

OS 2

MAY 24, 2018

ASSIGNMENT 3

PREPARED BY

GROUP 41

Abstract

In this assignment we will be creating an encrypted block device. This device will be built separately and be included in the kernel as a kernel module. The encrypted block device will have two modes; read and write. We will develop the module locally and import it into the kernel and mount it on the fly.

CONTENTS

1	Solution Design	2
2	questions	2
2.1	What Do you think the main point of this assignment is?	2
2.2	How did you personally approach the problem? Design decisions, algorithm, etc.	2
2.3	How did you ensure your solution was correct? Testing details, for instance. Ensure this is written in a way that the TA's can follow to ensure correctness.	2
2.4	What did you learn?	3
3	Version Control log	4
4	Work Log	4

1 SOLUTION DESIGN

For this solution we started off by taking the advice offered in the assignment write up. We followed the advice given and found an already developed block device driver as to not recreate the wheel. We chose to use the Simple Block Driver (SBD) that we found on a blog by Pat Patterson. We then studied this device driver and spent time figuring out where the reads and the writes occur. We looked for these occurrences because we knew that it would be where we would need to encrypt and decrypt the data that will be written to and read from the disk. Once we identified where the reads and writes were happening, we were able to identify the places we need to do the encryption and decryption. Once we knew where we needed to perform the encryption and decryption, it was fairly simple to encrypt and decrypt the data based on examples found in the kernel. The final step of our design was to use printk statements to ensure that the encryption and decryption were being handled correctly. We also kept a log of each command needed to build the module and import it correctly. We continuously updated and improved the log so that it was more correct with each iteration.

2 QUESTIONS

2.1 What Do you think the main point of this assignment is?

There were several points to this assignment. The first of which was to get familiar with block drivers. The point of this assignment was certainly not to write our own device driver however this assignment still gave us a chance to understand and learn about block drivers. The second main point of this assignment was to allow us to gain some experience using kernel modules. This includes building the module, importing it into the kernel and mounting this module on the fly. The final point of this assignment was to learn how to use the crypto service provided by the kernel.

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

We started by reading the Linux Device Driver (LDD3) resources to see how they implemented drivers to encrypt block devices on older kernel versions. Then we found a source online that had a majority of the functionality implemented already. We then modified functions that allowed us to encrypt block devices and decrypt them while reading.

2.3 How did you ensure your solution was correct? Testing details, for instance. Ensure this is written in a way that the TA's can follow to ensure correctness.

- 1) First, we need to source the necessary configuration file.

```
source /scratch/opt/environment-setup-i586-pocky-linux
```

- 2) clean the kernel to ensure a sanitary environment.

```
make clean
```

- 3) make the kernel.

```
make -j4 all
```

- 4) run the kernel.

```
qemu-system-i386 -redir tcp:5541::22 -nographic -kernel arch/x86/boot/bzImage -drive file=
```

- 5) make the block device driver.

make

- 6) copy the module over to the running kernel.

```
scp -P 5541 sbd.ko root@localhost:~
```

- 7) Set up and insert the kernel module containing your encrypted block device.

```
cd ~
insmod sbd.ko
shred -z /dev/sbd0
mkfs.ext2 /dev/sbd0
mount /dev/sbd0 /mnt
lsmod
```

- 8) finally check test that the module is working.

```
echo "This is my awesome test."
touch /mnt/testfile
ls -la /mnt
```

```
echo "Insert Test Data"
echo "This is my awesome test" > /mnt/testfile
cat /mnt/testfile
```

```
echo "Search for test data in module"
grep -a "Test Data" /dev/sbd0
```

```
echo "display the contents of the module"
cat /dev/sbd0
```

```
echo "Display the test file"
cat /mnt/testfile
```

```
echo "Delete test file"
rm /mnt/testfile
```

- 9) umount /dev/sbd0
rmmod sbd.ko
lsmod

2.4 What did you learn?

We learned how the kernel reads and writes encrypted data from a block device. We also learned how different kernels may implement the process of reading and writing encrypted data differently. While kernels can be very different, most block devices have many similarities and may be modified to fit the specific kernel you are working with.

3 VERSION CONTROL LOG

Author	Date	Message
James Zeng zengja@oregonstate.edu	Fri May 18 20:33:55 2018 -0700	Worked on getting block device up and working.
Thomas Noelcke noelcket@oregonstate.edu	Tue May 15 19:45:56 2018 -0700	Got Block device working.

4 WORK LOG

Name	Task	Date	Total Time
Thomas	Set up document	5/15/2018	0.5 Hours
Thomas and James	Work on Encrypted Block device design and Implementation	5/15/2018	2.0 Hours
Thomas and James	Work on debugging encrypted block device	5/15/2018	1.0 Hours
Thomas, James, Hayden	worked on getting write up started.	5/17/2018	2.0 hours.
Thomas	Started filling out sections in the write up.	5/17/2018	1.0 hours.
Thomas	Wrote the design section along with the point of the assignment question.	5/18/2018	1 hour
Thomas	Created git log section and filled in table.	5/20	0.5 hours
James	Answered the remaining questions on the report.	5/20/2018	0.5 Hours
Hayden	Edited entire report for accuracies and grammar	5/23/2018	1 hour