CS6570 : Secure Systems Engineering
Lab 2 : Return Oriented Programming

10th February 2023

In this lab, you will utilize Return-oriented programming tools to find and chain gadgets to execute different tasks using the same executable.

Resources:

Before you start, install ROPGagdet: https://github.com/JonathanSalwan/ROPgadget

```
$ pip install ropgadget
$ ROPGadget --binary lab_2_rop
```

All the teams are given the same statically linked executable, which you will be exploiting and constructing ROP gadgets.

Problem 1: 30 points

In this problem, you will chain gadgets found in *lab_2_rop* to compute the 6!. You need not implement loops/jumps to compute 6!. You should chain your gadgets such that step-by-step multiplication starting from 1 to 6 should be executed by the gadget. The final value (6!) should be printed on the screen. Following is a sample output:

```
Input 10 words:
Here are the first characters from the 10 words concatenated:
SaiGanesha
Value in glb is 720
Segmentation fault (core dumped)
```

Problem 2 : 70 points

In this problem, you will chain gadgets found in *lab_2_rop* to encrypt a character string using Caesar's cipher. You need to implement loops as part of the ROP logic. The final encrypted cipher text should be printed on the screen.
Following is the Caesar cipher program logic that you are expected to encode via gadgets:

```
int key = <xxx>; // between 1-26
int length_plaintext = <xxx>; // should work for any length
char plaintext[length_plaintext] = "<xxxx>"; // characters: [A-Z]
char ciphertext[length_plaintext];

for(int i=0; i<length_plaintext; i++)
    ciphertext[i] = (plaintext[i] - 'A' + key)%26 + 'A';
printf("%s\n", ciphertext);
```

Following is a sample output for the input string=*"ABCDEFGHIJKLMNOP"*, key=*25*:

```
Input 10 words:
Here are the first characters from the 10 words concatenated:
Sai@CS6570
ZABCDEFGHIJKLMNO
Value in glb is 0
Segmentation fault (core dumped)
```

Submission:

1. This is a team-based assignment.
2. The exploit generation script for the above 2 problems needs to be submitted along with a README file that contains instructions on how to generate the exploit string and pass the input to the executable.
3. No report/document explaining your gadget construction or technique needs to be submitted.