
Cybersecurity Internship Report

Web Application: User Management System
Submitted by: Syed Muhammad Hashir
Date: April 20, 2025

✓ Week 1: Security Assessment Report

1. Vulnerabilities Found

1.1 SQL Injection

- **Payload**: `admin' OR '1'='1`
- **Impact**: Bypassed login; unauthorized access.
- **Fix**: Use prepared statements with parameterized queries.

1.2 Reflected Cross-Site Scripting (XSS)

- **Payload**: `<script>alert('XSS')</script>`
- **Impact**: Script executed in the victim's browser.
- **Fix**: Implement HTML output encoding and CSP headers.

---

2. Security Misconfigurations

2.1 Plaintext Password Storage

- **Impact**: If DB is compromised, credentials are exposed.
- **Fix**: Replace with bcrypt hashing (with salt).

Note:
My code changes the plaintext password to bcrypt when the password is used to log in.

---

3. Areas of Improvement

- Enforce input validation & sanitization.
- Implement strong password policies & rate limiting.
- Add security headers (CSP, X-XSS-Protection, etc.).
- Apply least privilege to DB access.

✅ Week 2: Security Implementation Report

1. Fixes Implemented

✅ SQL Injection

- Replaced raw SQL with **prepared statements**:

```
$stmt = $con->prepare("SELECT id, fname, password FROM users WHERE email =
?");
$stmt->bind_param("s", $useremail);
```

✅ Input Validation

- Used `filter_var()` to validate email format:

```
if (!filter_var($useremail, FILTER_VALIDATE_EMAIL)) { ... }
```

✅ Password Hashing & Transition

- Implemented `password_verify()` and `password_hash()`:

```
if (password_verify($password, $user['password']) || $password ===
$user['password']) {
    if ($password === $user['password']) {
        $newHash = password_hash($password, PASSWORD_DEFAULT);
        ...
    }
}
```

- Automatically rehashes plaintext passwords to bcrypt on successful login.

---

2. Authentication Enhancements

- Used PHP sessions to track login securely.
- Dual-mode verification allows smooth migration from insecure passwords.
- Added redirect on successful authentication.

✅ Week 3: Advanced Security & Final Reporting (Partial)

1. Basic Penetration Testing

🔍 Nmap Scan Result

Command used:

```
nmap -sV localhost
```

Key Findings:

Port	Service	Version/Details
21	FTP	FileZilla 0.9.41 – insecure if not using SFTP.
80	HTTP	Apache 2.4.56 with PHP 8.2.4.
443	HTTPS	Secure — SSL on Apache.
135	Microsoft RPC	Common in Windows — patch regularly.
445	Microsoft DS	Vulnerable to SMB-based attacks — disable if not needed.
2869	SSDP/UPnP	Risky on public networks.
3306	MariaDB	Running on default port — ensure strong passwords.
1001	Unknown	Needs further investigation.

Suggestions:

- Disable unused services (e.g., FTP, SMB).
- Restrict MySQL access to `localhost` only.
- Apply firewall rules and patch services.
