PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques Against Browser Phishing Blacklists

Adam Oest*, Yeganeh Safaei*, Adam Doupé*, Gail-Joon Ahn*[§], Brad Wardman[†], Kevin Tyers[†]
*Arizona State University, [§] Samsung Research, [†]PayPal, Inc.
{aoest, ysafaeis, doupe, gahn}@asu.edu, {bwardman, ktyers}@paypal.com

***Abstract*—Phishing attacks have reached record volumes in recent years. Simultaneously, modern phishing websites are growing in sophistication by employing diverse cloaking techniques to avoid detection by security infrastructure. In this paper, we present *PhishFarm*: a scalable framework for methodically testing the resilience of anti-phishing entities and browser blacklists to attackers' evasion efforts. We use *PhishFarm* to deploy 2,380 live phishing sites (on new, unique, and previously-unseen *.com* domains) each using one of six different HTTP request filters based on real phishing kits. We reported subsets of these sites to 10 distinct anti-phishing entities and measured both the occurrence and timeliness of native blacklisting in major web browsers to gauge the effectiveness of protection ultimately extended to victim users and organizations. Our experiments revealed shortcomings in current infrastructure, which allows some phishing sites to go unnoticed by the security community while remaining accessible to victims. We found that simple cloaking techniques representative of real-world attacks— including those based on geolocation, device type, or JavaScript— were effective in reducing the likelihood of blacklisting by over 55% on average. We also discovered that blacklisting did not function as intended in popular mobile browsers (Chrome, Safari, and Firefox), which left users of these browsers particularly vulnerable to phishing attacks. Following disclosure of our findings, anti-phishing entities are now better able to detect and mitigate several cloaking techniques (including those that target mobile users), and blacklisting has also become more consistent between desktop and mobile platforms— but work remains to be done by anti-phishing entities to ensure users are adequately protected. Our *PhishFarm* framework is designed for continuous monitoring of the ecosystem and can be extended to test future state-of-the-art evasion techniques used by malicious websites.**

I. Introduction

Phishing has maintained record-shattering levels of volume in recent years [1] and continues to be a major threat to today's Internet users. In 2018, as many as 113,000 unique monthly phishing attacks were reported to the APWG [2]. Beyond damaging well-known brands and compromising victims' identities, financials, and accounts, cybercriminals annually inflict millions of dollars of indirect damage due to the necessity of an expansive anti-abuse ecosystem which serves to protect the targeted companies and consumers [3]. With an ever-increasing number of Internet users and services— in particular on mobile devices [4]— the feasibility of social engineering on a large scale is also increasing. Given the potential for lucrative data, phishers are engaged in a tireless cat-and-mouse game with the ecosystem and seek to stay a step ahead of mitigation efforts to maximize the effectiveness of their attacks. Although new phishing attack vectors are emerging (e.g. via social media as a distribution channel [5]), malicious actors still primarily deploy "classic" phishing websites [2]. These malicious sites are ultimately accessed by victim users who are tricked into revealing sensitive information.

Today's major web browsers, both on desktop and mobile platforms, natively incorporate anti-phishing blacklists and display prominent warnings when a user attempts to visit a known malicious site. Due to their ubiquity, blacklists are a user's main and at times only technical line of defense against phishing. Unfortunately, blacklists suffer from a key weakness: they are inherently reactive [6]. Thus, a malicious website will generally not be blocked until its nature is verified by the blacklist operator. Phishing sites actively exploit this weakness by leveraging cloaking techniques [7] to avoid or delay detection by blacklist crawlers. Cloaking has only recently been scrutinized in the context of phishing [8]; to date, there have been no formal studies of the impact of cloaking on blacklisting effectiveness (despite numerous empirical analyses of blacklists in general). This shortcoming is important to address, as cybercriminals could potentially be causing ongoing damage without the ecosystem's knowledge.

In this paper, we carry out a carefully-controlled experiment to evaluate how 10 different anti-phishing entities respond to reports of phishing sites that employ cloaking techniques representative of real-world attacks. We measure how this cloaking impacts the effectiveness (i.e. site coverage and timeliness) of native blacklisting across major desktop and mobile browsers. We performed preliminary tests in mid-2017, disclosed our findings to key entities (including *Google Safe Browsing*, *Microsoft*, browser vendors, and the *APWG*), and conducted a full-scale retest in mid-2018. Uniquely and unlike prior work, we created our own (innocuous) PayPal-branded phishing websites (with permission) to minimize confounding effects and allow for an unprecedented degree of control.

Our work reveals several shortcomings within the anti-phishing ecosystem and underscores the importance of robust, ever-evolving anti-phishing defenses with good data sharing. Through our experiments, we found that cloaking can prevent browser blacklists from adequately protecting users by *signif-*

icantly decreasing the likelihood that a phishing site will be blacklisted, or *substantially delaying blacklisting*, in particular when geolocation- or device-based request filtering techniques are applied. Moreover, we identified a gaping hole in the protection of top mobile web browsers: shockingly, mobile Chrome, Safari, and Firefox failed to show *any* blacklist warnings between mid-2017 and late 2018 despite the presence of security settings that implied blacklist protection. As a result of our disclosures, users of the aforementioned mobile browsers now receive comparable protection to desktop users, and anti-phishing entities now better protect against some of the cloaking techniques we tested. We propose a number of additional improvements which could further strengthen the ecosystem, and we will freely release the PhishFarm framework[1] to interested researchers and security organizations for continued testing of anti-phishing systems and potential adaptation for measuring variables beyond just cloaking. Thus, the contributions of this work are as follows:

- A controlled empirical study of the effects of server-side request filtering on blacklisting coverage and timeliness in modern desktop and mobile browsers.
- A reusable, automated, scalable, and extensible framework for carrying out our experimental design.
- Identification of actionable real-world limitations in the current anti-phishing ecosystem.
- Enhancements to blacklisting infrastructure after disclosure to anti-phishing entities, including phishing protection in mobile versions of Chrome, Safari, and Firefox.

II. Background

Phishing is a type of social engineering attack that seeks to trick victims into disclosing sensitive information, often via a fraudulent website which impersonates a real organization. Attackers (phishers) then use the stolen data for their own monetary gain [9], [10], [11]. Cybercriminals are vehement in their attacks and the scale of credential theft cannot be overstated. Between March 2016 and 2017, malware, phishing, and data breaches led to 1.9 *billion* usernames and passwords being offered for sale on black market communities [12].

A. Phishing Attacks

An online phishing attack consists of three stages: preparation, distribution, and data exfiltration, respectively. First, prior to involving any victims, an attacker deploys a spoofed version of a legitimate website (by copying its look and feel) such that it is difficult for an average user to discern that it is fake. This deployment can be done using a phishing kit, as discussed in Section II-B. Second, the attacker sends messages (such as spam e-mails) to the user (leveraging social engineering to insist that action is needed [13]) and lures the user to click on a link to the phishing site. 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. Finally, the phishing site transmits the victim's information back to the phisher, who will attempt to fraudulently use it for monetary gain either directly or indirectly [14].

Phishing attacks are ever-evolving in response to ecosystem standards and may include innovative components that seek to circumvent existing mitigations. A current trend (mimicking the wider web) is the adoption of HTTPS by phishing sites, which helps them avoid negative security indicators in browsers and may give visitors a false sense of security [15], [16]. At the end of 2017, over 31% of all phishing sites reported to the APWG used HTTPS, up from less than 5% a year prior [2]. Another recent trend is the adoption of redirection links, which allows attackers to distribute a link that differs from the actual phishing landing page. Redirection chains commonly consist of multiple hops [17], [18], each potentially leveraging a different URL shortening service or open redirection vulnerability [19]. Notably, redirection allows the number of unique phishing links being distributed to grow well beyond the number of unique phishing sites, and such links might thus better slip past spam filters or volume-based phishing detection [20]. Furthermore, redirection through well-known services such as *bit.ly* may better fool victims [5], though it also allows the intermediaries to enact mitigations.

Ultimately, phishers cleverly attempt to circumvent existing controls in an effort to maximize the effectiveness of their attacks. The anti-phishing community should seek to predict such actions and develop new defenses while ensuring that existing controls remain resilient.

B. Phishing Kits

A phishing kit is a unified collection of tools used to deploy a phishing site on a web server [21]. Kits are generally designed to be easy to deploy and configure; when combined with mass-distribution tools [9], they greatly lower the barrier to entry and enable phishing on a large scale [22]. They are part of the cybercrime-as-a-service economy [23] and are often customized to facilitate a specific attack [24]. In the wild, they are deployed on compromised infrastructure or infrastructure managed directly by malicious actors.

*1) **Cloaking**:* A recent study found that phishing kits commonly use server-side directives to filter (i.e. block or turn away) unwanted (i.e. non-victim) traffic, such as search engine bots, anti-phishing crawlers, researchers, or users in locations that are incompatible with the phishing kit [8]. Attributes such as the visitor's IP address, hostname, user agent, or referring URL are leveraged to implement these filtering techniques.

Similar approaches, known as *cloaking*, have historically been used by malicious actors to influence search engine rankings by displaying different web content to bots than human visitors [7]. Users follow a misleading search result and are thus tricked into visiting a site which could contain malware or adware. Relatively little research has focused on cloaking outside of search engines; our experiment in Section III is the first controlled study, to the best of our knowledge, that measures the impact of cloaking within phishing attacks.

[1] Framework details are available at https://phishfarm-project.com

TABLE I: Overview of market share and blacklist providers of major web browsers (* denotes browsers we tested).

Browser Name	*Phishing Blacklist Provider*	*Est. Market Share (Worldwide) [28], [4]*	
		7/2017	*7/2018*
Desktop Web Browsers			
Google Chrome*	*Google Safe Browsing (GSB)*	63.48%	67.60%
Mozilla Firefox*		13.82%	11.23%
Safari*		5.04%	5.01%
Internet Explorer (IE)*	Microsoft SmartScreen	9.03%	6.97%
Microsoft Edge*		3.95%	4.19%
Opera*	Opera	2.25%	2.48%
Others	Varies	2.43%	2.52%
Mobile Web Browsers			
Google Chrome*	GSB	50.07%	55.98%
Safari*	GSB	17.19%	17.70%
UC Browser	None	15.55%	12.49%
Samsung Internet	None	6.54%	5.12%
Opera* (non-mini)	Opera	5.58%	4.26%
Android Browser	None	3.41%	1.95%
Mozilla Firefox*	GSB	0.06%	0.31%
Others	Varies	1.60%	2.19%

C. Anti-phishing Ecosystem

Even though phishing attacks only directly involve the attacker, the victim, and the impersonated organization, a large amount of collateral damage is caused due to the abuse of a plethora of independent systems which ultimately facilitate the attack [25]. Moreover, credential re-use, fueled by the sale of credentials in underground economies [9], [26], causes phishing threats aimed at one organization to potentially affect others. Over time, an anti-phishing ecosystem has matured; it consists of commercial security vendors, consumer antivirus organizations, web hosts, domain registrars, e-mail and messaging platforms, and dedicated anti-phishing entities [27], [8]. These entities consist of enterprise firms that operate on behalf of victim organizations, clearinghouses that aggregate and share abuse data, and the blacklists which directly protect web browsers and other software.

D. Detecting Phishing

The distribution phase of phishing attacks is inevitably noisy, as links to each phishing site are generally sent to a large set of potential victims. Thus, the thousands of phishing attacks reported each month easily translate into messages with millions of recipients [29]. Detection can also occur during the preparation and exfiltration stages. As artifact trails propagate through the security community, they can be used to detect and respond to attacks. Detection methods include classification of e-mails [30], [31], analyzing underground tools and drop zones [10], identifying malicious hostnames through passive DNS [32], URL and content classification [33], [20], [34], [35], malware scanning by web hosts [36], monitoring domain registrations [26] and certificate issuance [37], and receiving direct reports. All of these potential detection methods can result in reports which may be forwarded to anti-phishing entities that power blacklists [6].

E. Browser Blacklists

The final piece of the puzzle in defending against phishing is the conversion of intelligence about attacks into the protection of users. Native browser blacklists are a key line of defense against phishing and malware sites, as they are enabled by default in modern web browsers and thus automatically protect even unaware users. In the absence of third-party security software, once a phishing message reaches a potential victim, browser blacklists are the only technical control that stands between the victim and the display of the phishing content. Studies have shown that blacklists are highly effective at stopping a phishing attack whenever a warning is shown [6]. Such warnings are prominent, but typically only appear *after* the blacklist operator's web crawling infrastructure verifies the attack; some are also based on proactive heuristics [6].

Today's major web browsers are protected by one of three different blacklist providers, as shown in Table I. While Google Safe Browsing (GSB) and SmartScreen are well-documented standalone blacklists, Opera does not publicly disclose the sources for its blacklist. Prior work by NSS Labs suggests that PhishTank and Netcraft are among Opera's current third-party partners [38]; this is supported by our experimental results. We know that blacklists are effective when they function as intended [39], but is this always the case? Do blacklists offer adequate protection against phishers' evasion efforts such as cloaking, or are phishers launching attacks with impunity? We focus on answering these questions in the rest of this paper.

III. Experimental Methodology

Our primary research goal is to measure how cloaking affects the occurrence and timeliness of blacklisting of phishing sites within browsers. On a technical level, cloaking relies on filtering logic that restricts access to phishing content based on metadata from an HTTP request. Filtering is widely used in phishing kits [8]; attackers aim to maximize their return on investment by only showing the phishing content to victims rather than anti-abuse infrastructure. If a phishing kit suspects that the visitor is not a potential victim, a 404 "not found", 403 "forbidden", or 30x redirection response code [40] may be returned by the server in lieu of the phishing page. Attackers could also choose to display benign content instead.

Prior studies of browser blacklists have involved observation or honeypotting of live phishing attacks [25], [6], [39], but these tests did not consider cloaking. It is also difficult to definitively identify cloaking techniques simply by observing live sites. Our experimental design addresses this limitation.

A. Overview

At a high level, we deploy our own (sterilized) phishing websites on a large scale, report their URLs to anti-phishing entities, and make direct observations of blacklisting times across major web browsers. We divide the phishing sites into multiple batches, each of which targets a different entity. We further sub-divide these batches into smaller groups with different types of cloaking. Once the sites are live, we report their URLs to the anti-phishing entity being tested, such that the entity sees a sufficient sample of each cloaking technique. We then monitor the entity's response, with our primary metric being the time of blacklisting relative to the time we submitted each report. We collect secondary metrics from web traffic logs of each phishing site. Our approach is thus empirical in nature, however it is distinct from prior work because we fully control the phishing sites in question, and, therefore, have ground truth on their deployment times and cloaking techniques.

All the phishing sites that we used for all of our experiments spoofed the *PayPal.com* login page (with permission from PayPal, Inc.) as it appeared in mid-2017 (we discuss the hosting approach in Section IV-C2). Our phishing sites each used new, unique, and previously-unseen *.com* domain names, and were hosted across a diverse set of IP addresses spanning three continents. As part of our effort to reliably measure the time between our report and browser blacklisting for each site, we never reported the same phishing site to more than one entity, nor did we re-use any domains. To summarize, each experiment proceeded as follows:

1) Selecting a specific anti-phishing entity to test.
2) Deploying a large set of new, previously-unseen PayPal phishing sites with desired cloaking techniques.
3) Reporting the sites to the entity.
4) Measuring if and when each site becomes blacklisted across major web browsers.

We split our experiments into two halves: preliminary testing of 10 anti-phishing entities (mid-2017) and full follow-up testing of five of the best-performing entities (mid-2018). The latter test involved a large sample size designed to support statistically significant inferences. In between the two tests, we disclosed our preliminary findings to key entities to afford them the opportunity to evaluate any shortcomings we identified. We discuss our approach in more detail in the following sub-sections and present the results in Section V.

*1) **Targeted Web Browsers**:* We strove to maximize total market share while selecting the browsers to be tested. In addition to traditional desktop platforms, we wanted to measure the extent of blacklisting on mobile devices— a market which has grown and evolved tremendously in recent years [41]. We thus considered all the major desktop and mobile web browsers with native anti-phishing blacklists, as listed in Table I. Although we also identified a handful of other web browsers with blacklist protection, such as *CM Browser*, we did not test them due to their low market share [4].

*2) **Filter Types**:* We chose a set of request filtering techniques based on high-level cloaking strategies found in a recent study of *.htaccess* files from phishing kits [8]. Exhaustively measuring every possible combination of request filters was not feasible with a large sample size; we therefore chose a manageable set of filtering strategies which we felt would be effective in limiting traffic to broad yet representative groups of potential victims while remaining simple (for criminals) to implement and drawing inspiration from techniques found in the wild [7]. Table III summarizes our filter selections.

TABLE II: Entities targeted by our experiments.

Entity (Report Location)	*Report Type*	*URLs*
Full + Preliminary Tests		
APWG (reportphishing@apwg.org)	E-mail	40 (prelim.) 396 (full) per entity
Google Safe Browsing ([42])	Web	
Microsoft SmartScreen ([43])	Web	
PayPal (spoof@paypal.com)	E-mail	
PhishTank (phish-{username}@phishtank.com)	E-mail	
Preliminary Tests Only		
ESET ([44])	Web	40 per entity
Netcraft ([45])	Web	
McAfee ([46])	Web	
US CERT (phishing-report@us-cert.gov)	E-mail	
WebSense (asa@websense.com)	E-mail	
10 Entities Total		**2,380 URLs**

TABLE III: Cloaking techniques used by our phishing sites.

Cloaking Filter Name		*HTTP Request Criteria*
A	Control	Allow all
B	Mobile Devices	Allow if user agent indicates: Android or iOS
C	US Desktop GSB Browsers	Allow if IP country (is/is not) US and user agent indicates: Chrome, Firefox, or Safari; and Windows, Macintosh, or Linux; and not Opera, IE, or Edge; and not iOS or Android
D	Non-US Desktop GSB Browsers	
E	Block Known Security Entities	Allow if user agent, referrer, hostname, and IP: not known to belong to a security entity or bot
F	“Real” Web Browsers	Allow all; content retrieved asynchronously during JavaScript onload event

It would not be responsible of us to disclose the exact conditions required of each filter, but we can discuss them at a high level. *Filter A* served as our control group; our expectation was for every site in this group to be blacklisted at least as quickly as other sites. *Filter B* sought to study how well mobile-only phishing sites are blacklisted, coinciding with the recent uptick in mobile users and phishing victims [28], [2]. *Filters C* and *D* focus specifically on desktop browsers protected by GSB, which span the majority of desktop users today. We also included geolocation, which while not as frequent in the wild as other cloaking types [8], is highly effective due to low detectability and characteristics of *spearphishing*. A secondary motivation was to see how well entities other than GSB protect this group. *Filter E* is the most elaborate, but it also directly emulates typical real-world phishing kits. It is based on top *.htaccess* filters from a large dataset of recent phishing kits [8]. This filter seeks to block anti-phishing entities by hundreds of IP addresses, hostnames, referrers, and user agents. Finally, *Filter F* only displays phishing content if the client browser can execute JavaScript, which may defeat simple script-based web crawlers. This filter is common in modern search engine cloaking [7] and further motivated by an ongoing study we are conducting of JavaScript use in real-world phishing sites. Although today’s phishing kits tend to favor straightforward filtering techniques (namely approaches such as *Filter E* or geolocation), growing sophistication and adoption of cloaking (and other types of evasion) is technically feasible and to be expected as a risk to the anti-phishing ecosystem.

*3) **Tested Entities**:* In addition to the blacklist operators themselves, major clearinghouses, and PayPal’s internal anti-phishing system, we wanted to test as many of the other types of anti-phishing entities discussed in Section II-C as possible. We started with a recently-published list of entities commonly targeted for evasion by phishers [8]. We then made selections from this list, giving priority to the more common (and thus potentially more impactful) entities in today’s ecosystem. We had to exclude some entities of interest, as not all accept direct external phishing reports and thus do not fit our experimental design. Also, we could not be exhaustive as we had to consider the domain registration costs associated with conducting each experiment. Table II summarizes our entity selections.

*4) **Reporting**:* When the time came for us to report our phishing sites to each entity being tested, we would submit the site URLs either via e-mail or the entity's web interface (if available and preferred by the entity). In all cases, we used publicly-available submission channels; we had no special agreements with the entities, nor did they know they were being tested. In the case of the web submission channel, we simply used a browser to submit each phishing site's exact URL to the entity. E-mail reports were slightly more involved, as the industry prefers to receive attachments with entire phishing e-mails in lieu of a bare URL. We thus used PayPal-branded HTML e-mail templates to create lookalike phishing messages. Each message contained a random customer name and e-mail address (i.e. of a hypothetical victim) within one of many body templates. We sent e-mails from unique accounts across five security-branded domains under our control.

Reports of real-world phishing sites might be submitted in the exact same manner by agents on behalf of victim brands. Our reporting approach is thus realistic, though many larger victim organizations contract the services of enterprise security firms, which in turn have private communication channels to streamline the reporting process.

B. Preliminary Tests

Our preliminary testing took place in mid-2017 and carried out the full experimental approach from Section III-A on a small scale. We will now detail the execution of each test.

One of our key goals was to minimize confounding effects on our experimental results. In other words, we did not want any factors other than our report submission and cloaking strategy to influence the blacklisting of our phishing sites. As part of this, we wanted to secure a large set of IP addresses, such that no anti-phishing entity would see two of our sites hosted on the same IP address. With the resources available to us, we were able to provision 40 web servers powered by Digital Ocean, a large cloud hosting provider [47] (we informed Digital Ocean about our research to ensure the servers would not be taken down for abuse [48]). Each server had a unique IP address hosted in one of the host's data centers in Los Angeles, New York, Toronto, Frankfurt, Amsterdam, or Singapore. Our batch size was thus 40 phishing sites per preliminary test, for a total of 400 sites across the 10 entities being tested. Within each batch, six sites used filter types *A* and *F*, while seven used filter types *B* through *E*.

We also wanted to mimic actual phishing attacks as closely as possible. We studied the classification of phishing URL types submitted to the Anti-phishing Working Group's eCrime Exchange in early 2017 and crafted our URLs for each of the 40 phishing sites while following this distribution as closely as possible [8], [49]. We registered only *.com* domains for our URLs, as the *.com* TLD accounts for the majority of real-world phishing attacks. In addition, we chose GoDaddy as our registrar, which is among the most-abused registrars by real phishers [1]. Furthermore, to prevent crawlers from landing on our phishing sites by chance (i.e. by requesting the bare hostname), paths were non-empty across all of our URLs.

TABLE IV: URL and filter distribution for each experiment.

Phishing URL Content *Sample URL*	*(Type [8], [49])*	*Qty.*	*Filters Used*
Full Tests			
Non-deceptive (random) *http://www.florence-central.com/logician/retch/*	(V)	396	A (66), B (66), C (66), D (66), E (66), F (66)
Preliminary Tests			
Non-deceptive (random) *http://receptorpeachtreesharp.com/cultivable/*	(V)	10	A (1), B (2), C (2), D (2), E (2), F (1)
Brand in Domain *http://www.https-official-verifpaypal.com/signin*	(IVa)	6	A (1), B (1), C (1), D (1), E (1), F (1)
Deceptive Domain *http://services-signin.com/login/services/account/*	(IVb)	6	
Brand in Subdomain *http://paypal1.com.835anastasiatriable.com/signin*	(IIIa)	6	
Deceptive Subdomain *http://services.account.secure.lopezben.com/signin*	(IIIb)	6	
Deceptive Path *http://simpsonclassman.com/paypa1.com/signin*	(II)	6	

Using URLs that appear deceptive is a double-edged sword: while it allows us to gain insight into how various URL types are treated by entities, it is also a factor which may skew blacklisting speed. However, we decided to proceed with this in mind as the purpose of the preliminary tests was to *observe* more than *measure*, given the use of a relatively small sample size per batch. Table IV shows the distribution of URL types and filters per batch of sites.

We registered the required domain names and finalized configuration of our infrastructure in May 2017. In July, we started our experiments by systematically deploying and reporting our phishing sites. For each day over the course of a 10-day period, we picked one untested entity at random (from those in Table II), fully reported a single batch of URLs (over the course of several minutes), and started monitoring blacklist status across each targeted web browser. Monitoring of each URL continued every 10 minutes for a total of 72 hours after deployment. Over this period and for several days afterward, we also logged web traffic information to each of our phishing sites in an effort to study crawler activity related to each entity. Prior empirical tests found blacklists to show their weakness in early hours of a phishing attack [6]. Our 72-hour observation window allows us to study blacklist effectiveness during this critical period while also observing slower entities and potentially uncovering new trends.

C. Responsible Disclosure

Our analysis of the preliminary tests yielded several security recommendations (discussed in Section VI). We immediately proceeded to disclose our findings to the directly impacted entities (i.e. browser vendors, the brand itself, and major blacklist operators) which we also intended to re-test. We held our first disclosure meeting with PayPal in August 2017. Following PayPal's legal approval, we also disclosed to Google, Microsoft, Apple, Mozilla, and the APWG in February 2018. Each meeting consisted of a detailed review of the entity's performance in our study, positive findings, specific actionable findings, and high-level comparisons to other entities. Our disclosures were generally positively received and resulted in close follow-up collaboration with Google, Mozilla, and the APWG; this ultimately resulted in the implementation of effective blacklisting within mobile GSB browsers and general mitigations against certain types of cloaking. We clearly stated that we would repeat our experiments in 4-6 months and thereafter publish our findings.

We did not immediately disclose to the blacklists powering Opera as we had not originally expected to have the resources to re-test a fifth entity. Additionally, given lesser short-term urgency with respect to the remaining entities (and in the absence of close relationships with them), at the time we felt it would be more impactful to disclose to them the preliminary findings *alongside* the full test findings. This approach ultimately allowed us to better guide our recommendations by sharing deeper insight into the vulnerabilities within the broader ecosystem. After the completion of the full tests, we reached out the all remaining entities via e-mail; all but PhishTank and Opera responded and acknowledged receipt of a textual report containing our findings. US CERT and ESET additionally followed up for clarifications once thereafter.

D. Full-scale Tests

We believed that key ecosystem changes resulting from our disclosures would still be captured by our original experimental design. Thus, we did not alter our retesting approach beyond increasing the scale to enable statistically significant observations. In addition, rather than using the URL distribution from the preliminary experiments, we solely used non-deceptive paths and hostnames (i.e. with randomly-chosen English words) in order to remove URL classification as a possible confounding factor in our cloaking evaluation.

We registered the required domains in May 2018 and initiated sequential deployment of our full-scale tests in early July. We re-tested all entities to which we disclosed. As our budget ultimately allowed for an additional entity, we decided to also include PhishTank for its promising performance in the preliminary test. Each of the resulting five experiment batches consisted of 396 phishing sites, evenly split into six groups for each cloaking technique. Our reporting method did not change, though we throttled e-mailing such that the reports were spread over a one-hour period. Reporting through Google and Microsoft's web interfaces spanned a slightly longer period of up to two hours due to the unavoidable manual work involved in solving required CAPTCHA challenges.

E. Sample Size Selection

For our full tests, we chose a sample size of 384 phishing sites for each entity. Our goal was to obtain a power of 0.95 at the significance level of 0.05 in a one-way independent ANOVA test, which, for each entity, could identify the presence of a statistically significant difference in mean blacklisting speed between the six cloaking filters. Based on Cohen's recommendation [50] and our preliminary test results, we assumed a *medium* effect size (f) of 0.25. We added 12 sites (2 per filter) to each experiment to serve as backups in case of unforeseen technical difficulties. All sites ultimately delivered 100% uptime during deployment, thus we ended up with an effective sample size of 396 per experiment. While the assumption of f=*0.25* introduced some risk into our experimental design, we accepted it given the high cost of registering a new *.com* domain for each site.

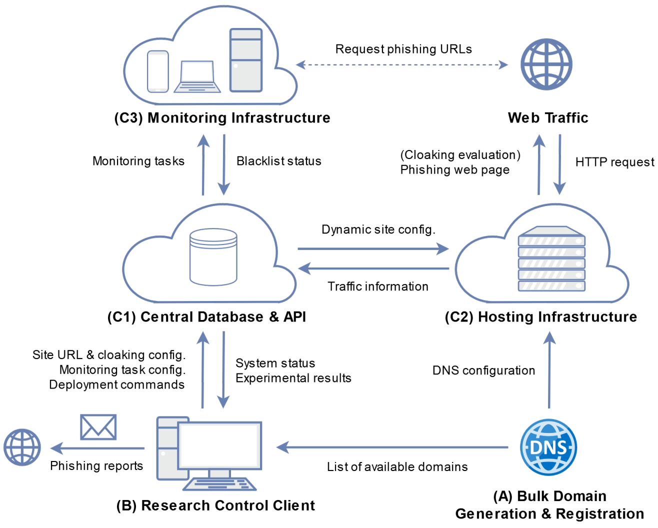


Fig. 1: PhishFarm framework architecture.

IV. PHISHFARM TESTBED FRAMEWORK

In order to execute our experimental design at scale, we designed *PhishFarm*: a comprehensive framework for deploying phishing sites, reporting them to anti-abuse entities, and measuring blacklist response. The framework operates as a web service and satisfies three important requirements: automation, scalability, and reliability. We eliminated as many manual actions as technically feasible to ensure that the difference between launching hundreds and thousands of sites was only a matter of minutes. Actual site deployment can thus happen instantly and on-demand. Apart from up-front bulk domain registration (IV-A) and reporting phishing through a web form, all framework components feature end-to-end automation.

The framework consists of five interconnected components as shown in Figure 1: bulk domain registration and DNS setup, stateless cloud web servers to display the phishing sites, an API and database that manages configuration, a client that issues commands through the API, and a monitoring system that regularly checks in which browsers each phishing is blacklisted. In total, we wrote over 11,000 lines of PHP, Java, and Python code for these backend components. We extensively tested each component— in particular, the hosting and monitoring infrastructure— to verify correct operation.

A. Domain and DNS Configuration

A core component of any phishing site is its URL, which consists of a hostname and path [49]. Our framework includes a script to automatically generate hostnames and paths per the experimental design. We then manually register the required domain names in bulk and point them to our cloud hosting provider's nameservers. Finally, we automatically create DNS records such that the domains for each experiment are evenly spread across the IP addresses under our control. We set up wildcard CNAME records [51] such that we could programmatically configure the subdomain of each phishing site on the server side. In total, registration of the 400 preliminary domains took about 15 minutes, while registration of the 1,980 domains for the full tests took 30 minutes; apart from the domain registration itself, no manual intervention is needed.

B. Research Client

We implemented a cross-platform client application to control the server-side components of the framework and execute our research. This client enables bulk configuration of phishing sites and cloaking techniques, bulk deployment of experiments, automated e-mail reporting, semi-automated web reporting, monitoring of status, and data analysis.

C. Server-Side components

The server-side components in our framework display the actual phishing sites based on dynamic configuration. They are also responsible for logging request data for later analysis and monitoring blacklist status.

*1) **Central Database and API**:* At the heart of our framework is a central API that serves as an interface to a database with our phishing site and system state. For each site, the database also maintains attributes such as date activated, reported, and disabled; blacklisting status; site and e-mail HTML templates; server-side and JavaScript request filtering code; and access logs. We interact with this API via the client whenever we define new sites or deploy sites as part of an experiment. All traffic to and from the API is encrypted.

*2) **Hosting Infrastructure**:* Our hosting infrastructure consists of Ubuntu cloud servers which run custom intermediate software on top of Apache. The intermediate software captures all requests and enables dynamic cloaking configuration and enhanced traffic logging. Each server obtains all of its configuration from the API, which means that the number of servers can flexibly be chosen based on testing requirements (40 in our case). When an HTTP request is received by any server, the server checks the request URL against the list of live phishing sites from the API, processes any cloaking rules, responds with the intended content, and logs the request. In order to quickly handle incoming requests, system state is cached locally on each server to minimize API queries.

We served all our phishing sites over HTTP rather than HTTPS, as this was typical among real-world phishing sites at the time we designed the preliminary tests [1]. While our framework supports HTTPS, we did not wish to alter the experimental design between tests. Both approaches generate potential artifacts that might benefit the anti-phishing ecosystem: unencrypted sites allow network-level packet sniffing, while encrypted sites leave an evidence trail at the time a certificate is issued. We mitigated the former risk to a large extent through the design of our monitoring system.

Our framework supports arbitrary, brand-agnostic web page content to be displayed for each phishing site. For our experiments, we hosted all resources (e.g. images and CSS) locally on each server to avoid confounding detection through external web requests to these files. In the wild, we have observed that sophisticated phishing kits follow a similar strategy to reduce detectability, though today's typical kits merely embed resources from the legitimate website.

*3) **Monitoring Infrastructure**:* The purpose of the monitoring system is to identify, at a fine granularity, how much time elapses before a reported phishing site is blacklisted (if at all). To obtain a clear picture of the early hours of blacklist response, we configured the monitoring system to access each live phishing URL at least once every 10 minutes in *each* target desktop browser (the shortest interval feasible given our resources). We check the blacklist status of each URL by analyzing a screenshot of the respective browser window; this is similar to the approach taken by Sheng et al [6]. We chose this approach for its universal applicability and lack of dependencies on browser automation libraries, which commonly force-disable phishing protection. For our purposes, it sufficed to check if the dominant color in each image [52] was similar to red (used in the warning messages of the browsers we considered). Although this image-based approach proved reliable and can be fully automated, it does not scale well because each browser requires exclusive use of a single system's screen.

To satisfy the scalability requirement, our monitoring system follows a distributed architecture with multiple autonomous nodes that communicate with the API; each node is a virtual machine (VM) that runs on a smaller set of host systems. Software on each VM points a browser to the desired URL via command line and sends virtual keystrokes, as needed, to close stale tabs and allow for quick page loads. During our experiments, we used three types of nodes: Mac OS X 10.12 VMs with Chrome, Safari, Opera, and Firefox; Windows 10 VMs with Edge and Internet Explorer (IE) 11; and Windows 8.1 VMs with IE 11. In total, the full experiments required 31 desktop nodes across four host machines (collectively with 18 Intel Core i7 CPU cores and 96 Gb of RAM) to deliver the required level of monitoring performance and redundancy. Each node informs the API of the browsers it supports; it then awaits a command consisting of a set of URLs for a target browser, and in real time, reports the monitoring results back to the API. We freshly installed the latest stable version of each browser at the time of each test and kept default security settings (or, in the case of IE and Edge, recommended settings when prompted).

We were unable to obtain such a large number of mobile devices, and official emulators at the time would force-disable blacklist warnings. Thus, we only tested mobile browsers hourly and relied on our observation that their behavior was tied to the behavior their desktop counterparts. We used a physical Samsung Galaxy S8 and Google Pixel phone to test mobile Chrome, Firefox, and Opera; and an iPhone 7 (preliminary) or 8 (full) to test mobile Safari. A future version of the framework could be improved to leverage Android VMs, in lieu of physical or emulated devices, to perform monitoring similar to that of desktop browsers.

TABLE V: Overview of crawler and blacklisting activity across all experiments.

	Phishing Sites Deployed		**Crawler(s) Attempted Retrieval**		**Successful Crawler Retrievals**		**Crawled Sites Blacklisted**		**Mean Time Before 1st Blacklisting**		**Mean Page Loads per Site**	
	w/ Cloaking	*w/o*	*w/*	*w/o*	*w/*	*w/o*	*w/*	*w/o*	*w/*	*w/o*	*w/*	*w/o*
Prelim.	340	60	294 (86.5%)	51 (85.0%)	156 (53.1%)	49 (96.1%)	124 (42.2%)	42 (82.4%)	173 min.	79 min	80	464
Full	1650	330	1333 (80.8%)	271 (82.7%)	818 (61.4%)	264 (97.4%)	306 (23.0%)	134 (49.4%)	238 min.	126 min.	162	334

Lastly, we wanted to ensure that the large number of requests from our monitoring system did not affect the blacklist status of our phishing sites (i.e. by triggering heuristics or network-level analysis [6]). Therefore, rather than displaying the phishing content in the monitoring system's browsers, we simply displayed a blank page with a 200 status code. While we had the option of querying the GSB API directly as a supplement to empirically monitoring the browsers it protects, we chose not to do so for the same reason. Finally, our monitors used an anonymous VPN service (NordVPN) in an effort to bypass any institution- and ISP-level sniffing.

D. Ethical and Security Concerns

Over the course of our experiments, we were careful in ensuring that no actual users would visit (let alone submit credentials to) our phishing sites: we only ever shared the site URLs directly with anti-phishing entities, and we sterilized the login forms such that passwords would not be transmitted in the event of form submission. In the hands of malicious actors with access to the necessary hosting and message distribution infrastructure, and with malicious modifications, our framework could potentially be used to carry out real and evasive phishing attacks on a large scale. We will thus not release the framework as open source software; however, we will share it privately with vetted security researchers.

Another potential concern is that testing such as ours could degrade the response time of live anti-phishing systems or negatively impact data samples used by their classifiers. Given the volume of live phishing attacks today [2], we do not believe that our experiments carried any adverse side-effects; the entities to which we disclosed did not raise any concerns regarding this. With respect to classifiers based on machine learning, effective methodology has already been proposed to ensure that high-frequency phishing sites do not skew training samples [20]. Nevertheless, it would be prudent for any party engaged in long-term or high-volume testing of this nature to first consult with any targeted entities.

V. Evaluation

PhishFarm proved to deliver reliable performance over the course of our tests and executed our experimental methodology as designed. Following the completion of all tests, we had collected timestamps of when blacklisting occurred, relative to the time we reported the site, for each of our phishing sites in each of the desktop and mobile browsers tested. This totaled over 20,000 data points across the preliminary and full tests, and had added dimensionality due to the various entities and cloaking techniques tested. Table V shows the breakdown of crawler and blacklisting activity on our phishing sites. Overall, we found that sites with cloaking were both slower and less likely to be blacklisted than sites without cloaking: in the full tests, just 23.0% of our sites with cloaking (which were crawled) ended up being blacklisted in at least one browser— far fewer than the 49.4% sites without cloaking which were blacklisted. Cloaking also slowed the average time to blacklisting from 126 minutes (for sites without cloaking) to 238 minutes.

However, closer scrutiny is required to make insightful conclusions about the conditions under which cloaking is effective, as we found that each anti-phishing entity exhibited distinctive blacklisting of different cloaking techniques alongside varying overall speed. For example, the mobile-only *Filter B* showed 100% effectiveness against blacklisting across all entities. On the other hand, the JavaScript-based *Filter F* was 100% effective for some entities, delayed blacklisting for others, but was in fact more likely to be blacklisted than *Filter A* by others still. In many cases, there was also a lack of blacklisting of *Filter A* sites (i.e. those *without* cloaking). Entities— in particular, the clearinghouses— at times failed to ultimately blacklist a reported site despite extensive crawling activity. Although it is possible that our direct reporting methodology led to some sites being ignored altogether, the exceptional performance of GSB with respect to *Filter A* shows that a very high standard is realistic. We detail each entity's behavior in Section V-F.

To provide meaningful measurements of entity performance with respect to different browsers and cloaking filters, we propose a scoring system in Section V-B which seeks to capture both the response time and number of sites blacklisted by each anti-phishing entity for each browser and/or filter. In addition, we summarize our findings visually in Figures 2 and 3 by plotting the cumulative percentage of sites blacklisted over time, segmented by each browser or by each cloaking technique, respectively. Lastly, we analyze web traffic logs to make observations about the distinctive behavior of each anti-phishing entity. Although much of our analysis focuses on the results of the large-scale full tests, we also make comparisons to the preliminary test data when appropriate.

A. Crawling Behavior

Of all sites that we launched during the preliminary and full tests, most (81.9%) saw requests from a web crawler, and a majority of sites therein (66.0%) was successfully retrieved at least once (i.e. bypassing cloaking if in use). In a handful of cases, our reports were ignored by the entities and thus resulted in no crawling activity, possibly due to volume or similarity

TABLE VI: Aggregate entity blacklisting performance scores in the full tests (colors denote subjective assessment: green— good, yellow— lacking, red— negligible blacklisting)

GSB		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.942	0	0.030	0.899	0.104	0.692	0.533	
	IE	0	0	0	0	0	0	0	
	Edge	0	0	0	0	0	0	0	
	Opera	0	0	0	0	0	0	0	
PB_f		0.970	0	0.031	0.953	0.106	0.712	**0.457**	S
TB_f *(min.)*		112	N/A	50	100	81	107	0.947	C

SmartScreen		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.005	0	0.005	0	0	0.009	0.004	
	IE	0.176	0	0.003	0	0.301	0.411	0.296	
	Edge	0.183	0	0.003	0	0.329	0.421	0.311	
	Opera	0	0	0.005	0	0	0	0	
PB_f		0.212	0	0.016	0	0.364	0.455	**0.038**	S
TB_f		548	N/A	2889	N/A	391	298	0.649	C

APWG		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.563	0	0.356	0	0.777	0	0.339	
	IE	0.113	0	0	0	0.626	0	0.246	
	Edge	0.129	0	0	0	0.761	0	0.297	
	Opera	0.242	0	0.185	0	0.545	0	0.262	
PB_f		0.576	0	0.344	0	0.803	0	**0.328**	S
TB_f		194	N/A	243	N/A	125	N/A	1	C

PhishTank		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.077	0	0	0	0	0.026	0.024	
	IE	0.096	0	0	0	0	0	0.026	
	Edge	0.085	0	0	0	0	0	0.028	
	Opera	0.074	0	0	0	0	0.024	0.032	
PB_f		0.106	0	0	0	0	0.136	**0.025**	S
TB_f		386	N/A	N/A	N/A	N/A	2827	0.467	C

PayPal		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.133	0	0.149	0.052	0.198	0.167	0.104	
	IE	0.102	0	0.040	0	0.074	0.137	0.140	
	Edge	0.123	0	0.056	0.046	0.193	0.163	0.160	
	Opera	0.119	0	0.029	0	0.191	0.120	0.143	
PB_f		0.167	0	0.172	0.078	0.288	0.182	**0.138**	S
TB_f		675	N/A	440	1331	1077	338	0.995	C

to previous reports; we mitigated this risk through the large sample size and discuss it in more detail in Section V-G. The distribution of crawler hits was skewed left, characterized by a small number of early requests from the entity to which we reported, followed by a large stream of traffic from it as well as other entities. Importantly, different cloaking techniques showed no significant effect on the time of the first crawling *attempt*; the median response time ranged from 50 to 53 minutes, from the time of reporting, for all filter types.

During our full experiments, *only* sites which were crawled were ultimately blacklisted. Generally, the crawling also had to result in successful retrieval of the phishing content for blacklisting to occur, though in 10 cases (all in the GSB experiment with *Filter D*), a site would be blacklisted despite a failed retrieval attempt; possible factors for this positive phishing classification are described in prior work [20].

B. Entity Scoring

For each entity tested, let U be the set of phishing URLs reported to the entity, let B be the set of browsers monitored, let T be the set of observed blacklisting times, let F be the set of cloaking filters used, and let MS denote the worldwide browser market share. Additionally, we define the value of the function *accessible(b, f)* to be true if and only if a phishing site with filter f is designed to be accessible in browser b.

For each URL-browser combination in $B \times U$, per Formula 1, we define a normalized performance score S_{URL_b} on the range $[0, 1]$, where 0 represents no blacklisting and 1 represents immediate blacklisting relative to the time reported. The score decays linearly over our 72-hour observation window (e.g. a site blacklisted after 36 hours would have $S_{URL_b} = 0.5$). We take the average of all these URL-browser scores for each browser-filter combination, as S_{bf}, per Formula 2.

TABLE VII: Formulas for aggregate scores (per test).

$$\forall\ b,\ URL \in B \times U,\ S_{URL_b} = \tag{1}$$

$$\begin{cases} 1 - \frac{T_{URL\ blacklisted_b} - T_{URL\ reported}}{T_{window}} & blacklisted\ in\ b \\ 0 & otherwise \end{cases}$$

$$\forall\ b,\ f \in B \times F,\ S_{bf} = \sum_{URL,\ F_{URL}=f} \frac{S_{URL_b}}{n} \tag{2}$$

$$\forall\ b \in B,\ S_b = \sum_{f,\ accessible(b,\ f)} \frac{S_{bf}}{n} \tag{3}$$

$$S = \sum_b \frac{MS_b}{MS_B} \cdot S_b \tag{4}$$

We further aggregate the S_{bf} scores to meaningfully summarize the protection of each browser by each entity: the browser score S_b, as per Formula 3, is the average of all S_{bf} for a given browser b, but limited to filters f *accessible* in b. To gauge the blacklisting of each cloaking filter, we report PB_f as the raw proportion of sites blacklisted in at least one browser for each respective filter. Note that PB_f will always be greater than or equal to any S_{bf} because the former is not reduced by poor timeliness; we thus additionally report TB_f, the mean time to blacklisting for each filter f (in minutes).

To capture the overall real-world performance of each entity, we average all S_b, weighted by the market share of each browser, to produce S, as per Formula 4. The scores S allow us to efficiently compare the performance between entities and would be useful in modeling long-term trends in future deployments of *PhishFarm*. We also include and the proportion of all sites crawled, C, to illustrate the entity's response effort.

We present the aforementioned aggregate scores in Table VI for all entities in the full tests. Indeed, the large number of *0* or *near-0* scores was disappointing and representative of multiple ecosystem weaknesses which we discuss in the following sections. Scores for the preliminary tests are found in Table VIII in Appendix II. Note that because Chrome, Firefox, and Safari showed nearly identical scores across all experiments, we simplify the table to report the highest respective score under the *GSB* heading. We make a similar simplification for IE 11 on Windows 10 and 8.1.

We do not separately report the performance of mobile browsers because we observed the behavior of mobile browsers to be directly related to their desktop counterparts. During the preliminary tests, mobile Firefox and Opera mirrored the blacklisting— and thus also the scores— of their desktop versions. Mobile Chrome and Mobile Safari showed no blacklist warnings *whatsoever* for any of our phishing sites and thus receive S_b scores of *0*. During the full tests, the behavior of mobile Chrome, Safari, and Opera remained unchanged. Firefox stopped showing blacklist warnings, and its scores thus dropped to *0* (the *0* scores of mobile browsers represented a serious issue; this was corrected after we contacted Google and Mozilla after the full tests, as discussed in Section VI-A1). We did not test mobile versions of Microsoft

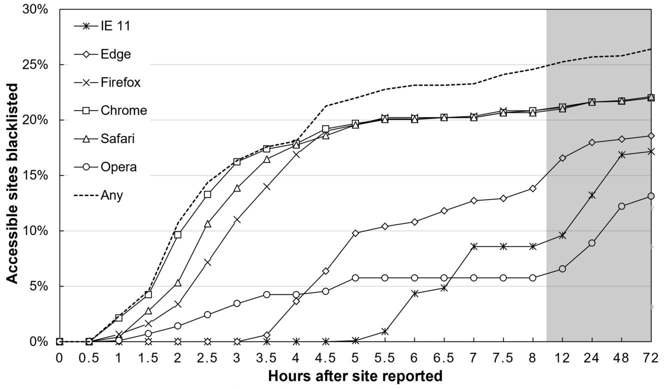


Fig. 2: Blacklisting over time in each browser (full tests).

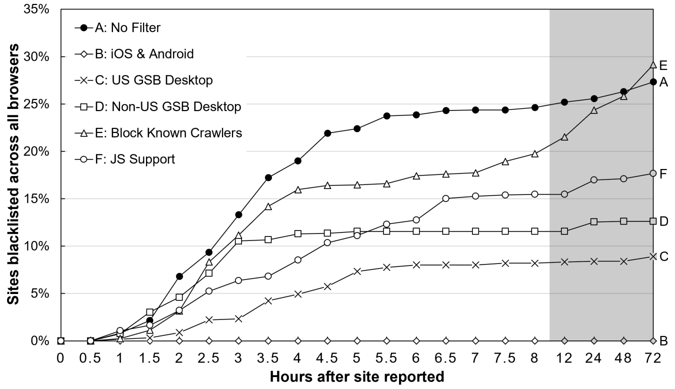


Fig. 3: Effect of cloaking on blacklisting over time (full tests).

browsers because mobile IE is no longer supported; Edge for Android and iOS was released after we began testing.

C. Preliminary vs. Full Tests

We observed many core similarities when comparing the performance same entity between the preliminary and full tests. We also saw improvements related to the recommendations we shared during our disclosure meetings, in particular with respect to the APWG's treatment of *Filters C* and *E*. Notably, during the full tests, crawler traffic to sites *with* cloaking increased by 44.7% relative to sites *without* cloaking, while the overall traffic volume also increased by 89.7%. We discuss all entity-specific improvements in section V-F.

The comparison also revealed some surprises, however. The main experimental difference between the two sets of tests, apart from sample size, was our exclusive use of random URLs in the full tests. On the other hand, the preliminary tests included a sampling of deceptive URLs. In the preliminary tests, we observed that Edge and IE were quick to display blacklist warnings for sites with certain deceptive URLs. In fact, many Type IV URLs (with domain names containing either the PayPal brand or deceptive keywords) saw proactive zero-hour warnings in these browsers without any prior crawler activity. Figure 5b in Appendix II shows the effect of URL type on blacklisting in the preliminary tests. During the full tests, no phishing site was blacklisted unless it had previously been visited by a crawler. In the absence of deceptive URLs, we thus observed all the blacklists to be purely reactive; this, in turn, led to lower S_b scores of Edge, IE, and sites with *Filter B*, and a lower overall score S for SmartScreen.

D. Browser Performance

Figure 2 shows the percentage of our phishing sites blacklisted over the course of all the full experiments, grouped by browser, but limited to sites intended to be *accessible* in each respective desktop browser (i.e. *Filter B* was excluded for all and *Filters C* and *D* were excluded for IE, Edge, and Opera).

Chrome, Safari, and Firefox consistently exhibited the highest overall blacklisting speed and coverage, but were still far from covering all cloaked sites. Opera generally outperformed the Microsoft browsers during the first four hours, but was later overtaken for the remainder of our experimental period. In the absence of deceptive phishing URLs, Edge and IE simply lost their edge in the full tests; Figure 5c in Appendix II shows their superior performance in the preliminary tests.

*1) **Consistency Between Browsers**:* Chrome, Firefox, and Safari are all protected by the GSB blacklist; our tests confirmed these browsers do consistently display blacklist warnings for the same set of websites. However, during our full tests, Chrome displayed warnings up to 30 minutes earlier than Safari, and up to one hour earlier than Firefox; the warnings became consistent after the five-hour mark. Upon further investigation, we believe that this disparity was caused by different caching implementations of the *GSB Update API (v4)* in each browser (this API ensures privacy and quick lookups but requires periodic refreshes of the entire blacklist [53]). Latency is a notable shortcoming of blacklists [6] and it appears that browser blacklist caching can still be improved from a security perspective.

Edge and IE 11 (both protected by SmartScreen) proved far less consistent. In the full tests, Edge displayed warnings up to two hours earlier *and* covered more sites than IE until the 24-hour mark. However, in the preliminary tests— which involved deceptive URLs detectable by heuristics— IE would often display preemptive warnings for sites which would not be blocked in Edge for several hours, if at all. These deviations were evident in the preliminary APWG, SmartScreen, and PayPal tests as per Table VIII in Appendix II.

E. Filter Performance

Figure 3 shows the percentage of our phishing sites blacklisted over the course of all the full experiments, grouped by cloaking filter type. Because this summary view masks some of the distinctive per-entity behavior we observed in Table VI, Figure 4 in Appendix I should be consulted as a supplement. Note that because these figures consider *all* browser-filter combinations, their Y-axes differ from Figure 2 and the S_{bf} scores in Table VI. Nevertheless, they all convey the finding that there exist considerable gaps in today's blacklists.

The overall earliest blacklisting we observed occurred after approximately 40 minutes, in Chrome. Significant growth took place between 1.5 and 6.5 hours and continued at a slower rate thereafter for the full 72-hour period. Compared to sites without cloaking, our cloaking techniques showed moderate to high effectiveness throughout the life of each phishing site. *Filter B* saw no blacklisting whatsoever across any desktop or mobile browser we tested. *Filters E* and *F* proved most effective in the early hours of deployment, while the geo-specific *Filters C* and *D* saw the lowest amount of blacklisting in the long term. Between 48 and 72 hours after deployment, sites with *Filter E* overtook *Filter A* sites by a small margin; upon further analysis, we found that this was due to a high level of interest in such sites following reporting to the APWG. All other types of sites with cloaking were on average less likely to be blacklisted than sites without.

F. Entity Performance

Although no single entity was able to overcome all of the cloaking techniques on its own, collectively the entities would be successful in doing so, with the exception of *Filter B* (this has since been corrected as per the discussion in Section VI-A1).

*1) **Google Safe Browsing**:* Due to the high market share of the browsers it protects, GSB is the most impactful anti-phishing blacklist today. It commanded the highest score S in both the preliminary and full tests. GSB's key strength lies in its speed and coverage: we observed that a crawler would normally visit one of our sites just seconds after we reported it. 94.7% of all the sites we reported in the full tests were in fact crawled, and 97% of sites without cloaking (*Filter A*) ended up being blacklisted. Blacklisting of *Filter D* was comparable, and *Filter F* improved over the preliminary tests.

However, GSB struggled with *Filter E*, which blocked hostnames specific to Google crawlers. It also struggled with *Filter C*, which denied non-US traffic. The reason the respective PB_f scores are low is that if the initial crawler hit on the phishing site failed, GSB would abandon further crawling attempts; the initial hit almost always originated from a predictable non-US IP address. Another weakness appears to be data sharing, as none of the sites we reported to GSB ended up being blacklisted in Edge, IE, or Opera.

*2) **Microsoft SmartScreen**:* SmartScreen proved to be the only native anti-phishing blacklist to leverage *proactive* URL heuristics to blacklist phishing sites, which allowed Microsoft browsers to achieve high scores during the preliminary tests. These heuristics were mainly triggered by URLs with a deceptive domain name. In the preliminary tests, Edge proved to be exceptionally well-protected, achieving a top S_b score of 0.87— the highest of any browser.

In the full tests, the performance of IE improved over the preliminary tests and became more consistent with that of Edge. Surprisingly, SmartScreen was more likely to blacklist sites *with* cloaking than those without, possibly due to use of classification of cloaking (which would be commendable) alongside low trust of our phishing reports (see *Limitations*).

Reporting to SmartScreen did not seem to significantly affect any non-Microsoft browsers; the entity thus shares a similar shortcoming with GSB. SmartScreen was also among the slowest entities to reactively respond to phishing reports, and its overall coverage was poor, which is its key weaknesses.

*3) **APWG**:* The APWG was the second-highest scoring entity in our full tests and showed consistent protection of all browsers. Its score S increased substantially compared to the preliminary tests due to improvements to bypass *Filters C* and *E*, which allowed APWG reports to result in blacklisting of such sites in GSB browsers— something not achieved when we reported directly to GSB. The APWG also generated the highest level of crawler traffic of any entity we tested.

Unfortunately, the APWG failed to blacklist any sites with *Filter D* or *F* in the full tests; its preliminary success proved to have been related solely to the detection of deceptive URLs by IE and Edge. Interestingly, we saw a large increase in the blacklisting of sites with *Filter E* after the 24-hour mark; after analyzing the traffic logs we believe that this is due to data sharing with PayPal (this trend is also reflected in Figure 4e).

*4) **PhishTank**:* PhishTank is a community-driven clearinghouse that allows human volunteers to identify phishing content [54]; it also leverages a network of crawlers and partners to aid in this effort. It was the second-highest performer in our preliminary tests thanks to its expeditious blacklisting in GSB browsers. In the full tests, we were surprised to see that only 46.7% of sites reported were crawled, and very few sites were ultimately blacklisted. Despite this, PhishTank generated the second-highest volume of total crawler traffic. We do not know the reasons for its shortcomings and PhishTank did not reply to our disclosure; we suspect that the manual nature of classification by PhishTank may be a limiting factor.

*5) **PayPal**:* During the preliminary tests, PayPal's own abuse reporting service struggled to bypass *Filters D* and *F*, but overcame the latter in the full experiments while maintaining a moderate degree of blacklisting. Its protection of Opera also improved between the two tests. Despite crawling all but two of the sites we reported in the full tests, the average response time and browser protection ended up being poor overall. We cannot disclose the reasons for this but expect to see a future improvement as a result of our findings.

*6) **Remaining Entities**:* Performance scores for entities only included in the preliminary tests are found in Table VIII within Appendix II. High-level descriptions of each follow.

All sites we reported to *ESET* ended up being crawled, but only a fraction of those– all with *Filter A*– were actually blacklisted. Overall timeliness was poor, though the blacklisting did span multiple browsers. *Netcraft* yielded the best protection of Opera in the preliminary tests, but overall it struggled with most of our cloaking techniques and did not deliver timely blacklisting. In retrospect, given the unexpectedly poor performance of *PhishTank* in the full tests, we would have been interested in re-testing *Netcraft*. Reports to the *US CERT* led to minimal crawler activity and blacklisting; disregarding heuristics from the Microsoft browsers, only a single site was blacklisted. Phishing reports we sent to *McAfee* effectively bypassed *Filter A* and *Filter E* but only appeared to lead to timely blacklisting in Microsoft Edge. Reports to *WebSense* had no effect beyond crawler traffic related to the URL heuristics used by Microsoft; while we were hopeful the e-mail reporting channel we used would prove fruitful, this is understandable given that the company focuses on protection of its enterprise customers.

G. Limitations

Our findings are based on a controlled (to the extent possible without sending out real spam e-mails) empirical experiment and observations from a large set of supporting metadata and a high volume of anti-abuse crawler traffic. Our study focuses exclusively on *native* phishing blacklist protection that is available by default in the browsers and platforms tested. Systems with third-party security software may see enhanced protection [6], though cloaking can also have an impact on systems powering such software.

Within its scope, our analysis should still be considered with certain limitations in mind. We suspect that real-world blacklisting of phishing attacks may be more timely than what our results otherwise suggest, as our efforts to isolate cloaking as a variable in our experimental design (i.e. by using randomized domain names, never rendering the actual phishing content in browsers being monitored, and using *.com* domains months after registration) also eliminated many of the methods that the ecosystem can use to detect or classify phishing (e.g. URL-, network-, or DNS-based analysis). However, this reduced detectability is offset, to an extent, by the possibility of malicious actors to likewise evade such forms of detection. We observed in the preliminary tests that only URLs containing brand names were quicker to be blacklisted than others; in the wild, there is also a shifting tendency to abuse compromised infrastructure and distribute random phishing URLs in lieu of more deceptive alternatives [8], [49]. In terms of factors under our control, it was not financially feasible to achieve a one-to-one mapping between IP addresses and all of our domains; this is a skewing factor which may have acted in favor of blacklists in the full tests, such as with the 10 *Filter D* sites which were blacklisted despite not being successfully retrieved during the GSB experiment.

Finally, we only submitted a *single* and *direct* report for each phishing site deployed. Although real-world phishing sites might see a much higher volume of automated reports (e.g. from sources such as spam filters), the volume of per-URL phishing reports in the wild can in fact be reduced by attackers (e.g. through the use of redirection links). More importantly, direct reports such as ours (in particular to the blacklist operators) might be subject to more suspicion because anti-phishing entities must account for adversaries who willingly submit false reports or seek to profile crawling infrastructure. Although the blacklist operators to which we disclosed did not express concern with our reporting methodology, we learned that crawling infrastructure used to respond to direct reports is indeed designed to be unpredictable to mitigate adversarial submissions. SmartScreen's lower crawl rate may be explained by this; GSB, on the other hand, consistently responded quickly and seemed to give our direct reports a high level of priority. It is therefore possible that either classification of reporting channel abuse works very accurately, or that reporting channels are more vulnerable to adversarial submissions than what the entities otherwise believe; regardless, to improve the future effectiveness of reporting, we propose an alternative approach in Section VI-B2.

Ultimately, if each report represents a chance that a phishing site will be blacklisted, we believe that our experimental design still captures trends therein; moreover, our findings with respect to cloaking effectiveness are consistent with internal PayPal e-mail and web traffic data pertaining to actual victims of phishing. To address its current limitations, our framework could be adapted to follow a different (possibly collaboratively arranged) reporting methodology, consider a broader range of cloaking techniques, or even be applied to proactively-detected live phishing URLs for which cloaking can be profiled.

1) ***Statistical Tests****:* In the full tests, we were surprised to find the blacklisting performance of each entity with respect to the different filters to have far more clear-cut gaps than in the preliminary tests; 11 of the 30 the per-entity filter groups saw no blacklisting whatsoever (per Table VI). Although ANOVA as originally planned could still be meaningfully applied to the subset of entities which had three or more filter groups that satisfied homogeneity of variance [50] (i.e. had some blacklisting activity), we chose not to perform such tests as the resulting power would be below our target, and instead relied on direct observations supported by crawler metadata. If our experiment were to be repeated to validate improvements in blacklisting or continuously test the ecosystem, we believe that statistical tests could be highly useful in assessing whether per-entity blacklisting performance improved significantly for each respective cloaking filter.

VI. Security Recommendations

Based on analysis of our experimental findings, we propose several possible improvements to the anti-phishing ecosystem.

A. Cloaking

Our first set of recommendations focuses specifically on the cloaking techniques we tested.

1) ***Mobile Users****:* We believe that the highest priority within the current ecosystem should be effective phishing protection for mobile users. Such users are not only inherently more vulnerable to phishing attacks [41], but now also comprise the majority of web traffic [4].

Our work has been impactful in better securing mobile users by enhancing existing anti-phishing systems. For over a year between mid-2017 and late 2018, GSB blacklists (with nearly 76% global market share) simply did not function properly on mobile devices: none of our phishing sites with *Filter A*, *E* or *F* (targeting both desktop and mobile devices) showed any warnings in mobile Chrome, Safari, or Firefox despite being blacklisted on desktop. We confirmed the disparity between desktop and mobile protection through periodic small-scale testing and by analyzing an undisclosed dataset of traffic to real phishing sites. Following our disclosure, we learned that the inconsistency in mobile GSB blacklisting was due to the transition to a new mobile API designed to optimize data usage, which ultimately did not function as intended. Because blacklisting was not rectified after our full tests, we contacted the entities anew. As a result, in mid-September 2018 Mozilla patched Firefox (from version 63) such that all desktop warnings were also shown on mobile. Google followed suit days thereafter with a GSB API fix that covered mobile GSB browsers retroactively; mobile Chrome and Safari now mirror desktop listings, albeit with a shorter-lived duration to lower bandwidth usage. Chrome, Safari, and Firefox thus again join Opera as mobile browsers with effective blacklists, though some popular mobile browsers still lack such protection [41].

Upon close inspection of the preliminary test results, we found that *Filter B* sites were solely blacklisted due to their suspicious URLs rather than our reports. During our full tests, not a single site with *Filter B* was blacklisted in any browser, interestingly despite the fact that crawlers did successfully retrieve many of these sites. GSB addressed this vulnerability in mid-September 2018, together with the aforementioned API fix. Through a subsequent final redeployment of PhishFarm, we verified that sites with *Filter B* were being blacklisted following reports to GSB, the APWG, and PayPal. Other entities— including ones we did not test— should ensure that sites targeted at mobile users are being effectively detected.

2) ***Geolocation****:* Although our experiments only considered two simple geolocation filters (US and non-US), our findings are indicative of exploitable weaknesses in this area. Given the overwhelming amount of crawler traffic from the US (79%, per Table IX in Appendix III), *Filter C* should *not* have been as effective as it proved to be. We hypothesize that other geo-specific filters would have similarly low blacklisting rates, in part due to the crawler characteristics discussed in Section V-F1. Country- or region-specific filtering paired with localized page content is not an unusual sight in real-world PayPal phishing kits that we have analyzed.

3) ***JavaScript****:* It is trivial to implement JavaScript-based cloaking such as *Filter F*. This technique proved to be effective in slowing blacklisting by three of the five entities in our full tests. Fortunately, SmartScreen bypasses this technique well, and PayPal started doing so following our disclosure. The broader ecosystem should better adapt to client-side cloaking, in particular if its sophistication increases over time.

B. Anti-phishing Entities

We also offer a number of more general recommendations for anti-phishing systems of tomorrow.

1) ***Continuous Testing****:* The mere failure of mobile blacklisting that we observed during our experiments is sufficient to warrant the need for continuous testing and validation of blacklist performance. Periodic deployment of PhishFarm could be used for such validation. In addition, continuous testing could help ensure that future cloaking techniques— which may grow in sophistication— can effectively be mitigated without compromising existing defenses. As an added benefit, trends in the relative performance of different entities and the overall timeliness and coverage of blacklisting could be modeled.

2) ***Trusted Reporting Channels****:* The phishing reporting channels we tested merely capture a suspected URL or a malicious e-mail. While the latter is useful in identifying spam origin, we believe a better solution would be the implementation of standardized trusted phishing reporting systems that allow the submission of specific metadata (such as victim geolocation or device). Trusted channels could allow detection efforts to more precisely target high-probability threats while minimizing abuse from deliberate false-negative submissions; they could also simplify and expedite collaboration efforts between anti-phishing entities and abused brands, which may hold valuable intelligence about imminent threats.

3) ***Blacklist Timeliness****:* The gap between the detection of a phishing website and its blacklisting across browsers represents the prime window for phishers to successfully carry out their attacks. At the level of an individual entity, cloaking has a stronger effect on the *occurrence* rather than the timeliness of blacklisting. However, if we look at the ecosystem as a whole in Figure 3, cloaking clearly delays blacklisting overall. Our test results show that blacklisting now typically occurs in a matter of hours— a stark improvement over the day- or week-long response time observed years ago [6], [55]. However, given the tremendous increase in total phishing attacks since then (on the order of well over 100 attacks per hour in 2018 [2]), we believe that even today's 40-minute best-case blacklist response time is too slow to deter phishers and effectively protect users. The gap needs to be narrowed, especially by slower entities (i.e. those not directly in control of blacklists). Future work should investigate the real-world impact of delays in blacklisting on users and organizations victimized by phishing attacks in order to accurately establish an appropriate response threshold.

4) ***Volume****:* GSB, the APWG, and PayPal crawled nearly all of the phishing sites we reported. In particular, GSB proved to deliver a consistently agile response time despite the high number of reports we submitted. Other entities fell short of this level of performance. With increasing volumes of phishing attacks, it is essential that all players in the ecosystem remain robust and capable of delivering a consistent response.

5) ***Data Sharing****:* Data sharing has long been a concern within the ecosystem [56]. We found that the two main blacklist operators (GSB and SmartScreen) did not appear to effectively share data with each other, as per Table VI. However, clearinghouse entities (APWG and PhishTank) and PayPal itself showed consistent protection across all browsers. Unfortunately, the timeliness and overall coverage of clearinghouses appear to be inferior to those of the blacklist operators in their respective browsers. Closer cooperation could thus not only speed up blacklisting, but also ensure that malicious sites are blocked universally. Strengthening this argument, perhaps a breakdown in communication between infrastructure used by different entities accounted for those of our sites which were successfully crawled but not ultimately blacklisted.

VII. Related Work

To the best of our knowledge, our work is the first controlled effort to measure the effects of cloaking in the context of phishing. Several prior studies measured the general effectiveness of anti-phishing blacklists and the behavior of phishing kits; none of the prior work we identified considered cloaking, which may have had a skewing effect on the datasets and ecosystem findings previously reported. Cloaking itself has previously been studied with respect to malicious search engine results; Invernizzi et al. [7] proposed a system to detect such cloaking with high accuracy. Oest et al. [8] later studied the nature of server-side cloaking techniques within phishing kits and proposed approaches for defeating each.

The work most similar to ours is NSS Labs' [39] recent use of a proprietary distributed testbed [57] to study the timeliness of native phishing blacklisting in Chrome, Firefox, and Edge. The main limitation of NSS Labs' approach is the reliance on feeds of known phishing attacks; any delay in the appearance of a site in each source feed can affect the accuracy of blacklisting time measurements. Furthermore, phishing sites could be overlooked in the case of successful cloaking against the feeds. We address these limitations by having full control over the deployment and reporting time of phishing sites.

Sheng et al. [6] took an empirical approach to measure the effectiveness of eight anti-phishing toolbars and browsers powered by five anti-phishing blacklists. The authors found that heuristics by Microsoft and Symantec proved effective in offering zero-hour protection against a small fraction of phishing attacks, and that full propagation across phishing blacklists spanned several hours. This work also found that false positive rates in blacklists are near-zero; we thus did not pursue such tests in our experiments. While Sheng et al.'s work was based on phishing sites only 30 minutes old, and was thus better controlled than earlier blacklist tests [58], the datasets studied were of limited size and heterogeneous in terms of victim brands; the anti-phishing tools evaluated are now dated. In addition, Sheng et al. checked blacklist status with a granularity of one hour— longer than our 10 minutes.

Han et al. [25] analyzed the lifecycle of phishing sites by monitoring cybercriminals' behavior on a honeypot web server. The authors timed the blacklisting of the uploaded sites across Google Safe Browsing and PhishTank. Uniquely, this work sheds light on the time between the creation and deployment of real phishing sites. A key difference in our work is our ability to customize and test different configurations of phishing kits instead of waiting for one to be uploaded. For instance, we could target specific brands or configure our own cloaking techniques to directly observe the ecosystem's response. Our experiments suggest a significantly faster blacklist response time by the ecosystem than what Han et al. found; our test sample size was also nearly five times larger.

Virvilis et al. [41] carried out an evaluation of mobile web browser phishing protection in 2014 and found that major mobile web browsers at the time included no phishing protection. Like that of Sheng et al., this evaluation was empirical and based on a set of known phishing URLs. In our work, we found that mobile Chrome, Safari, and Firefox now natively offer blacklist protection, but that this protection was not functioning as advertised during our tests.

The differences between today's phishing trends and those seen in prior work show that the ecosystem is evolving quickly. This warrants regular testing of defenses and re-evaluation of criminals' circumvention techniques and attack vectors; it also underscores the importance of scalable and automatable solutions. Our testbed shares some similarities with previous work [39], [6] but it is the only such testbed to offer full automation without the need for intervention during the execution of experiments, and the only one to actively deploy sites and directly send reports to entities.

VIII. Conclusion

By launching and monitoring a large set of phishing sites, we carried out the first controlled evaluation of how cloaking techniques can hamper the timeliness and occurrence of phishing blacklist warnings in modern web browsers. As a result of our disclosure to anti-phishing entities, mobile blacklisting is now more consistent, and some of the cloaking techniques we tested are no longer as effective; others represent ongoing vulnerabilities which could be addressed through tweaks to existing detection systems. Such tweaks should also seek to improve overall timeliness of blacklists to better counter the modern onslaught of phishing attacks. Blacklist protection should not be taken for granted; continuous testing— such as that supported by our framework— is key to ensuring that browsers are secured sufficiently, and as intended.

Cloaking carries a detrimental effect on the occurrence of blacklisting due to the fundamentally-reactive nature of the main detection approach currently used by blacklists; it thus has the potential to cause continuous damage to the organizations and users targeted by phishers. Although no single entity we tested was able to individually defeat all of our cloaking techniques, collectively the requisite infrastructure is already in place. In the short term, collaboration among existing entities could help address their individual shortcomings. We observed that proactive defenses (such as the URL heuristics of SmartScreen) proved to deliver superior protection— but only under the right circumstances. In the long term, the ecosystem should move to more broadly implement general-purpose proactive countermeasures to more reliably negate cloaking. It is important for the ecosystem to be able to effectively bypass cloaking because it is merely one way in which phishing sites can be evasive. For instance, with cloaking alongside redirection chains or bulletproof hosting, phishing sites might otherwise avoid existing mitigations far more successfully than what we have observed.

Phishing has proven to be a difficult problem to solve due to attackers' unyielding persistence, the cross-organizational nature of infrastructure abused to facilitate phishing, and the reality that technical controls cannot always compensate for the human weakness exploited by social engineers. We believe that continuous and close collaboration between all anti-abuse entities, which can lead to a deep understanding of current threats and development of intelligent defenses, is the crux of optimizing controls and delivering the best possible long-term protection for phishing victims.

Acknowledgments

The authors would like to thank the reviewers for their insightful feedback and suggestions. This work was partially supported by PayPal, Inc. and a grant from the Center for Cybersecurity and Digital Forensics at Arizona State University.

References

[1] "Anti-Phishing Working 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

[2] "Anti-Phishing Working Group: APWG Trends Report Q1 2018," (Date last accessed 31-August-2018). [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2018.pdf

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

[4] "Desktop vs mobile vs tablet market share worldwide," http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet, 2018, [Online; accessed 31-Aug-2018].

[5] 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.

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

[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] A. Oest, Y. Safaei, A. Doupé, G. Ahn, B. Wardman, and G. Warner, "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis," in *2018 APWG Symposium on Electronic Crime Research (eCrime)*, May 2018, pp. 1–12.

[9] 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.

[10] 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.

[11] D. K. McGrath and M. Gupta, "Behind phishing: An examination of phisher modi operandi." *LEET*, vol. 8, p. 4, 2008.

[12] 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.

[13] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2006, pp. 581–590.

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

[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.

[16] M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually prevent phishing attacks?" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2006, pp. 601–610.

[17] G. Stringhini, C. Kruegel, and G. Vigna, "Shady paths: Leveraging surfing crowds to detect malicious web pages," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13, 2013, pp. 133–144.

[18] Y. Takata, S. Goto, and T. Mori, "Analysis of redirection caused by web-based malware," *Proceedings of the Asia-Pacific advanced network*, vol. 32, pp. 53–62, 2011.

[19] "CWE-601: URL Redirection to Untrusted Site ('Open Redirect')." [Online]. Available: http://cwe.mitre.org/data/definitions/601.html

[20] 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

[21] 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.

[22] 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.

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

[24] 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*, July 2007, p. 3.

[25] 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.

[26] 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.

[27] Z. Ramzan, "Phishing attacks and countermeasures," in *Handbook of information and communication security*. Springer, 2010, pp. 433–448.

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

[29] 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.

[30] 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.

[31] 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)*, vol. 1. IEEE, 2016, pp. 408–416.

[32] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure: Finding malicious domains using passive dns analysis." in *NDSS*, 2011.

[33] 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.

[34] 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.

[35] 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*, 2007, pp. 639–648.

[36] 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*, 2013, pp. 177–188.

[37] A. Lukovenko, "Let's Automate Let's Encrypt," *Linux Journal*, no. 266, Jun. 2016.

[38] J. P. Randy Abrams, Orlando Barrera, "Browser Security Comparative Analysis - Phishing Protection," *NSS Labs*, 2013, https://www.helpnetsecurity.com/images/articles/Browser%20Security%20CAR%202013%20-%20Phishing%20Protection.pdf.

[39] NSS Labs, "NSS Labs Conducts First Cross-Platform Test of Leading Web Browsers," Oct 2017. [Online]. Available: https://www.nsslabs.com/company/news/press-releases/nss-labs-conducts-first-cross-platform-test-of-leading-web-browsers/

[40] 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.

[41] N. Virvilis, N. Tsalis, A. Mylonas, and D. Gritzalis, "Mobile devices: A phisher's paradise," *2014 11th International Conference on Security and Cryptography (SECRYPT)*, pp. 1–9, 2014.

[42] "Google Safe Browsing: Report Phishing Page." [Online]. Available: https://safebrowsing.google.com/safebrowsing/report_phish/

[43] "SmartScreen: Report a Website." [Online]. Available: https://feedback.smartscreen.microsoft.com/feedback.aspx?t=0url=

[44] "ESET: Report a Phishing Page." [Online]. Available: http://phishing.eset.com/report/

[45] "NetCraft: Report a Phishing URL." [Online]. Available: http://toolbar.netcraft.com/report_url

[46] “McAfee: Custom URL ticketing System.” [Online]. Available: ”https://www.trustedsource.org/en/feedback/url?action=checksingle

[47] “Digitalocean: Cloud computing, simplicity at scale.” [Online]. Available: https://www.digitalocean.com/

[48] 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.

[49] 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, pp. 1–8.

[50] J. Cohen, “Statistical power analysis for the behavioral sciences.” 1988.

[51] E. Lewis, “The role of wildcards in the domain name system,” United States, 2006.

[52] K. Ravishankar, B. Prasad, S. Gupta, and K. K. Biswas, “Dominant color region based indexing for cbir,” in *Image Analysis and Processing, 1999. Proceedings. International Conference on*. IEEE, 1999, pp. 887–892.

[53] “Overview — safe browsing apis (v4) — google developers.” [Online]. Available: https://developers.google.com/safe-browsing/v4/

[54] 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*, pp. 31–36.

[55] T. Moore, R. Clayton, and H. Stern, “Temporal correlations between spam and phishing websites.” in *LEET*, 2009.

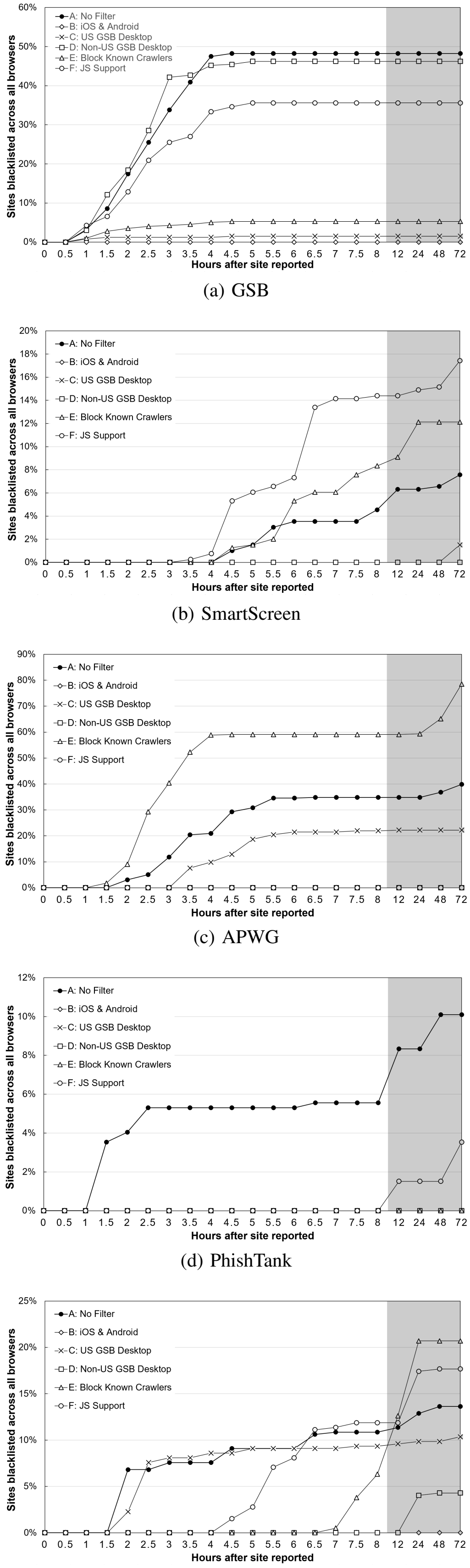
[56] T. Moore and R. Clayton, “How hard can it be to measure phishing?” *Mapping and Measuring Cybercrime*, 2010.

[57] NSS Labs, “Web browser security: Phishing protection test methodology v3.0,” Jul 2016. [Online]. Available: https://research.nsslabs.com/reports/free-90/files/TestMethodology_WebB/Page4

[58] C. Ludl, S. McAllister, E. Kirda, e. H. B. Kruegel, Christopher, and R. Sommer, “On the effectiveness of techniques to detect phishing sites,” in *Detection of Intrusions and Malware, and Vulnerability Assessment*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 20–39.

Appendix I: Full test Per-entity Blacklisting

Each entity we tested exhibited a characteristic blacklisting behavior with respect to the different cloaking techniques; the aggregate view of filter performance in Figure 3 masks some of these characteristics. Figure 4 includes similar charts for each individual entity as a supplement to Figure 3 and the entity scores in Table VI. Each chart shows detailed blacklisting performance over time, aggregated for all browsers, for each filter type.



(a) GSB

(b) SmartScreen

(c) APWG

(d) PhishTank

(e) PayPal

Fig. 4: Blacklisting over time by filter (full tests).

APPENDIX II: PRELIMINARY TEST DATA

Table VIII includes detailed performance scores for all entities in the preliminary tests. These scores are based on the formulas in Section V-B and are the basis of our comparative discussion in Section V-C. We visualize the preliminary test performance of only the subsequently re-tested entities (GSB, SmartScreen, AWPG, PhishTank, and PayPal) in Figure 5a.

Figure 5b illustrates the increased likelihood of URLs with a deceptive domain (Type IV [8], [49]) to be blacklisted during the preliminary tests. As previously discussed, this increase is linked to heuristics used by SmartScreen browsers; the positive effect that this had on IE and Edge blacklisting can be seen in the browser performance breakdown in Figure 5c. The latter two charts are based on data from all 10 preliminary tests.

TABLE VIII: Aggregate entity blacklisting performance scores in the preliminary tests.

GSB		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.988	0	0	0.846	0	0.493	0.466	
	IE	0	0.142	0	0	0	0.148	0.049	
	Edge	0.950	0.130	0	0.709	0	0.703	0.551	
	Opera	0.138	0	0	0.236	0	0	0.046	
PB_f		1.000	1.000	0	0.857	0	0.833	**0.421**	*S*
TB_f		38	10	N/A	43	N/A	151	0.900	*C*

SmartScreen		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0	0	0	0	0	0	0	
	IE	0	0.142	0.284	0.142	0.135	0.166	0.100	
	Edge	0.956	0	0	0	0.952	0.703	0.870	
	Opera	0	0	0	0	0	0	0	
PB_f		1.000	0.143	0.286	0.143	1.000	0.833	**0.045**	*S*
TB_f		10	10	14	18	162	7	1	*C*

APWG		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.648	0	0	0.141	0	0	0.158	
	IE	0.255	0.432	0.306	0.306	0.524	0.178	0.319	
	Edge	0.958	0.142	0.137	0	0.821	0.632	0.804	
	Opera	0.345	0	0	0	0	0	0.115	
PB_f		1.000	1.000	1.000	0.857	1.000	0.333	**0.198**	*S*
TB_f		73	286	199	154	276	188	1	*C*

PhishTank		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.971	0.141	0.261	0	0.399	0.303	0.387	
	IE	0.314	0	0	0	0.286	0.167	0.255	
	Edge	0.771	0	0	0.124	0.522	0.295	0.529	
	Opera	0.112	0.123	0	0	0	0	0.037	
PB_f		1.000	0.286	0.286	0.125	0.714	0.333	**0.372**	*S*
TB_f		95	309	344	3784	162	363	0.975	*C*

PayPal		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.637	0	0.276	0	0.408	0	0.264	
	IE	0.345	0.517	0.448	0.306	0.525	0	0.290	
	Edge	0.213	0	0	0.173	0	0.145	0.119	
	Opera	0.102	0	0.253	0	0	0	0.034	
PB_f		1.000	1.000	1.000	1.000	1.000	0.167	**0.255**	*S*
TB_f		121	181	128	179	154	3694	0.925	*C*

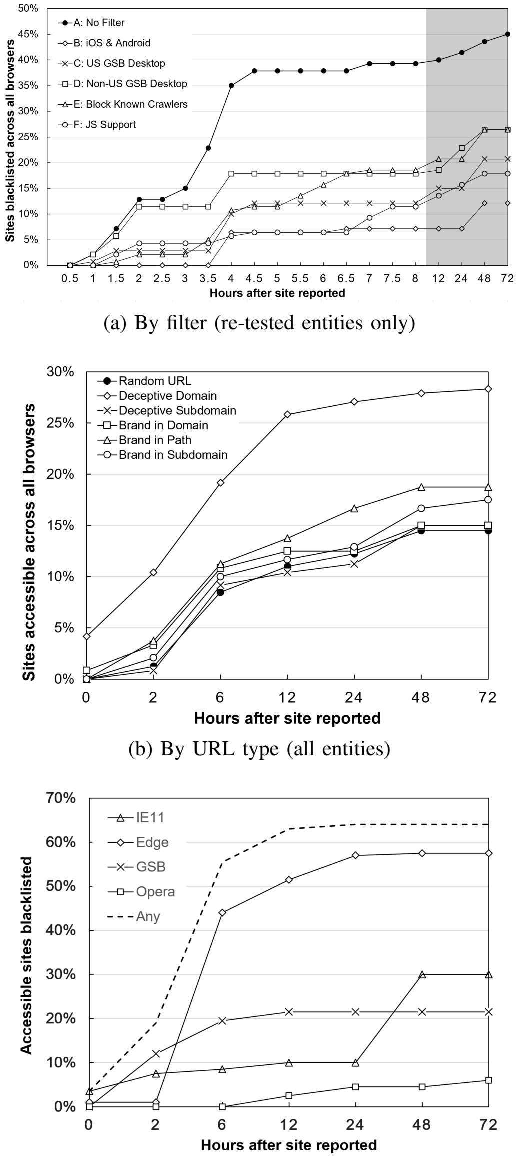
ESET		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0.297	0	0	0	0	0	0.059	
	IE	0	0	0	0	0	0	0	
	Edge	0.256	0	0	0	0	0	0.085	
	Opera	0.137	0	0	0	0	0	0.046	
PB_f		0.333	0	0	0	0	0	**0.055**	*S*
TB_f		444	N/A	N/A	N/A	N/A	N/A	1	*C*

WebSense		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0	0	0	0	0	0	0	
	IE	0	0	0.142	0	0	0.331	0.110	
	Edge	0	0	0.128	0	0	0.299	0.100	
	Opera	0	0	0	0	0	0	0	
PB_f		0	0	0.143	0	0	0.286	**0.014**	*S*
TB_f		444	N/A	4	N/A	N/A	6	0.275	***C***

Netcraft		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0	0.135	0	0.538	0	0	0.108	
	IE	0.166	0	0.142	0	0.134	0	0.100	
	Edge	0.389	0	0.129	0	0	0	0.130	
	Opera	0.302	0.260	0.130	0	0.130	0	0.144	
PB_f		0.667	0.429	0.182	0.571	0.182	0	**0.109**	*S*
TB_f		531	334	206	241	318	N/A	0.975	***C***

US CERT		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0	0	0.139	0	0	0	0.028	
	IE	0	0.127	0.142	0	0.134	0	0.045	
	Edge	0	0.127	0.127	0	0.127	0	0.042	
	Opera	0	0	0	0	0	0	0	
PB_f		0	0.143	0.286	0	0.143	0	**0.029**	*S*
TB_f		N/A	28	70	N/A	248	N/A	0.200	***C***

McAfee		*Filter A*	*Filter B*	*Filter C*	*Filter D*	*Filter E*	*Filter F*	S_b	
S_{bf}	*GSB*	0	0	0	0	0	0.160	0.032	
	IE	0.167	0.127	0.143	0.143	0	0	0.056	
	Edge	0.939	0	0	0	0.955	0.118	0.671	
	Opera	0	0	0	0	0	0	0	
PB_f		1	0.143	0.143	0.143	1	0.167	**0.059**	*S*
TB_f		134	466	3702	3702	180	172	1	***C***



(a) By filter (re-tested entities only)

(b) By URL type (all entities)

(c) By browser (all entities)

Fig. 5: Blacklisting over time (preliminary tests).

Appendix III: Crawler Traffic Analysis

Our 2,380 phishing sites logged a total of 2,048,606 HTTP requests originating from 100,959 distinct IP addresses. A substantial proportion of requests was characterized by crawlers scanning for phishing kit archives (i.e. zip files) or credential dump files; such requests resulted in 404 “not found” errors. It is beneficial for the security community to be able to identify compromised user information and study phishing kits [8], [10], but such crawling is noisy. By monitoring traffic to their phishing sites, cybercriminals could become aware of the ecosystem’s knowledge of their tools, and adapt accordingly.

Figure 6 aggregates the web traffic to all phishing sites from the full tests over the course of a 2-week period relative to initial deployment (along a logarithmic scale). Not unexpectedly, we observed the largest amount of traffic in the hours immediately after reporting each phishing site. Smaller spikes occurred several hours or days thereafter as additional infrastructure started accessing the sites. We automatically disabled each site at the end of its 72-hour deployment; crawlers would thus start seeing 404 errors thereafter. We observed a spike in traffic at this point with characteristics similar to the initial crawler traffic, followed by an immediate sharp decline (presumably once the offline state of each site was verified). Over the following seven days, we logged a consistent yet slowly-declining level of traffic. It is clear that an effort is being made to ensure that offline phishing content does not make a return. After about 10 days, we saw a second sharp decline, after which the traffic reached insignificant levels. We did not study blacklist warning persistence across browsers; this could be an interesting area to explore in the future.

*1) **Geographical Origin**:* Using the GeoLite2 IP database, we found that traffic to our phishing sites originated from 113 different countries across the majority of North America, Europe, and Asia, and some of Africa, as shown in Table IX. 79.02% of all unique IP addresses were based in the US; this accounted for a slightly lower 64.73% of all traffic but still constituted an overwhelming majority overall.

*2) **Entity Crawler Overlap**:* We provide a summary of IP address overlap between entities in Table X. The data is indicative of collaboration between certain entities, as discussed in Section VI-B5. A per-entity analysis of long-term crawler traffic is outside the scope of this work.

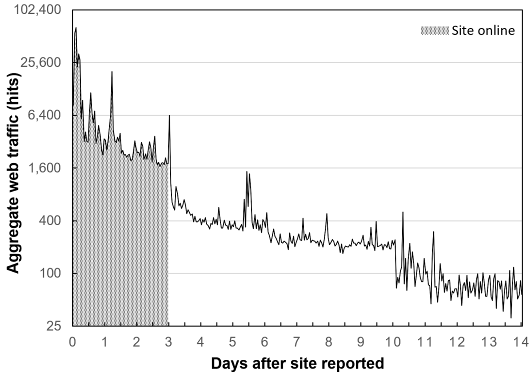


Fig. 6: Traffic to our phishing sites over time (full tests).

TABLE IX: Geographical distribution of requests to our sites.

Country	*Total Traffic*	*Unique IPs*
United States	64.73%	79.02%
United Kingdom	6.66%	2.42%
Germany	4.72%	1.30%
Brazil	1.99%	2.04%
Italy	1.80%	0.34%
Japan	1.73%	0.43%
Netherlands	1.73%	0.76%
India	1.54%	0.76%
Canada	1.36%	1.28%
France	1.21%	0.68%
Belgium	0.85%	0.13%
Singapore	0.65%	0.30%
Ireland	0.65%	0.66%
Norway	0.65%	0.18%
Australia	0.63%	0.34%
Korea	0.50%	0.17%
Denmark	0.50%	0.12%
Estonia	0.48%	0.07%
Austria	0.45%	0.15%
Russia	0.42%	2.23%
Unknown	3.99%	1.96%
93 Others	2.75%	4.65%

TABLE X: Crawler IP overlap between entities.

Entity	*Unique IPs*	*IP Overlap*				
		GSB	**MS**	**APWG**	**Phish-Tank**	**PayPal**
GSB	1,788		7.94%	31.20%	18.40%	53.52%
MS	475	29.89%		29.89%	23.16%	38.11%
APWG	6,165	11.08%	2.30%		11.13%	47.96%
PhishTank	2,409	13.66%	4.57%	28.48%		47.11%
PayPal	17,708	5.40%	1.02%	16.70%	6.41%	

TABLE XI: Web traffic during and after site deployment.

		Total HTTP Requests		**Unique IP Addresses**	
		Valid URL	*Invalid URL*	*Valid URL*	*Invalid URL*
Prelim.	*Sites Live*	271,943	452,049	6,528	11,869
	Day 3-14	262,141		7,230	
	Total	986,133		20,874	
Full	*Sites Live*	355,093	545,704	22,929	54,392
	Day 3-14	161,676		21,991	
	Total	1,062,473		80,085	