

Website Vulnerability Scanner Report (Light)



Unlock the full capabilities of this scanner



See what the DEEP scanner can do

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

Testing areas	Light scan	Deep 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://www[REDACTED].com/

Target added due to a redirect from https://www[REDACTED].com



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:

High

Risk ratings:

Critical:	0
High:	1
Medium:	1
Low:	6
Info:	3

Scan information:

Start time:	Aug 11, 2026 / 08:45:03 UTC+03
Finish time:	Aug 11, 2026 / 08:45:45 UTC+03
Scan duration:	42 sec
Tests performed:	41
Scan status:	Finished

Findings



Vulnerabilities found for php 8.2.30

port 443/tcp

CVE	CVSS	EPSS Score	EPSS Percentile	Summary
-----	------	---------------	--------------------	---------

CVE-2026-6722	9.5	0.00894	0.56143	In PHP versions 8.2.* before 8.2.31, 8.3.* before 8.3.31, 8.4.* before 8.4.21, and 8.5.* before 8.5.6, the SOAP extension's object deduplication mechanism stores pointers to PHP objects in a global map without incrementing their reference counts. When an apache:Map node contains duplicate keys, processing the second entry overwrites the first in the temporary result map, freeing the original PHP object while its stale pointer remains in the map. A subsequent href reference to the freed node can copy the dangling pointer into the result. As PHP string allocations can reclaim the freed memory region, an attacker with control over the SOAP request body can exploit this use-after-free to achieve remote code execution.
CVE-2026-17543	8.1	0.00468	0.38139	Improper escaping of backslashes in attacker-provided parameters would allow for trivial SQL injection in PHP versions from 8.2.* before 8.2.33, from 8.3.* before 8.3.33, from 8.4.* before 8.4.24, and from 8.5.* before 8.5.9.
CVE-2025-14179	7.4	0.00439	0.36111	In PHP versions 8.2.* before 8.2.31, 8.3.* before 8.3.31, 8.4.* before 8.4.21, and 8.5.* before 8.5.6, the PDO Firebird driver improperly handles NUL bytes when preparing SQL queries. During token-by-token query construction, a string token containing a NUL byte is copied via strncat(), which stops at the NUL byte, dropping the closing quote and causing subsequent SQL tokens to be interpreted as part of the string. This allows SQL injection when attacker-controlled values are quoted via PDO::quote() and embedded in SQL statements.
CVE-2026-6735	7.3	0.0021	0.11295	In PHP versions 8.2.* before 8.2.31, 8.3.* before 8.3.31, 8.4.* before 8.4.21, 8.5.* before 8.5.6, due to improper sanitation of user data, it allows an attacker to compose an URL, which will cause the target to execute arbitrary JavaScript code (XSS) on the target's machine when the target is viewing the PHP-FPM status page.
CVE-2026-7568	6.3	0.00455	0.37308	In PHP versions 8.2.* before 8.2.31, 8.3.* before 8.3.31, 8.4.* before 8.4.21, and 8.5.* before 8.5.6, the metaphone() function in ext/standard/metaphone.c uses a signed int variable to track the current position within the input string. If a string longer than 2,147,483,647 bytes is passed, a signed integer overflow occurs, resulting in undefined behavior. This can lead to an out-of-bounds read, causing a segmentation fault or access to unrelated memory, and may affect the availability of the PHP process.

▼ Details

Risk description:

The risk is that an attacker could search for an appropriate exploit (or create one himself) for any of these vulnerabilities and use it to attack the system.

Since the vulnerabilities were discovered using only version-based testing, the risk level for this finding will not exceed 'high' severity. Critical risks will be assigned to vulnerabilities identified through accurate active testing methods.

Recommendation:

In order to eliminate the risk of these vulnerabilities, we recommend you check the installed software version and upgrade to the latest version.

Classification:

EPSS score : 0.00894

EPSS percentile : 0.56143

CISA KEV: False

CVE : [CVE-2026-6722](#), [CVE-2026-17543](#), [CVE-2025-14179](#), [CVE-2026-6735](#), [CVE-2026-7568](#)

CVSS V3 : 9.5

CWE : [CWE-1035](#)

Vulnerabilities found for jquery 1.11.3

port 443/tcp

CVE	CVSS	EPSS Score	EPSS Percentile	Summary
CVE-2020-11023	6.9	0.8383	0.99664	In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.
CVE-2020-11022	6.9	0.99019	0.99926	In jQuery starting with 1.12.0 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.
CVE-2019-11358	6.1	0.87218	0.99734	jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.

CVE-2015-9251	6.1	0.29726	0.98018	jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.
---------------	-----	---------	---------	--

▼ Details

Risk description:

The risk is that an attacker could search for an appropriate exploit (or create one himself) for any of these vulnerabilities and use it to attack the system.

Since the vulnerabilities were discovered using only version-based testing, the risk level for this finding will not exceed 'high' severity. Critical risks will be assigned to vulnerabilities identified through accurate active testing methods.

Recommendation:

In order to eliminate the risk of these vulnerabilities, we recommend you check the installed software version and upgrade to the latest version.

Classification:

EPSS score : 0.99019

EPSS percentile : 0.99926

CISA KEV: False

CVE : [CVE-2020-11023](#), [CVE-2020-11022](#), [CVE-2019-11358](#), [CVE-2015-9251](#)

CVSS V3 : 6.9

CWE : [CWE-1035](#)

🚩 Missing security header: Strict-Transport-Security

CONFIRMED

port 443/tcp

URL	Evidence
https://www.████.com/	Response headers do not include the HTTP Strict-Transport-Security header Request / Response

▼ Details

Risk description:

The risk is that lack of this header permits an attacker to force a victim user to initiate a clear-text HTTP connection to the server, thus opening the possibility to eavesdrop on the network traffic and extract sensitive information (e.g. session cookies).

Recommendation:

The Strict-Transport-Security HTTP header should be sent with each HTTPS response. The syntax is as follows:

`Strict-Transport-Security: max-age=<seconds>[; includeSubDomains]`

The parameter `max-age` gives the time frame for requirement of HTTPS in seconds and should be chosen quite high, e.g. several months. A value below 7776000 is considered as too low by this scanner check.

The flag `includeSubDomains` defines that the policy applies also for sub domains of the sender of the response.

Classification:

CWE : [CWE-693](#)

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

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

OWASP Top 10 - 2025 : [A02 - Security Misconfiguration](#)

🚩 Missing security header: Referrer-Policy

CONFIRMED

port 443/tcp

URL	Evidence
https://www.████.com/	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](#)

OWASP Top 10 - 2025 : [A02 - Security Misconfiguration](#)

Unsafe security header: Content-Security-Policy

CONFIRMED

port 443/tcp

URL	Evidence
https://www[REDACTED].com/	Response headers include the HTTP Content-Security-Policy security header with the following security issues: <pre>script-src: script-src directive is missing. object-src: Missing object-src allows the injection of plugins which can execute JavaScript. We recommend setting it to 'none'. base-uri: Missing base-uri allows the injection of base tags. They can be used to set the base URL for all relative (script) URLs to an attacker controlled domain. We recommend setting it to 'none' or 'self'. default-src: The default-src directive should be set as a fall-back when other restrictions have not been specified.</pre> Request / Response

Details**Risk description:**

For example, if the unsafe-inline directive is present in the CSP header, the execution of inline scripts and event handlers is allowed. This can be exploited by an attacker to execute arbitrary JavaScript code in the context of the vulnerable application.

Recommendation:

Remove the unsafe values from the directives, adopt nonces or hashes for safer inclusion of inline scripts if they are needed, and explicitly define the sources from which scripts, styles, images or other resources can be loaded.

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-1021](#)

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

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

OWASP Top 10 - 2025 : [A02 - Security Misconfiguration](#)

Missing security header: X-Content-Type-Options

CONFIRMED

port 443/tcp

URL	Evidence
https://www[REDACTED].com/	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](#)
OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)
OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)
OWASP Top 10 - 2025 : [A02 - Security Misconfiguration](#)

🚩 Robots.txt file found

port 443/tcp

CONFIRMED

URL

<https://www.██████.com/robots.txt>

▼ Details

Risk description:

There is no particular security risk in having a robots.txt file. However, it's important to note that adding endpoints in it should not be considered a security measure, as this file can be directly accessed and read by anyone.

Recommendation:

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

References:

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

Classification:

OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)
OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)
OWASP Top 10 - 2025 : [A06 - Insecure Design](#)

🚩 Server software and technology found

port 443/tcp

UNCONFIRMED ⓘ

Software / Version	Category
 Facebook Chat Plugin	Live chat
 Font Awesome 4.7.0	Font scripts
 Google Font API	Font scripts
 Bootstrap	UI frameworks
 GSAP	JavaScript frameworks
 HTTP/3	Miscellaneous
 jQuery 1.11.3	JavaScript libraries
 MailChimp	Marketing automation, Email
 Google Maps	Maps
 MySQL	Databases
 PHP 8.2.30	Programming languages
 Spin.js	JavaScript libraries
 WordPress	CMS, Blogs
 Hostinger	Hosting
 MailChimp for WordPress	WordPress plugins, Marketing automation
 YouTube	Video players

▼ 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:

CWE : [CWE-200](#)

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

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

OWASP Top 10 - 2025 : [A02 - Security Misconfiguration](#)

 Security.txt file is missing**CONFIRMED**

port 443/tcp

URL

Missing: <https://www.████████.com/.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:

CWE : [CWE-1188](#)

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

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

OWASP Top 10 - 2025 : [A02 - Security Misconfiguration](#)

 HTTP OPTIONS enabled**CONFIRMED**

port 443/tcp

URL	Method	Summary
https://www.████████.com/	OPTIONS	We did a HTTP OPTIONS request. The server responded with a 200 status code and the header: Allow: GET,HEAD Request / Response

Details**Risk description:**

The only risk this might present nowadays is revealing debug HTTP methods that can be used on the server. This can present a danger if any of those methods can lead to sensitive information, like authentication information, secret keys.

Recommendation:

We recommend that you check for unused HTTP methods or even better, disable the OPTIONS method. This can be done using your webserver configuration.

References:

<https://techcommunity.microsoft.com/t5/iis-support-blog/http-options-and-default-page-vulnerabilities/ba-p/1504845>

<https://docs.nginx.com/nginx-management-suite/acm/how-to/policies/allowed-http-methods/>

Classification:

CWE : [CWE-16](#)

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

Email Address Exposure

port 443/tcp

UNCONFIRMED 

URL	Method	Parameters	Evidence
https://www[REDACTED].com/	GET	Headers: User-Agent=Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36	Email Address: info@s[REDACTED].com Request / Response

▼ Details

Risk description:

The risk is that exposed email addresses within the application could be accessed by unauthorized parties. This could lead to privacy violations, spam, phishing attacks, or other forms of misuse.

Recommendation:

Compartmentalize the application to have 'safe' areas where trust boundaries can be unambiguously drawn. Do not allow email addresses to go outside of the trust boundary, and always be careful when interfacing with a compartment outside of the safe area.

References:

https://owasp.org/Top10/A04_2021-Insecure_Design/

Classification:

CISA KEV: False

CVE : -1

CWE : [CWE-200](#)

OWASP Top 10 - 2017 : A6: Security Misconfiguration

OWASP Top 10 - 2021 : A4: Insecure Design

OWASP Top 10 - 2025 : [A06 - Insecure Design](#)

Scan coverage information

List of tests performed (41)

Port 443

- ✓ Scanned for version-based vulnerabilities of server-side software
- ✓ Scanned for version-based vulnerabilities of server-side software
- ✓ Scanned for missing HTTP header - Referrer
- ✓ Scanned for missing HTTP header - Strict-Transport-Security
- ✓ Scanned for missing HTTP header - X-Content-Type-Options
- ✓ Scanned for robots.txt file
- ✓ Scanned for website technologies
- ✓ Scanned for unsafe HTTP header Content Security Policy
- ✓ Scanned for emails
- ✓ Scanned for enabled HTTP OPTIONS method
- ✓ Scanned for absence of the security.txt file
- ✓ Scanned for client access policies
- ✓ Scanned for use of untrusted certificates
- ✓ Scanned for enabled HTTP debug methods
- ✓ Scanned for secure communication
- ✓ Scanned for directory listing
- ✓ Scanned for passwords submitted unencrypted
- ✓ Scanned for error messages
- ✓ Scanned for debug messages
- ✓ Scanned for code comments
- ✓ Scanned for missing HTTP header - Content Security Policy
- ✓ Scanned for passwords submitted in URLs
- ✓ Scanned for domain too loose set for cookies
- ✓ Scanned for mixed content between HTTP and HTTPS
- ✓ Scanned for cross domain file inclusion

- ✔ Scanned for internal error code
- ✔ Scanned for HttpOnly flag of cookie
- ✔ Scanned for Secure flag of cookie
- ✔ Scanned for login interfaces
- ✔ Scanned for secure password submission
- ✔ Scanned for sensitive data
- ✔ Scanned for OpenAPI files
- ✔ Scanned for file upload
- ✔ Scanned for SQL statement in request parameter
- ✔ Scanned for password returned in later response
- ✔ Scanned for Path Disclosure
- ✔ Scanned for Session Token in URL
- ✔ Scanned for API endpoints
- ✔ Scanned for missing HTTP header - Rate Limit
- ✔ Scanned for PEM-encoded private key material
- ✔ Scanned for partial PEM-encoded private key material

Scan parameters

target:	https://www. [REDACTED] .com/
scan_type:	Light
authentication:	False

Scan stats

Unique Injection Points Detected:	112
URLs spidered:	7
Total number of HTTP requests:	16
Average time until a response was received:	2264ms