Buffer Overflow Exploits allow the value of variables to be changed and for shell code to be executed. This is done by filling a fixed-size buffer on the stack with more than its capacity. For this exercise I changed the value of the integer variable balance and ran the shell code associated with the /bin/cat etc/password command by overflowing a buffer in the vul2 program.

This exploit was done by exploring the stack layout of vul2 and then preparing input for the buffer that would overflow it. For the first step I ran gdb with the vul2 executable, set a breakpoint before the execution of the proc\_input function, and ran print &buf, print &balance, and info registers to find the start of the buffer, the location of the balance variable, the ebp (stack base pointer), and the difference between the three. I discovered that ebp was at 0xffffd458, balance was at 0xffffd44c, and the buffer started at 0xffffd3ac. The difference between the buffer’s start and the ebp was 172, and the difference between the ebp and the balance was 12. Therefore, when creating the program to write the “abc” input file, the location of the ebp (0xffffd458) would be placed at index 172, followed by the buffer’s start (0xffffd3ac) at 176, and preceded by the new value for the balance variable at 160. The first 75 characters to be written were the 75 bytes of the shellcode used to execute the /bin/cat etc/password. Bytes 160-164 contained \x15\xcd\x5b\x07, the hexadecimal representation of the number 123456789. This character string was then written to the file “abc”. This same file was read into the buffer buf by vul2, which was passed to the proc\_input function and copied to a buffer of a smaller capacity, overflowing the buffer. This overwrote the balance variable from 0 to 123456789 and ran the shellcode, executing /bin/cat etc/password. The results were a very large balance and the output of password information.

I compiled the exploit code using gcc -m32 exp2.c -o exp2, and the vul2 program using gcc -m32 -g -fno-stack-protector -z execstack vul2.c -o vul2. I’m running a MacBook Pro, using the Jaguar server via XTerm.