ZAP Scanning Report

Site: http://localhost:3000

Generated on Sun, 23 Mar 2025 22:47:08

ZAP Version: 2.16.0

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	3
Low	1
Informational	1
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	1
Cross-Domain Misconfiguration	Medium	1
Hidden File Found	Medium	4
Cross-Domain JavaScript Source File Inclusion	Low	2
Modern Web Application	Informational	1

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://localhost:3000/
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
Instances	1
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	Cross-Domain Misconfiguration
Description	Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server.
URL	http://localhost:3000/
Method	GET
Parameter	
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	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.
Instances	1
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.
Reference	https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy
CWE Id	264
WASC Id	14
Plugin Id	10098

Medium	Hidden File Found
Description	A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.
URL	http://localhost:3000/._darcs
Method	GET
Parameter	
Attack	
Evidence	HTTP/1.1 200 OK
Other Info	
URL	http://localhost:3000/.bzr
Method	GET
Parameter	
Attack	
Evidence	HTTP/1.1 200 OK
Other Info	
URL	http://localhost:3000/.hg
Method	GET
Parameter	
Attack	
Evidence	HTTP/1.1 200 OK
Other Info	
URL	http://localhost:3000/BitKeeper

Method	GET
Parameter	
Attack	
Evidence	HTTP/1.1 200 OK
Other Info	
Instances	4
Solution	Consider whether or not the component is actually required in production, if it isn't then disable it. If it is then ensure access to it requires appropriate authentication and authorization, or limit exposure to internal systems or specific source IPs, etc.
Reference	https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html
CWE Id	538
WASC Id	13
Plugin Id	40035

Low	**Cross-Domain JavaScript Source File Inclusion**
Description	The page includes one or more script files from a third-party domain.
URL	http://localhost:3000/
Method	GET
Parameter	//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js
Attack	
Evidence	<script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
Other Info	
URL	http://localhost:3000/
Method	GET
Parameter	//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js
Attack	
Evidence	<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
Other Info	
Instances	2
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.
Reference	
CWE Id	829
WASC Id	15
Plugin Id	10017

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://localhost:3000/
Method	GET
Parameter	
Attack	
Evidence	<script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.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

Sequence Details

With the associated active scan results.