

Codio Activity - Exploring Python tools and features.

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.

This is the code in `bufoverflow.c` (also available in the Codio workspace):

```
#include <stdio.h>
int main(int argc, char **argv)
{
    char buf[8]; // buffer for eight characters
    printf("enter name:");
    gets(buf); // read from stdio (sensitive function!)
    printf("%s\n", buf); // print out data stored in buf
    return 0; // 0 as return value
}
```

Now compile and run the code. To test it, enter your first name (or at least the first 8 characters of it) you should get the output which is just your name repeated back to you.

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?

```

*
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: food-harmony.codio.io
*
Last login: Tue Aug 24 18:48:09 2021 from 192.168.11.51
codio@food-harmony:~/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-f
unction-declaration]
    gets(buf);           // read from stdio (sensitive function!)
    ^~~~~
    fgets
/tmp/cchYeezv.o: In function `main':
bufoverflow.c:(.text+0x3c): warning: the `gets' function is dangerous and should not be used.
Enter name: kikelomo
kikelomo
codio@food-harmony:~/workspace$ ./bufoverflow
Enter name: Kikelomo Obayemi
Kikelomo Obayemi
*** stack smashing detected ***: <unknown> terminated
Aborted (core dumped)
codio@food-harmony:~/workspace$ ./bufoverflow
Enter name: everything
everything
*** stack smashing detected ***: <unknown> terminated
Aborted (core dumped)
codio@food-harmony:~/workspace$

```

I entered a string over 10 characters twice and each time I got the error message highlighted in red in the screenshot above.

- What does the output message mean?

In C programming language, local variables and information pertaining to the control flow of a variable are stored in memory chunks known as stacks (O'REILLY, 2021). If more values are pushed onto an array on a stack than the available space, an attacker could take advantage of the situation to override the control flow of the variable also stored on the stack. This is known as stack smashing (O'REILLY, 2021).

Stack smashing also known as stack buffer overflows can be defined as a type of attack that occurs when bug in codes are exploited to enable buffer overflow (Arora, 2013). Technologies that have been developed to protect programs from this type of attacks include: those implemented in the compiler (IBM's pro police and stackguard's version of GCC) and some dynamic runtime ones such as Libsafe (O'REILLY, 2021.)

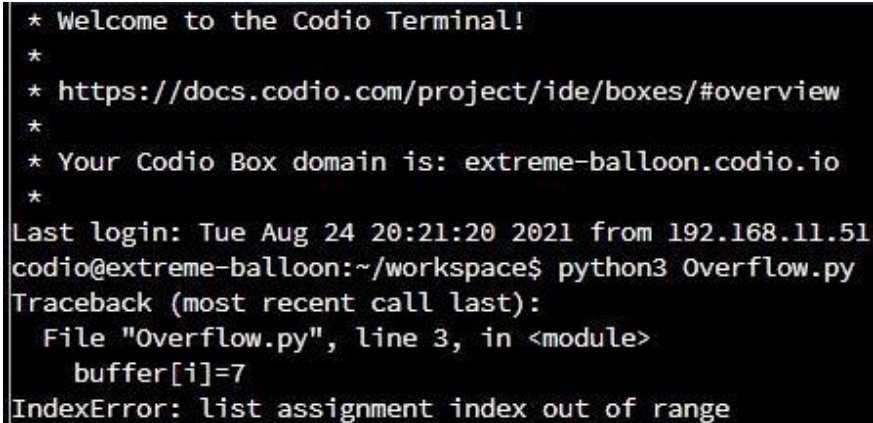
Part II

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:

```
buffer=[None]*10
for i in range (0,11):
    buffer[i]=7
print(buffer)
```

- Run your code using: Python overflow.py (or use the codio rocket icon)
- What is the result?

A screenshot of a terminal window with a black background and white text. The text shows a welcome message, a URL, and a login timestamp. Then, the command 'python3 Overflow.py' is executed, resulting in a 'Traceback (most recent call last):' error. The error message is 'File "Overflow.py", line 3, in <module> buffer[i]=7' followed by 'IndexError: list assignment index out of range'.

```
* Welcome to the Codio Terminal!
*
* https://docs.codio.com/project/ide/boxes/#overview
*
* Your Codio Box domain is: extreme-balloon.codio.io
*
Last login: Tue Aug 24 20:21:20 2021 from 192.168.11.51
codio@extreme-balloon:~/workspace$ python3 Overflow.py
Traceback (most recent call last):
  File "Overflow.py", line 3, in <module>
    buffer[i]=7
IndexError: list assignment index out of range
```

Screenshot show a buffer overflow error.

- 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:
- (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?

```

codio@extreme-balloon:~/workspace$ pylint Overflow.py
No config file found, using default configuration
***** Module Overflow
C: 1, 0: Exactly one space required around assignment
buffer=[None]*10
      ^ (bad-whitespace)
C: 2, 0: Exactly one space required after comma
for i in range(0,14):
      ^ (bad-whitespace)
C: 3, 0: Exactly one space required around assignment
    buffer[i]=7
          ^ (bad-whitespace)
C: 4, 0: Trailing whitespace (trailing-whitespace)
C: 5, 0: Final newline missing (missing-final-newline)
C: 5, 0: Unnecessary parens after 'print' keyword (superfluous-parens)
W: 1, 0: Redefining built-in 'buffer' (redefined-builtin)
C: 1, 0: Module name "Overflow" doesn't conform to snake_case naming styl
e (invalid-name)
C: 1, 0: Missing module docstring (missing-docstring)
C: 1, 0: Constant name "buffer" doesn't conform to UPPER_CASE naming styl
e (invalid-name)

-----
Your code has been rated at -15.00/10

```

Poor code ratings of -15/10 which shows that PEP 8 code standards have not been applied. Below is a screenshot after rewriting the code to conform with PEP 8

Edited Code:

```

"""

```

An example to show buffer overflow in python

```

"""

```

```

from __future__ import print_function //this would not be necessary in pylint version 3

```

```

BUFFER = [None]*10 // spacing around variable assignment and static variable name changed to
uppercase

```

```

for i in range(0, 10): //space inserted after comma

```

```

    BUFFER[i] = 7

```

```

print(BUFFER)

```

```

//newline entered to terminate program

```

```
-----
Your code has been rated at 5.00/10 (previous run: 5.00/10, +0.00)

codio@extreme-balloon:~/workspace$ pylint overflow.py
No config file found, using default configuration
***** Module overflow
C:  9, 0: Final newline missing (missing-final-newline)

-----

Your code has been rated at 8.00/10 (previous run: 5.00/10, +3.00)

codio@extreme-balloon:~/workspace$ pylint overflow.py
No config file found, using default configuration

-----

Your code has been rated at 10.00/10 (previous run: 8.00/10, +2.00)

codio@extreme-balloon:~/workspace$ █
```

NOTE: This does not solve the buffer overflow error shown in Fig 2.

The array has been declared as a fixed size of 10 elements but the code shows that we are trying to access the 11th element of an array which doesn't exist.

References

Arora H. (2013) How to Avoid Smashing Attacks with GCC. Available from:

<https://www.thegeekstuff.com/2013/02/stack-smashing-attacks-gcc/> [Accessed 24 August 2021]

O'REILLY (2021) Prevent Stack Smashing Attacks. Available from:

<https://www.oreilly.com/library/view/network-security-hacks/0596006438/ch01s13.html> [Accessed 24 August 2021]