Web Application Report

This report includes important security information about your web application.

Security Report

This report was created by HCL AppScan Standard 10.1.0
Scan started: 11/23/2025 9:08:35 AM

Table of Contents

Introduction

Summary

Introduction

This report contains the results of a web application security scan performed by HCL AppScan Standard.

Medium severity issues:	20
Low severity issues:	8
Informational severity issues:	37
Total security issues included in the report:	65
Total security issues discovered in the scan:	65

General Information

Scan file name:	KAY_DMIS_SCAN
Scan started:	11/23/2025 9:08:35 AM
Test policy:	Complete
Test optimization level:	Fastest

Host	drims.nginxdev.egovja.com
Port	443
Operating system:	Unknown
Web server:	Unknown
Application server:	Any

Login Settings

Login method:	Recorded login
Concurrent logins:	Enabled
In-session detection:	Enabled
In-session pattern:	`> Logout<`
Tracked or session ID cookies:	`session`
Tracked or session ID parameters:	

Login sequence:	`https://drims.nginxdev.egovja.com/`
	`https://drims.nginxdev.egovja.com/login?next=/`
	`https://drims.nginxdev.egovja.com/login`
	`https://drims.nginxdev.egovja.com/login`
	`https://drims.nginxdev.egovja.com/login`
	`https://drims.nginxdev.egovja.com/dashboard/`
	`https://drims.nginxdev.egovja.com/static/js/form-utils.js`

```
https://drims.nginxdev.egovja.com/notifications/api/unread_count
https://drims.nginxdev.egovja.com/notifications/api/unread_count
https://drims.nginxdev.egovja.com/users/
https://drims.nginxdev.egovja.com/static/js/form-utils.js
https://drims.nginxdev.egovja.com/notifications/api/unread_count
https://drims.nginxdev.egovja.com/notifications/api/unread_count
https://drims.nginxdev.egovja.com/notifications/api/unread_count
https://drims.nginxdev.egovja.com/users/26/edit
https://drims.nginxdev.egovja.com/static/js/form-utils.js
https://drims.nginxdev.egovja.com/notifications/api/unread_count
https://drims.nginxdev.egovja.com/users/
https://drims.nginxdev.egovja.com/notifications/api/unread_count
https://drims.nginxdev.egovja.com/notifications/api/unread_count
https://drims.nginxdev.egovja.com/users/27/edit
https://drims.nginxdev.egovja.com/static/js/form-utils.js
https://drims.nginxdev.egovja.com/notifications/api/unread_count
```

Summary

Issue Types 14

TOC

	Issue Type	Number of Issues
M	Cross-Site Request Forgery	17
M	Missing Secure Attribute in Encrypted Session (SSL) Cookie	1
M	Weak Cipher Suites - ROBOT Attack : Vulnerable cipher suites are supported by the server	1
M	Weak Ciphers are detected: Not all cipher suites support Perfect Forward Secrecy	1
L	Autocomplete HTML Attribute Not Disabled for Password Field	1
L	Cookie with Insecure or Improper or Missing SameSite attribute	1
L	Missing HttpOnly Attribute in Session Cookie	1
L	Missing or Insecure "Script-Src" policy in "Content-Security-Policy" header	1
L	Missing or insecure HTTP Strict-Transport-Security Header	1
L	Query Parameter in SSL Request	1
L	SHA-1 cipher suites were detected	1
L	Unnecessary Http Response Headers found in the Application	1
I	Cookie with SameSite attribute not Restrictive	1
I	Email Address Pattern Found	36

Vulnerable URLs 39

TOC

	URL	Number of Issues
M	https://drims.nginxdev.egovja.com/account-requests/	1
M	https://drims.nginxdev.egovja.com/agencies/	2
M	https://drims.nginxdev.egovja.com/agencies/6/deactivate	2
M	https://drims.nginxdev.egovja.com/agencies/8/edit	2
M	https://drims.nginxdev.egovja.com/custodians/create	2
M	https://drims.nginxdev.egovja.com/donors/2/edit	2
M	https://drims.nginxdev.egovja.com/items/	2
M	https://drims.nginxdev.egovja.com/items/20/inactivate	2
M	https://drims.nginxdev.egovja.com/items/create	2
M	https://drims.nginxdev.egovja.com/notifications/api/clear-all	1

M	https://drims.nginxdev.egovja.com/profile/edit	2
M	https://drims.nginxdev.egovja.com/profile/preferences	2
M	https://drims.nginxdev.egovja.com/uom/	2
M	https://drims.nginxdev.egovja.com/users/27/deactivate	2
M	https://drims.nginxdev.egovja.com/warehouses/	2
M	https://drims.nginxdev.egovja.com/warehouses/1/edit	2
M	https://drims.nginxdev.egovja.com/warehouses/create	2
M	https://drims.nginxdev.egovja.com/	10
L	https://drims.nginxdev.egovja.com/login	3
I	https://drims.nginxdev.egovja.com/agencies/6	1
I	https://drims.nginxdev.egovja.com/agencies/8	1
I	https://drims.nginxdev.egovja.com/agencies/create	1
I	https://drims.nginxdev.egovja.com/custodians/	1
I	https://drims.nginxdev.egovja.com/dashboard/	1
I	https://drims.nginxdev.egovja.com/donors/	1
I	https://drims.nginxdev.egovja.com/item-categories/6	1
I	https://drims.nginxdev.egovja.com/item-categories/6/edit	1
I	https://drims.nginxdev.egovja.com/item-categories/7	1
I	https://drims.nginxdev.egovja.com/item-categories/create	1
I	https://drims.nginxdev.egovja.com/items/20	1
I	https://drims.nginxdev.egovja.com/items/21	1
I	https://drims.nginxdev.egovja.com/items/21/activate	1
I	https://drims.nginxdev.egovja.com/profile/	1
I	https://drims.nginxdev.egovja.com/users/	1
I	https://drims.nginxdev.egovja.com/users/27	1
I	https://drims.nginxdev.egovja.com/warehouses/1	1
I	https://drims.nginxdev.egovja.com/warehouses/1/delete	1
I	https://drims.nginxdev.egovja.com/warehouses/12	1
I	https://drims.nginxdev.egovja.com/warehouses/12/delete	1

Fix Recommendations 11

TOC

	Remediation Task	Number of Issues
M	Add the 'Secure' attribute to all sensitive cookies	1
M	Change server's supported ciphersuites	3
M	Validate the value of the "Referer" header, and use a one-time-nonce for each submitted form	17
L	Add the 'HttpOnly' attribute to all session cookies	1
L	Always use SSL and POST (body) parameters when sending sensitive information.	1
L	Config your server to use the "Content-Security-Policy" header with secure policies	1
L	Correctly set the "autocomplete" attribute to "off"	1
L	Do not allow sensitive information to leak.	1

L	Implement the HTTP Strict-Transport-Security policy with a long "max-age"	1
L	Remove e-mail addresses from the website	36
L	Review possible solutions for configuring SameSite Cookie attribute to recommended values	2

Security Risks 9

TOC

	Risk	Number of Issues
M	It may be possible to force an end-user to execute unwanted actions on a web application in which they're currently authenticated.	17
M	It may be possible to steal user and session information (cookies) that was sent during an encrypted session	1
M	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user	4
L	It may be possible to bypass the web application's authentication mechanism	1
L	Prevent cookie information leakage by restricting cookies to first-party or same-site context, Attacks can extend to Cross-Site-Request-Forgery (CSRF) attacks if there are no additional protections in place (such as Anti-CSRF tokens).	1
L	It is possible to gather sensitive information about the web application such as usernames, passwords, machine name and/or sensitive file locations	39
L	It is possible to persuade a naive user to supply sensitive information such as username, password, credit card number, social security number etc.	2
L	It may be possible to steal sensitive data such as credit card numbers, social security numbers etc. that are sent unencrypted	1
I	Prevent cookie information leakage by restricting cookies to first-party or same-site context to Strict.	1

Causes 12

TOC

	Cause	Number of Issues
M	This vulnerability arises because the application allows the user to perform some sensitive action without verifying that the request was sent intentionally.	17
M	An attacker can cause a victim's browser to emit an HTTP request to an arbitrary URL in the application. When this request is sent from an authenticated victim's browser, it will include the victim's session cookie or authentication header. The application will accept this as a valid request from an authenticated user.	17
M	When a web server is designed to receive a request from a client without any mechanism for verifying that it was intentionally sent, an attacker may be able to trick a client into making an unintentional request from a different site, which will be treated as an authentic request by the application. This can be done by submitting a form, loading an image, sending an XMLHttpRequest in JavaScript, and more.	17
M	For example, this IMG tag can be embedded in an attacker's webpage, and the victim's browser will submit a request to retrieve the image. This valid request will be processed by the application, and the browser will not display a broken image. `<img src="https://myapp.com/transfer?acct=VICTIM&amount=10000" width=0 height=0 border=0>`. As a result, money is transferred from the victim's account to the attacker, using the victim’s session.	17

M	The web application sends non-secure cookies over SSL	1
M	Cipher Suites that support TLS-RSA key exchange were detected. The web server or application server with TLS implementation flaw may be vulnerable to ROBOT Attack. This issue does not necessarily mean that you are vulnerable. Please follow the advisory guideline.	1
M	The web server or application server are configured in an insecure way	2
L	Insecure web application programming or configuration	40
L	Sensitive Cookie with Improper or Insecure or Missing SameSite Attribute	1
L	The web application sets session cookies without the HttpOnly attribute	1
L	Query parameters were passed over SSL, and may contain sensitive information	1
I	Sensitive Cookie with Unrestrictive SameSite Attributes and Flags	1

WASC Threat Classification

TOC

Threat	Number of Issues
Cross-site Request Forgery	17
Information Leakage	43
Server Misconfiguration	5