

Website Vulnerability Scanner Report

✓ <http://testfire.net/login.jsp>

! The Light Website Scanner didn't check for critical issues like SQLi, XSS, Command Injection, XXE, etc. [Upgrade to run Deep scans](#) with 40+ tests and detect more vulnerabilities.

Summary

Overall risk level:

Medium

Risk ratings:

Critical: 0

High: 0

Medium: 3

Low: 5

Info: 44

Scan information:

Start time: Jun 07, 2025 / 13:15:32
UTC+0530

Finish time: Jun 07, 2025 / 13:32:12
UTC+0530

Scan duration: 16 min, 40 sec

Tests performed: 52/52

Scan status: **Finished**

Findings

Insecure cookie setting: missing Secure flag

port 80/tcp

CONFIRMED

URL	Cookie Name	Evidence
http://testfire.net/login.jsp	JSESSIONID	Set-Cookie: JSESSIONID=47698CCDCD2AED051CF1A33CDD245D3D Request / Response

▼ Details

Risk description:

The risk exists that an attacker will intercept the clear-text communication between the browser and the server and he will steal the cookie of the user. If this is a session cookie, the attacker could gain unauthorized access to the victim's web session.

Recommendation:

Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.

References:

https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html

Classification:

CWE : [CWE-614](#)

OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)

OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)

Communication is not secure

port 80/tcp

CONFIRMED

URL	Response URL	Evidence
http://testfire.net/login.jsp	http://testfire.net/login.jsp	Communication is made over unsecure, unencrypted HTTP.

▼ Details

Risk description:

The risk is that an attacker who manages to intercept the communication at the network level can read and modify the data transmitted (including passwords, secret tokens, credit card information and other sensitive data).

Recommendation:

We recommend you to reconfigure the web server to use HTTPS - which encrypts the communication between the web browser and the server.

Classification:

CWE : [CWE-311](#)

OWASP Top 10 - 2017 : [A3 - Sensitive Data Exposure](#)

OWASP Top 10 - 2021 : [A4 - Insecure Design](#)



Passwords are submitted unencrypted over the network

CONFIRMED

port 80/tcp

URL	Evidence
http://testfire.net/login.jsp	<p>Password input detected over insecure HTTP. Login form:</p> <pre><form action="doLogin" id="login" method="post" name="login" onsubmit="return (confirminput(login));"> <table> <tr> <td> Username: </td> <td> <input id="uid" name="uid" style="width: 150px;" type="text" value="" /> </td> <td> </td> </tr> <tr> <td> Password: </td> <td> <input id="passw" name="passw" style="width: 150px;" type="password" /> </td> <td> </td> </tr> <tr> <td></td> <td> <input name="btnSubmit" type="submit" value="Login" /> </td> </tr> </table> </form></pre> <p>Request / Response</p>

Details**Risk description:**

The risk is that malicious actors could employ various techniques, such as packet sniffing or man-in-the-middle attacks, to capture plaintext passwords. Once intercepted, the attacker gains unauthorized access to user accounts, potentially leading to identity theft, unauthorized data access, or other malicious activities. The risk remains unchanged even if the password's form submission triggers a redirect response to an HTTPS page.

Recommendation:

We recommend you to reconfigure the web server so it uses HTTPS - which encrypts the communication between the web browser and the server. This way, the attacker will not be able to obtain the clear-text passwords, even though he manages to intercept the network communication.

Classification:

CWE : [CWE-523](#)

OWASP Top 10 - 2017 : [A3 - Sensitive Data Exposure](#)

OWASP Top 10 - 2021 : [A2 - Cryptographic Failures](#)

Screenshot:

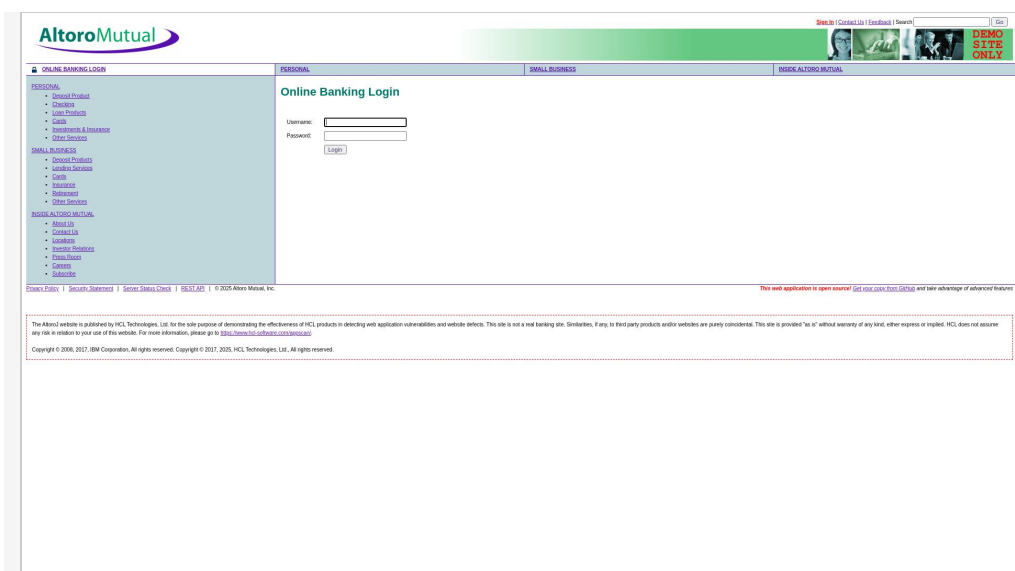


Figure 1. Password field found

Missing security header: Content-Security-Policy port 80/tcp

CONFIRMED

URL	Evidence
http://testfire.net/login.jsp	Response does not include the HTTP Content-Security-Policy security header or meta tag Request / Response

Details

Risk description:

The risk is that if the target application is vulnerable to XSS, lack of this header makes it easily exploitable by attackers.

Recommendation:

Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

References:

https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy>

Classification:

CWE : [CWE-693](#)
OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)
OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)

Missing security header: X-Content-Type-Options port 80/tcp

CONFIRMED

URL	Evidence
http://testfire.net/login.jsp	Response headers do not include the X-Content-Type-Options HTTP security header Request / Response

Details

Risk description:

The risk is that lack of this header could make possible attacks such as Cross-Site Scripting or phishing in Internet Explorer browsers.

Recommendation:

We recommend setting the X-Content-Type-Options header such as `X-Content-Type-Options: nosniff`.

References:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options>

Classification:

CWE : [CWE-693](#)

Missing security header: Referrer-Policy

CONFIRMED

port 80/tcp

URL	Evidence
http://testfire.net/login.jsp	Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with name 'referrer' is not present in the response. Request / Response

Details

Risk description:

The risk is that if a user visits a web page (e.g. "http://example.com/pricing/") and clicks on a link from that page going to e.g. "https://www.google.com", the browser will send to Google the full originating URL in the **Referer** header, assuming the Referrer-Policy header is not set. The originating URL could be considered sensitive information and it could be used for user tracking.

Recommendation:

The Referrer-Policy header should be configured on the server side to avoid user tracking and inadvertent information leakage. The value **no-referrer** of this header instructs the browser to omit the Referer header entirely.

References:

https://developer.mozilla.org/en-US/docs/Web/Security/Referer_header:_privacy_and_security_concerns

Classification:

CWE : [CWE-693](#)


OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)

OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)

DOM-based Open Redirect

CONFIRMED

port 80/tcp

URL	Method	Vulnerable Parameter	Evidence	Replay Attack
http://testfire.net/disclaimer.htm	GET	url (Query Parameter)	<p>The server redirects to the URL https://pentest-tools.com/file.txt when it is injected in the url query parameter.</p> <p>The redirection was triggered in the DOM by the JavaScript sink location.href which received our URL.</p> <p>The stack trace to this call was:</p> <pre>anonymous @ anonymo us: line 380, column 2 anonymous @ anonymo us: line 379, column 22 callFunction @ anonymo us: line 365, column 21 apply.element-6066-11e4-a52e-4f735466cecf @ anonymo us: line 402, column 23 executeScript @ anonymo us: line 397, column 29 eval @ anonymo us: line 2, column 37 onclick @ http://testfire.net/disclaimer.htm?url=https://pentest-tools.com/file.txt: line 56, column 49 go @ http://testfire.net/disclaimer.htm?url=https://pentest-tools.com/file.txt: line 18, column 30</pre> <p>Request / Response</p>	

Details

Risk description:

The risk is that attackers may use open redirect to redirect users to arbitrary domains of their choice. This can be used in phishing attacks, as targets will receive a trusted URL and might not notice the subsequent redirect, often placed in the query or fragment parameters.

Recommendation:

Avoid incorporating user input into JavaScript variables that can end up as arguments for request-triggering functions. Instead, use direct predefined links to redirect towards the target page. If, however, this is not possible, you should only accept relative URLs as input. To

check that the input represents a relative URL, you can make use of the `URL` class from JavaScript by instantiating it with the user input and the trusted base. Then ensure the resulting URL's origin still matches your domain before accepting it.

Classification:

CWE : [CWE-601](#)

OWASP Top 10 - 2021 : [A1 - Broken Access Control](#)

Server software and technology found

port 80/tcp

UNCONFIRMED ⓘ

Software / Version	Category
 Java	Programming languages
 Apache Tomcat	Web servers
 JSP	Web frameworks

Details

Risk description:

The risk is that an attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:

We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

References:

https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html

Classification:

OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)

OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)

Screenshot:

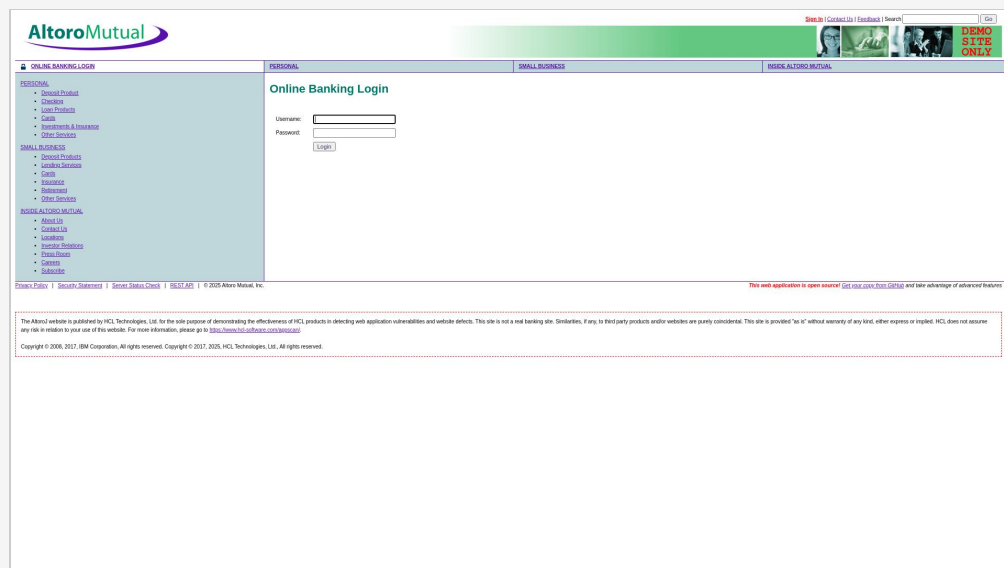


Figure 2. Website Screenshot

Login Interface Found

port 80/tcp

CONFIRMED

URL	Evidence
-----	----------

http://testfire.net/login.jsp	<pre><input id="uid" name="uid" style="width: 150px;" type="text" value=""/> <input id="passw" name="passw" style="width: 150px;" type="password"/> <input name="btnSubmit" type="submit" value="Login"/></pre> <p>Request / Response</p>
---	---

▼ Details

Risk description:

The risk is that an attacker could use this interface to mount brute force attacks against known passwords and usernames combinations leaked throughout the web.

Recommendation:

Ensure each interface is not bypassable using common knowledge of the application or leaked credentials using occasional password audits.

References:

<https://pentest-tools.com/network-vulnerability-scanning/password-auditor>
<http://capec.mitre.org/data/definitions/16.html>

Screenshot:

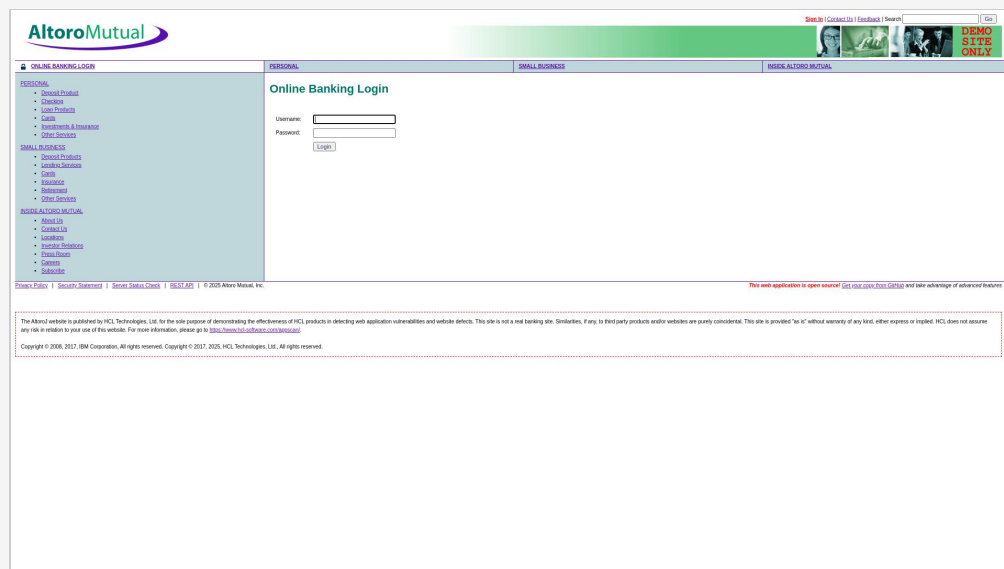


Figure 3. Login Interface

🚩 Security.txt file is missing port 80/tcp

CONFIRMED

URL
Missing: http://testfire.net/well-known/security.txt

▼ Details

Risk description:

There is no particular risk in not having a security.txt file for your server. However, this file is important because it offers a designated channel for reporting vulnerabilities and security issues.

Recommendation:

We recommend you to implement the security.txt file according to the standard, in order to allow researchers or users report any security issues they find, improving the defensive mechanisms of your server.

References:

<https://securitytxt.org/>

Classification:

OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)
OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)

Spider results

URL	Method	Parameters	Page Title	Page Size	Status Code
http://testfire.net/doLogin	GET		Altoro Mutual	9.18 KB	200
http://testfire.net/doSubscribe	POST	Body: btnSubmit=Subscribe txtEmail=1d3d2d231d2d4	Altoro Mutual	9.18 KB	200
http://testfire.net/'disclaimer.htm	GET	Query: url=http://www.netscape.com'	Altoro Mutual	6.79 KB	404
http://testfire.net/Privacypolicy.jsp	GET	Query: sec=Careers template=US	Altoro Mutual	6.79 KB	404
http://testfire.net/cgi.exe	GET		Altoro Mutual	6.79 KB	404
http://testfire.net/default.jsp	GET	Query: content=security.htm	Altoro Mutual	6.79 KB	404
http://testfire.net/disclaimer.htm	GET	Query: url=http://www.microsoft.com	Altoro Mutual: Link Disclaimer	2.03 KB	200
http://testfire.net/doLogin	POST	Body: btnSubmit>Login passw=Secure123456\$	Altoro Mutual	8.46 KB	200
http://testfire.net/doSubscribe	GET		HTTP Status 405 – Method Not Allowed	1.05 KB	405
http://testfire.net/feedback.jsp	GET		Altoro Mutual	8.33 KB	200
http://testfire.net/html	GET		Altoro Mutual	6.79 KB	404
http://testfire.net/images	GET		Altoro Mutual	6.79 KB	404
http://testfire.net/images/	GET		Altoro Mutual	6.79 KB	404
http://testfire.net/index.jsp	GET		Altoro Mutual	9.18 KB	200
http://testfire.net/index.jsp	GET	Query: content=inside_investor.htm	Altoro Mutual	8.13 KB	200
http://testfire.net/inside_points_of_interest.htm	GET		Altoro Mutual	6.79 KB	404
http://testfire.net/login.jsp	GET		Altoro Mutual	8.35 KB	200
http://testfire.net/retirement.htm	GET		Business Retirement Infromation	1.09 KB	200
http://testfire.net/search.jsp	GET		Altoro Mutual	6.84 KB	200
http://testfire.net/search.jsp	GET	Query: query=1d3d2d231d2dd4	Altoro Mutual	6.85 KB	200
http://testfire.net/sendFeedback	GET		HTTP Status 405 – Method Not Allowed	1.05 KB	405
http://testfire.net/sendFeedback	POST	Body: cfile=comments.txt comments=comments email_addr=1d3d2d231d2dd4 name=1d3d2d231d2dd4 reset= Clear Form subject=subject submit= Submit	Altoro Mutual	7.05 KB	200
http://testfire.net/status_check.jsp	GET		Altoro Mutual	9.87 KB	200

http://testfire.net/subscribe.jsp	GET		Altoro Mutual	8.34 KB	200
http://testfire.net/subscribe.swf	GET			2.31 KB	200
http://testfire.net/survey_questions.jsp	GET		Altoro Mutual	6.98 KB	200
http://testfire.net/swagger	GET		Altoro Mutual	6.79 KB	404
http://testfire.net/swagger/	GET		Altoro Mutual	6.79 KB	404
http://testfire.net/swagger/index.html	GET		Swagger UI	1.45 KB	200

▼ Details

Risk description:

The table contains all the unique pages the scanner found. The duplicated URLs are not available here as scanning those is considered unnecessary

Recommendation:

We recommend to advanced users to make sure the scan properly detected most of the URLs in the application.

References:

[All the URLs the scanner found, including duplicates](#) (available for 90 days after the scan date)

🚩 Website is accessible.

🚩 Nothing was found for vulnerabilities of server-side software.

🚩 Nothing was found for client access policies.

🚩 Nothing was found for robots.txt file.

🚩 Nothing was found for outdated JavaScript libraries.

🚩 Nothing was found for use of untrusted certificates.

🚩 Nothing was found for enabled HTTP debug methods.

🚩 Nothing was found for administration consoles.

🚩 Nothing was found for information disclosure.

🚩 Nothing was found for software identification.

🚩 Nothing was found for sensitive files.

🚩 Nothing was found for interesting files.

🚩 Nothing was found for enabled HTTP OPTIONS method.

🚩 Nothing was found for directory listing.

🚩 Nothing was found for error messages.

🚩 Nothing was found for debug messages.

🚩 Nothing was found for code comments.

🚩 Nothing was found for missing HTTP header - Strict-Transport-Security.

🚩 Nothing was found for missing HTTP header - Feature.

🚩 Nothing was found for Insecure Direct Object Reference.

🚩 Nothing was found for passwords submitted in URLs.

🚩 Nothing was found for domain too loose set for cookies.

🚩 Nothing was found for mixed content between HTTP and HTTPS.

🚩 Nothing was found for cross domain file inclusion.

🚩 Nothing was found for internal error code.

🚩 Nothing was found for HttpOnly flag of cookie.

🚩 Nothing was found for secure password submission.

🚩 Nothing was found for sensitive data.

🚩 Nothing was found for Server Side Request Forgery.

🚩 Nothing was found for Open Redirect.

🚩 Nothing was found for Exposed Backup Files.

🚩 Nothing was found for unsafe HTTP header Content Security Policy.

🚩 Nothing was found for OpenAPI files.

🚩 Nothing was found for file upload.

🚩 Nothing was found for SQL statement in request parameter.

🚩 Nothing was found for password returned in later response.

🚩 Nothing was found for Path Disclosure.

🚩 Nothing was found for Session Token in URL.

🚩 Nothing was found for API endpoints.

🚩 Nothing was found for emails.

🚩 Nothing was found for missing HTTP header - Rate Limit.

Scan coverage information

List of tests performed (52/52)

- ✓ Starting the scan...
- ✓ Checking for missing HTTP header - Content Security Policy...
- ✓ Checking for Secure flag of cookie...
- ✓ Checking for missing HTTP header - X-Content-Type-Options...
- ✓ Checking for secure communication...
- ✓ Checking for passwords submitted unencrypted...
- ✓ Checking for missing HTTP header - Referrer...
- ✓ Checking for login interfaces...
- ✓ Spidering target...
- ✓ Checking for website technologies...
- ✓ Checking for vulnerabilities of server-side software...
- ✓ Checking for client access policies...
- ✓ Checking for robots.txt file...
- ✓ Checking for absence of the security.txt file...
- ✓ Checking for outdated JavaScript libraries...
- ✓ Checking for use of untrusted certificates...
- ✓ Checking for enabled HTTP debug methods...
- ✓ Checking for administration consoles...
- ✓ Checking for information disclosure... (this might take a few hours)
- ✓ Checking for software identification...
- ✓ Checking for sensitive files...
- ✓ Checking for interesting files... (this might take a few hours)
- ✓ Checking for DOM-based Open Redirect...
- ✓ Checking for enabled HTTP OPTIONS method...
- ✓ Checking for directory listing...
- ✓ Checking for error messages...
- ✓ Checking for debug messages...
- ✓ Checking for code comments...
- ✓ Checking for missing HTTP header - Strict-Transport-Security...
- ✓ Checking for missing HTTP header - Feature...
- ✓ Checking for Insecure Direct Object Reference...
- ✓ Checking for passwords submitted in URLs...
- ✓ Checking for domain too loose set for cookies...
- ✓ Checking for mixed content between HTTP and HTTPS...

- ✓ Checking for cross domain file inclusion...
- ✓ Checking for internal error code...
- ✓ Checking for HttpOnly flag of cookie...
- ✓ Checking for secure password submission...
- ✓ Checking for sensitive data...
- ✓ Checking for Server Side Request Forgery...
- ✓ Checking for Open Redirect...
- ✓ Checking for Exposed Backup Files...
- ✓ Checking for unsafe HTTP header Content Security Policy...
- ✓ Checking for OpenAPI files...
- ✓ Checking for file upload...
- ✓ Checking for SQL statement in request parameter...
- ✓ Checking for password returned in later response...
- ✓ Checking for Path Disclosure...
- ✓ Checking for Session Token in URL...
- ✓ Checking for API endpoints...
- ✓ Checking for emails...
- ✓ Checking for missing HTTP header - Rate Limit...

Scan parameters

target: http://testfire.net/login.jsp
scan_type: Light
authentication: False

Scan stats

Unique Injection Points Detected:	27
URLs spidered:	85
Total number of HTTP requests:	950
Average time until a response was received:	168ms
Total number of HTTP request errors:	231
