README.DOCX

Deliverable for ZPARNOLD and SYOON

We wrote a function called (get\_prinfo) that would intercept CS3013\_SYSCALL1, and we intercepted this call by becoming the kernel and inserting a new address for our function call located in module2.ko. This module when loaded properly prints out statistics into the kernel log in Kernel-space, and in user space, returns to a struct pointed to in user-space from the kernel-space. Additionally, in our tests we printed out all info from the struct that was being pointed to in user-space.

Testing:

In order to test our program, we needed to see what was being printed out by several calls to the system table at varying levels. (3) For example, 3 levels of children, and 3 siblings of each parent.

In our test.c file, you will notice 3 test functions. (test1function, test2function, and test3function.)

Each of these tests a different aspect of the system call being intercepted. The first simply prints out a process being made. The second prints out three levels of children and a parent. And, the third prints out three levels of children and parent each with 3 siblings totaling 16 total outputs. (Each with a “super parent”)

In order to make this program. You will need the Makefile, getprinfo.c, and prinfo\_LKM.c

To compile the tests, (since this is done separately from the module,) use “gcc test.c” from a command window.