DFD Scanning Report ajax

Site: https://dfd.dev12.intersmarthosting.in

Generated on Mon, 11 Dec 2023 14:41:45

ZAP Version: 2.14.0

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	4
Low	5
Informational	5

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	2
Content Security Policy (CSP) Header Not Set	Medium	1
Hidden File Found	Medium	1
Missing Anti-clickjacking Header	Medium	1
Cookie No HttpOnly Flag	Low	1
Cookie Without Secure Flag	Low	2
Cross-Domain JavaScript Source File Inclusion	Low	7
Strict-Transport-Security Header Not Set	Low	1
X-Content-Type-Options Header Missing	Low	1
Information Disclosure - Suspicious Comments	Informational	3
Modern Web Application	Informational	1
Re-examine Cache-control Directives	Informational	1
Session Management Response Identified	Informational	5
User Agent Fuzzer	Informational	12

Alert Detail

Medium	Absence of Anti-CSRF Tokens
	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.

Description	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://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<form action="https://dfd.dev12.intersmarthosting.in/shop" method="GET">
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] was found in the following HTML form: [Form 1: "search"].
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<form action="https://dfd.dev12.intersmarthosting.in/shop" method="GET">
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] was found in the following HTML form: [Form 3: "search"].
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	http://projects.webappsec.org/Cross-Site-Request-Forgery https://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	10202

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://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	
Other Info	
Instances	1
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 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
WASC Id	15
Plugin Id	10038

Medium	Hidden File Found
Description	A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.
URL	https://dfd.dev12.intersmarthosting.in/BitKeeper
Method	GET
Attack	
Evidence	HTTP/1.1 200 OK
Other Info	

Instances	1
Solution	Consider whether or not the component is actually required in production, if it isn't then disable it. If it is then ensure access to it requires appropriate authentication and authorization, or limit exposure to internal systems or specific source IPs, etc.
Reference	https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html
CWE Id	538
WASC Id	13
Plugin Id	40035

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://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	
Other Info	
Instances	1
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	**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://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	Set-Cookie: XSRF-TOKEN
Other Info	
Instances	1
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://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	Set-Cookie: laravel_session
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	Set-Cookie: XSRF-TOKEN
Other Info	
Instances	2
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-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
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://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/gsap/3.5.1/gsap.min.js"></script>
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/jquery-mousewheel/3.0.6/jquery.mousewheel.min.js"> </script>
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/jquery.touchswipe/1.6.19/jquery.touchSwipe.min.js"> </script>

Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/owl.carousel.min.js"></script>
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<script src="https://cdn.jsdelivr.net/npm/bootstrap-select@1.14.0-beta2/dist/js/bootstrap-select.min.js"></script>
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/js/select2.min.js"></script>
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<script src="https://cdnjs.cloudflare.com/ajax/libs/aos/2.3.4/aos.js"></script>
Other Info	
Instances	7
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://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	
Other Info	

Instances	1
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 http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security http://caniuse.com/stricttransportsecurity http://tools.ietf.org/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://dfd.dev12.intersmarthosting.in/
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	1
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	http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx https://owasp.org/www-community/Security_Headers
CWE Id	693
WASC Id	15
Plugin Id	10021

Informational	**Information Disclosure - Suspicious Comments**
Description	The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	from
Other Info	The following pattern was used: \bFROM\b and was detected in the element starting with: "<!-- <div class="ti", see evidence field for the suspicious comment/snippet.
URL	https://dfd.dev12.intersmarthosting.in/

Method	GET
Attack	
Evidence	select
Other Info	The following pattern was used: \bSELECT\b and was detected 3 times, the first in the element starting with: "<!-- <select class="selectpicker langDrop" id="language-change"> <option selected data-content="", see evidence field for the suspicious comment/snippet.
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	user
Other Info	The following pattern was used: \bUSER\b and was detected 16 times, the first in the element starting with: "<!-- <img src="assets/images/user.jpg" alt=""> < /div> -->", see evidence field for the suspicious comment/snippet.
Instances	3
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://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	<a data-bs-toggle="modal" data-bs-target="#SearchModal" href="#" onclick="return false;" class="mobSearch"> <svg viewBox="0 0 20.579 20.756"> <path id="Path_14" data-name=" Path 14" class="cls-1" d="M27.2,26.469l-3.54-3.716a9.643,9.643,0,1,0-.907.907l3.54,3.716 a.641.641,0,0,0,.907-.907ZM10.532,22.321a8.336,8.336,0,1,1,5.895,2.442A8. 283,8.283,0,0,1,10.532,22.321Z" transform="translate(-6.808 -6.808)" /> </svg> Search < /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://dfd.dev12.intersmarthosting.in/
Method	GET

Attack	
Evidence	no-cache, private
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-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	**Session Management Response Identified**
Description	The given response has been identified as containing a session management token. The 'Other
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	eyJpdiI6IkZ2bzlOcXo3RWQ3N1VuVU96ZjFtL2c9PSIsInZhbHVlIjoidjBCMDc4Z1JxK0ZYZ05TZ0
Other Info	cookie:laravel_session cookie:XSRF-TOKEN
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	eyJpdiI6InBSczRNMGxVMVZwYkJEdG12blJqN1E9PSIsInZhbHVlIjoiUVIraEJJUFBpZjNZQTJU 3D
Other Info	cookie:laravel_session cookie:XSRF-TOKEN
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	eyJpdiI6InV4NWFDemsvMHNsRHA5RjNEVXRzbmc9PSIsInZhbHVlIjoidWxJNjhjY0xpbCt4T2Fv 3D
Other Info	cookie:laravel_session cookie:XSRF-TOKEN
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	
Evidence	eyJpdiI6InZUaml2NkRFaWYzWTZrOU0vaU5CTWc9PSIsInZhbHVlIjoicTR5OHhsMktraElubDVv
Other Info	cookie:laravel_session cookie:XSRF-TOKEN
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	

Evidence	eyJpdiI6InpjMlJzK0xaRE44U1VyS2t4TFg4OWc9PSIsInZhbHVlIjoiWGt3Z3E0L3EwQjM1V3I2dX
Other Info	cookie:laravel_session cookie:XSRF-TOKEN
Instances	5
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

Informational	User Agent Fuzzer
Description	Check for differences in response based on fuzzed User Agent (eg. mobile sites, access as a Search Engine Crawler). Compares the response statuscode and the hashcode of the response body with the original response.
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Trident/7.0; rv:11.0) like Gecko
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3739.0 Safari/537.36 Edg/75.0.109.0
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/

Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:93.0) Gecko/20100101 Firefox/91.0
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/5.0 (iPhone; CPU iPhone OS 8_0_2 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Version/8.0 Mobile/12A366 Safari/600.1.4
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_0 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7A341 Safari/528.16
Evidence	
Other Info	
URL	https://dfd.dev12.intersmarthosting.in/
Method	GET
Attack	msnbot/1.1 (+http://search.msn.com/msnbot.htm)
Evidence	
Other Info	
Instances	12

Solution	
Reference	https://owasp.org/wstg
CWE Id	
WASC Id	
Plugin Id	10104