Capstone Project — Full VAPT Cycle

Attacker: Kali Linux (192.168.1.15)
Target: Metasploitable2 + DVWA (192.168.1.7)
Tools: Nmap, Nikto, sqlmap, Metasploit, OpenVAS, tcpdump/Wireshark
Test dates: 2025-09-07 (evidence timestamps included below)

Report

Executive Summary:
On 2025-09-07 a penetration test was conducted from Kali (192.168.1.15) against host 192.168.1.7 (Metasploitable2 + DVWA) to evaluate the external attack surface, validate exploitable vulnerabilities, and produce prioritized remediation recommendations. Reconnaissance and automated scanning identified numerous legacy services and a vulnerable web application. Exploitation confirmed both web-appive data exposure and remote code execution on the host, demonstrating a high risk of data compromise and full system control by an attacker.

Findings:
Active service discovery (Nmap) revealed open FTP (vsftpd 2.3.4), Apache/PHP (Apache 2.2.8, PHP 5.2.4), Tomcat (8180), Samba, MySQL and other legacy services. Nikto enumerated exposed administration pages and phpinfo. Web testing (DVWA, Security=Low) and `sqlmap` validated a SQL Injection in the `id` parameter, allowing enumeration of the `dvwa` database (evidence: `~/capstone_evidence/sqlmap_dvwa/`, sqlmap log @ 2025-09-07 10:37). Metasploit exploitation of Tomcat manager successfully deployed a WAR and opened a Java Meterpreter session (evidence: `~/capstone_evidence/msf/msf_tomcat_deploy_*.log`, session at 2025-09-07 10:27:41). vsftpd and other legacy services are exploitable via known public modules.

Recommendations:
Immediate priorities: patch or remove vulnerable services (vsftpd, Apache/PHP, Tomcat), mitigate SQL injection by using parameterized queries and input validation, restrict management interfaces (phpMyAdmin/Tomcat) to trusted admin networks, and enforce strong authentication. Implement network segmentation, regular automated scanning

(OpenVAS), and a formal patch management process. Re-scan after remediation to verify closure.

Non-Technical Briefing (≈100 words)

A simulated security test of the lab server at 192.168.1.7 discovered critical weaknesses enabling attackers to read and modify data and, in some cases, take full control of the system. The web application (DVWA) had a database injection flaw allowing database enumeration, and multiple server components were old or misconfigured (FTP, web server, Tomcat). These issues could lead to data theft, service disruption, or unauthorized access. Immediate actions: remove or patch outdated services, fix the web application (input validation and prepared statements), restrict administrative interfaces to internal networks, and schedule routine automated scans to prevent recurrence.

Findings (table — copy into report)

Timestamp (UTC/local)	Target IP	Vulnerability / Vector	Tool / Evidence (files)
2025-09-07 09:36	192.168.1.7	Network scan — many legacy services found	`~/capstone_evidence/nmap/ms2_dvwa_full.nmap`
2025-09-07 09:41	192.168.1.7	Web server issues (phpinfo, phpMyAdmin, dir listing)	`~/capstone_evidence/nikto/ms2_nikto.txt`
2025-09-07 10:37	192.168.1.7	SQL Injection (DVWA `id` parameter) — DB enumeration	`~/capstone_evidence/sqlmap_dvwa/` (sqlmap logs)

2025-09-07 10:27:41	192.168.1.7	Tomcat manager — WAR deployed → Java Meterpreter (RCE)	`~/capstone_evidence/msf/msf_tomcat_deploy_*.log`, `screenshots/screenshot_tomcat_session.png`
2025-09-07 (scan)	192.168.1.7	vsftpd 2.3.4 backdoor (known RCE)	`~/capstone_evidence/msf/msf_vsftpd_*.log`, `screenshots/screenshot_vsftpd.png`

Note: include full OpenVAS export PDFs (before/after) in the evidence folder: `~/capstone_evidence/openvas/ms2_dvwa_openvas_before.pdf` and `..._after.pdf`.

Remediation & Verification (concise steps)

1. **Immediate (Critical)**
    a. Take the host offline from production-equivalent networks or isolate via firewall rules.
    b. Remove or disable unnecessary legacy services (FTP, rsh, distccd) and update packages to supported versions.
2. **Web Application (High priority)**
    a. Fix SQL Injection: use prepared statements/parameterized queries; validate & sanitize input server-side.
    b. Remove public phpMyAdmin or restrict to admin IPs + require strong authentication and MFA where possible.
    c. Set DVWA security to production settings or replace the app with a patched codebase.
3. **Application Servers & Admin Interfaces**
    a. Update Tomcat; change/disable manager credentials; restrict manager to admin network and require strong credentials. Remove uploaded WARs used during testing.
    b. Update Apache/PHP to supported versions; disable directory indexing; remove test pages (phpinfo.php).
4. **Hardening & Process**
    a. Enforce least privilege for services and users.

b. Implement host-based firewall rules and network segmentation to limit lateral movement.
c. Adopt a patch-management schedule and automated vulnerability scanning (OpenVAS/Nessus) weekly/monthly.

5. **Verification**
    a. After remediation, re-run Nmap, OpenVAS and application-level tests (sqlmap/Burp) and produce before/after reports and screenshots. Save to evidence folder (`openvas/..._after.pdf`, `nmap/..._rescan.nmap`).

Evidence & Appendix (what to attach to Google Doc)

- `~/capstone_evidence/nmap/` — Nmap scans (initial and rescan)
- `~/capstone_evidence/nikto/ms2_nikto.txt` — web scan output
- `~/capstone_evidence/sqlmap_dvwa/` — sqlmap logs & DB dumps (proof)
- `~/capstone_evidence/msf/` — Metasploit transcripts (vsftpd, tomcat logs)
- `~/capstone_evidence/pcaps/` — PCAPs captured during exploits
- `~/capstone_evidence/screenshots/` — ordered screenshots (recon → vuln → exploit → post-exploit)
- OpenVAS exported PDF reports (before & after remediation)