

Website Vulnerability Scanner Report (Light)



Get a PRO Account to unlock the FULL capabilities of this scanner



See what the FULL scanner can do

Perform in-depth website scanning and discover high risk vulnerabilities.

Testing areas	Light scan	Full scan
Website fingerprinting	✓	✓
Version-based vulnerability detection	✓	✓
Common configuration issues	✓	✓
SQL injection	✗	✓
Cross-Site Scripting	✗	✓
Local/Remote File Inclusion	✗	✓
Remote command execution	✗	✓
Discovery of sensitive files	✗	✓

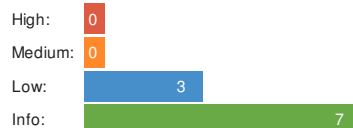
✓ https://circloopstation.surge.sh/

Summary

Overall risk level:

Low

Risk ratings:





Scan information:

Start time: 2020-06-09 23:15:40 UTC+03
Finish time: 2020-06-09 23:15:45 UTC+03
Scan duration: 5 sec
Tests performed: 10/10
Scan status: **Finished**

Findings

Server software and technology found

Software / Version	Category
 Firebase	Databases
 jQuery 1.9.1	JavaScript Frameworks

▼ Details

Risk description:

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 permit the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

More information about this issue:

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

🚩 Missing HTTP security headers

HTTP Security Header	Header Role	Status
X-Frame-Options	Protects against Clickjacking attacks	Not set
X-XSS-Protection	Mitigates Cross-Site Scripting (XSS) attacks	Not set
Strict-Transport-Security	Protects against man-in-the-middle attacks	Not set
X-Content-Type-Options	Prevents possible phishing or XSS attacks	Not set

▼ Details

Risk description:

Because the **X-Frame-Options** header is not sent by the server, an attacker could embed this website into an iframe of a third party website. By manipulating the display attributes of the iframe, the attacker could trick the user into performing mouse clicks in the application, thus performing activities without user's consent (ex: delete user, subscribe to newsletter, etc). This is called a Clickjacking attack and it is described in detail here:

<https://owasp.org/www-community/attacks/Clickjacking>

The **X-XSS-Protection** HTTP header instructs the browser to stop loading web pages when they detect reflected Cross-Site Scripting (XSS) attacks. Lack of this header exposes application users to XSS attacks in case the web application contains such vulnerability.

The HTTP **Strict-Transport-Security** header instructs the browser not to load the website via plain HTTP connection but always use HTTPS. Lack of this header exposes the application users to the risk of data theft or unauthorized modification in case the attacker implements a man-in-the-middle attack and intercepts the communication between the user and the server.

The HTTP **X-Content-Type-Options** header is addressed to Internet Explorer browser and prevents it from reinterpreting the content of a web page (MIME-sniffing) and thus overriding the value of the Content-Type header). Lack of this header could lead to attacks such as Cross-Site Scripting or phishing.

Recommendation:

We recommend you to add the **X-Frame-Options** HTTP response header to every page that you want to be protected against Clickjacking attacks.

More information about this issue:

https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html

We recommend setting the **X-XSS-Protection** header to "X-XSS-Protection: 1; mode=block".

More information about this issue:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>

We recommend setting the **Strict-Transport-Security** header.

More information about this issue:

https://www.owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet

We recommend setting the **X-Content-Type-Options** header to "X-Content-Type-Options: nosniff".

More information about this issue:

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

🚩 Robots.txt file found

<https://circloopstation.surge.sh/robots.txt>

▼ Details

Risk description:

There is no particular security risk in having a robots.txt file. However, this file is often misused to try to hide some web pages from the users. This should not be done as a security measure because these URLs can easily be read from the robots.txt file.

Recommendation:

We recommend you to remove the entries from robots.txt which lead to sensitive locations in the website (ex. administration panels, configuration files, etc).

More information about this issue:

<https://www.theregister.co.uk/2015/05/19/robotstxt/>

🚩 No vulnerabilities found for server-side software

🚩 No security issue found regarding HTTP cookies

🚩 Communication is secure

🚩 No security issue found regarding client access policies

🚩 Directory listing not found (quick scan)

🚩 No password input found (auto-complete test)

🚩 No password input found (clear-text submission test)

Scan coverage information

List of tests performed (10/10)

- ✓ Fingerprinting the server software and technology...
- ✓ Checking for vulnerabilities of server-side software...
- ✓ Analyzing the security of HTTP cookies...
- ✓ Analyzing HTTP security headers...
- ✓ Checking for secure communication...
- ✓ Checking robots.txt file...
- ✓ Checking client access policies...
- ✓ Checking for directory listing (quick scan)...
- ✓ Checking for password auto-complete (quick scan)...
- ✓ Checking for clear-text submission of passwords (quick scan)...

Scan parameters

Website URL: <https://circloopstation.surge.sh/>
Scan type: Light
Authentication: False
