ZAP by Checkmarx Scanning Report

Site: http://project.test

Generated on Tue, 6 May 2025 14:27:42

ZAP Version: 2.16.1

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	4
Low	5
Informational	2

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	2
Content Security Policy (CSP) Header Not Set	Medium	5
Hidden File Found	Medium	5
Missing Anti-clickjacking Header	Medium	5
Cookie No HttpOnly Flag	Low	3
Cookie without SameSite Attribute	Low	3
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	5
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	5
X-Content-Type-Options Header Missing	Low	5
Authentication Request Identified	Informational	1
Session Management Response Identified	Informational	5

Alert Detail

Medium	Absence of Anti-CSRF Tokens
	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

Description	/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	http://project.test/login.php
Method	GET
Attack	
Evidence	<form method="POST">
Other Info	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: "password" "username"].
URL	http://project.test/login.php
Method	POST
Attack	
Evidence	<form method="POST">
Other Info	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: "password" "username"].
Instances	2
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	https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html https://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	10202

Medium	Content Security Policy (CSP) Header Not Set
Description	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 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	http://project.test/index.php
Method	GET
Attack	
Evidence	
Other Info	
URL	http://project.test/login.php
Method	GET
Attack	
Evidence	
Other Info	
URL	http://project.test/robots.txt
Method	GET
Attack	
Evidence	
Other Info	
URL	http://project.test/sitemap.xml
Method	GET
Attack	
Evidence	
Other Info	

URL	http://project.test/login.php
Method	POST
Attack	
Evidence	
Other Info	
Instances	5
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
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 https://www.w3.org/TR/CSP/ https://w3c.github.io/webappsec-csp/ https://web.dev/articles/csp https://caniuse.com/#feat=contentsecuritypolicy https://content-security-policy.com/
CWE Id	693
WASC Id	15
Plugin Id	10038

Medium	**Hidden File Found**
Description	A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.
URL	http://project.test/._darcs
Method	GET
Attack	
Evidence	HTTP/1.1 200 OK
Other Info	
URL	http://project.test/.bzr
Method	GET
Attack	
Evidence	HTTP/1.1 200 OK
Other Info	
URL	http://project.test/.hg
Method	GET
Attack	
Evidence	HTTP/1.1 200 OK
Other Info	
URL	http://project.test/BitKeeper
Method	GET
Attack	
Evidence	HTTP/1.1 200 OK

Other Info	
URL	http://project.test/.git/config
Method	GET
Attack	
Evidence	HTTP/1.1 200 OK
Other Info	git_dir
Instances	5
Solution	Consider whether or not the component is actually required in production, if it isn't then disable it. If it is then ensure access to it requires appropriate authentication and authorization, or limit exposure to internal systems or specific source IPs, etc.
Reference	https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html https://git-scm.com/docs/git-config
CWE Id	538
WASC Id	13
Plugin Id	40035

Medium	Missing Anti-clickjacking Header
Description	The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
URL	http://project.test/index.php
Method	GET
Attack	
Evidence	
Other Info	
URL	http://project.test/login.php
Method	GET
Attack	
Evidence	
Other Info	
URL	http://project.test/robots.txt
Method	GET
Attack	
Evidence	
Other Info	
URL	http://project.test/sitemap.xml
Method	GET
Attack	
Evidence	
Other Info	

URL	http://project.test/login.php
Method	POST
Attack	
Evidence	
Other Info	
Instances	5
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	**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://project.test/login.php
Method	GET
Attack	
Evidence	Set-Cookie: PHPSESSID
Other Info	
URL	http://project.test/robots.txt
Method	GET
Attack	
Evidence	Set-Cookie: PHPSESSID
Other Info	
URL	http://project.test/sitemap.xml
Method	GET
Attack	
Evidence	Set-Cookie: PHPSESSID
Other Info	
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
Plugin Id	10010

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	http://project.test/login.php
Method	GET
Attack	
Evidence	Set-Cookie: PHPSESSID
Other Info	
URL	http://project.test/robots.txt
Method	GET
Attack	
Evidence	Set-Cookie: PHPSESSID
Other Info	
URL	http://project.test/sitemap.xml
Method	GET
Attack	
Evidence	Set-Cookie: PHPSESSID
Other Info	
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	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	http://project.test/index.php
Method	GET
Attack	
Evidence	X-Powered-By: PHP/8.4.4
Other Info	
URL	http://project.test/login.php
Method	GET
Attack	
Evidence	X-Powered-By: PHP/8.4.4
Other	

Info	
URL	http://project.test/robots.txt
Method	GET
Attack	
Evidence	X-Powered-By: PHP/8.4.4
Other Info	
URL	http://project.test/sitemap.xml
Method	GET
Attack	
Evidence	X-Powered-By: PHP/8.4.4
Other Info	
URL	http://project.test/login.php
Method	POST
Attack	
Evidence	X-Powered-By: PHP/8.4.4
Other Info	
Instances	5
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.
Reference	https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/01-Information_Gathering/08-Fingerprint_Web_Application_Framework https://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html
CWE Id	497
WASC Id	13
Plugin Id	10037

Low	**Server Leaks Version Information via "Server" HTTP Response Header Field**
Description	The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.
URL	http://project.test/index.php
Method	GET
Attack	
Evidence	nginx/1.25.2
Other Info	
URL	http://project.test/login.php
Method	GET
Attack	
Evidence	nginx/1.25.2
Other Info	

URL	http://project.test/robots.txt
Method	GET
Attack	
Evidence	nginx/1.25.2
Other Info	
URL	http://project.test/sitemap.xml
Method	GET
Attack	
Evidence	nginx/1.25.2
Other Info	
URL	http://project.test/login.php
Method	POST
Attack	
Evidence	nginx/1.25.2
Other Info	
Instances	5
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.
Reference	https://httpd.apache.org/docs/current/mod/core.html#servertokens https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff648552(v=pandp.10) https://www.troyhunt.com/shhh-dont-let-your-response-headers/
CWE Id	497
WASC Id	13
Plugin Id	10036

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://project.test/index.php
Method	GET
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.
URL	http://project.test/login.php
Method	GET
Attack	
Evidence	
	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still

Other Info	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.
URL	http://project.test/robots.txt
Method	GET
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.
URL	http://project.test/sitemap.xml
Method	GET
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.
URL	http://project.test/login.php
Method	POST
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	5
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	**Authentication Request Identified**
Description	The given request has been identified as an authentication request. The 'Other Info' field contains a set of key=value lines which identify any relevant fields. If the request is in a context which has an Authentication Method set to "Auto-Detect" then this rule will change the authentication to match the request identified.
URL	http://project.test/login.php
Method	POST
Attack	
Evidence	password
Other	userParam=username userValue=ZAP passwordParam=password referer=http://project.test

Info	/login.php
Instances	1
Solution	This is an informational alert rather than a vulnerability and so there is nothing to fix.
Reference	https://www.zaproxy.org/docs/desktop/addons/authentication-helper/auth-req-id/
CWE Id	
WASC Id	
Plugin Id	10111

Informational	Session Management Response Identified
Description	The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.
URL	http://project.test/login.php
Method	GET
Attack	
Evidence	439a288120d256d076d0d1afa710c1a0
Other Info	cookie:PHPSESSID
URL	http://project.test/login.php
Method	GET
Attack	
Evidence	6f4f549e4026c3514d9bfe1211b55ad8
Other Info	cookie:PHPSESSID
URL	http://project.test/robots.txt
Method	GET
Attack	
Evidence	54bac2b8bc030400193d4a575121a1a4
Other Info	cookie:PHPSESSID
URL	http://project.test/sitemap.xml
Method	GET
Attack	
Evidence	3b5ea924bcc4f1ba883c1b51728b5481
Other Info	cookie:PHPSESSID
URL	http://project.test/sitemap.xml
Method	GET
Attack	
Evidence	3b5ea924bcc4f1ba883c1b51728b5481
Other Info	cookie:PHPSESSID
Instances	5
Solution	This is an informational alert rather than a vulnerability and so there is nothing to fix.

Reference	https://www.zaproxy.org/docs/desktop/addons/authentication-helper/session-mgmt-id
CWE Id	
WASC Id	
Plugin Id	10112