IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

Analyzing Attacker Behaviour Using Honeypot and Log Analysis

Yash Rajput[1], Krutika Pawar[2], Siddhika Salunke[3], Aarti Sahane[4]
Students, Department of Computer Engineering[1,2,3,4]
Matoshri College of Engineering & Research Centre, Nashik, Maharashtra, India

Abstract: *Cyber threats have become increasingly sophisticated, with attackers continuously exploiting vulnerabilities in online systems. Understanding their behaviour and attack patterns is vital for designing proactive defense strategies. This research focuses on analyzing attacker behaviour through a medium-interaction honeypot named Cowrie, which emulates vulnerable SSH and Telnet services to attract attackers. The honeypot captures comprehensive logs of login attempts, executed commands, and session data. These logs are then processed through a Python-based log analyzer that extracts meaningful insights, such as the frequency of attacks, commonly used credentials, and origin IPs. The system is containerized using Docker for isolation and tested using Hydra, a brute-force tool from Kali Linux, to simulate attacks safely. Visualization tools like Matplotlib and Pandas are used to identify temporal and behavioural attack trends. The findings highlight how honeypots can effectively collect threat intelligence and aid in the development of adaptive cybersecurity systems.*

Keywords: Honeypot, Cybersecurity, Cowrie, Log Analysis, Docker, Hydra, Python, Threat Intelligence

I. INTRODUCTION

Background:
As the internet continues to expand, cyberattacks have become a critical threat to both individuals and organizations. Attackers use automated scripts, botnets, and advanced persistent threats (APTs) to exploit vulnerabilities. Traditional security mechanisms such as firewalls, intrusion prevention systems (IPS), and antivirus software often block known threats but fail to understand why and how these attacks occur.To address this limitation, honeypots serve as controlled decoy environments that simulate vulnerable systems. They attract malicious actors, allowing researchers to record and study attacker behaviour without exposing real assets. This collected data can then be analyzed to uncover tactics, techniques, and procedures (TTPs), providing valuable insights for threat intelligence and intrusion detection systems.

Problem Statement:
Conventional log monitoring tools primarily focus on identifying anomalies in legitimate systems but lack controlled data about attacker intentions. Security analysts struggle to trace the attacker's complete behaviour chain, especially in brute-force and reconnaissance attacks. Therefore, a structured environment capable of capturing detailed attacker interactions, analyzing their activities, and visualizing patterns is essential for modern cybersecurity research.

Objectives: This project aims to:

- Deploy a Cowrie honeypot in an isolated Docker environment to record attack activity safely.
- Collect and parse attacker logs to extract behavioural data such as IP addresses, login attempts, and executed commands.
- Develop a Python-based log analysis tool to visualize and summarize attacker trends.
- Simulate controlled brute-force attacks using Hydra to verify honeypot functionality.
- Provide visual intelligence about attack frequency, origin, and methods, helping improve defensive measures.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

II. LITERATURE SURVEY

Honeypot Systems: Previous studies such as Spitzner (2003) describe honeypots as valuable research tools for capturing malicious activity. Lightweight honeypots like Kippo and Honeyd inspired the development of Cowrie, which supports SSH/Telnet emulation and full session logging.

Log Analysis for Threat Intelligence: Huang et al. (2020) emphasize how automated log analysis helps uncover behavioural patterns in brute-force attacks, supporting the identification of botnets and compromised credentials.

Docker for Secure Deployment: Containerized environments (e.g., Docker) are widely adopted to isolate honeypot instances, minimizing risk while allowing controlled exposure.

Attack Simulation Tools: Van Hauser (2021) introduced Hydra, a flexible password-cracking tool used for generating controlled attack traffic.

Machine Learning for Security Analytics: Rao and Patel (2023) highlight how data-driven log analysis can enhance early warning systems by detecting anomalies in attacker activity.

III. METHODOLOGY

System Design and Architecture: The proposed system follows a modular and layered architecture (Fig. 1), consisting of the following modules:

Honeypot Deployment Module:

- Cowrie honeypot is deployed inside a Docker container on Ubuntu Server.
- It listens on SSH (port 22) and Telnet (port 23) to mimic a vulnerable Linux system.
- All login attempts, executed commands, and session metadata are stored in structured log files (JSON and text).

Attack Simulation Module:

- The Hydra tool is used to perform simulated brute-force attacks.
- The module ensures that Cowrie correctly logs all events.
- Real-world exposure tests are optionally performed to collect genuine attacker data from open internet traffic.

Log Collection and Parsing Module:

Python scripts read Cowrie's log files and extract key attributes such as:

- Timestamp
- Source IP
- Username/Password pairs
- Command sequences

The data is normalized and stored for further analysis.

Log Analysis and Visualization Module:

- Analytical operations are performed using Pandas and Matplotlib.
- Charts visualize attack frequencies, command usage, and IP origin heatmaps.
- These insights enable pattern recognition and comparative analysis.

Alert and Reporting Module:

- Generates automated reports highlighting high-frequency IPs and common attack vectors.
- Optionally integrates email notifications for new activity detection.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

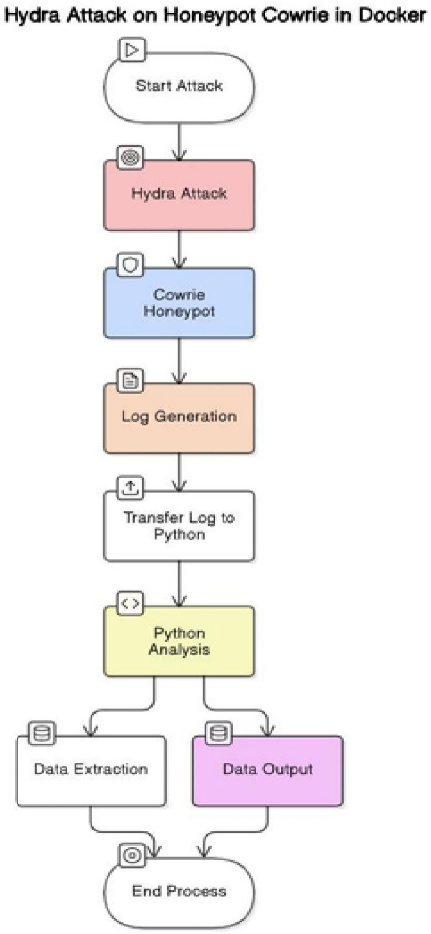

Fig. 1. System Architecture of the Honeypot Environment

Component	Technology Used
Honeypot	Cowrie (Python)
Containerization	Docker
Attack Simulation	Hydra (Kali Linux)
Programming Language	Python 3
Libraries	Pandas, Matplotlib, GeoIP
OS Environment	Ubuntu 22.04 LTS
Deployment	AWS EC2 / Local VM
Visualization	CSV and Graphical Dashboards
Security Layer	IP Filtering, Docker Isolation

IV. SYSTEM IMPLEMENTATION

A. Honeypot Configuration

Cowrie was configured through a Dockerfile that automates dependency installation. Once deployed, the honeypot starts listening for SSH and Telnet connections. When attackers attempt logins or commands, the session is stored as both plaintext and JSON logs, including timestamps, source IPs, and interaction details.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

Configuration highlights include:

- Port redirection via Docker Compose (host port 2222 → container 22).
- Logging via /var/log/cowrie/cowrie.json.
- Custom banners and fake file systems to maintain realism.

B. Data Collection Phase

The system ran continuously for several days, exposed to the public internet on a test EC2 instance. Thousands of SSH attempts were recorded. Each log entry contained metadata (timestamp, IP, username, command) that provided a realistic view of global brute-force patterns.

C. Log Parsing and Analysis

Python scripts read Cowrie logs and filtered fields like:

```
with open('cowrie.json') as file:
    for line in file:
        entry = json.loads(line)
        if entry['eventid'] == 'cowrie.login.failed':
            # Extract failed login attempts
```

After extraction, the data was converted into structured DataFrames. Aggregations such as "top 10 usernames" or "hourly attack count" were calculated. Visualization through Matplotlib produced:

Line charts for time-based attack frequency.

Bar charts for username/password frequency.

GeoIP maps showing attacker distribution.

D. Visualization

Visual outputs were embedded into reports (Fig. 2 – "Top Attacked Usernames", Fig. 3 – "Hourly Attack Distribution").

The results clearly demonstrated peak attack windows during night hours (02:00 – 05:00 IST), indicating automated bots operating from global time zones.

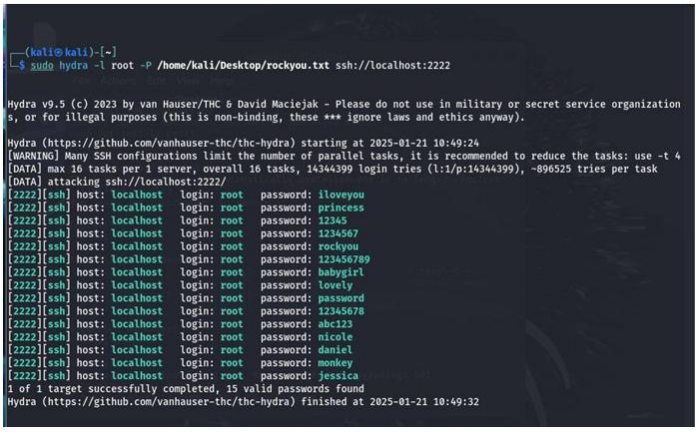

Fig. 2. Top Attacked Usernames

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, November 2025
Impact Factor: 7.67

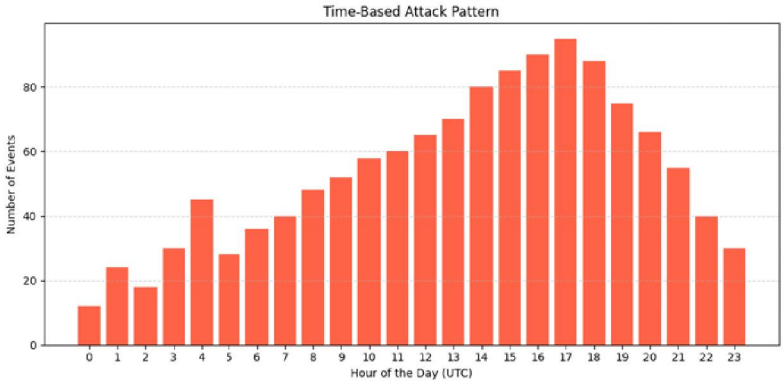

Fig. 3. Hourly Attack Distribution

E. Testing and Validation

Controlled brute-force attacks using Hydra validated the honeypot's response accuracy:

hydra -l root -P passwords.txt ssh://<honeypot_ip>

Cowrie captured every attempt correctly, verifying system reliability.

V. RESULT

The honeypot successfully collected and analyzed comprehensive attacker data.

Quantitative Findings:

- Over 12,000 login attempts recorded within 7 days.
- Top Usernames: root (35%), admin (22%), test (18%).
- Top Passwords: 123456, password, admin.
- Peak Attack Hours: 02:00–05:00 IST.
- Top Source Regions: Russia, USA, and China (via GeoIP lookup).

Qualitative Insights:

- Top Source Regions: Russia, USA, and China (via GeoIP lookup).
- Repeated command sequences indicated scanning and privilege escalation attempts.
- Logs contained traces of malware download commands (e.g., wget and curl), revealing common exploit strategies.

System Performance:

- Average log parsing time < 2 seconds per MB.
- Real-time analysis mode processed > 10 events/sec.
- Python analyzer successfully visualized datasets of > 100k records without failure.

User Evaluation:

Security students and testers reported the tool as intuitive, effective for training and visualization, and valuable for understanding attack evolution.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 1, November 2025 Impact Factor: 7.67

VI. CONCLUSION

The project demonstrates the effectiveness of honeypots in collecting real-world attacker data and transforming it into actionable intelligence. By integrating Cowrie with Python-based analysis tools, this system provides a scalable and educational platform for studying cyberattacks.The modular architecture allows integration with external SIEM tools like ELK or Splunk for enterprise-level analysis.

Future Enhancements:

- Introduce Machine Learning models for anomaly and behaviour prediction.
- Integrate with EventBridge or AWS Athena for cloud-scale log queries.
- Develop real-time dashboards using Grafana or Kibana.
- Expand into a honeynet with multiple geographic nodes to compare attacker behaviour globally.
- Overall, the project contributes to the growing field of threat intelligence and proactive cybersecurity, providing both academic and practical insights into modern attack methodologies.

REFERENCES

[1] L. Spitzner, *Honeypots: Tracking Hackers*, Addison-Wesley, 2003.FinOps Foundation, "The State of FinOps 2022 Report," FinOps Foundation, 2022.
[2]. J. Huang and S. Lee, "Automated Log Analysis for Cyberattack Detection," Journal of Information Security Research, 2020.
[3]. M. Van Hauser, Hydra Password Cracker Toolkit, 2021.
[4]. S. Rao and D. Patel, "Predictive Analytics for Cybersecurity Log Data," IJARSCT, Vol. 5, Issue 9, 2023.
[5]. Cowrie Documentation, 2024, https://github.com/cowrie/cowrie
[6]. N. Banerjee et al., "Containerization for Secure Research Environments," IEEE Access, 2021.
[7]. FinOps Foundation, "The State of FinOps 2022 Report."
[8]. R. Mehta, "Security Automation in Cloud Systems," IJARSCT, 2024.
[9]. N. Gupta, "Machine Learning for Threat Detection," IEEE Access, 2023.
[10]. A. Patil and K. Deshmukh, "SSH Honeypot Deployment for Attack Analysis," International Journal of Cyber Research, 2024