Longitudinal Analysis of SSH Honeypot Logs

Bachelor's thesis proposal

Dominic Rudigier
Dominic.Rudigier@student.uibk.ac.at

11832156

May 5, 2021

1 Motivation

A honeypot is a sacrificial computer system which is intended to attract potential hackers. It mimics a target and is built to gain information about how a attack was intended to be executed. A research honeypot is built to extract as much information about the specific methods and tactics used as possible, unlike a production honeypot which is used by businesses to detect attacks in their internal networks. Once connected to the internet the honeypot generates log files for all events, but the gathered information is hard to handle.

While a large number of honeypot related tools exist,[1] they generally focus on high-level aggregated statistics and not about individual log anomalies. These anomalies can be user/password combinations, weird looking strings or other executables, indicating new device vulnerabilities of specific vendors. For example, an attacker may find out that a particular device can be accessed with hardcoded configuration credentials, which is a common problem in software and hardware.[2] If someone finds this vulnerability they can harm the system. This is crucial for a company to know to protect their users and data from malicious actors. Attacks happen regularly by hackers and botnets which are trying to exploit already known or new vulnerabilities of specific versions of software systems.

Potentially new exploits can be found using honeypots analyzing millions of attacks happening every day.[3]. O rganizations and providers deploy thousands of honeypots. Hackers might find out they connected to a honeypot and disconnect, so their configurations need to be optimized. The main problem is that there is no easy way to analyze and visualize the log data of many honeypots over time. This is necessary to find weak spots in honeypot configurations. Batch-processing log files of multiple honeypots can be used to further improve our systems. One goal is to find out why the attacker disconnected and view the previously executed commands, which might give us ways to improve it. Malware could have found paths to detect our system by executing common commands and matching the results with possible honeypot attributes. The main contribution of this thesis is a tool that visualizes attacks encountered by a network of SSH honeypots over time. This helps their operators to detect new attacks and supports honeypot developers in adapting their honeypots.

As every honeypot is generating its own log files, it can be seen that it would be beneficial to analyze and process data of all honeypots in parallel. Using programming models like MapReduce [DG04] provide the possibility to aggregate files discretely (map) and connect the aggregated results to overall results (reduce).

Therefore the main focus lies on techniques to adapt a honeypot using the information queried from potentially thousands of batch-processed log files across different systems.

[1] https://github.com/paralax/awesome-honeypots

[2] https://cwe.mitre.org/data/definitions/798.html

[3] https://www.sicherheitstacho.eu/start/main

2 Status quo

The current state of the art for logging brute-force attacks and shell interactions of connection-based intrusion attacks is a SSH and Telnet honeypot called Cowrie[1] by micheloosterhof (Github). Cowrie grants the possibility to catch attacker actions and provides logs of all kinds like JSON, UML, TTY, Splunk HEC and different databases like MongoDB, SQLite and MySQL [Cab+19]. Although there exist data analyzation platforms like Splunk Enterprise[2] (which is not for free) with Cowrie analysis apps like Tango[3] they are all more concentrated on the individual analysis of honeypots and collection of statistics than on a severe analysis of individual attacker behaviors and action paths. Malware has become more intelligent recently which enlights the need for analysis and improvement of intrusion detection systems.[4]

3 Modus operandi

3.1 Goals

The primary goal of the work is to batch-process many connection-based honeypot log data and extract useful information to improve existing systems and have as well a possibility to view interaction based attacker behavior. There are tons of attacks happening all the time, the tool should provide a possibility to detect changes in attacker behaviour over time like sudden disconnects after specific commands or general log data anomalies not previously shown up. As an initial try the MapReduce programming model [DG04] is used on local files to process and gather information of the honeypots. As this is non-trivial there will only be time for an easy visualization and probably no time for machine learning attempts for detection of outliers, this will be left for future work.

[1] https://github.com/cowrie/cowrie

[2] https://www.splunk.com/

[3] https://github.com/aplura/Tango

[4] https://www.avira.com/en/blog/new-mirai-variant-aisuru-detects-cowrie-opensource-honeypots

3.2 Timeline

February	Research ssh honeypots
March	Set up multiple cowrie honeypots, test Splunk, test logging functionalities
April	Research MapReduce programming model
April	Research batch-processing log data & aggregating information
May	Run MapReduce job across multiple log files locally
May	Develop initial MapReduce job extracting information of local log files
May	Visualize MapReduce data
June	Extract pre-disconnect commands frequency across all nodes
June	Detect outliers for different logs
June	Handle bugs/improvements
July	Dream: Visualize everything and provide hints to improve honeypots
July	Finish batch-processing system
August	Write thesis

3.3 Risks

- Gathering Data
  As the goal is to have test data for our web application in the first place there is a need to set up honeypots and create those log data for cowrie. With the attraction of attackers there is always a slight risk of them escaping our sandbox and using a honeypot as pivot node to penetrate productive systems. There were used dedicated droplets on DigitalOcean[1] to secure this.

[1] `https://www.digitalocean.com/`

References

[Cab+19] W. Cabral et al. “Review and Analysis of Cowrie Artefacts and Their Potential to be Used Deceptively”. In: *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2019, pp. 166–171. DOI: 10.1109/CSCI49370.2019.00035.

[DG04] Jeffrey Dean and Sanjay Ghemawat. “MapReduce: Simplified Data Processing on Large Clusters”. In: *OSDI’04: Sixth Symposium on Operating System Design and Implementation*. San Francisco, CA, 2004, pp. 137–150.

[GGL03] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. “The Google file system”. In: *Proceedings of the nineteenth ACM symposium on Operating systems principles*. 2003, pp. 29–43.

[KJE19] S. Kumar, B. Janet, and R. Eswari. “Multi Platform Honeypot for Generation of Cyber Threat Intelligence”. In: *2019 IEEE 9th International Conference on Advanced Computing (IACC)*. 2019, pp. 25–29. DOI: 10.1109/IACC48062.2019.8971584.

[KS18] A. Kyriakou and N. Sklavos. “Container-Based Honeypot Deployment for the Analysis of Malicious Activity”. In: *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. 2018, pp. 1–4. DOI: 10.1109/GIIS.2018.8635778.

[San20] Chris Sanders. *Intrusion Detection Honeypots, Detection Through Deception. Detection Through Deception*. Applied Network Defense, 2020. ISBN: 978-1735188300.