Professional Penetration Test Report

Target: `testphp.vulnweb.com`
Date: April 30, 2025
Prepared by: [Your Name / Organization]

---

1. Executive Summary

A penetration test was conducted against `testphp.vulnweb.com`, revealing a **critical SQL injection vulnerability** in the `artist` parameter of the `artists.php` page. Exploitation of this flaw allowed unauthorized access to sensitive database records, including user credentials (`pass`, `email`, `cc`).

Key Findings:

- **SQL Injection (Boolean-based blind)** in `artist` (GET parameter).
- **Database Compromise:** Full extraction of table structures and data from the `acuart` database.
- **Sensitive Data Exposure:** Usernames, passwords (`test`), email addresses, and potential credit card information (`cc` column).

Risk Rating:

- **CVSS Score:** 9.8 (Critical)
- **Impact:** High (Data breach, application compromise)
- **Exploitability:** Trivial (No authentication required)

Recommendations Summary:

1. **Patch SQL Injection** using parameterized queries.
2. **Upgrade PHP** (5.6.40 is end-of-life).
3. **Implement WAF** rules to block injection attempts.

---

2. Methodology

Scope & Approach

- **Target:** `http://testphp.vulnweb.com/artists.php?artist=1`
- **Tools:** `sqlmap` (v1.8.3) for automated exploitation.
- **Safety Measures:**
  - Testing limited to the provided test environment.
  - No brute-forcing or disruptive attacks performed.

Steps Reproduced:

1. **Vulnerability Identification:**
   - Sent a malformed `artist=1` payload to detect SQLi.
   - **Screenshot 1:** sqlmap launch and legal disclaimer.
   - **Screenshot 2:** Confirmed injection point (`boolean-based blind`).
2. **Database Enumeration:**
   - Extracted database names (`acuart`, `information_schema`).
   - **Screenshot 3:** Database listing output.
3. **Table & Data Extraction:**
   - Dumped the `users` table structure (`name`, `pass`, `cc`, etc.).
   - Retrieved plaintext passwords (e.g., `test`).
   - **Screenshot 4:** Extracted password from `acuart.users`.

---

3. Findings

Vulnerability Details

Vulnerability	**Description**	**Evidence**
SQL Injection (GET)	Unsanitized `artist` parameter allows database queries.	Screenshot 1 (sqlmap detection).
Data Exposure	`acuart.users` table contains plaintext passwords and PII.	Screenshot 4 (`pass="test"`).
Outdated Stack	PHP 5.6.40 (EOL) with known vulnerabilities.	Screenshot 3 (server info).

Proof of Concept

1. **Exploit Command:**
2. bash
3. sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbs
4. **Extracted Data:**
    - Database: `acuart` → Table: `users` → Columns: `uname`, `pass`, `cc`, `email`.
    - **Screenshot 2 & 4:** Table schema and extracted credentials.

---

4. Recommendations

Immediate Actions:

1. **Input Validation:**
    - Use prepared statements (e.g., PDO, MySQLi) for all SQL queries.
    - Example fix:
    - php

$stmt = $pdo->prepare("SELECT * FROM artists WHERE id = ?");

   - $stmt->execute([$_GET['artist']]);

2. **Upgrade Software:**
    - Migrate from **PHP 5.6.40** to a supported version (≥8.0).
    - Update **Nginx 1.19.0** to the latest stable release.
3. **Sensitive Data Protection:**
    - Hash passwords using **bcrypt/Argon2** (plaintext `test` is unacceptable).
    - Encrypt `cc` (credit card) data if storage is necessary.

Long-Term Mitigations:

- **Web Application Firewall (WAF):** Deploy ModSecurity or Cloudflare to filter SQLi payloads.
- **Logging & Monitoring:** Alert on repeated SQLi attempts.

---

5. Conclusion

The SQL injection vulnerability in `testphp.vulnweb.com` poses a severe risk, enabling full database compromise. While this test was conducted in a controlled environment, real-world exploitation could lead to **data breaches** or **system takeover**.

Evidence Attachments:

- **Screenshot 1:** sqlmap execution and injection confirmation.
- **Screenshot 2:** `users` table structure.
- **Screenshot 3:** Database enumeration (`acuart`).
- **Screenshot 4:** Extracted password (`test`).

Next Steps:

- Patch vulnerabilities within **7 days**.
- Conduct a retest after remediation.

Report End

Appendices

A. Screenshot Index

Ref	Description
1	sqlmap launch and injection point detection.
2	`users` table schema (columns: `pass`, `cc`, etc.).
3	Database list (`acuart`, `information_schema`).
4	Extracted password (`test`) from `acuart.users`.

B. Tool Output Logs

Full sqlmap logs available at:
`/home/user/.local/share/sqlmap/output/testphp.vulnweb.com`