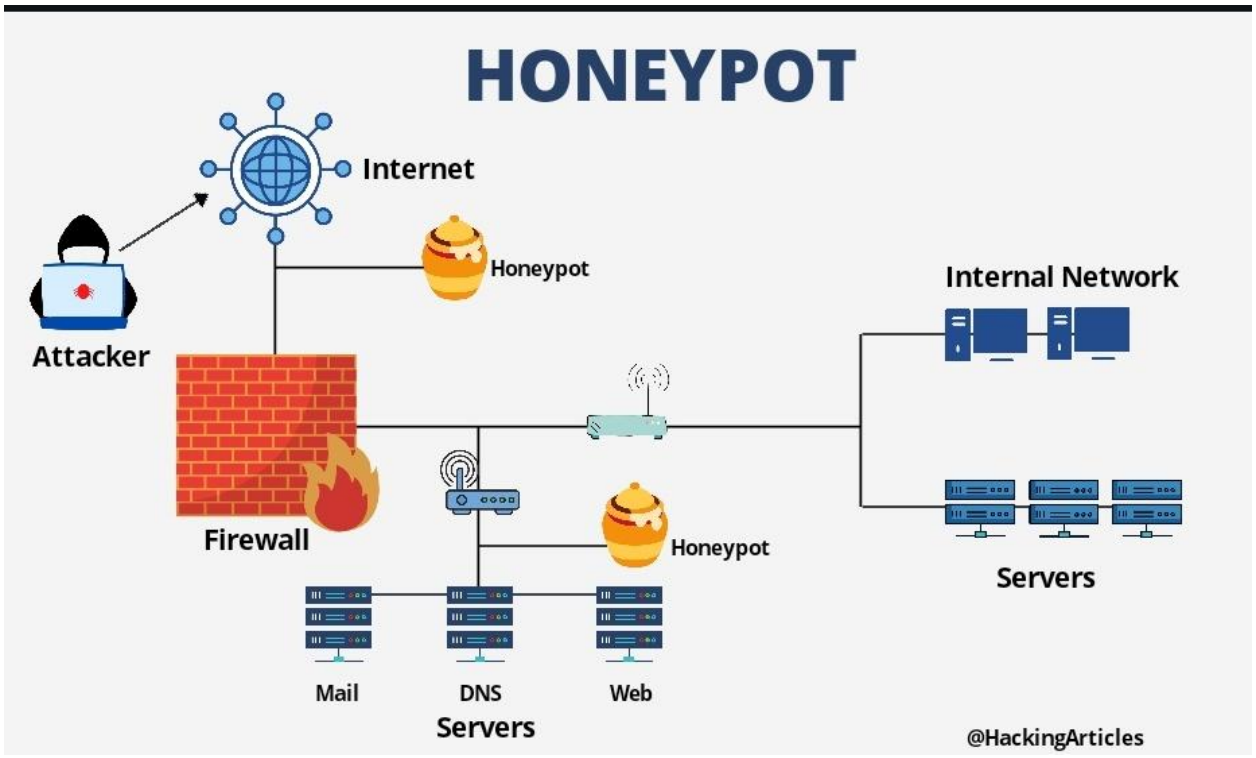
HONEY TRAP

Detect Log Activities

The honeyTrap is built using Flask, a lightweight Python web framework

Honeypot Project Report



Report on Honeypot

1. Introduction

A honeypot is a cybersecurity mechanism designed to detect, deflect, or study cyberattack attempts. It acts as a decoy system to attract attackers and monitor their activities. In this project, a Fake Admin Panel Honeypot was created using Flask to log unauthorized login attempts and gather intelligence on potential cyber threats.

2. Objective

The primary goals of this honeypot are:

- To capture unauthorized login attempts.
- To log attacker details, including IP address and User-Agent.
- To attract bots and malicious actors by simulating an admin panel.
- To analyze attacker behavior and commonly used credentials.

3. Implementation

The honeypot was implemented using Python and Flask. The following functionalities were developed:

- A fake login page (`/`) that mimics a real authentication system.
- A fake admin panel (`/admin`) designed to attract automated bots and attackers.
- Logging of IP addresses, usernames, passwords, and User-Agent information.

4. Code Summary

The honeypot is built using Flask, a lightweight Python web framework. Below are key implementation details:

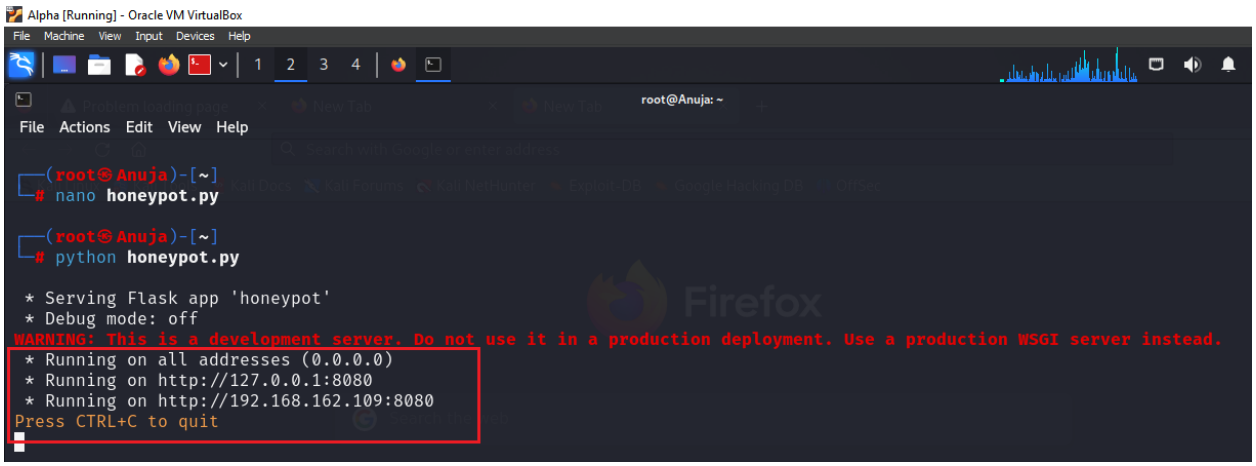
- The main page (`/`) provides a login form where user inputs are logged.
- The admin panel (`/admin`) acts as bait for attackers trying to gain privileged access.
- Logs are stored in a file (`honeypot.log`) for further analysis.

5. Data Collection and Logging

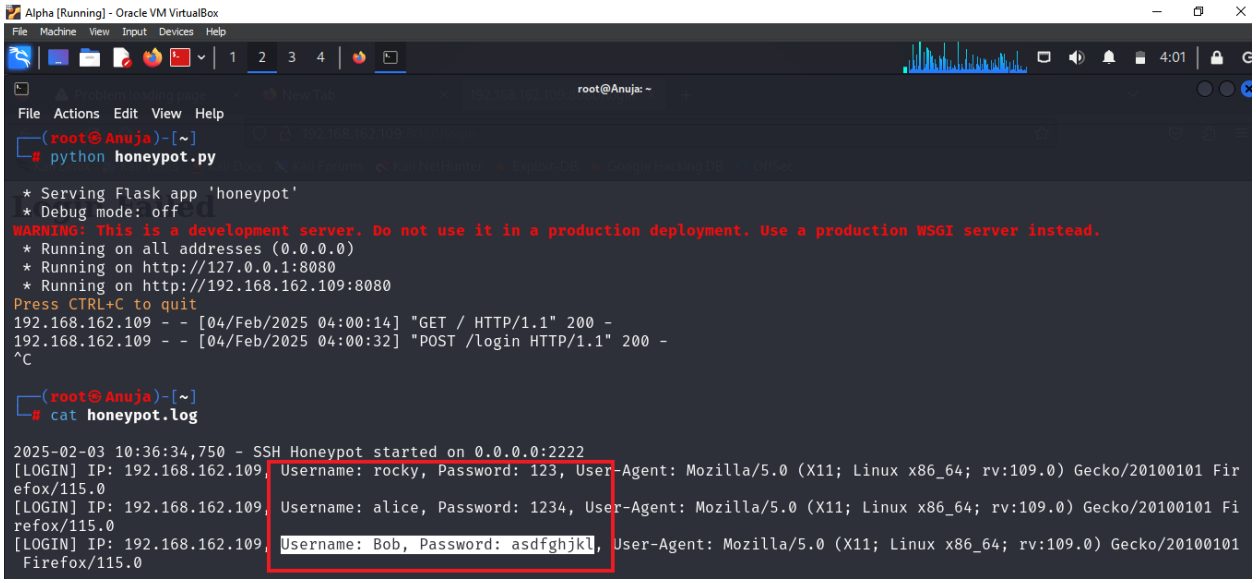
Each attack attempt is logged with the following information:

- **IP Address:** Identifies the attack source.
- **Username & Password:** Captures credentials used by attackers.
- **User-Agent:** Provides insights into the attacker's browser or automated tool.

- **Endpoint Accessed:** Distinguishes whether the attacker targeted the main login or the admin panel.



Sample Log Entry:



6. Results and Observations

During initial deployment, the honeypot successfully logged multiple unauthorized login attempts. Notable observations include:

- **Common usernames:** `admin`, `root`, `user`, `test`
- **Common passwords:** `123456`, `password`, `admin123`, `qwerty`
- **Automated attack tools:** Many attempts were made using `curl` and `Python-requests`, indicating bot activity.
- **Frequent access to `/admin` endpoint:** Suggests attackers are scanning for weak admin portals.

7. Potential Enhancements

To improve the honeypot, the following features can be added:

- Email or SMS alerts when multiple failed logins occur.
- Integration with a database for more structured logging.
- Deployment on a public server to attract real-world attacks.
- Visualization tools to analyze trends in attack patterns.

8. Conclusion

This Fake Admin Panel Honeypot successfully demonstrated how attackers attempt to gain unauthorized access. By logging attacker behavior, security teams can analyze threats and implement better defenses. The project serves as an excellent starting point for cybersecurity research and further honeypot development.