

ZAP Scanning Report

Summary of Alerts

Generated on Mon, 21 Feb 2022 22:19:43

Risk Level	Number of Alerts
High	0
Medium	2
Low	3
Informational	2

Alerts

Name	Risk Level	Number of Instances
Vulnerable JS Library	Medium	1
X-Frame-Options Header Not Set	Medium	2
Absence of Anti-CSRF Tokens	Low	2
Cookie No HttpOnly Flag	Low	3
X-Content-Type-Options Header Missing	Low	7
Information Disclosure - Suspicious Comments	Informational	1
Timestamp Disclosure - Unix	Informational	15

Alert Detail

Medium (Medium)	Vulnerable JS Library
-----------------	-----------------------

Description	The identified library jquery, version 3.4.1 is vulnerable.
URL	http://10.71.97.90/assets/js/app.min.js
Method	GET
Evidence	{jquery:"3.4.1"
Instances	1
Solution	Please upgrade to the latest version of jquery.
Other information	CVE-2020-11023 CVE-2020-11022
Reference	https://blog.jquery.com/2020/04/10/jquery-3-5-0-released/
CWE Id	829
Source ID	3
Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://10.71.97.90/
Method	GET
Parameter	X-Frame-Options
URL	http://10.71.97.90
Method	GET
Parameter	X-Frame-Options
Instances	2
Solution	Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (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
Source ID	3
Low (Medium)	Absence of Anti-CSRF Tokens
Description	<p>No Anti-CSRF tokens were found in a HTML submission form.</p> <p>A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.</p> <p>CSRF attacks are effective in a number of situations, including:</p> <ul style="list-style-type: none"> * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site. <p>CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.</p>
URL	http://10.71.97.90/
Method	GET
Evidence	<form action="http://10.71.97.90/login" method="POST" class="margin-bottom-0">
URL	http://10.71.97.90

Method	GET
Evidence	<form action="http://10.71.97.90/login" method="POST" class="margin-bottom-0">
Instances	2
Solution	<p>Phase: Architecture and Design</p> <p>Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.</p> <p>For example, use anti-CSRF packages such as the OWASP CSRFGuard.</p> <p>Phase: Implementation</p> <p>Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.</p> <p>Phase: Architecture and Design</p> <p>Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).</p> <p>Note that this can be bypassed using XSS.</p> <p>Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.</p> <p>Note that this can be bypassed using XSS.</p> <p>Use the ESAPI Session Management control.</p> <p>This control includes a component for CSRF.</p> <p>Do not use the GET method for any request that triggers a state change.</p>

	<p>Phase: Implementation</p> <p>Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.</p>
Other information	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF] was found in the following HTML form: [Form 1: "_token" "email" "password" "remember"].
Reference	<p>http://projects.webappsec.org/Cross-Site-Request-Forgery</p> <p>http://cwe.mitre.org/data/definitions/352.html</p>
CWE Id	352
WASC Id	9
Source ID	3
Low (Medium)	Cookie No HttpOnly Flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://10.71.97.90/login
Method	POST
Parameter	XSRF-TOKEN
Evidence	Set-Cookie: XSRF-TOKEN
URL	http://10.71.97.90/
Method	GET
Parameter	XSRF-TOKEN
Evidence	Set-Cookie: XSRF-TOKEN
URL	http://10.71.97.90

Method	GET
Parameter	XSRF-TOKEN
Evidence	Set-Cookie: XSRF-TOKEN
Instances	3
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	https://owasp.org/www-community/HttpOnly
CWE Id	1004
WASC Id	13
Source ID	3
Low (Medium)	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://10.71.97.90/assets/css/material/theme/red.min.css
Method	GET
Parameter	X-Content-Type-Options
URL	http://10.71.97.90/
Method	GET
Parameter	X-Content-Type-Options
URL	http://10.71.97.90/assets/js/app.min.js
Method	GET
Parameter	X-Content-Type-Options
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET

Parameter	X-Content-Type-Options
URL	http://10.71.97.90
Method	GET
Parameter	X-Content-Type-Options
URL	http://10.71.97.90/assets/img/logo/logo-admin1.png
Method	GET
Parameter	X-Content-Type-Options
URL	http://10.71.97.90/robots.txt
Method	GET
Parameter	X-Content-Type-Options
Instances	7
Solution	<p>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.</p> <p>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.</p>
Other information	<p>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.</p> <p>At "High" threshold this scan rule will not alert on client or server error responses.</p>
Reference	<p>http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx</p> <p>https://owasp.org/www-community/Security_Headers</p>
CWE Id	693
WASC Id	15
Source ID	3
Informational (Low)	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	http://10.71.97.90/assets/js/app.min.js
Method	GET
Evidence	username
Instances	1
Solution	Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.
Other information	The following pattern was used: \bUSERNAME\b and was detected in the element starting with: "(function(){function t(t,e){function i(){this.constructor=t}for(var n in e)G.call(e,n)&&(t[n]=e[n]);return i.prototype=e.prototy", see evidence field for the suspicious comment/snippet.
Reference	
CWE Id	200
WASC Id	13
Source ID	3
Informational (Low)	Timestamp Disclosure - Unix
Description	A timestamp was disclosed by the application/web server - Unix
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET
Evidence	23333333
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET
Evidence	23666666
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET

Evidence	23555555
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET
Evidence	00000000
URL	http://10.71.97.90/assets/js/app.min.js
Method	GET
Evidence	0123456789
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET
Evidence	23444444
URL	http://10.71.97.90
Method	GET
Evidence	12670055
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET
Evidence	80000000
URL	http://10.71.97.90/
Method	GET
Evidence	12440311
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET
Evidence	23777620
URL	http://10.71.97.90/assets/css/material/app.min.css

Method	GET
Evidence	23009688
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET
Evidence	23777777
URL	http://10.71.97.90/assets/css/material/app.min.css
Method	GET
Evidence	23454545
URL	http://10.71.97.90/
Method	GET
Evidence	12670055
URL	http://10.71.97.90
Method	GET
Evidence	12440311
Instances	15
Solution	Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.
Other information	23333333, which evaluates to: 1970-09-27 21:28:53
Reference	http://projects.webappsec.org/w/page/13246936/Information%20Leakage
CWE Id	200
WASC Id	13
Source ID	3