Harvesting Wild Honey from Webmail Beehives

Jeremiah Onaolapo, Enrico Mariconti, and Gianluca Stringhini

Department of Computer Science, University College London, UK
{j.onaolapo,e.mariconti,g.stringhini}@cs.ucl.ac.uk

Abstract. Cybercriminals steal access credentials to online accounts in a bid to derive profit from the valuable content of such accounts. The research community lacks a comprehensive understanding of what these stolen accounts are used for. This is largely because it is hard for researchers to collect data on compromised online accounts. To bridge this gap, we present an infrastructure that is able to monitor accesses and activity of cybercriminals in compromised Gmail accounts. We leaked credentials to 100 accounts under our control through information-stealing malware, public paste sites, and underground forums. We then monitored accesses and activity observed in these accounts over a period of 7 months.

Keywords: honeypots·honey accounts·webmail·information theft

1 Introduction

Malicious online activity has increased rapidly in recent times. Victims suffer massive harm, and cybercriminals make a lot of profit. Malicious activity perpetrated by cybercriminals includes spamming [8, 10, 17], malware dissemination [6], phishing [7, 10], information theft [14], account hijacking [5, 8], and Denial-of-Service (DoS) attacks. The massive scale of cybercrime nowadays is supported by underground ecosystems of interconnected merchants trading fake and compromised accounts, botmasters in control of massive botnet infrastructure, malware developers and distributors, and more [16, 18]. There is limited literature on these underground ecosystems. A direct implication of this limited understanding is that "indirect and defense costs" of cybercrime are at least ten times the earnings of cybercriminals' earnings, according to a recent study [3]. Hence, a deeper understanding of underground ecosystems is necessary to effectively truncate cybercrime operations.

Cybercriminals steal credentials to online accounts, since such accounts often contain valuable information such as credit card details and bank account information, which they sell in underground markets. As earlier stated, there is limited research literature on this problem [5]. There are two main reasons for this. First, it is hard for researchers to obtain data of compromised accounts from large online service providers, such as *Google* and *Yahoo*. Second, only some parts of a cybercrime operation are observable, for instance, information shared publicly by compromised accounts [8, 9].

To bridge this gap, we present a system that is able to monitor attacker activity in Gmail accounts instrumented with *Google Apps Script* [1]. We call such accounts *honey accounts*, and we created 100 of them. We leaked them through multiple outlets, namely, paste sites, underground forums, and information-stealing malware. The idea was to entice cybercriminals to access the honey accounts. We monitored accesses and actions in the honey accounts for 7 months.

In summary, this paper makes the following contributions: First, we develop an extensible system to monitor malicious activity in Gmail accounts. We will publicly release the source code of our system, to allow other researchers to deploy their own honey accounts. The idea is to deepen the understanding of the security community on malicious activity in online accounts. Second, we deployed 100 honey accounts on Gmail, and leaked their credentials through multiple outlets, as earlier described. We collected data for 7 months.

2 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. Figure 1 shows an overview of our monitoring system. We 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.

2.1 System overview

Our system comprises a number of components, namely, honey accounts and monitoring infrastructure. These components are described in this section and a system overview diagram is shown in Figure 1.

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 [2]. We included functions that periodically scan the emails in each honey account to report all discovered changes, namely, read, sent, "starred," and draft emails, back to us. We also added a "heartbeat message" function to attest that the account was still functional and had not been blocked by Google. To mitigate spam, we configured the honey accounts to redirect emails sent from them to a sinkhole SMTP server under our control.

Monitoring infrastructure. In addition to Google Apps Scripts in the honey accounts, we set up some scripts on our server, to periodically login to the honey accounts and automatically collect information about accesses. Notifications of

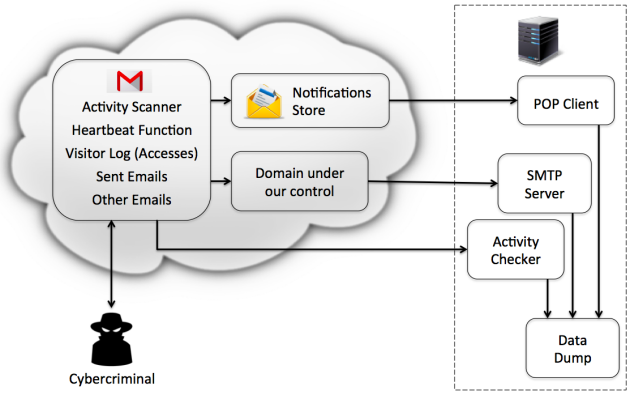

Fig. 1: Overview of our monitoring infrastructure.

actions on honey accounts are sent by Google Apps Scripts we hid in the honey accounts, to a dedicated webmail account that we set up as notifications store. Another script periodically connects to the dedicated webmail account, and retrieves all notifications in it. 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.

2.2 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 [15]. We populated the freshly-created accounts with modified emails from the public Enron email dataset [12], to make them believable.

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 places where to leak 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 [4]. Information-stealing malware collects form data that users input on particular sites, and then uploads it to the C&C server.

Finally, to study activity of information-stealing malware in honey accounts, we leaked some honey credentials to information-stealing malware samples, using the malware honeypot infrastructure described in the following 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.

Malware honeypot infrastructure. We set up a malware sandbox and infected it with malware samples; these samples captured the honey credentials during logins. Our sandbox works as follows:

- Host machine creates Virtual Machine (VM) running on a vulnerable guest operating system.
- VM requests malware and honey credentials from web server running on host machine.
- VM self-infects with malware.
- VM drives web browser and performs login to Gmail using honey credentials.
- Active malware in VM steals honey credentials and sends them to C&C server.
- Host machine deletes VM, and repeats the process with different malware sample and honey credentials.

The structure is similar to the one presented in [11]. To minimize abuse, we followed the practices presented in [13], such as limiting the bandwidth of the network interfaces to avoid DoS attacks, and limiting the lifetime of each VM.

2.3 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. 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 in our virtual environment would not be put in condition to harm third parties. As we said, we followed common practices [13] 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 run in an ethical fashion, we sought and obtained IRB approval by our institution.

3 Preliminary Results and Discussion

In this section, we highlight preliminary results, some limitations of our method, and also present some ideas for future work.

Preliminary results. We ran experiments for 7 months on 100 honey accounts, and observed 327 unique accesses. The accesses originated from 29 countries.
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, 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.

4 Conclusion

In this paper, we presented a honey account system that is 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 multiple outlets. We measured the accesses received on our honey accounts for a period of 7 months.

5 Acknowledgments

We wish to thank anonymous reviewers. This work was supported by the EPSRC under grant EP/N008448/1 and by a Google Research Award. Jeremiah Onaolapo was funded by the Petroleum Technology Development Fund (PTDF), Nigeria. Enrico Mariconti was supported by the EPSRC under grant 1490017.

References

1. Apps Script. https://developers.google.com/apps-script/?hl=en
2. Overview of Google Apps Script. https://developers.google.com/apps-script/overview
3. Anderson, R., Barton, C., Böhme, R., Clayton, R., Van Eeten, M.J., Levi, M., Moore, T., Savage, S.: Measuring the cost of cybercrime. In: The economics of information security and privacy, pp. 265–300. Springer (2013)
4. Binsalleeh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M., Wang, L.: On the analysis of the zeus botnet crimeware toolkit. In: Privacy Security and Trust (PST) (2010)
5. Bursztein, E., Benko, B., Margolis, D., Pietraszek, T., Archer, A., Aquino, A., Pitsillidis, A., Savage, S.: Handcrafted fraud and extortion: Manual account hijacking in the wild. In: ACM SIGCOMM Conference on Internet Measurement (2014)
6. Caballero, J., Grier, C., Kreibich, C., Paxson, V.: Measuring Pay-per-Install: The Commoditization of Malware Distribution. In: USENIX Security Symposium (2011)
7. Dhamija, R., Tygar, J.D., Hearst, M.: Why phishing works. In: SIGCHI conference on Human Factors in computing systems (2006)
8. Egele, M., Stringhini, G., Kruegel, C., Vigna, G.: Compa: Detecting compromised accounts on social networks. In: Symposium on Network and Distributed System Security (NDSS) (2013)
9. Egele, M., Stringhini, G., Kruegel, C., Vigna, G.: Towards detecting compromised accounts on social networks. In: Transactions on Dependable and Secure Systems (TDSC) (2015)
10. Jagatic, T., Johnson, N., Jakobsson, M., Jagatif, T.: Social Phishing. Communications of the ACM (2007)
11. John, J.P., Moshchuk, A., Gribble, S.D., Krishnamurthy, A.: Studying Spamming Botnets Using Botlab. In: USENIX Symposium on Networked Systems Design and Implementation (NSDI) (2009)
12. Klimt, B., Yang, Y.: Introducing the Enron Corpus. In: Conference on Email and Anti-Spam (CEAS) (2004)
13. Rossow, C., Dietrich, C.J., Grier, C., Kreibich, C., Paxson, V., Pohlmann, N., Bos, H., van Steen, M.: Prudent practices for designing malware experiments: Status quo and outlook. In: IEEE Symposium on Security and Privacy (2012)
14. Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., Kruegel, C., Vigna, G.: Your Botnet is My Botnet: Analysis of a Botnet Takeover. In: ACM Conference on Computer and Communications Security (CCS) (2009)
15. Stringhini, G., Kruegel, C., Vigna, G.: Detecting Spammers on Social Networks. In: Annual Computer Security Applications Conference (ACSAC) (2010)
16. Stringhini, G., Hohlfeld, O., Kruegel, C., Vigna, G.: The harvester, the botmaster, and the spammer: on the relations between the different actors in the spam landscape. In: ACM Symposium on Information, computer and communications security (ASIACCS) (2014)
17. Taylor, B.: Sender reputation in a large webmail service. In: Conference on Email and Anti-Spam (CEAS) (2006)
18. 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)