Project 3 Write Up

Sam Quinn

November 10, 2014

What do you think the main point of this assignment is?

I think the main point of this assignment was to incorporate existing source code provided by the Linux Kernel in to a new function. There is a lot of reusable code within the Linux kernel and when ever possible you should use that instead of trying to implement it yourself. I think that since there was little documentation on the *Linux Crypto API* it made us have to analyze the source code directly which is not an easy task.

How did you personally approach the problem? Design decisions, algorithm, etc.

I first wanted to implement a *RAM Disk* without encryption and then implement the encryption portion later. After doing some research on *RAM Disks* I found that there is a command that you can create a *RAM Drive* but to have it **Encrypt** and **Decrypt** in the background without any configuring I needed to create a Driver within the Kernel. I read the LDD3 section on creating a Linux *RAM Disk* driver and began to examine the source code provided. I continued my research and found a blog post by "Pat Paterson" which took the same source code provided by LDD3 book and fixed many of the compile errors that would occur with Linux Kernel 2.6 and up. I took his code which looks fairly identical to the one out of the book and implemented it in to our Kernel. Once it booted and I figured out how to mount it we began the wild goose chase of figuring out how to implement the Linux crypto API. We came across a few example which described the process allocate cipher, initialize key, encrypt/decrypt, and then free the cipher. After we got that information we began to implement it into our RAM Disk. While every example we found online about the ciphers used "AES" as their encryption method. Trying to stand out we at first tried to implement the "Blowfish" cipher, which ended up not working. We eventually got the cipher to encrypt and decrypt one byte at a time which requires us to loop through our data to encrypt and decrypt.

How did you ensure your solution was correct? Testing details, for instance.

We implemented a few printk statements that would display the data as unsigned chars before and after the encryption. It was quite easy to see that the data which was already quite foreign because it was all numbers was getting more jumbled after the encryption process. Even when the data being passed through the cipher are all zeros you can see that the output became something else.

What did you learn?

This project interested me a lot. I am currently going in to the security field of computer science and this was the fist time that I have worked with encryption at this low of a level. I think that I might even try to implement the same concept to one of my extraneous drive in my personal computer. I learned how to take advantage of reusable code within the Linux Kernel as well as get first hand experience in the exact field I hope to be apart of after school.