VULNERABILITY ASSESSMENT REPORT

Targated Website:
http://demo.testfire.net

Prepared by

Affan Mongal

Cybersecurity Analyst Intern - Future Interns

Date: 2nd February 2026

Executive Summary

This report presents the results of a vulnerability assessment conducted on the publicly accessible website demo.testfire.net.

The objective of this assessment was to identify common security misconfigurations and potential risks using ethical and passive analysis techniques.

No exploitation or intrusive testing was performed during the assessment. The findings are focused on improving the overall security posture of the website and reducing potential risks to users and business operations.

Scope & Ethics

Scope of Assessment:

- Public-facing pages only
- Passive scanning and configuration analysis
- Security header and cookie inspection

Out of Scope:

- Authentication bypass
- Brute-force attacks
- Denial-of-Service attacks
- Exploitation of vulnerabilities

This assessment strictly followed ethical security testing guidelines.

Tools Used

Tools & Techniques:

- Nmap – Basic exposure and port analysis
- OWASP ZAP (Zed Attack Proxy) – Passive vulnerability scanning
- Browser Developer Tools – Header and cookie inspection
- Canva – Professional security report design

Risk Classification

Risk Level	Description
Low	Minimal impact, informational issues
Mideum	Security misconfigurations requiring attention
High	Critical vulnerabilities (not identified in this assessment)

Findings Summary Table

Sr. No.	Issue	Risk
1	Cookie without SameSite attribute	Low
2	Content security policy (CSP) header not set	Medium
3	Missing anti-clickjacking header	Medium
4	Server leaks version information via “Server” HTTP response header field	Low
5	X-Content-Type-Options Header Missing	Low

Detailed Findings

Finding 1: Cookie Without SameSite Attribute

Risk Level: Low
Confidence: Medium
CWE ID: CWE-1275

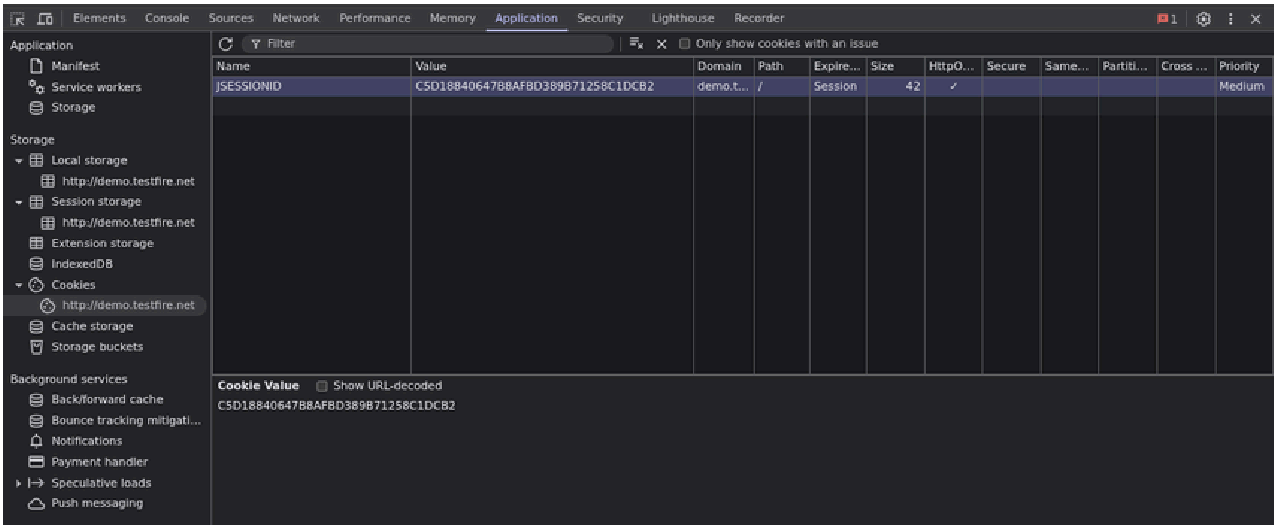
Description:
The session cookie (JSESSIONID) is set without the SameSite attribute. This attribute restricts how cookies are sent with cross-site requests and helps protect against CSRF attacks.

Evidence:

- URL Tested: http://demo.testfire.net/
- Cookie Name: JSESSIONID
- Set-Cookie Header Observed without SameSite
- Browser Developer Tools confirm missing SameSite flag

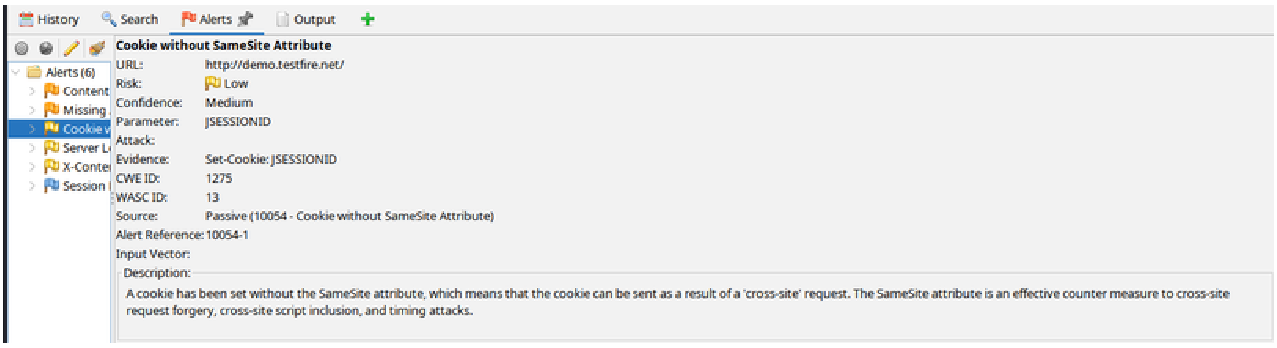
Impact”
Cookies may be sent in cross-site requests, increasing the risk of Cross-Site Request Forgery (CSRF) and session-related attacks.

Recommendation:
Set the SameSite attribute for session cookies:
“Set-Cookie: JSESSIONID=VALUE; SameSite=Strict; Secure; HttpOnly”



Name	Value	Domain	Path	Expire...	Size	HttpO...	Secure	Same...	Partiti...	Cross ...	Priority
JSESSIONID	C5D18840647B8AFBD389B71258C1DCB2	demo.t...	/	Session	42	✓					Medium

Cookie Value □ Show URL-decoded
C5D18840647B8AFBD389B71258C1DCB2



Cookie without SameSite Attribute

URL: http://demo.testfire.net/
Risk: Low
Confidence: Medium
Parameter: JSESSIONID
Attack:
Evidence: Set-Cookie: JSESSIONID
CWE ID: 1275
WASC ID: 13
Source: Passive (10054 - Cookie without SameSite Attribute)
Alert Reference: 10054-1
Input Vector:
Description:
A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Detailed Findings

Finding 2: Content Security Policy (CSP) Header Not Set

Risk Level: Medium
Confidence: High
CWE ID: CWE-693

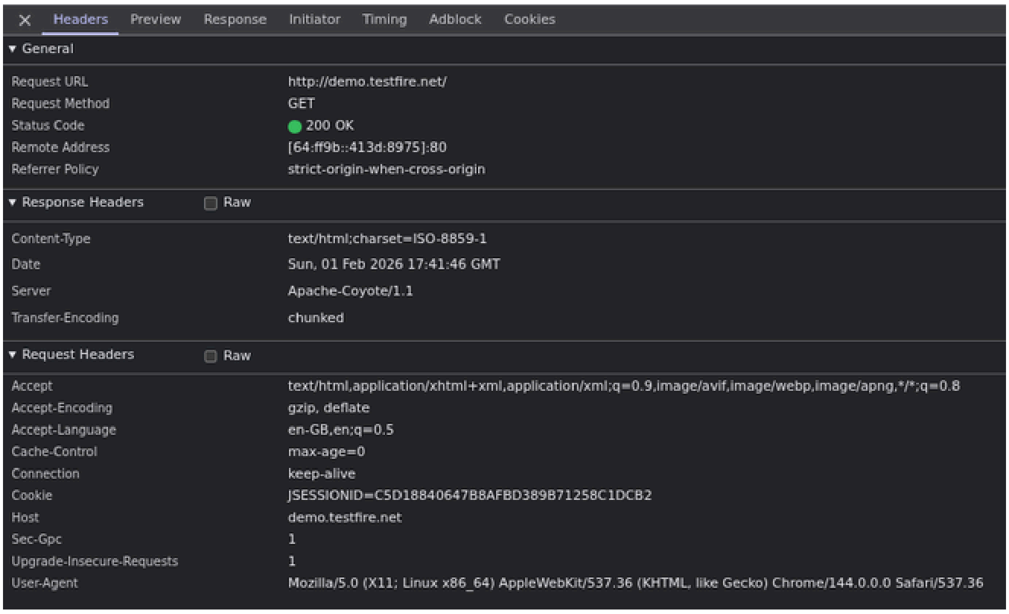
Description:
The application does not implement a Content Security Policy (CSP). CSP helps mitigate attacks such as Cross-Site Scripting (XSS), data injection, and malicious content loading.
Without CSP, the browser has no restrictions on the sources from which scripts, styles, or other resources can be loaded.

Evidence:

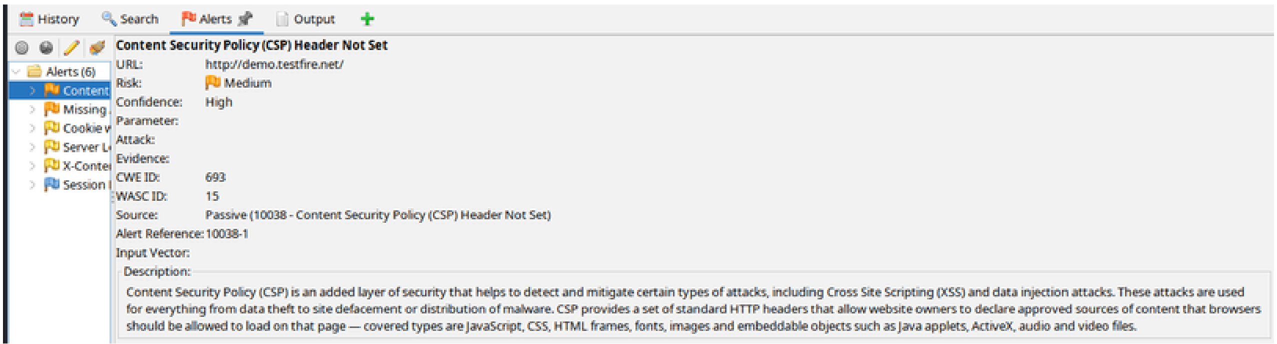
- URL Tested: http://demo.testfire.net/
- OWASP ZAP Alert: Content Security Policy (CSP) Header Not Set

Impact:
Increases the risk of client-side attacks such as XSS, which may lead to session hijacking or data theft.

Recommendation:
Implement a strict CSP policy. Example:
“Content-Security-Policy: default-src 'self';”
This can be adjusted based on application requirements.



```
▾ General

Request URL                 http://demo.testfire.net/
Request Method              GET
Status Code                 ● 200 OK
Remote Address              [64:ff9b::413d:8975]:80
Referrer Policy             strict-origin-when-cross-origin

▾ Response Headers          □ Raw

Content-Type                text/html;charset=ISO-8859-1
Date                        Sun, 01 Feb 2026 17:41:46 GMT
Server                      Apache-Coyote/1.1
Transfer-Encoding           chunked

▾ Request Headers           □ Raw

Accept                      text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Accept-Encoding             gzip, deflate
Accept-Language             en-GB,en;q=0.5
Cache-Control               max-age=0
Connection                  keep-alive
Cookie                      JSESSIONID=C5D18840647B8AFBD389B71258C1DCB2
Host                        demo.testfire.net
Sec-Gpc                     1
Upgrade-Insecure-Requests   1
User-Agent                  Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36
```



Content Security Policy (CSP) Header Not Set

```
URL:             http://demo.testfire.net/
Risk:            Medium
Confidence:      High
Parameter:
Attack:
Evidence:
CWE ID:          693
WASC ID:         15
Source:          Passive (10038 - Content Security Policy (CSP) Header Not Set)
Alert Reference: 10038-1
Input Vector:
```

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. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

Detailed Findings

Finding 3: Missing Anti-Clickjacking Header

Risk Level: Medium
Confidence: Medium
CWE ID: CWE-1021

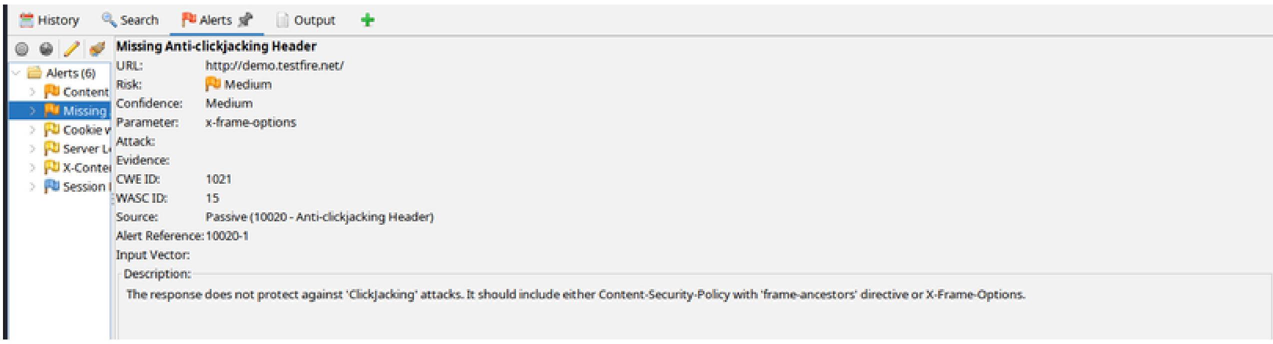
Description:
The application does not include an anti-clickjacking protection mechanism such as the X-Frame-Options header or a proper Content-Security-Policy (CSP) frame-ancestors directive. This makes the application vulnerable to clickjacking attacks, where a malicious site tricks users into interacting with hidden elements.

Evidence:

- URL Tested: http://demo.testfire.net/
- Missing Header: X-Frame-Options
- OWASP ZAP Alert: Missing Anti-clickjacking Header

Impact:
Attackers may embed the website within an invisible iframe to trick users into performing unintended actions, potentially leading to account compromise.

Recommendation:
Add one of the following headers:
“X-Frame-Options: DENY”
OR
“Content-Security-Policy: frame-ancestors 'none';”



Missing Anti-clickjacking Header

```
URL:             http://demo.testfire.net/
Risk:            Medium
Confidence:      Medium
Parameter:       x-frame-options
Attack:
Evidence:
CWE ID:          1021
WASC ID:         15
Source:          Passive (10020 - Anti-clickjacking Header)
Alert Reference: 10020-1
Input Vector:
```

Description:
The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.

Detailed Findings

Finding 4: Server Leaks Version Information via "Server" HTTP Response Header

Risk Level: Low
Confidence: High
CWE ID: CWE-497

Description:
The application discloses server version information in the HTTP Server response header. In this case, the server identifies itself as Apache-Coyote/1.1.
Exposing server version details can help attackers tailor attacks based on known vulnerabilities of the disclosed server software.

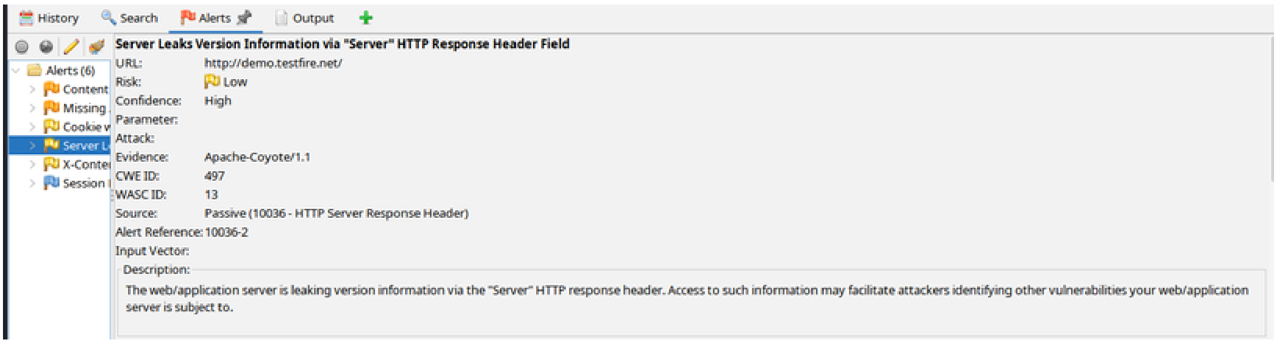
Evidence:

- URL Tested: http://demo.testfire.net/
- Response Header: Server: Apache-Coyote/1.1
- OWASP ZAP Alert: Server Leaks Version Information

Impact:
Information disclosure increases the risk of targeted attacks, such as exploiting known vulnerabilities specific to the server version.

Recommendation:
Configure the web server to suppress or generalize the Server header. For example:

- Remove version numbers
- Use a generic server name



Server Leaks Version Information via "Server" HTTP Response Header Field

URL:	http://demo.testfire.net/
Risk:	Low
Confidence:	High
Parameter:	
Attack:	
Evidence:	Apache-Coyote/1.1
CWE ID:	497
WASC ID:	13
Source:	Passive (10036 - HTTP Server Response Header)
Alert Reference:	10036-2
Input Vector:	

Description:
The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Detailed Findings

Finding 5: X-Content-Type-Options Header Missing

Risk Level: Low
Confidence: Medium
CWE ID: CWE-693
OWASP WASC ID: 15

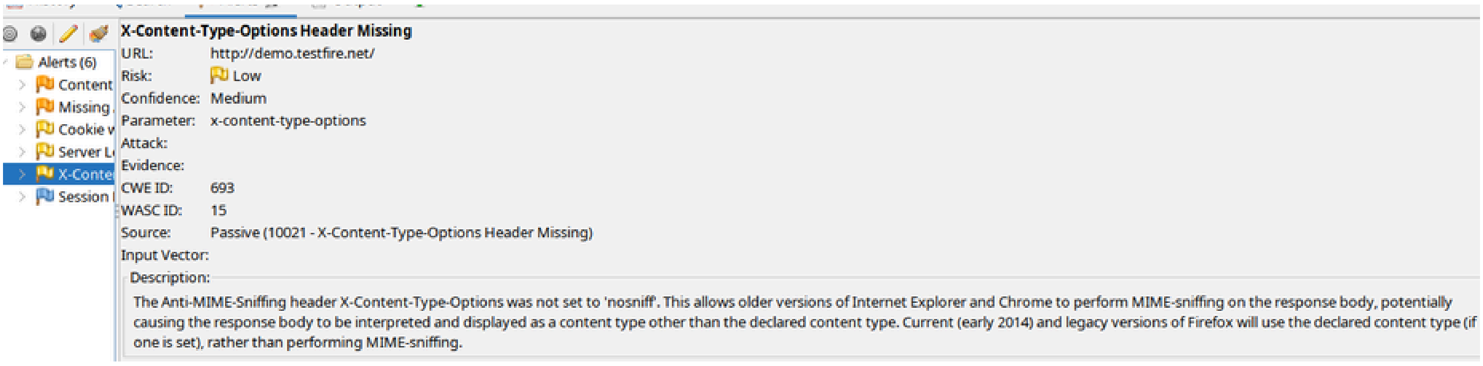
Description:
The HTTP response from the application does not include the X-Content-Type-Options security header. This header is used to prevent MIME-type sniffing by browsers. Without this header, browsers may interpret files as a different content type than intended.
This behavior may allow attackers to inject malicious scripts that could be executed by the browser if content is incorrectly interpreted.

Evidence:

- URL Tested: http://demo.testfire.net/
- OWASP ZAP Alert: X-Content-Type-Options Header Missing
- Parameter: x-content-type-options

Impact:
An attacker may exploit MIME-sniffing behavior to execute malicious content in the victim's browser, potentially leading to Cross-Site Scripting (XSS) attacks.

Recommendation:
Configure the web server to include the following HTTP response header: "X-Content-Type-Options: nosniff"
This ensures browsers strictly follow the declared Content-Type.



X-Content-Type-Options Header Missing

URL: http://demo.testfire.net/
Risk: Low
Confidence: Medium
Parameter: x-content-type-options
Attack:
Evidence:
CWE ID: 693
WASC ID: 15
Source: Passive (10021 - X-Content-Type-Options Header Missing)
Input Vector:

Description:
The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Conclusion

The vulnerability assessment identified several medium and low-risk security issues primarily related to configuration weaknesses.

Addressing these issues will improve the website’s security posture, reduce potential attack surfaces, and enhance user trust.

Regular security reviews and secure configuration practices are recommended.