Passive Reconnaissance Scan Report

Scan target : Halisans.com (66.29.153.49)

Date of recon: 20 Jan 2026

Cybersecurity Analyst
Onwusa Precious

Scope: halisans.com and publicly resolvable subdomains only (passive OSINT).

Out-of-scope: Active vulnerability exploitation, authenticated access and service disruption, rate-aggressive crawling.

Executive Summary

- The target domain resolves and serves a public website with recent content.
- This report enumerates: WHOIS/RDAP, authoritative DNS data, HTTPS surface (at a high level), presence of a WAF/CDN (fingerprinted passively) and open-source footprint across common OSINT sources.
- No intrusive scans were performed; all findings are from passive lookups and single-request fetches of public pages.

Methodology (Passive Only)

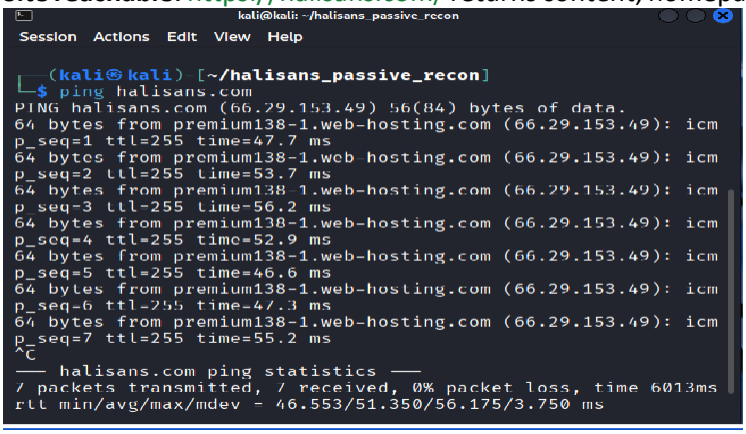
Tools & Modes

- **whois / RDAP:** Registration & registrar metadata
- **dig, host, dnsrecon:** Passive DNS lookups (A/AAAA, NS, MX, TXT/SOA/CAA where present) via public resolvers.
- **wafw00f:** Single HTTP(S) request fingerprint (headers/body markers) to infer WAF/CDN; *no evasion, no burst*.
- **SpiderFoot (SF):** *Passive modules only* (DNS, CT logs, WHOIS, netblocks, leak/site mentions, social).
- **Wapiti:** *Listing only* and passive banner/headers check.
- **OSINT Framework:** As a directory to guide passive pivoting, CT logs, public paste sites, reputation lists, search operators.

Findings

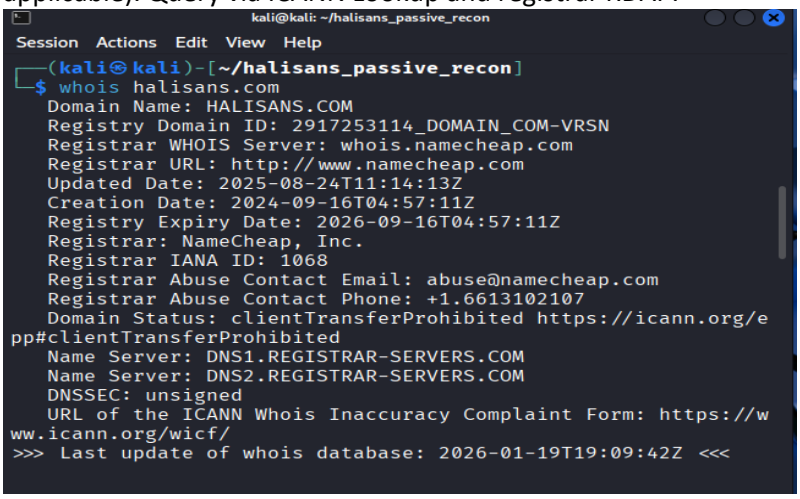
3.1 Public Web Presence (Landing Page)

- **Site reachable:** https://halisans.com/ returns content; homepage shows recent posts.



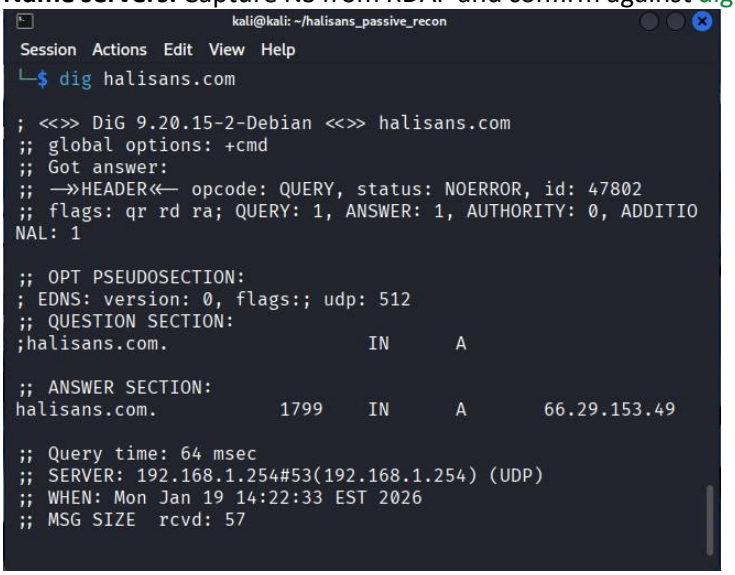
3.2 Registration (WHOIS)

- **Registrar / Dates:** Use ICANN RDAP as the primary source of truth (GDPR-redacted where applicable). Query via ICANN Lookup and registrar RDAP.



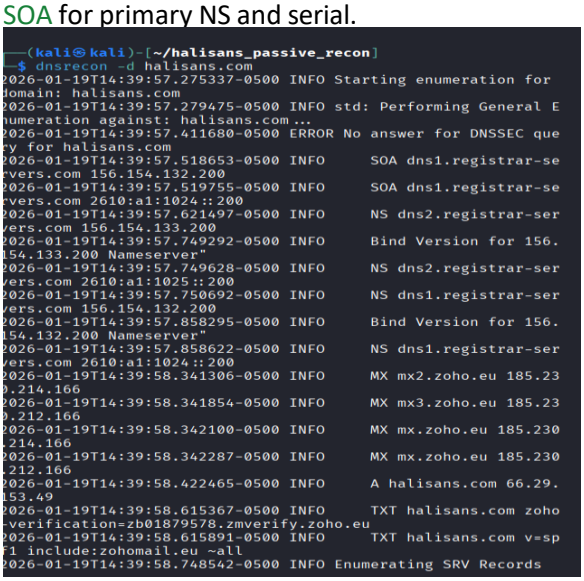
- vulnerability search against **Namecheap**, as a web application and domain registrar, was performed to identify any publicly disclosed CVEs directly associated with its services or infrastructure. By querying the CVE.org database using “Namecheap” as a keyword, the intent was to determine whether reported vulnerabilities affected Namecheap systems themselves or referenced Namecheap in a supporting or mitigation role.

- **Name servers:** Capture NS from RDAP and confirm against dig NS.



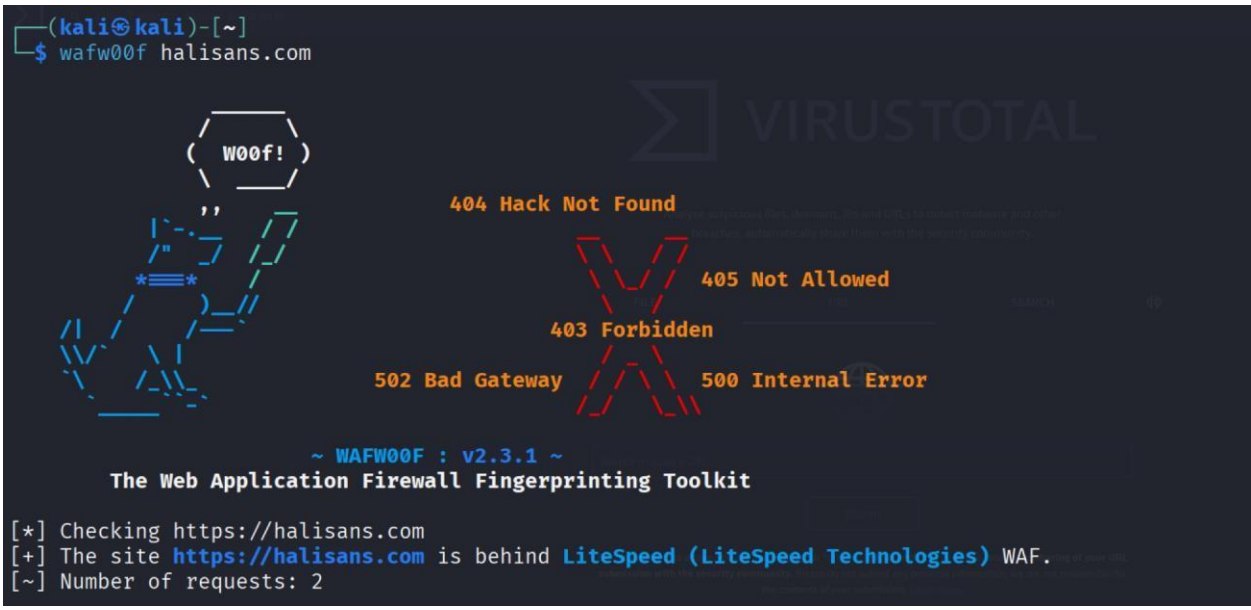
3.3 DNS Surface (A/AAAA, NS, MX, TXT, SOA, CAA)

- **Records collected passively:**
  - A/AAAA for apex and www.
  - NS to identify hosting/DNS provider.
  - MX for mail handling (and whether any third-party service is used).
  - TXT for SPF/DMARC/DKIM indicators.
  - SOA for primary NS and serial.



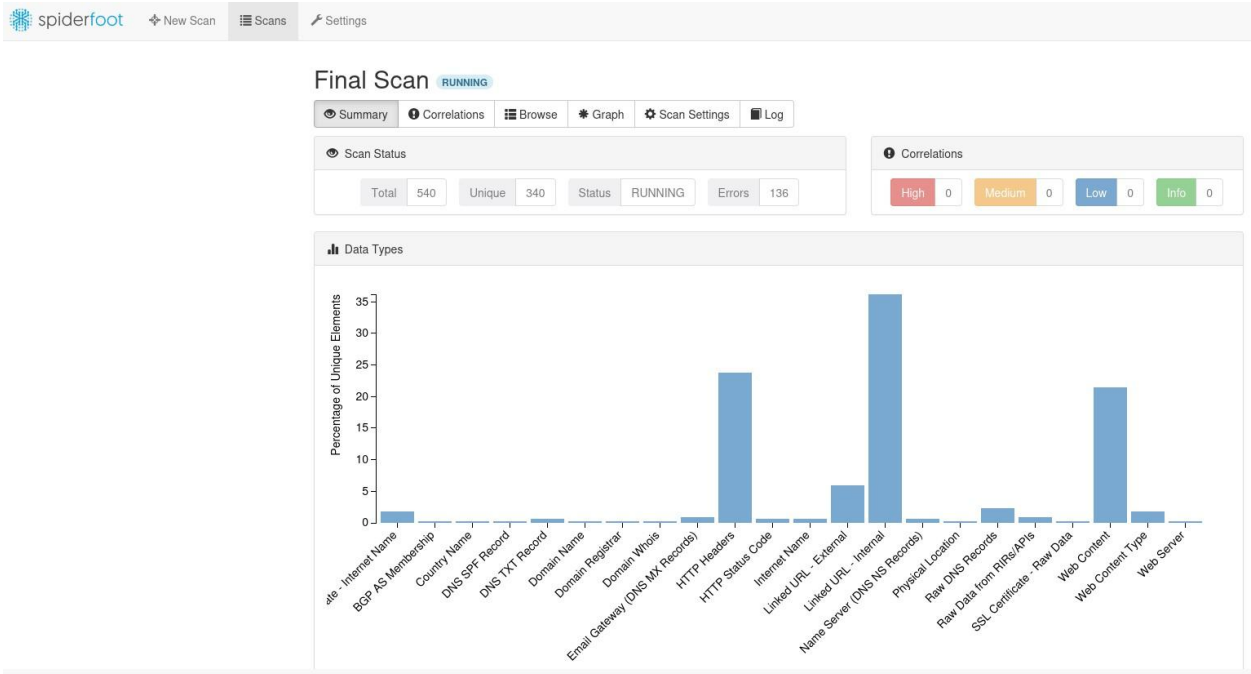
3.5 Web Application Firewall

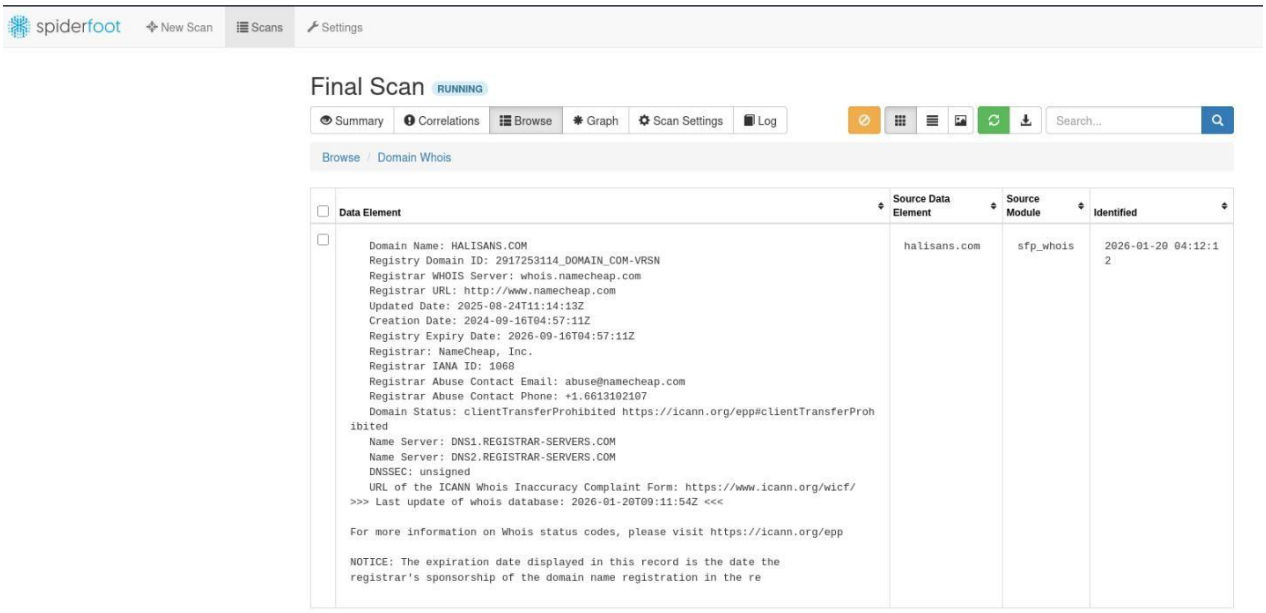
- **Passive approach:**



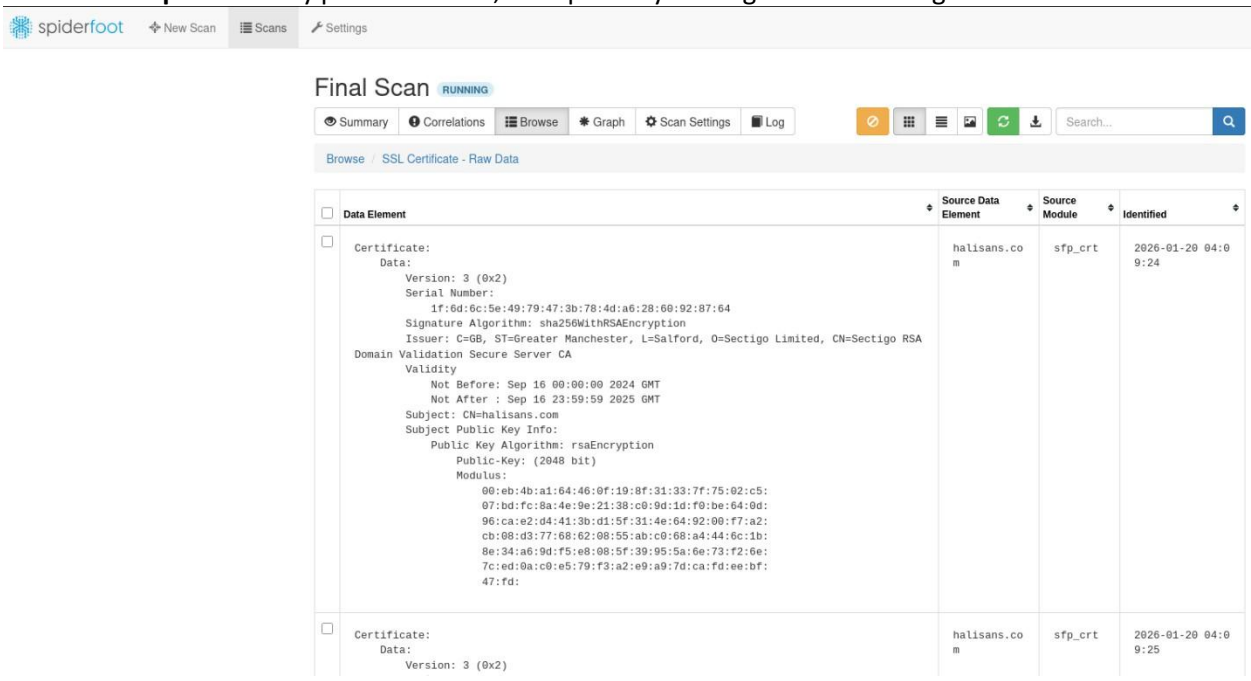
3.6 OSINT: Mentions, Accounts, and Exposure

- **SpiderFoot (passive modules):**
  - **DNS/Hosts:** Passive resolution of subdomains from CT/DNS.

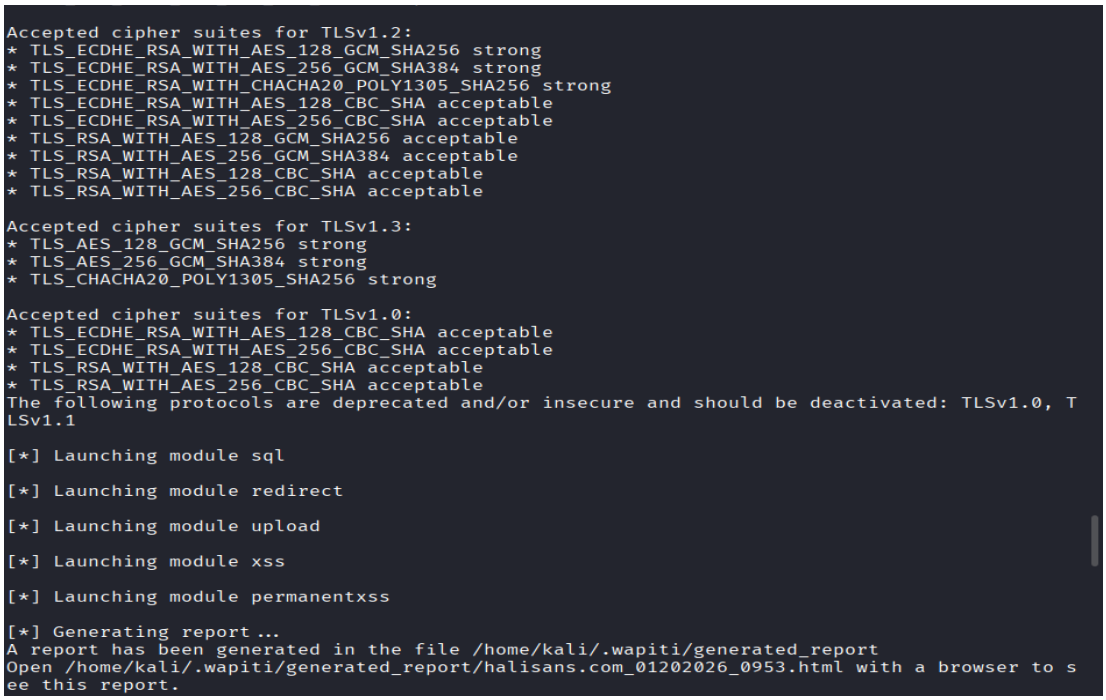




- **Credential exposure:** Only passive checks; no repository cloning or brute forcing.



Wapiti: Vulnerability Assessment on Web Application



Recommendations (Based on Passive Posture Only)

1. **HTTP Security Headers:** Verify presence of Strict-Transport-Security, Content-Security-Policy, X-Content-Type-Options, Referrer-Policy, and Permissions-Policy.

Wapiti vulnerability report

Target: https://halisans.com/

Date of the scan: Tue, 20 Jan 2026 09:53:32 +0000. Scope of the scan: folder. Crawled pages: 105

Summary

Category	Number of vulnerabilities found
Backup file	0
Cleartext Submission of Password	0
Weak credentials	0
CRLF Injection	0
Content Security Policy Configuration	1
Cross Site Request Forgery	0
Potentially dangerous file	0
Command execution	0
Path Traversal	0
Fingerprint web application framework	0
Fingerprint web server	0
Htaccess Bypass	0
HTML Injection	0
Clickjacking Protection	1
HTTP Strict Transport Security (HSTS)	1

Category	Number of vulnerabilities found
MIME Type Confusion	1
HttpOnly Flag cookie	0
Unencrypted Channels	0
Inconsistent Redirection	0
Information Disclosure - Full Path	1
LDAP Injection	0
Log4Shell	0
NS takeover	0
Open Redirect	0
Reflected Cross Site Scripting	0
Secure Flag cookie	0
Spring4Shell	0
SQL Injection	0
TLS/SSL misconfigurations	10
Server Side Request Forgery	0
Stored HTML Injection	0
Stored Cross Site Scripting	0
Subdomain takeover	0
Blind SQL Injection	0
Unrestricted File Upload	0

Category	Number of vulnerabilities found
Vulnerable software	0
Internal Server Error	0
Resource consumption	0
Review Webserver Metafiles for Information Leakage	0
Fingerprint web technology	0
HTTP Methods	0
TLS/SSL misconfigurations	8

Content Security Policy Configuration

Description

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.

🟡 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
CSP is not set for URL: https://halisans.com/
```

Solutions

Configuring Content Security Policy involves adding the Content-Security-Policy HTTP header to a web page and giving it values to control what resources the user agent is allowed to load for that page.

References

- Mozilla: Content Security Policy (CSP)
- OWASP: Content Security Policy Cheat Sheet
- OWASP: How to do Content Security Policy (PDF)

- OWASP: Content Security Policy

Clickjacking Protection

Description

Clickjacking is a technique that tricks a user into clicking something different from what the user perceives, potentially revealing confidential information or taking control of their computer.

🟡 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
X-Frame-Options is not set
```

Solutions

Implement X-Frame-Options or Content Security Policy (CSP) frame-ancestors directive.

References

- OWASP: Clickjacking
- KeyCDN: Preventing Clickjacking

HTTP Strict Transport Security (HSTS)

Description

HSTS is a web security policy mechanism that helps to protect websites against man-in-the-middle attacks such as protocol downgrade attacks and cookie hijacking.

🟡 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Strict-Transport-Security is not set
```

Solutions

Implement the HTTP Strict Transport Security header to enforce secure connections to the server.

References

- OWASP: HTTP Strict Transport Security
- KeyCDN: Enabling HSTS

MIME Type Confusion

Description

MIME type confusion can occur when a browser interprets files as a different type than intended, which could lead to security vulnerabilities like cross-site scripting (XSS).

🟡 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
X-Content-Type-Options is not set
```

Solutions

Implement X-Content-Type-Options to prevent MIME type sniffing.

References

- OWASP: MIME Sniffing
- KeyCDN: Preventing MIME Type Sniffing

Information Disclosure - Full Path

Description

The application response discloses full system paths. This information can help attackers understand the server environment, directory structure, and operating system, which can facilitate further attacks.

🟡 Vulnerability found in /steps-involved-in-performing-sql-injectionfor-

educational-purposes-only/

Description | HTTP Request | cURL command line | WSTG Code

```
Response contains potential system path: /usr/share/webshells/sql/
```

Solutions

Ensure that error messages and application responses do not disclose full filesystem paths or other sensitive system information. Use generic error messages for end users and log detailed errors only on the server side.

References

- OWASP: Information Leakage
- CWE-209: Generation of Error Message Containing Sensitive Information
- WASC-13: Information Leakage

TLS/SSL misconfigurations

Description

The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. Over the years numerous vulnerabilities have been discovered in some SSL/TLS version or specific ciphers making the integrity of the communications at risk (eavesdropping, alteration...)

🔥 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Requested hostname doesn't match those in the certificate
```

🟠 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Certificate doesn't use Extended Validation
```

🟠 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
OCSP Must-Staple extension is missing
```

🔴 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Certificate transparency: No
```

🔴 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Strict Transport Security (HSTS) is not set
```

🔥 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Server is vulnerable to OpenSSL CCS (CVE-2014-0224
```

🟠 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
The following ciphers are acceptable for TLSv1.1: TLS_ECDHE_RSA_WITH_AES_128_CBC_
```

🟠 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
The following ciphers are acceptable for TLSv1.2: TLS_ECDHE_RSA_WITH_AES_128_CBC_
```

🟠 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
The following ciphers are acceptable for TLSv1.0: TLS_ECDHE_RSA_WITH_AES_128_CBC_
```

🔥 Vulnerability found in /

Description | HTTP Request | cURL command line | WSTG Code

```
The following protocols are deprecated and/or insecure and should be deactivated:
```

Solutions

To protect against SSL/TLS vulnerabilities, make sure that deprecated versions of the protocol are disabled. Refer to up-to-date recommendations to only allow modern versions of TLS with Perfect Forward Secrecy.

References

- SSL Labs: SSL and TLS Deployment Best Practices
- Mozilla: Server Side TLS recommended configurations
- Beagle Security: Importance of TLS 1.3, SSL and TLS Vulnerabilities
- Security of TLS cipher suites
- Trail of Bits: What Application Developers Need To Know About TLS Early Data (0RTT)
- OWASP: Weak SSL TLS Ciphers Insufficient Transport Layer Protection

---

TLS/SSL misconfigurations

Description

The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications. Over the years numerous vulnerabilities have been discovered in some SSL/TLS version or specific ciphers making the integrity of the communications at risk (eavesdropping, alteration...)

🕵 Additional found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Certificate subject: *.web-hosting.com
```

🕵 Additional found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Alt. names: *.web-hosting.com, web-hosting.com
```

🕵 Additional found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Issuer: Sectigo RSA Domain Validation Secure Server CA
```

🕵 Additional found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Key: RSA 2048 bits
```

🕵 Additional found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Signature Algorithm: sha256WithRSAEncryption
```

🕵 Additional found in /

Description | HTTP Request | cURL command line | WSTG Code

```
Certificate expires in 2 months, 13 days, 21 hours, 35 minutes and 59.22 seconds
```

🕵 Additional found in /

Description | HTTP Request | cURL command line | WSTG Code

```
The following ciphers are strong for TLSv1.2: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA2
```

🕵 Additional found in /

Description | HTTP Request | cURL command line | WSTG Code

```
The following ciphers are strong for TLSv1.3: TLS_AES_128_GCM_SHA256, TLS_AES_256
```

Solutions

To protect against SSL/TLS vulnerabilities, make sure that deprecated versions of the protocol are disabled. Refer to up-to-date recommendations to only allow modern versions of TLS with Perfect Forward Secrecy.

References

- SSL Labs: SSL and TLS Deployment Best Practices
- Mozilla: Server Side TLS recommended configurations
- Beagle Security: Importance of TLS 1.3, SSL and TLS Vulnerabilities
- Security of TLS cipher suites
- Trail of Bits: What Application Developers Need To Know About TLS Early Data (0RTT)
- OWASP: Weak SSL TLS Ciphers Insufficient Transport Layer Protection

---

Conclusion

This assessment combined passive OSINT reconnaissance with a non-intrusive Wapiti scan to evaluate the external security posture of halisans.com within a clearly defined scope. The findings show that while the website is publicly accessible, functional, and does not expose critical application-level vulnerabilities such as SQL injection, XSS, or authentication weaknesses, there are notable security hygiene gaps at the configuration and transport layers.

The most significant issues relate to missing HTTP security headers (CSP, HSTS, X-Frame-Options, X-Content Type Options), instances of information disclosure, and multiple TLS/SSL misconfigurations, including deprecated protocol support and certificate inconsistencies. These weaknesses do not indicate active exploitation but increase the attack surface and could be leveraged in chained or future attacks if left unaddressed.

Overall, the security posture can be classified as moderate but improvable. By implementing recommended header hardening, tightening TLS configurations to modern standards, and reducing informational leakage, the site’s resilience against common web-based threats can be significantly

strengthened. This work demonstrates the value of passive reconnaissance as an effective first step in identifying security risks without disrupting production systems, while highlighting areas that would benefit from deeper, authorized active testing in the future.