Project 2 Write Up

Group 22

November 10, 2014

Our Plan

To create an encrypted *RAM Disk* we decided to first get a unencrypted version working before we begin to work with the *Linux Crypto API*. After reading over the assignment details and requirements we took a look at the "Linux Device Drivers" example we discovered that finding a working example of a *RAM Disk* would not be too hard to find. After hearing about how undocumented the *Linux Crypto API* is in class we will begin our research early.

Our Solution

We took advantage of reusing the code from the "Linux Device Drivers" book that was recommended for us to get used to programming Linux Drivers. We wanted to get a working *RAM Disk* working before we implemented our *cipher*. We added the sbd.c text that we grabbed from the book and after a few compile warnings we found a blog by "Pat Patterson" that said he had fixed the warnings and errors from the sbd.c file from the "Linux Device Drivers". Once we compiled the kernel with the ramdisk included it was we needed to mount it. Using the command *fdisk -l* we discovered that our new *RAM Disk* was mapped to */dev/sbd0* since the *RAM Disk* is just a unformated block of memory we decided to make a ext2 filesystem and locally mounted it to a folder. After the Disk was mounted we copied a few files to it and made sure it worked OK before we continued.
Finding examples and any documentation for that mater was very difficult for the *Linux Crypto API* but after a bit of searching we eventually came to the conclusion that you need to

1. Allocate crypto API
2. Set cipher key
3. Encrypt/Decrypt one byte at a time
4. Free the cipher

We attempted to use the "Blowfish" cipher algorithm instead of the standard "AES" to try something different from the few examples we came across. It was hard to use the *Linux Crypto API* macros since they were fairly ambiguous. After a while of fiddling we got our Kernel to compile and boot. Once the system

started we formated the drive and mounted it like we did before.
To verify that our solution was correct we took advantage of the printk function and printed the RAW data to the console before and after it was *encrypted*. It was very easy to examine the differences from before and after the *encryption* took place. Even though we were printing unsigned chars directly from the buffer and already couldn't read the data it was clear that something changed when the *encrypted* data was printed.

Work Log

Date	**Author**	**Commit**	**Summary**
Wed Nov 5 19:25	Bob	3f23e27044ff6f1d896075a36974c258c17a84f4	Finnished Makefile and tar.bz2 for Concurrency 3.
Wed Nov 5 19:21	Bob	ce1c354c64c832cd137e17617b258c1c116034e0	Finnised concurrency 3.
Sun Nov 2 12:20	Bob	8feccdab21ba8750a0c5e9efe128aa44aa4fd3c4	Merge branch 'master' of github.com:quinnsam/cs444
Sun Nov 2 12:20	Bob	2cebcc999819ba670771c438eb14f813e99fc989	Added hw3 folder and some documentation.
Mon Nov 10 10:45	Sam Quinn	b907ba3f829fae8178c7a05eb62e925eed699f56	Added a working version of the encrypted Ramdisk.
Mon Nov 10 10:49	Bob	3899e2b81125105ff1e1c6d3c08b6e7e4e8b152f	Added a mounting shell script to mount the encrypted ramdisk in one command.
Mon Nov 10 10:53	Bob	3d5d384d6174a0667bb7c0b07e0bd21702b8151f	Added original unecrypted ramdisk code from LLD3
Mon Nov 10 10:56	Bob	3a9436d55efb23125f914d22181830a5e5f2d85d	Updated the .gitignore file.
Mon Nov 10 11:01	Bob	93a01769fcd7d5cb670419a7c68baa4a91dda6a6	Added the outline for Sam's Writeup.
Mon Nov 10 18:08	Bob	689a1ac1ff20bd4d32c1f318984277c05cb8cf85	Updated Sam's writeup with finished version.
Mon Nov 10 18:09	Bob	e9e015e98d3b8e04c2a8762b4af20029ce72245a	Added a pdf version of Sam;s Writeup.
Mon Nov 10 18:13	Bob	b3e07347b213607769bfcf75739449ecc6a4be62	Added the tar ball for Sam's writeup.
Mon Nov 10 18:27	Bob	e744aa955ac823fb16abccbf23752374e396500f	Added the kernel patch file for the encrypted ram disk.
Mon Nov 10 20:53	Bob	cd1a064e295c87c3cafeee62842ca5b379735bc3	Added the group Writeup tex and makefile
Mon Nov 10 21:12	Bob	af5c9f7eab1da3f867639174c34c4a1b97f2a394	Finnished the group witeup.