ZAP Scanning Report

Site: http://gateway:8080

Generated on Sat, 26 Apr 2025 15:13:07

ZAP Version: 2.16.1

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	0
Low	2
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
Insufficient Site Isolation Against Spectre Vulnerability	Low	1
X-Content-Type-Options Header Missing	Low	1
Content-Type Header Missing	Informational	1
Non-Storable Content	Informational	3
Storable and Cacheable Content	Informational	2

Alert Detail

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://gateway:8080/device
Method	GET
Parameter	Cross-Origin-Resource-Policy
Attack	

Evidence	
Other Info	
Instances	1
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	**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://gateway:8080/device
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	1
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	**Content-Type Header Missing**
Description	The Content-Type header was either missing or empty.
URL	http://gateway:8080/products
Method	GET

Parameter	content-type
Attack	
Evidence	
Other Info	
Instances	1
Solution	Ensure each page is setting the specific and appropriate content-type value for the content being delivered.
Reference	https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85)
CWE Id	345
WASC Id	12
Plugin Id	10019

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	http://gateway:8080/products
Method	GET
Parameter	
Attack	
Evidence	401
Other Info	
URL	http://gateway:8080/device/register
Method	POST
Parameter	
Attack	
Evidence	415
Other Info	
URL	http://gateway:8080/products
Method	POST
Parameter	
Attack	
Evidence	415
Other Info	
Instances	3
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
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://gateway:8080/device
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://gateway:8080/weirdo
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	2
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.