ZAP Scanning Report

Site: https://himanshu240601.github.io

Generated on Tue, 9 Apr 2024 04:31:14

ZAP Version: 2.14.0

ZAP is supported by the Crash Override Open Source Fellowship

Summary of Alerts

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

Alerts

Name	Risk Level	Number of Instances
CSP: Wildcard Directive	Medium	6
CSP: style-src unsafe-inline	Medium	3
Content Security Policy (CSP) Header Not Set	Medium	2
Cross-Domain Misconfiguration	Medium	26
Missing Anti-clickjacking Header	Medium	2
Strict-Transport-Security Header Not Set	Low	3
X-Content-Type-Options Header Missing	Low	26
CSP: Header & Meta	Informational	3
Information Disclosure - Suspicious Comments	Informational	6
Modern Web Application	Informational	1
Re-examine Cache-control Directives	Informational	2
Retrieved from Cache	Informational	30

Alert Detail

Medium	CSP: Wildcard Directive
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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://himanshu240601.github.io/
Method	GET

Attack	
Evidence	default-src 'none'; style-src 'unsafe-inline'; img-src data:; connect-src 'self'
Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: form-action The directive(s): form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
URL	https://himanshu240601.github.io/
Method	GET
Attack	
Evidence	default-src 'none'; style-src 'unsafe-inline'; img-src data:; connect-src 'self'
Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: frame-ancestors, form-action The directive(s): frame-ancestors, form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
URL	https://himanshu240601.github.io/robots.txt
Method	GET
Attack	
Evidence	default-src 'none'; style-src 'unsafe-inline'; img-src data:; connect-src 'self'
Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: form-action The directive(s): form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
URL	https://himanshu240601.github.io/robots.txt
Method	GET
Attack	
Evidence	default-src 'none'; style-src 'unsafe-inline'; img-src data:; connect-src 'self'
Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: frame-ancestors, form-action The directive(s): frame-ancestors, form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
URL	https://himanshu240601.github.io/sitemap.xml
Method	GET
Attack	
Evidence	default-src 'none'; style-src 'unsafe-inline'; img-src data:; connect-src 'self'
Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: form-action The directive(s): form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
URL	https://himanshu240601.github.io/sitemap.xml
Method	GET
Attack	
Evidence	default-src 'none'; style-src 'unsafe-inline'; img-src data:; connect-src 'self'
Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: frame-ancestors, form-action The directive(s): frame-ancestors, form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.
Instances	6
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
	https://www.w3.org/TR/CSP/ https://caniuse.com/#search=content+security+policy https://content-security-policy.com/

Reference	https://github.com/HtmlUnit/htmlunit-csp https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Medium	**CSP: style-src unsafe-inline**
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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://himanshu240601.github.io/
Method	GET
Attack	
Evidence	default-src 'none'; style-src 'unsafe-inline'; img-src data:; connect-src 'self'
Other Info	style-src includes unsafe-inline.
URL	https://himanshu240601.github.io/robots.txt
Method	GET
Attack	
Evidence	default-src 'none'; style-src 'unsafe-inline'; img-src data:; connect-src 'self'
Other Info	style-src includes unsafe-inline.
URL	https://himanshu240601.github.io/sitemap.xml
Method	GET
Attack	
Evidence	default-src 'none'; style-src 'unsafe-inline'; img-src data:; connect-src 'self'
Other Info	style-src includes unsafe-inline.
Instances	3
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://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
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	https://himanshu240601.github.io/ecommerce-website/
Method	GET
Attack	
Evidence	
Other Info	
URL	https://himanshu240601.github.io/ecommerce-website/user/products_page/index.html
Method	GET
Attack	
Evidence	
Other Info	
Instances	2
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	**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://himanshu240601.github.io/ecommerce-website/
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://himanshu240601.github.io/ecommerce-website/assets/brands/adidas.png
Method	GET
Attack	
Evidence	Access-Control-Allow-Origin: *
	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

Other Info	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://himanshu240601.github.io/ecommerce-website/assets/brands/Calvin%20Klein.png
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://himanshu240601.github.io/ecommerce-website/assets/brands/chanel.png
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://himanshu240601.github.io/ecommerce-website/assets/brands/flying%20machine.png
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://himanshu240601.github.io/ecommerce-website/assets/brands/H&M.png
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://himanshu240601.github.io/ecommerce-website/assets/brands/levis.png
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://himanshu240601.github.io/ecommerce-website/assets/brands/louis%20philippe.png
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://himanshu240601.github.io/ecommerce-website/assets/brands/nike.png
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://himanshu240601.github.io/ecommerce-website/assets/brands/pepe.png
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://himanshu240601.github.io/ecommerce-website/assets/brands/puma.png
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://himanshu240601.github.io/ecommerce-website/assets/product%20images/tshirt.jpg
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://himanshu240601.github.io/ecommerce-website/assets/product%20images/white%20shoes.png
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://himanshu240601.github.io/ecommerce-website/assets/product%20images/whitetop.png
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://himanshu240601.github.io/ecommerce-website/css/style.css
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://himanshu240601.github.io/ecommerce-website/js/carousel.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://himanshu240601.github.io/ecommerce-website/js/loadProducts.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://himanshu240601.github.io/ecommerce-website/js/nav_footer.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://himanshu240601.github.io/ecommerce-website/js/slickslider.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://himanshu240601.github.io/ecommerce-website/slick/slick-theme.css
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://himanshu240601.github.io/ecommerce-website/slick/slick.css
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://himanshu240601.github.io/ecommerce-website/slick/slick.min.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://himanshu240601.github.io/ecommerce-website/user/products_page/index.html

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://himanshu240601.github.io/ecommerce-website/user/products_page/js/filtertoggle.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://himanshu240601.github.io/ecommerce-website/user/products_page/js/shop.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://himanshu240601.github.io/ecommerce-website/user/products_page/shop.css
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	26
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?id=desc.config.dotnet.html5_overly_permissive_cors_policy
CWE Id	264
WASC Id	14
Plugin Id	10098

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	https://himanshu240601.github.io/ecommerce-website/
Method	GET
Attack	
Evidence	
Other Info	
URL	https://himanshu240601.github.io/ecommerce-website/user/products_page/index.html
Method	GET
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	**Strict-Transport-Security Header Not Set**
Description	HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.
URL	https://himanshu240601.github.io/
Method	GET
Attack	
Evidence	
Other Info	
URL	https://himanshu240601.github.io/robots.txt
Method	GET
Attack	
Evidence	
Other Info	
URL	https://himanshu240601.github.io/sitemap.xml
Method	GET
Attack	

Evidence	
Other Info	
Instances	3
Solution	Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.
Reference	https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html https://owasp.org/www-community/Security_Headers https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security https://caniuse.com/stricttransportsecurity https://datatracker.ietf.org/doc/html/rfc6797
CWE Id	319
WASC Id	15
Plugin Id	10035

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	https://himanshu240601.github.io/ecommerce-website/
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/adidas.png
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/Calvin%20Klein.png
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/chanel.png
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/flying%20machine.png
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/H&M.png
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/levis.png
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/louis%20philippe.png
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/nike.png
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/pepe.png
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	https://himanshu240601.github.io/ecommerce-website/assets/brands/puma.png
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	https://himanshu240601.github.io/ecommerce-website/assets/product%20images/tshirt.jpg
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	https://himanshu240601.github.io/ecommerce-website/assets/product%20images/white%20shoes.png
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	https://himanshu240601.github.io/ecommerce-website/assets/product%20images/whitetop.png
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	https://himanshu240601.github.io/ecommerce-website/css/style.css
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	https://himanshu240601.github.io/ecommerce-website/js/carousel.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	https://himanshu240601.github.io/ecommerce-website/js/loadProducts.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	https://himanshu240601.github.io/ecommerce-website/js/nav_footer.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	https://himanshu240601.github.io/ecommerce-website/js/slickslider.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	https://himanshu240601.github.io/ecommerce-website/slick/slick-theme.css
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	https://himanshu240601.github.io/ecommerce-website/slick/slick.css
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	https://himanshu240601.github.io/ecommerce-website/slick/slick.min.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	https://himanshu240601.github.io/ecommerce-website/user/products_page/index.html
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	https://himanshu240601.github.io/ecommerce-website/user/products_page/js/filtertoggle.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	https://himanshu240601.github.io/ecommerce-website/user/products_page/js/shop.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	https://himanshu240601.github.io/ecommerce-website/user/products_page/shop.css
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	26
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	**CSP: Header & Meta**

Description	The message contained both CSP specified via header and via Meta tag. It was not possible to union these policies in order to perform an analysis. Therefore, they have been evaluated individually.
URL	https://himanshu240601.github.io/
Method	GET
Attack	
Evidence	
Other Info	
URL	https://himanshu240601.github.io/robots.txt
Method	GET
Attack	
Evidence	
Other Info	
URL	https://himanshu240601.github.io/sitemap.xml
Method	GET
Attack	
Evidence	
Other Info	
Instances	3
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://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Informational	**Information Disclosure - Suspicious Comments**
Description	The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
URL	https://himanshu240601.github.io/ecommerce-website/
Method	GET
Attack	
Evidence	user
Other Info	The following pattern was used: \bUSER\b and was detected in the element starting with: "<script> loadproducts() loadNavAndFooter("user") // jquery for search box $(document). ready(", see evidence field for the suspicious comment/snippet.
URL	https://himanshu240601.github.io/ecommerce-website/js/carousel.js
Method	GET
Attack	

Evidence	from
Other Info	The following pattern was used: \bFROM\b and was detected 2 times, the first in the element starting with: " "offer_info": "Get premium quality shoes from Nike.", ", see evidence field for the suspicious comment/snippet.
URL	https://himanshu240601.github.io/ecommerce-website/js/carousel.js
Method	GET
Attack	
Evidence	user
Other Info	The following pattern was used: \bUSER\b and was detected in the element starting with: " Shop Now ", see evidence field for the suspicious comment/snippet.
URL	https://himanshu240601.github.io/ecommerce-website/js/loadProducts.js
Method	GET
Attack	
Evidence	from
Other Info	The following pattern was used: \bFROM\b and was detected 3 times, the first in the element starting with: " "p_information": "Latest smart watch from watches&time with body sensors that will keep track of your healt. The watch h", see evidence field for the suspicious comment/snippet.
URL	https://himanshu240601.github.io/ecommerce-website/js/loadProducts.js
Method	GET
Attack	
Evidence	user
Other Info	The following pattern was used: \bUSER\b and was detected 5 times, the first in the element starting with: " ${products[i].rating} ", see evidence field for the suspicious comment/snippet.
URL	https://himanshu240601.github.io/ecommerce-website/slick/slick.min.js
Method	GET
Attack	
Evidence	select
Other Info	The following pattern was used: \bSELECT\b and was detected in the element starting with: "!function(i){"use strict";"function"==typeof define&&define.amd?define(["jquery"],i):" undefined"!=typeof exports?module.exports=", see evidence field for the suspicious comment/snippet.
Instances	6
Solution	Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.
Reference	
CWE Id	200
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	https://himanshu240601.github.io/ecommerce-website/user/products_page/index.html
Method	GET
Attack	

Evidence	All
Other Info	Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
Instances	1
Solution	This is an informational alert and so no changes are required.
Reference	
CWE Id	
WASC Id	
Plugin Id	10109

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://himanshu240601.github.io/ecommerce-website/
Method	GET
Attack	
Evidence	max-age=600
Other Info	
URL	https://himanshu240601.github.io/ecommerce-website/user/products_page/index.html
Method	GET
Attack	
Evidence	max-age=600
Other Info	
Instances	2
Solution	For secure content, 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 https://grayduck.mn/2021/09/13/cache-control-recommendations/
CWE Id	525
WASC Id	13
Plugin Id	10015

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://himanshu240601.github.io/ecommerce-website/
Method	GET
Attack	

Evidence	HIT
Other Info	
URL	https://himanshu240601.github.io/
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/adidas.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/Calvin%20Klein.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/chanel.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/flying%20machine.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/H&M.png
Method	GET
Attack	
Evidence	Age: 0
Other	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is

Info	in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/levis.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/louis%20philippe.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/nike.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/pepe.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/brands/puma.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/product%20images/tshirt.jpg
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/assets/product%20images/white%20shoes.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.

URL	https://himanshu240601.github.io/ecommerce-website/assets/product%20images/whitetop.png
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/css/style.css
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/js/carousel.js
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/js/loadProducts.js
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/js/nav_footer.js
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/js/slickslider.js
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/slick/slick-theme.css
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/slick/slick.css

Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/slick/slick.min.js
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/user/products_page/index.html
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/user/products_page/js/filtertoggle.js
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/user/products_page/js/shop.js
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/ecommerce-website/user/products_page/shop.css
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/robots.txt
Method	GET
Attack	
Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
URL	https://himanshu240601.github.io/sitemap.xml
Method	GET
Attack	

Evidence	Age: 0
Other Info	The presence of the 'Age' header indicates that that a HTTP/1.1 compliant caching server is in use.
Instances	30
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://tools.ietf.org/html/rfc7234 https://tools.ietf.org/html/rfc7231 https://www.rfc-editor.org/rfc/rfc9110.html
CWE Id	
WASC Id	
Plugin Id	10050