

Codio Activity - Buffer Overflow

Part I

In this example, you will compile and run a program in C using the Codio workspace provided (Buffer Overflow in C). The program is already provided as `bufoverflow.c` - a simple program that creates a buffer and then asks you for a name, and prints it back out to the screen.

Run the code a second time (from the command window this can be achieved by entering `./bufoverflow` on the command line). This time, enter a string of 10 or more characters.

- What happens?
- What does the output message mean?

The following code was contained in `bufoverflow.c`:

```
# include <stdio.h>
int main (int argc, char **argv)
{
char buf[8];
printf("Enter name: ")
gets(buf);
printf("%s\n", buf);
return 0;
}
```

When performing the above, the following code execution was observed in *Figure 1*:

```
codio@wheelplastic-membermiami:~/workspace$ gcc bufoverflow.c -o bufoverflow && ./bufoverflow
bufoverflow.c: In function 'main':
bufoverflow.c:8:5: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
    gets(buf);           // read from stdio (sensitive function!)
    ^~~~~
    fgets
/tmp/ccVPJb0d.o: In function `main':
bufoverflow.c:(.text+0x3c): warning: the `gets' function is dangerous and should not be used.
Enter name: leesaxto
leesaxto
codio@wheelplastic-membermiami:~/workspace$ gcc bufoverflow.c -o bufoverflow && ./bufoverflow
bufoverflow.c: In function 'main':
bufoverflow.c:8:5: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
    gets(buf);           // read from stdio (sensitive function!)
    ^~~~~
    fgets
/tmp/ccGL43j9.o: In function `main':
bufoverflow.c:(.text+0x3c): warning: the `gets' function is dangerous and should not be used.
Enter name: leesaxton
leesaxton
*** stack smashing detected ***: <unknown> terminated
Aborted (core dumped)
codio@wheelplastic-membermiami:~/workspace$
```

Figure 1: Buffer Overflow in C

In the first instance, the name entered is within the buffer range so the input is reflected back as output.

In the second instance, the name entered exceeds the buffer range and thus results in the following message:

```
*** stack smashing detected ***; <unknown> terminated
Aborted (core dumped)
```

The message is comprised of two main components:

1. "stack smashing detected" (Narang, 2022)
 - defense mechanism to prevent a possible buffer overflow
 - uses a sequence of bits to check for a buffer overflow
 - is a runtime error and results in program termination (<unknown> terminated)
2. Aborted (core dumped) (knobs-dials, n.d.)
 - intentional program termination during debugging
 - performed to better figure out what went wrong
 - prevents further data corruption through abrupt termination

Part II

Now carry out a comparison of this code with one in Python (Buffer Overflow in Python), following these instructions:

In the Codio workspace, you will be using the file called Overflow.py:

Run your code using: Python overflow.py (or use the codio rocket icon)

What is the result?

Read about Pylint at <http://pylint.pycqa.org/en/latest/tutorial.html>

Install pylint using the following commands:

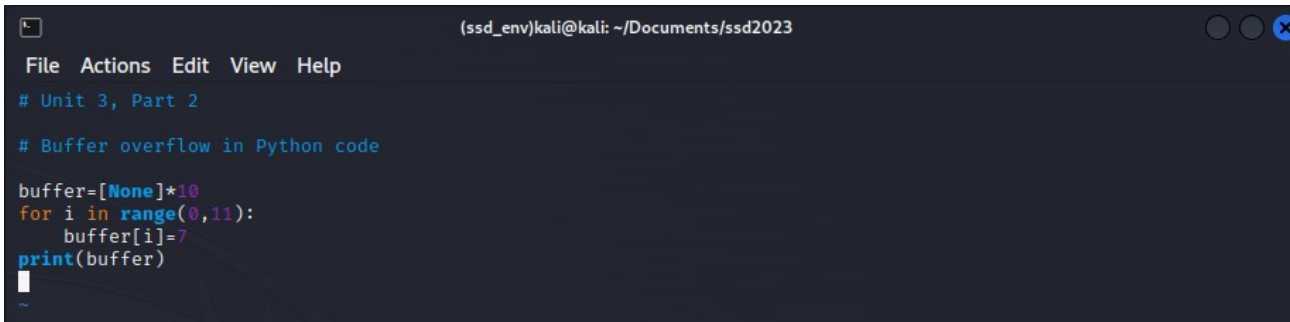
```
pip install pylint (in the command shell/ interpreter)
```

Run pylint on one of your files and evaluate the output:

- Pylint your_file
- (Make sure you are in the directory where your file is located before running Pylint)

What is the result? Does this tell you how to fix the error above?

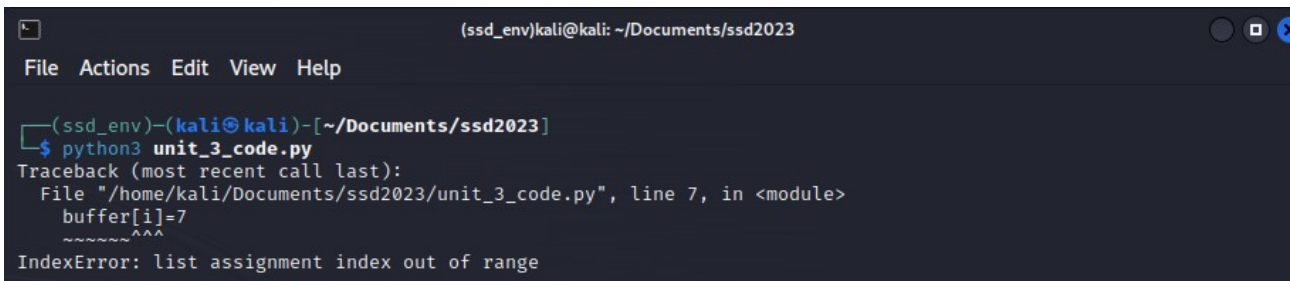
Part 2 was performed on the command line of a Kali Linux VM. The code was written on VIM and can be reviewed in *Figure 2*.



```
(ssd_env)kali@kali: ~/Documents/ssd2023
File Actions Edit View Help
# Unit 3, Part 2
# Buffer overflow in Python code
buffer=[None]*10
for i in range(0,11):
    buffer[i]=7
print(buffer)
```

Figure 2: Python Script for Buffer Overflow

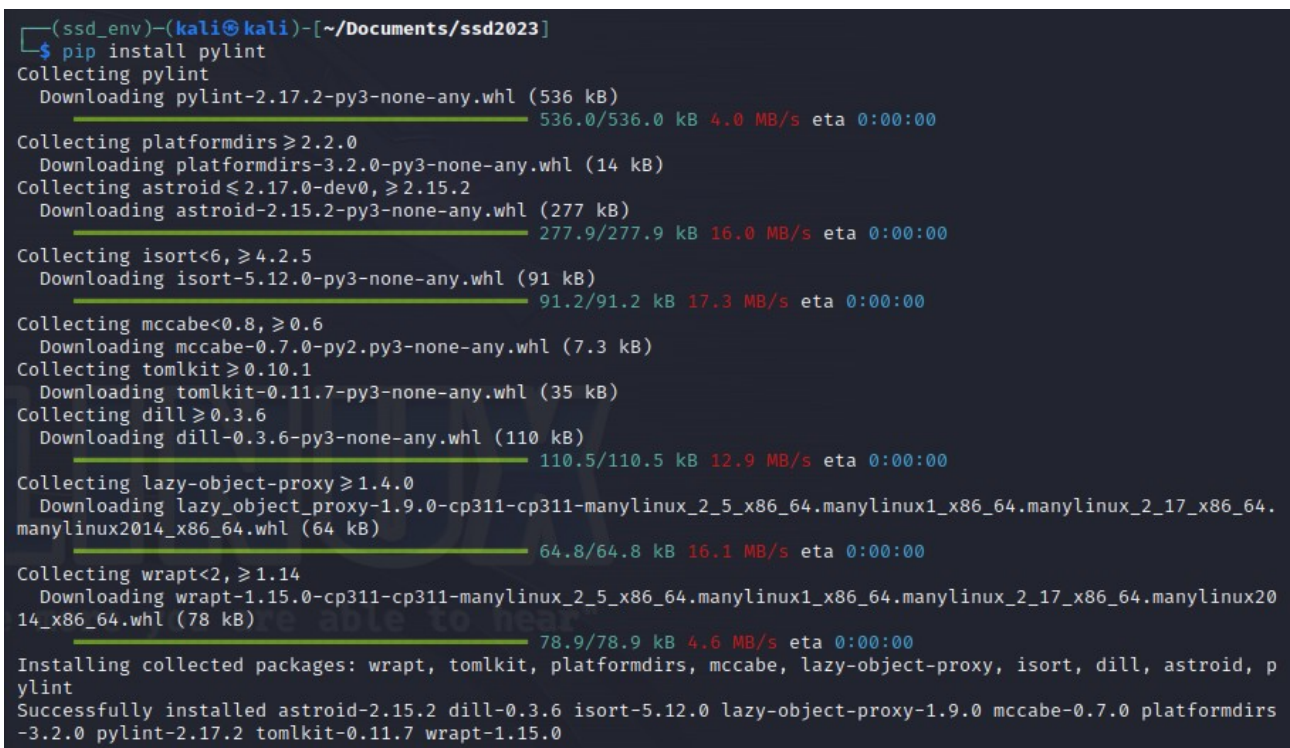
When performing the above, the following code execution was observed:



```
(ssd_env)kali@kali: ~/Documents/ssd2023
File Actions Edit View Help
(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ python3 unit_3_code.py
Traceback (most recent call last):
  File "/home/kali/Documents/ssd2023/unit_3_code.py", line 7, in <module>
    buffer[i]=7
    ~~~~~^
IndexError: list assignment index out of range
```

Figure 3: Python Code Execution

Pylint was then downloaded to parse the code for possible errors (Figure 3), the results of which can be seen in *Figure 4*.



```
(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ pip install pylint
Collecting pylint
  Downloading pylint-2.17.2-py3-none-any.whl (536 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 536.0/536.0 kB 4.0 MB/s eta 0:00:00
Collecting platformdirs ≥ 2.2.0
  Downloading platformdirs-3.2.0-py3-none-any.whl (14 kB)
Collecting astroid ≤ 2.17.0-dev0, ≥ 2.15.2
  Downloading astroid-2.15.2-py3-none-any.whl (277 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 277.9/277.9 kB 16.0 MB/s eta 0:00:00
Collecting isort < 6, ≥ 4.2.5
  Downloading isort-5.12.0-py3-none-any.whl (91 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 91.2/91.2 kB 17.3 MB/s eta 0:00:00
Collecting mccabe < 0.8, ≥ 0.6
  Downloading mccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
Collecting tomlkit ≥ 0.10.1
  Downloading tomlkit-0.11.7-py3-none-any.whl (35 kB)
Collecting dill ≥ 0.3.6
  Downloading dill-0.3.6-py3-none-any.whl (110 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 110.5/110.5 kB 12.9 MB/s eta 0:00:00
Collecting lazy-object-proxy ≥ 1.4.0
  Downloading lazy_object_proxy-1.9.0-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl (64 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 64.8/64.8 kB 16.1 MB/s eta 0:00:00
Collecting wrapt < 2, ≥ 1.14
  Downloading wrapt-1.15.0-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl (78 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.9/78.9 kB 4.6 MB/s eta 0:00:00
Installing collected packages: wrapt, tomlkit, platformdirs, mccabe, lazy-object-proxy, isort, dill, astroid, pylint
Successfully installed astroid-2.15.2 dill-0.3.6 isort-5.12.0 lazy-object-proxy-1.9.0 mccabe-0.7.0 platformdirs-3.2.0 pylint-2.17.2 tomlkit-0.11.7 wrapt-1.15.0
```

Figure 4: Download of Pylint

```
(ssd_env)-(kali@kali)-[~/Documents/ssd2023]
└─$ pylint unit_3_code.py
***** Module unit_3_code
unit_3_code.py:1:0: C0114: Missing module docstring (missing-module-docstring)

-----
Your code has been rated at 7.50/10
```

Figure 5: Pylint Results

Conclusions about the code output and errors are as follows:

1. 'Index Error: list assignment index out of range' (Gallagher, 2020)
 1. occurs when trying to access a variable outside the range of a list
2. 'Missing module docstring (missing-module-docstring) (Alonso, 2022)
 1. occurs when a description of the module doc-string is missing
 1. example: `'''Enter doc-string here'''`

The errors shown above would result in a code error and thus leave the program vulnerable to a buffer overflow attack. As the index has not been set properly and module doc-string is absent, the module would not run upon execution and this piece of security would not be implemented.