Low-Interaction Honeypot for Suspicious Activity Detection in Café Wi-Fi Networks

Presented by:
Shaikh Muqtasida & Arya Gawit
DigiSuraksha Cybersecurity Internship Program – 2025

📄 Abstract

Public Wi-Fi networks, like those in cafés, are highly convenient but vulnerable to cyber threats such as network scanning, unauthorized access, and brute-force login attempts. In this project, we developed a lightweight honeypot using Python and Flask that simulates a café-style Wi-Fi login portal. The system does not authenticate any user but logs every login attempt — including username, password, IP address, and timestamp — for analysis. This low-interaction honeypot quietly monitors suspicious behavior without affecting normal users or requiring high system resources. Our tool offers a cost-effective and educational approach to monitoring threat activity on open networks.

📄 Problem Statement & Objective

Problem:
Café Wi-Fi networks often lack monitoring tools, making them easy targets for attackers using scanning tools or

probing techniques to find weak devices or services. Traditional cybersecurity systems are too complex or expensive for such public setups.

Objective:
To design and implement a simple, low-risk honeypot that:

- Mimics a café-style login page on port 8000
- Captures and logs login attempts
- Records attacker metadata (IP, timestamp, username, password)
- Provides a passive monitoring tool for public network environments

📄 Literature Review

We studied previous honeypot research and public network threats:

- “Honeypots: Concepts, Approaches, and Challenges” – IEEE, 2022
- “Using Deception for Cyber Defense in SMEs” – Springer, 2021
- “Low-Interaction Honeypots for IoT and Public Spaces” – Elsevier, 2023
- OWASP SQL Injection Attacks – owasp.org
- CISA Guidance on Insider Threats and Small Businesses – cisa.gov

These references highlight how honeypots can serve as quiet observers, helping detect early-stage threats without disrupting services.

📄 Research Methodology

Environment:

- Simulated café Wi-Fi network (localhost testing)
- Single-device test for safe proof-of-concept
- Flask web server run on port 8000

Tool Components:

- Flask web application (Python 3)
- HTML login form with two input fields (username and password)
- Logging logic using request.form and request.remote_addr
- “Access Denied” message served on form submission

Testing Process:

- Login attempts were manually made using a browser
- Credentials like “admin:123” or “arya:abc” were submitted
- Log entries were printed to terminal and recorded as text

📄 Tool Implementation

The honeypot is a Flask application running a fake login page. When a user submits credentials, it:

- Logs the username and password entered
- Captures the IP address of the user (request.remote_addr)
- Logs the timestamp (using datetime.now())
- Displays a message: “Access Denied. This activity has been logged.”

The goal is not to prevent login, but to detect someone interacting with a fake service that should never be used.

Example terminal output:

```
[2025-05-12 15:12:20] IP: 127.0.0.1 | Username: arya | Password: abc
```

This simulates real-world login probing or credential-stuffing attempts.

📄 Results & Observations

- All login attempts were successfully logged, including local tests
- Each log recorded time, IP, username, and password
- The system worked smoothly on both Mac and Windows using Python 3

- Multiple form submissions produced real-time log entries
- Example log entry:

```
[2025-05-12 15:12:20] IP: 127.0.0.1 | Username: guest | Password: test123
```

This demonstrates that even basic interaction with the system creates a valid log, supporting the honeypot's purpose as a behavioral trap.

📄 Ethical Impact & Real-World Relevance

Ethics:
This tool is designed for educational and detection purposes only. It does not exploit, track, or harm users. The data logged is anonymous and used only to demonstrate suspicious behavior in a simulated setup.

Real-World Use:
This tool could be useful in cafés, libraries, or coworking spaces to detect potential attackers or rogue users. It reflects real deception-based tools used in the industry (e.g., Thinkst Canary, The Honeynet Project), where fake services are deployed to monitor malicious behavior without affecting real users.

📄 Future Scope

- Add email or SMS alerts when login attempts occur
- Export logs to CSV or JSON for SIEM tools (e.g., Splunk)
- Deploy the honeypot on a Raspberry Pi for real café trials
- Introduce additional ports or fake services (SSH, FTP)
- Optionally expand to detect attack types like SQLi using input patterns

📄 References

- https://www.honeynet.org
- https://canary.tools
- https://owasp.org/www-community/attacks/SQL_Injection
- https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-115.pdf
- https://www.cisa.gov/news-events/news/insider-threats-and-small-businesses
- IEEE Xplore, SpringerLink, ScienceDirect, and ACM articles on honeypots and deception