Code Smell (101)





Python static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PYTHON code

R Bug (55)

Compliant Solution

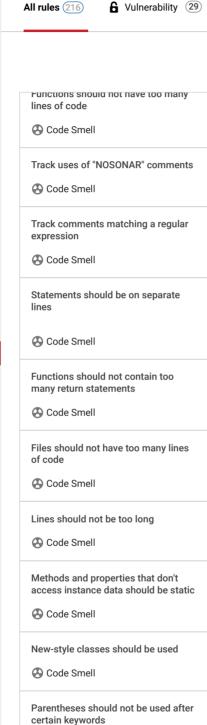
by default shell=False, a shell is not spawn

subprocess.run(cmd) # Compliant

subprocess.Popen(cmd) # Compliant
subprocess.call(cmd) # Compliant
subprocess.check_call(cmd) # Compliant

Python 3

Security Hotspot (31)



A Code Smell

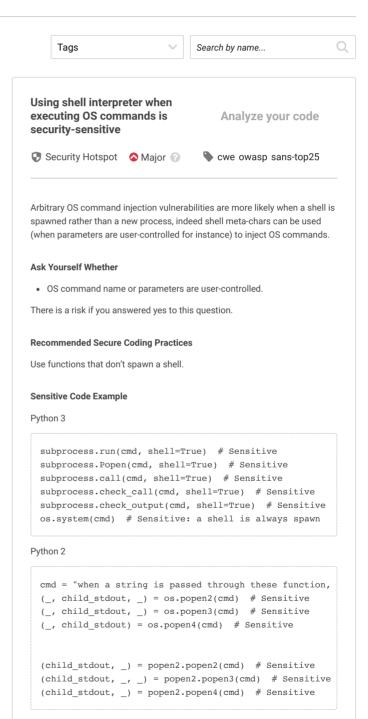
Code Smell

naming convention

person

Track "TODO" and "FIXME" comments that do not contain a reference to a

Module names should comply with a



Code Smell

Comments should not be located at the end of lines of code

Code Smell

Lines should not end with trailing whitespaces

Code Smell

Files should contain an empty newline at the end

Code Smell

Long suffix "L" should be upper case

A Code Smell

```
subprocess.check output(cmd) # Compliant
# always in a subprocess:
os.spawnl(mode, path, *cmd) # Compliant
os.spawnle(mode, path, *cmd, env) # Compliant
os.spawnlp(mode, file, *cmd) # Compliant
os.spawnlpe(mode, file, *cmd, env) # Compliant
os.spawnv(mode, path, cmd) # Compliant
os.spawnve(mode, path, cmd, env) # Compliant
os.spawnvp(mode, file, cmd) # Compliant
os.spawnvpe(mode, file, cmd, env) # Compliant
(child_stdout) = os.popen(cmd, mode, 1) # Compliant
(_, output) = subprocess.getstatusoutput(cmd) # Compli
out = subprocess.getoutput(cmd) # Compliant
os.startfile(path) # Compliant
os.execl(path, *cmd) # Compliant
os.execle(path, *cmd, env) # Compliant
os.execlp(file, *cmd) # Compliant
os.execlpe(file, *cmd, env) # Compliant
os.execv(path, cmd) # Compliant
os.execve(path, cmd, env) # Compliant
os.execvp(file, cmd) # Compliant
os.execvpe(file, cmd, env) # Compliant
```

Python 2

```
cmdsargs = ("use", "a", "sequence", "to", "directly", "

(_, child_stdout) = os.popen2(cmdsargs) # Compliant
(_, child_stdout, _) = os.popen3(cmdsargs) # Compliant
(_, child_stdout) = os.popen4(cmdsargs) # Compliant

(child_stdout, _) = popen2.popen2(cmdsargs) # Complian
(child_stdout, _, _) = popen2.popen3(cmdsargs) # Complian
(child_stdout, _) = popen2.popen4(cmdsargs) # Complian
```

See

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

Deprecated

This rule is deprecated, and will eventually be removed.

Available In:

sonarcloud 👌 | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved. Privacy Policy