

SIV Assignment

In the SIV submission folder, there are two versions of SIV system (one for Python2.7 and another for the Python3.7), but their functions are the same. When I used test_siv.py script to test my SIV code, I found it needs to do some modifications that can be compatible with my SIV. This README.pdf introduces what should be adjusted to follow the real circumstance.

1. Test_siv.py

1.2 For Python2.7 (Ref. siv2/test_siv.py)

-Change the way that print the message into file. For example,

```
with open(fname, "w") as fh:
    print>> fh, text
```

change as python2 version - Tag1

-Function 'print' in Python2 instead of the 'print()' in Python 3.0. Example,

```
print "*" * 10 + "INIT COMPLETED" + "*" * 10
```

-Add 'python2' pre-command into siv_init(sivexec, envdir) and siv_verify(sivexec, envdir) to compatible with my SIV

```
def siv_init(sivexec, envdir):
    siv_cmd = ["python2", sivexec, '-i', '-D', envdir + '/data', '-V', envdir + '/vdb', '-R', envdir + '/init.txt', '-H', 'sha1']
    print ' '.join(siv_cmd)
    # My SIV has to have 'python' as a command- Tag2
    # change as python2 version- Tag3

def siv_verify(sivexec, envdir):
    siv_cmd = ["python2", sivexec, '-v', '-D', envdir + '/data', '-V', envdir + '/vdb', '-R', envdir + '/init.txt']
    # My SIV has to have 'python' as a command- Tag2
    # The report file name has to be same with the report name in initialization mode- Tag6
```

It should also depend on how many versions you have installed in your Ubuntu server. If you just have one default python in your server, please specify the 'python' as the first command place. If your testing sever contains Python2 and Python3, please specify the version, in this case is 'python2'. If you are using Python3, please refer 1.3.

Use the command to change your python version:

```
`sudo vi ~/.bashrc`
```

```
alias python='/usr/bin/python2.7'
```

Save and quit, then `./~/.bashrc`

Testing use `python -V`

-Function 'raw_input()' in Python2 instead of 'input()' in Python3

```
if os.path.exists(envdir):
    overwrite = raw_input('Environment exists! Overwrite (y/n)?')
```

change as python2 version- Tag4

-Function **subprocess.call()** in Python2 instead of the function **subprocess.run()** which run in Python3

```
def siv_init(sivexec, envdir):
    # My SIV has to have 'python'/'python2' as a command- Tag2
    siv_cmd = ['python2', sivexec, '-i', '-D', envdir + '/data', '-V', envdir + '/vDB', '-R', envdir + '/init.txt', '-H', 'sha1']
    print(' '.join(siv_cmd))
    # result = subprocess.run(siv_cmd, stdout=subprocess.PIPE)
    result = subprocess.call(siv_cmd)
    # change as python2 version- Tag5

def siv_verify(sivexec, envdir):
    # My SIV has to have 'python' as a command- Tag2
    #The report file name has to be same with the report name in initialization mode- Tag6
    siv_cmd = ['python2', sivexec, '-v', '-D', envdir + '/data', '-V', envdir + '/vDB', '-R', envdir + '/init.txt']
    #siv_cmd = ['python', sivexec, '-v', '-D', envdir + '/data', '-V', envdir + '/vDB', '-R', envdir + '/verify.txt']
    print(' '.join(siv_cmd))
    # result = subprocess.run(siv_cmd, stdout=subprocess.PIPE)
    result = subprocess.call(siv_cmd)
    # print(result.stdout.decode('utf-8'))
    print(int(result))
    # change as python2 version- Tag5
    # if result.returncode == 0:
    #     print(""*10 + "VERIFY COMPLETED" + ""*10)
    #     return True
    # else:
    #     print(""*10 + "VERIFY FAILED" + ""*10)
    #     return False
```

-Make the report file name is the same in two modes

```
def siv_verify(sivexec, envdir):
    # My SIV has to have 'python' as a command- Tag2
    #The report file name has to be same with the report name in initialization mode- Tag6
    siv_cmd = ['python2', sivexec, '-v', '-D', envdir + '/data', '-V', envdir + '/vDB', '-R', envdir + '/init.txt']
    #siv_cmd = ['python', sivexec, '-v', '-D', envdir + '/data', '-V', envdir + '/vDB', '-R', envdir + '/verify.txt']
    print(' '.join(siv_cmd))
    # result = subprocess.run(siv_cmd, stdout=subprocess.PIPE)
    # result = subprocess.call(siv_cmd)
```

For the system security reason, there is logic judgement: the report file name should be the same as the one which created in verification mode. Because the initialization mode one contains the verification file path, it can help judge whether the verification file has modified or not.

1.3 For Python3.5 (Ref. siv3/test_siv.py)

Change the version as Python3.5, and source it.

```
alias python='/usr/bin/python3.5'
```

-Modify functions **siv_init** and **siv_verif**, supplement **'python3'** as pre-command.

If your server has downloaded and installed two version Python, it might need to specify the 'python3'/'python'. For example,

```
def siv_init(sivexec, envdir):
    siv_cmd = ['python3', sivexec, '-i', '-D', envdir + '/data', '-V', envdir + '/vDB', '-R', envdir + '/init.txt', '-H', 'sha1'] #Tag1
    print(' '.join(siv_cmd))
    result = subprocess.run(siv_cmd)
    print(result)
    if result.returncode == 0:
        print(""*10 + "INIT COMPLETED" + ""*10)
        return True
    else:
        print(""*10 + "INIT FAILED" + ""*10)
        return False

def siv_verify(sivexec, envdir):
    siv_cmd = ['python3', sivexec, '-v', '-D', envdir + '/data', '-V', envdir + '/vDB', '-R', envdir + '/init.txt'] #Tag1
    print(' '.join(siv_cmd))
    result = subprocess.run(siv_cmd)
    print(result)
    if result.returncode == 0:
        print(""*10 + "VERIFY COMPLETED" + ""*10)
        return True
    else:
        print(""*10 + "VERIFY FAILED" + ""*10)
        return False
```

For example, you face the following issue:

```
rongpeng@RONGPENG:~/Documents/SIV/siv3$ ./test_siv.py -s siv.py -e orig
siv.py -i -D orig/data -V orig/vDB -R orig/init.txt -H sha1
Traceback (most recent call last):
  File "./test_siv.py", line 151, in <module>
    main()
  File "./test_siv.py", line 138, in main
    if siv_init(args.sivexec, args.envdir) is False:
  File "./test_siv.py", line 94, in siv_init
    result = subprocess.run(siv_cmd)
              #Tag2
  File "/usr/lib/python3.5/subprocess.py", line 693, in run
    with Popen(*popenargs, **kwargs) as process:
  File "/usr/lib/python3.5/subprocess.py", line 947, in __init__
    restore_signals, start_new_session)
  File "/usr/lib/python3.5/subprocess.py", line 1551, in _execute_child
    raise child_exception_type(errno_num, err_msg)
FileNotFoundError: [Errno 2] No such file or directory: 'siv.py'
```

It must lack of the 'python3'/'python' in siv_init and siv_verif **siv_cmd**.

-Modify the subprocess.run() function, **remove the stdout=subprocess.PIPE** subprocess.PIPE actually provides a buffer for the text stream. But it doesn't really require, specify in here will influence the program running.

And the 'result' print doesn't need to decode as 'utf-8', just let it be simple print.

```
def siv_init(sivexec, envdir):
    siv_cmd = ['python3', sivexec, '-i', '-D', envdir + '/data', '-V', envdir + '/vDB', '-R', envdir + '/init.txt', '-H', 'sha1']
    print(' '.join(siv_cmd))
    result = subprocess.run(siv_cmd)
    print(result)
    if result.returncode == 0:
        print("#"*10 + "INIT COMPLETED" + "#"*10)
        return True
    else:
        print("#"*10 + "INIT FAILED" + "#"*10)
        return False

def siv_verify(sivexec, envdir):
    siv_cmd = ['python3', sivexec, '-v', '-D', envdir + '/data', '-V', envdir + '/vDB', '-R', envdir + '/init.txt']
    print(' '.join(siv_cmd))
    result = subprocess.run(siv_cmd)
    print(result)
    if result.returncode == 0:
        print("#"*10 + "VERIFY COMPLETED" + "#"*10)
        return True
    else:
        print("#"*10 + "VERIFY FAILED" + "#"*10)
        return False
```

Execution:

```

rongpeng@RONGPENG:~/Documents/siv2$ ./test_siv.py -s siv.py -e orig
python2 siv.py -i -D orig/data -V orig/vDB -R orig/init.txt -H sha1
The specified monitored directory exists.
The specified file orig/vDB not exists. SIV created it automatically.
The specified file orig/init.txt not exists. SIV created it automatically.
The specified verification file is outside the monitored directory.
The specified report file is outside the monitored directory.
Notion: The file you entered orig/vDB already existed,do you want to overwrite the existing file? Y/N
n
No Overwrite
Notion: The file you entered orig/init.txt already existed,do you want to overwrite the existing file? Y/N
n
Invalid input
Notion: The file you entered orig/init.txt already existed,do you want to overwrite the existing file? Y/N
n
No Overwrite
0
*****INIT COMPLETED*****
[sudo] password for rongpeng:
/bin/chown: invalid user: 'bob:alice'
python2 siv.py -v -D orig/data -V orig/vDB -R orig/init.txt
The specified verification file exists.
Begin parsing file...
The specified verification file is outside the monitored directory.
The specified report file is outside the monitored directory.
Notion: The file you entered orig/init.txt already existed,do you want to overwrite the existing file? Y/N
n
No Overwrite
Warning:the /home/rongpeng/Documents/siv2/orig/data/a directory access right has changed to 0777 .
Warning:the /home/rongpeng/Documents/siv2/orig/data/x.txt file size has changed to 4.
Warning:the /home/rongpeng/Documents/siv2/orig/data/x.txt file last_modified_date has changed to 1970-01-01 .
Warning:the /home/rongpeng/Documents/siv2/orig/data/b has been removed .
Warning:the /home/rongpeng/Documents/siv2/orig/data/a/blahblah.gz has been removed .
0
*****VERIFY COMPLETED*****
COMPLETED
rongpeng@RONGPENG:~/Documents/siv2$ python -V
Python 2.7.12

```

```

rongpeng@RONGPENG:~/Documents/siv3$ ./test_siv.py -s siv.py -e orig
python3 siv.py -i -D orig/data -V orig/vDB -R orig/init.txt -H sha1
The specified monitored directory exists.
The specified file /home/rongpeng/Documents/siv3/orig/vDB not exists. SIV created it automatically.
The specified file /home/rongpeng/Documents/siv3/orig/init.txt not exists. SIV created it automatically.
The specified verification file is outside the monitored directory.
The specified report file is outside the monitored directory.
Notion: The file you entered orig/vDB already existed,do you want to overwrite the existing file? Y/N
n
No Overwrite
Notion: The file you entered orig/init.txt already existed,do you want to overwrite the existing file? Y/N
n
No Overwrite
CompletedProcess(args=['python3', 'siv.py', '-i', '-D', 'orig/data', '-V', 'orig/vDB', '-R', 'orig/init.txt', '-H', 'sha1'], returncode=0)
*****INIT COMPLETED*****
[sudo] password for rongpeng:
/bin/chown: invalid user: 'bob:alice'
python3 siv.py -v -D orig/data -V orig/vDB -R orig/init.txt
The specified verification file exists.
Begin parsing file...
The specified verification file is outside the monitored directory.
The specified report file is outside the monitored directory.
Notion: The file you entered orig/init.txt already existed,do you want to overwrite the existing file? Y/N
n
No Overwrite
Warning:the /home/rongpeng/Documents/siv3/orig/data/a directory access right has changed to 0o777 .
Warning:the /home/rongpeng/Documents/siv3/orig/data/x.txt file size has changed to 4.
Warning:the /home/rongpeng/Documents/siv3/orig/data/x.txt file last_modified_date has changed to 1970-01-01 .
Warning:the /home/rongpeng/Documents/siv3/orig/data/b has been removed .
Warning:the /home/rongpeng/Documents/siv3/orig/data/a/blahblah.gz has been removed .
CompletedProcess(args=['python3', 'siv.py', '-v', '-D', 'orig/data', '-V', 'orig/vDB', '-R', 'orig/init.txt'], returncode=0)
*****VERIFY COMPLETED*****
COMPLETED
rongpeng@RONGPENG:~/Documents/siv3$ python -V
Python 3.5.2

```

2. Decision Choice

In the SIV code, if you want to see the initial record which written in verification file and report.txt, please choose **'N'/'n'** for most of cases when ask if want to overwrite the file.

3. Command format

For initialization mode,

python siv.py -i -D <Mon_Dir> -V <Ver_Path> -R <Rep_Path> -H sha1/md5

For verification mode,

```
python siv.py -v -D <Mon_Dir> -V <Ver_Path> -R <Rep_Path>
```

If still contain issue, you can check Test Cases Excel for reference.

Or contact my email: ropeaudrey@gmail.com

