Wapiti vulnerability report

Target: http://localhost/dvwa/

Date of the scan: Wed, 11 Jun 2025 06:45:13 +0000. Scope of the scan: folder. Crawled pages: 3

Summary

Category	Number of vulnerabilities found
Backup file	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)	0
MIME Type Confusion	1
HttpOnly Flag cookie	0
Unencrypted Channels	0
LDAP Injection	0

Category	Number of vulnerabilities found
Log4Shell	0
Open Redirect	0
Reflected Cross Site Scripting	0
Secure Flag cookie	2
Spring4Shell	0
SQL Injection	0
TLS/SSL misconfigurations	0
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
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	0

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 /dvwa/

Description | HTTP Request | cURL command line | WSTG Code

```
CSP is not set
```

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 /dvwa/

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

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 /dvwa/

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

Secure Flag cookie

Description
The secure flag is an option that can be set by the application server when sending a new cookie to the user within an HTTP Response. The purpose of the secure flag is to prevent cookies from being observed by unauthorized parties due to the transmission of a the cookie in clear text.

🟡 Vulnerability found in /dvwa/

Description | HTTP Request | cURL command line | WSTG Code

```
Secure flag is not set in the cookie : security
```

Vulnerability found in /dvwa/

Description | HTTP Request | cURL command line | WSTG Code

```
Secure flag is not set in the cookie : PHPSESSID
```

Solutions

When generating the cookie, make sure to set the Secure Flag to True.

References

- OWASP: Testing for Cookies Attributes
- OWASP: Secure Cookie Attribute

Wapiti 3.2.4 © Nicolas SURRIBAS 2006-2025