CSE 406 (July 2022)
Buffer Overflow Online (A - 2)

You are given the following vulnerable C program ***A2.c***. Replace <param_1> and <param_2> in the source code with the corresponding values of Table-1.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

/* Changing this size will change the layout of the stack.
 * Instructors can change this value each year, so students
 * won't be able to use the solutions from the past.
 */
#ifndef BUF_SIZE
#define BUF_SIZE1 <param_1>
#define BUF_SIZE2 <param_2>
#endif

void dummy_function(char *str);
int gl = 0;
char* str_1, *str_2;
FILE *badfile, *badfile_2;

int bof_1(char *str)
{
    char buffer[BUF_SIZE1];
    printf("In bof_1\n");
    strcpy(buffer, str);
    return 0;
}

char* foo(int a, int b){
    printf("In Foo\n");
    gl = 1;
    if(a == 5 && b == 7){
     fread(str_2, sizeof(char), 500, badfile_2);
    }
    return str_2;
}

int bof_2(char * str){
    if (gl != 1){
     printf("Not Allowed\n");
     return 1;
```

```
    }
    char buffer[BUF_SIZE2];
    printf("In bof_2\n");
    strcpy(buffer, str);
    return 0;
}

int main(int argc, char **argv)
{
    str_1 = (char*)malloc(500 * sizeof(char));
    str_2 = (char*)malloc(500 * sizeof(char));

    badfile = fopen("badfile", "r");
    badfile_2 = fopen("badfile_2", "r");
    if (!badfile || !badfile_2) {
       perror("Opening badfile"); exit(1);
    }
    fread(str_1, sizeof(char), 500, badfile);
    bof_1(str_1);
    fprintf(stdout, "==== Returned Properly ====\n");
    return 1;
}
```

Tasks:

1. First, compile the program with the 32 bit flag set as demonstrated in the class. Do not forget to turn off address space randomization and stack protection. Also, make sure that the stack is executable while compiling the program.
2. Prepare a payload (e.g. badfile) which will cause the program to open a shell with root's privilege after calling the secret function ***foo***. Please note that there are two badfiles. The machine code to open the shell is to be stored in ***badfile_2***.
3. Put the exploit file(s) in a directory and rename the directory with your 7 digit student ID. Zip the directory and submit on moodle.

   Final output would look like this:

```
[06/12/22]seed@VM:~/.../code_bufrange (copy)$ python3 exploit.py
[06/12/22]seed@VM:~/.../code_bufrange (copy)$ python3 exploit_2.py
[06/12/22]seed@VM:~/.../code_bufrange (copy)$ ./A2
In bof_1
In Foo
In bof_2
# exit
[06/12/22]seed@VM:~/.../code_bufrange (copy)$
```

Marks Distribution

Item	Marks
Opening shell from bof_1	3
Calling foo with proper parameters to satisfy the ***if*** condition	4
Opening the shell from bof_2	8
Viva	5
Total	**20**

Table 1

Student ID	Param_1	Param_2
1605021	105	76
1605036	101	60
1605051	88	58
1605085	147	49
1705031	124	51
1705032	119	42
1705033	132	59
1705034	132	66
1705035	92	75
1705036	138	32
1705037	100	47
1705038	93	39
1705039	103	52
1705040	96	59
1705041	95	51
1705042	100	47
1705043	115	73
1705044	144	70

Student ID	Param_1	Param_2
1705045	107	56
1705046	97	42
1705047	113	36
1705048	116	68
1705049	152	31
1705050	142	31
1705051	106	38
1705052	111	42
1705053	128	45
1705054	145	56
1705055	139	43
1705056	99	71
1705057	93	46
1705058	114	65
1705059	120	34
1705060	143	62