ZAP by Checkmarx Scanning Report

Report 2

Sites: https://www.google.com https://unpkg.com http://localhost:4000 http://localhost:3000

Generated on Tue, 2 Dec 2025 09:35:24

ZAP Version: 2.16.1

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	4
Low	3
Informational	3

Alerts

Name	Risk Level	Number of Instances
CSP: Failure to Define Directive with No Fallback	Medium	1
Content Security Policy (CSP) Header Not Set	Medium	4
Cross-Domain Misconfiguration	Medium	4
Missing Anti-clickjacking Header	Medium	4
Cross-Domain JavaScript Source File Inclusion	Low	8
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	3
X-Content-Type-Options Header Missing	Low	6
Information Disclosure - Suspicious Comments	Informational	2
Modern Web Application	Informational	4
Retrieved from Cache	Informational	6

Alert Detail

Medium	CSP: Failure to Define Directive with No Fallback
Description	The Content Security Policy fails to define one of the directives that has no fallback. Missing /excluding them is the same as allowing anything.
URL	http://localhost:4000/

Method	GET
Attack	
Evidence	default-src 'none'
Other Info	The directive(s): frame-ancestors, form-action is/are among the directives that do not fallback to default-src.
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	https://www.w3.org/TR/CSP/ https://caniuse.com/#search=content+security+policy https://content-security-policy.com/ https://github.com/HtmlUnit/htmlunit-csp https://web.dev/articles/csp#resource-options
CWE Id	693
WASC Id	15
Plugin Id	10055

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://localhost:3000/
Method	GET
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/favicon.ico
Method	GET
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/robots.txt
Method	GET
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/sitemap.xml
Method	GET
Attack	
Evidence	
Other	

Info	
Instances	4
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/HTTP/Guides/CSP 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	Cross-Domain Misconfiguration
Description	Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server.
URL	https://unpkg.com/react-dom@18.3.1/umd/react-dom.development.js
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	https://unpkg.com/react-dom@18/umd/react-dom.development.js
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	https://unpkg.com/react@18.3.1/umd/react.development.js
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
URL	https://unpkg.com/react@18/umd/react.development.js
Method	GET
Attack	

Evidence	Access-Control-Allow-Origin: *
Other Info	The CORS misconfiguration on the web server permits cross-domain read requests from arbitrary third party domains, using unauthenticated APIs on this domain. Web browser implementations do not permit arbitrary third parties to read the response from authenticated APIs, however. This reduces the risk somewhat. This misconfiguration could be used by an attacker to access data that is available in an unauthenticated manner, but which uses some other form of security, such as IP address white-listing.
Instances	4
Solution	Ensure that sensitive data is not available in an unauthenticated manner (using IP address white-listing, for instance). Configure the "Access-Control-Allow-Origin" HTTP header to a more restrictive set of domains, or remove all CORS headers entirely, to allow the web browser to enforce the Same Origin Policy (SOP) in a more restrictive manner.
Reference	https://vulncat.fortify.com/en/detail?category=HTML5&subcategory=Overly%20Permissive%20CORS%20Policy
CWE Id	264
WASC Id	14
Plugin Id	10098

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://localhost:3000/
Method	GET
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/favicon.ico
Method	GET
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/robots.txt
Method	GET
Attack	
Evidence	
Other Info	
URL	http://localhost:3000/sitemap.xml
Method	GET
Attack	
Evidence	
Other Info	
Instances	4

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/Reference/Headers/X-Frame-Options
CWE Id	1021
WASC Id	15
Plugin Id	10020

Low	**Cross-Domain JavaScript Source File Inclusion**
Description	The page includes one or more script files from a third-party domain.
URL	http://localhost:3000/
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
Other Info	
URL	http://localhost:3000/
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react@18/umd/react.development.js"></script>
Other Info	
URL	http://localhost:3000/favicon.ico
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
Other Info	
URL	http://localhost:3000/favicon.ico
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react@18/umd/react.development.js"></script>
Other Info	
URL	http://localhost:3000/robots.txt
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
Other Info	
URL	http://localhost:3000/robots.txt
Method	GET

Attack	
Evidence	<script src="https://unpkg.com/react@18/umd/react.development.js"></script>
Other Info	
URL	http://localhost:3000/sitemap.xml
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
Other Info	
URL	http://localhost:3000/sitemap.xml
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react@18/umd/react.development.js"></script>
Other Info	
Instances	8
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.
Reference	
CWE Id	829
WASC Id	15
Plugin Id	10017

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://localhost:4000/
Method	GET
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:4000/api/me
Method	GET
Attack	
Evidence	X-Powered-By: Express
Other Info	
URL	http://localhost:4000/api/me
Method	OPTIONS
Attack	
Evidence	X-Powered-By: Express
Other	

Info	
Instances	3
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/shhh-dont-let-your-response-headers/
CWE Id	497
WASC Id	13
Plugin Id	10037

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://localhost:3000/
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://localhost:3000/app.js
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://localhost:3000/favicon.ico
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://localhost:3000/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://localhost:3000/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://localhost:4000/api/me
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.
Instances	6
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	Information Disclosure - Suspicious Comments
Description	The response appears to contain suspicious comments which may help an attacker.
URL	https://unpkg.com/react-dom@18.3.1/umd/react-dom.development.js
Method	GET
Attack	
Evidence	TODO
Other Info	The following pattern was used: \bTODO\b and was detected in likely comment: "// TODO: Need to review this code one more time before landing", see evidence field for the suspicious comment/snippet.
URL	https://unpkg.com/react@18.3.1/umd/react.development.js
Method	GET
Attack	
Evidence	bugs
Other Info	The following pattern was used: \bBUGS\b and was detected in likely comment: "// No known bugs, but needs performance testing", see evidence field for the suspicious comment /snippet.
Instances	2
Solution	Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.

Reference	
CWE Id	615
WASC Id	13
Plugin Id	10027

Informational	Modern Web Application
Description	The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.
URL	http://localhost:3000/
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react@18/umd/react.development.js"></script>
Other Info	No links have been found while there are scripts, which is an indication that this is a modern web application.
URL	http://localhost:3000/favicon.ico
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react@18/umd/react.development.js"></script>
Other Info	No links have been found while there are scripts, which is an indication that this is a modern web application.
URL	http://localhost:3000/robots.txt
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react@18/umd/react.development.js"></script>
Other Info	No links have been found while there are scripts, which is an indication that this is a modern web application.
URL	http://localhost:3000/sitemap.xml
Method	GET
Attack	
Evidence	<script src="https://unpkg.com/react@18/umd/react.development.js"></script>
Other Info	No links have been found while there are scripts, which is an indication that this is a modern web application.
Instances	4
Solution	This is an informational alert and so no changes are required.
Reference	
CWE Id	
WASC Id	
Plugin Id	10109

Informational	Retrieved from Cache
Description	The content was retrieved from a shared cache. If the response data is sensitive, personal or user-specific, this may result in sensitive information being leaked. In some cases, this may even result in a user gaining complete control of the session of another user, depending on the configuration of the caching components in use in their environment. This is primarily an issue where caching servers such as "proxy" caches are configured on the local network. This configuration is typically found in corporate or educational environments, for instance.

URL	https://unpkg.com/react-dom@18.3.1/umd/react-dom.development.js
Method	GET
Attack	
Evidence	Age: 286965
Other Info	The presence of the 'Age' header indicates that a HTTP/1.1 compliant caching server is in use.
URL	https://unpkg.com/react-dom@18.3.1/umd/react-dom.development.js
Method	GET
Attack	
Evidence	Age: 286968
Other Info	The presence of the 'Age' header indicates that a HTTP/1.1 compliant caching server is in use.
URL	https://unpkg.com/react-dom@18.3.1/umd/react-dom.development.js
Method	GET
Attack	
Evidence	Age: 286969
Other Info	The presence of the 'Age' header indicates that a HTTP/1.1 compliant caching server is in use.
URL	https://unpkg.com/react@18.3.1/umd/react.development.js
Method	GET
Attack	
Evidence	Age: 45018
Other Info	The presence of the 'Age' header indicates that a HTTP/1.1 compliant caching server is in use.
URL	https://unpkg.com/react@18.3.1/umd/react.development.js
Method	GET
Attack	
Evidence	Age: 45021
Other Info	The presence of the 'Age' header indicates that a HTTP/1.1 compliant caching server is in use.
URL	https://unpkg.com/react@18.3.1/umd/react.development.js
Method	GET
Attack	
Evidence	Age: 45022
Other Info	The presence of the 'Age' header indicates that a HTTP/1.1 compliant caching server is in use.
Instances	6
Solution	Validate that the response does not contain sensitive, personal or user-specific information. If it does, consider the use of the following HTTP response headers, to limit, or prevent the content being stored and retrieved from the cache by another user: Cache-Control: no-cache, no-store, must-revalidate, private Pragma: no-cache Expires: 0

	This configuration directs both HTTP 1.0 and HTTP 1.1 compliant caching servers to not store the response, and to not retrieve the response (without validation) from the cache, in response to a similar request.
Reference	https://datatracker.ietf.org/doc/html/rfc7234 https://datatracker.ietf.org/doc/html/rfc7231 https://www.rfc-editor.org/rfc/rfc9110.html
CWE Id	525
WASC Id	
Plugin Id	10050