Inside a Phisher's Mind: Understanding the Anti-phishing Ecosystem Through Phishing Kit Analysis

Adam Oest*, Yeganeh Safaei*, Adam Doupé*, Gail-Joon Ahn*§, Brad Wardman†, Gary Warner‡
*Arizona State University, § Samsung Research, †PayPal, Inc., ‡Cofense, Inc.
{aoest, ysafaeis, doupe, gahn}@asu.edu, bwardman@paypal.com, gar@cis.uab.edu

***Abstract*—Phishing attacks are becoming increasingly prevalent: 2016 saw more phishing attacks than any previous year on record according to the Anti-Phishing Working Group. At the same time, the growing level of sophistication of cybercriminals must be understood for the development of effective anti-phishing systems, as phishers have extensive control over the content they serve to their victims. By examining two large, real-world datasets of phishing kits and URLs from 2016 through mid-2017, we paint a clear picture of today's anti-phishing ecosystem while inferring the higher-level motives and thought processes of phishers. We analyze the nature of server-side *.htaccess* filtering techniques used by phishers to evade detection by the security community. We also propose a new generic classification scheme for phishing URLs which corresponds to modern social engineering techniques and reveals a correlation between URL type and compromised infrastructure use. Our analysis identifies measures that can be taken by the security community to defeat phishers' countermeasures and increase the likelihood of a timely response to phishing. We discover that phishers have a keen awareness of the infrastructure used against them, which illustrates the ever-evolving struggle between cybercriminals and security researchers and motivates future work to positively impact online security.**

I. Introduction

Phishing is a type of social engineering attack that seeks to trick victims into disclosing account credentials or other sensitive information through a fraudulent message (e.g. e-mail) that leads to a website which impersonates a real organization. Attackers (phishers) then use the stolen data for their own monetary gain [1], [2]. The global volume of phishing attacks is on the rise: in 2016, the Anti-Phishing Working Group (APWG) recorded over 1.2 million total attacks, more than any previous year on record and a 65% increase over 2015 [3].

The behemoth scale of credential theft cannot be overstated. Between March 2016 and March 2017, malware, data breaches, and phishing led to 1.9 billion usernames and passwords being offered for sale on black market communities [4]. While phishing attacks are conceptually simple, they are difficult to effectively counter because phishers and anti-phishing entities are engaged in an endless cat-and-mouse game. The technological tools used by both are ever-evolving in response to the other's actions [5].

Phishing attacks are particularly damaging not only due to their prevalence, but because their impact extends beyond the individuals who are directly targeted. The organizations being impersonated in such attacks (such as financial institutions or e-mail providers) expend vast resources to minimize their losses and must work together with security firms and researchers to address the increasing level of sophistication being observed in phishing. This gives rise to an anti-phishing ecosystem comprised of many diverse entities working toward the same goal of reducing the billions of dollars of annual damage attributed to phishing [6].

In this paper, we portray the anti-phishing ecosystem as a whole. We leverage two unique datasets to identify the key players in this ecosystem and expose specific techniques that phishers employ to avoid detection while maximizing their return on investment. Ultimately, our goal is to obtain a clear understanding of the current ecosystem, including phishers, victims, and abuse-reporting entities. We propose immediate solutions to counter sophisticated phishing attacks and identify future research directions for evaluating and improving phishing countermeasures.

We first analyze a dataset of over 2300 real-world "phishing kits" (retrieved by Cofense, Inc. between Q1 and Q2 2016) to gain insight into the different server-side approaches that phishers take to evade existing phishing site detection infrastructure, specifically focusing on filtering directives found in *.htaccess* server configuration files. Many of these directives allow us to identify security organizations commonly targeted by phishers. Because phishing kits are ready-to-deploy, reusable packages used to carry out phishing attacks, we are also able to observe patterns in the distribution and adoption of such kits across multiple attacks.

We then use over 170,000 phishing URLs (submitted to the APWG during the first half of 2017) to identify the extent to which compromised infrastructure and domains are used for phishing. We propose an up-to-date URL classification scheme and take a novel approach to combine URL classification with domain age to fingerprint each phishing attack.

Analyzing the software and URLs of phishing kits allows us to not only understand the goals of phishers, but also reveals evasion techniques and allows each attack to be profiled. To date, no detailed insight into the server-side evasion techniques we discuss in Section IV has been published in the context of

978-1-5386-4922-0/18/$31.00 ©2018 IEEE

phishing, yet understanding these techniques can help anti-phishing entities identify and blacklist attacks more quickly and reliably [7]. Phishers also select the URLs that host their attack sites, either in whole or partial form. These URLs are often crafted to deceive victims [8], but they can also be formulated to evade detection. With minimal effort from the phisher, request filtering and cleverly formulated URLs can dramatically bolster the effectiveness of a phishing attack; we thus focus our study on these two areas.

Our immediate contributions can help organizations involved in the fight against phishing consider the dynamics within the entire anti-phishing ecosystem, understand what influences patterns in phishers' URL and hosting selections, and understand the nature and purpose of server-side filtering that phishers employ. Our findings are the precursors to a larger study that we will conduct to measure the true effectiveness of phishers' evasion techniques, motivated by the potential for a more effective and timely incident response by anti-phishing entities to ultimately improve the security of potential phishing victims.

II. Phishing Attack Anatomy

To make a positive impact on the fight against phishing, we must first familiarize ourselves with the nature of the phishing attack, tools at the disposal of phishers, as well as the industry's defenses. We must then understand how phishers respond to those defenses so that we can think ahead to what might come next.

A. The Classic Phishing Attack

The stages of a typical phishing scenario are illustrated in Figure 1. First, prior to involving any victims, an attacker spoofs a website by copying its look and feel such that it is difficult for an average user to distinguish between the legitimate website and the fake one (0). This can be done using a phishing kit, as discussed in Section II-D. Next, the attacker sends messages (such as spam e-mails) to the user, leveraging social engineering to insist that an action is needed [9] and lures the user to click on a link to the phishing site (1). If the victim is successfully fooled, he or she then visits the site and submits sensitive information such as account credentials or credit card numbers (2). Victims will often be shown a reassuring confirmation message to minimize suspicion of the attack after the fact. Finally, the phishing site transmits the victim's information back to the phisher (3), who will attempt to fraudulently use it for monetary gain [10] either directly (4a, 5a) or indirectly (4b, 5b).

B. Anti-Phishing Defenses

Phishing is a difficult problem to solve, as phishers are becoming increasingly convincing when tricking users into disclosing personal information. However, the industry has developed security mechanisms that, if used alongside proper phishing awareness training, can help users identify and avoid fake websites [11].

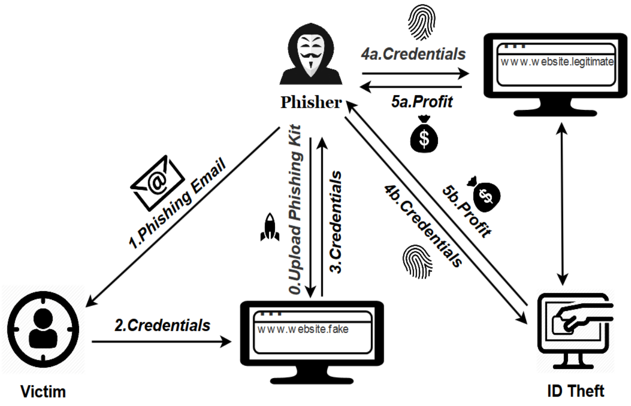

Fig. 1. The classic phishing attack.

E-mail messages are a common starting point for typical phishing attacks. Spam blacklists, heuristic filters, and reporting tools can be used to protect a user from potentially harmful messages [6], [12]. Targeted attacks against specific individuals and organizations, known as *spearphishing*, are increasing in pervasiveness and sophistication. Early identification of suspicious behavior in communication can help thwart this type of custom-tailored social engineering [13], [14].

In web browsers, using HTTPS with extended validation (EV) SSL certificates and showing security indicators such as the green lock or a user selected image are techniques that website owners and browser developers can employ to positively identify sites as legitimate [15] and establish user trust, thus distinguishing known sites from attack sites. Although phishers are now commonly using standard SSL certificates (e.g. via free services such as *LetsEncrypt* [16]), EV certificates require business verification and are thus harder to obtain fraudulently. On the client side, native browser blacklists are the first line of defense, as these are enabled by default in major web browsers. Blacklists prevent phishing content from being displayed to the user and instead generate a warning. Heuristic filter suites such as toolbars or antivirus programs can also be used for similar protection [17]. Such tools can combine various data sources, including URL content, domain age, search engine rankings [18], and page content [19]. Under the hood, security companies also use a sophisticated network of systems to track and respond to phishing threats (e.g. by updating blacklists). The effectiveness of these technical defenses hinges on early detection, which might be thwarted by a clever phisher as we discuss in Section IV.

In the case of credential theft, even if a phishing attack is carried out successfully, victim organizations can employ multi-factor authentication schemes in an effort to mitigate malicious login attempts [4].

C. Phishing Kits

A phishing kit is a unified collection of tools used to deploy a phishing site on a web server [20]. Some phishing kits are closely held by their creators, while others are offered as part of the cybercrime-as-a-service economy [21]. Certain criminals specialize in creating and selling phishing kits and will even accept custom requests for kit creation [22]. Kit creators compete based on the believability, ease of use, or perceived security of their kits. Other criminal service providers sell or barter to provide pre-hacked web servers (sometimes called "shells" or "cpanels" in criminal marketplaces). Still others offer lists of spam recipient e-mails and tools for sending the e-mails [1]. This lowers the barrier to entry, allowing criminals with very minimal technical skills or limited capabilities in English to become successful phishers [23]. The phisher can simply buy a kit, customize it by replacing the destination e-mail address, upload, and unzip the kit on a pre-hacked web server. The phisher then loads a pre-written message and a list of target e-mails into his or her spamming tool, hits "send," and waits for stolen credentials to arrive in his or her inbox.

Basic components of a phishing kit include a template that mimics the design of the website being impersonated, server-side code to capture and send submitted data to the phisher, and optionally code to filter out unwanted traffic or implement other countermeasures against the anti-phishing community. Such countermeasures might include URL shortening or redirection, URL randomization, or code obfuscation [24].

D. *Deploying a Phishing Scam*

To carry out a traditional phishing scam, attackers first need to have access to a live web server to host the phishing site. In most cases, creating the site merely involves uploading a phishing kit archive to the server and extracting its contents to the desired path. Compromised infrastructure or free hosting solutions are particularly common hosting targets (as we discuss in Section V-C), because using an existing live URL bypasses the requirement to purchase a new domain name and saves phishers both time and money [25]. Otherwise, the phisher must also register a domain name that will point to the phishing content.

In the case of compromised infrastructure, the phisher gains access to upload malicious files to a web server he or she does not own by exploiting a known web vulnerability or by using default or stolen credentials to access administrative software running on the server [26]. Exploitation is often automated and results in the uploading of a shell script on the server which can then be used to remotely execute commands. Based on the dataset in Section IV-A, we found that Wordpress installations are a particularly common target for phishers. This can likely be attributed to the prevalence of Wordpress, the vast library of third-party extensions, and technically-unsavvy users who fail to install security patches.

Once the phishing site is online, phishers distribute its URL through means such as e-mail, social media, or direct messaging [27]. Messages are crafted to deceive the user and often convey a sense of urgency to encourage action [9].

The phishing campaign will remain online for some period of time during which the phishing site collects credentials from victims who fall for the scam and forwards them to the phisher. Eventually, the site will be blacklisted, abandoned by the phisher, or forcefully taken offline by the web host [28]. Security efforts aim to minimize the amount of time that passes between phishing site deployment and blacklisting or take-down [17].

III. Anti-Phishing Ecosystem

Phishers have extensive control over the configuration of phishing sites they deploy. As we established in Section II-D, this includes the location (URL and hosting provider) of the site, the software the site uses to display the malicious content and capture user input, and the deceptive messages distributed to victims. Each of these areas involves an extensive network of entities who are either exploited during the phishing attack, who seek to fight phishing, or who are adversaries. This gives rise to a complex anti-phishing ecosystem.

By combining our findings from Sections IV and V with previous research of individual parts of the ecosystem [4], [1], [29], [2], [23], we, for the first time, paint a picture of the ecosystem as a whole, expose a host of potential weak points throughout the ecosystem, and address these weaknesses.

A. *Ecosystem Components*

As shown in Figure 2, at the heart of the phishing attack lie the phisher (1), phishing message (2), victim user (3), and organization being impersonated (4). Without these basic components, there would exist no basis of trust between the victim user and organization and no means of exploitation by the phisher [9].

Users are prone to re-using the same credentials across different services [4]. This means that the damage from each successful phishing attack can potentially cause a chain reaction spanning multiple organizations. Thus, the organization directly targeted by the phisher (4) expands to a set of indirectly targeted organizations of interest to the phisher (5) that use the same authentication scheme (such as username and password). Given the risk of damage that arises as a result, the organizations implement mitigation strategies consisting of their own security teams, third-party anti-phishing vendors, or law enforcement (6a).

1) ***Phishing Content:*** Phishing websites are displayed to the user through a web browser, which necessitates browser-based defenses such as those discussed in Section II-B. To support these defenses, there exist organizations that maintain blacklists of known phishing sites, organizations that verify phishing reports, and native web browser functionality that checks the blacklists [17] and blocks known sites as a baseline protection against phishing attacks (6b). Consumer-oriented security firms (6c) also offer software for end users who want additional protection (e.g. antivirus and internet security tools). While the former three classes of organizations all contribute to the anti-phishing effort, they have different priorities and scopes of operation, and are thus worth distinguishing.

Because native browser blacklists (discussed in Section IV-C5) accept user phishing reports, a community of

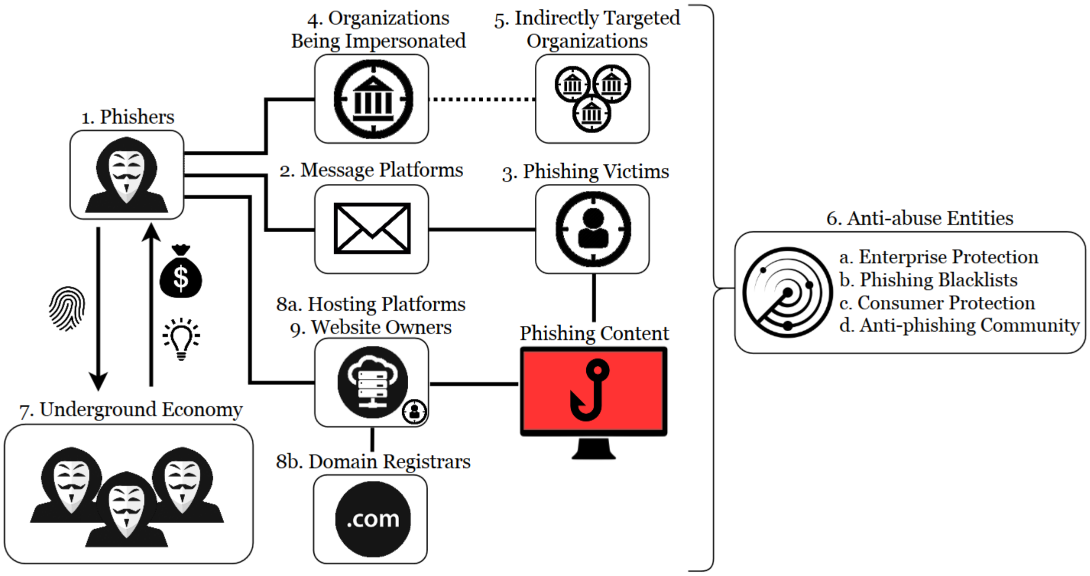

Fig. 2. Components of the anti-phishing ecosystem.

savvy web users and researchers joins the fight against phishing (6d). These groups gave rise to organized community-driven efforts to list and confirm phishing sites, such as PhishTank or the APWG [30].

Phishing content itself often stems from phishing kits, which can be obtained through forums or dark web communities that fuel cybercrime (7). Credentials stolen through successful attacks are sold by phishers via illicit underground economies [1] which in turn yield tools and motivation for future attacks.

*2) **Hosting Infrastructure**:* The hosting platforms (8a) on which phishing sites get deployed fall into two main categories: those controlled directly by the phisher and abused to carry out attacks, and those belonging to legitimate web sites that get hijacked by the phisher [24]. In the case of the former scenario, domain registrars (8b), both paid and free, can be the subject of further abuse through malicious domain registrations. In the case of the latter, the web site owner (9) may suffer collateral damage, such as disruption of regular business operations or loss of productivity as a result of an incident response from the hosting provider or one of the security vendors previously discussed (6a). The hosting provider may also make efforts to reduce hijacking through diligent patching or intrusion detection.

*3) **Message Distribution**:* Phishers require a communication channel to initiate their scam. Major e-mail providers and social media networks inevitably capture a large volume of phishing messages (2) through their platforms [31], thus they are dragged into the ecosystem once they start dedicating resources to protect their users. E-mail providers such as Gmail check incoming messages, mark them as spam or alert the user if malicious content is found, and forward abuse reports of identified phishing URLs to concerned entities (6a, 6b) [4], [32]. With the rapidly changing state of social media and web browser landscape [33], we expect phishers to develop innovative ways to bypass the protection available in traditional communication channels such as e-mail.

B. Observations

We have presented the first high-level overview of the different entities involved in today's anti-phishing ecosystem as well as the services exploited by phishers to carry out their work. Because phishers are likely to gravitate toward whatever tools best facilitate their attacks, beyond the entities already listed, the ecosystem can evolve in response to new hosting or distribution platforms for phishing content and continued innovation in evasion techniques implemented in phishing kits. In the following sections we show how the phishing kits and URLs we studied were able to reveal such extensive information about the ecosystem's components.

IV. Analysis of Server-Side Filtering

Many phishing kits employ request filtering, which requires some set of conditions to pass (based on information contained in the HTTP request or server state) before the phishing site is displayed to the client. Filters are of particular interest to researchers as they allow us to gain insight into the organizations that a phishing kit is trying to evade, or the group of users being targeted, thus making it possible to fingerprint the kit. Filtering in phishing kits bears a similarity to web cloaking [7], a technique used by spam sites to serve malicious content to victims while showing seemingly benign content to web crawlers.

Denying requests through server-side filtering seems paradoxical at first glance, as the phisher's goal is to scam as many victims as possible. However, the phisher also wishes to evade detection, which is where filtering plays an important role. It is thus in the phisher's interest to serve the phishing content to a legitimate victim while denying access to search engines, security firms, researchers, and blacklist crawlers, all of which

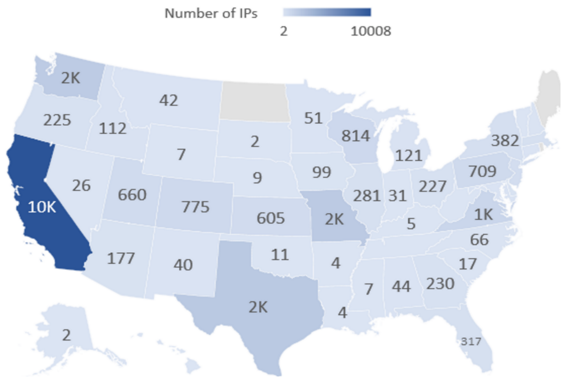

Fig. 3. Distribution of blocked IPs in the United States.

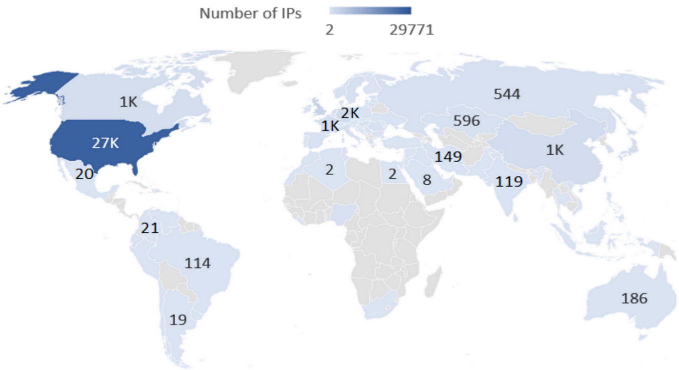

Fig. 4. Distribution of blocked IPs worldwide.

could trigger an anti-phishing response. Successfully blocking these entities decreases the likelihood of timely detection and blacklisting of the phishing site, ultimately increasing the phisher's return on investment.

Filtering can be implemented in various places, including server directives, server-side scripts (written in languages such as PHP and Python), or Javascript that runs in the user's web browser. The former two approaches are common and allow for very similar types of filtering with certain trade-offs as discussed below. The latter is an emerging technology seen in more sophisticated phishing kits, suitable for evaluation in a future work.

.htaccess files are used to supply configuration information for the Apache web server, the most common server software used for 44.99% of active web sites as of August, 2017 [33]. Such files are placed in directories containing web content and scripts. They allow configuration to be specified without root-level access to the server, which makes them possible to deploy without a complete breach of a system (gaining access to upload files to a public folder is generally sufficient). They are also simple for phishers to write and maintain through different iterations of a kit as they carry no dependencies. All this ease makes .htaccess files particularly appealing and they are therefore a common sight in phishing kits. .htaccess files lend themselves well to analysis as they are comprised nearly entirely of filtering directives in a homogeneous format, as opposed to arbitrary server-side code which can be written in many different ways, and can be obfuscated.

We examined a dataset of PHP scripts from phishing kits as part of our research and found them to implement filtering strategies in the same manner as .htaccess files. The main benefit of using scripts over server-side directives is the ability to track a user, something which has been previously studied [23], [24]. We thus focus our analysis on .htaccess files.

In the following sections we study a large dataset of .htaccess files in detail to reveal the nature and prevalence of request filtering techniques employed by phishers while identifying their underlying motivation. We propose methods to defeat each type of identified filter. We also synthesize a list of the anti-phishing organizations that phishers attempt to evade based on the contents of the .htaccess files. Finally, we consider metadata of the .htaccess files to plot age and discuss phishing kit re-use.

A. Dataset Overview

We examine a sample of 2,313 .htaccess files extracted from 1,794 live phishing kits hosted on 933 different domains. These kits were retrieved between January 1$^{\text{st}}$, 2016 and June 30$^{\text{th}}$, 2016 and provided to us by Cofense (formerly PhishMe), a company that focuses exclusively on phishing-related security solutions. In addition to the contents of the .htaccess files, the dataset contained the file modification date, date of retrieval, and URL of each kit.

1) ***Cleaning the Data:*** Because .htaccess files consist of plain text and generally contain one directive per line, they are straightforward to parse. We started by identifying duplicate files in the dataset by stripping comments and empty lines (we preserved inline comments as metadata for later analysis). This left 153 unique .htaccess files out of the total of 2,313 (6.6%). Of these unique files, 73 were seen only once, 66 appeared an average of 7.5 times, and 14 outliers appeared an average of 124 times. The outliers were attributable to a handful of kits with a large number of sub-directories all containing the same .htaccess file.

We then identified syntactic variations of directives with the same semantics (such as IP block rules) and iterated through each .htaccess file to obtain an aggregate overview of its filters.

B. Filter Types and Frequency

We discovered five different major types of filters used in the .htaccess files, distributed as shown in Table I. The most common *deny IP* filter takes a blacklist approach to block requests from specific IP addresses, partial IP addresses, or CIDR ranges; at least one such rule was present in 64% of the unique files and 95% of all files. Three other blacklist filters checked the *requester's hostname*, *referring URL*, or *user agent string* to deny requests matching the specified strings. On the opposite end of the spectrum, the *allow IP* filter took a whitelist approach to grant access only to specific IP addresses. In the case of our dataset, the *allow IP* filter was only present in a handful of files that performed geolocation

TABLE I
OCCURRENCE OF DIFFERENT FILTER TYPES IN .HTACCESS FILES.

Approach	Filter Type	All Files (2313)		Unique Files (153)	
		Filter Count	Files w/ Filter	Filter Count	Files w/ Filter
Blacklist	Deny IP	1,046,397	2194	234,125	98
Blacklist	Hostname	16,540	913	794	76
Blacklist	Referrer	4,177	572	976	48
Blacklist	User Agent	111,255	462	11,904	41
Whitelist	Allow IP	315,907	9	94,515	5

to restrict traffic to single countries. The four blacklist filters provide insight into the specific entities that phishers are trying to exclude. The whitelist filters instead reveal the location of the victims of the phishing scam.

In addition to these filters, we found that 30% of the unique .htaccess files and 61% of all files limited HTTP methods to *GET* and *POST*. For the sake of completeness, other less interesting directives included the specification an index script, disabling of server-level directory indexes, and allowing only certain file extensions.

C. Organizations of Interest

In this section we discuss the nature of the blocked IP addresses, hostnames, referring URLs, and user agents as we unmask why phishers selected them as part of their filtering strategy.

```
order allow,deny
allow from all
deny from 60.51.63.      # websense bandwidth waster
deny from 87.233.31.45   # bot rips way too fast
deny from 46.134.202.86

deny from paypal.com
deny from apple.com

RewriteEngine on

RewriteCond %{HTTP_REFERER} google\.com [NC,OR]
RewriteCond %{HTTP_REFERER} firefox\.com
RewriteRule .* - [F]

RewriteCond %{HTTP_USER_AGENT} ^googlebot
RewriteRule ^.* - [F,L]
```

Listing 1. Partial .htaccess file with all 4 blacklist filters and real comments left by phishers.

*1) **IP Address**:* We identify the entity targeted by each IP address through a variety of techniques: performing a reverse DNS lookup to obtain the hostname, visiting the IP, querying an ISP or IP geolocation database (we used IP2Location and GeoLite2, respectively), and manually interpreting the comments left behind by phishers. For some 4,300 IP addresses out of the total of 29,971 unique blacklisted IPs we extracted, phishers included a comment describing the entity believed to be tied to the address. These comments could be found in 23% of the files in the dataset. Examples of IP filtering as well as comments are found in Listing 1.

For the purpose of our analysis, we combined all of these techniques to obtain as much information as possible about each IP address. For each IP address in the dataset, we recorded the frequency as well as the associated entity. We categorized these entities based on their primary business type.

Our analysis of the data revealed that the phishers developing these .htaccess files focus heavily on blocking requests from web hosts, web crawlers, and internet service providers, with a secondary focus on security companies, universities, and organizations involved in DNS administration. Per the GeoLite2 database, over 90% of the unique IP addresses in our dataset were located in the US, with approximately half originating in tech-heavy California as shown in the maps in Figures 3 and 4. Areas with concentrated IP blocks coincide with the headquarters and datacenters of major organizations involved in internet security.

Avoiding traffic from these entities is important to phishers because, for instance, detection by a web host might lead to deactivation of the platform hosting the phishing site or identification of the person behind the attack [28]. Detection by a search engine crawler or security company would likely result in blacklisting of the phishing content, which could terminate the phishing campaign before the phisher finishes his or her work [24]. Generally speaking, the phisher does not want anyone but the victims to be able to access the phishing page.

We can conclude that phishers make a considerable effort to identify and attempt to bypass the anti-phishing infrastructure being used against them. While this dataset does not allow us to make any measurements of the effectiveness the phishers' evasion efforts, the security industry can regardless respond by using a diverse and ever-changing network of systems and IP addresses.

*2) **Hostname**:* By total count, filtering by hostname (of the IP address of the user making the HTTP request) was the least common filter type in our dataset. This rarity is likely because such filters require the server to perform a reverse DNS lookup for every HTTP request which is costly in terms of time and could impact the availability of the phishing site. However, nearly half of the unique .htaccess files contained at least one hostname filter, suggesting that phishers trust their effectiveness.

Hostname filters showed a heavy bias toward the victim organizations as well as anti-phishing organizations, and also included some antivirus vendors. Some were designed to match keywords that might be present in a security-related hostname, such as "phish," "spam," or ".edu."

This filter can be evaded by ensuring that no *PTR* reverse DNS record is configured for the IP address accessing the kit, or that the record is not revealing.

TABLE II
MOST COMMONLY OBSERVED ENTITIES IN OUR .HTACCESS DATASET.

Freq.	Ecosystem Entity	Type
257	Google	Crawler Blacklist
96	PayPal	Victim Org.
91	Internet Identity	Security
81	Bit Defender	Security
49	McAfee	Antivirus
42	Forcepoint	Security
42	Mark Monitor	Security
39	Brand Protect	Security
37	Looking Glass Cyber	Security
35	AVG	Antivirus
34	Eset	Antivirus
33	Kaspersky	Antivirus
27	Firefox / Mozilla	Browser Blacklist
25	TrendMicro	Antivirus
22	Apple	Victim Org.
21	Symantec	Antivirus
21	Netcraft	Security
20	F-secure	Antivirus
19	Dr Web	Antivirus
15	Avast	Antivirus
14	Avira	Antivirus
14	ClamAV	Antivirus
12	Spamcop	Security
11	Yandex	Crawler
11	Comodo	Security
10	Microsoft	Blacklist
10	PhishTank	Security

*3) **Referrer**:* When a human makes a web request by following a link in a browser, the browser will typically transmit the URL of the referring web page in the *Referer* HTTP header [34] of the new request. For instance, if an employee at a security company were to manually verify a phishing URL by clicking on a link in an e-mail or internal database, the company's name may be revealed in the referring URL. Phishers can take advantage of this behavior by blocking requests containing certain referring URLs, as shown in Lines 12–14 of Listing 1.

We found that the referrer filters focused exclusively on antivirus companies, security companies, and blacklist providers. These filters merely contained the primary public domain of these companies (e.g. google.com or mcafee.com), suggesting that phishers may have been guessing rather than basing these filters on known referrers.

The anti-phishing industry can easily bypass such filtering by configuring browsers or crawlers used for phishing detection to never transmit referrer information or to transmit a benign-looking URL.

*4) **User Agent**:* The user agent string (defined in RFC 2616) identifies the software issuing the HTTP request on behalf of the user, such as a browser or robot [34]. In our dataset, user agent filters were used exclusively to block known web crawling and scraping software.

Lines similar to 16–17 in Listing 1 were commonly found in our dataset and seek to block the Google crawler. We identified no other references to the anti-phishing entities that we saw in prior filters. This absence suggests that although phishers certainly wish to prevent automated tools (such as *wget* or a software library) from fetching their sites, they perhaps do not know the specific user agent strings used by security infrastructure, or they know how trivially these can be changed. Regardless, because the string can be spoofed, phishing blacklist crawlers can and should frequently take advantage of user agent spoofing.

*5) **Combined Summary**:* In Table II, we summarize the most frequently observed entities in our dataset of 153 .htaccess files, specifically those with 10 more distinct appearances. We aggregated the data by assigning a unique identifier to each entity, then combined the total number of occurrences of each entity within each filter type. Many of the entities listed are of integral importance in the anti-phishing landscape. Of particular note are Google Safe Browsing and Microsoft SmartScreen, who operate the blacklists that natively protect Google Chrome, Safari, Firefox, Internet Explorer, and Edge, accounting for over 97% of global desktop traffic as of August, 2017 [35]. Similarly, it is no surprise that PayPal and Apple appeared in this list as these companies ranked second and third as the most targeted brands in our APWG dataset and were common victims in prior studies of phishing kits by Cova et al. [23] and Han et al. [24]. The large, user-driven anti-phishing community PhishTank saw a disproportionately low representation in .htaccess files, possibly due to the distributed nature of the community [30].

It is evident that phishers work hard to thwart anti-phishing efforts by evading detection to ultimately bypass the defenses offered by blacklists, security firms, and antivirus vendors. Furthermore, it is arguably eye-opening that such a clear picture can be painted from data created by and known to phishers. At the same time, the .htaccess data does show some cracks: filters in older kits (discussed in the following section) lag behind in the past, as many references are made to defunct companies or companies that have merged with others within the past two years. Measuring the true effectiveness of phishers' filtering techniques within the ecosystem would be an interesting research problem.

D. Phishing Kit Sharing and Age

We found that on average 87 days elapsed between the first and the last retrieval of the duplicate .htaccess files discussed in Section IV-A1. Furthermore, almost every kit was retrieved from a distinct domain. We can thus conclude that phishing kits see regular re-use, potentially as part of a single phisher's campaign.

Frequent phishing kit re-use is further supported by analyzing the modification date metadata of the .htaccess files. The majority of files were last modified over a year before deployment, as shown in Figure 5. Interestingly, a handful of kits dated as far back as 2009. Regular changes to the configuration of anti-phishing infrastructure would quickly render the filtering efforts in such kits obsolete.

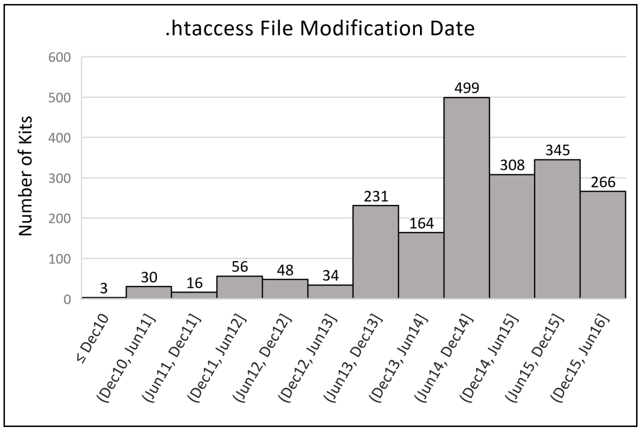

Fig. 5. .htaccess file modification date histogram.

V. Analysis of Phishing URLs

Cleverly-chosen URLs can be used by phishers to deceive victims or make a page appear benign in the absence of other context. In this section, we propose an up-to-date classification scheme reflecting the latest trends in phishing URLs motivated by evolving social engineering techniques of phishers in the context of URL creation. Moreover, we show that analyzing the domain age alongside the classification of these URLs can reveal information about the infrastructure being used to host phishing content. Patterns in phishing URLs have been shown to allow for automated classification and detection of phishing content, as studied in the work discussed in Section VI, and such classification can be enhanced by considering recent developments in URL crafting and understanding the intent behind each type of phishing URL.

A. Dataset Overview

The APWG has operated a clearinghouse of cybercrime data since 2004. The online eCrime eXchange database, available to APWG members, contains over 2.7 million real-world phishing attacks reported since 2015. Each entry contains the phish URL, organization targeted, date detected, and a confidence score. We focus on 172,620 URLs submitted during the first two quarters of 2017, which we fetched from the database on a daily basis and annotated with the domain registration date based on WHOIS data [36]. We further narrow our focus to “traditional” phishing URLs rather than social media URLs, which are also recorded by the APWG but differ substantially from a classic phishing attack in terms of deployment [31].

A key aspect of this dataset is that each URL is paired with the victimized organization, because such URLs are primarily submitted to the database by agents of the organizations themselves. This pairing allows us to reliably identify the presence of the brand within the URL’s hostname or path and addresses the limitation of datasets in prior work [23], [24].

After removing partial URLs without a hostname, social media URLs, and entries with a confidence score below 100%, we parsed each URL to generate additional attributes including hostname, top level domain, path, and subdomain level. We then removed hostnames as duplicates if they differed only in the presence of a hash or user ID in the subdomain, a common technique employed by phishers to evade blacklist hits [24]. This pruning left 66,752 unique hostnames. Finally, we used substring extraction [37] to identify common tokens in the URLs and manually classified them as *brand* if similar to the targeted brand name or *misleading* if related to account security and potentially misleading to a user (e.g. “secure,” “https,” “service”). Having classified individual tokens, we created parameters indicating their presence in the path and hostname of each URL, and in turn could classify the entire URL.

B. Classifying Phishing URLs

Using the attributes we added to the URL dataset, we propose a phishing URL classification in Table III that builds on the model of Garera et al. [8], an early and now outdated classification that identified four different types of hostnames in a phishing URL.

In an effort to comprehensively capture recent and emerging patterns in phishing URLs, we identify URLs as one of five mutually exclusive types by looking at both the hostname and path. Types I through IV are divided into a further two sub-types: the more common (a) for URLs recognizably containing the brand name, or (b) URLs containing misleading keywords. Both sub-types aim to trick a user to visit the URL. It was common for the (a) sub-type to contain a slight misspelling of the brand such as “paypol” instead of Paypal or “appel” instead of Apple; this enables top-level domain registration and typosquatting, and may thus also evade heuristic classifiers that expect the exact brand name [32] or specific keywords [38].

Apart from the introduction of the two sub-types to each, Type I, II, and III URLs are otherwise unchanged compared to Garera et al.’s classification [8]. Type IV URLs contain a deceptive top-level domain name registered by the phisher. Type V URLs are unintelligible in the absence of other metadata and contain a seemingly random hostname (which can be either a domain or IP address) and no brand or deceptive keywords.

Most browsers show the URL of the current page to the user, but the way the URL is displayed differs across browsers. Phishers may thus opt for a specific URL type depending on the targeted browser or platform. For example, the Google Chrome desktop browser highlights the top-level domain, so a Type IV URL might appear legitimate to unsuspecting users. A Type III URL would be suitable for a mobile browser which only shows the first several characters of the URL due to screen width limitations. For instance, a top level domain of *fakesite.com* would not visible in the displayed portion of a URL such as *www.paypal.com.signin.fakesite.com* when viewed on a small screen. A Type I, II, or III URL would be appropriate for display in e-mails, due to the possibility of long deceptive strings spanning much of the URL. Finally, a Type V URL can be advantageous for evading detection tools expecting specific patterns. We have thus identified not

TABLE III

PROPOSED HIGH-LEVEL CLASSIFICATION OF PHISHING URLS, WITH EXAMPLES FROM DATASET.

Type I *IP address as hostname, deceptive path contents*
(a) http://66.196.233.2/www.paypal.com/webscr.html?cmd=_login-run
(b) http://93.182.172.145/info/Verify.php?=&securessl=true
Type II *Random domain, deceptive path contents*
(a) http://resqplus.net/css/ssl/secure.paypal.com.au/au/cgi-bin/webscr/
(b) http://www.nae4ha2012.com/logos/login/secure.my.private-settings.support
Type III *Long, deceptive subdomain*
(a) http://statements.visa.com.upetkiti.be/cards/myvisa/transactionsphp
(b) http://https.secure.update.customer-update.extrasecure.profilecontinue.charityliberia.org/
Type IV *Deceptive top-level domain*
(a) http://change-paypal.com/ma/webapps/mpp/home/
(b) http://support-center-confirm.com/support/Payment-update-0.html
Type V *Unintelligible URL*
http://offto.net/5d8ucl/?action=redirect&nick=5d8ucl&m=1
http://69.93.204.33/∼startrac/cash69.html

only technical reasons for URL selection, but also motivations deeply rooted in social engineering.

In the entire dataset, we identified 156 Type I, 6,899 Type II, 4,186 Type III, 14,289 Type IV, and 41,213 Type V URLs. However, these numbers are far more meaningful when viewed alongside the age of the domain name within the latter four URL types, as discussed in the following section.

C. New vs. Compromised Infrastructure

Although URL content and domain age have historically been used in phishing site classifiers based on machine learning, other heuristic attributes based on page content and search engine metadata have proven to be much stronger indicators of a phishing attack [38]. Therefore, rather than using URL type and domain solely to identify phishing attacks, we propose a different use for these attributes: predicting the underlying hosting infrastructure, which can then be leveraged to mount an appropriate incident response by anti-phishing entities.

We found that 28.9% of the URLs in our dataset were reported within 1 month (30 days) of the domain registration, while another 53.3% of URLs had domains older than a year. The remainder was fairly uniformly split between 1 and 12 months. These findings are consistent with Hao et al.'s analysis of spam URLs distributed via e-mail [29] and show that the use of compromised infrastructure is still a significant problem today. While it is tempting to outright conclude that old domains belong to benign websites that have been compromised, a much stronger case can be made if we also consider URL classification along with the requirements to deploy each URL type.

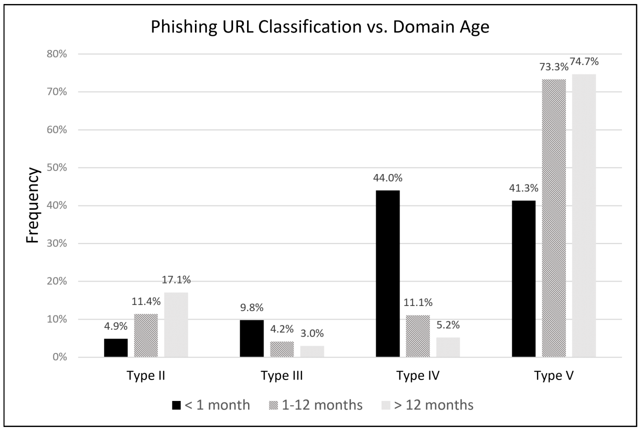

Fig. 6. Impact of domain age on URL type distribution (Type II: random domain with deceptive path; Type III: long deceptive subdomain; Type IV: deceptive top-level domain; Type V: unintelligible URL).

In Figure 6 we plot domain age versus URL type in our dataset and find that Type IV (deceptive TLD) URLs are the most common in newly-registered phishing domains, with Type III (long subdomain) URLs also occurring more frequently early on. Over time, the frequency of Type II (deceptive path) and Type V (unintelligible) URLs increases significantly while other types decline. Type I URLs are omitted as IPs do not have registration dates, unlike domains; they were also rare in the dataset as a whole.

Type I, II, and V URLs only rely on the folder structure of uploaded files, which can easily be controlled via phishing kits. Exploiting a web vulnerability to upload a phishing kit would grant the access necessary to deploy such URLs. On the other hand, a typical web hosting environment requires DNS changes for configuring Type III and Type IV domains; such access would require a larger-scale breach for deployment on a compromised system. We therefore hypothesize that older Type II and Type V phishing URLs generally represent compromised infrastructure, while newer Type III and Type IV URLs are more likely to be found on infrastructure deployed by phishers. While we do not verify this correlation, a future study involving a dataset with attributes such as the content and search engine rankings of each URL could be used to definitively classify the underlying infrastructure.

An intriguing case is our observation of a small proportion of Type IV URLs with domains older than 1 year. Because paid domain names must be renewed annually, it would not make economic sense for a phisher to pay renewal fees prior to using the domain name for the first time. After analyzing the TLD distribution of these URLs, we found that nearly all of these domains had free ccTLDs including *.tk*, *.ga*, *.ml*, *.cf*, and *.gq*. Such domains are owned by the registrar rather than the phisher and can often be renewed for free or re-acquired for free following expiration; a secondary advantage is of course anonymity since no payment details need to be provided. A handful of outliers included compromised legitimate websites that happen to contain a brand name, such as *apple-medical.com*. Type IV URLs consisted of 41.8% *.com* domains (lower than the 47.0% for the entire dataset) and 14.6% of the aforementioned ccTLDs (higher than the 5.1% dataset average). It is also worth noting that these ccTLDs are administered by only a handful of commonly-abused registrars.

In this dataset, *.coms* dwarf the second most common single TLD, *.net*, which accounted for only 3.6% of URLs. Full TLD statistics are publicly available from the APWG [3].

VI. Related Work

The work most closely related to our analysis includes that of Thomas et al. [4], who carried out a comprehensive study of the underground credential re-use ecosystem. As part of their study, the authors briefly discuss web cloaking through IP address blacklists in .htaccess files found in phishing kits, but they do not delve into any specific details. Cova et al. [23] performed an analysis of phishing kits freely available through underground sources or left behind by phishers as archives on live sites. The authors focused on identifying backdoors in these free phishing kits and found trends in e-mail provider usage, victimization, drop technique, and URL type. While they considered URL obfuscation techniques implemented through PHP code as a blacklist deterrent, they did not analyze if the kits performed request filtering. Han et al. [24] collected a large dataset of phishing kits through a honeypot server in order to study the anatomy and timeline of phishing attacks in great detail, which is important in understanding how to best respond to a phishing attack.

Much prior research has studied URLs used for phishing attacks. Hao et al. [29] analyze the domain registration behavior of real phishers and reveal commonly abused registrars. Garera et al. [8] provide a high-level classification scheme for phishing URLs. Further studies developed effective machine learning techniques to perform the classification itself [18], [39], [38] in order to automatically identify malicious websites on a large scale. Such systems are among those that respond to phishing content reported to the security community [40]. With the boom of social media in recent years, shortened redirection URLs and social media links have also seen a prominent use in phishing, as studied by Chhabra et al. [31].

To the best of our knowledge, no prior work has analyzed *.htaccess files* in detail in the context of phishing. With respect to phishing URLs, we expand on the classification of Garera et al. [8] by introducing a new URL type frequently observed in our dataset and combine methodology originally proposed by Matsuoka et al. [36] to further identify the URLs deployed on compromised infrastructure. Our datasets are unique because they are based entirely on real-world, live phishing attacks, contain extensive metadata, and not directly available to the public.

VII. Discussion

Server-side request filtering can easily be deployed through .htaccess files placed inside distributable phishing kits, which spread quickly due to their appeal to phishers with limited programming knowledge. Fortunately, countermeasures exist for each filtering technique as discussed in Section IV.

As web browsers and anti-phishing technologies mature, phishers respond by adapting their URLs for maximum effectiveness against their victims while aiming to avoid detection. Automated phishing site classification systems should likewise adapt to detect more subtle phishing URLs with partial brand names, homographs, and deceptive keywords. The generic URL classification scheme that we propose addresses common modern-day social engineering techniques and can be used as a part of such automated systems. Specifically, while modern web browsers leverage native anti-phishing blacklists to protect users from known malicious URLs, classification based on the URL alone is not widely used. Our findings could be used to reduce the false positive rates of such classification by considering context such as the type of device being used.

Continuous monitoring of the most recent phishing URLs can further improve this classification over time by identifying changing trends. With the evolution of the use of social media platforms and URL shortener services in phishing [31], future analysis should also focus on how these new platforms are used alongside the traditionally-hosted phishing site.

The URL-based attributes that we have identified could be incorporated alongside existing detection techniques such as passive DNS in order to respond during earlier stages of a phishing campaign, or with greater confidence. Also, considering the age of the domain along with the URLs classification can help determine whether or not a legitimate website has been compromised and allow for an effective response by the hosting provider. Finally, we could leverage URL-based predictions about the intended phishing victims in an attempt to bypass some of the server-side filtering techniques discussed in Section IV.

Phishers heavily rely on compromised infrastructure to carry out their work, but they also obtain their own domains through key bottlenecks including *.com* registrars and free domain providers. Further efforts should be focused on quickly detecting and responding to malicious registrations, especially when no payment is required. With access to their own domains, phishers otherwise have full flexibility when it comes to crafting a deceptive URL.

A. Future Work

In a future study motivated by the findings from our analysis of .htaccess evasion techniques, we will measure how effectively key security organizations in the anti-phishing ecosystem (from Table II) implement countermeasures to phishers' attempts at evasion, with a focus on the real-world security of the end user. If anti-phishing tools do not effectively counter request filters, phishers can cause extensive damage to their victims without being detected or stopped in a timely manner. Our preliminary efforts in this area, which involve monitoring of traffic to dummy phishing sites, have shown that even simple request filtering is effective at delaying a response from the security community. We plan to expand this study to more complex and innovative combinations of filters targeting typical groups of victims. Delayed blacklisting can lead to costly damage as a result of the phishing attack, both to the victims and brands being phished.

B. Limitations

Our findings should be considered with certain limitations in mind. The APWG database, while large and maintained by security professionals, is skewed toward organizations [3] that directly or indirectly partner with the APWG. It contains malicious URLs reported by the community, which are a subset of all phishing attacks, and may be skewed toward URLs that are harder to detect using existing heuristics. Also, we trust the confidence levels assigned by APWG submitters as an indication that the phishing URL listed was a true positive, as in many cases phishing content has already been removed by the time a URL appears in the database, so we cannot re-verify it.

The Cofense dataset is a year older than the APWG dataset, though Cofense does use a brand-agnostic approach to verifying and documenting phishing attacks which is divorced from its customer list. While many or even most phishing sites may contain .htaccess files, once deployed an .htaccess file is not retrievable via a web request, and thus we have only analyzed those files which were found in phishing kit archives that could be retrieved from phishing sites (i.e. zip files that a phisher extracted onto a compromised server). This means that the .htaccess files ultimately deployed on phishing sites could differ from those retrieved, though this is partially negated by the sheer number of directives across our dataset.

Nevertheless, such datasets are highly relevant to the phishing landscape as they correspond to the most common victim organizations and therefore warrant future analysis by the research community. Given our finding that phishers' filtering techniques target the most prominent entities involved in the fight against phishing, future datasets built collaboratively could prove to be even more eye-opening.

VIII. Conclusion

We have studied server-side request techniques employed in phishing kits, which paint a picture of the anti-phishing ecosystem from the perspective of criminals. We observed that the ecosystem spreads far beyond just the victims and organizations being targeted by phishers, yet those phishers have a keen awareness of the tools being used against them by the security community. Phishers seek to maximize their return on investment by avoiding detection by tools they know of, increasing the volume of attacks by using compromised infrastructure when possible, and bolstering the effectiveness of attacks by registering highly-deceptive domain names (preferably at no cost) to trick their victims. Security researchers and all involved entities must likewise understand phishers and respond to thwart them before phishing methodology evolves further.

It is clear that phishers have a wide gamut of paths available to them when it comes to deploying an attack. Our study provides the building blocks for an enhanced, custom-tailored response to phishing attacks that can be aided by automated technologies. Identifying a URL as a phish is only a basic first step. By combining our URL classification scheme and analyzing domain age, we can profile not only *where* a phishing attack likely originated in terms of infrastructure, but also *why* that URL was chosen. Building on our .htaccess findings, a crawler that is able to bypass and profile server-side filtering efforts can then reveal information about *who was targeted* for a faster blacklist response time. Knowing this information will allow anti-phishing efforts to focus on the most effective response, potentially saving the ecosystem time and money while improving the security of the end user. In a future work, we will identify the effectiveness to each filtering technique and measure the timeliness of the ecosystem's response.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant 1703644. This work was also partially supported by a grant from the Center for Cybersecurity and Digital Forensics at Arizona State University.

References

[1] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking fraudulent accounts: The role of the underground market in Twitter spam and abuse." in *USENIX Security Symposium*, 2013, pp. 195–210.

[2] T. Holz, M. Engelberth, and F. Freiling, "Learning more about the underground economy: A case-study of keyloggers and dropzones," *Computer Security–ESORICS 2009*, pp. 1–18, 2009.

[3] "Anti-Phishing Work Group: APWG Trends Report Q4 2016," (Date last accessed 23-August-2017). [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q4_2016.pdf

[4] K. Thomas, F. Li, A. Zand, J. Barrett, J. Ranieri, L. Invernizzi, Y. Markov, O. Comanescu, V. Eranti, A. Moscicki, D. Margolis, V. Paxson, and E. Bursztein, Eds., *Data breaches, phishing, or malware? Understanding the risks of stolen credentials*, 2017.

[5] B. Liang, M. Su, W. You, W. Shi, and G. Yang, "Cracking classifiers for evasion: A case study on Google's phishing pages filter," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 345–356.

[6] T.-C. Chen, T. Stepan, S. Dick, and J. Miller, "An anti-phishing system employing diffused information," *ACM Transactions on Information and System Security (TISSEC)*, vol. 16, no. 4, p. 16, 2014.

[7] L. Invernizzi, K. Thomas, A. Kapravelos, O. Comanescu, J.-M. Picod, and E. Bursztein, "Cloak of visibility: Detecting when machines browse a different web," in *Proceedings of the 37th IEEE Symposium on Security and Privacy*, 2016.

[8] S. Garera, N. Provos, M. Chew, and A. D. Rubin, “A framework for detection and measurement of phishing attacks,” in *Proceedings of the 2007 ACM Workshop on Recurring Malcode*, ser. WORM ’07. New York, NY, USA: ACM, 2007, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/1314389.1314391

[9] R. Dhamija, J. D. Tygar, and M. Hearst, “Why phishing works,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’06. New York, NY, USA: ACM, 2006, pp. 581–590. [Online]. Available: http://doi.acm.org/10.1145/1124772.1124861

[10] A. Emigh, “ITTC report on online identity theft technology and countermeasures 1: Phishing technology, chokepoints and countermeasures,” *Radix Labs*, Oct 2005.

[11] W. Yang, A. Xiong, J. Chen, R. W. Proctor, and N. Li, “Use of phishing training to improve security warning compliance: Evidence from a field experiment,” in *Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp*, ser. HoTSoS. New York, NY, USA: ACM, 2017, pp. 52–61. [Online]. Available: http://doi.acm.org/10.1145/3055305.3055310

[12] S. Duman, K. Kalkan-Cakmakci, M. Egele, W. Robertson, and E. Kirda, “Emailprofiler: Spearphishing filtering with header and stylometric features of emails,” in *Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual*, vol. 1. IEEE, 2016, pp. 408–416.

[13] Y. Han and Y. Shen, “Accurate spear phishing campaign attribution and early detection,” in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, ser. SAC ’16. New York, NY, USA: ACM, 2016, pp. 2079–2086. [Online]. Available: http://doi.acm.org/10.1145/2851613.2851801

[14] G. Stringhini and O. Thonnard, “That aint you: Blocking spearphishing through behavioral modelling,” in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2015, pp. 78–97.

[15] C. Jackson, D. Simon, D. Tan, and A. Barth, “An evaluation of extended validation and picture-in-picture phishing attacks.” Microsoft Research, January 2007. [Online]. Available: https://www.microsoft.com/en-us/research/publication/an-evaluation-of-extended-validation-and-picture-in-picture-phishing-attacks/

[16] A. Lukovenko, “Let’s Automate Let’s Encrypt,” *Linux J.*, vol. 2016, no. 266, Jun. 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969938.2969939

[17] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, “An empirical analysis of phishing blacklists,” 2009.

[18] L. Fang, W. Bailing, H. Junheng, S. Yushan, and W. Yuliang, “A proactive discovery and filtering solution on phishing websites,” in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2348–2355.

[19] Y. Zhang, J. I. Hong, and L. F. Cranor, “Cantina: A content-based approach to detecting phishing web sites,” in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW ’07. New York, NY, USA: ACM, 2007, pp. 639–648. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242659

[20] H. McCalley, B. Wardman, and G. Warner, “Analysis of back-doored phishing kits.” in *IFIP Int. Conf. Digital Forensics*, vol. 361. Springer, 2011, pp. 155–168.

[21] D. Manky, “Cybercrime as a service: a very modern business,” *Computer Fraud & Security*, vol. 2013, no. 6, pp. 9–13, 2013.

[22] D. Birk, S. Gajek, F. Grobert, and A. R. Sadeghi, “Phishing phishers - observing and tracing organized cybercrime,” in *Second International Conference on Internet Monitoring and Protection (ICIMP 2007)*, July 2007, p. 3.

[23] M. Cova, C. Kruegel, and G. Vigna, “There is no free phish: An analysis of ”free” and live phishing kits,” in *Proceedings of the 2nd Conference on USENIX Workshop on Offensive Technologies*, ser. WOOT’08. Berkeley, CA, USA: USENIX Association, 2008, pp. 4:1–4:8. [Online]. Available: http://dl.acm.org/citation.cfm?id=1496702.1496706

[24] X. Han, N. Kheir, and D. Balzarotti, “Phisheye: Live monitoring of sandboxed phishing kits,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1402–1413.

[25] T. Moore and R. Clayton, “Examining the impact of website take-down on phishing,” in *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. ACM, 2007, pp. 1–13.

[26] D. Canali, D. Balzarotti, and A. Francillon, “The role of web hosting providers in detecting compromised websites,” in *Proceedings of the 22nd International Conference on World Wide Web*, ser. WWW ’13. New York, NY, USA: ACM, 2013, pp. 177–188. [Online]. Available: http://doi.acm.org/10.1145/2488388.2488405

[27] J. Hong, “The state of phishing attacks,” *Communications of the ACM*, vol. 55, no. 1, pp. 74–81, 2012.

[28] P. J. Nero, B. Wardman, H. Copes, and G. Warner, “Phishing: Crime that pays,” in *eCrime Researchers Summit (eCrime), 2011*. IEEE, 2011, pp. 1–10.

[29] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck, “Understanding the domain registration behavior of spammers,” in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 63–76.

[30] D. G. Dobolyi and A. Abbasi, “Phishmonger: A free and open source public archive of real-world phishing websites,” in *Intelligence and Security Informatics, 2016 IEEE Conference on*. IEEE, 2016, pp. 31–36.

[31] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru, “Phi. sh/$ocial: the phishing landscape through short urls,” in *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*. ACM, 2011, pp. 92–101.

[32] M. Khonji, Y. Iraqi, and A. Jones, “Enhancing phishing e-mail classifiers: A lexical url analysis approach,” *International Journal for Information Security Research (IJISR)*, vol. 2, no. 1/2, p. 40, 2012.

[33] “NetCraft August 2017 Web Server Survey,” Jul 2017. [Online]. Available: https://news.netcraft.com/archives/2017/08/29/august-2017-web-server-survey.html

[34] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, “Hypertext transfer protocol – HTTP/1.1,” United States, 1999.

[35] “Statcounter: Desktop browser market share worldwide,” http://gs.statcounter.com/browser-market-share/, 2017, [Online; accessed 20-Aug-2017].

[36] M. Matsuoka, N. Yamai, K. Okayama, K. Kawano, M. Nakamura, and M. Minda, “Domain registration date retrieval system of urls in e-mail messages for improving spam discrimination,” in *Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual*. IEEE, 2013, pp. 587–592.

[37] N. Provos, J. McClain, and K. Wang, “Search worms,” in *Proceedings of the 4th ACM Workshop on Recurring Malcode*, ser. WORM ’06. New York, NY, USA: ACM, 2006, pp. 1–8. [Online]. Available: http://doi.acm.org/10.1145/1179542.1179544

[38] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, “Cantina+: A feature-rich machine learning framework for detecting phishing web sites,” *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 2, pp. 21:1–21:28, Sep. 2011. [Online]. Available: http://doi.acm.org/10.1145/2019599.2019606

[39] A. Blum, B. Wardman, T. Solorio, and G. Warner, “Lexical feature based phishing url detection using online learning,” in *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*, ser. AISec ’10. New York, NY, USA: ACM, 2010, pp. 54–60. [Online]. Available: http://doi.acm.org/10.1145/1866423.1866434

[40] C. Whittaker, B. Ryner, and M. Nazif, “Large-scale automatic classification of phishing pages,” in *NDSS ’10*, 2010. [Online]. Available: http://www.isoc.org/isoc/conferences/ndss/10/pdf/08.pdf