Zachary D. Weaver - Senior Design Project Requirements
Department of Computer Science
4/8/2024

Table of Contents

Portable Honey Pot

The portable honey pot will be a small, lightweight, deployable security mechanism which can be used to capture important threat actor information. With it's primary goal being to gain a deeper understanding of intrusion detection systems, the honeypot will be able to trap attackers and view their actions. The hardware used to construct the device will be a raspberry pi 4 unit which is to be purchased online. To begin, a decoy ssh server will be setup using both docker and python socket programming. This decoy server which is to be hosted inside a docker container will be able to grant a threat actor access to a fake ssh session and emulate common commands used in the command line interface. Commands executed by the attacker as well as other valuable attacker information will logged for further inspection. Additional honeypot implementation will include a decoy sql database that will hold what appears to be sensitive information. After any type of successful connection to the honey pot is made, the clients actions and strategies will be logged and sent back to a host machine for further examination and data capture. Dashboard metrics will additionally be used to view the status of containerized applications and system information. Due to the discreteness of the raspberry pi unit, this device will be lightweight, portable, and able to be easily connected wirelessly to any available internet for intrusion detection and threat analysis.

Description

This document will list and describe all the requirements for the honey pot device while also representing a complete description of the project. Each requirement will include a requirement number and type, along with a description, rationale, fit criterion, priority and any other dependencies. The types of requirements included on this document will be functional, look and feel, usability, performance, maintainability and support, security, and cultural. A priority will also be set on a scale from 1 to 10 to highlight the importance of the requirement to the device being created.

Research to be conducted:

- Configuring decoy ssh server inside a container
- what specific linux OS to place in the containers
- Container optimization and automation
- Sending container metrics and info to a dashboard
- Container load balancing through orchestration (Docker Swarm/Kubernetes/etc…)

Requirements - *Functional*

Requirement ID: **01**

Description:
The device will be able to emulate real world services and protocols.
Rationale:
Service and protocol emulation has the ability to attract a wide variety of attackers.
Fit Criterion:
The Services used for the honeypot will be compared to their real-world counterparts.
Priority: 10

Dependencies: none

Requirement ID: **02**

Description:
The device will automatically connect to an available wireless network.
Rationale:
Network connectivity wil be a quick and easy way to allow the device to be discoverable by attackers and connected to by the host.
Fit Criterion:
Automated script will be ran to connect the device to an available network (light module will show active connection)
Priority: 10

Dependencies: none

Requirement ID: **03**

Description:
Docker will be installed and configured on the raspberry pi.

Rationale:
Using docker containers is an efficient way to isolate vulnerable servers while conserving system resources.

Fit Criterion:

User will verify successful installation (“docker ps, docker info”)

Priority: 10

Dependencies: none

Requirement ID: **04**

Description:

Open SSH Service will be installed and configured on the raspberry pi.

Rationale:

SSH will be used to configure and maintain the honeypot remotely.

Fit Criterion:

User will verify successful installation (“ssh -v”)

Priority: 10

Dependencies: none

Requirement ID: **05**

Description:

Raspberry pi resource usage will be optimized to ensure that the containers can operate effectively on the unit.

Rationale:

Optimizing resources will allow the containers to be deployed safely in various environments

Fit Criterion:

Critical active processes such as SSH and Docker will be prioritized by the CPU

Priority: 8

Dependencies: 03, 04

Requirement ID: **06**

Description:

The raspberry pi will be able to be remotely accessed from a trusted client machine.

Rationale:

Remote access will allow for safe configuration and authenticated access to monitor the docker containers.

Fit Criterion:

User will successfully be logged into raspberry pi admin account from a remote client machine

Priority: 10

Dependencies: 02, 05

Requirement ID: **07**

Description:

The device will have security measures in place including access controls and intrusion detection mechanisms.

Rationale:

Security measures will protect the raspberry pi from total exploitation.

Fit Criterion:

Raspberry pi will not be accessible to an attacker

Priority: 10

Dependencies:

Requirements - *Look and Feel*

Requirement ID: **08**

Description:

The device will be discrete and placed inside a robust outer case for easy transportation.

Rationale:

Having a small, lightweight chassis will promote quick and easy access for a more pleasant user experience.

Fit Criterion:

Case will be purchased online and built during initial construction

Priority: 5

Dependencies: none

Requirement ID: **09**

Description:
The containers will appear as a vulnerable targets on a specified network.
Rationale:
By seeming vulnerable, attackers are more likely to try and connect to the honey pot.
Fit Criterion:
Scanning for open ports and services will return controlled vulnerabilities in the containers.
Priority: 10

Dependencies: none

Requirements - *Usability*

Requirement ID: **10**

Description:
The device and its services services will have automatic startup enabled upon powering on.
Rationale:
Being able to start the honeypot as soon as the device is powered on will ensure that the device is quick, easy to use, and user-friendly.
Fit Criterion:
Docker container status will be monitored initially by the user through raspberry pi remote connection.
Priority: 9

Dependencies: 03, 04

Requirement ID: **11**

Description:
A guided setup checklist/process will be provided to allow the user easy deployment in their environment.
Rationale:

This will make it easier for the user to change or update any specific configuration files when needed.

Fit Criterion:

Checklist will be provided in github project repository

Priority: 7

Dependencies: none

Requirement ID: **12**

Description:

The device will automatically start docker containers hosting emulated services (ssh, sql, etc).

Rationale:

Using docker containers is an efficient way to isolate vulnerable servers while conserving system resources.

Fit Criterion:

Containers will be individually built and maintained by the user

Priority: 10

Dependencies: 03, 10

Requirement ID: **13**

Description:

Dashboard metrics will be available to monitor the devices status and alert of any suspicious activities via container orchestration.

Rationale:

Viewing logs and metrics will allow the user to effectively analyze any possible attacks

Fit Criterion:

Container metrics will be sent through orchestration service

Priority: 9

Dependencies: 03

Requirement ID: **14**

Description:
The device will provide error handling and feedback messages to alert the user if something is not working properly.

Rationale:
Informative error messages and handling will be useful in case there are any configuration errors or problems encountered during they honeypot setup or operation.

Fit Criterion:
Errors will be viewed either on the raspberry pi itself or on a dashboard with other metrics.

Priority: 9

Dependencies: 13, 12, 09, 02

Requirement ID: **15**

Description:
The device will be able to monitor network traffic.

Rationale:
Monitoring will allow for the detection of attacker connection attempts.

Fit Criterion:
Any connection attempts will be recorded through logging capabilities for inspection

Priority: 9

Dependencies: 12, 03, 02

Requirement ID: **16**

Description:
The device will be able to log successful connection attempts, commands entered, and payloads administered throught the docker containers.

Rationale:

Logging and reporting to the raspberry pi will allow for report generation to analyze attacker data.

Fit Criterion:

Loggs will provide sensitive attacker data that can be used for future use

Priority: 9

Dependencies: 15

Requirements - *Performance*

Requirement ID: **17**

Description:

The raspberry pi will be equipped with an on board fan and heat sinks for cooling

Rationale:

Keeping the system cool will improve hardware efficiency

Fit Criterion:

On board fan and cooling will be purchased online and built onto the device

Priority: 5

Dependencies: 08

Requirement ID: **18**

Description:

There will be little to no latency or time delay when connecting to one of the docker containers

Rationale:

This will ensure the attacker believes he/she is accessing a legitimate target system

Fit Criterion:

The attacker will experience minimal delay when attacking to the honey pot

Priority: 8

Dependencies: none

Requirement ID: **19**

Description:
Power management settings will be optimized to minimize unnecessary background processes and conserve battery life
Rationale:
Emphasizing power efficiency will make it easer to accurately use the device in portable deployment settings
Fit Criterion:
Battery life and device usage will be prolonged
Priority: 8

Dependencies: none

Requirement ID: **20**

Description:
Container load balancing will be setup to handle multiple connections that are being made
Rationale:
Ensure high availability to the honeypot services that are being hosted in the containers
Fit Criterion:
Load balancing will be made possible through a containerized orchestration service such as Docker Swarm or Kubernetes

Priority: 8

Dependencies: 03, 02

Requirements - *Maintainability & Support*

Requirement ID: **21**

Description:
A full description of how to setup and run the device will be available in the github repository

Rationale:
Making the project open source allows anyone the opportunity to recreate the device for their own use

Fit Criterion:
All project documentation and files will be uploaded to github
Priority: 5

Dependencies: none

Requirement ID: **22**

Description:
Any information on updates or recent developments made for the device will be available via github

Rationale:
Allows the device to be properly maintained and sustainable
Fit Criterion:
All device updates will be properly tested before being uploaded to blackboard
Priority: 5

Dependencies: 21

Requirements - *Security*

Requirement ID: **23**

Description:
SSH will be configured to a different port on the raspberry pi
Rationale:
Switching SSH to a different port will strengthen the security posture of the raspberry pi ensuring that attackers cannot access the container configurations
Fit Criterion:
SSH connectivity will be tested from users client machine to ensure ssh is properly configured
Priority: 9

Dependencies: 04

Requirement ID: **24**

Description:
A secure firewall will be configured on the raspberry pi to restrict incoming and outgoing network traffic

Rationale:

Specific firewall rules will only allow necessary ports and services that the honey pot needs to function.

Fit Criterion:

Tests will be conducted on trying to connect to the pi through closed ports and services

Priority: 10

Dependencies: none

Requirement ID: **25**

Description:

Docker containers will be ran with minimal privileges (ie. non-root users)

Rationale:

Strengthening the security of the individual containers will allow them to only be vulnerable where needed for honeypot functionality

Fit Criterion:

Root user access inside the containers will be inaccessible to attackers

Priority: 8

Dependencies: 03

Requirement ID: **26**

Description:

Each container will have its own firewall with specific configurations and vulnerabilities

Rationale:

Will allow the threat actor access to the containers simulating a real-world vulnerability

Fit Criterion:

Test will be conducted on trying to connect to the containers

Priority: 8

Dependencies: 01, 02, 03

Requirement ID: **27**

Description:
The device will have alerting mechanisms to notify the admin of any potential security incidents
Rationale:
Alerting mechanisms will help by ensuring no suspicious events inside the containers go unnoticed
Fit Criterion:
Alerting will be done through container orchestration and sent to the dashboard metrics

Priority: 8

Dependencies: 26, 20, 16, 14, 12

Requirement ID: **28**

Description:
Decoy ssh server and sql database will divert attackers attention to the vulnerable containers
Rationale:
Using enticing bait and deception techniques will enhance the honeypots effectiveness
Fit Criterion:
Tests will be conducted on the decoy services being hosted inside the machines
Priority: 10

Dependencies: 03, 04

Requirements - *Cultural*

Requirement ID: **29**

Description:
The purpose of this device is to gain a deeper understanding of possible cyber threats and threat actor strategies
Rationale:
Analyzing common attack vectors can provide valuable feedback for further system hardening

Fit Criterion:

When powered on, the honey pot will successfully log all connection attempts, and attacks made by any attacker

Priority: 10

Dependencies: none

Requirement ID: **30**

Description:

Continuous project expansion will allow for an ever growing list of attack strategies and research conducted from using the honeypot

Rationale:

The device has the ability to expand utilizing more docker containers and honeypot techniques in the years to come

Fit Criterion:

All project advancements will be tested before implementation and uploaded in the updates/development section in the githib repository

Priority: 5

Dependencies: 03, 21, 22
