Lack of Intentionality: Honeypots Show Us Wandering Drones

Author: Jesse La Grew

Abstract

Many honeypot studies focus on the trends, sources, and motivations behind recorded attacks. For organizations choosing hosting providers, it is compelling to understand any differences in the attacks between providers. Honeypots were deployed to multiple cloud environments, including Amazon Web Services (AWS), Digital Ocean, Google Cloud Platform (GCP), and Microsoft Azure. These were compared with a residentially hosted honeypot to understand any differences between SSH, telnet, and web attacks. The data was processed to compare differences in attack volume, destination ports probed, terminal commands run, malware submitted, Uniform Resource Locator (URL) paths accessed, and credentials supplied for brute force access attempts. The information demonstrated that there are few differences between providers. Activity seen is a combination of automated scanning and botnet activity. There are some unique outliers seen in different target networks. The data highlights that addressing the primary cyber defenses may hold more significance than focusing on targeted attacks in a particular hosting environment.

**1. Introduction**

In 1999, putting an unprotected device on the internet could result in a compromise within 15 minutes (Spitzner, 2003). In 2024, it could be even faster. The wide availability of broadband internet and powerful computing devices creates new opportunities and risks. The timeframe for a device to be probed by some external source on the internet can be as short as a few seconds and rarely more than a minute (SANS, 2024).

Researchers use honeypots worldwide to understand the landscape of internet attacks. These honeypots range based on the services provided to attackers and the level of interactivity that they allow. Low-interaction honeypots gather information passively from attack sources. They may not reply to communication requests or respond with minimal customized responses. Medium-interaction honeypots allow for more interaction by an attacker. The interaction is usually within an emulated system rather than a real one. They can allow someone to log into a virtualized system, run commands, and even download malware. It does not allow someone to make system changes. Mediuminteraction honeypots are also limited in the information they are provided to emulate. High-interaction honeypots give an attacker a functional system, allowing for more interaction. A proxy connection is often used to record interactions. Due to the increased risk of providing attackers with a functional system to interact with, researchers must address those risks through defensible architecture (Melese, S. Z., & Avadhani, P. S., 2016).

The DShield honeypot is made available by the SANS Internet Storm Center (ISC), combining low and medium-interaction honeypot services. One of the lowinteraction honeypot services is the web honeypot, facilitated by Twisted Python (Twisted Project, 2024). In addition to the web honeypot, iptables is used on DShield honeypots to log all incoming traffic. Cowrie is used to emulate a shell connection through telnet or SSH (Secure Shell) and is the medium interaction honeypot that allows some user interactivity. These tools have been used to understand attacks on the internet and their source. They often prove useful when looking at emerging threats like Log4Shell (Yamamoto & Yamaguchi, 2023).

Organizations consider many factors when hosting their public-facing services with a hosting provider. These include service resilience, cost, and support. This research questions whether the threat in one hosting environment is different from another. The similarities and differences in these environments can be reviewed by hosting honeypots in different cloud hosting environments and comparing them with a residential honeypot.

**2. Research Method**

This research examines the similarities and differences in data received from DShield honeypots distributed within different cloud hosting providers. The focus is on the attack surface made available from these honeypots, which includes telnet, SSH, and web services. Firewall data was also used to understand attack volume and destination port communication attempts. Comparing these datasets with a residentially hosted honeypot will also help describe differences between residential and cloud-hosted networks. Additionally, usernames and passwords submitted to the Cowrie service were compared and evaluated to determine targeted attacks for specific products and services. Data was collected between 4/21/2024 and 5/21/2024.

**2.1. Distributed Honeypots**

DShield honeypots were distributed in various popular cloud hosting environments within Linux virtual machines (Appendix B). These devices were installed using default settings outside of changes made to maintain additional logs. Additional logs were forwarded to a DShield-Elk instance (Bruneau, 2024) for ad-hoc analysis. Services utilized and zone locations are outlined in Figure 1.

|  |  |
| --- | --- |
| **Service** | **Zone** |
| Amazon AWS, Lightsail | (Virginia, Zone A) |
| Google Compute | (us-central1-a) |
| Digital Ocean | (SFO3, San Francisco) |
| Microsoft Azure | (East US 2) |

**2.2. Honeypot Services**

DShield Honeypots used multiple TCP ports to receive information from telnet, SSH, and web services, as outlined in Figure 2.

|  |  |
| --- | --- |
| **Service** | **Publicly Available Ports** |
| Web Honeypot (ISC Agent) | TCP 80, 8080, 7547, 5555, 8000, 8443, 9000 |
| Cowrie | TCP 22, 23, 2222, 2223, 2323 |
| Firewall (iptables) | N/A |

Figure 2: Honeypot data sources and network ports

**2.3. Data Processing**

JSON logs were collected from honeypots and processed using a custom Python package. The Python package and example processing scripts are available on GitHub (La Grew, 2024).

**3. Findings and Discussion**

The information from the honeypots showed that most of the attacks seen by the individual honeypots are similar. There is little network targeting based on this dataset. Only one network showed significantly different behavior and volume. The residential network appears to have had at least one large Qualys scan. The scan was identified based on data containing the string “QUALYS” such as the user agent “${jndi:corba://10.10.11.42:45518/QUALYSTEST}” and the URL “/QUALYS730141”. The other honeypots did not indicate any activity from the scanner source and similar user agent strings. The intent behind this scan was unknown.

**3.1. Data Outliers**

The data uncovered several outliers that required additional review. Several peaks were seen within the Amazon, Digital Ocean, and residentially hosted honeypots and highlighted in Figure 3 and Figure 4

Comparing the URL paths seen by exact match alone does not address small URL changes that may have a similar intent. For example, phpMyAdmin discovery attacks are often seen, but the URL may change based on the target and phpMyAdmin software versions, as seen in Figure 10. Another example from the top 10 URLs above is “/<honeypot IP>:9000/.env”, which includes the IP address of a honeypot, shown in Figure 11. This URL path included three unique values and was manually consolidated when the data was compared. From the other honeypots, there may be examples of similar paths but with a different IP address or port, as shown in Figure 10.

A subset of the data was analyzed to help determine any URL path uniqueness between honeypots. This subset comprises any URL paths with 0 results from any honeypot. The rows with 0 results for any honeypot helped identify paths that may appear unique when using literal comparisons but may be grouped using clustering techniques. These were then clustered using DBSCAN (Appendix D).

The different clusters demonstrate that there are other commonalities between URL paths. For example, a cluster of 646 unique paths accounted for almost 500,000 access attempts, nearly three times greater than the total access attempts for the root path (“/”). This cluster included access attempts for “.env” configuration files but with different directory paths, as demonstrated in Figure 11. The attempted access of configuration files was commonly seen between all honeypots. AWS had a higher overall volume of web access attempts and had the most significant diversity of directory paths used when accessing “.env” files. The data included paths referencing “admirer,” not seen in other honeypots. This traffic originated from three unique sources that were not seen scanning the other honeypots. The lack of these source IP addresses in other honeypot data could indicate scanning sources with limited resources to scan large internet populations.

When comparing all the resulting clusters, of which there were 120 in total, 54 of them contained only AWS honeypot URL paths. The rest of the clusters included at least two sources of honeypot data. This indicates that similar URL paths were only received by the AWS honeypot. The most common examples of these access attempts include:

• /\*/ConnectionStrings.config

• /\*/server.xml

• /\*/.gitlab-ci.yml

• /\*/api-config.ini

• /\*/remote-sync.json

• /\*/api-config.ini

**3.5. Malware**

Files were generated on honeypots through the terminal, download commands, and directly uploaded through a secure copy (SCP). 99% of individual submissions were from two files. The most common is an SSH key uploaded to honeypots to attempt persistence. The second most common is an empty file with one newline character. The number of malware submissions to the AWS honeypot was lower than the others. This highlights previous data regarding the increased popularity of web-based attacks for AWS honeypots and the lower frequency of SSH or telnet attacks. Additional metadata was used from VirusTotal to compare the files created on the honeypots. Anything with a blank “description” or “classification” was removed. A comparison of VirusTotal descriptions for files created on honeypots is shown in Figure 12. The malware descriptions overlapped, for the most part, outside of several instances of Windows executables submitted to the Digital Ocean honeypot. Some were malicious files associated with the Prometei botnet (Bruneau, 2024). Others were valid executables from 7-Zip, a utility used by many Windows users. There were many generic classifications related to Linux trojan variants (Appendix E). Most of these “trojan.r002c0\*” variants upload an “sshd” file. There is no indication from the honeypots that there is any other activity outside of uploading this file.

**3.6. Terminal Commands**

Terminal commands were collected for interactive telnet or SSH sessions. The most common commands were shared among all honeypots. The top 10 commands accounted for 47% of all commands run. The top 20 commands accounted for 89% of all commands run. The comparison of the proportions of the top 10 commands seen for each honeypot is shown in Figure 13.

All these commands include some password changes but were uniquely performed with appended or piped commands. The password change from the Digital Ocean honeypot is unique due to the length of the password change command. There were no other “Dolphinscheduler…” password change attempts from any honeypot, which is likely an attempt to impact systems running Apache DolphinScheduler.

**3.7. Usernames**

Usernames can often indicate specific systems or services being attacked. Default usernames and passwords are regularly targeted by brute force and credential-stuffing attacks. The most frequently seen usernames, shown in Figure 16, were seen in high proportions among all the distributed honeypots.

**4. Recommendations and Implications**

A comparison of data from different distributed honeypots showed an overlap in attacks, regardless of the placement of any host within any target network. There was a higher volume of web attacks in AWS and an increased variety of directories tested using discover attacks. Otherwise, commands ran, URL paths accessed, and malware submitted were similar. Firewall traffic revealed a higher interest in different services during the period. With more data collected to compare, the differences between honeypots may lessen.

**4.1. Recommendations for Practice**

For defenders, this data highlights that basic cyber hygiene practices may be more broadly impactful than focusing on defenses for specific technical attacks, at least when considering publicly facing resources. Many ports, such as telnet and SSH, should not be publicly available. In addition, the impacts of web-based attacks can be offset using web application firewalls (WAFs), content distribution networks, and keeping web applications up to date.

**4.2. Implications for Future Research**

This research still leaves some additional questions based on the data collected. More data would help to understand whether current attack differences are maintained, such as password use, port probing, or attack volume. Digital Ocean had a higher volume of MySQL port probes. With a MySQL honeypot, exciting data could be collected from various networks and compared. To expand on this research, multiple avenues could be taken, such as:

• Expanding geographic distribution

• Expanding service providers/target networks

• Expanding honeypot services

• Collecting and processing additional data (PCAPs)

• Developing better processes to identify network scanning organizations

**5. Conclusion**

Many studies have focused on attack attribution and how attacks may change depending on the source. Reviewing how attacks change based on the target network rather than the source helps network defenders consider how attacks may be targeted in different environments. It has demonstrated that, for the most part, attacks are very similar between hosting environments. Some differences exist in attack volume, services probed, and the diversity of URL paths used. Overall, attacks on the internet are broad and distributed widely. Focusing on foundational network protections may be the best focus for any defender.