Web Vulnerability Scanning Mastery Guide (Day 1-30)

This guide covers web vulnerability scanning tools, commands, and techniques, organized day-by-day for progressive learning.

Day 1: Whois Lookup

- **Command:** `whois example.com`
- **Purpose:** Retrieve domain registration details.

Day 2: DNS Resolution

- **Command:** `nslookup example.com`
- **Purpose:** Resolve domain to IP.

Day 3: DNS Records Lookup

- **Command:** `dig example.com any`
- **Purpose:** Obtain all DNS records.

Day 4: Host Availability

- **Command:** `ping example.com`
- **Purpose:** Check if the host is reachable.

Day 5: Network Path Mapping

- **Command:** `traceroute example.com`
- **Purpose:** Trace the route packets take to reach the host.

Day 6: Web Port Scan

- **Command:** `nmap -p 80,443 example.com`
- **Purpose:** Scan common web service ports.

Day 7: Service Version Detection

- **Command:** `nmap -sV example.com`
- **Purpose:** Identify versions of services running on open ports.

Day 8: Grab Web Page Title

- **Command:** `nmap --script=http-title example.com`
- **Purpose:** Retrieve title of the website.

Day 9: Enumerate Common Directories

- **Command:** `nmap --script=http-enum example.com`
- **Purpose:** Discover common web directories.

Day 10: Basic Web Server Vulnerability Scan

- **Command:** `nikto -h example.com`
- **Purpose:** Scan web server for known vulnerabilities.

Day 11: Directory Brute-force

- **Command:** `gobuster dir -u http://example.com -w wordlist.txt`
- **Purpose:** Discover hidden directories.

Day 12: Fast Fuzzing

- **Command:** `ffuf -u http://example.com/FUZZ -w wordlist.txt`
- **Purpose:** Quickly find hidden resources.

Day 13: Technology Detection

- **Command:** `whatweb example.com`
- **Purpose:** Identify technologies used by the site.

Day 14: WordPress Vulnerability Scan

- **Command:** `wpscan --url example.com`
- **Purpose:** Scan WordPress installations for vulnerabilities.

Day 15: Web Framework Detection

- **Command:** `wappalyzer-cli example.com`
- **Purpose:** Detect web frameworks and CMS.

Day 16: Automated SQL Injection Scan

- **Command:** `sqlmap -u "http://example.com/page?id=1" --batch`
- **Purpose:** Detect SQL injection vulnerabilities automatically.

Day 17: SQL Injection on Login Forms

- **Command:** `sqlmap -u "http://example.com/login" --data="username=admin&password=pass" --batch`
- **Purpose:** Test login forms for SQL injection.

Day 18: Weak Credentials Test

- **Command:** Use Hydra for HTTP/FTP/DB logins.
- **Purpose:** Identify weak passwords and default credentials.

Day 19: Enumerate Users & Roles

- **Command:** `sqlmap -u "http://example.com/page?id=1" --users --roles`
- **Purpose:** Determine user accounts and privileges.

Day 20: Session Management Test

- **Command:** Analyze cookies and tokens with Burp Suite.
- **Purpose:** Identify session management weaknesses.

Day 21: Cross-Site Scripting (XSS) Scan

- **Command:** Use OWASP ZAP/Burp Suite to test reflected, stored, and DOM XSS.
- **Purpose:** Identify XSS vulnerabilities.

Day 22: Cross-Site Request Forgery (CSRF) Scan

- **Command:** Use OWASP ZAP/Burp to test forms for CSRF tokens.
- **Purpose:** Detect missing CSRF protection.

Day 23: Input Validation & Open Redirects

- **Command:** Test URL parameters and forms for unsafe input handling.
- **Purpose:** Detect validation issues and redirect vulnerabilities.

Day 24: Security Headers Check

- **Command:** `curl -I https://example.com`
- **Purpose:** Check headers like CSP, X-Frame-Options, HSTS.

Day 25: File Inclusion / RCE Scan

- **Command:** Use payloads to test LFI/RFI vulnerabilities.
- **Purpose:** Detect potential file inclusion and remote code execution.

Day 26: API Endpoint Testing

- **Command:** Scan JSON/XML endpoints for parameter tampering, IDOR.
- **Purpose:** Test APIs for vulnerabilities.

Day 27: Automated Vulnerability Scripts

- **Command:** Combine Nikto, Gobuster, ffuf, sqlmap in scripts.
- **Purpose:** Automate web vulnerability scanning.

Day 28: Capture & Analyze HTTP Traffic

- **Command:** Use Burp Suite/OWASP ZAP or tcpdump.
- **Purpose:** Analyze requests for weaknesses.

Day 29: Advanced Exploitation Techniques

- **Command:** Test discovered vulnerabilities with safe payloads.
- **Purpose:** Validate exploitability of vulnerabilities.

Day 30: Reporting & Documentation

- **Command:** Compile findings into a penetration testing report.
- **Purpose:** Document vulnerabilities and remediation recommendations.

Note: Only scan websites you are authorized to test. Unauthorized scanning is illegal.