Exploit1:

Approach: Buffer overflow (in the copy\_file function)

Vulnerability:

The copy\_file function uses a counter to track the file length and assign buffer value,

But there is no check on the counter to make sure it is less than the

Buffer limit. So it is possible to feed the buffer more than 4000 chars and overwrite the return address to execute the shellcode.

Implementation:

I create a buffer than is 4017 chars in length. And first fill with dot (same as the warm up video). Then I put in the shellcode and the eip location (found using gdb to check the stack frame). Then, put the char array into a txt file and upload. But I noticed that the check virus function will scan the /bin/sh and stop the file being copied by the copy\_file function. Then after I noticed that the check for viruses function use a buffer with length of 1024. I decided to add dot in front of the shellcode to separate bin/sh into two scan loop to avoid this issue. After that, the file will be able to make the buffer in copy\_file function overflow and overwrite the eip and point to the start position of shellcode (also found using gdb) and execute the shellcode when it returns.

Fix:

Limit the copy\_file counter same as (smaller than) the buffer size to avoid overwriting the eip.