Activity - Exploring Python tools and features

Part 1

Upon compiling the code (*`gcc bufoverflow.c -o executable`*), I receive the below warning, seen in Figure 1.

```
(base) andaziemele@Andas-MacBook-Pro-2 buffer-overflow-in-c % gcc bufoverflow.c
-o executable
bufoverflow.c:8:5: warning: 'gets' is deprecated: This function is provided for compatibilit
y reasons only.  Due to security concerns inherent in the design of gets(3), it is highly re
commended that you use fgets(3) instead. [-Wdeprecated-declarations]
    8 |     gets(buf);              // read from stdio (sensitive function!)
      |     ^
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/_stdio.h:257:1: note: 'gets'
 has been explicitly marked deprecated here
  257 | __deprecated_msg("This function is provided for compatibility reasons only.  Due to
security concerns inherent in the design of gets(3), it is highly recommended that you use f
gets(3) instead.")
      | ^
/Library/Developer/CommandLineTools/SDKs/MacOSX.sdk/usr/include/sys/cdefs.h:218:48: note: ex
panded from macro '__deprecated_msg'
  218 |         #define __deprecated_msg(_msg) __attribute__((__deprecated__(_msg)))
      |                                                       ^
1 warning generated.
```

Figure 1. Screenshot of terminal after code compilation.

I receive a shorter warning implying unsafeness of `gets()` upon executing it. When I enter my name, it repeats the name back to me and exits (Figure 2). When entering a string longer than 10 characters, it still repeats the character string, however, then also prints the string `zsh: abort  ./executable` (Figure 2).

```
(base) andaziemele@Andas-MacBook-Pro-2 buffer-overflow-in-c % ./executable
warning: this program uses gets(), which is unsafe.
Enter name: Anda
Anda
(base) andaziemele@Andas-MacBook-Pro-2 buffer-overflow-in-c % ./executable
warning: this program uses gets(), which is unsafe.
Enter name: jdjdkjdkjskmaka;a,
jdjdkjdkjskmaka;a,
zsh: abort      ./executable
```

Figure 2. Screenshot of terminal after executing code with expected and unexpected inputs.

The programme is technically meant to read and store up to 8 characters in memory, however the `gets()` function takes a standard input and reads it until a newline character or end-of-file is present (GeeksForGeeks, 2024). The function does not account for bounds, hence why will take and read an input beyond what has been allocated in memory, causing a buffer overflow. A buffer overflow happens when a programme attempts to push more data into the allocated memory buffer than it can handle, causing the programme to crash or corrupt data, or even execute malicious code (OWASP, n.d.).

Part 2

Upon running Overflow.py, the script fails with an IndexError (Figure 3).

```
(base) andaziemele@Andas-MacBook-Pro-2 buffer-overflow-in-python % python Overflow.py
Traceback (most recent call last):
  File "Overflow.py", line 3, in <module>
    buffer[i]=7
IndexError: list assignment index out of range
```

Figure 3. Screenshot of terminal after running Python script.

When parsing the programme code with Pylint, it highlights issues with formatting and documentation of the code (Figure 4).

```
(base) andaziemele@Andas-MacBook-Pro-2 buffer-overflow-in-python % pylint Overflow.py
************* Module Overflow
Overflow.py:4:0: C0303: Trailing whitespace (trailing-whitespace)
Overflow.py:5:0: C0304: Final newline missing (missing-final-newline)
Overflow.py:1:0: C0103: Module name "Overflow" doesn't conform to snake_case naming style (invali
d-name)
Overflow.py:1:0: C0114: Missing module docstring (missing-module-docstring)

------------------------------------
Your code has been rated at 0.00/10
```

Figure 4. Screenshot of terminal after running Python script through linter.

With respect to the IndexError, Pylint does not specify how to fix it. Only upon running the code, the IndexError occurs, and Pylint is a static code analyser, meaning that it will not run the programme to analyse it (Pylint, 2024).

References:

GeeksForGeeks (2024) fgets() and gets() in C language. *GeeksForGeeks*. Available at: https://www.geeksforgeeks.org/fgets-gets-c-language/ [Accessed 10 November 2024]

OWASP (n.d.) Buffer Overflow. *OWASP*. Available at: https://owasp.org/www-community/vulnerabilities/Buffer_Overflow [Accessed 10 November 2024]

Pylint (2024) pylint 3.3.1. *PyPi*. Available at: https://pypi.org/project/pylint/ [Accessed 10 November 2024]