ZAP Scanning Report

Site: http://192.168.244.209

Generated on Sat, 14 Oct 2023 04:21:33

ZAP Version: 2.13.0

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	5
Low	5
Informational	1
False Positives:	0

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	1
Application Error Disclosure	Medium	4
Content Security Policy (CSP) Header Not Set	Medium	7
Directory Browsing	Medium	4
Missing Anti-clickjacking Header	Medium	6
Cookie No HttpOnly Flag	Low	1
Cookie without SameSite Attribute	Low	1
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	4
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	10
X-Content-Type-Options Header Missing	Low	7
Session Management Response Identified	Informational	2

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	http://192.168.244.209/login.php
Method	GET
Parameter	
Attack	
Evidence	<form action="/login.php" method="POST">
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	Application Error Disclosure
Description	This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page.
URL	http://192.168.244.209/documents/
Method	GET
Parameter	
Attack	
Evidence	Parent Directory
URL	http://192.168.244.209/documents/?C=M;O=A
Method	GET
Parameter	
Attack	
Evidence	Parent Directory
URL	http://192.168.244.209/documents/?C=N;O=D
Method	GET
Parameter	
Attack	
Evidence	Parent Directory
URL	http://192.168.244.209/passwords/
Method	GET
Parameter	
Attack	
Evidence	Parent Directory
Instances	4
Solution	Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.
Reference	
CWE Id	200
WASC Id	13
Plugin Id	90022

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://192.168.244.209/admin/
Method	GET

Parameter	
Attack	
Evidence	
URL	http://192.168.244.209/documents/
Method	GET
Parameter	
Attack	
Evidence	
URL	http://192.168.244.209/documents/?C=M;O=A
Method	GET
Parameter	
Attack	
Evidence	
URL	http://192.168.244.209/documents/?C=N;O=D
Method	GET
Parameter	
Attack	
Evidence	
URL	http://192.168.244.209/login.php
Method	GET
Parameter	
Attack	
Evidence	
URL	http://192.168.244.209/passwords/
Method	GET
Parameter	
Attack	
Evidence	
URL	http://192.168.244.209/sitemap.xml
Method	GET
Parameter	
Attack	
Evidence	
Instances	7
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 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

Medium	Directory Browsing
Description	It is possible to view a listing of the directory contents. Directory listings may reveal hidden scripts, include files, backup source files, etc., which can be accessed to reveal sensitive information.
URL	http://192.168.244.209/documents/
Method	GET
Parameter	
Attack	
Evidence	<title>Index of /documents</title>
URL	http://192.168.244.209/documents/?C=M;O=A
Method	GET
Parameter	
Attack	
Evidence	<title>Index of /documents</title>
URL	http://192.168.244.209/documents/?C=N;O=D
Method	GET
Parameter	
Attack	
Evidence	<title>Index of /documents</title>
URL	http://192.168.244.209/passwords/
Method	GET
Parameter	
Attack	
Evidence	<title>Index of /passwords</title>
Instances	4
Solution	Configure the web server to disable directory browsing.
Reference	https://cwe.mitre.org/data/definitions/548.html
CWE Id	548
WASC Id	16
Plugin Id	10033

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	http://192.168.244.209/admin/
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
	http://192.168.244.209/documents/

URL	
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
URL	http://192.168.244.209/documents/?C=M;O=A
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
URL	http://192.168.244.209/documents/?C=N;O=D
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
URL	http://192.168.244.209/login.php
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
URL	http://192.168.244.209/passwords/
Method	GET
Parameter	x-frame-options
Attack	
Evidence	
Instances	6
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://192.168.244.209/portal.php
Method	GET
Parameter	PHPSESSID

Attack	
Evidence	Set-Cookie: PHPSESSID
Instances	1
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://192.168.244.209/portal.php
Method	GET
Parameter	PHPSESSID
Attack	
Evidence	Set-Cookie: PHPSESSID
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	**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://192.168.244.209/
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: PHP/5.5.9-1ubuntu4.14
URL	http://192.168.244.209/admin/
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: PHP/5.5.9-1ubuntu4.14
URL	http://192.168.244.209/login.php
Method	GET
Parameter	

Attack	
Evidence	X-Powered-By: PHP/5.5.9-1ubuntu4.14
URL	http://192.168.244.209/portal.php
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: PHP/5.5.9-1ubuntu4.14
Instances	4
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	**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://192.168.244.209/
Method	GET
Parameter	
Attack	
Evidence	Apache/2.4.7 (Ubuntu)
URL	http://192.168.244.209/admin/
Method	GET
Parameter	
Attack	
Evidence	Apache/2.4.7 (Ubuntu)
URL	http://192.168.244.209/documents/
Method	GET
Parameter	
Attack	
Evidence	Apache/2.4.7 (Ubuntu)
URL	http://192.168.244.209/documents/?C=M;O=A
Method	GET
Parameter	
Attack	
Evidence	Apache/2.4.7 (Ubuntu)
URL	http://192.168.244.209/documents/?C=N;O=D
Method	GET
Parameter	

Attack	
Evidence	Apache/2.4.7 (Ubuntu)
URL	http://192.168.244.209/login.php
Method	GET
Parameter	
Attack	
Evidence	Apache/2.4.7 (Ubuntu)
URL	http://192.168.244.209/passwords/
Method	GET
Parameter	
Attack	
Evidence	Apache/2.4.7 (Ubuntu)
URL	http://192.168.244.209/portal.php
Method	GET
Parameter	
Attack	
Evidence	Apache/2.4.7 (Ubuntu)
URL	http://192.168.244.209/robots.txt
Method	GET
Parameter	
Attack	
Evidence	Apache/2.4.7 (Ubuntu)
URL	http://192.168.244.209/sitemap.xml
Method	GET
Parameter	
Attack	
Evidence	Apache/2.4.7 (Ubuntu)
Instances	10
Solution	Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.
Reference	http://httpd.apache.org/docs/current/mod/core.html#servertokens http://msdn.microsoft.com/en-us/library/ff648552.aspx#ht_urlscan_007 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	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.
	http://192.168.244.209/admin/

URL	
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
URL	http://192.168.244.209/documents/
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
URL	http://192.168.244.209/documents/?C=M;O=A
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
URL	http://192.168.244.209/documents/?C=N;O=D
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
URL	http://192.168.244.209/login.php
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
URL	http://192.168.244.209/passwords/
Method	GET
Parameter	x-content-type-options
Attack	
Evidence	
URL	http://192.168.244.209/robots.txt
Method	GET
Parameter	x-content-type-options
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	**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://192.168.244.209/portal.php
Method	GET
Parameter	PHPSESSID
Attack	
Evidence	1bsni5qce9qknt8u3p6r6qlva1
URL	http://192.168.244.209/portal.php
Method	GET
Parameter	PHPSESSID
Attack	
Evidence	1bsni5qce9qknt8u3p6r6qlva1
Instances	2
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