Vulnerability Assessment and Penetration Testing (VAPT) Weekly Report – Week 1

TABLE OF CONTENTS

1. Introduction

I performed this task to understand Security Assessment and Vulnerability Assessment & Penetration Testing (VAPT) using only open-source tools**.** I conducted the assessment in a controlled lab environment using Kali Linux as the attacking machine and an intentionally vulnerable system as the target. This setup allowed me to safely test and analyze security weaknesses without affecting any real systems.

During this task, I set up the lab environment, performed network discovery, scanned the target for vulnerabilities, exploited selected weaknesses, assessed the risks, and documented the results. Each step helped me understand how attackers identify and use security flaws. This report explains the complete process I followed, the vulnerabilities I identified, the risks involved, and the security recommendations based on my findings.

2. Executive Summary

I found several critical, high, and medium-risk security issues in the target system during the assessment. Most of these issues were caused by old software versions, poor security settings, open network ports, and services that were not properly protected. I used Nmap to discover open ports and running services. I used OpenVAS and Nikto to scan the system and identify known vulnerabilities. I used the Metasploit Framework to confirm whether these weaknesses could actually be misused.

I observed that if these vulnerabilities are not fixed, an attacker could enter the system without permission, run harmful commands, or access sensitive data. Based on these findings, I strongly recommend updating the system, securing services, and applying proper security controls to reduce the risk.

3. Objectives

I aimed to understand the basics of security assessment and Vulnerability Assessment & Penetration Testing (VAPT) through this task. I learned how to scan systems for vulnerabilities using open-source tools and how to find security weaknesses in a target system. I also learned how to judge and prioritize risks using CVSS scores and simple risk matrices. Through controlled exploitation in a lab setup, I learned how attackers misuse system weaknesses. Finally, I learned how to document my work and write a clear and professional cybersecurity report.

4. Scope

I tested only the intentionally vulnerable virtual machine within a controlled lab environment. During this task, I performed network scanning to identify open ports and services, followed by vulnerability scanning and basic risk assessment to understand the severity of the issues found. I also carried out limited and controlled exploitation to observe how these weaknesses could be misused, and I documented all findings clearly in this report. I avoided testing any real or live systems, avoided Denial-of-Service attacks and social engineering techniques, and ensured that all activities remained strictly within the defined lab environment.

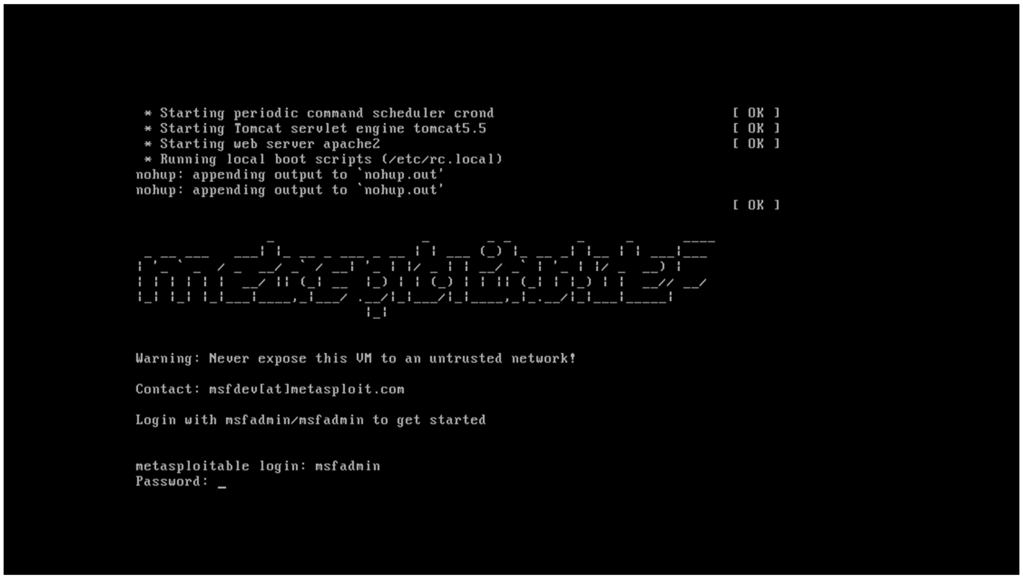
5. Setup Testing Environment

I used virtualization to create a safe and isolated testing environment
Environment Details:

- Attacker Machine: Kali Linux
- Target Machine: Metasploitable
- Virtualization Tool: VirtualBox
- Network Mode: Host-only / Internal Network

Screenshot 1: Kali Linux desktop

Screenshot 2: Target VM running



```
 * Starting periodic command scheduler crond                          [ OK ]
 * Starting Tomcat servlet engine tomcat5.5                           [ OK ]
 * Starting web server apache2                                        [ OK ]
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'
                                                                      [ OK ]

                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password: _
```

6. VAPT Methodology

I followed a structured VAPT methodology to ensure that the security assessment was performed in a clear and systematic manner. This approach helped me complete each step properly and avoid missing any important part of the assessment.

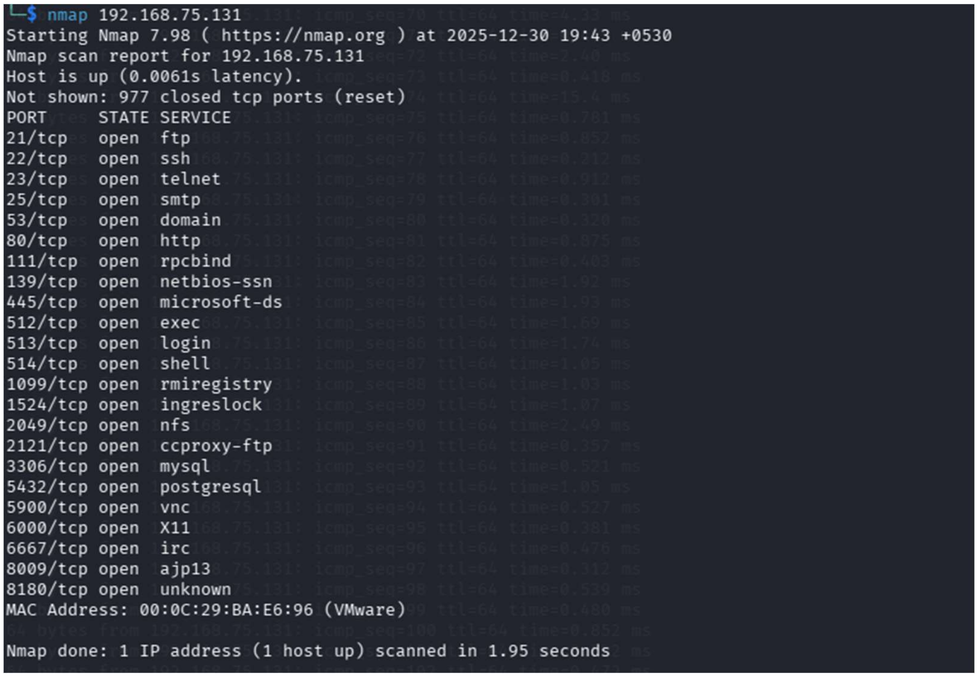
I performed the assessment using the following phases:

- **Planning** – I defined the scope of testing and selected the required tools.
- **Discovery** – I performed network scanning to identify open ports and running services.
- **Vulnerability Scanning** – I scanned the target system to identify known vulnerabilities and weak configurations.
- **Exploitation** – I performed controlled exploitation of selected critical vulnerabilities in the lab environment.
- **Risk Assessment** – I analyzed and prioritized vulnerabilities based on their severity and impact.
- **Reporting** – I documented all findings, risks, and security recommendations in a clear and professional manner.

7. Discovery – Nmap

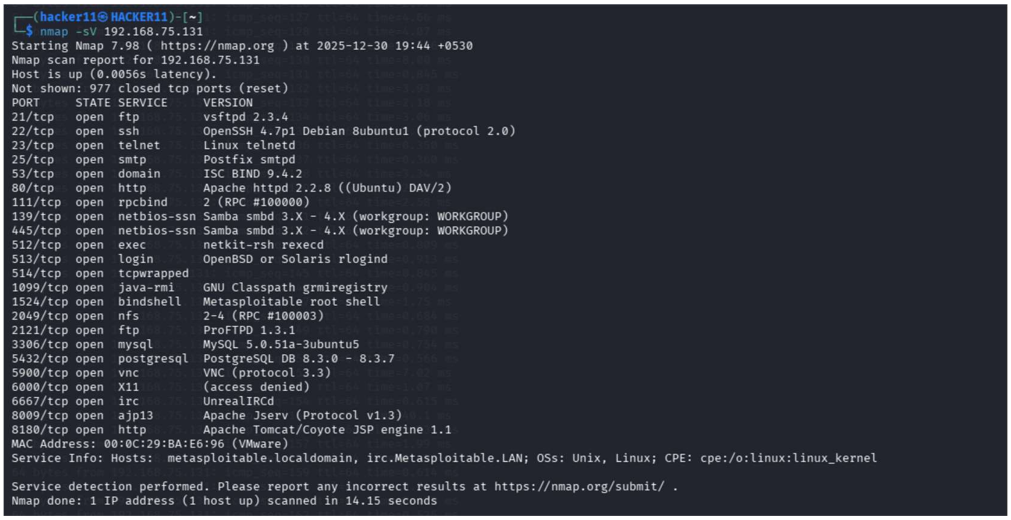
I found open ports and running services on the target system by scanning it with Nmap. I observed that many services were exposed, which increased the attack surface of the system. I also found service version details, and I observed that some services were outdated and could be vulnerable to attacks.

Screenshot 3: Nmap scan command and execution



```
└─$ nmap 192.168.75.131
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-30 19:43 +0530
Nmap scan report for 192.168.75.131
Host is up (0.0061s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:BA:E6:96 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.95 seconds
```

Screenshot 4: Nmap results showing open ports and services



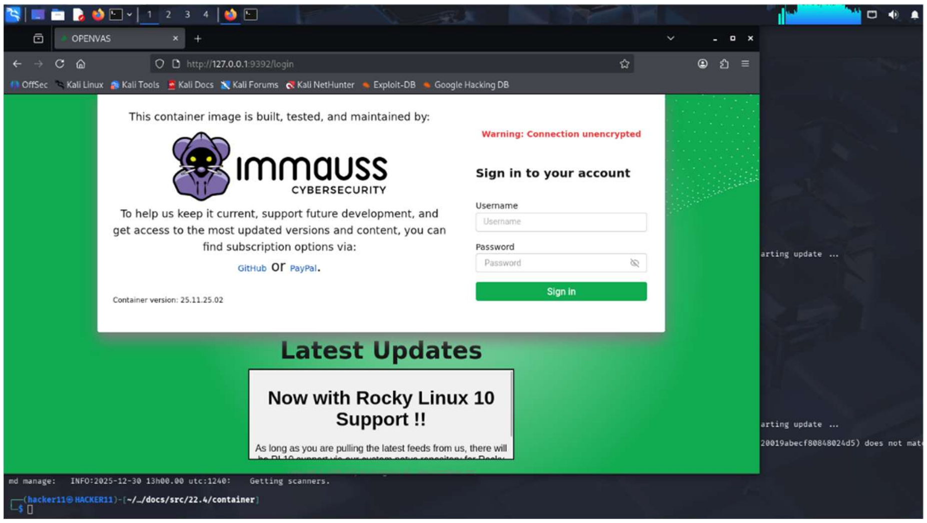
```
┌──(hacker11㉿HACKER11)-[~]
└─$ nmap -sV 192.168.75.131
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-30 19:44 +0530
Nmap scan report for 192.168.75.131
Host is up (0.0056s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (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 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
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)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:BA:E6:96 (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.15 seconds
```

8. Vulnerability Scanning – OpenVAS

I used OpenVAS to perform an automated vulnerability scan on the target system. First, I configured the target properly and then executed a full vulnerability scan to check for known security issues. After the scan was completed, I reviewed the results by analyzing the severity levels, CVSS scores, and related CVE references.

The scan results showed several Critical and High-risk vulnerabilities**.** These issues were mainly related to outdated operating system components**,** insecure or misconfigured services**,** and known vulnerabilities that could be exploited by attackers**.** This highlighted the need for proper patching and system hardening.

Screenshot 5: OpenVAS Login Interface



This container image is built, tested, and maintained by:

To help us keep it current, support future development, and get access to the most updated versions and content, you can find subscription options via:

GitHub or PayPal.

Container version: 25.11.25.02

Warning: Connection unencrypted

Sign in to your account

Username: Username

Password: Password

Sign in

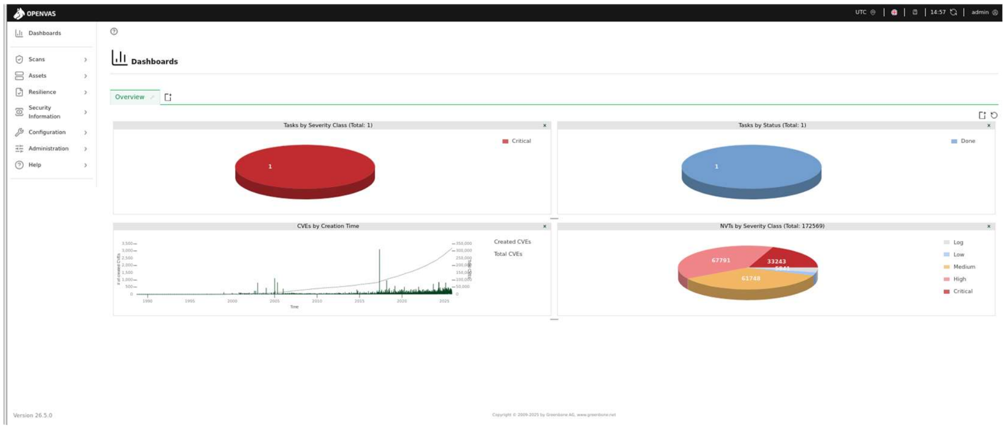
Latest Updates

Now with Rocky Linux 10 Support !!

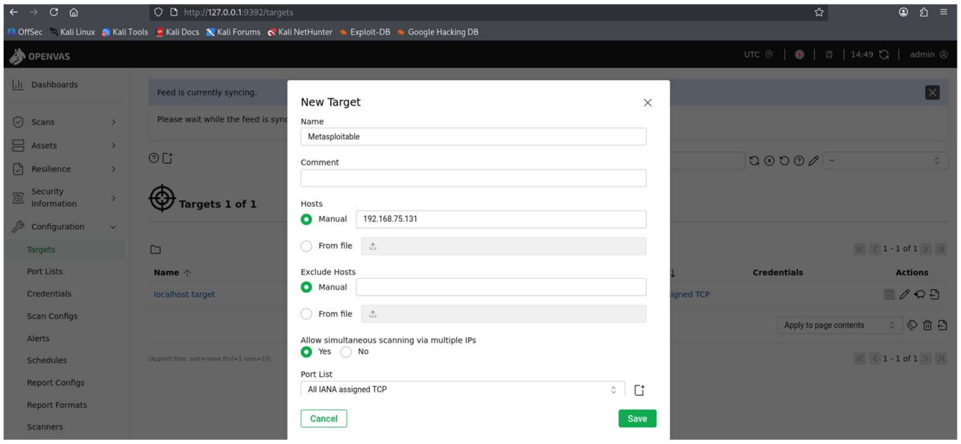
As long as you are pulling the latest feeds from us, there will

```
arting update ...
arting update ...
20019abecf80848024d5) does not mat
md manage:   INFO:2025-12-30 13h00.00 utc:1240:    Getting scanners.
┌──(hacker11㉿HACKER11)-[~/…/docs/src/22.4/container]
└─$ ▯
```

Screenshot 6: OpenVAS dashboard



Screenshot 7: Target configuration



http://127.0.0.1:9392/targets

New Target

Name: Metasploitable

Comment:

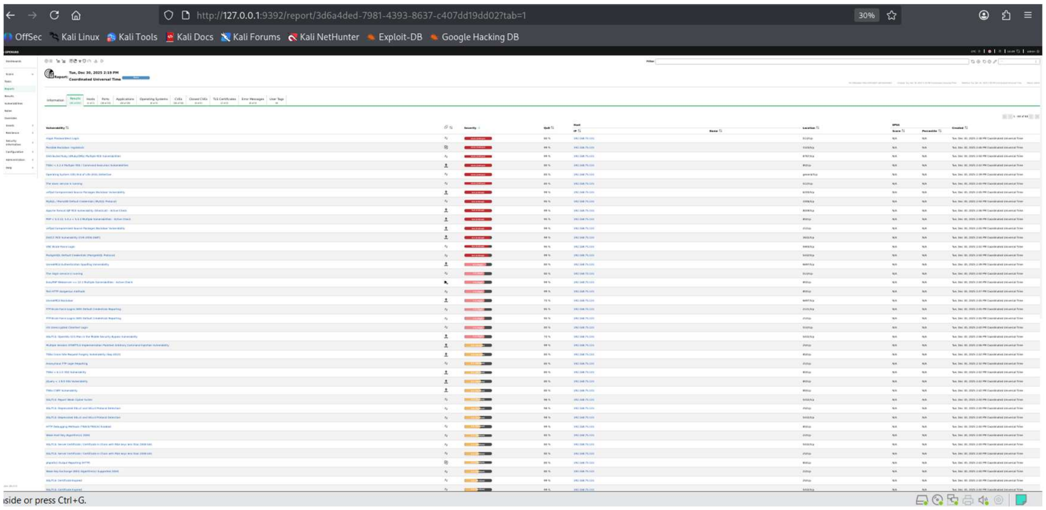
Hosts: ● Manual 192.168.75.131 ○ From file

Exclude Hosts: ● Manual ○ From file

Allow simultaneous scanning via multiple IPs: ● Yes ○ No

Port List: All IANA assigned TCP

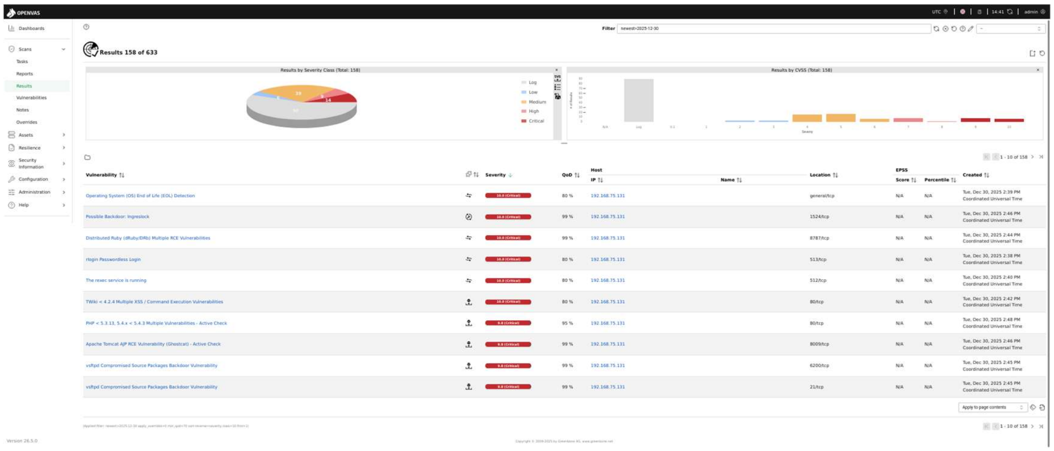
Screenshot 8: Vulnerability list



http://127.0.0.1:9392/report/3d6a4ded-7981-4393-8637-c407dd19dd02?tab=1

Report: Tue, Dec 30, 2025 2:19 PM Coordinated Universal Time

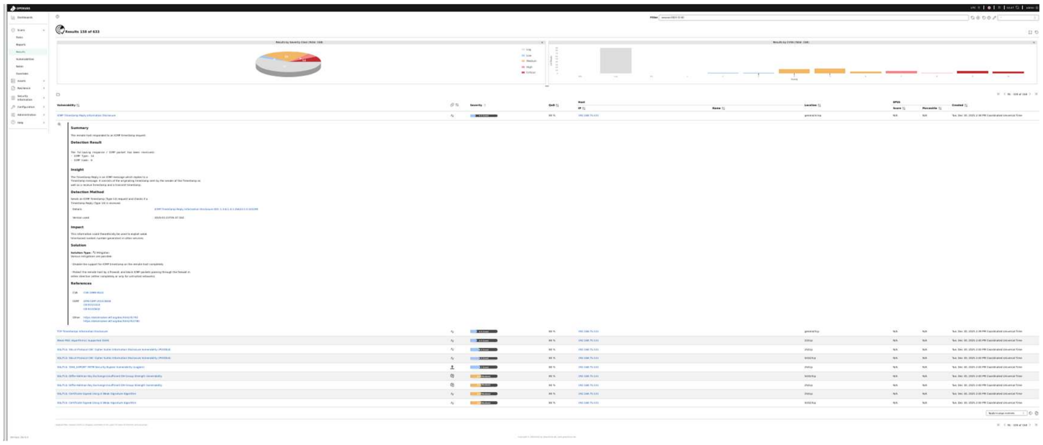
Screenshot 9: CVSS severity details



Results 158 of 633

Vulnerability	Severity	QoD	Host IP	Host Name	Location	EPSS Score	EPSS Percentile	Created
Operating System (OS) End of Life (EOL) Detection	10.0 (Critical)	80 %	192.168.75.131		general/tcp	N/A	N/A	Tue, Dec 30, 2025 2:39 PM Coordinated Universal Time
Possible Backdoor: Ingreslock	10.0 (Critical)	99 %	192.168.75.131		1524/tcp	N/A	N/A	Tue, Dec 30, 2025 2:46 PM Coordinated Universal Time
Distributed Ruby (dRuby/DRb) Multiple RCE Vulnerabilities	10.0 (Critical)	99 %	192.168.75.131		8787/tcp	N/A	N/A	Tue, Dec 30, 2025 2:44 PM Coordinated Universal Time
rlogin Passwordless Login	10.0 (Critical)	80 %	192.168.75.131		513/tcp	N/A	N/A	Tue, Dec 30, 2025 2:38 PM Coordinated Universal Time
The rexec service is running	10.0 (Critical)	80 %	192.168.75.131		512/tcp	N/A	N/A	Tue, Dec 30, 2025 2:40 PM Coordinated Universal Time
TWiki < 4.2.4 Multiple XSS / Command Execution Vulnerabilities	10.0 (Critical)	80 %	192.168.75.131		80/tcp	N/A	N/A	Tue, Dec 30, 2025 2:42 PM Coordinated Universal Time
PHP < 5.3.13, 5.4.x < 5.4.3 Multiple Vulnerabilities - Active Check	9.8 (Critical)	95 %	192.168.75.131		80/tcp	N/A	N/A	Tue, Dec 30, 2025 2:48 PM Coordinated Universal Time
Apache Tomcat AJP RCE Vulnerability (Ghostcat) - Active Check	9.8 (Critical)	99 %	192.168.75.131		8009/tcp	N/A	N/A	Tue, Dec 30, 2025 2:46 PM Coordinated Universal Time
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (Critical)	99 %	192.168.75.131		6200/tcp	N/A	N/A	Tue, Dec 30, 2025 2:45 PM Coordinated Universal Time
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (Critical)	99 %	192.168.75.131		21/tcp	N/A	N/A	Tue, Dec 30, 2025 2:45 PM Coordinated Universal Time

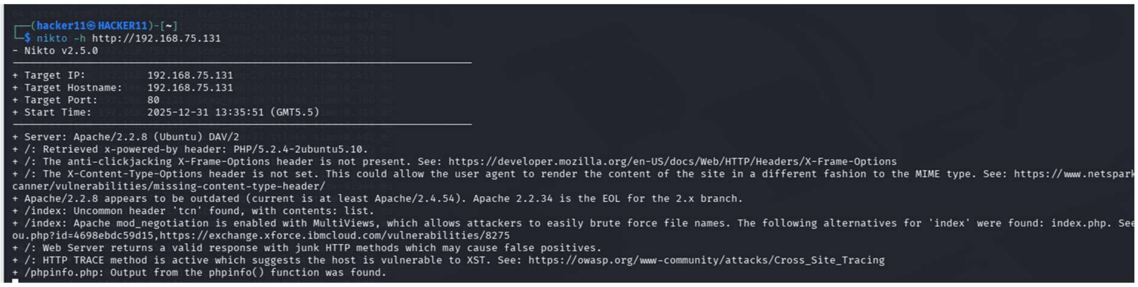
Screenshot 10: Individual vulnerability view



9. Web Scanning – Nikto

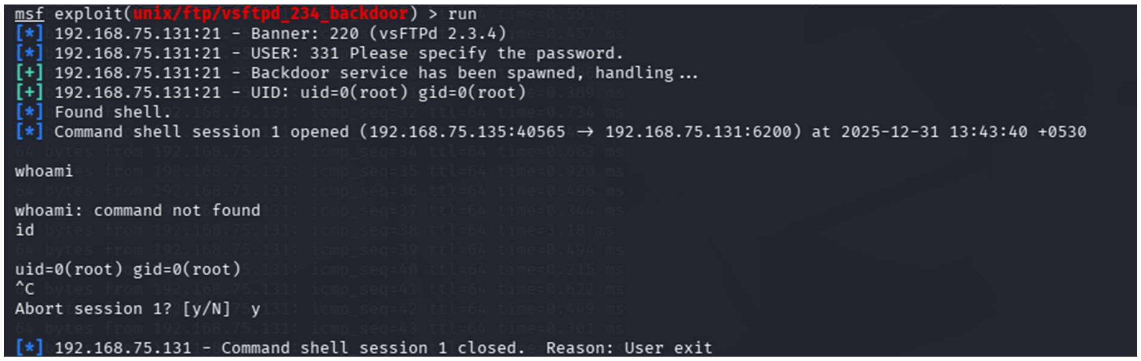
I used Nikto to scan the web server for security issues and misconfigurations. During the scan, Nikto identified problems such as insecure HTTP headers, outdated web components, and possible information disclosure. These findings indicate that the web server is not securely configured and may expose sensitive information if not properly secured.

Screenshot 11: Nikto scan command



```
┌──(hacker11㉿HACKER11)-[~]
└─$ nikto -h http://192.168.75.131
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.75.131
+ Target Hostname:    192.168.75.131
+ Target Port:        80
+ Start Time:         2025-12-31 13:35:51 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netspark
canner/vulnerabilities/missing-content-type-header/
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See
ou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross_Site_Tracing
+ /phpinfo.php: Output from the phpinfo() function was found.
```

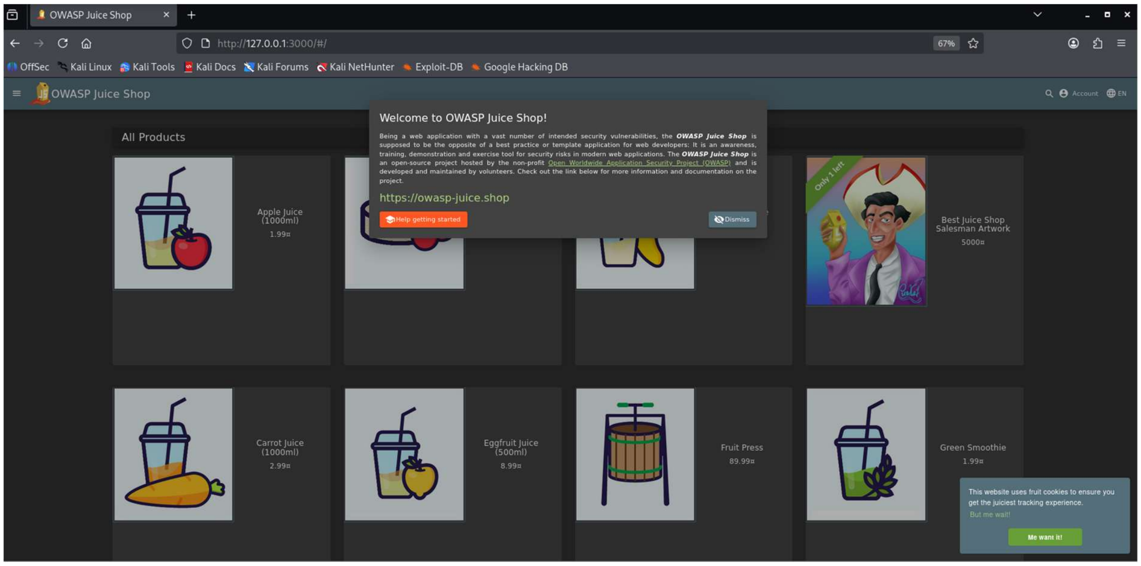
Screenshot 12: Nikto scan results



```
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.75.131:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.75.131:21 - USER: 331 Please specify the password.
[+] 192.168.75.131:21 - Backdoor service has been spawned, handling...
[+] 192.168.75.131:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.75.135:40565 → 192.168.75.131:6200) at 2025-12-31 13:43:40 +0530

whoami

whoami: command not found
id

uid=0(root) gid=0(root)
^C
Abort session 1? [y/N]  y

[*] 192.168.75.131 - Command shell session 1 closed.  Reason: User exit
```

Nikto + OWASP ZAP

OWASP ZAP automated scan detecting SQL Injection in Juice Shop



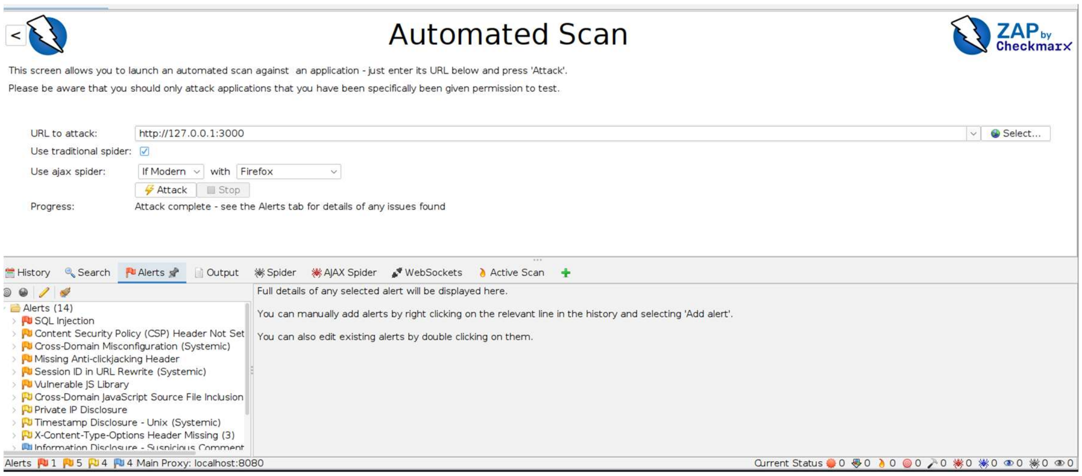
Welcome to OWASP Juice Shop!

Being a web application with a vast number of intended security vulnerabilities, the ***OWASP Juice Shop*** is supposed to be the opposite of a best practice or template application for web developers: It is an awareness, training, demonstration and exercise tool for security risks in modern web applications. The ***OWASP Juice Shop*** is an open-source project hosted by the non-profit Open Worldwide Application Security Project (OWASP) and is developed and maintained by volunteers. Check out the link below for more information and documentation on the project.

https://owasp-juice.shop

Help getting started   Dismiss

This website uses fruit cookies to ensure you get the juiciest tracking experience.
But me wait!
Me want it!



Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://127.0.0.1:3000   Select...
Use traditional spider: ☑
Use ajax spider: If Modern with Firefox
Attack   Stop
Progress: Attack complete - see the Alerts tab for details of any issues found

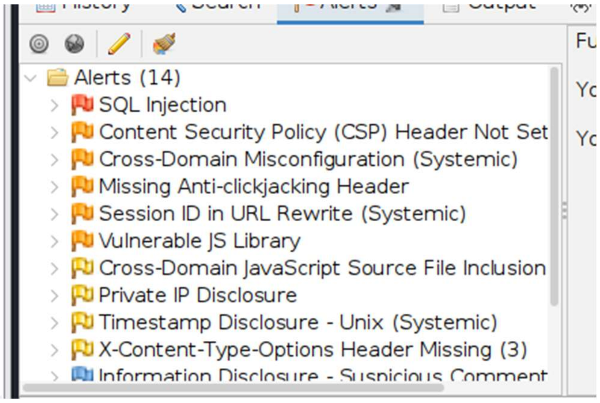
Alerts (14)
- SQL Injection
- Content Security Policy (CSP) Header Not Set
- Cross-Domain Misconfiguration (Systemic)
- Missing Anti-clickjacking Header
- Session ID in URL Rewrite (Systemic)
- Vulnerable JS Library
- Cross-Domain JavaScript Source File Inclusion
- Private IP Disclosure
- Timestamp Disclosure - Unix (Systemic)
- X-Content-Type-Options Header Missing (3)
- Information Disclosure - Suspicious Comment

Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

Alerts 1 5 4 4 Main Proxy: localhost:8080   Current Status 0 0 0 0 0 0 0 0 0 0

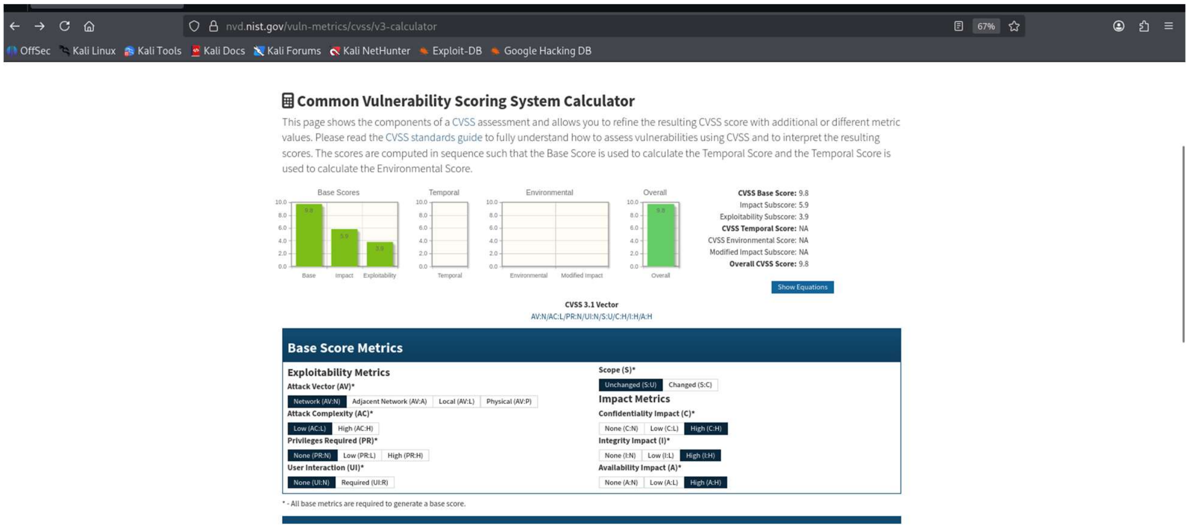


Alerts (14)
- SQL Injection
- Content Security Policy (CSP) Header Not Set
- Cross-Domain Misconfiguration (Systemic)
- Missing Anti-clickjacking Header
- Session ID in URL Rewrite (Systemic)
- Vulnerable JS Library
- Cross-Domain JavaScript Source File Inclusion
- Private IP Disclosure
- Timestamp Disclosure - Unix (Systemic)
- X-Content-Type-Options Header Missing (3)
- Information Disclosure - Suspicious Comment

10. Risk Assessment

I assessed the risks of the identified vulnerabilities using the NVD CVSS Calculator and a simple risk matrix. I considered how likely each vulnerability was to be exploited and how it could affect the system's confidentiality, integrity, and availability. I also considered the CVSS score and severity level to decide which vulnerabilities required the highest priority.

Screenshot 13: CVSS scoring calculation



Common Vulnerability Scoring System Calculator

This page shows the components of a CVSS assessment and allows you to refine the resulting CVSS score with additional or different metric values. Please read the CVSS standards guide to fully understand how to assess vulnerabilities using CVSS and to interpret the resulting scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.

CVSS Base Score: 9.8
Impact Subscore: 5.9
Exploitability Subscore: 3.9
CVSS Temporal Score: NA
CVSS Environmental Score: NA
Modified Impact Subscore: NA
Overall CVSS Score: 9.8

CVSS 3.1 Vector
AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*
[✓ Network (AV:N)] Adjacent Network (AV:A) Local (AV:L) Physical (AV:P)

Attack Complexity (AC)*
[✓ Low (AC:L)] High (AC:H)

Privileges Required (PR)*
[✓ None (PR:N)] Low (PR:L) High (PR:H)

User Interaction (UI)*
[✓ None (UI:N)] Required (UI:R)

Scope (S)*
[✓ Unchanged (S:U)] Changed (S:C)

Impact Metrics

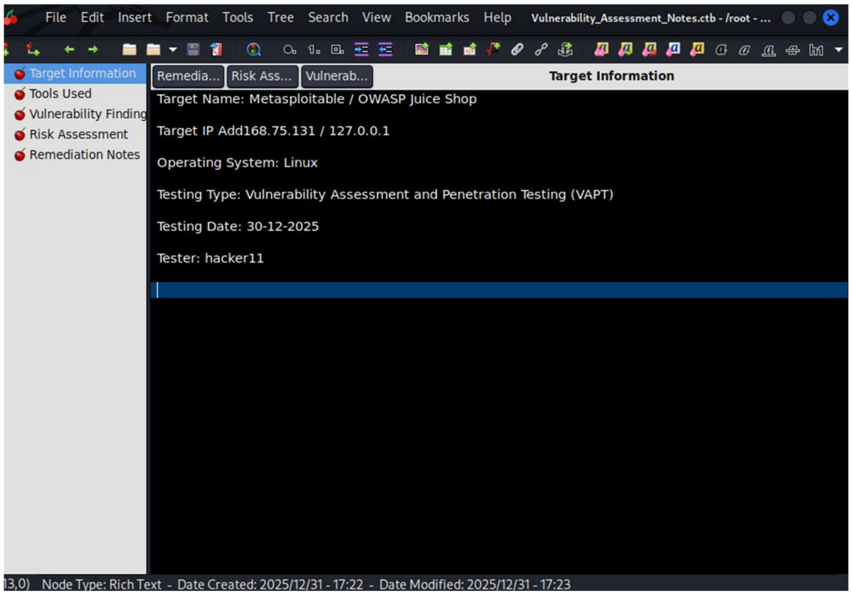
Confidentiality Impact (C)*
None (C:N) Low (C:L) [✓ High (C:H)]

Integrity Impact (I)*
None (I:N) Low (I:L) [✓ High (I:H)]

Availability Impact (A)*
None (A:N) Low (A:L) [✓ High (A:H)]

* - All base metrics are required to generate a base score.

Documented vulnerabilities and observations recorded using CherryTree for structured risk assessment and reporting.



Target Name: Metasploitable / OWASP Juice Shop

Target IP Add168.75.131 / 127.0.0.1

Operating System: Linux

Testing Type: Vulnerability Assessment and Penetration Testing (VAPT)

Testing Date: 30-12-2025

Tester: hacker11

13,0) Node Type: Rich Text - Date Created: 2025/12/31 - 17:22 - Date Modified: 2025/12/31 - 17:23

Risk assessment table

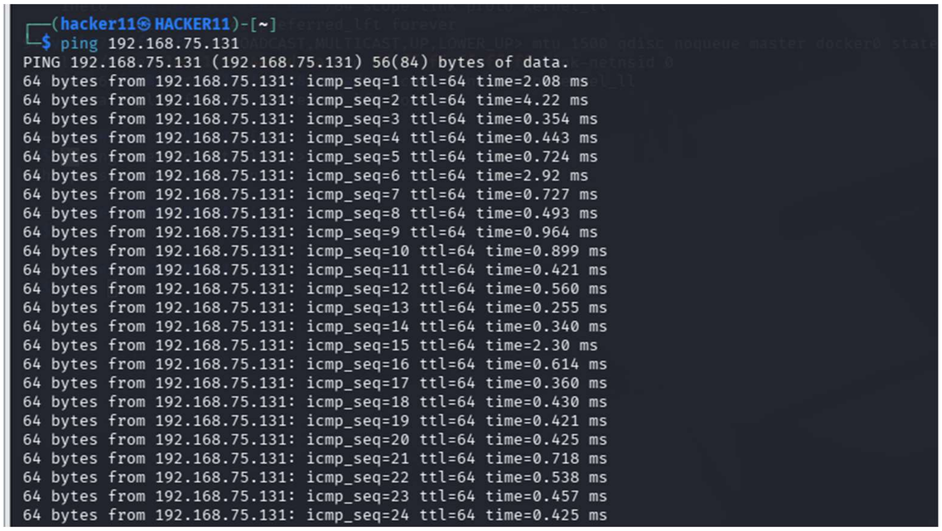
No.	Vulnerability Name	Affected Port / Service	CVSS Score	Severity	Likelihood	Impact	Overall Risk	Justification
1	Operating System (OS) End of Life (EOL)	General	10.0	Critical	High	High	**High**	Unsupported OS no longer receives security patches, making the system highly vulnerable to known exploits.
2	rlogin Passwordless Login	513/tcp	10.0	Critical	High	High	**High**	Allows attackers to gain root access without authentication, leading to full system compromise.
3	Possible Backdoor: Ingreslock	1524/tcp	10.0	Critical	High	High	**High**	Indicates presence of a potential backdoor which can allow unauthorized remote access.
4	Distributed Ruby (dRuby) Multiple RCE Vulnerabilities	8787/tcp	10.0	Critical	High	High	**High**	Remote Code Execution vulnerabilities can allow attackers to execute arbitrary commands remotely.
5	Apache Tomcat AJP RCE (Ghostcat)	8009/tcp	9.8	Critical	High	High	**High**	Exploitable vulnerability allowing file inclusion and remote code execution on the server.
6	PostgreSQL Default Credentials	5432/tcp	9.0	Critical	High	High	**High**	Use of default credentials allows easy unauthorized database access.
7	VNC Brute Force Login	5900/tcp	9.0	Critical	Medium	High	**High**	Weak authentication allows attackers to brute-force credentials and gain remote access.

8	FTP Unencrypted Cleartext Login	21/tcp	4.8	Medium	Medium	Medium	**Medium**	Credentials transmitted in plaintext can be intercepted using network sniffing attacks.
9	SSL/TLS Weak Cipher Suites	5432/tcp	5.0	Medium	Medium	Medium	**Medium**	Weak encryption algorithms increase the risk of data exposure.
10	Directory Browsing Enabled	80/tcp	5.0	Medium	Medium	Low	**Medium**	Allows attackers to view directory contents, exposing sensitive files or information.

11. Exploitation – Metasploit

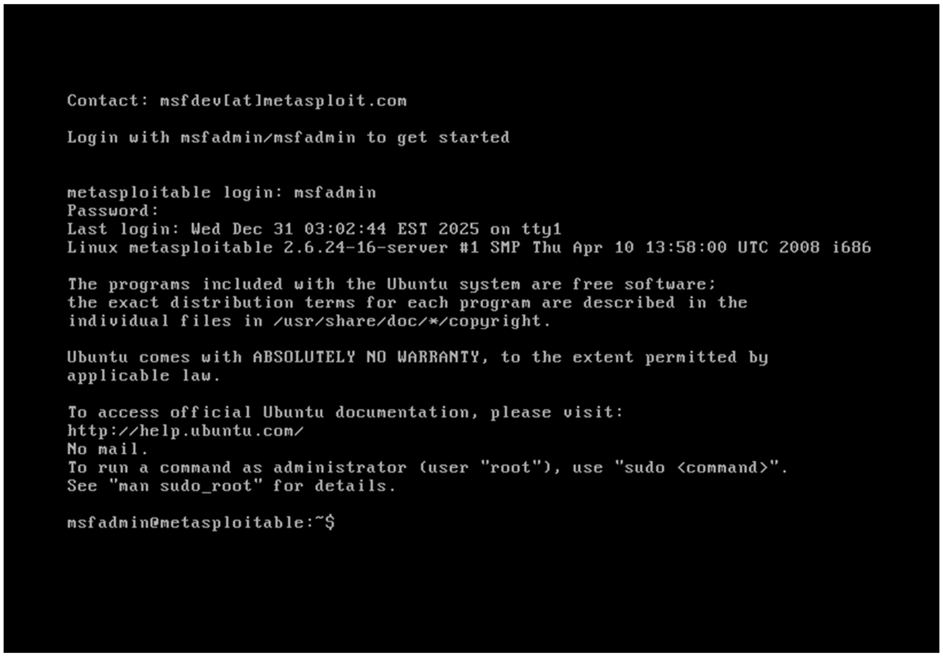
I tested selected critical vulnerabilities using the Metasploit Framework in a controlled lab environment to verify whether they could be exploited. Through this testing, I confirmed that the identified vulnerabilities were exploitable and could allow an attacker to gain unauthorized access or take control of the system if they remain unpatched.

Screenshot 14: Metasploit exploit setup



```
┌──(hacker11㉿HACKER11)-[~]
└─$ ping 192.168.75.131
PING 192.168.75.131 (192.168.75.131) 56(84) bytes of data.
64 bytes from 192.168.75.131: icmp_seq=1 ttl=64 time=2.08 ms
64 bytes from 192.168.75.131: icmp_seq=2 ttl=64 time=4.22 ms
64 bytes from 192.168.75.131: icmp_seq=3 ttl=64 time=0.354 ms
64 bytes from 192.168.75.131: icmp_seq=4 ttl=64 time=0.443 ms
64 bytes from 192.168.75.131: icmp_seq=5 ttl=64 time=0.724 ms
64 bytes from 192.168.75.131: icmp_seq=6 ttl=64 time=2.92 ms
64 bytes from 192.168.75.131: icmp_seq=7 ttl=64 time=0.727 ms
64 bytes from 192.168.75.131: icmp_seq=8 ttl=64 time=0.493 ms
64 bytes from 192.168.75.131: icmp_seq=9 ttl=64 time=0.964 ms
64 bytes from 192.168.75.131: icmp_seq=10 ttl=64 time=0.899 ms
64 bytes from 192.168.75.131: icmp_seq=11 ttl=64 time=0.421 ms
64 bytes from 192.168.75.131: icmp_seq=12 ttl=64 time=0.560 ms
64 bytes from 192.168.75.131: icmp_seq=13 ttl=64 time=0.255 ms
64 bytes from 192.168.75.131: icmp_seq=14 ttl=64 time=0.340 ms
64 bytes from 192.168.75.131: icmp_seq=15 ttl=64 time=2.30 ms
64 bytes from 192.168.75.131: icmp_seq=16 ttl=64 time=0.614 ms
64 bytes from 192.168.75.131: icmp_seq=17 ttl=64 time=0.360 ms
64 bytes from 192.168.75.131: icmp_seq=18 ttl=64 time=0.430 ms
64 bytes from 192.168.75.131: icmp_seq=19 ttl=64 time=0.421 ms
64 bytes from 192.168.75.131: icmp_seq=20 ttl=64 time=0.425 ms
64 bytes from 192.168.75.131: icmp_seq=21 ttl=64 time=0.718 ms
64 bytes from 192.168.75.131: icmp_seq=22 ttl=64 time=0.538 ms
64 bytes from 192.168.75.131: icmp_seq=23 ttl=64 time=0.457 ms
64 bytes from 192.168.75.131: icmp_seq=24 ttl=64 time=0.425 ms
```

Screenshot 16: Successful exploitation result



```
Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password:
Last login: Wed Dec 31 03:02:44 EST 2025 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

msfadmin@metasploitable:~$
```

12. Security Standards & Compliance

GDPR (General Data Protection Regulation)
GDPR is mainly about keeping personal and sensitive data safe. It expects systems to be properly configured so that only authorized users can access data. During this assessment, open ports and exposed services were found, which could allow attackers to enter the system and access data in a real environment. Finding and fixing these issues helps protect data and follows the basic idea of GDPR.

HIPAA (Health Insurance Portability and Accountability Act)
HIPAA focuses on protecting sensitive information by ensuring confidentiality, integrity, and availability. It requires strong login controls and secure communication methods. In this assessment, insecure services like Telnet and outdated FTP were identified, which can expose usernames and passwords. Detecting such weaknesses shows why secure configurations are important, especially in systems that handle sensitive data.

ISO/IEC 27001 (Information Security Management System)
ISO/IEC 27001 is a standard that helps organizations manage security risks in a structured way. It requires identifying risks, understanding their impact, and taking steps to reduce them. In this project, vulnerability scanning and basic

exploitation were used to find technical risks. The risk assessment table in the report matches this approach by ranking issues based on their severity.

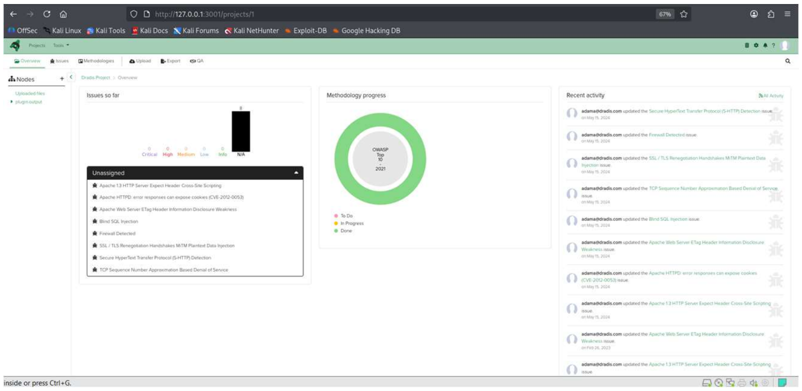
OWASP Top 10 (Web Application Security)

OWASP Top 10 is a widely used guide that lists the most common and serious web security problems. It was used to understand and prioritize issues found during web scanning with Nikto. By matching vulnerabilities to OWASP Top 10 categories, the findings are easier to understand and relate to real attack methods.

Vulnerability Mapping: OWASP Top 10 + ISO 27001

Identified Vulnerability	OWASP Top 10 Mapping	ISO 27001 Control Mapping
Open FTP service (vsftpd)	A5: Security Misconfiguration	A.12.6 – Technical Vulnerability Management
Telnet enabled (cleartext login)	A2: Cryptographic Failures	A.9 – Access Control
Outdated Apache HTTP Server	A5: Security Misconfiguration	A.12.6.1 – Management of Technical Vulnerabilities
Exposed MySQL/PostgreSQL ports	A4: Insecure Design	A.13 – Network Security
Weak or default credentials	A7: Identification & Authentication Failures	A.9.2 – User Access Management
Vulnerable Tomcat service	A5: Security Misconfiguration	A.14 – System Acquisition, Development & Maintenance
Web server misconfigurations (Nikto)	A5: Security Misconfiguration	A.13.1 – Network Security Controls
Successful exploitation via Metasploit	A3: Injection / A8: Software Integrity Failures	A.12 – Operations Security

Dradis OWASP methodology progress screen



Issues so far

Critical	High	Medium	Low	Info	N/A
0	0	0	0	0	8

Unassigned

- Apache 1.3 HTTP Server Expect Header Cross-Site Scripting
- Apache HTTPD: error responses can expose cookies (CVE-2012-0053)
- Apache Web Server ETag Header Information Disclosure Weakness
- Blind SQL Injection
- Firewall Detected
- SSL / TLS Renegotiation Handshakes MiTM Plaintext Data Injection
- Secure HyperText Transfer Protocol (S-HTTP) Detection
- TCP Sequence Number Approximation Based Denial of Service

Methodology progress

OWASP Top 10 - 2021

- To Do
- In Progress
- Done

Recent activity

adama@dradis.com updated the Secure HyperText Transfer Protocol (S-HTTP) Detection issue.
on May 15, 2024

adama@dradis.com updated the Firewall Detected issue.
on May 15, 2024

adama@dradis.com updated the SSL / TLS Renegotiation Handshakes MiTM Plaintext Data Injection issue.
on May 15, 2024

adama@dradis.com updated the TCP Sequence Number Approximation Based Denial of Service issue.
on May 15, 2024

adama@dradis.com updated the Blind SQL Injection issue.
on May 15, 2024

adama@dradis.com updated the Apache Web Server ETag Header Information Disclosure Weakness issue.
on May 15, 2024

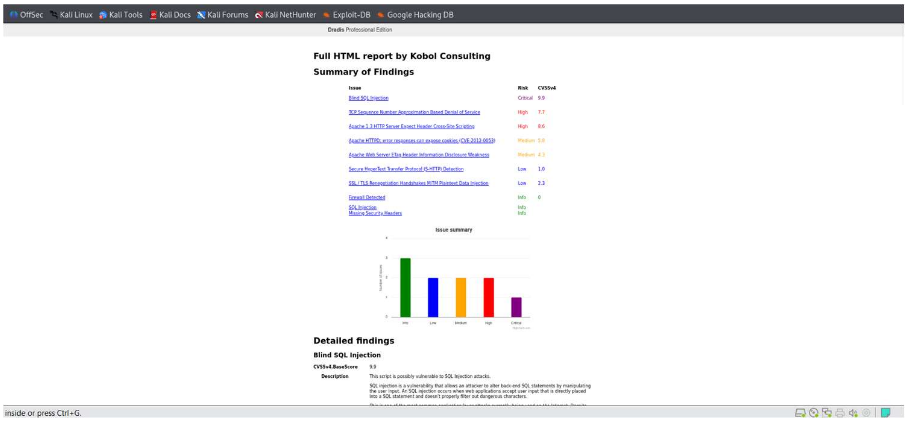
adama@dradis.com updated the Apache HTTPD: error responses can expose cookies (CVE-2012-0053) issue.
on May 15, 2024

adama@dradis.com updated the Apache 1.3 HTTP Server Expect Header Cross-Site Scripting issue.
on May 15, 2024

adama@dradis.com updated the Apache Web Server ETag Header Information Disclosure Weakness issue.
on Feb 26, 2023

adama@dradis.com updated the Apache 1.3 HTTP Server Expect Header Cross-Site Scripting issue.

Web Vulnerability Report Aligned with OWASP Standards



Full HTML report by Kobol Consulting

Summary of Findings

Issue	Risk	CVSSv4
Blind SQL Injection	Critical	9.9
TCP Sequence Number Approximation Based Denial of Service	High	7.7
Apache 1.3 HTTP Server Expect Header Cross-Site Scripting	High	8.6
Apache HTTPD: error responses can expose cookies (CVE-2012-0053)	Medium	5.8
Apache Web Server ETag Header Information Disclosure Weakness	Medium	4.3
Secure HyperText Transfer Protocol (S-HTTP) Detection	Low	1.0
SSL / TLS Renegotiation Handshakes MiTM Plaintext Data Injection	Low	2.3
Firewall Detected	Info	0
SQL Injection	Info	
Missing Security Headers	Info	

Detailed findings

Blind SQL Injection

CVSSv4.BaseScore 9.9

Description This script is possibly vulnerable to SQL Injection attacks.

SQL injection is a vulnerability that allows an attacker to alter back-end SQL statements by manipulating the user input. An SQL injection occurs when web applications accept user input that is directly placed into a SQL statement and doesn't properly filter out dangerous characters.

13. Remediation Recommendations

To improve the security of the system, it is important to apply the latest security patches and software updates to fix known vulnerabilities. Unnecessary and insecure services should be disabled to reduce the attack surface, and all default credentials must be removed to prevent unauthorized access. Secure communication protocols should be used to protect data in transit. In addition, regular vulnerability scans should be performed to identify new security issues, and proper access control and monitoring mechanisms should be implemented to detect and respond to suspicious activities.

14. Challenges Faced

I faced several challenges during this task, especially while setting up and configuring OpenVAS for the first time. The initial configuration took longer than expected because the vulnerability feeds required proper synchronization, and some services were not immediately available. I also faced difficulty in understanding the vulnerability severity levels and CVSS scores during the first few scans, as it was challenging to differentiate between critical, high, medium, and low risks and their actual impact on the system.

Another challenge I faced was related to **understanding and interpreting the vulnerability scan results** generated by the tools. During the initial scans, the large number of vulnerabilities and technical terms made it difficult to clearly understand which issues were critical and required immediate attention. Differentiating between false positives and real security risks was also challenging. I resolved this difficulty by carefully reviewing vulnerability descriptions, CVSS scores, and severity levels provided by the tools.

15. Key Learnings

I learned how a complete security assessment and VAPT process is performed, starting from planning and scanning to exploitation and final reporting. I gained hands-on experience with open-source security tools and learned how vulnerabilities are discovered, analyzed, and validated in a safe lab environment. I also learned how to prioritize risks using CVSS scores and severity levels instead of treating all vulnerabilities equally. In addition, I gained an understanding of the importance of clear documentation and professional reporting in cybersecurity assessments.

16. Conclusion

I learned how a complete security assessment lifecycle is carried out using open-source tools, from planning and scanning to exploitation and final reporting. I realized the importance of combining theoretical knowledge with hands-on practice to clearly understand real-world security testing. I also learned how vulnerabilities are identified, validated, and prioritized based on their risk and impact. Overall, this task helped me build confidence in performing security assessments and preparing professional cybersecurity reports.