National Taiwan Normal University
CSIE Information Security: A Hands-on Approach

Instructor: Po-Wen Chi
Due Date: Dec 20, 2021, AM 11:59

Assignment 5

Policies:

- Zero tolerance for late submission.
- Please pack all your submissions in one zip file. **RAR is not allowed!!**
- I only accept **PDF**. MS Word is not allowed.
- Hand-writing is not allowed.

5.1 SEED Lab (40 pts)

Format-String Vulnerability Lab
https://seedsecuritylabs.org/Labs_20.04/Software/Format_String/

Please record all your steps with screen captures and answer all questions. Undoubtedly, you need to complete all tasks even they are marked with **optional**.

5.2 Data Modification (20 pts)

In this class, I have shown you how to use the format string vulnerability to modify the memory data. We use the conversion specifier **n**, which is the number of characters written so far is stored into the integer pointed to by the corresponding argument, to change the data value. Undoubtedly, the number of characters written grows and cannot be less than zero. Is it possible to write a negative value to a signed integer variable? Please write down your answer and give an example to support your answer.

5.3 _FORTIFY_SOURCE (20 pts)

There is a GCC option called **_FORTIFY_SOURCE**. What is this option? Can this option stop the format string attack? Why? Please give an example to support your description.

5.4 sprintf (20 pts)

Please read the following code.

```
#include <stdio.h>

void fmtstr( char *str )
{
    unsigned int *framep;
    unsigned int *ret;

    // Copy ebp into framep
    asm("movl %%ebp, %0" : "=r" (framep));
    ret = framep + 1;

    /* print out information for experiment purpose */
    printf( "The address of the input array:  0x%.8x\n", (unsigned ) str );
    printf( "The value of the frame pointer:  0x%.8x\n", (unsigned ) framep );
    printf( "The value of the return address: 0x%.8x\n", *ret );

  char buf[BUFSIZE] = {0};
    sprintf( buf, str ); // The vulnerable place

    printf( "\nThe value of the return address: 0x%.8x\n", *ret );
}

int main( int argc, char **argv )
{
    FILE *badfile;
    char str[300];

    badfile = fopen( "badfile", "rb" );
    fread( str, sizeof( char ), 200, badfile );
    fmtstr( str );

    return 1;
}
```

The difference is that in this problem, we use **sprintf** instead of **printf**. The build command should be

```
$ gcc -D BUFSIZE=5000 -m32 fmtvul.c
```

where BUFSIZE is from 5000 to 6000. Please hack this program. Again, you need to describe each step in **detail**!!

5.5 Bonus: Cyberbit (5 pts)

How about the Cyberbit class? Please write down your comment for me to plan the future class.