universität innsbruck

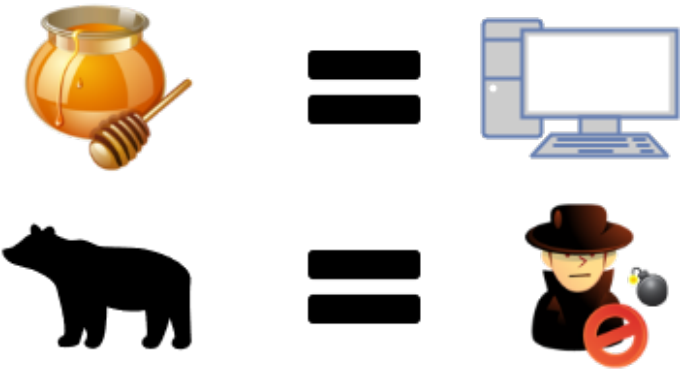
Longitudinal Analysis of SSH Honeypot Logs

Security and Privacy Lab, Department of Computer Science

Dominic Rudigier, 11832156 | Supervisor: Maximilian Hils

Motivation

Motivation



Motivation

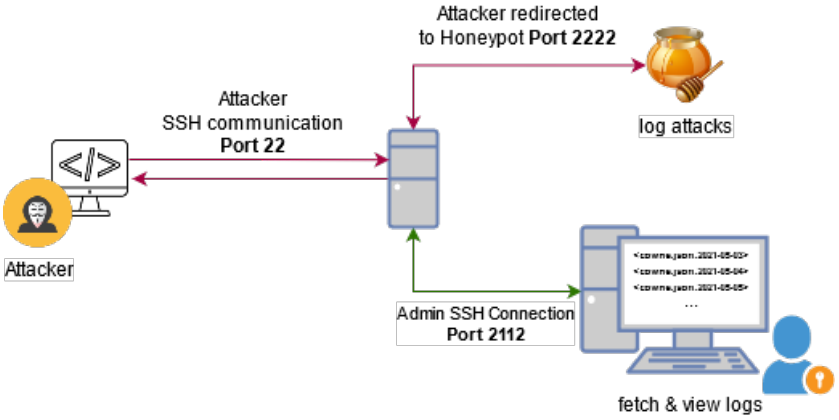
Honeypot:

- Mimics easy target and attracts attackers
- Generates logs about connection data (commands, files uploaded)
- Gained information can be used to improve systems

Example

1. Hardcoded credentials in software
2. Attacker found out somehow
3. Analyzing honeypot logs shows that hacker knows
4. Vendor can patch vulnerability

Motivation


Attacker redirected
to Honeypot Port 2222
log attacks
Attacker
SSH communication
Port 22
Attacker
Admin SSH Connection
Port 2112
fetch & view logs

Status quo

Cowrie:

- SSH and Telnet honeypot
- Different log formats for connection data (JSON, UML, MongoDB,..)

Problem

- Malware has become more intelligent
- e.g. Aisuru detects Cowrie honeypots
  - existence of "@LocalHost:]"
  - existence of a service, started on Jun 22nd, or Jun 23rd
  - user exists on the device named "richard"
- Improve honeypot configurations

Status quo

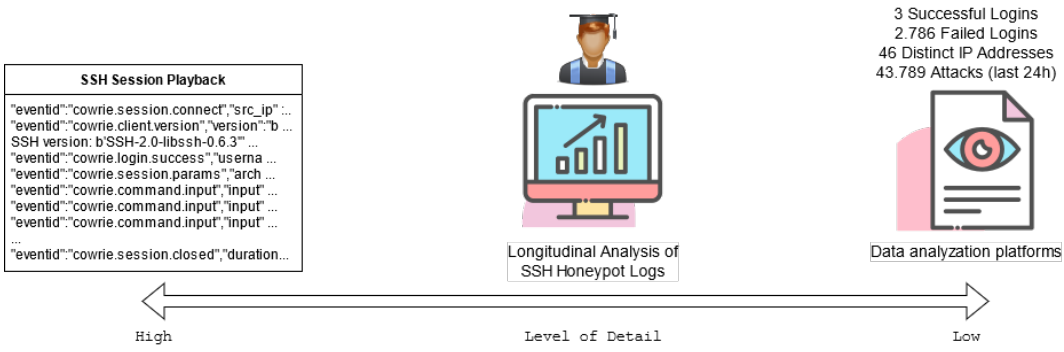
Already existing:

- SSH Session Playback (single connection)
- Data analyzation platforms (aggregated statistics)

Problem

- Providers
  - Multiple thousands of honeypots
  - Each honeypot logs thousands of attacks per hour
  - Too much information gathered, hard to handle.

Thesis contribution



Thesis contribution

Longitudinal analysis:

- Research design that involves repeated observations of the same variables.
- Research attacker behaviour over time.

Batch-processing of log files

1. Cowrie generates log files **<cowrie.json.2021-05-01, 100 to 200 MB>**
   - per honeypot (instance)
   - per day (multiple)
   - 1k honeypots / 30 days = 3-6 TB log data per month
2. Batch-process using MapReduce model with Python
3. Visualize (Python or Flask + ReactJS)

Batch-processing of log files

cowrie.json.2021-05-01: Log files for each honeypot each day

Generated logs
{"eventid":"cowrie.session.connect","src_ip":"5.253.24.65" ..
{"eventid":"cowrie.client.version","version":"b'SSH-2.0-li ..
{"eventid":"cowrie.client.kex","hassh":"51cba57125523ce4b9 ..
{"eventid":"cowrie.login.failed","username":"minecraft","p ..
{"eventid":"cowrie.session.closed","duration":1.7365803718 ..
{"eventid":"cowrie.command.input","input":"cat /proc/cpuin ..

Batch-processing of log files

MapReduce: Programming model for performant data analysis

Procedure	Functionality
`\split_func{...}`	split into JSON objects
`\map_func{...}`	map, filter and sort objects
`\shuffle_func{...}`	combine same events
`\reduce_func{...}`	aggregate event data to summary

Batch-processing of log files

Method	Output
Input	`{"eventid":"cowrie.session.connect"..}`

Map	`{`
	`  honeypot: "honeypotA",`
	`  date: "2021-04-25",`
	`  passwords: [ {user: "foo", password: "bar",`
	`  count: 7}, ... /* top N attempts today */ ]`
	`}`

Reduce	`[{.., count:329}, {.., count:31}, {..}]`

Goal

Extract information:

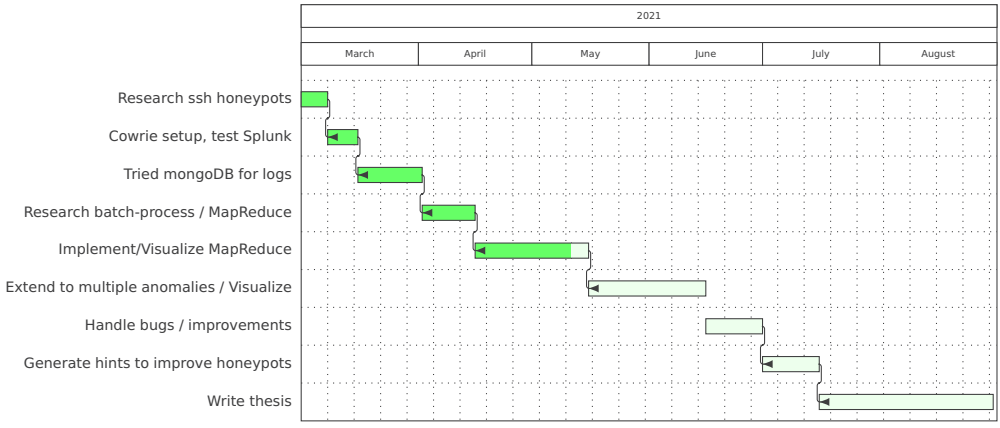
1. Detect changes in attacker behavior over time

Changes over time might indicate new vulnerabilities

- User:Password combination changes
- Commands executed before disconnect
- Command quantity changes
- All log anomalies not previously shown up

2. Visualize
3. Find ways to improve honeypot configurations

Timeline


2021
March
April
May
June
July
August
Research ssh honeypots
Cowrie setup, test Splunk
Tried mongoDB for logs
Research batch-process / MapReduce
Implement/Visualize MapReduce
Extend to multiple anomalies / Visualize
Handle bugs / improvements
Generate hints to improve honeypots
Write thesis

References I

[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.

References II

[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.

References III

[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.

Thank you for your attention!

Dominic Rudigier, 11832156 | Supervisor: Maximilian Hils

Appendix 1

https://www.sicherheitstacho.eu/start/main
https://www.avira.com/en/blog/
new-mirai-variant-aisuru-detects-cowrie-opensource-honeypots