Semgrep SAST Scan Report for Repository: Semgrep-Demo/go-app

Report Generated at 2024-09-04 21:52

SAST Scan Summary

Vulnerability Severity	Vulnerability Count
Findings- SAST High Severity	10
Findings- SAST Medium Severity	60
Findings- SAST Low Severity	0

Findings Summary- High Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location
remote-property-injection	Bracket object notation with user input is present, this might allow an attacker to access all properties of the object and even it's prototype. Use literal values for object properties.	high	open	main	public/js/jquery-3.2.1-min.js#L4
gosql-sqli	Detected string concatenation with a non-literal variable in a "database/sql" Go SQL statement. This could lead to SQL injection if the variable is user-controlled and not properly sanitized. In order to prevent SQL injection, use parameterized queries or prepared statements instead. You can use prepared statements with the 'Prepare' and 'PrepareContext' calls.	high	open	main	util/database/database.go#L24
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	open	main	templates/template.header.html#L13
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	high	open	main	templates/template.header.html#L14

Finding Title	Finding Description & Remediation	severity	status	ref	location
	tag and retrieving the data in your script by using the element ID (e.g., `document.getElementById`).				
err-nil-check	superfluous nil err check before return	high	open	main	setup/function.go#L58
err-nil-check	superfluous nil err check before return	high	open	main	setup/function.go#L74
err-nil-check	superfluous nil err check before return	high	open	main	vulnerability/idor/function.go#L50
err-nil-check	superfluous nil err check before return	high	open	main	vulnerability/idor/function.go#L76
err-nil-check	superfluous nil err check before return	high	open	main	vulnerability/sqli/function.go#L76
deprecated-ioutil-readfile	ioutil.ReadFile is deprecated	high	open	main	util/config/config.go#L24

Findings Summary- Medium Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location
session-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 Options struct.	medium	open	main	user/session/session.go#L28
session-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 Options struct.	medium	open	main	user/session/session.go#L67
session-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	user/session/session.go#L28
session-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	user/session/session.go#L67
use-of-md5	Detected MD5 hash algorithm which is considered insecure. MD5 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.	medium	open	main	user/user.go#L160
use-of-md5	Detected MD5 hash algorithm which is considered insecure. MD5 is not collision resistant and is therefore not	medium	open	main	vulnerability/csa/csa.go#L62

Finding Title	Finding Description & Remediation	severity	status	ref	location
	suitable as a cryptographic signature. Use SHA256 or SHA3 instead.				
use-of-md5	Detected MD5 hash algorithm which is considered insecure. MD5 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.	medium	open	main	vulnerability/idor/idor.go#L164
string-formatted-query	String-formatted SQL query detected. This could lead to SQL injection if the string is not sanitized properly. Audit this call to ensure the SQL is not manipulable by external data.	medium	open	main	util/database/database.go#L24
string-formatted-query	String-formatted SQL query detected. This could lead to SQL injection if the string is not sanitized properly. Audit this call to ensure the SQL is not manipulable by external data.	medium	open	main	vulnerability/sqli/function.go#L37
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	util/cookie.go#L32
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	util/cookie.go#L48
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	util/cookie.go#L32

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	util/cookie.go#L48
formatted-template-string	Found a formatted template string passed to 'template.HTML()'. 'template.HTML()' does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability.	medium	open	main	vulnerability/xss/xss.go#L52
formatted-template-string	Found a formatted template string passed to 'template.HTML()'. 'template.HTML()' does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability.	medium	open	main	vulnerability/xss/xss.go#L53
formatted-template-string	Found a formatted template string passed to 'template.HTML()'. 'template.HTML()' does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability.	medium	open	main	vulnerability/xss/xss.go#L61
formatted-template-string	Found a formatted template string passed to 'template.HTML()'. 'template.HTML()' does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability.	medium	open	main	vulnerability/xss/xss.go#L96
no-direct-write-to-responsewriter	Detected directly writing or similar in 'http.ResponseWriter.write()'. This bypasses HTML escaping that prevents cross-site scripting vulnerabilities.	medium	open	main	util/template.go#L35

Finding Title	Finding Description & Remediation	severity	status	ref	location
	Instead, use the 'html/template' package and render data using 'template.Execute()'.				
unsafe-template-type	Semgrep could not determine that the argument to 'template.HTML()' is a constant. 'template.HTML()' and similar does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability. Instead, do not use this function and use 'template.Execute()'.	medium	open	main	util/template.go#L45
unsafe-template-type	Semgrep could not determine that the argument to 'template.HTML()' is a constant. 'template.HTML()' and similar does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability. Instead, do not use this function and use 'template.Execute()'.	medium	open	main	vulnerability/xss/xss.go#L58
unsafe-template-type	Semgrep could not determine that the argument to 'template.HTML()' is a constant. 'template.HTML()' and similar does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability. Instead, do not use this function and use 'template.Execute()'.	medium	open	main	vulnerability/xss/xss.go#L59
unsafe-template-type	Semgrep could not determine that the argument to 'template.HTML()' is a constant. 'template.HTML()' and similar does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability. Instead, do not use this function and use 'template.Execute()'.	medium	open	main	vulnerability/xss/xss.go#L62

Finding Title	Finding Description & Remediation	severity	status	ref	location
unsafe-template-type	Semgrep could not determine that the argument to 'template.HTML()' is a constant. 'template.HTML()' and similar does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability. Instead, do not use this function and use 'template.Execute()'.	medium	open	main	vulnerability/xss/xss.go#L63
unsafe-template-type	Semgrep could not determine that the argument to 'template.HTML()' is a constant. 'template.HTML()' and similar does not escape contents. Be absolutely sure there is no user-controlled data in this template. If user data can reach this template, you may have a XSS vulnerability. Instead, do not use this function and use 'template.Execute()'.	medium	open	main	vulnerability/xss/xss.go#L100
raw-html-format	Detected user input flowing into a manually constructed HTML string. You may be accidentally bypassing secure methods of rendering HTML by manually constructing HTML and this could create a cross-site scripting vulnerability, which could let attackers steal sensitive user data. Use the `html/template` package which will safely render HTML instead, or inspect that the HTML is rendered safely.	medium	open	main	vulnerability/xss/xss.go#L96
formatted-template-string-taint	Untrusted input could be used to tamper with a web page rendering, which can lead to a Cross-site scripting (XSS) vulnerability. XSS vulnerabilities occur when untrusted input executes malicious JavaScript code, leading to issues such as account compromise and sensitive information leakage. To prevent this vulnerability, validate the user input, perform contextual output encoding or sanitize the input. For more information, see: [Go XSS prevention] (https://semgrep.dev/docs/cheat-sheets/go-xss/).	medium	open	main	vulnerability/xss/xss.go#L100

Finding Title	Finding Description & Remediation	severity	status	ref	location
missing-integrity	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted resource, this could lead to XSS and other types of attacks. To prevent this, include the base64-encoded cryptographic hash of the resource (file) youâ€™re telling the browser to fetch in the 'integrity' attribute for all externally hosted files.	medium	open	main	templates/cart.html#L7
missing-integrity	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted resource, this could lead to XSS and other types of attacks. To prevent this, include the base64-encoded cryptographic hash of the resource (file) youâ€™re telling the browser to fetch in the 'integrity' attribute for all externally hosted files.	medium	open	main	templates/ template.login.html#L7
missing-integrity	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted resource, this could lead to XSS and other types of attacks. To prevent this, include the base64-encoded cryptographic hash of the resource (file) youâ€™re telling the browser to fetch in the 'integrity' attribute for all externally hosted files.	medium	open	main	templates/ template.login.html#L8
		medium	open	main	

Finding Title	Finding Description & Remediation	severity	status	ref	location
plaintext-http-link	This link points to a plaintext HTTP URL. Prefer an encrypted HTTPS URL if possible.				templates/template.idor1.html#L56
plaintext-http-link	This link points to a plaintext HTTP URL. Prefer an encrypted HTTPS URL if possible.	medium	open	main	templates/template.idor2.html#L57
plaintext-http-link	This link points to a plaintext HTTP URL. Prefer an encrypted HTTPS URL if possible.	medium	open	main	templates/template.sqli1.html#L19
plaintext-http-link	This link points to a plaintext HTTP URL. Prefer an encrypted HTTPS URL if possible.	medium	open	main	templates/template.sqli2.html#L18
plaintext-http-link	This link points to a plaintext HTTP URL. Prefer an encrypted HTTPS URL if possible.	medium	open	main	templates/template.sqli2.html#L19
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	templates/setup.sidebar.html#L12
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	templates/setup.sidebar.html#L18
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	medium	open	main	templates/template.sidebar.html#L12

Finding Title	Finding Description & Remediation	severity	status	ref	location
	slash and concatenate the URL, like this: href='/{{link}}'. You may also consider setting the Content Security Policy (CSP) header.				
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	templates/ template.sidebar.html#L18
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	templates/ template.sidebar.html#L27
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	templates/ template.sidebar.html#L28
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	templates/ template.sidebar.html#L36

Finding Title	Finding Description & Remediation	severity	status	ref	location
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	templates/ template.sidebar.html#L37
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	templates/ template.sidebar.html#L44
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	templates/ template.sidebar.html#L45
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	templates/ template.sidebar.html#L56
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)	medium	open	main	templates/ template.sidebar.html#L63

Finding Title	Finding Description & Remediation	severity	status	ref	location
	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.				
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	templates/template.sidebar.html#L68
detect-non-literal-regexp	RegExp() called with a `a` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled input, consider performing input validation or use a regex checking/sanitization library such as https://www.npmjs.com/package/recheck to verify that the regex does not appear vulnerable to ReDoS.	medium	open	main	public/js/jquery-3.2.1-min.js#L2
detect-non-literal-regexp	RegExp() called with a `b` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled input, consider performing input validation or use a regex checking/sanitization library such as https://www.npmjs.com/package/recheck to verify that the regex does not appear vulnerable to ReDoS.	medium	open	main	public/js/jquery-3.2.1-min.js#L3
detect-non-literal-regexp	RegExp() called with a `b` function argument, this might allow an attacker to cause a Regular Expression Denial-of-	medium	open	main	public/js/jquery-3.2.1-min.js#L4

Finding Title	Finding Description & Remediation	severity	status	ref	location
	Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled input, consider performing input validation or use a regex checking/sanitization library such as https://www.npmjs.com/package/recheck to verify that the regex does not appear vulnerable to ReDoS.				
prototype-pollution-loop	Possibility of prototype polluting function detected. By adding or modifying attributes of an object prototype, it is possible to create attributes that exist on every object, or replace critical attributes with malicious ones. This can be problematic if the software depends on existence or non-existence of certain attributes, or uses pre-defined attributes of object prototype (such as hasOwnProperty, toString or valueOf). Possible mitigations might be: freezing the object prototype, using an object without prototypes (via Object.create(null)), blocking modifications of attributes that resolve to object prototype, using Map instead of object.	medium	open	main	public/js/jquery-3.2.1-min.js#L2
prototype-pollution-loop	Possibility of prototype polluting function detected. By adding or modifying attributes of an object prototype, it is possible to create attributes that exist on every object, or replace critical attributes with malicious ones. This can be problematic if the software depends on existence or non-existence of certain attributes, or uses pre-defined attributes of object prototype (such as hasOwnProperty, toString or valueOf). Possible mitigations might be: freezing the object prototype, using an object without prototypes (via Object.create(null)), blocking modifications of attributes that resolve to object prototype, using Map instead of object.	medium	open	main	public/js/jquery-3.2.1-min.js#L2
		medium	open	main	

Finding Title	Finding Description & Remediation	severity	status	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 the 'url' template tag to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.				templates/template.sidebar.html#L28
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	templates/template.sidebar.html#L37
django-no-csrf-token	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks.	medium	open	main	templates/template.login.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 'url_for()' to safely generate a URL. You may also consider setting the Content Security Policy (CSP) header.	medium	open	main	templates/template.sidebar.html#L28
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	templates/template.sidebar.html#L37
no-new-privileges	Service 'db-mysql' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	medium	open	main	docker-compose.yml#L14

Finding Title	Finding Description & Remediation	severity	status	ref	location
writable-filesystem-service	Service 'db-mysql' 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	open	main	docker-compose.yml#L14
formatted-template-string-taint-copy	Untrusted input could be used to tamper with a web page rendering, which can lead to a Cross-site scripting (XSS) vulnerability. XSS vulnerabilities occur when untrusted input executes malicious JavaScript code, leading to issues such as account compromise and sensitive information leakage. To prevent this vulnerability, validate the user input, perform contextual output encoding or sanitize the input. For more information, see: [Go XSS prevention] (https://semgrep.dev/docs/cheat-sheets/go-xss/).	medium	open	main	vulnerability/xss/xss.go#L100

Findings Summary- Low Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location