Project 1 – Basic Vulnerability Assessment for small Business Network

Group Number - 21

Team Members :

Ranjith M C

Akash V

Sanjana Mondal

Jaishri Mahalia

Project 1 – Basic Vulnerability Assessment for small Business Network

Introduction

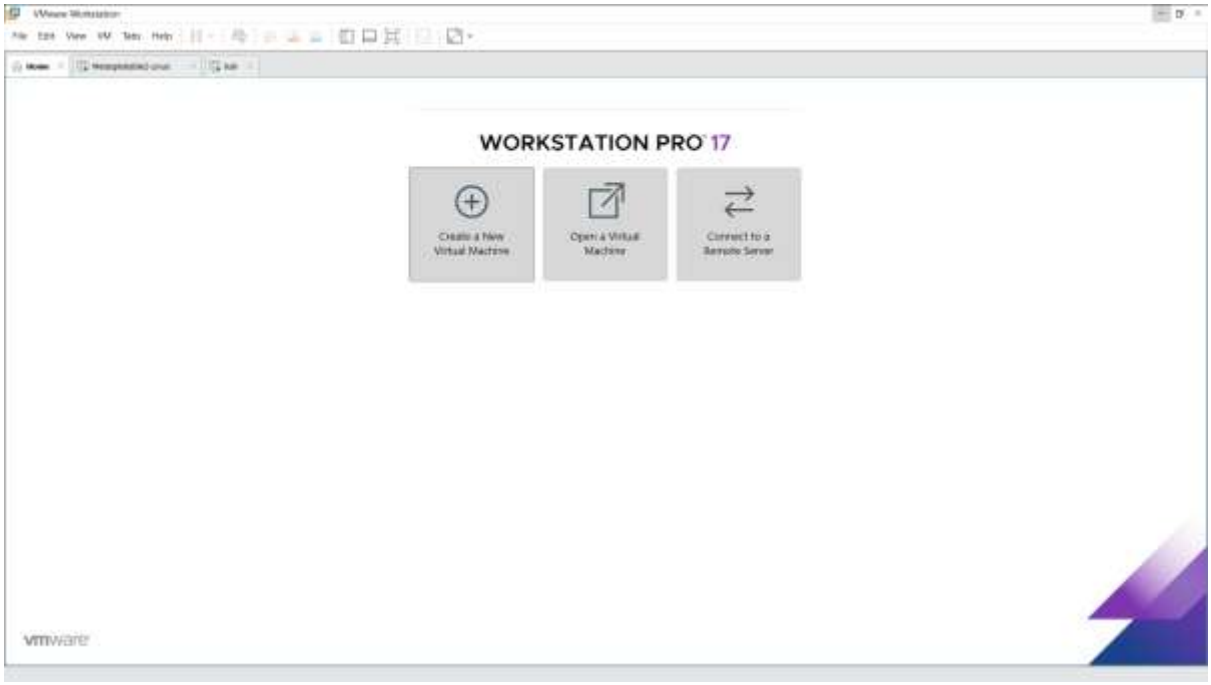
Project Overview

This project involves conducting a basic vulnerability assessment for a small business IT network. The goal is to identify security gaps, prioritize risks, and provide mitigation strategies to improve the overall security posture.

Set Up Virtual Lab

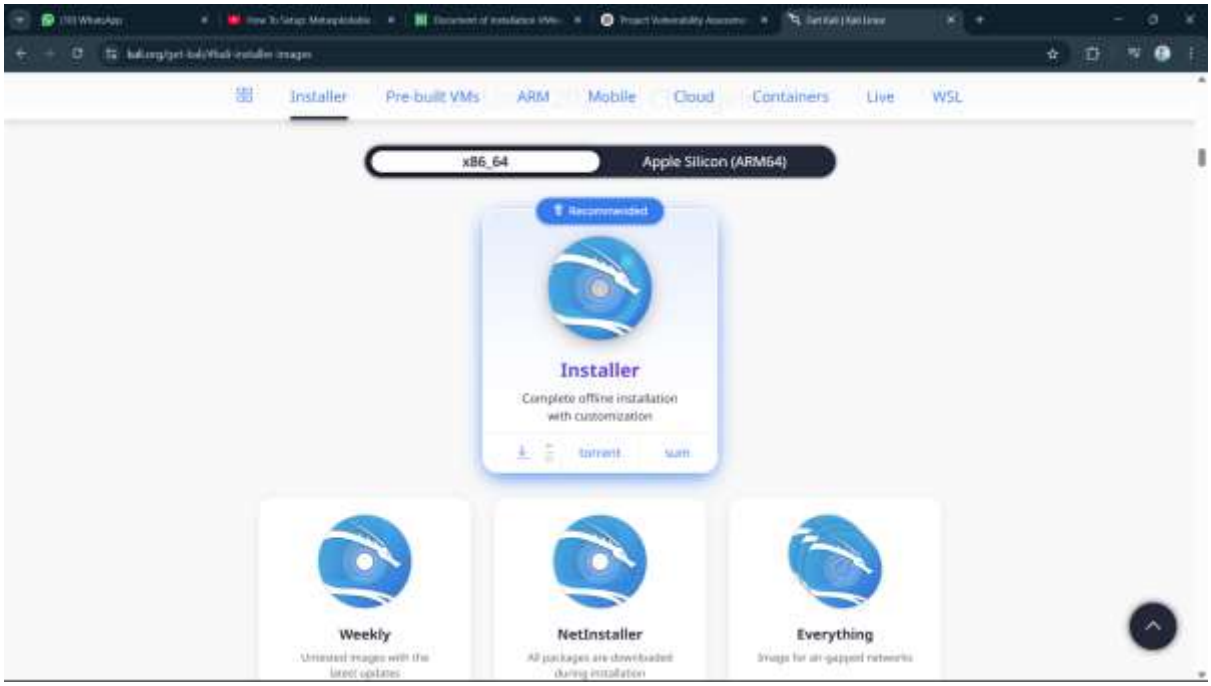
1. Installation of VMware Workstation

- Download VMware Workstation from the official website.
- Run the installer and follow the on-screen instructions.
- Accept the terms and conditions and complete the installation.
- Launch VMware Workstation.



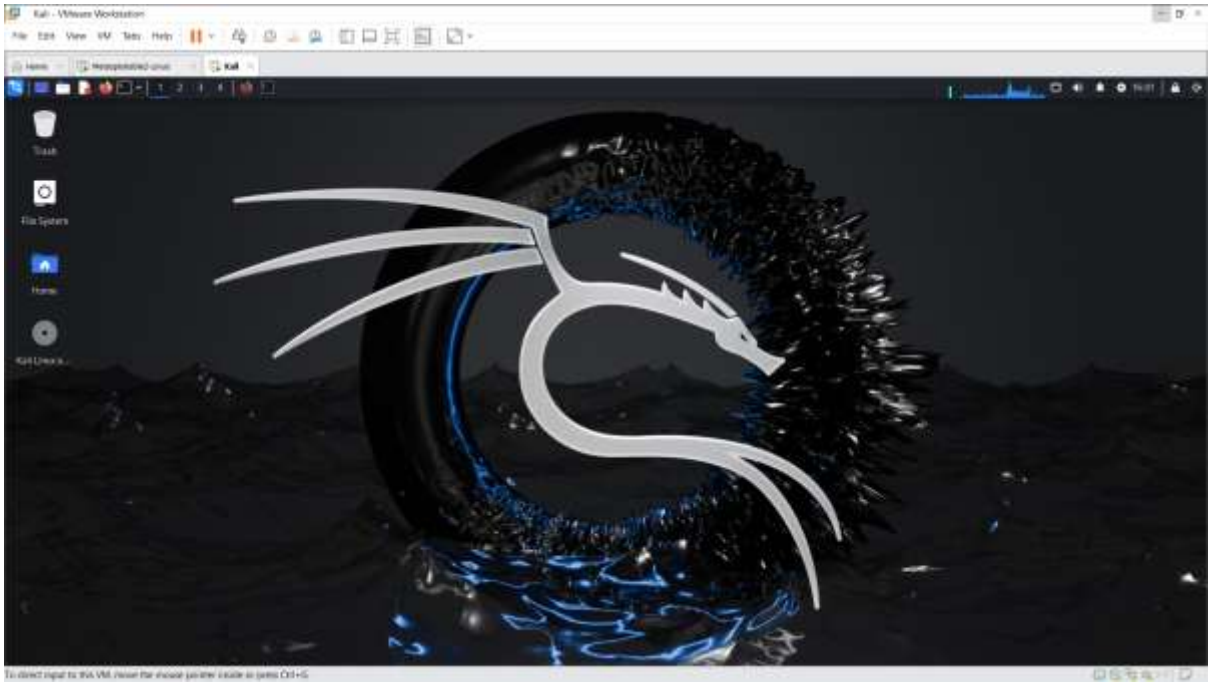
2. Installation of Kali Linux

- Download the Kali Linux ISO file from the official website https://www.kali.org/get-kali/#kali-installer-images



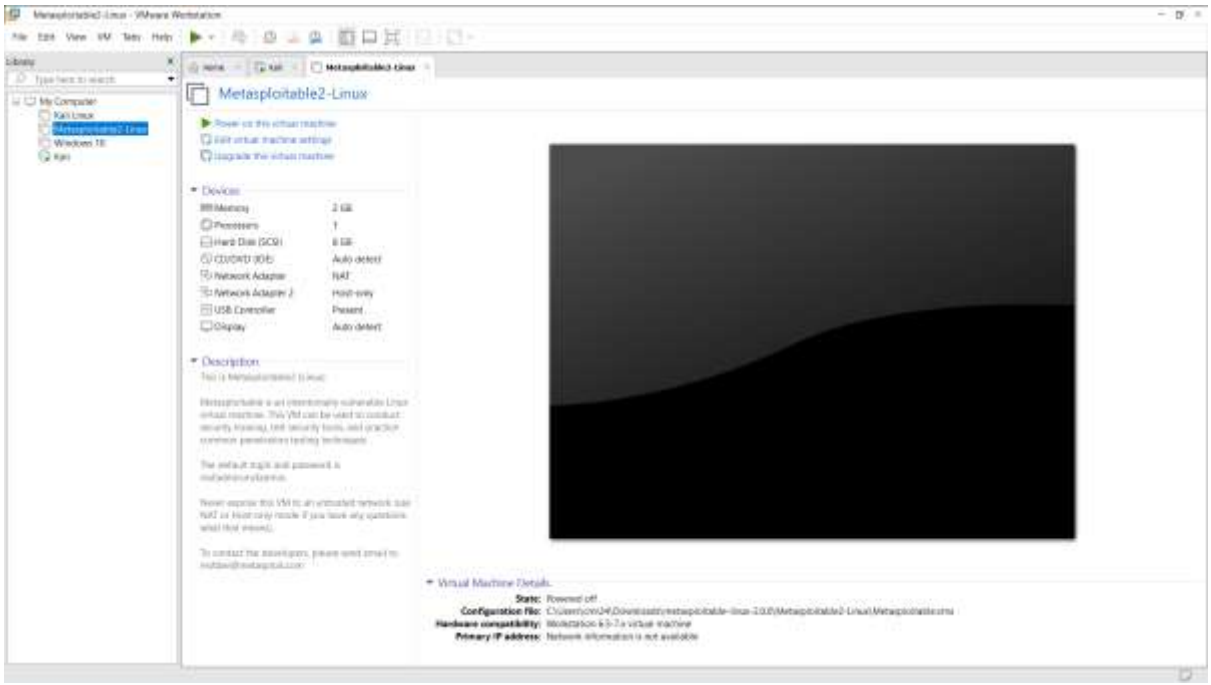
- Create a new virtual machine in VMware Workstation.
- Select the downloaded ISO file during the setup.
- Allocate sufficient RAM and disk space.

- Start the virtual machine and follow the installation steps.
- Set up a username and password during installation.
- Then reboot the Kali Linux and enter the username and password and click enter.



3. Installation of Metasploitable2

- Go to the official Rapid7 Metasploitable page. Download the Metasploitable2 VMware image .zip file.
- Extract the downloaded file.
- Open VMware Workstation and select "Open a Virtual Machine."
- Browse to the extracted file and add it to VMware.
- Start the virtual machine.



- Log in to Metasploitable2 the default credentials are:
  1. Username: **msfadmin**

2. Password: **msfadmin**

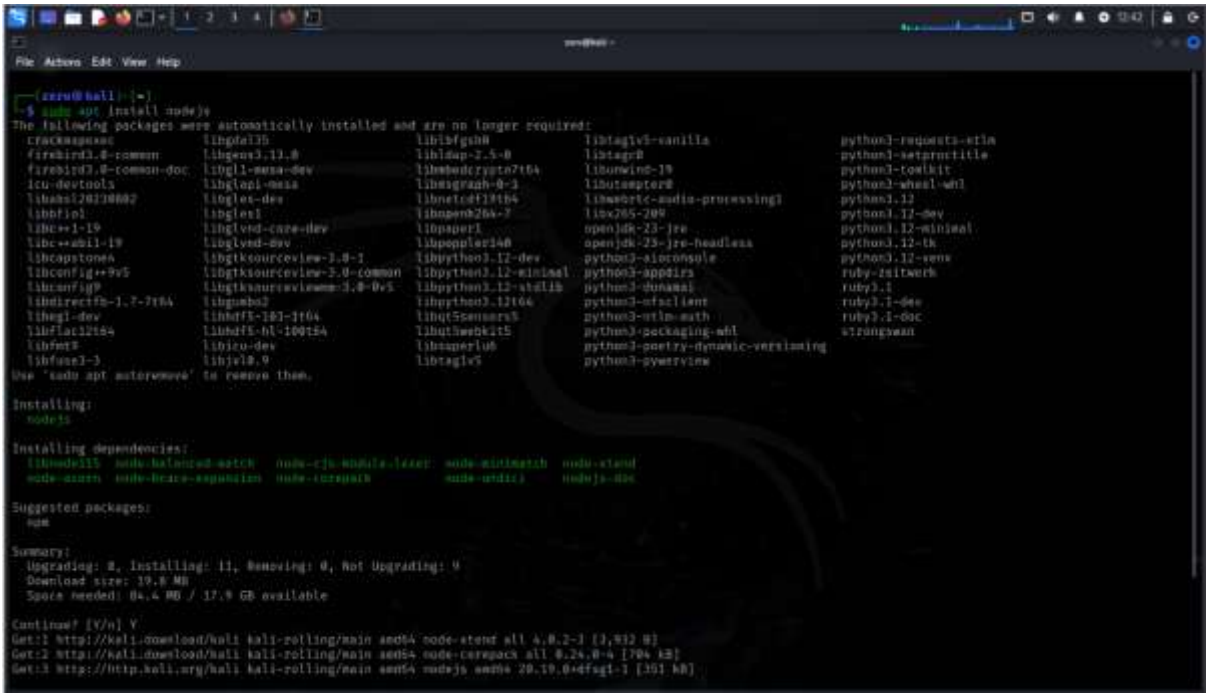
Installation of OWASP Juice Shop in Kali :

Step 1: Update the packages

- Open a terminal in Kali Linux.
- Update the package list using the command:
  sudo apt update && sudo apt upgrade

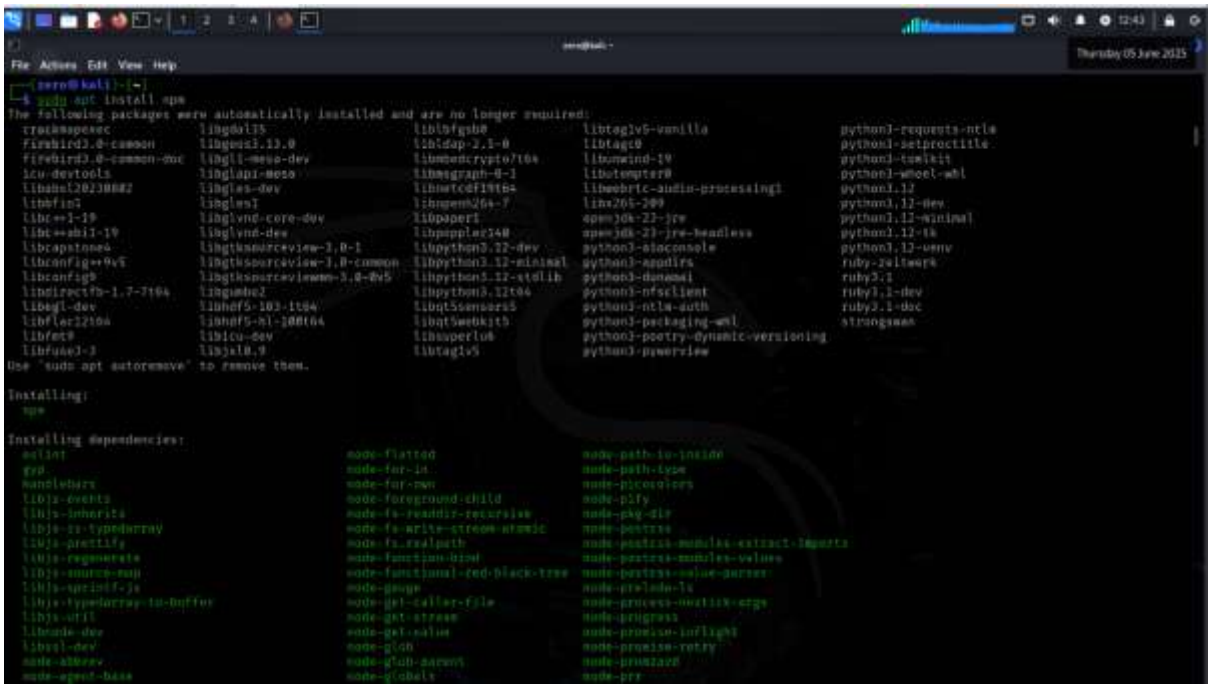
Step 2: Install node.js & npm Dependencies

- Install Node.js using the command :
  sudo apt install nodejs

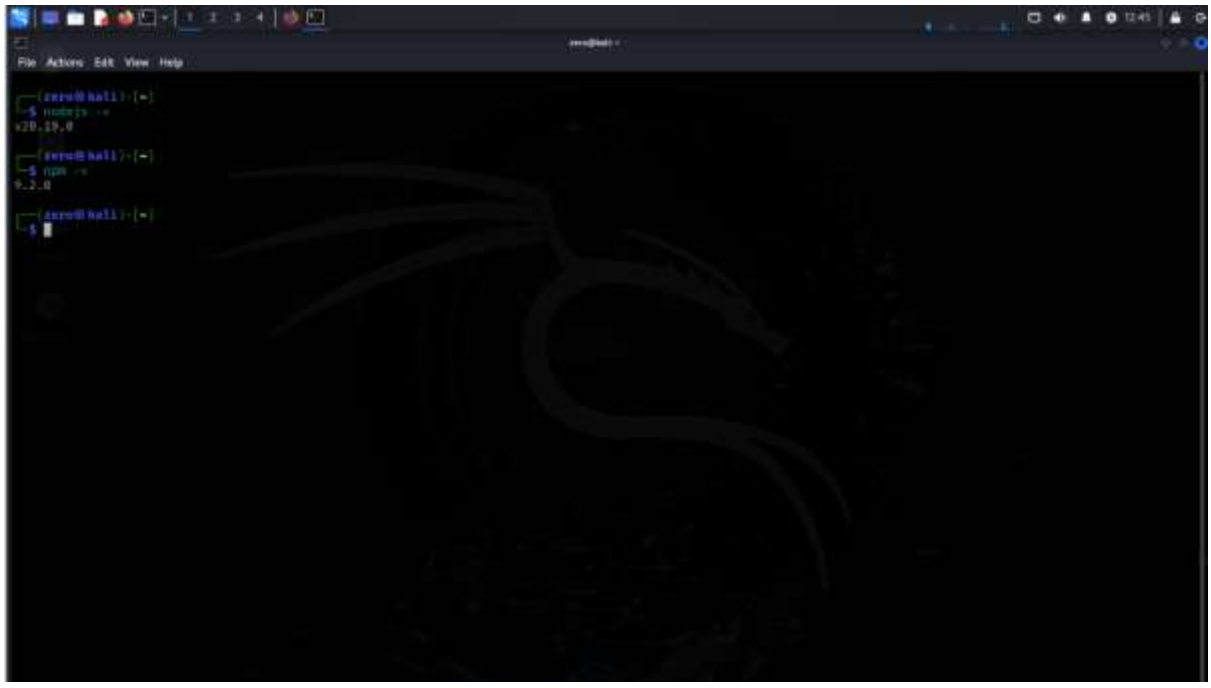


- Install npm using the command :

  sudo apt install npm



- Verify installation with **node -v** and **npm -v**.

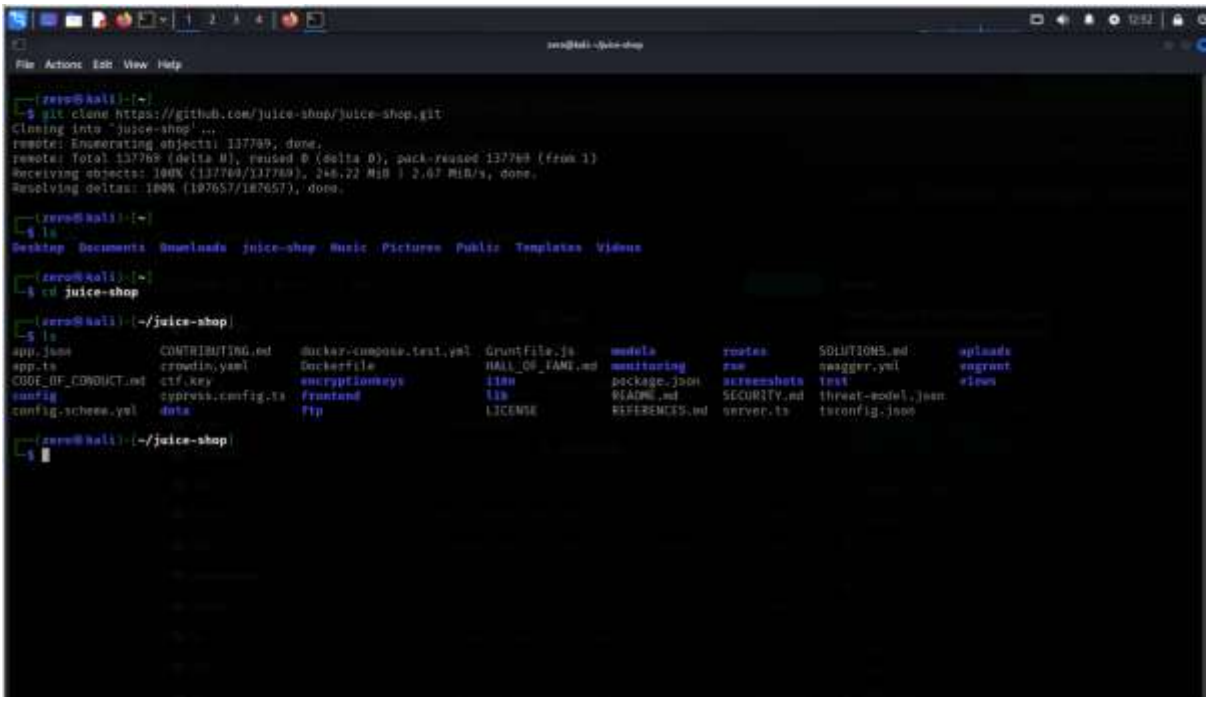


Step 3: Clone the Juice Shop Repository

- Use Git to download the official OWASP Juice Shop project:

  git clone https://github.com/juice-shop/juice-shop.git

  cd juice-shop

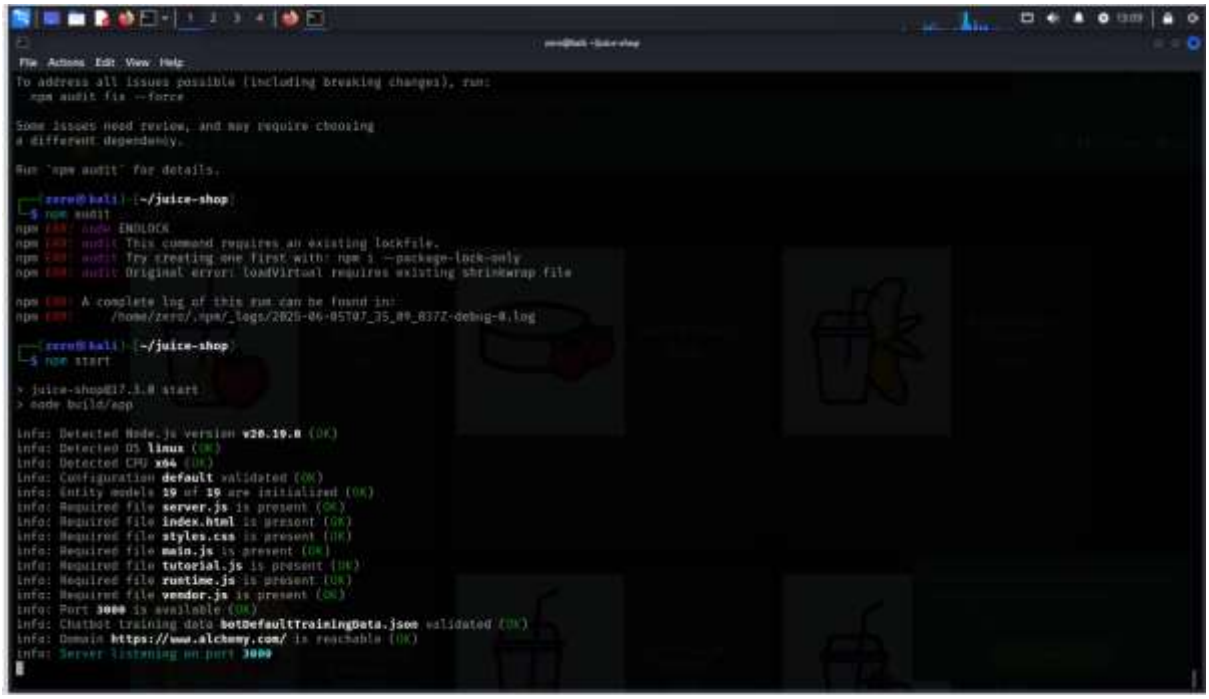


Step 4: Start the Juice shop server

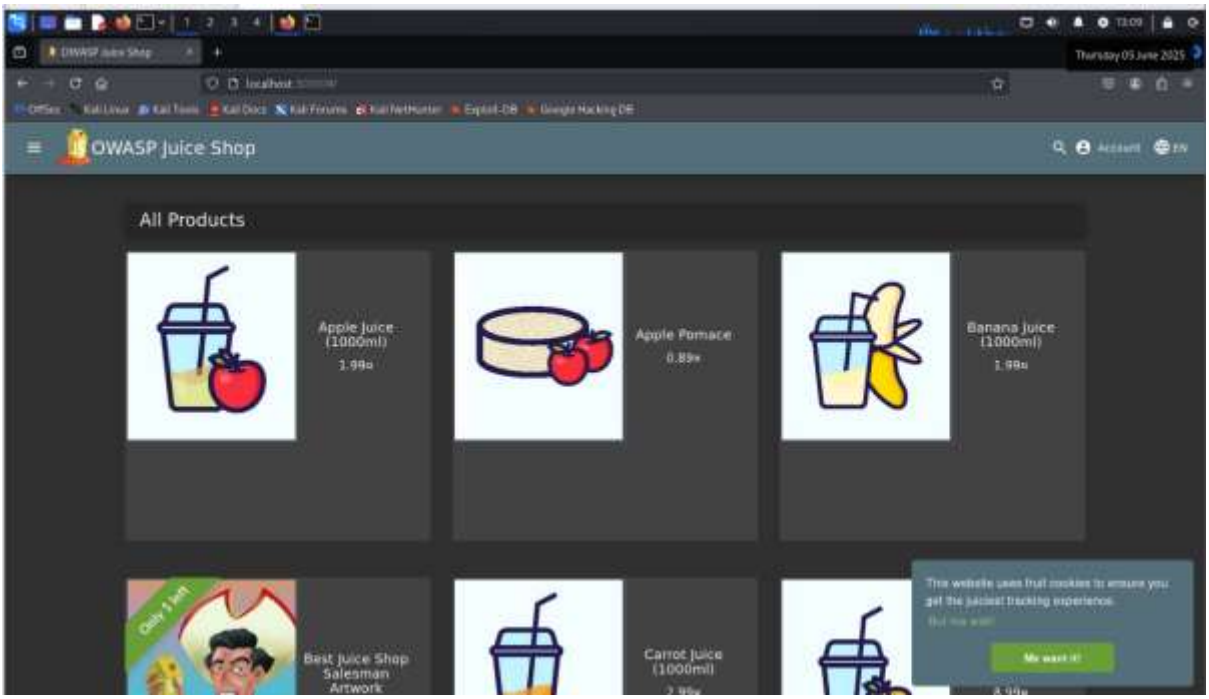
- Enter the following command

  npm install

  Start the application: **npm start**.



- Server initialized on port 3000 and access Juice Shop in a browser at http://localhost:3000



Installing OpenVAS on Kali Linux :

Step 1: Update and Upgrade System

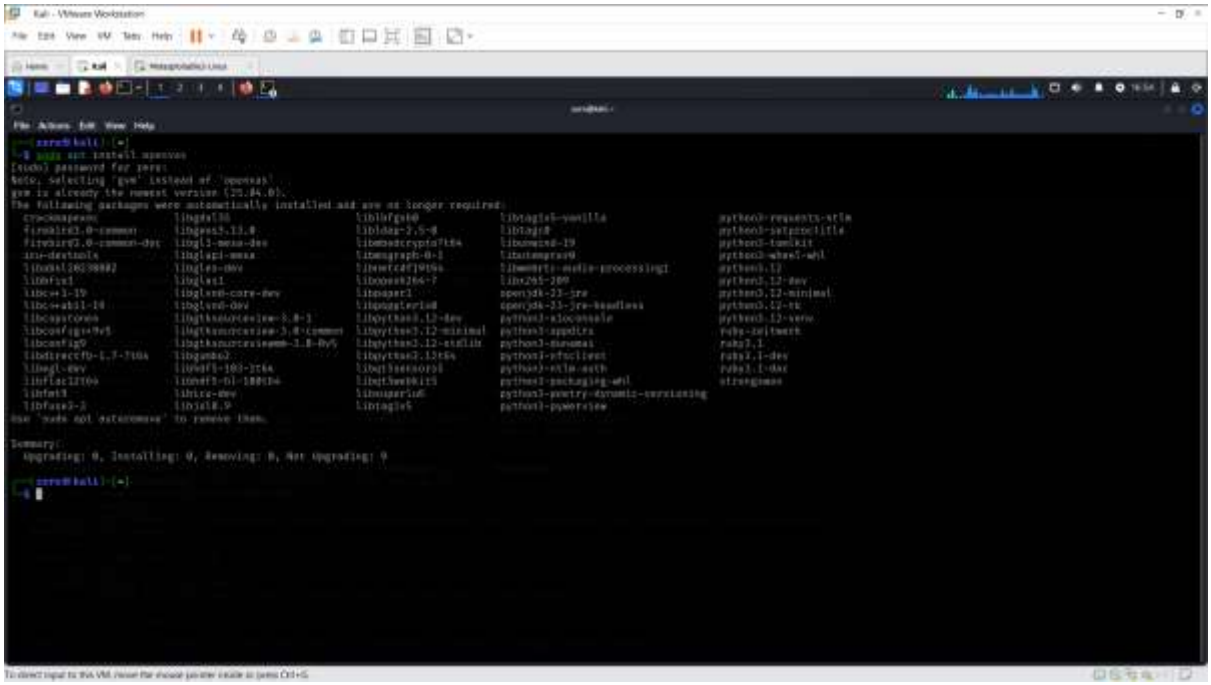
- Ensure your system is up to date:

  sudo apt update && sudo apt upgrade -y

Step 2: Install OpenVAS

- Install OpenVAS by running the following command:

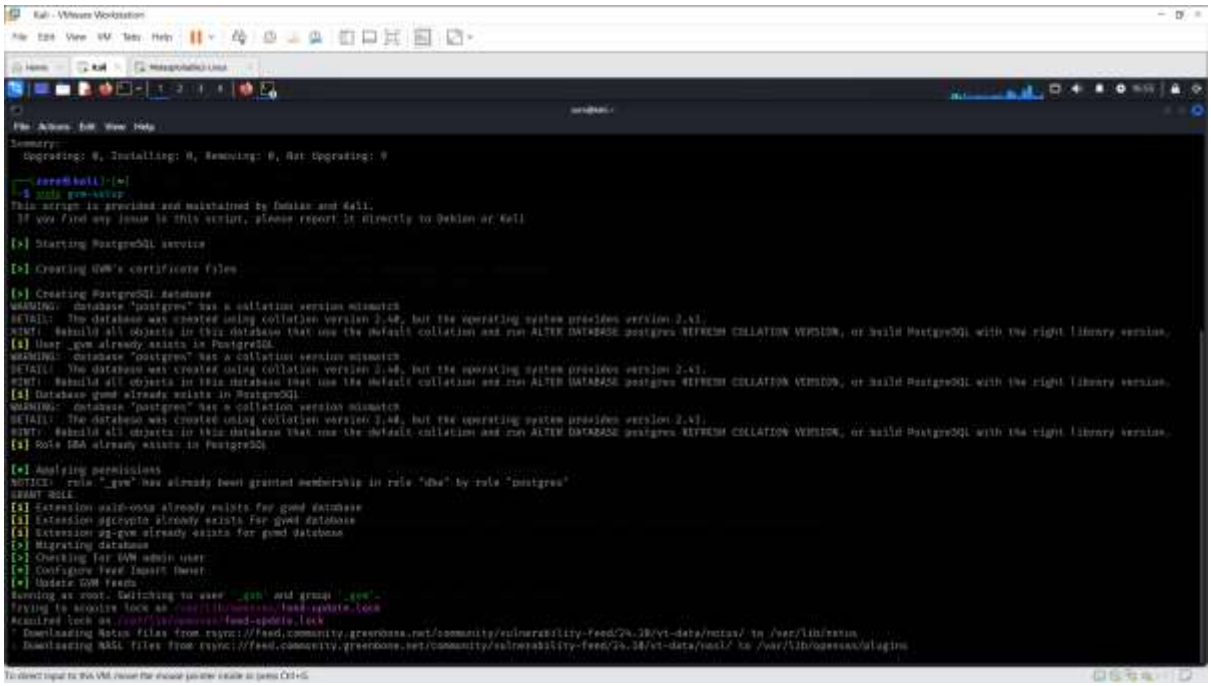
  sudo apt install openvas



Step 3: Initialize OpenVAS

- Initialize OpenVAS for the first time. This will set up the databases and services:

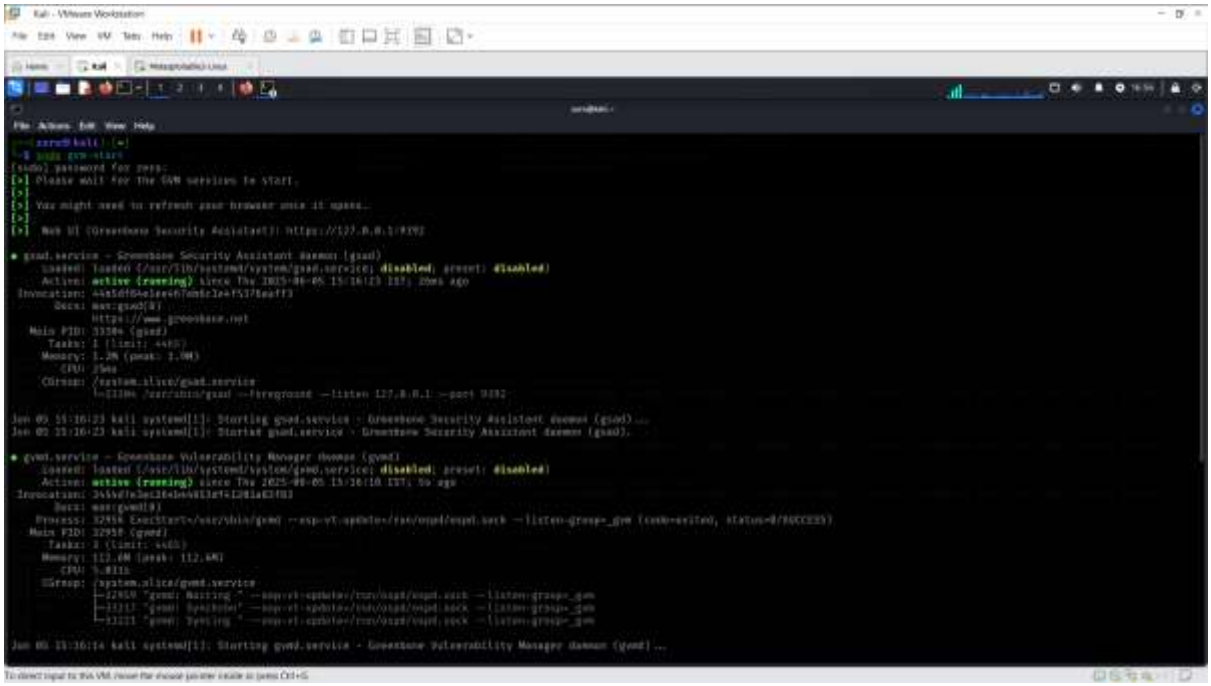
  sudo gvm-setup



Step 4: Start OpenVAS Services

- Start the OpenVAS services:

  sudo gvm-start

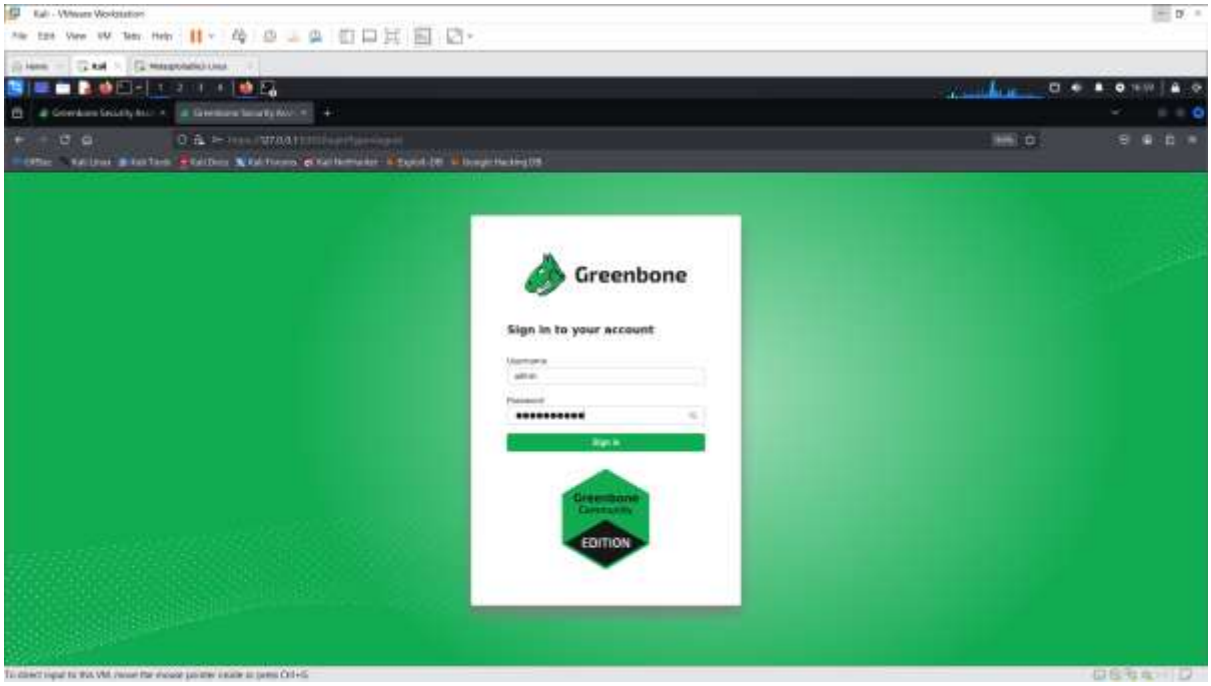


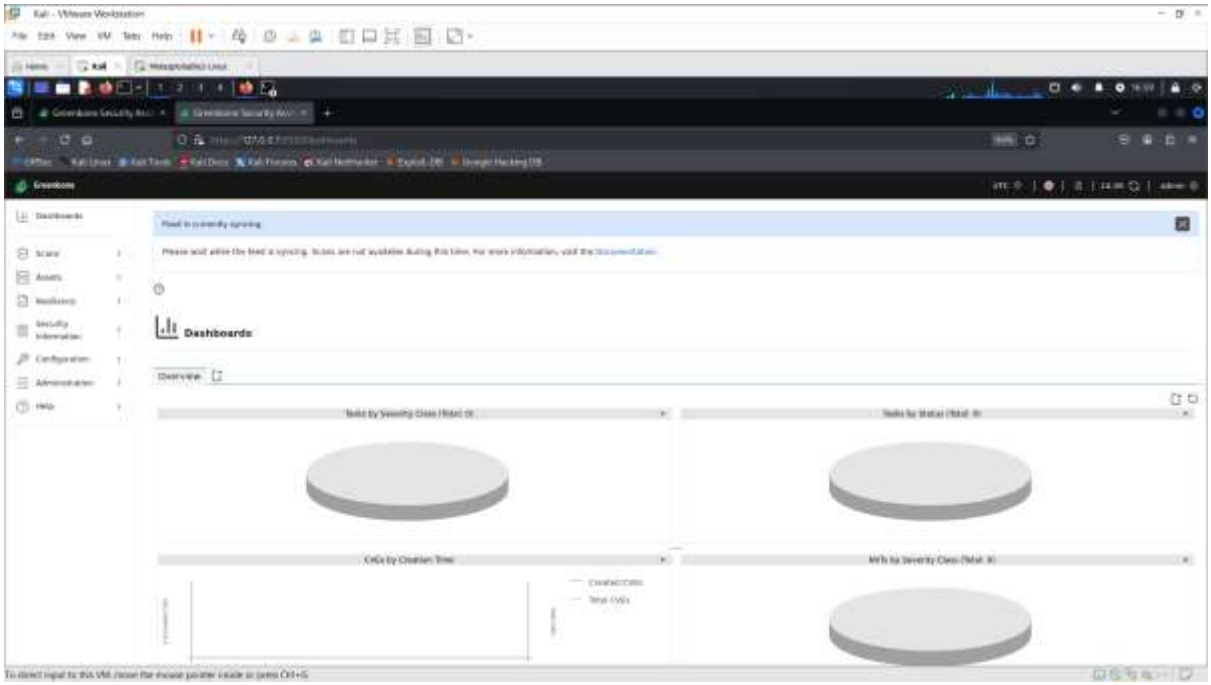
Step 5: Access the OpenVAS Web Interface

- Open your browser and navigate to:

  https://127.0.0.1:9392

- Login credentials are generated during the setup process. Use the admin username and password displayed in the terminal.



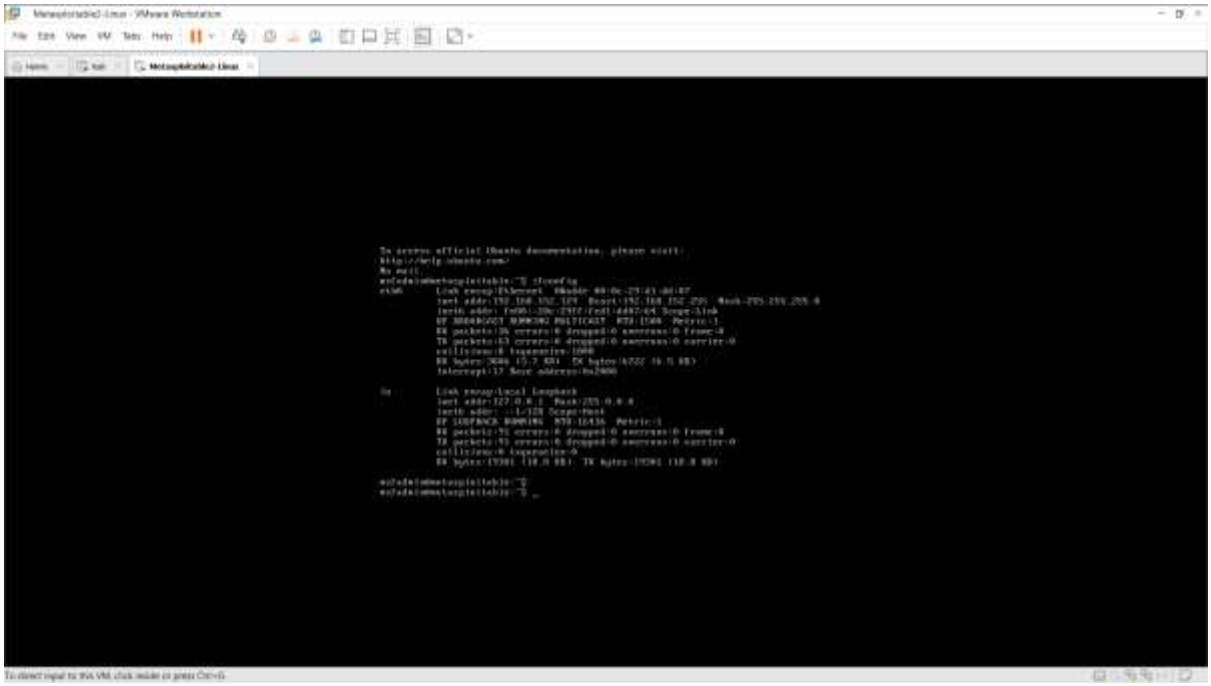


Nmap and Metasploitable2 Port Scan :

Step 1: Set Up Metasploitable2

- Ensure Metasploitable2 is running in your virtual environment.
- Note the IP address of the Metasploitable2 machine use the following command to find the IP address.

  Ifconfig



Step 2: Run a Comprehensive Nmap Scan

- Use the following command to scan all 65,535 ports, detect services and OS, and save the output in XML format:

  nmap -v -sV -A -p1-65535 192.168.152.129 -oX port.xml

  - **-v**: Verbose mode for detailed output.
  - **-sV**: Detect service versions.
  - **-A**: Aggressive scan (includes OS detection, script scanning, and traceroute).
  - **-p1-65535**: Scan all ports from 1 to 65535.
  - **-oX port.xml**: Save the scan results in XML format.

Step 3: Convert XML to HTML

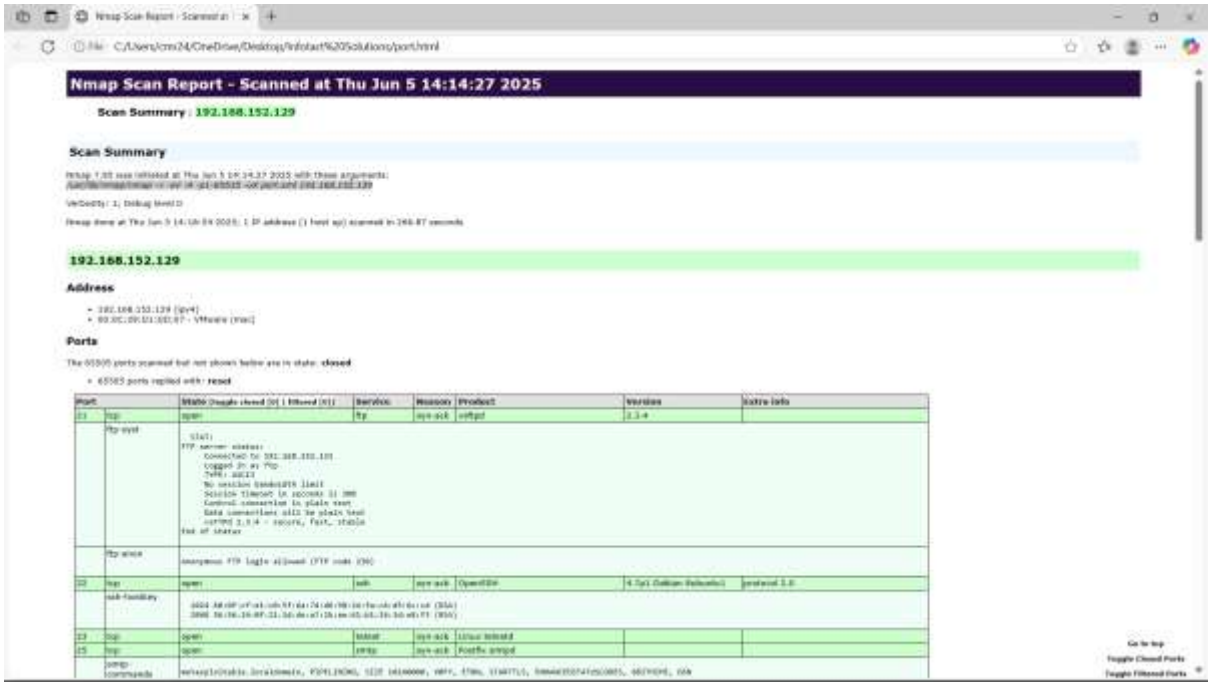
- Use xsltproc to convert the XML file into an HTML report:

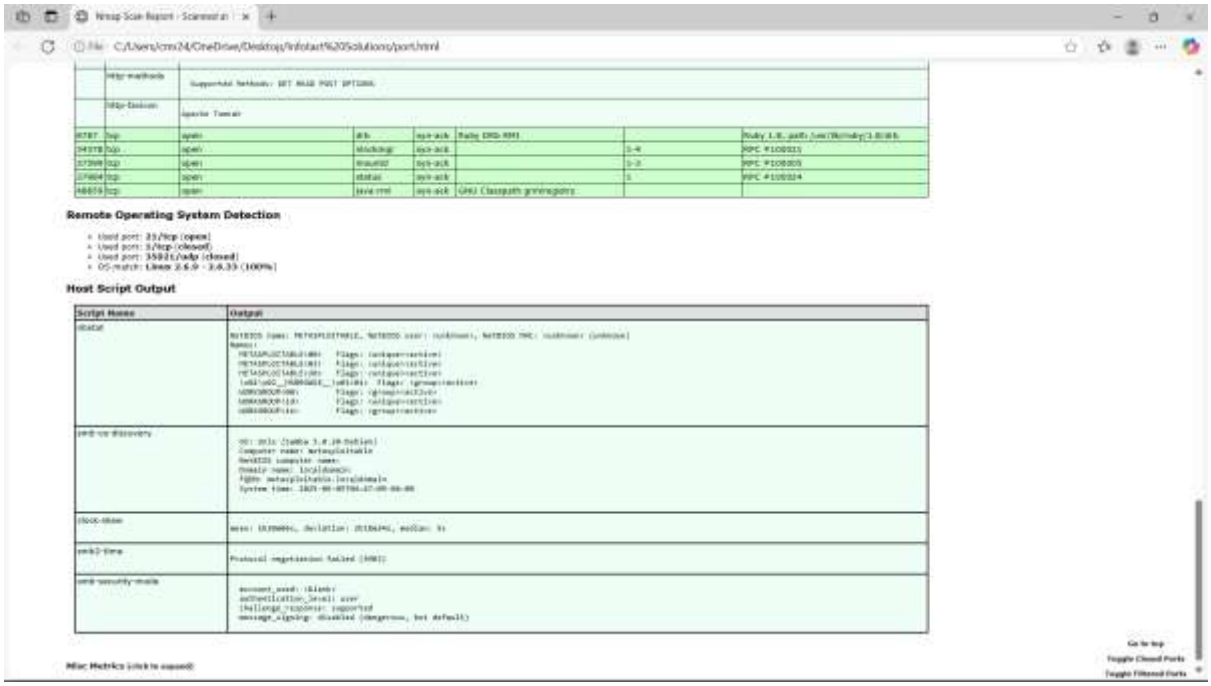
  xsltproc port.xml -o port.html

- port.xml: Input file with scan results.
- port.html: Output HTML file for viewing in a web browser.

Step 4: View the HTML Report

- Open the generated port.html in any web browser to view the detailed scan results.
- To view my nmap scan report click here





Vulnerability Found using nmap :

Port	Service	Version	Vulnerability
21	FTP	vsftpd 2.3.4	Known backdoor vulnerability in older versions.
22	SSH	OpenSSH 4.7p1	Outdated version susceptible to vulnerabilities.
23	Telnet	Linux telnetd	Plain text protocol; vulnerable to sniffing and unauthorized access.
25	SMTP	Postfix smtpd	Supports SSLv2; susceptible to POODLE attack and other SSL vulnerabilities.
53	DNS	ISC BIND 9.4.2	Outdated version; vulnerable to cache poisoning and denial-of-service attacks.
80	HTTP	Apache 2.2.8	Outdated version; vulnerable to cross-site scripting and denial-of-service attacks.

What's the risk?
Nmap reveals open ports and services, which can give attackers information about what is running on a system.

✅ **Mitigation Steps:**

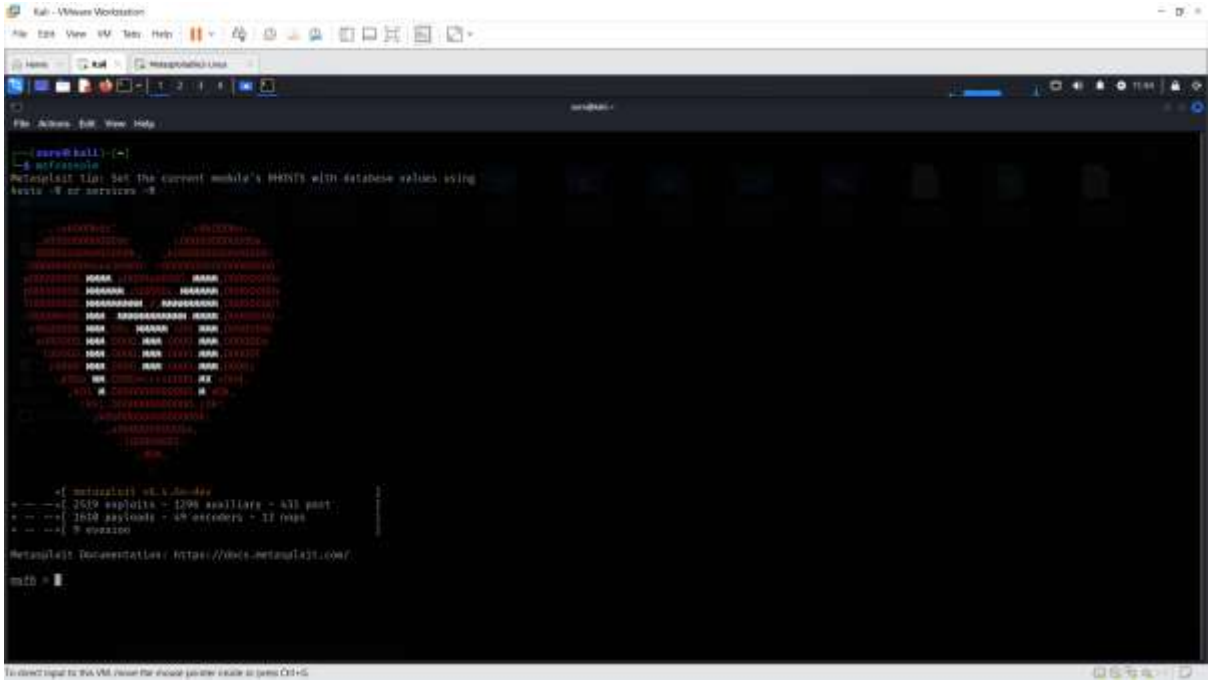
1. **Close Unused Ports**:
    - Use a firewall to block ports that aren't needed.
    - Disable services that aren't in use.
    - For example, if FTP is not used, close port 21.
2. **Use Secure Configurations**:
    - Only allow trusted IP addresses to access certain ports.
    - Implement IP whitelisting for critical services.
3. **Regular Monitoring**:
    - Scan your system frequently to identify and address newly opened ports or vulnerabilities.

Exploiting vsftpd 2.3.4 with Metasploit2 :

Step 1: Open Metasploit Console

- Open a terminal and start Metasploit:

  msfconsole

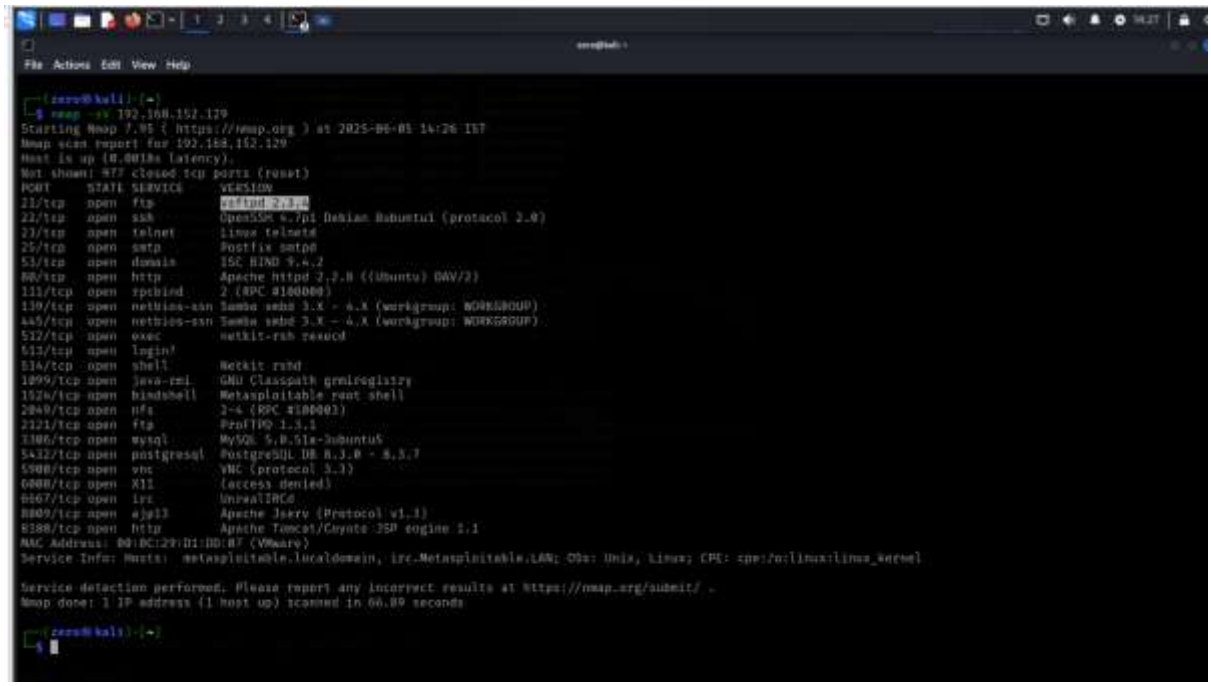


Step 2: Perform an Nmap Scan

- In a separate terminal, run the following Nmap command to find the which service is active:

  nmap -sV 192.168.152.129

- Look for FTP running **vsftpd 2.3.4** on port 21 in the results.



Step 3: Search for the Exploit

- In the Metasploit console, search for the exploit module:

search vsftpd 2.3.4

- You should see a matching module like exploit/unix/ftp/vsftpd_234_backdoor

Step 4: Use the Exploit Module

- Select the exploit module:

  use 0

- The 0 refers to the first result in the search.

Step 5: Set the Target IP Address

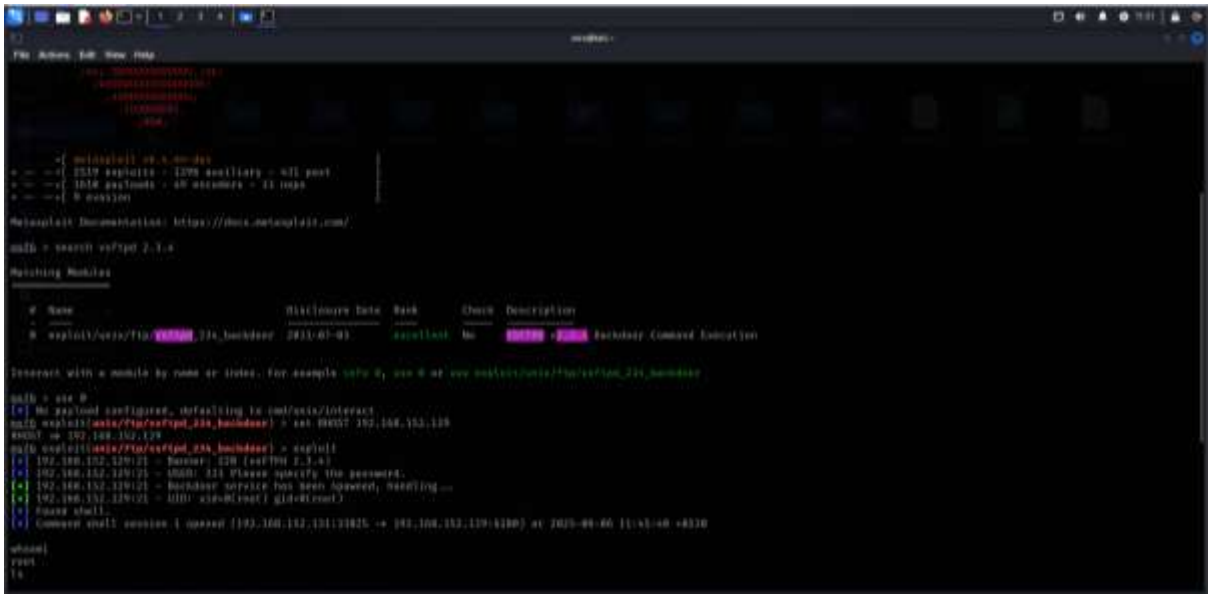
- Set the RHOST parameter to the IP of the target system:

  set RHOST 192.168.152.129

Step 6: Run the Exploit

- Launch the exploit:

  exploit

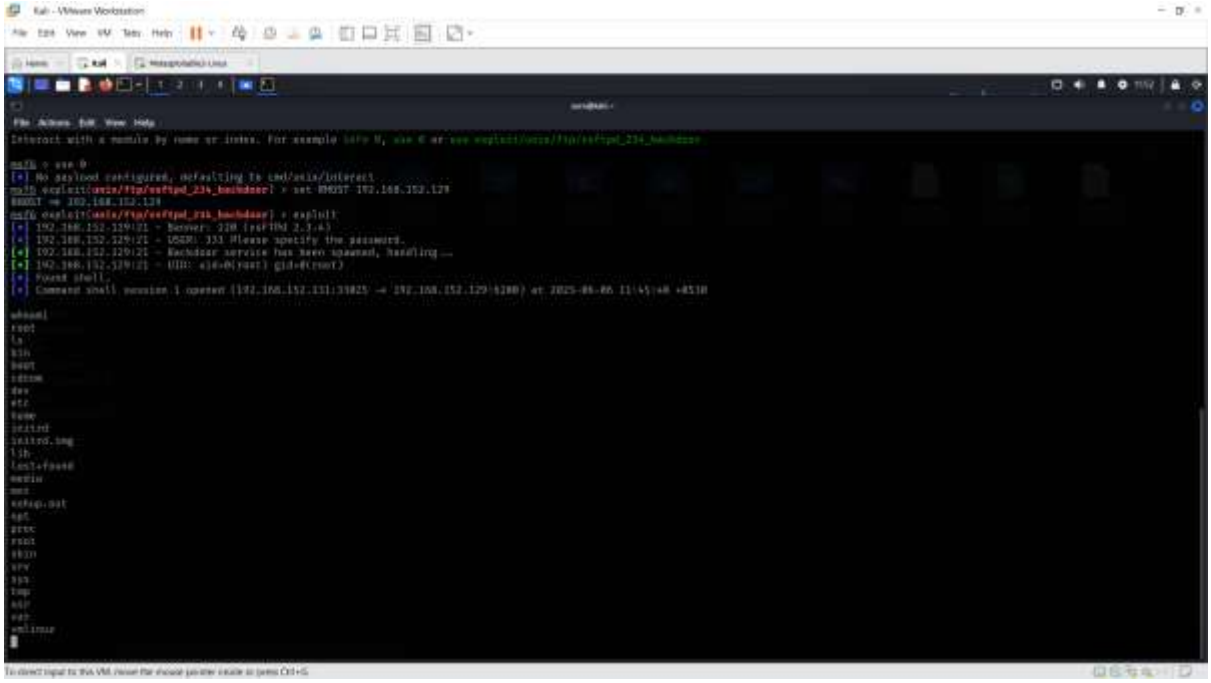


Step 7: Verify Access

- If successful, you will gain a shell or other form of access to the target system. Look for confirmation in the terminal, such as:
- Command shell session opened!

Post-Exploitation

- Explore the target system using commands like **ls**, **pwd**, etc.



Vulnerability Found in Metasploit2:

Vulnerability	Description	Risk Level	Impact
vsftpd 2.3.4 Backdoor	Hidden backdoor in FTP service allows remote attackers to get full control (root access) of the system	High	Attacker can take over the server, steal or damage data

What's the risk?

The vsftpd 2.3.4 vulnerability allows attackers to take control of the system remotely.

✅ **Mitigation Steps :**

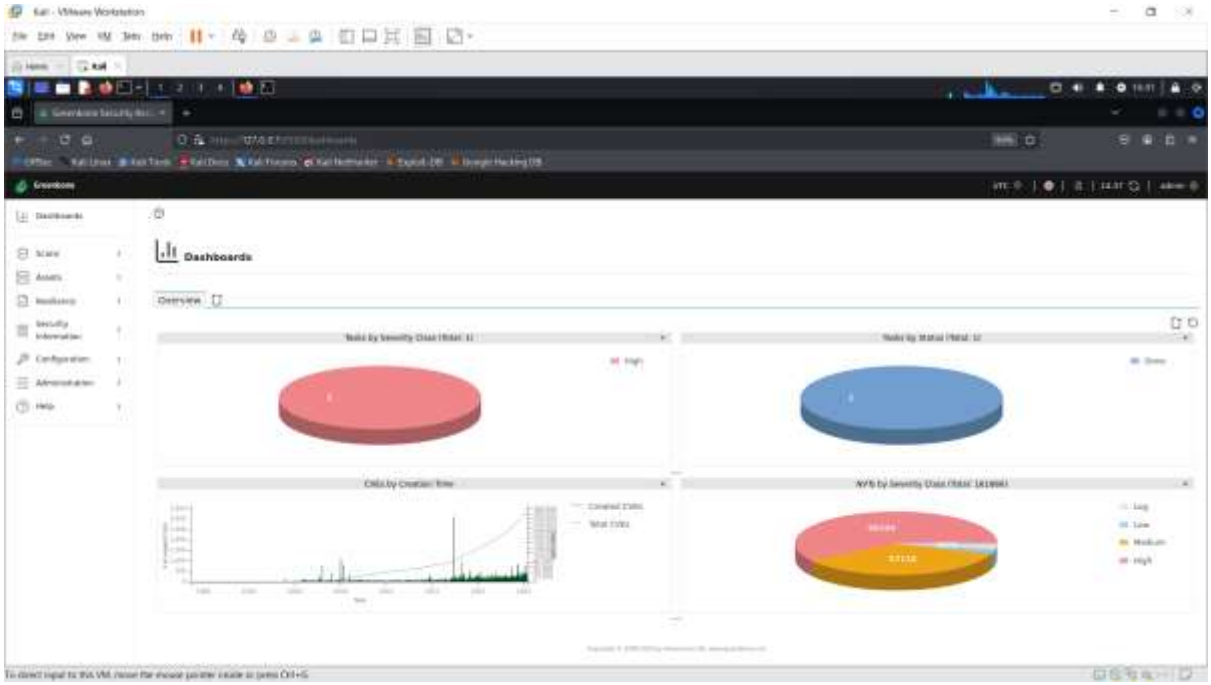
1. **Update the Software:** Install the latest version of vsftpd to fix the issue.
2. **Turn Off FTP:** If you don't need FTP, disable it to avoid risks.
3. **Use Safer Options:** Replace FTP with secure methods like SFTP or FTPS.
4. **Limit Access:** Only allow trusted devices to connect using firewall rules.

5. **Check Activity:** Regularly look at system logs to find unusual activity.
6. **Strengthen Security:** Use tools like IDS and strong passwords for all services.

Using OpenVAS for Vulnerability Scanning :

Start OpenVAS

- Launch OpenVAS and log in using your credentials.
- The dashboard will appear after successful login.

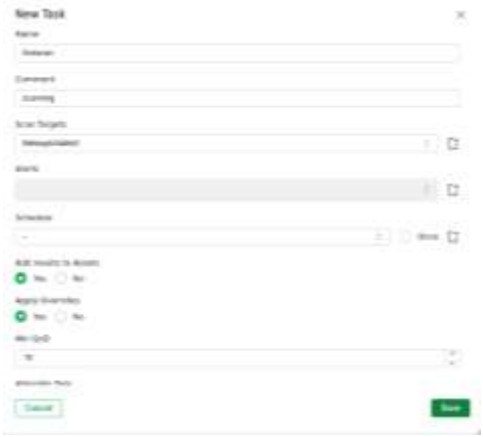


Add a Target

- Navigate to **Configuration > Targets** and click on **New Target**.
- Fill in the target details (e.g., name and IP address).
- Example: Add "Metasploitable2" with its IP address.

Create a New Task

- Go to **Scans > Tasks** and click on **New Task**.
- Fill in the task details such as task name, assigned target, and scan type.

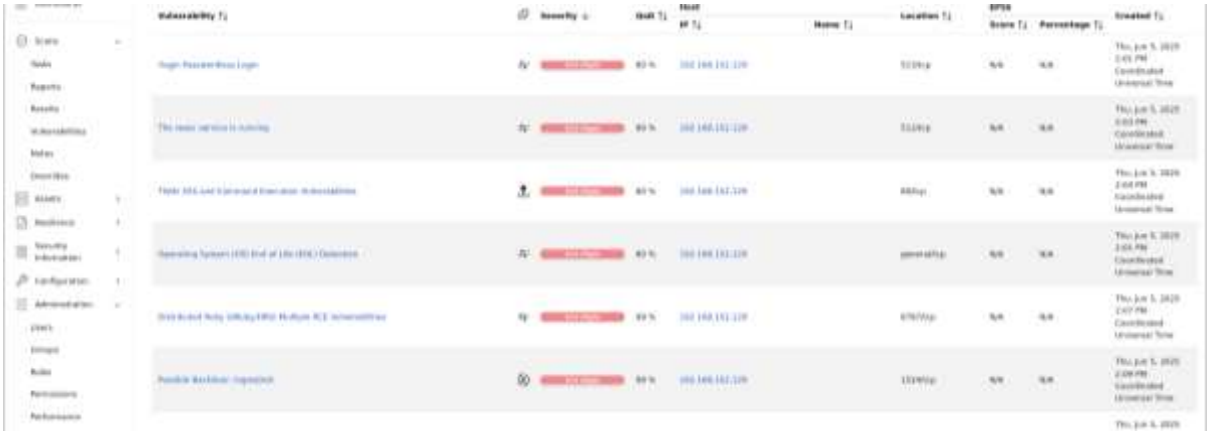


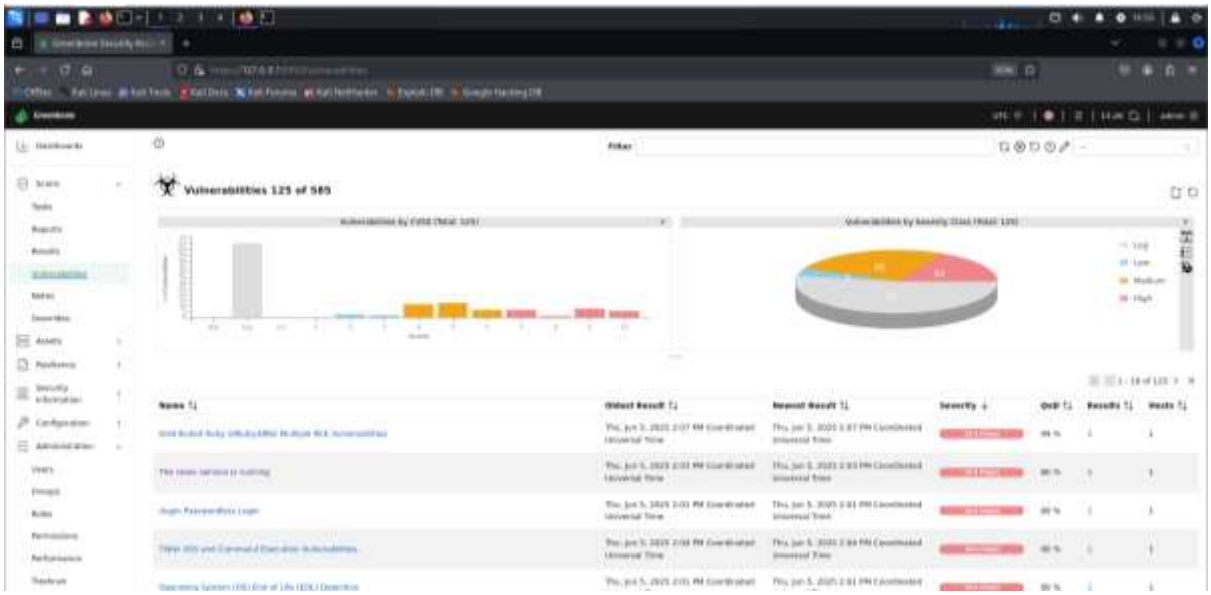
Run the Scan

- In the **Tasks** view, click the play icon next to your task.
- The task status will change from **Requested → Running → Done**.
- The scan may take 30 minutes.

View the Scanning Results

- Navigate to **Scans > Reports** or **Vulnerabilities**





Automated Scan Report

- We can download the scan results in various formats such as **XML, PDF, Text, or CSV**.
- To view my automated scan report click here

Vulnerability Found in OpenVas Scanning :

🔴 **High-Risk Vulnerabilities**

1. **Port: general** – *Outdated Ubuntu 8.04 OS*
   The system is running a very old OS that no longer gets security updates. Easy for attackers to exploit.
2. **Port: 1099** – *Java RMI Remote Code Execution*
   The Java RMI service can be tricked into executing malicious code from a remote attacker.
3. **Port: 1524** – *Ingreslock Backdoor*
   A secret backdoor is present that gives remote control to an attacker.
4. **Port: 6697** – *UnrealIRCd Backdoor & Spoofing*
   Fake login possible, and attackers can run commands remotely.
5. **Port: 512** – *rexec Service*
   Lets users run remote commands without any encryption or proper login.
6. **Port: 514** – *rsh Service*
   Allows remote login with plain text username and password.
7. **Port: 2121** – *FTP Default Credentials*
   Can log in with weak/default accounts like msfadmin, postgres, or user.

8. **Port: 21** – *vsftpd Backdoor*
   A hacked version of vsftpd is installed, allowing attackers to open a shell.
9. **Port: 5900** – *VNC Weak Password*
   VNC remote desktop access is possible using the simple password password.
10. **Port: 8787** – *Distributed Ruby RCE*
    Ruby's dRuby service can be misused to execute system commands remotely.
11. **Port: 8009** – *Apache Tomcat Ghostcat*
    A flaw in the AJP connector lets attackers read sensitive config files.
12. **Port: 80** – *TWiki & PHP Vulnerabilities*
    The website allows script injection and remote code execution.
13. **Port: 513** – *rlogin Passwordless Access*
    rlogin allows root access without any password at all.
14. **Port: 5432** – *PostgreSQL Default Login*
    You can log in using the default postgres:postgres account.
15. **Port: 6200** – *FTP Backdoor Shell*
    Backdoor opens a hidden remote shell on port 6200.
16. **Port: 3306** – *MySQL Default Login*
    MySQL root account has no password — full access is possible.
17. **Port: 3632** – *DistCC Remote Execution*
    distcc compiler allows attackers to run remote shell commands.
18. **Port: 80** – *HTTP PUT and DELETE Enabled*
    Web server allows file uploads and deletions directly through browser.

🟠 Medium-Risk Vulnerabilities

1. **Port: 22** – *SSH Weak Key Exchange*
   SSH is using old encryption methods like DH group1, which are considered insecure.
2. **Port: 2121** – *FTP TLS Weak Ciphers*
   FTP service uses weak encryption (TLS), which can be cracked easily.
3. **Port: 21** – *FTP TLS Weak Ciphers*
   Same as above but on standard FTP port.
4. **Port: 5900** – *VNC No Encryption*
   VNC traffic is unencrypted, making it easy to spy on.
5. **Port: 80** – *Old PHP Version*
   Website is running an outdated PHP version with known vulnerabilities.

6. **Port: 5432** – *PostgreSQL Weak SSL/TLS*
   PostgreSQL uses outdated SSL protocols that can be attacked.
7. **Port: 445** – *SMBv1 Enabled*
   Server Message Block v1 is outdated and has known exploits (like EternalBlue).
8. **Port: 25** – *SMTP Open Relay*
   Email server may allow sending spam or spoofed emails.
9. **Port: 23** – *Telnet Login Unencrypted*
   Telnet sends usernames and passwords in plain text.

🟢 **Low-Risk Vulnerabilities**

1. **Port: general** – *TCP Timestamp Enabled*
   System reveals its uptime, which helps in attack planning.
2. **Port: 22** – *SSH Info Disclosure*
   Reveals detailed SSH version and algorithms — useful to attackers.
3. **Port: ICMP** – *ICMP Timestamp Reply*
   System replies to ping timestamp requests, revealing its clock info.
4. **Port: 5432** – *SSLv3 Weak Cipher on PostgreSQL*
   PostgreSQL accepts outdated SSLv3 connections.
5. **Port: 25** – *DHE_EXPORT Cipher Detected*
   Very weak encryption settings in mail server (vulnerable to LogJam attack).
6. **Port: general** – *Info Leakage on Ports*
   System reveals details about running services and ports.

✅ **Mitigation Steps :**

- **Update the Operating System:** Use a newer version of Ubuntu or another OS that still gets security updates.
- **Change All Default Password:** Set strong passwords for all accounts like msfadmin, postgres, root, etc.
- **Remove Backdoors & Unused Services**: Stop or uninstall services like rexec, rlogin, rsh, UnrealIRCd, or anything suspicious.
- **Update All Old Software:** Upgrade tools like PHP, vsftpd, Tomcat, PostgreSQL, MySQL to the latest versions.
- **Fix Web Server Settings:** Disable dangerous methods like PUT/DELETE, and fix weak SSL settings.
- **Limit Access with Firewall:** Use a firewall to block unused ports and limit access to trusted IP addresses.

Testing SQL Injection in OWASP Juice Shop :

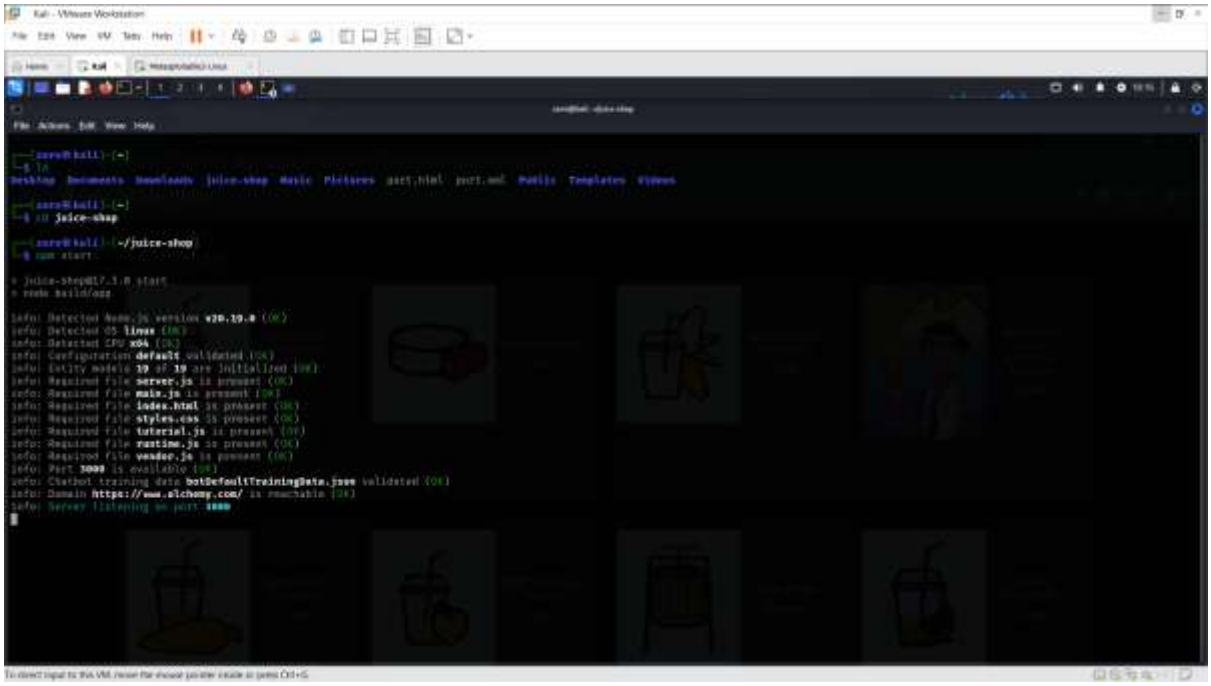
Step 1: Start the Juice Shop Application

- Navigate to the Juice Shop directory:

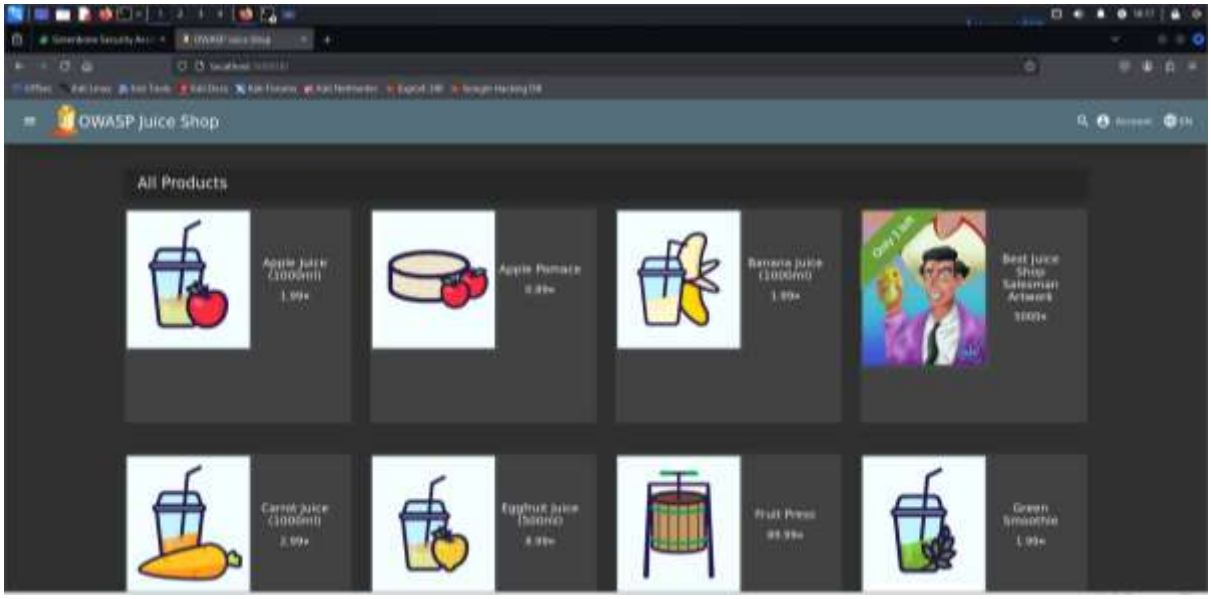
  cd juice-shop

- Start the application:

  npm start

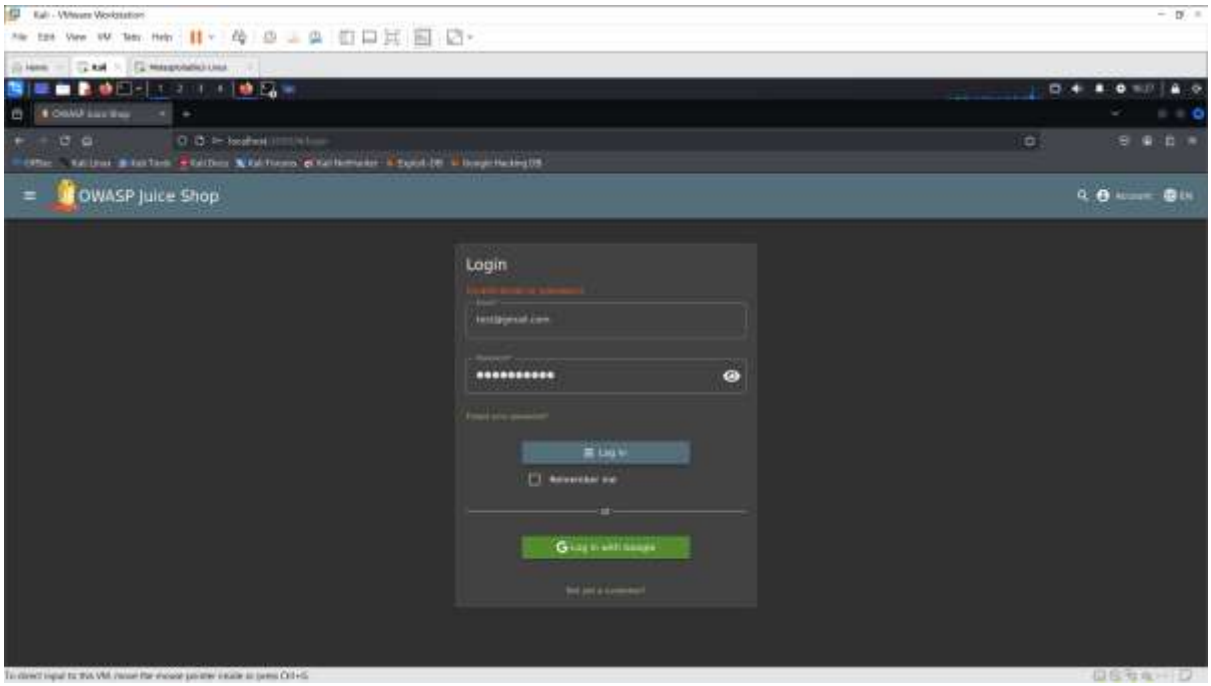


- Ensure the application is running on http://localhost:3000



Step 2: Test SQL Injection

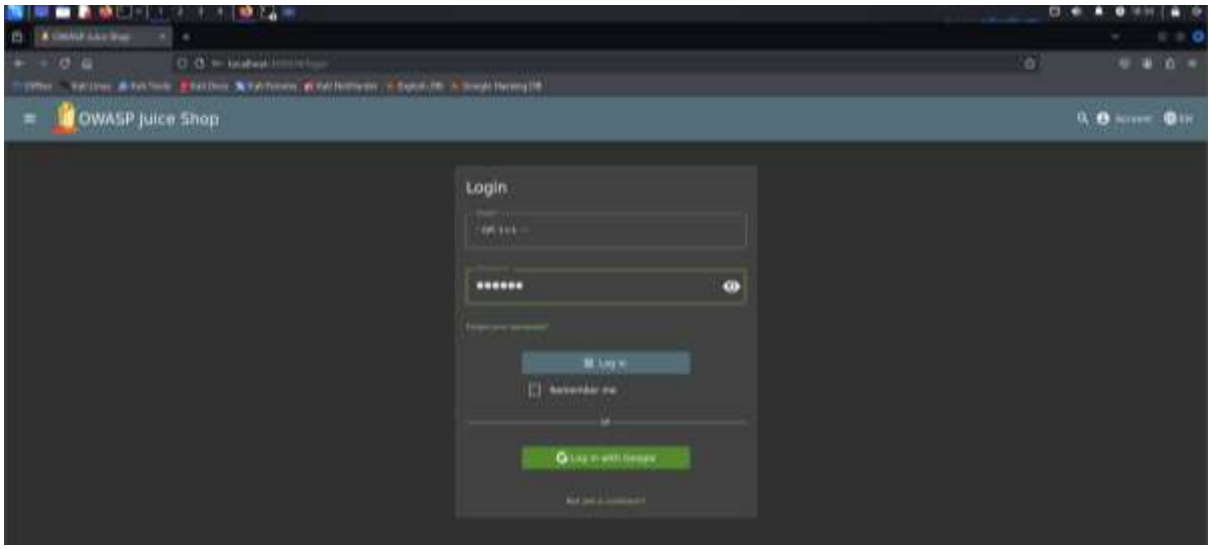
- **Login Form Test**:
    - Navigate to the login page.

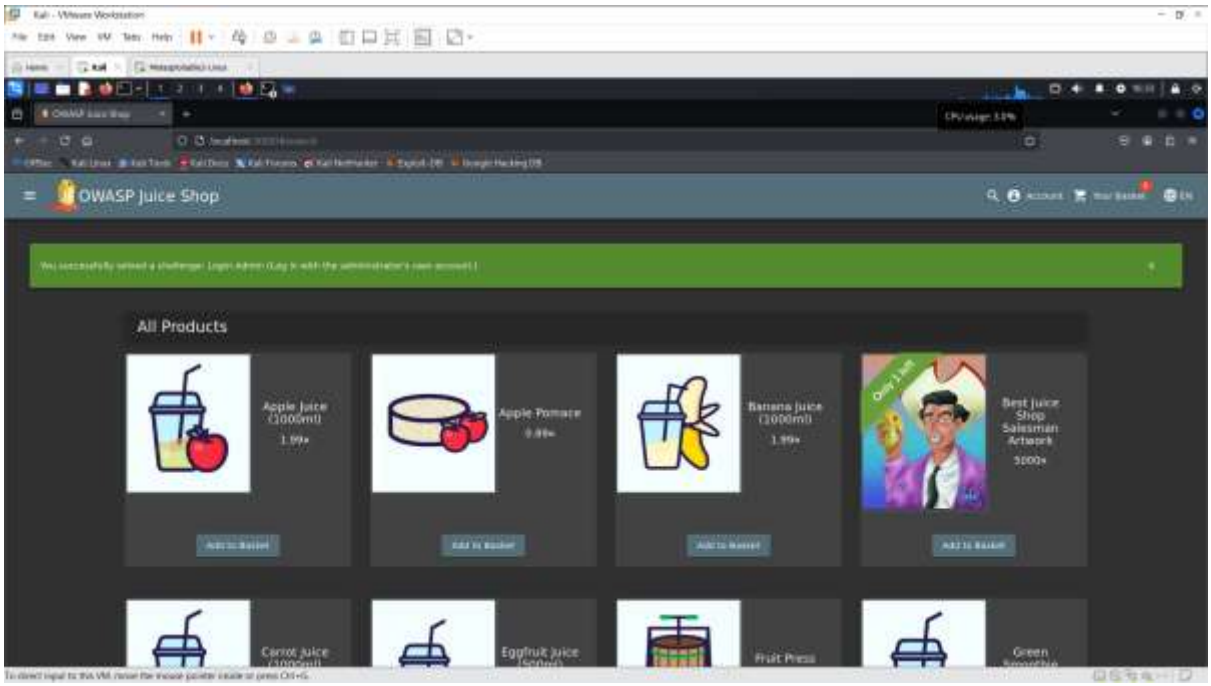


2. **SQL Injection Commands for Testing**

Authentication Bypass

- Use these payload on login forms:
    - Payload : **' OR 1=1 --** or **' OR '1'='1' –**
    - Use this above sql command for email and enter any password as your own.
- If successful, this bypasses authentication and logs you in as an admin.





Vulnerability found in OWASP Juice Shop :

Vulnerability	What is it?	Where?	Risk
SQL Injection	Bad input let attackers change database command.	Login Page	Can login without permission and steal data.

What's the risk?

SQL injection allows attackers to execute harmful SQL queries, steal data, or take control of the database.

✅ **Mitigation Steps:**

1. **Validate Inputs:** Check and clean user inputs to remove harmful characters.
2. **Parameterized Queries:** Use safe coding methods to separate user inputs from SQL commands.
3. **Minimal Access Rights:** Limit what the application can do in the database.
4. **Web Application Firewall (WAF):** Use a firewall to block attacks.
5. **Hide Errors:** Don't show technical details in error messages.
6. **Regular Updates:** Keep systems and software patched and secure.

Conclusion :

The vulnerability assessment for the small business network identified several security weaknesses, including misconfigured services, open ports, and exploitable web applications. Tools like Nmap, Metasploit, and OpenVAS were used to scan, test, and analyze these vulnerabilities. Key findings showed risks in outdated software, weak authentication, and exposed network configurations. To improve security, recommendations included updating software regularly, restricting access to critical services, using firewalls, and validating user inputs to prevent attacks like SQL injection. Regular vulnerability scans and monitoring were also suggested to maintain a secure network. This project highlighted the importance of proactive security measures for small businesses to protect their data and operations from cyber threats. By following these steps, businesses can reduce risks and improve their overall security.