Website Admin Panel Security Testing Mastery Guide (Day 1-30)

This guide covers ethical security testing of web admin panels, organized day-by-day with commands, purposes, and tips for authorized testing.

Day 1: Setup Lab Environment

- **Command:** `docker run -d -p 8080:80 vulnerables/web-goat`
- **Purpose:** Deploy a vulnerable web application for testing.

Day 2: Install Tools

- **Command:** `apt install burpsuite nikto gobuster nmap`
- **Purpose:** Set up essential security testing tools.

Day 3: Identify Admin URLs

- **Command:** `gobuster dir -u http://localhost:8080 -w /usr/share/wordlists/dirb/common.txt`
- **Purpose:** Discover hidden directories like `/admin`.

Day 4: Scan for Vulnerabilities

- **Command:** `nikto -h http://localhost:8080`
- **Purpose:** Scan for known vulnerabilities in web server and admin panel.

Day 5: Map Web Application

- **Command:** `nmap -p 80,443 --script http-enum localhost`
- **Purpose:** Enumerate web application endpoints.

Day 6: Test Login Page Response

- **Command:** `curl -I http://localhost:8080/admin`
- **Purpose:** Check HTTP headers and authentication requirements.

Day 7: Check HTTP Methods

- **Command:** `curl -X OPTIONS http://localhost:8080/admin -i`
- **Purpose:** Identify allowed HTTP methods.

Day 8: Enumerate Users (Lab Only)

- **Command:** `hydra -L users.txt -P passwords.txt localhost http-post-form "/login:username=^USER^&password=^PASS^:F=incorrect"`
- **Purpose:** Test login security with a wordlist.

Day 9: Test for Default Credentials

- **Command:** `curl -u admin:admin http://localhost:8080/admin`
- **Purpose:** Check if default credentials are still valid.

Day 10: Check Session Cookies

- **Command:** `curl -I http://localhost:8080/admin`
- **Purpose:** Review cookie flags like Secure, HttpOnly.

Day 11: SQL Injection Test

- **Command:** `sqlmap -u "http://localhost:8080/admin?id=1" --batch`
- **Purpose:** Test input fields for SQL injection vulnerabilities.

Day 12: Cross-Site Scripting (XSS) Test

- **Command:** `xsser -u "http://localhost:8080/admin/search?q=test"`
- **Purpose:** Test for XSS vulnerabilities in input fields.

Day 13: CSRF Token Check

- **Command:** Inspect token in `curl -c cookie.txt -b cookie.txt` request headers.
- **Purpose:** Verify CSRF protections are implemented.

Day 14: Analyze Page Source

- **Command:** `curl http://localhost:8080/admin -o admin.html`
- **Purpose:** Review HTML for hidden fields, endpoints, or comments.

Day 15: Check SSL/TLS

- **Command:** `nmap --script ssl-cert -p 443 localhost`
- **Purpose:** Inspect certificate details and encryption strength.

Day 16: Scan for Subdomains

- **Command:** `sublist3r -d localhost`
- **Purpose:** Identify subdomains or admin portals.

Day 17: Test Directory Traversal

- **Command:** `curl http://localhost:8080/admin/../../etc/passwd`
- **Purpose:** Check if path traversal is possible.

Day 18: Test File Upload Security

- **Command:** `curl -F "file=@test.txt" http://localhost:8080/admin/upload`
- **Purpose:** Ensure file uploads are properly restricted.

Day 19: Test IDOR

- **Command:** `curl http://localhost:8080/admin/user/2`
- **Purpose:** Check if users can access others' data.

Day 20: Enumerate Admin Panel APIs

- **Command:** `nmap --script http-enum -p 80,443 localhost`
- **Purpose:** Identify exposed API endpoints.

Day 21: Automated Vulnerability Scan

- **Command:** `nikto -h http://localhost:8080 -o nikto_results.txt`
- **Purpose:** Save scan results for review.

Day 22: Analyze JavaScript Files

- **Command:** `wget -r -l1 -A .js http://localhost:8080/admin`
- **Purpose:** Inspect JS for hidden functionality or sensitive endpoints.

Day 23: Check HTTP Security Headers

- **Command:** `curl -I http://localhost:8080/admin`
- **Purpose:** Verify headers like X-Frame-Options, Content-Security-Policy.

Day 24: Test Authentication Bypass

- **Command:** `curl -i -H "X-Forwarded-For: 127.0.0.1" http://localhost:8080/admin`
- **Purpose:** Test lab-based bypass attempts.

Day 25: Analyze Error Messages

- **Command:** `curl http://localhost:8080/admin/nonexistent`
- **Purpose:** Ensure error messages do not leak sensitive info.

Day 26: Test Rate Limiting

- **Command:** `for i in {1..50}; do curl -X POST -d "username=admin&password=test" http://localhost:8080/admin; done`
- **Purpose:** Test login rate limiting and lockout.

Day 27: Automated Admin Panel Enumeration

- **Command:** `wfuzz -c -w /usr/share/wordlists/dirb/common.txt --hc 404 http://localhost:8080/FUZZ`
- **Purpose:** Enumerate hidden directories/endpoints.

Day 28: Test Security Configurations

- **Command:** `nmap -p 80,443 --script http-security-headers localhost`
- **Purpose:** Verify security headers, TLS versions, and weak ciphers.

Day 29: Documentation

- **Task:** Compile findings, logs, screenshots, and commands.
- **Purpose:** Prepare professional pentesting report.

Day 30: Review & Cleanup

- **Task:** Revert VM snapshots and clean test data.
- **Purpose:** Ensure lab environment is safe and ready for next testing cycle.

Note: Only test web admin panels on systems you own or are explicitly authorized to assess. Unauthorized access is illegal.