Semgrep SAST Scan Report for Repository: mojito

Report Generated at 2024-02-28 09:28

SAST Scan Summary

Vulnerability Severity	Vulnerability Count
Findings- SAST High Severity	11
Findings- SAST Medium Severity	14
Findings- SAST Low Severity	1

Findings Summary- HIGH Severity

Finding Title	Finding Description & Remediation	severity	state	ref	location
missing-user-entrypoint	By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attacker can control a process running as root, they may have control over the container. Ensure that the last USER in a Dockerfile is a USER other than 'root'.	high	unresolved	master	webapp/src/main/docker/Dockerfile#L6
command-injection-process-builder	A formatted or concatenated string was detected as input to a ProcessBuilder call. This is dangerous if a variable is controlled by user input and could result in a command injection. Ensure your variables are not controlled by users or sufficiently sanitized.	high	unresolved	master	cli/src/main/java/com/box/l10n/mojito/cli/command/RetryCommand.java#L131
command-injection-process-builder	A formatted or concatenated string was detected as input to a ProcessBuilder call. This is dangerous if a variable is controlled by user input and could result in a command injection. Ensure your variables are not controlled by users or sufficiently sanitized.	high	unresolved	master	common/src/main/java/com/box/l10n/mojito/shell/Shell.java#L21
formatted-sql-string	Detected a formatted string in a SQL statement. This could lead to SQL injection if variables in the SQL statement are not properly sanitized. Use a prepared statements (java.sql.PreparedStatement) instead. You can obtain a PreparedStatement using 'connection.prepareStatement'.	high	unresolved	master	webapp/src/main/java/com/box/l10n/mojito/nativecriteria/JpaQueryProvider.java#L17

Finding Title	Finding Description & Remediation	severity	state	ref	location
documentbuilderfactory-disallow-doctype-decl-missing	DOCTYPE declarations are enabled for this DocumentBuilderFactory. This is vulnerable to XML external entity attacks. Disable this by setting the feature "http://apache.org/xml/features/disallow-doctype-decl" to true. Alternatively, allow DOCTYPE declarations and only prohibit external entities declarations. This can be done by setting the features "http://xml.org/sax/features/external-general-entities" and "http://xml.org/sax/features/external-parameter-entities" to false.	high	unresolved	master	common/src/main/java/com/box/l10n/mojito/okapi/filters/XMLFilter.java#L95
documentbuilderfactory-disallow-doctype-decl-missing	DOCTYPE declarations are enabled for this DocumentBuilderFactory. This is vulnerable to XML external entity attacks. Disable this by setting the feature "http://apache.org/xml/features/disallow-doctype-decl" to true. Alternatively, allow DOCTYPE declarations and only prohibit external entities declarations. This can be done by setting the features "http://xml.org/sax/features/external-general-entities" and "http://xml.org/sax/features/external-parameter-entities" to false.	high	unresolved	master	webapp/src/main/java/com/box/l10n/mojito/android/strings/AndroidStringDocumentUtils.java#L15
crlf-injection-logs-deepsemgrep	When data from an untrusted source is put into a logger and not neutralized correctly, an attacker could forge log entries or include malicious content.	high	unresolved	master	webapp/src/main/java/com/box/l10n/mojito/security/ShowPageAuthenticationSuccessHandler.java#L30

Finding Title	Finding Description & Remediation	severity	state	ref	location
var-in-script-tag	Detected a template variable used in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent cross-site scripting (XSS) attacks when used directly in JavaScript. If you need this data on the rendered page, consider placing it in the HTML portion (outside of a script tag). Alternatively, use a JavaScript-specific encoder, such as the one available in OWASP ESAPI. For Django, you may also consider using the 'json_script' template tag and retrieving the data in your script by using the element ID (e.g., `document.getElementById`).	high	unresolved	master	webapp/src/main/resources/templates/index.html#L8
var-in-script-tag	Detected a template variable used in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent cross-site scripting (XSS) attacks when used directly in JavaScript. If you need this data on the rendered page, consider placing it in the HTML portion (outside of a script tag). Alternatively, use a JavaScript-specific encoder, such as the one available in OWASP ESAPI. For Django, you may also consider using the 'json_script' template tag and retrieving the data in your script by using the element ID (e.g., `document.getElementById`).	high	unresolved	master	webapp/src/main/resources/templates/index.html#L9

Finding Title	Finding Description & Remediation	severity	state	ref	location
var-in-script-tag	Detected a template variable used in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent cross-site scripting (XSS) attacks when used directly in JavaScript. If you need this data on the rendered page, consider placing it in the HTML portion (outside of a script tag). Alternatively, use a JavaScript-specific encoder, such as the one available in OWASP ESAPI. For Django, you may also consider using the 'json_script' template tag and retrieving the data in your script by using the element ID (e.g., `document.getElementById`).	high	unresolved	master	webapp/src/main/resources/templates/index.html#L15
var-in-script-tag	Detected a template variable used in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent cross-site scripting (XSS) attacks when used directly in JavaScript. If you need this data on the rendered page, consider placing it in the HTML portion (outside of a script tag). Alternatively, use a JavaScript-specific encoder, such as the one available in OWASP ESAPI. For Django, you may also consider using the 'json_script' template tag and retrieving the data in your script by using the element ID (e.g., `document.getElementById`).	high	unresolved	master	webapp/src/main/resources/templates/index.html#L16

Findings Summary- MEDIUM Severity

Finding Title	Finding Description & Remediation	severity	state	ref	location
unsafe-reflection	If an attacker can supply values that the application then uses to determine which class to instantiate or which method to invoke, the potential exists for the attacker to create control flow paths through the application that were not intended by the application developers. This attack vector may allow the attacker to bypass authentication or access control checks or otherwise cause the application to behave in an unexpected manner.	medium	unresolved	master	cli/src/main/java/com/box/l10n/mojito/cli/command/checks/AbstractCliChecker.java#L16
unsafe-reflection	If an attacker can supply values that the application then uses to determine which class to instantiate or which method to invoke, the potential exists for the attacker to create control flow paths through the application that were not intended by the application developers. This attack vector may allow the attacker to bypass authentication or access control checks or otherwise cause the application to behave in an unexpected manner.	medium	unresolved	master	webapp/src/main/java/com/box/l10n/mojito/service/assetintegritychecker/integritychecker/IntegrityCheckerFactory.java#L84

Finding Title	Finding Description & Remediation	severity	state	ref	location
unsafe-reflection	If an attacker can supply values that the application then uses to determine which class to instantiate or which method to invoke, the potential exists for the attacker to create control flow paths through the application that were not intended by the application developers. This attack vector may allow the attacker to bypass authentication or access control checks or otherwise cause the application to behave in an unexpected manner.	medium	unresolved	master	webapp/src/main/java/com/box/l10n/mojito/service/drop/exporter/DropExporterService.java#L87
unvalidated-redirect	Application redirects to a destination URL specified by a user-supplied parameter that is not validated. This could direct users to malicious locations. Consider using an allowlist to validate URLs.	medium	unresolved	master	webapp/src/main/java/com/box/l10n/mojito/rest/cli/CliWS.java#L52

Finding Title	Finding Description & Remediation	severity	state	ref	location
unrestricted-request-mapping	Detected a method annotated with 'RequestMapping' that does not specify the HTTP method. CSRF protections are not enabled for GET, HEAD, TRACE, or OPTIONS, and by default all HTTP methods are allowed when the HTTP method is not explicitly specified. This means that a method that performs state changes could be vulnerable to CSRF attacks. To mitigate, add the 'method' field and specify the HTTP method (such as 'RequestMethod.POST').	medium	unresolved	master	webapp/src/main/java/com/box/l10n/mojito/react/ReactAppController.java#L58

Finding Title	Finding Description & Remediation	severity	state	ref	location
open-redirect-pathname	The application builds a URL using user-controlled input which can lead to an open redirect vulnerability. An attacker can manipulate the URL and redirect users to an arbitrary domain. Open redirect vulnerabilities can lead to issues such as Cross-site scripting (XSS) or redirecting to a malicious domain for activities such as phishing to capture users' credentials. To prevent this vulnerability perform strict input validation of the domain against an allowlist of approved domains. Notify a user in your application that they are leaving the website. Display a domain where they are redirected to the user. A user can then either accept or deny the redirect to an untrusted site.	medium	unresolved	master	webapp/src/main/resources/public/js/app.js#L152

Finding Title	Finding Description & Remediation	severity	state	ref	location
var-in-script-src	Detected a template variable used as the 'src' in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent malicious URLs from being injected and could results in a cross-site scripting (XSS) vulnerability. Prefer not to dynamically generate the 'src' attribute and use static URLs instead. If you must do this, carefully check URLs against an allowlist and be sure to URL-encode the result.	medium	unresolved	master	webapp/src/main/resources/templates/index.html#L15
var-in-script-src	Detected a template variable used as the 'src' in a script tag. Although template variables are HTML escaped, HTML escaping does not always prevent malicious URLs from being injected and could results in a cross-site scripting (XSS) vulnerability. Prefer not to dynamically generate the 'src' attribute and use static URLs instead. If you must do this, carefully check URLs against an allowlist and be sure to URL-encode the result.	medium	unresolved	master	webapp/src/main/resources/templates/index.html#L16

Finding Title	Finding Description & Remediation	severity	state	ref	location
template-explicit-unescape	Detected an explicit unescape in a Mustache template, using triple braces '{{{...}}}' or ampersand '&'. If external data can reach these locations, your application is exposed to a cross-site scripting (XSS) vulnerability. If you must do this, ensure no external data can reach this location.	medium	unresolved	master	webapp/src/main/resources/templates/index.html#L8
var-in-href	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. If using a relative URL, start with a literal forward slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.	medium	unresolved	master	webapp/src/main/resources/templates/email/sla/openIncident.html#L29
template-href-var	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. Use the 'url' template tag to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.	medium	unresolved	master	webapp/src/main/resources/templates/email/sla/openIncident.html#L29

Finding Title	Finding Description & Remediation	severity	state	ref	location
template-href-var	Detected a template variable used in an anchor tag with the 'href' attribute. This allows a malicious actor to input the 'javascript:' URI and is subject to cross- site scripting (XSS) attacks. Use 'url_for()' to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.	medium	unresolved	master	webapp/src/main/resources/templates/email/sla/openIncident.html#L29
no-new-privileges	Service 'db' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	medium	unresolved	master	docker/docker-compose.yml#L3
writable-filesystem-service	Service 'db' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this service to prevent this.	medium	unresolved	master	docker/docker-compose.yml#L3

Findings Summary- LOW Severity

Finding Title	Finding Description & Remediation	severity	state	ref	location
xml-custom-entityresolver	The application is using an XML parser that has not been safely configured. This might lead to XML External Entity (XXE) vulnerabilities when parsing user-controlled input. An attacker can include document type definitions (DTDs) which can interact with internal or external hosts. XXE can lead to other vulnerabilities, such as Local File Inclusion (LFI), Remote Code Execution (RCE), and Server-side request forgery (SSRF), depending on the application configuration. An attacker can also use DTDs to expand recursively, leading to a Denial-of-Service (DoS) attack, also known as a Billion Laughs Attack. By setting a custom `EntityResolver` for all previous security configurations for are bypassed. It is your responsibility to handle security in the `EntityResolver` implementation instead. For more information, see: [Java XXE prevention](https://semgrep.dev/docs/cheat-sheets/java-xxe/)	low	unresolved	master	common/src/main/java/com/box/l10n/mojito/okapi/filters/XMLFilter.java#L101