Name: Student ID:

This is a piece of shellcode. Can you figure out its C equivalence? Write out your answer in the form of several C statements. For example, for the shellcode we learned in class, the answer is:

char \*path = "/bin/sh";

char \*argv[2] = {path, 0 };

execve(path, argv, 0);

*shellcode:*

*pushl $0x31*

*popl %eax*

*cltd or xorl %edx, %edx*

*int $0x80*

*movl %eax, %ebx*

*movl %eax, %ecx*

*pushl $0x46*

*popl %eax*

*int $0x80*

*xorl %eax, %eax*

*pushl %eax*

*pushl $0x68732f6e*

*pushl $0x69622f2f*

*movl %esp, %ebx*

*pushl %eax*

*pushl %ebx*

*movl %esp, %ecx*

*xorl %edx, %edx*

*movb $0xb, %al*

*int $0x80*

Answer:

Note:

1: This is part of the system call table for x86 Linux kernel. eax stores the system call number. Ebx, ecx, edx, esi and edi are parameters.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | %eax | %ebx | %ecx | %edx | %esi | %edi |
| exit | 1 | int |  |  |  |  |
| read | 3 | unsigned int | chart \* | size\_t |  |  |
| execve | 11 | char \* | char \*\* | char \*\* |  |  |
| getpid | 20 |  |  |  |  |  |
| geteuid | 49 |  |  |  |  |  |
| setreuid | 70 | unsigned int | unsigned int |  |  |  |

2: A brief description of each system call.

**void exit(int status);**

The exit() function causes normal process termination and the value of status & 0377 is returned to the parent (see wait(2)).

**ssize\_t read(int fd, void \*buf, size\_t count);**

read() attempts to read up to count bytes from file descriptor fd into the buffer starting at buf.

**int execve(const char \*filename, char \*const argv[], char \*const envp[]);**

execve() executes the program pointed to by filename. argv is an array of argument strings passed to the new program. envp is an array of strings, conventionally of the form key=value, which are passed as environment to the new program. Both argv and envp must be terminated by a null pointer.

**pid\_t getpid(void);**

getpid() returns the process ID of the calling process.

**uid\_t geteuid(void);**

geteuid() returns the effective user ID of the calling process.

**int setreuid(uid\_t ruid, uid\_t euid);**

setreuid() sets real and effective user IDs of the calling process.

3: The ***cltd*** instruction converts the signed long in EAX to a signed double long in EDX:EAX by extending the most-significant bit (sign bit) of EAX into all bits of EDX.

4: ASSIC encoding

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **0x2f** | **0x62** | **0x68** | **0x69** | **0x6e** | **0x73** |
| / | b | h | i | n | s |

|  |
| --- |
|  |