Virtuelity
Vulnerability Scanner

Vulnerability Scan Report

ZAP Scanning Report

Sites: https://Thethrone.in https://thethrone.in

Generated on Tue, 15 Apr 2025 15:27:54

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	6
Low	5
Informational	4
False Positives:	0

Summary of Sequences

For each step: result (Pass/Fail) - risk (of highest alert(s) for the step, if any).

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	2
CSP: Failure to Define Directive with No Fallback	Medium	1
CSP: Wildcard Directive	Medium	1
CSP: script-src unsafe-inline	Medium	1
CSP: style-src unsafe-inline	Medium	1
Content Security Policy (CSP) Header Not Set	Medium	3

Cookie No HttpOnly Flag	Low	4
Cookie Without Secure Flag	Low	6
Cross-Domain JavaScript Source File Inclusion	Low	2
Strict-Transport-Security Header Not Set	Low	3
Timestamp Disclosure - Unix	Low	15
Information Disclosure - Suspicious Comments	Informational	1
Modern Web Application	Informational	1
Re-examine Cache-control Directives	Informational	1
Session Management Response Identified	Informational	1

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://thethrone.in/
Method	GET
Parameter	

	Attack	
	Evidence	<form action="/cart" id="CartDrawer-Form" class="cart__contents cart-drawer__form" method="post" >
	Other Info	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 1: ""].
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	<form method="post" action="/contact#ContactFooter" id="ContactFooter" accept-charset="UTF-8" class="footer__newsletter newsletter-form">
	Other Info	No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 3: "contact[tags]" "form_type" "NewsletterForm--sections--23415186129206__footer" "utf8"].
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	https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.h https://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	10202

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		https://thethrone.in/
	Method	GET
	Parameter	content-security-policy
	Attack	
	Evidence	block-all-mixed-content; frame-ancestors 'none'; upgrade-insecure-requests;
	Other Info	The directive(s): 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://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_reso
CWE Id		693
WASC Id		15
Plugin Id		10055
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://thethrone.in/
	Method	GET
	Parameter	content-security-policy
	Attack	
	Evidence	block-all-mixed-content; frame-ancestors 'none'; upgrade-insecure-requests;

	Other Info	The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: script-src, style-src, img-src, connect-src, frame-src, font-src, media-src, object-src, manifest-src, worker-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://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_reso
CWE Id		693
WASC Id		15
Plugin Id		10055
Medium		**CSP: script-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://thethrone.in/
	Method	GET
	Parameter	content-security-policy
	Attack	
	Evidence	block-all-mixed-content; frame-ancestors 'none'; upgrade-insecure-requests;
	Other Info	script-src includes unsafe-inline.
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://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_reso

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://thethrone.in/
	Method	GET
	Parameter	content-security-policy
	Attack	
	Evidence	block-all-mixed-content; frame-ancestors 'none'; upgrade-insecure-requests;
	Other Info	style-src includes unsafe-inline.
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://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_reso
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://Thethrone.in
	Method	GET
	Parameter	
	Attack	
	Evidence	
	Other Info	
URL		https://Thethrone.in/robots.txt
	Method	GET
	Parameter	
	Attack	
	Evidence	
	Other Info	
URL		https://Thethrone.in/sitemap.xml
	Method	GET
	Parameter	
	Attack	
	Evidence	
	Other Info	
Instances		3
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

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://thethrone.in/
Method	GET
Parameter	_shopify_s
Attack	
Evidence	set-cookie: _shopify_s
Other Info	
URL	https://thethrone.in/
Method	GET
Parameter	_shopify_y
Attack	
Evidence	set-cookie: _shopify_y
Other Info	
URL	https://thethrone.in/
Method	GET
Parameter	_tracking_consent
Attack	
Evidence	set-cookie: _tracking_consent
Other Info	
URL	https://thethrone.in/
Method	GET
Parameter	localization
Attack	
Evidence	set-cookie: localization

Other Info	
Instances	4
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://thethrone.in/
Method	GET
Parameter	_landing_page
Attack	
Evidence	set-cookie: _landing_page
Other Info	
URL	https://thethrone.in/
Method	GET
Parameter	_orig_referrer
Attack	
Evidence	set-cookie: _orig_referrer
Other Info	
URL	https://thethrone.in/
Method	GET
Parameter	_shopify_s
Attack	
Evidence	set-cookie: _shopify_s
Other Info	

	URL	https://thethrone.in/
	Method	GET
	Parameter	_shopify_y
	Attack	
	Evidence	set-cookie: _shopify_y
	Other Info	
	URL	https://thethrone.in/
	Method	GET
	Parameter	_tracking_consent
	Attack	
	Evidence	set-cookie: _tracking_consent
	Other Info	
	URL	https://thethrone.in/
	Method	GET
	Parameter	localization
	Attack	
	Evidence	set-cookie: localization
	Other Info	
Instances		6
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-Se
CWE Id		614
WASC Id		13
Plugin Id		10011
Low		**Cross-Domain JavaScript Source File Inclusion**
Description		The page includes one or more script files from a third-party domain.

URL		https://thethrone.in/
	Method	GET
	Parameter	https://unpkg.com/@google/model-viewer/dist/model-viewer-legacy.js
	Attack	
	Evidence	<script nomodule src="https://unpkg.com/@google/model-viewer/dist/model-viewer-legacy.js"></script>
	Other Info	
URL		https://thethrone.in/
	Method	GET
	Parameter	https://unpkg.com/@google/model-viewer/dist/model-viewer.js
	Attack	
	Evidence	<script type="module" src="https://unpkg.com/@google/model-viewer/dist/model-viewer.js"></script>
	Other Info	
Instances		2
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		**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://Thethrone.in
	Method	GET
	Parameter	
	Attack	
	Evidence	

Other Info	
URL	https://Thethrone.in/robots.txt
Method	GET
Parameter	
Attack	
Evidence	
Other Info	
URL	https://Thethrone.in/sitemap.xml
Method	GET
Parameter	
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	**Timestamp Disclosure - Unix**
Description	A timestamp was disclosed by the application/web server. - Unix
URL	https://thethrone.in/
Method	GET
Parameter	

	Attack	
	Evidence	1478001846
	Other Info	1478001846, which evaluates to: 2016-11-01 17:34:06.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1729863339
	Other Info	1729863339, which evaluates to: 2024-10-25 19:05:39.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1729869869
	Other Info	1729869869, which evaluates to: 2024-10-25 20:54:29.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1729869923
	Other Info	1729869923, which evaluates to: 2024-10-25 20:55:23.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1729871768

	Other Info	1729871768, which evaluates to: 2024-10-25 21:26:08.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1729871915
	Other Info	1729871915, which evaluates to: 2024-10-25 21:28:35.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1729875707
	Other Info	1729875707, which evaluates to: 2024-10-25 22:31:47.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1729875746
	Other Info	1729875746, which evaluates to: 2024-10-25 22:32:26.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1729877982
	Other Info	1729877982, which evaluates to: 2024-10-25 23:09:42.
URL		https://thethrone.in/

	Method	GET
	Parameter	
	Attack	
	Evidence	1729878089
	Other Info	1729878089, which evaluates to: 2024-10-25 23:11:29.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1729943922
	Other Info	1729943922, which evaluates to: 2024-10-26 17:28:42.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1742395481
	Other Info	1742395481, which evaluates to: 2025-03-19 20:14:41.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	1744711069
	Other Info	1744711069, which evaluates to: 2025-04-15 15:27:49.
URL		https://thethrone.in/
	Method	GET
	Parameter	server-timing
	Attack	

	Evidence	1744711069
	Other Info	1744711069, which evaluates to: 2025-04-15 15:27:49.
URL		https://thethrone.in/
	Method	GET
	Parameter	x-request-id
	Attack	
	Evidence	1744711069
	Other Info	1744711069, which evaluates to: 2025-04-15 15:27:49.
Instances		15
Solution		Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.
Reference		https://cwe.mitre.org/data/definitions/200.html
CWE Id		497
WASC Id		13
Plugin Id		10096
Informational		**Information Disclosure - Suspicious Comments**
Description		The response appears to contain suspicious comments which may help an attacker.
URL		https://thethrone.in/
	Method	GET
	Parameter	
	Attack	
	Evidence	from
	Other Info	The following pattern was used: \bFROM\b and was detected in likely comment: "//cdn.shopify.com/shopifycloud/storefront-forms-hcaptcha/ce_storefront_forms_captcha_hcaptcha.v1.5.2 see evidence field for the suspicious comment/snippet.
Instances		1
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	https://thethrone.in/
Method	GET
Parameter	
Attack	
Evidence	<a class="frontpage-menu-link" href=""></a>
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://thethrone.in/
Method	GET
Parameter	cache-control
Attack	
Evidence	
Other Info	
Instances	1
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-c 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	**Session Management Response Identified**
Description	The given response has been identified as containing a session management token. The 'Other Info' field contains a set of header tokens that can be used in the Header Based Session Management Method. If the request is in a context which has a Session Management Method set to "Auto-Detect" then this rule will change the session management to use the tokens identified.
URL	https://thethrone.in/
Method	GET
Parameter	_shopify_y
Attack	
Evidence	2D7789EB-6f9f-4313-ad49-2c80c368756a
Other Info	cookie:_shopify_y cookie:_shopify_s cookie:_tracking_consent
Instances	1
Solution	This is an informational alert rather than a vulnerability and so there is nothing to fix.
Reference	https://www.zaproxy.org/docs/desktop/addons/authentication-helper/session-mgmt-id
CWE Id	
WASC Id	
Plugin Id	10112

Sequence Details

With the associated active scan results.

Report generated by VirtuesTech Security Scanner

Virtuelity

Vulnerability Scanner