CSE 406 (July 2022)
Buffer Overflow Online (B - 1)

You are given the following vulnerable C program *B1.c*. Replace <param_1> , <param_2> and <param_3> in the source code with the corresponding values of Table-1.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
#define BUF_SIZE <param_1>

int check;
int last_val;

int foo(int a, int b, int session){
    if( session != check){
     printf("Session Terminated. Try Again!\n");
     return 0;
    }

    if((a+b) == <param_2> && b!=last_val){
     printf("Verified session = %d\n", session);
     return 1;
    }

    printf("Unauthorized call to foo\n");
    return 0;
}

void basic_func(char* str, int a ,int b, int session){
    char buffer1[BUF_SIZE];

    printf("Requested session = %d\n", session);
    strcpy(buffer1, str);
    if( !foo(a, b, session) ) exit(1);
}

int main(int argc, char **argv){
    char str[517];
    FILE *badfile;

    srand(time(NULL));
    check = rand();
    last_val = rand() % <param_3>;
```

```
    badfile = fopen("badfile", "r");
    if (!badfile) {
       perror("Opening badfile"); exit(1);
    }

    int length = fread(str, sizeof(char), 517, badfile);
    printf("Input size: %d\n", length);
    basic_func(str, 12345, last_val, check);
    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 and print exactly the lines shown in the figure below

```
[06/19/22]seed@VM:~/.../Online 2$ python3 exploit.py
[06/19/22]seed@VM:~/.../Online 2$ ./stack
Input size: 517
Requested session = 1841171308
Verified session = 1841171308
# whoami
root
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=100
o),30(dip),46(plugdev),120(lpadmin),131(lxd),132(samb
#
```

.

3. Rename your ***exploit.py*** file with ***1705XXX.py*** and submit on moodle.

Marks Distribution

Item	Marks
Calling the function ***foo***	3
Returning 1 from ***foo***	7
Opening the shell	5
Viva	5
Total	**20**

Table 1

Student ID	param_1	param_2	param_3
1605021	284	21	21
1605036	258	16	36
1605051	281	17	51
1605085	203	25	85
1705061	316	14	61
1705062	271	16	62
1705063	303	21	63
1705064	264	13	64
1705065	289	14	65
1705066	233	17	66
1705067	222	16	67
1705068	237	25	68
1705069	233	24	69
1705070	201	20	70
1705071	220	21	71
1705072	287	20	72
1705073	268	17	73
1705074	275	25	74
1705075	310	11	75

Student ID	param_1	param_2	param_3
1705076	257	13	76
1705077	241	22	77
1705078	259	20	78
1705079	318	24	79
1705080	284	12	80
1705081	290	24	81
1705082	252	15	82
1705083	264	18	83
1705084	228	22	84
1705085	209	19	85
1705086	293	20	86
1705087	225	11	87
1705088	310	15	88
1705089	267	16	89
1705090	204	23	90