Title: Full VAPT Report - DVWA Application
Target: Damn Vulnerable Web App (DVWA) on localhost
Tools: sqlmap, OpenVAS

PTES Phase Log:

Timestamp	Target IP	Vulnerability	PTES Phase
2025-08-28 14:16:00	127.0.0.1	SQL Injection (Blind)	Exploitation
2025-08-28 14:17:00	127.0.0.1	Database Dump & Password Cracking	Post-Exploitation

Report:

1. Introduction: This engagement involved a full VAPT cycle against the Damn Vulnerable Web Application (DVWA) running locally. The objective was to identify and exploit the SQL Injection vulnerability.

2. Findings: A critical SQL Injection vulnerability was identified in the id parameter of the DVWA application. The vulnerability was of type Union-based and Time-based blind, allowing for complete database enumeration.

3. Exploitation: The tool sqlmap was used to automate the exploitation. The attack successfully:

- Identified the backend database as MySQL.
- Enumerated database names, confirming the dvwa database.
- Dumped the entire structure of the dvwa database, including the users table.
- Extracted all user records, including hashed passwords.
- Cracked the weak MD5 password hashes, revealing passwords like password, charley, and letmein.

4. Remediation: The root cause is a lack of input sanitization. It is critically recommended to:

- Use parameterized queries (prepared statements) to prevent SQL injection.
- Implement strong password policies to avoid weak, crackable passwords.
- Store passwords using strong, salted hashing algorithms (e.g., bcrypt).

5. Rescan: A follow-up scan after code remediation is mandatory to confirm the fix.

Non-Technical Summary :

During our security test of the training website, we found a severe flaw that allowed us to trick the website into giving us its entire user database. This included everyone's usernames and cracked passwords, some of which were very simple like "password". This happened because the website wasn't properly checking the information we sent it. We recommend the development team fix the website's code to properly validate all user input and ensure stronger passwords are used. This will prevent a real attacker from stealing sensitive user information.

Evidence: See screenshots 08_sqlmap_Injection_Found.png and 09_sqlmap_Password_Cracking.png.