ZAP Scanning Report

Site: https://qa.contilink.com

Generated on Fri, 23 Sept 2022 11:36:18

Summary of Alerts

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

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	1
Content Security Policy (CSP) Header Not Set	Medium	2
Cookie with SameSite Attribute None	Low	1
Cookie without SameSite Attribute	Low	3
Timestamp Disclosure - Unix	Low	1
Information Disclosure - Sensitive Information in URL	Informational	1
Information Disclosure - Suspicious Comments	Informational	5

Alert Detail

Medium	Absence of Anti-CSRF Tokens
Description	No Anti-CSRF tokens were found in a HTML submission form. 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. CSRF attacks are effective in a number of situations, including: * 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. 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.
URL	https://qa.contilink.com/cas/login?service=https%3A%2F%2Fqa.contilink.com%2Fcas%2Foauth2.0%2FcallbackAuthorize%3Fclient_id%3Dcocgateway%26redirect_uri%3Dhttps%253A%252F%252Fqa.contilink.com%252Flogin%252Foauth2%252Fcode%252Fcocgateway%26response_type%3Dcode%26state%3Dr4TfXTNFrvdiDF9TyfisFUnoWY_brsWLzeOjgHVZAJ4%253D%26nonce%3D7KKBt1lEko6sFBR_x0PkDnZtqegx6Ab1JSEqKZPbZzI%26client_name%3DCasOAuthClient
Method	GET
Attack	
Evidence	<form method="post" name="login" id="login" cssClass="form-signin" commandName="${commandName}" htmlEscape="true" role="form">
Instances	1
Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design 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). Note that this can be bypassed using XSS. 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. Note that this can be bypassed using XSS. Use the ESAPI Session Management control. This control includes a component for CSRF. Do not use the GET method for any request that triggers a state change. Phase: Implementation 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.
Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	10202

Medium	**Content Security Policy (CSP) Header Not Set**
	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

Description	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	https://qa.contilink.com/cas/login?service=https%3A%2F%2Fqa.contilink.com%2Fcas%2Foauth2.0%2FcallbackAuthorize%3Fclient_id%3Dcocgateway%26redirect_uri%3Dhttps%253A%252F%252Fqa.contilink.com%252Flogin%252Foauth2%252Fcode%252Fcocgateway%26response_type%3Dcode%26state%3Dr4TfXTNFrvdiDF9TyfisFUnoWY_brsWLzeOjgHVZAJ4%253D%26nonce%3D7KKBt1lEko6sFBR_x0PkDnZtqegx6Ab1JSEqKZPbZzI%26client_name%3DCasOAuthClient
Method	GET
Attack	
Evidence	
URL	https://qa.contilink.com/login?error
Method	GET
Attack	
Evidence	
Instances	2
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header, to achieve optimal browser support: "Content-Security-Policy" for Chrome 25+, Firefox 23+ and Safari 7+, "X-Content-Security-Policy" for Firefox 4.0+ and Internet Explorer 10+, and "X-WebKit-CSP" for Chrome 14+ and Safari 6+.
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 http://www.w3.org/TR/CSP/ http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html http://www.html5rocks.com/en/tutorials/security/content-security-policy/ http://caniuse.com/#feat=contentsecuritypolicy http://content-security-policy.com/
CWE Id	693
WASC Id	15
Plugin Id	10038

Low	**Cookie with SameSite Attribute None**
Description	A cookie has been set with its SameSite attribute set to "none", which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
URL	https://qa.contilink.com/cas/login?service=https%3A%2F%2Fqa.contilink.com%2Fcas%2Foauth2.0%2FcallbackAuthorize%3Fclient_id%3Dcocgateway%26redirect_uri%3Dhttps%253A%252F%252Fqa.contilink.com%252Flogin%252Foauth2%252Fcode%252Fcocgateway%26response_type%3Dcode%26state%3Dr4TfXTNFrvdiDF9TyfisFUnoWY_brsWLzeOjgHVZAJ4%253D%26nonce%3D7KKBt1lEko6sFBR_x0PkDnZtqegx6Ab1JSEqKZPbZzI%26client_name%3DCasOAuthClient
Method	POST
Attack	
Evidence	Set-Cookie: TGC
Instances	1
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site

CWE Id	1275
WASC Id	13
Plugin Id	10054

Low	Cookie without SameSite Attribute
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
URL	https://qa.contilink.com/cas/js/conti/login.js
Method	GET
Attack	
Evidence	Set-Cookie: org.springframework.web.servlet.i18n.CookieLocaleResolver.LOCALE
URL	https://qa.contilink.com/cas/login?service=https%3A%2F%2Fqa.contilink.com%2Fcas%2Foauth2.0%2FcallbackAuthorize%3Fclient_id%3Dcocgateway%26redirect_uri%3Dhttps%253A%252F%252Fqa.contilink.com%252Flogin%252Foauth2%252Fcode%252Fcocgateway%26response_type%3Dcode%26state%3Dr4TfXTNFrvdiDF9TyfisFUnoWY_brsWLzeOjgHVZAJ4%253D%26nonce%3D7KKBt1lEko6sFBR_x0PkDnZtqegx6Ab1JSEqKZPbZzI%26client_name%3DCasOAuthClient
Method	GET
Attack	
Evidence	Set-Cookie: org.springframework.web.servlet.i18n.CookieLocaleResolver.LOCALE
URL	https://qa.contilink.com/cas/login?service=https%3A%2F%2Fqa.contilink.com%2Fcas%2Foauth2.0%2FcallbackAuthorize%3Fclient_id%3Dcocgateway%26redirect_uri%3Dhttps%253A%252F%252Fqa.contilink.com%252Flogin%252Foauth2%252Fcode%252Fcocgateway%26response_type%3Dcode%26state%3Dr4TfXTNFrvdiDF9TyfisFUnoWY_brsWLzeOjgHVZAJ4%253D%26nonce%3D7KKBt1lEko6sFBR_x0PkDnZtqegx6Ab1JSEqKZPbZzI%26client_name%3DCasOAuthClient
Method	POST
Attack	
Evidence	Set-Cookie: org.springframework.web.servlet.i18n.CookieLocaleResolver.LOCALE
Instances	3
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	1275
WASC Id	13
Plugin Id	10054

Low	Timestamp Disclosure - Unix
Description	A timestamp was disclosed by the application/web server - Unix
URL	https://qa.contilink.com/cas/js/vegas/2.5.1/vegas.min.css
Method	GET
Attack	
Evidence	41335068
Instances	1
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

Informational	Information Disclosure - Sensitive Information in URL
Description	The request appeared to contain sensitive information leaked in the URL. This can violate PCI a organizational compliance policies. You can configure the list of strings for this check to add or r values specific to your environment.
URL	https://qa.contilink.com/cas/oauth2.0/callbackAuthorize?client_id=cocgateway&redirect_uri=http 2Fqa.contilink.com%2Flogin%2Foauth2%2Fcode%2Fcocgateway&response_type=code&state=r4TfXTNFrvdiDF9TyfisFUnoWY_brsWLzeOjgHVZ/ 3D&nonce=7KKBt1lEko6sFBR_x0PkDnZtqegx6Ab1JSEqKZPbZzI&client_name=CasOAuthClie 2315-MYPEq63PHjV--vZwLn2qAzf7IDA-4302d5db2d2a
Method	GET
Attack	
Evidence	ticket
Instances	1
Solution	Do not pass sensitive information in URIs.
Reference	
CWE Id	200
WASC Id	13
Plugin Id	10024

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://qa.contilink.com/cas/js/conti/global.js
Method	GET
Attack	
Evidence	where
URL	https://qa.contilink.com/cas/js/conti/login.js
Method	GET
Attack	
Evidence	username
URL	https://qa.contilink.com/cas/js/jquery/jquery-3.6.1.min.js
Method	GET
Attack	
Evidence	username
URL	https://qa.contilink.com/cas/login?service=https%3A%2F%2Fqa.contilink.com%2Fcas%2Foauth2.0%2FcallbackAuthorize%3Fclient_id%3Dcocgateway%26redirect_uri%3Dhttps%253A%252F%252Fqa.contilink.com%252Flogin%252Foauth2%252Fcode%252Fcocgateway%26response_type%3Dcode%26state%3Dr4TfXTNFrvdiDF9TyfisFUnoWY_brsWLzeOjgHVZAJ4%253D%26nonce%3D7KKBt1lEko6sFBR_x0PkDnZtqegx6Ab1JSEqKZPbZzI%26client_name%3DCasOAuthClient
Method	GET
Attack	
Evidence	from

URL	https://qa.contilink.com/cas/login?service=https%3A%2F%2Fqa.contilink.com%2Fcas%2Foauth2.0%2FcallbackAuthorize%3Fclient_id%3Dcocgateway%26redirect_uri%3Dhttps%253A%252F%252Fqa.contilink.com%252Flogin%252Foauth2%252Fcode%252Fcocgateway%26response_type%3Dcode%26state%3Dr4TfXTNFrvdiDF9TyfisFUnoWY_brsWLzeOjgHVZAJ4%253D%26nonce%3D7KKBt1lEko6sFBR_x0PkDnZtqegx6Ab1JSEqKZPbZzI%26client_name%3DCasOAuthClient
Method	GET
Attack	
Evidence	username
Instances	5
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