Kenneth Woodard

CS 366

Written Assignment 2

1.

A. You can exploit this function using buffer underflow. With a reversed stack, strcpy will write upwards, to “func’s” stack frame. This leads to overwrite of strcpy’s return address. This just means that you fill the buffer with less than 128 bytes. If you choose to fill 64 bytes, the other 64 bytes will be filled with other bytes from memory. These 64 leaked bytes may contain the data we need to attack.

B. Finding the canary value is necessary in exploiting this system with buffer overflow. The canary is always checked before the calling function is returned, so the canary must be inserted back into the stack beforehand. For example, if the canary is a NULL value, you can use a NULL terminated string to write the canary back to the buffer in the proper position (before the return address of the function. If the canary is correct upon the return of the function, then this is when the attacker is able to influence the instruction pointer however they desire to.

2.

A. If the Unix system is older, one vulnerability would be the /etc/passwd file because it is world readable. The main vulnerability to exploit here is path injection. Because the list of permission files is in Alice’s programs, you cannot access and change which files can be altered. But, you could move and rename desired files to the exact location of a file that’s on the list. If the file is a shell_code that has root privileges, you can pretty much get whatever you need/want without changing alice_in or alice_out.

B. If you check the file info using the stat (or whatever system call is necessary) you can access the time the file was last modified. You can just store the times values for each file in file

list. If they don’t line up, then quit alice_in or alice_out because the file isn’t what it is supposed to be. Because you can modify the mod time of files, it isn’t the most fool-proof solution. But it does stop the simple exploitation of just moving and renaming the files to the desired path and name.