December 1, 2024

Vulnerability Scan Report

Prepared By

HostedScan Security

hostedscan.com

Overview

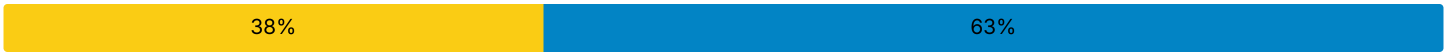
1 Executive Summary

Vulnerability scans were conducted on select servers, networks, websites, and applications. This report contains the discovered potential vulnerabilities from these scans. Vulnerabilities have been classified by severity. Higher severity indicates a greater risk of a data breach, loss of integrity, or availability of the targets.

1.1 Total Vulnerabilities

Below are the total number of vulnerabilities found by severity. Critical vulnerabilities are the most severe and should be evaluated first. An accepted vulnerability is one which has been manually reviewed and classified as acceptable to not fix at this time, such as a false positive detection or an intentional part of the system's architecture.

Critical	High	Medium	Low	Accepted
0	0	3	5	0



1.2 Report Coverage

This report includes findings for 1 target scanned. Each target is a single URL, IP address, or fully qualified domain name (FQDN).

Vulnerability Categories

8

Passive Web Application Vulnerabilities

2 Vulnerabilities By Target

This section contains the vulnerability findings for each scanned target. Prioritization should be given to the targets with the highest severity vulnerabilities. However, it is important to take into account the purpose of each system and consider the potential impact a breach or an outage would have for the particular target.

2.1 Targets Summary

The number of potential vulnerabilities found for each target by severity.

Target	● Critical	● High	● Medium	● Low	● Accepted
● https://myprojectnetwork.com/	0	0	3	5	0

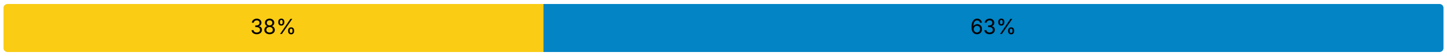
2.2 Target Breakdowns

Details for the potential vulnerabilities found for each target by scan type.

https://myprojectnetwork.com/

Total Risks

0	0	3	5	0



Passive Web Application Vulnerabilities	Severity	First Detected	Last Detected
Missing Anti-clickjacking Header	● Medium	0 days ago	0 days ago
Content Security Policy (CSP) Header Not Set	● Medium	0 days ago	0 days ago
Vulnerable JS Library	● Medium	0 days ago	0 days ago
X-Content-Type-Options Header Missing	● Low	0 days ago	0 days ago
Cross-Domain JavaScript Source File Inclusion	● Low	0 days ago	0 days ago
Cookie No HttpOnly Flag	● Low	0 days ago	0 days ago
Strict-Transport-Security Header Not Set	● Low	0 days ago	0 days ago
Big Redirect Detected (Potential Sensitive Information Leak)	● Low	0 days ago	0 days ago

3 Passive Web Application Vulnerabilities

The OWASP ZAP Passive Web Application scan crawls the pages of a website or web application. The passive scan inspects each page as well as the requests and responses sent between the server. The passive scan checks for vulnerabilities such as cross-domain misconfigurations, insecure cookies, vulnerable Javascript dependencies, and more.

3.1 Total Vulnerabilities

Total number of vulnerabilities found by severity.

Critical	High	Medium	Low	Accepted
0	0	3	5	0

3.2 Vulnerabilities Breakdown

Summary list of all detected vulnerabilities.

Title	Severity	Open	Accepted
Missing Anti-clickjacking Header	● Medium	1	0
Content Security Policy (CSP) Header Not Set	● Medium	1	0
Vulnerable JS Library	● Medium	1	0
X-Content-Type-Options Header Missing	● Low	1	0
Cross-Domain JavaScript Source File Inclusion	● Low	1	0
Cookie No HttpOnly Flag	● Low	1	0
Strict-Transport-Security Header Not Set	● Low	1	0
Big Redirect Detected (Potential Sensitive Information Leak)	● Low	1	0

3.3 Vulnerability Details

Detailed information about each potential vulnerability found by the scan.

Missing Anti-clickjacking Header

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description

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

Solution

Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.

If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.

Instances (1 of 6)

uri: https://myprojectnetwork.com/
method: GET
param: x-frame-options

References

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

Vulnerable Target	First Detected	Last Detected
https://myprojectnetwork.com/	0 days ago	0 days ago

Content Security Policy (CSP) Header Not Set

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

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.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Instances (1 of 9)

uri: https://myprojectnetwork.com/
method: GET

References

https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://www.w3.org/TR/CSP/
https://w3c.github.io/webappsec-csp/
https://web.dev/articles/csp
https://caniuse.com/#feat=contentsecuritypolicy
https://content-security-policy.com/

Vulnerable Target	First Detected	Last Detected
https://myprojectnetwork.com/	0 days ago	0 days ago

Vulnerable JS Library

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description

The identified library bootstrap, version 67211654735dd822d79ac35c--tangerine-genie-222549.netlify.app is vulnerable.

Solution

Please upgrade to the latest version of bootstrap.

Instances (1 of 2)

uri: https://67211654735dd822d79ac35c--tangerine-genie-222549.netlify.app/js/bootstrap.min.js
method: GET
evidence: /67211654735dd822d79ac35c--tangerine-genie-222549.netlify.app/js/bootstrap.min.js
otherinfo: CVE-2018-14041 CVE-2018-20677 CVE-2018-20676 CVE-2018-14042

References

https://github.com/advisories/GHSA-pj7m-g53m-7638
https://github.com/twbs/bootstrap/issues/20184
https://github.com/advisories/GHSA-ph58-4vrj-w6hr
https://github.com/twbs/bootstrap/issues/20631
https://github.com/twbs/bootstrap/pull/3421
https://nvd.nist.gov/vuln/detail/CVE-2018-20676

Vulnerable Target	First Detected	Last Detected
https://myprojectnetwork.com/	0 days ago	0 days ago

X-Content-Type-Options Header Missing

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

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.

Solution

Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

Instances (1 of 7)

uri: https://myprojectnetwork.com/
method: GET
param: x-content-type-options
otherinfo: This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.

References

https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85)
https://owasp.org/www-community/Security_Headers

Vulnerable Target	First Detected	Last Detected
https://myprojectnetwork.com/	0 days ago	0 days ago

Cross-Domain JavaScript Source File Inclusion

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

The page includes one or more script files from a third-party domain.

Solution

Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

Instances (1 of 8)

```
uri: https://myprojectnetwork.com/
method: GET
param: https://67211654735dd822d79ac35c--tangerine-genie-222549.netlify.app/js/bootstrap.min.js
evidence: <script src="https://67211654735dd822d79ac35c--tangerine-genie-222549.netlify.app/js/bootstrap.min.js"></script>
```

Vulnerable Target	First Detected	Last Detected
https://myprojectnetwork.com/	0 days ago	0 days ago

Cookie No HttpOnly Flag

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.

Solution

Ensure that the HttpOnly flag is set for all cookies.

Instances (1 of 11)

uri: https://myprojectnetwork.com/
method: GET
param: XSRF-TOKEN
evidence: set-cookie: XSRF-TOKEN

References

https://owasp.org/www-community/HttpOnly

Vulnerable Target	First Detected	Last Detected
https://myprojectnetwork.com/	0 days ago	0 days ago

Strict-Transport-Security Header Not Set

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.

Instances (1 of 11)

uri: https://myprojectnetwork.com/
method: GET

References

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html
https://owasp.org/www-community/Security_Headers
https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
https://caniuse.com/stricttransportsecurity
https://datatracker.ietf.org/doc/html/rfc6797

Vulnerable Target	First Detected	Last Detected
https://myprojectnetwork.com/	0 days ago	0 days ago

Big Redirect Detected (Potential Sensitive Information Leak)

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

The server has responded with a redirect that seems to provide a large response. This may indicate that although the server sent a redirect it also responded with body content (which may include sensitive details, PII, etc.).

Solution

Ensure that no sensitive information is leaked via redirect responses. Redirect responses should have almost no content.

Instances (1 of 5)

uri: https://myprojectnetwork.com/auth/redirect
method: GET
otherinfo: Location header URI length: 283 [https://accounts.google.com/o/oauth2/auth?client_id=962765230566-khjvlf3ld4mu21mclq3uolms3ap0es7b.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fmyprojectnetwork.com%2Fauth%2Fcallback&scope=openid+profile+email&response_type=code&state=T1M6MSJMOfZWQHw9EKQ178Dw0BBn32ixVTM87YCc]. Predicted response size: 583. Response Body Length: 1,442.

Vulnerable Target	First Detected	Last Detected
https://myprojectnetwork.com/	0 days ago	0 days ago

4 Glossary

Accepted Vulnerability

An accepted vulnerability is one which has been manually reviewed and classified as acceptable to not fix at this time, such as a false positive scan result or an intentional part of the system's architecture.

Fully Qualified Domain Name (FQDN)

A fully qualified domain name is a complete domain name for a specific website or service on the internet. This includes not only the website or service name, but also the top-level domain name, such as .com, .org, .net, etc. For example, 'www.example.com' is an FQDN.

Passive Web Application Vulnerabilities

The OWASP ZAP Passive Web Application scan crawls the pages of a website or web application. The passive scan inspects each page as well as the requests and responses sent between the server. The passive scan checks for vulnerabilities such as cross-domain misconfigurations, insecure cookies, vulnerable Javascript dependencies, and more.

Vulnerability

A weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability. Mitigation of the vulnerabilities in this context typically involves coding changes, but could also include specification changes or even specification deprecations (e.g., removal of affected protocols or functionality in their entirety).

Target

A target represents target is a single URL, IP address, or fully qualified domain name (FQDN) that was scanned.

Severity

Severity represents the estimated impact potential of a particular vulnerability. Severity is divided into 5 categories: Critical, High, Medium, Low and Accepted.

CVSS Score

The CVSS 3.0 score is a global standard for evaluating vulnerabilities with a 0 to 10 scale. CVSS maps to threat levels:
0.1 - 3.9 = Low
4.0 - 6.9 = Medium
7.0 - 8.9 = High
9.0 - 10.0 = Critical

This report was prepared using

HostedScan Security ®

For more information, visit **hostedscan.com**

Founded in Seattle, Washington in 2019, HostedScan, LLC. is dedicated to making continuous vulnerability scanning and risk management much more easily accessible to more businesses.

HostedScan, LLC.

2212 Queen Anne Ave N
Suite #521
Seattle, WA 98109

Terms & Policies
hello@hostedscan.com