Using shell interpreter when executing OS commands is security-sensitive

Analyze your code

Security Hotspot Major cwe owasp sans-top25

Arbitrary OS command injection vulnerabilities are more likely when a shell is spawned rather than a new process, indeed shell meta-chars can be used (when parameters are user-controlled for instance) to inject OS commands.

Ask Yourself Whether

- OS command name or parameters are user-controlled.

There is a risk if you answered yes to this question.

Recommended Secure Coding Practices

Use functions that don't spawn a shell.

Sensitive Code Example

```
const cp = require('child_process');

// A shell will be spawn in these following cases:
cp.exec(cmd); // Sensitive
cp.execSync(cmd); // Sensitive

cp.spawn(cmd, { shell: true }); // Sensitive
cp.spawnSync(cmd, { shell: true }); // Sensitive
cp.execFile(cmd, { shell: true }); // Sensitive
cp.execFileSync(cmd, { shell: true }); // Sensitive
```

Compliant Solution

```
const cp = require('child_process');

cp.spawnSync("/usr/bin/file.exe", { shell: false }); // Comp
```

See

- OWASP Top 10 2021 Category A3 - Injection
- OWASP Top 10 2017 Category A1 - Injection
- MITRE, CWE-78 - Improper Neutralization of Special Elements used in an OS Command
- SANS Top 25 - Insecure Interaction Between Components

Available In:

sonarcloud | sonarqube

Code Smell

Unnecessary imports should be removed

Code Smell

Boolean literals should not be used in comparisons

Code Smell

Extra semicolons should be removed

Code Smell

Class names should comply with a naming convention