentions to understand the root cause of fingerprinting in the phishing ecosystem.

3 Methodology

At the core of our study lies a comprehensive dataset of phishing websites recently detected by a commercial detection solu-
We implemented a custom Chromium-based crawler to analyze each phishing site. Following existing phishing analysis approaches [2, 5, 42], our crawler did not perform subsequent page analysis or interacted with the websites (e.g., by submitting forms). Due to this limitation, we could have missed fingerprinting activities performed by phishers only after certain actions were performed or only present in specific pages of the website. Therefore, the result shown in this work should be considered a lower bound.

To ensure we could perform our analysis before the sites were taken down by the attackers, we launched the crawler as soon as a new website was reported by the commercial phishing detection solution. As the process of detecting phishing is out of scope of the paper, we wanted to ensure that we are analyzing the exact same pages detected by the vendor that provided us with the data. Thus, our crawler took a screenshot of the page and visually compared it with the one provided by the detection system, to ensure the phishing website was still active and matched the report. We use pHash [53] to create and compare the perceptual image hashes of the phishing website and the final website reported by the commercial solution [76]. If the images did not match, we discarded the webpage for the remainder of the analysis pipeline.

As discussed in Section 2, over the last decade, a large number of fingerprinting techniques have been proposed. In order to conduct the most comprehensive fingerprinting study of the ecosystem, we collected all fingerprinting functions discussed in academic studies [35] and used by available privacy solutions (more concretely, DuckDuck Go [66], Mozilla Firefox [27], Apple WebKit [58] and Brave [13]). This resulted in 109 functions, 50 of which fall into the Basic fingerprinting category, while the rest are Advanced fingerprinting.

We also performed a comparison of our proposed solution against OpenWPM [52], a web privacy measurement framework previously used in other studies [23, 32]. In particular, we analyzed a random sample of 1,000 phishing websites using both techniques. In total, we found 728 websites calling fingerprint-related APIs. Our crawler was able to detect and log all invocations, while OpenWPM missed calls in 348 (47.8%) of the websites. The main reason behind this result lies in the fact that OpenWPM monitors a smaller set of fingerprinting APIs [51]. In this experiment 48 different types of APIs calls were detected, out of which OpenWPM supported only 36. More concretely, from those not present in OpenWPM, 5 APIs fall under the basic category (e.g., screen.availWidth and window.matchMedia), and 7 under advanced (e.g., htmlmediaelement.canPlayType and webglrenderingcontext.getExtension). Additionally, we instrumented the missing API calls in OpenWPM and rerun the analysis to test for potential false positives and negatives. After this update, both our crawler and OpenWPM were able to detect all the 2092 calls performed on the test set, showing the reliability of the solution. In the following section, we will describe how our approach is able to understand the specific intention behind the calls performed (e.g., content manipulation or harvesting the data through requests).

Our crawler is designed to collect information about each fingerprinting function (e.g., in HTMLCanvasElement and RTCPeerConnection objects), and to track APIs that can be used to modify pages dynamically (e.g., Element.ReplaceChild, and Node.insertBefore). The system also tracks the callers of each function (either first-party or third-party script), by using a custom instrumentation based on the Chrome debugging protocol (CDP) [18]. By cross-referencing a state-of-the-art tracker list [21, 44, 60], known trackers are matched with some of the third-party scripts identified to call fingerprinting functions. In fact, while phishers often employ their own custom fingerprinting scripts, in some cases they leverage third-party tracking solutions.
Moreover, our crawler monitors the network activity to gather redirections and all requests and responses performed by the browser and fingerprinting scripts, together with their parameters (i.e., main pages and frames). Finally, the system analyzes cookie creation events (i.e., `Document.cookie`) and their associated scripts to identify fingerprints that are stored as cookies.

### 3.3 Fingerprinting Intention

Our crawler can automatically associate fingerprinting functions with two intentions: (i) using fingerprints for content manipulation/redirections, and (ii) storing fingerprints in cookies or sending fingerprints within parameters of GET and POST requests. This is achieved by combining code dependency analysis and timing element factors.

The process starts by comparing the time in which different parts of the document object model (DOM) are loaded (either statically through HTTP/HTML or dynamically by using JavaScript) with the time in which the browser fingerprinting API and functions are invoked. Then, our tool inspects the stack traces of all function calls and events, and associates each call with its corresponding script. In the simplest case, the script that perform fingerprinting and the corresponding intention is the same. For example, when a number of fingerprinting functions are called followed by multiple inclusions in the DOM or by redirecting the page, we can flag it as content manipulation. Similarly, if the script calls a set of fingerprinting functions and after it stores data in a cookie or it makes a request (GET/POST) with parameters, we mark it as a case of fingerprinting. In more complex scenarios, however, the scripts responsible for fingerprinting are not the same as the one implementing the corresponding intention. For such cases, we employ a methodology to associate the fingerprinting script with its intention script. Our crawler checks the page hierarchy by using a dynamically generated resource tree in order to find a connection between the two scripts. In particular, our analysis looks for parent-child or shared-parent relationships. In Sections 5 and 7, we will analyze and discuss both content manipulation and fingerprinting, showcasing these scenarios and their different types in more detail.

### 3.4 Target Websites Analysis

In our paper, we do not only aim at measuring the extent of browser fingerprinting adoption among phishing websites, but also at shedding light into the main motives of the attackers (i.e., content manipulation or fingerprinting). Therefore, once the aforementioned intentions are identified on phishing websites, we look at the target legitimate page to understand whether the phishers implement the same fingerprinting techniques that exists on the original target page or whether they collect more information and share it with other parties. To perform a more accurate comparison, we first need to identify the exact set of fingerprint functions used by each page targeted by the phishing sites in our dataset. Unfortunately, the information we obtained for each phishing page included only the name of the targeted company and not the particular login page that was impersonated. Therefore, we proceeded to manually locate the corresponding login pages.

Then, to record the fingerprinting functions invoked during a login process, we implemented a browser extension for Chromium following the same logic explained earlier for the crawler, but this time inside an extension. This allowed us to record everything that happened dynamically in the browser. We then manually accessed the original login pages and provided dummy data to trigger the login process, while our extension collected all the fingerprinting functions invoked until the form submission, together with information about which party triggered the event: first or the third party (e.g., known trackers, unknown third-parties). This full process took roughly a week of work for two researchers.

### 4 The General Picture

During the 21 months of data collection, our system analyzed 1,709,810 active phishing sites. Of these, 18.7% (319,922) invoked at least one fingerprinting function on our list. Among those, 82.2% call three or more different functions. This finding shows that fingerprinting is a much more prevalent behavior on phishing websites than on legitimate websites. In fact, according to a very recent study [32], only 10% of the top 100K websites include fingerprinting scripts.

Recent studies found a higher percentage of phishing sites that perform fingerprinting. However, they used different (and much smaller) datasets which might not have included the long tail of less popular sites [40]. In fact, if we restrict our analysis to the most popular phishing pages – computed according to the number of victims in the telemetry – we observe a different distribution. For instance, the percentage of those using fingerprinting APIs increases to 44.5% if we consider the top 50K phishing pages. These results are in line with
what was previously reported in other studies [75]. We will present a deeper analysis on the real-world impact of these results later in this section.

In Figure 1, we plot the percentage of phishing websites that perform fingerprinting across the length of our study, grouped on a monthly basis. Over the past two years, we observe a dramatic increase on the number of phishing sites that adopt browser fingerprinting techniques, from around 12% at the beginning of our study to over 26.6% on the last months. This emphasizes the ever-growing interest of both phishers and legitimate websites in using fingerprinting functions.

Following the methodology described in Section 3, we are able to distinguish between two reasons for the usage of fingerprinting functions: 104,279 (32.6%) of the phishing websites use it for content manipulation, and 101,030 (31.6%) creates fingerprints and shares them through POST/GET requests or stored in cookies. Finally, 114,613 (35.8%) use fingerprinting for both intentions.

![Figure 2: CDF, functions per website (by device type).](image)

**Summary**: The fraction of phishing websites that use fingerprinting has more than doubled over the past two years. Today, one in four phishing sites invokes fingerprinting APIs, with more than 80% combining over 3 functions.

### 4.1 Real-World Impact

As previously mentioned, we leveraged telemetry data to measure the number of users who encountered the phishing websites in our dataset. In total, 307,883 (22.2%) of the phishing websites were accessed by at least one user. Note that, due to privacy reasons, user telemetry did not contain unique user identifiers, but only the country of the user. Therefore, we cannot provide statistics about the number of users who encountered phishing attacks. Out of the phishing pages accessed by the users in our telemetry, 24% perform fingerprinting. If we combine this information with the fact that 18.7% of the phishing pages in our dataset invoked fingerprinting APIs, we conclude that pages that perform fingerprinting are also more successful at attracting victims.

The telemetry also contained the type of device used to access the phishing sites. This additional information allowed us to categorize the phishing websites into three categories: those that targeted only mobile users, those that targeted only PCs, and those visited by both. In particular, we found that 5,894 (8%) of the phishing websites were encountered only in mobile phones. The vast majority (60.8%) of the phishing sites were visited instead only by PC users. Finally, 31.2% of phishing websites had victims using both mobile and PC platforms. Figure 2 shows the distribution in the number of fingerprinting functions for the three aforementioned categories (mean values are 5.9, 8.2 and 6.3 respectively for mobile, PC, and mixed). Again, these results could support the hypothesis that phishers who deploy more sophisticated phishing sites and adopt browser fingerprinting techniques are also more successful in attracting their victims, or vice versa. More concretely, desktop users are fingerprinted with more functions than mobile ones (2.4 more functions on average).

Finally, we divide phishing pages in two categories: country-specific phishers and international phishers. As the name suggests, the country-specific phishers (73.9% of phishing websites) target users from only one specific country while the international ones (26.1%) target users from multiple countries. We observe a higher percentage of fingerprinters in the international category (36.6%) than in the country-specific category (22.1%). Similar to what happens with phishers targeting mixed platforms, those that attract wider range of victim seen to implement fingerprinting functions more frequently.

**Summary**: Phishing websites that are more successful at attracting users, that target more devices, or whose victims span more countries, tend also to be the ones that use fingerprinting functions the most.

### 5 Additional Fingerprinting on Phishing Sites

Our system keeps track of the parties who initiate the scripts that call fingerprinting functions. The actors who are involved conclude that pages that perform fingerprinting are also more successful at attracting victims.

The telemetry also contained the type of device used to access the phishing sites. This additional information allowed us to categorize the phishing websites into three categories: those that targeted only mobile users, those that targeted only PCs, and those visited by both. In particular, we found that 5,894 (8%) of the phishing websites were encountered only in mobile phones. The vast majority (60.8%) of the phishing sites were visited instead only by PC users. Finally, 31.2% of phishing websites had victims using both mobile and PC platforms. Figure 2 shows the distribution in the number of fingerprinting functions for the three aforementioned categories (mean values are 5.9, 8.2 and 6.3 respectively for mobile, PC, and mixed). Again, these results could support the hypothesis that phishers who deploy more sophisticated phishing sites and adopt browser fingerprinting techniques are also more successful in attracting their victims, or vice versa. More concretely, desktop users are fingerprinted with more functions than mobile ones (2.4 more functions on average).

Finally, we divide phishing pages in two categories: country-specific phishers and international phishers. As the name suggests, the country-specific phishers (73.9% of phishing websites) target users from only one specific country while the international ones (26.1%) target users from multiple countries. We observe a higher percentage of fingerprinters in the international category (36.6%) than in the country-specific category (22.1%). Similar to what happens with phishers targeting mixed platforms, those that attract wider range of victim seen to implement fingerprinting functions more frequently.

**Summary**: Phishing websites that are more successful at attracting users, that target more devices, or whose victims span more countries, tend also to be the ones that use fingerprinting functions the most.

---

1 Because the collection of country information in the shared telemetry started after we begun our study, we can provide results for only 77.6% (238,958) of the phishing websites.
might be the first-party, third-party trackers, or other third-parties. We will now focus on cases where phishers create fingerprints and share them through POST/GET requests or store them in cookies. In Table 1, we provide the breakdown for the actors on the phishing pages, cross-checking their existence on the target page as well (note that a given website can include multiple actors). On 53.7% of the phishing pages, fingerprinting content is sent directly by the first-party, i.e., by the phisher herself.

Interestingly, 77.8% of the websites which use third parties for fingerprinting do not copy them from the legitimate website they impersonate (in other words, the original site did not include that same third-party script). This indicates that the phishers collect additional information that might not necessarily be only for matching the authentication credentials of the user. 74.2% third-parties sharing data in phishing sites are known to be trackers. Overall, 91.4% (197,177) of the phishing websites send the output of the fingerprints by using new code (from first parties or additional third parties) that does not exist on the target page.

Figure 3 shows the top trackers identified performing fingerprinting-based tracking in phishing sites and their targets. While the top trackers observed on legitimate websites [19, 60] are present in both cases, we also observe less popular entries. For example, Histats, Kampyle, and 5ILA, (just to name a few) are not among the common trackers observed in legitimate websites, due to their low prevalence. This shows that phishers tend to use different trackers than legitimate pages.

By performing the same analysis on content manipulation cases, we found very similar patterns. On 66.8% of the phishing pages, this behavior is performed by scripts from the site itself (i.e., first party). In a similar manner, 73.5% of the third parties are newly included, and were not present in the target site the phishing page is mimicking. Of all phishing websites that perform content manipulation calling fingerprinting functions, 89.6% (196,010) use code that is directly included by the phishing page itself or third parties not present in the original targeted website.

Summary: Over 90% of the phishing websites that send or store fingerprints, compute them from newly included code (not present in their target site).

5.1 Fingerprinting Functions

We now look at the fingerprinting functions that are used by the phishing sites in our dataset. We will focus on new fingerprinting code, as this is directly performed by the phishers, and not simply inherited from the original website (e.g., due to cut-and-paste) when developing the phishing page. Overall, we find 75 distinct fingerprinting functions in the code of phishing sites, 43 of which are basic and 32 advanced.

In total, our system identified 890,795 fingerprinting function calls throughout the data collection period. 65.9% of them were triggered by first-party scripts. 24.1% of the calls were instead performed by known tracking domains that did not exist on the impersonated original webpages. The remaining 10% of the functions were called by other third-parties included by the phisher.

Figure 4 shows the distribution of the number of fingerprinting functions invoked by the phishing websites. The graph breaks down the number of functions called by first-party, tracker and third-party scripts. The average number of fingerprinting functions we observed in phishing pages is 6.3, 5.4 and 5.7 respectively. 10.2% of these phishing websites use advanced fingerprinting functions. There is a clear increase in the fraction of phishing websites that adopt advanced fingerprinting techniques compared to the previous study which analyzed phishing sites from 2018 (7%) [75]. Note that here we report statistics about scripts that are not in the target websites. If we include all cases, the number goes up to 24.6%.

In Tables 2 and 3, we list the basic and complex fingerprinting functions most frequently used by phishers. 78.2% of the phishing websites collect the UserAgent, 22.5% (44,445) the
Table 2: Top Basic Fingerprinting Functions

<table>
<thead>
<tr>
<th>Fingerprinting Function</th>
<th>Websites</th>
</tr>
</thead>
<tbody>
<tr>
<td>navigator.userAgent</td>
<td>154205</td>
</tr>
<tr>
<td>screen.width</td>
<td>44445</td>
</tr>
<tr>
<td>screen.height</td>
<td>44072</td>
</tr>
<tr>
<td>navigator.appVersion</td>
<td>40528</td>
</tr>
<tr>
<td>date.getTimezoneOffset</td>
<td>40076</td>
</tr>
<tr>
<td>navigator.plugins</td>
<td>30864</td>
</tr>
<tr>
<td>screen.colorDepth</td>
<td>28894</td>
</tr>
<tr>
<td>navigator.appName</td>
<td>28120</td>
</tr>
<tr>
<td>navigator.language</td>
<td>27010</td>
</tr>
<tr>
<td>window.devicePixelRatio</td>
<td>25917</td>
</tr>
<tr>
<td>navigator.cookieEnabled</td>
<td>21427</td>
</tr>
<tr>
<td>navigator.javaEnabled</td>
<td>20780</td>
</tr>
<tr>
<td>navigator.platform</td>
<td>20521</td>
</tr>
<tr>
<td>navigator.vendor</td>
<td>17591</td>
</tr>
<tr>
<td>navigator.product</td>
<td>16067</td>
</tr>
</tbody>
</table>

Table 3: Top Advanced Fingerprinting Functions

<table>
<thead>
<tr>
<th>Fingerprinting Function</th>
<th>Websites</th>
</tr>
</thead>
<tbody>
<tr>
<td>document.createElement(&quot;canvas&quot;)</td>
<td>12014</td>
</tr>
<tr>
<td>htmlcanvaselement.toDataURL</td>
<td>7943</td>
</tr>
<tr>
<td>webglrendercontext.getParameter</td>
<td>3642</td>
</tr>
<tr>
<td>webglrendercontext.getExtension</td>
<td>3189</td>
</tr>
<tr>
<td>intl.datetimeformat.resolvedOptions</td>
<td>2252</td>
</tr>
<tr>
<td>htmlmediaelement.canPlayType</td>
<td>943</td>
</tr>
<tr>
<td>element.getClients</td>
<td>678</td>
</tr>
<tr>
<td>RTCPeerConnection</td>
<td>520</td>
</tr>
<tr>
<td>audiobuffer.getChannelData</td>
<td>511</td>
</tr>
<tr>
<td>OfflineAudioContext</td>
<td>506</td>
</tr>
<tr>
<td>canvarendercontext2d.measureText</td>
<td>480</td>
</tr>
<tr>
<td>animation.currentTime</td>
<td>467</td>
</tr>
<tr>
<td>Notification.permission</td>
<td>269</td>
</tr>
<tr>
<td>rtcpereconnectionnoncevent.candidate</td>
<td>258</td>
</tr>
<tr>
<td>console.memory</td>
<td>238</td>
</tr>
</tbody>
</table>

screen width and 20.5% (40,528) accessed the appVersion. These are among the most common basic fingerprinting techniques. If we consider more advanced forms of fingerprinting, such as using the canvas and webGL APIs (6% and 2% respectively), and compare it with previous studies [75], we observe a similar percentage for canvas, but we observe a large increase in the usage of webGL fingerprinting (which was previously reported as less than 1%). Among the techniques that are less popular in our data, we can observe a long tail of phishing sites that adopt advanced techniques such as audio fingerprinting and font fingerprinting.

Summary: Out of the nearly 900k fingerprinting call we analyzed, the vast majority originated from first party scripts or newly included trackers. Phishers use a broad range of APIs, both basic (e.g., almost 80% retrieve the user agent) and advanced (e.g., 6% use canvas).

5.2 Target Brand Analysis

The phishing websites in our dataset impersonate the login pages of 500 websites from 27 categories and 64 countries. Table 4 lists the categories that are targeted the most by phishing websites that include additional fingerprinting code. In the last column, we report the number of phishing sites that called more than 3 distinct fingerprinting functions. On average, 73% of the sites in these categories incorporated at least 3 fingerprinting functions, which shows an alignment with unique identifiers generation activities [57]. Financial Services, Social Networking and TechnologyInternet are the most targeted categories with between half and three-quarters of them adopting complex fingerprinting. Around 90% of the phishers call multiple functions to create fingerprints when targeting companies from Games and Online Chats, but less when targeting Shopping and Government/Legal websites, as their legitimate counterparts include less too.

Table 5 presents the country-based breakdown of the targets. The most complicated fingerprints are created for companies in countries that do not get targeted as much. For instance, all phishing sites targeting websites in Hungary incorporate fingerprinting functions. Finally, in Table 6, we list the top 10 impersonated websites. As it can be seen, the phishing websites that target more popular websites put more effort to create richer fingerprints, as on average 81.6% of them use more than 3 functions. Again, since some targets (such as USPS) invokes less fingerprinting APIs on their website, their phishing counterparts also include less fingerprinting functions.

Summary: On average, over 80% of highly targeted brands, categories or countries, implement more than 3 fingerprinting functions, as their benign counterparts do.

6 Comparing Phishing with Their Targets

In the previous section, we have shown that fingerprinting is a prevalent phenomenon on phishing sites and the number of phishing websites that fingerprint their victims has been increasing over the last couple of years. We also found that various phishing websites perform different levels of fingerprinting: while the majority mainly rely on basic fingerprinting functions, a significant fraction in our dataset (i.e., 24.6%) adopt more sophisticated techniques such as canvas fingerprinting, webGL fingerprinting and audio fingerprinting – considered the most advanced techniques leveraged today by trackers [35].

Malicious actors are already selling stolen fingerprints on the dark web [45]. However, there is no way to know for sure whether this is the case also for the ones we detected in our
In the previous section, we found that 91% phishing sites include new scripts with fingerprinting activity that is not observed on the original impersonated websites. Here, we perform deeper investigation on this phenomenon and provide more detailed comparison results between the phishing sites and their targets.

Unfortunately, it is not possible to tell whether phishers use these identifiers for authentication [10, 28, 45], or for tracking purposes. A possible way to indirectly answer this question is to look at what the target websites do. For instance, Lin et al. [75] proposed a manual methodology to study the original websites and separate their fingerprinting APIs in the two categories. The authors created valid accounts on 16 victim websites and then iteratively removed one by one each of the fingerprinting functions – each time verifying whether the authentication was still working. If they received an alert, this proved that the information was used for authentication. If not, they concluded it might be simply used for tracking. Even if this methodology is very accurate, it can only be applied to the victim website and not to the phishing page. Moreover, it requires valid accounts (which are extremely difficult to get in case of financial institutions or government organizations) and therefore it cannot be scaled to analyze thousands of login pages, as the case here presented.

While we cannot be certain that all functions found on the legitimate website are used for authentication, it is still interesting to compare this set of functions with the ones adopted by their corresponding phishing sites. To make the comparison more meaningful, we remove the targets that use only a handful of functions. Following the approach followed in Section 4, we adopt the threshold of over 3 fingerprints and only focus on these targets for the rest of our analysis. After filtering, 368 target websites remained, impersonated by 218,611 different phishing pages.

At first, one might expect phishers to simply copy the exact same fingerprinting functionality used by their targets. However, this would be a poor and very time-consuming strategy. Instead, we believe a better strategy would be to deploy generic fingerprinting code snippets (or to provide them as part of phishing kits), which could be easily reused for different targets. These templates could invoke a superset of the fingerprinting functions used by legitimate websites – such that attackers can later reuse this information for authentication without having to tediously reverse engineer each target website individually. This seems to be the case for several

---

**Table 4: The top categories targeted by phishers**

<table>
<thead>
<tr>
<th>Target</th>
<th># Phishing Sites</th>
<th>&gt;= 3 FPs</th>
<th>Pct (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial Services</td>
<td>50932</td>
<td>32783</td>
<td>64.4</td>
</tr>
<tr>
<td>Social Networking</td>
<td>39773</td>
<td>28846</td>
<td>72.5</td>
</tr>
<tr>
<td>Technology/Internet</td>
<td>32322</td>
<td>21654</td>
<td>67.0</td>
</tr>
<tr>
<td>Government/Legal</td>
<td>11879</td>
<td>6342</td>
<td>53.4</td>
</tr>
<tr>
<td>Shopping</td>
<td>8966</td>
<td>3754</td>
<td>41.9</td>
</tr>
<tr>
<td>Online Chats</td>
<td>7546</td>
<td>6775</td>
<td>89.8</td>
</tr>
<tr>
<td>Business/Economy</td>
<td>5824</td>
<td>5150</td>
<td>88.4</td>
</tr>
<tr>
<td>Office Apps</td>
<td>2817</td>
<td>2467</td>
<td>87.6</td>
</tr>
<tr>
<td>Search Engines</td>
<td>2575</td>
<td>1840</td>
<td>71.5</td>
</tr>
<tr>
<td>Gaming</td>
<td>1220</td>
<td>1130</td>
<td>92.6</td>
</tr>
</tbody>
</table>

**Table 5: The top countries targeted by phishers**

<table>
<thead>
<tr>
<th>Target</th>
<th># Phishing Sites</th>
<th>&gt;= 3 FPs</th>
<th>Pct (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>118595</td>
<td>80870</td>
<td>68.2</td>
</tr>
<tr>
<td>France</td>
<td>14780</td>
<td>7342</td>
<td>49.7</td>
</tr>
<tr>
<td>Japan</td>
<td>10357</td>
<td>7361</td>
<td>71.1</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>4707</td>
<td>3563</td>
<td>75.7</td>
</tr>
<tr>
<td>Netherlands</td>
<td>4282</td>
<td>3464</td>
<td>80.9</td>
</tr>
<tr>
<td>Germany</td>
<td>1950</td>
<td>1407</td>
<td>72.2</td>
</tr>
<tr>
<td>Canada</td>
<td>1768</td>
<td>1371</td>
<td>77.5</td>
</tr>
<tr>
<td>Colombia</td>
<td>1200</td>
<td>662</td>
<td>55.2</td>
</tr>
<tr>
<td>Australia</td>
<td>1110</td>
<td>763</td>
<td>68.7</td>
</tr>
<tr>
<td>Hungary</td>
<td>943</td>
<td>941</td>
<td>99.8</td>
</tr>
</tbody>
</table>

**Table 6: The top brands targeted by phishers**

<table>
<thead>
<tr>
<th>Target</th>
<th># Phishing Sites</th>
<th>&gt;= 3 FPs</th>
<th>Pct (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>29462</td>
<td>21051</td>
<td>71.5</td>
</tr>
<tr>
<td>USPS</td>
<td>10728</td>
<td>5711</td>
<td>53.2</td>
</tr>
<tr>
<td>Instagram</td>
<td>10180</td>
<td>7702</td>
<td>75.7</td>
</tr>
<tr>
<td>Microsoft</td>
<td>9587</td>
<td>7730</td>
<td>80.6</td>
</tr>
<tr>
<td>Discord</td>
<td>7018</td>
<td>6425</td>
<td>91.6</td>
</tr>
<tr>
<td>Wells Fargo</td>
<td>3495</td>
<td>2954</td>
<td>84.5</td>
</tr>
<tr>
<td>SMBC</td>
<td>3448</td>
<td>2985</td>
<td>86.6</td>
</tr>
<tr>
<td>Orange</td>
<td>2761</td>
<td>2703</td>
<td>97.9</td>
</tr>
<tr>
<td>DHL</td>
<td>2750</td>
<td>2417</td>
<td>87.9</td>
</tr>
<tr>
<td>Citibank</td>
<td>2557</td>
<td>2209</td>
<td>86.4</td>
</tr>
</tbody>
</table>
black market tools, which are used for multiple sites and support multiple fingerprinting information [10, 28, 45].

According to our analysis, 20% (43, 581) of the phishing websites include all fingerprinting functions of the impersonated webpage. From the targets’ perspective, 29.1% of the login pages are targeted by at least one phishing page that can match all its fingerprinting functions. Even though we observe an increase in this trend, still a great fraction of the phishing sites are not able to match the fingerprinting functions used by their corresponding targets. Table 7 lists the top fingerprint functions used by the targets but not by their corresponding phishing pages. As it can be seen, a considerable number of advanced fingerprinting functions are not implemented by the phishers (marked with an asterisk in the table). One example being the fingerprint based on WebRTC and the extraction of the internal IP, which 72, 093 phishing pages did not include even if it was present in the target website. These results support our intuition about the optimal strategy from the phisher’s perspective as they indicate that a small fraction of the phishing sites put the effort to reverseengineer the targets to match their fingerprinting choices.

In further support to our initial claim about the optimal fingerprinting strategy of the phishers, we found that 65.1% (142, 384) of the phishing websites call additional fingerprinting functions that do not exist in their targets at all. On average, the numbers of additional functions included is 5.7. In general, 48.9% of them were invoked by first-party scripts, 30.9% of the times the script belonged to a known trackers, and only 20.2% from third parties. Table 8 lists the most common basic and advanced fingerprinting functions called by the phishing websites but not by their targets. Notably, 36, 200 of the phishing websites include additional canvas fingerprint functions, for example. This phenomenon can be directly related to the fact that many more target website are starting to include advanced fingerprinting functions such as canvas fingerprinting in their flows [32].

**Summary:** Nearly 30% of the targets have at least one phishing page that matches their fingerprinting functions, and 65% of the phishers invoke additional functions.

### 7 Who are the Heavy Fingerprinters?

One of the findings of our study is that some phishing pages employ a large number of fingerprinting functions, up to 31 different ones. Since the likelihood of using the exact same set is small, this information can be used to cluster phishing pages together, and potentially attribute them to the same phisher or phishing kit [30].

In an attempt to make accurate attribution, we experimentally evaluated different thresholds and decided to adopt 10 different functions (twice the amount used by the average benign websites [57]) as the threshold for identifying unique
signatures that could be attributed to the same entity. Additionally, we require at least one of these functions to belong to the advanced category and we remove cases in which all functions are invoked by third-party scripts. In our dataset, 6.5% (17, 178) of the phishing websites satisfy these requirements. Figure 5 presents the temporal evolution of these websites, showing a steady increase during all months of our analysis. This indicates that not only the general number of phishing pages incremented during the analysis period, but that this is also the case for those that can be clustered by using their fingerprinting functionalities.

To increase the precision of the signatures we generate, we look not only at the functions called, but also incorporate the information about the callee. If the callee is a third party, we include the domain name in our data structure (e.g., [canvasrenderingcontext2d.measureText, domain.com]). If on the other hand, it is the first party, we mark it as first. We then compute a cryptographic hash of the list of ordered function and callee pairs to generate the signatures. In total, we obtain 603 unique signatures present in more than one phishing page, out of which 32 are observed in phishing sites that impersonate more than one target brand. By analyzing the top fingerprinting APIs invoked in these clusters, we found a very similar trend to the one described in Section 5, with just some small changes in the order. For example, navigator.javaEnabled and Notification.permission are more common among clustered pages than in the general phishing dataset.

7.1 Understanding Clusters

One important point for phishing websites is data harvesting. The analysis of this process through the different clusters allows us to better understand how these campaigns work. In particular, we observed that scripts first retrieve the information and compute the corresponding fingerprints, and then immediately (milliseconds after, depending on the victim machine) send back the obtained data. As previously indicated, attacker can perform this action using either GET/POST requests or cookies. 81.3% (490) of the clusters harvest this information by including it inside a request. A closer look at those requests reveal that 72.7% of the cases are associated with parameters in GET requests, and the remaining 27.3% include them in the body of POST requests. Moreover, 18.7% of the clusters store the information in cookies that are attached to subsequent page requests.

We also investigated the destination of those requests. Our results show that all clusters send information to their own domains (i.e., first party). Furthermore, 36.7% (221) of the clusters also sent fingerprinting data to third parties. In this case, we found that all those third party domain were classified as trackers according to the lists indicated in Section 3, with Google being the most prominent one. These behaviors indicate that attackers do not use certain domains as hubs to collect data from different websites even if present in the same cluster, and that they mainly rely on first-party data harvesting approaches.

Table 9 presents the signatures we observe most frequently in our phishing data, along with the number of fingerprinting functions they use, the number of different targets they impersonate, and the number of website categories these target belong to. The results show that phishers indeed seem to reuse fingerprint code when they create phishing pages for multiple of their targets and they do that across multiple targets, in some cases up to 15 from 7 different categories. In the next section, we will present two case studies where we manually check the source code of the phishing pages and provide more details about the campaigns.

In Tables 10 and 11 we report the categories and companies most commonly targeted by the groups we identified. It is interesting to see that the order of the categories is quite different compared to the general category statistics provided in the previous section. For example, the Technology/Internet category surpasses the Social Networking category which was the second most prevalent. The table also lists the most targeted companies by the campaigns. The distribution is quite diverse and includes a big tech company, a crypto exchange service, a service provider, a bank, and a social network.

Table 9: Top more prevalent signatures.

<table>
<thead>
<tr>
<th>MD5 Signature</th>
<th>FPs</th>
<th>Targets</th>
<th>Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>19c0e2ae1..298838d452a4</td>
<td>31</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>4c81f5a77..6d9cabdb3945d</td>
<td>27</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>52260ccf2cc..8b9a9e97069</td>
<td>12</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>169a53833..d71bd5b2199</td>
<td>11</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>ce1708c9ee..3645ab2240f8</td>
<td>16</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 10: Top target categories in multi-target signatures.

<table>
<thead>
<tr>
<th>Target</th>
<th>Signatures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial Services</td>
<td>15</td>
</tr>
<tr>
<td>Technology/Internet</td>
<td>13</td>
</tr>
<tr>
<td>Social Networking</td>
<td>4</td>
</tr>
<tr>
<td>Search Engines</td>
<td>3</td>
</tr>
<tr>
<td>Business/Economy</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 11: Top targets in multi-target signatures.

<table>
<thead>
<tr>
<th>Target</th>
<th>Signatures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td>7</td>
</tr>
<tr>
<td>Paxful</td>
<td>7</td>
</tr>
<tr>
<td>Orange</td>
<td>3</td>
</tr>
<tr>
<td>Wells Fargo</td>
<td>3</td>
</tr>
<tr>
<td>Facebook</td>
<td>3</td>
</tr>
</tbody>
</table>

7.2 Case Studies

To obtain a deeper understanding about the fingerprinting practices of the phishers and how they reuse the fingerprinting functionalities on different targets, we perform a deeper investigation through two case studies. We chose the signature that targets the most number of targets (the first signature in Table 9) and the runner-up signature in the same table for our manual investigation.

During the crawling phase, we collected all information related to each phishing websites – including source code, HTTP headers, all scripts that were created and executed (even if they were deleted afterwards), all fingerprinting analysis results, and the page screenshot. This allowed us perform a post-mortem manual analysis for a random sample of ten phishing websites belonging to the two mentioned signatures.

Signature 1 (19c0e2a2e1..298838d452a4)

All of the phishing websites included in this campaign call exactly the same 31 fingerprinting functions, and all calls are made by the first party (the website itself). Our tool classified 88% of the fingerprinting calls for content manipulation purposes. 44% of them used advanced methods, such as WebGL fingerprinting (e.g., webglrenderingcontext.getShaderPrecisionFormat), canvas fingerprinting, and audioFingerprinting. Among the basic fingerprinting functions used by these pages, some are used to obtain detailed information about the device, such as navigator.mediaDevices and navigator.maxTouchPoints and are not very common in other phishing websites.

Some of the impersonated pages made exactly the same calls, indicating that very likely attackers included these functions to match the target webpages. Due to the high sophistication of the presented case study, we speculate that the attackers may be saving the information used for content manipulation also for fingerprinting.

This campaign target 15 companies (such as Apple, Steam, M&T Bank, Citizens Bank, Discord, Instagram, Avito, or Coinbase) in 7 different categories (Technology/Internet, Games, Financial Services, Chat, Social Networking, Cryptocurrency, and Shopping). This operation started in July 2021 and continued to create new phishing pages until the end of our data collection period. The attackers launched phishing attacks to target a specific company every month, and successive attacks were never overlapping as the attacker always waited to finish a campaign before starting a new one. For example, the Discord campaign ran only from mid-September to mid-October 2021. We also looked at the domains used by the phishers and found several cases of state-of-the-art techniques such as typosquating (e.g., website and websit), hyphenation (e.g., website and web-site) or letter replacements (e.g., website and webite), all adopted to lure the victims not only by impersonating the target webpage visually but also by creating look-alike domains.

The JavaScript in the phishing websites is heavily obfuscated. Even so, we managed to reverse-engineer their code and identify the fingerprinting activities detected by our crawler. The 10 pages we manually reversed had almost exactly the same source code, proving that they were created by the same attacker group. We were able to confirm that the majority of the functions (30 out of 34) are indeed used for content manipulation (cloaking more specifically). The usage of such a sophisticated fingerprint only for ensuring the access to the webpage was only done by real users, shows the effort put by the attackers for making their attacks as stealthy as possible. When the user accesses the website a message in English is shown to inform the user will be redirected shortly. The users’ whose browser language is configured to “ru-Ru” are shown the Russian version of the same text. It is interesting that in all targets (except those that target Avito, a Russian website that offers anything from general goods, to jobs, car or even real estate), we observe this same language behavior.

Taking into account the profile of the impersonated websites, it seems that the attacker targeted individuals that are 1) younger (e.g., Discord and Instagram), 2) Russian (e.g., Avito and “ru-Ru”), 3) interested in technology (e.g., Apple, Coinbase), 4) living in US (e.g., M&T Bank and Citizens Bank). Moreover, the attackers implemented very advanced fingerprinting techniques and registered expensive domains using state-of-the-art domain look-alike methods, hinting that the phishers are well funded and knowledgeable.

Signature 2 (4c81f54a77..6d9c9db3945d)

For our second case study, we look at the runner-up signature that also includes a large number of fingerprinting functions: 27. Unlike the previous campaign, in this case not all fingerprinting calls are performed by first-party code. Precisely, 23.68% of the functions are called by the phishing websites, and all are used for fingerprinting. Of these, 3 are canvas fin-
gerprint functions. The remaining calls are instead invoked by known trackers, such as Kliken.

This campaign targeted 3 categories: Social Networking, Technology/Internet and Search Engines, and companies such as Twitter, Orange, British Telecom, AOL, and Yahoo. We observed pages belonging to this prolific campaign for the whole analysis period of 21 months. Unlike the previous attackers, campaigns against different targets were often overlapping. For example, a British Telecom campaign lasted for more than 2 months, in the middle of which the phishers also run a one-week campaign against Yahoo. Also, for these attacks, attackers do not purchase domains that could increase their effectiveness. They used instead the free version of website creator services and their sub-domains to host the phishing websites. As a result, many of these pages were created at zero cost, with no financial overhead.

We also checked the code of the phishing websites and identified the adoption of heavy obfuscation techniques as well. We reverse-engineered the pages to confirm they were created by the same attacker (group). As we mentioned earlier, the majority of the fingerprint is sourced by trackers. When we looked at how web pages are created using the website creation service, we saw that the webpage encouraged the usage of tracking to perform analytics about the website.

The canvas fingerprinting that was made by the first party exhibits instead an interesting behavior. While in general more basic fingerprinting functions are used by this group, the canvas fingerprinting is done multiple times with different combinations to increase the precision of the created identifier. In conclusion, given the lower sophistication and the multiple overlapping campaigns against different targets, this could be the result of a group of phishers using the same phishing kit.

8 Discussion and Conclusion

While phishing and browser fingerprinting are two research areas that, independently, have been the focus of a large body of research over the years, there is very little knowledge about how intricate interconnections between the two subjects are. In this paper we try to shed some light on current browsing fingerprinting practices adopted by phishing pages. Are they a common practice? How are they implemented? Do we see an increase on fingerprinting adoption? Which API functions are primarily used and for what purpose? Can we identify fingerprinting patterns that are shared between phishing websites? Can these patterns be useful to identify distinct groups or campaigns?

The main finding of our paper is that fingerprinting is a very common phenomenon in the phishing ecosystem, and the adoption rates are constantly increasing (x2 increase during the 21 months of our analysis). Today, 26.6% of the phishing pages use browser fingerprinting functions, a percentage that increases consistently among the most successful (in terms of number of victims) phishing campaigns. We also observe a different adoption depending on the specific targets, categories, countries, or even device types. For instance, phishing pages aiming at both PC and mobile devices tend to include more fingerprinting functions, granting attackers the ability to obtain a wider range of data.

These techniques are either implemented by the phishers themselves (i.e., first party scripts) or loaded from external sources. Sometimes the code is hosted as an attacker-controlled external service, and other times it is implemented by using well known web trackers. More concretely, over 90% of the script following a fingerprinting intention, are newly included code not present in the target. In fact, the choice depends on the objective of collected data: phishers often go beyond using basic approaches, and are starting to implement advanced methods (24.6%) such as WebGL, canvas or audio fingerprinting, to refine the information needed for operations.

Next, we detail our analysis by attempting to automatically identify the reasons behind the fingerprinting performed by the phishers. It is important to point out that it is impossible to provide a definitive answer, as that would require knowing the exact actions taken by the attackers on the server-side. In fact, the exact same fingerprint can be used for authentication bypass or for general tracking, or for both purposes at the same time. Through our systematic analysis, however, we were able to make an assessment and provide possible explanations. For instance, we verified whether the victim website’s use fingerprinting functions, and compare them to the one extracted from the corresponding phishing pages. However, due to the existing limitations to identify whether a website incorporates fingerprinting as a multi-factor authentication scheme [75] and to find the exact sequence of fingerprinting functions used for it, we cannot be certain that the target websites incorporate fingerprint for authentication and therefore whether phishing websites match them intentionally to steal the credentials. Instead, we perform comparisons between the fingerprinting functions called by the target websites and the phishing websites that impersonate them, and find that 29.1% of the target sites have at least one phishing page that matches all the fingerprinting functions. Therefore, if the targets used multi-factor authentication, the phishers could generate the correct authentication credentials for logins.

Lastly, in order to investigate the possible existence of distinct groups behind advanced phishing campaigns, we computed a signature based on the list of fingerprinting functions used by each site. We were able to extract 32 unique signatures impersonating more than one target brand, matching up to 15 targets from 7 different categories. Through a manual analysis of some such cases, we discovered clusters that suggest specific attacker groups with distinct modes of operations. In summary, we performed the largest, most comprehensive study of browser fingerprinting on phishing websites, allowing us to expose many important insights that were previously difficult or impossible to understand.
Acknowledgment

This project was supported by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 101019206 (TESTABLE).

References


urlScan: Website scanner for suspicious and malicious URLs. urlscan.io, 2022.


Appendix

A. Fingerprint Computation

Early solutions [22] computed a browser fingerprint by incorporating features extracted from the HTTP headers, Javascript, and the list of installed plugins. For instance, a fingerprint can include information from the User-Agent, Content encoding, and Content language headers, the list of plugins [62, 67], and other browser configurations such as whether cookies are enabled, the timezone, the screen resolution and color depth, and the list of available fonts. Today, approaches based on this information are commonly referred as “basic fingerprinting” [35], to distinguish them from more advanced forms of fingerprinting that were proposed later by researchers [16, 33, 34, 46, 63]. For instance, canvas fingerprinting leverages the browser Canvas API, which provides methods and objects to draw and create graphics on a canvas drawing surface. Differences in the underlying graphic card, operating system, fonts, sub pixel hinting, and version of the browser make the canvas of each user unique, and therefore a great candidate for fingerprinting [46]. Mowery et. al. [43] explored instead the use of the WebGL API for fingerprinting as they discovered that differences in the processing pipeline could lead to different WebGL profiles. This motivated future work to heavily utilize WebGL to identify unique devices [16, 34]. The Web Audio API was also found to leave unique traces that could be used for fingerprinting [23]. Finally, other methods are able to very accurately distinguish machines, by detecting hardware imperfections, such as those in the internal quartz crystal clocks used in modern computers [33, 63].

In summary, in this paper we consider basic fingerprinting information those that are obtainable from general browser objects (such as screen, navigator, Document, window, and date) and advance fingerprinting anything else.

Browser fingerprinting is now such an important threat to the Internet user’s privacy that the World Wide Web Consortium (W3C) decided to test each new API for its fingerprintability before it is publicly made available [74]. In our work, we implemented techniques to identify the adoption of all state-of-the-art fingerprinting functions available to date (more than 100) and measure their prevalence on the phishing website ecosystem.