Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection

Hugo Bijmans, Tim Booij, and Anneke Schwedersky, *Netherlands Organisation for Applied Scientific Research* (*TNO*); Aria Nedgabat, *Eindhoven University of Technology;* Rolf van Wegberg, *Delft University of Technology*

https://www.usenix.org/conference/usenixsecurity21/presentation/bijmans

This paper is included in the Proceedings of the 30th USENIX Security Symposium.

August 11–13, 2021

978-1-939133-24-3

Open access to the Proceedings of the 30th USENIX Security Symposium is sponsored by USENIX.

Catching Phishers By Their Bait: Investigating the Dutch Phishing Landscape through Phishing Kit Detection

Hugo Bijmans[1], Tim Booij[1], Anneke Schwedersky[1], Aria Nedgabat[2], and Rolf van Wegberg[3]

[1]*Netherlands Organisation for Applied Scientific Research* (TNO)
[2]*Eindhoven University of Technology*
[3]*Delft University of Technology*

Abstract

Off-the-shelf, easy-to-deploy phishing kits are believed to lower the threshold for criminal entrepreneurs going phishing. That is, the practice of harvesting user credentials by tricking victims into disclosing these on fraudulent websites. But, how do these kits impact the phishing landscape? And, how often are they used? We leverage the use of TLS certificates by phishers to uncover possible Dutch phishing domains aimed at the financial sector between September 2020 and January 2021. We collect 70 different Dutch phishing kits in the underground economy, and identify 10 distinct kit families. We create unique fingerprints of these kits to measure their prevalence in the wild. With this novel method, we identify 1,363 Dutch phishing domains that deploy these phishing kits, and capture their end-to-end life cycle – from domain registration, kit deployment, to take-down. We find the median uptime of phishing domains to be just 24 hours, indicating that phishers do act fast. Our analysis of the deployed phishing kits reveals that only a small number of different kits are in use. We discover that phishers increase their luring capabilities by using decoy pages to trick victims into disclosing their credentials. In this paper, we paint a comprehensive picture of the tactics, techniques and procedures (TTP) prevalent in the Dutch phishing landscape and present public policy takeaways for anti-phishing initiatives.

1 Introduction

Phishing is a pervasive type of social engineering that harvests user credentials by tricking targets into disclosing personal or financial information – e.g., credit card details – on a fraudulent website. Deploying a phishing website has become trivial with so-called *'phishing kits'*, which can be bought, leased or even downloaded for free in the underground economy – like dark net markets [34], social media platforms or secure messaging services like Telegram [29]. A phishing kit contains full-fledged phishing websites [9], mimicking popular banks or financial service providers. Phished credentials are exfiltrated either through e-mail [47] or collected within an administrator panel. As phishing attacks are often tailored to a specific audience and country [44], understanding the impact of phishing kits on the entire landscape, should be investigated per linguistic or geographical area to create coherent insights on phishing tactics, techniques and procedures (TTP). This specific focus aligns with earlier work finding that deployed phishing kits often victimize a particular audience and target banks in a single country [17].

Given our information position in the Dutch cybercrime ecosystem, enabling us to capture the supply of phishing kits, we take phishing targeted at the Dutch financial sector as the focus of our research. The Dutch retail banking sector is very concentrated, as just three large retail banks and a few smaller ones make up the entire market [2]. More importantly, they all primarily service customers through online banking, which is therefore widespread and popular in The Netherlands [51].

Where executing a phishing attack has become quite simple, responding swiftly and adequately to this phenomenon is far from trivial. By the time phishing domains are reported to law enforcement agencies (LEA), many of them are already offline. They can be either taken down by the phishers themselves or by hosting providers, often initiated by notice-and-takedown requests by banks who's clients get phished. All of this makes phishing campaign attribution rather difficult, as the window wherein evidence can be collected closes fast. To overcome this challenge, it is essential to pro-actively detect phishing domains and get a minute-to-minute overview of the phishing landscape. Measuring the scale and operations is crucial for defining robust countermeasures and deploying them before these attacks can cause any harm. Additionally, the recent adoption of SMS and WhatsApp as a means of phishing message delivery [37] has sped up the execution of these attacks even more. Therefore, decreasing the time between the start of the attack and detection – before the arrival of the first victim – is crucial. In this paper, we present a novel, multi-stage method to detect phishing domains at scale in real time, capture their attributes and identify the presence of phishing kits.

We leverage the fact that many phishing domains are secured by TLS connections [16] and that newly issued X.509 certificates can be monitored in real time by observing Certificate Transparency Logs [19]. By continuously monitoring these logs for 'phishy' domains and subsequently crawling them, we create a dataset of potential malignant domains. By fingerprinting parts of the source code and structure of gathered phishing kits, we measure their prevalence in the wild by detecting these fingerprints on live phishing domains. We group related kits into families, analyze their deployments and gain more insights into the TTP used by these phishers.

Our analyses aims to create an overview of the impact of off-the-shelf kits on the Dutch phishing landscape and to identify commonly used TTP. In this paper, we make the following contributions:

- We present the first empirical, longitudinal measurement study of the end-to-end life cycle of Dutch phishing campaigns.
- We collect 70 different Dutch phishing kits, identify 10 different families and create unique fingerprints in order to examine the prevalence of these kits in the wild.
- We leverage the use of TLS certificates by phishers and Certificate Transparency Logs to find 1,363 confirmed Dutch phishing domains deploying these kits between September 2020 and January 2021.
- We compile a comprehensive overview of the Dutch phishing landscape including commonly used (decoy) tactics, phishing kit characteristics and preferred hosting providers.

The remainder of this paper is structured as follows: We analyze the anatomy of a phishing campaign in Section 2, explain our methodology in Section 3 and present our results in the subsequent sections. In Section 4, we discuss the results of our analysis on gathered phishing kits. In Section 5, we examine the domains used by phishers and show how phishing kits are deployed in Section 6. We benchmark and validate our methodology with external data in Section 7 and depict the end-to-end life cycle of phishing campaigns with an example in Section 8. An overview of related work on phishing measurements and phishing kit analysis is given in Section 9. Finally, we critically discuss our results and methods in Section 10, share our public policy takeaways and conclude our work in Section 11.

2 Anatomy of a phishing campaign

A successful phishing expedition is the result of many crucial steps a phisher needs to take successively. In this section, we examine common techniques to lure in victims and make them disclose their credentials. Next, we depict the complete end-to-end life cycle of a typical phishing campaign. We end this section with the scope of our work before we elaborate on our measurement methodology.

2.1 Luring in victims

The chances of successfully executing a phishing attack are highly dependent on the credibility of the phishing message – the *bait*. Therefore, phishers use a wide range of techniques and narratives to craft sophisticated phishing messages to trick victims into disclosing their credentials without thinking twice. We can analyze such techniques by utilizing the work of Robert Caldini, the author of *The Psychology of Persuasion*, who identified several principles that explain how *'mental shortcuts'* can be exploited for the persuasion of others [8]. Recent work by Van der Heijden & Allodi [49] employed Caldini's principles on phishing e-mails and have shown that *scarcity* – time is limited, so the victim should act quickly – and *consistency* – victim is already a customer of this bank, so communication is expected – are the most popular persuasion techniques among phishers.

Although the contents of e-mails or text messages are unknown when analyzing phishing websites, we were able to identify these two principles on pages included in the various phishing kits we examine in Section 4, as persuasion techniques are exemplified there. Like a request to pay additional shipping costs for postal packages (*scarcity*), an identification request for DigiD – the Dutch online identity to interact with governmental organizations (*consistency*) – or a request to return debit cards to the bank for safe destruction and renewal (both *scarcity* and *consistency*). We noticed that besides the traditional approach of demanding victims to login to their online banking account directly, attackers also deployed more subtle, multi-staged, approaches. The first two examples are part of such an approach phishers follow to improve the credibility of their attack. In such a staged approach, victims are directed towards a decoy page like one of the aforementioned examples first, as shown schematically in Figure 1. There are no user credentials harvested on this page, but the victim is directed to a page on which a variety of banks can be chosen to initiate further steps eventually. As the victim is already on a 'trusted' website, it is likely to be less observant. Any irregularities are unlikely to be spotted, making disclosing credentials to one of the fake bank login pages deployed by phishers the final step of the fall trap. Phishing kits employing these techniques and containing templates for multiple banks are called *multipanels*, which we will examine in more detail in Section 4.

2.2 End-to-end life cycle of a phishing campaign

Whether or not advanced luring techniques are used, the steps to setup a phishing campaign are near-identical. A typical

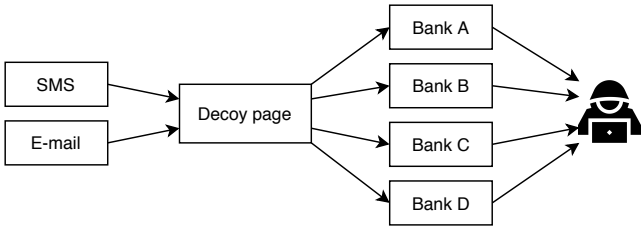

Figure 1: Luring technique with a decoy landing page and various fake banking login pages – a so-called *multipanel*

phishing attack consists out of five steps, which we illustrate in Figure 2. First, a phisher has to obtain a phishing kit that contains a website created to trick victims into disclosing their credentials. Although phishers could make this website themselves, it is much easier to deploy an off-the-shelf phishing kit that contains all the necessary resources. These phishing kits can be obtained through various sources, such as dark net markets [50] and online forums, but they have become available on public chat applications like Telegram [29] as well. Second, the phisher needs a domain where the phishing website is located. This can either be done by hijacking an insecure and unrelated website – no costs, more effort – or by simply registering a new domain name – small costs, less effort. Third, when a new domain is registered and a phishing kit obtained, the phisher needs a Web hosting provider to store the phishing kit files. Consequently, phishers often rent a Virtual Private Server (VPS), which allows them to install a Web server capable of hosting their website. Fourth, to make the phishing website look even more legitimate, the attacker acquires an X.509 (TLS) certificate to create a secure connection between victim and website over HTTPS. According to the Anti-Phishing Working Group, 78% of all phishing in 2020 is served over HTTPS [16]. This practice plays into the expectation of Internet users to observe a (green) padlock icon in the browser's address bar when visiting their bank's website – to indicate a secure connection. As Google Chrome started marking Web pages served over HTTP as *'not secure'* in September 2018 [42], potential victims could hesitate filling in their credentials when the website is not served over a secured connection. Obtaining these TLS certificates is easy and often free through certificate authorities like *Let's Encrypt* [10]. With the website in place, the phisher delivers the *bait* to potential victims by e-mail, text message or through other means and waits for victims to fill in their credentials.

As we will show in Section 4, these steps are often explained in great detail by the supplier of phishing kits, allowing their 'customers' to easily setup a phishing website

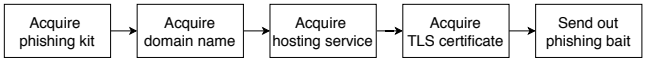

Figure 2: End-to-end life cycle of a phishing campaign

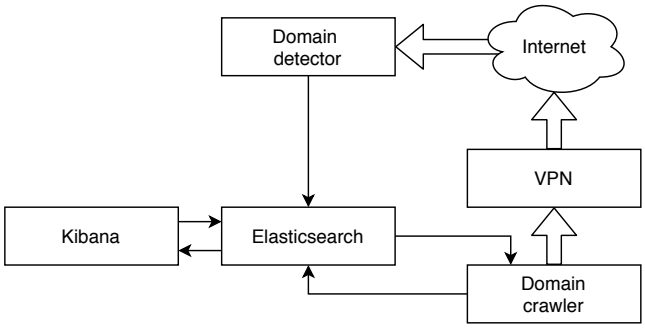

Figure 3: Architecture of our measurement system

themselves. We, on the other hand, examined these steps in the life cycle of a phishing campaign and identified the fourth step, obtaining the TLS certificate, as a valuable data source for detecting potential phishing domains. More importantly, this is also the only real-time and public data source available to us. For the remainder of this paper, we follow the steps in this life cycle to present and structure our findings. As the work on examinations and observations of phishing websites in the wild is limited [35, 39] and insights into the complete life cycle of a phishing campaign combined with thorough phishing kit analysis are absent, we designed and implemented a measurement system to monitor and analyze the Dutch phishing landscape. The focus on this one consumer market is logical as Han et al. [17] stated that phishing victims are often originating from the same country, which underlines the necessity for country specific phishing research. Likewise, earlier work on this topic highlighted the fast disappearance of phishing domains [39], making attribution rather difficult. Therefore, it is essential to create a system that could assist law enforcement to quickly respond to these attacks.

3 Measurement methodology

To study the Dutch phishing landscape, we follow the life cycle of a phishing campaign as explained in the previous section. Our measurement approach consists out of the following three steps: 1) collect phishing kits on Telegram employing snowball sampling, 2) identify possible phishing domains based on issued TLS certificates, and 3) crawl the corresponding Web pages to identify the used phishing kit and the capture the end-to-end life cycle of the attack. The methodology used to analyze each of these steps is explained in the following subsections. We store the data produced by all our measurement steps in an Elasticsearch instance, together with Kibana for easy data visualization and monitoring. The complete measurement system is deployed in Docker containers on a cloud server and presented in Figure 3.

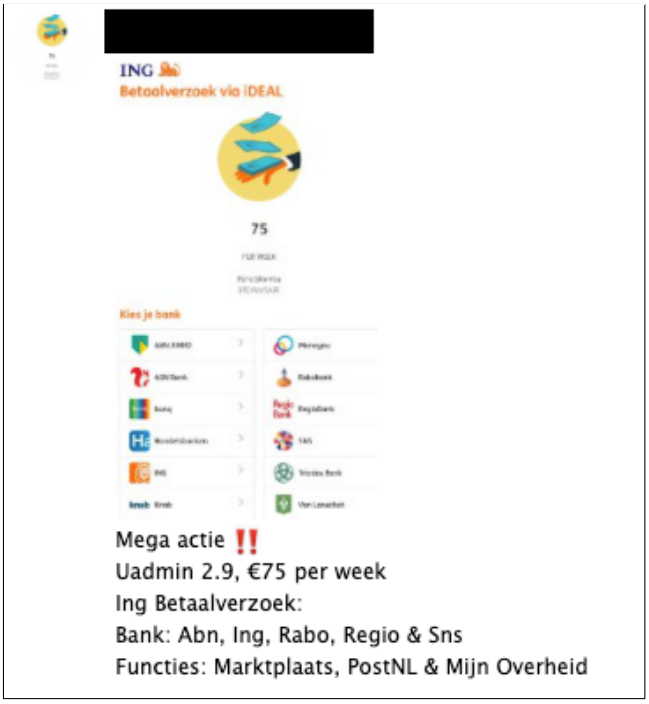

Figure 4: Example of a phishing kit offered on Telegram. This vendor offers a phishing-page-for-hire for €75 per week with templates for multiple Dutch banks included – a *multipanel*.

3.1 Phishing kit acquisition

We use two approaches to gather phishing kits that target Dutch banking clients. First, we collect phishing kits on public Telegram channels employing a so-called *'snowball sampling'* approach. In addition, we automatically download kits from open directories on crawled phishing domains. We explain both approaches in the following paragraphs.

Telegram is an instant messenger application which allows for secure communications on multiple platforms. The chat application offers a wide variety of channel types, ranging from public broadcast channels to secret chats with more security features. Encryption is applied to all messages, making it difficult to eavesdrop communications [46]. The ease of use and the high sense of security on Telegram makes it popular among criminals [29], and much easier to use compared to dark net markets or underground forums. Criminals offer illegal drugs, weapons and phishing kits on public Telegram channels, whereas direct messages on the platform allow them to negotiate prices and make deals with potential customers in private. An example of an advertisement can be found in Figure 4, which shows a vendor offering a fake *ING Betaalverzoek* (payment request) decoy page which includes templates for multiple dutch banks, a so-called *multipanel* as we have explained in Section 2.1.

To gather phishing kits from Telegram, we manually inspected fraud-related Telegram channels, searched for shared

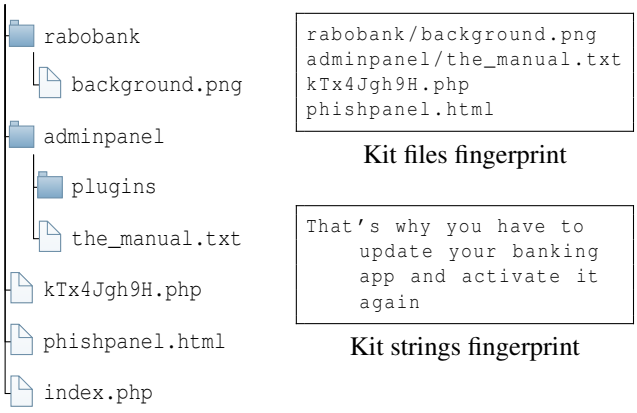

Figure 5: Phishing kit file structure and the corresponding fingerprints for both file structure and landing page strings

phishing kits and discovered related channels by following shared links in the chat. This snowball approach is a common sampling technique, that allows to reach saturation in data collection when the total population is hidden or hard to reach [1]. Our data collection saturated after we did not find any new links to our sample of public fraud-related Telegram channels ($n = 50$). Phishing kits shared in these channels are often free – e.g., as a trial version with limited possibilities – can be leased for a customized period of time – as is the case in Figure 4 – or bought from the creator or reseller for a fixed price. Kits offered in the latter category are often shared for free (*'leaked'*) afterwards to frustrate the seller.

The second approach to obtain phishing kits is to capture them from suspected phishing domains. As will be explained in Section 3.3, we crawl each suspected phishing domain and when such a domain returns an open directory, we follow the same methodology as Cova et al. [9], and search for `.zip` files to find new phishing kits that we then download automatically. Note, we did not search by trying to guess the names of popular phishing kit `.zip` files.

Fingerprinting kits We manually examined each phishing kit and created *fingerprints* based on the unique properties of these kits. Both the file names, including the full path from the root of the website, as well as strings found on the main page of the website are used to derive this fingerprint. For example, uncommon file names are considered good candidates for a fingerprint. Next, we inspect the home page of the domain to find uncommon strings in the HTML source code. This could be text shown to the victim, but also invisible HTML or JavaScript code included on the page. These fingerprints are used by our crawler to detect the phishing kits deployed on domains in the wild. An example of a phishing kit with the corresponding fingerprint is shown in Figure 5.

3.2 Domain detector

To discover new phishing domains, we leverage the fact that 78% of all phishing in 2020 is served over HTTPS – which requires the use of X.509 certificates – according to the Anti-Phishing Working group [16]. As soon as TLS certificates are issued, they appear in the Transparency Logs Project [19] – a project initiated by Google that collects all issued X.509 certificates. These logs are designed to audit the validity of these certificates, but we use this continuous stream of certificates to find new potential phishing domains. The logs can be monitored continuously using `certstream` – an intelligence feed that shares real-time updates from the Certificate Transparency Log network [6]. We thereby limit ourselves to phishing domains within two of the five categories of the taxonomy created by Oest et al. [38]. Namely, long, deceptive subdomains (type III) and deceptive top-level domains (type IV). Since TLS certificates do not contain paths after the domain name, we can not detect type I and II domains. In addition, as IP addresses – which can be used within TLS certificates – do not contain potentially malignant words, we are unable to detect type V phishing domains.

We advance on the `certstream` Python library [7] to create an application that monitors these logs for potential phishing domains. Just like Lin et al. [28], we were inspired by PhishCatcher [52], an open-source PoC demonstrating the possibilities of finding phishing domains through Certificate Transparency Logs. Our application analyzes all domains present in each certificate and calculates a score based on the features listed in Table 1, along with their assigned weighted scores. The first feature extracted is the use of Punycode within the domain name. If that is found, we increase the score with 30 and normalize the domain name for further analysis by converting the Punycode symbols to their regular counterparts. For instance, we convert `xn-pypl-loac.com` to `paypal.com`, which we then use in further steps. We increase the score with 20 for domains hosted on the 10 most abused TLDs according to Spamhaus [41]. Afterwards, we split this domain name into words and search for fake TLDs (which could be part of domain names of targeted Dutch banks, so `.com`, `.nl`, `.me`), brand names (of the 13 targeted Dutch banks) and suspicious keywords (a list of 78 words we made ourselves). We also identify typosquatted variations of the latter two by searching for words with a Levenshtein distance of 1 within the domain name. Additionally, we count the number of hyphens and subdomains and inspect the certificate. The score for domains listed in a free certificate is increased with 20. For domains included in a (paid) certificate with *Extended Validity*, we decrease the score with 100, as we do not expect attackers to pay and complete verification process. Finally, we disregard domains from Dutch banks and a number of cloud service providers through a white list to prevent false positives. When a threshold of 110 is reached, the domain is marked as potentially malicious and added to the Elasticsearch index along with the extracted features and the complete X.509 certificate. This threshold was determined after our testing period in June-August, 2020, and was considered a good balance between true and false positives. Do note that we aim to collect as many *potential* phishing domains, while keeping the number of false positives manageable. This means that the threshold is not fully optimized to a specific value. Ultimately, our domain crawler – explained in the next section – is responsible for the actual identification of phishing domains.

Table 1: Features used to detect potential phishing websites

Domain feature	Example & references	Score
Punycode usage	`xn-pypl-loac.com` [11, 30]	30
Suspicious TLDs	`.xyz`, `.icu`, `.top` [16, 41]	20
TLD as subdomain	`x.com.domain.net` [16, 27]	20
Brand name	`brand.domain.net` [16, 27]	40-150
Typosquatted brand	`paypa1.com` [22, 27]	0-110
Suspicious keyword	`login`, `verify` [27, 31]	25-50
Hyphens count	`brand-n--ame.net` [18, 27]	$3x$
Subdomain count	`sub.x.domain.net` [27, 32]	$3x$
Free certificate	`Let's Encrypt` [16, 48]	20
Fake `www`	`wwwbrand.com` [22]	45

3.3 Domain crawler

To find traces of the gathered and fingerprinted phishing kits, we crawl each of the domains detected by our domain detector. Every hour, the crawler retrieves new possible phishing domains from the Elasticsearch index and starts processing them subsequently. First, it determines if the domain is online, and if so, a FireFox browser controlled by the Selenium WebDriver [43] is launched and visits the domain just like a regular user would. All outgoing Web traffic is routed through a VPN connection to obfuscate our IP address and to easily change our IP address when necessary. While visiting the Web page, the IP address is resolved, HTML sources are stored, and a screenshot is taken. The *favicon* is extracted and hashed using an average hashing function [23], similar to the method suggested by Geng et al. [13]. They showed that more than 83% of phishing websites employ fake *favicons* mimicking the targeted brand or organization. Geng et al. created an algorithm that is able to identify similar *favicons* by comparing the gray values of pixel rows to detect the slightly changed ones. Such hashing is thus perceptual, meaning that small changes in the image result in only minor hash changes. We used their methods to identify domains that do not mimic one of the targeted brands by comparing the *favicon*'s hash to the hashes of Dutch banks *favicons* (12 different brands, 24 icons in total). A domain is omitted from further analysis when the Hamming distance between the found hash and all the hashes Dutch banks differs more than 10%. If no *favicon* is present,

the domain is analyzed further. Another perceptual hash is generated for the screenshot of the visited page. This hash is used to spot any differences on the page since the last visit. If the hash has not changed since the last visit, we skip further analysis. Otherwise, we continue the analysis by retrieving the `WHOIS` record, which reveals the registrar and the creation date of the domain.

Finally, we start the phishing kit identification phase. In this phase, we adopt a three-layer approach. First, the crawler starts with a search through the list of loaded resources of the Web page. The format of the fingerprints allows us to search for partial file path matches within this list of resources. Given the example in Figure 5, resource `https://domain.com/rabobank/background.png` matches fingerprint `rabobank/background.png`. Secondly, we perform a string-based search on the landing page to find matching string fingerprints – e.g., if the page includes the sentence from Figure 5, it will be detected. To be able to detect phishing kit resources that are not loaded on the landing page of the website, we perform an extensive search for files and directories on the server using `wFuzz` [33], which tries to `HTTP GET` all resources included in the fingerprint. Given the example in Figure 5, resource `adminpanel/the_manual.txt` is not loaded on the landing page of the website, but can be detected in this third phase. To harden our detection method against minor changes in phishing kits, we decided to classify a domain as true phishing and identify it as being made with a particular phishing kit when at least 10% of a fingerprint is found in one of these steps. We removed false positives due to this low threshold from our dataset manually in Section 5. Each domain that is inserted into the Elasticsearch index is monitored on an hourly basis for a maximum of seven days after the initial analysis.

3.4 Deployment and testing

Figure 6 gives an overview of the process of deploying our measurement setup and data collection period. As elaborated on in Section 3.1, the research started with an exploration on Telegram for phishing kits. These kits were dissected to create fingerprints, and then utilized to detect phishing activity on domains. In parallel, we started building our measurement system and as one can can see, we dedicated a significant portion of time on developing, reviewing, and upgrading our deployment.

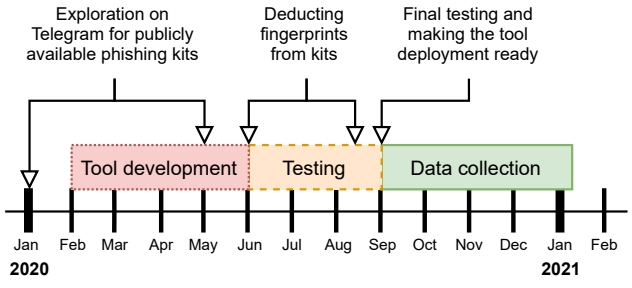

Figure 6: Timeline of the creation and testing of our measurement methodology

During our testing phase, newly found phishing kits from open directories are constantly added manually to the crawler application. During this same testing phase, we also identified five new, unknown, phishing kits on domains labeled as potentially malicious by our domain detector. However, the crawler could not find any matching fingerprints and labeled these domains as *potentially phishing*. After manual inspection, we determined that these domains were indeed phishing, and we created fingerprints based on the characteristics of these live domains, similar to what we did for the phishing kits in Section 4.2. We completed this iterative process five times during our testing period and grouped these phishing kits as *unknown*. In September, 2020, we stopped testing, made no further changes and started the data collection.

4 Phishing kit analysis

As discussed in Section 2, phishing campaigns hinge on successful deployment, which can be made easy with a phishing kit. To collect these kits, we manually inspected public Telegram channels following a snowball sampling approach and downloaded `.zip` files from open directories on potential phishing websites. Our initial search in January, 2020, resulted in a collection of 36 phishing kits discovered by manually inspecting 50 public Telegram channels. In the following months, we continued to monitor these channels periodically and gathered yet another 10 phishing kits in May, 2020. Additionally, as explained in Section 3.3, we automatically downloaded `.zip` files from open directories on phishing websites, which resulted in a collection of another 24 phishing kits retrieved in the period July – December 2020. In total, we gathered 70 different phishing kits, which we then manually dissected. We analyzed their operating procedures and techniques, came to understand the anatomy of a typical phishing kit and clustered their features to discern phishing kit families. The results of these analyses are outlined in the following subsections.

4.1 Anatomy of a phishing kit

A phishing kit consists out of many files that together ensure the functionality of the kit when deployed. Among these files, we typically find:

- **Front-end pages** impersonate the original login screens of the targeted banks or can be categorized as decoy landing pages (as explained in Section 2.1), which direct the victims to fake login screens afterwards.

- **Resources** are the files behind the front-end pages, such as JavaScript, CSS and images. These can either be hosted on the same server – hence included in the phishing kit – or retrieved from the website of the targeted organization.

- **Manuals** are often located in the root folder of the phishing kit and include detailed instructions on how to setup a VPS, acquire a TLS certificate and install the phishing kit. These files often mention default login credentials and a reference to the creator of the kit.

- **Control panel**, allowing the phisher to access the backend of the phishing kit, view the phished credentials, or trigger new events for the victim. These panels range from simple text files to extensive dashboards with live visitor manipulations, statistics and third-party integrations like Jabber – a XMPP instant message service.

- **Anti-detection (cloaking) methods** are present in some kits to prevent detection by law enforcement agencies, independent researchers like us or anti-phishing services such as Google SafeBrowsing [20]. For example, setting up strict IP blockades on the server-side in an `.htaccess` file as discussed by Oest et al. [38] or by redirecting certain visitors based on their IP address, geolocation or User-Agent string through PHP scripts. This can also be done client-side by utilizing JavaScript as discussed by Invernizzi et al. [21].

4.2 Phishing kit families

Precise distinctions between the 70 phishing kits are difficult to make, due to the unstructured nature of phishing kit development. During our manual dissection of the gathered kits, we noticed that a large portion of these kits contained copies, older versions or modifications of one another. Creating unique fingerprints for each of these kits is therefore difficult, as such fingerprints could easily match a slightly changed or copied version of another kit. To solve this problem and enable analysis on their usage, we categorized the gathered phishing kits into 10 families by comparing the files present within each kit. For each of the gathered phishing kits, we calculated the percentage of overlapping files by comparing them pairwise and counting file path matches. Following a similar methodology as Bijmans et al. [3], we used a graph structure to find clusters of similar phishing kits that we can group into families. Displayed in Figure 7 we find a directed graph with phishing kits shown as nodes and edges created due to overlapping files. An edge between two phishing kits is created if 75% of the files in a kit are overlapping. To find families of kits that belong together, we employed a community extraction technique proposed by Blondel et al. [4]. This is a heuristic method based on modularity optimization. The resulting structure describes how the network can be compartmentalized into smaller sub networks. Utilizing this technique we determined 10 families of at least two phishing kits per family, in which we group 53 phishing kits. The remaining 17 phishing kits have no significant overlap with others and are thus considered not part of any family. An overview of the five largest phishing kit families can be found in Table 2.

When taking a closer look at Figure 7, we clearly observe one large interconnected network containing four different phishing kit families - the *uAdmin*, *tikkie*, *ics*, and *livepanel* families. From this large community we can confirm the hypothesis that phishing kits 'learn' – or steal – a lot from each other. The *uAdmin* and *tikkie* families have a lot of overlapping files, but are nevertheless separated in two families. By examining the codebase of both these families more closely we can see that, whilst they both build upon the same framework – which will be explained in the following paragraph – they have slightly different possibilities. Following this same logic, we took a closer look at the *ics* family. These kits are connected to the larger network through merely one kit. The framework used in that phishing kit connects the *ics* family to the network and is again built upon the same codebase as the rest of the cluster. However, it is interesting that the other three kits in the *ics* family are not built upon this framework, but do have the same target as the connecting phishing kit. This indicates that this family has 'evolved' into using this framework to perform their phishing activities, adapting to newer technologies. The other, smaller families, positioned to the right in Figure 7, clearly employ different tactics compared to the large interconnected network. For example, the five phishing kits in the *bonken* family are all built upon the ASP.NET Core platform, and have nothing in common with the other clusters. As the two largest families and 26 phishing kits in our dataset are build upon the same framework, we highlight its characteristics in the following paragraph.

uAdmin framework Universal Admin – better known as the uAdmin control panel – is a framework written in PHP and uses a SQLite3 database for information storage. As PHP can be found on almost every Web server and has built-in support for SQLite, this panel can be deployed very easily. It allows for many different templates for most Dutch banks, as well as various decoy pages (as explained in Section 2.1). An unique feature is that the administrator panel can be hosted separately

Table 2: Analysis on the five major phishing kit families

Family	# kits	Technology	Type	Decoys
uAdmin	17	PHP, SQLite3	multipanel	✓
tikkie	9	PHP, SQLite3	multipanel	✓
bonken	5	ASP.NET	multipanel	✗
ics	4	PHP, MySQL	multipanel	✗
livepanel	4	PHP	single page	✗

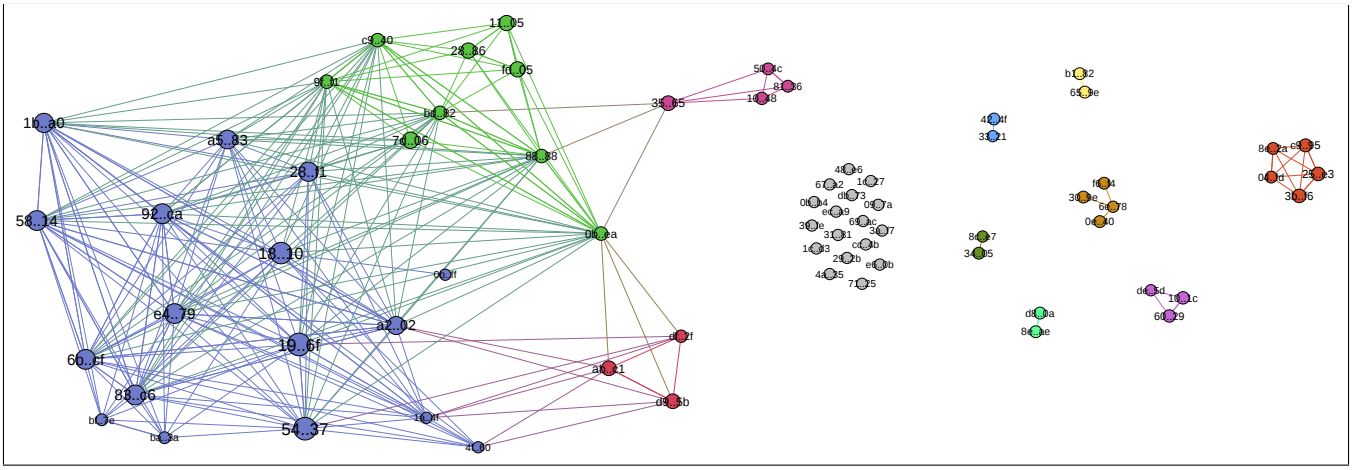

Figure 7: Graph of phishing kit families with the first and last two characters of the MD5 hash of the phishing kit name ($n = 70$)

from the phishing page. This makes it easy to setup multiple phishing domains and proxy all their connections to a control panel hosted elsewhere. Part of the uAdmin framework is the *O-token* plugin, enabling real-time interaction with the victim. It includes a detailed log of all entered information, as well as buttons to prompt the victim for more input and the possibility to integrate Jabber notifications. This real-time interaction with the victim also allows the attacker to act as a man-in-the-middle to defeat two-factor authentication defenses. uAdmin employs a number of anti-detection methods. There is an `antibot.php` script, which blocks a list of IP addresses, host names and User-Agents. Additionally, when a victim visits one of the pages, a unique folder is created on the Web server, all necessary resources are copied into it and the victim is redirected to that folder after a timeout of 1 second, as shown in Listing 1.

```
$random = rand(0, 10000000);
$md5    = md5("$random");
$base   = base64_encode($md5);
$dst    = 'a1b2c3/'.md5("$base");
...
$src = "def";
dublicate($src, $dst);
...
<script type="text/javascript">
 setTimeout(function(){
   window.top.location.href='<?php echo $dst."?".
       $_SERVER["QUERY_STRING"]; ?>'; },1000)
</script>
```

Listing 1: Anti-detection techniques employed (file copy and a JavaScript redirect) by the uAdmin phishing kit family

This code snippet is very similar to the code mentioned by Han et al. [17] and Oest et al. [39] and tries to prevent detection by anti-phishing services like PhishTank [40] or Google Safe Browsing [20]. Han et al. [17] discovered that these anti-phishing bodies crawl submitted domains themselves and place the landing URL on their block lists. In this case, this is a random path on the Web server, hereby preventing detection. In February, 2021, the Ukrainian attorney general's office reported that they arrested the developer of the uAdmin phishing kit, after reports that it was used in more than half of all phishing attacks in Australia in 2019 [25]. The Australian Federal Police stated that *"Pretty much every Australian received a half dozen of these phishing attempts."* [25]. Financial institutions in 11 countries including the United States, Italy and the Netherlands were suffering from phishing attacks through uAdmin.

5 Phishing domain analysis

Using the method of analyzing the stream of issued TLS certificates in real time as described in Section 3.2, our domain detector labeled 7,936 domains as potentially malicious, which meant that these domains reached the threshold value and are further analyzed by our crawler. The domain crawler could match fingerprints of known phishing kits on 1,504 of these domains, which we all manually checked for false positives. We removed 61 domains on which our crawler discovered fingerprints, but no actual phishing took place. On most of these domains, this was the result of a fingerprint not being specific enough, and in some cases the domain responded successfully to all `HTTP GET` requests and thus matched all fingerprints. Finally, as we are investigating the complete end-to-end life cycle of phishing campaigns, we only included domains able to complete a life cycle. This meant that we excluded domains that were discovered in the final week of the data collection period and therefore omitted another 80 domains from our dataset. Our final dataset contained 1,363 verified phishing fully qualified domain names (FQDN) which have been online for at least one hour. These were hosted on 1,112 different registered domain names (RDN), as some domains hosted multiple phishing pages on different subdomains. A summary of our dataset is listed in Table 3.

Table 3: Summary of our phishing domains data collection

Data collection start	September 6, 2020
Data collection end	January 6, 2021
Amount of visits made by crawler	499,497
Amount of potential phishing domains found	7,936
Amount of identified phishing FQDN	1,363
Amount unique phishing RDN	1,112
Average amount of FQDN online every day	31
Median time online (h)	24

5.1 Domain name characteristics

Setting up a new phishing domain requires a balance between the right amount of persuasion of the victim and stealth to prevent early detection by anti-phishing organizations. As explained in Section 3.2, common practices to hide malicious activity are to obfuscate (parts of) the URL by using deceptive subdomains, Punycode or typosquatting. The use of deceptive subdomains is categorized as type III by Oest et al. [38] and we could discover only 66 of such domains in our dataset. As listed in Table 4, we identified much more type IV domains (1,297) in our dataset. 16 of the 66 type III phishing FQDNs increased their credibility by including the full FQDN of the target brand as subdomains. This practice can be the result of either one of the following techniques: this RDN could be hijacked or especially chosen to increase stealth. In the case of hijacked domains, attackers have taken control over the domain and made (multiple) subdomains for their phishing page, a practice discussed extensively by Han et al. [17]. For the other technique, adding the domain of the targeted bank as a subdomain is done to increase the credibility of the URL, which works especially well on mobile devices on which the complete URL is not always shown in the GUI. Distinctions between these two categories are difficult to make, as we can not determine whether a domain is hijacked or chosen on purpose by the attacker to avoid early detection.

Although mentioned in related and previous work on this phenomenon [26, 30], we did not find any successful usage of Punycode obfuscated domains in our dataset. The use of Punycode did increase the malicious score of a domain in our domain detector, and we identified 21 of such domains, but our crawler did not find matching fingerprints on any of them. This could indicate that the use of Punycode is less popular among attackers focused on Dutch consumers, as we did find references to other banks outside our scope. On the other hand, typosquatting – also known as URL-hijacking – is found 36 times in our dataset. The practice of replacing the character `i` with `l` in domains mimicking the *ING Bank* and *ICS Cards* is popular, as we found respectively 16 and 20 of such domains.

However, most phishing FQDNs in our dataset simply obfuscate their malicious intents by not mentioning the name of the target organization. As shown in Table 5, more than half of the domains in our dataset (770) did not include any references to Dutch banks, but were detected because of other words mentioned, which we included in our methodology as *suspicious keywords*. These words refer to either banking related matters – e.g., *payment*, *verification* or *debit card* – or to completely different matters, often related to the decoys mentioned in Section 2.1.

Targeted banks An analysis of the FQDNs that do refer to one of the targeted banks results in insights into their popularity. Note however, indicators in the domain name are not always directly linked to the actual Web page on that domain – e.g., a domain including a reference to bank A contains the login screen of bank B. Our domain detector searched for references to the ten largest Dutch retail banks and two daughter brands of ABN AMRO – Tikkie and ICS Cards – within all domains and was able to identify 593 FQDNs referring to one of them. As shown in Table 5, we found 194 domains referring to the Rabobank, which makes it the prime target for attackers. In contrast, only ten domains contained references to Regiobank, making this bank to seem a less attractive target.

5.2 Domain registrations

When choosing a top-level domain (TLD) as an attacker, it is important to keep in mind that different registries have different policies when it comes to monitoring and cleaning of their TLD. Some registries allow registrars – the companies selling the domains used for phishing to the attackers – to sell large quantities of domain names to attackers and are hereby knowingly contributing to online abuse. As The Spamhaus project states: *"Some registrars and resellers knowingly sell high volumes of domains to these actors for profit, and many registries do not do enough to stop or limit this endless supply of domains."* [41]. The Spamhaus Project monitors domains in SPAM messages and calculates the percentage of bad domains within each TLD zone. We compare their data with our results to find out whether phishers focused on Dutch consumers favour these TLDs over the more regularly used TLDs in the Netherlands. The results of our analysis – listed in in the first columns of Table 6 – show that `.info` is the most commonly used TLD in our dataset, followed by `.xyz`. These phishers tend to choose one of the many 'bad' TLDs, but they also stick to the more commonly used TLDs in the Netherlands, such as `.com` and `.nl`.

Domain registrars Using the retrieved `WHOIS` records, we were able to identify the registrar of 933 of the 1,112 RDNs in our dataset, we thus have no information about the registrar for 179 RDNs. Inspecting the `WHOIS` records of the 933 domains, revealed that *Namecheap* is by far the most popular registrar used by phishers, as 72.6% of all phishing domains was registered through that registrar. Other large registrars, such as

Table 4: High-level classification of detected domains with examples from our study

Type III Long, deceptive subdomain	ics-beveiligingsprocedure.zap123456-7.plesk11.zap-webspace.com mijn.ing.nl.u1234567.cp.regruhosting.ru	66	4.8%
Type IV Deceptive top-level domain	betalingsverzoek-online.link ing-verificatiepagina.eu	1,297	95.2%

Table 5: Popularity of targeted banks and suspicious keywords

Brand name	#	**Suspicious word**	*(translation)*	#
Rabobank	194	Betaal	*(pay)*	300
ING Bank	135	Verzoek	*(request)*	271
ICS Cards	48	Mijn	*(my)*	217
Tikkie	40	Veilig	*(secure)*	159
Knab	37	Betaling	*(payment)*	153
ABN AMRO	25	Omgeving	*(environment)*	119
Bunq	16	Platform	*(platform)*	116
SNS Bank	13	Verificatie	*(verification)*	87
Regiobank	10	iDeal	*(iDeal)*	73
Triodos	8	DigiD	*(DigiD)*	70
Not mentioned	770	Not mentioned		125

Porkbun and *GoDaddy.com* are significantly less popular than one would expect. Another interesting observation is the use of *REG.RU*, a Russian domain registrar, which is found 46 times in our dataset. An overview of the 10 most popular domain registrars can be found in the middle of Table 6.

Certificates authorities The fourth step in the end-to-end life cycle of a phishing campaign is acquiring a TLS certificate. As explained in Section 3.2, we leverage this step to detect phishing domains in our analysis. *Let's Encrypt* is the main supplier of TLS certificates in our dataset, as 67% of all FQDNs use such free certificates. Additionally, we found 146 domains with a certificate issued by *cPanel*, software often used to manage the domain. Most certificates (99%) are Domain Validated (DV), but we gathered 33 TLS certificates issued through *CloudFlare*'s free certificate service which were Organisation Validated (OV). These certificates require additional validation steps which are highly unlikely for a phisher to fulfill, as this would disclose their identity.

6 Phishing website deployments

In the four month data collection period, our domain crawler made a total of 499,497 visits to 7,936 unique FQDNs. As explained in Section 3.3, the crawler visits every domain labeled as potential phishing by our domain detector and monitors it for a period of a maximum of seven days after initial discovery. Properties such as the used phishing kit, the IP address and `WHOIS` record are gathered during this process.

Besides choosing a suitable TLD and a domain name to be used for their phishing attack, phishers also need a place to host their website. By resolving the IP addresses of identified phishing domains and mapping them to their corresponding Autonomous System Numbers (ASNs), we determined the hosting provider of each domain. An overview of the top ten providers can be found in Table 6. Similar to the domain registrations mentioned in Section 5, *Namecheap* is the most popular hosting provider among attackers in our dataset. The overall popularity of *Namecheap* has various reasons. First of all, it is – like it says – cheap and as attackers want to maximise their profits, it makes sense to rent an inexpensive VPS instead of an expensive one. Second, *Namecheap* accepts payments in Bitcoin [36], which offers more operations security to attackers due to the relative anonymity of Bitcoin transactions. Finally, it is mentioned explicitly by various phishing kit creators in their manuals.

Surprisingly, none of the hosting providers in this list can be regarded as *bulletproof* – i.e., very reluctant to LEA requests – except from HS, short for Host Sailor. This provider does have a disreputable background [24], but is used by only 58 domains in our dataset. Another interesting entry in Table 6 is *Combahton*, an inexpensive German hosting provider used by services like *zap-webspace.de* and *gamingweb.de*. From the lack of bulletproof hosting providers we derive that these phishers are not concerned about an extended lifespan of their domain. As long as they act quickly, they are long-gone before their domain is taken offline by third-parties. However, the choice of these services does open avenues up for possible law enforcement interventions, as mainstream hosting providers – such as *Namecheap* – are willing to cooperate with law enforcement.

6.1 Phishing kit prevalence

As stated in Section 4.2, we obtained a total of 70 phishing kits, which we dissected and grouped into 10 families of similar kits. During the data collection period, our crawler found matching fingerprints for 7 of the 10 different families. We show the size of the Dutch phishing landscape and the popularity of the different phishing kit families in Figure 8, in which the total number of active and online phishing domains are shown per day, categorized per phishing kit family.

Although we expected a wide variety of phishing kits to be used, the opposite turned out to be true. The overwhelming majority of phishing domains our detector found was made

Table 6: Overview of the top 10 top-level domains (TLDs), domain registrars and hosting providers used by attackers

TLD ($n = 1,112$)	#	%	**Registrar** ($n = 933$)	#	%	**Hosting provider** ($n = 836$)	#	%
.info	202	18.2	Namecheap	678	72.6	Namecheap	280	33.5
.xyz	159	14.3	REG.RU LLC	46	4.9	Combahton	84	10.0
.com	149	13.4	Porkbun LLC	30	3.2	HS	58	6.9
.nl	102	9.2	NameSilo, LLC	21	2.3	Alibaba (US) Technology Co.	56	6.7
.me	74	6.7	Eranet International Ltd.	17	2.4	Cherryservers	29	3.5
.icu	71	6.4	GoDaddy.com, LLC	12	1.3	First Colo	26	3.1
.online	57	5.1	Tucows Domains Inc.	12	1.3	NCONNECT-AS	24	2.9
.site	50	4.5	AXC	10	1.1	Serverion	23	2.8
.net	28	2.5	Hosting Concepts B.V. - Openprovider	8	0.9	YURTEH-AS	14	1.8
.top	23	2.1	Registrar.eu	6	0.6	OVH	14	1.7

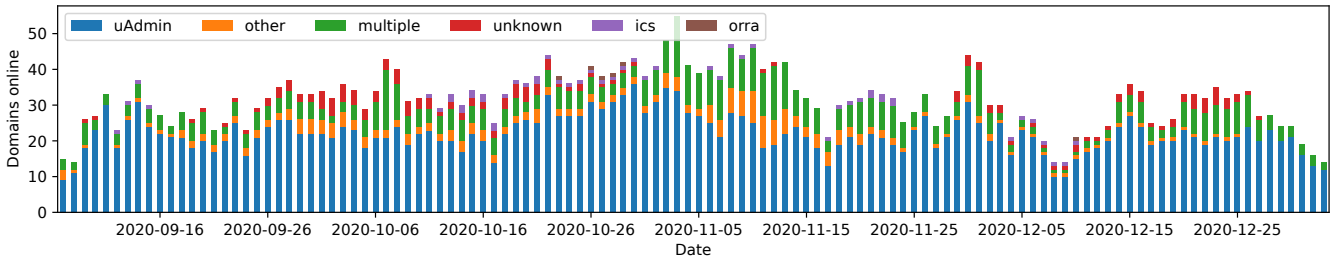

Figure 8: Number of domains active per day, grouped per phishing kit family ($n = 1,363$)

using one of many variants of phishing kits within the *uAdmin* family. Almost 89% of all identified phishing websites were made with a kit within this family, shown by the size of the lower blue bars in Figure 8. As explained in Section 4.2, these phishing kits contain many templates for different banks and often include decoy pages, making them attractive to aspiring phishers. The support for many different bank login templates also explains why many of the domains are labeled as *'multiple'* in Figure 8. These domains have fingerprint matches of both *uAdmin* and another phishing kit family. It seems that phishing kit creators are integrating as many templates as possible from different kits into the uAdmin framework. The structure of this framework remains often unchanged, as we could locate the control panel on its default location on 775 of the 1,211 FQDNs (64%) that matched a fingerprint of a phishing kit in the *uAdmin* family. Finally, as shown in red in Figure 8, the category *unknown* consists of new, unknown phishing kits found on live phishing domains. As explained in Section 3.4, we manually verified that these domains were indeed phishing, and created fingerprints of the used kits according to the characteristics of these live domains.

6.2 Campaign duration

Since our crawler monitored each identified phishing domain for a maximum of seven days (168 hours), we were able to closely follow these domains and capture the end-to-end life cycle of a typical phishing campaign. Additionally, as stated before in Section 5, we manually checked the dataset

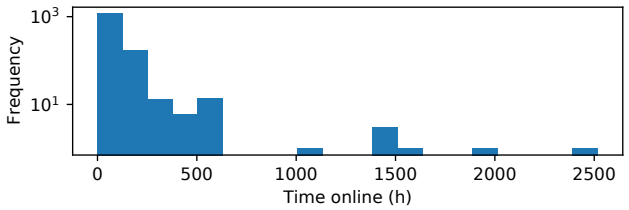

Figure 9: Histogram of phishing domain uptimes, domains with multiple certificates included ($n = 1,363$)

to prevent any false positives from being included in the data and only included domains with a complete end-to-end life cycle in our analysis, which allowed us to analyze this in the next paragraphs.

First, we plot a histogram of the uptimes of all domains in our dataset in Figure 9 with a logarithmic Y-axis. As one can see, the majority of domains have an uptime of 0 to 200 hours, which coincides with our maximum analysis period of 168 hours. However, there are 75 domains with an uptime of more than 168 hours (7 days). After manually inspecting this unexpected result, we found that these domains requested multiple TLS certificates during their uptime, which caused our domain crawler to restart the crawling cycle as soon as a new certificate was issued. Since these outliers heavily influence the results and prevent us from determining timestamps of the steps in the life cycle, we exclude them for the remainder of the analysis in this paragraph.

Now, we are able to calculate the uptime of the 1,288 remaining phishing domains in our dataset. On average, a phishing domain in our dataset is online for 45 hours, but we find a median uptime of 24 hours. The uptimes are shown as a cumulative distribution function (CDF) in Figure 10. Thus, 50% of all domains have a lifespan less than a day, whereas just over 30% is online for more than two days. These numbers again stress the fact that speed is key in anti-phishing initiatives.

Installation of phishing kits Although it is hard to determine which actors are behind phishing attacks on Dutch consumers, the timestamps of the first identification of an active phishing kit installation does give some clues into the region of the world these attackers operate from. And as shown in Figure 11, the phishing kit installation times (in UTC+1) line up nicely with the Dutch circadian rhythm. Most phishing kits are installed successfully during the day, whilst almost none of them are installed in the middle of the night. This finding, and the fact that most manuals of the gathered phishing kits are written in Dutch, extends the conclusions of Han et al. [17], as this would indicate that both victim and attacker originate from the same country.

During installation and testing of the kit, visitors are occasionally redirected to popular benign domains like Google or Bing, or to the website of the target organization. During our crawls, we observed 49 different phishing domains doing this before their phishing kit was fully deployed and operational.

End-to-end life cycle steps We can determine timestamps of all steps within a typical phishing campaign – shown in Figure 2 and explained in Section 2 – by combining the retrieved `WHOIS` records and crawling timestamps of all identified phishing domains. Unfortunately, 460 FQDNs in our dataset lack `WHOIS` information due to inconsistent information formatting or server errors beyond our control. Therefore, these domains are excluded from the analysis in this paragraph. Additionally, we focus this analysis on type IV phishing domains only. As type III domains include hijacked domains which have not been registered purposefully for phishing.

The end-to-end life cycle analysis on the remaining 818 domains is summarized in Figure 12 as a horizontal box plot, with the hours since domain registration on a logarithmic X-axis. As indicated by the red bars inside the boxes, a phishing domain is online – i.e., returns a successful `HTTP 200` response – three hours after registration on average. Oftentimes, quickly followed by the installation of a phishing kit, on average only one hour later. After a successful installation, the phisher sends out the *bait* to its potential victims and waits for credentials to be filled in. The domain goes offline after 40 hours on average. The majority of domains complete this full life cycle within a couple of days. Note however, that there are also outliers. In these cases, the domain was registered many days in advance, waiting to be used by the attacker. In our dataset, only 114 of the 818 domains (14%) were registered more than 24 hours before coming online.

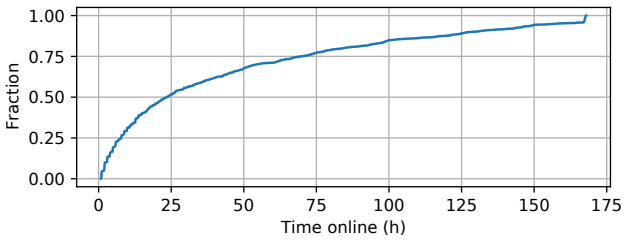

Figure 10: CDF of a phishing domain uptime ($n = 1,288$), domains with multiple certificates issued excluded

6.3 External resources & evasion techniques

During our analysis of phishing domains in the wild, we noticed that some websites make external connections. As explained in the Section 4.1, phishing websites could either include all impersonated resources – e.g., JavaScript, CSS and images – on the domain, or refer to resources hosted externally. Analyzing the resources loaded by all identified phishing domains tells us that only 104 domains (7.6% of the total dataset) load their resources directly from their benign counterparts. This finding contradicts the assumption underlying the work of Oest et al. [39] and makes their method of analyzing Web server logs for malicious external requests less robust, as only a very small portion of websites in our dataset is pursuing this method. However, it does confirm the findings of Han et al. [17] and Cova et al. [9], who also observed a negligible portion of phishing kits with resources loaded from the target organization. These authors studied attacker behavior on honeypot domains, which is based on the assumption that attackers hijack domains to use for phishing. Although our measurement methodology is not perfectly suited to find such hijacked domains – as these domains often already have TLS certificates – we did find 18 of them. All of these domains include the full FQDN of the target organization as subdomains and have a slightly longer uptime of 72 hours on average.

Evasion techniques As explained in Section 4, some phishing kits deploy evasion techniques to prevent detection by anti-

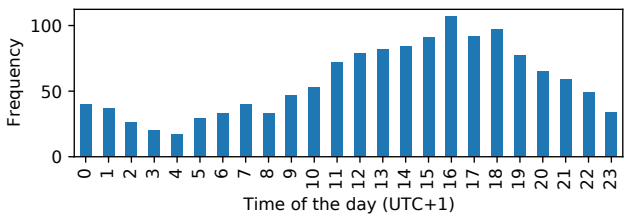

Figure 11: Histogram of kit installation hours ($n = 1,363$)

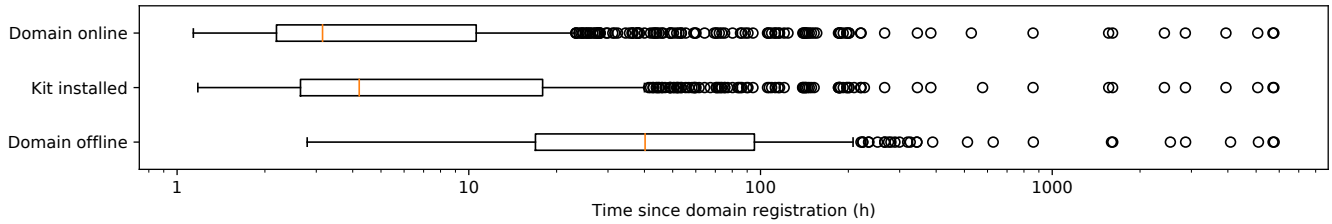

Figure 12: Boxplot of timestamps in the end-to-end life cycle of the identified phishing domains ($n = 818$)

phishing services such as APWG [14] and Google SafeBrowsing [20]. These techniques, often referred to as *cloaking*, allow phishers to show a different page to a potential victim than to a crawler [21, 53]. Although our methodology is focused on detecting the use of specific phishing kits in the wild and not to identify cloaking, we did observe such evasion techniques many times. In fact, 946 (69%) of the detected phishing domains returned a blank screen – and no favicon – to our crawler when we visited the domain, meaning that the phishing website detected us and deployed cloaking techniques. However, our phishing kit detection was still possible because these websites returned a successful response for the files included in the fingerprint. The phishing kit responsible for most of these cloaking activities was again the *uAdmin* kit, which combined some server-side and client-side cloaking. On the server-side, it checked the IP address with a block list and created a random path for every visitor, as explained in Section 4.2. On the client-side, it deployed a simple JavaScript timeout to evade non-JavaScript crawlers. The combination of both techniques is shown in Listing 1.

7 External validation

To benchmark and validate our methodology, we compare our results with data from the APWG eCrime Exchange (eCX) [15]. This repository contains phishing activity from all over the world, including many Dutch phishing domains. A comparison shows that our methodology covers a much broader spectrum of phishing domains, capturing known differentiations in the phishing landscape. In total, only 77 phishing domains detected using our methodology, overlap with the APWG database, meaning that 1,286 domains are not listed in their repository. By comparing the date on which a phishing domain was initially detected by our crawler with the data it was submitted to the eCX, we find that our method was able to identify phishing domains much faster. In 76 out of the 77 cases (99%), our crawler detected the phishing domains faster than APWG, with a median time difference of 11.3 hours (almost half a day) earlier. Interestingly, the domains that overlap with the eCX repository had clearly more bank names included in their domain name. 61 of the domains (79%) overlapping with eCX contained a reference to a bank, whereas only 44% had this in the complete dataset.

This external validation shows that our methodology has the potential to detect phishing websites very swiftly which could save unsuspecting people from this kind of fraud.

8 Throwing out the bait

In the previous sections of this paper, we have unraveled the characteristics of every step in the end-to-end life cycle of a phishing campaign, except for the last step: sending out the text messages, e-mails or social media posts, the so-called *bait*. Although our measurement system does not contain the input data necessary to thoroughly analyze this step of the life cycle, the authors are among the target population of phishers and thus regularly receive the thrown out bait themselves. During our data collection period, we collected these messages and looked into the ones that contained links to domains in our dataset. This allows us to show the complete timeline of events in a phishing campaign life cycle. We discuss an example in the following section.

Verify your identity Within the first two weeks of our research, we received a text message seemingly originating from DigiD, the official Dutch digital identity service. The message shown in Figure 14, stated that a suspicious login was detected and that immediate action was necessary to prevent cancellation of the account. This is a prime example of the *scarcity* and *consistency* luring techniques as described in Section 2.1. The link included in the message directed victims to `https://deblokkeren-digid.xyz`, a type IV domain made with a phishing kit belonging to the *uAdmin* family. This website was registered only six hours before the message was received and fully operational just three hours later. On the website, potential victims were asked to verify their identity by logging into their online bank account. Multiple options are displayed on the decoy page as shown in Figure 13a, allowing the victim to choose their preferred bank. Upon clicking on one of the buttons, the victim is redirected to yet another phishing page as shown in Figure 13b, which mimics the chosen bank's login screen. That page eventually captures the login credentials of the victim. The use of the DigiD decoy page is a prime example of the technique depicted in Figure 1. Within a day, only 12 hours later, the domain was taken offline.

(a) (b)

Figure 13: Landing (decoy) page in 13a: indicating that verification through a bank account is necessary to prevent account deactivation and the actual phishing page in 13b after clicking on a bank of choice on which user credentials are harvested

SMS Message – 17-09-2020 21:32 *(translated)*

[My DigiD] There has been a suspicious login in your My DigiD account. Verify this directly to prevent cancellation of your My DigiD account through: `https://deblokkeren-digid.xyz/inloggen`

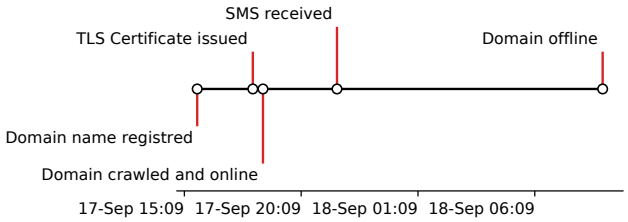

Figure 14: Text message demanding DigiD verification and corresponding timeline of `deblokkeren-digid.xyz`

9 Related work

Earlier work on phishing involves many different points of view and subjects. Ranging from creating robust domain detection methods [5, 12, 13, 27, 32, 45], phishing kit analysis [9, 17, 38], evasion techniques [21, 53] to research focused on victim behavior [49]. Much effort has been devoted to the creation of robust detection techniques, but less is known about the life cycle, ecosystem and actors behind such attacks. Only a limited number of researchers have investigated this part of phishing [35, 39], which we deem essential to fully understand the ecosystem and to be able to create robust countermeasures.

Analysis on phishing kits Early work on phishing kits in 2008 by Cova et al. [9] focused on the analysis of *'free'* phishing kits. They noticed that packages containing easy-to-deploy phishing websites often contained backdoors which exfiltrated the gathered information also to third parties and that 100% of the investigated kits were written in the PHP language.

In their PhishEye study, Han et al. [17] share insights into live phishing websites created by deploying phishing kits on honeypot domains. Using their sandboxed approach, they were able to lure phishers into installing phishing kits on their honeypot servers of which the behavior was closely monitored. The authors analyzed both phisher and victim actions on the phishing website, showed that phishing kits are only active for less than 10 days since their installation and that most of the victims share the same country of origin. During their 5 months analysis period (Sep 2015 – Jan 2016), they collected 643 unique phishing kits of which 74% were correctly installed by 471 distinct attackers. Additionally, they discovered that only 10 phishing kits loaded the resources directly from the website of the targeted organization.

Measurements on live phishing domains In recent work, Oest et al. [38] analyzed `.htaccess` files – commonly used on Apache Web servers – to capture the evasive behavior of phishers. These files allow phishers to protect themselves against anti-phishing or search engine crawlers. Their paper states that *deny IP* and *User-Agent* filters are the most prevailing blacklisting technologies, whilst the *allow IP* filter type is often used to target specific countries. Additionally, they proposed a new high-level classification scheme for phishing URLs that builds upon the work of Garera et al. [12]. This taxonomy categorizes phishing URLs into five categories with different hiding and lure strategies. We also used that taxonomy to classify the URLs detected by our measurements in Section 5.1.

The work closest to ours is from the same authors, who continued their research by investigating the end-to-end life cycle of phishing attacks in 2020. This work relied on the observation that a substantial proportion of phishing pages make requests for Web resources to the websites that the attackers impersonate [39]. A unique collaboration with a large payment provider enabled them to link such Web requests to

the phishing websites they are originating from. This gave the authors an in-depth look into phishing campaigns from the moment the attacker installs the phishing page to the moment victims disclose their credentials. They found that the average phishing attack spans 21 hours and that modern Web browsers display a warning for a detected domain after 16 hours. Oest et al. [39] called the gap between the launch of the attack and detection by anti-phishing bodies the *'golden hours'* of phishing, in which the attackers gather 38% of their phished credentials. As our work shared a similar goal – analyzing the end-to-end life cycle of a phishing campaign – we share a number of findings. Namely, the use of extensive use of server-side cloaking, victim-specific paths and the presence of MITM-proxies in phishing kits. Additionally, our conclusions regarding the duration of an average phishing attack are comparable. However, there are also notable differences. Their work is focused on one single organization and includes both HTTP and HTTPS traffic whereas our work focuses on the entire Dutch financial sector, but was limited to domains served over HTTPS only. Furthermore, they relied on the assumption that phishing domains load resources directly from the target website, whereas we discovered that only a negligible portion of domains in our analysis did so.

10 Discussion

Limitations Analyzing a phenomenon like phishing always brings its inherent limitations and so does this study. As all other work on this topic, our methodology is only able to capture part of the phishing landscape. We identify the following limitations:

We are aware of the fact that by our choice of methodology, we are limited to phishing domains secured by HTTPS connections only. Yet we believe, as 78% of all phishing in 2020 is delivered through HTTPS according to the APWG [16] and the fact that Oest et al. [39] concluded that phishing served over HTTPS was three times more effective, the effects of that concise decision to be limited. Also note that our approach results in our ability to identify type III and IV phishing domains only, and thus miss the three other types. Another limitation of this work is that we are limited to identifying known phishing kits. Phishing domains that do not match any of our predefined fingerprints are simply not marked as phishing. Besides these missing kits, phishers could also change the file names or structure of their phishing kits, which would also render our detection methodology less effective. However, the main advantage of phishing kits is that they are easy to deploy for any criminal that wants to go phishing. Therefor, we do not expect that phishers that deploy these kits are either capable or willing to make numerous changes each time they deploy a new phishing website. On the other hand, our fingerprinting methodology also has a detection advantage for websites that deploy certain cloaking strategies. As explained in Section 4.1, some phishing websites ban IP ranges or User-Agents known to be used by anti-phishing services through PHP scripts on the homepage, or show a different landing page depending on the country of origin. These methods make detection based on the characteristics of the page – e.g., login forms, bank icons, etc. – rather difficult. However, searching for known files – our fingerprints – on such domains bypasses these evasion methods and results in a robust detection of a phishing kit.

We started our crawling infrastructure three months before data collection started, which allowed us to carefully examine the domains missed by our crawler. As explained in Section 3.4, we created fingerprints based on source code of live phishing websites missed by our crawling during testing. So even without obtaining the actual phishing kit, we were able to create robust fingerprints.

Unfortunately, the largest limitation is in missing data we do not see. As explained in Section 5.1, many domain names do not contain references to bank names, but only use common words. Before data collection started, we added 78 of such words to our suspicious keywords list, but we have definitely missed some. As these domains did not reached the threshold set in our domain detector, they remain undetected. The validation with eCrime Exchange data in Section 7 showed that such domains are less prevalent in this anti-phishing repository and it is therefore important to include such words. We identify the validation with only one data source also as a limitation of our work, but leave validation with more datasets for future work.

Public policy takeaways Taking decisive action on phishing is complex. Ironically, the standardized notice-and-takedown (NTD) procedure, that banks generally outsource to the security industry, has resulted in a game of whack-a-mole, leaving the police chasing these criminals often empty handed. And, as concluded by Moore & Clayton [35] in 2017, website removal is only part of the answer to phishing, but is not fast enough to completely mitigate the problem. If and when phishing campaigns are reported to law enforcement agencies (LEA), phishing domains are often already taken down, making attribution of the actors behind phishing campaigns near to impossible. Therefore, implementing a system as presented in this paper would be very beneficial for LEA investigations.

With WhatsApp and text messages being a popular delivery mechanism [37], the interaction with victims has sped up, highlighting the need for early-stage detection even more. This paper presented a measurement methodology leveraging the increasing use of phishing kits and TLS certificates in the phishing scene to make early-stage detection possible. This would open a window where phishers have their phishing gear ready, but have not yet thrown out the bait. Our findings pinpoint clear choke points in using phishing kits in campaigns, which law enforcement agencies in turn might exploit for disruption before a takedown occurs. Our measurements of the life cycle of campaigns using phishing kits, shows a pattern

wherein a persistent time gap exists between domain registration, deployment and sending out the bait. This is a window of opportunity that can be used to take preventive action, when the campaign did not make any victims yet. Leveraging our methodology, kit fingerprints can be used to automate detection of domains where a kit is ready to be deployed. We show that the use of these kits is widespread in the Dutch phishing landscape and have found that distinct families of kits exist, wherein certain common characteristics are identified – likely because the source code of one kit has evolved into the next. When these characteristics relate to a vulnerability – e.g., the standard admin password is `'password'` and the control panel can be approached via a typical subdomain – this brings novel opportunities for automated exploitation for law enforcement purposes towards attribution rather than disruption. Having a clear picture of the popularity of phishing kits could assist LEA in prioritizing their anti-phishing efforts to dominant kits. Interventions – e.g., exploiting a vulnerability – on these kits would immediately impact a large portion of campaigns. Next, these shared traits can also be used to keep track of the phishing landscape. For instance, *uAdmin* allows for multiple domains contacting the same control panel, making in-depth analysis possible on these domains to find new, related campaigns or actors.

A system like ours could complement the threat intelligence process of many organizations, especially financial institutions that suffer from these attacks. Additional measurements in the landscape, can also be enriched by a repository of phishing kit fingerprints. Similar to repositories for malware fingerprints, the community – from hosting providers, to volunteers and researchers – can contribute their analyses on phishing kits, so to keep track of this pervasive phishing tactic. In turn, standardization on how to describe phishing kits, their tactics and detection methods is necessary before such an exchange can be successful. The creation of such a standard is a gap future work can fill. In the meantime, our system can be extended with (semi)automatic submission to anti-phishing services and block lists, which would hopefully lead to quicker responses.

11 Conclusions

In this paper we have presented the results of our investigation of the Dutch phishing landscape. We designed an empirical methodology to study phishing campaigns in the wild using phishing kit fingerprints. We leverage the fact that phishers are using TLS certificates to capture the end-to-end life cycle of phishing campaigns. We were able to find 1,363 confirmed phishing domains that deploy such kits, in a four months time period – with on average 31 phishing domains online every day, waiting for victims to arrive. Most of these domains are online for only 24 hours, but half of them (much) longer. External validation with APWG data has shown that our methodology has the potential to detect phishing websites swiftly and that it covers a complementary spectrum of phishing domains. Additionally, we show that attackers have increased their abilities to lure victims into disclosing their credentials by using decoy pages, which do not directly demand credentials from the victim but do so eventually. These decoy pages split the target organization from the organization impersonated on the phishing page, which allows for numerous possibilities for attackers. Referring to the target organization in the domain name is less prevailing, as regular words are more often used to trick victims into clicking on a phishing link. Through a combination of our analysis on the anatomy of phishing kits and the crawls of phishing domains in the wild, we demonstrate that the Dutch phishing landscape is less diverse than expected and that many phishers are building their campaigns on the same framework, *uAdmin*. The arrest of the developer of this framework in February, 2021, and the corresponding news coverage allows us to conclude that our findings are also useful outside the Netherlands, as uAdmin is actively used all around the world. Through both data analyses and a real-world example, we have reconstructed a timeline of the complete end-to-end life cycle of a typical phishing campaign – proving that phishers move fast. In turn, these fast moving campaigns require swift and decisive interventions. We believe the insights of this work will help LEA and intermediaries design faster responses to this ever evolving threat and we encourage them to do a similar analysis of their local phishing landscape.

References

[1] Rowland Atkinson and John Flint. Accessing hidden and hard-to-reach populations: Snowball research strategies. *Social research update*, 33(1):1–4, 2001.

[2] Banken.nl. Banken.nl: Marktaandeel, 2019. https://www.banken.nl/bankensector/marktaandeel.

[3] Hugo L. J. Bijmans, Tim M. Booij, and Christian Doerr. Inadvertently making cyber criminals rich: A comprehensive study of cryptojacking campaigns at internet scale. In *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pages 1627–1644. USENIX Association, 2019.

[4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[5] Aaron Blum, Brad Wardman, Thamar Solorio, and Gary Warner. Lexical feature based phishing url detection using online learning. In *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*, pages 54–60, 2010.

[6] Calidog. Certstream. https://certstream.calidog.io/.

[7] Calidog. Certstream-python. https://github.com/CaliDog/certstream-python.

[8] Robert B Cialdini. *Influence: The new psychology of modern persuasion*. Morrow, 1984.

[9] Marco Cova, Christopher Kruegel, and Giovanni Vigna. There is no free phish: An analysis of "free" and live phishing kits. In *2nd USENIX Workshop on Offensive Technologies, WOOT'08, San Jose, CA, USA, July 28, 2008, Proceedings*. USENIX Association, 2008.

[10] Let's Encrypt. Let's Encrypt - Free SSL/TLS Certificates. https://letsencrypt.org/.

[11] Evgeniy Gabrilovich and Alex Gontmakher. The homograph attack. *Commun. ACM*, 45(2):128, 2002.

[12] Sujata Garera, Niels Provos, Monica Chew, and Aviel D Rubin. A framework for detection and measurement of phishing attacks. In *Proceedings of the 2007 ACM workshop on Recurring malcode*, pages 1–8, 2007.

[13] Guang-Gang Geng, Xiao-Dong Lee, Wei Wang, and Shian-Shyong Tseng. Favicon - a clue to phishing sites detection. In *2013 APWG eCrime Researchers Summit*, pages 1–10. IEEE, 2013.

[14] Anti-Phishing Working Group. APWG - Unifying the global response to cybercrime. https://apwg.org/.

[15] Anti-Phishing Working Group. The APWG eCrime Exchange (eCX). https://apwg.org/ecx/.

[16] Anti-Phishing Working Group. Phishing Activity Trends Report: 2nd Quarter 2020, August 2020. https://docs.apwg.org/reports/apwg_trends_report_q2_2020.pdf.

[17] Xiao Han, Nizar Kheir, and Davide Balzarotti. Phisheye: Live monitoring of sandboxed phishing kits. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1402–1413. ACM, 2016.

[18] Crane Hassold. The Mobile Phishing Threat You'll See Very Soon: URL Padding, June 2017. https://info.phishlabs.com/blog/the-mobile-phishing-threat-youll-see-very-soon-url-padding.

[19] Google Inc. Certificate Transparency. https://www.certificate-transparency.org/.

[20] Google Inc. Safe Browsing - Google Safe Browsing. https://safebrowsing.google.com/.

[21] Luca Invernizzi, Kurt Thomas, Alexandros Kapravelos, Oxana Comanescu, Jean Michel Picod, and Elie Bursztein. Cloak of visibility: Detecting when machines browse a different web. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 743–758. IEEE Computer Society, 2016.

[22] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 569–586. ACM, 2017.

[23] Neal Krawetz. Looks Like It - The Hacker Factor Blog. http://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html.

[24] Brian Krebs. The Reincarnation of a Bulletproof Hoster, April 2016. https://krebsonsecurity.com/2016/08/the-reincarnation-of-a-bulletproof-hoster.

[25] Brian Krebs. Arrest, Raids Tied to 'U-Admin' Phishing Kit, February 2021. https://krebsonsecurity.com/2021/02/arrest-raids-tied-to-u-admin-phishing-kit/.

[26] Mohit Kumar. This Phishing Attack is Almost Impossible to Detect On Chrome, Firefox and Opera, April 2017. https://thehackernews.com/2017/04/unicode-Punycode-phishing-attack.html.

[27] Yukun Li, Zhenguo Yang, Xu Chen, Huaping Yuan, and Wenyin Liu. A stacking model using URL and HTML features for phishing webpage detection. *Future Gener. Comput. Syst.*, 94:27–39, 2019.

[28] Yun Lin, Ruofan Liu, Dinil Mon Divakaran, Jun Yang Ng, Qing Zhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and Jin Song Dong. Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[29] Pim Lindeman. Criminelen handelen op berichtenapp: 'Heb je geld? Ik heb een pistool voor 3k', August 2019. https://www.ad.nl/dossier-weekend/criminelen-handelen-op-berichtenapp-heb-je-geld-ik-heb-een-pistool-voor-3k~ab66bdd0/.

[30] Baojun Liu, Chaoyi Lu, Zhou Li, Ying Liu, Hai-Xin Duan, Shuang Hao, and Zaifeng Zhang. A reexamination of internationalized domain names: The good, the bad and the ugly. In *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018, Luxembourg City, Luxembourg, June 25-28, 2018*, pages 654–665. IEEE Computer Society, 2018.

[31] Stephen Lynch. OpenDNS Unveils 'NLPRank,' a New Model for Advanced Threat Detection, March 2020. https://umbrella.cisco.com/blog/opendns-unveils-nlprank-a-new-model-for-advanced-threat-detection.

[32] Samuel Marchal, Kalle Saari, Nidhi Singh, and N. Asokan. Know your phish: Novel techniques for detecting phishing sites and their targets. In *36th IEEE International Conference on Distributed Computing Systems, ICDCS 2016, Nara, Japan, June 27-30, 2016*, pages 323–333. IEEE Computer Society, 2016.

[33] Xavi Mendez. Wfuzz - The Web Fuzzer. https://github.com/xmendez/wfuzz/.

[34] Simon Migliano. The Dark Web is Democratizing Cybercrime, August 2018. https://hackernoon.com/the-dark-web-is-democratizing-cybercrime-75e951e2454.

[35] Tyler Moore and Richard Clayton. Examining the impact of website take-down on phishing. In *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit 2007, Pittsburgh, Pennsylvania, USA, October 4-5, 2007*, volume 269 of *ACM International Conference Proceeding Series*, pages 1–13. ACM, 2007.

[36] Namecheap. What payment methods do you accept for domain registrations?, December 2018. https://www.namecheap.com/support/knowledgebase/article.aspx/35/7/what-payment-methods-do-you-accept-for-domain-registrations.

[37] Betaalvereniging Nederland. Veel meer valse SMS-berichten, zogenaamd van banken. https://www.betaalvereniging.nl/actueel/nieuws/veel-meer-valse-sms-berichten-zogenaamd-van-banken/.

[38] Adam Oest, Yeganeh Safaei, Adam Doupé, Gail-Joon Ahn, Brad Wardman, and Gary Warner. Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In *2018 APWG Symposium on Electronic Crime Research, eCrime 2018, San Diego, CA, USA, May 15-17, 2018*, pages 1–12. IEEE, 2018.

[39] Adam Oest, Penghui Zhang, Brad Wardman, Eric Nunes, Jakub Burgis, Ali Zand, Kurt Thomas, Adam Doupé, and Gail-Joon Ahn. Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In *29th USENIX Security Symposium USENIX Security 20)*, 2020.

[40] PhishTank. PhishTank - Out of the Net, into the tank. https://www.phishtank.com/.

[41] The Spamhaus Project. The Spamhaus Project - The Top 10 Most Abused TLDs. https://www.spamhaus.org/statistics/tlds/.

[42] Emily Schechter. Evolving Chrome's security indicators. https://blog.chromium.org/2018/05/evolving-chromes-security-indicators.html.

[43] Selenium. WebDriver: Documentation for Selenium. https://www.selenium.dev/documentation/en/webdriver/.

[44] Camelia Simoiu, Ali Zand, Kurt Thomas, and Elie Bursztein. Who is targeted by email-based phishing and malware?: Measuring factors that differentiate risk. In *IMC '20: ACM Internet Measurement Conference, Virtual Event, USA, October 27-29, 2020*, pages 567–576. ACM, 2020.

[45] Choon Lin Tan, Kang-Leng Chiew, KokSheik Wong, and San-Nah Sze. Phishwho: Phishing webpage detection via identity keywords extraction and target domain name finder. *Decis. Support Syst.*, 88:18–27, 2016.

[46] Telegram. Telegram FAQ: So how do you encrypt data?, 2021. https://telegram.org/faq#q-so-how-do-you-encrypt-data.

[47] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Daniel Margolis, Vern Paxson, and Elie Bursztein, editors. *Data breaches, phishing, or malware? Understanding the risks of stolen credentials*, 2017.

[48] Ivan Torroledo, Luis David Camacho, and Alejandro Correa Bahnsen. Hunting malicious TLS certificates with deep neural networks. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, CCS 2018, Toronto, ON, Canada, October 19, 2018*, pages 64–73. ACM, 2018.

[49] Amber van der Heijden and Luca Allodi. Cognitive triaging of phishing attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1309–1326, Santa Clara, CA, August 2019. USENIX Association.

[50] Rolf Van Wegberg, Samaneh Tajalizadehkhoob, Kyle Soska, Ugur Akyazi, Carlos Hernandez Ganan, Bram Klievink, Nicolas Christin, and Michel Van Eeten. Plug and prey? measuring the commoditization of cybercrime via online anonymous markets. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1009–1026, 2018.

[51] Centraal Bureau voor de Statistiek. The Netherlands on the European scale: Internet, May 2019. https://longreads.cbs.nl/european-scale-2019/internet/.

[52] x0rz. phishing_catcher. https://github.com/x0rz/phishing_catcher.

[53] Penghui Zhang, Adam Oest, Haehyun Cho, Zhibo Sun, RC Johnson, Brad Wardman, Shaown Sarker, Alexandros Kapravelos, Tiffany Bao, Ruoyu Wang, et al. Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2021.