ZAP Scanning Report

Site: https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com

Generated on Sun, 13 Apr 2025 07:22:14

ZAP Version: 2.16.1

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
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
CORS Misconfiguration	Medium	1
Cross-Domain Misconfiguration	Medium	1
Strict-Transport-Security Header Not Set	Low	4
Non-Storable Content	Informational	4

Alert Detail

Medium	CORS Misconfiguration
Description	This CORS misconfiguration could allow an attacker to perform AJAX queries to the vulnerable website from a malicious page loaded by the victim's user agent. In order to perform authenticated AJAX queries, the server must specify the header "Access-Control-Allow-Credentials: true" and the "Access-Control-Allow-Origin" header must be set to null or the malicious page's domain. Even if this misconfiguration doesn't allow authenticated AJAX requests, unauthenticated sensitive content can still be accessed (e.g intranet websites). A malicious page can belong to a malicious website but also a trusted website with flaws (e.g XSS, support of HTTP without TLS allowing code injection through MITM, etc).
URL	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/prod

Method	GET
Parameter	
Attack	origin: https://tGBAdBGC.com
Evidence	
Other Info	
Instances	1
Solution	If a web resource contains sensitive information, the origin should be properly specified in the Access-Control-Allow-Origin header. Only trusted websites needing this resource should be specified in this header, with the most secured protocol supported.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS https://portswigger.net/web-security/cors
CWE Id	942
WASC Id	14
Plugin Id	40040

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	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/prod
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

Low	**Strict-Transport-Security Header Not Set**
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.
URL	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/
Method	GET

Parameter	
Attack	
Evidence	
Other Info	
URL	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/prod
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/robots.txt
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/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 enforce Strict-Transport-Security.
Reference	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
CWE Id	319
WASC Id	15
Plugin Id	10035

Informational	**Non-Storable Content**
Description	The response contents are not storable by caching components such as proxy servers. If the response does not contain sensitive, personal or user-specific information, it may benefit from being stored and cached, to improve performance.
URL	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/
Method	GET
Parameter	

Attack	
Evidence	403
Other Info	
URL	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/prod
Method	GET
Parameter	
Attack	
Evidence	403
Other Info	
URL	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/robots.txt
Method	GET
Parameter	
Attack	
Evidence	403
Other Info	
URL	https://a5i6dlwoka.execute-api.ap-southeast-1.amazonaws.com/sitemap.xml
Method	GET
Parameter	
Attack	
Evidence	403
Other Info	
Instances	4
Solution	The content may be marked as storable by ensuring that the following conditions are satisfied: The request method must be understood by the cache and defined as being cacheable ("GET", "HEAD", and "POST" are currently defined as cacheable) The response status code must be understood by the cache (one of the 1XX, 2XX, 3XX, 4XX, or 5XX response classes are generally understood) The "no-store" cache directive must not appear in the request or response header fields For caching by "shared" caches such as "proxy" caches, the "private" response directive must not appear in the response For caching by "shared" caches such as "proxy" caches, the "Authorization" header field must not appear in the request, unless the response explicitly allows it (using one of the "must-revalidate", "public", or "s-maxage" Cache-Control response directives) In addition to the conditions above, at least one of the following conditions must also be satisfied by the response: It must contain an "Expires" header field It must contain a "max-age" response directive For "shared" caches such as "proxy" caches, it must contain a "s-maxage" response directive It must contain a "Cache Control Extension" that allows it to be cached It must have a status code that is defined as cacheable by default (200, 203, 204, 206, 300, 301, 404, 405, 410, 414, 501).

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.