Report – HoneyPot Server to Detect Attack Patterns

1. Introduction

With the increase in cyber threats, organizations need to proactively understand and counteract attack patterns. Honeypots are decoy systems that mimic real servers, attracting attackers so that their techniques can be studied in a controlled environment. This project implements an SSH/Telnet honeypot using Cowrie, coupled with custom analysis scripts, to observe, analyze, and visualize attacker behaviour.

2. Abstract

The project aims to deploy a secure and isolated honeypot using Cowrie on Kali Linux. The honeypot records attack attempts, which are then parsed and analyzed with Python scripts. Attacker IPs are geolocated using the GeoLite2 database, and their global distribution visualized on an interactive map. This provides actionable insight into common attack vectors and the geographical spread of threats.

3. Tools Used

- Cowrie: An SSH/Telnet honeypot for capturing and logging attack attempts.
- Python 3: For scripting and data analysis.
- GeoLite2-City.mmdb: MaxMind's geolocation database for IP lookup.
- Folium & Matplotlib: Python libraries for data visualization and map generation.
- Kali Linux: The operating system used for deployment and testing.

4. Steps Involved in Building the Project

4.1. System Preparation: Install required packages (`python3`, `pip`, `venv`, `git`, and build essentials) on Kali Linux.

4.2. Cowrie Installation: Clone Cowrie from GitHub, create a Python virtual environment, install dependencies, and copy the example configuration files.

4.3. Configuration:
- Edit SSH port and logging options in `cowrie.cfg` as needed.
- Set up fake credentials in `userdb.txt` to allow easy login for simulated attacks.

4.4. Running the Honeypot: Start Cowrie in the virtual environment. SSH connections to the honeypot are logged in `cowrie.json`.

4.5. Simulating Attacks: SSH logins (from localhost or other machines), running typical attacker commands (e.g., `ls`, `cat /etc/passwd`, `wget ...`).

4.6. Analysis Environment:
- Set up a separate Python virtual environment for analysis scripts (`cowrie-analysis`).
- Install required libraries (geoip2, folium, matplotlib).

4.7. Parsing and Geolocation:
- Run `parse_logs.py` to extract attacker IPs and commands from Cowrie logs.
- Use `geoip_report.py` to geolocate public attacker IPs via GeoLite2.

4.8. Visualization:
- Run `visualize_attack_map.py` to generate an interactive world map (`attack_map.html`).
- Open the map to view real-time attacker locations and activity.

5. Conclusion

This project demonstrates the effective use of honeypots for practical cybersecurity research. By deploying Cowrie and analysing real or simulated attacks, valuable insight is gained into attacker techniques and geographic origins. The modular pipeline—from data capture to visualization—can be adapted for broader organizational security monitoring and threat intelligence.

6. References

- Cowrie Honeypot (Official Documentation) https://cowrie.readthedocs.io/en/latest
- Cowrie GitHub Repository https://github.com/cowrie/cowrie
- GeoLite2 Free Geolocation Database (MaxMind) https://dev.maxmind.com/geoip/geolite2-free-geolocation-data
- Folium — Python Data, Leaflet.js Maps https://python-visualization.github.io/folium
- Matplotlib — Visualization with Python https://matplotlib.org/stable/index.html
- geoip2 Python Library Documentation https://geoip2.readthedocs.io/en/latest
- Kali Linux Official Documentation https://www.kali.org/docs
- OpenSSH Manual Pages https://man.openbsd.org/ssh
- Python 3 Official Documentation https://docs.python.org/3
- SSH Honeypot Concepts (OWASP) https://owasp.org/www-community/attacks/Honeypot

7. Note

For demo purposes in isolated labs, attack geolocation and map visualization require either real external attacks or manual editing of logs to simulate public IPs.
