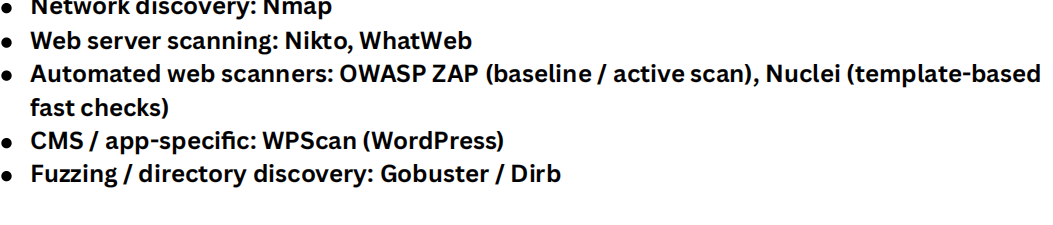
**Web Application Scanning: Automated**  **Vulnerability Discovery**

**BY CHESTA V DUBEY…**

**OBJECTIVE:**

l learn to perform safe, legal vulnerability scanning of a web application using automated scanners. You'll discover how to plan a scan, run and tune multiple scanners (light → aggressive), interpret results, remove false positives, and produce a professional vulnerability report with remediation advice.

**TOOLS USED(Kali Linux):**



1. **RECON AND DISCOVERY**

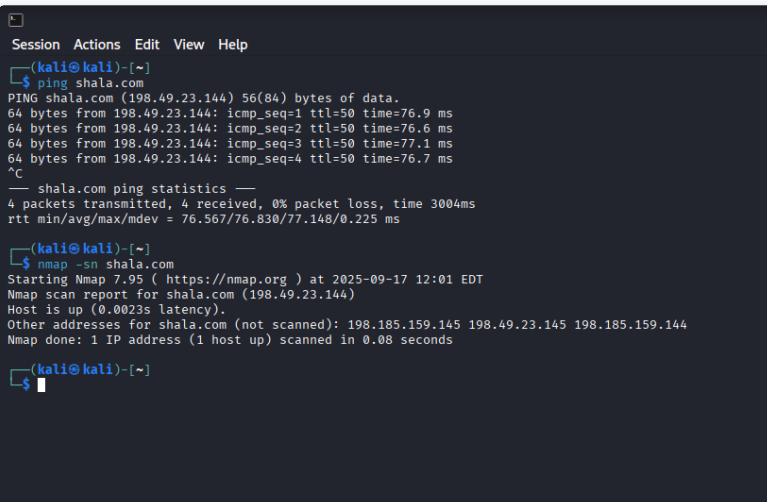
**STEP 1:HOST DISCOVERY**

**Command: i) ping shala.com**

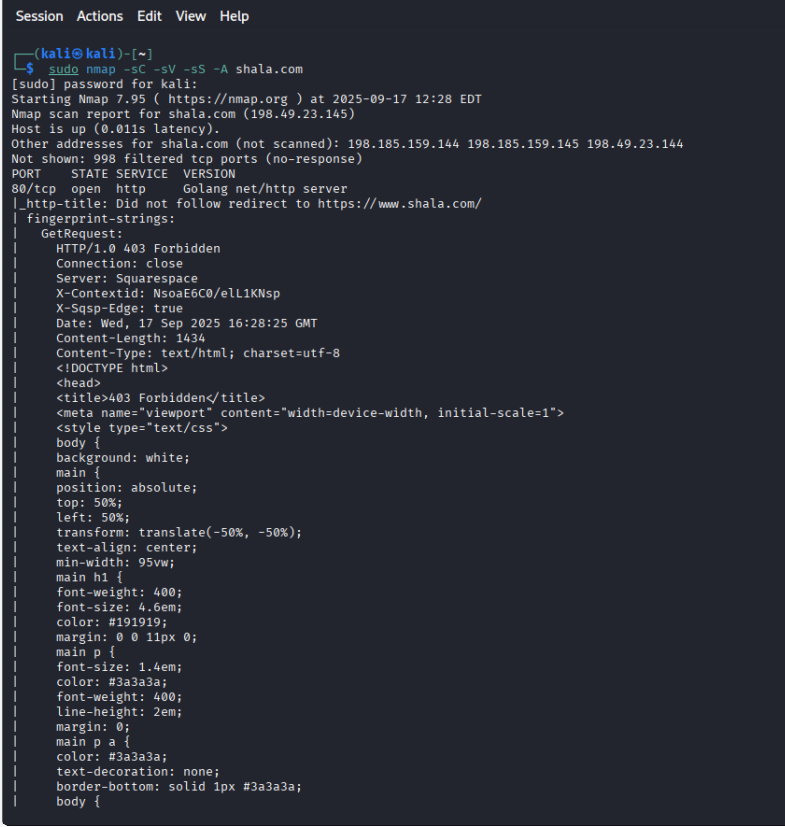
**Ii) nmap -sn shala.com**

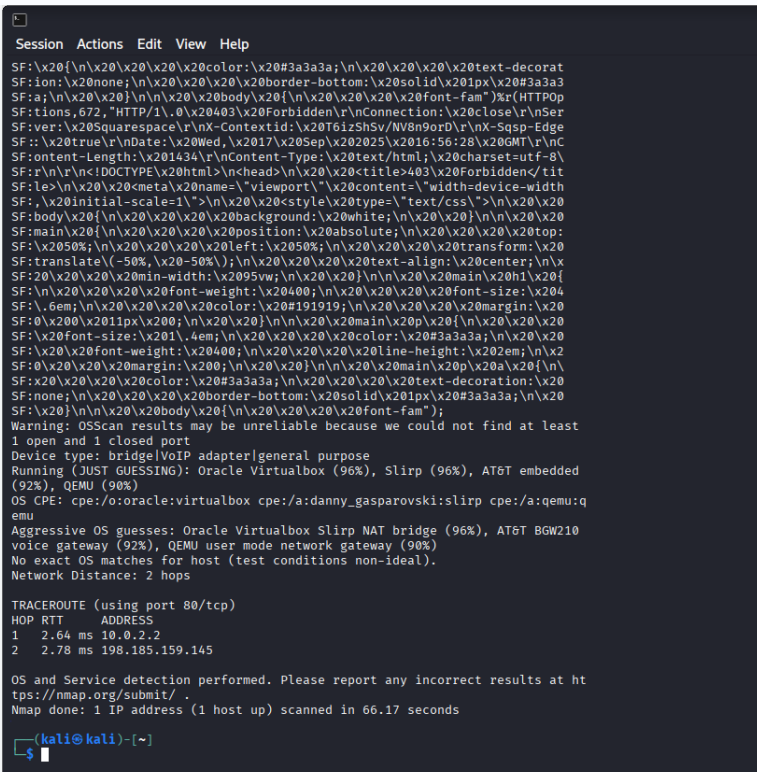
**Finding:** To find open port on target and runningh services and version and to perform half tcp scan.

**Screenshort:**



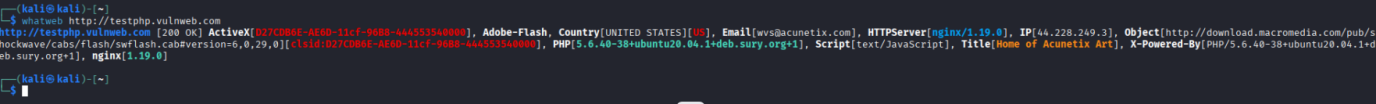
**Command:sudo nmap -sC -sV -sS -A shala.com**





Port:80,53and443 is open

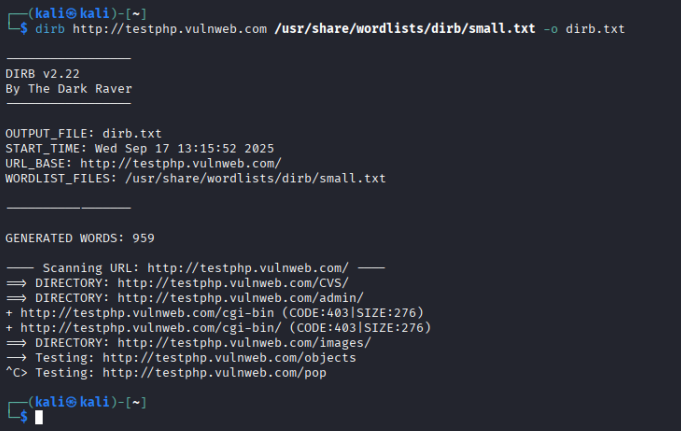
1. **TO IDENTIFY WEB TECHNOLOGY OF TARGET USE WHATWEB .**

**Command:whatweb http://testphp.vulnweb.com**

1. **Directory Discovery**

**Findings:** found multiple directories (/uploads/,/images/,/admins/) that could be explored later for sensitive files.

**Command:dirb http://testphp.vulnweb.com /usr/share/wordlists/dirb/small.txt -o dirb.txt**

**Screenshort:**

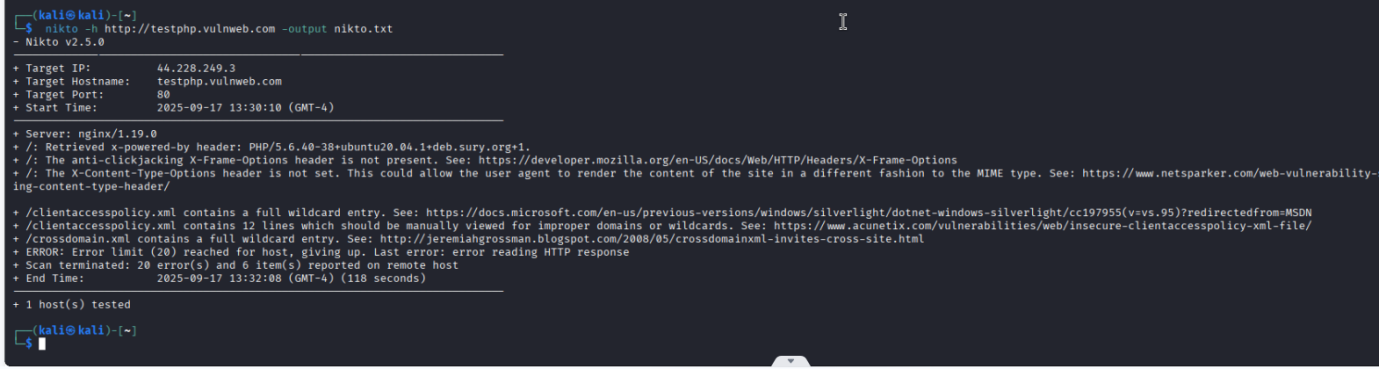
**STEP 2:LIGHT AUTOMATED SCANS**

1. **NIKTO-WEB VULNERABILILTY SCAN**

**Command: nikto -h http://testphp.vulnweb.com -output nikto.txt**

**Findings:** Missing security headers (X-Frame-Options, X-Content-Type-Options)

Possible directory indexing Outdated Apache version → potential CVEs

**Screenshort:**

1. **Nuclei — Template-based Checks**

**Commmad :nuclei -u http://testphp.vulnweb.com -as -o nuclei.txt -c 10**

**Finding:** Detected several low- to medium-severity issues such as missing headers and common misconfigurations.

**Screenshot:**



1. **OWASP ZAP (BASELINE SCAN)**

**Command:sudo docker run -v $(pwd):/zap/wrk/:rw --network="host" ghcr.io/zaproxy/zaproxy:stable zap-baseline.http://testphp.vulnweb.com -r zap\_baseline.html**

**Findings:**

14 warnings, including:

Missing anti-clickjacking, CSP, X-Content-Type-Options headers

Information leak via Server & X-Powered-By headers

Potential XSS in guestbook.php & search.php

Absence of anti-CSRF tokens on key pages

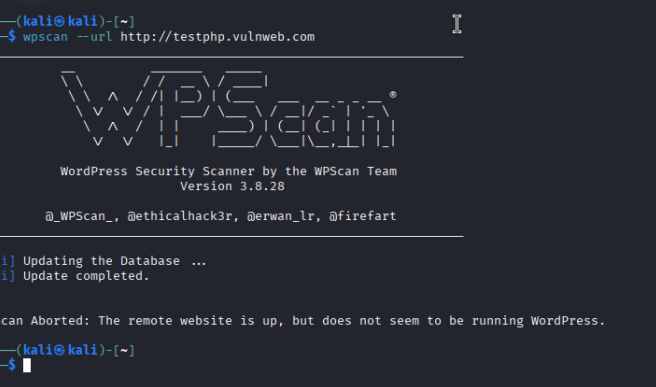
**Screenshort:**

1. **WPScan — WordPress Check**

**Command:wpscan --url <http://testphp.vulnweb.com>**

**Target reachable**

**No WordPress detected → Skipped plugin/theme/user enumeration**

**Screenshot:**

**Conclusion :**

From this project:

Target is live and running Apache/2.2.8 with PHP 5.2.4 (outdated → risk).

Key issues found:

Missing security headers (CSP, X-Frame-Options, HSTS)

Potential XSS in input fields

Server information disclosure (Server, X-Powered-By)

No anti-CSRF protections on sensitive forms

No WordPress detected, so WP-specific attacks are not applicable.

Takeaway: This scanning process maps the attack surface, identifies misconfigurations, and highlights priority fixes (e.g., apply latest Apache/PHP patches, implement security headers, validate input to prevent XSS).