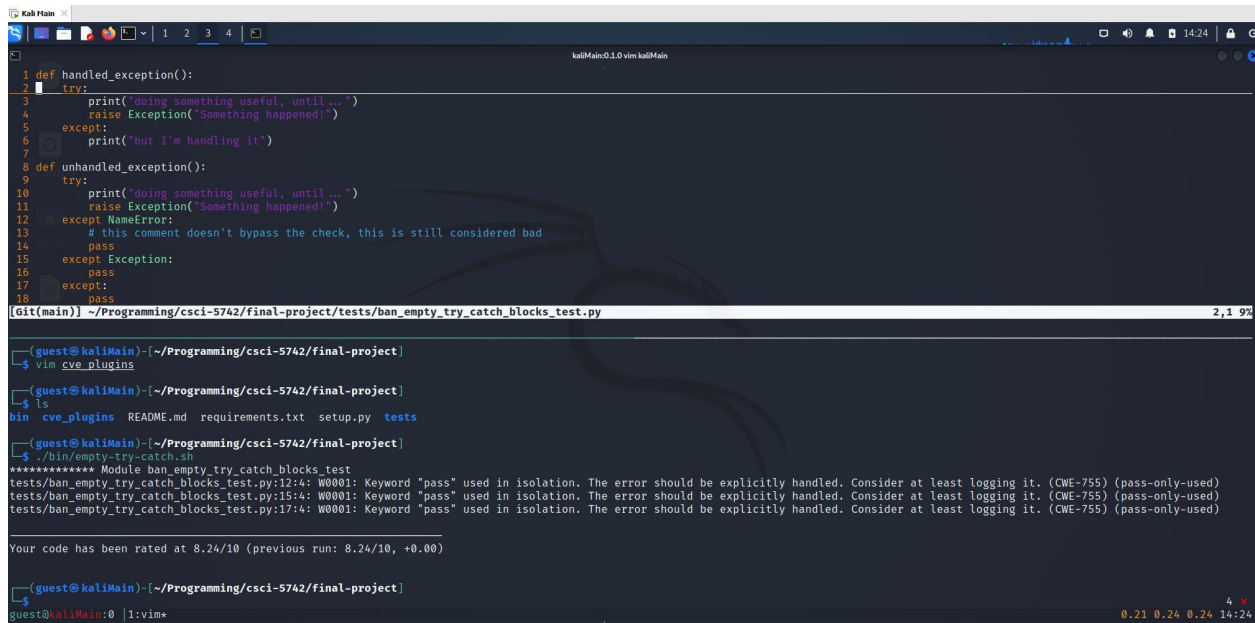


Intro

All screenshots below are taken on the kali linux image provided for the class. The setup instructions from the README.md that is included with the project were followed before execution. Screenshots are of the pylint plugins being executed via shell wrappers provided in the bin directory found in the root of the project. No animals were hurt in the making of these screenshots.

ScreenShots



```
Kali Main
1 def handled_exception():
2     try:
3         print("doing something useful, until...")
4         raise Exception("Something happened!")
5     except:
6         print("but I'm handling it")
7
8 def unhandled_exception():
9     try:
10        print("doing something useful, until...")
11        raise Exception("Something happened!")
12    except NameError:
13        # this comment doesn't bypass the check, this is still considered bad
14        pass
15    except Exception:
16        pass
17    except:
18        pass
19
[git(main)] ~/Programming/csci-5742/final-project/tests/ban_empty_try_catch_blocks_test.py 2,1 9%

(guest@kaliMain)-[~/Programming/csci-5742/final-project]
$ vim cve_plugins

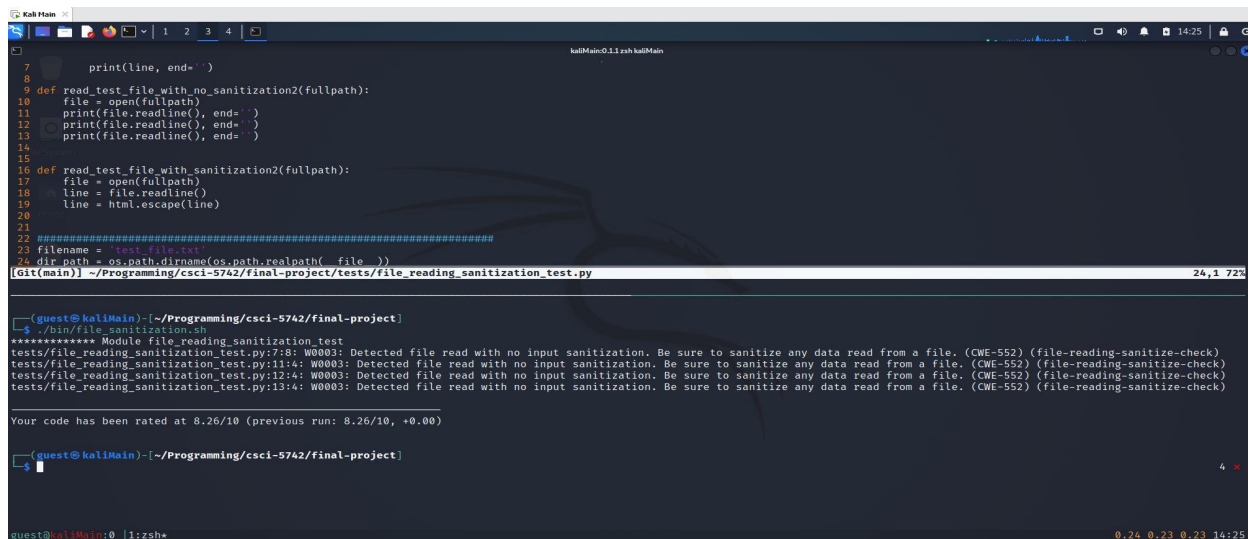
(guest@kaliMain)-[~/Programming/csci-5742/final-project]
$ ls
bin cve_plugins README.md requirements.txt setup.py tests

(guest@kaliMain)-[~/Programming/csci-5742/final-project]
$ ./bin/empty-try-catch.sh
***** Module ban_empty_try_catch_blocks_test
tests/ban_empty_try_catch_blocks_test.py:12:4: W0001: Keyword "pass" used in isolation. The error should be explicitly handled. Consider at least logging it. (CWE-755) (pass-only-used)
tests/ban_empty_try_catch_blocks_test.py:15:4: W0001: Keyword "pass" used in isolation. The error should be explicitly handled. Consider at least logging it. (CWE-755) (pass-only-used)
tests/ban_empty_try_catch_blocks_test.py:17:4: W0001: Keyword "pass" used in isolation. The error should be explicitly handled. Consider at least logging it. (CWE-755) (pass-only-used)

Your code has been rated at 8.24/10 (previous run: 8.24/10, +0.00)

(guest@kaliMain)-[~/Programming/csci-5742/final-project]
$
guest@kaliMain:0 |1:vim+
0.21 0.24 0.24 14:24
```

Figure 1) Empty Try/Except blocks (with pass call only) plugin being called on unit test.



```
Kali Main
7         print(line, end=' ')
8
9 def read_test_file_with_no_sanitization2(fullpath):
10    file = open(fullpath)
11    print(file.readline(), end=' ')
12    print(file.readline(), end=' ')
13    print(file.readline(), end=' ')
14
15
16 def read_test_file_with_sanitization2(fullpath):
17    file = open(fullpath)
18    line = file.readline()
19    line = html.escape(line)
20
21
22 #####
23 filename = 'test_file.txt'
24 dir_path = os.path.dirname(os.path.realpath( file ))
[git(main)] ~/Programming/csci-5742/final-project/tests/file_reading_sanitization_test.py 24,1 72%

(guest@kaliMain)-[~/Programming/csci-5742/final-project]
$ ./bin/file_sanitization.sh
***** Module file_reading_sanitization_test
tests/file_reading_sanitization_test.py:7:8: W0003: Detected file read with no input sanitization. Be sure to sanitize any data read from a file. (CWE-552) (file-reading-sanitize-check)
tests/file_reading_sanitization_test.py:11:4: W0003: Detected file read with no input sanitization. Be sure to sanitize any data read from a file. (CWE-552) (file-reading-sanitize-check)
tests/file_reading_sanitization_test.py:12:4: W0003: Detected file read with no input sanitization. Be sure to sanitize any data read from a file. (CWE-552) (file-reading-sanitize-check)
tests/file_reading_sanitization_test.py:13:4: W0003: Detected file read with no input sanitization. Be sure to sanitize any data read from a file. (CWE-552) (file-reading-sanitize-check)

Your code has been rated at 8.26/10 (previous run: 8.26/10, +0.00)

(guest@kaliMain)-[~/Programming/csci-5742/final-project]
$
guest@kaliMain:0 |1:zsh+
0.24 0.23 0.23 14:25
```

Figure 2) File Sanitization plugin being called on unit test

```
Kali Main
1 import html
2
3 def get_user_input_without_sanitization():
4     input("Please enter something")
5
6 def get_user_input_without_sanitization2():
7     return input("Please enter something")
8
9 def get_user_input_without_sanitization3():
10    user_input = input("Please enter something")
11    return user_input
12
13 def get_user_input_with_sanitization():
14    print("Please enter something")
15    user_input = input()
16    user_input = html.escape(user_input)
17    return user_input
18
19 def get_user_input_with_sanitization2():
20    print("Please enter something")
21    user_input = input()
22    user_input = html.escape(user_input)
23    return user_input
24
25 def get_user_input_with_sanitization3():
26    print("Please enter something")
27    user_input = input()
28    user_input = html.escape(user_input)
29    return user_input
30
[Git(main)] ~/Programming/csci-5742/final-project/tests/input_sanitization_test.py 5,0 16% [Git(main)] ~/Programming/csci-5742/final-project/tests/input_sanitization_test.py 30,0 100%

(guest@kaliMain) ~/Programming/csci-5742/final-project
$ ls bin
empty-try-catch.sh file_sanitization.sh input_sanitization.sh os-subprocess.sh quick-install.sh subprocess.sh
130 x

(guest@kaliMain) ~/Programming/csci-5742/final-project
$ ./bin/input_sanitization.sh
***** Module input_sanitization_test
tests/input_sanitization_test.py:4:4: W0002: The input() method was used, but there doesn't seem to be any input sanitization.It is highly recommended some be added. (CWE-20) (input-sanitiz
e-check)
tests/input_sanitization_test.py:7:4: W0002: The input() method was used, but there doesn't seem to be any input sanitization.It is highly recommended some be added. (CWE-20) (input-sanitiz
e-check)
tests/input_sanitization_test.py:11:4: W0002: The input() method was used, but there doesn't seem to be any input sanitization.It is highly recommended some be added. (CWE-20) (input-sanitiz
e-check)

Your code has been rated at 8.57/10 (previous run: 8.57/10, +0.00)

(guest@kaliMain) ~/Programming/csci-5742/final-project
$
guest@kaliMain:0 |1:zsh*
```

Figure 3) Input Sanitization plugin being called on unit test. Note 1 of the 4 methods performs sanitization, so there should only be 3 warnings.

```
Kali Main
1 """
2 Doc String
3 """
4 from os import system as s
5 from os import fork as f
6
7
8 def from_os_system_test():
9     """
10    This test uses OS.System and should be flagged!
11    """
12    return NULL
13
14 s("echo \"Hello World\"")
15
16 def from_os_system_test_with_input(arbitrary_input):
17     """
18    This test uses OS.System and should be flagged!
19    """
20    return NULL
21
22 s(arbitrary_input)
23
24 def from_os_run_fork():
25     """
26    This test uses OS.Fork and should be flagged!
27    """
28    f()
29
30
31 from os import system
32 from os import fork
33
34 s("echo \"Hello World\"")
35
[Git(main)] ~/Programming/csci-5742/final-project/tests/from_os_system_test.py 1,1 3% [Git(main)] ~/Programming/csci-5742/final-project/tests/from_os_system_test.py 31,21 96%

(guest@kaliMain) ~/Programming/csci-5742/final-project
$ ls bin
empty-try-catch.sh file_sanitization.sh input_sanitization.sh os-subprocess.sh quick-install.sh subprocess.sh
130 x

(guest@kaliMain) ~/Programming/csci-5742/final-project
$ ./bin/os-subprocess.sh
***** Module os_system_test
tests/os_system_test.py:10:7: W0005: "system" should not be used, try multiprocessing. Opens up your program to CWE-78Remember to sanitize your paths when execing paths (CWE-78) (subprocess
-creation-is-banned)
tests/os_system_test.py:19:7: W0005: "system" should not be used, try multiprocessing. Opens up your program to CWE-78Remember to sanitize your paths when execing paths (CWE-78) (subprocess
-creation-is-banned)

Your code has been rated at 7.50/10 (previous run: 7.50/10, +0.00)

(guest@kaliMain) ~/Programming/csci-5742/final-project
$
guest@kaliMain:0 |1:zsh*
```

Figure 4) OS system calls plugin being executed against the unit tests.

```
Kali Main
1 ***
2 A Test file for the ban_arbitrary_execution_subprocess_plugin
3 ***
4 from subprocess import run as s
5 from subprocess import Popen as p
6 # Confounding imports
7 from array import typecodes, array
8 import string as x
9 import math
10
11
12 def run_test(value):
13     """
14     Call subprocess run alias
15     """
16     s(value)
17     print("Done")
18
19
20 def function_ptr_run_test(inputs):
21     """
22     Call Subprocess run alias on inputs map
23     """
24     map(s, inputs)
25
26
27 def Popen_test(value):
28     """
29     Call subprocess Popen alias
30     """
31     p(value)
32
33
34 run_test('echo "hello"')
35 Popen_test('ls -lsh')
```

[Git(main)] ~/Programming/csci-5742/final-project/tests/subprocess_test.py 2,1 5%

Your code has been rated at 7.50/10 (previous run: 7.50/10, +0.00)

```
(guest@kaliMain)-[~/Programming/csci-5742/final-project]
$ :q
:q: command not found
4 x

(guest@kaliMain)-[~/Programming/csci-5742/final-project]
$ ./bin/subprocess.sh
127 x
***** Module subprocess_test
tests/subprocess_test.py:5:23: W0008: "Popen" should not be used, opens up your program to CWE-78 (ban-arbitrary-execution-subprocess)
tests/subprocess_test.py:1:0: W0008: "subprocess.run" should not be used, opens up your program to CWE-78 (ban-arbitrary-execution-subprocess)
tests/subprocess_test.py:1:0: W0008: "subprocess.run" should not be used, opens up your program to CWE-78 (ban-arbitrary-execution-subprocess)

Your code has been rated at 7.86/10 (previous run: 7.86/10, +0.00)
```

```
(guest@kaliMain)-[~/Programming/csci-5742/final-project]
$
0.68 0.36 0.27 14:28
```

Figure 5) Subprocess module usage detection plugin being executed against the unit tests.