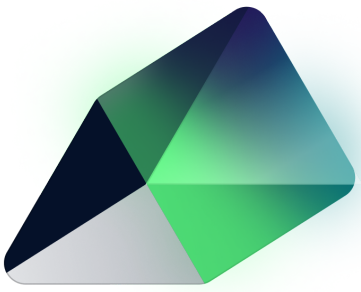
July 1, 2025

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	5	2	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

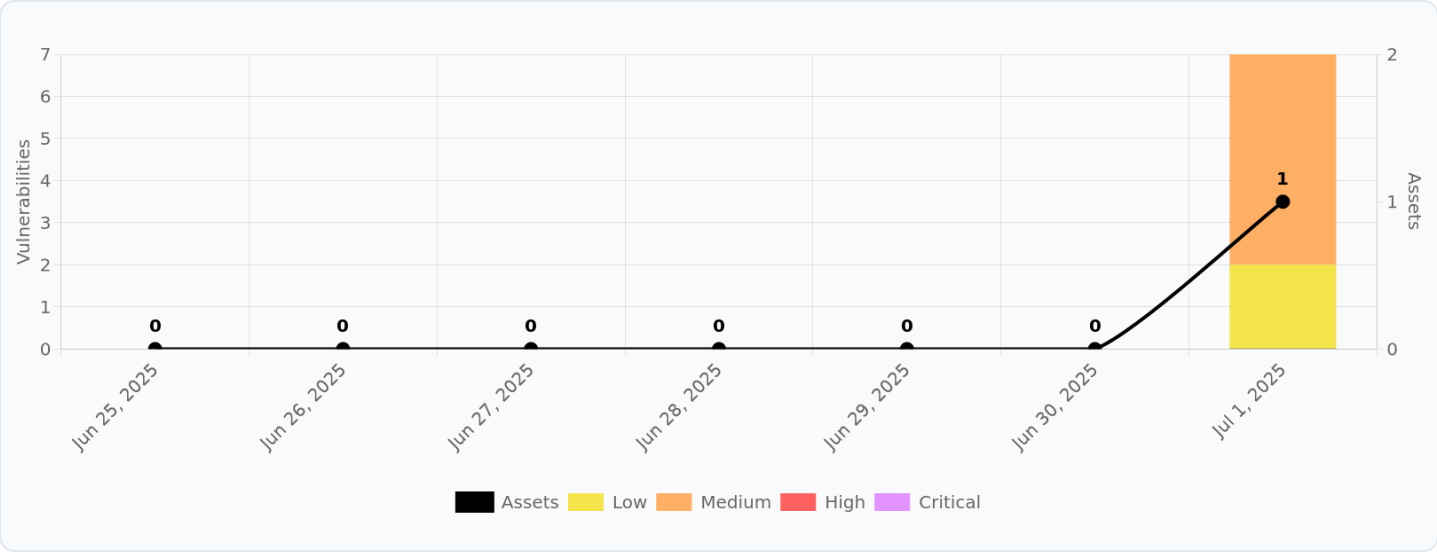
7

Passive Web Application Vulnerabilities

2 Trends

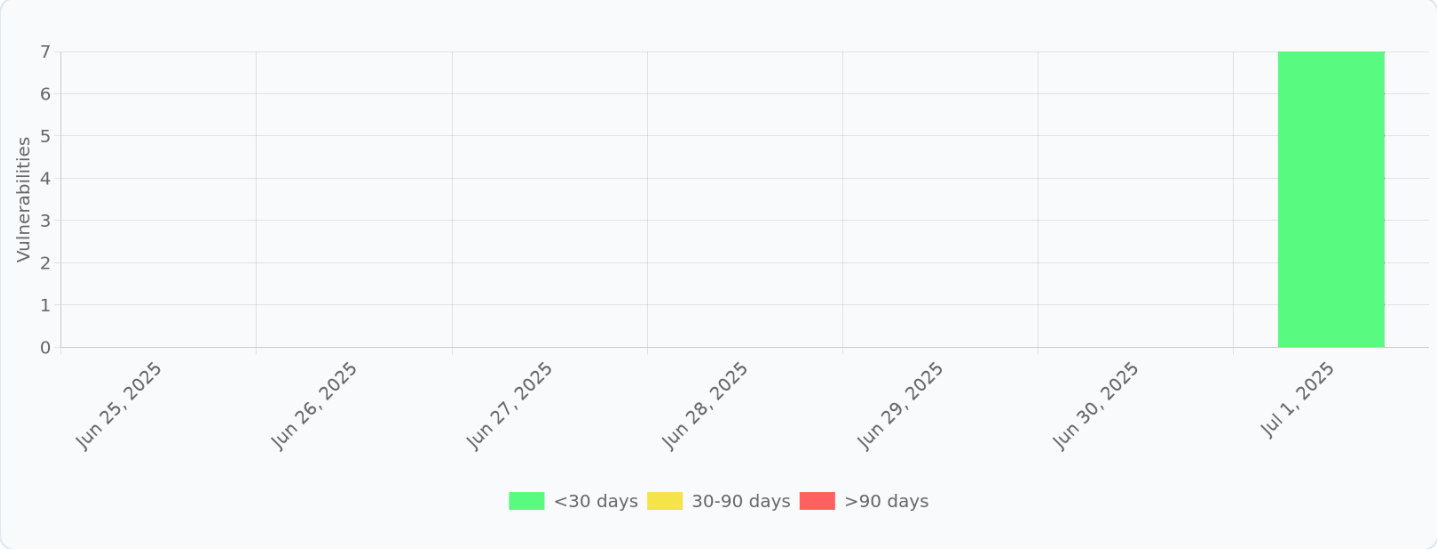
2.1 Open Risks

Total number of vulnerabilities grouped by severity level.



2.2 Exposure Window

Total number of unresolved vulnerabilities grouped by age (time since first detection).



3 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.

3.1 Targets Summary

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

Target	Critical	High	Medium	Low	Accepted
https://www.smecai.online/	0	0	5	2	0

3.2 Target Breakdowns

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

https://www.smecai.online/

Total Risks

0	0	5	2	0

Passive Web Application Vulnerabilities	Severity	First Detected	Last Detected
Cross-Domain Misconfiguration	Medium	0 days ago	0 days ago
CSP: Wildcard Directive	Medium	0 days ago	0 days ago
CSP: script-src unsafe-inline	Medium	0 days ago	0 days ago
CSP: style-src unsafe-inline	Medium	0 days ago	0 days ago
CSP: script-src unsafe-eval	Medium	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

4 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.

4.1 Total Vulnerabilities

Total number of vulnerabilities found by severity.

Critical	High	Medium	Low	Accepted
0	0	5	2	0

4.2 Vulnerabilities Breakdown

Summary list of all detected vulnerabilities.

Title	Severity	Open	Accepted
Cross-Domain Misconfiguration	Medium	1	0
CSP: Wildcard Directive	Medium	1	0
CSP: script-src unsafe-inline	Medium	1	0
CSP: style-src unsafe-inline	Medium	1	0
CSP: script-src unsafe-eval	Medium	1	0
Strict-Transport-Security Header Not Set	Low	1	0
Big Redirect Detected (Potential Sensitive Information Leak)	Low	1	0

4.3 Vulnerability Details

Detailed information about each potential vulnerability found by the scan.

Cross-Domain Misconfiguration

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description

Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server.

Solution

Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance).

Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.

Instances (1 of 21)

uri: https://www.smecai.online/_next/image?q=75&url=%2Flogo-h-b.png&w=828
method: GET
evidence: Access-Control-Allow-Origin: *
otherinfo: The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.

References

https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy

Vulnerable Target	First Detected	Last Detected
https://www.smecai.online/	0 days ago	0 days ago

CSP: Wildcard Directive

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 (but not limited to) 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 properly configured to set the Content-Security-Policy header.

Instances (1 of 15)

uri: https://www.smecai.online/
method: GET
param: Content-Security-Policy
evidence: default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self' 'unsafe-inline'; img-src 'self' data: https: blob:; font-src 'self' data:; connect-src 'self' https://api.openai.com https://*.anthropic.com https://*.neon.tech; media-src 'self'; object-src 'none'; base-uri 'self'; form-action 'self'; frame-ancestors 'none'; upgrade-insecure-requests
otherinfo: The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: img-src

References

https://www.w3.org/TR/CSP/
https://caniuse.com/#search=content+security+policy
https://content-security-policy.com/
https://github.com/HtmlUnit/htmlunit-csp
https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources

Vulnerable Target	First Detected	Last Detected
https://www.smecai.online/	0 days ago	0 days ago

CSP: script-src unsafe-inline

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 (but not limited to) 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 properly configured to set the Content-Security-Policy header.

Instances (1 of 15)

uri: https://www.smecai.online/
method: GET
param: Content-Security-Policy
evidence: default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self' 'unsafe-inline'; img-src 'self' data: https: blob:; font-src 'self' data:; connect-src 'self' https://api.openai.com https://*.anthropic.com https://*.neon.tech; media-src 'self'; object-src 'none'; base-uri 'self'; form-action 'self'; frame-ancestors 'none'; upgrade-insecure-requests
otherinfo: script-src includes unsafe-inline.

References

https://www.w3.org/TR/CSP/
https://caniuse.com/#search=content+security+policy
https://content-security-policy.com/
https://github.com/HtmlUnit/htmlunit-csp
https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources

Vulnerable Target	First Detected	Last Detected
https://www.smecai.online/	0 days ago	0 days ago

CSP: style-src unsafe-inline

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 (but not limited to) 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 properly configured to set the Content-Security-Policy header.

Instances (1 of 15)

uri: https://www.smecai.online/
method: GET
param: Content-Security-Policy
evidence: default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self' 'unsafe-inline'; img-src 'self' data: https: blob:; font-src 'self' data:; connect-src 'self' https://api.openai.com https://*.anthropic.com https://*.neon.tech; media-src 'self'; object-src 'none'; base-uri 'self'; form-action 'self'; frame-ancestors 'none'; upgrade-insecure-requests
otherinfo: style-src includes unsafe-inline.

References

https://www.w3.org/TR/CSP/
https://caniuse.com/#search=content+security+policy
https://content-security-policy.com/
https://github.com/HtmlUnit/htmlunit-csp
https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources

Vulnerable Target	First Detected	Last Detected
https://www.smecai.online/	0 days ago	0 days ago

CSP: script-src unsafe-eval

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 (but not limited to) 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 properly configured to set the Content-Security-Policy header.

Instances (1 of 15)

uri: https://www.smecai.online/
method: GET
param: Content-Security-Policy
evidence: default-src 'self'; script-src 'self' 'unsafe-inline' 'unsafe-eval'; style-src 'self' 'unsafe-inline'; img-src 'self' data: https: blob:; font-src 'self' data:; connect-src 'self' https://api.openai.com https://*.anthropic.com https://*.neon.tech; media-src 'self'; object-src 'none'; base-uri 'self'; form-action 'self'; frame-ancestors 'none'; upgrade-insecure-requests
otherinfo: script-src includes unsafe-eval.

References

https://www.w3.org/TR/CSP/
https://caniuse.com/#search=content+security+policy
https://content-security-policy.com/
https://github.com/HtmlUnit/htmlunit-csp
https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources

Vulnerable Target	First Detected	Last Detected
https://www.smecai.online/	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 1)

uri: https://www.smecai.online/_next/image?url=%2Flogo-h-b.png&w=640&q=75
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://www.smecai.online/	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 12)

uri: https://www.smecai.online/
method: GET
otherinfo: Location header URI length: 37 [/07ee9d54-83b1-4a44-b22d-b41c175e5019]. Predicted response size: 337. Response Body Length: 5,514.

Vulnerable Target	First Detected	Last Detected
https://www.smecai.online/	0 days ago	0 days ago

5 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

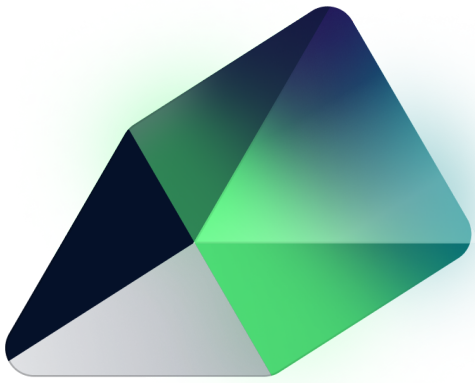
The CVSS 3.0 score is a global standard for evaluating vulnerabilities with a 0 to 10 scale. CVSS maps to threat levels:
0.1 - 3.9 = Low
4.0 - 6.9 = Medium
7.0 - 8.9 = High
9.0 - 10.0 = Critical

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