Semgrep SAST Scan Report for Repository: Semgrep-Demo/new-project

Report Generated at 2024-09-04 21:52

SAST Scan Summary

Vulnerability Severity	Vulnerability Count
Findings- SAST High Severity	12
Findings- SAST Medium Severity	21
Findings- SAST Low Severity	0

Findings Summary- High Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location
grpc-server-insecure-connection	Found an insecure gRPC server without 'grpc.Creds()' or options with credentials. This allows for a connection without encryption to this server. A malicious attacker could tamper with the gRPC message, which could compromise the machine. Include credentials derived from an SSL certificate in order to create a secure gRPC connection. You can create credentials using 'credentials.NewServerTLSFromFile("cert.pem", "cert.key")'.	high	reviewing	main	src/shippingservice/main.go#L85
grpc-server-insecure-connection	Found an insecure gRPC server without 'grpc.Creds()' or options with credentials. This allows for a connection without encryption to this server. A malicious attacker could tamper with the gRPC message, which could compromise the machine. Include credentials derived from an SSL certificate in order to create a secure gRPC connection. You can create credentials using 'credentials.NewServerTLSFromFile("cert.pem", "cert.key")'.	high	open	main	src/shippingservice/main.go#L88
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	open	main	src/loadgenerator/Dockerfile#L35
	Found an insecure gRPC connection. This creates a connection without encryption to a gRPC client/	high	open	main	src/currencyservice/client.js#L21

Finding Title	Finding Description & Remediation	severity	status	ref	location
grpc-nodejs-insecure-connection	server. A malicious attacker could tamper with the gRPC message, which could compromise the machine.				
arbitrary-sleep	time.sleep() call; did you mean to leave this in?	high	open	main	src/emailservice/email_server.py#L134
arbitrary-sleep	time.sleep() call; did you mean to leave this in?	high	open	main	src/emailservice/email_server.py#L158
arbitrary-sleep	time.sleep() call; did you mean to leave this in?	high	open	main	src/recommendationservice/recommendation_server.py#L61
arbitrary-sleep	time.sleep() call; did you mean to leave this in?	high	open	main	src/recommendationservice/recommendation_server.py#L151
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	open	refs/pull/2/merge	src/assistant-fix-custom-message.java#L14
crlf-injection-logs-deepsemgrep-javaorg-copy	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. Please use the Jsoup.clean() function to sanitize data.	high	open	refs/pull/2/merge	src/assistant-fix-custom-message.java#L14
tainted-sql-string	Detected user input used to manually construct a SQL string. This is usually bad practice because manual construction could accidentally result in a SQL injection. An attacker could use a SQL injection to steal or modify contents of the database. Instead, use a parameterized query which is available by default in most database engines. Alternatively, consider using an object-	high	open	refs/pull/1/merge	src/assistant-fix-sqli-sequelize.ts#L5

Finding Title	Finding Description & Remediation	severity	status	ref	location
	relational mapper (ORM) such as Sequelize which will protect your queries.				
express-sequelize-injection	Detected a sequelize statement that is tainted by user-input. This could lead to SQL injection if the variable is user-controlled and is not properly sanitized. In order to prevent SQL injection, it is recommended to use parameterized queries or prepared statements.	high	open	refs/pull/1/merge	src/assistant-fix-sqli-sequelize.ts#L5

Findings Summary- Medium Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location
path-join-resolve-traversal	Detected possible user input going into a `path.join` or `path.resolve` function. This could possibly lead to a path traversal vulnerability, where the attacker can access arbitrary files stored in the file system. Instead, be sure to sanitize or validate user input first.	medium	open	main	src/paymentservice/server.js#L37
path-join-resolve-traversal	Detected possible user input going into a `path.join` or `path.resolve` function. This could possibly lead to a path traversal vulnerability, where the attacker can access arbitrary files stored in the file system. Instead, be sure to sanitize or validate user input first.	medium	open	main	src/paymentservice/server.js#L38
math-random-used	Do not use `math/rand`. Use `crypto/rand` instead.	medium	open	main	src/frontend/handlers.go#L21
math-random-used	Do not use `math/rand`. Use `crypto/rand` instead.	medium	open	main	src/shippingservice/tracker.go#L19
cookie-missing-httponly	A session cookie was detected without setting the 'HttpOnly' flag. The 'HttpOnly' flag for cookies instructs the browser to forbid client-side scripts from reading the cookie which mitigates XSS attacks. Set the 'HttpOnly' flag by setting 'HttpOnly' to 'true' in the Cookie.	medium	open	main	src/frontend/handlers.go#L418
cookie-missing-httponly	A session cookie was detected without setting the 'HttpOnly' flag. The 'HttpOnly' flag for cookies instructs the browser to forbid client-side scripts from reading the cookie which mitigates XSS attacks. Set the 'HttpOnly' flag by setting 'HttpOnly' to 'true' in the Cookie.	medium	open	main	src/frontend/middleware.go#L97

Finding Title	Finding Description & Remediation	severity	status	ref	location
cookie-missing-secure	A session cookie was detected without setting the 'Secure' flag. The 'secure' flag for cookies prevents the client from transmitting the cookie over insecure channels such as HTTP. Set the 'Secure' flag by setting 'Secure' to 'true' in the Options struct.	medium	open	main	src/frontend/handlers.go#L418
cookie-missing-secure	A session cookie was detected without setting the 'Secure' flag. The 'secure' flag for cookies prevents the client from transmitting the cookie over insecure channels such as HTTP. Set the 'Secure' flag by setting 'Secure' to 'true' in the Options struct.	medium	open	main	src/frontend/middleware.go#L97
use-tls	Found an HTTP server without TLS. Use 'http.ListenAndServeTLS' instead. See https://golang.org/pkg/net/http/#ListenAndServeTLS for more information.	medium	open	main	src/frontend/main.go#L159
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	open	main	src/frontend/templates/ad.html#L21
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	open	main	src/frontend/templates/ad.html#L21
django-no-csrf-token	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks.	medium	open	main	src/frontend/templates/cart.html#L45

Finding Title	Finding Description & Remediation	severity	status	ref	location
django-no-csrf-token	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks.	medium	open	main	src/frontend/templates/header.html#L71
django-no-csrf-token	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks.	medium	open	main	src/frontend/templates/product.html#L38
direct-use-of-jinja2	Detected direct use of jinja2. If not done properly, this may bypass HTML escaping which opens up the application to cross-site scripting (XSS) vulnerabilities. Prefer using the Flask method 'render_template()' and templates with a '.html' extension in order to prevent XSS.	medium	open	main	src/emailservice/email_server.py#L45
direct-use-of-jinja2	Detected direct use of jinja2. If not done properly, this may bypass HTML escaping which opens up the application to cross-site scripting (XSS) vulnerabilities. Prefer using the Flask method 'render_template()' and templates with a '.html' extension in order to prevent XSS.	medium	open	main	src/emailservice/email_server.py#L90
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	open	main	src/frontend/templates/ad.html#L21
string-to-int-signedness-cast	Downcasting or changing sign of an integer with `$CAST_METHOD` method	medium	open	main	src/frontend/handlers.go#L214
context-todo	Consider to use well-defined context	medium	open	main	src/checkoutservice/main.go#L357
	These functions do not allow to set a a timeout value for reading requests. As a result, the app server may be vulnerable to a Slowloris Denial-of-Service (DoS) attack. Slowloris	medium	open	main	src/frontend/main.go#L159

Finding Title	Finding Description & Remediation	severity	status	ref	location
slowloris-dos-functions	attacks exploit the fact that HTTP servers keep the connection active if the request received is incomplete. To mitigate this, implement a `Server` and set the timeout with `ReadHeaderTimeout`.				
crlf-injection-logs	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.	medium	open	refs/pull/2/merge	src/assistant-fix-custom-message.java#L13

Findings Summary- Low Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location