CSE 406
Online -1 (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.

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

int bof(char *str)
{
   int arr[10];
   arr[5] = 7;
   {
            char buffer[<Param_1>];
            /* The following statement has a buffer overflow problem */
            strcpy(buffer, str);
            printf("%s",buffer);
   }
   return 1;
}

int foo(char *str)
{
   int arr[<Param_2>];
   arr[120] = 23;
   bof(str);
   return 1;
}

int secret()
{
   printf("Inside a Secret function\n");
}

int main(int argc, char **argv)
{
   char str[<Param_3>];
   FILE *badfile;

   bof("Normal Execution\n");
   badfile = fopen("badfile", "r");
```

```
    fread(str, sizeof(char), <Param_3>, badfile);
    foo(str);

    printf("Try Again\n");
    return 1;
}
```

Tasks:

1. First, compile the program from the root's privilege and set its UID as shown in the lab. 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 call the ***"secret"*** function and then open a shell with root's privilege when executed by other users (seed).
3. Rewrite the ***"bof"*** function to defend against buffer overflow attack.
4. Rename your ***exploit.py*** file with ***16050xx.py.*** Put ***16050xx.py*** and modified ***B1.c*** in a folder, zip it (***16050xx.zip***) and submit it in moodle.

The output would look like this:

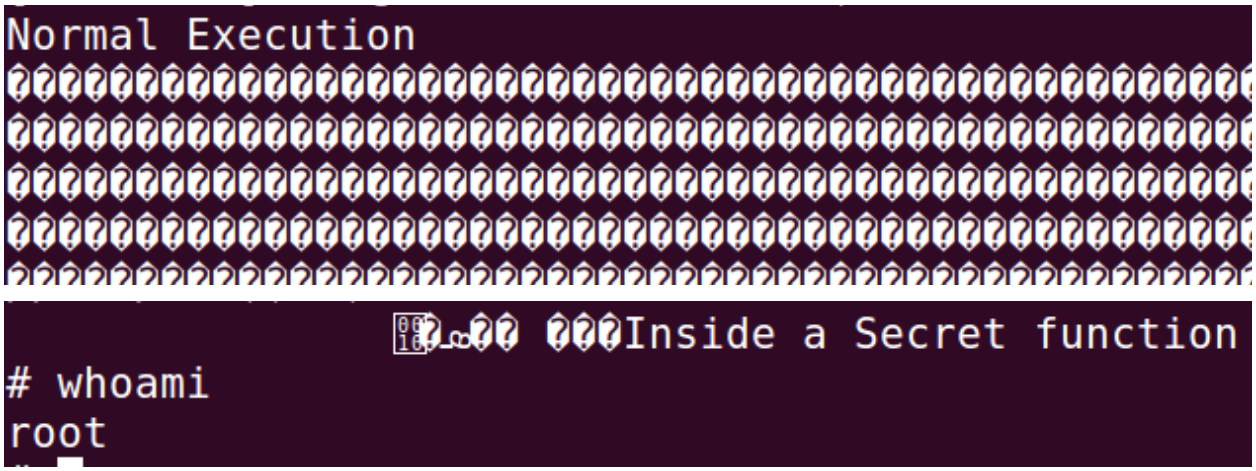

Table-1

Student ID	Param_1	Param_2	Param_3
1605061	614	1633	1014
1605062	1042	1791	1442
1605063	668	1606	1068
1605064	603	1664	1003
1605065	841	1953	1241
1605066	1075	1769	1475
1605067	1074	1839	1474

1605068	916	1645	1316
1605069	665	1689	1065
1605070	985	1960	1385
1605071	627	1797	1027
1605072	659	1815	1059
1605073	721	1898	1121
1605074	1071	1780	1471
1605075	902	1831	1302
1605076	663	1941	1063
1605077	908	1683	1308
1605078	645	1720	1045
1605079	1047	1793	1447
1605080	655	1884	1055
1605081	929	1931	1329
1605082	709	1882	1109
1605083	717	1701	1117
1605084	864	1890	1264
1605085	960	1739	1360
1605086	693	1825	1093
1605087	632	1726	1032
1605088	1096	1647	1496
1605089	1021	1868	1421
1605090	727	1754	1127
0905081	1070	1837	1470
1405059	731	1910	1131
1405081	697	1751	1097
1605012	945	1980	1345
1605013	632	1986	1032
1605019	619	1682	1019
1605027	797	1770	1197