

# Data Breaches, Phishing, or Malware? Understanding the Risks of Stolen Credentials

Kurt Thomas<sup>◊</sup> Frank Li<sup>†</sup> Ali Zand<sup>◊</sup> Jacob Barrett<sup>◊</sup> Juri Ranieri<sup>◊</sup> Luca Invernizzi<sup>◊</sup>  
Yarik Markov<sup>◊</sup> Oxana Comanescu<sup>◊</sup> Vijay Eranti<sup>◊</sup> Angelika Moscicki<sup>◊</sup> Daniel Margolis<sup>◊</sup>  
Vern Paxson<sup>†\*</sup> Elie Bursztein<sup>◊</sup>

Google<sup>◊</sup> University of California, Berkeley<sup>†</sup> International Computer Science Institute<sup>\*</sup>

## ABSTRACT

In this paper, we present the first longitudinal measurement study of the underground ecosystem fueling credential theft and assess the risk it poses to millions of users. Over the course of March, 2016–March, 2017, we identify 788,000 potential victims of off-the-shelf keyloggers; 12.4 million potential victims of phishing kits; and 1.9 billion usernames and passwords exposed via data breaches and traded on blackmarket forums. Using this dataset, we explore to what degree the stolen passwords—which originate from thousands of online services—enable an attacker to obtain a victim’s valid email credentials—and thus complete control of their online identity due to transitive trust. Drawing upon Google as a case study, we find 7–25% of exposed passwords match a victim’s Google account. For these accounts, we show how hardening authentication mechanisms to include additional risk signals such as a user’s historical geolocations and device profiles helps to mitigate the risk of hijacking. Beyond these risk metrics, we delve into the global reach of the miscreants involved in credential theft and the blackhat tools they rely on. We observe a remarkable lack of external pressure on bad actors, with phishing kit playbooks and keylogger capabilities remaining largely unchanged since the mid-2000s.

## 1 INTRODUCTION

As the digital footprint of Internet users expands to encompass social networks, financial records, and data stored in the cloud, often a single account underpins the security of this entire identity—an email address. This root of trust is jeopardized by the exposure of a victim’s email password or recovery questions. Once subverted, a hijacker can reset a victim’s passwords to other services as a stepping stone attack; download all of the victim’s private data; remotely wipe the victim’s data and backups; or impersonate the victim to spew out spam or worse.

Highly visible hijacking incidents include attacks on journalists such as Mat Honan and the Associated Press [21, 26], as well as

politicians and government officials including Sarah Palin, John Podesta, and Emmanuel Macron [17, 32, 40]. However, the threat of hijacking extends to millions of users [14, 36]. Indeed, a user study by Shay et al. in 2014 found 30% of 294 participants reported having at least one of their accounts compromised [33]. Yet, despite the prevalence of hijacking, there are few details about the largest sources of stolen credentials, or the degree to which hardening authentication mechanisms to include additional risk signals like a user’s historical geolocation or device profiles helps to mitigate the threat of compromise.

In this paper, we present the first longitudinal measurement study of the underground ecosystem fueling credential theft and the risks it poses to users. Our study captures three market segments: (1) forums that trade credential leaks exposed via data breaches; (2) phishing kits that deceive users into submitting their credentials to fake login pages; and (3) off-the-shelf keyloggers that harvest passwords from infected machines. We measure the volume of victims affected by each source of credential theft, identify the most popular blackhat tools responsible, and ultimately evaluate the likelihood that attacks obtain valid email credentials and subsequently bypass risk-based authentication protections to hijack a victim’s account.

To conduct our study, we develop an automated framework that monitors blackmarket actors and stolen credentials. Over the course of March, 2016–March, 2017, we identify 788,000 potential victims of keylogging; 12.4 million potential victims of phishing; and 1.9 billion usernames and passwords exposed by data breaches. We emphasize our dataset is strictly a sample of underground activity, yet even our sample demonstrates the massive scale of credential theft occurring in the wild. We observe victims from around the globe, with credential leaks and phishing largely affecting victims in the United States and Europe, while keyloggers disproportionately affect victims in Turkey, the Philippines, Malaysia, Thailand, and Iran.

We find that the risk of a full email takeover depends significantly on how attackers first acquire a victim’s (re-used) credentials. Using Google as a case study, we observe only 7% of victims in third-party data breaches have their current Google password exposed, compared to 12% of keylogger victims and 25% of phishing victims. Hijackers also have varying success at emulating the historical login behavior and device profile of targeted accounts. We find victims of phishing are 400x more likely to be successfully hijacked compared

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CCS’17, Oct. 30–Nov. 3, 2017, Dallas, TX, USA.

© 2017 Copyright held by the owner/author(s). ISBN 978-1-4503-4946-8/17/10.

DOI: 10.1145/3133956.3134067

to a random Google user. In comparison, this rate falls to 10x for data breach victims and roughly 40x for keylogger victims. This discrepancy results from phishing kits actively stealing risk profile information to impersonate a victim, with 83% of phishing kits collecting geolocations, 18% phone numbers, and 16% User-Agent data.

Behind the scenes, we find 4,069 distinct phishing kits and 52 keyloggers were responsible for the active attacks in our year-long monitoring sample. The most popular phishing kit—a website emulating Gmail, Yahoo, and Hotmail logins—was used by 2,599 blackhat actors to steal 1.4 million credentials. Likewise, the most popular keylogger—HawkEye—was used by 470 blackhat actors to generate 409,000 reports of user activity on infected devices. We find the operators of both phishing kits and keyloggers concentrate in Nigeria, followed by other nations in Africa and South-East Asia. Our findings illustrate the global reach of the underground economy surrounding credential theft and the necessity of a defense-in-depth approach to authenticating users.

## 2 BACKGROUND & RELATED WORK

Miscreants involved in credential theft rely on an array of underground markets and blackhat tools. We provide a brief background on some of these services, the details of which inform our study.

### 2.1 Credential Leaks

Media headlines reporting data breaches at major online services have become a regular occurrence in recent years. Indeed, 26% of 2,618 adults surveyed in the United States reported receiving a notice related to a data breach in the last year [1]. Prominent examples of affected online services include Yahoo, MySpace, LinkedIn, Adobe, and Dropbox, which combined revealed the username and password details for over a billion users [12, 15, 16, 18, 25]. The password storage policies of each of these companies varied, with some breaches exposing plaintext or unsalted password hashes, while others exposed salted SHA-1, bcrypt, or even symmetrically encrypted passwords. Though many of these credential leaks purportedly date back to 2012–2014, they have only recently percolated through underground networks and ultimately appeared on more public blackhat forums or paste sites [6], or on sites like *leakedsources.com*, *leakbase.pw*, or *breachalarm.com* that charge companies and users to lookup whether their accounts were impacted.

Apart from the loss of user faith in online services after massive password resets, credential leaks pose a broader risk to the online ecosystem due to weak password selection and re-use [34]. Das et al. examined the password strategies for users who appeared in multiple credential leaks and estimated 43% of passwords were re-used [9], while Wash et al. found users re-used 31% of their passwords based on a study of 113 participants [37]. Even if passwords are hashed or include subtle transformations from service to service, a wealth of prior work has examined how to invert the hashes via dictionary attacks or modeling password selection behavior [2, 7, 9, 10, 24, 29, 38]. We re-evaluate the risk of stolen passwords due to long-term re-use and the susceptibility of hashed passwords to trivial dictionary attacks.

### 2.2 Phishing Kits

Phishing kits are “ready-to-deploy” packages for creating and configuring phishing content that also provide built-in support for reporting stolen credentials [8]. The type of information stolen depends on the kits, but prior studies have shown that they harvest a victim’s username, password, and geolocation information among other sensitive data [8, 19, 30, 39]. Han et al. estimated the success rate of kits by monitoring the activity of real visitors to infected honeypots, of which 9% submitted some data to the phishing page [19].

Kits forward stolen credentials to the operator in one of three ways: through SMTP to an email address controlled by the operator, via FTP, or by connecting to a remote database. The number of phishing websites that rely on kits (as opposed to custom deployments) is unknown, but previous work by Zawoad et al. found 10% of phishing sites active in 2013 left trace evidence of phishing kits [39]. This is a lower bound due to a limited coverage in the detection technique for phishing kits and because miscreants may delete traces of the kit after deployment. Moore et al. demonstrated how to develop inbound email rules deployed at a large, undisclosed email provider to discover the email accounts of SMTP-based phishing kit operators, for which they detected 120–160 different miscreants [30]. We use this initial system as motivation for our design to capture the behavior of over ten thousand phishing kits, discussed in Section 3.

### 2.3 Keyloggers

Keyloggers have evolved beyond their moniker, with off-the-shelf families like HawkEye and Predator Pain [28] providing built-in functionality to steal on-device password stores, harvest clipboard content, and screenshot a victim’s activity in addition to monitoring keystrokes. As with phishing kits, keyloggers use a variety of techniques for reporting stolen credentials including SMTP, FTP, or remote databases. Holz et al. studied public ‘dropzones’ where keyloggers would upload stolen data and identified more than 10,700 stolen online bank account credentials and over 149,000 stolen email passwords over a 7 month period in 2008. [20]. While we focus on off-the-shelf keyloggers in this study, malware families may broadly include similar capabilities: Stone-Gross et al. examined the Torpig botnet and found it harvested over 54,000 email accounts from password stores and over 400,000 other credentials from HTTP forms over a 10 day period [35].

### 2.4 Hijacker Behavior

While not the focus of our research, a number of studies have investigated how hijackers subsequently abuse stolen credentials. Onalapo et al. leaked 100 email accounts via paste sites, underground forums, and virtual machines infected with malware [31]. They found a majority of miscreants searched the email history of accounts for financial data, while a smaller set used the accounts for spamming. Bursztein et al. reported a similar strategy where hijackers searched each victim’s email history for financial records and credentials related to third-party services [5]. Shay et al. conducted a user study of the harm caused by hijacking and found most participants self-reported being angry or embarrassed, but that their accounts were mostly used for spam [33]. In the realm

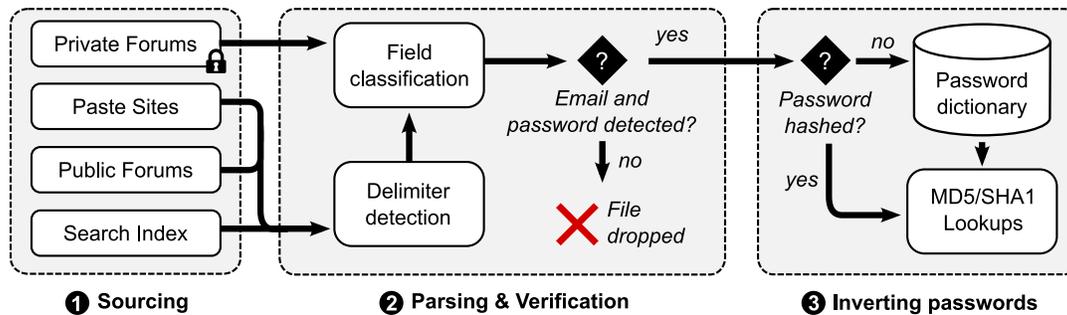


Figure 1: Collection framework for identifying credential leaks on public websites and private forums.

of social networks, Gao et al. identified 57,000 Facebook accounts that created 200,000 spam posts; they estimated 97% of the accounts were in fact compromised [14]. Finally, Thomas et al. examined cascades of hijacking campaigns on Twitter [36]. They identified 13.8 million compromised accounts used for both infecting other users and for posting spam. These behaviors illustrate a variety of strategies for monetizing stolen credentials—spam, financial fraud, and stepping stone access to other accounts.

### 3 METHODOLOGY

Our study of hijacking risk necessitates access to a significant corpus of stolen credentials. As such, we develop an automated collection framework that combines proprietary data from Google Search and Gmail to identify over a billion victims of credential leaks, phishing kits, and off-the-shelf keyloggers. Table 1 contains a detailed breakdown of the dataset collected by our system. We discuss the design decisions of our framework, its limitations, and ethical considerations that guide our approach.

#### 3.1 Credential Leaks

We present our high-level strategy for identifying usernames and passwords exposed via data breaches in Figure 1. Our design hinges on the idea that credential leaks sold privately on underground markets eventually surface for free. We detect when this happens by regularly crawling a set of paste sites and blackhat forums, as well as the public Internet at-large in order to identify content that may contain emails and passwords (1). We then parse and classify these documents to confirm whether they contain leaked credentials (2). Finally, when possible, we invert any non-salted, hashed passwords (3). We supplement this framework with credential leaks that we manually obtain from private, member-only forums. We discuss each of these steps in detail.

**Sourcing potential credential leaks.** As previously demonstrated by Butler et al. [6], blackhat forums and paste sites are common haunts for publicly sharing credential leaks. We leverage Google’s crawler to monitor activity on five public blackhat subforums<sup>1</sup> that deal exclusively with stolen credentials, along with 115 paste sites. As not every paste or forum thread will contain usernames and passwords, we treat each page as a *candidate document* that requires further verification. To avoid applying an expensive

<sup>1</sup>For operational reasons, we do not reveal the forums that we monitor.

Table 1: Summary of datasets from our collection pipelines.

Dataset	Samples	Time Frame
Credential leaks	3,785	06/2016–03/2017
Phishing kits	10,037	03/2016–03/2017
Keyloggers	15,579	03/2016–03/2017
Credential leak victims	1,922,609,265	06/2016–03/2017
Phishing kit victims	3,779,664	03/2016–03/2017
Keylogger victims	2,992	03/2016–03/2017
Phishing victim reports	12,449,036	03/2016–03/2017
Keylogger victim reports	788,606	03/2016–03/2017

verification process to every new paste, we apply a pre-filter to omit any pastes without at least 100 email addresses. Forum threads are far less frequent so we skip pre-filtering.

Given that our coverage of public paste sites and blackhat forums is undoubtedly incomplete, we improve our recall by identifying any document in Google’s recent crawl history that contains at least 10 of 1,000 common passwords (e.g., 123456, password) or their MD5 or SHA-1 equivalent, along with email suffixes for popular mail providers. We bootstrap our list of the most common passwords from previously collected credential leaks. We note this technique will miss credential leaks that are compressed, password protected, or encrypted. Together with paste sites and forums, we surface a combined total of 31,446 candidate documents, as detailed in Table 2, based on a snapshot of the search index in June, 2016.

We supplement this automated pipeline by manually collecting credential leaks shared on 11 private, member-only blackhat forums where we have access. Over the course of June, 2016–May, 2017<sup>2</sup>, we periodically monitored new forum threads and obtained 258 large credential leaks containing a combined 1.79 billion non-unique usernames and passwords. We emphasize that we never trade or purchase credential dumps—our activity on these sites is strictly passive. All of the files we obtained were compressed and shared via torrents or on file hosting sites, thus being missed by our automated detection.

<sup>2</sup>We note that the time we obtain a copy of the leaks is independent from the time the data breach actually occurred, which might have been many years prior.

**Parsing using delimiter detection.** For our automatically detected candidate documents, we first apply a *delimiter detection* heuristic to columnize the data. Based on a manual investigation of a random sample of candidate documents, we find that confirmed credential leaks conform to a small number of highly-structured formats: delimiter-separated values, key-value pairs, JSON blobs, and SQL query outputs. Intuitively, this stems from the fact most leaks are programmatically produced and consumed. In order to detect the correct file format, we apply multiple parsers configured with a small list of delimiter characters and then evaluate which parser generates rows of equal length that consist of at least two columns.

**Classification and verification.** After identifying and applying the optimal parser for a document, we scan over the produced columns to determine if they correspond to a leak. All leaks must have a column containing email addresses and a column containing a password (or password hash). We use a regular expression to identify which column contains an email address for every row. While not required, we also detect IP addresses using regular expressions; and User-Agent strings and mailing addresses based on a keyword dictionary (e.g., Mozilla, France). Verifying whether a candidate document contains a password is more challenging. To handle the possibility of MD5 and SHA-1 hashed passwords, we detect columns of fixed-length strings consisting entirely of hexadecimal characters. Plaintext passwords lack this typical structure. Here, we use a logistic regression that models character n-grams to identify likely password columns. We emphasize this learning approach takes into account every string in a column simultaneously rather than operating on a row by row basis.

To train our password classifier, we manually parsed and labeled the columns of plaintext credential leaks in our private forum dataset. We used all password columns as positive samples, and all other columns (e.g., usernames, and any other data) as negative examples. We then featurized every column into a binary vector of character n-grams. To determine the size of the n-grams and a threshold on the quantity of n-grams to include, we ran a grid search using 10-fold cross validation on the training data. Our search considered n-grams of length of 1 to 10 and feature vectors that included the top 1,000 to 100,000 most frequent n-grams, excluding common n-grams shared by both classes. Our final classifier consists of n-grams of length 2 to 5, and a feature vector of the top 10,000 such n-grams per class. To test our classifier, we manually labeled 230 candidate documents: 157 contained stolen credentials and 73 did not. Our classifier correctly classified 93.9% of test documents. The classifier favors precision over recall, whereby it failed to identify an existing password column in 8.3% of leaks and misidentified a password column in 3.1% of documents.

Training and testing aside, we apply our classifier to every candidate document and drop any document that fails to contain an email and potential hashed password, or password detected by the classifier. We present a breakdown of confirmed credential leaks per collection source in Table 2. In total, our automated collection identifies 3,527 documents from public sources which combined contain 123,055,697 emails and passwords. In comparison, we managed to acquire 1.7 billion passwords from just 258 leaks on private forums, indicating there still is a gap between public and private sets. Most

**Table 2: Breakdown of where we source credential leaks.**

Source	Candidate documents	Confirmed leaks	Credentials extracted
Paste sites	3,317	1,666	4,855,780
Search index	26,208	1,304	10,856,227
Public forums	1,921	557	107,343,690
Private forums	–	258	1,799,553,568

**Table 3: Top 20 largest credential leaks in our dataset and the fraction of inverted (or existing plaintext) passwords.**

Rank	Source	Number of credentials	Plaintext after inversion
1	Unknown <sup>P</sup>	558,862,722	100.0%
2	MySpace <sup>P</sup>	322,014,681	100.0%
3	Badoo	125,322,081	33.0%
4	Adobe <sup>◊</sup>	123,947,902	0.0%
5	LinkedIn	112,322,695	85.6%
6	VK <sup>P</sup>	76,865,954	99.6%
7	Tumblr <sup>*</sup>	73,355,694	0.0%
8	Dropbox <sup>†</sup>	68,669,208	0.0%
9	Zoosk	57,085,529	68.2%
10	IMesh <sup>‡</sup>	51,283,424	0.0%
11	LastFM	41,631,844	85.4%
12	Fling <sup>P</sup>	40,724,332	100.0%
13	Neopets <sup>P</sup>	35,822,980	100.0%
14	Mate1 <sup>P</sup>	27,383,966	100.0%
15	Unknown <sup>P</sup>	26,351,372	99.8%
16	000webhost <sup>P</sup>	15,249,241	100.0%
17	Taobao <sup>P</sup>	15,051,549	100.0%
18	NexusMods <sup>P</sup>	6,759,631	100.0%
19	Unknown <sup>P</sup>	5,728,163	99.7%
20	Unknown <sup>P</sup>	4,901,088	100.0%
–	Total	1,922,609,265	76.0%

<sup>P</sup> Password leak acquired in plaintext format; no dictionary attack required.

<sup>◊</sup> Adobe passwords were encrypted and could not be reversed.

<sup>\*</sup> Tumblr passwords were salted SHA-1. Salt was not present in the leak acquired.

<sup>†</sup> Dropbox passwords were a mixture of bcrypt and salted SHA-1. Salt was not present in the leak acquired.

<sup>‡</sup> IMesh passwords were salted MD5. Salt was not present in the leak acquired.

public leaks are small: 48% contain fewer than 1,000 credentials, and 86% fewer than 10,000. We list the the largest 20 confirmed leaks in our dataset in Table 3.

**Inverting passwords.** Based on the character length and distribution of passwords in each confirmed leak, we estimate 14.8% of all passwords in our dataset are hashed using MD5 and 9.8% are SHA-1. We attempt to invert these passwords using a dictionary

**Table 4: Top 10 passwords across all plaintext leaks.**

Rank	Top Passwords	Number of Credentials	Percent of Credentials
1	123456	6,387,184	0.35%
2	password	2,759,747	0.15%
3	123456789	2,249,344	0.12%
4	abc123	985,709	0.10%
5	password1	888,836	0.05%
6*	homelesspa	855,477	0.05%
7	111111	855,257	0.05%
8	qwerty	829,835	0.05%
9	12345678	828,848	0.05%
10	1234567	740,464	0.04%

\* This was the most common password in the MySpace credential leak, but appears to be automatically generated as all email addresses begin with “msmhomelessartist”.

of 3,416,701,663 keywords. We source this list by combining non-hashed passwords identified in the previous stages of our pipeline with supplemental dictionaries curated by Weakpass<sup>3</sup> and Crackstation.<sup>4</sup> In total, we successfully invert 35.8% of hashed passwords. We note this low hit rate may result for two reasons. First, black-market actors may have previously inverted credential leaks and uploaded a new leak file with both discovered plaintext passwords and any remaining uninverted hashes. In this case, we are only iterating on the previous inversion step. Secondly, this approach fails when applied to salted passwords (as was the case for Dropbox, Tumblr, and IMesh). We summarize our final dataset, post-inversion in Table 3, with the most popular passwords listed in Table 4.

### 3.2 Phishing Kits & Victims

Through an undisclosed source, we obtain a sample of 10,037 phishing kits (including the PHP and HTML source code) and 3,779,664 usernames and passwords belonging to victims of those kits along with the time they were phished. Both the kits and victims were identified over the course of March, 2016–March, 2017. Leveraging this data, we develop a pipeline to understand the life cycle of phishing kits and the volume of potential victims they deceive as shown in Figure 2. Our pipeline hinges on the observation that phishing kits frequently use email as a mechanism for reporting stolen credentials (discussed previously in Section 2). Using our phishing kit corpus (❶), we first statically analyze the source code of each kit to extract its email template used to report stolen credentials (❷). We then develop rules to match the subject and body of these messages (❸) and finally adapt Gmail’s anti-abuse classification pipeline to identify all inbound messages containing stolen credentials (❹).

**Template extraction.** All of the phishing kits in our corpus use PHP’s `mail()` command to report stolen credentials to an *exfiltration point*. We provide an example in Figure 3. This particular kit prompts a victim for their username and password before populating a message template containing the stolen information, which

<sup>3</sup><https://weakpass.com/>

<sup>4</sup><https://crackstation.net/>

**Table 5: Breakdown of the top five email providers used by miscreants as exfiltration points to receive stolen credentials.**

Phishing Kits		Keyloggers	
Mail provider	Popularity	Mail provider	Popularity
Gmail	72.3%	Gmail	39.0%
Yahoo	6.8%	Yandex	12.3%
Yandex	5.1%	Mail.ru	8.5%
Hotmail	4.2%	Hotmail	3.6%
Outlook	2.2%	Zoho	1.3%
Other	9.4%	Other	35.3%

is then mailed to a hardcoded email. Using static analysis, we automatically identify calls to `mail()` and then determine the string values supplied to each variable of the call. This search handles all *include* operations as well as nested variable instantiations. In the end, we output a template that includes the target email (e.g., exfiltration point), the message’s subject, and the message’s body stripped of variables the kit determines at run-time. We provide a breakdown of email providers for 7,780 unique exfiltration points hard coded into kits in Table 5, of which 72.3% relate to *gmail.com*. This heavy use indicates our pipeline should detect a significant amount of all messages related to stolen credentials.

**Rule generation.** The next phase of our pipeline tokenizes the message templates and outputs a set of rules to match the subject and body of inbound emails that contain stolen credentials along with a maximum expected message size. We opt for rules rather than identifying the exfiltration points in our kit corpus because a single kit may be re-configured to use multiple exfiltration points (potentially by multiple actors). For the sample kit in Figure 3, a message would match the template if its subject contained *Result from Gmail*, while its body contained *Username, Vict!m Info*, and all other strings from the template’s *\$message* that are not run-time variables. In total, we generate 7,325 rules which cover our 10,037 kits. As an initial trial to evaluate false positives, we applied these to a corpus of 100,000 messages unrelated to phishing kit templates, of which there were 0 false positives.

**Email flagging.** We modify Gmail’s anti-abuse detection systems to apply our rules to all inbound messages over the course of March, 2016–March, 2017 (the same period over which the kits were collected) to identify the exfiltration points receiving stolen credentials, the volume of messages each account receives, and the volume of messages per kit template. We require any exfiltration point to receive at least 20 stolen credentials before we include it in our study. In total, we flag 12,449,036 messages (excluding 0.8% filtered due to lacking at least 20 credentials) sent to 19,311 exfiltration points. We caution this is a strict underestimate of messages generated by phishing kits as our coverage of kits is non-exhaustive and kits may use non-Gmail addresses or even non-SMTP mechanisms to report stolen credentials (as discussed in Section 2).

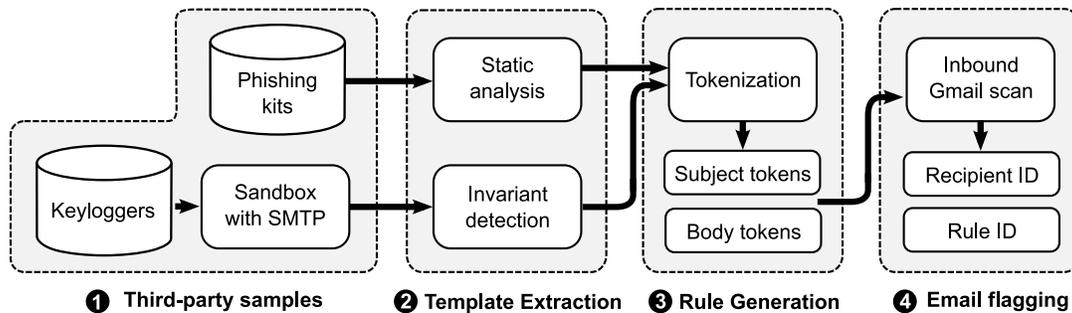


Figure 2: Framework for identifying inbound messages that contain credentials stolen by phishing kits and keyloggers.

```

$subject = "Result from Gmail";
$message .= "-----Gmail Info-----";
$message .= "Username : ".$gmailuser."\n";
$message .= "Password : ".$gmailpassword."\n";
$message .= "-----Vict!m Info-----";
$message .= "Client IP : ".$ip."\n";
$message .= "Browser : ".$browserAgent."\n";
$message .= "country : ".$country."\n";
$message .= "-----Created BY Dropbox Wire-----";

mail("xxx@gmail.com", $subject, $message);
  
```

Figure 3: Sample phishing kit that collects a victim’s user-name, password, browser user-agent, and IP address.

### 3.3 Keyloggers & Victims

Through an undisclosed source, we obtain a corpus of 15,579 keylogger binaries and information related to 2,992 victims including the timestamp they were infected and the passwords stolen. As many keyloggers are pre-configured to use email as an exfiltration point (discussed in Section 2), we extend our phishing detection logic (Figure 2) to include keyloggers. To start, we execute each binary in a Windows sandbox seeded with a Chrome and Firefox password store containing a set of honey credentials. During execution, we monitor outbound SMTP connections to identify message bodies that contain the credentials. We then extract string invariants that do not contain system information related to the sandbox environment, time of execution, or honey credentials and convert these invariants into rules. In total, we require only 315 rules to cover all of the keyloggers in our corpus, a marked lack of diversity compared to phishing kits that likely stems from the complexity of writing desktop software versus PHP. As before, we trial these rules on a test corpus of 100,000 emails which generates 0 false positives. Using the same deployment strategy as phishing kits, we flag a total of 788,606 messages (excluding 1.7% filtered by our minimum requirement of 20 messages per exfiltration point) sent to 1,034 exfiltration points. We expect our keylogger coverage to be less than phishing kits as only 39.0% of samples in our corpus use Gmail as an exfiltration point (Table 5).

### 3.4 Limitations

Our heuristic approach to credential leak identification and reliance on samples of phishing kits and keyloggers leads to incomplete coverage. As such, our perspective of the *volume* of the problem should be treated only as a lower bound. Nevertheless, our dataset of victims and blackhat tools provides a significant enough sample to evaluate the hijacking risk to users who have had their password exposed. Finally, we caution that our corpus of victims is not a random sample: victims of credential leaks are biased towards the user base of major US services (though many other non-US companies may have suffered breaches), while the victims of phishing kits and keyloggers are biased towards the unknown blackhat distribution strategies beyond our visibility.

### 3.5 Ethics

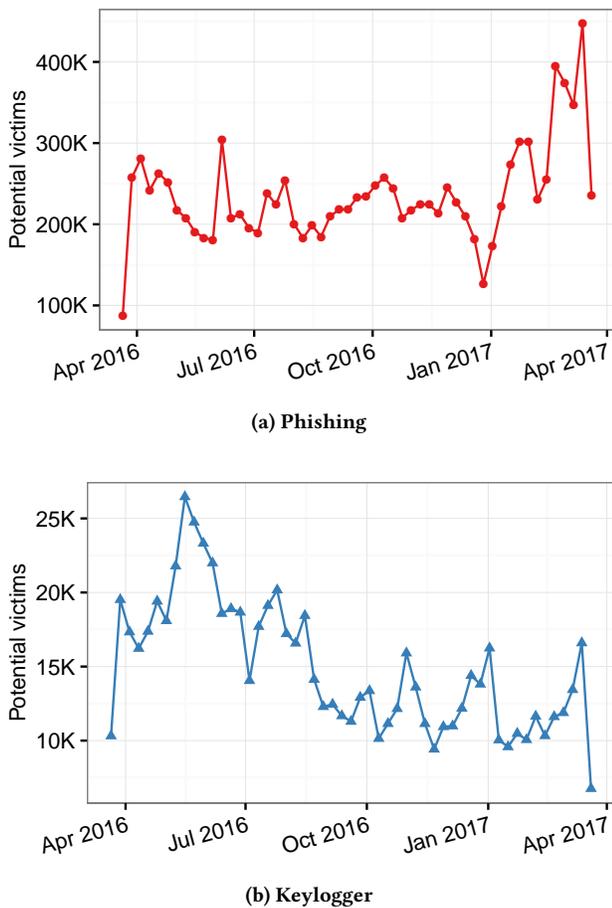
The ethics of using data exposed by data breaches or other illicit origins remains a hotly debated topic in the security community [11]. We never purchase or trade credential leaks. We exclusively use the password information in our dataset to evaluate hijacking risks. Furthermore, for all Google users in our dataset, we re-secure all accounts via a forced password reset in the event their real credentials were exposed. This remains one of Google’s long standing policies [4]. We do not attempt to check the validity of passwords for any other site. Finally, when measuring the volume of phishing and keylogger activity on Gmail, we reiterate that all detection occurs by our anti-abuse systems in a fully programmatic fashion.

## 4 VICTIMS WITH STOLEN CREDENTIALS

As a first step towards understanding the landscape of stolen credentials, we examine the frequency that users fall victim to credential theft, demographic information related to those victims, and other information beyond passwords that attackers also steal.

### 4.1 Volume of Victims

In order to offer a perspective of the sheer amount of credential information accessible to miscreants, we provide a breakdown across each of our collection sources. We identify 1,484,680,141 unique username, password combinations that belong to 1,092,567,042 unique usernames. For phishing kits and keyloggers, we provide a weekly breakdown in Figure 4 of the number of messages flagged by our rules as containing stolen credentials. On average, miscreants



**Figure 4: Weekly breakdown of the number of messages our rules flag as containing stolen credential information.**

using the phishing kits in our corpus collect 234,887 potentially valid credentials every week, and for keyloggers, 14,879 per week. There is a noticeable dip in phishing kit activity around the holiday season in 2016, potentially due to limited campaign activity by miscreants or limited online activity by victims. Our results illustrate that credential theft is a multi-pronged problem. Even absent relatively rare data breaches that expose hundreds of millions of credentials in a single incident, there are still hundreds of thousands of users that fall victim to phishing and keyloggers every week—and that covers only what we detect. As such, we argue there remains a significant gap between the threats that users are exposed to and protections in place, such as education efforts, warnings, or automated defenses.

## 4.2 Demographics

In terms of unique users, our dataset includes 1,092,567,042 credential leak victims, 3,779,664 phishing victims, and 2,992 keylogger victims. We provide a breakdown of the top 10 mail providers used by each victim (in the event their username is an email address) in

Table 6. This list reflects the mail providers who might be most impacted by stolen credentials, though this approach does not capture social media accounts or commercial and banking accounts that rely on service-specific usernames rather than an email address. We find that Gmail, Yahoo, and Hotmail make up 50% of victims regardless of the origin of the credentials. In total, however, there are over 25 million email domains captured by our dataset, reflecting the challenge of re-securing exposed password material across tens of millions of services.

As an approximation of the geographic distribution of victims, we examine the sign-up locations of Google users appearing in our dataset. Table 7 provides a breakdown of the top 10 geolocations. For credential leaks, we find the geographic distribution of victims reflects the aggregate user bases of major US online services at the time of their breach in 2012–2014, with the United States, India, and Brazil accounting for 49.3% of victims. In contrast, phishing attacks heavily favor the United States, followed in popularity by South Africa and Canada—all countries with large English-speaking populations. This may result from bias in our phishing kit corpus, or from the selection criteria of attackers. Keylogging victims are the most distinct, where Turkey, the Philippines, Malaysia, Thailand, Iran, and Nigeria appear in the top 10 geolocations, unlike for phishing victims or credential leak victims. While we lack visibility into the keylogger distribution campaigns, it is possible these infections stem from governments or authorities as previously examined by Marczak et al. [27]. Taken as a whole, our demographic breakdown suggests that miscreants tailor their distribution campaigns to specific regions or even targets.

## 4.3 Additional Information Exposed

While our study focuses on stolen credentials, additional sensitive user information may be exposed by credential leaks, phishing kits, and keyloggers. Based on the output of our leak classifier, only 3.8% of victims of credential leaks have their geolocation exposed in the form of a postal address or IP address, and approximately 0.000009% of leaks included User-Agent information. To evaluate the same metrics for phishing kits and keyloggers, we manually compile a list of keywords in multiple languages related to real names, credit cards, addresses, phones, device information, and secret questions, and then search the message templates of phishing kits and keyloggers for these keywords.

Our results, shown in Table 8, indicate that phishing kits frequently collect additional authentication factors such as secret questions, geolocation details, and device-related information, likely to bypass login challenges for services that attempt to detect suspicious sign-ins (discussed further in Section 5). This behavior is less common for keyloggers, where fewer than 0.1% of keylogger variants explicitly gather phone details or secret questions that might be used as login challenges (though it nevertheless may appear in keystroke logs). Among our phishing kit corpora, we also observe that 39.9% of variants collect credit card information and 8.8% of variants collect social security numbers (US-specific). Our findings indicate that while credential leaks may expose the largest number of passwords, phishing kits and keyloggers provide more flexibility to adapt to new account protections.

**Table 6: Distribution of emails providers used by victims of credential leaks, phishing kits, and keyloggers. For credential leaks, we note that none of the exposed credentials in our study originate from a breach at an email provider (to our knowledge). All email addresses were exposed due to a third-party breach where the company used email addresses as identifiers.**

Credential Leak Victims		Phishing Victims		Keylogger Victims	
Email provider	Popularity	Email provider	Popularity	Email provider	Popularity
yahoo.com	19.5%	gmail.com	27.8%	gmail.com	29.8%
hotmail.com	19.0%	yahoo.com	12.0%	yahoo.com	11.5%
gmail.com	12.2%	hotmail.com	11.3%	hotmail.com	9.4%
mail.ru	4.7%	outlook.com	1.0%	aol.com	3.3%
aol.com	3.6%	mail.ru	0.8%	hotmail.fr	1.6%
yandex.ru	1.4%	live.com	0.6%	msn.com	1.1%
hotmail.fr	1.3%	yahoo.co.in	0.5%	hotmail.co.uk	0.9%
hotmail.co.uk	1.0%	orange.fr	0.5%	comcast.net	0.8%
live.com	1.0%	yml.com	0.4%	sbcglobal.net	0.8%
rambler.ru	0.8%	hotmail.fr	0.4%	163.com	0.7%
Other	35.4%	Other	44.7%	Other	44.7%

**Table 7: Distribution of geolocations for victims of credential leaks, phishing kits, and keyloggers with Google accounts.**

Credential Leak Victims		Phishing Victims		Keylogger Victims	
Signup location	Popularity	Signup location	Popularity	Signup location	Popularity
United States	38.8%	United States	49.9%	Brazil	18.3%
India	7.9%	South Africa	3.6%	India	9.8%
Brazil	2.6%	Canada	3.3%	United States	8.0%
Spain	2.5%	India	2.8%	Turkey	5.8%
France	2.1%	United Kingdom	2.5%	Philippines	3.8%
Italy	1.9%	France	1.9%	Malaysia	3.3%
United Kingdom	1.8%	Spain	1.9%	Thailand	3.1%
Canada	1.7%	Australia	1.8%	Iran	3.1%
Japan	1.5%	Malaysia	1.1%	Nigeria	2.8%
Indonesia	1.4%	Italy	1.0%	Indonesia	2.7%
Other	37.8%	Other	30.2%	Other	39.5%

**Table 8: Additional information stolen by phishing kits and keyloggers. We note that some phishing kits exclusively collect credit card details rather than usernames and passwords.**

Data type	Phishing kits	Keyloggers
Email	81.4%	97.8%
Password	83.0%	100%
Geolocation	82.9%	73.6%
Phone number	18.1%	0.1%
Device information	16.2%	67.9%
Secret questions	7.4%	0.1%
Full name	45.8%	85.3%
Credit card	39.9%	2.1%
SSN	8.8%	0.1%

## 5 RISK OF STOLEN CREDENTIALS

With billions of potentially valid passwords available to miscreants, we evaluate the risk of an attacker accessing a victim’s email account—and through transitive trust, the victims entire online identity. We quantify this risk along multiple dimensions: current password match rates; historical password match rates; successful hijacking attacks; brute force login attempts; and challenges with recovery and re-compromise.

### 5.1 Current password match rate

Stolen credentials pose a significant risk to authentication systems built solely on usernames and passwords. To approximate this risk, we scan through all email addresses and passwords in our dataset in search of Google users. We extend this search to non-email usernames by looking up whether any Google account exists under the same username (e.g., for Cindy001, we would check Cindy001@gmail.com and other Google-related domains). For existing users, we then verify whether the password exposed to miscreants matches the user’s current Google password. In order to stem any future abuse, we force all victims with exposed, valid credentials to reset their passwords. We note that in the case of credential leaks, our experiment measures long-term password re-use, while for phishing kits and keyloggers there are multiple factors at play we discuss shortly.

**Credential leaks re-use rate.** We identify a total of 751,133,653 Google users affected by third-party breaches,<sup>5</sup> of which 51,754,113 had valid passwords—a match rate of 6.9%, if we include unsalted passwords that we failed to invert. If we exclude all victims with non-inverted passwords (thus biasing towards weak passwords which victims may re-use more frequently), the match rate increases to 7.5%. We note these rates are likely underestimates, as users may have changed their passwords between the time the password was exposed and our check; we may have previously reset the victim’s password from a prior credential leak; or because our approach to mapping usernames to Google accounts may over-count the number of existing accounts which are in fact unrelated to the victim of a credential leak. Across all of our credential leaks, the median match rate per file including non-inverted passwords is 7.0%. While this percentage is small, in aggregate, data breaches continue to expose millions of valid passwords.

**Phishing & Keylogger match rate.** For victims of phishing kits, we identify a total of 2,335,289 Google users, of which 578,434 had valid passwords—a match rate of 24.8%. For keyloggers, our sample is much smaller and consists of 1,616 Google users, only 192 of which had valid passwords—a match rate of 11.9%. Using the timestamps reported through our undisclosed source, we compare the match rate of passwords exposed in the first 6 months of our time window versus the last 6 months and find the rate falls within the margin of error (23.1% vs. 24.1% for phishing kits; 12.2% vs. 10.4% for keyloggers). This suggests that even a year later, victims remain unaware their passwords were ever exposed and at risk.

<sup>5</sup>We exclude victims appearing in salted or encrypted credential leaks so as not to underestimate long-term password re-use.

Table 9: Risk associated with passwords stolen through leaks, phishing kits, or keyloggers.

Password source	Password match rate	Hijacking odds ratio	Failed login odds ratio
Credential leak	6.9%	11.6x	1.4x
Phishing kit	24.8%	463.4x	1.7x
Keylogger	11.9%	38.5x	1.5x

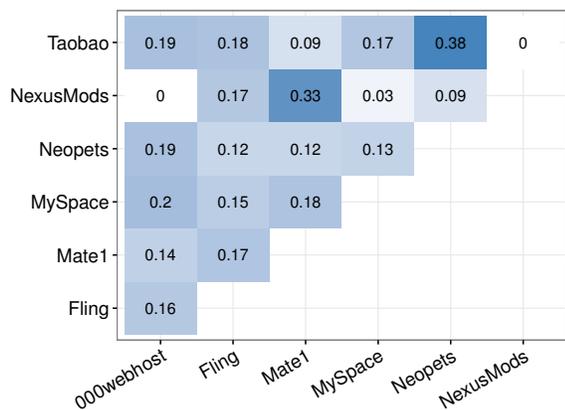
Having ruled out staleness, one explanation for the low match rates is that numerous services rely on email addresses for usernames, though victims may diversify passwords between their email provider and these other services (a promising sign). As our password dataset lacks an annotation of the service targeted, we cannot limit our analysis to Google-specific attacks. Likewise, victims may supply intentionally incorrect information to phishing pages to “test” whether the login is real. We find 5% of *gmail.com* emails provided to phishing pages do not exist, indicating either a savvy user or security service submitting fake credentials.

### 5.2 Historical password match rate

As stolen credentials become stale over time, our analysis of credential leaks dating back to 2012–2014 may underestimate the risk of password re-use. To better understand contemporaneous password re-use metrics, we examine the overlap of passwords for users affected by multiple data breaches. We restrict our analysis to fully inverted leaks (with 100% plaintext passwords after inversion) where we have a high confidence in the origin due to manual acquisition—otherwise, copies or portions of the same leak appearing on multiple paste sites may cause us to overestimate re-use. Across the 7 largest fully inverted leaks of known distinct origins (listed in Table 3), we observe that 17.0% of the 22 million email addresses in multiple leaks re-used a password at least once. Figure 5 depicts the pairwise password re-use rate for each set of leaks. While the Taobao and Neopets leaks had the most extensive password re-use (for 38% of common emails), the majority of password re-use rates varied between 12% and 19%. Interestingly, for email addresses in at least three leaks, only 7.1% re-used a password two times or more, indicating that while users may re-use passwords across multiple sites, universal use of the same password is less common. For comparison, Das et al. analyzed 10 plaintext (or fully inverted) leaks from 2006–2012 and found a 43% re-use rate for 6,077 accounts [9]. Our sample from 2012–2014 is much larger and indicates that password re-use is less frequent.

### 5.3 Hijacking risk

We evaluate the likelihood a user falls victim to hijacking given they appear in our dataset of stolen credentials. In order to mitigate the risk of exposed passwords, Google blocks or requires additional authentication information when a login falls outside a user’s risk profile, similar to an approach by Freeman et al. [13]. This profile encapsulates a user’s historical access patterns, known devices, and known locations. The full details of this calculation are beyond the scope of this work. Independently, Google also monitors account activity to detect suspicious behaviors for authenticated users in



**Figure 5: Heatmap of password re-use rates, comparing leaks pairwise.**

order to detect victims who have been hijacked. This detection is unbiased as to *how* the victim was first compromised.

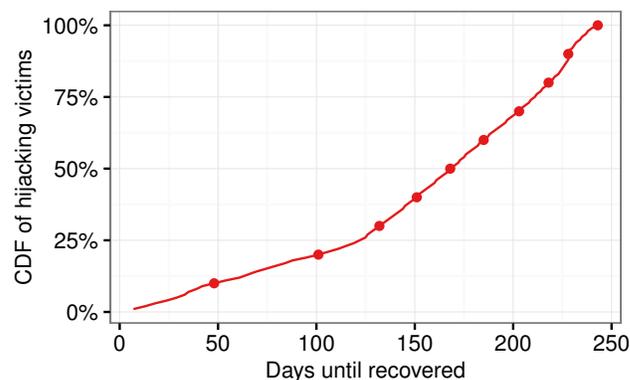
To understand the impact of risk-based login scores, we calculate the odds ratio that a Google account in our dataset with a valid password was hijacked between March, 2016–March, 2017 compared to a random sample of all Google accounts. We rely on odds ratios rather than raw likelihoods as hijacking detection has an unquantified number of false negatives (though near-zero false positives). We present a breakdown of our results in Table 9. We find that once a user’s valid credentials are exposed to a phishing kit, the likelihood they become compromised is over 400x more than a random user. In contrast, for victims affected by data breaches, the odds of becoming compromised are an order of magnitude less: roughly 10x. This discrepancy stems in part from phishing kits collecting additional information related to victims including their login location, User-Agent, and even recovery questions (discussed in Section 4.3), whereas credential leaks often only include the username and password. Keyloggers fall in between these extremes, with an odds ratio of roughly 40x. As such, while credential leaks represent the largest source of passwords in our dataset (even taking into account match rates), phishing victims are the most likely to become hijacked.

#### 5.4 Brute force password guessing

For accounts without valid passwords, we also examine whether attackers attempt to brute force access to the account, potentially trying variations of the exposed password as an initial seed. Using a sample of all logins from a one week period, we calculate the odds that a victim appearing in our dataset receives at least 10 failed logins compared to a random sample of users. Table 9 shows our results. While we find some evidence of inflated failed sign-ins—1.4x to 1.7x more than random users—we do not find any strong evidence that attackers are permuting invalid passwords. That said, our analysis window is fairly limited due to privacy reasons; it may be attackers behave differently in other time periods.

#### 5.5 Recovery rate

In the event Google detects an account as hijacked, it proactively disables login access and invalidates any existing sessions. While



**Figure 6: Duration an account remains temporarily disabled due to hijacking before the true owner recovers access.**

this lock down prevents further abuse, it raises a subsequent challenge where users must prove ownership beyond a password in order to re-gain access to their account. Google relies on a spectrum of challenges to prove historical access: sending a code to a pre-configured second email address or phone number; answering a secret question; or identifying prior product usage times. Roughly 70.5% of hijacked users successfully pass these challenges to recover their account. We provide a breakdown of the time frame between when an account was disabled and re-enabled in Figure 6. A median user takes 168 days to re-secure their account. This long delay arrives in part from users being unaware they are hijacked, and Google lacking an alternate notification mechanism in the absence of a recovery phone or recovery email. Furthermore, users may be confused why they cannot login.

For those users that do successfully recover from a hijacking incident, we examine what fraction change their security posture post-recovery. We find only limited evidence of improving account security: roughly 3.1% of users enable second-factor authentication. Our results suggest there is a significant gap in educating users about how to protect their accounts from further risk. This mirrors previous findings by Ablon et al. where only 4% of users migrated to password managers after being notified their data was exposed by a breach [1], as well as results by Ion et al. who found that while experts commonly favor using two-factor authentication or password managers, these tools are virtually absent from the security posture of regular users [23].

#### 5.6 Recompromise rate

As a final metric, we examine the likelihood that victims hijacked in the last year become hijacked again in the same time window. We restrict this analysis to only accounts that were successfully recovered. In total, only 2% of users fall victim to repeat hijacking. This indicates that password resetting may be a sufficient response to address account compromise. For repeat victims, one possibility is that malware infections harvest newly changed passwords, or that they were deceived by a phishing attack after their initial recovery.

## 6 INSIGHTS INTO BLACKHAT TOOLS

Zooming out, we leverage our unique dataset to explore the most influential phishing kits and keyloggers fueling the ecosystem of credential theft. We examine which blackhat tools are most popular in the wild and how they changed during the course of our study. We also explore the miscreants deploying these tools and the regions where they are most active.

### 6.1 Popularity of tools

With 4,069 phishing kit variants and 52 keylogger variants in operation during our measurement window, which ones see the most widespread use? As a metric of popularity, we examine the number of unique Gmail exfiltration emails per variant (e.g., the exfiltration email received at least one message matching that particular phishing kit or keylogger’s reporting template). We note it is possible that multiple kit variants all report to the same exfiltration point. Our results in Figure 7 indicate that the majority of blackhat tools are unpopular: 69% of phishing kit variants have fewer than 10 associated exfiltration points, while the same is true for 48% of keyloggers.

However, if we rank the tools by the number of potential victims they impact, we find that a handful of popular tools have a significant negative impact as shown in Table 10 and Table 11. The most popular phishing kit—a file portal that supports “logins” from Yahoo, Hotmail, AOL, Gmail, and other mail providers—generated 1,448,890 reports of stolen credentials to 2,599 different exfiltration emails. The other top phishing kits spoof a variety of brands including file storage services like Dropbox and Office 365; web mail providers like Workspace Webmail (operated by GoDaddy) and AOL; and even business services like Docusign (legal signing service) and ZoomInfo (business information service).

The most popular keylogger, HawkEye, sent 409,837 reports of victim activity to 470 exfiltration emails. Originally available on *hawkeyeproducts.com* for \$35, a second “Hawkeye Reborn” version was released via *hawkspy.net* along with multiple cracked versions. Hawkeye supports stealing credentials from browsers, mail clients, and chat clients in addition to multiple exfiltration options (SMTP, FTP, and HTTP). Other popular kits in use during our year-long investigation include Cyborg, Predator Pain, Limitless, and Olympic Vision—the majority of which are free (or cracked) keyloggers available on blackmarket forums. The large discrepancy in the number of exfiltration emails between keyloggers and phishing kits likely stems from the ease of deploying a website and attracting visitors versus deceiving users into installing a binary and contending with anti-virus protections.

### 6.2 Usage over time

As an alternative measure to popularity, we also examine the long-term usage of phishing kit and keylogger variants. Figure 8 provides a breakdown of the number of days we saw activity from each blackhat tool. Roughly 50% of keylogger variants remained active for the entire year duration of our study. This likely stems from a lack of diversity in keyloggers and a slow release cycle for new variants. In contrast, only 21% of phishing kits remained active over the course of the year. Re-examining the most popular tools in Table 10 and Table 11, all remained active for the entirety of the

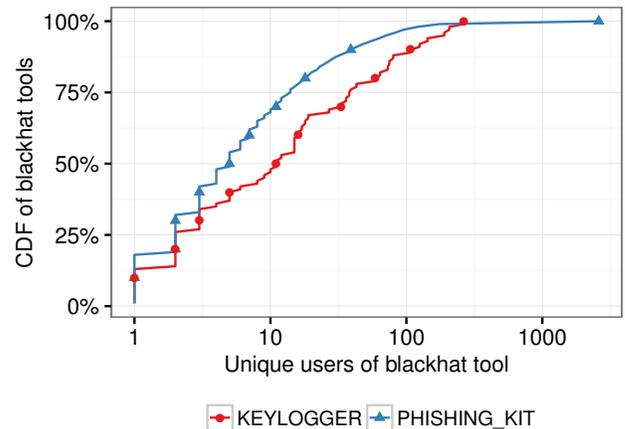


Figure 7: Number of email addresses configured to receive stolen credentials for each blackhat tool (log scale).

Table 10: Top 10 phishing kits and the brands they target, ranked by number of potential victims.

Brand impersonated	Potential victims	Exfiltration emails	Days active
Yahoo, Hotmail, Gmail	1,448,890	2,599	365
Workspace Webmail	1,292,778	814	365
Dropbox	323,689	976	365
Dropbox	195,758	862	365
Google Drive	185,966	382	365
Docusign	152,242	180	365
ZoomInfo	151,282	19	364
Docusign	142,761	175	365
Office 365	133,044	166	284
AOL	130,898	507	365

Table 11: Top 10 keylogger families, ranked by the number of potential victims.

Keylogger	Activity reports	Exfiltration emails	Days active
HawkEye	409,837	470	365
Cyborg Logger	173,662	60	365
Predator Pain	118,197	326	365
Limitless Stealer	24,371	44	365
iSpy Keylogger	16,495	162	365
Olympic Vision	9,056	19	363
Unknown Logger	8,561	17	352
Saint Andrew’s	6,802	1	352
Infinity Logger	4,690	15	363
Redpill Spy	3,668	15	363

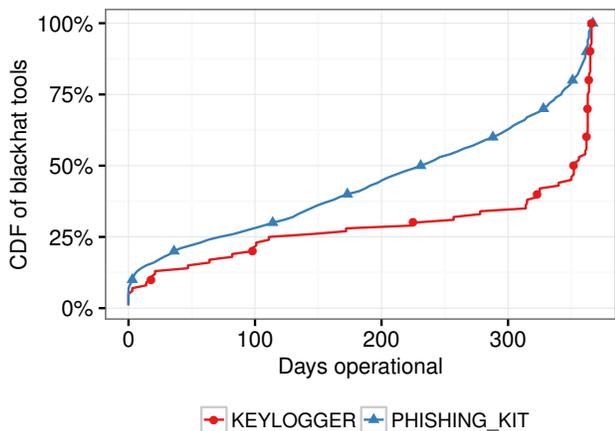


Figure 8: Phishing kit and keylogger daily usage, where usage indicates generating at least one message on a given day.

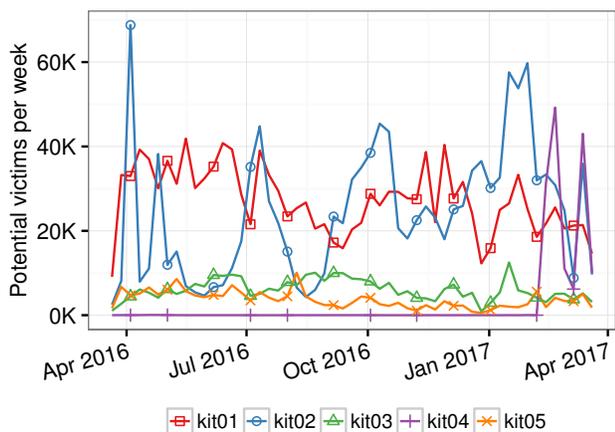


Figure 9: Weekly breakdown of potential stolen credentials collected by the top 5 phishing kits.

year with the exception of the phishing kit targeting Office 365. This stability indicates a remarkable lack of external pressure on blackhat developers to modify their phishing landing pages.

Zooming in on the weekly activity for the top five phishing kits, we find bursty, campaign-like behavior as shown in Figure 9. This suggests a handful of actors have an outsized influence in the space. For example, the second most popular kit (Workspace Webmail) affected only 2,500 potential victims each week during lulls, but then jumped to 40,000–69,000 victims per week during periods of coordinated action. Likewise, the fourth most popular kit (Dropbox) netted fewer than 100 victims each week until two campaigns collected over 40,000 credentials each week while active. Indeed, if we examine the exfiltration points receiving stolen credentials, the top 50 receive 26% of all credentials. As such, while the blackhat tools we investigate exhibit year-round activity from hundreds of actors, just a handful of actors truly drive the market.

Table 12: Top 10 geolocations associated with the last sign-in to email accounts receiving stolen credentials.

Phishing Kit Users		Keylogger Users	
Geolocation	Popularity	Geolocation	Popularity
Nigeria	41.5%	Nigeria	11.2%
United States	11.4%	Brazil	7.8%
Morocco	7.6%	Senegal	7.3%
South Africa	6.4%	United States	6.4%
United Kingdom	3.3%	Malaysia	5.8%
Malaysia	3.2%	India	5.7%
Indonesia	3.1%	Philippines	4.6%
Tunisia	2.0%	Turkey	3.2%
Egypt	1.6%	Thailand	2.8%
Algeria	1.3%	Egypt	2.7%
Other	18.6%	Other	42.7%

### 6.3 Location of users

We examine the last login geolocation for all Gmail accounts that received stolen credentials to provide a perspective on the geographic usage of blackhat tools. We note that in the event a single miscreant controls multiple exfiltration points, we may over count their location. Likewise, we cannot rule out the possibility of proxies obfuscating the true location of miscreants. We provide a breakdown of the top 10 geolocations for phishing kit users and keylogger users in Table 12.

We find that 41.5% of exfiltration points for phishing kits were last accessed in Nigeria, followed in popularity by the United States, Morocco, and South Africa. This differs significantly from the geolocations of hijackers accessing the accounts of victims as reported by Bursztein et al. [5] (predominantly China) and by Onaolapo et al. [31] (predominantly tailored to the victim’s historical geolocation to evade risk detection, or Tor). This suggests that the infrastructure that miscreants use to steal credentials is independent from the devices that hijackers use to access accounts. Compared to the targets of phishing (discussed previously in Table 7), victims largely come from North America and Europe, while blackhat users originate from Africa and South-East Asia.

For keyloggers, we find a much more diverse ecosystem, where a small fraction of users span every continent. Still, Nigeria takes the top place, accounting for 11.2% of exfiltration points. We also observe a coupling between the targets of keyloggers (discussed previously in Table 7) and the geolocations of miscreants. Brazil, India, the United States, Turkey, the Philippines, Malaysia, Thailand, and Nigeria appear in both top 10 lists. Not listed in Table 7, Iran makes up 2.7% of blackhat users, and Indonesia 1.4%. This coupling suggests that blackhat users favor local victims.

### 6.4 Web cloaking

Phishing kits commonly have built-in cloaking capabilities to redirect crawlers or users behind proxies (perceived to be security analysts) away from deceptive content in order to impede detection. By manually investigating a random sample of phishing kits, we surface two dominant cloaking implementations. The first, present in

652 kits, relies on setting an `htaccess` policy to deny access to over 500 IP address prefixes, some of which relate to brand and trademark monitoring services. The second technique, present in 182 kits, relies on a PHP script that checks an embedded `blacklist.dat` consisting of over 1,300 IP address prefixes. Based on annotations embedded in the blacklist file, these addresses belong to cloud providers including Google, Amazon, and OVH; security crawlers including PhishTank and other anti-virus brands; and anonymous proxies like Tor. Other cloaking strategies rely on anti-fraud protections provided by MaxMind and FraudLabs Pro that detect proxies or anonymous access—in this case re-purposed to flag inorganic users accessing phishing pages. Overall, the cloaking strategies of kits match those reported by Invernizzi et al. as popular among blackhat search engine optimization [22], indicating a common core of blackhat technologies.

## 7 DISCUSSION

**Mitigation techniques.** The sheer scale of blackmarket activity surrounding stolen credentials highlights the fragility of authentication schemes built solely on usernames and passwords. A number of online services have enhanced their identity services beyond “something you know” to include approximations of “something you have,” while avoiding the overhead of requiring a mobile device or hardware token. As we discussed in Section 5, this can include a user’s device profiles (e.g., User-Agents or machine identifiers) or a user’s physical location. As we have shown, these additional factors are not intractable for attackers to identify, collect, and impersonate. In the case of geolocation, this means selecting a network proxy in the vicinity of a hijacking target. Despite these limitations, adapting authentication schemes to include risk profiles significantly reduces the danger from having valid passwords exposed in credential leaks, or bundled and traded absent the full user profile.

Immediate solutions to the shortcomings of risk profiles include migrating users to unphishable two-factor authentication (2FA) or password managers that associate credentials with specific domains. While these schemes are susceptible to malware, our results suggest that the threat posed by credential leaks and phishing is orders of magnitude larger than keyloggers at present. Nevertheless, as Bonneau et al. point out, there are various barriers to adopting 2FA and password managers with respect to ease-of-use, recovery from loss, and trusting third parties [3]. Likewise, user knowledge of these schemes is spotty [1, 23]. Our own results indicate that less than 3.1% who fall victim to hijacking subsequently enable any form of two-factor authentication after recovering their account. As such, user education remains a major initiative for enhancing account security.

**Evolution of blackmarket tools.** Compared to the capabilities of keyloggers and phishing kits dating back to the mid-2000s (Section 2), we observe a marked lack of pressure on blackhat developers to evolve their core technologies. Phishing kits reported nearly a decade ago still rely on the same PHP skeleton and approach for reporting stolen credentials. The only modifications have been to collect new challenge questions and ancillary authentication details. Likewise, all of the keylogger variants in our study provide identical capabilities; it is only the interface and customer support that differ.

Our findings illustrate that despite significant research in the space, Internet users continue to fall victim to the same threats.

## 8 CONCLUSION

In this work we presented the first longitudinal measurement study of how miscreants obtain stolen credentials and subsequently bypass risk-based authentication schemes to hijack a victim’s account. In total, we identified 788,000 potential victims of off-the-shelf keyloggers; 12.4 million potential victims of phishing kits; and 1.9 billion usernames and passwords exposed via data breaches and traded on blackmarket forums. Through a combination of password re-use across thousands of online services and targeted collection, we estimated 7–25% of stolen passwords in our dataset would enable an attacker to log in to a victim’s Google account and thus take over their online identity due to transitive trust. However, we showed how blocking login attempts that fail to match a user’s historical login behavior or device profile help mitigate the risk of data breaches and keyloggers, and to a lesser extent phishing. We are now using these insights to improve our login defenses for all users. Our findings illustrate the global reach of the underground economy surrounding credential theft and the need to educate users about password managers and unphishable two-factor authentication as a potential solution.

## 9 ACKNOWLEDGEMENTS

This work was supported in part by a National Science Foundation fellowship and by NSF award CNS-1237265. Opinions expressed in this paper do not necessarily reflect those of the research sponsors.

## REFERENCES

- [1] Lillian Ablon, Paul Heaton, Diana Catherine Lavery, and Sasha Romanosky. Consumer attitudes toward data breach notifications and loss of personal information. In *Proceedings of the Workshop on Economics of Information Security (WEIS)*, 2016.
- [2] Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2012.
- [3] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. The quest to replace passwords: a framework for comparative evaluation of web authentication schemes. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2012.
- [4] Tadek Pietraszek, Borbala Benko, Elie Bursztein and Mark Risher. Cleaning up after password dumps. <https://security.googleblog.com/2014/09/cleaning-up-after-password-dumps.html>, 2014.
- [5] Elie Bursztein, Borbala Benko, Daniel Margolis, Tadek Pietraszek, Andy Archer, Allan Aquino, Andreas Pitsillidis, and Stefan Savage. Handcrafted fraud and extortion: manual account hijacking in the wild. In *Proceedings of the Internet Measurement Conference*, 2014.
- [6] Blake Butler, Brad Wardman, and Nate Pratt. REAPER: an automated, scalable solution for mass credential harvesting and OSINT. In *eCrime Researchers Summit*, 2016.
- [7] Hsien-Cheng Chou, Hung-Chang Lee, Hwan-Jeu Yu, Fei-Pei Lai, Kuo-Hsuan Huang, and Cih-Wen Hsueh. Password cracking based on learned patterns from disclosed passwords. *IJICIC*, 2013.
- [8] Marco Cova, Christopher Kruegel, and Giovanni Vigna. There is no free phish: an analysis of “free” and live phishing kits. In *Proceedings of the USENIX Workshop on Offensive Technologies*, 2008.
- [9] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The tangled web of password reuse. In *Symposium on Network and Distributed System Security (NDSS)*, 2014.
- [10] Matteo Dell’Amico, Pietro Michiardi, and Yves Roudier. Password strength: an empirical analysis. In *Proceedings of IEEE INFOCOM*, 2010.
- [11] Serge Egelman, Joseph Bonneau, Sonia Chiasson, David Dittrich, and Stuart Schechter. It’s not stealing if you need it: a panel on the ethics of performing research using public data of illicit origin. In *International Conference on Financial Cryptography and Data Security*, 2012.

- [12] Lorenzo Franceschi-Bicchieri. Hacker tries to sell 427 million stolen myspace passwords for \$2,800. [https://motherboard.vice.com/en\\_us/article/427-million-myspace-passwords-emails-data-breach](https://motherboard.vice.com/en_us/article/427-million-myspace-passwords-emails-data-breach), 2016.
- [13] David Mandell Freeman, Sakshi Jain, Markus Dürmuth, Battista Biggio, and Giorgio Giacinto. Who are you? a statistical approach to measuring user authenticity. In *Symposium on Network and Distributed System Security (NDSS)*, 2016.
- [14] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010.
- [15] Samuel Gibbs. Dropbox hack leads to leaking of 68m user passwords on the internet. <https://www.theguardian.com/technology/2016/aug/31/dropbox-hack-k-passwords-68m-data-breach>, 2016.
- [16] Vinu Goel and Nicole Perloth. Yahoo says 1 billion user accounts were hacked. <https://www.nytimes.com/2016/12/14/technology/yahoo-hack.html>, 2016.
- [17] Andy Greenberg. Hackers hit macron with huge email leak ahead of french election. <https://www.wired.com/2017/05/macron-email-hack-french-election/>, 2017.
- [18] Robert Hackett. LinkedIn lost 167 million account credentials in data breach. <http://fortune.com/2016/05/18/linkedin-data-breach-email-password/>, 2016.
- [19] Xiao Han, Nizar Kheir, and Davide Balzarotti. Phisheye: live monitoring of sandboxed phishing kits. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2016.
- [20] Thorsten Holz, Markus Engelberth, and Felix Freiling. Learning more about the underground economy: a case-study of keyloggers and dropzones. In *European Symposium on Research in Computer Security (ESORICS)*, 2009.
- [21] Mat Honan. How apple and amazon security flaws led to my epic hacking. <https://www.wired.com/2012/08/apple-amazon-mat-honan-hacking/>, 2012.
- [22] Luca Invernizzi, Kurt Thomas, Alexandros Kapravelos, Oxana Comanescu, Jean-Michel Picod, and Elie Bursztein. Cloak of visibility: detecting when machines browse a different web. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2016.
- [23] Iulia Ion, Rob Reeder, and Sunny Consolvo. ... no one can hack my mind: comparing expert and non-expert security practices. In *Symposium on Usable Privacy and Security (SOUPS)*, 2015.
- [24] Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. Guess again (and again and again): measuring password strength by simulating password-cracking algorithms. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2012.
- [25] Brian Krebs. Adobe breach impacted at least 38 million users. <https://krebsonsecurity.com/2013/10/adobe-breach-impacted-at-least-38-million-users/>, 2013.
- [26] Edmund Lee. Ap twitter account hacked in market-moving attack. <https://www.bloomberg.com/news/articles/2013-04-23/dow-jones-drops-recovers-after-false-report-on-ap-twitter-page>, 2013.
- [27] William R Marczak, John Scott-Railton, Morgan Marquis-Boire, and Vern Paxson. When governments hack opponents: a look at actors and technology. In *Proceedings of the USENIX Security Symposium*, 2014.
- [28] Bakuei Matsukawa, David Sancho, Lord Alfred Remorin, Robert McArdle, and Ryan Flores. Predator pain and limitless when cybercrime turns into cyberspying. <https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-predator-pain-and-limitless.pdf>, 2014.
- [29] William Melicher, Blase Ur, Sean M Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Fast, lean and accurate: modeling password guessability using neural networks. In *Proceedings of the USENIX Security Symposium*, 2016.
- [30] Tyler Moore and Richard Clayton. Discovering phishing dropboxes using email metadata. In *eCrime Researchers Summit*, 2012.
- [31] Jeremiah Onaolapo, Enrico Mariconti, and Gianluca Stringhini. What happens after you are pwnd: understanding the use of leaked account credentials in the wild. In *Proceedings of the Internet Measurement Conference*, 2016.
- [32] Nicole Perloth and Michael D. Shear. Private security group says russia was behind John Podesta's email hack. <https://www.nytimes.com/2016/10/21/us/private-security-group-says-russia-was-behind-john-podestas-email-hack.html>, 2016.
- [33] Richard Shay, Iulia Ion, Robert W Reeder, and Sunny Consolvo. "My religious aunt asked why I was trying to sell her viagra": experiences with account hijacking. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, 2014.
- [34] Elizabeth Stobert and Robert Biddle. The password life cycle: user behaviour in managing passwords. In *Proc. SOUPS*, 2014.
- [35] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. Your botnet is my botnet: analysis of a botnet takeover. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2009.
- [36] Kurt Thomas, Frank Li, Chris Grier, and Vern Paxson. Consequences of connectivity: characterizing account hijacking on Twitter. In *Proceedings of the Conference on Computer and Communications Security*, 2014.
- [37] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. Understanding password choices: how frequently entered passwords are re-used across websites. In *Symposium on Usable Privacy and Security (SOUPS)*, 2016.
- [38] Matt Weir, Sudhir Aggarwal, Breno De Medeiros, and Bill Glodek. Password cracking using probabilistic context-free grammars. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2009.
- [39] Shams Zawoad, Amit Kumar Dutta, Alan Sprague, Ragib Hasan, Jason Britt, and Gary Warner. Phish-net: investigating phish clusters using drop email addresses. In *eCrime Researchers Summit*, 2013.
- [40] Kim Zetter. Group posts e-mail hacked from Palin account – update. <https://www.wired.com/2008/09/group-posts-e-m>, 2008.