Semgrep SAST Scan Report for Repository: securingsoftware/forum-service

Report Generated at 2024-09-04 21:52

SAST Scan Summary

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

Findings Summary- High Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location
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	update_products	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	update_products	sqli-sequelize.ts#L5
deno-dangerous-run	Detected non-literal calls to Deno.run(). This could lead to a command injection vulnerability.	high	open	main	src/deno-dangerous-run.js#L12
shelljs-exec-injection	If unverified user data can reach the `exec` method it can result in Remote Code Execution	high	open	main	src/shelljs-exec-injection.js#L5

Findings Summary- Medium Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location

Findings Summary- Low Severity

Finding Title	Finding Description & Remediation	severity	status	ref	location