CHARLESTON SOUTHERN UNIVERSITY

Senior Design Project Proposal
Department of Computer Science
4/8/2024

---

**Student Name:** Zachary D. Weaver
**Degree and Major:** Cybersecurity
**Project Advisor Name:** Julie Henderson
**Expected Graduation Date:** May 5, 2025

## Problem Statement:

In recent years, there has been a significant increase in cyber based incidents that end in data theft through unauthorized access to compromised computer systems. Implementing modern cyber defense solutions must be done in order to strengthen security postures and prevent malicious activity. With an ever increasing frequency of sophisticated cyber attacks, personal computer systems are specifically at risk. One way these risks may be mitigated and avoided is through a careful analysis of possible attack strategies that may be implemented by a threat actor. This project aims to address this problem by creating a portable security mechanism that can be used to detect, counteract, and analyze cyberattacks directed at a personal system or network. This device will be configured to appear as a legitimate target on a network and can be used to both lure and trap attackers using what appears to be important information, services, and vulnerabilities.

## Project Description:

The HoneyPi will be a small, lightweight, portable honeypot IDS which can be used to capture important threat actor information. Its primary goal will be to gain a deeper understanding of intrusion detection systems and threat actor analysis. The hardware used to construct the Honey Pot will be a Raspberry Pi 4 unit to be purchased online. For honeypot applications, a decoy SSH server will be set up using both docker and python socket programming. This decoy server will be able to grant a threat actor access to a fake SSH session and emulate common commands used in the command line interface. Additional honeypot implementation will include a decoy SQL database that will hold what appears to be sensitive information. All emulated services and applications will be hosted in docker containers for a safe, isolated, and controlled environment. After any successful connection to the Honey Pot containers is made, the attacker's actions and strategies will be logged and sent back to a host machine for further examination and data

capture. Dashboard metrics will also be used to analyze common attack methods used. Due to the discreteness of the Raspberry Pi unit, the device will be portable and easily connected wirelessly to any available internet for reconnaissance and intrusion detection.

## Proposed Implementation Languages:

-Python
-SQL programming
-Various Unix-based shell scripting techniques

## Libraries, Packages, and Development Kits:

-Paramiko (python ssh implementation)
-Socket (Python library that allows client-host connections)
-Docker containerization (Docker Swarm)

## Additional Equipment Needed:

-Rasberry Pi 4 SBC Unit
- 64gb Micro SD Card
-Stable Internet Connection
-Host machine used to monitor honeypot status

## Personal Motivation:

Throughout my academic career, one of my greatest interests has been intrusion detection and prevention systems as they are a crucial aspect of cybersecurity. Moreover I have always been fascinated with physical pen testing tools that are often used by ethical hackers and penetration testers. I desire to grow deeper in my knowledge of intrusion detection systems so that I can aid others and myself against pertinent cyber threats. We live in a time where so many devices are vulnerable which is why I am creating the portable honey pot in order to aid myself and others in common attack method analysis.

## Outline of Future Research Efforts:

The portable Honey Pot will be created and developed primarily through configuration of a Raspberry Pi 4 unit and implementation of docker containers. The containers will be used to store different services posing as vulnerabilities to an attacker. For this project, I currently have little knowledge on the functionality of how these services will be hosted in docker. While I do have understanding on the vulnerable services that will be used, I look forward to learning more about container orchestration and load balancing to send both metrics and threat actor data to one centralized location for closer examination. Additionally, I plan on extensively learning more about python scripting, socket programming, and all of the possible libraries that could be used theren.

# Project Schedule:

| *Task to be Completed* | *Description* | *Due Date* |
|---|---|---|
| Project *Proposal* Deliverable:<br>**Problem statement.** | A brief overview of the problem that this project aims to solve. | **1/26/24** |
| Project *Proposal* Deliverable:<br>**-Project Description.**<br>**-Languages, Libraries, and equipment.**<br>**-Personal Motivation, outline of future efforts.** | -A brief description of the project itself and how it will function.<br><br>-Research into any programming languages, software libraries, and additional hardware that will be used.<br><br>-Explain how this project will further my knowledge in the future and any possible expansion. | **2/23/24** |
| Project *Requirements* Deliverable:<br>**Sections 1-3** | -Functionality<br>-Look & Feel<br>-Usability | **3/15/24** |
| Project *Requirements* Deliverable:<br>**Section 4** | -Performance | **3/15/24** |
| Project *Requirements* Deliverable:<br>**Section 5-7** | -Maintainability & Support<br>-Security<br>-Cultural | **3/22/24** |
| Raspberry Pi Construction & Initial Configuration | Purchase and build the raspberry pi. Setup for further prototyping | **3/29/24** |
| Prototyping | Setup docker containers on the Raspberry Pi to run honeypot services. Begin prototyping | **3/29/24** |

| | | |
|---|---|---|
| Project *Proposal* Deliverable: **Final Draft**<br>Project *Requirements* Deliverable: **Final Draft** | -Final draft of the project proposal document.<br><br>-Detailed specifications that must be met for the project to be successful. | **4/5/24** |
| Configure **SSH** on raspberry pi | | **9/01/24** |
| Configure **docker** on raspberry pi | | **9/01/24** |
| Build SSH container | | **10/01/24** |
| *Decoy SSH Server*:<br>**SSH server creation/interaction** | -Build docker file for the container<br><br>-Use python socket programming to establish client-server connections<br><br>-Use paramiko library to build the decoy server and create a fake ssh session | **9/30/24** |
| *Decoy SSH Server:*<br>**Logging/emulating commands** | Incorperate logging client information and emulated commands to fake SSH session | **10/09/24** |
| *Decoy SSH Server:*<br>**Test Container Connection** | Host the decoy server in a docker container and test connectivity/logging capability | **10/09/24** |
| *Decoy SSH Server:*<br>**Debugging/Automatic Start** | Create systemd service file to start the docker container whenever the Raspberry pi is powered on | **10/18/24** |

| | | |
|---|---|---|
| *SQL Database:*<br>**Run SQL in the docker container** | Create the docker container that will hold the database | **11/01/24** |
| *SQL Database:*<br>**Populate Database** | Use SQL commands to manually enter fake information into the database | **11/8/24** |
| *SQL Database:*<br>**Logging, Filtering, and Analysis** | Create logging mechanism to record queries used to access the database. (Use scrips to filter the logs for specific patterns?) | **11/15/24** |
| Configure container orchestration | Scrape the containers for data such as active connections, administered payloads, attempted logons, etc.. | **12/07/24** |
| Build dashboard web application (grafana?) | | **12/07/24** |
| Send container metrics to grafana (prometheus) | | **12/07/24** |
| **TBD** | **---** | **---** |
| **TBD** | **---** | **---** |
| **TBD** | **---** | **---** |
| Implement light matrix to visualize active connections made to the containers | **TBD** | **TBD** |