Semgrep SAST Scan Report for Repository: Semgrep-Demo/sharpcompress

Report Generated at 2024-09-04 21:52

SAST Scan Summary

Vulnerability Severity	Vulnerability Count
Findings- SAST High Severity	4
Findings- SAST Medium Severity	9
Findings- SAST Low Severity	0

Findings Summary- High Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location
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-relational mapper (ORM) such as Sequelize which will protect your queries.	high	open	refs/pull/1/merge	src/assistant-fix-sqli-sequelize.ts#L5
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
narrow-to-wide-string-mismatch	A byte-string (narrow string) is used in an API that expects a wide-string. This can trigger an out-of-bounds read.	medium	open	master	reference/unrar/pathfn.cpp#L165
narrow-to-wide-string-mismatch	A byte-string (narrow string) is used in an API that expects a wide-string. This can trigger an out-of-bounds read.	medium	open	master	reference/unrar/pathfn.cpp#L167
narrow-to-wide-string-mismatch	A byte-string (narrow string) is used in an API that expects a wide-string. This can trigger an out-of-bounds read.	medium	open	master	reference/unrar/pathfn.cpp#L864
narrow-to-wide-string-mismatch	A byte-string (narrow string) is used in an API that expects a wide-string. This can trigger an out-of-bounds read.	medium	open	master	reference/unrar/strfn.cpp#L320
narrow-to-wide-string-mismatch	A byte-string (narrow string) is used in an API that expects a wide-string. This can trigger an out-of-bounds read.	medium	open	master	reference/unrar/strfn.cpp#L323
tainted-allocation-size	Externally controlled data influences the size of an allocation. This can usually lead to overflow or underflow and later trigger an out of bounds conditions.	medium	open	master	reference/unrar/cmddata.cpp#L39
wide-to-narrow-string-mismatch	A wide-string is used in an API that should consume byte-string (narrow string). This can trigger an out-of-bounds read.	medium	open	master	reference/unrar/pathfn.cpp#L790
world-writable-file	This call makes a world-writable file which allows any user on a machine to write to the file. This may allow	medium	open	master	reference/unrar/file.cpp#L192

Finding Title	Finding Description & Remediation	severity	status	ref	location
	attackers to influence the behaviour of this process by writing to the file.				
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