ZAP Scanning Report

Site: https://ehr-online.co.uk

Generated on Mon, 20 Jun 2022 20:18:00

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	4
Informational	1
False Positives:	0

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	2
Vulnerable JS Library	Medium	1
Cookie No HttpOnly Flag	Low	2
Cookie Without Secure Flag	Low	2
Cookie without SameSite Attribute	Low	2
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	7
Re-examine Cache-control Directives	Informational	3

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://ehr-online.co.uk/interface/login/login.php

Method	GET
Parameter	
Attack	
Evidence	<form method="POST" action="../main/main_screen.php?auth=login&site=default" target="_top" name="login_form" onsubmit="return imsubmitted();">
URL	https://ehr-online.co.uk/interface/login/login.php?site=default
Method	GET
Parameter	
Attack	
Evidence	<form method="POST" action="../main/main_screen.php?auth=login&site=default" target="_top" name="login_form" onsubmit="return imsubmitted();">
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	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	**Vulnerable JS Library**
Description	The identified library jquery, version 1.4.3 is vulnerable.
URL	https://ehr-online.co.uk/library/js/jquery-1.4.3.min.js
Method	GET
Parameter	
Attack	
Evidence	jquery-1.4.3.min.js
Instances	1

Solution	Please upgrade to the latest version of jquery.
Reference	https://nvd.nist.gov/vuln/detail/CVE-2012-6708 https://github.com/jquery/jquery/issues/2432 http://research.insecurelabs.org/jquery/test/ https://bugs.jquery.com/ticket/9521 http://blog.jquery.com/2016/01/08/jquery-2-2-and-1-12-released/ http://bugs.jquery.com/ticket/11290 https://blog.jquery.com/2019/04/10/jquery-3-4-0-released/ https://nvd.nist.gov/vuln/detail/CVE-2019-11358 https://nvd.nist.gov/vuln/detail/CVE-2015-9251 https://github.com/jquery/jquery/commit/753d591aea698e57d6db58c9f722cd0808619b1b https://blog.jquery.com/2020/04/10/jquery-3-5-0-released/ https://nvd.nist.gov/vuln/detail/CVE-2011-4969
CWE Id	829
WASC Id	
Plugin Id	10003

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	https://ehr-online.co.uk/interface/login/login.php
Method	GET
Parameter	LibreHealthEHR
Attack	
Evidence	Set-Cookie: LibreHealthEHR
URL	https://ehr-online.co.uk/interface/main/main_screen.php?auth=login&site=default
Method	POST
Parameter	LibreHealthEHR
Attack	
Evidence	Set-Cookie: LibreHealthEHR
Instances	2
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 Secure Flag**
Description	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
URL	https://ehr-online.co.uk/interface/login/login.php
Method	GET
Parameter	LibreHealthEHR
Attack	
Evidence	Set-Cookie: LibreHealthEHR
URL	https://ehr-online.co.uk/interface/main/main_screen.php?auth=login&site=default
Method	POST
Parameter	LibreHealthEHR
Attack	

Evidence	Set-Cookie: LibreHealthEHR
Instances	2
Solution	Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.
Reference	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
CWE Id	614
WASC Id	13
Plugin Id	10011

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://ehr-online.co.uk/interface/login/login.php
Method	GET
Parameter	LibreHealthEHR
Attack	
Evidence	Set-Cookie: LibreHealthEHR
URL	https://ehr-online.co.uk/interface/main/main_screen.php?auth=login&site=default
Method	POST
Parameter	LibreHealthEHR
Attack	
Evidence	Set-Cookie: LibreHealthEHR
Instances	2
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	https://ehr-online.co.uk
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: PHP/7.4.29
URL	https://ehr-online.co.uk/interface/
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: PHP/7.4.29
URL	https://ehr-online.co.uk/interface/login/login.php
Method	GET

Parameter	
Attack	
Evidence	X-Powered-By: PHP/7.4.29
URL	https://ehr-online.co.uk/interface/login/login.php?site=default
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: PHP/7.4.29
URL	https://ehr-online.co.uk/interface/login_screen.php?error=1&site
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: PHP/7.4.29
URL	https://ehr-online.co.uk/interface/login_screen.php?error=1&site=
Method	GET
Parameter	
Attack	
Evidence	X-Powered-By: PHP/7.4.29
URL	https://ehr-online.co.uk/interface/main/main_screen.php?auth=login&site=default
Method	POST
Parameter	
Attack	
Evidence	X-Powered-By: PHP/7.4.29
Instances	7
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

Informational	**Re-examine Cache-control Directives**
Description	The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like css, js, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.
URL	https://ehr-online.co.uk/acknowledge_license_cert.html
Method	GET
Parameter	Cache-Control
Attack	
Evidence	max-age=3600, must-revalidate
URL	https://ehr-online.co.uk/mpl_license.txt
Method	GET
Parameter	Cache-Control
Attack	
Evidence	max-age=3600, public, must-revalidate

URL	https://ehr-online.co.uk/robots.txt
Method	GET
Parameter	Cache-Control
Attack	
Evidence	max-age=3600, public, must-revalidate
Instances	3
Solution	Whenever possible ensure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate". If an asset should be cached consider setting the directives "public, max-age, immutable".
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