What Happens After You Are Pwnd: Understanding the Use of Leaked Webmail 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 webmail 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.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences; K.6.5 [**Security and Protection**]: Unauthorized Access

IMC 2016, November 14-16, 2016, Santa Monica, CA, USA
© 2016 ACM. ISBN 978-1-4503-4526-2/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2987443.2987475

Keywords

Cybercrime, Webmail, Underground Economy, Malware

1. INTRODUCTION

The wealth of information that users store in webmail accounts on services such as Gmail, Yahoo! Mail, or Outlook.com, as well as the possibility of misusing them for illicit activities has attracted cybercriminals, who actively engage in compromising such accounts. Miscreants obtain the credentials to victims' online accounts by performing phishing scams [17], by infecting users with information-stealing malware [29], or by compromising large password databases, leveraging the fact that people often use the same password across multiple services [16]. Such credentials can be used by the cybercriminal privately, or can then be 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 [30].

Cybercriminals can use compromised accounts in multiple ways. First, they can use them to send spam [18]. 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 [20]. 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) [33]. 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 [13]. Despite the importance of stolen accounts for the underground economy, there is surprisingly little work on the topic. Bursztein et al. [13] 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. Since their work only focused on one possible way used by criminals to steal user login credentials, it leaves questions unanswered on how general their observations are, compared to credentials acquired through other means. Most importantly, [13] relies on proprietary information from Google, and therefore it is not possible for other researchers to replicate their results or build on top of their work.

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 posted on Twitter by compromised accounts [18, 19].

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 using *Google Apps Script* [1]; by doing so, we were able to monitor any time an email was opened, favorited, sent, or a new draft was 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*.

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 leaked the credentials to such accounts on multiple outlets, modeling the different ways in which cybercriminals share and get access to such credentials. First, we leaked credentials on paste sites, such as `pastebin` [5]. Paste sites are commonly used by cybercriminals to post account credentials after data breaches [2]. We also leaked them to underground forums, which have been shown to be the place where cybercriminals gather to trade stolen commodities such as account credentials [30]. Finally, we logged in to our honey accounts on virtual machines that were previously infected with information stealing malware. By doing this, the credentials will be sent to the cybercriminal behind the malware's command and control infrastructure, and will then be used directly by her or placed on the black market for sale [29]. We know that there are other outlets that attackers use, for instance, phishing and data breaches, but we decided to focus on 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 were 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 allowed us to draw a taxonomy of the different actions performed by criminals on stolen Gmail accounts, and provided 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 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 the accounts from locations that are closer 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. Our findings complement what was reported by previous work in the case of manual account hijacking [13], and show that the modus operandi of miscreants varies considerably depending on how they obtain the credentials to stolen accounts.

In summary, this paper makes the following contributions:

- We developed a system to monitor the activity of Gmail accounts. We 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. To the best of our knowledge, this is the first publicly available Gmail honeypot infrastructure.

- 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

Gmail accounts. 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 over other webmail platforms 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 the 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 option 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 many ways. First, they can perform social engineering-based scams, such as setting up phishing web pages that resemble the login pages of popular online services [17] or sending spearphishing emails pretending to be members of customer support teams at such online services [32]. As a second way of obtaining user credentials, cybercriminals can install malware on victim computers and configure it to report back any account credentials issued by the user to the command and control server of the botnet [29]. As a third way of obtaining access to user credentials, cybercriminals can exploit vulnerabilities in the databases used by online services to store them [6]. User credentials can also be obtained illegitimately through targeted online password guessing techniques [36], often aided by the problem of password reuse across various online services [16]. Finally, cybercriminals can steal user credentials and access tokens by running network sniffers [14] or mounting Man-in-the-Middle [11] attacks against victims.

After stealing account credentials, a cybercriminal can either use them privately for their 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 [13]. In this work, we study a broader threat model in which we mimic cybercriminals leaking credentials on paste sites [5] as well as miscreants advertising them for sale on underground forums [30]. In particular, previous research showed that cybercriminals often offer a small number of account credentials for free to test their "quality" [30]. 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 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.

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 accesses and activity on Gmail accounts. We set up accounts and leaked them through different outlets. In the following sections, we describe our system architecture and experiment setup in detail.

3.1 System overview

Our system comprises two components, namely, *honey accounts* and a *monitor infrastructure*.

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, designed to augment the functionality of Gmail accounts and Google Drive documents, in addition to building web apps [4]. The scripts we embedded in the honey accounts send notifications to a dedicated webmail account under our control whenever an email is opened, sent, or "starred." In addition, the scripts send us copies of all draft emails created in the honey accounts. We also added a "heartbeat message" function, to send us a message once a day from each honey account, to attest that the account was still functional and had not been blocked by Google. In each honey account, we hid the script in a Google Docs spreadsheet. 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 mailserver under our control. All emails sent from the honeypot accounts 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 location information and IP addresses of accesses to webmail accounts. To track those accesses, we set up external scripts to drive a web browser and periodically login into 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 page in each honey account, and dump the pages to disk, for offline parsing. By collecting information from the visitor activity pages, we obtain location and system configuration information of accesses, as provided by Google's geolocation and system configuration fingerprinting system.

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 release the source code of our system[1].

3.2 Experiment setup

As part of our experiments, we first set up a number of honey accounts on Gmail, and then leaked them through 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 [31]. Creating and setting up these accounts is a manual process. Google also rate-limits the creation of new accounts from the same IP address by presenting a phone verification page after a few accounts have been created. These factors imposed limits on the number of honey accounts we could set up in practice.

We populated the freshly-created accounts with emails from the public Enron email dataset [22]. 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 (i.e., the fictitious "owners" of the honey accounts), 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.

In order to have realistic email timestamps, we translated the old Enron email timestamps to recent timestamps slightly earlier than our experiment start date. For instance, given two email timestamps t_1 and t_2 in the Enron dataset such that t_1 is earlier than t_2, we translate them to more recent timestamps T_1 and T_2 such that T_1 is earlier than T_2. We then schedule those particular emails to be sent to the recipient honey accounts at times T_1 and T_2 respectively. We sent between $200 - 300$ emails from the Enron dataset to each honey account in the process of populating them.

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. In addition, we leaked some credentials through malware, since this is a popular way by which professional cybercriminals steal credentials and compromise accounts [10]. We divided the honeypot accounts 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 leaked 10 account credentials on Russian paste websites (p.for-us.nl and paste.org.ru). For the remaining 20 accounts, we leaked username and password pairs along with UK and US location information of the fictitious personas that we associated with the honey accounts. We also included date of birth information of each persona.

Group	Accounts	Outlet of leak
1	30	paste websites (no location)
2	20	paste websites (with location)
3	10	forums (no location)
4	20	forums (with location)
5	20	malware (no location)

Table 1: List of account honeypot groupings.

We leaked 30 account credentials on underground forums. For 10 of them, we only specified username and password pairs, without additional information. In a manner similar to the paste site leaks described earlier, we appended UK and US location information to underground forum leaks, claiming that our fictitious personas lived in those locations. We also included date of birth information for each persona.

To leak credentials, we used these forums: offensivecommunity.net, bestblackhatforums.eu, hackforums.net, and blackhatworld.com. We selected them 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 [30]. 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 [7]. When leaking credentials on underground forums, we mimicked the modus operandi of cybercriminals that was outlined by Stone-Gross et al. in [30]. In the 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 we received on underground forums, mostly inquiring about obtaining the full dataset, but we did not follow up to them.

Finally, to study the activity of criminals obtaining credentials through information-stealing malware in honey accounts, we leaked access credentials of 20 accounts to information-stealing malware samples. To this end, we selected malware samples from the Zeus family, which is one of the most popular malware families performing information stealing [10], 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. Similarly, we provide decoy location

[1] https://bitbucket.org/gianluca_students/gmail-honeypot

information in some leaks, and not in others, with the idea of observing differences in malicious activity depending on the amount and type of information available to cybercriminals. 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. Our malware sandbox system is structured as follows. A web server entity manages the honey credentials (usernames and passwords) and the malware samples. The host machine creates a Virtual Machine (VM), which contacts the web server to request an executable malware file and a honey credential file. The structure is similar to the one explained in [21]. The malware file is then executed in the VM (that is, the VM is infected with malware), after which a script drives a browser in the VM to login to Gmail using the downloaded credentials. The idea is to expose the honey credentials to the malware that is already running in the VM. After some time, the infected VM is deleted and a fresh one is created. This new VM downloads another malware sample and a different honey credential file, and it repeats the infection and login operation. To maximize the efficiency of the configuration, before the experiment we carried out a test without the Gmail login process to select only samples whose C&C servers were still up and running.

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. However, it is necessary to note that the Enron dataset is the only large publicly available email corpus, to the best of our knowledge. To make the emails believable, we changed the names in the emails, dates, and company name. In the future, we will work towards obtaining or generating a better email dataset, if possible. Also, some visitors may notice that the honey accounts did not receive any new emails during the period of observation, and this may affect the way in which criminals interact with the accounts. Another threat is that we only leaked honey credentials through the outlets listed previously (namely paste sites, underground forums, and malware), therefore, our results reflect the activity of participants present on those outlets only. Finally, since we selected only underground forums that are publicly accessible, our observations might not reflect the modus operandi of actors who are active on closed forums that require vetting for signing up.

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 accesses 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 engaged in suspicious activity. It is important to note, however, that our approach does not rely on help from Google to work. Our main reason for enlisting Google's help to disable suspicious login filters was to ensure that all accesses get through to the honey accounts (most accesses would be blocked if Google did not disable the login filters). This does not impact directly on our methodology, and as a result does not reduce the wider applicability of our approach. It is also important to note that Google did not share with us any details on the techniques used internally for the detection of malicious activity on Gmail. Another point of risk is ensuring that the malware in our VMs would not be able to harm third parties. We followed common practices [28] such as restricting the bandwidth available to our virtual machines and sinkholing all email traffic sent by them. Finally, our experiments involve deceiving cybercriminals by providing them fake accounts with fake personal information in them. To ensure that our experiments were run in an ethical fashion, we obtained IRB approval from 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 an overview of our results. We then discuss 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 the 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.

Google records each unique access to a Gmail account and labels the access with a unique cookie identifier. These unique cookie identifiers, along with more information including times of accesses, are included in the visitor activity pages of Gmail accounts. Our scripts extract this data, which we analyze in this section. For the sake of convenience, we will use the terms “cookie” and “unique access” interchangeably in the remainder of this paper.

4.1 Overview

We created, instrumented, and leaked 100 Gmail accounts for our experiments. 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. After this filtering operation, we observed 326 unique accesses to the accounts during the experiments, during which 147 emails were opened, 845 emails were sent, and there were 12 unique draft emails composed by cybercriminals.

In total, 90 accounts received accesses during the experiment, comprising 41 accounts leaked to paste sites, 30 accounts leaked to underground forums, and 19 accounts leaked through malware. 42 accounts were blocked by Google during the course of the experiment, due to suspicious activity. We were able to log activity in those accounts for some time before Google blocked them. 36 accounts were hijacked by cybercriminals, that is, the passwords of such accounts were changed by the cybercriminals. As a result, we lost control of those accounts. We did not observe any attempt by attackers to change the default *send-from* address of our honey accounts. However, assuming that happened and attackers started sending spam messages, Google would block such accounts since we asked them to monitor the accounts with particular attention. A dataset containing the parsed metadata of the accesses received from our honey accounts during our experiments is publicly available at http://dx.doi.org/10.14324/000.ds.1508297

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. Afterwards, they do not perform any additional action. The majority of the observed accesses belong to this category, accounting for 224 accesses. We acknowledge that this large number of curious accesses may be due in part to experienced attackers avoiding interactions with the accounts after logging in, probably after some careful observations indicating that the accounts do not look real. This could potentially introduce some bias into our results.

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 [13]. 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 [30]. We consider accesses to belong to this category if they send any email. We observed 8 accounts of this type that recorded such accesses. This low number of accounts shows that sending spam appears not to be one of the main purposes that cybercriminals use stolen accounts for, when stolen through the outlets that we studied.

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. A change of password prevents us from scraping the visitor activity page, and therefore we are unable to collect further information about the accesses performed to that account.

It is important to note that the taxonomy classes that we described are not exclusive. For example, an attacker might use an account to send spam emails, therefore falling in the “spammer” category, and then change the password of that account, therefore falling into the “hijacker” category. 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 that sent spam through our honey accounts also acted as “hijackers” or as “gold diggers,” searching for sensitive information in the account.

We wanted to understand the distribution of different types of accesses in accounts that were leaked through different means. Figure 1 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 accounts. 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

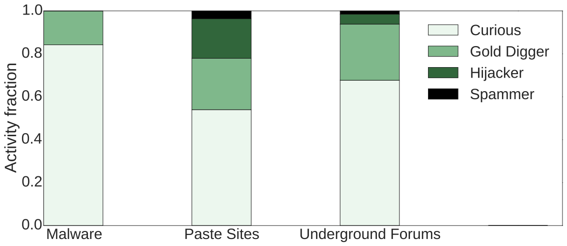


Figure 1: Distribution of types of accesses for different credential leak accesses. As it can be seen, most accesses belong to the "curious" category. It is possible to spot differences in the types of activities for different leak outlets. For example, accounts leaked by malware do not present activity of "hijacker" type. Hijackers, on the other hand, are particularly common among miscreants who obtain stolen credentials through paste sites.

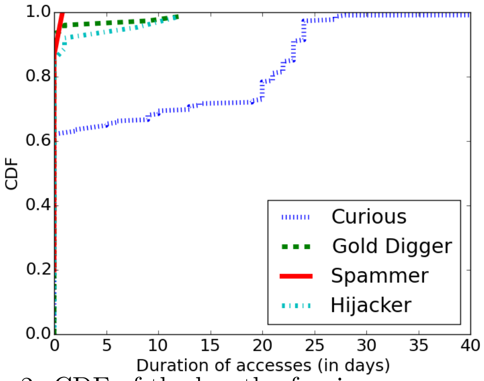


Figure 2: CDF of the length of unique accesses for different types of activity on our honey accounts. The vast majority of unique accesses lasts a few minutes. Spammers tend to use accounts aggressively for a short time and then disconnect. The other types of accesses, and in particular "curious" ones, come back after some time, likely to check for new activity in the honey accounts.

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

In the following, we provide detailed analysis on the unique accesses that we recorded for our honey accounts.

4.3.1 *Duration of accesses*

For each cookie identifier, we recorded the time that the cookie first appeared in a particular honey account as t_0, and the last time it appeared in the honey account as t_{last}. From this information, we computed the duration of activity of each cookie as $t_{last} - t_0$. It is necessary to note that t_{last} of each cookie is a lower bound, since we cease to obtain information about cookies if the

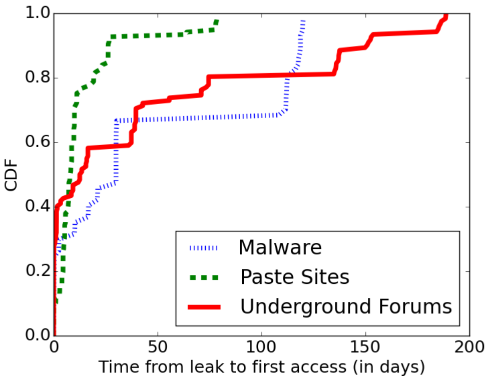


Figure 3: CDF of the time passed between account credentials leaks and the first visit by a cookie. Accounts leaked through paste sites receive on average accesses earlier than accounts leaked through other outlets.

password of the honey account that is recording cookies is changed, for instance. Figure 2 shows the Cumulative Distribution Function (CDF) of the length of unique accesses of different types of attackers. As it can be seen, the vast majority of accesses are very short, lasting only a few minutes and never coming back. "Spammer" accesses, 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. 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 [13], which states that most cybercriminals connect to a compromised webmail account once, to assess its value within a few minutes. However, [13] focused only on accounts compromised via phishing pages, while we look at a broader range of ways in which criminals can obtain such credentials. Our results show that the modes of operation of cybercriminals vary, depending on the outlets they obtain stolen credentials from.

4.3.2 *Time between leak and first access*

We then studied how long it takes after credentials are leaked on different outlets until our infrastructure records accesses from cybercriminals. Figure 3 reports a CDF of the time between leak and first access for accounts leaked through different outlets. As it can be seen, within the first 25 days after leak, we 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 two sharp inflection points.

Figure 4 sheds more light into what happened at those points. The figure reports the unique accesses to each of our honey accounts over time. 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 visited again by the same criminal who operated the malware infrastructure, or that the accounts were sold on the underground market and that another miscreant 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.

In addition, Figure 4 shows that the 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 ten 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 maybe they did not believe that the accounts were real, thus not bothering to access them.

4.3.3 System configuration of accesses

We observed a wide variety of system configurations for the accesses across groups of leaked accounts, by leveraging Google's system fingerprinting information available to us inside the honey accounts. As shown in Figure 5a, accesses to accounts leaked on paste sites were made through a variety of popular browsers, with Firefox and Chrome taking the lead. We also recorded many accesses from unknown browsers. It is possible for an attacker to hide browser information from Google servers by presenting an empty user agent and hiding other fingerprintable information [27]. About 50% of accesses to accounts leaked through paste sites were not identifiable. Chrome and Firefox take the lead in groups leaked in underground forums as well, but there is less variety of browsers there. Interestingly, all accesses to accounts in malware groups were made from unknown browsers. This shows that cybercriminals that accessed groups leaked through malware were stealthier than others. While analyzing the operating systems used by criminals, we observed that honey accounts leaked through malware mostly received accesses from Windows computers, followed by Mac OS X and Linux. This is shown in Figure 5b. In the paste sites and underground forum groups, we observe a wider range of operating systems. More than 50% of computers in the three categories ran on Windows. It is interesting to note that Android devices were also used to connect to the honey accounts in paste sites and underground forum groups.

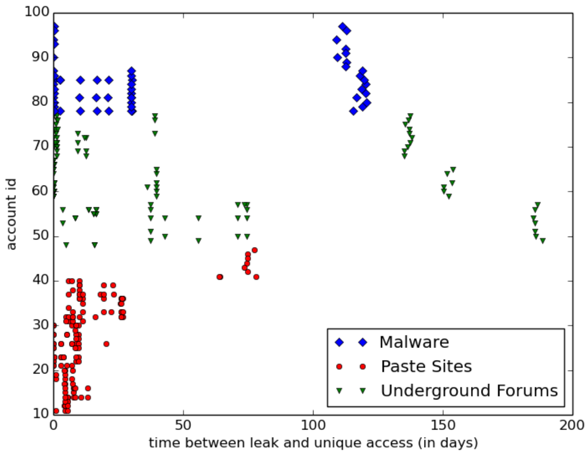
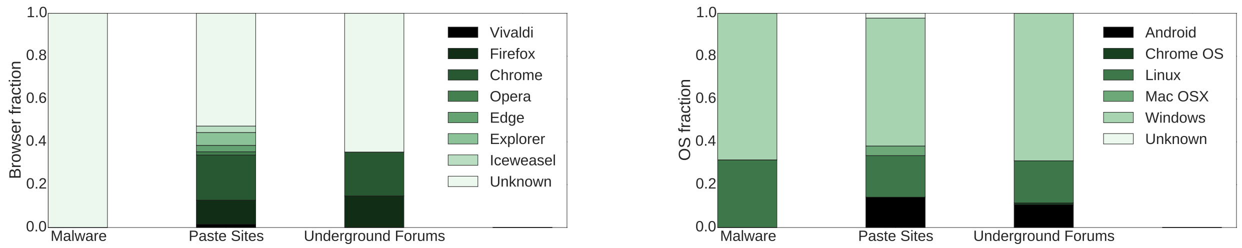


Figure 4: Plot of duration between time of leak and unique accesses in accounts leaked through different outlets. As it can be seen, accounts leaked to malware experience a sudden increase in unique accesses after 30 days and 100 days from the leak, indicating that they may have been sold or transferred to some other party by the cybercriminals behind the malware command and control infrastructure.

The diversity of devices and browsers in the paste sites and underground forums groups indicates a motley mix of cybercriminals with various motives and capabilities, compared to the malware groups that appear to be more homogeneous. It is also obvious that attackers that steal credentials through malware make more efforts to cover their tracks by evading browser fingerprinting.

4.3.4 Location of accesses

We recorded the location information that we found in the accesses that were logged by our infrastructure. Our goal was to understand patterns in the locations (or proxies) used by criminals to access stolen accounts. Out of the 326 accesses logged, 132 were coming from Tor exit nodes. More specifically, 28 accesses to accounts leaked on paste sites were made via Tor, out of a total of 144 accesses to accounts leaked on paste sites. 48 accesses to accounts leaked on forums were made through Tor, out of a total of 125 accesses made to accounts leaked on forums. We observed 57 accesses to accounts leaked through malware, and all except one of those accesses were made via Tor. We removed these accesses for further analysis, since they do not provide information on the physical location of the criminals. After removing Tor nodes, 173 unique accesses presented



(a) Distribution of browsers of honey account accesses

(b) Distribution of operating systems of honey account accesses

Figure 5: Distribution of browsers and operating systems of the accesses that we logged to our honey accounts. As it can be seen, accounts leaked through different outlets attracted cybercriminals with different system configurations.

location information. To determine this location information, we used the geolocation provided by Google on the account activity page of the honey accounts. 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 the 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 were used by cybercriminals to connect to the stolen accounts.

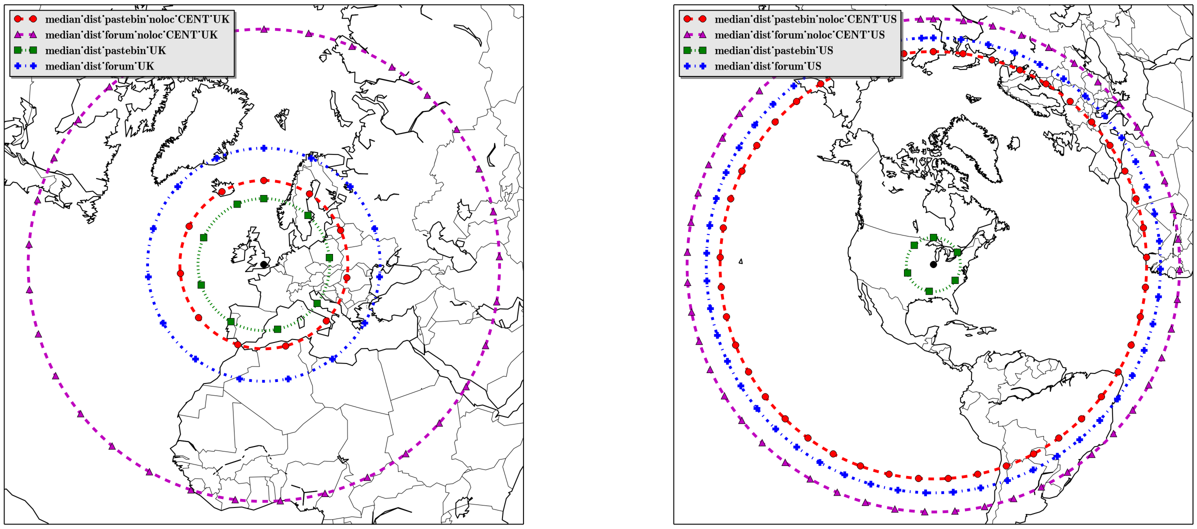
One of our goals was to observe if cybercriminals attempt to evade location-based login risk analysis systems by tweaking access origins. In particular, we wanted to assess whether telling criminals the location where the owner of an account is based influences the locations that they will use to connect to the 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 the malware operators that accessed our accounts 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.

While leaking the honey credentials, we chose London and Pontiac, MI as our decoy UK and US locations respectively. The idea was to claim that the honey accounts leaked with location details belonged to fictitious personas living in either London or Pontiac. However, we realized that leaking multiple accounts with the same location might cause suspicion. As a result, we chose decoy UK and US locations such that London and Pontiac, IL were the midpoints of those locations respectively.

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 decoy locations in our account leaks. For example, for the 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 the $dist\_paste\_UK$ vector as $distance(L_A, mid_{UK})$, where mid_{UK} are 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 6a and 6b.

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 6a shows that connections to accounts leaked on paste sites and forums 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 6a, while the one of accounts leaked in underground forums, along with UK location information, is smaller.

We obtained similar results in the US plot, with some interesting distinctions. As shown in Figure 6b, 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 median circle of accounts leaked on paste sites without location information, which 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 as pronounced. This again supports the indication that


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 6: 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. As it can be seen, account credentials leaked with location information attract logins from hosts that are closer to the advertised midpoint than credentials that are posted without any location information.

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 the studied forums are less sophisticated, or care less than the ones on paste sites.

Statistical significance. As we explained, Figures 6a and 6b show that accesses to leaked accounts happen closer to advertised locations if this information is included in the leak. To confirm the statistical significance of this finding, we performed a Cramer Von Mises test [15]. The Anderson version [8] of this test is used to understand if two vectors of values do 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., that the two vectors of distances have the same distribution), otherwise it is not possible to state with statistical significance that the two distance vectors come from different distributions. The p-value from the test on paste sites vectors (p-values of 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, thus stating that the two vectors come from different distributions while we cannot say the same observing the p-values for the tests on forum vectors (p-values of 0.272883 for the UK case and 0.272011 for 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 6a and 6b indicate that there are some location differences in this case too.

4.3.5 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, and 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. 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 opened. To overcome this limitation, we employed Term Frequency–Inverse Document Frequency (TF-IDF). TF-IDF is used to rank words in a corpus by importance. As a result we re-

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	0.0420	0.1531	-0.1111
localbitcoins	0.1009	0.0	0.1009	would	0.0864	0.1493	-0.0630
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	0.0768	about	0.1342	0.1226	0.0116
below	0.1236	0.0496	0.0740	email	0.1402	0.1196	0.0207
listed	0.0858	0.0207	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 opened 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.

lied on TF-IDF to infer the words that cybercriminals searched for in the honey accounts. 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 opened by cybercriminals to the important words in all emails in the decoy 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.

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 opened by the attackers. The intuition is that the words that have a large importance in the emails that have been opened by a criminal, but have a lower importance in the overall dataset, are likely to be keywords that the attackers searched for in the Gmail account. 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 $TFIDF_A$ are presented in Table 2. 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.” 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 opened by cybercriminals, and most of them have negative values in $TFIDF_R - TFIDF_A$. This is a strong indicator that the emails opened in the honey accounts were not opened at random, but were the result of searches for sensitive information.

Originally, the Enron dataset had no “bitcoin” term. However, that term was introduced into the opened 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 [3], requesting ransoms in bitcoin, in exchange for silence. In the process, many draft emails containing information about “bitcoin” were created and abandoned by the cybercriminal, and other cybercriminals opened 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.

4.4 Interesting case studies

In this section, we present some interesting case studies that 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 created and abandoned many drafts emails targeted at more Ashley Madison victims, which as we have already mentioned some other visitors to the accounts opened, thus contributing to the opened emails 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 opened by an attacker, and we received notifications about the opening 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 further illicit activity.

4.5 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, location filter evading – by connecting from locations close to the advertised decoy location if provided, and stealthiness – avoiding performing clearly malicious actions such as hijacking and spamming. Attackers accessing the different groups of honey accounts exhibit different types of sophistication. Those accessing accounts leaked through malware are stealthier than others – they don't hijack the accounts, and they don't send spam from them. They also access the accounts through Tor, and they hide their system configuration, for instance, their web browser is not fingerprintable by Google. 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 do not make significant attempts to stay stealthy or to connect from closer locations. These differences in sophistication could be used to characterize attacker behavior in future work.

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 with location information about the online accounts, they then tend to connect from places that are closer to those advertised locations. 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 show that there is an arms race going on, with attackers attempting to actively evade the location-based anomaly detection systems employed by Google. We also observed that many accesses were received through Tor exit nodes, so it is hard to determine the exact origins of logins. This problem shows the importance of defense in depth in protecting online systems, in which multiple detection systems are employed at the same time to identify and block miscreants.

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 by the legitimate account owner over a period of time. A deviation of search behavior would then be flagged as anomalous, indicating that the account may have been compromised. Similarly, anomaly detection systems could be trained on the 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 accounts. 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 to set up monitoring scripts, and it would be impossible for attackers to evade their scrutiny. Finally, while evaluating what cybercriminals were looking for in the honey accounts, we were able to observe the emails that they found interesting in the honey accounts, not everything they searched for. This is because we do not have access to the search logs of the honey accounts.

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, 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 than generic corporate accounts, as we did in this paper). We would also like to study if demographic information, as well as the language that the emails in honey accounts are written in, influence the way in which cybercriminals interact with these accounts. To mitigate the fact that our infrastructure can only identify search terms for emails that were found in the accounts, we plan to seed the honey accounts with some specially crafted emails containing decoy sensitive information, for instance, fake bank account information and login credentials, along with other regular email messages. Hopefully, this type of specialized email seeding will help to increase the variety of hits when cybercriminals search for content in the honey accounts, by improving the seeding of the honey accounts. We believe this will improve our insight into what criminals search for.

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. [13] investigated manual hijacking of online accounts through phishing pages. The study focuses on cybercriminals that steal user credentials and use them privately, and shows that manual hijacking is not as common as automated hijacking by botnets. This paper illustrates the usefulness of honey credentials (account honeypots), in the study of hijacked accounts. Compared to the work by Bursztein et al., which focused on phishing, we analyzed a much broader threat model, looking at account credentials automatically stolen by malware, as well as the behavior of cybercriminals that 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. We also 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. To the best of our knowledge, our infrastructure is the first publicly available Gmail honeypot infrastructure. Despite the fact that the authors of [13] 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 326 malicious accesses to 100 accounts, while they studied 575 high-confidence hijacked accounts.

A number of papers looked at abuse of accounts on social networks. Thomas et al. [34] studied Twitter accounts under the control of spammers. Stringhini et al. [31] 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 [9,12,24,38]. Thomas et al. [35] 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. [37] proposed the use of patterns of click events to find fake accounts in online services.

Researchers also looked at developing systems to detect compromised accounts. Egele et al. [18] presented a system that detects malicious activity in online social networks using statistical models. Stringhini et al. [32] developed a tool for detecting compromised email accounts based on the behavioral modeling of senders. Other papers investigated the use of stolen credentials and stolen files by setting up honeyfiles. Liu et al. [25] deployed honeyfiles containing honey account credentials in P2P shared spaces. The study used a similar approach to ours, especially in the placement of honey account credentials. However, they placed more emphasis on honeyfiles than on honey credentials. Besides, they studied P2P networks while our work focuses on compromised accounts in webmail services. Nikiforakis et al. [26] studied privacy leaks in file hosting services by deploying honeyfiles on them. In our previous work [23], we developed an infrastructure to study malicious activity in online spreadsheets, using an approach similar to the one described in this paper. Stone-Gross et al. [30] 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. 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.

7. CONCLUSION

In this paper, we presented a honey account system able to monitor the activity of cybercriminals that gain access to Gmail account credentials. Our system is publicly available to encourage researchers to set up additional experiments and improve the knowledge of our community regarding what happens after webmail accounts are compromised[2]. We leaked 100 honey accounts on paste sites, underground forums, and virtual machines infected with malware, and provided detailed statistics of the activity of cybercriminals on the accounts, together with a taxonomy of the criminals. Our findings help the research community to get a better understanding of the ecosystem of stolen online accounts, and potentially help researchers to develop better detection systems against this malicious activity.

8. ACKNOWLEDGMENTS

We wish to thank our shepherd Andreas Haeberlen for his advice on how to improve our paper, and Mark Risher and Tejaswi Nadahalli from Google for their support throughout the project. We also thank the anonymous reviewers for their comments. This work was supported by the EPSRC under grant EP/N008448/1, and by a Google Faculty Award. Jeremiah Onaolapo was supported by the Petroleum Technology Development Fund (PTDF), Nigeria, while Enrico Mariconti was funded by the EPSRC under grant 1490017.

9. 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] Hackers Finally Post Stolen Ashley Madison Data. https://www.wired.com/2015/08/happened-hackers-posted-stolen-ashley-madison-data/.

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

[5] Pastebin. pastebin.com.

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

[7] 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.

[8] T. W. Anderson and D. A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 1952.

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

[10] 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.

[11] D. Boneh, S. Inguva, and I. Baker. SSL MITM Proxy. *http://crypto.stanford.edu/ssl-mitm*, 2007.

[12] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. The socialbot network: when bots socialize for fame and money. In *Annual Computer Security Applications Conference (ACSAC)*, 2011.

[13] 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 Internet Measurement Conference (IMC)*, 2014.

[14] E. Butler. Firesheep. *http://codebutler.com/firesheep*, 2010.

[15] H. Cramèr. On the composition of elementary errors. *Skandinavisk Aktuarietidskrift*, 1928.

[16] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang. The Tangled Web of Password Reuse. In *Symposium on Network and Distributed System Security (NDSS)*, 2014.

[17] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2006.

[18] 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.

[19] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna. Towards Detecting Compromised Accounts on Social Networks. In *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 2015.

[20] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social Phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[21] 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.

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

[2] https://bitbucket.org/gianluca_students/gmail-honeypot

[23] M. Lazarov, J. Onaolapo, and G. Stringhini. Honey Sheets: What Happens to Leaked Google Spreadsheets? In *USENIX Workshop on Cyber Security Experimentation and Test (CSET)*, 2016.

[24] K. Lee, J. Caverlee, and S. Webb. The social honeypot project: protecting online communities from spammers. In *World Wide Web Conference (WWW)*, 2010.

[25] 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 (WPES)*, 2012.

[26] N. Nikiforakis, M. Balduzzi, S. Van Acker, W. Joosen, and D. Balzarotti. Exposing the Lack of Privacy in File Hosting Services. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2011.

[27] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *IEEE Symposium on Security and Privacy*, 2013.

[28] 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.

[29] 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.

[30] B. Stone-Gross, T. Holz, G. Stringhini, and G. Vigna. The underground economy of spam: A botmaster's perspective of coordinating large-scale spam campaigns. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2011.

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

[32] G. Stringhini and O. Thonnard. That Ain't You: Blocking Spearphishing Through Behavioral Modelling. In *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 2015.

[33] B. Taylor. Sender Reputation in a Large Webmail Service. In *Conference on Email and Anti-Spam (CEAS)*, 2006.

[34] K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of Twitter spam. In *ACM Internet Measurement Conference (IMC)*, 2011.

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

[36] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang. Targeted Online Password Guessing: An Underestimated Threat. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

[37] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao. You are How You Click: Clickstream Analysis for Sybil Detection. In *USENIX Security Symposium*, 2013.

[38] 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.