GOJ DMIS Scan Report

Project Name	GOJ DMIS
Scan Start	Monday, December 1, 2025 9:37:12 AM
Preset	Checkmarx Default
Scan Time	00h:01m:23s
Lines Of Code Scanned	88889
Files Scanned	514
Report Creation Time	Monday, December 1, 2025 10:12:01 AM
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7
Team	CxServer
Checkmarx Version	9.7.4.1001 HF6
Scan Type	Full
Source Origin	LocalPath
Density	4/1000 (Vulnerabilities/LOC)
Visibility	Public

Filter Settings

Severity

Included:	Critical, High, Medium, Low, Information
Excluded:	None

Result State

Included:	To Verify, Not Exploitable, Confirmed, Urgent, Proposed Not Exploitable
Excluded:	None

Assigned to

Included:	All

Categories

Included:

Uncategorized	All
OWASP Top 10 2013	All
FISMA 2014	All
NIST SP 800-53	All
OWASP Top 10 2017	All
OWASP Mobile Top 10 2016	All
OWASP Top 10 API	All
OWASP Top 10 2010	All
ASD STIG 4.10	All
Custom	All
CWE top 25	All
MOIS(KISA) Secure Coding 2021	All
OWASP ASVS	All

OWASP Top 10 2021	All
PCI DSS v3.2.1	All
SANS top 25	All
ASA Mobile Premium	All
ASA Premium	All
Top Tier	All
Base Preset	All
PCI DSS v4.0	All
OWASP Top 10 API 2023	All
ASD STIG 6.1	All
OWASP Mobile Top 10 2024	All

Excluded:

Uncategorized	None
OWASP Top 10 2013	None
FISMA 2014	None
NIST SP 800-53	None
OWASP Top 10 2017	None
OWASP Mobile Top 10 2016	None
OWASP Top 10 API	None
OWASP Top 10 2010	None
ASD STIG 4.10	None
Custom	None
CWE top 25	None
MOIS(KISA) Secure Coding 2021	None
OWASP ASVS	None
OWASP Top 10 2021	None
PCI DSS v3.2.1	None
SANS top 25	None
ASA Mobile Premium	None
ASA Premium	None
Top Tier	None
Base Preset	None

PCI DSS v4.0	None
OWASP Top 10 API 2023	None
ASD STIG 6.1	None
OWASP Mobile Top 10 2024	None

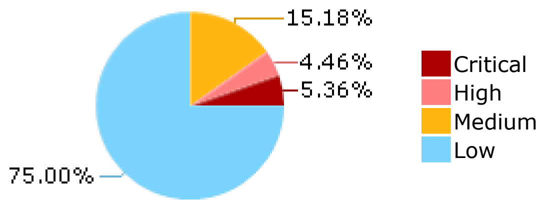
Results Limit

Results limit per query was set to 50

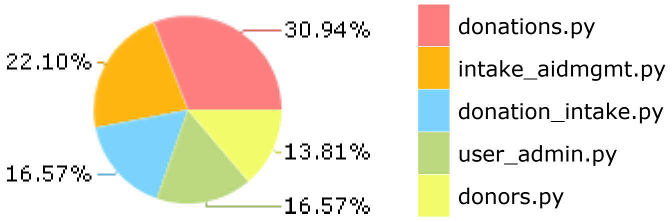
Selected Queries

Selected queries are listed in Result Summary

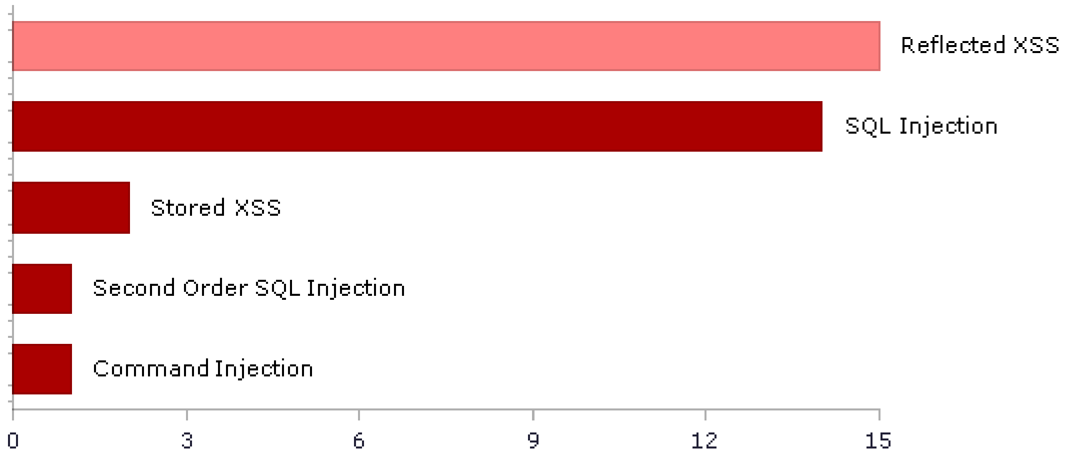
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

Further details and elaboration about vulnerabilities and risks can be found at: OWASP Top 10 2017

Category	Threat Agent	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection*	App. Specific	EASY	COMMON	EASY	SEVERE	App. Specific	250	54
A2-Broken Authentication*	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	1	1
A3-Sensitive Data Exposure*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	0	0
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	5	4
A6-Security Misconfiguration *	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	15	15
A7-Cross-Site Scripting (XSS)*	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	19	10
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	0	0
A10-Insufficient Logging & Monitoring	App. Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App. Specific	0	0

* Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - OWASP Top 10 2021

Category	Issues Found	Best Fix Locations
A1-Broken Access Control*	17	9
A2-Cryptographic Failures*	0	0
A3-Injection*	43	19
A4-Insecure Design*	257	71
A5-Security Misconfiguration	0	0
A6-Vulnerable and Outdated Components	0	0
A7-Identification and Authentication Failures*	4	4
A8-Software and Data Integrity Failures*	11	11
A9-Security Logging and Monitoring Failures*	6	3
A10-Server-Side Request Forgery	0	0

* Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - OWASP Mobile Top 10 2024

Category	Issues Found	Best Fix Locations
M1: Improper Credential Usage	0	0
M2: Inadequate Supply Chain Security	0	0
M3: Insecure Authentication/Authorization	0	0
M4: Insufficient Input/Output Validation	0	0
M5: Insecure Communication	0	0
M6: Inadequate Privacy Controls	0	0
M7: Insufficient Binary Protections	0	0
M8: Security Misconfiguration	0	0
M9: Insecure Data Storage	0	0
M10: Insufficient Cryptography	0	0

Scan Summary - PCI DSS v3.2.1

Category	Issues Found	Best Fix Locations
PCI DSS (3.2.1) - 6.5.1 - Injection flaws - particularly SQL injection	29	15
PCI DSS (3.2.1) - 6.5.2 - Buffer overflows*	0	0
PCI DSS (3.2.1) - 6.5.3 - Insecure cryptographic storage	0	0
PCI DSS (3.2.1) - 6.5.4 - Insecure communications	0	0
PCI DSS (3.2.1) - 6.5.5 - Improper error handling	0	0
PCI DSS (3.2.1) - 6.5.7 - Cross-site scripting (XSS)*	17	8
PCI DSS (3.2.1) - 6.5.8 - Improper access control	17	9
PCI DSS (3.2.1) - 6.5.9 - Cross-site request forgery	0	0
PCI DSS (3.2.1) - 6.5.10 - Broken authentication and session management*	218	38

***** Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - FISMA 2014

Category	Description	Issues Found	Best Fix Locations
Access Control	Organizations must limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems) and to the types of transactions and functions that authorized users are permitted to exercise.	0	0
Audit And Accountability*	Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.	0	0
Configuration Management	Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems.	15	15
Identification And Authentication*	Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	0	0
Media Protection*	Organizations must: (i) protect information system media, both paper and digital; (ii) limit access to information on information system media to authorized users; and (iii) sanitize or destroy information system media before disposal or release for reuse.	0	0
System And Communications Protection	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity*	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	274	61

* Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	0	0
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)*	6	3
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	10	10
SC-23 Session Authenticity (P1)	0	0
SC-28 Protection of Information at Rest (P1)*	0	0
SC-4 Information in Shared Resources (P1)*	0	0
SC-5 Denial of Service Protection (P1)	0	0
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	251	50
SI-11 Error Handling (P2)*	15	15
SI-15 Information Output Filtering (P0)*	17	8
SI-16 Memory Protection (P1)*	0	0

* Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - OWASP Mobile Top 10 2016

Category	Description	Issues Found	Best Fix Locations
M1-Improper Platform Usage	This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, misuse of TouchID, the Keychain, or some other security control that is part of the mobile operating system. There are several ways that mobile apps can experience this risk.	0	0
M2-Insecure Data Storage	This category covers insecure data storage and unintended data leakage.	0	0
M3-Insecure Communication	This category covers poor handshaking, incorrect SSL versions, weak negotiation, cleartext communication of sensitive assets, etc.	0	0
M4-Insecure Authentication	This category captures notions of authenticating the end user or bad session management. This can include: -Failing to identify the user at all when that should be required -Failure to maintain the user's identity when it is required -Weaknesses in session management	0	0
M5-Insufficient Cryptography	The code applies cryptography to a sensitive information asset. However, the cryptography is insufficient in some way. Note that anything and everything related to TLS or SSL goes in M3. Also, if the app fails to use cryptography at all when it should, that probably belongs in M2. This category is for issues where cryptography was attempted, but it wasnt done correctly.	0	0
M6-Insecure Authorization	This is a category to capture any failures in authorization (e.g., authorization decisions in the client side, forced browsing, etc.). It is distinct from authentication issues (e.g., device enrolment, user identification, etc.). If the app does not authenticate users at all in a situation where it should (e.g., granting anonymous access to some resource or service when authenticated and authorized access is required), then that is an authentication failure not an authorization failure.	0	0
M7-Client Code Quality	This category is the catch-all for code-level implementation problems in the mobile client. That's distinct from server-side coding mistakes. This would capture things like buffer overflows, format string vulnerabilities, and various other code-level mistakes where the solution is to rewrite some code that's running on the mobile device.	0	0
M8-Code Tampering	This category covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification. Once the application is delivered to the mobile device, the code and data resources are resident there. An attacker can either directly modify the code, change the contents of memory dynamically, change or replace the system APIs that the	0	0

	application uses, or modify the application's data and resources. This can provide the attacker a direct method of subverting the intended use of the software for personal or monetary gain.		
M9-Reverse Engineering	This category includes analysis of the final core binary to determine its source code, libraries, algorithms, and other assets. Software such as IDA Pro, Hopper, otool, and other binary inspection tools give the attacker insight into the inner workings of the application. This may be used to exploit other nascent vulnerabilities in the application, as well as revealing information about back end servers, cryptographic constants and ciphers, and intellectual property.	0	0
M10-Extraneous Functionality	Often, developers include hidden backdoor functionality or other internal development security controls that are not intended to be released into a production environment. For example, a developer may accidentally include a password as a comment in a hybrid app. Another example includes disabling of 2-factor authentication during testing.	0	0

Scan Summary - Custom

Category	Issues Found	Best Fix Locations
Must audit	0	0
Check	0	0
Optional	0	0

Scan Summary - PCI DSS v4.0

Category	Issues Found	Best Fix Locations
PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development*	329	110
PCI DSS (4.0) - 8.6.2 Vulnerabilities related to passwords/passphrases usage*	1	1

* Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - ASD STIG 4.10

Category	Issues Found	Best Fix Locations
APSC-DV-000640 - CAT II The application must provide audit record generation capability for the renewal of session IDs.	0	0
APSC-DV-000650 - CAT II The application must not write sensitive data into the application logs.	0	0
APSC-DV-000660 - CAT II The application must provide audit record generation capability for session timeouts.	0	0
APSC-DV-000670 - CAT II The application must record a time stamp indicating when the event occurred.	0	0
APSC-DV-000680 - CAT II The application must provide audit record generation capability for HTTP headers including User-Agent, Referer, GET, and POST.	0	0
APSC-DV-000690 - CAT II The application must provide audit record generation capability for connecting system IP addresses.	0	0
APSC-DV-000700 - CAT II The application must record the username or user ID of the user associated with the event.	0	0
APSC-DV-000710 - CAT II The application must generate audit records when successful/unsuccessful attempts to grant privileges occur.	0	0
APSC-DV-000720 - CAT II The application must generate audit records when successful/unsuccessful attempts to access security objects occur.	0	0
APSC-DV-000730 - CAT II The application must generate audit records when successful/unsuccessful attempts to access security levels occur.	0	0
APSC-DV-000740 - CAT II The application must generate audit records when successful/unsuccessful attempts to access categories of information (e.g., classification levels) occur.	0	0
APSC-DV-000750 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify privileges occur.	0	0
APSC-DV-000760 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify security objects occur.	0	0
APSC-DV-000770 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify security levels occur.	0	0
APSC-DV-000780 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify categories of information (e.g., classification levels) occur.	0	0
APSC-DV-000790 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete privileges occur.	0	0
APSC-DV-000800 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete security levels occur.	0	0
APSC-DV-000810 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete application database security objects occur.	0	0
APSC-DV-000820 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete categories of information (e.g., classification levels) occur.	0	0
APSC-DV-000830 - CAT II The application must generate audit records when successful/unsuccessful logon attempts occur.	0	0
APSC-DV-000840 - CAT II The application must generate audit records for privileged activities or other system-level access.	0	0
APSC-DV-000850 - CAT II The application must generate audit records showing starting and ending time for user access to the system.	0	0
APSC-DV-000860 - CAT II The application must generate audit records when successful/unsuccessful accesses to objects occur.	0	0

APSC-DV-000870 - CAT II The application must generate audit records for all direct access to the information system.	0	0
APSC-DV-000880 - CAT II The application must generate audit records for all account creations, modifications, disabling, and termination events.	0	0
APSC-DV-000910 - CAT II The application must initiate session auditing upon startup.	0	0
APSC-DV-000940 - CAT II The application must log application shutdown events.	0	0
APSC-DV-000950 - CAT II The application must log destination IP addresses.	0	0
APSC-DV-000960 - CAT II The application must log user actions involving access to data.	0	0
APSC-DV-000970 - CAT II The application must log user actions involving changes to data.	0	0
APSC-DV-000980 - CAT II The application must produce audit records containing information to establish when (date and time) the events occurred.	0	0
APSC-DV-000990 - CAT II The application must produce audit records containing enough information to establish which component, feature or function of the application triggered the audit event.	0	0
APSC-DV-001000 - CAT II When using centralized logging; the application must include a unique identifier in order to distinguish itself from other application logs.	0	0
APSC-DV-001010 - CAT II The application must produce audit records that contain information to establish the outcome of the events.	0	0
APSC-DV-001020 - CAT II The application must generate audit records containing information that establishes the identity of any individual or process associated with the event.	0	0
APSC-DV-001030 - CAT II The application must generate audit records containing the full-text recording of privileged commands or the individual identities of group account users.	0	0
APSC-DV-001040 - CAT II The application must implement transaction recovery logs when transaction based.	0	0
APSC-DV-001050 - CAT II The application must provide centralized management and configuration of the content to be captured in audit records generated by all application components.	0	0
APSC-DV-001070 - CAT II The application must off-load audit records onto a different system or media than the system being audited.	0	0
APSC-DV-001080 - CAT II The application must be configured to write application logs to a centralized log repository.	0	0
APSC-DV-001090 - CAT II The application must provide an immediate warning to the SA and ISSO (at a minimum) when allocated audit record storage volume reaches 75% of repository maximum audit record storage capacity.	0	0
APSC-DV-001100 - CAT II Applications categorized as having a moderate or high impact must provide an immediate real-time alert to the SA and ISSO (at a minimum) for all audit failure events.	0	0
APSC-DV-001110 - CAT II The application must alert the ISSO and SA (at a minimum) in the event of an audit processing failure.	0	0
APSC-DV-001120 - CAT II The application must shut down by default upon audit failure (unless availability is an overriding concern).	0	0
APSC-DV-001130 - CAT II The application must provide the capability to centrally review and analyze audit records from multiple components within the system.	0	0
APSC-DV-001140 - CAT II The application must provide the capability to filter audit records for events of interest based upon organization-defined criteria.	0	0
APSC-DV-001150 - CAT II The application must provide an audit reduction capability that supports on-demand reporting requirements.	0	0
APSC-DV-001160 - CAT II The application must provide an audit reduction capability that supports on-demand audit review and analysis.	0	0
APSC-DV-001170 - CAT II The application must provide an audit reduction capability that supports after-the-fact investigations of security incidents.	0	0
APSC-DV-001180 - CAT II The application must provide a report generation capability that supports on-demand audit review and analysis.	0	0
APSC-DV-001190 - CAT II The application must provide a report generation capability that	0	0

supports on-demand reporting requirements.		
APSC-DV-001200 - CAT II The application must provide a report generation capability that supports after-the-fact investigations of security incidents.	0	0
APSC-DV-001210 - CAT II The application must provide an audit reduction capability that does not alter original content or time ordering of audit records.	0	0
APSC-DV-001220 - CAT II The application must provide a report generation capability that does not alter original content or time ordering of audit records.	0	0
APSC-DV-001250 - CAT II The applications must use internal system clocks to generate time stamps for audit records.	0	0
APSC-DV-001260 - CAT II The application must record time stamps for audit records that can be mapped to Coordinated Universal Time (UTC) or Greenwich Mean Time (GMT).	0	0
APSC-DV-001270 - CAT II The application must record time stamps for audit records that meet a granularity of one second for a minimum degree of precision.	0	0
APSC-DV-001280 - CAT II The application must protect audit information from any type of unauthorized read access.	0	0
APSC-DV-001290 - CAT II The application must protect audit information from unauthorized modification.	0	0
APSC-DV-001300 - CAT II The application must protect audit information from unauthorized deletion.	0	0
APSC-DV-001310 - CAT II The application must protect audit tools from unauthorized access.	0	0
APSC-DV-001320 - CAT II The application must protect audit tools from unauthorized modification.	0	0
APSC-DV-001330 - CAT II The application must protect audit tools from unauthorized deletion.	0	0
APSC-DV-001340 - CAT II The application must back up audit records at least every seven days onto a different system or system component than the system or component being audited.	0	0
APSC-DV-001570 - CAT II The application must electronically verify Personal Identity Verification (PIV) credentials.	0	0
APSC-DV-001350 - CAT II The application must use cryptographic mechanisms to protect the integrity of audit information.	0	0
APSC-DV-001360 - CAT II Application audit tools must be cryptographically hashed.	0	0
APSC-DV-001370 - CAT II The integrity of the audit tools must be validated by checking the files for changes in the cryptographic hash value.	0	0
APSC-DV-001390 - CAT II The application must prohibit user installation of software without explicit privileged status.	0	0
APSC-DV-001410 - CAT II The application must enforce access restrictions associated with changes to application configuration.	0	0
APSC-DV-001420 - CAT II The application must audit who makes configuration changes to the application.	0	0
APSC-DV-001430 - CAT II The application must have the capability to prevent the installation of patches, service packs, or application components without verification the software component has been digitally signed using a certificate that is recognized and approved by the orga	0	0
APSC-DV-001440 - CAT II The applications must limit privileges to change the software resident within software libraries.	0	0
APSC-DV-001460 - CAT II An application vulnerability assessment must be conducted.	0	0
APSC-DV-001480 - CAT II The application must prevent program execution in accordance with organization-defined policies regarding software program usage and restrictions, and/or rules authorizing the terms and conditions of software program usage.	0	0
APSC-DV-001490 - CAT II The application must employ a deny-all, permit-by-exception (whitelist) policy to allow the execution of authorized software programs.	0	0
APSC-DV-001500 - CAT II The application must be configured to disable non-essential capabilities.	0	0

APSC-DV-001510 - CAT II The application must be configured to use only functions, ports, and protocols permitted to it in the PPSM CAL.	0	0
APSC-DV-001520 - CAT II The application must require users to reauthenticate when organization-defined circumstances or situations require reauthentication.	0	0
APSC-DV-001530 - CAT II The application must require devices to reauthenticate when organization-defined circumstances or situations requiring reauthentication.	0	0
APSC-DV-001540 - CAT I The application must uniquely identify and authenticate organizational users (or processes acting on behalf of organizational users).	0	0
APSC-DV-001550 - CAT II The application must use multifactor (Alt. Token) authentication for network access to privileged accounts.	0	0
APSC-DV-001560 - CAT II The application must accept Personal Identity Verification (PIV) credentials.	0	0
APSC-DV-001580 - CAT II The application must use multifactor (e.g., CAC, Alt. Token) authentication for network access to non-privileged accounts.	0	0
APSC-DV-001590 - CAT II The application must use multifactor (Alt. Token) authentication for local access to privileged accounts.	0	0
APSC-DV-001600 - CAT II The application must use multifactor (e.g., CAC, Alt. Token) authentication for local access to non-privileged accounts.	0	0
APSC-DV-001610 - CAT II The application must ensure users are authenticated with an individual authenticator prior to using a group authenticator.	0	0
APSC-DV-001620 - CAT II The application must implement replay-resistant authentication mechanisms for network access to privileged accounts.	0	0
APSC-DV-001630 - CAT II The application must implement replay-resistant authentication mechanisms for network access to non-privileged accounts.	0	0
APSC-DV-001640 - CAT II The application must utilize mutual authentication when endpoint device non-repudiation protections are required by DoD policy or by the data owner.	0	0
APSC-DV-001650 - CAT II The application must authenticate all network connected endpoint devices before establishing any connection.	0	0
APSC-DV-001660 - CAT II Service-Oriented Applications handling non-releasable data must authenticate endpoint devices via mutual SSL/TLS.	0	0
APSC-DV-001670 - CAT II The application must disable device identifiers after 35 days of inactivity unless a cryptographic certificate is used for authentication.	0	0
APSC-DV-001680 - CAT I The application must enforce a minimum 15-character password length.	0	0
APSC-DV-001690 - CAT II The application must enforce password complexity by requiring that at least one upper-case character be used.	0	0
APSC-DV-001700 - CAT II The application must enforce password complexity by requiring that at least one lower-case character be used.	0	0
APSC-DV-001710 - CAT II The application must enforce password complexity by requiring that at least one numeric character be used.	0	0
APSC-DV-001720 - CAT II The application must enforce password complexity by requiring that at least one special character be used.	0	0
APSC-DV-001730 - CAT II The application must require the change of at least 8 of the total number of characters when passwords are changed.	0	0
APSC-DV-001740 - CAT I The application must only store cryptographic representations of passwords.	0	0
APSC-DV-001850 - CAT I The application must not display passwords/PINs as clear text.	0	0
APSC-DV-001750 - CAT I The application must transmit only cryptographically-protected passwords.	0	0
APSC-DV-001760 - CAT II The application must enforce 24 hours/1 day as the minimum password lifetime.	0	0
APSC-DV-001770 - CAT II The application must enforce a 60-day maximum password lifetime restriction.	0	0
APSC-DV-001780 - CAT II The application must prohibit password reuse for a minimum of five generations.	0	0

APSC-DV-001790 - CAT II The application must allow the use of a temporary password for system logons with an immediate change to a permanent password.	0	0
APSC-DV-001795 - CAT II The application password must not be changeable by users other than the administrator or the user with which the password is associated.	0	0
APSC-DV-001800 - CAT II The application must terminate existing user sessions upon account deletion.	0	0
APSC-DV-001820 - CAT I The application, when using PKI-based authentication, must enforce authorized access to the corresponding private key.	0	0
APSC-DV-001830 - CAT II The application must map the authenticated identity to the individual user or group account for PKI-based authentication.	0	0
APSC-DV-001870 - CAT II The application must uniquely identify and authenticate non-organizational users (or processes acting on behalf of non-organizational users).	0	0
APSC-DV-001810 - CAT I The application, when utilizing PKI-based authentication, must validate certificates by constructing a certification path (which includes status information) to an accepted trust anchor.	0	0
APSC-DV-001840 - CAT II The application, for PKI-based authentication, must implement a local cache of revocation data to support path discovery and validation in case of the inability to access revocation information via the network.	0	0
APSC-DV-001860 - CAT II The application must use mechanisms meeting the requirements of applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance for authentication to a cryptographic module.	0	0
APSC-DV-001880 - CAT II The application must accept Personal Identity Verification (PIV) credentials from other federal agencies.	0	0
APSC-DV-001890 - CAT II The application must electronically verify Personal Identity Verification (PIV) credentials from other federal agencies.	0	0
APSC-DV-002050 - CAT II Applications making SAML assertions must use FIPS-approved random numbers in the generation of SessionIndex in the SAML element AuthnStatement.	0	0
APSC-DV-001900 - CAT II The application must accept FICAM-approved third-party credentials.	0	0
APSC-DV-001910 - CAT II The application must conform to FICAM-issued profiles.	0	0
APSC-DV-001930 - CAT II Applications used for non-local maintenance sessions must audit non-local maintenance and diagnostic sessions for organization-defined auditable events.	0	0
APSC-DV-000310 - CAT III The application must have a process, feature or function that prevents removal or disabling of emergency accounts.	0	0
APSC-DV-001940 - CAT II Applications used for non-local maintenance sessions must implement cryptographic mechanisms to protect the integrity of non-local maintenance and diagnostic communications.	0	0
APSC-DV-001950 - CAT II Applications used for non-local maintenance sessions must implement cryptographic mechanisms to protect the confidentiality of non-local maintenance and diagnostic communications.	0	0
APSC-DV-001960 - CAT II Applications used for non-local maintenance sessions must verify remote disconnection at the termination of non-local maintenance and diagnostic sessions.	0	0
APSC-DV-001970 - CAT II The application must employ strong authenticators in the establishment of non-local maintenance and diagnostic sessions.	0	0
APSC-DV-001980 - CAT II The application must terminate all sessions and network connections when non-local maintenance is completed.	0	0
APSC-DV-001995 - CAT II The application must not be vulnerable to race conditions.	0	0
APSC-DV-002000 - CAT II The application must terminate all network connections associated with a communications session at the end of the session.	0	0
APSC-DV-002010 - CAT II The application must implement NSA-approved cryptography to protect classified information in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, and standards.	0	0
APSC-DV-002020 - CAT II The application must utilize FIPS-validated cryptographic modules when signing application components.	0	0
APSC-DV-002030 - CAT II The application must utilize FIPS-validated cryptographic modules when generating cryptographic hashes.	0	0

APSC-DV-002040 - CAT II The application must utilize FIPS-validated cryptographic modules when protecting unclassified information that requires cryptographic protection.	0	0
APSC-DV-002150 - CAT II The application user interface must be either physically or logically separated from data storage and management interfaces.	0	0
APSC-DV-002210 - CAT II The application must set the HTTPOnly flag on session cookies.	0	0
APSC-DV-002220 - CAT II The application must set the secure flag on session cookies.	0	0
APSC-DV-002230 - CAT I The application must not expose session IDs.	0	0
APSC-DV-002240 - CAT I The application must destroy the session ID value and/or cookie on logoff or browser close.	0	0
APSC-DV-002250 - CAT II Applications must use system-generated session identifiers that protect against session fixation.	0	0
APSC-DV-002260 - CAT II Applications must validate session identifiers.	0	0
APSC-DV-002270 - CAT II Applications must not use URL embedded session IDs.	0	0
APSC-DV-002280 - CAT II The application must not re-use or recycle session IDs.	0	0
APSC-DV-002290 - CAT II The application must use the Federal Information Processing Standard (FIPS) 140-2-validated cryptographic modules and random number generator if the application implements encryption, key exchange, digital signature, and hash functionality.	0	0
APSC-DV-002300 - CAT II The application must only allow the use of DoD-approved certificate authorities for verification of the establishment of protected sessions.	0	0
APSC-DV-002310 - CAT I The application must fail to a secure state if system initialization fails, shutdown fails, or aborts fail.	0	0
APSC-DV-002320 - CAT II In the event of a system failure, applications must preserve any information necessary to determine cause of failure and any information necessary to return to operations with least disruption to mission processes.	0	0
APSC-DV-002330 - CAT II The application must protect the confidentiality and integrity of stored information when required by DoD policy or the information owner.	0	0
APSC-DV-002340 - CAT II The application must implement approved cryptographic mechanisms to prevent unauthorized modification of organization-defined information at rest on organization-defined information system components.	0	0
APSC-DV-002350 - CAT II The application must use appropriate cryptography in order to protect stored DoD information when required by the information owner or DoD policy.	0	0
APSC-DV-002360 - CAT II The application must isolate security functions from non-security functions.	0	0
APSC-DV-002370 - CAT II The application must maintain a separate execution domain for each executing process.	0	0
APSC-DV-002380 - CAT II Applications must prevent unauthorized and unintended information transfer via shared system resources.	0	0
APSC-DV-002390 - CAT II XML-based applications must mitigate DoS attacks by using XML filters, parser options, or gateways.	0	0
APSC-DV-002400 - CAT II The application must restrict the ability to launch Denial of Service (DoS) attacks against itself or other information systems.	0	0
APSC-DV-002410 - CAT II The web service design must include redundancy mechanisms when used with high-availability systems.	0	0
APSC-DV-002420 - CAT II An XML firewall function must be deployed to protect web services when exposed to untrusted networks.	0	0
APSC-DV-002610 - CAT II The application must remove organization-defined software components after updated versions have been installed.	0	0
APSC-DV-002440 - CAT I The application must protect the confidentiality and integrity of transmitted information.	0	0
APSC-DV-002450 - CAT II The application must implement cryptographic mechanisms to prevent unauthorized disclosure of information and/or detect changes to information during transmission unless otherwise protected by alternative physical safeguards, such as, at a minimum, a Prot	0	0
APSC-DV-002460 - CAT II The application must maintain the confidentiality and integrity of	0	0

information during preparation for transmission.		
APSC-DV-002470 - CAT II The application must maintain the confidentiality and integrity of information during reception.	0	0
APSC-DV-002480 - CAT II The application must not disclose unnecessary information to users.	0	0
APSC-DV-002485 - CAT I The application must not store sensitive information in hidden fields.	0	0
APSC-DV-002490 - CAT I The application must protect from Cross-Site Scripting (XSS) vulnerabilities.	0	0
APSC-DV-002500 - CAT II The application must protect from Cross-Site Request Forgery (CSRF) vulnerabilities.	0	0
APSC-DV-002510 - CAT I The application must protect from command injection.	0	0
APSC-DV-002520 - CAT II The application must protect from canonical representation vulnerabilities.	0	0
APSC-DV-002530 - CAT II The application must validate all input.	0	0
APSC-DV-002540 - CAT I The application must not be vulnerable to SQL Injection.	0	0
APSC-DV-002550 - CAT I The application must not be vulnerable to XML-oriented attacks.	0	0
APSC-DV-002560 - CAT I The application must not be subject to input handling vulnerabilities.	0	0
APSC-DV-002570 - CAT II The application must generate error messages that provide information necessary for corrective actions without revealing information that could be exploited by adversaries.	0	0
APSC-DV-002580 - CAT II The application must reveal error messages only to the ISSO, ISSM, or SA.	0	0
APSC-DV-002590 - CAT I The application must not be vulnerable to overflow attacks.	0	0
APSC-DV-002630 - CAT II Security-relevant software updates and patches must be kept up to date.	0	0
APSC-DV-002760 - CAT II The application performing organization-defined security functions must verify correct operation of security functions.	0	0
APSC-DV-002900 - CAT II The ISSO must ensure application audit trails are retained for at least 1 year for applications without SAMI data, and 5 years for applications including SAMI data.	0	0
APSC-DV-002770 - CAT II The application must perform verification of the correct operation of security functions: upon system startup and/or restart; upon command by a user with privileged access; and/or every 30 days.	0	0
APSC-DV-002780 - CAT III The application must notify the ISSO and ISSM of failed security verification tests.	0	0
APSC-DV-002870 - CAT II Unsigned Category 1A mobile code must not be used in the application in accordance with DoD policy.	0	0
APSC-DV-002880 - CAT II The ISSO must ensure an account management process is implemented, verifying only authorized users can gain access to the application, and individual accounts designated as inactive, suspended, or terminated are promptly removed.	0	0
APSC-DV-002890 - CAT I Application web servers must be on a separate network segment from the application and database servers if it is a tiered application operating in the DoD DMZ.	0	0
APSC-DV-002910 - CAT II The ISSO must review audit trails periodically based on system documentation recommendations or immediately upon system security events.	0	0
APSC-DV-002920 - CAT II The ISSO must report all suspected violations of IA policies in accordance with DoD information system IA procedures.	0	0
APSC-DV-002930 - CAT II The ISSO must ensure active vulnerability testing is performed.	0	0
APSC-DV-002980 - CAT II New IP addresses, data services, and associated ports used by the application must be submitted to the appropriate approving authority for the organization, which in turn will be submitted through the DoD Ports, Protocols, and Services Management (DoD PPS	0	0
APSC-DV-002950 - CAT II Execution flow diagrams and design documents must be created	0	0

to show how deadlock and recursion issues in web services are being mitigated.		
APSC-DV-002960 - CAT II The designer must ensure the application does not store configuration and control files in the same directory as user data.	0	0
APSC-DV-002970 - CAT II The ISSO must ensure if a DoD STIG or NSA guide is not available, a third-party product will be configured by following available guidance.	0	0
APSC-DV-002990 - CAT II The application must be registered with the DoD Ports and Protocols Database.	0	0
APSC-DV-002995 - CAT II The Configuration Management (CM) repository must be properly patched and STIG compliant.	0	0
APSC-DV-003000 - CAT II Access privileges to the Configuration Management (CM) repository must be reviewed every three months.	0	0
APSC-DV-003010 - CAT II A Software Configuration Management (SCM) plan describing the configuration control and change management process of application objects developed by the organization and the roles and responsibilities of the organization must be created and maintained.	0	0
APSC-DV-003020 - CAT II A Configuration Control Board (CCB) that meets at least every release cycle, for managing the Configuration Management (CM) process must be established.	0	0
APSC-DV-003030 - CAT II The application services and interfaces must be compatible with and ready for IPv6 networks.	0	0
APSC-DV-003040 - CAT II The application must not be hosted on a general purpose machine if the application is designated as critical or high availability by the ISSO.	0	0
APSC-DV-003050 - CAT II A disaster recovery/continuity plan must exist in accordance with DoD policy based on the applications availability requirements.	0	0
APSC-DV-003060 - CAT II Recovery procedures and technical system features must exist so recovery is performed in a secure and verifiable manner. The ISSO will document circumstances inhibiting a trusted recovery.	0	0
APSC-DV-003070 - CAT II Data backup must be performed at required intervals in accordance with DoD policy.	0	0
APSC-DV-003080 - CAT II Back-up copies of the application software or source code must be stored in a fire-rated container or stored separately (offsite).	0	0
APSC-DV-003090 - CAT II Procedures must be in place to assure the appropriate physical and technical protection of the backup and restoration of the application.	0	0
APSC-DV-003100 - CAT II The application must use encryption to implement key exchange and authenticate endpoints prior to establishing a communication channel for key exchange.	0	0
APSC-DV-003110 - CAT I The application must not contain embedded authentication data.	0	0
APSC-DV-003120 - CAT I The application must have the capability to mark sensitive/classified output when required.	0	0
APSC-DV-003130 - CAT III Prior to each release of the application, updates to system, or applying patches; tests plans and procedures must be created and executed.	0	0
APSC-DV-003150 - CAT II At least one tester must be designated to test for security flaws in addition to functional testing.	0	0
APSC-DV-003140 - CAT II Application files must be cryptographically hashed prior to deploying to DoD operational networks.	0	0
APSC-DV-003160 - CAT III Test procedures must be created and at least annually executed to ensure system initialization, shutdown, and aborts are configured to verify the system remains in a secure state.	0	0
APSC-DV-003170 - CAT II An application code review must be performed on the application.	0	0
APSC-DV-003180 - CAT III Code coverage statistics must be maintained for each release of the application.	0	0
APSC-DV-003190 - CAT II Flaws found during a code review must be tracked in a defect tracking system.	0	0
APSC-DV-003200 - CAT II The changes to the application must be assessed for IA and accreditation impact prior to implementation.	0	0
APSC-DV-003210 - CAT II Security flaws must be fixed or addressed in the project plan.	0	0

APSC-DV-003215 - CAT III The application development team must follow a set of coding standards.	0	0
APSC-DV-003220 - CAT III The designer must create and update the Design Document for each release of the application.	0	0
APSC-DV-003230 - CAT II Threat models must be documented and reviewed for each application release and updated as required by design and functionality changes or when new threats are discovered.	0	0
APSC-DV-003235 - CAT II The application must not be subject to error handling vulnerabilities.	0	0
APSC-DV-003250 - CAT I The application must be decommissioned when maintenance or support is no longer available.	0	0
APSC-DV-003236 - CAT II The application development team must provide an application incident response plan.	0	0
APSC-DV-003240 - CAT I All products must be supported by the vendor or the development team.	0	0
APSC-DV-003260 - CAT III Procedures must be in place to notify users when an application is decommissioned.	0	0
APSC-DV-003270 - CAT II Unnecessary built-in application accounts must be disabled.	0	0
APSC-DV-003280 - CAT I Default passwords must be changed.	0	0
APSC-DV-003330 - CAT II The system must alert an administrator when low resource conditions are encountered.	0	0
APSC-DV-003285 - CAT II An Application Configuration Guide must be created and included with the application.	0	0
APSC-DV-003290 - CAT II If the application contains classified data, a Security Classification Guide must exist containing data elements and their classification.	0	0
APSC-DV-003300 - CAT II The designer must ensure uncategorized or emerging mobile code is not used in applications.	0	0
APSC-DV-003310 - CAT II Production database exports must have database administration credentials and sensitive data removed before releasing the export.	0	0
APSC-DV-003320 - CAT II Protections against DoS attacks must be implemented.	0	0
APSC-DV-003340 - CAT III At least one application administrator must be registered to receive update notifications, or security alerts, when automated alerts are available.	0	0
APSC-DV-003360 - CAT III The application must generate audit records when concurrent logons from different workstations occur.	0	0
APSC-DV-003345 - CAT III The application must provide notifications or alerts when product update and security related patches are available.	0	0
APSC-DV-003350 - CAT II Connections between the DoD enclave and the Internet or other public or commercial wide area networks must require a DMZ.	0	0
APSC-DV-003400 - CAT II The Program Manager must verify all levels of program management, designers, developers, and testers receive annual security training pertaining to their job function.	0	0
APSC-DV-000010 - CAT II The application must provide a capability to limit the number of logon sessions per user.	0	0
APSC-DV-000060 - CAT II The application must clear temporary storage and cookies when the session is terminated.	0	0
APSC-DV-000070 - CAT II The application must automatically terminate the non-privileged user session and log off non-privileged users after a 15 minute idle time period has elapsed.	0	0
APSC-DV-000080 - CAT II The application must automatically terminate the admin user session and log off admin users after a 10 minute idle time period is exceeded.	0	0
APSC-DV-000090 - CAT II Applications requiring user access authentication must provide a logoff capability for user initiated communication session.	0	0
APSC-DV-000100 - CAT III The application must display an explicit logoff message to users indicating the reliable termination of authenticated communications sessions.	0	0
APSC-DV-000110 - CAT II The application must associate organization-defined types of security attributes having organization-defined security attribute values with information in	0	0

storage.		
APSC-DV-000120 - CAT II The application must associate organization-defined types of security attributes having organization-defined security attribute values with information in process.	0	0
APSC-DV-000130 - CAT II The application must associate organization-defined types of security attributes having organization-defined security attribute values with information in transmission.	0	0
APSC-DV-000160 - CAT II The application must implement DoD-approved encryption to protect the confidentiality of remote access sessions.	0	0
APSC-DV-000170 - CAT II The application must implement cryptographic mechanisms to protect the integrity of remote access sessions.	0	0
APSC-DV-000190 - CAT I Messages protected with WS_Security must use time stamps with creation and expiration times.	0	0
APSC-DV-000180 - CAT II Applications with SOAP messages requiring integrity must include the following message elements:-Message ID-Service Request-Timestamp-SAML Assertion (optionally included in messages) and all elements of the message must be digitally signed.	0	0
APSC-DV-000200 - CAT I Validity periods must be verified on all application messages using WS-Security or SAML assertions.	0	0
APSC-DV-000210 - CAT II The application must ensure each unique asserting party provides unique assertion ID references for each SAML assertion.	0	0
APSC-DV-000220 - CAT II The application must ensure encrypted assertions, or equivalent confidentiality protections are used when assertion data is passed through an intermediary, and confidentiality of the assertion data is required when passing through the intermediary.	0	0
APSC-DV-000230 - CAT I The application must use the NotOnOrAfter condition when using the SubjectConfirmation element in a SAML assertion.	0	0
APSC-DV-000240 - CAT I The application must use both the NotBefore and NotOnOrAfter elements or OneTimeUse element when using the Conditions element in a SAML assertion.	0	0
APSC-DV-000250 - CAT II The application must ensure if a OneTimeUse element is used in an assertion, there is only one of the same used in the Conditions element portion of an assertion.	0	0
APSC-DV-000260 - CAT II The application must ensure messages are encrypted when the SessionIndex is tied to privacy data.	0	0
APSC-DV-000290 - CAT II Shared/group account credentials must be terminated when members leave the group.	0	0
APSC-DV-000280 - CAT II The application must provide automated mechanisms for supporting account management functions.	0	0
APSC-DV-000300 - CAT II The application must automatically remove or disable temporary user accounts 72 hours after account creation.	0	0
APSC-DV-000320 - CAT III The application must automatically disable accounts after a 35 day period of account inactivity.	0	0
APSC-DV-000330 - CAT II Unnecessary application accounts must be disabled, or deleted.	0	0
APSC-DV-000420 - CAT II The application must automatically audit account enabling actions.	0	0
APSC-DV-000340 - CAT II The application must automatically audit account creation.	0	0
APSC-DV-000350 - CAT II The application must automatically audit account modification.	0	0
APSC-DV-000360 - CAT II The application must automatically audit account disabling actions.	0	0
APSC-DV-000370 - CAT II The application must automatically audit account removal actions.	0	0
APSC-DV-000380 - CAT III The application must notify System Administrators and Information System Security Officers when accounts are created.	0	0
APSC-DV-000390 - CAT III The application must notify System Administrators and Information System Security Officers when accounts are modified.	0	0
APSC-DV-000400 - CAT III The application must notify System Administrators and Information System Security Officers of account disabling actions.	0	0

APSC-DV-000410 - CAT III The application must notify System Administrators and Information System Security Officers of account removal actions.	0	0
APSC-DV-000430 - CAT III The application must notify System Administrators and Information System Security Officers of account enabling actions.	0	0
APSC-DV-000440 - CAT II Application data protection requirements must be identified and documented.	0	0
APSC-DV-000520 - CAT II The application must audit the execution of privileged functions.	0	0
APSC-DV-000450 - CAT II The application must utilize organization-defined data mining detection techniques for organization-defined data storage objects to adequately detect data mining attempts.	0	0
APSC-DV-000460 - CAT I The application must enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.	0	0
APSC-DV-000470 - CAT II The application must enforce organization-defined discretionary access control policies over defined subjects and objects.	0	0
APSC-DV-000480 - CAT II The application must enforce approved authorizations for controlling the flow of information within the system based on organization-defined information flow control policies.	0	0
APSC-DV-000490 - CAT II The application must enforce approved authorizations for controlling the flow of information between interconnected systems based on organization-defined information flow control policies.	0	0
APSC-DV-000500 - CAT II The application must prevent non-privileged users from executing privileged functions to include disabling, circumventing, or altering implemented security safeguards/countermeasures.	0	0
APSC-DV-000510 - CAT I The application must execute without excessive account permissions.	0	0
APSC-DV-000530 - CAT I The application must enforce the limit of three consecutive invalid logon attempts by a user during a 15 minute time period.	0	0
APSC-DV-000560 - CAT III The application must retain the Standard Mandatory DoD Notice and Consent Banner on the screen until users acknowledge the usage conditions and take explicit actions to log on for further access.	0	0
APSC-DV-000540 - CAT II The application administrator must follow an approved process to unlock locked user accounts.	0	0
APSC-DV-000550 - CAT III The application must display the Standard Mandatory DoD Notice and Consent Banner before granting access to the application.	0	0
APSC-DV-000570 - CAT III The publicly accessible application must display the Standard Mandatory DoD Notice and Consent Banner before granting access to the application.	0	0
APSC-DV-000580 - CAT III The application must display the time and date of the users last successful logon.	0	0
APSC-DV-000630 - CAT II The application must provide audit record generation capability for the destruction of session IDs.	0	0
APSC-DV-000590 - CAT II The application must protect against an individual (or process acting on behalf of an individual) falsely denying having performed organization-defined actions to be covered by non-repudiation.	0	0
APSC-DV-000600 - CAT II For applications providing audit record aggregation, the application must compile audit records from organization-defined information system components into a system-wide audit trail that is time-correlated with an organization-defined level of tolerance	0	0
APSC-DV-000610 - CAT II The application must provide the capability for organization-identified individuals or roles to change the auditing to be performed on all application components, based on all selectable event criteria within organization-defined time thresholds.	0	0
APSC-DV-000620 - CAT II The application must provide audit record generation capability for the creation of session IDs.	0	0

Scan Summary - ASD STIG 6.1

Category	Issues Found	Best Fix Locations
APSC-DV-000640 - CAT II The application must provide audit record generation capability for the renewal of session IDs.	0	0
APSC-DV-000650 - CAT II The application must not write sensitive data into the application logs.	0	0
APSC-DV-000660 - CAT II The application must provide audit record generation capability for session timeouts.	0	0
APSC-DV-000670 - CAT II The application must record a time stamp indicating when the event occurred.	0	0
APSC-DV-000680 - CAT II The application must provide audit record generation capability for HTTP headers including User-Agent, Referer, GET, and POST.	0	0
APSC-DV-000690 - CAT II The application must provide audit record generation capability for connecting system IP addresses.	0	0
APSC-DV-000700 - CAT II The application must record the username or user ID of the user associated with the event.	0	0
APSC-DV-000710 - CAT II The application must generate audit records when successful/unsuccessful attempts to grant privileges occur.	0	0
APSC-DV-000720 - CAT II The application must generate audit records when successful/unsuccessful attempts to access security objects occur.	0	0
APSC-DV-000730 - CAT II The application must generate audit records when successful/unsuccessful attempts to access security levels occur.	0	0
APSC-DV-000740 - CAT II The application must generate audit records when successful/unsuccessful attempts to access categories of information (e.g., classification levels) occur.	0	0
APSC-DV-000750 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify privileges occur.	0	0
APSC-DV-000760 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify security objects occur.	0	0
APSC-DV-000770 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify security levels occur.	0	0
APSC-DV-000780 - CAT II The application must generate audit records when successful/unsuccessful attempts to modify categories of information (e.g., classification levels) occur.	0	0
APSC-DV-000790 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete privileges occur.	0	0
APSC-DV-000800 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete security levels occur.	0	0
APSC-DV-000810 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete application database security objects occur.	0	0
APSC-DV-000820 - CAT II The application must generate audit records when successful/unsuccessful attempts to delete categories of information (e.g., classification levels) occur.	0	0
APSC-DV-000830 - CAT II The application must generate audit records when successful/unsuccessful logon attempts occur.	0	0
APSC-DV-000840 - CAT II The application must generate audit records for privileged activities or other system-level access.	0	0
APSC-DV-000850 - CAT II The application must generate audit records showing starting and ending time for user access to the system.	0	0
APSC-DV-000860 - CAT II The application must generate audit records when successful/unsuccessful accesses to objects occur.	0	0

APSC-DV-000870 - CAT II The application must generate audit records for all direct access to the information system.	0	0
APSC-DV-000880 - CAT II The application must generate audit records for all account creations, modifications, disabling, and termination events.	0	0
APSC-DV-000910 - CAT II The application must initiate session auditing upon startup.	0	0
APSC-DV-000940 - CAT II The application must log application shutdown events.	0	0
APSC-DV-000950 - CAT II The application must log destination IP addresses.	0	0
APSC-DV-000960 - CAT II The application must log user actions involving access to data.	0	0
APSC-DV-000970 - CAT II The application must log user actions involving changes to data.	0	0
APSC-DV-000980 - CAT II The application must produce audit records containing information to establish when (date and time) the events occurred.	0	0
APSC-DV-000990 - CAT II The application must produce audit records containing enough information to establish which component, feature or function of the application triggered the audit event.	0	0
APSC-DV-001000 - CAT II When using centralized logging; the application must include a unique identifier in order to distinguish itself from other application logs.	0	0
APSC-DV-001010 - CAT II The application must produce audit records that contain information to establish the outcome of the events.	0	0
APSC-DV-001020 - CAT II The application must generate audit records containing information that establishes the identity of any individual or process associated with the event.	0	0
APSC-DV-001030 - CAT II The application must generate audit records containing the full-text recording of privileged commands or the individual identities of group account users.	0	0
APSC-DV-001040 - CAT II The application must implement transaction recovery logs when transaction based.	0	0
APSC-DV-001050 - CAT II The application must provide centralized management and configuration of the content to be captured in audit records generated by all application components.	0	0
APSC-DV-001070 - CAT II The application must off-load audit records onto a different system or media than the system being audited.	0	0
APSC-DV-001080 - CAT II The application must be configured to write application logs to a centralized log repository.	0	0
APSC-DV-001090 - CAT II The application must provide an immediate warning to the SA and ISSO (at a minimum) when allocated audit record storage volume reaches 75% of repository maximum audit record storage capacity.	0	0
APSC-DV-001100 - CAT II Applications categorized as having a moderate or high impact must provide an immediate real-time alert to the SA and ISSO (at a minimum) for all audit failure events.	0	0
APSC-DV-001110 - CAT II The application must alert the ISSO and SA (at a minimum) in the event of an audit processing failure.	0	0
APSC-DV-001120 - CAT II The application must shut down by default upon audit failure (unless availability is an overriding concern).	0	0
APSC-DV-001130 - CAT II The application must provide the capability to centrally review and analyze audit records from multiple components within the system.	0	0
APSC-DV-001140 - CAT II The application must provide the capability to filter audit records for events of interest based upon organization-defined criteria.	0	0
APSC-DV-001150 - CAT II The application must provide an audit reduction capability that supports on-demand reporting requirements.	0	0
APSC-DV-001160 - CAT II The application must provide an audit reduction capability that supports on-demand audit review and analysis.	0	0
APSC-DV-001170 - CAT II The application must provide an audit reduction capability that supports after-the-fact investigations of security incidents.	0	0
APSC-DV-001180 - CAT II The application must provide a report generation capability that supports on-demand audit review and analysis.	0	0
APSC-DV-001190 - CAT II The application must provide a report generation capability that	0	0

supports on-demand reporting requirements.		
APSC-DV-001200 - CAT II The application must provide a report generation capability that supports after-the-fact investigations of security incidents.	0	0
APSC-DV-001210 - CAT II The application must provide an audit reduction capability that does not alter original content or time ordering of audit records.	0	0
APSC-DV-001220 - CAT II The application must provide a report generation capability that does not alter original content or time ordering of audit records.	0	0
APSC-DV-001250 - CAT II The applications must use internal system clocks to generate time stamps for audit records.	0	0
APSC-DV-001260 - CAT II The application must record time stamps for audit records that can be mapped to Coordinated Universal Time (UTC) or Greenwich Mean Time (GMT).	0	0
APSC-DV-001270 - CAT II The application must record time stamps for audit records that meet a granularity of one second for a minimum degree of precision.	0	0
APSC-DV-001280 - CAT II The application must protect audit information from any type of unauthorized read access.	0	0
APSC-DV-001290 - CAT II The application must protect audit information from unauthorized modification.	0	0
APSC-DV-001300 - CAT II The application must protect audit information from unauthorized deletion.	0	0
APSC-DV-001310 - CAT II The application must protect audit tools from unauthorized access.	0	0
APSC-DV-001320 - CAT II The application must protect audit tools from unauthorized modification.	0	0
APSC-DV-001330 - CAT II The application must protect audit tools from unauthorized deletion.	0	0
APSC-DV-001340 - CAT II The application must back up audit records at least every seven days onto a different system or system component than the system or component being audited.	0	0
APSC-DV-001570 - CAT II The application must electronically verify Personal Identity Verification (PIV) credentials.	0	0
APSC-DV-001350 - CAT II The application must use cryptographic mechanisms to protect the integrity of audit information.	0	0
APSC-DV-001360 - CAT II Application audit tools must be cryptographically hashed.	0	0
APSC-DV-001370 - CAT II The integrity of the audit tools must be validated by checking the files for changes in the cryptographic hash value.	0	0
APSC-DV-001390 - CAT II The application must prohibit user installation of software without explicit privileged status.	0	0
APSC-DV-001410 - CAT II The application must enforce access restrictions associated with changes to application configuration.	0	0
APSC-DV-001420 - CAT II The application must audit who makes configuration changes to the application.	0	0
APSC-DV-001430 - CAT II The application must have the capability to prevent the installation of patches, service packs, or application components without verification the software component has been digitally signed using a certificate that is recognized and approved by the orga	0	0
APSC-DV-001440 - CAT II The applications must limit privileges to change the software resident within software libraries.	0	0
APSC-DV-001460 - CAT II An application vulnerability assessment must be conducted.	0	0
APSC-DV-001480 - CAT II The application must prevent program execution in accordance with organization-defined policies regarding software program usage and restrictions, and/or rules authorizing the terms and conditions of software program usage.	0	0
APSC-DV-001490 - CAT II The application must employ a deny-all, permit-by-exception (whitelist) policy to allow the execution of authorized software programs.	0	0
APSC-DV-001500 - CAT II The application must be configured to disable non-essential capabilities.	0	0

APSC-DV-001510 - CAT II The application must be configured to use only functions, ports, and protocols permitted to it in the PPSM CAL.	0	0
APSC-DV-001520 - CAT II The application must require users to reauthenticate when organization-defined circumstances or situations require reauthentication.	0	0
APSC-DV-001530 - CAT II The application must require devices to reauthenticate when organization-defined circumstances or situations requiring reauthentication.	0	0
APSC-DV-001540 - CAT I The application must uniquely identify and authenticate organizational users (or processes acting on behalf of organizational users).	0	0
APSC-DV-001550 - CAT II The application must use multifactor (Alt. Token) authentication for network access to privileged accounts.	0	0
APSC-DV-001560 - CAT II The application must accept Personal Identity Verification (PIV) credentials.	0	0
APSC-DV-001580 - CAT II The application must use multifactor (e.g., CAC, Alt. Token) authentication for network access to non-privileged accounts.	0	0
APSC-DV-001590 - CAT II The application must use multifactor (Alt. Token) authentication for local access to privileged accounts.	0	0
APSC-DV-001600 - CAT II The application must use multifactor (e.g., CAC, Alt. Token) authentication for local access to non-privileged accounts.	0	0
APSC-DV-001610 - CAT II The application must ensure users are authenticated with an individual authenticator prior to using a group authenticator.	0	0
APSC-DV-001620 - CAT II The application must implement replay-resistant authentication mechanisms for network access to privileged accounts.	0	0
APSC-DV-001630 - CAT II The application must implement replay-resistant authentication mechanisms for network access to non-privileged accounts.	0	0
APSC-DV-001640 - CAT II The application must utilize mutual authentication when endpoint device non-repudiation protections are required by DoD policy or by the data owner.	0	0
APSC-DV-001650 - CAT II The application must authenticate all network connected endpoint devices before establishing any connection.	0	0
APSC-DV-001660 - CAT II Service-Oriented Applications handling non-releasable data must authenticate endpoint devices via mutual SSL/TLS.	0	0
APSC-DV-001670 - CAT II The application must disable device identifiers after 35 days of inactivity unless a cryptographic certificate is used for authentication.	0	0
APSC-DV-001680 - CAT I The application must enforce a minimum 15-character password length.*	0	0
APSC-DV-001690 - CAT II The application must enforce password complexity by requiring that at least one upper-case character be used.	0	0
APSC-DV-001700 - CAT II The application must enforce password complexity by requiring that at least one lower-case character be used.	0	0
APSC-DV-001710 - CAT II The application must enforce password complexity by requiring that at least one numeric character be used.	0	0
APSC-DV-001720 - CAT II The application must enforce password complexity by requiring that at least one special character be used.	0	0
APSC-DV-001730 - CAT II The application must require the change of at least 8 of the total number of characters when passwords are changed.	0	0
APSC-DV-001740 - CAT I The application must only store cryptographic representations of passwords.*	0	0
APSC-DV-001850 - CAT I The application must not display passwords/PINs as clear text.	0	0
APSC-DV-001750 - CAT I The application must transmit only cryptographically-protected passwords.	0	0
APSC-DV-001760 - CAT II The application must enforce 24 hours/1 day as the minimum password lifetime.	0	0
APSC-DV-001770 - CAT II The application must enforce a 60-day maximum password lifetime restriction.	0	0
APSC-DV-001780 - CAT II The application must prohibit password reuse for a minimum of five generations.	0	0

APSC-DV-001790 - CAT II The application must allow the use of a temporary password for system logons with an immediate change to a permanent password.	0	0
APSC-DV-001795 - CAT II The application password must not be changeable by users other than the administrator or the user with which the password is associated.	0	0
APSC-DV-001800 - CAT II The application must terminate existing user sessions upon account deletion.	0	0
APSC-DV-001820 - CAT I The application, when using PKI-based authentication, must enforce authorized access to the corresponding private key.	0	0
APSC-DV-001830 - CAT II The application must map the authenticated identity to the individual user or group account for PKI-based authentication.	0	0
APSC-DV-001870 - CAT II The application must uniquely identify and authenticate non-organizational users (or processes acting on behalf of non-organizational users).	0	0
APSC-DV-001810 - CAT I The application, when utilizing PKI-based authentication, must validate certificates by constructing a certification path (which includes status information) to an accepted trust anchor.	0	0
APSC-DV-001840 - CAT II The application, for PKI-based authentication, must implement a local cache of revocation data to support path discovery and validation in case of the inability to access revocation information via the network.	0	0
APSC-DV-001860 - CAT II The application must use mechanisms meeting the requirements of applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance for authentication to a cryptographic module.	0	0
APSC-DV-001880 - CAT II The application must accept Personal Identity Verification (PIV) credentials from other federal agencies.	0	0
APSC-DV-001890 - CAT II The application must electronically verify Personal Identity Verification (PIV) credentials from other federal agencies.	0	0
APSC-DV-002050 - CAT II Applications making SAML assertions must use FIPS-approved random numbers in the generation of SessionIndex in the SAML element AuthnStatement.	0	0
APSC-DV-001900 - CAT II The application must accept FICAM-approved third-party credentials.	0	0
APSC-DV-001910 - CAT II The application must conform to FICAM-issued profiles.	0	0
APSC-DV-001930 - CAT II Applications used for non-local maintenance sessions must audit non-local maintenance and diagnostic sessions for organization-defined auditable events.	0	0
APSC-DV-000310 - CAT III The application must have a process, feature or function that prevents removal or disabling of emergency accounts.	0	0
APSC-DV-001940 - CAT II Applications used for non-local maintenance sessions must implement cryptographic mechanisms to protect the integrity of non-local maintenance and diagnostic communications.	0	0
APSC-DV-001950 - CAT II Applications used for non-local maintenance sessions must implement cryptographic mechanisms to protect the confidentiality of non-local maintenance and diagnostic communications.	0	0
APSC-DV-001960 - CAT II Applications used for non-local maintenance sessions must verify remote disconnection at the termination of non-local maintenance and diagnostic sessions.	0	0
APSC-DV-001970 - CAT II The application must employ strong authenticators in the establishment of non-local maintenance and diagnostic sessions.	0	0
APSC-DV-001980 - CAT II The application must terminate all sessions and network connections when non-local maintenance is completed.	0	0
APSC-DV-001995 - CAT II The application must not be vulnerable to race conditions.	0	0
APSC-DV-002000 - CAT II The application must terminate all network connections associated with a communications session at the end of the session.	0	0
APSC-DV-002010 - CAT II The application must implement NSA-approved cryptography to protect classified information in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, and standards.	0	0
APSC-DV-002020 - CAT II The application must utilize FIPS-validated cryptographic modules when signing application components.	0	0
APSC-DV-002030 - CAT II The application must utilize FIPS-validated cryptographic modules when generating cryptographic hashes.	0	0

APSC-DV-002040 - CAT II The application must utilize FIPS-validated cryptographic modules when protecting unclassified information that requires cryptographic protection.	0	0
APSC-DV-002150 - CAT II The application user interface must be either physically or logically separated from data storage and management interfaces.	0	0
APSC-DV-002210 - CAT II The application must set the HTTPOnly flag on session cookies.	0	0
APSC-DV-002220 - CAT II The application must set the secure flag on session cookies.	0	0
APSC-DV-002230 - CAT I The application must not expose session IDs.	0	0
APSC-DV-002240 - CAT I The application must destroy the session ID value and/or cookie on logoff or browser close.	0	0
APSC-DV-002250 - CAT II Applications must use system-generated session identifiers that protect against session fixation.	0	0
APSC-DV-002260 - CAT II Applications must validate session identifiers.	0	0
APSC-DV-002270 - CAT II Applications must not use URL embedded session IDs.	0	0
APSC-DV-002280 - CAT II The application must not re-use or recycle session IDs.	0	0
APSC-DV-002290 - CAT II The application must use the Federal Information Processing Standard (FIPS) 140-2-validated cryptographic modules and random number generator if the application implements encryption, key exchange, digital signature, and hash functionality.*	0	0
APSC-DV-002300 - CAT II The application must only allow the use of DoD-approved certificate authorities for verification of the establishment of protected sessions.	0	0
APSC-DV-002310 - CAT I The application must fail to a secure state if system initialization fails, shutdown fails, or aborts fail.	0	0
APSC-DV-002320 - CAT II In the event of a system failure, applications must preserve any information necessary to determine cause of failure and any information necessary to return to operations with least disruption to mission processes.	0	0
APSC-DV-002330 - CAT II The application must protect the confidentiality and integrity of stored information when required by DoD policy or the information owner.	1	1
APSC-DV-002340 - CAT II The application must implement approved cryptographic mechanisms to prevent unauthorized modification of organization-defined information at rest on organization-defined information system components.	0	0
APSC-DV-002350 - CAT II The application must use appropriate cryptography in order to protect stored DoD information when required by the information owner or DoD policy.	0	0
APSC-DV-002360 - CAT II The application must isolate security functions from non-security functions.	218	38
APSC-DV-002370 - CAT II The application must maintain a separate execution domain for each executing process.	0	0
APSC-DV-002380 - CAT II Applications must prevent unauthorized and unintended information transfer via shared system resources.	0	0
APSC-DV-002390 - CAT II XML-based applications must mitigate DoS attacks by using XML filters, parser options, or gateways.	0	0
APSC-DV-002400 - CAT II The application must restrict the ability to launch Denial of Service (DoS) attacks against itself or other information systems.	0	0
APSC-DV-002410 - CAT II The web service design must include redundancy mechanisms when used with high-availability systems.	0	0
APSC-DV-002420 - CAT II An XML firewall function must be deployed to protect web services when exposed to untrusted networks.	0	0
APSC-DV-002610 - CAT II The application must remove organization-defined software components after updated versions have been installed.	0	0
APSC-DV-002440 - CAT I The application must protect the confidentiality and integrity of transmitted information.	0	0
APSC-DV-002450 - CAT II The application must implement cryptographic mechanisms to prevent unauthorized disclosure of information and/or detect changes to information during transmission unless otherwise protected by alternative physical safeguards, such as, at a minimum, a Prot	0	0
APSC-DV-002460 - CAT II The application must maintain the confidentiality and integrity of	0	0

information during preparation for transmission.		
APSC-DV-002470 - CAT II The application must maintain the confidentiality and integrity of information during reception.	0	0
APSC-DV-002480 - CAT II The application must not disclose unnecessary information to users.	0	0
APSC-DV-002485 - CAT I The application must not store sensitive information in hidden fields.	0	0
APSC-DV-002490 - CAT I The application must protect from Cross-Site Scripting (XSS) vulnerabilities.*	17	8
APSC-DV-002500 - CAT II The application must protect from Cross-Site Request Forgery (CSRF) vulnerabilities.	0	0
APSC-DV-002510 - CAT I The application must protect from command injection.	1	1
APSC-DV-002520 - CAT II The application must protect from canonical representation vulnerabilities.	0	0
APSC-DV-002530 - CAT II The application must validate all input.	0	0
APSC-DV-002540 - CAT I The application must not be vulnerable to SQL Injection.	15	2
APSC-DV-002550 - CAT I The application must not be vulnerable to XML-oriented attacks.	0	0
APSC-DV-002560 - CAT I The application must not be subject to input handling vulnerabilities.*	46	29
APSC-DV-002570 - CAT II The application must generate error messages that provide information necessary for corrective actions without revealing information that could be exploited by adversaries.	15	15
APSC-DV-002580 - CAT II The application must reveal error messages only to the ISSO, ISSM, or SA.	0	0
APSC-DV-002590 - CAT I The application must not be vulnerable to overflow attacks.	0	0
APSC-DV-002630 - CAT II Security-relevant software updates and patches must be kept up to date.	0	0
APSC-DV-002760 - CAT II The application performing organization-defined security functions must verify correct operation of security functions.	0	0
APSC-DV-002900 - CAT II The ISSO must ensure application audit trails are retained for at least 1 year for applications without SAMI data, and 5 years for applications including SAMI data.	0	0
APSC-DV-002770 - CAT II The application must perform verification of the correct operation of security functions: upon system startup and/or restart; upon command by a user with privileged access; and/or every 30 days.	0	0
APSC-DV-002780 - CAT III The application must notify the ISSO and ISSM of failed security verification tests.	0	0
APSC-DV-002870 - CAT II Unsigned Category 1A mobile code must not be used in the application in accordance with DoD policy.	0	0
APSC-DV-002880 - CAT II The ISSO must ensure an account management process is implemented, verifying only authorized users can gain access to the application, and individual accounts designated as inactive, suspended, or terminated are promptly removed.	0	0
APSC-DV-002890 - CAT I Application web servers must be on a separate network segment from the application and database servers if it is a tiered application operating in the DoD DMZ.	0	0
APSC-DV-002910 - CAT II The ISSO must review audit trails periodically based on system documentation recommendations or immediately upon system security events.	0	0
APSC-DV-002920 - CAT II The ISSO must report all suspected violations of IA policies in accordance with DoD information system IA procedures.	0	0
APSC-DV-002930 - CAT II The ISSO must ensure active vulnerability testing is performed.	0	0
APSC-DV-002980 - CAT II New IP addresses, data services, and associated ports used by the application must be submitted to the appropriate approving authority for the organization, which in turn will be submitted through the DoD Ports, Protocols, and Services Management (DoD PPS	0	0
APSC-DV-002950 - CAT II Execution flow diagrams and design documents must be created	0	0

to show how deadlock and recursion issues in web services are being mitigated.		
APSC-DV-002960 - CAT II The designer must ensure the application does not store configuration and control files in the same directory as user data.	0	0
APSC-DV-002970 - CAT II The ISSO must ensure if a DoD STIG or NSA guide is not available, a third-party product will be configured by following available guidance.	0	0
APSC-DV-002990 - CAT II The application must be registered with the DoD Ports and Protocols Database.	0	0
APSC-DV-002995 - CAT II The Configuration Management (CM) repository must be properly patched and STIG compliant.	0	0
APSC-DV-003000 - CAT II Access privileges to the Configuration Management (CM) repository must be reviewed every three months.	0	0
APSC-DV-003010 - CAT II A Software Configuration Management (SCM) plan describing the configuration control and change management process of application objects developed by the organization and the roles and responsibilities of the organization must be created and maintained.	0	0
APSC-DV-003020 - CAT II A Configuration Control Board (CCB) that meets at least every release cycle, for managing the Configuration Management (CM) process must be established.	0	0
APSC-DV-003030 - CAT II The application services and interfaces must be compatible with and ready for IPv6 networks.	0	0
APSC-DV-003040 - CAT II The application must not be hosted on a general purpose machine if the application is designated as critical or high availability by the ISSO.	0	0
APSC-DV-003050 - CAT II A disaster recovery/continuity plan must exist in accordance with DoD policy based on the applications availability requirements.	0	0
APSC-DV-003060 - CAT II Recovery procedures and technical system features must exist so recovery is performed in a secure and verifiable manner. The ISSO will document circumstances inhibiting a trusted recovery.	0	0
APSC-DV-003070 - CAT II Data backup must be performed at required intervals in accordance with DoD policy.	0	0
APSC-DV-003080 - CAT II Back-up copies of the application software or source code must be stored in a fire-rated container or stored separately (offsite).	0	0
APSC-DV-003090 - CAT II Procedures must be in place to assure the appropriate physical and technical protection of the backup and restoration of the application.	0	0
APSC-DV-003100 - CAT II The application must use encryption to implement key exchange and authenticate endpoints prior to establishing a communication channel for key exchange.	0	0
APSC-DV-003110 - CAT I The application must not contain embedded authentication data.	0	0
APSC-DV-003120 - CAT I The application must have the capability to mark sensitive/classified output when required.	0	0
APSC-DV-003130 - CAT III Prior to each release of the application, updates to system, or applying patches; tests plans and procedures must be created and executed.	0	0
APSC-DV-003150 - CAT II At least one tester must be designated to test for security flaws in addition to functional testing.	0	0
APSC-DV-003140 - CAT II Application files must be cryptographically hashed prior to deploying to DoD operational networks.	0	0
APSC-DV-003160 - CAT III Test procedures must be created and at least annually executed to ensure system initialization, shutdown, and aborts are configured to verify the system remains in a secure state.	0	0
APSC-DV-003170 - CAT II An application code review must be performed on the application.	0	0
APSC-DV-003180 - CAT III Code coverage statistics must be maintained for each release of the application.	0	0
APSC-DV-003190 - CAT II Flaws found during a code review must be tracked in a defect tracking system.	0	0
APSC-DV-003200 - CAT II The changes to the application must be assessed for IA and accreditation impact prior to implementation.	0	0
APSC-DV-003210 - CAT II Security flaws must be fixed or addressed in the project plan.	0	0

APSC-DV-003215 - CAT III The application development team must follow a set of coding standards.	0	0
APSC-DV-003220 - CAT III The designer must create and update the Design Document for each release of the application.	0	0
APSC-DV-003230 - CAT II Threat models must be documented and reviewed for each application release and updated as required by design and functionality changes or when new threats are discovered.	0	0
APSC-DV-003235 - CAT II The application must not be subject to error handling vulnerabilities.*	0	0
APSC-DV-003250 - CAT I The application must be decommissioned when maintenance or support is no longer available.	0	0
APSC-DV-003236 - CAT II The application development team must provide an application incident response plan.	0	0
APSC-DV-003240 - CAT I All products must be supported by the vendor or the development team.	0	0
APSC-DV-003260 - CAT III Procedures must be in place to notify users when an application is decommissioned.	0	0
APSC-DV-003270 - CAT II Unnecessary built-in application accounts must be disabled.	0	0
APSC-DV-003280 - CAT I Default passwords must be changed.	0	0
APSC-DV-003330 - CAT II The system must alert an administrator when low resource conditions are encountered.	0	0
APSC-DV-003285 - CAT II An Application Configuration Guide must be created and included with the application.	0	0
APSC-DV-003290 - CAT II If the application contains classified data, a Security Classification Guide must exist containing data elements and their classification.	0	0
APSC-DV-003300 - CAT II The designer must ensure uncategorized or emerging mobile code is not used in applications.	10	10
APSC-DV-003310 - CAT II Production database exports must have database administration credentials and sensitive data removed before releasing the export.	0	0
APSC-DV-003320 - CAT II Protections against DoS attacks must be implemented.	0	0
APSC-DV-003340 - CAT III At least one application administrator must be registered to receive update notifications, or security alerts, when automated alerts are available.	0	0
APSC-DV-003360 - CAT III The application must generate audit records when concurrent logons from different workstations occur.	0	0
APSC-DV-003345 - CAT III The application must provide notifications or alerts when product update and security related patches are available.	0	0
APSC-DV-003350 - CAT II Connections between the DoD enclave and the Internet or other public or commercial wide area networks must require a DMZ.	0	0
APSC-DV-003400 - CAT II The Program Manager must verify all levels of program management, designers, developers, and testers receive annual security training pertaining to their job function.	0	0
APSC-DV-000010 - CAT II The application must provide a capability to limit the number of logon sessions per user.	0	0
APSC-DV-000060 - CAT II The application must clear temporary storage and cookies when the session is terminated.	0	0
APSC-DV-000070 - CAT II The application must automatically terminate the non-privileged user session and log off non-privileged users after a 15 minute idle time period has elapsed.	0	0
APSC-DV-000080 - CAT II The application must automatically terminate the admin user session and log off admin users after a 10 minute idle time period is exceeded.	0	0
APSC-DV-000090 - CAT II Applications requiring user access authentication must provide a logoff capability for user initiated communication session.	0	0
APSC-DV-000100 - CAT III The application must display an explicit logoff message to users indicating the reliable termination of authenticated communications sessions.	0	0
APSC-DV-000110 - CAT II The application must associate organization-defined types of security attributes having organization-defined security attribute values with information in	0	0

storage.		
APSC-DV-000120 - CAT II The application must associate organization-defined types of security attributes having organization-defined security attribute values with information in process.	0	0
APSC-DV-000130 - CAT II The application must associate organization-defined types of security attributes having organization-defined security attribute values with information in transmission.	0	0
APSC-DV-000160 - CAT II The application must implement DoD-approved encryption to protect the confidentiality of remote access sessions.	0	0
APSC-DV-000170 - CAT II The application must implement cryptographic mechanisms to protect the integrity of remote access sessions.	0	0
APSC-DV-000190 - CAT I Messages protected with WS_Security must use time stamps with creation and expiration times.	0	0
APSC-DV-000180 - CAT II Applications with SOAP messages requiring integrity must include the following message elements:-Message ID-Service Request-Timestamp-SAML Assertion (optionally included in messages) and all elements of the message must be digitally signed.	0	0
APSC-DV-000200 - CAT I Validity periods must be verified on all application messages using WS-Security or SAML assertions.	0	0
APSC-DV-000210 - CAT II The application must ensure each unique asserting party provides unique assertion ID references for each SAML assertion.	0	0
APSC-DV-000220 - CAT II The application must ensure encrypted assertions, or equivalent confidentiality protections are used when assertion data is passed through an intermediary, and confidentiality of the assertion data is required when passing through the intermediary.	0	0
APSC-DV-000230 - CAT I The application must use the NotOnOrAfter condition when using the SubjectConfirmation element in a SAML assertion.	0	0
APSC-DV-000240 - CAT I The application must use both the NotBefore and NotOnOrAfter elements or OneTimeUse element when using the Conditions element in a SAML assertion.	0	0
APSC-DV-000250 - CAT II The application must ensure if a OneTimeUse element is used in an assertion, there is only one of the same used in the Conditions element portion of an assertion.	0	0
APSC-DV-000260 - CAT II The application must ensure messages are encrypted when the SessionIndex is tied to privacy data.	0	0
APSC-DV-000290 - CAT II Shared/group account credentials must be terminated when members leave the group.	0	0
APSC-DV-000280 - CAT II The application must provide automated mechanisms for supporting account management functions.	0	0
APSC-DV-000300 - CAT II The application must automatically remove or disable temporary user accounts 72 hours after account creation.	0	0
APSC-DV-000320 - CAT III The application must automatically disable accounts after a 35 day period of account inactivity.	0	0
APSC-DV-000330 - CAT II Unnecessary application accounts must be disabled, or deleted.	0	0
APSC-DV-000420 - CAT II The application must automatically audit account enabling actions.	0	0
APSC-DV-000340 - CAT II The application must automatically audit account creation.	0	0
APSC-DV-000350 - CAT II The application must automatically audit account modification.	0	0
APSC-DV-000360 - CAT II The application must automatically audit account disabling actions.	0	0
APSC-DV-000370 - CAT II The application must automatically audit account removal actions.	0	0
APSC-DV-000380 - CAT III The application must notify System Administrators and Information System Security Officers when accounts are created.	0	0
APSC-DV-000390 - CAT III The application must notify System Administrators and Information System Security Officers when accounts are modified.	0	0
APSC-DV-000400 - CAT III The application must notify System Administrators and Information System Security Officers of account disabling actions.	0	0

APSC-DV-000410 - CAT III The application must notify System Administrators and Information System Security Officers of account removal actions.	0	0
APSC-DV-000430 - CAT III The application must notify System Administrators and Information System Security Officers of account enabling actions.	0	0
APSC-DV-000440 - CAT II Application data protection requirements must be identified and documented.	0	0
APSC-DV-000520 - CAT II The application must audit the execution of privileged functions.	0	0
APSC-DV-000450 - CAT II The application must utilize organization-defined data mining detection techniques for organization-defined data storage objects to adequately detect data mining attempts.	0	0
APSC-DV-000460 - CAT I The application must enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.	0	0
APSC-DV-000470 - CAT II The application must enforce organization-defined discretionary access control policies over defined subjects and objects.	0	0
APSC-DV-000480 - CAT II The application must enforce approved authorizations for controlling the flow of information within the system based on organization-defined information flow control policies.	0	0
APSC-DV-000490 - CAT II The application must enforce approved authorizations for controlling the flow of information between interconnected systems based on organization-defined information flow control policies.	0	0
APSC-DV-000500 - CAT II The application must prevent non-privileged users from executing privileged functions to include disabling, circumventing, or altering implemented security safeguards/countermeasures.	0	0
APSC-DV-000510 - CAT I The application must execute without excessive account permissions.	0	0
APSC-DV-000530 - CAT I The application must enforce the limit of three consecutive invalid logon attempts by a user during a 15 minute time period.	0	0
APSC-DV-000560 - CAT III The application must retain the Standard Mandatory DoD Notice and Consent Banner on the screen until users acknowledge the usage conditions and take explicit actions to log on for further access.	0	0
APSC-DV-000540 - CAT II The application administrator must follow an approved process to unlock locked user accounts.	0	0
APSC-DV-000550 - CAT III The application must display the Standard Mandatory DoD Notice and Consent Banner before granting access to the application.	0	0
APSC-DV-000570 - CAT III The publicly accessible application must display the Standard Mandatory DoD Notice and Consent Banner before granting access to the application.	0	0
APSC-DV-000580 - CAT III The application must display the time and date of the users last successful logon.	0	0
APSC-DV-000630 - CAT II The application must provide audit record generation capability for the destruction of session IDs.	0	0
APSC-DV-000590 - CAT II The application must protect against an individual (or process acting on behalf of an individual) falsely denying having performed organization-defined actions to be covered by non-repudiation.	0	0
APSC-DV-000600 - CAT II For applications providing audit record aggregation, the application must compile audit records from organization-defined information system components into a system-wide audit trail that is time-correlated with an organization-defined level of tolerance	0	0
APSC-DV-000610 - CAT II The application must provide the capability for organization-identified individuals or roles to change the auditing to be performed on all application components, based on all selectable event criteria within organization-defined time thresholds.	0	0
APSC-DV-000620 - CAT II The application must provide audit record generation capability for the creation of session IDs.	0	0

***** Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - OWASP Top 10 API 2023

Category	Issues Found	Best Fix Locations
API1-Broken Object Level Authorization	3	2
API2-Broken Authentication*	0	0
API3-Broken Object Property Level Authorization	0	0
API4-Unrestricted Resource Consumption	20	15
API5-Broken Function Level Authorization	0	0
API6-Unrestricted Access to Sensitive Business Flows	0	0
API7-Server Side Request Forgery	0	0
API8-Security Misconfiguration	16	16
API9-Improper Inventory Management	0	0
API10-Unsafe Consumption of APIs	17	9

***** Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - MOIS(KISA) Secure Coding 2021

Category	Issues Found	Best Fix Locations
MOIS(KISA) API misuse*	0	0
MOIS(KISA) Code error	0	0
MOIS(KISA) Encapsulation	0	0
MOIS(KISA) Error processing	15	15
MOIS(KISA) Security Functions*	4	3
MOIS(KISA) Time and status	0	0
MOIS(KISA) Verification and representation of input data*	69	37

***** Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - SANS top 25

Category	Issues Found	Best Fix Locations
SANS top 25*****	65	41

***** Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - CWE top 25

Category	Issues Found	Best Fix Locations
CWE top 25*	53	29

* Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - Top Tier

Category	Issues Found	Best Fix Locations
Top Tier	35	13

Scan Summary - OWASP ASVS

Category	Issues Found	Best Fix Locations
V01 Architecture, Design and Threat Modeling	3	2
V02 Authentication*	1	1
V03 Session Management	0	0
V04 Access Control	0	0
V05 Validation, Sanitization and Encoding*	68	36
V06 Stored Cryptography*	0	0
V07 Error Handling and Logging*	6	3
V08 Data Protection	0	0
V09 Communication	0	0
V10 Malicious Code*	0	0
V11 Business Logic	0	0
V12 Files and Resources	0	0
V13 API and Web Service	0	0
V14 Configuration	19	19

***** Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - ASA Mobile Premium

Category	Issues Found	Best Fix Locations
ASA Mobile Premium*	0	0

* Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - ASA Premium

Category	Issues Found	Best Fix Locations
ASA Premium*	312	93

* Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - Base Preset

Category	Issues Found	Best Fix Locations
Base Preset	19	5

Scan Summary - OWASP Top 10 2013

Further details and elaboration about vulnerabilities and risks can be found at: OWASP Top 10 2013

Category	Threat Agent	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	AVERAGE	SEVERE	ALL DATA	234	41
A2-Broken Authentication and Session Management*	EXTERNAL, INTERNAL USERS	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	AFFECTED DATA AND FUNCTIONS	1	1
A3-Cross-Site Scripting (XSS)*	EXTERNAL, INTERNAL, ADMIN USERS	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	AFFECTED DATA AND SYSTEM	19	10
A4-Insecure Direct Object References*	SYSTEM USERS	EASY	COMMON	EASY	MODERATE	EXPOSED DATA	3	2
A5-Security Misconfiguration	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	EASY	MODERATE	ALL DATA AND SYSTEM	15	15
A6-Sensitive Data Exposure*	EXTERNAL, INTERNAL, ADMIN USERS, USERS BROWSERS	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	EXPOSED DATA	0	0
A7-Missing Function Level Access Control	EXTERNAL, INTERNAL USERS	EASY	COMMON	AVERAGE	MODERATE	EXPOSED DATA AND FUNCTIONS	0	0
A8-Cross-Site Request Forgery (CSRF)	USERS BROWSERS	AVERAGE	COMMON	EASY	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0
A9-Using Components with Known Vulnerabilities*	EXTERNAL USERS, AUTOMATED TOOLS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	17	9

* Please note, the report only includes the presets/filters you applied to the scan results.

Scan Summary - OWASP Top 10 API

Category	Issues Found	Best Fix Locations
API1-Broken Object Level Authorization	0	0
API2-Broken Authentication	0	0
API3-Excessive Data Exposure	0	0
API4-Lack of Resources and Rate Limiting	0	0
API5-Broken Function Level Authorization	0	0
API6-Mass Assignment	0	0
API7-Security Misconfiguration	0	0
API8-Injection	0	0
API9-Improper Assets Management	0	0
API10-Insufficient Logging and Monitoring	0	0

Scan Summary - OWASP Top 10 2010

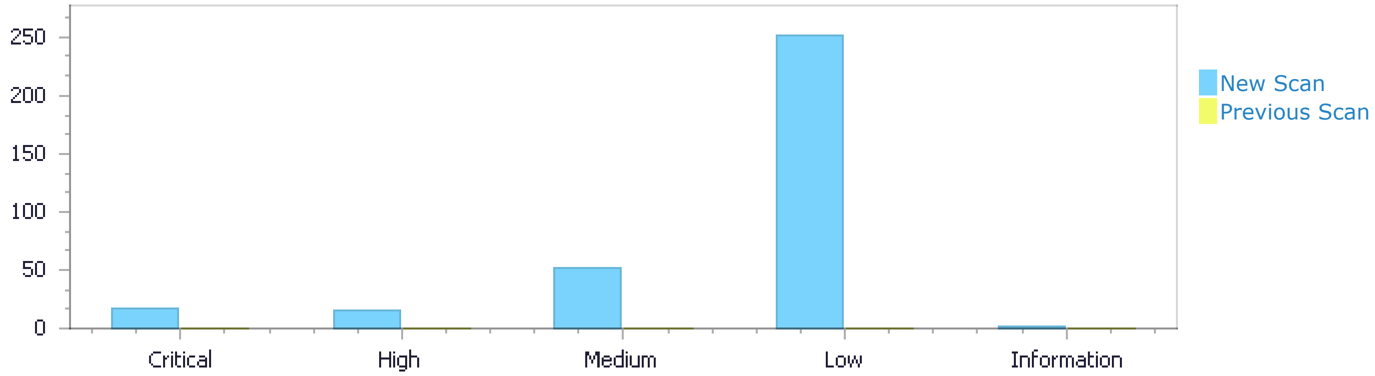
Category	Issues Found	Best Fix Locations
A1-Injection	0	0
A2-Cross-Site Scripting (XSS)*	0	0
A3-Broken Authentication and Session Management	0	0
A4-Insecure Direct Object References	0	0
A5-Cross-Site Request Forgery (CSRF)	0	0
A6-Security Misconfiguration	0	0
A7-Insecure Cryptographic Storage*	0	0
A8-Failure to Restrict URL Access	0	0
A9-Insufficient Transport Layer Protection	0	0
A10-Unvalidated Redirects and Forwards	0	0

* Please note, the report only includes the presets/filters you applied to the scan results.

Results Distribution By Status

First scan of the project

	Critical	High	Medium	Low	Information	Total
New Issues	18	15	51	252	2	338
Recurrent Issues	0	0	0	0	0	0
Total	18	15	51	252	2	338
Fixed Issues	0	0	0	0	0	0



Results Distribution By State

	Critical	High	Medium	Low	Information	Total
To Verify	18	15	51	252	2	338
Not Exploitable	0	0	0	0	0	0
Confirmed	0	0	0	0	0	0
Urgent	0	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0	0
Total	18	15	51	252	2	338

Result Summary

Vulnerability Type	Occurrences	Severity
SQL Injection	14	Critical
Stored XSS	2	Critical
Command Injection	1	Critical
Second Order SQL Injection	1	Critical
Reflected XSS	15	High
Unchecked Input for Loop Condition	20	Medium

Open Redirect	17	Medium
Stored Command Argument Injection	6	Medium
Parameter Tampering	3	Medium
OS Access Violation	2	Medium
Insufficiently Protected Credentials	1	Medium
Missing HSTS Header	1	Medium
Missing HSTS Header	1	Medium
Trust Boundary Violation in Session Variables	218	Low
Information Exposure Through an Error Message	15	Low
Client Dangerous File Inclusion	10	Low
Log Forging	6	Low
Missing Content Security Policy	1	Low
Permissive Content Security Policy	1	Low
Potential Clickjacking on Legacy Browsers	1	Low
Client Potential XSS	2	Information

10 Most Vulnerable Files

Critical High and Medium Vulnerabilities

File Name	Issues Found
GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	17
GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	12
GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html	12
GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py	10
GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py	7
GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py	5
GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py	4
GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	4
GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py	3
GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py	3

Scan Results Details

SQL Injection

Query Path:
Python\Cx\Python Critical Risk\SQL Injection Version:3

Categories

OWASP Top 10 2013: A1-Injection
FISMA 2014: System And Information Integrity
NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection
CWE top 25: CWE top 25
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
OWASP ASVS: V05 Validation, Sanitization and Encoding
OWASP Top 10 2021: A3-Injection
PCI DSS v3.2.1: PCI DSS (3.2.1) - 6.5.1 - Injection flaws - particularly SQL injection
SANS top 25: SANS top 25
ASA Premium: ASA Premium
Top Tier: Top Tier
Base Preset: Base Preset
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
ASD STIG 6.1: APSC-DV-002540 - CAT I The application must not be vulnerable to SQL Injection.

Description

SQL Injection\Path 1:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=5
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 248 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 227 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	227	248
Object	form	execute

Code Snippet	
File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
227.   donor.phone_no = (request.form.get('phone_no') or '').strip()
....
248.   countries = db.session.execute(
```

SQL Injection\Path 2:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=6
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 227 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	227	262
Object	form	execute

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
227.   donor.phone_no = (request.form.get('phone_no') or '').strip()
....
262.   countries = db.session.execute(
```

SQL Injection\Path 3:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=7
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 248 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create

method at line 228 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	228	248
Object	form	execute

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
228.   donor.email_text = request.form.get('email_text',
'').strip().lower() or None
....
248.   countries = db.session.execute(
```

SQL Injection\Path 4:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=8
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.

An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 228 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	228	262
Object	form	execute

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
228.   donor.email_text = request.form.get('email_text',
'').strip().lower() or None
....
262.   countries = db.session.execute(
```

SQL Injection\Path 5:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=9
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 248 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 225 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	225	248
Object	form	execute

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
225.   donor.address2_text = request.form.get('address2_text',
'').strip() or None
....
248.   countries = db.session.execute(
```

SQL Injection\Path 6:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=10
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.

An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 225 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	225	262
Object	form	execute

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
225.  donor.address2_text = request.form.get('address2_text',
'').strip() or None
....
262.  countries = db.session.execute(
```

SQL Injection\Path 7:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=11
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 248 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 224 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	224	248
Object	form	execute

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
224.  donor.address1_text = (request.form.get('address1_text') or
'').strip()
....
248.  countries = db.session.execute(
```

SQL Injection\Path 8:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=12
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 224 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization.
This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	224	262
Object	form	execute

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
224.  donor.address1_text = (request.form.get('address1_text') or
'').strip()
....
262.  countries = db.session.execute(
```

SQL Injection\Path 9:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=13
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 248 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.

An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 223 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	223	248
Object	form	execute

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
223.  donor.org_type_desc = request.form.get('org_type_desc',
'').strip() or None
....
248.  countries = db.session.execute(
```

SQL Injection\Path 10:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=14
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 223 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	223	262
Object	form	execute

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
223.  donor.org_type_desc = request.form.get('org_type_desc',
'').strip() or None
....
262.  countries = db.session.execute(
```

SQL Injection\Path 11:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=15
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 248 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 222 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	222	248
Object	form	execute

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
222.  donor.donor_name = (request.form.get('donor_name') or
'').strip().upper()
....
248.  countries = db.session.execute(
```

SQL Injection\Path 12:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=16
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.

An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 222 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	222	262
Object	form	execute

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
222.  donor.donor_name = (request.form.get('donor_name') or
'').strip().upper()
....
262.  countries = db.session.execute(
```

SQL Injection\Path 13:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=17
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 248 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 221 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization. This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	221	248
Object	form	execute

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
221.   donor.donor_code = (request.form.get('donor_code') or
'').strip().upper()
....
248.   countries = db.session.execute(
```

SQL Injection\Path 14:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=18
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's create method executes an SQL query with execute, at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly.
An attacker would be able to inject arbitrary syntax and data into the SQL query, by crafting a malicious payload and providing it via the input form; this input is then read by the create method at line 221 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code, into a query and to the database server - without sanitization.
This may enable an SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	221	262
Object	form	execute

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
221.   donor.donor_code = (request.form.get('donor_code') or
'').strip().upper()
....
262.   countries = db.session.execute(
```

Stored XSS

Query Path:
Python\Cx\Python Critical Risk\Stored XSS Version:4

Categories

OWASP Top 10 2013: A3-Cross-Site Scripting (XSS)
FISMA 2014: System And Information Integrity
NIST SP 800-53: SI-15 Information Output Filtering (P0)
OWASP Top 10 2017: A7-Cross-Site Scripting (XSS)
CWE top 25: CWE top 25
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
OWASP ASVS: V05 Validation, Sanitization and Encoding

OWASP Top 10 2021: A3-Injection
PCI DSS v3.2.1: PCI DSS (3.2.1) - 6.5.7 - Cross-site scripting (XSS)
SANS top 25: SANS top 25
ASA Premium: ASA Premium
Top Tier: Top Tier
Base Preset: Base Preset
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
ASD STIG 6.1: APSC-DV-002490 - CAT I The application must protect from Cross-Site Scripting (XSS) vulnerabilities.

Description

Stored XSS\Path 1:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=2
Status	New
Detection Date	12/1/2025 9:38:28 AM

The method %} embeds untrusted data in generated output with url_for, at line 219 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by saving malicious data in a data-store ahead of time. The attacker's modified data is then read from the database by the funds_donations method with filter, at line 141 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This untrusted data then flows through the code straight to the output web page, without sanitization.
This can enable a Stored Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	141	219
Object	filter	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method def funds_donations():

```
....
141.  ).filter(
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method {% block content %}

```
....
219.  <a class="page-link" href="{{ url_for('reports.funds_donations', page=pagination.prev_num, country_id=filters.country_id, date_from=filters.date_from, date_to=filters.date_to, currency_code=filters.currency_code) }}">
```

Stored XSS\Path 2:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=3
Status	New
Detection Date	12/1/2025 9:38:28 AM

The method %} embeds untrusted data in generated output with url_for, at line 251 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by saving malicious data in a data-store ahead of time. The attacker's modified data is then read from the database by the funds_donations method with filter, at line 141 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This untrusted data then flows through the code straight to the output web page, without sanitization.
This can enable a Stored Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	141	251
Object	filter	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method def funds_donations():

```
....
141.   ).filter(
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method {% block content %}

```
....
251.  <a class="page-link" href="{{ url_for('reports.funds_donations', page=pagination.next_num, country_id=filters.country_id, date_from=filters.date_from, date_to=filters.date_to, currency_code=filters.currency_code) }}">
```

Command Injection

Query Path:
Python\Cx\Python Critical Risk\Command Injection Version:4

Categories

OWASP Top 10 2013: A1-Injection
FISMA 2014: System And Information Integrity
NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection

CWE top 25: CWE top 25
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
OWASP Top 10 2021: A3-Injection
PCI DSS v3.2.1: PCI DSS (3.2.1) - 6.5.1 - Injection flaws - particularly SQL injection
ASA Premium: ASA Premium
Top Tier: Top Tier
Base Preset: Base Preset
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
ASD STIG 6.1: APSC-DV-002510 - CAT I The application must protect from command injection.

Description

Command Injection\Path 1:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=1
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's main method calls an OS (shell) command with input, at line 105 of GOJ_DMIS-feature-hadr-aid-tracking/scripts/migrate_phone_numbers.py, using an untrusted string with the command to execute.
This could allow an attacker to inject an arbitrary command, and enable a Command Injection attack.
The attacker may be able to inject the executed command via user input, input, which is retrieved by the application in the main method, at line 105 of GOJ_DMIS-feature-hadr-aid-tracking/scripts/migrate_phone_numbers.py.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/scripts/migrate_phone_numbers.py	GOJ_DMIS-feature-hadr-aid-tracking/scripts/migrate_phone_numbers.py
Line	105	105
Object	input	input

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/scripts/migrate_phone_numbers.py
Method def main():

```
....
105.   response = input("Are you sure you want to proceed? (yes/no): ")
```

Second Order SQL Injection

Query Path:
Python\Cx\Python Critical Risk\Second Order SQL Injection Version:5

Categories

OWASP Top 10 2013: A1-Injection
FISMA 2014: System And Information Integrity
NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection
CWE top 25: CWE top 25
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
OWASP ASVS: V05 Validation, Sanitization and Encoding

OWASP Top 10 2021: A3-Injection
PCI DSS v3.2.1: PCI DSS (3.2.1) - 6.5.1 - Injection flaws - particularly SQL injection
SANS top 25: SANS top 25
ASA Premium: ASA Premium
Top Tier: Top Tier
Base Preset: Base Preset
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
ASD STIG 6.1: APSC-DV-002540 - CAT I The application must not be vulnerable to SQL Injection.

Description

Second Order SQL Injection\Path 1:

Severity	Critical
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=4
Status	New
Detection Date	12/1/2025 9:38:28 AM

The application's init_database method executes an SQL query with execute, at line 42 of GOJ_DMIS-feature-hadr-aid-tracking/scripts/init_db.py. The application constructs this SQL query by embedding an untrusted string into the query without proper sanitization. The concatenated string is submitted to the database, where it is parsed and executed accordingly. The attacker may be able to write arbitrary data to the database, which is then retrieved by the application with read in init_database method at line 39 of GOJ_DMIS-feature-hadr-aid-tracking/scripts/init_db.py. This data then flows through the code, until it is used directly in the SQL query without sanitization, and then submitted to the database server for execution.
This may enable a Second-Order SQL Injection attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/scripts/init_db.py	GOJ_DMIS-feature-hadr-aid-tracking/scripts/init_db.py
Line	39	42
Object	read	execute

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/scripts/init_db.py
Method def init_database():

```
....
39.   sql_content = f.read()
....
42.   cursor.execute(sql_content)
```

Reflected XSS

Query Path:
Python\Cx\Python High Risk\Reflected XSS Version:5

Categories

OWASP Top 10 2013: A3-Cross-Site Scripting (XSS)
FISMA 2014: System And Information Integrity
NIST SP 800-53: SI-15 Information Output Filtering (P0)
OWASP Top 10 2017: A7-Cross-Site Scripting (XSS)
CWE top 25: CWE top 25

MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
OWASP ASVS: V05 Validation, Sanitization and Encoding
OWASP Top 10 2021: A3-Injection
PCI DSS v3.2.1: PCI DSS (3.2.1) - 6.5.7 - Cross-site scripting (XSS)
SANS top 25: SANS top 25
ASA Premium: ASA Premium
Top Tier: Top Tier
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
ASD STIG 6.1: APSC-DV-002490 - CAT I The application must protect from Cross-Site Scripting (XSS) vulnerabilities.

[Description](#)

Reflected XSS\Path 1:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=19
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method login embeds untrusted data in generated output with ReturnStmt, at line 238 of GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the login method at line 236 of GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py
Line	236	238
Object	args	ReturnStmt

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py
Method	def login():

```
....
236.   next_page = request.args.get('next')
....
238.   return redirect(next_page)
```

Reflected XSS\Path 2:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=20
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 97 of GOJ_DMIS-feature-hadr-aid-tracking/templates/uom/list.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the list_uom method at line 101 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/uom/list.html
Line	101	97
Object	args	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py
Method def list_uom():

```
....
101.   filter_type = request.args.get('filter', 'all')
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/uom/list.html
Method {% block content %}

```
....
97.   <a href="{{ url_for('uom.list_uom', filter=filter_type) }}"
class="btn-relief-secondary">
```

Reflected XSS\Path 3:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=21
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 97 of GOJ_DMIS-feature-hadr-aid-tracking/templates/item_categories/list.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the list_categories method at line 108 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/item_categories.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/item_categories.p	GOJ_DMIS-feature-hadr-aid-tracking/templates/item_categories/list.h

	y	tml
Line	108	97
Object	args	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/item_categories.py
Method def list_categories():

```
....
108.   filter_type = request.args.get('filter', 'all')
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/item_categories/list.html

Method {% block content %}

```
....
97.  <a href="{{ url_for('item_categories.list_categories',
filter=filter_type) }}" class="btn-relief-secondary">
```

Reflected XSS\Path 4:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=22
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 219 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 121 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	121	219
Object	args	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method def funds_donations():

```
....
121.   country_filter = request.args.get('country_id', '', type=str)
```

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method	{% block content %}

```
....
219.   <a class="page-link" href="{{ url_for('reports.funds_donations',
page=pagination.prev_num, country_id=filters.country_id,
date_from=filters.date_from, date_to=filters.date_to,
currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 5:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=23
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 251 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 121 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	121	251
Object	args	url_for

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method	def funds_donations():

```
....
121.   country_filter = request.args.get('country_id', '', type=str)
```

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method	{% block content %}

```
....
251.  <a class="page-link" href="{{ url_for('reports.funds_donations',
page=pagination.next_num, country_id=filters.country_id,
date_from=filters.date_from, date_to=filters.date_to,
currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 6:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=24
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 219 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 122 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	122	219
Object	args	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method def funds_donations():

```
....
122.  date_from = request.args.get('date_from', '', type=str)
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method {% block content %}

```
....
219.  <a class="page-link" href="{{ url_for('reports.funds_donations',
page=pagination.prev_num, country_id=filters.country_id,
date_from=filters.date_from, date_to=filters.date_to,
currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 7:

Severity	High
Result State	To Verify

Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=25
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 251 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 122 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	122	251
Object	args	url_for

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method	def funds_donations():

```
....
122.   date_from = request.args.get('date_from', '', type=str)
```

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method	{% block content %}

```
....
251.   <a class="page-link" href="{{ url_for('reports.funds_donations', 
page=pagination.next_num, country_id=filters.country_id, 
date_from=filters.date_from, date_to=filters.date_to, 
currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 8:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=26
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 219 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 123 of GOJ_DMIS-

feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	123	219
Object	args	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method def funds_donations():

```
....
123.   date_to = request.args.get('date_to', '', type=str)
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method {% block content %}

```
....
219.   <a class="page-link" href="{{ url_for('reports.funds_donations', 
page=pagination.prev_num, country_id=filters.country_id, 
date_from=filters.date_from, date_to=filters.date_to, 
currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 9:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=27
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 251 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 123 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	123	251

Object	args	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method def funds_donations():

```
....
123.  date_to = request.args.get('date_to', '', type=str)
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method {% block content %}

```
....
251.  <a class="page-link" href="{{ url_for('reports.funds_donations',
page=pagination.next_num, country_id=filters.country_id,
date_from=filters.date_from, date_to=filters.date_to,
currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 10:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=28
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 219 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 124 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	124	219
Object	args	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method def funds_donations():

```
....
124.  currency_filter = request.args.get('currency_code', '', type=str)
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html

Method {% block content %}

```
....
219.  <a class="page-link" href="{{ url_for('reports.funds_donations', page=pagination.prev_num, country_id=filters.country_id, date_from=filters.date_from, date_to=filters.date_to, currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 11:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=29
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 251 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 124 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	124	251
Object	args	url_for

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py

Method def funds_donations():

```
....
124.  currency_filter = request.args.get('currency_code', '', type=str)
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html

Method {% block content %}

```
....
251.  <a class="page-link" href="{{ url_for('reports.funds_donations', page=pagination.next_num, country_id=filters.country_id, date_from=filters.date_from, date_to=filters.date_to, currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 12:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=30
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 132 of GOJ_DMIS-feature-hadr-aid-tracking/templates/agencies/list.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the list_agencies method at line 150 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/agencies.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/agencies.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/agencies/list.html
Line	150	132
Object	args	url_for

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/agencies.py
Method	def list_agencies():

```
....
150.   filter_type = request.args.get('filter', 'all')
```

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/agencies/list.html
Method	{% block content %}

```
....
132.   <a href="{{ url_for('agencies.list_agencies', filter=filter_type)
}}" class="btn-relief-secondary">
```

Reflected XSS\Path 13:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=31
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 219 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.

The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 118 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	118	219
Object	args	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method def funds_donations():

```
....
118.   page = request.args.get('page', 1, type=int)
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method {% block content %}

```
....
219.   <a class="page-link" href="{{ url_for('reports.funds_donations',
page=pagination.prev_num, country_id=filters.country_id,
date_from=filters.date_from, date_to=filters.date_to,
currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 14:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=32
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with url_for, at line 251 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the funds_donations method at line 118 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html

Line	118	251
Object	args	url_for

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/reports.py
Method def funds_donations():

```
....
118.   page = request.args.get('page', 1, type=int)
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method {% block content %}

```
....
251.   <a class="page-link" href="{{ url_for('reports.funds_donations',
page=pagination.next_num, country_id=filters.country_id,
date_from=filters.date_from, date_to=filters.date_to,
currency_code=filters.currency_code) }}">
```

Reflected XSS\Path 15:

Severity	High
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=33
Status	New
Detection Date	12/1/2025 9:38:29 AM

The method %} embeds untrusted data in generated output with email_text, at line 116 of GOJ_DMIS-feature-hadr-aid-tracking/templates/donors/list.html. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.
The attacker would be able to alter the returned web page by simply providing modified data in the user input args, which is read by the list_donors method at line 150 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This input then flows through the code straight to the output web page, without sanitization.
This can enable a Reflected Cross-Site Scripting (XSS) attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/templates/donors/list.html
Line	150	116
Object	args	email_text

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def list_donors():

```
....
150.   search_query = request.args.get('search', '').strip()
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/donors/list.html

Method {% block content %}

```
....
116.  <a href="mailto:{{ donor.email_text }}" class="text-decoration-
none">
```

Unchecked Input for Loop Condition

Query Path:
Python\Cx\Python Medium Threat\Unchecked Input for Loop Condition Version:2

Categories

OWASP Top 10 2021: A4-Insecure Design
ASA Premium: ASA Premium
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
OWASP Top 10 API 2023: API4-Unrestricted Resource Consumption
ASD STIG 6.1: APSC-DV-002560 - CAT I The application must not be subject to input handling vulnerabilities.

Description

Unchecked Input for Loop Condition\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=90
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method strip_sensitive_query_params at line 198 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py obtains user input from args - the range of this value is not validated, and is eventually used in a loop condition in strip_sensitive_query_params at line 198 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Line	198	198
Object	args	keys

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py

Method def strip_sensitive_query_params():

```
....
198.  for param_name in request.args.keys():
```

Unchecked Input for Loop Condition\Path 2:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=91
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create_donation at line 227 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in create_donation at line 227 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Line	227	227
Object	form	keys

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Method	def create_donation():

```
....
227.   for key in request.form.keys():
```

Unchecked Input for Loop Condition\Path 3:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=92
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method decorated_function at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py obtains user input from args - the range of this value is not validated, and is eventually used in a loop condition in decorated_function at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Line	262	262
Object	args	keys

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method	def decorated_function(*args, **kwargs):

```
....
262.   for param_name in request.args.keys():
```

Unchecked Input for Loop Condition\Path 4:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=93
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method edit_donation at line 700 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in edit_donation at line 700 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Line	700	700
Object	form	keys

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Method	def edit_donation(donation_id):

```
....
700.   for key in request.form.keys():
```

Unchecked Input for Loop Condition\Path 5:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=95
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method verify_donation_detail at line 1364 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in verify_donation_detail at line 1364 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Line	1364	1364
Object	form	keys

Code Snippet	
File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Method	def verify_donation_detail(donation_id):

```
....
1364.   for key in request.form.keys():
```

Unchecked Input for Loop Condition\Path 6:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=97
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method _process_allocations at line 1795 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in _process_allocations at line 1795 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Line	1795	1795
Object	form	keys

Code Snippet	
File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Method	def _process_allocations(relief_request, validate_complete=False):

```
....
1795.   allocation_keys = [k for k in request.form.keys() if
k.startswith(f'batch_allocation_{item_id}_')]
```

Unchecked Input for Loop Condition\Path 7:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=99
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create at line 232 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in create at line 233 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py

Line	232	233
Object	form	warehouse_ids

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Method def create():

```
....
232.   warehouse_ids = request.form.getlist('warehouses')
233.   for warehouse_id in warehouse_ids:
```

Unchecked Input for Loop Condition\Path 8:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=101
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method get_safe_query_params at line 346 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py obtains user input from args - the range of this value is not validated, and is eventually used in a loop condition in sanitize_query_string at line 160 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Line	346	160
Object	args	items

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method def get_safe_query_params():

```
....
346.   sanitized, _ = sanitize_query_string(request.args)
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method def sanitize_query_string(query_args):

```
....
160.   for key, value in query_args.items():
```

Unchecked Input for Loop Condition\Path 9:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018

	&projectid=7&pathid=103
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method edit at line 531 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in edit at line 532 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Line	531	532
Object	form	warehouse_ids

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Method def edit(user_id):

```
....
531.   warehouse_ids = request.form.getlist('warehouses')
532.   for warehouse_id in warehouse_ids:
```

Unchecked Input for Loop Condition\Path 10:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=105
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create_intake at line 44 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in create_intake at line 49 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Line	44	49
Object	form	zip

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method def create_intake():

```
....
44.  usable_qtys = request.form.getlist('usable_qty[]')
....
49.  for u, d, e in zip(usable_qtys, defective_qtys, expired_qtys):
```

Unchecked Input for Loop Condition\Path 11:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=106
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create_intake at line 45 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in create_intake at line 49 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Line	45	49
Object	form	zip

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method def create_intake():

```
....
45.  defective_qtys = request.form.getlist('defective_qty[]')
....
49.  for u, d, e in zip(usable_qtys, defective_qtys, expired_qtys):
```

Unchecked Input for Loop Condition\Path 12:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=107
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create_intake at line 46 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in create_intake at line 49 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.p	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.p

	y	y
Line	46	49
Object	form	zip

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method def create_intake():

```
....
46.   expired_qtys = request.form.getlist('expired_qty[]')
....
49.   for u, d, e in zip(usable_qtys, defective_qtys, expired_qtys):
```

Unchecked Input for Loop Condition\Path 13:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=108
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create_intake at line 44 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in create_intake at line 67 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Line	44	67
Object	form	zip

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method def create_intake():

```
....
44.   usable_qtys = request.form.getlist('usable_qty[]')
....
67.   for item_id, usable_qty, defective_qty, expired_qty in
zip(item_ids, usable_qtys, defective_qtys, expired_qtys):
```

Unchecked Input for Loop Condition\Path 14:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=109
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create_intake at line 45 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in create_intake at line 67 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Line	45	67
Object	form	zip

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method	def create_intake():

```
....
45.  defective_qtys = request.form.getlist('defective_qty[]')
....
67.  for item_id, usable_qty, defective_qty, expired_qty in
zip(item_ids, usable_qtys, defective_qtys, expired_qtys):
```

Unchecked Input for Loop Condition\Path 15:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=110
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create_intake at line 46 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in create_intake at line 67 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Line	46	67
Object	form	zip

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method	def create_intake():

```
....
46.  expired_qtys = request.form.getlist('expired_qty[]')
....
67.  for item_id, usable_qty, defective_qty, expired_qty in
zip(item_ids, usable_qtys, defective_qtys, expired_qtys):
```

Unchecked Input for Loop Condition\Path 16:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=111
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create_donation at line 413 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py obtains user input from files - the range of this value is not validated, and is eventually used in a loop condition in create_donation at line 419 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Line	413	419
Object	files	enumerate

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Method	def create_donation():

```
....
413.   uploaded_files = request.files.getlist('document_files')
....
419.   for idx, uploaded_file in enumerate(uploaded_files):
```

Unchecked Input for Loop Condition\Path 17:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=112
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method verify_donation_detail at line 1533 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py obtains user input from files - the range of this value is not validated, and is eventually used in a loop condition in verify_donation_detail at line 1539 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py

Line	1533	1539
Object	files	enumerate

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Method def verify_donation_detail(donation_id):

```
....
1533.   uploaded_files = request.files.getlist('document_files')
....
1539.   for idx, uploaded_file in enumerate(uploaded_files):
```

Unchecked Input for Loop Condition\Path 18:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=113
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method _process_entry_submission at line 332 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in _process_entry_submission at line 452 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py
Line	332	452
Object	form	intake_items_data

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py
Method def _process_entry_submission(donation, warehouse, existing_intake, action):

```
....
332.   item_comments = request.form.get(f'item_comments_{item_id}',
'').strip()
....
452.   for item_data in intake_items_data:
```

Unchecked Input for Loop Condition\Path 19:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=114
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method _process_verification_submission at line 637 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in _process_verification_submission at line 749 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py
Line	637	749
Object	form	verified_items_data

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py
Method	def _process_verification_submission(intake, donation, warehouse):

```
....
637.   item_comments = request.form.get(f'item_comments_{item_id}',
intake_item.comments_text or '').strip()
....
749.   for item_data in verified_items_data:
```

Unchecked Input for Loop Condition\Path 20:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=115
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method _process_verification_submission at line 634 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py obtains user input from form - the range of this value is not validated, and is eventually used in a loop condition in _process_verification_submission at line 749 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py, allowing attackers to provide a very high number of iterations.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py
Line	634	749
Object	form	verified_items_data

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py
Method	def _process_verification_submission(intake, donation, warehouse):

```
....
634.  batch_no_raw = request.form.get(f'batch_no_{item_id}',
batch_no_default).strip().upper()
....
749.  for item_data in verified_items_data:
```

Open Redirect

Query Path:
Python\Cx\Python Medium Threat\Open Redirect Version:3

Categories

OWASP Top 10 2013: A10-Unvalidated Redirects and Forwards
FISMA 2014: System And Information Integrity
NIST SP 800-53: SI-10 Information Input Validation (P1)
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
OWASP ASVS: V05 Validation, Sanitization and Encoding
OWASP Top 10 2021: A1-Broken Access Control
PCI DSS v3.2.1: PCI DSS (3.2.1) - 6.5.8 - Improper access control
ASA Premium: ASA Premium
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
OWASP Top 10 API 2023: API10-Unsafe Consumption of APIs
ASD STIG 6.1: APSC-DV-002560 - CAT I The application must not be subject to input handling vulnerabilities.

Description

Open Redirect\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=39
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by url in GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py at line 36 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py at line 36, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py	GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py
Line	36	36
Object	url	redirect

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py
Method def decorated_function(*args, **kwargs):

```
....
36.  return redirect(url_for('login', next=request.url))
```

Open Redirect\Path 2:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=40
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by url in GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py at line 72 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py at line 72, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py	GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py
Line	72	72
Object	url	redirect

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py
Method	def decorated_function(*args, **kwargs):

```
....
72.    return redirect(url_for('login', next=request.url))
```

Open Redirect\Path 3:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=41
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by url in GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py at line 111 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py at line 111, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py	GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py
Line	111	111
Object	url	redirect

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/core/decorators.py
Method	def decorated_function(*args, **kwargs):

```
....
111.   return redirect(url_for('login', next=request.url))
```

Open Redirect\Path 4:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=42
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by get in GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py at line 236 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py at line 238, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py
Line	236	238
Object	get	redirect

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py

Method def login():

```
....
236.   next_page = request.args.get('next')
....
238.   return redirect(next_page)
```

Open Redirect\Path 5:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=43
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py at line 155 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py at line 162, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py
Line	155	162
Object	form	redirect

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py

Method def create_intake():

```
....
155.  donation_id = request.form.get('donation_id')
....
162.  return redirect(url_for('donation_intake.intake_form',
```

Open Redirect\Path 6:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=44
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py at line 156 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py at line 162, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py
Line	156	162
Object	form	redirect

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donation_intake.py
Method	def create_intake():

```
....
156.  inventory_id = request.form.get('inventory_id')
....
162.  return redirect(url_for('donation_intake.intake_form',
```

Open Redirect\Path 7:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=45
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py at line 25 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py at line 93, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	25	93

Object	form	redirect

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py

Method def create():

```
....
25.   transport_mode = request.form.get('transport_mode', '').strip()
....
93.   return redirect(url_for('transfers.view',
transfer_id=new_transfer.transfer_id))
```

Open Redirect\Path 8:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=46
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py at line 26 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py at line 93, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	26	93
Object	form	redirect

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py

Method def create():

```
....
26.   comments_text = request.form.get('comments_text', '').strip()
....
93.   return redirect(url_for('transfers.view',
transfer_id=new_transfer.transfer_id))
```

Open Redirect\Path 9:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=47
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py at line 21 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py at line 93, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	21	93
Object	form	redirect

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Method	def create():

```
....
21.   to_warehouse_id = request.form.get('to_warehouse_id', type=int)
....
93.   return redirect(url_for('transfers.view',
transfer_id=new_transfer.transfer_id))
```

Open Redirect\Path 10:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=48
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py at line 22 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py at line 93, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	22	93
Object	form	redirect

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Method	def create():

```
....
22.   item_id = request.form.get('item_id', type=int)
....
93.   return redirect(url_for('transfers.view',
transfer_id=new_transfer.transfer_id))
```

Open Redirect\Path 11:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=49
Status	New

Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py at line 44 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py at line 186, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Line	44	186
Object	form	redirect

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method	def create_intake():

```
....
44.   usable_qtys = request.form.getlist('usable_qty[]')
....
186.   return redirect(url_for('intake.view_intake', 
reliefpkg_id=reliefpkg_id, inventory_id=first_inventory_id))
```

Open Redirect\Path 12:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=50
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py at line 45 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py at line 186, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Line	45	186
Object	form	redirect

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method	def create_intake():

```
....
45.  defective_qtys = request.form.getlist('defective_qty[]')
....
186.  return redirect(url_for('intake.view_intake', 
reliefpkg_id=reliefpkg_id, inventory_id=first_inventory_id))
```

Open Redirect\Path 13:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=51
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py at line 46 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py at line 186, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Line	46	186
Object	form	redirect

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method	def create_intake():

```
....
46.  expired_qtys = request.form.getlist('expired_qty[]')
....
186.  return redirect(url_for('intake.view_intake', 
reliefpkg_id=reliefpkg_id, inventory_id=first_inventory_id))
```

Open Redirect\Path 14:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=52
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py at line 43 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py at line 186, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.p	GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.p

	y	y
Line	43	186
Object	form	redirect

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/intake_aidmgmt.py
Method def create_intake():

```
....
43.  item_ids = request.form.getlist('item_id[]')
....
186.  return redirect(url_for('intake.view_intake',
reliefpkg_id=reliefpkg_id, inventory_id=first_inventory_id))
```

Open Redirect\Path 15:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=53
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py at line 152 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py at line 184, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py
Line	152	184
Object	form	redirect

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py
Method def create_uom():

```
....
152.  is_valid, errors, normalized_data =
validate_uom_data(request.form)
....
184.  return redirect(url_for('uom.view_uom', uom_code=uom.uom_code))
```

Open Redirect\Path 16:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=54
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py at line 240 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py at line 269, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py
Line	240	269
Object	form	redirect

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/uom.py
Method def edit_uom(uom_code):

```
....
240.  is_valid, errors, normalized_data =
validate_uom_data(request.form, is_update=True, uom_code=uom_code)
....
269.  return redirect(url_for('uom.view_uom', uom_code=uom.uom_code))
```

Open Redirect\Path 17:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=55
Status	New
Detection Date	12/1/2025 9:38:32 AM

The potentially tainted value provided by form in GOJ_DMIS-feature-hadr-aid-tracking/app/features/agencies.py at line 224 is used as a destination URL by redirect in GOJ_DMIS-feature-hadr-aid-tracking/app/features/agencies.py at line 269, potentially allowing attackers to perform an open redirection.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/agencies.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/agencies.py
Line	224	269
Object	form	redirect

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/agencies.py
Method def create_agency():

```
....
224.  is_valid, errors, normalized_data =
validate_agency_data(request.form)
....
269.  return redirect(url_for('agencies.view_agency',
agency_id=new_agency.agency_id))
```

Stored Command Argument Injection

Query Path:
Python\Cx\Python Medium Threat\Stored Command Argument Injection Version:2

Categories

CWE top 25: CWE top 25
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
OWASP ASVS: V05 Validation, Sanitization and Encoding
OWASP Top 10 2021: A3-Injection
SANS top 25: SANS top 25

Description

Stored Command Argument Injection\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=56
Status	New
Detection Date	12/1/2025 9:38:33 AM

An argument is passed to an external OS command by execute_values at GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py in line 217. This could allow an attacker to attack the external program by injecting malicious arguments into its execution.
The attacker may be able to inject a command or argument to execute into storage, which is then retrieved by the application with connect, at line 184 of GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py, in the insert_records method.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py	GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py
Line	184	217
Object	connect	execute_values

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py
Method def insert_records(records: list[dict], db_url: str, create_by_id: str = 'IMPORT'):

```
....
184.   conn = psycopg2.connect(db_url)
....
217.   execute_values(cur, insert_sql, values, page_size=1000)
```

Stored Command Argument Injection\Path 2:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=57
Status	New
Detection Date	12/1/2025 9:38:33 AM

An argument is passed to an external OS command by execute_values at GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py in line 217. This could allow an attacker to attack the external program by injecting malicious arguments into its execution.

The attacker may be able to inject a command or argument to execute into storage, which is then retrieved by the application with read_excel, at line 97 of GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py, in the parse_excel_data method.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py	GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py
Line	97	217
Object	read_excel	execute_values

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py

Method def parse_excel_data(excel_path: str) -> list[dict]:

```
....
97.   df = pd.read_excel(xlsx, sheet_name=sheet_name, header=None)
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/scripts/import_hadr_aid_staging.py

Method def insert_records(records: list[dict], db_url: str, create_by_id: str = 'IMPORT'):

```
....
217.   execute_values(cur, insert_sql, values, page_size=1000)
```

Stored Command Argument Injection\Path 3:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=58
Status	New
Detection Date	12/1/2025 9:38:33 AM

An argument is passed to an external OS command by execute_values at GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py in line 238. This could allow an attacker to attack the external program by injecting malicious arguments into its execution.

The attacker may be able to inject a command or argument to execute into storage, which is then retrieved by the application with connect, at line 205 of GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py, in the insert_records method.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py
Line	205	238
Object	connect	execute_values

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py
Method	def insert_records(records: list[dict], db_url: str, create_by_id: str = 'IMPORT'):

```
....
205.   conn = psycopg2.connect(db_url)
....
238.   execute_values(cur, insert_sql, values, page_size=1000)
```

Stored Command Argument Injection\Path 4:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=59
Status	New
Detection Date	12/1/2025 9:38:33 AM

An argument is passed to an external OS command by execute_values at GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py in line 238. This could allow an attacker to attack the external program by injecting malicious arguments into its execution.
The attacker may be able to inject a command or argument to execute into storage, which is then retrieved by the application with read_excel, at line 111 of GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py, in the parse_excel_data method.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py
Line	111	238
Object	read_excel	execute_values

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py
Method	def parse_excel_data(excel_path: str) -> list[dict]:

```
....
111.   df = pd.read_excel(xlsx, sheet_name=sheet_name, header=None)
```

File Name	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_hadr_aid_staging_1764348663793.py
Method	def insert_records(records: list[dict], db_url: str, create_by_id: str = 'IMPORT'):

```
....
238.   execute_values(cur, insert_sql, values, page_size=1000)
```

Stored Command Argument Injection\Path 5:

Severity	Medium

Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=60
Status	New
Detection Date	12/1/2025 9:38:33 AM

An argument is passed to an external OS command by execute_values at GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_mlss_warehouse_staging (1)_1764433314146.py in line 741. This could allow an attacker to attack the external program by injecting malicious arguments into its execution.
The attacker may be able to inject a command or argument to execute into storage, which is then retrieved by the application with connect, at line 708 of GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_mlss_warehouse_staging (1)_1764433314146.py, in the insert_records method.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_mlss_warehouse_staging (1)_1764433314146.py	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_mlss_warehouse_staging (1)_1764433314146.py
Line	708	741
Object	connect	execute_values

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/attached_assets/import_mlss_warehouse_staging (1)_1764433314146.py
Method	def insert_records(records: list[dict], db_url: str, create_by_id: str = 'MLSS_IMPORT'):

```
....
708.   conn = psycopg2.connect(db_url)
....
741.   execute_values(cur, insert_sql, values, page_size=1000)
```

Stored Command Argument Injection\Path 6:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=61
Status	New
Detection Date	12/1/2025 9:38:33 AM

An argument is passed to an external OS command by execute_values at GOJ_DMIS-feature-hadr-aid-tracking/import_mlss_warehouse_staging.py in line 753. This could allow an attacker to attack the external program by injecting malicious arguments into its execution.
The attacker may be able to inject a command or argument to execute into storage, which is then retrieved by the application with connect, at line 720 of GOJ_DMIS-feature-hadr-aid-tracking/import_mlss_warehouse_staging.py, in the insert_records method.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/import_mlss_warehouse_staging.py	GOJ_DMIS-feature-hadr-aid-tracking/import_mlss_warehouse_staging.py

Line	720	753
Object	connect	execute_values

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/import_mlss_warehouse_staging.py
Method def insert_records(records: list[dict], db_url: str, create_by_id: str = 'MLSS_IMPORT'):

```
....
720.   conn = psycopg2.connect(db_url)
....
753.   execute_values(cur, insert_sql, values, page_size=1000)
```

Parameter Tampering

Query Path:
Python\Cx\Python Medium Threat\Parameter Tampering Version:3

Categories

OWASP Top 10 2013: A4-Insecure Direct Object References
OWASP Top 10 2017: A5-Broken Access Control
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Security Functions
OWASP ASVS: V01 Architecture, Design and Threat Modeling
OWASP Top 10 2021: A4-Insecure Design
PCI DSS v3.2.1: PCI DSS (3.2.1) - 6.5.1 - Injection flaws - particularly SQL injection
ASA Premium: ASA Premium
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
OWASP Top 10 API 2023: API1-Broken Object Level Authorization
ASD STIG 6.1: APSC-DV-002560 - CAT I The application must not be subject to input handling vulnerabilities.

Description

Parameter Tampering\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=62
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create at line 199 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This input is later concatenated by the application directly into a string variable containing SQL commands, without being validated. This string is then used in method validate_donor_data to query the database execute, at line 116 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py, without any additional filtering by the database. This could allow the user to tamper with the filter parameter.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	199	116
Object	form	execute

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
199.   is_valid, errors = validate_donor_data(request.form)
```

▼

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def validate_donor_data(form_data, is_update=False, donor_id=None):

```
....
116.   country_exists = db.session.execute(
```

Parameter Tampering\Path 2:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=63
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method edit at line 321 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This input is later concatenated by the application directly into a string variable containing SQL commands, without being validated. This string is then used in method validate_donor_data to query the database execute, at line 116 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py, without any additional filtering by the database. This could allow the user to tamper with the filter parameter.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	321	116
Object	form	execute

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def edit(donor_id):

```
....
321.   is_valid, errors = validate_donor_data(request.form,
is_update=True, donor_id=donor_id)
```

▼

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def validate_donor_data(form_data, is_update=False, donor_id=None):

```
....
116.   country_exists = db.session.execute(
```

Parameter Tampering\Path 3:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=64
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method create_request at line 21 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/account_requests.py gets user input from element form. This input is later concatenated by the application directly into a string variable containing SQL commands, without being validated. This string is then used in method create_request to query the database filter, at line 28 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/account_requests.py, without any additional filtering by the database. This could allow the user to tamper with the filter parameter.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/account_requests.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/account_requests.py
Line	21	28
Object	form	filter

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/account_requests.py
Method	def create_request():

```
....
21.   contact_email = request.form.get('contact_email',
'').strip().lower()
....
28.   existing = AgencyAccountRequest.query.filter(
```

OS Access Violation

Query Path:
Python\Cx\Python Medium Threat\OS Access Violation Version:5

Categories

OWASP Top 10 2017: A5-Broken Access Control
CWE top 25: CWE top 25
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
OWASP Top 10 2021: A3-Injection
ASA Premium: ASA Premium
Top Tier: Top Tier
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development

Description

OS Access Violation\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=36
Status	New
Detection Date	12/1/2025 9:38:32 AM

The user-provided input from files in GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py in line 413 is used by the file operation saved_path in GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py in line 500 without validation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Line	413	500
Object	files	saved_path

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Method def create_donation():

```
....
413.   uploaded_files = request.files.getlist('document_files')
....
500.   os.remove(saved_path)
```

OS Access Violation\Path 2:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=37
Status	New
Detection Date	12/1/2025 9:38:32 AM

The user-provided input from files in GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py in line 1533 is used by the file operation saved_path in GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py in line 1622 without validation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Line	1533	1622
Object	files	saved_path

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donations.py
Method def verify_donation_detail(donation_id):

```
....
1533.   uploaded_files = request.files.getlist('document_files')
....
1622.   os.remove(saved_path)
```

Insufficiently Protected Credentials

Query Path:
Python\Cx\Python Medium Threat\Insufficiently Protected Credentials Version:2

Categories

OWASP Top 10 2013: A2-Broken Authentication and Session Management
OWASP Top 10 2017: A2-Broken Authentication
CWE top 25: CWE top 25
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Security Functions
OWASP ASVS: V02 Authentication
OWASP Top 10 2021: A4-Insecure Design
SANS top 25: SANS top 25
ASA Premium: ASA Premium
PCI DSS v4.0: PCI DSS (4.0) - 8.6.2 Vulnerabilities related to passwords/passphrases usage

Description

Insufficiently Protected Credentials\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=34
Status	New
Detection Date	12/1/2025 9:38:31 AM

Method login at line 223 of GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py gets a user password from the first element. This element’s value then flows through the code without being encrypted and is written to the database in login at line 225 of GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py. This may enable passwords to be stolen by an attacker.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py
Line	223	225
Object	first	password_hash

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py
Method def login():

```
....
223.   user = User.query.filter_by(email=email).first()
....
225.   if user and password and check_password_hash(user.password_hash,
password):
```

Missing HSTS Header

Query Path:
Python\Cx\Python Medium Threat\Missing HSTS Header Version:2

Categories

OWASP ASVS: V14 Configuration
OWASP Top 10 2021: A7-Identification and Authentication Failures
ASA Premium: ASA Premium
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
OWASP Top 10 API 2023: API8-Security Misconfiguration

Description

Missing HSTS Header\Path 1:

Severity	Medium

Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=35
Status	New
Detection Date	12/1/2025 9:38:31 AM

The web-application does not define an HSTS header, leaving it vulnerable to attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py
Line	22	22
Object	app	app

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py
Method	app = Flask(__name__)

```
....
22.   app = Flask(__name__)
```

Missing HSTS Header

Query Path:
JavaScript\Cx\JavaScript Medium Threat\Missing HSTS Header Version:3

Categories

OWASP ASVS: V14 Configuration
OWASP Top 10 2021: A7-Identification and Authentication Failures
ASA Premium: ASA Premium
Base Preset: Base Preset
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development

Description

Missing HSTS Header\Path 1:

Severity	Medium
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=38
Status	New
Detection Date	12/1/2025 9:38:32 AM

The web-application does not define an HSTS header, leaving it vulnerable to attack.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/static/js/approve.js	GOJ_DMIS-feature-hadr-aid-tracking/static/js/approve.js
Line	110	110
Object	json	json

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/static/js/approve.js
Method	.then(response => response.json())

```
....
110.   .then(response => response.json())
```

Trust Boundary Violation in Session Variables

Query Path:
Python\Cx\Python Low Visibility\Trust Boundary Violation in Session Variables Version:4

Categories

OWASP Top 10 2013: A1-Injection
FISMA 2014: System And Information Integrity
NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection
OWASP Top 10 2021: A4-Insecure Design
PCI DSS v3.2.1: PCI DSS (3.2.1) - 6.5.10 - Broken authentication and session management
ASA Premium: ASA Premium
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
ASD STIG 6.1: APSC-DV-002360 - CAT II The application must isolate security functions from non-security functions.

Description

Trust Boundary Violation in Session Variables\Path 1:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=121
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 146 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 156 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Line	146	156
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Method	def create():

```
....
146.   address1_text=request.form.get('address1_text').strip(),
....
156.   db.session.add(custodian)
```

Trust Boundary Violation in Session Variables\Path 2:

Severity	Low
Result State	To Verify

Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=122
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 147 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 156 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Line	147	156
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Method	def create():

```
....
147.   address2_text=request.form.get('address2_text', '').strip() or 
None,
....
156.   db.session.add(custodian)
```

Trust Boundary Violation in Session Variables\Path 3:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=123
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 148 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 156 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Line	148	156
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Method	def create():

```
....
148.   parish_code=request.form.get('parish_code').strip(),
....
156.   db.session.add(custodian)
```

Trust Boundary Violation in Session Variables\Path 4:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=124
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 150 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 156 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Line	150	156
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Method	def create():

```
....
150.   phone_no=request.form.get('phone_no').strip(),
....
156.   db.session.add(custodian)
```

Trust Boundary Violation in Session Variables\Path 5:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=125
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 151 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 156 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py

Line	151	156
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Method def create():

```
....
151.   email_text=request.form.get('email_text', '').strip() or None
....
156.   db.session.add(custodian)
```

Trust Boundary Violation in Session Variables\Path 6:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=126
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 145 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 156 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Line	145	156
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Method def create():

```
....
145.   custodian_name=request.form.get('custodian_name').strip().upper(),
....
156.   db.session.add(custodian)
```

Trust Boundary Violation in Session Variables\Path 7:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=127
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 149 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 156 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Line	149	156
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/custodians.py
Method def create():

```
....
149.   contact_name=request.form.get('contact_name').strip().upper(),
....
156.   db.session.add(custodian)
```

Trust Boundary Violation in Session Variables\Path 8:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=128
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 258 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	258	263
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method def create_warehouse():

```
....
258.   warehouse.reason_desc = request.form.get('reason_desc',
'').strip() or None
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 9:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018

	&projectid=7&pathid=129
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 257 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	257	263
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method	def create_warehouse():

```
....
257.   warehouse.status_code = request.form.get('status_code').strip()
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 10:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=130
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 256 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	256	263
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method	def create_warehouse():

```
....
256.   warehouse.custodian_id = int(request.form.get('custodian_id'))
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 11:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=131
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 255 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	255	263
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method def create_warehouse():

```
....
255.   warehouse.email_text = request.form.get('email_text', '').strip()
or None
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 12:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=132
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 254 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py

Line	254	263
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method	def create_warehouse():

```
....
254.   warehouse.phone_no = request.form.get('phone_no').strip()
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 13:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=133
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 252 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	252	263
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method	def create_warehouse():

```
....
252.   warehouse.parish_code = request.form.get('parish_code').strip()
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 14:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=134
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 253 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	253	263
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method def create_warehouse():

```
....
253.   warehouse.contact_name =
request.form.get('contact_name').strip().upper()
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 15:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=135
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 251 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	251	263
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method def create_warehouse():

```
....
251.   warehouse.address2_text = request.form.get('address2_text',
'').strip() or None
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 16:

Severity	Low
Result State	To Verify

Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=136
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 250 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	250	263
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method	def create_warehouse():

```
....
250.  warehouse.address1_text =
request.form.get('address1_text').strip()
....
263.  db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 17:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=137
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 249 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	249	263
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method	def create_warehouse():

```
....
249.   warehouse.warehouse_type =
request.form.get('warehouse_type').strip()
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 18:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=138
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_warehouse at line 248 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_warehouse at line 263 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Line	248	263
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/warehouses.py
Method def create_warehouse():

```
....
248.   warehouse.warehouse_name =
request.form.get('warehouse_name').strip().upper()
....
263.   db.session.add(warehouse)
```

Trust Boundary Violation in Session Variables\Path 19:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=139
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 21 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 62 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-	GOJ_DMIS-feature-hadr-aid-

	tracking/app/features/transfers.py	tracking/app/features/transfers.py
Line	21	62
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Method def create():

```
....
21.   to_warehouse_id = request.form.get('to_warehouse_id', type=int)
....
62.   db.session.add(to_inventory)
```

Trust Boundary Violation in Session Variables\Path 20:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=140
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 22 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 62 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	22	62
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Method def create():

```
....
22.   item_id = request.form.get('item_id', type=int)
....
62.   db.session.add(to_inventory)
```

Trust Boundary Violation in Session Variables\Path 21:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=141
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 22 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 89 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	22	89
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Method def create():

```
....
22.   item_id = request.form.get('item_id', type=int)
....
89.   db.session.add(transfer_item)
```

Trust Boundary Violation in Session Variables\Path 22:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=142
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 23 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 89 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	23	89
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Method def create():

```
....
23.   quantity = request.form.get('quantity', type=float)
....
89.   db.session.add(transfer_item)
```

Trust Boundary Violation in Session Variables\Path 23:

Severity	Low

Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=143
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 24 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 89 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	24	89
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Method	def create():

```
....
24.   uom_code = request.form.get('uom_code')
....
89.   db.session.add(transfer_item)
```

Trust Boundary Violation in Session Variables\Path 24:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=144
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 26 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 89 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	26	89
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Method	def create():

```
....
26.   comments_text = request.form.get('comments_text', '').strip()
....
89.   db.session.add(transfer_item)
```

Trust Boundary Violation in Session Variables\Path 25:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=145
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 25 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 89 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Line	25	89
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/transfers.py
Method	def create():

```
....
25.   transport_mode = request.form.get('transport_mode', '').strip()
....
89.   db.session.add(transfer_item)
```

Trust Boundary Violation in Session Variables\Path 26:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=146
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 105 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 210 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py

Line	105	210
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Method	def create():

```
....
105.   is_active = request.form.get('is_active') == 'on'
....
210.   db.session.add(new_user)
```

Trust Boundary Violation in Session Variables\Path 27:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=147
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 100 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 210 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Line	100	210
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Method	def create():

```
....
100.   first_name = request.form.get('first_name', '').strip()
....
210.   db.session.add(new_user)
```

Trust Boundary Violation in Session Variables\Path 28:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=148
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 101 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 210 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Line	101	210
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Method	def create():

```
....
101.   last_name = request.form.get('last_name', '').strip()
....
210.   db.session.add(new_user)
```

Trust Boundary Violation in Session Variables\Path 29:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=149
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 103 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 210 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Line	103	210
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Method	def create():

```
....
103.   job_title = request.form.get('job_title', '').strip()
....
210.   db.session.add(new_user)
```

Trust Boundary Violation in Session Variables\Path 30:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=150

Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 104 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 210 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Line	104	210
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Method def create():

```
....
104.   phone = request.form.get('phone', '').strip()
....
210.   db.session.add(new_user)
```

Trust Boundary Violation in Session Variables\Path 31:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=151
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 99 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 210 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Line	99	210
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Method def create():

```
....
99.  password = request.form.get('password', '')
....
210.   db.session.add(new_user)
```

Trust Boundary Violation in Session Variables\Path 32:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=152
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 97 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 210 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Line	97	210
Object	form	session

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py

Method def create():

```
....
97.   email = request.form.get('email', '').strip().lower()
....
210.   db.session.add(new_user)
```

Trust Boundary Violation in Session Variables\Path 33:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=153
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 98 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 210 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py
Line	98	210
Object	form	session

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/user_admin.py

Method	def create():

```
....
98.   user_name = request.form.get('user_name', '').strip().upper()[:20]
....
210.   db.session.add(new_user)
```

Trust Boundary Violation in Session Variables\Path 34:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=154
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 227 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 233 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	227	233
Object	form	session

Code Snippet	
File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
227.   donor.phone_no = (request.form.get('phone_no') or '').strip()
....
233.   db.session.add(donor)
```

Trust Boundary Violation in Session Variables\Path 35:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=155
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 228 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 233 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py

Line	228	233
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
228.   donor.email_text = request.form.get('email_text',
'').strip().lower() or None
....
233.   db.session.add(donor)
```

Trust Boundary Violation in Session Variables\Path 36:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=156
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 226 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 233 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	226	233
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
226.   donor.country_id = int(request.form.get('country_id') or 388)
....
233.   db.session.add(donor)
```

Trust Boundary Violation in Session Variables\Path 37:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=157
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 225 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 233 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	225	233
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
225.  donor.address2_text = request.form.get('address2_text',
'').strip() or None
....
233.  db.session.add(donor)
```

Trust Boundary Violation in Session Variables\Path 38:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=158
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 224 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 233 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	224	233
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method def create():

```
....
224.  donor.address1_text = (request.form.get('address1_text') or
'').strip()
....
233.  db.session.add(donor)
```

Trust Boundary Violation in Session Variables\Path 39:

Severity	Low

Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=159
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 223 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 233 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	223	233
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
223.   donor.org_type_desc = request.form.get('org_type_desc',
'').strip() or None
....
233.   db.session.add(donor)
```

Trust Boundary Violation in Session Variables\Path 40:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=160
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 222 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 233 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	222	233
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
222.  donor.donor_name = (request.form.get('donor_name') or
'').strip().upper()
....
233.  db.session.add(donor)
```

Trust Boundary Violation in Session Variables\Path 41:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=161
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create at line 221 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create at line 233 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Line	221	233
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/donors.py
Method	def create():

```
....
221.  donor.donor_code = (request.form.get('donor_code') or
'').strip().upper()
....
233.  db.session.add(donor)
```

Trust Boundary Violation in Session Variables\Path 42:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=162
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_item at line 273 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py gets user input from element form. This element's value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_item at line 318 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-	GOJ_DMIS-feature-hadr-aid-

	tracking/app/features/items.py	tracking/app/features/items.py
Line	273	318
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Method	def create_item():

```
....
273.   status_code = request.form.get('status_code', 'A')
....
318.   db.session.add(item)
```

Trust Boundary Violation in Session Variables\Path 43:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=163
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_item at line 271 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_item at line 318 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Line	271	318
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Method	def create_item():

```
....
271.   issuance_order = request.form.get('issuance_order', 'FIFO')
....
318.   db.session.add(item)
```

Trust Boundary Violation in Session Variables\Path 44:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=164
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_item at line 272 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_item at line 318 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Line	272	318
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Method def create_item():

```
....
272.   comments_text = (request.form.get('comments_text', '') or
'').strip() or None
....
318.   db.session.add(item)
```

Trust Boundary Violation in Session Variables\Path 45:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=165
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_item at line 270 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_item at line 318 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Line	270	318
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Method def create_item():

```
....
270.   can_expire_flag = request.form.get('can_expire_flag') == 'on'
....
318.   db.session.add(item)
```

Trust Boundary Violation in Session Variables\Path 46:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=166
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_item at line 269 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_item at line 318 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Line	269	318
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Method	def create_item():

```
....
269.   is_batched_flag = request.form.get('is_batched_flag') == 'on'
....
318.   db.session.add(item)
```

Trust Boundary Violation in Session Variables\Path 47:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=167
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_item at line 268 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_item at line 318 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Line	268	318
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Method	def create_item():

```
....
268.  storage_desc = (request.form.get('storage_desc', '') or
'').strip() or None
....
318.  db.session.add(item)
```

Trust Boundary Violation in Session Variables\Path 48:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=168
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_item at line 266 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_item at line 318 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Line	266	318
Object	form	session

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Method	def create_item():

```
....
266.  units_size_vary_flag = request.form.get('units_size_vary_flag') ==
'on'
....
318.  db.session.add(item)
```

Trust Boundary Violation in Session Variables\Path 49:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=169
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_item at line 267 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_item at line 318 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-	GOJ_DMIS-feature-hadr-aid-

	tracking/app/features/items.py	tracking/app/features/items.py
Line	267	318
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Method def create_item():

```
....
267.   usage_desc = (request.form.get('usage_desc', '') or '').strip() or
None
....
318.   db.session.add(item)
```

Trust Boundary Violation in Session Variables\Path 50:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=170
Status	New
Detection Date	12/1/2025 9:38:35 AM

Method create_item at line 265 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py gets user input from element form. This element’s value flows through the code without being properly sanitized or validated and is eventually stored in the server-side Session object, in create_item at line 318 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py. This constitutes a Trust Boundary Violation.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Line	265	318
Object	form	session

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/items.py
Method def create_item():

```
....
265.   default_uom_code = request.form.get('default_uom_code',
'').strip()
....
318.   db.session.add(item)
```

Information Exposure Through an Error Message

Query Path:
Python\Cx\Python Low Visibility\Information Exposure Through an Error Message Version:2

Categories

OWASP Top 10 2013: A5-Security Misconfiguration
FISMA 2014: Configuration Management

NIST SP 800-53: SI-11 Error Handling (P2)
OWASP Top 10 2017: A6-Security Misconfiguration
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Error processing
OWASP ASVS: V14 Configuration
OWASP Top 10 2021: A4-Insecure Design
SANS top 25: SANS top 25
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
OWASP Top 10 API 2023: API8-Security Misconfiguration
ASD STIG 6.1: APSC-DV-002570 - CAT II The application must generate error messages that provide information necessary for corrective actions without revealing information that could be exploited by adversaries.

Description

Information Exposure Through an Error Message\Path 1:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=75
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method setup_optimistic_locking, at line 48 of GOJ_DMIS-feature-hadr-aid-tracking/app/core/optimistic_locking.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to warning, in method setup_optimistic_locking of GOJ_DMIS-feature-hadr-aid-tracking/app/core/optimistic_locking.py, line 49.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/core/optimistic_locking.py	GOJ_DMIS-feature-hadr-aid-tracking/app/core/optimistic_locking.py
Line	48	49
Object	e	warning

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/core/optimistic_locking.py
Method	def setup_optimistic_locking(db):

```
....
48.   except Exception as e:
49.   logger.warning(f"Could not configure optimistic locking for
{model_name}: {e}")
```

Information Exposure Through an Error Message\Path 2:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=76
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method get_cached_rate, at line 134 of GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to error, in method get_cached_rate of GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py, line 135.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py
Line	134	135
Object	e	error

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py

Method def get_cached_rate(currency_code: str, rate_date: date) -> Optional[Decimal]:

```
....
134.   except Exception as e:
135.   logger.error(f"Error getting cached rate for {currency_code}:
{e}")
```

Information Exposure Through an Error Message\Path 3:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=77
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method store_rate, at line 182 of GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to error, in method store_rate of GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py, line 184.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py
Line	182	184
Object	e	error

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py

Method def store_rate(currency_code: str, rate_date: date, rate_to_jmd: Decimal,

```
....
182.   except Exception as e:
....
184.   logger.error(f"Error storing rate for {currency_code}: {e}")
```

Information Exposure Through an Error Message\Path 4:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018

	&projectid=7&pathid=78
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method get_donation_currencies, at line 323 of GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to error, in method get_donation_currencies of GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py, line 324.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py
Line	323	324
Object	e	error

Code Snippet	
File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py
Method	def get_donation_currencies() -> List[str]:

```
....
323.   except Exception as e:
324.   logger.error(f"Error getting donation currencies: {e}")
```

Information Exposure Through an Error Message\Path 5:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=79
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method list_cached_rates, at line 386 of GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to error, in method list_cached_rates of GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py, line 387.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py
Line	386	387
Object	e	error

Code Snippet	
File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/services/currency_service.py
Method	def list_cached_rates(limit: int = 50) -> List[CurrencyRate]:

```
....
386.   except Exception as e:
387.   logger.error(f"Error listing cached rates: {e}")
```

Information Exposure Through an Error Message\Path 6:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=80
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method submit_for_dispatch, at line 142 of GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to error, in method submit_for_dispatch of GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.py, line 147.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.py	GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.py
Line	142	147
Object	e	error

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.py
Method	def submit_for_dispatch(

```
....
142.   except SQLAlchemyError as e:
....
147.   logger.error(f"Database error during dispatch for package
{reliefpkg_id}: {str(e)}")
```

Information Exposure Through an Error Message\Path 7:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=81
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method submit_for_dispatch, at line 149 of GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to error, in method submit_for_dispatch of GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.py, line 154.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.p	GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.p

	y	y
Line	149	154
Object	e	error

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/services/dispatch_service.py
Method def submit_for_dispatch(

```
....
149.   except Exception as e:
....
154.   logger.error(f"Unexpected error during dispatch for package
{reliefpkg_id}: {str(e)}")
```

Information Exposure Through an Error Message\Path 8:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=82
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method review_approval, at line 265 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to warning, in method review_approval of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, line 269.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Line	265	269
Object	e	warning

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Method def review_approval(reliefrqst_id):

```
....
265.   except Exception as e:
....
269.   logger.warning(f'Failed to send approval notification: {str(e)}')
```

Information Exposure Through an Error Message\Path 9:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=83
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method cancel_package, at line 354 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to error, in method cancel_package of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, line 360.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Line	354	360
Object	e	error

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Method def cancel_package(reliefpkg_id):

```
....
354.   except Exception as e:
....
360.   logger.error(f'Unexpected error in cancel_package route:
{str(e)}', exc_info=True)
```

Information Exposure Through an Error Message\Path 10:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=84
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method _approve_and_dispatch, at line 700 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to warning, in method _approve_and_dispatch of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, line 704.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Line	700	704
Object	e	warning

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Method def _approve_and_dispatch(relief_request, relief_pkg, relief_request_version, package_version):

```
....
700.   except Exception as e:
....
704.   logger.warning(f'Failed to send approval notification: {str(e)}')
```

Information Exposure Through an Error Message\Path 11:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=85
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method submit_for_dispatch, at line 833 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to warning, in method submit_for_dispatch of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, line 837.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Line	833	837
Object	e	warning

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Method	def submit_for_dispatch(reliefpkg_id):

```
....
833.   except Exception as e:
....
837.   logger.warning(f'Failed to send dispatch notification: {str(e)}')
```

Information Exposure Through an Error Message\Path 12:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=86
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method submit_for_dispatch, at line 844 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to error, in method submit_for_dispatch of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, line 848.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Line	844	848
Object	e	error

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Method	def submit_for_dispatch(reliefpkg_id):

```
....
844.  except Exception as e:
....
848.  logger.error(f'Error in submit_for_dispatch: {str(e)}',
exc_info=True)
```

Information Exposure Through an Error Message\Path 13:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=87
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method _submit_for_approval, at line 1619 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to error, in method _submit_for_approval of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, line 1622.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Line	1619	1622
Object	e	error

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py

Method def _submit_for_approval(relief_request, relief_request_version, package_version):

```
....
1619.  except Exception as e:
....
1622.  logger.error(f'Failed to send LM approval notification:
{str(e)}', exc_info=True)
```

Information Exposure Through an Error Message\Path 14:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=88
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method _send_for_dispatch, at line 1706 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to warning, in method _send_for_dispatch of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, line 1710.

	Source	Destination

File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Line	1706	1710
Object	e	warning

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Method def _send_for_dispatch(relief_request, relief_request_version, package_version):

```
....
1706.   except Exception as e:
....
1710.   logger.warning(f'Failed to send dispatch notification: {str(e)}')
```

Information Exposure Through an Error Message\Path 15:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=89
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method mark_handover, at line 2504 of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, handles an Exception or runtime Error e. During the exception handling code, the application exposes the exception details to warning, in method mark_handover of GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py, line 2508.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py	GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Line	2504	2508
Object	e	warning

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/features/packaging.py
Method def mark_handover(reliefpkg_id):

```
....
2504.   except Exception as e:
....
2508.   logger.warning(f'Failed to send handover notification: {str(e)}')
```

Client Dangerous File Inclusion

Query Path:
JavaScript\Cx\JavaScript Low Visibility\Client Dangerous File Inclusion Version:4

Categories

NIST SP 800-53: SC-18 Mobile Code (P2)
OWASP Top 10 2017: A1-Injection
CWE top 25: CWE top 25
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data

OWASP ASVS: V05 Validation, Sanitization and Encoding
OWASP Top 10 2021: A8-Software and Data Integrity Failures
PCI DSS v3.2.1: PCI DSS (3.2.1) - 6.5.1 - Injection flaws - particularly SQL injection
SANS top 25: SANS top 25
ASA Premium: ASA Premium
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
ASD STIG 6.1: APSC-DV-003300 - CAT II The designer must ensure uncategorized or emerging mobile code is not used in applications.

Description

Client Dangerous File Inclusion\Path 1:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=65
Status	New
Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js", at line 365 of GOJ_DMIS-feature-hadr-aid-tracking/templates/requests/edit_items.html. An attacker might be able to exploit this and cause the application to load arbitrary code.
Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/requests/edit_items.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/requests/edit_items.html
Line	365	365
Object	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/requests/edit_items.html
Method	<script nonce="{{ csp_nonce() }}" src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js" integrity="sha384-5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6" crossorigin="anonymous"></script>

```
....
365.  <script  nonce="{{ csp_nonce() }}"
src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"
integrity="sha384-
5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6"
crossorigin="anonymous"></script>
```

Client Dangerous File Inclusion\Path 2:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=66
Status	New

Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js", at line 272 of GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html. An attacker might be able to exploit this and cause the application to load arbitrary code.
Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Line	272	272
Object	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/reports/funds_donations.html
Method	<script nonce="{{ csp_nonce() }}" src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js" integrity="sha384-5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6" crossorigin="anonymous"></script>

```
....
272.  <script nonce="{{ csp_nonce() }}"
src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"
integrity="sha384-
5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6"
crossorigin="anonymous"></script>
```

Client Dangerous File Inclusion\Path 3:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=67
Status	New
Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js", at line 161 of GOJ_DMIS-feature-hadr-aid-tracking/templates/packaging/create_request_on_behalf.html. An attacker might be able to exploit this and cause the application to load arbitrary code.
Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/packaging/create_request_on_behalf.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/packaging/create_request_on_behalf.html
Line	161	161

Object	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/packaging/create_request_on_behalf.html

Method <script nonce="{{ csp_nonce() }}" src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js" integrity="sha384-5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6" crossorigin="anonymous"></script>

```
....
161.  <script nonce="{{ csp_nonce() }}"
src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"
integrity="sha384-
5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6"
crossorigin="anonymous"></script>
```

Client Dangerous File Inclusion\Path 4:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=68
Status	New
Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js", at line 108 of GOJ_DMIS-feature-hadr-aid-tracking/templates/login.html. An attacker might be able to exploit this and cause the application to load arbitrary code.

Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/login.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/login.html
Line	108	108
Object	"https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"	"https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"

Code Snippet

File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/login.html

Method <script nonce="{{ csp_nonce() }}" src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

```
....
108.  <script  nonce="{{ csp_nonce() }}"
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bund
le.min.js" integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
```

Client Dangerous File Inclusion\Path 5:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=69
Status	New
Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js", at line 162 of GOJ_DMIS-feature-hadr-aid-tracking/templates/events/create.html. An attacker might be able to exploit this and cause the application to load arbitrary code.
Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/events/create.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/events/create.html
Line	162	162
Object	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/events/create.html
Method	<script nonce="{{ csp_nonce() }}" src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js" integrity="sha384-5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6" crossorigin="anonymous"></script>

```
....
162.  <script  nonce="{{ csp_nonce() }}"
src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"
integrity="sha384-
5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6"
crossorigin="anonymous"></script>
```

Client Dangerous File Inclusion\Path 6:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=70
Status	New
Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/flatpickr", at line 398 of GOJ_DMIS-feature-hadr-aid-tracking/templates/donation_intake/verify_form.html. An attacker might be able to exploit this and cause the application to load arbitrary code.
Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/donation_intake/verify_form.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/donation_intake/verify_form.html
Line	398	398
Object	"https://cdn.jsdelivr.net/npm/flatpickr"	"https://cdn.jsdelivr.net/npm/flatpickr"

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/donation_intake/verify_form.html
Method <script src="https://cdn.jsdelivr.net/npm/flatpickr" integrity="sha384-1a8Wi8gJdDPWfEWuJf7Z4p4C6aOqEYw6cqC7DFKfEWDnsjf/dz+pzzCz2fv1ZHzS" crossorigin="anonymous"></script>

```
....
398.  <script src="https://cdn.jsdelivr.net/npm/flatpickr"
integrity="sha384-
1a8Wi8gJdDPWfEWuJf7Z4p4C6aOqEYw6cqC7DFKfEWDnsjf/dz+pzzCz2fv1ZHzS"
crossorigin="anonymous"></script>
```

Client Dangerous File Inclusion\Path 7:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=71
Status	New
Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js", at line 773 of GOJ_DMIS-feature-hadr-aid-tracking/templates/donation_intake/intake_form.html. An attacker might be able to exploit this and cause the application to load arbitrary code.
Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/donation_intake/intake_form.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/donation_intake/intake_form.html
Line	773	773
Object	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/donation_intake/intake_form.html
Method	<script nonce="{{ csp_nonce() }}" src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js" integrity="sha384-5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6" crossorigin="anonymous"></script>

```
....
773.  <script  nonce="{{ csp_nonce() }}"
src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"
integrity="sha384-
5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6"
crossorigin="anonymous"></script>
```

Client Dangerous File Inclusion\Path 8:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=72
Status	New
Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js", at line 308 of GOJ_DMIS-feature-hadr-aid-tracking/templates/dashboard/aid_movement_dashboard.html. An attacker might be able to exploit this and cause the application to load arbitrary code.
Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/dashboard/aid_movement_dashboard.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/dashboard/aid_movement_dashboard.html
Line	308	308
Object	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/dashboard/aid_movement_dashboard.html
Method	<script nonce="{{ csp_nonce() }}" src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js" integrity="sha384-5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6" crossorigin="anonymous"></script>

```
....
308.  <script nonce="{{ csp_nonce() }}"
src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"
integrity="sha384-
5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6"
crossorigin="anonymous"></script>
```

Client Dangerous File Inclusion\Path 9:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=73
Status	New
Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js", at line 769 of GOJ_DMIS-feature-hadr-aid-tracking/templates/base.html. An attacker might be able to exploit this and cause the application to load arbitrary code.
Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/base.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/base.html
Line	769	769
Object	"https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"	"https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/base.html
Method	<script nonce="{{ csp_nonce() }}" src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

```
....
769.  <script  nonce="{{ csp_nonce() }}"
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bund
le.min.js" integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
```

Client Dangerous File Inclusion\Path 10:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=74
Status	New
Detection Date	12/1/2025 9:38:33 AM

The application loads an external library or source code file using "https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js", at line 176 of GOJ_DMIS-feature-hadr-aid-tracking/templates/agency_requests/edit_items.html. An attacker might be able to exploit this and cause the application to load arbitrary code.
Note that the client application retrieves the external JavaScript library from a remote 3rd party server. It might be possible to exploit this trust model and cause the user's browser to load and execute arbitrary code.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/agency_requests/edit_items.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/agency_requests/edit_items.html
Line	176	176
Object	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"	"https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/templates/agency_requests/edit_items.html
Method	<script nonce="{{ csp_nonce() }}" src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js" integrity="sha384-5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6" crossorigin="anonymous"></script>

```
....
176.  <script  nonce="{{ csp_nonce() }}"
src="https://cdn.jsdelivr.net/npm/flatpickr/dist/flatpickr.min.js"
integrity="sha384-
5JqMv4L/Xa0hfvtF06qboNdhvuYXUku9ZrhZh3bSk8VXF0A/RuSLHpLsSV9Zqhl6"
crossorigin="anonymous"></script>
```

Log Forging

Query Path:
Python\Cx\Python Low Visibility\Log Forging Version:4

Categories

FISMA 2014: System And Information Integrity
NIST SP 800-53: AU-9 Protection of Audit Information (P1)
OWASP Top 10 2017: A1-Injection
OWASP ASVS: V07 Error Handling and Logging
OWASP Top 10 2021: A9-Security Logging and Monitoring Failures
ASA Premium: ASA Premium
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
ASD STIG 6.1: APSC-DV-002560 - CAT I The application must not be subject to input handling vulnerabilities.

Description

Log Forging\Path 1:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=94
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method sanitize_query_string at line 165 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py gets user input from element path. This element’s value flows through the code without being properly sanitized or validated, and is eventually used in writing an audit log in sanitize_query_string at line 163 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py.
This may enable Log Forging.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Line	165	163
Object	path	warning

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method	def sanitize_query_string(query_args):

```
....
165.   f"(IP: {request.remote_addr}, Path: {request.path})"
....
163.   logger.warning(
```

Log Forging\Path 2:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=96
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method get_safe_query_params at line 346 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py gets user input from element args. This element’s value flows through the code without being properly sanitized or validated, and is eventually used in writing an audit log in sanitize_query_string at line 163 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py.
This may enable Log Forging.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Line	346	163
Object	args	warning

Code Snippet

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method	def get_safe_query_params():

```
....
346.   sanitized, _ = sanitize_query_string(request.args)
```

▼

File Name	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method	def sanitize_query_string(query_args):

```
....
163.   logger.warning(
```

Log Forging\Path 3:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=98
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method strip_sensitive_query_params at line 208 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py gets user input from element path. This element's value flows through the code without being properly sanitized or validated, and is eventually used in writing an audit log in strip_sensitive_query_params at line 204 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py.
This may enable Log Forging.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Line	208	204
Object	path	warning

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method def strip_sensitive_query_params():

```
....
208.   f"Path: {request.path} | "
....
204.   logger.warning(
```

Log Forging\Path 4:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=100
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method strip_sensitive_query_params at line 209 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py gets user input from element method. This element's value flows through the code without being properly sanitized or validated, and is eventually used in writing an audit log in strip_sensitive_query_params at line 204 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py.
This may enable Log Forging.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_prote	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_prote

	ction.py	ction.py
Line	209	204
Object	method	warning

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method def strip_sensitive_query_params():

```
....
209.   f"Method: {request.method}"
....
204.   logger.warning(
```

Log Forging\Path 5:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=102
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method strip_sensitive_query_params at line 198 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py gets user input from element args. This element’s value flows through the code without being properly sanitized or validated, and is eventually used in writing an audit log in strip_sensitive_query_params at line 204 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py.
This may enable Log Forging.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Line	198	204
Object	args	warning

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method def strip_sensitive_query_params():

```
....
198.   for param_name in request.args.keys():
....
204.   logger.warning(
```

Log Forging\Path 6:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=104
Status	New
Detection Date	12/1/2025 9:38:33 AM

Method decorated_function at line 262 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py gets user input from element args. This element's value flows through the code without being properly sanitized or validated, and is eventually used in writing an audit log in decorated_function at line 267 of GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py.
This may enable Log Forging.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py	GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Line	262	267
Object	args	warning

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/security/query_string_protection.py
Method def decorated_function(*args, **kwargs):

```
....
262.   for param_name in request.args.keys():
....
267.   logger.warning(
```

Missing Content Security Policy

Query Path:
Python\Cx\Python Low Visibility\Missing Content Security Policy Version:1

Categories

OWASP ASVS: V14 Configuration
OWASP Top 10 2021: A7-Identification and Authentication Failures
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development

Description

Missing Content Security Policy\Path 1:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=116
Status	New
Detection Date	12/1/2025 9:38:33 AM

A Content Security Policy is not explicitly defined within the web-application.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py	GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py
Line	22	22
Object	app	app

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/drims_app.py

Method app = Flask(__name__)

```
....
22.   app = Flask(__name__)
```

Permissive Content Security Policy

Query Path:
Python\Cx\Python Low Visibility\Permissive Content Security Policy Version:1

Categories

OWASP ASVS: V14 Configuration
OWASP Top 10 2021: A7-Identification and Authentication Failures
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development

Description

Permissive Content Security Policy\Path 1:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=117
Status	New
Detection Date	12/1/2025 9:38:33 AM

The Content Security Policy header build_csp_header set via add_csp_headers at line 95 of the file GOJ_DMIS-feature-hadr-aid-tracking/app/security/csp.py is overly permissive.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/app/security/csp.py	GOJ_DMIS-feature-hadr-aid-tracking/app/security/csp.py
Line	95	95
Object	build_csp_header	build_csp_header

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/app/security/csp.py
Method def add_csp_headers(response):

```
....
95.   response.headers['Content-Security-Policy'] = build_csp_header()
```

Potential Clickjacking on Legacy Browsers

Query Path:
JavaScript\Cx\JavaScript Low Visibility\Potential Clickjacking on Legacy Browsers Version:1

Categories

CWE top 25: CWE top 25
MOIS(KISA) Secure Coding 2021: MOIS(KISA) Verification and representation of input data
OWASP ASVS: V05 Validation, Sanitization and Encoding
OWASP Top 10 2021: A8-Software and Data Integrity Failures
SANS top 25: SANS top 25
PCI DSS v4.0: PCI DSS (4.0) - 6.2.4 Vulnerabilities in software development
ASD STIG 6.1: APSC-DV-002330 - CAT II The application must protect the confidentiality and integrity of stored information when required by DoD policy or the information owner.

Description

Potential Clickjacking on Legacy Browsers\Path 1:

Severity	Low
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=118
Status	New
Detection Date	12/1/2025 9:38:34 AM

The application does not protect the web page GOJ_DMIS-feature-hadr-aid-tracking/templates/account_requests/list.html from clickjacking attacks in legacy browsers, by using framebusting scripts.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/templates/account_requests/list.html	GOJ_DMIS-feature-hadr-aid-tracking/templates/account_requests/list.html
Line	6	6
Object	<	<

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/templates/account_requests/list.html
Method <div class="container-fluid">

```
....
6.   <div class="container-fluid">
```

Client Potential XSS

Query Path:
JavaScript\Cx\JavaScript Best Coding Practice\Client Potential XSS Version:4

Categories

OWASP Top 10 2013: A3-Cross-Site Scripting (XSS)
OWASP Top 10 2017: A7-Cross-Site Scripting (XSS)
OWASP ASVS: V05 Validation, Sanitization and Encoding
OWASP Top 10 2021: A3-Injection

Description

Client Potential XSS\Path 1:

Severity	Information
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=119
Status	New
Detection Date	12/1/2025 9:38:34 AM

The method finalAllowedCodes.forEach embeds untrusted data in generated output with option, at line 257 of GOJ_DMIS-feature-hadr-aid-tracking/static/js/prepare.js. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.

	Source	Destination

File	GOJ_DMIS-feature-hadr-aid-tracking/static/js/prepare.js	GOJ_DMIS-feature-hadr-aid-tracking/static/js/prepare.js
Line	239	257
Object	value	option

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/static/js/prepare.js
Method function updateAllowedStatusOptions(itemId, allocated, requested) {

```
....
239.   const currentValue = statusDropdown.value;
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/static/js/prepare.js

Method finalAllowedCodes.forEach(code => {

```
....
257.   statusDropdown.appendChild(option);
```

Client Potential XSS\Path 2:

Severity	Information
Result State	To Verify
Online Results	https://thaumas.egovja.com/CxWebClient/ViewerMain.aspx?scanid=1000018&projectid=7&pathid=120
Status	New
Detection Date	12/1/2025 9:38:34 AM

The method finalAllowedCodes.forEach embeds untrusted data in generated output with option, at line 265 of GOJ_DMIS-feature-hadr-aid-tracking/static/js/approve.js. This untrusted data is embedded into the output without proper sanitization or encoding, enabling an attacker to inject malicious code into the generated web-page.

	Source	Destination
File	GOJ_DMIS-feature-hadr-aid-tracking/static/js/approve.js	GOJ_DMIS-feature-hadr-aid-tracking/static/js/approve.js
Line	247	265
Object	value	option

Code Snippet
File Name GOJ_DMIS-feature-hadr-aid-tracking/static/js/approve.js
Method function updateAllowedStatusOptions(itemId, allocated, requested) {

```
....
247.   const currentValue = statusDropdown.value;
```

File Name GOJ_DMIS-feature-hadr-aid-tracking/static/js/approve.js

Method finalAllowedCodes.forEach(code => {

```
....
265.   statusDropdown.appendChild(option);
```

Command Injection

Risk

What might happen

An attacker could run arbitrary system-level OS commands on the application server host. Depending on the application’s OS permissions, these could include:

- File actions (read / create / modify / delete)
- Open a network connection to the attacker’s server
- Start and stop system services
- Modify the running application
- Complete server takeover

Cause

How does it happen

The application runs an OS system-level command to complete it's task, rather than via the application code. The command includes untrusted data, that may be controllable by an attacker. This untrusted string may contain malicious system-level commands engineered by an attacker, which could be executed as though the attacker were running commands directly on the application server.
In this case, the application receives data from the user input, and passes it as a string to the Operating System. This unvalidated data is then executed by the OS as a system command, running with the same system privileges as the application.

General Recommendations

How to avoid it

- Refactor the code to avoid any direct shell command execution. Instead, use platform provided APIs or library calls.
- If it is impossible to remove the command execution, execute only static commands that do not include dynamic, user-controlled data.
- Validate all input, regardless of source. Validation should be based on a whitelist: accept only data fitting a specified format, rather than rejecting bad patterns (blacklist). Parameters should be limited to an allowed character set, and non-validated input should be dropped. In addition to characters, check for:
  - Data type
  - Size
  - Range
  - Format
  - Expected values
- In order to minimize damage as a measure of defense in depth, configure the application to run using a restricted user account that has no unnecessary OS privileges.
- If possible, isolate all OS commands to use a separate dedicated user account that has minimal privileges only for the specific commands and files used by the application, according to the Principle of Least Privilege.
- If absolutely necessary to call a system command or execute an external program with user input, do not use unsafe methods that call the system shell, such as `os.system()` or `popen2.popen4()`.
- Instead, use safer methods such as `subprocess.run()`, with the `shell` parameter set to `False`.
- Always pass the user input as the 2nd element in the `args` list, with the first element in the list set to a hard-coded (or application-controlled) system command or program path.
- Do not pass the user argument as a string parameter, or as the first element in the `args` list.

Source Code Examples

Python

Execute System (Shell) Command With User Input

```
@app.route('/execute')
def execute_user_command_unsafe():
    user_command = request.args.get('command')

      output = os.system(user_command)

      return output
```

Call External Program with Safe Parameters

```
@app.route('/execute')
def execute_command_with_user_argument_safe():
    user_param = request.args.get('Parameter')
      user_param = shlex.quote(user_param)

      proc = subprocess.run( [PATH_TO_EXTERNAL_PROGRAM, user_param], shell=False)

      return proc.returncode
```

Refactor Code to Call Method in External Package

```
@app.route('/execute')
def perform_specific_action_package_api():
    user_param = request.args.get('Parameter')

      api = OpenSysLibrary()
      output = api.specific_action(user_param)

      return output
```

Stored XSS

Risk

What might happen

A successful XSS exploit would allow an attacker to rewrite web pages and insert malicious scripts which would alter the intended output. This could include HTML fragments, CSS styling rules, arbitrary JavaScript, or references to third party code. An attacker could use this to steal users' passwords, collect personal data such as credit card details, provide false information, or run malware. From the victim's point of view, this is performed by the genuine website, and the victim would blame the site for incurred damage.

An attacker could use legitimate access to the application to submit modified data to the application's data-store. This would then be used to construct the returned web page, triggering the attack.

Cause

How does it happen

The application creates web pages that include untrusted data, whether from user input, the application's database, or from other external sources. The untrusted data is embedded directly in the page's HTML, causing the browser to display it as part of the web page. If the input includes HTML fragments or JavaScript, these are displayed too, and the user cannot tell that this is not the intended page. The vulnerability is the result of directly embedding arbitrary data without first encoding it in a format that would prevent the browser from treating it like HTML or code instead of plain text.

In order to exploit this vulnerability, an attacker would load the malicious payload into the data-store, typically via regular forms on other web pages. Afterwards, the application reads this data from the data-store, and embeds it within the web page as displayed for another user.

General Recommendations

How to avoid it

- Fully encode all dynamic data, regardless of source, before embedding it in output.
- Encoding should be context-sensitive. For example:
  - HTML encoding for HTML content
  - HTML Attribute encoding for data output to attribute values
  - JavaScript encoding for server-generated JavaScript
- It is recommended to use the platform-provided encoding functionality, or known security libraries for encoding output.
- Implement a Content Security Policy (CSP) with explicit whitelists for the application's resources only.
- As an extra layer of protection, validate all untrusted data, regardless of source (note this is not a replacement for encoding). Validation should be based on a whitelist: accept only data fitting a specified structure, rather than reject bad patterns. Check for:
  - Data type
  - Size
  - Range
  - Format
  - Expected values
- In the `Content-Type` HTTP response header, explicitly define character encoding (charset) for the entire page.
- Set the `HTTPOnly` flag on the session cookie for "Defense in Depth", to prevent any successful XSS exploits from stealing the cookie.

Source Code Examples

Python

Unsanitized Database Inserted into an HttpRepsonse in Django

```
def myPage(request):
    uid = str(request.GET.get('userId'))
    username = User.objects.get(id=uid)
    welcomeFormat = '<h1>Welcome, {}!</h1>' #Setting the username to
<script>alert(1)</script> will generate an alert, demonstrating stored XSS.
    content =  welcomeFormat.format(username)
    response = HttpResponse(content)
    return response
```

Unsanitized Database Inserted into an HttpRepsonse's Javascript Context in Django, Enabling XSS

```
def myPage(request):
    uid = str(request.GET.get('userId'))
    username = User.objects.get(id=uid)
    welcomeFormat = '<script>alert(`Welcome, {}!`)</script>' #Setting the username to aaaa`-
prompt(1)-`1 will generate a prompt, demonstrating XSS.
    content =  welcomeFormat.format(username)
    response = HttpResponse(content)
    return response
```

HTML Encoded Database Output Inserted into an HttpRepsonse's HTML Context in Django

```
def myPage(request):
    uid = str(request.GET.get('userId'))
    username = html.escape(User.objects.get(id=uid))
    welcomeFormat = '<h1>Welcome, {}!</h1>' #Setting the username to
<script>alert(1)</script> will be replaced by encoded alternative
    content =  welcomeFormat.format(username)
    response = HttpResponse(content)
    return response
```

Second Order SQL Injection

Risk

What might happen

An attacker could directly access all of the system's data. The attacker would likely be able to steal any sensitive information stored by the system, including private user information, credit card details, proprietary business data, and any other secret data. Likewise, the attacker could possibly modify or erase existing data, or even add new bogus data. In some scenarios, it may even be possible to execute code on the database.

In addition to disclosing or altering confidential information directly, this vulnerability might also be used to achieve secondary effects, such as bypassing authentication, subverting security checks, or forging a data trail.

Further increasing the likelihood of exploit is the fact that this flaw is easy for attackers to find, and easy to exploit.

Cause

How does it happen

The application stores and manages data in a database, by submitting a textual SQL query to the database engine for processing. The application creates the query by simple string concatenation, embedding untrusted data. However, there is no separation between data and code; furthermore, the embedded data is neither checked for data type validity nor subsequently sanitized. Thus, the untrusted data could contain SQL commands, or modify the intended query. The database would interpret the altered query and commands as if they originated from the application, and execute them accordingly.

In order to exploit this vulnerability, an attacker would load the malicious payload into the database, typically via forms on other web pages. Afterwards, the application reads this data from the database, and embeds it within the SQL query, as SQL commands.

General Recommendations

How to avoid it

- Validate all untrusted data, regardless of source. Validation should be based on a whitelist: accept only data fitting a specified structure, rather than reject bad patterns.
- In particular, check for:
  - Data type
  - Size
  - Range
  - Format
  - Expected values.
- Restrict access to database objects and functionality, according to the Principle of Least Privilege.
- Do not use dynamically concatenate strings to construct SQL queries.
- Prefer using DB Stored Procedures for all data access, instead of ad-hoc dynamic queries.
- Instead of unsafe string concatenation, use secure database components such as parameterized queries and object bindings (for example, commands and parameters).
- Alternatively, an even better solution is to use an ORM library, in order to pre-define and encapsulate the allowed commands enabled for the application, instead of dynamically accessing the database directly. In this way the code plane and data plane should be isolated from each other.

Source Code Examples

Python

Raw Query DB Without Parameterized Queries

```
def getCustomerHistory(request):
  name = request.GET['name']
  query = 'select * from my_first_app_customer where first_name = %s'
  customer = Customer.objects.raw(query, [name])
  customer_history = 'select * from my_first_app_purchase where name = %s' % customer.name
  qs = Customer.objects.raw(customer_history)
  return HttpResponse(qs)
```

All DB Queries Are Parameterized

```
def getCustomerHistory(request):
  name = request.GET['name']
  query = 'select * from my_first_app_customer where first_name = %s'
  customer = Customer.objects.raw(query, [name])
  customer_history = 'select * from my_first_app_purchase where name = %s'
  qs = Customer.objects.raw(customer_history, [customer.name])
  return HttpResponse(qs)
```

SQL Injection

Risk

What might happen

An attacker could directly access all of the system's data. The attacker would likely be able to steal any sensitive information stored by the system, including private user information, credit card details, proprietary business data, and any other secret data. Likewise, the attacker could possibly modify or erase existing data, or even add new bogus data. In some scenarios, it may even be possible to execute code on the database.

In addition to disclosing or altering confidential information directly, this vulnerability might also be used to achieve secondary effects, such as bypassing authentication, subverting security checks, or forging a data trail.

Further increasing the likelihood of exploit is the fact that this flaw is easy for attackers to find, and easy to exploit.

Cause

How does it happen

The application stores and manages data in a database, by submitting a textual SQL query to the database engine for processing. The application creates the query by simple string concatenation, embedding untrusted data. However, there is no separation between data and code; furthermore, the embedded data is neither checked for data type validity nor subsequently sanitized. Thus, the untrusted data could contain SQL commands, or modify the intended query. The database would interpret the altered query and commands as if they originated from the application, and execute them accordingly.

Note that an attacker can exploit this vulnerability either by modifying the URL, or by submitting malicious data in the user input or other request fields.

General Recommendations

How to avoid it

- Validate all untrusted data, regardless of source. Validation should be based on a whitelist: accept only data fitting a specified structure, rather than reject bad patterns.
- In particular, check for:
  - Data type
  - Size
  - Range
  - Format
  - Expected values.
- Restrict access to database objects and functionality, according to the Principle of Least Privilege.
- Do not use dynamically concatenate strings to construct SQL queries.
- Prefer using DB Stored Procedures for all data access, instead of ad-hoc dynamic queries.
- Instead of unsafe string concatenation, use secure database components such as parameterized queries and object bindings (for example, commands and parameters).
- Alternatively, an even better solution is to use an ORM library, in order to pre-define and encapsulate the allowed commands enabled for the application, instead of dynamically accessing the database directly. In this way the code plane and data plane should be isolated from each other.

Source Code Examples

Python

Query DB Using Flask-SQLAlchemy ORM Method

```
@app.route('/getUsers', methods=['POST', 'GET'])
```

```
def getUsers(query=None):
    query = request.form.get('query')
    return render_template('query.html',
users=User.query.filter_by(username="{}".format(query)))
```

Query DB Using Flask-SQLAlchemy With Raw SQL as Input

```
@app.route('/getUsers', methods=['POST', 'GET'])
def getUsers(query=None):
    query = request.form.get('query')
    return render_template('query.html',
users=User.query.filter(text("user_username={}".format(query))))
```

Reflected XSS

Risk

What might happen

A successful XSS exploit would allow an attacker to rewrite web pages and insert malicious scripts which would alter the intended output. This could include HTML fragments, CSS styling rules, arbitrary JavaScript, or references to third party code. An attacker could use this to steal users' passwords, collect personal data such as credit card details, provide false information, or run malware. From the victim’s point of view, this is performed by the genuine website, and the victim would blame the site for incurred damage.

The attacker could use social engineering to cause the user to send the website modified input, which will be returned in the requested web page.

Cause

How does it happen

The application creates web pages that include untrusted data, whether from user input, the application’s database, or from other external sources. The untrusted data is embedded directly in the page's HTML, causing the browser to display it as part of the web page. If the input includes HTML fragments or JavaScript, these are displayed too, and the user cannot tell that this is not the intended page. The vulnerability is the result of directly embedding arbitrary data without first encoding it in a format that would prevent the browser from treating it like HTML or code instead of plain text.

Note that an attacker can exploit this vulnerability either by modifying the URL, or by submitting malicious data in the user input or other request fields.

General Recommendations

How to avoid it

- Fully encode all dynamic data, regardless of source, before embedding it in output.
- Encoding should be context-sensitive. For example:
  - HTML encoding for HTML content
  - HTML Attribute encoding for data output to attribute values
  - JavaScript encoding for server-generated JavaScript
- It is recommended to use the platform-provided encoding functionality, or known security libraries for encoding output.
- Implement a Content Security Policy (CSP) with explicit whitelists for the application's resources only.
- As an extra layer of protection, validate all untrusted data, regardless of source (note this is not a replacement for encoding). Validation should be based on a whitelist: accept only data fitting a specified structure, rather than reject bad patterns. Check for:
  - Data type
  - Size
  - Range
  - Format
  - Expected values
- In the `Content-Type` HTTP response header, explicitly define character encoding (charset) for the entire page.
- Set the `HTTPOnly` flag on the session cookie for "Defense in Depth", to prevent any successful XSS exploits from stealing the cookie.

Source Code Examples

Python

Outputting Unsanitized User Input into an HttpRepsonse in Django

```
def myPage(request):
    name = str(request.GET.get('name'))
    welcomeFormat = '<h1>Welcome, {}!</h1>' #Providing the payload
name=<script>alert(1)</script> will generate an alert, demonstrating XSS.
    content =  welcomeFormat.format(name)
    response = HttpResponse(content)
    return response
```

Outputting HTML Encoded User Input into an HttpRepsonse's Javascript Context in Django, Enabling XSS

```
def myPage(request):
    name = html.escape(str(request.GET.get('name')))
    welcomeFormat = '<script>alert(`Welcome, {}!`)</script>' #Providing the payload
name=aaaa`-prompt(1)-`1 will generate a prompt, demonstrating XSS.
    content =  welcomeFormat.format(name)
    response = HttpResponse(content)
    return response
```

Outputting HTML Encoded User Input into an HttpRepsonse's HTML Context in Django

```
def myPage(request):
    name = html.escape(str(request.GET.get('name')))
    welcomeFormat = '<h1>Welcome, {}!</h1>'
    content =  welcomeFormat.format(name)
    response = HttpResponse(content)
    return response
```

Insufficiently Protected Credentials

Risk

What might happen

An attacker could steal user credentials, enabling access to user accounts and confidential data.

Cause

How does it happen

User passwords are written to the database without being properly encrypted with a cryptographic hash. The application reads clear passwords straight from the database.

General Recommendations

How to avoid it

Store passwords using a cryptographic hash designed as a password protection scheme, such as:

- Argon2
- bcrypt
- scrypt
- PBKDF2 (with random salt) These need to be configured with an appropriately high work effort.

Source Code Examples

Python

Always Use a Secure Password Protection Scheme To Store Passwords, Such As bcrypt:

```
master_secret_key = getpass('WERsdfvkjerVDSFGRTc')
raw_password = request.form.get('pwd')
salt = bcrypt.gensalt()
combo_password = raw_password + salt + master_secret_key
hashed_password = bcrypt.hashpw(combo_password, salt)
```

For Password Verification, Use The Matching Function:

```
passwd = request.form.get('pwd')
salt = bcrypt.gensalt()
hashed = bcrypt.hashpw(passwd, salt)
if bcrypt.checkpw(passwd, hashed):
    print("match")
else:
    print("does not match")
```

Insecure Hashing Method Used:

```
pwd = request.form.get('pwd')
hashed_pwd = hashlib.md5(pwd.encode('utf-8')).hexdigest()
```

Missing HSTS Header

Risk

What might happen

Failure to set an HSTS header and provide it with a reasonable "max-age" value of at least one year may leave users vulnerable to Man-in-the-Middle attacks.

Cause

How does it happen

Many users browse to websites by simply typing the domain name into the address bar, without the protocol prefix. The browser will automatically assume that the user's intended protocol is HTTP, instead of the encrypted HTTPS protocol.
When this initial request is made, an attacker can perform a Man-in-the-Middle attack and manipulate it to redirect users to a malicious web-site of the attacker's choosing. To protect the user from such an occurence, the HTTP Strict Transport Security (HSTS) header instructs the user's browser to disallow use of an unsecure HTTP connection to the the domain associated with the HSTS header.
Once a browser that supports the HSTS feature has visited a web-site and the header was set, it will no longer allow communicating with the domain over an HTTP connection.
Once an HSTS header was issued for a specific website, the browser is also instructed to prevent users from manually overriding and accepting an untrusted SSL certificate for as long as the "max-age" value still applies. The recommended "max-age" value is for at least one year in seconds, or 31536000.

General Recommendations

How to avoid it

- Before setting the HSTS header - consider the implications it may have:
  - Forcing HTTPS will prevent any future use of HTTP, which could hinder some testing
  - Disabling HSTS is not trivial, as once it is disabled on the site, it must also be disabled on the browser
- Set the HSTS header either explicitly within application code, or using web-server configurations.
- Ensure the "max-age" value for HSTS headers is set to 31536000 to ensure HSTS is strictly enforced for at least one year.
- Include the "includeSubDomains" to maximize HSTS coverage, and ensure HSTS is enforced on all sub-domains under the current domain
  - Note that this may prevent secure browser access to any sub-domains that utilize HTTP; however, use of HTTP is very severe and highly discouraged, even for websites that do not contain any sensitive information, as their contents can still be tampered via Man-in-the-Middle attacks to phish users under the HTTP domain.
- Once HSTS has been enforced, submit the web-application's address to an HSTS preload list - this will ensure that, even if a client is accessing the web-application for the first time (implying HSTS has not yet been set by the web-application), a browser that respects the HSTS preload list would still treat the web-application as if it had already issued an HSTS header. Note that this requires the server to have a trusted SSL certificate, and issue an HSTS header with a maxAge of 1 year (31536000)
- Note that this query is designed to return one result per application. This means that if more than one vulnerable response without an HSTS header is identified, only the first identified instance of this issue will be highlighted as a result. If a misconfigured instance of HSTS is identified (has a short lifespan, or is missing the "includeSubDomains" flag), that result will be flagged. Since HSTS is required to be enforced across the entire application to be considered a secure deployment of HSTS functionality, fixing this issue only where the query highlights this result is likely to produce subsequent results in other sections of the application; therefore, when adding this header via code, ensure it is uniformly deployed across the entire application. If this header is added via configuration, ensure that this configuration applies to the entire application.
- Note that misconfigured HSTS headers that do not contain the recommended max-age value of at least one year or the "includeSubDomains" flag will still return a result for a missing HSTS header.

Source Code Examples

Python

Adding HSTS in Django's settings.py

```
SECURE_HSTS_SECONDS = 31536000, SECURE_HSTS_INCLUDE_SUBDOMAINS = True
```

Setting HSTS Header with Code in Django

```
response['Strict-Transport-Security'] = 'max-age=31536000; includeSubDomains'
```

Setting HSTS Header with Code in Flask

```
@app.after_request
def add_header(response):
    response.headers['Strict-Transport-Security'] = 'max-age=31536000; includeSubDomains'
    return response
```

Wrapping a Flask App with Talisman Will Set HSTS Header By Default

```
app = Flask(__name__)
talisman = Talisman(app)
```

OS Access Violation

Risk

What might happen

An attacker could prepare malicious input data which would cause an access violation, a private data leak, data corruption or a denial of service (DEP violation and application crash)

Cause

How does it happen

The Python's OS module provides a portable interface intended for using host operating system functionality. The interface of the OS module includes operations for creating, deleting and manipulating host files, directories and links.Python's OS module allows arbitrary files access and manipulation. In case an attacker is able to pass a special-crafted input path to the OS module, access violation, information leakage or data corruption may occur.

General Recommendations

How to avoid it

1. Do not perform file manipulation based on inputs received from an untrusted or a user-controlled source.
2. Make sure path to file which is begin manipulated validated properly:
   - Avoid depending on user input for path to file, if possible.
   - Ensure that path to file is fully canonicalized.
   - Restrict access to file path within a specific directory (sandbox).
3. Create a white list of files or directories which can be manipulated safely and allow access to only these files or directories.

Source Code Examples

Python

Application Removes a File Based on User Input

```
import os
import sys

path = sys.stdin.readline()[:-1]
os.remove(path)
```

Application Validates Path to File Provided by User Input Before Deletion

```
import os
import sys

def is_safe_path(basedir, path):
    return os.path.abspath(path).startswith(basedir)

path = sys.stdin.readline()[:-1]

if not is_safe_path('/tmp/userfiles', path):
    sys.stdout.write('Not allowed!\n')
    sys.exit()
```

```
os.remove(path)
```

Missing HSTS Header

Risk

What might happen

Failure to set an HSTS header and provide it with a reasonable "max-age" value of at least one year may leave users vulnerable to Man-in-the-Middle attacks.

Cause

How does it happen

Many users browse to websites by simply typing the domain name into the address bar, without the protocol prefix. The browser will automatically assume that the user's intended protocol is HTTP, instead of the encrypted HTTPS protocol.

When this initial request is made, an attacker can perform a Man-in-the-Middle attack and manipulate it to redirect users to a malicious web-site of the attacker's choosing. To protect the user from such an occurence, the HTTP Strict Transport Security (HSTS) header instructs the user's browser to disallow use of an unsecure HTTP connection to the the domain associated with the HSTS header.

Once a browser that supports the HSTS feature has visited a web-site and the header was set, it will no longer allow communicating with the domain over an HTTP connection.

Once an HSTS header was issued for a specific website, the browser is also instructed to prevent users from manually overriding and accepting an untrusted SSL certificate for as long as the "max-age" value still applies. The recommended "max-age" value is for at least one year in seconds, or 31536000.

General Recommendations

How to avoid it

- Before setting the HSTS header - consider the implications it may have:
    - Forcing HTTPS will prevent any future use of HTTP, which could hinder some testing
    - Disabling HSTS is not trivial, as once it is disabled on the site, it must also be disabled on the browser
- Set the HSTS header either explicitly within application code, or using web-server configurations.
- Ensure the "max-age" value for HSTS headers is set to 31536000 to ensure HSTS is strictly enforced for at least one year.
- Include the "includeSubDomains" to maximize HSTS coverage, and ensure HSTS is enforced on all sub-domains under the current domain
    - Note that this may prevent secure browser access to any sub-domains that utilize HTTP; however, use of HTTP is very severe and highly discouraged, even for websites that do not contain any sensitive information, as their contents can still be tampered via Man-in-the-Middle attacks to phish users under the HTTP domain.
- Once HSTS has been enforced, submit the web-application's address to an HSTS preload list - this will ensure that, even if a client is accessing the web-application for the first time (implying HSTS has not yet been set by the web-application), a browser that respects the HSTS preload list would still treat the web-application as if it had already issued an HSTS header. Note that this requires the server to have a trusted SSL certificate, and issue an HSTS header with a maxAge of 1 year (31536000)
- Note that this query is designed to return one result per application. This means that if more than one vulnerable response without an HSTS header is identified, only the first identified instance of this issue will be highlighted as a result. If a misconfigured instance of HSTS is identified (has a short lifespan, or is missing the "includeSubDomains" flag), that result will be flagged. Since HSTS is required to be enforced across the entire application to be considered a secure deployment of HSTS functionality, fixing this issue only where the query highlights this result is likely to produce subsequent results in other sections of the application; therefore, when adding this header via code, ensure it is uniformly deployed across the entire application. If this header is added via configuration, ensure that this configuration applies to the entire application.
- Note that misconfigured HSTS headers that do not contain the recommended max-age value of at least one year or the "includeSubDomains" flag will still return a result for a missing HSTS header.

Source Code Examples

JavaScript
Using Helmet with Express

```
var express = require('express')
var helmet = require('helmet') // Helmet includes HSTS, defined to one year and with
"includeSubDomains", as a built-in header

var app = express()
app.use(helmet())
```

Using Explicit HSTS Package - Built into Helmet, So Either 'HSTS' or 'Helmet' Can Be Used

```
var hsts = require('hsts')

app.use(hsts({
  maxAge: 31536000,
  includeSubDomains: true // Also enabled by default
}))
```

Explicitly Setting HSTS Header in Code

```
res.setHeader("Strict-Transport-Security", "max-age=31536000; includeSubDomains");
```

Open Redirect

Risk

What might happen

An attacker could use social engineering to get a victim to click a link to the application, so that the user will be immediately redirected to another site of the attacker's choice. An attacker can then craft a destination website to fool the victim; for example - they may craft a phishing website with an identical looking UI as the previous website's login page, and with a similar looking URL, convincing the user to submit their access credentials in the attacker's website. Another example would be a phishing website with an identical UI as that of a popular payment service, convincing the user to submit their payment information.

Cause

How does it happen

The application redirects the user’s browser to a URL provided by a tainted input, without first ensuring that URL leads to a trusted destination, and without warning users that they are being redirected outside of the current site. An attacker could use social engineering to get a victim to click a link to the application with a parameter defining another site to which the application will redirect the user’s browser. Since the user may not be aware of the redirection, they may be under the misconception that the website they are currently browsing can be trusted.

General Recommendations

How to avoid it

1. Ideally, do not allow arbitrary URLs for redirection. Instead, create a mapping from user-provided parameter values to legitimate URLs.
2. If it is necessary to allow arbitrary URLs:
    - For URLs inside the application site, first filter and encode the user-provided parameter, and then either:
        - Create a white-list of allowed URLs inside the application
        - Use variables as a relative URL as an absolute one, by prefixing it with the application site domain - this will ensure all redirection will occur inside the domain
    - For URLs outside the application (if necessary), either:
        - White-list redirection to allowed external domains by first filtering URLs with trusted prefixes. Prefixes must be tested up to the third slash [/] - `scheme://my.trusted.domain.com/`, to prevent evasion. For example, if the third slash [/] is not validated and scheme://my.trusted.domain.com is trusted, the URL scheme://my.trusted.domain.com.evildomain.com would be valid under this filter, but the domain actually being browsed is evildomain.com, not domain.com.
        - For fully dynamic open redirection, use an intermediate disclaimer page to provide users with a clear warning that they are leaving the site.

Source Code Examples

Python

Open Redirection in Flask

```
@app.route("/redirect/")
def redirect():
    redirectUrl = request.args.get('redirect')
    return redirect(redirectUrl)
```

Using Flask's url_for to Ensure Redirected URL is Within the Local URL Router

```python
@app.route("/redirect/")
def redirect():
    redirectUrl = request.args.get('redirect')
    return flask.redirect(url_for(redirectUrl))
```

Open Redirection in Django

```python
def redirectTo(request):
    redirectUrl = request.GET.get('redirectUrl')
    return redirect(redirectUrl)
```

Ensuring Redirection is Only Done to a Trusted Domain

```python
def redirectTo(request):
    redirectUrl = request.GET.get('redirectUrl')
    if redirectUrl.startswith("https://" + TRUSTED_DOMAIN + "/"):
        return redirect(redirectUrl)
    else:
        return redirect("/")
```

Stored Command Argument Injection

Risk

What might happen

The impact of an attacker-controlled argument passed to an external program depends on the functionality, capability, implementation and permissions granted to this program. Invoking a program that allows OS commands may allow command injection, an external compiler or interpreter may allow for code injection or a file path can be manipulated to allow path traversal. In other cases, the external program may itself be vulnerable to attack, such as a buffer overflow.

Cause

How does it happen

A potentially tainted value is passed as an argument to an external program, which is executed by code.

General Recommendations

How to avoid it

- Refactor the code to avoid any direct shell command execution. Instead, use platform provided APIs or library calls.
- If it is impossible to remove the command execution, execute only static commands that do not include dynamic, user-controlled data.
- Validate all input, regardless of source. Validation should be based on a whitelist: accept only data fitting a specified format, rather than rejecting bad patterns (blacklist). Parameters should be limited to an allowed character set, and non-validated input should be dropped. In addition to characters, check for:
  - Data type
  - Size
  - Range
  - Format
  - Expected values
- In order to minimize damage as a measure of defense in depth, configure the application to run using a restricted user account that has no unnecessary OS privileges.
- If possible, isolate all OS commands to use a separate dedicated user account that has minimal privileges only for the specific commands and files used by the application, according to the Principle of Least Privilege.

Source Code Examples

Python

Execute a File By ID From User Input

```
# Django application

def exec_file(request):
    num = request.GET['index']
    file = File.objects.get(pk=num)
    filename = getattr(file, 'filename')
    res = subprocess([EXTERNAL_EXECUTABLE, filename])
    return HttpResponse(res)
```

Parameter Tampering

Risk

What might happen

A malicious user could access other users' information. By requesting information directly, such as by an account number, authorization may be bypassed and the attacker could steal confidential or restricted information (for example, a bank account balance), using a direct object reference.

Cause

How does it happen

The application provides user information without filtering by user ID. For example, it may provide information solely by a submitted account ID. The application concatenates the user input directly into the SQL query string, without any additional filtering. The application also does not perform any validation on the input, nor constrain it to a pre-computed list of acceptable values.

General Recommendations

How to avoid it

Generic Guidance:

- Enforce authorization checks before providing any access to sensitive data, including the specific object reference.
- Explicitly block access to any unauthorized data, especially to other users' data.
- If possible, avoid allowing the user to request arbitrary data by simply sending a record ID. For example, instead of having the user send an account ID, the application should look up the account ID for the current authenticated user session.

Specific Mitigation:

- Do not concatenate user input directly into SQL queries.
- Include a user-specific identifier as a filter in the WHERE clause of the SQL query.
- Map the user input to an indirect reference, e.g. via a prepared list of allowable values.

Source Code Examples

Python

Unfiltered Direct Object Reference

```
@app.route('/getUser', methods=['POST'])
def getUser():
    userID = request.form.get('userID')
    return render_template('display.html',
users=User.query.filter_by(userID="{}".format(userID)))
```

Record References are Now Filtered and Indirect

```
@app.route('/getUser', methods=['POST'])
def getUser():
    index = request.form.get('userID')
    realAccountId = userAccountList[userID]
    userID = session['userID']
    return render_template('query.html',
users=User.query.filter_by(userID="{}".format(userID), accountID="{}".format(realAccountId)))
```

Unchecked Input for Loop Condition

Risk

What might happen

An attacker could input a very high value, keeping application logic busy on very long loops and potentially causing a denial of service (DoS).

Cause

How does it happen

The application performs some repetitive task in a loop, and defines the number of times to perform the loop according to user input. A very high value could cause the application to get stuck in the loop and to be unable to continue to other operations.

General Recommendations

How to avoid it

- Ideally, don’t base a loop on user-provided data.
- If it is necessary to have dynamic values in iterations - any user input must first be validated, and its range should be limited.

Source Code Examples

Python

Loop Condition Is Not Bounded By Any Value

```
@app.route('/fib')
def fib():
    result = [0,1]
    limit = int(request.args.get('limit'))
    for i in range(0, limit):
        result.append(result[i] + result[-1])
    return ','.join(map(str, result))
```

Loop Condition is Bounded With MAX_ITERATION

```
@app.route('/fib')
def fib():
    result = [0,1]
    limit = int(request.args.get('limit'))
    if limit > MAX_ITERATION:
        limit = MAX_ITERATION
    for i in range(0, limit):
        result.append(result[i] + result[-1])
    return ','.join(map(str, result))
```

Client Dangerous File Inclusion

Risk

What might happen

If an attacker can select the name of the library, or the location of the code file that is loaded by the application, they would be able to cause the application to execute arbitrary code. This effectively allows the attacker to control the code run by the application.
This could enable a remote attacker to modify the pages displayed by the user's browser, execute arbitrary code in the context of the web application, and even manipulate or leak any requests sent to the webserver.

Cause

How does it happen

The application uses untrusted data to specify the library or code file, without proper sanitization. This causes the application to load any arbitrary code, as specified. The loaded code will then be executed.
While the URL for the remote code file is defined at development time, it is possible for an untrusted 3rd party hosting the remote code to replace the intended code file with arbitrary JavaScript. Likewise, it is possible the remote server is exploited, and even without their knowledge an attacker might replace their code files.

General Recommendations

How to avoid it

- Do not dynamically load code libraries, especially not based on user input.
- If it is necessary to use untrusted data to select the library to be loaded, verify the selected library name matches a predefined set of whitelisted library names. Alternatively, use the value as an identifier to select from the whitelisted libraries.
- Validate any untrusted data used to load or process libraries or code files by performing an integrity check on the requested resource.
- Specifically, avoid referencing remote third-party scripts in the client application, except for well-known infrastructure libraries, such as jQuery and Angular.

Source Code Examples

JavaScript

Client Remote Script Tag

```
<html>
      <script src="http://remote.thirdparty.com/coolwhiz_functions.js" />

      <body>
      <script language="JavaScript">
             doCoolScript(document.cookie);
        </script>
      </body>
</html>
```

Client Dynamic Script Tag With Integrity Check

```
var url = new URL(window.location.href);
script = document.createElement('script');
script.type = 'text/javascript';
script.src = url.searchParams.get("script");
script.integrity = 'sha256-xNzN2a4ltkB44Mc/Jz3pT4iU1cmeR0FkXs4pru/JxaQ=';
document.getElementsByTagName('head')[0].appendChild(script);
```

Information Exposure Through an Error Message

Risk

What might happen

Exposed details about the application’s environment, users, or associated data (for example, stack trace) could enable an attacker to find another flaw and help the attacker to mount an attack. This may also leak sensitive data, e.g. passwords or database fields.

Cause

How does it happen

The application handles exceptions in an insecure manner, including raw details directly in the error message. This could occur in various ways: by not handling the exception; printing it directly to the output or file; explicitly returning the exception object; or by configuration. These exception details may include sensitive information that could leak to the users due to the occurrence of the runtime error.

General Recommendations

How to avoid it

- Do not expose exception data directly to the output or users, instead return an informative, generic error message. Log the exception details to a dedicated log mechanism.
- Any method that could throw an exception should be wrapped in an exception handling block that:
    - Explicitly handles expected exceptions.
    - Includes a default solution to explicitly handle unexpected exceptions.
- Configure a global handler to prevent unhandled errors from leaving the application.

Source Code Examples

Log Forging

Risk

What might happen

An attacker could engineer audit logs of security-sensitive actions and lay a false audit trail, potentially implicating an innocent user or hiding an incident.

Cause

How does it happen

The application writes audit logs upon security-sensitive actions. Since the audit log includes user input that is neither checked for data type validity nor subsequently sanitized, the input could contain false information made to look like legitimate audit log data,

General Recommendations

How to avoid it

1. Validate all input, regardless of source. Validation should be based on a whitelist: accept only data fitting a specified structure, rather than reject bad patterns. Check for:
    - Data type
    - Size
    - Range
    - Format
    - Expected values
2. Validation is not a replacement for encoding. Fully encode all dynamic data, regardless of source, before embedding it in logs.
3. Use a secure logging mechanism.

Source Code Examples

Python

User Input is Used Directly in Logs

```
@app.route('/login', methods=['POST'])
def login():
    user = request.form.get('username')
    pwd = request.form.get('pwd')
    if doLogin(user, pwd):
        app.logger.info('%s logged in successfully' % user)
        return render_template('index.html', user)
    else:
        app.logger.info('%s failed to log in' % user)
        return 'Failed to log in'
```

Missing Content Security Policy

Risk

What might happen

The Content-Security-Policy header enforces that the source of content, such as the origin of a script, embedded (child) frame, embedding (parent) frame or image, are trusted and allowed by the current web-page; if, within the web-page, a content's source does not adhere to a strict Content Security Policy, it is promptly rejected by the browser. Failure to define a policy may leave the application's users exposed to Cross-Site Scripting (XSS) attacks, Clickjacking attacks, content forgery and more.

Cause

How does it happen

The Content-Security-Policy header is used by modern browsers as an indicator for trusted sources of content, including media, images, scripts, frames and more. If these policies are not explicitly defined, default browser behavior would allow untrusted content.

General Recommendations

How to avoid it

Explicitly set the Content-Security-Policy headers for all applicable policy types (frame, script, form, script, media, img etc.) according to business requirements and deployment layout of external file hosting services. Specifically, do not use a wildcard, '*', to specify these policies, as this would allow content from any external resource.

The Content-Security-Policy can be explicitly defined within web-application code, as a header managed by web-server configurations, or within `<meta>` tags in the HTML `<head>` section.

Source Code Examples

PHP

Restricting Content-Security-Policy to Only Obtain Embedded Content from Current Web-Application

```
<?php
      header("Content-Security-Policy: default-src 'none'; script-src 'self'; connect-src
'self'; img-src 'self'; style-src 'self';");
?>
```

Permissive Content Security Policy

Risk

What might happen

The Content-Security-Policy header enforces that the source of content, such as the origin of a script, embedded (child) frame, embedding (parent) frame or image, are trusted and allowed by the current web-page; if, within the web-page, a content's source does not adhere to a strict Content Security Policy, it is promptly rejected by the browser. Failure to enforce strict content behavior by policy may leave the application's users exposed to Cross-Site Scripting (XSS) attacks, Clickjacking attacks, content forgery and more.

Cause

How does it happen

The Content-Security-Policy header is used by modern browsers as an indicator for trusted sources of content, including media, images, scripts, frames and more. If these policies are broadly defined, they are ineffective in blocking untrusted content.

Application code is used to set a Content-Security-Policy; however, it sets an overly permissive policy.

General Recommendations

How to avoid it

Set the Content-Security-Policy headers for all applicable policy types (frame, frame-ancestors, script, form-actions, script, media, img etc.) according to business requirements and deployment layout of external file hosting services. Specifically, do not use a wildcard, '*', to specify these policies, as this would allow content from any external resource.

The Content-Security-Policy can be explicitly defined within web-application code, as a header managed by web-server configurations, and within `<meta>` tags in HTML pages' `<head>` section.

Source Code Examples

Python

Permissive Content-Security-Policy Set Inline for Talisman

```
csp = {
    'default-src': '*',
    'script-src':  '*'
}
@app.route('/embeddable')
@talisman(content_security_policy=csp)
def embeddable():
    return 'Embeddable'
```

Permissive Content-Security-Policy Configuration Set for Talisman

```
app = Flask(__name__)
csp = {
    'default-src': '*',
    'script-src':  '*'
}
talisman = Talisman(app, content_security_policy=csp)
```

Permissive Content-Security-Policy in Response Header

```
# Django

response = HttpResponse()
response["Content-Security-Policy"] = "default-src *"
```

Content-Security-Policy Only Allows Resources Loaded from Same Origin

```
# Django

response = HttpResponse()
response["Content-Security-Policy"] = "default-src 'self'"
```

Content-Security-Policy Only Allows Resources Loaded from Same Origin

```
# Django

response = HttpResponse()
response.setitem("Content-Security-Policy", "default-src 'self'")
```

Potential Clickjacking on Legacy Browsers

Risk

What might happen

Clickjacking attacks allow an attacker to "hijack" a user's mouse clicks on a webpage, by invisibly framing the application, and superimposing it in front of a bogus site. When the user is convinced to click on the bogus website, e.g. on a link or a button, the user's mouse is actually clicking on the target webpage, despite being invisible.

This could allow the attacker to craft an overlay that, when clicked, would lead the user to perform undesirable actions in the vulnerable application, e.g. enabling the user's webcam, deleting all the user's records, changing the user's settings, or causing clickfraud.

Cause

How does it happen

The root cause of vulnerability to a clickjacking attack, is that the application's web pages can be loaded into a frame of another website. The application does not implement a proper frame-busting script, that would prevent the page from being loaded into another frame. Note that there are many types of simplistic redirection scripts that still leave the application vulnerable to clickjacking techniques, and should not be used.

When dealing with modern browsers, applications mitigate this vulnerability by issuing appropriate Content-Security-Policy or X-Frame-Options headers to indicate to the browser to disallow framing. However, many legacy browsers do not support this feature, and require a more manual approach by implementing a mitigation in Javascript. To ensure legacy support, a framebusting script is required.

General Recommendations

How to avoid it

Generic Guidance:

- Define and implement a a Content Security Policy (CSP) on the server side, including a frame-ancestors directive. Enforce the CSP on all relevant webpages.
- If certain webpages are required to be loaded into a frame, define a specific, whitelisted target URL.
- Alternatively, return a "X-Frame-Options" header on all HTTP responses. If it is necessary to allow a particular webpage to be loaded into a frame, define a specific, whitelisted target URL.
- For legacy support, implement framebusting code using Javascript and CSS to ensure that, if a page is framed, it is never displayed, and attempt to navigate into the frame to prevent attack. Even if navigation fails, the page is not displayed and is therefore not interactive, mitigating potential clickjacking attacks.

Specific Recommendations:

- Implement a proper framebuster script on the client, that is not vulnerable to frame-buster-busting attacks.
    - Code should first disable the UI, such that even if frame-busting is successfully evaded, the UI cannot be clicked. This can be done by setting the CSS value of the "display" attribute to "none" on either the "body" or "html" tags. This is done because, if a frame attempts to redirect and become the parent, the malicious parent can still prevent redirection via various techniques.
    - Code should then determine whether no framing occurs by comparing self === top; if the result is true, can the UI be enabled. If it is false, attempt to navigate away from the framing page by setting the top.location attribute to self.location.

Source Code Examples

JavaScript
Clickjackable Webpage

```
<html>
    <body>

     <button onclick="clicked();">
            Click here if you love ducks
        </button>
    </body>

</html>
```

Bustable Framebuster

```
<html>
    <head>

     <script>
            if ( window.self.location != window.top.location ) {
                   window.top.location = window.self.location;
            }
        </script>
    </head>


    <body>

     <button onclick="clicked();">
            Click here if you love ducks
        </button>
    </body>

</html>
```

Proper Framebusterbusterbusting

```
<html>
    <head>

     <style> html {display : none; } </style>
        <script>
            if ( self === top ) {
                   document.documentElement.style.display = 'block';
            }
            else {
                   top.location = self.location;
            }
        </script>
    </head>


    <body>

     <button onclick="clicked();">
            Click here if you love ducks
        </button>
    </body>

</html>
```

Trust Boundary Violation in Session Variables

Risk

What might happen

Code that reads from Session variables may trust them as server-side variables, but they may have been tainted by user inputs. This can lead to tampering with parameters used to authenticate or authorize users. Further, tainted Session variables offer an additional attack surface against the application - if untrusted data taints a Session variable, and that Session variable is then used elsewhere without sanitization as if it were trusted, it could lead to further attacks such as Cross-Site Scripting, SQL Injection and more.

Cause

How does it happen

Server-side Session variables, or objects, are values assigned to a specific session, which is associated with a specific user. Often, they hold data relevant to that user's session, such as specific identifiers, user-type, authorization, authentication information and more. As such, the paradigm often associated to the Session object is that its contents can be trusted, as users cannot generally set these values themselves.

The application places user input, which is untrusted data, in the server-side Session object, which is considered a trusted location. This could lead developers to treat untrusted data as trusted.

General Recommendations

How to avoid it

1. Validate and sanitize all input, regardless of source. Validation should be based on a whitelist: accept only data fitting a specified structure, rather than reject bad patterns. Check for:
   - Data type
   - Size
   - Range
   - Format
   - Expected values
2. Don’t mix untrusted user input with trusted data.

Source Code Examples

CSharp

Input from the user is added to the current session without sanitizing it

```
public class TrustBoundaryViolation
{
        public void foo()
        {
                string input = Console.ReadLine();
                HttpContext.Current.Session["val"] = input;
        }
}
```

The numbers are extracted from the user inputed data before use

```
public class TrustBoundaryViolationFixed
{
        public void foo()
        {
                string input = Console.ReadLine();
                string inputValue = int.Parse(input).ToString();
                HttpContext.Current.Session["val"] = inputValue;
        }
}
```

Client Potential XSS

Risk

What might happen

A successful XSS exploit would allow an attacker to rewrite web pages and insert malicious scripts which would alter the intended output. This could include HTML fragments, CSS styling rules, arbitrary JavaScript, or references to third party code. An attacker could use this to steal users' passwords, collect personal data such as credit card details, provide false information, or run malware. From the victim’s point of view, this is performed by the genuine website, and the victim would blame the site for incurred damage.
An additional risk with DOM XSS is that, unlike reflected or stored XSS, tainted values do not have to go through the server. Since the server is not involved in sanitization of these inputs, server-side validation is not likely to not be aware XSS attacks have been occurring, and any server-side security solutions, such as a WAF, are likely to be ineffective in DOM XSS mitigation.

Cause

How does it happen

The application creates web pages that include untrusted data, whether from user input, the application’s database, or from other external sources. The untrusted data is embedded directly in the page's HTML, causing the browser to display it as part of the web page. If the input includes HTML fragments or JavaScript, these are displayed too, and the user cannot tell that this is not the intended page. The vulnerability is the result of directly embedding arbitrary data without first encoding it in a format that would prevent the browser from treating it like HTML or code instead of plain text.
When a DOM XSS occurs, it is the client-side code itself that manipulates the local web-page's DOM, extracting data from some client-based storage, introducing potentially malicious content.

General Recommendations

How to avoid it

- Fully encode all dynamic data, regardless of source, before embedding it in output.
- Encoding should be context-sensitive. For example:
  - HTML encoding for HTML content
  - HTML Attribute encoding for data output to attribute values
  - JavaScript encoding for server-generated JavaScript
- It is recommended to use the platform-provided encoding functionality, or known security libraries for encoding output.
- Implement a Content Security Policy (CSP) with explicit whitelists for the application's resources only.
- As an extra layer of protection, validate all untrusted data, regardless of source (note this is not a replacement for encoding). Validation should be based on a whitelist: accept only data fitting a specified structure, rather than reject bad patterns. Check for:
  - Data type
  - Size
  - Range
  - Format
  - Expected values
- In the `Content-Type` HTTP response header, explicitly define character encoding (charset) for the entire page.
- Set the `HTTPOnly` flag on the session cookie for "Defense in Depth", to prevent any successful XSS exploits from stealing the cookie.

Source Code Examples

JavaScript

Stored DOM XSS in img Attribute

```
var imgsrc = localStorage.get("imgsrc");
document.write('<img id="myImage" src=' + imgsrc +' ></img>'); // // If the local storage
value "imgsrc" is set to "1 onerror=alert(1)" will result in an alert prompt, demonstrating
XSS
```

Use Javascript to Construct DOM Elements, Rather Than Manually Concatenating Values

```
var imgsrc = localStorage.get("imgsrc");
var myImg = document.createElement("IMG");
myImg.src = imgsrc;
someDiv.append(myImg);
```

Stored DOM XSS When Using "eval()" to Parse JSON in Javascript

```
var val = localStorage.get("val");
var json = `[{"val": "${val}"}]`;
var obj = eval(json); // If the local storage value "val" is set to ","a":alert(1),"b":" will
result in an alert prompt, demonstrating XSS
```

Replacing "eval()" with "JSON.parse()" to Avoid XSS

```
var val = localStorage.get("val");
var json = `[{"val": "${val}"}]`;
var obj =  JSON.parse(json); // JSON.parse() does not eval JS code
```

DOM XSS in iFrame "src" Attribute

```
var iframeLocation = localStorage.get("iframeLocation");
document.getElementById("myFrame").src = iframeLocation; // If the local storage value
"iframeLocation" is set to "javascript:alert(1)" will result in an alert prompt,
demonstrating XSS. This is also vulnerable to open redirection.
```

Prepending iFrame "src" Attribute to Prevent Malicious URI Schemes

```
var iframeLocation = localStorage.get("iframeLocation");
document.getElementById("myFrame").src = "/example/"+iframeLocation; // Prepending
iframeLocation prevents changing the URI scheme to "javascript:", mitigating XSS
```

Scanned Languages

Language	Hash Number	Change Date
JavaScript	2103811659559524	11/25/2025
VbScript	0742915089703437	11/25/2025
PLSQL	4496420313191342	11/25/2025
Python	0902023474167420	11/25/2025
Common	2250261308531610	11/25/2025