ZAP Scanning Report

Site: https://sec.cse.csusb.edu

Generated on Thu, 21 Nov 2024 05:49:50

ZAP Version: 2.15.0

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	6
Informational	5
False Positives:	0

Alerts

Name	Risk Level	Number of Instances
Content Security Policy (CSP) Header Not Set	Medium	4
Missing Anti-clickjacking Header	Medium	2
Permissions Policy Header Not Set	Low	4
Permissions Policy Header Not Set	Low	4
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	6
X-Content-Type-Options Header Missing	Low	3
Modern Web Application	Informational	1
Retrieved from Cache	Informational	2
Storable and Cacheable Content	Informational	4
Storable but Not Cacheable Content	Informational	1
Sec-Fetch-Dest	Informational	10

Alert Detail

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: https://sec.cse.csusb.edu/
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/fonts-world/
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/styles.txt
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/sitemap.xml
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/robots.txt
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/favicon.ico
  - Method: GET
  - Parameter:
  - 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/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 — 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: https://sec.cse.csusb.edu/
  - Method: GET
  - Parameter: x-frame-options
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/fonts-world/
  - Method: GET
  - Parameter: x-frame-options
  - Attack:
  - Evidence:
  - Other Info:

Instances: 2

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 — Permissions Policy Header Not Set

Description: Permissions Policy Header is an added layer of security that helps to restrict from unauthorized access or usage of browser/client features by web resources. This policy ensures the user privacy by limiting or specifying the features of the browsers can be used by the web resources. Permissions Policy provides a set of standard HTTP headers that allow website owners to limit which features of browsers can be used by the page such as camera, microphone, location, full screen etc.

- URL: https://sec.cse.csusb.edu/
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/fonts-world/
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/styles.txt
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/sitemap.xml
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/robots.txt
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:
- URL: https://sec.cse.csusb.edu/favicon.ico
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence:
  - Other Info:

Instances: 6

Solution: Ensure that your web server, application server, load balancer, etc. is configured to set the Permissions-Policy header.

Reference: https://developer.mozilla.org/en-US/docs/Web/Security/Feature-Policy
https://developer.chrome.com/blog/feature-policy/
https://scotthelme.co.uk/a-new-security-header-feature-policy/
https://w3c.github.io/webappsec-feature-policy/
https://www.smashingmagazine.com/2018/12/feature-policy/

CWE Id: 693
WASC Id: 15
Plugin Id: 10063

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: https://sec.cse.csusb.edu/
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence: Apache/2.4.51 (Debian)
  - Other Info:
- URL: https://sec.cse.csusb.edu/fonts-world/
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence: Apache/2.4.51 (Debian)
  - Other Info:
- URL: https://sec.cse.csusb.edu/styles.txt
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence: Apache/2.4.51 (Debian)
  - Other Info:
- URL: https://sec.cse.csusb.edu/sitemap.xml
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence: Apache/2.4.51 (Debian)
  - Other Info:
- URL: https://sec.cse.csusb.edu/robots.txt
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence: Apache/2.4.51 (Debian)
  - Other Info:
- URL: https://sec.cse.csusb.edu/favicon.ico
  - Method: GET
  - Parameter:
  - Attack:
  - Evidence: Apache/2.4.51 (Debian)
  - Other Info:

Instances: 6

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
https://learn.microsoft.com/en-us/iis/manage/provisioning-and-managing-iis/changing-server-name
https://cheatsheetseries.owasp.org/cheatsheets/Fingerprinting_Cheat_Sheet.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 UAs (old Firefox, old Opera, old Safari, early versions of Firefox with use the declared content type if one is sent, rather than performing MIME-sniffing.

- URL: https://sec.cse.csusb.edu/
  - Method: GET
  - Parameter: x-content-type-options
  - 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 on alerts or some error responses.
- URL: https://sec.cse.csusb.edu/sitemap.xml
  - Method: GET
  - Parameter: x-content-type-options
  - 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 on alerts or some error responses.

Instances: 3

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/microsoft-edge/devtools-guide-chromium/network/content-type-mismatch-issues
https://owasp.org/www-community/Vulnerabilities/Unchecked_Content_Type_Header

CWE Id: 693
WASC Id: 15
Plugin Id: 10021

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: https://sec.cse.csusb.edu/sitemap.xml
  - Method: GET
  - Parameter: content-type
  - Attack: text/plain;charset=utf-8;charset=iso-8859-1;charset=iso-8859-2;charset=iso-8859-3;charset=iso-8859-4;charset=iso-8859-5;charset=iso-8859-6;charset=iso-8859-7;charset=iso-8859-8;charset=iso-8859-9;charset=iso-8859-10;charset=iso-8859-13;charset=iso-8859-14;charset=iso-8859-15;charset=iso-8859-16;charset=iso-2022-jp;charset=iso-2022-jp-2;charset=iso-2022-jp-3;charset=iso-2022-kr;charset=utf-7;charset=utf-8;charset=utf-16;charset=utf-16be;charset=utf-16le;charset=utf-32;charset=utf-32be;charset=utf-32le;charset=euc-jp;charset=euc-kr;charset=euc-tw;charset=gb2312;charset=gb18030;charset=hz-gb-2312;charset=gb18030;charset=big5;charset=big5-hkscs;charset=shift_jis