Software Security via Program Analysis

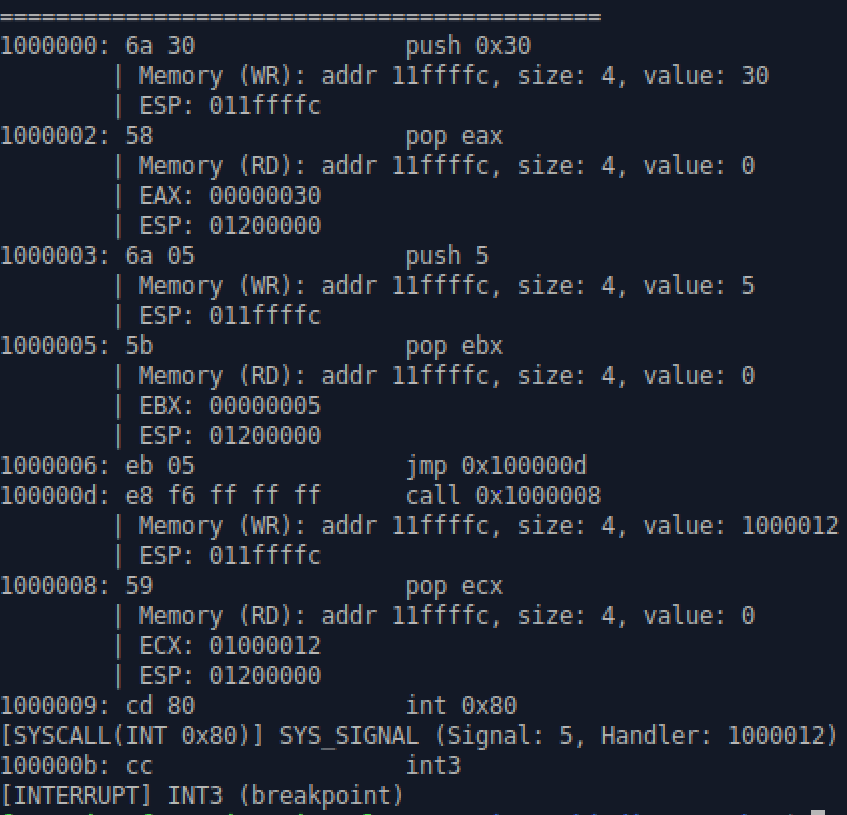
HW4 Running the Hidden Code

Yu-Sheng Wang (bqk3dj)

**Example 1**

b"\x6a\x30\x58\x6a\x05\x5b\xeb\x05\x59\xcd\x80\xcc\x40\xe8\xf6\xff\xff\xff\x99\xb0\x0b\x52\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x52\x53\x54\xeb\xe1"

Stop at xcd\x80



eax is 30. It calls the system call number 30 which is SYS\_SIGNAL() and sets up signal at handler 12.

We can continue the execution as following:  
A screenshot of a computer

Description automatically generated with medium confidence

eax is equal to 1. It calls the system call number 1 which is exit().

Continue the execution:  
Text

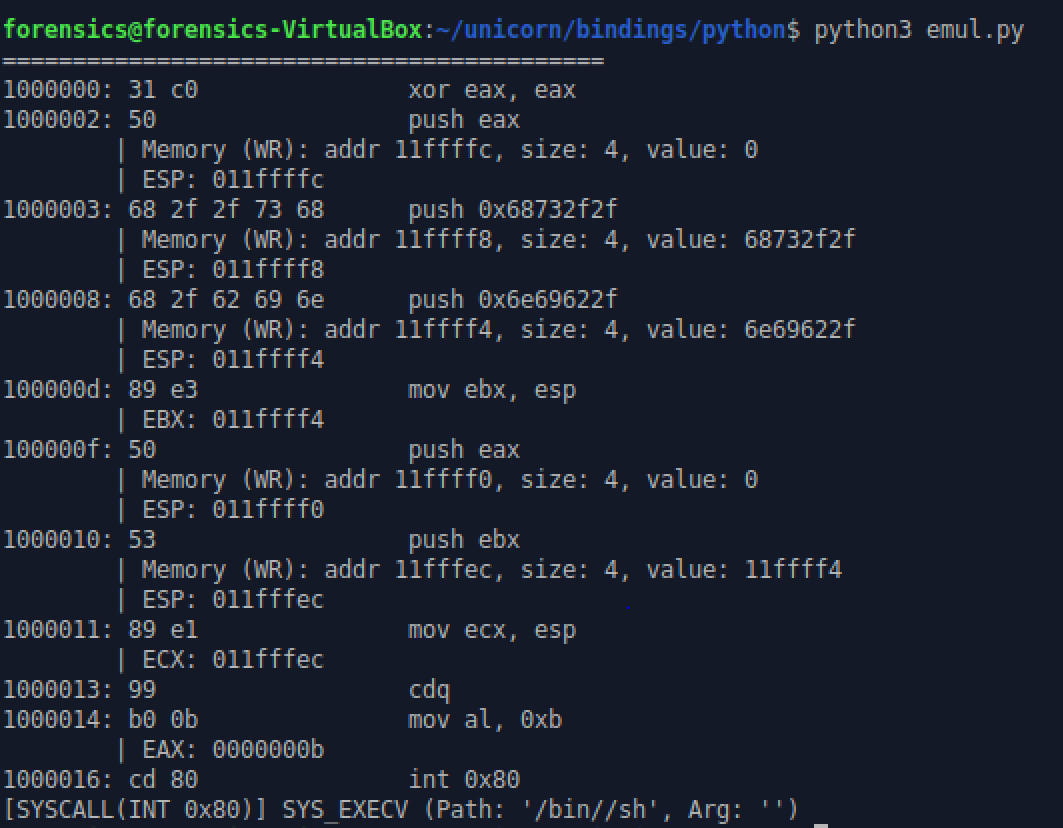
Description automatically generated

Now, we finally cover all the code. Starting from x99 is the malicious code the hacker tries to harm the host machine. At 1000025, it calls an interrupt (SYSCALL(INT 0x80)) and SYS\_EXECV with EAX set as 0b. *Path* is constructed with all those *push* instructions. If we run this on a real machine, it first sets up an exception handler. If exception happens, it gains access to a bin shell.

**Example 2**

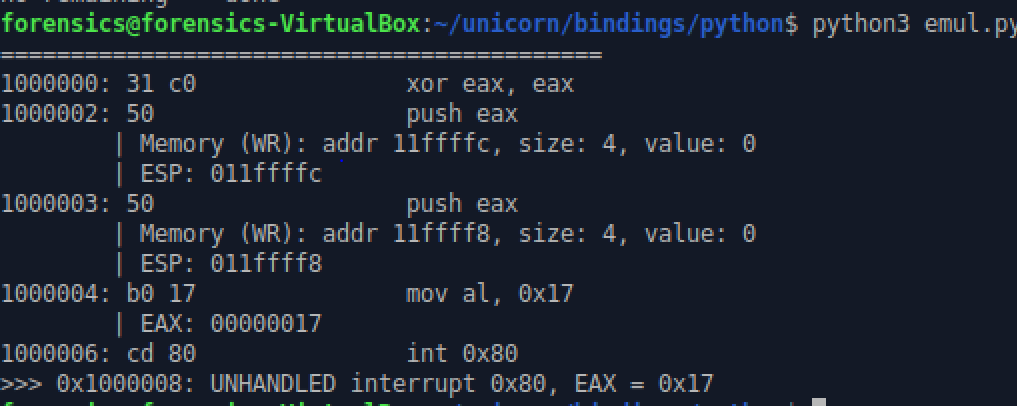
From 1000002, it push value to stack memory at address 11ffffc, which overwrite the value in ESP. The code snippet would launch a bin shell at the end.

At 10000014, the eax value is modified as 0b. Then, the interrupt comes and it calls system call number 11 SYS\_EXECV.



**Example 3**

The original code does not support the system call 0x17 (setuid).



Add function in hook\_interrupt():  
Text

Description automatically generated

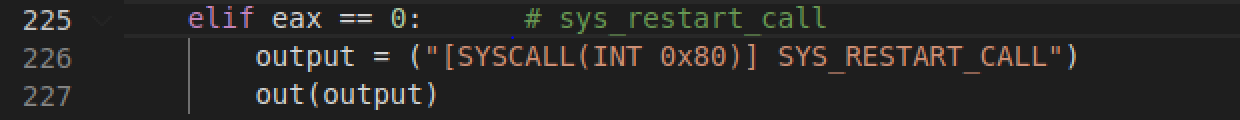
Now the program can run:  
A screenshot of a computer

Description automatically generated with medium confidence

Ai 1000008, it jumps to the function *load\_file* and starts to write return address with multiple push. It’s used to get the current instruction address.

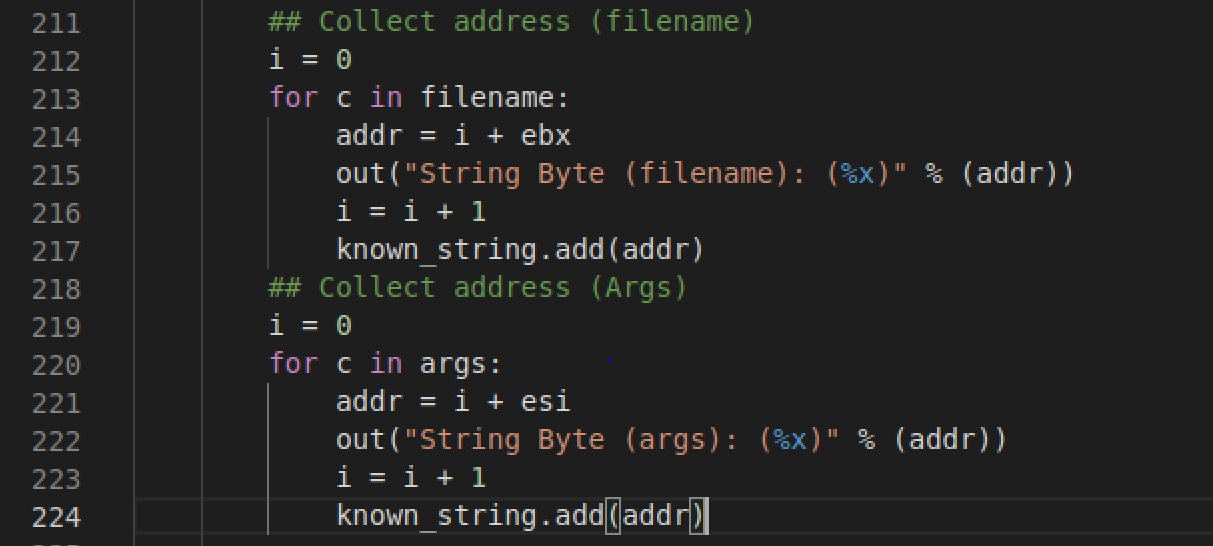
To continue but the program crashed because the lack of support of system call:  
Text

Description automatically generated

Again, to extend the function *hook\_interrupt:*

Something wrong in the next step:  
Graphical user interface, text

Description automatically generated

Modify *emul.py* to output and record the string address:  
Text

Description automatically generated

By recording the addresses of strings, we can skip executing them.

Text

Description automatically generated