What Happens After You Leak Your Password: Understanding Credential Sharing on Phishing Sites

Peng Peng, Chao Xu, Luke Quinn, Hang Hu, Bimal Viswanath, Gang Wang
Department of Computer Science, Virginia Tech
{pengp17, chaoxu18, lquinn, hanghu, vbimal, gangwang}@vt.edu

ABSTRACT

Phishing has been a big concern due to its active roles in recent data breaches and state-sponsored attacks. While existing works have extensively analyzed phishing websites and their operations, there is still a limited understanding of the information sharing flows throughout the end-to-end phishing process. In this paper, we perform an empirical measurement on the transmission and sharing of stolen login credentials. Over 5 months, our measurement covers more than 179,000 phishing URLs (47,000 live phishing sites). First, we build a measurement tool to feed fake credentials to live phishing sites. The goal is to monitor how the credential information is shared with the phishing server and potentially third-party collectors on the client side. Second, we obtain phishing kits from a subset of phishing sites to analyze how credentials are sent to attackers and third-parties on the server side. Third, we set up honey accounts to monitor the post-phishing exploitation activities from attackers. Our study reveals the key mechanisms for information sharing during phishing, particularly with third-parties. We also discuss the implications of our results for phishing defenses.

CCS CONCEPTS

• **Security and privacy** → *Web application security*;

KEYWORDS

Phishing; Measurement; Honey Account

ACM Reference Format:
Peng Peng, Chao Xu, Luke Quinn, Hang Hu, Bimal Viswanath, Gang Wang. 2019. What Happens After You Leak Your Password: Understanding Credential Sharing on Phishing Sites. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS '19), July 9–12, 2019, Auckland, New Zealand*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3321705.3329818

1 INTRODUCTION

Phishing attack is a persistent threat on the Internet. It exploits human factors to lure the target users to give away critical information. In recent years, phishing becomes an even bigger concern due to its prevalent usage in facilitating major data breaches [3], particularly the recent breaches in hospitals and health care companies [4, 5]. In addition, phishing plays an important role in many state-sponsored attacks. One of the recent examples is the spear phishing attack against John Podesta, the campaign manager of Hillary Clinton, during the US election in 2016 [1].

The research community has been studying phishing attacks from different aspects. While some existing works analyzed phishing emails [20], the vast majority focus on the *phishing websites* that are set up by attackers to trick users to reveal important information (*e.g.*, login credentials) [36, 38, 40, 42]. These phishing sites often impersonate other reputable entities to gain the victim's trust. More recently, researchers analyze phishing kits, the software packages for running phishing websites, to understand how phishing sites are deployed and operated [12, 19, 29]. However, these works only looked into the disconnected parts of phishing. There is a limited end-to-end understanding of *the information flow after user credentials are leaked to the phishing sites.*

In this paper, we perform an empirical measurement by piecing together the different stages of phishing to understand the information flow. We collect a large set of *live* phishing sites and feed fake login credentials to these sites. In this process, we monitor how the information is shared to the attackers who deployed the phishing site, and more importantly, any other third-parties. For the *client-side* measurement, we build a measurement tool to automatically detect a login form, fill in the fake credentials, and monitor the network traffic to external parties. For the *phishing-server* measurement, we build a crawler to retrieve phishing kits, and run them in a sandbox to detect first-party and third-party information collectors. Finally, to examine what attackers do after obtaining the login credentials, we set up our own honey accounts (in email services) to monitor the potential post-phishing exploiting activities. These steps allow us to provide an end-to-end view of the phishing process and credential sharing.

We performed the measurement from August 2018 to January 2019 covering 179,865 phishing URLs. The client-side measurement covers 41,986 live phishing sites, and the server-side measurement is based on the analysis of 2,064 detected phishing kits. Our post-phishing exploitation analysis uses 100 honey accounts from Gmail and 50 accounts from ProtonMail for data collection. We explore how likely attackers would attempt to use the leaked password to further hijack the associated email account (in addition to the original online account).

Our study leads to a number of key findings. First, we show that user credentials are shared in real time on both the client-side and the server-side. This easily exposes the stolen credentials to more malicious parties. Second, while the client-side sharing is not very common (about 5%), the third-party servers are often located in a different country (compared to the phishing server), which may

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
AsiaCCS '19, July 9–12, 2019, Auckland, New Zealand
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6752-3/19/07…$15.00
https://doi.org/10.1145/3321705.3329818

create difficulties to take them down. In particular, many "good" websites were used to receive stolen credentials (*e.g.*, Google Ads are used to track the phishing statistics for attackers). Third, server-side credential sharing is primarily done via emails. 20% of the phishing kits send the credentials to two or more email addresses. About 5% of the phishing kits contain backdoors that stealthily leak the credentials to third-parties. Finally, from our honey email accounts, we observe that attackers indeed attempted to exploit the honey accounts shortly after phishing (within tens of minutes or 1–2 days). A single leakage can attract multiple attackers, which indicates credential sharing.

Our paper makes three key contributions:

- *First*, we perform a large-scale empirical measurement on the information flow of credential sharing during phishing attacks. Our measurement covers both client-side, and server-side information sharing, and post-phishing exploitation.
- *Second*, we build a new measurement tool to automatically seed fake credentials to phishing sites to measure the information sharing in real time. We will make the tool available for sharing with the research community.
- *Third*, our measurements provide new insights into the credential sharing mechanisms (to third-parties) during the phishing process.

In the end of the paper (§7), we discuss how third-party sharing and backdoors can be potentially used by defenders for good purposes. For example, the defender may leverage the third-party sharing channel to establish a vantage point to back-track phishing kit usage, and provide early alerts for phishing victims.

2 BACKGROUND & MOTIVATIONS

We start by introducing the background of phishing, and the different ways for attackers collect the leaked information. Then we describe our high-level research goals and approaches.

2.1 Background of Phishing

Figure 1 shows the typical steps of a phishing attack. Attackers first need to trick users into visiting a phishing website. To gain the victim's trust, a phishing website often impersonates other reputable services. In step1, the victim user submits the login credential via the phishing page in the browser. After that, the information is then sent to the phishing server (step2.1). The phishing server either directly sends the collected credentials via emails to the attacker (step3.1), or the attacker will (manually) log into the phishing server to retrieve the information (step3.2). Once the login credentials are obtained by the attacker, they can proceed further with malicious activities against users or their organizations (*e.g.*, stealing data, compromising enterprise/government networks).

Phishing Kits. Attackers often deploy phishing websites using a collection of software tools called phishing kits [12]. Phishing kits allow people with little technical skills to run phishing attacks. A typical kit contains a website component, and an information processing component. The website component contains the code, images, and other content to create a fake website. The information processing tool will automatically record and store the received information (password, login time, IP), and send the information

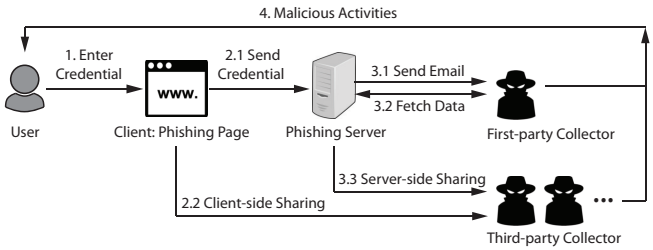


Figure 1: Phishing attack process.

to the attacker. Some phishing kits also contain a spamming tool, which can send spam emails to lead users to the phishing sites.

Third-party Information Sharing. During a phishing attack, it is possible that the user credentials are also shared to third-parties, in both the *client-side* and the *server-side*.

- **Client-side Third Parties.** Step2.2 shows that client-side third-parties collect the user credential. In this case, the phishing server that directly hosts the phishing page is the first-party and any other servers that also collect the credential are third-parties. The information sharing happens in real time when the user clicks on the "submit" button.
- **Server-side Third Parties.** Step3.3 represents the server-side third-parties. Certain phishing kits contain "back-doors" planted by other parties (*e.g.*, the phishing kit developer) [12]. After the login credentials are received by the phishing server, the information will be sent to the first-party (who deployed the phishing website), and also possibly to the third-party (who planted the back-door in the phishing kit).

2.2 Our Motivations

Phishing is an extensively-studied topic, and yet there is still a lack of empirical understanding of the information flow after the credential leakage. Most existing works focus on step1 to analyze the characteristics of phishing websites and their hosting domains to build detection systems [36, 38, 40, 42]. More recently, researchers analyze the phishing kits to understand how phishing websites are deployed [19, 29]. However, these works are usually limited in scale and scope. More importantly, there is no existing work that systematically measures the real-time credential sharing to third-parties, or examines the post-phishing exploitation activities.

In this paper, we seek to provide a more comprehensive view of the information flow of phishing attacks via a large-scale empirical measurement. We examine the *end-to-end* process: from leaking the login credentials to the phishing websites, to analyzing the phishing servers and phishing kits, and monitoring attacker's exploitation activities using the leaked credentials. More importantly, for the first time, we want to measure the real-time credential sharing to third-party collectors at both client and server sides. Regarding monitoring the account exploitation, the most related work is a study from Google [10] that monitored the activities of manually hijacked Google accounts. Another study leaked email accounts to *underground forums* and monitored the account activities [30]. These works focus on the generic accounts hijacking, while we specifically focus on the account exploitation after the phishing attack as part of the end-to-end analysis.

2.3 Methodology Overview

In this section, we describe our methodology to track the information flow in each step in Figure 1. Here, we only describe the high-level idea, and leave the detailed design and analysis to the corresponding sections in the later part of the paper.

First, to track the information flow at step1, step2.1, and particularly step2.2, we design a measurement tool to automatically feed (fake) login credentials to real-world phishing websites via the login forms. The tool will also keep track any redirections and real-time credential sharing during this process (§3 and §4).

Second, to infer the information flow of step3.1, step3.2, and step3.3, we try to obtain the phishing kits from phishing servers and analyze how the phishing kits work. We extract the email addresses that first-party attackers use to collect the user credentials. We also perform a dynamic analysis in a sandbox to identify potential backdoors planted by third-parties (§5).

Third, to shed light on step4, we intentionally leak email addresses and their *real* passwords via phishing sites, and monitor how attackers would exploit the email accounts after the phishing. These "honey accounts" are created by ourselves and do not affect any real users (§6).

3 TOOL DESIGN & DATA COLLECTION

We start by introducing our measurement tool to track the information flow on the *client side*. Given a phishing website, our tool can automatically detect the login form, fill in the fake credential (email address and password), and submit the information to the phishing server. In this process, our tool records all the HTTP and HTTPS traffic and detect those that transmit the credential to remote servers. In the following, we describe the detailed designs of this tool, and how we collect our datasets.

3.1 Measurement Tool

Our tool is a web crawler implemented using Selenium[1]. It controls a headless ChromeDriver browser to complete a series of actions and records the network traffic in the ChromeDriver log.

Detecting the Login Form. We focus on phishing sites that collect login credentials, excluding those that collect other information such as credit card information or social security numbers. We detect the login form by looking for three fields: username, password, and the "submit" button. We look for related tags in HTML including `FORM` tags, `INPUT` tags and `BUTTON` tags. We also extract the form attributes such as `type`, `placeholder`, `name`, and `class`). We don't consider any read-only or invisible tags.

To make sure that the form is indeed a login form instead of other irrelevant forms (*e.g.*, searching bar, survey forms), we compile a list of login related keywords and search them within the form attributes. We select keywords manually analyzing the login forms of 500 randomly phishing websites. In total, we select 40 keywords including 14 keywords for username (*e.g.*, "user name", "id", "online id", "email", "email address"), 8 keywords for password (*e.g.*, "password", "passwd", "passcode"), and 18 keywords for the submit button (*e.g.*, "log in", "sign in", "submit"). The main challenge is that phishing websites often have unconventional designs, or even intentionally hide keywords to evade detection [36]. It is not always possible to locate all three fields. Below, we list the key problems and how to address them.

- **Keywords in images**: The most common challenge is that attackers use an image to contain the "Login" keyword for the submit button, instead of placing the keyword to the `placeholder`. Our solution is to use the Tesseract Open Source OCR Engine[2] to extract the texts from images, and then perform the keyword search.
- **No FORM tags**: Phishing pages may intentionally leave out the `FORM` tags (to evade detection). Our solution is to search `INPUT` tags and keywords in the whole HTML page, instead of just within the `FORM` tags.
- **Two-step login**: In some phishing pages, users need to enter the username on the first page, and type in the password on the next page. Our tool can handle two-step login by tracking the log-in progress.
- **Previous unseen keywords**: the keywords may occasionally fail to match the corresponding input fields. To increase our success rate, we perform a simple inference based on the order of input fields. For example, if the username and button fields are matched, then we guess the unmatched input field in the middle is for the password.

Filling in the Fake Credential. After detecting the login form, our tool will automatically fill in the username and password fields and click the submit button. The username is an email address that belongs to us. The password is a random string of 8 characters which is uniquely created by us. The unique password is helpful later to detect the network requests that send out the password. This email address is never used to register any online account. The password is also not the real password for the email address. In this way, we make sure the leaked information would not affect any real users. We test the tool on 300 phishing sites (different from those that contributed the keywords). We show that the tool has a success rate of 90% to complete the login.

Here, we also want to make sure that using *fake credentials* does not affect our measurement result. We did a small experiment to see if the phishing site would react to real and fake password differently. We create 4 real accounts with PayPal, Microsoft, LinkedIn, and AT&T respectively. Then we select 60 live phishing websites from eCrimeX that impersonate these brands (15 websites per brand). We feed the real and fake passwords in separate runs, and find that the collected network traffic has no difference.

3.2 Data Collection

Using the measurement tool, we collect data from August 2018 to January 2019 by crawling 4 large phishing blacklists: PhishTank, PhishBank, eCrimeX, and OpenPhish. The detailed data statistics are shown in Table 1. For each phishing URL, all four blacklists share the timestamp when the phishing URL was reported/detected. Three of the blacklists also show the target brand (or website) that the phishing page is trying to impersonate. OpenPhish shares the target brand information only for the premium API (not the free-API we used). We notice that many phishing URLs become

[1] https://www.seleniumhq.org/

[2] https://github.com/tesseract-ocr/tesseract

Blacklist	Crawling Time Span	Target Brand	Detection Time	# All	# Live	# w/ Login Form	# Success
OpenPhish	09/24/2018 - 01/03/2019	✗	✓	75,687	44,553	24,202	19,720
eCrimeX	08/20/2018 - 01/03/2019	✓	✓	65,465	33,319	21,161	19,172
PhishTank	09/24/2018 - 01/03/2019	✓	✓	50,608	41,682	7,406	6,430
PhishBank	09/24/2018 - 01/03/2019	✓	✓	3,093	2,027	1,010	864
Total	08/20/2018 - 01/03/2019	–	–	179,865	110,934	47,703	41,986

Table 1: Dataset summary.

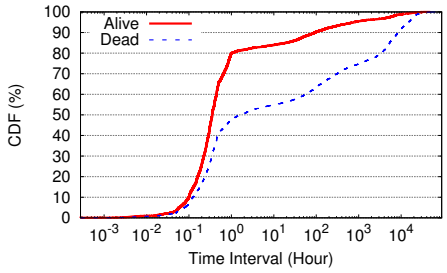


Figure 2: The gap between the time when a URL was blacklisted and the time when our crawler visited the URL.

inaccessible quickly after they are blacklisted. To interact with the *live* phishing server, we build a crawler to fetch phishing URLs from the four blacklists every 30 minutes. Then we immediately use our measurement tool to load the phishing page, feed the fake credential, and record the network traffic.

We also considered that situation where the phishing servers use cloaking techniques. More specifically, the phishing server may check the IP and User-Agent of the incoming request to see if the request is coming from a university, a security company, or a web crawler. In those cases, the phishing server may drop the request or return a benign page to avoid being detected. As such, we put our crawler behind web proxies and use a realistic User-Agent.

As shown in Table 1, we collected 190,087 unique phishing URLs (after removing duplicated URLs between the four blacklists). Among them, 68,751 (38.26%) are "dead", and the rest 110,934 (61.74%) are still alive. Figure 2 shows that the live pages are typically more recently-reported compared to the dead ones. 80% of the live pages were reported just 1 hour ago (by the time we visited the pages), while the dead pages were reported much earlier.

Login Results. Not all the live URLs are still phishing pages. In fact, many of the live URLs have been reset to legitimate/blank pages. Among 110,934 (61.74%) live URLs, only 47,703 (26.55%) still contain a login form. We use our measurement tool to feed the fake credentials to and record all the network traffic. Out of the 47,703 phishing sites, we successfully submitted the login form for 41,986 sites (88.01%), We manually checked the pages with failed logins. Some of the forms not only asked for username and password, but also required answering security questions by clicking a drop-down list. Other failure cases are caused by the special format constraints for the input data. We admit that there is still room for improving our measurement tool.

Identifying Relevant Network Traffic. Among all the network requests, we look for those that contain the seeded password. We consider both `POST` and `GET` HTTP/HTTPS requests. We expect that some phishing pages may encode or hash the credentials before transmission. As such, in addition to matching the plaintext, we also attempt to match the hashed/encoded versions of the password. We apply 31 hash/encoding function on the password and look for a match in the traffic (Table 2). After the filtering, we identified 41,986 network requests that contain the leaked password (either plaintext or hashed).

Hash or encoding functions (31 in total)
MD2, MD4, MD5, RIPEMD, SHA1, SHA224, SHA256, SHA384, SHA512, SHA3_224, SHA3_256, SHA3_384, SHA3_512, blake2b, blake2s, crc32, adler32, murmurhash 3 32 bit, murmurhash 3 64 bit, murmurhash 3 128 bit, whirlpool, b16 encoding, b32 encoding, b64 encoding, b85 encoding, url encoding, gzip, zlib, bz2, yenc, entity

Table 2: Functions used to obfuscate login credentials.

Rk.	Domain Name	# Unique URLs	Category
1	kylelierman.com	3,257 (6.82%)	Uncategorized
2	datarescue.cl	545 (1.14%)	Phishing & frauds
3	psycheforce.com	519 (1.09%)	Sex Education
4	4-6-3baseball.com	447 (0.94%)	Web Hosting
5	serveirc.com	424 (0.89%)	Dynamic DNS
6	galton.pila.pl	303 (0.63%)	Retail and Wholesale
7	lexvidhi.com	287 (0.60%)	Business Marketing
8	xsitedleadpages.com	262 (0.55%)	Uncategorized
9	stcroixlofts.com	233 (0.49%)	Dynamic Content
10	colorsplashstudio.com	230 (0.48%)	Blogs & shopping

Table 3: Top 10 domains of phishing URLs.

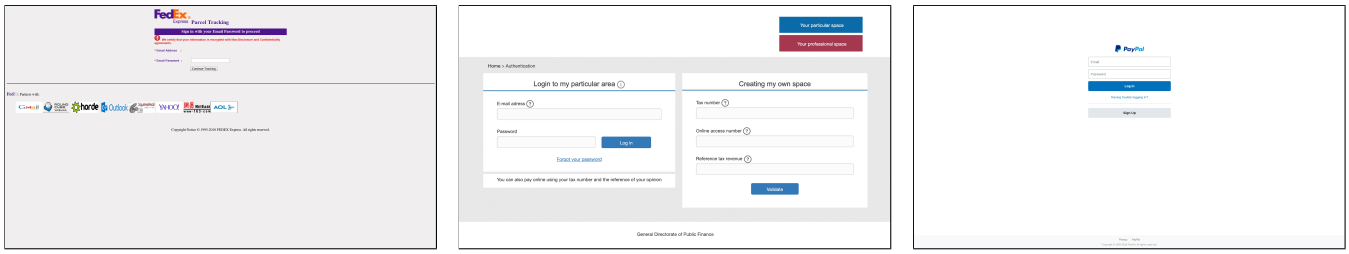
4 CLIENT SIDE ANALYSIS

We now analyze the collected dataset to examine the various aspects of the phishing websites including their target brands, domains and server geolocations. Then we inspect the information flow to understand how the login credentials are shared with third-party information collectors. The analysis of this section is based on the 47,703 phishing sites with a login form.

4.1 Understanding Phishing Sites

HTTPS Scheme. HTTPS is already widely used by the phishing sites. Among the 47,703 sites, 16,128 (33.81%) are hosting the phishing pages under HTTPS. We suspect that HTTPs helps to further deceive the users. More specifically, most modern browsers display a green padlock as the security indicator for HTTPS sites (with a valid certificate). This means, if a phishing site enables HTTPS, the green padlock would also show up when a user visits it. This could give the user a false sense of "security" given that user may not fully understand the meaning of the security indicator [18].

Domain Analysis. The 47,703 phishing sites are hosted under 24,199 full qualified domain names (FQDNs) which correspond to



(a) acm.org (b) sigcomm.org (c) nsw.gov.au

Figure 3: Compromised domains and their hosted phishing pages.

Domain	Alexa rank	# URLs
archive.org	269	1
bathandbodyworks.com	1,224	4
etherscan.io	3,162	3
nsw.gov.au	3,182	1
acm.org	3,676	11
tillys.com	9,506	1
krakow.pl	10,902	5
ugm.ac.id	11,198	1
kemkes.go.id	12385	4
mun.ca	13036	1

Table 4: Compromised domains that host phishing pages.

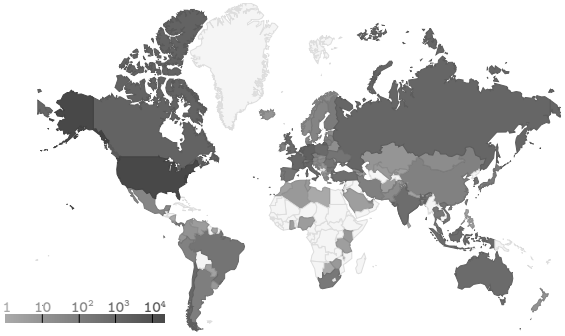


Figure 4: Geolocation distribution of phishing URLs.

16,939 unique domain names. Table 3 shows the top 10 domains ranked by the number of unique phishing URLs. There is no single domain that has a dominating contribution to the phishing URLs.

Interestingly, 417 domains are ranked within Alexa top 1 million[3]. We then manually investigate those domains, and classify them into four categories: 159 domains belong to web hosting services, 3 domains belong to dynamic DNS services, and 31 domains belong to URL shortener services. The rest 224 domains can not be easily categorized since they look like good websites that got compromised. In Table 4, we list the top 10 domains (based on their Alexa ranking) that are likely compromised for phishing.

Figure 3 shows three examples of compromised websites. Figure 3a is a phishing page hosted under `acm.org`. The phishing URL is "`http://iccps.acm.org/admin/.certified/***`" deployed under the ICCPS conference site to impersonate the FedEx website. Figure 3b is a phishing URL "`http://conferences.sigcomm.org/css/***`" hosted under the SIGCOMM conference website to impersonate a tax agency in France. Figure 3c is a phishing URL hosted under a government website of New South Wales in Australia "`http://councillorportal.ashfield.nsw.gov.au/Service/***`" to impersonate Paypal.

Geolocation Analysis. We further examine the geolocation of the phishing servers[4]. In this analysis, we do not consider phishing pages under web hosting services or compromised domains since these servers are not dedicated phishing servers. In total, we have 10,192 unique IP addresses, and their geolocation distribution is shown in Figure 4. The majority of the phishing sites are hosted in North America and Europe, especial in the United States. This result, in part, can be biased due to the fact that the phishing URLs are collected from four US-based phishing blacklists.

Target Brands. The phishing sites are impersonating a wide range of popular "brands". Recall that three of the four blacklists provide the target brand information, which covers 28,614 URLs (59.99% out of 47,703). For the rest 19,089 phishing URLs, we need to identify the target brands by ourselves. Our method is based on those in [40, 42]. The intuition is that a target brand that the phishing website is impersonating is typically more popular (*i.e.*, ranked higher in the search engine). For each of the 19,089 phishing pages, we first apply the OCR technique [23] to extract keywords from the webpage screenshot. Here, we use screenshots instead of the HTML file because attackers often use obfuscation techniques to hide the keywords in HTML [36]. Then we use RAKE (Rapid Automatic Keyword Extraction) [32] to extract keywords from the texts to remove less important keywords (*e.g.*, stop-words). We search the keywords using Google, and take the first returning page as the target brand. For example, if we search the keywords in Figure 3c, Google will return `paypal.com` as the first return result (*i.e.*, the target brand).

We evaluate this approach using phishing pages with known target brands. We first test the method on 500 phishing pages that impersonate Paypal, and get a 100% accuracy. Then we test the method on 500 phishing pages targeting Microsoft, and get a 99.8% accuracy. Finally, we test the method on 500 randomly phishing pages, which returns an accuracy of 88%. We believe this is good enough to proceed with our analysis.

In total, we find 298 unique target brands. The most popular target brand is Paypal, followed by Microsoft, AT&T, Desjardins, and Linkedin. We further categorize the target brands into 6 sectors based on their Standard Industrial Classification (SIC) code. We get SIC code information from siccode.com. As shown in Table 5, more than 40% of phishing URLs are targeting finance and insurance

[3]https://www.alexa.com/topsites

[4]For geolocation service, we use the GeoPlugin (https://www.geoplugin.com/).

Target Sectors	# Phishing Sites	# Brand	1st brand	2nd brand	3rd brand
Finance and Insurance	18,648 (39.09%)	150	PayPal (15,083)	Desjardins (960)	Wells Fargo (646)
Computer and Software Services	9,304 (19.50%)	58	Microsoft (4,484)	LinkedIn (761)	Yahoo (603)
Electronic and Communication	1,262 (2.65%)	23	AT&T (927)	Apple (161)	Verizon (29)
Transportation Services	583 (1.22%)	9	Federal Express (393)	DHL (13)	Delta (40)
Other	5,456 (11.44%)	48	eBay (159)	Craigslist (126)	IRS (124)
Not Applicable	12,450 (26.10%)	10	—	—	—

Table 5: Target sectors and top brands in each sector.

Format	Plaintext	URL Encoding	Other Encoding
# Phishing sites	6,324 (15.06%)	35,616 (84.83%)	46 (0.11%)

Table 6: Data format of credentials sent from the client-side.

# 3rd-parties	0	1	2	≥ 3
# Phish sites	39,967 (95.19%)	1,963 (4.68%)	48 (0.11%)	8 (0.02%)

Table 7: Distribution of third-party collectors. About 95% phishing sites don't have third-party collectors and they only send credentials to the original hosting domain.

services. Paypal alone is associated with 15,083 phishing URLs (32%). Note that 12,450 (26%) phishing sites don't have an informative target brand. For example, the blacklist may label them as "Generic" or "United States". Manual inspection reveals that these phishing sites are impersonating small organizations.

4.2 Client-Side Information Flow

In this section, we investigate the information flows of sending credentials from the client side. To identify HTTP requests containing user credentials, we follow the methodology discussed earlier in §3.2. Out of the 47,703 phishing sites with a login form, we are able to track credential information flow for 41,986 phishing sites.

Credential Sending Format. Recall that credential information could be transmitted in plaintext or using some encoding/hashing schemes (*e.g.*, MD5, SHA256). Table 6 shows statistics of different types of data formats used across phishing sites. Interestingly, most phishing sites (99%) use human interpretable formats (*i.e.*, either plaintext or URL encoding), and only a small fraction, 0.11% use other more advanced encoding schemes. This implies that most attackers did not try to obfuscate the information flow.

Identifying Third-party Collectors. Any domain that collects credential information, and is not a direct phishing server domain, is considered to be a third-party collector. In total, we identify 694 third-party collector domains that include 1,021 URLs. These are entities that collect stolen credentials, and would be a vital component to target while building phishing defenses.

But do all phishing sites share credentials with third-party collectors? Table 7 shows the distribution of phishing sites that share credentials with different number of third-party collectors. There are about 5% of phishing sites sharing credentials with third-party collectors from the *client side*. The percentage is not high, but there is a sizeable number. There are 2,019 phishing sites that interact with one or more third-party collectors. In fact, 56 phishing sites share with more than 2 third-party collectors.

Third-party Collectors vs. Phishing Sites. Next, we look at two aspects of third-party collectors that have implications for disrupting their network. *First*, do third-party collectors link with multiple phishing sites? If each third-party collector served a single phishing site, we would have to take down as many collector domains as the number of phishing sites. But we observe a different trend. Figure 6 shows the distribution of fraction of phishing sites covered by different external collectors. We find that the top 100 external collectors (out of 694) link with a majority, 68.76% of the phishing sites. Thus, even targeting a small fraction of external collectors can disrupt many phishing efforts.

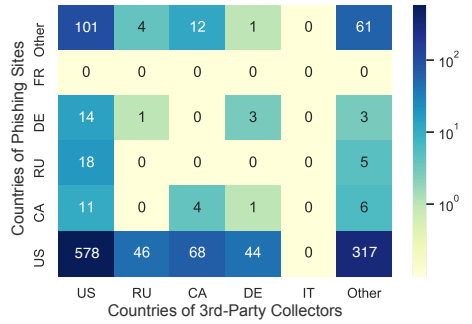


Figure 5: Countries of phishing sites and third-party collectors.

Second, we further examine the geographical locations of third-party collectors. Third-party collectors are spread over 37 countries, but 42% of them are located in the U.S. When third-party collectors are based in a country different from the phishing site they link with, it would require different law enforcement efforts to take down their domains. We analyze the relative locations of phishing sites and their associated third-party collectors. Among 1,408 IP address pairs made of phishing sites, and their connected collector domains[5], 44% are co-located in the same country. A significant fraction of this number can be attributed to the U.S.—96% of co-located pairs are located within the U.S. The remaining 56% non-co-located pairs include phishing sites that are spread over 52 countries, and collectors over 37 countries. We also note that a significant fraction, 88% of non-co-located pairs involve phishing sites or collectors based in the U.S. The detailed breakdown for is shown in Figure 5. We only show the top 5 countries of phishing servers and third-party collectors and group the rest into "other". Overall, this means that a majority of pairs are not based in the same country, and this could raise challenges to disrupt their network.

How Reputed are Third-Party Collectors? We investigate whether the third-party collectors are already known malicious entities or those with poor reputation.

[5]In total, there were 2,170 pairs, but we were unable to determine the geolocation for all of them.

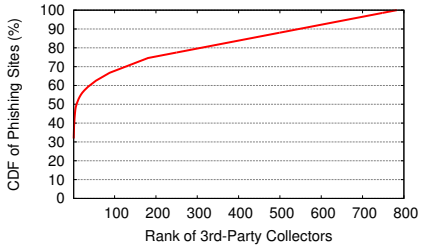


Figure 6: Distribution of fraction of phishing sites that connect to different third-party collectors. On the x-axis, third-party collectors are ranked based on # of phishing sites connected.

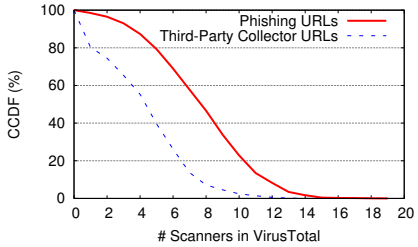


Figure 7: CCDF of Number of VirusTotal scanners that flagged the given URL as malicious. The majority of the third-party collectors are already flagged by VirusTotal scanners.

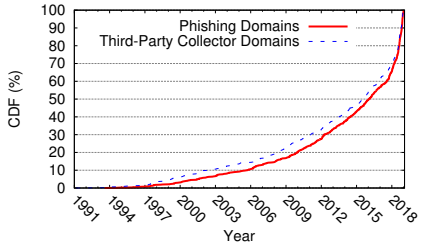


Figure 8: Registration time of phishing domains and third-party collector domains. Third-party collector domains have a similar distribution with phishing domains.

	# Phishing Sites w/ Third-party Collectors	# Third-party Collector URLs
Total	2,019	1,021
"Phishing Site"	1970 (97.57%)	823 (80.63%)
"Malicious Site"	1,840 (91.13%)	777 (76.10%)
"Malware Site"	239 (13.13%)	176 (17.24%)

Table 8: Number of URLs detected by VirusTotal.

We start by analyzing the reputation of third-party collector domains using *The Talos IP and Domain Reputation Center (by Cisco)*[6]. The Talos IP and Domain Reputation Center is a real-time threat detection network. They provide a reputation score of "Good", "Neutral" and "Poor". Here "Good" means little or no threat activity has been observed. On the contrary, "Poor" indicates a problematic level of threat activity has been observed, while "Neutral" means the domain is within acceptable parameters. Note that "Neutral" is a common case for most domains, even well-known ones such as `facebook.com`. Among all 694 third-party collector domains, we obtain reports for 508 (73.20%) domains. We find that 14 of them are labeled "Good", 146 are "Poor" and the rest 348 are "Neutral".

We take a closer look at these scores—*First*, it is interesting to see that a significant fraction, 29% of domains already have poor reputation, but still managed to stay alive and form a collector network. *Second*, it is surprising to see 14 domains marked as "Good". We find these are indeed legitimate domains, *e.g.*, `delta.com`, `google.com`, `doubleclick.net`, `dropbox.com`. On examining the HTTP logs for these "Good" collectors, we find there are different reasons for them acting as third-party collectors. For example, certain phishing sites were sending the credentials to the legitimate sites that they were trying to impersonate (*e.g.*, `delta.com`). We suspect that they were trying to check the validity of credentials. Some good sites were collecting credentials because they were used by attackers as a web hosting service (*e.g.*, `dropbox.com`). Finally, popular ads platforms or tracking services such as Google Ads and `doubleclick.net` also received the stolen credentials. A close inspection shows that the phishing sites were connecting to these tracking services to keep track of the number of victims. While doing so, the stolen credential was "accidentally" placed within the referer URL of the request.

Only analyzing domain reputation does not provide the full picture. There can be legitimate domains that host malicious URLs. We leverage VirusTotal[7] to scan external collector URLs. VirusTotal has been widely used by the security community in prior work [28, 36]. For each submitted URL, VirusTotal provides a report from 66 diverse scanners that may classify it into one or more categories that indicate whether a URL is problematic, clean or unrated. Problematic categories include "Malware site", "Phishing site", "Malicious site", "Suspicious site", and "Spam site".

Figure 7 shows the distribution of collector URLs detected by VirusTotal scanners that fall into any one of the problematic categories. A small fraction, 16% of URLs are not flagged by any scanner, and will likely remain under the radar for a long time. On the other hand, a large majority, 84% of collector URLs are classified as problematic by at least one scanner. Table 8 shows a further breakdown of collector URLs that are flagged by at least one scanner. Interestingly, 81% of them are flagged as 'Phishing sites'. This suggests the possibility of a network of phishing sites that exchange credential information with each other.

To summarize, while a majority of third-party collector domains do not have a poor reputation, a large majority of their URLs are already known to be problematic, *e.g.*, for phishing. In spite of the poor URL reputation, it is surprising that these collector URLs are still alive. To understand the age of the collector domains, we examine WHOIS records to determine their domain registration dates. Figure 8 shows that the distribution of domain registration time of third-party collectors is quite close to that of the phishing servers. Many of the collector domains are actually aged domains. 20% of them were registered 10 years ago. About half of them were registered before 2016. This suggests that the collector network has largely remained undisrupted.[8]

The top information collectors ranked by the number of phishing sites they serve is presented in Table 9. The largest information collectors here is "w32.info". This site was once hosting many phishing kits for downloading (not anymore). We confirm this by checking the achieved versions of this website[9]. It is possible that the kit developers were using this site to collect a copy of the stolen credentials from people who use their kits to perform phishing. We

[6] urlhttps://www.talosintelligence.com/

[7] https://www.virustotal.com

[8] We removed known web hosting domains (as reported by Alexa top 1 Million) from this plot to avoid a possible wrong interpretation. Malicious collector URLs hosted on a legitimate webhosting service would show up as being long-lived, while the exact age of the URL would be hard to determine.

[9] https://web.archive.org/web/20151128133828/http://w32.info:80/

Rk.	Third-party Collector	Phish URLs	Domain Category	Collector URLs
1	w32.info	731	Infection source	1
2	jquerymobile.ga	168	Uncategorized	2
3	ip-api.org	89	Geolocation API	1
4	serveirc.com	57	Dynamic DNS	57
5	imgur-photobox.com	50	Uncategorized	1
6	000webhostapp.com	28	Web hosting	26
7	ptpjm.co.id	17	known infection	3
8	servehttp.com	16	Dynamic DNS	8
9	redirectme.net	16	Dynamic DNS	16
10	fitandfirmonline.com	14	Uncategorized	14

Table 9: Top 10 third-party collectors.

also notice that web hosting services or dynamic DNS services are often used to collect credentials for multiple collector URLs (possibly for different attackers). One interesting case is `ip-api.org`, a website that provides a lookup service for IP geolocations. 89 phishing websites were sending stolen credentials to this server via "`http://cdn.images.ip-api.org/s.png`". We suspect that this service might have been compromised.

5 SERVER SIDE ANALYSIS

In this section, we move to the server side to analyze the information flow of credential transmission. The challenge here is that we don't have internal access to the phishing servers. Our solution is based on the fact that some (careless) attackers may have left the phishing kit in publicly accessible locations on the phishing server [12]. As such, we attempt to retrieve these phishing kits and infer the server-side information flow by combining static and dynamic analysis.

5.1 Collecting Phishing Kits

We search for phishing kits on servers that host the phishing websites. Unlike §4, we inspect all 179,865 phishing URLs (*i.e.*, not just sites that were still alive) for possible phishing kits. The main reason is that even if a phishing site has been disabled[10], it is possible that phishing kits are still left accessible on the server [29].

Since we have no knowledge of possible file names to query for (on the phishing server), we start with phishing servers that enable directory listing to obtain a list of files available on the server. Prior work suggests that phishing kits are usually compressed/archive files (*e.g.*, zip, tar, rar) [12]. For each phishing site URL, we do the following steps: (1) Check if directory listing is available for each path segment in the URL (*i.e.*, separated by '/'). (2) If we find a directory listing, we download all compressed/archive files. (3) For each downloaded file, we decompress it and check the PHP/Python/Ruby/HTML files to make sure it is indeed a phishing kit. To further increase our chance to retrieve more phishing kits, we identify the most frequent 50 kit names (based on the first 1000 kits downloaded earlier). Then given a phishing URL, we exhaustively query each path segment for these 50 file names, in addition to checking the directory listing. This helps us to obtain kits from servers that disabled the directory listing.

We applied the above method to querying 179,865 phishing sites, and obtained 2,064 phishing kits in total. Compared to earlier work [2, 19], our hit rate for finding a phishing kit on phishing servers is lower—we observe a hit rate of 1.15%, compared to 11.8% in prior work. We suspect that phishers are being more careful, and avoid leaving publicly visible traces of their malicious activity.

[10]By disabled we mean the phishing site has been reset to a legitimate website by phisher or the web administrator.

5.2 Server-side Information Flow

Unlike client-side analysis, where we only investigate outgoing HTTP/HTTPS requests, information flow on the server side can use other channels too—via Email [19]. Our goal is to capture the information flow on the server side, and also detect those related to third-party credential sharing.

Identifying Third-party Collectors. On the server side, the stolen credentials can be sent to third-parties in addition to the attacker who deployed the phishing kit. More specifically, prior work shows that phishing kits may contain backdoors [12] that allow third-parties to collect the stolen credentials. Often cases, the backdoors are stealthily inserted into the phishing kit code by the kit developers. When the kit is used by attackers to perform phishing, the kit developer also receives a copy of the credentials.

To differentiate backdoor collectors, we conduct both dynamic and static analysis. The methodology is inspired by that in [12]. The assumption is that backdoors are usually planted stealthily, which are not directly visible in plaintext in the kit code. As such, we first apply *static analysis* by performing a text search within files in a kit to identify email addresses, and URL endpoints (for HTTP requests) that collect credentials. Then we put the phishing kit in a sandbox for a *dynamic analysis* to capture all the outbound HTTP and email traffic that transmit the stolen credentials. Any collector identified from dynamic analysis, but not identifiable via plain text search through static analysis, can be considered to be a backdoor collector (*i.e.*, the third-party). Note that throughout our dynamic analysis, we did not observe any outbound HTTP/HTTPS traffic from any phishing kits. For brevity, we only introduce the details of the email channel analysis below.

Static and Dynamic Analysis. Our *static analysis* is based on a simple method to extract the collectors in plaintext. The idea is to locate the `mail(to,subject,...,header)` function and identify their "to" and "header" variables. The "to" address is considered to be a collector on the server side. Out of 2,064 phishing kits in total, we successfully detected email addresses in 1,974 phishing kits. In total, we extracted 1,222 valid email addresses (as receivers).

For the *dynamic analysis*, we build up an Apache web server and upload all phishing kits to it. We record all the outbound traffic but block the corresponding ports (*e.g.*, port 25 for email) to avoid actually sending data to the attackers. For each phishing kit, since we do not know which files build the phishing pages, we run our tool described in §3.1 to detect login forms to locate the phishing page. Then like before, we use our measurement tool to automatically fill in the username and password, and submit the information to the experimental server. To capture the server-side actions, we dump all the emails in the mail queue and all the HTTP logs.

We run the dynamic analysis on all of the 2,064 phishing kits. Using tools described in §3.1, we successfully logged into 1,181 (57%) phishing kits. Note that for 88 (9%) of these phishing kits, we did not find any outbound emails. It is possible that these attackers would rather log into the phishing server to retrieve the stolen credentials

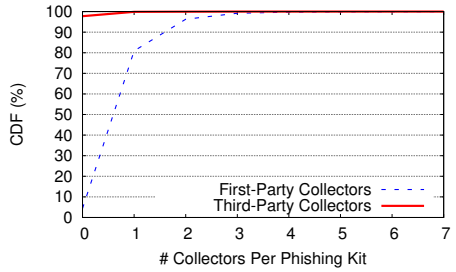


Figure 9: Number of server-side collectors per phishing kit.

Rk.	3rd-parties	# Phishing Kits	# Domains
1	equallib12@gmail.com	10	6
2	hhforexxx@gmail.com	5	4
3	ebay1235x@gmail.com	4	2
4	sesurityas@yandex.com	3	2
5	boxnr1234@gmail.com	2	2

Table 10: Top 5 third-party collectors on the server side.

Rk.	1st-parties	# Phishing Kits	# Domains
1	nosaplanter@gmail.com	76	10
2	chrismason601@gmail.com	27	6
3	mrgodwin2233@gmail.com	21	3
4	work-hard@dreambig.com	15	13
5	samzysoprano2@gmail.com	13	6

Table 11: Top 5 first-party collectors on the server side.

(step3.2 in Figure 1). For the rest of the phishing kits, we search the leaked password in their outbound emails to make sure they are sending the stolen credentials. We only find 6 emails that did not contain the password (the emails were for status reports). For these 1,093 phishing kits, we compare the result of dynamic analysis and that of static analysis, and find 46 phishing kits with backdoor emails (4.2%).

Server-side Collectors. Figure 9 shows the number of server-side collectors per phishing kit. Each collector is identified as a receiver email address. Most phishing kits (96%) do not have a backdoor (third-party) collector. Among the 46 kits that have a backdoor, there is usually only one backdoor collector per kit. In total, there are 24 unique backdoor email addresses. Table 10 further displayed the top 5 third-party email addresses, ranked by the number of associated phishing kits. Some collectors (*e.g.*, `equallib12@gmail.com`) were embedded into multiple phishing kits.

Regarding the first-party collectors, Figure 9 shows that most phishing kits have one first-party collector, but about 20% kits have more than one collectors. As shown in Table 11, some of the first-party collectors are associated with multiple kits, which indicates coordinated phishing campaigns, *i.e.*, one attacker deployed the kits onto multiple phishing servers.

Comparing Client-side and Server-side Flows. We next compare the flows of the client side with those of the server side. Among 179,865 phishing URLs, we find 2,064 phishing kits from 1,286 phishing server domains. 437 (34.0%) of these phishing domains overlap with those live phishing sites analyzed in §4. Given a phishing server domain, we examine the number of client-side collectors and the number of server-side collectors (combining first- and third-parties). The results are shown in Table 12. The majority of the domains (296 domains, 67.7%) has one collector on the server-side without any client-side collector. Only a small number of domains (7 domains, 1.6%) have collectors at both sides. There are 18 domains that have no collectors at neither sides. In this case, attackers would need to login to the phishing server to collect the stolen credentials.

Client Collectors \ Server Collectors	0	1	2	>2
0	18	296	94	29
1	3	4	2	1
2	0	0	0	0
>2	0	0	0	0

Table 12: Collectors on both client and server side.

6 POST-PHISHING EXPLOITATION

So far we explored different ways of information leakage, but what would the attackers do with the stolen credentials? To shed light on the post-phishing exploitation activities, we set up *honeypot accounts* whose credentials are intentionally leaked by us to phishing sites. Then by developing a honeypot account monitoring system, we can uncover activities that access and use our honeypot accounts. This idea is inspired by prior work by Onaolapo et al. on monitoring the activities after account hijacking [30]. While Onaolapo et al. investigated account hijacking more broadly, our focus is specifically on the fate of accounts that have credentials stolen by phishing attacks. This analysis helps to complete a comprehensive end-to-end view of what happens after the information leakage.

6.1 Experiment Setup

Our goal is to understand the post-phishing exploitation on the *email accounts*. For example, suppose the attacker sets up a phishing site to impersonate "`paypal.com`" to steal PayPal accounts, we expect the attacker will first try to login to the PayPal account (*e.g.*, to steal money). As the second-step exploitation, the attacker may also try to hijack the email account that is associated to the PayPal account using the same password (assuming users reuse the password). Intuitively, the email account can be used to hijack other online accounts registered under this email (*e.g.*, through password reset), and thus has value. In the following, we set up honey email accounts to study this second-step exploitation.

Honeypot Accounts Setup. Our honeypot accounts include two different types of email accounts: Gmail and ProtonMail[11]. Gmail is a large popular email service provided by Google, while ProtonMail is a less popular end-to-end encrypted email service based in Switzerland. We manually created 100 Gmail and 50 ProtonMail accounts and assigned them random combinations of popular first and last names. To make the freshly-created email accounts believable and realistic, we populated them with emails from the public Enron email dataset [24]. Enron dataset contains emails sent by executives of the energy corporation Enron, and was publicly released as evidence for the bankruptcy trial of the company. To avoid causing suspicion from attackers, we applied the following method

[11] https://protonmail.com/

Id	Honey Account	Phishing URL	Country	Target Brand	Leak Time	First Login Time	#Login (#IP)	Login Country	#Email Read
1	Gmail	https://donboscoschoolsindia.com/sigin/customer_center/customer-IDPP00C323/myaccount/signin/	US	PayPal	11-09-2018 17:30	11-09-2018 18:05	9 (1)	US	6
2	Protonmail	http://ceoclubscollections.com/yscom2/Login/122b53d78b50b4c05f117f4fab4bfb8c/	US	PayPal	11-28-2018 17:36	11-29-2018 9:28	1 (1)	MA	1
3	Protonmail	http://www.radioinkasurfm.com/new/	DE	Generic Email	10-26-2018 15:27	10-26-2018 16:50	7 (4)	NG	0
4	Protonmail	https://uddoktahub.com/bplbuzz/wp-content/login.php	US	LinkedIn	10-26-2018 15:21	10-26-2018 20:15	6 (4)	NG, CN	4
5	Protonmail	https://referring.ga/dinn/log/linkedin/Linkedin/SignIn.php	US	LinkedIn	10-26-2018 15:20	10-28-2018 14:14	1 (1)	GH	2
6	Protonmail	https://withium.xyz/rex/signin.html	US	Microsoft	12-21-2018 1:08	12-22-2018 1:46	1 (1)	PK	2
7	Protonmail	http://www.cafedepot.com/christmasgifts/drop/Login.html	US	ABSA Bank	12-21-2018 1:06	12-24-2018 19:14	17 (10)	NG, US, CA	1

Table 13: Account exploitation activities in our honey accounts.

to modify those emails before putting them into the inbox of the honey accounts. First, we translated the old Enron email timestamps to recent timestamps slightly earlier than our experiment start date. Second, we replaced the sender domain with some popular email domain such as `gmail.com` and `outlook.com`. Third, we replaced all instances of "Enron" with a fictitious company name.

For all the honey accounts, we did not enable any type of two-factor authentications. This is to make sure the attackers can perform the login using username and password alone. We also perform a quick confirmation test. We attempted to log in to these honey accounts from different countries (using web proxies), and found that the logins were all successful.

Leaking Real Credentials. To leak the credentials of the honey accounts, we choose phishing sites from 4 categories based on their target brands: "PayPal", "Finance and Insurance", "Computer and Software Services", and "Others". We treat PayPal as a separate category since a major portion of the phishing sites target the PayPal brand (see Table 5). Phishing sites that target "Electronic and Communication" and "Transportation Services", account for less than 10% of our data, so we count them as "Others". We choose 150 phishing sites (roughly 40 phishing sites from each category), and leak one email credential to each site (thus using all our honeypot accounts). The freshly created honey account is exclusively leaked to one phishing site only, which helps us to accurately attribute the exploitation activities to the original phishing site.

Monitoring Infrastructure. We develop our own monitoring system to collect data about the account activities. For Gmail, we obtain the information of recent logins from the "last account activity" page[12]. Each login record contains the IP, device information, and timestamp of login. Similarly, ProtonMail also provides such logs in its security settings. For both providers, we develop a script that can automatically login to each account and crawl the information of recent login records. To further monitor attacker activities after login, we obtain the scripts used in [30] to scan the inbox and detect any changes. The activity logs are periodically sent to a separate email account (created for data collection) under our control.

[12]https://support.google.com/mail/answer/45938?hl=en

Ethical Considerations. The above experiment requires ethical considerations. *First*, all the honey accounts are freshly created by us, and the experiment would not affect any real users of Gmail or ProtonMail. *Second*, to run this experiment, we need to give attackers the access to honey accounts. A potential risk is the attackers may use the honey accounts for other malicious activities. To reduce the risk, we restrict our ourselves to a *small-scale* experiment. This means attackers do not get many accounts. In addition, all the historical emails and previous contacts in these accounts are synthetically created. This means attackers cannot use these honey accounts to further phish their contacts (a common way of performing spear phishing). Throughout the experiment, these honey accounts are never used to send any emails. *Third*, we make sure to delete the honey accounts after the experiment.

6.2 Activities on Honeypot Accounts

Starting in November 2018, we performed the experiment by manually leaking the honey account credentials (email address and password) to different phishing sites. The credentials were not all leaked at once. After the credentials were leaked, we monitored the honey account for at least 50 days. Out of the 150 honey accounts, we observe that 7 accounts (leaked to different phishing sites) have received logins. Table 13 summarizes the account activities.

Overall Observations. First, we observe that the exploitation happened very quickly after the credential leakage. It can be shortly within an hour or only after 1–2 days. Second, most of the times, the attackers logged in from countries different from where the original phishing sites were located. Third, for some honey accounts, there are often multiple login attempts from different IP addresses. The result echoes our early analysis that the stolen credentials can be leaked or shared to multiple attackers.

Detailed Activity Analysis. Next, we provide more detailed results for each of the honey accounts.

- **Account-1** is the only Gmail account that received logins. The original phishing site is hosted in Arizona, US. After 35 minutes of the credential leakage, attackers first logged in

from Boston, US. After that, the attacker registered an Amazon Web Service (AWS) account using the honey account which left a confirmation email in the inbox. A few minutes later, the honey account received an email that indicated AWS payment failure. In the following 5 days, the attacker kept logging into the account for 8 additional times from the same IP address, but did not have other observable activities.

- **Account-2, 5 and 6** has one login each. All three phishing sites are hosted in the U.S., but all the logins are originated from a different country—Morocco, Ghana, and Pakistan. In addition, in Account-2, 5, and 6, the attacker read 1, 2, and 2 emails each, respectively. We suspect they were searching for something of value in the account, *e.g.*, banking information, social security numbers, credentials to other services.
- **Account-3** has 7 logins using 4 IPs from Nigeria, despite the phishing site being hosted in France. We did not observe any patterns in account access; they did not check the account on consecutive days.
- **Account-4** is more interesting as we observe activities from 2 different countries. After about 5 hours of the leakage, the attacker first logged in from Nigeria. Then 3 days later, we saw two logins from Beijing, China. Half a month later, the first attacker from Nigeria (*i.e.*, using the same IP) checked the account again. This phishing site is also hosted in the US. It is possible that the credential is leaked to multiple attackers during phishing[13]. The attackers read 4 emails.
- **Account-7** is another one with login activities from different locations—5 different cities (3 countries). There are 17 different logins over a period of one month. First, the attacker logged in from Lagos, Nigeria. Two days later, another attacker logged in from Atlanta, US. And then, on Jan 3, 2019, there were two logins from Burnaby, Canada and one from Miami, US. The last login was found from Los Angeles, US. We believe this could be evidence for credential sharing. Also, 1 email was read.

From our analysis, we conclude that attackers indeed log in to the email accounts and check whether they can find anything of value (by reading emails). Recall that the email accounts were not the initial targets of the phishing attack—the initial targets were online accounts of PayPal, LinkedIn, Microsoft. This explains why only 5% of attackers would go the extra miles to the hijacking of the associated email accounts. The irregular patterns of the account activities also suggest that the exploitation is likely done manually.

7 DISCUSSION

Implications of Results. Our measurement results have several key implications. *First*, credentials sharing happens throughout the phishing process at both client and server side, which exposes the stolen credentials to more malicious parties. The good news is that third-party sharing is not yet prevalent. *Second*, from the phisher's perspective, credential sharing can be both intended (*e.g.*, for validating the stolen credentials and tracking attack statistics) or unintended (*e.g.*, due to backdoors planted by phishing kit developers). *Third*, from the defender's perspective, client-side phishing efforts are easier to detect. In §4, we find that over 80% of client-side 3rd-party collectors are already flagged by VirusTotal. However, the problem is that they were not effectively taken down (they are usually in a different country compared to the phishing site). Nevertheless, defense schemes can still add these domains into local network blacklists to block credential sharing. *Fourth*, server-side efforts are harder to measure and disrupt. Web-hosting platforms can significantly contribute to phishing defenses by searching for phishing kits, and take action to block such sites, or issue a warning to the site moderator (in case they were compromised).

Using third-party Sharing Channel for Defense. We believe that third-party sharing (and backdoors) can also be used by defenders for good purposes. For example, for known third-party collectors (backdoor email addresses or client-side collectors), instead of directly shutting them down, the defenders (*e.g.*, law enforcement, service providers) may keep them alive but *take away the ownership from the malicious parties*. For example, Google can block the attacker from accessing the Gmail account that acts as the backdoor collector. Then Gmail's security team can keep this account alive as a vantage point to monitor the phishing activities from the same class of phishing kits. The benefit is that whenever the corresponding phishing kits are used to perform phishing in the wild, the defenders can directly pinpoint the location of the attackers (since the phishing kits will contact the backdoor collector). In addition, the defender will also receive a copy of the victim list, which allows defenders to take early actions to alert the victims.

Limitations. Our study has a few limitations. *First*, while we obtain a complete view of client-side sharing, we still do not have the complete picture on the server-side. We only observe instantaneous sharing of credentials on the server-side, *i.e.*, as soon as the credentials are received by the server. This is a limitation because it is still possible that the server-side scripts may send credentials at a later point of time, *e.g.*, based on pre-set timers. Unfortunately, given the large number of phishing kits we need to test, we cannot monitor them for a long time. *Second*, our server-side analysis is based on the phishing kits—we have no information about phishing sites that do not leave kits publicly accessible. *Third*, we acknowledge that our dataset is biased due to the use of the four phishing blacklists which are skewed towards English speaking countries. However, our dataset still covers phishing sites that target major sectors and a broad set of brands (Table 5). *Fourth*, our view of post-phishing activities is limited due to the small scale of the experiment. For ethical concerns, the small scale is intended.

8 RELATED WORK

Password Leakage. While existing works have studied password leakage [11] and password re-use [13, 34, 37], credentials sharing during the *phishing process* wasn't well understood. A related study [35] examined the potential victims of off-the-shelf keyloggers, phishing kits and previous data breaches. They explored how stolen passwords enabled attackers to hijack Gmail accounts.

Phishing Kit. Zawoad et al. found 10% of phishing sites had evidence of using phishing kits [41]. Phishers' motivation and thought processes are inferred by analyzing phishing kits [2, 12, 25, 29]. Previous work has also sandboxed phishing kits to monitor their

[13]We cannot confirm whether there was server-side sharing since the phishing kit was not accessible. We did not observe any client-side sharing on this phishing site.

mechanisms and behavior of criminals [19]. Phishers usually use phishing kits to create a series of similar phishing pages [9].

Phishing Detection & Warning. Content-based detection methods have been studied extensively. Cantina and Cantina+ [40, 42] base their detection on DOM and search engines information. Researchers also looked into other detection methods based on visual similarities [38], URL properties [8, 27, 36], OCR features [6, 16], and user behavior patterns [15, 33]. Going deeper, phishing hosts have also been extensively studied including compromised sites [14] and malicious web infrastructure [26]. Phishing emails are used to distribute phishing URLs. Phishers can use email spoofing techniques [21, 22] or email header injection [31] to deceive users. Other researchers looked into the effectiveness of phishing websites warning and prevention in web browsers [7, 17, 39]. A key novelty of our work is to track the information flow for credential sharing across different phases of phishing.

9 CONCLUSION

In this paper, we performs an empirical measurement on the information flows of credential sharing during phishing attacks. Our analysis covers more than 179,000 phishing URLs (47,000 live phishing sites). We show that user credentials are shared in real-time to multiple parties at both the client side and the server side. Although third-party sharing exposes user credentials to even more malicious parties, we argue that defenders may make use of these channels to back-track phishing servers and alert phishing victims.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their helpful feedback. This project was supported in part by NSF grants CNS-1750101 and CNS-1717028, and Google Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agencies.

REFERENCES

[1] 2016. How John Podesta's Emails Were Hacked And How To Prevent It From Happening To You. https://www.forbes.com/sites/kevinmurnane/2016/10/21/how-john-podestas-emails-were-hacked-and-how-to-prevent-it-from-happening-to-you/.
[2] 2017. Phish in a Barrel: Hunting and Analyzing Phishing Kits at Scale. https://duo.com/blog/phish-in-a-barrel-hunting-and-analyzing-phishing-kits-at-scale.
[3] 2018. Data Breach Investigations Report. https://enterprise.verizon.com/resources/reports/dbir/.
[4] 2018. UnityPoint Health Notifies 1.4M Patients of Data Breach Caused by Phishing Attack. https://www.healthcare-informatics.com/news-item/cybersecurity/unitypoint-health-notifies-14m-patients-data-breach-caused-phishing-attack.
[5] 2019. The biggest healthcare data breaches of 2018. https://www.healthcareitnews.com/projects/biggest-healthcare-data-breaches-2018-so-far.
[6] Sadia Afroz and Rachel Greenstadt. 2011. PhishZoo: Detecting Phishing Websites by Looking at Them. In *Proc. of ICSC'11.*
[7] Devdatta Akhawe and Adrienne Porter Felt. 2013. Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In *Proc. of USENIX Security'13.*
[8] Aaron Blum, Brad Wardman, Thamar Solorio, and Gary Warner. 2010. Lexical feature based phishing URL detection using online learning. In *Proc. of AISec'10.*
[9] Jason Britt, Brad Wardman, Alan Sprague, and Gary Warner. 2012. Clustering Potential Phishing Websites Using DeepMD5. In *Proc. of LEET'12.*
[10] Elie Bursztein, Borbala Benko, Daniel Margolis, Tadek Pietraszek, Andy Archer, Allan Aquino, Andreas Pitsillidis, and Stefan Savage. 2014. Handcrafted Fraud and Extortion: Manual Account Hijacking in the Wild. In *Proc. of IMC'14.*
[11] Blake Butler, Brad Wardman, and Nate Pratt. 2016. REAPER: an automated, scalable solution for mass credential harvesting and OSINT. In *Proc. of eCrime'16.*
[12] Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2008. There Is No Free Phish: An Analysis of "Free" and Live Phishing Kits. In *WOOT'08.*
[13] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The Tangled Web of Password Reuse. In *Proc. of NDSS'14.*
[14] Joe DeBlasio, Stefan Savage, Geoffrey M Voelker, and Alex C Snoeren. 2017. Tripwire: Inferring internet site compromise. In *Proc. of IMC'17.*
[15] Xun Dong, John A Clark, and Jeremy L Jacob. 2008. User behaviour based phishing websites detection. In *Proc. of IMCSIT'08.*
[16] Matthew Dunlop, Stephen Groat, and David Shelly. 2010. GoldPhish: Using Images for Content-Based Phishing Analysis. In *Proc. of ICIMP'10.*
[17] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. 2008. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proc. of CHI'08.*
[18] Adrienne Porter Felt, Robert W. Reeder, Alex Ainslie, Helen Harris, Max Walker, Christopher Thompson, Mustafa Embre Acer, Elisabeth Morant, and Sunny Consolvo. 2016. Rethinking Connection Security Indicators. In *Proc. of SOUPS'16.*
[19] Xiao Han, Nizar Kheir, and Davide Balzarotti. 2016. Phisheye: Live monitoring of sandboxed phishing kits. In *Proc. of CCS'16.*
[20] Grant Ho, Aashish Sharma, Mobin Javed, Vern Paxson, and David Wagner. 2017. Detecting Credential Spearphishing in Enterprise Settings. In *Proc. of USENIX Security'17.*
[21] Hang Hu, Peng Peng, and Gang Wang. 2018. Towards Understanding the Adoption of Anti-Spoofing Protocols in Email Systems. In *Proc. of SecDev'18.*
[22] Hang Hu and Gang Wang. 2018. End-to-End Measurements of Email Spoofing Attacks. In *Proc. of USENIX Security'18.*
[23] Anthony Kay. 2007. Tesseract: an open-source optical character recognition engine. In *Linux Journal'17.* Belltown Media.
[24] Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *Proc. of ECML'04.*
[25] Luda Lazar. 2018. Our Analysis of 1,019 Phishing Kits. https://www.imperva.com/blog/our-analysis-of-1019-phishing-kits/.
[26] Zhou Li, Sumayah Alrwais, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2013. Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In *Proc. of IEEE S&P'13.*
[27] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. 2011. Learning to detect malicious urls. (2011).
[28] Rima Masri and Monther Aldwairi. 2017. Automated malicious advertisement detection using virustotal, urlvoid, and trendmicro. In *Proc. of ICICS'17.*
[29] Adam Oest, Yeganeh Safei, Adam Doupé, Gail-Joon Ahn, Brad Wardman, and Gary Warner. 2018. Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In *Proc. of eCrime'18.*
[30] Jeremiah Onaolapo, Enrico Mariconti, and Gianluca Stringhini. 2016. What happens after you are pwnd: Understanding the use of leaked webmail credentials in the wild. In *Proc. of IMC'16.*
[31] Sai Prashanth Chandramouli, Pierre-Marie Bajan, Christopher Kruegel, Giovanni Vigna, Ziming Zhao, Adam Doup, and Gail-Joon Ahn. 2018. Measuring E-mail header injections on the world wide web. In *Proc. of SAC'18.*
[32] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. In *Text Mining: Applications and Theory.* Wiley Online Library.
[33] Routhu Srinivasa Rao and Alwyn R Pais. 2017. Detecting phishing websites using automation of human behavior. In *Proc. of the ACM Workshop on Cyber-Physical System Security'17.*
[34] Elizabeth Stobert and Robert Biddle. 2014. The password life cycle: user behaviour in managing passwords. In *Proc. of SOUPS'14.*
[35] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, et al. 2017. Data breaches, phishing, or malware?: Understanding the risks of stolen credentials. In *Proc. of CCS'17.*
[36] Ke Tian, Steve TK Jan, Hang Hu, Danfeng Yao, and Gang Wang. 2018. Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild. In *Proc. of IMC'18.*
[37] Chun Wang, Steve TK Jan, Hang Hu, Douglas Bossart, and Gang Wang. 2018. The Next Domino to Fall: Empirical Analysis of User Passwords across Online Services. In *Proc. of CODASPY'18.*
[38] Liu Wenyin, Guanglin Huang, Liu Xiaoyue, Zhang Min, and Xiaotie Deng. 2005. Detection of phishing webpages based on visual similarity. In *Proc. of WWW'05.*
[39] Min Wu, Robert C Miller, and Simson L Garfinkel. 2006. Do security toolbars actually prevent phishing attacks?. In *Proc. of CHI'06.*
[40] Guang Xiang, Jason Hong, Carolyn P Rose, and Lorrie Cranor. 2011. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *Proc. of TISSEC'11* (2011).
[41] Shams Zawoad, Amit Kumar Dutta, Alan Sprague, Ragib Hasan, Jason Britt, and Gary Warner. 2013. Phish-net: investigating phish clusters using drop email addresses. In *Proc. of eCRS'13.*
[42] Yue Zhang, Jason I Hong, and Lorrie F Cranor. 2007. Cantina: a content-based approach to detecting phishing web sites. In *Proc. of WWW'07.*