实验 - Linux 实验环境初探

实验简介

ROP -- Return Oriented programming

基础部分

ret2sc 30分

```
// gcc -fno-stack-protector -zexecstack -znow hw1.c -o hw1
// ./hw1

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

void vul()
{
    char buffer[0x40];
    printf("buffer address: %p\n", buffer);
    gets(buffer);
}

int main(void)
{
    vul();

    return 0;
}
```

要求通过ret2sc的攻击手法来获取本地shell

要求报告中给出 exploit 代码并截图，截图的内容为获取到shell后执行任意的shell命令（如pwd，id等）

ret2libc 30分

```
// gcc -no-pie -fno-stack-protector -znow hw2.c -o hw2
// ./hw2

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

char string[] = "/bin/sh";

void prepare()
```

```
{
    puts("Hello!");
    printf("There is a gift: %p, enjoy it!\n", puts);
}

void vul()
{
    char buffer[0x50];
    gets(buffer);
}

int main(void)
{
    prepare();
    vul();

    return 0;
}
```

要求通过ret2libc的攻击手法来获取本地shell

要求报告中给出 exploit 代码并截图，截图的内容为获取到shell后执行任意的shell命令（如pwd，id等）

ret2csu 40

```
// gcc -no-pie -fno-stack-protector -znow hw3.c -o hw3
// ./hw3

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

char string[] = "/bin/sh";

void useless()
{
    /* This won't work, but you can use execve to get shell */
    execve("useless", NULL, NULL);
}

void vul()
{
    char buffer[0x30];
    read(0, buffer, 0x200);
}

int main(void)
{
    vul();

    return 0;
}
```

要求通过ret2scu的攻击手法来获取本地shell，即通过ROP控制三个参数并执行`execve("/bin/sh", NULL, NULL)`来获取shell。

要求报告中给出 exploit 代码并截图，截图的内容为获取到shell后执行任意的shell命令（如pwd，id等）

挑战部分

- 多次ROP完成getshell，拿到shell后获取靶机上的flag
  - 附件以二进制形式提供，为rop1，远程靶机地址：10.12.77.33:20001
- 栈迁移？栈迁移！，拿到shell后获取靶机上的flag
  - 附件以二进制形式提供，为rop2，远程靶机地址：10.12.77.33:20002

拓展问题和阅读

- 安装one_gadget工具

```
sudo apt -y install ruby
sudo gem install one_gadget
```

- CTFwiki：https://ctf-wiki.org/