honeyView
A multiple cowrie honeypot log analysis and tracing tool

Bachelor's thesis proposal

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

11832156

March 30, 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 compare data of multiple honeypots or even track a specific intruder through our network across multiple honeypots. Further there is no convenient method to alert on specific connection attempts or connections to new systems, currently happening only on e.g. an operating system level.

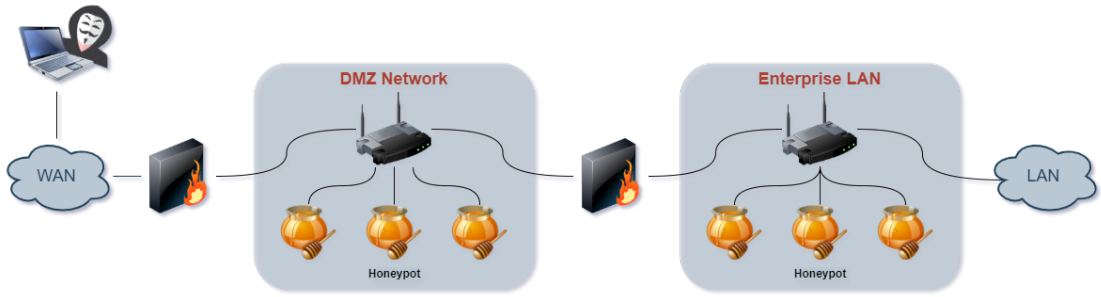

Figure 1: Possible network infrastructure

Imagine a scenario where an attacker has already access to our prevalent honeypots in a demilitarized zone and has connected to it. It is essential to view what the hacker is doing and alert on a possible connection to the enterprise LAN for prevention of further damage. A trace of the intruder path would be beneficial for analyzing the security vulnerability used by the attacker without manually analyzing several log files across multiple honeypots. It is also desired to have an easy way for later forensic analysis.

[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 path tracing of individual attackers through our different honeypots and a comparison of the gathered data and connection attempts.

3 Modus operandi

3.1 Goals

The primary goal of the work is to research connection-based honeypots like Cowrie and develop an easy-to-use log analysis tool for comparison of multiple cowrie honeypots. The tool displays common attributes shared between multiple honeypots extracted from the log data. Finally it should enable the possibility to trace attackers through our system of honeypots and view the commands executed at specific timestamps and on different machines. As an initial try the FReMP Stack[4] is used to develop a web application to analyze logs across multiple honeypots. There is a need to find ways to display gathered information in a useful way as well as visualizing the path of individuals. The software structure of the intended system can be viewed in 2.

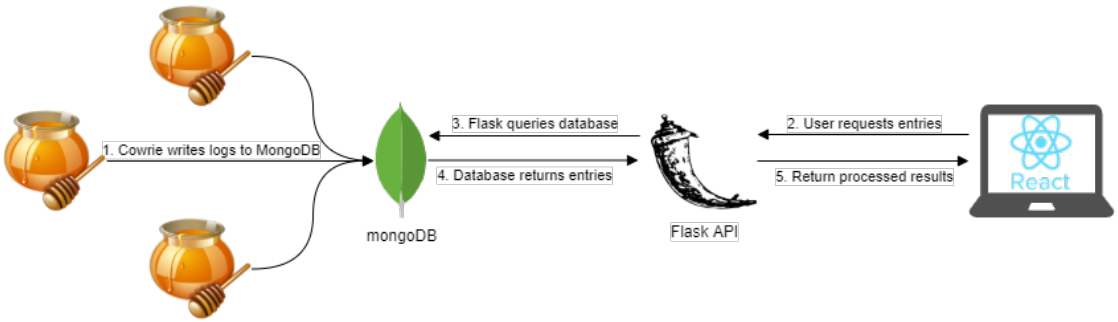

Figure 2: Web application infrastructure

[1] https://github.com/cowrie/cowrie
[2] https://www.splunk.com/
[3] https://github.com/aplura/Tango
[4] https://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	Visualize common attacker attributes between honeypots
April	Path tracing for IP
April	Locate IP addresses
May	E-Mail alerts on new IPs arising in log files
May	Extend functionality on n honeypots
June	Handle bugs/improvements appeared during development process
July	Finish web 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.

- 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] Cabral, W., Valli, C., Sikos, L., and Wakeling, S. 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.

[2] Kumar, S., Janet, B., and Eswari, R. Multi platform honeypot for generation of cyber threat intelligence. In *2019 IEEE 9th International Conference on Advanced Computing (IACC)* (2019), pp. 25–29.

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

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