ZAP Scanning Report

Site: http://192.168.1.166:5173

Generated on Sat, 3 May 2025 19:20:38

ZAP Version: 2.16.1

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	6
Informational	3
False Positives:	0

Summary of Sequences

For each step: result (Pass/Fail) - risk (of highest alert(s) for the step, if any).

Alerts

Name	Risk Level	Number of Instances
Content Security Policy (CSP) Header Not Set	Medium	2
Missing Anti-clickjacking Header	Medium	1
In Page Banner Information Leak	Low	2
Insufficient Site Isolation Against Spectre Vulnerability	Low	6
Permissions Policy Header Not Set	Low	4
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	6
Timestamp Disclosure - Unix	Low	1
X-Content-Type-Options Header Missing	Low	4
Information Disclosure - Suspicious Comments	Informational	1
Modern Web Application	Informational	1
Storable and Cacheable Content	Informational	6

Alert Detail

Medium	Content Security Policy (CSP) Header Not Set
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.
URL	http://192.168.1.166:5173
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://192.168.1.166:5173/robots.txt
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
Instances	2
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Reference	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/
CWE Id	693
WASC Id	15
Plugin Id	10038
Medium	**Missing Anti-clickjacking Header**
Description	The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
URL	http://192.168.1.166:5173
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Other Info	
Instances	1
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.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id	1021
WASC Id	15
Plugin Id	10020
Low	**In Page Banner Information Leak**
Description	The server returned a version banner string in the response content. Such information leaks may allow attackers to further target specific issues impacting the product and version in use.
URL	http://192.168.1.166:5173/robots.txt
Method	GET
Parameter	
Attack	
Evidence	nginx/1.28.0
Other Info	There is a chance that the highlight in the finding is on a value in the headers, versus the actual matched string in the response body.
URL	http://192.168.1.166:5173/sitemap.xml
Method	GET
Parameter	
Attack	
Evidence	nginx/1.28.0
Other Info	There is a chance that the highlight in the finding is on a value in the headers, versus the actual matched string in the response body.
Instances	2
Solution	Configure the server to prevent such information leaks. For example: Under Tomcat this is done via the "server" directive and implementation of custom error pages. Under Apache this is done via the "ServerSignature" and "ServerTokens" directives.
Reference	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/08-Testing_for_Error_Handling/
CWE Id	497
WASC Id	13
Plugin Id	10009
Low	**Insufficient Site Isolation Against Spectre Vulnerability**
Description	Cross-Origin-Resource-Policy header is an opt-in header designed to counter side-channels attacks like Spectre. Resource should be specifically set as shareable amongst different origins.
URL	http://192.168.1.166:5173
Method	GET
Parameter	Cross-Origin-Resource-Policy

Attack	
Evidence	
Other Info	
URL	http://192.168.1.166:5173/assets/index-CPJTrgpv.js
Method	GET
Parameter	Cross-Origin-Resource-Policy
Attack	
Evidence	
Other Info	
URL	http://192.168.1.166:5173/assets/index-rmb8RL8g.css
Method	GET
Parameter	Cross-Origin-Resource-Policy
Attack	
Evidence	
Other Info	
URL	http://192.168.1.166:5173/vite.svg
Method	GET
Parameter	Cross-Origin-Resource-Policy
Attack	
Evidence	
Other Info	
URL	http://192.168.1.166:5173
Method	GET
Parameter	Cross-Origin-Embedder-Policy
Attack	
Evidence	
Other Info	
URL	http://192.168.1.166:5173
Method	GET
Parameter	Cross-Origin-Opener-Policy
Attack	
Evidence	
Other Info	
Instances	6
Solution	Ensure that the application/web server sets the Cross-Origin-Resource-Policy header appropriately, and that it sets the Cross-Origin-Resource-Policy header to 'same-origin' for all web pages. 'same-site' is considered as less secured and should be avoided. If resources must be shared, set the header to 'cross-origin'. If possible, ensure that the end user uses a standards-compliant and modern web browser that supports the Cross-Origin-Resource-Policy header (https://caniuse.com/mdn-http_headers_cross-origin-resource-policy).
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Cross-Origin_Resource_Policy
CWE Id	693
WASC Id	14
Plugin Id	90004
Low	**Permissions Policy Header Not Set**
Description	Permissions Policy Header is an added layer of security that helps to restrict from unauthorized access or usage of browser/client features by web resources. This policy ensures the user privacy by limiting or specifying the features of the browsers can be used by the web resources. Permissions Policy provides a set of standard HTTP headers that allow website owners to limit which features of browsers can be used by the page such as camera, microphone, location, full screen etc.
URL	http://192.168.1.166:5173
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://192.168.1.166:5173/assets/index-CPJTrgpv.js
Method	GET
Parameter	
Attack	
Evidence	

Other Info	
URL	http://192.168.1.166:5173/robots.txt
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	http://192.168.1.166:5173/sitemap.xml
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
Instances	4
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Permissions-Policy header.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Permissions-Policy https://developer.chrome.com/blog/feature-policy/ https://scotthelme.co.uk/a-new-security-header-feature-policy/ https://w3c.github.io/webappsec-feature-policy/ https://www.smashingmagazine.com/2018/12/feature-policy/
CWE Id	693
WASC Id	15
Plugin Id	10063

Low	Server Leaks Version Information via "Server" HTTP Response Header Field
Description	The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.
URL	http://192.168.1.166:5173
Method	GET
Parameter	
Attack	
Evidence	nginx/1.28.0
Other Info	
URL	http://192.168.1.166:5173/assets/index-CPJTrgpv.js
Method	GET
Parameter	
Attack	
Evidence	nginx/1.28.0
Other Info	
URL	http://192.168.1.166:5173/assets/index-rmb8RL8g.css
Method	GET
Parameter	
Attack	
Evidence	nginx/1.28.0
Other Info	
URL	http://192.168.1.166:5173/robots.txt
Method	GET
Parameter	
Attack	
Evidence	nginx/1.28.0
Other Info	
URL	http://192.168.1.166:5173/sitemap.xml
Method	GET
Parameter	
Attack	
Evidence	nginx/1.28.0
Other Info	
URL	http://192.168.1.166:5173/vite.svg
Method	GET
Parameter	
Attack	
Evidence	nginx/1.28.0
Other Info	

Instances	6
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.
Reference	https://httpd.apache.org/docs/current/mod/core.html#servertokens https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff648552(v=pandp.10) https://www.troyhunt.com/shhh-dont-let-your-response-headers/
CWE Id	497
WASC Id	13
Plugin Id	10036
Low	**Timestamp Disclosure - Unix**
Description	A timestamp was disclosed by the application/web server. - Unix
URL	http://192.168.1.166:5173/assets/index-CPJTrgpv.js
Method	GET
Parameter	
Attack	
Evidence	1540483477
Other Info	1540483477, which evaluates to: 2018-10-25 16:04:37.
Instances	1
Solution	Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.
Reference	https://cwe.mitre.org/data/definitions/200.html
CWE Id	497
WASC Id	13
Plugin Id	10096
Low	**X-Content-Type-Options Header Missing**
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.
URL	http://192.168.1.166:5173
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
Other Info	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.
URL	http://192.168.1.166:5173/assets/index-CPJTrgpv.js
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
Other Info	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.
URL	http://192.168.1.166:5173/assets/index-rmb8RL8g.css
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
Other Info	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.
URL	http://192.168.1.166:5173/vite.svg
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
Other Info	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.
Instances	4
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.
Reference	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
CWE Id	693
WASC Id	15
Plugin Id	10021

Informational	**Information Disclosure - Suspicious Comments**
Description	The response appears to contain suspicious comments which may help an attacker.
URL	http://192.168.1.166:5173/assets/index-CPJTrgpv.js
Method	GET
Parameter	
Attack	
Evidence	select
Other Info	The following pattern was used: \bSELECT\b and was detected in likely comment: "//www.w3.org/2000/svg";case"math":return"http://www.w3.org/1998/Math/MathML";default:return"http://www.w3.org/1999/xhtml"}}funct", see evidence field for the suspicious comment/snippet.
Instances	1
Solution	Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.
Reference	
CWE Id	615
WASC Id	13
Plugin Id	10027

Informational	**Modern Web Application**
Description	The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.
URL	http://192.168.1.166:5173
Method	GET
Parameter	
Attack	
Evidence	<script type="module" crossorigin src="/assets/index-CPJTrgpv.js"></script>
Other Info	No links have been found while there are scripts, which is an indication that this is a modern web application.
Instances	1
Solution	This is an informational alert and so no changes are required.
Reference	
CWE Id	
WASC Id	
Plugin Id	10109

Informational	**Storable and Cacheable Content**
Description	The response contents are storable by caching components such as proxy servers, and may be retrieved directly from the cache, rather than from the origin server by the caching servers, in response to similar requests from other users. If the response data is sensitive, personal or user-specific, this may result in sensitive information being leaked. In some cases, this may even result in a user gaining complete control of the session of another user, depending on the configuration of the caching components in use in their environment. This is primarily an issue where "shared" caching servers such as "proxy" caches are configured on the local network. This configuration is typically found in corporate or educational environments, for instance.
URL	http://192.168.1.166:5173
Method	GET
Parameter	
Attack	
Evidence	
Other Info	In the absence of an explicitly specified caching lifetime directive in the response, a liberal lifetime heuristic of 1 year was assumed. This is permitted by rfc7234.
URL	http://192.168.1.166:5173/assets/index-CPJTrgpv.js
Method	GET
Parameter	
Attack	
Evidence	
Other Info	In the absence of an explicitly specified caching lifetime directive in the response, a liberal lifetime heuristic of 1 year was assumed. This is permitted by rfc7234.
URL	http://192.168.1.166:5173/assets/index-rmb8RL8g.css

Method	GET
Parameter	
Attack	
Evidence	
Other Info	In the absence of an explicitly specified caching lifetime directive in the response, a liberal lifetime heuristic of 1 year was assumed. This is permitted by rfc7234.
URL	http://192.168.1.166:5173/robots.txt
Method	GET
Parameter	
Attack	
Evidence	
Other Info	In the absence of an explicitly specified caching lifetime directive in the response, a liberal lifetime heuristic of 1 year was assumed. This is permitted by rfc7234.
URL	http://192.168.1.166:5173/sitemap.xml
Method	GET
Parameter	
Attack	
Evidence	
Other Info	In the absence of an explicitly specified caching lifetime directive in the response, a liberal lifetime heuristic of 1 year was assumed. This is permitted by rfc7234.
URL	http://192.168.1.166:5173/vite.svg
Method	GET
Parameter	
Attack	
Evidence	
Other Info	In the absence of an explicitly specified caching lifetime directive in the response, a liberal lifetime heuristic of 1 year was assumed. This is permitted by rfc7234.
Instances	6
Solution	Validate that the response does not contain sensitive, personal or user-specific information. If it does, consider the use of the following HTTP response headers, to limit, or prevent the content being stored and retrieved from the cache by another user: Cache-Control: no-cache, no-store, must-revalidate, private Pragma: no-cache Expires: 0 This configuration directs both HTTP 1.0 and HTTP 1.1 compliant caching servers to not store the response, and to not retrieve the response (without validation) from the cache, in response to a similar request.
Reference	https://datatracker.ietf.org/doc/html/rfc7234 https://datatracker.ietf.org/doc/html/rfc7231 https://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html
CWE Id	524
WASC Id	13
Plugin Id	10049

Sequence Details

With the associated active scan results.