Design and Implementation of a Windows Kernel Driver for LUKS2-encrypted Volumes

I do not know yet whether I want to have a subtitle, have a placeholder for now

Max Ihlenfeldt

Universität Augsburg
Lehrstuhl für Organic Computing
Bachelorarbeit im Studiengang Informatik

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contents

1 Introduction **1**

2 Background **2**
2.1 The LUKS2 disk encryption specification 2
2.2 Introduction to Windows kernel driver development 2
2.2.1 Structure and Hierarchy of the Windows operating system 2
2.2.2 The Windows Driver Model for kernel drivers 2
2.2.3 Communication between kernel and userspace 2

3 Related work **3**
3.1 Linux kernel implementation of LUKS2 3
3.2 Measuring filesystem driver performance 3
3.3 Other implementations of encrypted filesystems 3

4 Design and implementation of our driver **4**
4.1 Failed attempts 4
4.2 The working WDM driver 4
4.2.1 Initialization and configuration 4
4.2.2 De-/encrypting reads and writes 4
4.2.3 Handling other request types 4
4.3 Security considerations 4

5 Performance of our driver **5**
5.1 Experimental setup 5
5.2 Results 5

6 Discussion **6**

7 Conclusion **7**

List of Figures **8**

List of Tables **9**

List of Listings **10**

References **11**

1 Introduction

Explain use case etc.

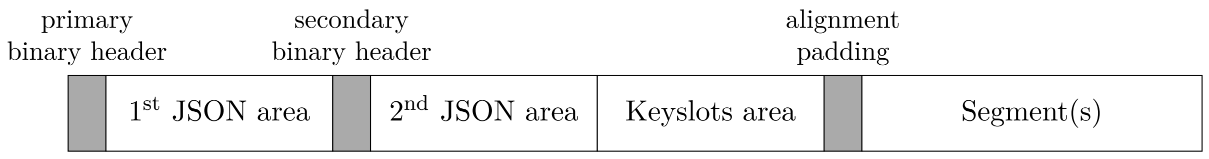

Figure 1: LUKS2 on-disk format (modified after [2])

2 Background

2.1 The LUKS2 disk encryption specification

also [1]

Linux Unified Key Setup 2, or short LUKS2, is the second version of a disk encyption standard. It provides a specification [2] for a on-disk format for storing the encryption metadata as well as the encrypted user data. Unlocking an encrypted disk is achieved by providing one of possibly multiple passphrases or keyfiles. The intended usage of LUKS2 is together with the Linux dm-crypt subsystem, but that is not mandatory[1].

What are the differences between LUKS2 and LUKS? Besides new password hashing functions (I think)

Mention own `luks2` Rust crate

2.2 Introduction to Windows kernel driver development

This section gives an introduction on the development of Windows kernel drivers and related important concepts.

2.2.1 Structure and Hierarchy of the Windows operating system

Roughly summarize important concepts from chapters 1 and 2 of [3]

2.2.2 The Windows Driver Model for kernel drivers

Also explain how it gets loaded (if not done already)

2.2.3 Communication between kernel and userspace

Via ports

[1] As we show in this thesis it is possible to make the combination of LUKS2 and Windows work.

3 Related work

3.1 Linux kernel implementation of LUKS2

3.2 Measuring filesystem driver performance

3.3 Other implementations of encrypted filesystems

VeraCrypt

4 Design and implementation of our driver

4.1 Failed attempts

FilterManager framework

Mention KMDF / UMDF and why we didn't use that if not already done in earlier section

4.2 The working WDM driver

Why WDM?

4.2.1 Initialization and configuration

`luks2filterstart.exe`

4.2.2 De-/encrypting reads and writes

custom AES implementation

```
VOID
EncryptWriteBuffer(
    PUINT8 Buffer,
    PLUKS2_VOLUME_INFO VolInfo,
    PLUKS2_VOLUME_CRYPTO CryptoInfo,
    UINT64 OrigByteOffset,
    UINT64 Length
)
{
    UINT64 Sector = OrigByteOffset / VolInfo->SectorSize;
    UINT64 Offset = 0;
    UINT8 Tweak[16];

    while (Offset < Length) {
        ToLeBytes(Sector, Tweak);
        CryptoInfo->Encrypt(
            &CryptoInfo->Xts, Buffer + Offset,
            VolInfo->SectorSize, Tweak
        );
        Offset += VolInfo->SectorSize;
        Sector += 1;
    }
}
```

4.2.3 Handling other request types

4.3 Security considerations

How does `cryptsetup` send the master key to `dm-crypt`?

5 Performance of our driver

5.1 Experimental setup

5.2 Results

6 Discussion

7 Conclusion

List of Figures

1 LUKS2 on-disk format (modified after [2]) 2

List of Tables

List of Listings

References

[1] C. Fruwirth, *LUKS1 On-Disk Format Specification Version 1.2.3*, 2018. [Online]. Available: https://mirrors.edge.kernel.org/pub/linux/utils/cryptsetup/LUKS_docs/on-disk-format.pdf

[2] M. Broz, *LUKS2 On-Disk Format Specification Version 1.0.0*, 2018. [Online]. Available: https://gitlab.com/cryptsetup/LUKS2-docs/-/raw/861197a9de9cba9cc3231ad15da858c9f88b0252/luks2_doc_wip.pdf

[3] P. Yosifovich, D. A. Solomon, and A. Ionescu, *Windows Internals, Part 1: System architecture, processes, threads, memory management, and more.* Microsoft Press, 2017.