HoneyHints
Extract non-trivial cowrie log information
Bachelor's thesis proposal

Dominic Rudigier

Dominic.Rudigier@student.uibk.ac.at

11832156

April 8, 2021

1 Motivation

Additional to our firewall and other security solutions a honeypot helps protect a network from hackers, designed to catch the hacker's attention. Vulnerable as an easy target it grants possibilities to track the hacker's activities. Once connected to the internet the honeypot generates log files for all events, but the gathered information is hard to handle. If the firewall fails to prevent attacks we know how far the hacker has invaded our network and whether there are dangers of accessing company data.

One might ask why there is a need for additional tools as the quantity of honeypots and related tools is already slightly overwhelming.[1]

There are multiple tools available showing log files and statistics of honeypots but there are none to perform an intelligent and useful analyzation to extract non-trivial information to detect anomalies in the log data. These anomalies might be user/password combinations indicating new device vulnerabilities of specific vendors or similar. Furthermore, analyzing commands executed prior to disconnects of the honeypot might give us a chance to find ways to improve it as malware might have found some ways to detect honeypot systems executing common commands known to be honey. The goal of this work is to find new ways to batch process cowrie logs and extract useful information to provide a possibility to be one step ahead of the attacker and iteratively improve our deployed honeypots. Imagine a scenario where an attacker has already access to our
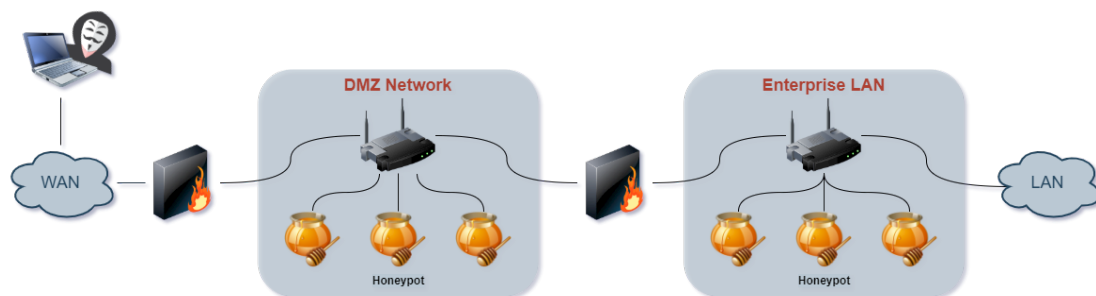

Figure 1: Possible network infrastructure

prevalent honeypots in a demilitarized zone and has connected to it. If our setup in this secured zone is similar to our enterprise network chances are high that we can spot common exploits or hopefully also zero-days if our honeypot is not detected. This possibility provides a huge advantage to secure our system from the maleficent actor and be one step ahead as we can further secure our enterprise LAN with all the knowledge gained. Therefore the main focus lies on techniques to adapt a honeypot to further hide it from the attacker to get quantitative information of attacker behavior as a result.

---

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

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 [1]. 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 connection-based honeypot log data and extract useful information to improve existing systems and have 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 anomalies in attacker behavior like sudden disconnects after specific commands or general log data anomalies not previously shown up. As an initial try the FReMP Stack[5] is used to develop a web application to easily visualize python processed log files from a database. Research will show whether machine learning attempts could be helpful in detecting outliers for some cases or a manual analyzation is preferable, which is not clear yet. The software structure of the intended system can be viewed in 2.
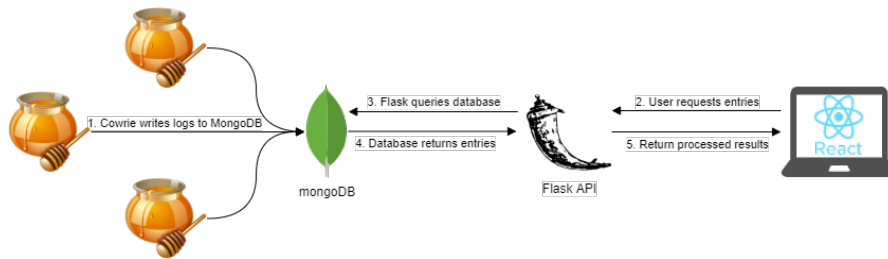

Figure 2: Web application infrastructure

---

[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

[5] https://fremp.github.io/

3.2 Timeline

February	Research ssh honeypots
March	Set up multiple cowrie honeypots, test Splunk, test logging functionalities
April	Start developing FReMP honeyview application (initially 2 honeypots)
April	Research machine learning attempts of log anomaly detection
April	Research batch-processing log data & aggregating information
April	Visualize pre-disconnect commands frequency
May	Detect outliers for different log tables and display
May / June	Extend functionality on n honeypots
June	Handle bugs/improvements appeared during development process
July	Dream: Provide hints to improve honeypots
July	Finish application
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. Props for the free credits.

- Database
  A database system with potential user connection data might attract some black-hats as well. A basic mongodb authentication and a firewall denying all incoming connections except for our honeypots and our webapp will grant proper security.

---

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

References

[1] W. Cabral, C. Valli, L. Sikos, and S. Wakeling. Review and analysis of cowrie artefacts and their potential to be used deceptively. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 166–171, 2019.

[2] 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)*, pages 25–29, 2019.

[3] A. Kyriakou and N. Sklavos. Container-based honeypot deployment for the analysis of malicious activity. In *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–4, 2018.

[4] C. Sanders. *Intrusion Detection Honeypots, Detection Through Deception.* Applied Network Defense, 2020.