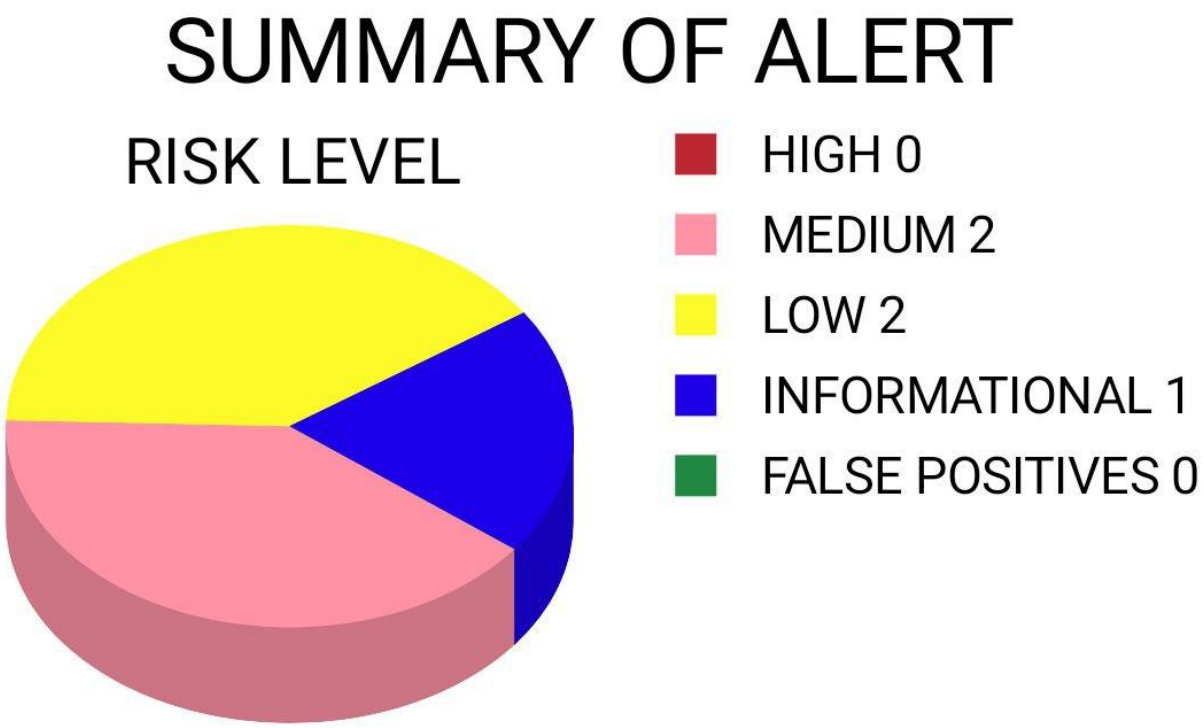
TECHNICAL REPORT OF THE SCANNING ENGINES

Generated automatically on 01/17/2023 at 23:47:00 (UTC)
by Website Security Scanning Monitor
H-X Technologies. Monitor version 6.1.14.

Target object: website https://oliva.in.ua
Date and time of request: 18.01.2023 at 01:14:44 (Europe/Kiev time zone)
Customer: Volodymyr (andruh131090@gmail.com) Group 3#45
IP address of the customer: 93.171.247.158

Resume

Automatic scanning revealed the following number of real and potential technical vulnerabilities 5 in 15 instances. This chart shows the number of vulnerabilities at different risk levels:



#	Website	Vulnerability Name	Count	Risk Level	Weighted Risk
1	https://oliva.in.ua	Content Security Policy (CSP) Header Not Set	3	Medium	69%
2	https://oliva.in.ua	Sub Resource Integrity Attribute Missing	3	Medium	69%
3	https://oliva.in.ua	Strict-Transport-Security Header Not Set	3	Low	36%
4	https://oliva.in.ua	Permissions Policy Header Not Set	3	Low	25%
5	https://oliva.in.ua	Non-Storable Content	3	Informational	2%

Conclusions regarding the security of the target object: the website https://oliva.in.ua is potentially VULNERABLE!

Technical vulnerabilities lead to unauthorized access, compromise, website shutdown, leakage of confidential information and other security incidents.

TECHNICAL REPORT OF THE SCANNING ENGINES

Vulnerability Name (ID)	Content Security Policy (CSP) Header Not Set (10038)	
Risk Level	Medium	
Confidence	High	
Weighted Risk Level	69%	
Confidence Description	High	
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.	
Instances of Vulnerability	URI	https://oliva.in.ua
	Method	GET
	Parameter	
	attack	
	Evidence	
	URI	https://oliva.in.ua/robots.txt
	Method	GET
	Parameter	
	attack	
	Evidence	
	URI	https://oliva.in.ua/sitemap.xml
	Method	GET
	Parameter	
	attack	
	Evidence	
Count	3	
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header, to achieve optimal browser support: "Content-Security-Policy" for Chrome 25+, Firefox 23+ and Safari 7+, "X-Content-Security-Policy" for Firefox 4.0+ and Internet Explorer 10+, and "X-WebKit-CSP" for Chrome 14+ and Safari 6+.	
Other information		
Reference	https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html http://www.w3.org/TR/CSP/ http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html http://www.html5rocks.com/en/tutorials/security/content-security-policy/ http://caniuse.com/#feat=contentsecuritypolicy http://content-security-policy.com/	
CWE ID	693 - https://cwe.mitre.org/data/definitions/693.html	
WASC ID	15	
Vulnerability Name (ID)	Sub Resource Integrity Attribute Missing (90003)	
Risk Level	Medium	
Confidence	High	
Weighted	69%	

Risk Level		
Confidence Description	High	
Description	The integrity attribute is missing on a script or link tag served by an external server. The integrity tag prevents an attacker who have gained access to this server from injecting a malicious content.	
Instances of Vulnerability	URI	https://oliva.in.ua
	Method	GET
	Parameter	
	attack	
	Evidence	<link href='https://fonts.googleapis.com/css?family=Open+Sans:400,700&subset=latin,cyrillic' rel='stylesheet' type='text/css'>
	URI	https://oliva.in.ua/robots.txt
	Method	GET
	Parameter	
	attack	
	Evidence	<link href='https://fonts.googleapis.com/css?family=Open+Sans:400,700&subset=latin,cyrillic' rel='stylesheet' type='text/css'>
	URI	https://oliva.in.ua/sitemap.xml
	Method	GET
	Parameter	
	attack	
	Evidence	<link href='https://fonts.googleapis.com/css?family=Open+Sans:400,700&subset=latin,cyrillic' rel='stylesheet' type='text/css'>
Count	3	
Solution	Provide a valid integrity attribute to the tag.	
Other information		
Reference	https://developer.mozilla.org/en/docs/Web/Security/Subresource_Integrity	
CWE ID	345 - https://cwe.mitre.org/data/definitions/345.html	
WASC ID	15	
Vulnerability Name (ID)	Permissions Policy Header Not Set (10063-1)	
Risk Level	Low	
Confidence	Medium	
Weighted Risk Level	25%	
Confidence Description	Medium	
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.	
Instances of Vulnerability	URI	https://oliva.in.ua
	Method	GET
	Parameter	
	attack	
	Evidence	
	URI	https://oliva.in.ua/robots.txt
	Method	GET

	Parameter	
	attack	
	Evidence	
	URI	https://oliva.in.ua/sitemap.xml
	Method	GET
	Parameter	
	attack	
	Evidence	
Count	3	
Solution	Ensure that your web server, application server, load balancer, etc. is configured to set the Permissions-Policy header.	
Other information		
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Feature-Policy https://developers.google.com/web/updates/2018/06/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 - https://cwe.mitre.org/data/definitions/693.html	
WASC ID	15	
Vulnerability Name (ID)	Strict-Transport-Security Header Not Set (10035)	
Risk Level	Low	
Confidence	High	
Weighted Risk Level	36%	
Confidence Description	High	
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.	
Instances of Vulnerability	URI	https://oliva.in.ua
	Method	GET
	Parameter	
	attack	
	Evidence	
	URI	https://oliva.in.ua/robots.txt
	Method	GET
	Parameter	
	attack	
	Evidence	
	URI	https://oliva.in.ua/sitemap.xml
	Method	GET
	Parameter	
	attack	
	Evidence	
Count	3	
Solution	Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.	
Other information		
Reference	https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.htm	

	l https://owasp.org/www-community/Security_Headers http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security http://caniuse.com/stricttransportsecurity http://tools.ietf.org/html/rfc6797	
CWE ID	319 - https://cwe.mitre.org/data/definitions/319.html	
WASC ID	15	
Vulnerability Name (ID)	Non-Storable Content (10049)	
Risk Level	Informational	
Confidence	Medium	
Weighted Risk Level	2%	
Confidence Description	Medium	
Description	The response contents are not storable by caching components such as proxy servers. If the response does not contain sensitive, personal or user-specific information, it may benefit from being stored and cached, to improve performance.	
Instances of Vulnerability	URI	https://oliva.in.ua
	Method	GET
	Parameter	
	attack	
	Evidence	403
	URI	https://oliva.in.ua/robots.txt
	Method	GET
	Parameter	
	attack	
	Evidence	403
	URI	https://oliva.in.ua/sitemap.xml
	Method	GET
	Parameter	
	attack	
	Evidence	403
Count	3	
Solution	The content may be marked as storable by ensuring that the following conditions are satisfied: The request method must be understood by the cache and defined as being cacheable ("GET", "HEAD", and "POST" are currently defined as cacheable) The response status code must be understood by the cache (one of the 1XX, 2XX, 3XX, 4XX, or 5XX response classes are generally understood) The "no-store" cache directive must not appear in the request or response header fields For caching by "shared" caches such as "proxy" caches, the "private" response directive must not appear in the response For caching by "shared" caches such as "proxy" caches, the "Authorization" header field must not appear in the request, unless the response explicitly allows it (using one of the "must-revalidate", "public", or "s-maxage" Cache-Control response directives) In addition to the conditions above, at least one of the following conditions must also be satisfied by the response: It must contain an "Expires" header field It must contain a "max-age" response directive For "shared" caches such as "proxy" caches, it must contain a "s-maxage" response directive It must contain a "Cache Control Extension" that allows it to be cached It must have a status code that is defined as cacheable by default (200, 203, 204, 206, 300, 301, 404, 405, 410, 414, 501).	
Other information		

Reference	https://tools.ietf.org/html/rfc7234 https://tools.ietf.org/html/rfc7231 http://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html (obsoleted by rfc7234)
CWE ID	524 - https://cwe.mitre.org/data/definitions/524.html
WASC ID	13