ZAP Scanning Report

Site: https://192.168.0.16:4201

Generated on Thu, 3 Feb 2022 11:42:55

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	3
Low	4
Informational	1

Alerts

Name	Risk Level	Number of Instances
CSP: Wildcard Directive	Medium	2
Cross-Domain Misconfiguration	Medium	9
Missing Anti-clickjacking Header	Medium	1
Incomplete or No Cache-control Header Set	Low	1
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	9
Timestamp Disclosure - Unix	Low	7
X-Content-Type-Options Header Missing	Low	7
Information Disclosure - Suspicious Comments	Informational	3

Alert Detail

Medium	CSP: Wildcard Directive
Description	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: frame-ancestors, form-action The directive(s): frame-ancestors, form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
URL	https://192.168.0.16:4201/robots.txt
Method	GET
Attack	
Evidence	default-src 'none'
URL	https://192.168.0.16:4201/sitemap.xml
Method	GET
Attack	
Evidence	default-src 'none'

Instances	2
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy http://content-security-policy.com/ https://github.com/shapesecurity/salvation https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

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://192.168.0.16:4201/
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
URL	https://192.168.0.16:4201/favicon.ico
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
URL	https://192.168.0.16:4201/main.js
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
URL	https://192.168.0.16:4201/polyfills.js
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
URL	https://192.168.0.16:4201/robots.txt
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
URL	https://192.168.0.16:4201/runtime.js
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
URL	https://192.168.0.16:4201/sitemap.xml
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *

URL	https://192.168.0.16:4201/styles.css
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
URL	https://192.168.0.16:4201/styles.js
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
Instances	9
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	**Missing Anti-clickjacking Header**
Description	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
URL	https://192.168.0.16:4201/
Method	GET
Attack	
Evidence	
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	**Incomplete or No Cache-control Header Set**
Description	The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content.
URL	https://192.168.0.16:4201/
Method	GET
Attack	
Evidence	
Instances	1

Solution	Whenever possible ensure the cache-control HTTP header is set with no-cache, no-store, must-revalidate.
Reference	https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#web-content-caching https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control
CWE Id	525
WASC Id	13
Plugin Id	10015

Low	**Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)**
Description	The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.
URL	https://192.168.0.16:4201/
Method	GET
Attack	
Evidence	X-Powered-By: Express
URL	https://192.168.0.16:4201/favicon.ico
Method	GET
Attack	
Evidence	X-Powered-By: Express
URL	https://192.168.0.16:4201/main.js
Method	GET
Attack	
Evidence	X-Powered-By: Express
URL	https://192.168.0.16:4201/polyfills.js
Method	GET
Attack	
Evidence	X-Powered-By: Express
URL	https://192.168.0.16:4201/robots.txt
Method	GET
Attack	
Evidence	X-Powered-By: Express
URL	https://192.168.0.16:4201/runtime.js
Method	GET
Attack	
Evidence	X-Powered-By: Express
URL	https://192.168.0.16:4201/sitemap.xml
Method	GET
Attack	
Evidence	X-Powered-By: Express
URL	https://192.168.0.16:4201/styles.css
Method	GET

Attack	
Evidence	X-Powered-By: Express
URL	https://192.168.0.16:4201/styles.js
Method	GET
Attack	
Evidence	X-Powered-By: Express
Instances	9
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.
Reference	http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx http://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html
CWE Id	200
WASC Id	13
Plugin Id	10037

Low	**Timestamp Disclosure - Unix**
Description	A timestamp was disclosed by the application/web server - Unix
URL	https://192.168.0.16:4201/main.js
Method	GET
Attack	
Evidence	2147483647
URL	https://192.168.0.16:4201/polyfills.js
Method	GET
Attack	
Evidence	2147483647
URL	https://192.168.0.16:4201/styles.css
Method	GET
Attack	
Evidence	00000005
URL	https://192.168.0.16:4201/styles.css
Method	GET
Attack	
Evidence	00000024
URL	https://192.168.0.16:4201/styles.css
Method	GET
Attack	
Evidence	00000042
URL	https://192.168.0.16:4201/styles.css
Method	GET
Attack	
Evidence	00000061
URL	https://192.168.0.16:4201/styles.js

Method	GET
Attack	
Evidence	2147483647
Instances	7
Solution	Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.
Reference	http://projects.webappsec.org/w/page/13246936/Information%20Leakage
CWE Id	200
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	https://192.168.0.16:4201/
Method	GET
Attack	
Evidence	
URL	https://192.168.0.16:4201/favicon.ico
Method	GET
Attack	
Evidence	
URL	https://192.168.0.16:4201/main.js
Method	GET
Attack	
Evidence	
URL	https://192.168.0.16:4201/polyfills.js
Method	GET
Attack	
Evidence	
URL	https://192.168.0.16:4201/runtime.js
Method	GET
Attack	
Evidence	
URL	https://192.168.0.16:4201/styles.css
Method	GET
Attack	
Evidence	
URL	https://192.168.0.16:4201/styles.js
Method	GET

Attack	
Evidence	
Instances	7
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	http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx 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. Note: Matches made within script blocks or files are against the entire content not only comments.
URL	https://192.168.0.16:4201/main.js
Method	GET
Attack	
Evidence	query
URL	https://192.168.0.16:4201/polyfills.js
Method	GET
Attack	
Evidence	query
URL	https://192.168.0.16:4201/styles.js
Method	GET
Attack	
Evidence	query
Instances	3
Solution	Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.
Reference	
CWE Id	200
WASC Id	13
Plugin Id	10027