Long-Term Study of Honeypots in a Public Cloud

Rakshit Agrawal*, Jack W. Stokes†, Lukas Rist‡, Ryan Littlefield†, Xun Fan†, Ken Hollis†,
Zane Coppedge†, Noah Chesterman†, Christian Seifert†
*Camio, San Mateo, CA 94401 USA
†Microsoft, Redmond, WA 98052 USA
‡Vivino, Copenhagen, Denmark

***Abstract*—Public cloud hosting environments offer convenient computation and storage resources for cloud service providers, and these resources are also beneficial for adversaries to host malicious web-based attacks. As a result, cloud-based virtual machines are often attacked. In the paper, we conduct a long-term deployment and analysis of honeypots in a public cloud hosting environment. In particular, we deploy five low-interaction honeypots and one medium-interaction honeypot and measure the attack patterns over eleven months. In our study, we found that the low-interaction honeypots were attacked repeatedly, but the activity on the medium-interaction honeypot was small. We first provide an overview of the attack traffic activity. We then use Latent Dirichlet Allocation (LDA) to discover topics in the log data.**

***Index Terms*—honeypot, low-interaction, medium-interaction, public cloud**

I. Introduction

Public cloud providers offer virtual computational and storage resources which can be dynamically provisioned to rapidly adjust according to increasing loads. In addition, the cloud-based computing environments are designed to easily develop and host web-based cloud services. These services are also attractive to adversaries who seek to compromise legitimate cloud resources to launch their attack, run command and control operations, and mine cryptocurrencies. As a result, cloud-based resources are often attacked after they are deployed.

Previous research [1] has investigated short-term brute force attempts on cloud-based VMs. In this work, we instead seek to understand long-term attack trends on virtual machines (VMs) hosted in a public cloud environment. To this end, we deployed a collection of five low-interaction honeypots and one medium-interaction honeypot in VMs hosted in the Azure public cloud. In order to observe attack patterns due to the adversary's behavior alone, we did not update the honeypots after their initial deployment. Furthermore, we left these honeypots running for eleven months and report the attack patterns observed in the honeypots' activity log files. Our goal from this research is to provide new insights into attack techniques for both public, private and hybrid cloud providers and users of these services.

From the immutable honeypot logs, we conduct two types of analysis. First, we analyze all of the traffic to understand how attackers interact with the honeypots. Second, we investigate the content of the files which were dropped on the honeypots. To this end, we analyze all of the content with latent Dirichlet allocation (LDA). The main contributions of this paper include: 1) we construct both low- and medium-interaction honeypots and deploy them in a public cloud environment over a period of eleven months, and 2) we report on the external activity directed towards these honeypots and the content of the dropped files.

II. Honeypots

A honeypot is a tool used to collect intelligence on adversarial tactics, tools, and procedures. This is generally done by using deception to make the attacker believe they are interacting with and potentially exploiting a real system. The more time the adversary spends on and with the system and the more they interact, the more data is collected. The deployment and development of honeypots often entails a feedback loop where the information gathered is used to improve the system's camouflage, deception, and interaction levels.

Honeypots are classified into low-, medium-, and high-interaction [2] based on their level of system emulation or in the case of high-interaction, using a real system. Low- and medium-interaction honeypots, like emulated SSH, web, or email services, are often preferred due to the simplicity of deployment, management, and their low increase to the security risk. They often only implement a subset of a services features and are vulnerable to fingerprinting. Low- and medium-interaction honeypots are often used to collect quantitative data and allow a more high-level view on the threat landscape. Although they could pose a security risk as they expose a real system and require additional ethical considerations, high-interaction honeypots are considered to produce more high-quality data as they are vulnerable to previously unknown exploits.

III. SYSTEM

To measure the long-term interaction between the attackers with cloud-hosted VMs, we deployed a collection of honeypots (HPs) in the Azure public cloud. Figure 1 shows that five low-interaction honeypots (LIHPs) and one medium-interaction honeypot (MIHP) were deployed in Azure over a period of eleven months. In order to protect the log data generated by the honeypots in case they were compromised, new log entries were quickly rewritten into an isolated and immutable data store in Azure Data Lake storage.

Infrastructure. The honeypots, including both the LIHPs and the MIHP, were hosted as individually isolated docker containers on an Azure Web App service. Each honeypot

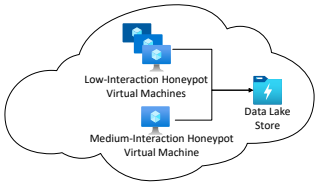


Fig. 1. Honeypot deployment in the Azure Public Cloud.

presented a web frontend resembling regular websites. LIHPs resembled blogs and general textual websites, additionally including hidden text inviting attacks. The MIHP instead looked more like a *Content Management System* or a *Forum*, where users could add content and interact with a database driven system. Furthermore, the MIHP included a mocked back-admin system which could be accessed using common vulnerable paths and credentials.

Logging. The largest source of data collection with the HPs was verbose logging of each interaction with the system. Every request made to the server was logged and analyzed using a family of *handlers*. The logs were broken into daily segments and pushed to Azure Data Lake storage for analysis.

Handlers. The handlers are modules that are deployed to respond to different kinds of attacks. Real-time analysis of the interaction generated responses where the handlers acted in a way the attacker desired within a secure setup and allowing the capture of more content. A common category of handlers involved shell commands including *wget* to fetch the malicious scripts, which were captured and stored for analysis by the handlers. Another handler often used by the system detected the use of the *sqlmap* SQL injection tool and generated responses similar to vulnerabilities.

IV. Evaluation

The evaluation and analysis of the activity and content captured by the honeypots uses the interaction logs on each honeypot along with the data collected by the handlers.

Overall Statistics. *The number of interactions depends on the honeypot. Also, the medium-interaction honeypot did not encounter significant activity.* We analyze this data under two categories including activity and content which are summarized in Table I for each of the deployed honeypots. The *Total Log Sessions* aggregates all of the session activity recorded by each honeypot, while the *Content Upload Sessions* provides the number of sessions where an individual honeypot's handlers were able to successfully store uploaded content files.

Since the vast majority of the overall activity occurred on LIHP-4, we analyze and report the activity on this honeypot in the remainder of this analysis. We investigated the activity for the other honeypots, and they showed similar, but less frequent activity. A few attackers found the medium-interaction honeypot, but they did not upload any content.

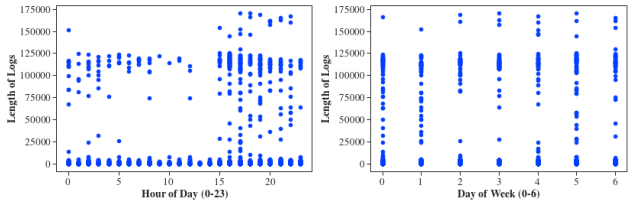
Activity Analysis. We first investigate different aspects of the honeypots' activity patterns and make several observations from this activity.

TABLE I
Honeypot interaction metrics.

Honeypot	Total Log Sessions	Content Upload Sessions
MIHP-1	138	0
LIHP-1	331	1
LIHP-2	186	0
LIHP-3	1,863	28
LIHP-4	31,204	78
LIHP-5	378	0

Hour of the Day: Short reconnaissance activity happens consistently during all hours of the day, but longer attack sessions are more likely to occur during the evening hours at the data center.

Figure 2a depicts the distribution of the length of total number of log entries beginning at each hour in the day. Each entry in the log records a network command received by the honeypot from a distinct IP address. The length is the total number of lines in the logs corresponding to all individual sessions. The hour represents the starting time for the attack and is based upon the local time of the honeypot in its data center. Figure 2a indicates log activity with significantly higher number of lines was specifically observed during late evenings as compared to other hours during the day. Also, attackers visited the honeypots for short sessions with similar frequencies over each hour in the day.



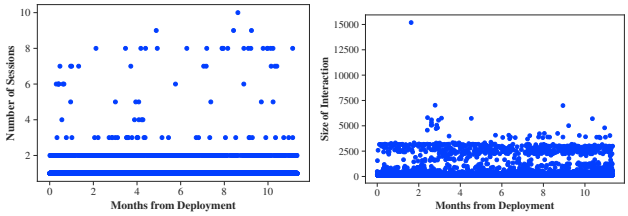
(a) (b)

Fig. 2. (a) Log activity length versus the time of day (hours). The hour represents the data center local time when the attack interaction started. (b) Starting day of individual attacks. The x-axis indicates the day when the attack interaction began.

Day of the Week: Attack session patterns are consistent and do not vary according to the day of the week.

Next, we plot the distribution of the attack length for the days of the week in Figure 2b. The figure shows that the attack patterns are relatively consistent for each day of the week. These results indicate that these attacks are most likely automated.

Number of Sessions: Most of the attacks per day are short and include only one or two sessions. The number of sessions per day are shown in Figure 3a, where each point is considered to be an attacker with a distinct IP address. The figure indicates that the honeypot (LIHP-4) was consistently attacked while it was deployed. The overall majority of these attacks were short and consisted of only one or two sessions per day. However, a few of the interactions included between three and ten sessions per attack.

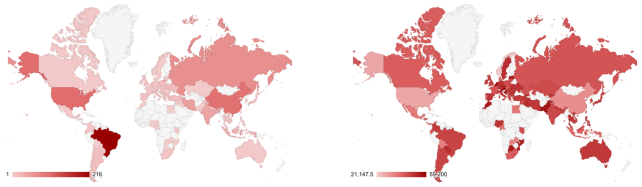


(a) (b)

Fig. 3. (a) Number of sessions over time. (b) Interaction sizes over time.

Attack Interaction Size: The total number of log entries per session is small. We report the interaction size per day in Figure 3b which shows the majority of log entries per session is small.

Attack Origination: Attacks were launched from all over the world. The countries corresponding to the attack location are shown in Figure 4a. The majority of the attacks were initiated from Brazil. A comparable number of attacks were launched from the United States and China. The country with the next highest number of attacks was Russia.



(a) (b)

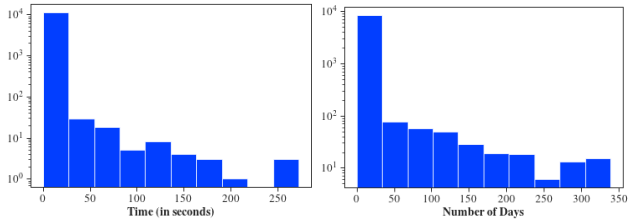
Fig. 4. (a) Total number of interactions based on the originating country. (b) Average attack payload size based on the originating country.

Attack Size from Different Countries: The average message payload size based on the originating country is depicted in Figure 4b, and we can observe that it is significantly different than the attack origination. The average size of the payload from regions like Russia, Australia, and Europe is much larger than the averages from the United States, even with a lower interaction count.

Attack Patterns over Time: We also report the distribution of session lengths (in seconds) for each attack on the honeypot in Figure 5a. Since we were able to extract the host IP address from the requests, we also report the duration between the first and last attack by each IP address and present the distribution over the number of days in Figure 5b.

Another metric for understanding the attacks is the size of the message payload that the adversary is transmitting to the honeypot. As shown in Figure 5a, most of the attacking IPs only interacted with the honeypot for a short duration, but for the ones with longer interaction periods, we computed their message payload sizes over the months. A sample interaction from one attacker is shown in Figure 6 where the message payload size is reported in bytes.

Content Analysis. The content generated by the activity on LIHP-4 is further analyzed to draw relevant patterns. The logging methodology followed by our system allowed the



(a) (b)

Fig. 5. (a) Log-scale histogram of session lengths in seconds. (b) Log-scale histogram of overall interaction length per IP address over days.

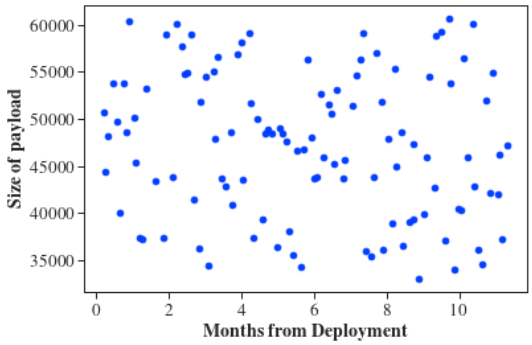


Fig. 6. Message payload sizes in bytes over time from a single host IP.

interaction logs to be captured in two categories: 'raw' and 'request'. Since each interaction log is through an HTTPS request, we also noted the request type for the interaction. Among the requests, 90.96% were *GET* requests, while 8.35% were *POST*. We also observed 0.69% *HEAD* requests.

Raw logs: These are pure traffic logs recorded by the server, capturing everything in the requests made to the server. The logs, containing text in the 'utf-8' character set, are then analyzed using Latent Dirichlet Allocation (LDA) [3] models. After training the model to learn 20 topics, the model was able to place the logs in buckets of words with a topic defining words that occur commonly together across all the logs. The number of individual log entries, serving as the documents for the LDA model was 166,416, with 9,049 unique tokens used after filtering the extremes.

Request logs: The request logs are pre-processed at the request stage and capture cleaner request data in the form of arguments attached to the messages. By analyzing this data separately, we are able to capture cleaner interaction data without being overpowered by the rest of the payload. By learning LDA topic models on these logs, using 20,135 entries and 176 unique tokens, we derived 20 topics. The top 5 topics are shown in Table II. As can be seen, these topics represent different families of vulnerabilities and attacks. For instance, Topic 1 is associated with attacks on the *PHP* framework-based systems, whereas Topic 4 shows the commands involved with the *NMAP* network scanner. An additional advantage of learning these topics from interactive logs is that real-time network activity can be validated against these topics, and in

case of strong inclination towards a topic, the request can be handled accordingly.

TABLE II
TOPIC MODELS TRAINED ON INTERACTION KEYWORDS.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
php	cgi	html	Nmap	txt
index	tmUnblock	manager	folder	robots
echo	vars	sitemap	scripts	known
public	call_user_func_array	xml	admin	security
TP	function		asp	
htm	invokefunction			

V. Related Work

In [4], Pouget and Holz deployed low-interaction alongside high-interaction honeypots for a qualitative and quantitative comparison of the two approaches. They concluded that "high interaction honeypots are somehow superfluous in the large-scale deployment of statistical sensors" but complement low-interaction honeypots by providing high accuracy data and the identification of "[...]very specific attacks and weird phenomena". This complementation was also demonstrated in [5] with the additional observation that there seems to be no attempts by adversaries made to identify the honeypots as such. In [6], it was shown that high-interaction IoT honeypots are generally more attractive for attackers than their emulated counterparts. It was also observed that the location and novelty of the attack vector had significant effects on the number of events observed. Faust [7] deployed five SSH honeypots in Asia, Europe, and Northern America. In his analysis, he focused on the credential combinations and the distribution over time looking for recurring patterns. In [8], Udhani et al. found a way to distinguish between an automated and a human attack. Their model can be used to identify advanced attacks against an SSH server. In [9], Fraunholz et al. stated multiple hypotheses regarding the data captured in a long-term, distributed deployment and showed a diverse spatial and temporal distribution of attacks, correlation between the number of attempts and the protocol, and a significant number of Mirai related events. For a better understanding of the Mirai and Mirai like botnets, Antonakakis et al. extensively researched [10] this malware family. Vetterl and Clayton investigated the potential to systemically fingerprint honeypots in [11] and found multiple low- and medium-interaction honeypots to be vulnerable to a trivial single packet detection method. Together with Walden they also researched how up to date a popular SSH honeypot is in the wild and found in [12] that many deployments are outdated and often in the default configuration making them very prone to fingerprinting. Similar research was done by Mukkamala et al. focusing on virtual environments and low interaction honeypots. Holz and Raynal also considered debuggers in their research on detecting honeypots and suspicious environments in [13]. Tsikerdekis et al. analyzed the methodologies behind attackers identifying honeypots and provided recommendations on how to improve their evasion strategies [14].

VI. Conclusion

Cloud services are constantly attacked by those who seek to use compromised resources for malicious purposes. To understand the current attack strategies, it is important to create and deploy honeypots in data centers. We found that low-interaction honeypots are attacked frequently, however we could not find much evidence that attackers visited and interacted with our medium-interaction honeypot.

References

[1] B. Arzani, S. Ciraci, S. Saroiu, A. Wolman, J. W. Stokes, G. Outhred, and L. Diwu, "Privateeye: Scalable and privacy-preserving compromise detection in the cloud," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. Santa Clara, CA: USENIX Association, Feb. 2020, pp. 797–815. [Online]. Available: https://www.usenix.org/conference/nsdi20/presentation/arzani

[2] M. Nawrocki, M. Wählisch, T. Schmidt, C. Keil, and J. Schönfelder, "A survey on honeypot software and data analysis," *ArXiv*, vol. abs/1608.06249, 2016.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 993–1022, Mar. 2003.

[4] F. Pouget and T. Holz, "A pointillist approach for comparing honeypots," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, K. Julisch and C. Kruegel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 51–68.

[5] E. Alata, V. Nicomette, M. Kaaniche, M. Dacier, and M. Herrb, "Lessons learned from the deployment of a high-interaction honeypot," in *2006 Sixth European Dependable Computing Conference*, 11 2006, pp. 39 – 46.

[6] J. Guarnizo, A. Tambe, S. Bhunia, M. Ochoa, N. O. Tippenhauer, A. Shabtai, and Y. Elovici, "Siphon: Towards scalable high-interaction physical honeypots," in *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*, ser. CPSS '17. Association for Computing Machinery, 04 2017, p. 57–68.

[7] J. Faust, "Distributed analysis of ssh brute force and dictionary based attacks," https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1083&context=msia_etds, Tech. Rep., 05 2018.

[8] S. Udhani, A. Withers, and M. Bashir, "Human vs bots: Detecting human attacks in a honeypot environment," in *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*, 2019, pp. 1–6.

[9] D. Fraunholz, M. Zimmermann, A. Hafner, and H. Schotten, "Data mining in long-term honeypot data," *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 649–656, 2017.

[10] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[11] A. Vetterl and R. Clayton, "Bitter harvest: Systematically fingerprinting low- and medium-interaction honeypots at internet scale," in *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. Baltimore, MD: USENIX Association, Aug. 2018. [Online]. Available: https://www.usenix.org/conference/woot18/presentation/vetterl

[12] A. Vetterl, R. Clayton, and I. Walden, "Counting outdated honeypots: Legal and useful," in *2019 IEEE Security and Privacy Workshops (SPW)*, 2019, pp. 224–229.

[13] T. Holz and F. Raynal, "Detecting honeypots and other suspicious environments," *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, pp. 29–36, 2005.

[14] M. Tsikerdekis, S. Zeadally, A. Schlesener, and N. Sklavos, "Approaches for preventing honeypot detection and compromise," in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, 2018, pp. 1–6.