What Happens After You Are Pwnd: Understanding The Use Of Leaked Account Credentials In The Wild

Jeremiah Onaolapo, Enrico Mariconti, and Gianluca Stringhini
University College London
{j.onaolapo,e.mariconti,g.stringhini}@cs.ucl.ac.uk

ABSTRACT

Cybercriminals steal access credentials to online accounts and then misuse them for their own profit, release them publicly, or sell them on the underground market. Despite the importance of this problem, the research community still lacks a comprehensive understanding of what these stolen accounts are used for. In this paper, we aim to shed light on the modus operandi of miscreants accessing stolen Gmail accounts. We developed an infrastructure that is able to monitor the activity performed by users on Gmail accounts, and leaked credentials to 100 accounts under our control through various means, such as having information-stealing malware capture them, leaking them on public paste sites, and posting them on underground forums. We then monitored the activity recorded on these accounts over a period of 7 months. Our observations allowed us to devise a taxonomy of malicious activity performed on stolen Gmail accounts, to identify differences in the behavior of cybercriminals that get access to stolen accounts through different means, and to identify systematic attempts to evade the protection systems in place at Gmail and blend in with the legitimate user activity. This paper gives the research community a better understanding of a so far understudied, yet critical aspect of the cybercrime economy.

1. INTRODUCTION

The wealth of information that users store in accounts on online services such as Gmail, Dropbox, and Facebook, as well as the possibility of misusing them for illicit activities have attracted cybercriminals, who actively engage in compromising such accounts. Miscreants obtain the credentials to victims' online accounts by performing phishing scams [13], by infecting users with information-stealing malware [23] or by compromising the databases of websites that contain such information [5]. Such credentials are then sold on the black market to other cybercriminals who wish to use the stolen accounts for profit. This ecosystem has become a very sophisticated market in which only vetted sellers are allowed to join [24].

Cybercriminals can use compromised accounts in multiple ways. First, they can use them to send spam [14]. This practice is particularly effective because of the established reputation of such accounts: the already-established contacts of the account are likely to trust its owner, and are therefore more likely to open the messages that they receive from her [16]. Similarly, the stolen account is likely to have a history of good behavior with the online service, and the malicious messages sent by it are therefore less likely to be detected as spam, especially if the recipients are within the same service (e.g., a Gmail account used to send spam to other Gmail accounts) [27]. Because of these advantages, the developers of large spamming botnets include the opportunity to instruct their bots to use stolen webmail service accounts to deliver spam [24]. Alternatively, cybercriminals can use the stolen accounts to collect sensitive information about the victim. Such information can include financial credentials (credit card numbers, bank account numbers), login information to other online services, and personal communications of the victim [11].

Despite the importance of stolen accounts for the underground economy, there is surprisingly little work on the topic. Bursztein et al. [11] studied the modus operandi of cybercriminals collecting Gmail account credentials through phishing scams. Their paper shows that criminals access these accounts to steal financial information from their victims, or use these accounts to send fraudulent emails. Despite the interesting insights, the narrowness of their threat model keeps many questions unanswered. Other researchers did not attempt studying the activity of criminals on compromised online accounts because it is usually difficult to monitor what happens to them without being a large online service. The rare exceptions are studies that look at information that is publicly observable, such as the messages shared on Twitter by compromised accounts [14, 15].

To close this gap, in this paper we present a system that is able to monitor the activity performed by attackers on Gmail accounts. To this end, we instrument the accounts by using *Google Apps Script* [1]; by doing so, we are able to monitor any time an email is read, favorited, sent, or a new draft is created. We also monitor the accesses that the accounts receive, with particular attention to their system configuration and their origin. We call such accounts *honey accounts*. To allow researchers to set up their own honeypot infrastructures, we will release the source code of our honey account system publicly.

We set up 100 honey accounts, each resembling the Gmail account of the employee of a fictitious company. To understand how criminals use these accounts after they get compromised, we leak the credentials to such accounts on multiple outlets, modeling the different ways in which cybercriminals share and get access to such credentials. First, we leak credentials on paste sites, such as `pastebin` [4]. Paste sites are commonly used by cybercriminals to post account credentials after data breaches [2]. We also leak them to underground forums, which have been shown to be the place where cybercriminals gather to trade stolen commodities such as account credentials [24]. Finally, we login to our honey accounts on virtual machines that have been previously infected with information stealing malware. By doing this, the credentials will be sent to the cybercriminal behind the command and control infrastructure, and will then be used directly or placed on the black market for sale [23]. We know that there are other outlets that attackers use, for instance, phishing and data breaches, but we decided to focus on the paste sites, underground forums, and malware in this paper. We worked in close collaboration with the Google anti-abuse team, to make sure that any unwanted activity by the compromised accounts would be promptly blocked. The accounts are configured to send any email to a mail server under our control, to prevent them from successfully delivering spam.

After leaking our credentials, we recorded any interaction with our honey accounts for a period of 7 months. Our analysis allows us to draw a taxonomy of the different actions performed by criminals on stolen Gmail accounts, as well as provide us interesting insights on the keywords that criminals typically search for when looking for valuable information on these accounts. We also show that criminals who obtain access to stolen accounts through certain outlets appear more skilled than others, and make additional efforts to avoid detection from Gmail. For instance, criminals who steal account credentials via malware make more efforts to hide their identity, by connecting from open proxies and the Tor network and disguising their browser user agent. Criminals who obtain access to stolen credentials through paste sites, on the other hand, tend to connect to these accounts from locations that are close to the typical location used by the owner of the account, if this information is shared with them. At the lowest level of sophistication are criminals who browse free underground forums looking for free samples of stolen accounts: these individuals do not take significant measures to avoid detection, and are therefore easier to detect and block.

In summary, this paper makes the following contributions:

- We develop a system to monitor the activity of Gmail accounts. We will publicly release the source code of our system, to allow other researchers to deploy their own Gmail honey accounts and further the understanding that the security community has of malicious activity on online services.
- We deployed 100 honey accounts on Gmail, and leaked credentials through three different outlets: underground forums, public paste sites, and virtual machines infected with information-stealing malware.
- We provide detailed measurements of the activity logged by our honey accounts over a period of 7 months. We show that certain outlets on which credentials are leaked appear to be used by more skilled criminals, who act stealthy and actively attempt to evade detection systems.

2. BACKGROUND

Webmail accounts. Webmail service providers such as Gmail, Yahoo!, and Outlook.com provide their users with a convenient place to store, sort, and manage their emails and contacts. In this paper we focus on Gmail accounts, with particular attention to the actions performed by cybercriminals once they obtain access to someone else's account. We made this choice because Gmail allows users to set up scripts that augment the functionality of their accounts, and it was therefore the ideal platform for developing webmail–based honeypots. To ease the understanding of the rest of this paper, we briefly summarize the capabilities offered by webmail accounts in general, and by Gmail in particular.

In Gmail, after logging in, users are presented with a view of their inbox. The inbox contains all the emails that the user received, and highlights the ones that have not been read yet by displaying them in boldface font. Users have the possibility to mark emails that are important to them and that need particular attention by *starring* them. Users are also given a *search* functionality, which allows them to find emails of interest by typing related keywords. They are also given the possibility to organize their email by placing related messages in folders, or assigning them descriptive labels. Such operations can be automated by creating rules that automatically process received emails. When writing emails, content is saved in a *Drafts* folder until the user decides to send it. Once this happens, sent emails can be found in a dedicated folder, and they can be searched similarly to what happens for received emails.

Threat model. Cybercriminals can get access to account credentials in three ways. First, they can perform social engineering-based scams, such as setting up phishing web pages that resemble the login page of popular online services [13] or sending spearphishing emails pretending to be members of customer support teams at such online services [26]. As a second way of obtaining user credentials, cybercriminals can install malware on victim computers and configure it to report back any account credentials typed by the user to the command and control server of the botnet [23]. As a third way of obtaining access to user credentials, cybercriminals can exploit vulnerabilities in the databases used by online services to store them [5].

After stealing account credentials, a cybercriminal can either use them privately for his own profit, release them publicly, or sell them on the underground market. Previous work studied the modus operandi of cybercriminals stealing user accounts through phishing and using them privately [11]. In this work, we study a broader threat model in which we mimic cybercriminals leaking credentials on paste sites [4] as well as miscreants advertising them for sale on underground forums [24]. In particular, previous research showed that cybercriminals often offer a small number of account credentials for free to test their "quality." We followed a similar approach, pretending to have more accounts for sale, but never following up to any further inquiries. In addition, we simulate infected victim machines in which the malware steals the user's credentials and sends them to the cybercriminal. We describe our setup and how we leaked account credentials on each outlet in detail in Section 3.2.

Finally, there are a number of actions that a cybercriminal who obtains access to a victim account can perform. Such actions include sending spam [14] or collecting sensitive information from the account [11]. In Section 3.1 we describe how we developed an infrastructure to monitor the activity of cybercriminals on our honey accounts, while in Section 4.2 we analyze in detail the types of activity that we observed.

3. METHODOLOGY

Our overall goal was to gain a better understanding of malicious activity in compromised webmail accounts. To achieve this goal, we developed a system able to monitor the activity on Gmail accounts. We instrumented these accounts to log all accesses they receive, and actions taken on them by cybercriminals. We set up a monitor infrastructure to keep track of this activity information. We then deployed 100 honey accounts on Gmail. We proceeded to leak the accounts on different outlets, namely popular paste websites, underground forums, and information-stealing malware. The idea is to study differences in malicious activity across these outlets. In the following section, we describe our system architecture and our experiment setup in detail.

3.1 System overview

Our system comprises a number of components, namely, honey accounts and monitor infrastructure. These components are described in this section.

Honey accounts. Our honey accounts are webmail accounts instrumented with Google Apps Script, to monitor activity in them. Google Apps Script is a cloud-based scripting language based on JavaScript, originally designed to augment the functionality of Gmail accounts and Google Drive documents, in addition to building web apps [3]. Google Apps Script provides APIs for performing time-triggered and event-triggered tasks, for instance, sending an email reminder once every day, about emails marked "important." We incorporated scripts into each account to monitor the emails in the account honeypots. The scripts send notifications to a dedicated webmail account under our control whenever an email is read, sent or "starred." In addition to information on actions taken on emails, the scripts also send copies of all draft emails created in the honey accounts to us.

We included functions to scan the emails in each honey account every 10 minutes, to report all discovered changes, namely, read, sent, starred and draft emails, back to us. We also added a "heartbeat message" function, to send us a predefined message once a day from each honey account, to attest that the account was still functional and had not been blocked by Google. The Google Apps Script was well hidden in a Google Docs Spreadsheet stored in the honey accounts. We believe that this measure makes it unlikely for attackers to find and delete our scripts. To minimize abuse, we changed each honeypot account's default *send-from* address to an email address pointing to a modified mailserver under our control. All emails sent from the account honeypots are delivered to the mailserver, which simply dumps the emails to disk and does not forward them to the intended destination.

Monitoring infrastructure. Google Apps Scripts are quite powerful, but they do not provide enough information in some cases. For example, they do not provide information about the location of IP addresses of accesses to webmail accounts. To keep track of such accesses, we set up scripts to drive a web browser and periodically login to each honey account, and record information about visitors (cookie identifier, geolocation information and times of accesses, among others). The scripts navigate to the visitor activity pages in the honey accounts, and dump the pages to disk, for offline parsing. It is interesting to note that by taking advantage of the Gmail activity page we are able to leverage Google's geolocation system, as well as their operating system fingerprinting techniques. In addition, notifications of actions on honey accounts are sent to a dedicated webmail account we set up as a notifications store. The notifications are sent by Google Apps Scripts and retrieved by another script managed by us.

We believe that our honey account and monitoring framework unleashes multiple possibilities for researchers who want to further study the behavior of attackers in webmail accounts. For this reason, we will release the source code of our system upon publication of this paper and will release it publicly.

3.2 Experiment setup

As part of our experiments, we first set up a number of honey accounts on Gmail, and then leaked them on multiple outlets used by cybercriminals.

Honey account setup. We created 100 Gmail accounts and assigned them random combinations of popular first and last names, similar to what was done in [25]. Creating and setting up these accounts is a manual process, and this is why it was not possible for us to create more than 100 accounts. Besides, Google rate-limits the opening of new accounts from the same IP address by presenting a phone verification page after a few accounts have been created. This limits the number of honey accounts we can set up in practice.

We populated the freshly-created accounts with emails from the public Enron email dataset [19]. This dataset contains the emails sent by the executives of the energy corporation Enron and was publicly released as evidence for the bankruptcy trial of the company. This dataset is suitable for our purposes, since the emails that it contains are the typical emails exchanged by corporate users. To make the honey accounts believable and avoid raising suspicion from cybercriminals accessing them, we mapped distinct recipients in the Enron dataset to our fictional characters, and replaced the original first names and last names in the dataset with our honey first names and last names. In addition, we changed all instances of "Enron" to a fictitious company name that we came up with, and updated all dates contained in the emails to reflect the time in which the accounts were populated.

Leaking account credentials. To achieve our objectives, we had to entice cybercriminals to interact with our account honeypots, while we logged their accesses. We selected paste sites and underground forums as appropriate venues for leaking account credentials, since they tend to be misused by cybercriminals for dissemination of stolen credentials. We also decided to set up virtual machines and perform logins to honey accounts after infecting them with information-stealing malware, since this is a popular way in which professional cybercriminals obtain access to stolen accounts [9]. Information stealing malware collects users' form data, and uploads it to the C&C server.

We divided the account honeypots in groups and leaked their credentials in different locations, as shown in Table 1. We leaked 50 accounts in total on paste sites. For 20 of them we leaked basic credentials (username and password pairs) on the popular paste sites `pastebin.com` and `pastie.org`. We then leaked 10 account credentials on popular Russian paste websites (`p.for-us.nl` and `paste.org.ru`). For the remaining 20 accounts we leaked username and password pairs along with location and date of birth information of the fictional character that "owned" each account. We specified locations close to London, UK for 10 of them, while we specified locations in the Midwest of the US for the other 10 accounts. By averaging the longitude and latitude of these locations, the middle point falls in Pontiac, Illinois.

group	number of honey accounts	outlet of leak
1	30	popular paste websites (no location information)
2	20	popular paste websites (including location information)
3	10	underground forums (no location information)
4	20	underground forums (including location information)
5	20	malware (no location information)

Table 1: List of account honeypot groupings and where we leaked them.

We leaked 30 account credentials on underground forums. For 10 of them we only specified username and password pairs, without additional information. Similar to what was done for paste sites, we leaked a group of 10 accounts each with additional information that claimed that the "owners" of such accounts were based near London, UK and in Midwestern US cities with a middle point in Pontiac, respectively. To leak credentials, we used these forums: `offensivecommunity.net`, `bestblackhatforums.eu`, `hackforums.net`, and `blackhatworld.com`. We selected these forums because they were open for anybody to register, and were highly ranked in Google results. We acknowledge that some underground forums are not open, and have a strict vetting policy to let users in [24]. Unfortunately, however, we did not have access to any private forum. In addition, the same approach of studying open underground forums has been used by previous work [6]. When leaking credentials on underground forums, we mimicked the modus operandi of cybercriminals that was outlined by Stone-Gross et al. in [24]. In this paper, the authors showed that cybercriminals often post a sample of their stolen datasets on the forum to show that the accounts are real, and promise to provide additional data in exchange for a fee. We logged the messages that the accounts we created received on underground forums, mostly inquiring about obtaining the full dataset, but we did not follow up to them.

Finally, to study activity of information-stealing malware in honey accounts, we leaked basic credentials of 20 accounts to information-stealing malware samples, using the malware honeypot infrastructure described in the following section. To this end, we selected malware samples from the Zeus family, which is one of the most popular malware families performing information stealing [9], as well as from the Corebot family. We will provide detailed information on our malware honeypot infrastructure in the next section.

The reason for leaking different accounts on different outlets is to study differences in the behavior of cybercriminals getting access to stolen credentials through different sources. The reason for providing different additional information (in some cases, no additional information) across the different groups is to observe differences in malicious activity depending on the amount and type of information available to the cybercriminal. As we will show in Section 4, the accesses that were observed in our honey accounts were heavily influenced by the presence of additional location information in the leaked content.

Malware honeypot infrastructure. We set up a sandbox and infected it with malware samples; these samples captured the credentials that a web browser was using to login on Gmail. Our sandbox works as follows: a web server entity manages the honey credentials (usernames and passwords) and malware samples; once the host creates the Virtual Machine (VM), it sends a request to the web server for the malware executable file and another one for the honey credential. The structure is similar to the one explained in [17]. The malware sample is then executed; after a timeout, a script carries out the login operation using the downloaded credential, in order to expose the honey credential to the malware that is already running in the VM. After a certain lifetime the VM is deleted and a new one is created; this new VM downloads another malware sample and different credential, and repeats the login operation.

To maximize the efficiency of the configuration, before the experiment we carried out a test without the Gmail login component to select only samples whose C&C servers were still up and running. To avoid contributing to spam campaigns or other malicious actions we set up a sinkhole server and followed other prudent practices described by [22], such as limiting the bandwidth of the network interfaces to avoid Denial of Service attacks, and limiting the lifetime of each VM.

3.3 Threats to validity

We acknowledge that seeding the honey accounts with emails from the Enron dataset may introduce bias into our results, and may make the honey accounts less believable to visitors. Another threat to validity is that the honey accounts were pre-populated with emails, and we did not send more emails to them while leaking them. Some visitors may notice that the honey accounts did not receive any new emails during the period of observation, and this may affect our results. Finally, since we only leaked honey credentials to the outlets listed previously (namely paste sites, underground forums, and malware), our results reflect the activity of participants on present on those outlets only. Besides, in the case of the underground forums, it is possible that the sophisticated cybercriminals from the underground forums did not interact with our honey accounts. This is because we leaked credentials through freshly created accounts on those forums, and freshly created accounts lack the level of high reputation it takes to interact with sophisticated cybercriminals in the underground forums. We also acknowledge that to generalize our findings, we would require more than 100 Gmail accounts.

Despite these issues, we believe that our methodology will help researchers to further understand the underground economy of stolen accounts, and that our results shed light into what happens to stolen accounts. It is currently hard to gather data on compromised accounts, and our system provides a novel way to do so, especially for researchers. Our methodology can be applied to other online accounts, for instance, those hosting documents, and our results can be used in the process of building tools to detect and mitigate illegitimate accesses to online accounts. As it can be seen, our results already show interesting trends and findings, which we intend to explore further in future work.

3.4 Ethics

The experiments performed in this paper require some ethical considerations. First of all, by giving access to our honey accounts to cybercriminals, we incur the risk that these accounts will be used to damage third parties. To minimize this risk, as we said, we configured our accounts in a way that all emails would be forwarded to a sinkhole mailserver under our control and never delivered to the outside world. We also established a close collaboration with Google and made sure to report to them any malicious activity that needed attention. Although the suspicious login filters that Google typically uses to protect their accounts from unauthorized accessed were disabled for our honey accounts, all other malicious activity detection algorithms were still in place, and in fact Google suspended a number of accounts under our control that attempted to send spam. To mitigate the possibility of cybercriminals finding our honey scripts, deleting them, and locking us out of the honey accounts we provided Google with a full list of our honey accounts, so that they could monitor them in case something went wrong. Another point of risk is ensuring that the malware ran in our virtual environment would not be put in condition to harm third parties. As we said, we followed common practices [22] such as restricting the bandwidth available to our virtual machines and sinkholing all email traffic sent by them. Finally, our experiments are inherently deceiving cybercriminals by providing them fake accounts with fake personal information in them. To make sure that our experiments were ran in an ethical fashion, we sought and obtained IRB approval by our institution.

4. DATA ANALYSIS

We monitored the activity on our honey accounts for a period of 7 months, from 25th June, 2015 to 16th February, 2016. In this section, we first provide a taxonomy of the types of activity that we observed. We provide a detailed analysis of the type of activity monitored on our honey accounts, focusing on the differences in modus operandi shown by cybercriminals who obtain credentials to our honey accounts from different outlets. We then investigate whether cybercriminals attempt to evade location-based detection systems by connecting from locations that are closer to where the owner of account typically connects from. We also develop a metric to infer which keywords attackers search for when looking for interesting information in an email account. Finally, we analyze how certain types of cybercriminals appear to be stealthier and more advanced than others.

4.1 Overview

We created and instrumented 100 Gmail accounts for our experiments. We observed 327 unique accesses to the accounts during the experiment, during which 147 emails were read, 845 emails were sent, and there were 12 unique draft emails composed by cybercriminals. To avoid biasing our results, we removed all accesses made to honey accounts by IP addresses from our monitoring infrastructure. We also removed all accesses that originated from the city where our monitoring infrastructure is located. 42 accounts were blocked by Google during the course of the experiment, since the cybercriminals that accessed them violated Google's Terms and Conditions while doing so.We developed a taxonomy of cybercriminals accessing the honey accounts, which we explain in detail in the next section.

4.2 A taxonomy of account activity

From our dataset of activity observed in the honey accounts, we devise a taxonomy of attackers based on unique accesses to such accounts. We identify four types of attackers, described in detail in the following.

Curious. These accesses constitute the most basic type of access to stolen accounts. After getting hold of account credentials, people login on those accounts to check if such credentials work. Afterward, they do not perform any additional action. The majority of the observed accesses belong to this category, accounting for 224 accesses.

Gold Diggers. When getting access to a stolen account, attackers often want to understand its worth. For this reason, on logging into honey accounts, some attackers search for sensitive information, such as account information and attachments that have financial-related names. They also seek information that may be useful in spearphishing attacks. We call these accesses "gold diggers." Previous research showed that this practice is quite common for manual account hijackers [11]. In this paper, we confirm that finding, provide a methodology to assess the keywords that cybercriminals search for, and analyze differences in the modus operandi of gold digger accesses for credentials leaked through different outlets. In total, we observed 82 accesses of this type.

Spammers. One of the main capabilities of webmail accounts is sending emails. Previous research showed that large spamming botnets have code in their bots and in their C&C infrastructure to take advantage of this capability, by having the bots directly connect to such accounts and send spam [24]. We consider accesses to belong to this category if they send any email. We observed 8 accounts of this type. This low number of accesses shows that sending spam appears not to be one of the main purposes that cybercriminals use stolen accounts for.

Hijackers. A stealthy cybercriminal is likely to keep a low profile when accessing a stolen account, to avoid raising suspicion from the account's legitimate owner. Less concerned miscreants, however, might just act to lock the legitimate owner out of their account by changing the account's password. We call these accesses "hijackers." In total, we observed 36 accesses of this type. It is interesting to note that a change of password prevents us from scraping the accesses page, and therefore we are unable to collect further information about the accesses performed to that account. However, since the password to the account now does not match the leaked one, it is reasonable to assume that the person performing further logins to such honey accounts is now either the hijacker himself, or someone that the hijacker gave (perhaps sold) the account credentials to. Interestingly, even after losing control of the accounts, our monitoring scripts embedded in the accounts keep running, and therefore we keep receiving information on the emails that are read, sent, or starred in that account.

It is important to note that the taxonomy classes that we described are not exclusive. For example, an at-

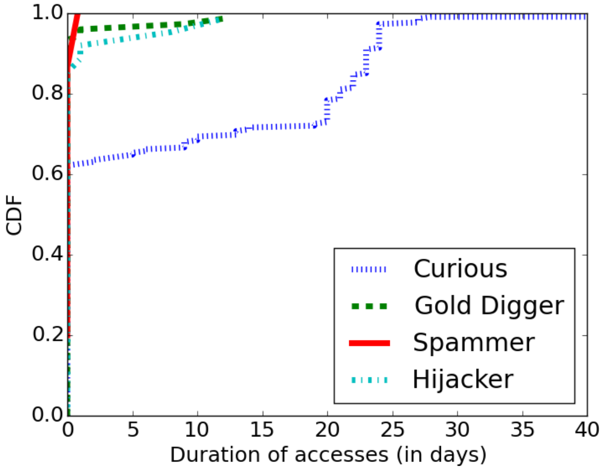


Figure 1: CDF of the length of unique accesses for different types of activity on our honey accounts.

tacker might change the password of an account, therefore falling into the "hijacker" category and then use it to send spam emails, therefore falling in the "spammer" category too. Such overlaps happened often for the accesses recorded in our honey accounts. It is interesting to note that there was no access that behaved exclusively as "spammer." Miscreants who sent spam through our honey accounts also acted as "hijackers" or as "gold diggers," searching for sensitive information in the account.

Figure 1 shows plots of the Cumulative Distribution Function (CDF) of the length of accesses of different types of attackers. As for the previous analysis, these accesses identify cookies, and account for the time between the first and the last time a cookie is observed on a certain honey account. As it can be seen, the vast majority of accesses are very short, lasting only a few minutes and never coming back. "Spammer" accounts, in particular, tend to send emails in burst for a certain period and then disconnect. "Hijacker" and "gold digger" accesses, on the other hand, have a long tail of about 10% accesses that keep coming back for several days in a row. For "hijacker" accesses, the statistics reported are only a lower bound of the actual time that a cookie was active on an account: as we said, after the account password is changed, we lose the capability of tracking accesses (but not the information on the interaction with the accounts, for example which emails were read). The CDF shows that most "curious" accesses are repeated over many days, indicating that the cybercriminals keep coming back to find out if there is new information in the accounts. This conflicts with the finding in [11], which states that most cybercriminals connect to a compromised webmail account once, to assess its value within a few minutes. However, [11] focused only accounts compromised via phishing pages, while we have on a broader threat model, as earlier stated.

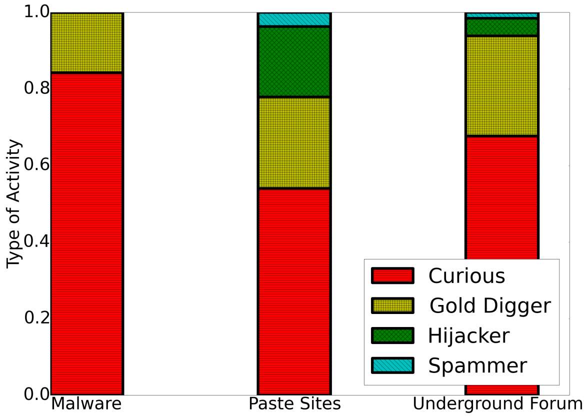


Figure 2: Distribution of types of accesses for different credential leak accesses.

We wanted to understand the distribution of different types of accesses in accounts that were leaked through different means. Figure 2 shows a breakdown of this distribution. As it can be seen, cybercriminals who get access to stolen accounts through malware are the stealthiest, and never lock the legitimate users out of their account. Instead, they limit their activity to checking if such credentials are real or searching for sensitive information in the account inbox, perhaps in an attempt to estimate the value of the accounts. Accounts leaked through paste sites and underground forums see the presence of "hijackers." 20% of the accesses to accounts leaked through paste sites, in particular, belong to this category. Accounts leaked through underground forums, on the other hand, see the highest percentage of "gold digger" accesses, with about 30% of all accesses belonging to this category.

4.3 Activity on honey accounts

Google identifies each access to a Gmail account with a cookie identifier. For each unique access observed in each account honeypot, we recorded the cookie identifier, the time of first access as t_0, and the time of last known access as t_{last}. In the case of repeated accesses with the same cookie identifier (i.e., multiple visits by the same attacker), we chose t_{last} as the time of the last recorded visit, while t_0 remains the time of first access. From this information, we computed the duration of activity as $t_{last} - t_0$ per cookie. The majority of accesses are short, especially in groups leaked on paste sites and underground forums. We observed that 80% of visitors to accounts leaked on paste sites and underground forums never came back after their first accesses, while 80% of visitors to accounts leaked to malware came back after some days.

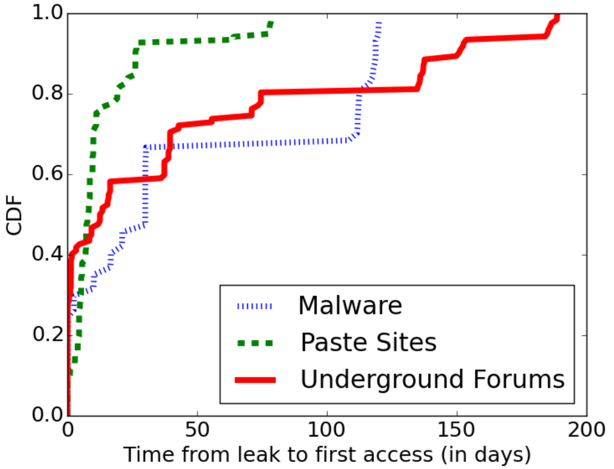


Figure 3: CDF of the time passed after account credentials are leaked and the first access by a cookie is recorded.

We also studied how long it takes for cybercriminals to access accounts in the different groups, after we leak them. We measured the duration between time of first leak in each group, and the time of first access. We generated a CDF to compare durations across malware, paste sites and underground forums, as shown in Figure 3. Similarly, we generated a time-based plot of activity in the honey accounts, by computing duration between time of first leak in each group, and the time of unique access. This is shown in Figure 4. The idea is to identify time patterns of unique accesses.

Figure 3 shows that within the first 25 days, the following were recorded: 80% of all unique accesses to accounts leaked to paste sites, 60% of all unique accesses to accounts leaked to underground forums, and 40% of all unique accesses to accounts leaked to malware. A particularly interesting observation is the nature of unique accesses to accounts leaked to malware. A close look at Figure 3 reveals rapid increases in unique accesses to honey accounts leaked to malware, about 30 days after the leak, and also after 100 days, indicated by 2 sharp inflection points.

Figure 4 sheds more light into what happened at those points. An interesting aspect to note is that accounts that are leaked on public outlets such as forum and paste sites can be accessed by multiple cybercriminals at the same time. Account credentials leaked through malware, on the other hand, are available only to the botmaster that stole them, until they decide to sell them or to give them to someone else. Seeing bursts in accesses to accounts leaked through malware months after the actual leak happened could indicate that the accounts were aggregated by the same criminal from different C&C servers or that the accounts were sold on the underground market and that another criminal is now using them. This hypothesis is somewhat confirmed by the fact that these bursts in accesses were of the "gold digger" type, while all previous accesses to the same accounts were of the "curious" type. Another interesting fact shown by Figure 4 is that accounts leaked to malware are accessed multiple times by the malware controllers who stole them, perhaps to check whether the accounts are active. As discussed previously, this is in contrast with what was discovered by previous work [11].

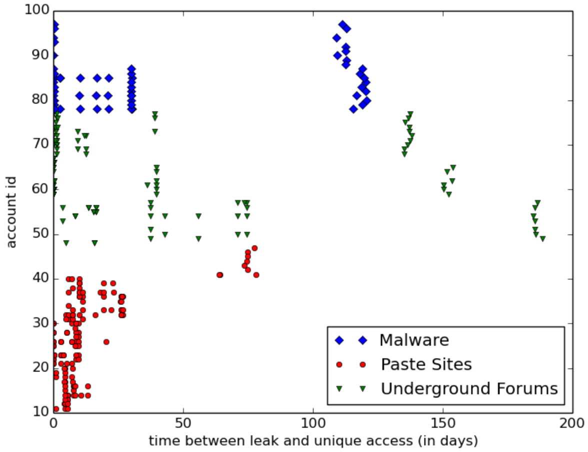


Figure 4: Plot of duration between time of leak and unique accesses in accounts leaked through different outlets.

In addition, Figure 4 shows that majority of accounts leaked to paste sites were accessed within a few days of leak, while a particular subset was not accessed for more than 2 months. That subset refers to the 10 credentials we leaked to Russian paste sites. The corresponding honey accounts were not accessed for more than 2 months from the time of leak. This either indicates that cybercriminals are not many on the Russian paste sites, or they maybe they did not believe that the accounts were real, thus not bothering to access them.

4.4 System configuration of accesses

Using Google's system fingerprinting information available in the honey accounts, we collected information about devices that accessed the honey accounts. Accounts leaked through paste sites and underground forums recorded a fraction of accesses from Android devices, while the accounts leaked through malware only recorded accesses from traditional computers. On the browser side we notice that accesses on accounts leaked through malware always consisted of an empty user agent, making it impossible for Google to identify the browser. Accounts leaked through underground forums and paste sites, on the other hand, present accesses from all the most popular browsers.

4.5 Location of accesses

We recorded the location information that we found in 173 unique accesses, and analyzed them to identify differences in origin location patterns across honey account groups. This was possible because Google provides location information (that is, the city from which a user logged into a Gmail account) in the account activity page of each Gmail account. 154 unique accesses did not have location information in them. According to information we got from Google, they are mostly accesses that originated from Tor exit nodes or anonymous proxies.

We observed accesses from a total of 29 countries. To understand whether the IP addresses that connected to our honey accounts had been recorded in previous malicious activity, we ran checks on all IP addresses we observed, against Spamhaus blacklist. We found 20 IP addresses that accessed our honey accounts in the Spamhaus blacklist. Because of the nature of this blacklist, we believe that the addresses belong to malware-infected machines that are used by cybercriminals to connect to the stolen accounts.

One of our goals was to observe if cybercriminals attempt to evade location-based login risk analysis systems by tweaking access origins. In particular, we wanted to assess whether telling criminals the location where the owner of an account is based influences the locations that they will use to connect to this account. Despite observing 57 accesses to our honey accounts leaked through malware, we discovered that all these connections except one originated from Tor exit nodes. This shows that malware operators prefer to hide their location through the use of anonymizing systems rather than modifying their location based on where the stolen account is typically connecting from. Because of this observation, we focused the further experiments in this section on the accounts that are stolen through paste sites and underground forums.

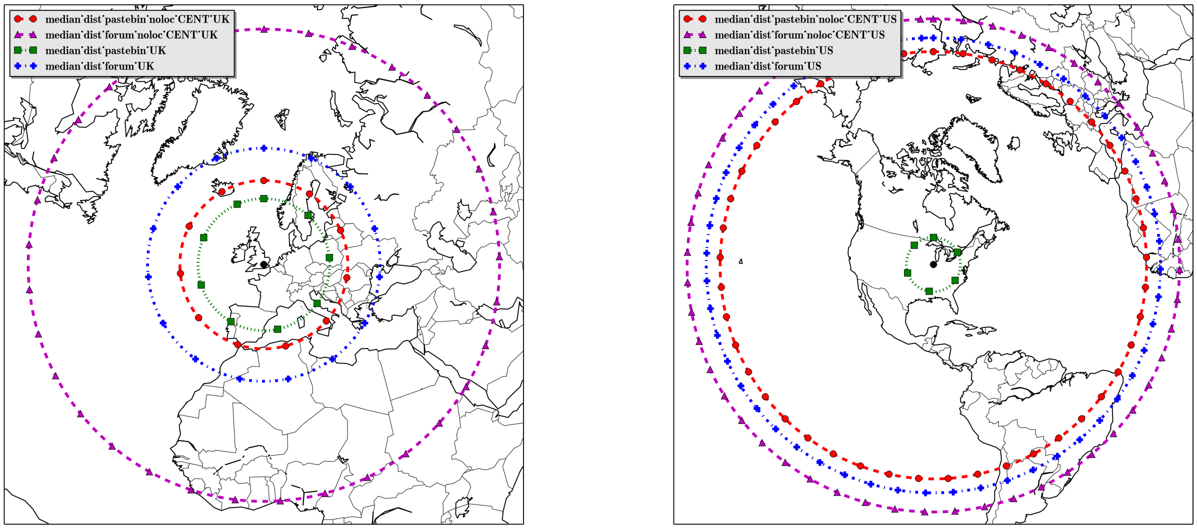
To observe the impact of availability of location information about the honey accounts on the locations that cybercriminals connect from, we calculated the median values of distances of the locations recorded in unique accesses, from the midpoints of the advertised locations in our account leaks. As we mentioned, the midpoint of advertised UK locations is London, while the midpoint of the advertised US locations is in Pontiac, Illinois. For example, for all unique accesses A to honey accounts leaked on paste sites, advertised with UK information, we extracted location information, translated them to coordinates L_A, and computed $dist\_paste\_UK$ vector as $distance(L_A, mid_{UK})$, where mid_{UK} is London's coordinates. All distances are in kilometers. We extracted the median values of all distance vectors obtained, and plotted circles on UK and US maps, specifying those median distances as radii of the circles, as shown in Figures 5a and 5b.

Interestingly, we observe that connections to accounts with advertised location information originate from places closer to the midpoints than accounts with leaked information containing usernames and passwords only. Figure 5a shows that connections to accounts leaked on paste sites result in the smaller median circles, that is, the connections originate from locations closer to London, the UK midpoint. The smallest circle is for the accounts leaked on paste sites, with advertised UK location information (radius 1400 kilometers). In contrast, the circle of accounts leaked on paste sites without location information has a radius of 1784 kilometers. The median circle of the accounts leaked in underground forums, with no advertised location information, is the largest circle in Figure 5a, while the one of accounts leaked in underground forums, along with UK location information, is smaller.

We obtained similar results in US plot, with some interesting distinctions. As shown in Figure 5b, connections to honey accounts leaked on paste sites, with advertised US locations are clustered around the US midpoint, as indicated by the circle with a radius of 939 kilometers, compared to the the median circle of accounts leaked on paste sites without location information, that has a radius of 7900 kilometers. However, despite the fact that the median circle of accounts leaked in underground forums with advertised location information is less than that of the one without advertised location information, the difference in their radii is not much. This again supports the indication that cybercriminals on paste sites exhibit more *location malleability*, that is, masquerading their origins of accesses to appear closer to the advertised location, when provided. It also shows that cybercriminals on forums are less sophisticated, or care less than the ones on paste sites.

Statistical significance. As we explained, Figures 5a and 5b show that accesses to leaked accounts happen closer to advertised locations if this information is included in the leak. To confirm this finding we performed a Cramer Von Mises test [12]. The Anderson version [7] of this test is used to understand if two vectors of values are likely have the same statistical distribution or not. The p value has to be under 0.01 to let us state that it is possible to reject the null hypothesis (i.e. the two vectors of distances have the same distribution), otherwise it is not possible to state with statistical significance if the two distance vectors are derived from the same distribution.

The p value from the test on paste sites vectors (values 0.0017415 for UK location information versus no location and 0.0000007 for US location information versus no location) allows us to reject the null hypothesis stating that the two vectors come from different distributions while we cannot say the same observing the


median˙dist˙pastebin˙noloc˙CENT˙UK
median˙dist˙forum˙noloc˙CENT˙UK
median˙dist˙pastebin˙UK
median˙dist˙forum˙UK
median˙dist˙pastebin˙noloc˙CENT˙US
median˙dist˙forum˙noloc˙CENT˙US
median˙dist˙pastebin˙US
median˙dist˙forum˙US

(a) Distance of login locations from the UK midpoint

(b) Distance of login locations from the US midpoint

Figure 5: Distance of login locations from the midpoints of locations advertised while leaking credentials. Red lines indicate credentials leaked on paste sites with no location information, green lines indicate credentials leaked on paste sites with location information, purple lines indicate credentials leaked on underground forums without location information, while blue lines indicate credentials leaked on underground forums with location information.

p values for the tests on forum vectors (0.272883 in the UK case and 0.272011 in the US one). Therefore, we can conclusively state that the statistical test proves that criminals using paste sites connect from closer locations when location information is provided along with the leaked credentials. We cannot reach that conclusion in the case of accounts leaked to underground forums, although Figures 5a and 5b indicate that there are some location differences in this case too.

4.6 What are "gold diggers" looking for?

Cybercriminals compromise online accounts due to the inherent value of those accounts. As a result, they assess accounts to decide how valuable they are, to decide exactly what to do with such accounts. We decided to study the words that they searched for in the honey accounts, in order to understand and potentially characterize anomalous searches in the accounts, to aid the design of detection systems in the future. A limiting factor in this case was the fact that we did not have access to search logs of the honey accounts, but only to the content of the emails that were read. To overcome this limitation, we employed Term Frequency–Inverse Document Frequency (TF-IDF). TF-IDF is a product of two metrics, namely Term Frequency (TF) and Inverse Document Frequency (IDF). The idea is that we can infer the words that cybercriminals searched for, by comparing the important words in the emails read by cybercriminals to the important words in all emails in the decoy accounts. TF-IDF is used to rank words in a corpus, by importance. As a result we relied on TF-IDF to infer the words that cybercriminals searched for in the honey accounts.

In its simplest form, TF is a measure of how frequently term t is found in document d, as shown in Equation 1. IDF is a logarithmic scaling of the fraction of the number of documents containing term t, as shown in Equation 2 where D is the set of all documents in the corpus, N is the total number of documents in the corpus, $|d \in D : t \in d|$ is the number of documents in D, that contain term t. Once TF and IDF are obtained, TF-IDF is computed by multiplying TF and IDF, as shown in Equation 3.

$$tf(t,d) = f_{t,d} \tag{1}$$

$$idf(t,D) = log\frac{N}{|d \in D : t \in d|} \tag{2}$$

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) \tag{3}$$

The output of TF-IDF is a weighted metric that ranges between 0 and 1. The closer the weighted value is to 1, the more important the term is, in the corpus.

Searched words	$tfidf_R$	$tfidf_A$	$tfidf_R - tfidf_A$	Common words	$tfidf_R$	$tfidf_A$	$tfidf_R - tfidf_A$
results	0.2250	0.0127	0.2122	transfer	0.2795	0.2949	-0.0154
bitcoin	0.1904	0.0	0.1904	please	0.2116	0.2608	-0.0493
family	0.1624	0.0200	0.1423	original	0.1387	0.1540	-0.0154
seller	0.1333	0.0037	0.1296	company	0.042	0.1531	-0.1111
localbitcoins	0.1009	0.0	0.1009	would	0.0864	0.1493	-0.063
account	0.1114	0.0247	0.0866	energy	0.0618	0.1471	-0.0853
payment	0.0982	0.0157	0.0824	information	0.0985	0.1308	-0.0323
bitcoins	0.0768	0.0	0.07684	about	0.1342	0.1226	0.0116
below	0.1236	0.0496	0.074	email	0.1402	0.1196	0.0207
listed	0.0858	0.02068	0.0651	power	0.0462	0.1175	-0.0713

Table 2: List of top 10 words by $tfidf_R - tfidf_A$ (on the left) and list of top 10 words by $tfidf_A$ (on the right). The words on the left are the ones that have the highest difference in importance between the emails read by attackers and the emails in the entire corpus. For this reason, they are the words that attackers most likely searched for when looking for sensitive information in the stolen accounts. The words on the right, on the other hand, are the ones that have the highest importance in the entire corpus.

We evaluated TF-IDF on all terms in a corpus of text comprising two documents, namely, all emails d_A in the honey accounts, and all emails d_R read by the attackers. The idea is that the words that have a large importance in the emails that have been read by a criminal, but have a lower importance in the overall dataset, are likely to be keywords that the attackers searched for on the Gmail site. We preprocessed the corpus by filtering out all words that have less than 5 characters, and removing all known header-related words, for instance "delivered" and "charset," honey email handles, and also removing signaling information that our monitoring infrastructure introduced into the emails. After running TF-IDF on all remaining terms in the corpus, we obtained their TF-IDF values as vectors $TFIDF_A$ and $TFIDF_R$, the TF-IDF values of all terms in the corpus $[d_A, d_R]$. We proceeded to compute the vector $TFIDF_R - TFIDF_A$. The top 10 words by $TFIDF_R - TFIDF_A$, compared to the top 10 words by $TFIDF_A$, is presented in Table 2. The idea is that words that have $TFIDF_R$ values that are higher than $TFIDF_A$ will rank higher in the list, and those are the words that the cybercriminals likely searched for.

As seen in Table 2, the top 10 important words by $TFIDF_R - TFIDF_A$ are sensitive words, such as "bitcoin," "family," and "payment." These top 10 words constitute words with the greatest difference importance between the emails read by attackers, and all emails in the corpus. Comparing these words with the most important words in the entire corpus reveals the indication that attackers likely searched for sensitive information, especially financial information. In addition, words with the highest importance in the entire corpus (for example, "company" and "energy"), shown in the right side of Table 2, have much lower importance in the emails read by cybercriminals, and most of them have negative values in $TFIDF_R - TFIDF_A$. This is a strong indicator that the honey accounts were not searched for random terms, but were searched for sensitive information.

Originally, the Enron dataset had no "bitcoin" term. However, that term was introduced into the read emails document d_R, through the actions of one of the cybercriminals that accessed some of the honey accounts. The cybercriminal attempted to send blackmail messages from some of our honey accounts to Ashley Madison scandal victims, requesting ransoms in bitcoin, in exchange for silence. In the process, a lot of draft emails containing information about 'bitcoin' were created and abandoned by the cybercriminal, and other cybercriminals read them during later accesses. That way, our monitoring infrastructure picked up 'bitcoin' related terms, and they rank high in Table 2, showing that cybercriminals showed a lot of interest in those emails.

4.7 Interesting case studies

In this section, we present some interesting case studies we encountered during our experiments. They help to shed further light into actions that cybercriminals take on compromised webmail accounts.

Three of the honey accounts were used by an attacker to send multiple blackmail messages to some victims of the Ashley Madison scandal. The blackmailer threatened to expose the victims, unless they made some payments in bitcoin to a specified bitcoin wallet. Tutorials on how to make bitcoin payments were also included in the messages. The blackmailer also created and abandoned many drafts emails targeted at more Ashley Madison victims, which some other visitors to the accounts read, thus contributing to read content that our monitoring infrastructure recorded.

Two of the honey accounts received notification emails about the hidden Google Apps Script in both honey accounts "using too much computer time." The notifications were read by an attacker, and we received notifications about the reading actions.

Finally, an attacker registered on an carding forum using one of the honey accounts as registration email address. As a result, registration confirmation information was sent to the honey account This shows that some of the accounts were used as stepping stones by cybercriminals to perform other attacks.

4.8 Sophistication of attackers

From the accesses we recorded in the honey accounts, we identified 3 peculiar behaviors of cybercriminals that indicate their level of sophistication, namely, configuration hiding - for instance by hiding user agent information, detection evading - by connecting from locations close to the advertised decoy location if provided, and stealth - avoiding actions like hijacking and spamming. Attackers accessing the different groups of honey accounts exhibit different types of sophistication. Those accessing accounts leaked through malware are more stealthy than others - they don't hijack the accounts, and they don't send spam from the accounts. They also access the accounts through Tor, and they hide their system configuration, for instance, there was no user agent string information in the accesses we recorded from them. Attackers accessing accounts leaked on paste sites tend to connect from locations closer to the ones specified as decoy locations in the leaked account. They do this in a bid to evade detection. Attackers accessing accounts leaked in underground forums don't make much attempts to stay stealthy or to connect from closer locations. These differences in sophistication could be used to characterize attacker behavior.

5. DISCUSSION

In this section, we discuss the implications of the findings we made in this paper. First, we talk about what our findings mean for current mitigation techniques against compromised online service accounts, and how they could be used to devise better defenses. Then, we talk about some limitations of our method. Finally, we present some ideas for future work.

Implications of our findings. In this paper, we made multiple findings that provide the research community with a better understanding of what happens when online accounts get compromised. In particular, we discovered that if attackers are provided location information about the online accounts they then tend to connect from places that are closer to that advertised location. We believe that this is an attempt to evade current security mechanisms employed by online services to discover suspicious logins. Such systems often rely on the origin of logins, to assess how suspicious those login attempts are. Our findings cast some doubts on the effectiveness of such suspicious login anomaly detection systems, especially in the presence of skilled attackers. We also observed that many accesses were received through proxies and Tor exit nodes, so it is hard to determine the exact origins of logins. This problem renders these security mechanisms less effective.

Despite confirming existing evasion techniques in use by cybercriminals, our experiments also highlighted interesting behaviors that could be used to develop effective systems to detect malicious activity. For example, our observations about the words searched for by the cybercriminals show that behavioral modeling could work in identifying anomalous behavior in online accounts. Anomaly detection systems could be trained adaptively on words being searched for over a period of time, by the legitimate account owner. A deviation of searches from those words would then be flagged as anomalous, indicating that the account may have been compromised. Similarly, anomaly detection systems could be trained on durations of connections during benign usage, and deviations from those could be flagged as anomalous.

Limitations. We encountered a number of limitations in the course of the experiments. For example, we were able to leak the honey accounts only on a few outlets, namely paste sites, underground forums and malware. In particular, we could only target underground forums that were open to the public and for which registration was free. Similarly, we could not study some of the most recent families of information-stealing malware such as Dridex, because they would not execute in our virtual environment.

Attackers could find the scripts we hid in the honey accounts and remove them, making it impossible for us to monitor the activity of the account. This is an intrinsic limitation of our monitoring architecture, but in principle studies similar to ours could be performed by the online service providers themselves, such as Google and Facebook. By having access to the full logs of their systems, such entities would have no need of setting up monitoring scripts, and it would be impossible for attackers to evade their scrutiny.

Future work. In the future, we plan to continue exploring the ecosystem of stolen accounts, and gaining a better understanding of the underground economy surrounding them. We would explore ways to make the decoy accounts more believable, in order to attract more cybercriminals and keep them engaged with the decoy accounts. We intend to set up additional scenarios, such as studying attackers who have a specific motivation, for example compromising accounts that belong to political activists (rather then generic corporate accounts, as we did in this paper). We want to study the modus operandi of cybercriminals taking over other types of accounts, such as Online Social Networks (OSNs) and cloud storage accounts. We could modify and adapt the monitoring infrastructure that we already have to other types of accounts. We also plan to devise a wider taxonomy of attackers in the future.

6. RELATED WORK

In this section, we briefly compare this paper with previous work, noting that most previous work focused on spam and social spam. Only a few focused on manual hijacking of accounts and their activity.

Bursztein et al. [11] investigated manual hijacking of online accounts, through phishing pages. The study focuses on cybercriminals who steal user credentials and use them privately, and shows that manual hijacking is not as common as automated hijacking by botnets. The study illustrates the usefulness of honey credentials (account honeypots), in the study of hijacked accounts. Compared to the work by Bursztein et al., in this paper we analyze a much broader threat model, looking at account credentials automatically stolen by malware, as well as the behavior of cybercriminals who obtain account credentials through underground forums and paste sites. By focusing on multiple types of miscreants, we were able to show differences in their modus operandi, and provide multiple insights on the activities that happen on hijacked Gmail accounts in the wild. In addition, we provide an open source framework that can be used by other researchers to set up experiments similar to ours and further explore the ecosystem of stolen Google accounts. Despite the fact that the authors of [11] had more visibility on the account hijacking phenomenon than we did, since they were operating the Gmail service, the dataset that we collected is of comparable size to theirs: we logged 327 malicious accesses to 100 accounts, while they studied 575 high-confidence hijacked accounts.

Thomas et al. [28] studied Twitter accounts under the control of spammers. Stringhini et al. [25] studied social spam using 300 honeypot profiles, and presented a tool for detection of spam on Facebook and Twitter. Similar work was also carried out in [8, 10, 20, 32]. Egele et al. [14] presented a system that detects malicious activity in online social networks by building statistical models of "normal" behavior patterns of users. Stringhini et al. [26] developed a tool for spearphishing detection based on behavioral modeling of senders.

Stone-Gross et al. [24] studied a large-scale spam operation by analyzing 16 C&C servers of *Pushdo/Cutwail* botnet. In the paper, the authors highlight that the Cutwail botnet, one of the largest of its time, has the capability of connecting to webmail accounts to send spam. This capability seemed not to be used by cybercriminals at the time of the study. During our measurement we did not observe any mass-sending email activity that might make us think our honey accounts were used by a large botnet to automatically send spam. In their paper, Stone-Gross et al. also describe the activity of cybercriminals on `spamdot`, a large underground forum. They show that cybercriminals were actively trading account information such as the one provided in this paper, providing free "teasers" of the overall datasets for sale. In this paper, we used a similar approach to leak account credentials on underground forums.

Thomas et al. [29] studied underground markets in which fake Twitter accounts are sold and then used to spread spam and other malicious content. Unlike this paper, they focus on fake accounts and not on legitimate ones that have been hijacked. Wang et al. [30] proposed the use of patterns of click events to spot fake accounts in online services, by building clickstream models of real users and fake accounts. In 2012, Liu et al. [21] studied content privacy issues in Peer-to-Peer (P2P) networks, by deploying honeyfiles containing honey account credentials in P2P shared spaces. They monitored downloading events and concluded that attackers that downloaded the honeyfiles had malicious intent, to make economic gain from the private data they obtained. The study used a similar approach to ours, especially in the placement of honey account credentials. However, they placed more emphasis on honeyfiles than honey credentials. Besides, they studied P2P networks while our work focuses on compromised accounts in the World Wide Web (WWW). Kapravelos et al. [18] studied malicious browser extensions, and emphasized the huge risks that malicious browser extensions pose to users. They configured dynamic web pages as honeypots, to study the behavior of browser extensions. On the other hand, Wang et al. [31] investigated malicious web pages by deploying VM-based honeypots that ran on vulnerable operating systems, and patrolled the World Wide Web to find malicious web pages.

7. CONCLUSION

In this paper, we presented a honey account system able to monitor the activity of cybercriminals that gain access to Gmail credentials. We will open source our system, to encourage researchers to set up additional experiments and improve the knowledge of our community regarding what happens after an account is compromised. We set up 100 honey accounts, and leaked them on paste sites, underground forums, and virtual machines infected with malware. We measured the accesses received on our honey accounts for a period of 7 months, and provided detailed statistics of the activity of cybercriminals on these accounts, together with a taxonomy of such criminals. Our findings help the research community to get a better understanding of the ecosystem of stolen online accounts, and will help researchers develop better detection systems against this malicious activity.

8. REFERENCES

[1] Apps Script. https://developers.google.com/apps-script/?hl=en.

[2] Dropbox User Credentials Stolen: A Reminder To Increase Awareness In House. http://www.symantec.com/connect/blogs/dropbox-user-credentials-stolen-reminder-\increase-awareness-house.

[3] Overview of Google Apps Script. https://developers.google.com/apps-script/overview.

[4] Pastebin. pastebin.com.

[5] The Target Breach, By the Numbers. http://krebsonsecurity.com/2014/05/the-target-breach-by-the-numbers/.

[6] S. Afroz, A. C. Islam, A. Stolerman, R. Greenstadt, and D. McCoy. Doppelgänger finder: Taking stylometry to the underground. In *IEEE Symposium on Security and Privacy*, 2014.

[7] T. Anderson and D. Darling. Asymptotic theory of certain 'goodness of fit' criteria based on stochastic processes. *Annals of Mathematical Statistics*, 1952.

[8] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting Spammers on Twitter. In *Conference on Email and Anti-Spam (CEAS)*, 2010.

[9] H. Binsalleeh, T. Ormerod, A. Boukhtouta, P. Sinha, A. Youssef, M. Debbabi, and L. Wang. On the analysis of the zeus botnet crimeware toolkit. In *Privacy Security and Trust (PST)*, 2010.

[10] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *Proceedings of the 27th Annual Computer Security Applications Conference*, 2011.

[11] E. Bursztein, B. Benko, D. Margolis, T. Pietraszek, A. Archer, A. Aquino, A. Pitsillidis, and S. Savage. Handcrafted fraud and extortion: Manual account hijacking in the wild. In *ACM SIGCOMM Conference on Internet Measurement*, 2014.

[12] H. Cramèr. On the composition of elementary errors. *Skandinavisk Aktuarie- tidskrift*, 1928.

[13] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *SIGCHI conference on Human Factors in computing systems*, 2006.

[14] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Compa: Detecting compromised accounts on social networks. In *Symposium on Network and Distributed System Security (NDSS)*, 2013.

[15] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Towards detecting compromised accounts on social networks. In *Transactions on Dependable and Secure Systems (TDSC)*, 2015.

[16] T. Jagatic, N. Johnson, M. Jakobsson, and T. Jagatif. Social Phishing. *Communications of the ACM*, 2007.

[17] J. P. John, A. Moshchuk, S. D. Gribble, and A. Krishnamurthy. Studying Spamming Botnets Using Botlab. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[18] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson. Hulk: Eliciting malicious behavior in browser extensions. In *USENIX Security Symposium*, 2014.

[19] B. Klimt and Y. Yang. Introducing the Enron Corpus. In *Conference on Email and Anti-Spam (CEAS)*, 2004.

[20] K. Lee, J. Caverlee, and S. Webb. The social honeypot project: protecting online communities from spammers. In *Proceedings of the 19th international conference on World wide web*, 2010.

[21] B. Liu, Z. Liu, J. Zhang, T. Wei, and W. Zou. How many eyes are spying on your shared folders? In *ACM workshop on Privacy in the electronic society*, 2012.

[22] C. Rossow, C. J. Dietrich, C. Grier, C. Kreibich, V. Paxson, N. Pohlmann, H. Bos, and M. van Steen. Prudent practices for designing malware experiments: Status quo and outlook. In *IEEE Symposium on Security and Privacy*, 2012.

[23] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[24] B. Stone-Gross, T. Holz, G. Stringhini, and G. Vigna. The underground economy of spam: A botmasters perspective of coordinating large-scale spam campaigns. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2011.

[25] G. Stringhini, C. Kruegel, and G. Vigna. Detecting Spammers on Social Networks. In *Annual Computer Security Applications Conference (ACSAC)*, 2010.

[26] G. Stringhini and O. Thonnard. That aint you: Blocking spearphishing through behavioral modelling. In *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2015.

[27] B. Taylor. Sender reputation in a large webmail service. In *Conference on Email and Anti-Spam (CEAS)*, 2006.

[28] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011.

[29] Thomas, K. and McCoy, D. and Grier, C. and Kolcz, A. and Paxson, V. Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse. In *USENIX Security Symposium*, 2013.

[30] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao. You are how you click: Clickstream analysis for sybil detection. *USENIX Security Symposium*, 2013.

[31] Y.-M. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King. Automated web patrol with strider honeymonkeys. In *Symposium on Network and Distributed System Security (NDSS)*, 2006.

[32] S. Webb, J. Caverlee, , and C.Pu. Social Honeypots: Making Friends with a Spammer Near You. In *Conference on Email and Anti-Spam (CEAS)*, 2008.