Forensic Issues in IoT devices using NAND Storage

Suchit Reddi

Under supervision
of
Dr. Sonal Singhal

Abstract

- This project focuses on rendering data unrecoverable on IoT devices using NAND-based storage like SSDs.
- The recoverability of a file depends on the sanitization technique used and determination of attempting entity.
- Working of modern drives differs from classic mechanical drives, rendering old sanitization methods ineffective
- Methods suitable for SSDs were researched, and the ones compatible with IoT devices were selected.
- The selected methods were implemented as a command-line utility.

Introduction

- How is NAND flash storage advantageous?
- How does the NAND flash controller trick the operating system?
- Garbage collection, TRIM, and wear leveling were introduced in SSD devices. Are they helping or hindering sanitization?
- What is verification? How is it done?
- Are all the methods compatible with every device?
- Why should IoT devices care about sanitization?
- And what happens if it is neglected?

IoT in major crimes

IoT devices are an easy target compared to more powerful machines like laptops and mobiles.

- Target Breach - Credit card data of over 70 million customers was stolen. Data stored in RAM and disk storage of POS systems was scraped using memory scraping.
- Mirai Botnet - This botnet had over 6,00,000 IoT devices. It only targeted IoT devices, and most of them used NAND flash storage, which allowed it to spread faster.
- WannaCry - WannaCry affected over 2,30,000 in 150 countries. This devastating malware spread and locked down the firmware stored on the boot partition of storage devices.

Firmware-based SANITIZE

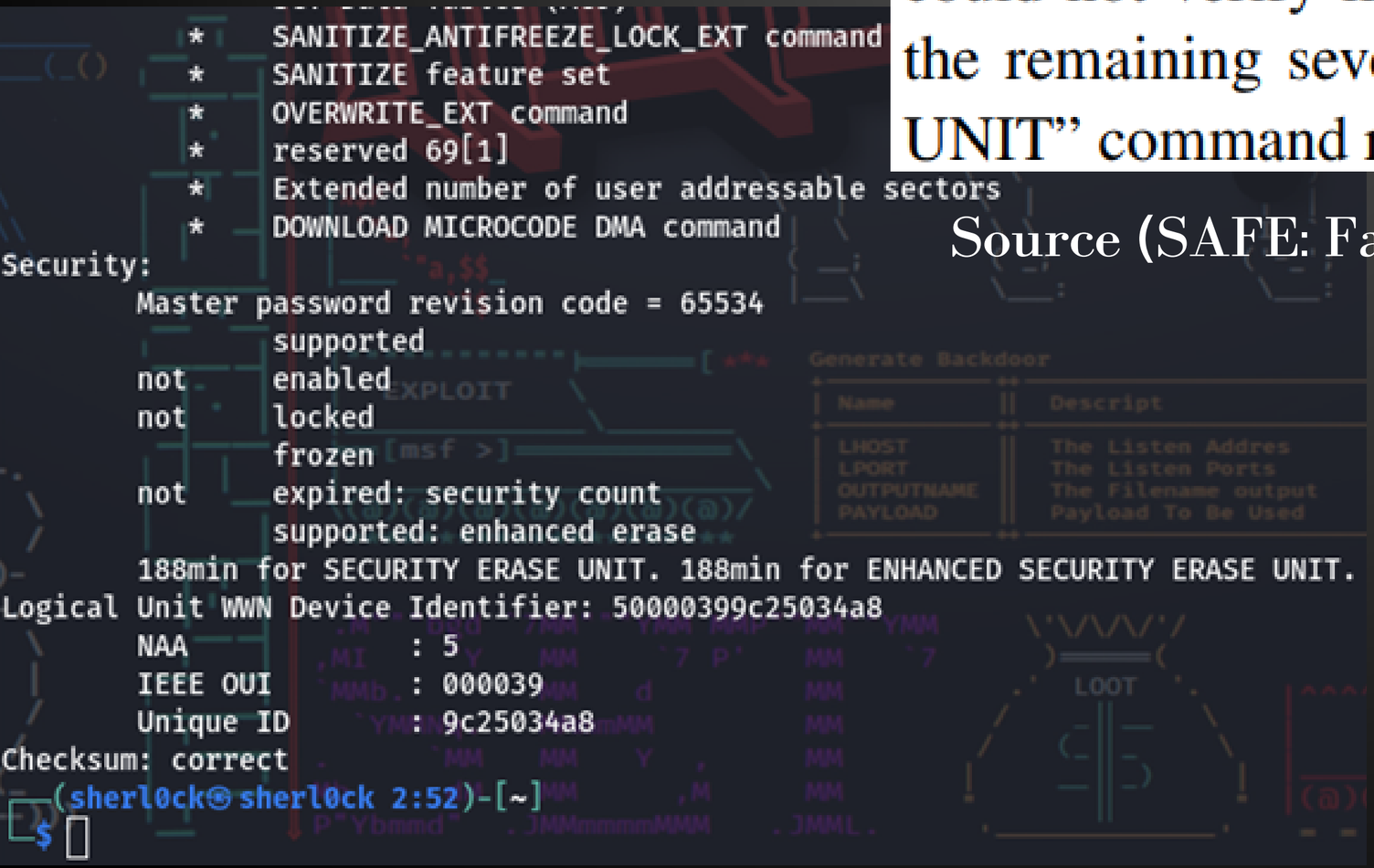
- Manufacturer provides sanitization commands in the device firmware, accessible by specific tools.
- How the storage device is interfaced determines its compatibility with an IoT device. (ATA, SATA, SCSI, UAS)
- SANITIZE feature set in hdparm must be supported.

hdparm --user-master u --security-erase-enhanced $strongp $partition

- The enhanced secure erase command claims to wipe all data from every block, including the overprovisioning area.
- But do you trust the manufacturer to sanitize your personal data? Some researchers found that these companies can lie!

is not surprising, since the standard is not yet final. Eight of the drives reported that they supported the ATA SECURITY feature set. One of these encrypts data, so we could not verify if the sanitization was successful. Of the remaining seven, only four executed the “ERASE UNIT” command reliably.

Source (SAFE: Fast, Verifiable Sanitization for SSDs)



Compatibility of SANITIZE feature set

Other methods

In the firmware-based method, hdparm, there are a few more methods:

- Block erase: It raises each block to a voltage higher than the standard program voltage (erase voltage) and drops it to the ground, returning the block to a “Fresh-out-of-box” state.
- Crypto scrambling: It rotates the internal cryptographic key in self-encrypting drives by sanitizing the key storage area.
- Trim sector ranges: It can only allocate certain sectors as free for the garbage collector. But it does not erase data.

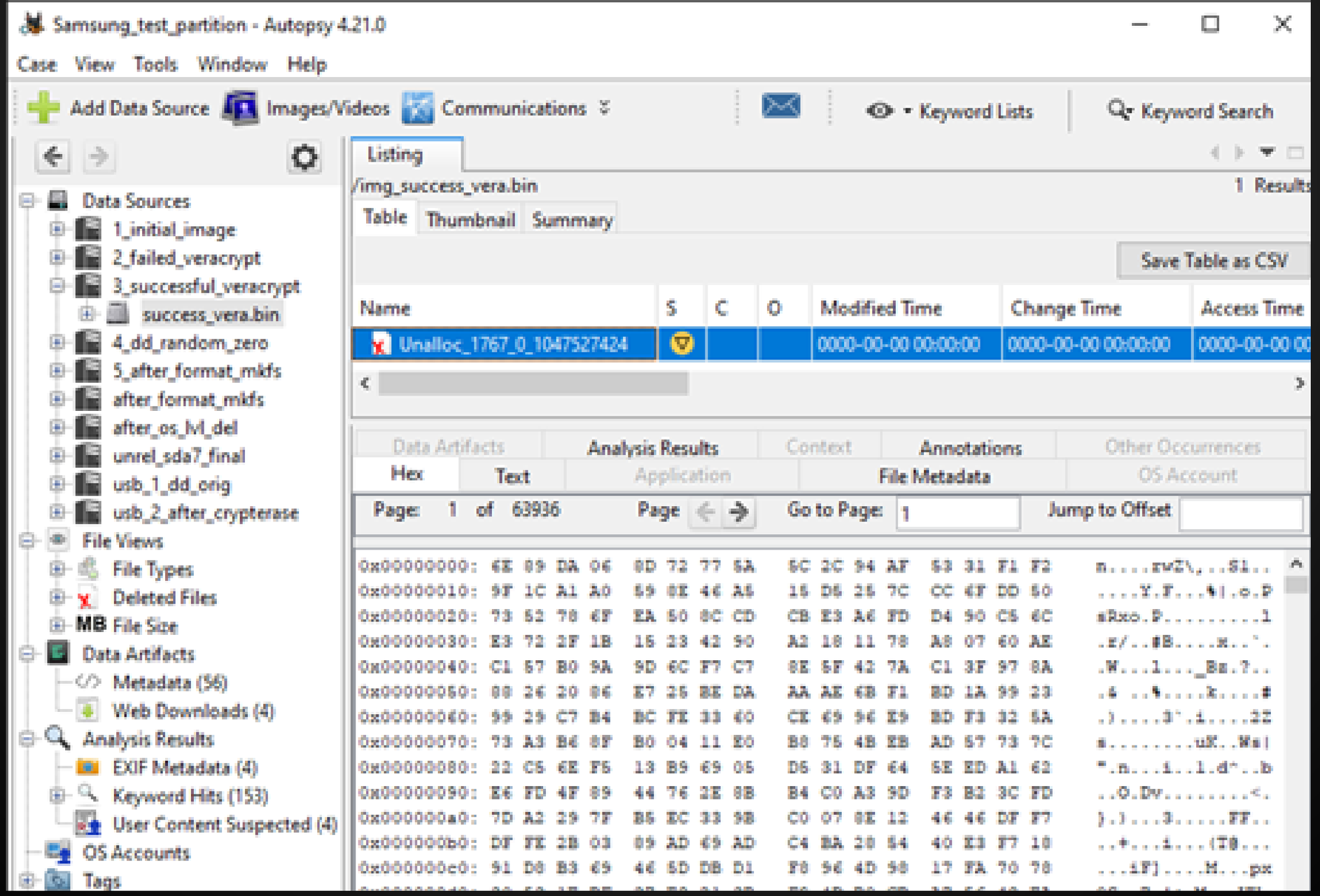
hdparm --trim-sector-ranges 66634:56665 … /dev/sda

Cryptographic Wipe

- We came up with this to compensate for incompatibility.
- The complete storage device is encrypted using VeraCrypt, which supports arm architecture (used by IoT devices like Pi).
- The encrypted drive, along with the key, is overwritten with random values. But the OP area might still have bits of data.

dd if=/dev/random of=$partition bs=1M status=progress

- We overwrite the drive again with zeros to force the controller to either swap OP blocks for encrypted/randomized/zeroed blocks or to erase existing blocks. We format it in the end.
- This process replaces personal with useless data in OP area, greatly reducing the chances of recovery.



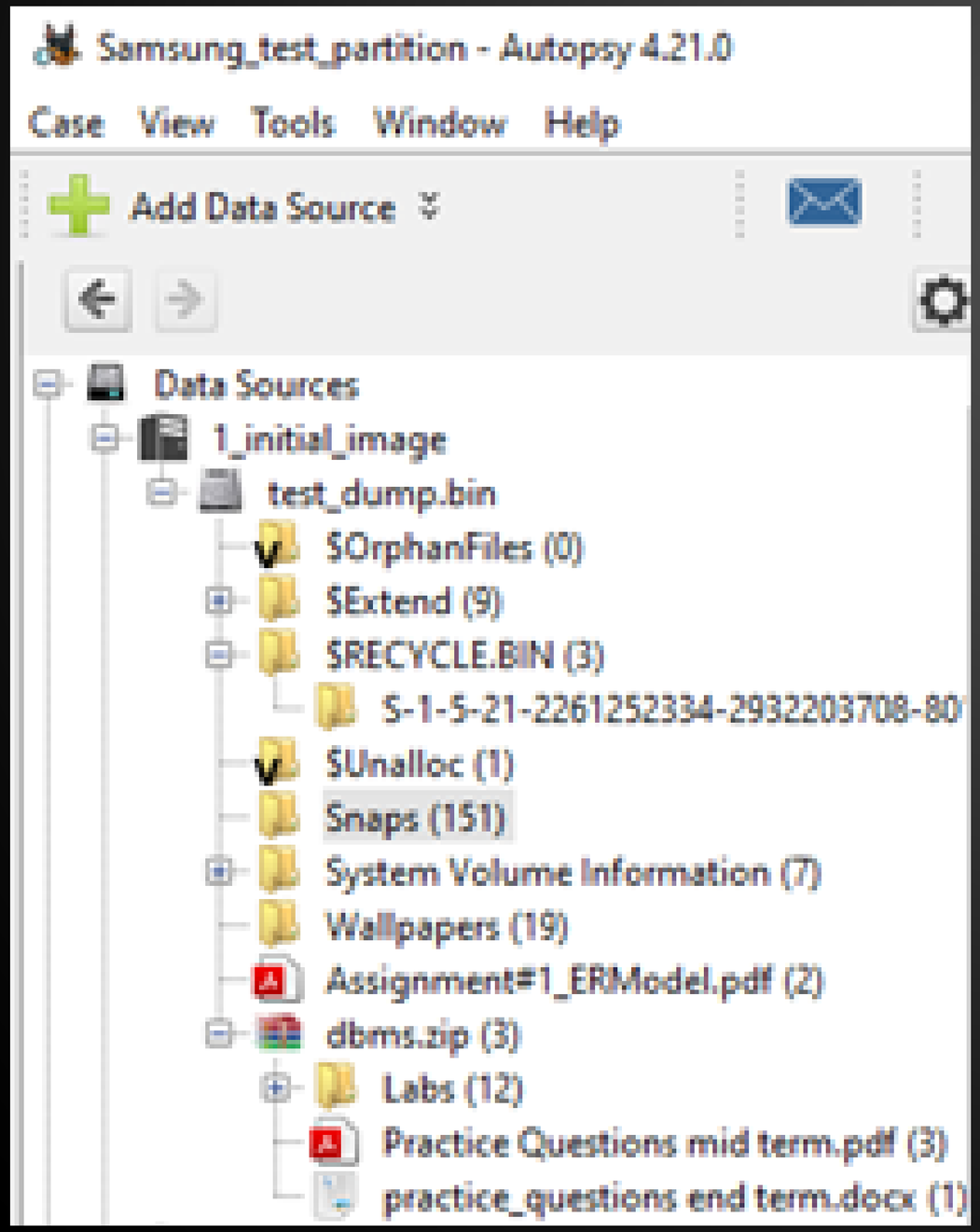
What does encryption do to data?

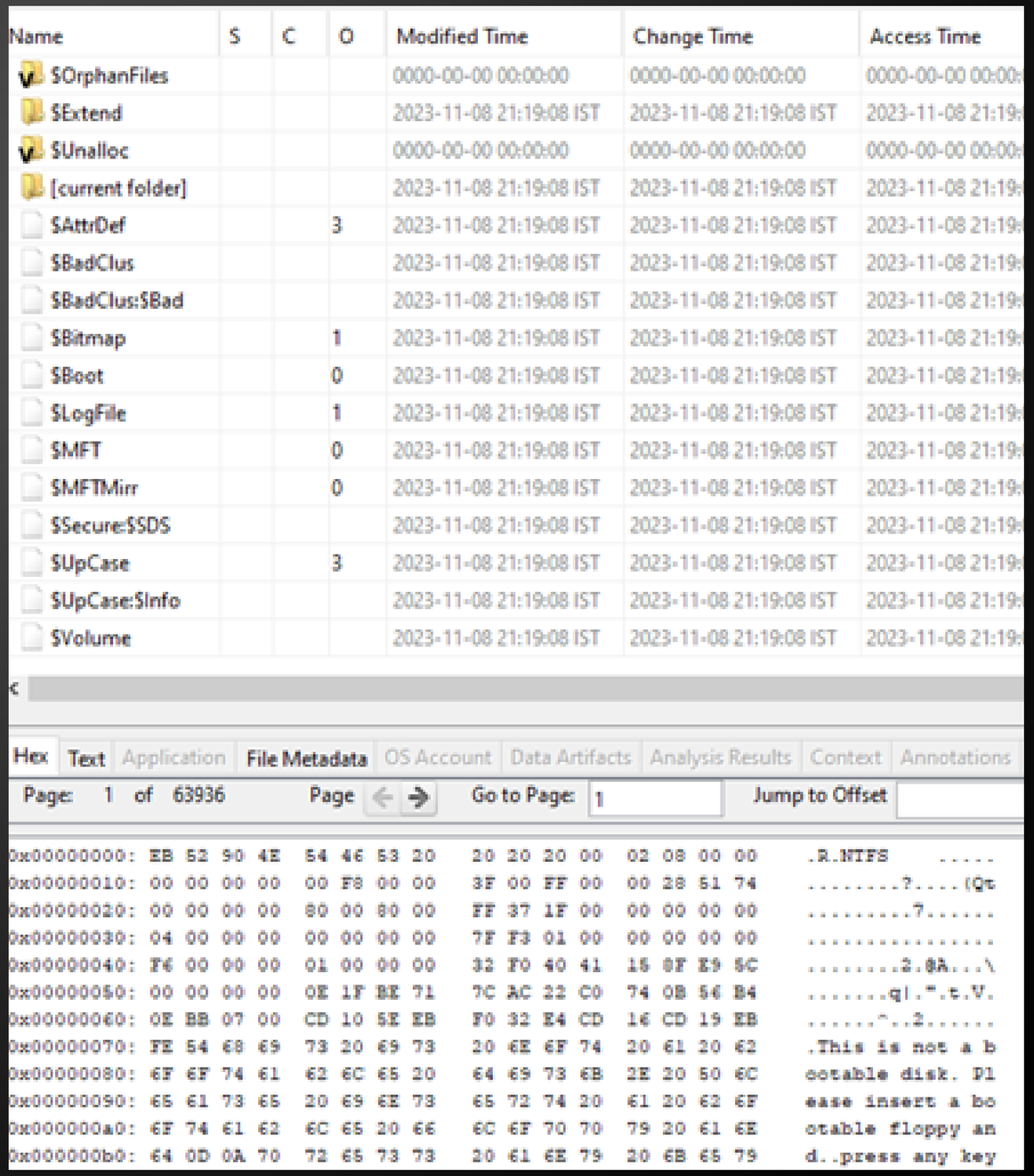
Verification

- Is the sanitization successful? How will you check it?
- Take a binary image of the storage device at each necessary step of the process using the *dd* command.

  dd if=$partition of=/<out location>/image.bin status=progress

- Use a forensic recovery tool (Autopsy) to recover still readable files from these images.
- A command-line utility tool was prepared, which included all the discussed methods and a few more variations of it.
- This tool was used for testing sanitization and verification processes in an automated way.





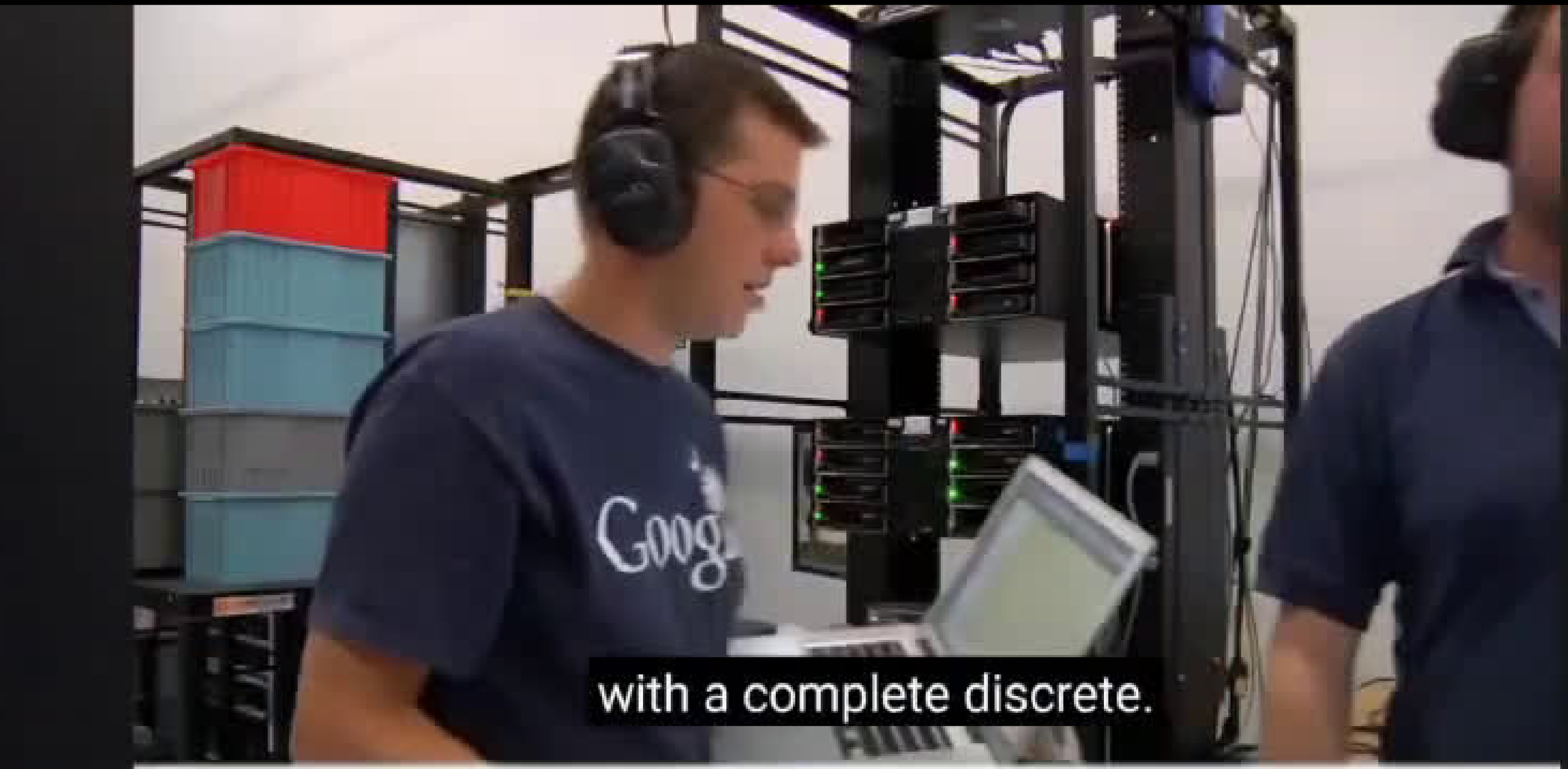
Did it actually work?

Memorywipe Demo

GitHub Link: https://github.com/SuchitReddi/memorywipe

- Demo - Running on a Raspberry Pi connected to an SSD.
- Demo - Verification results on Autopsy

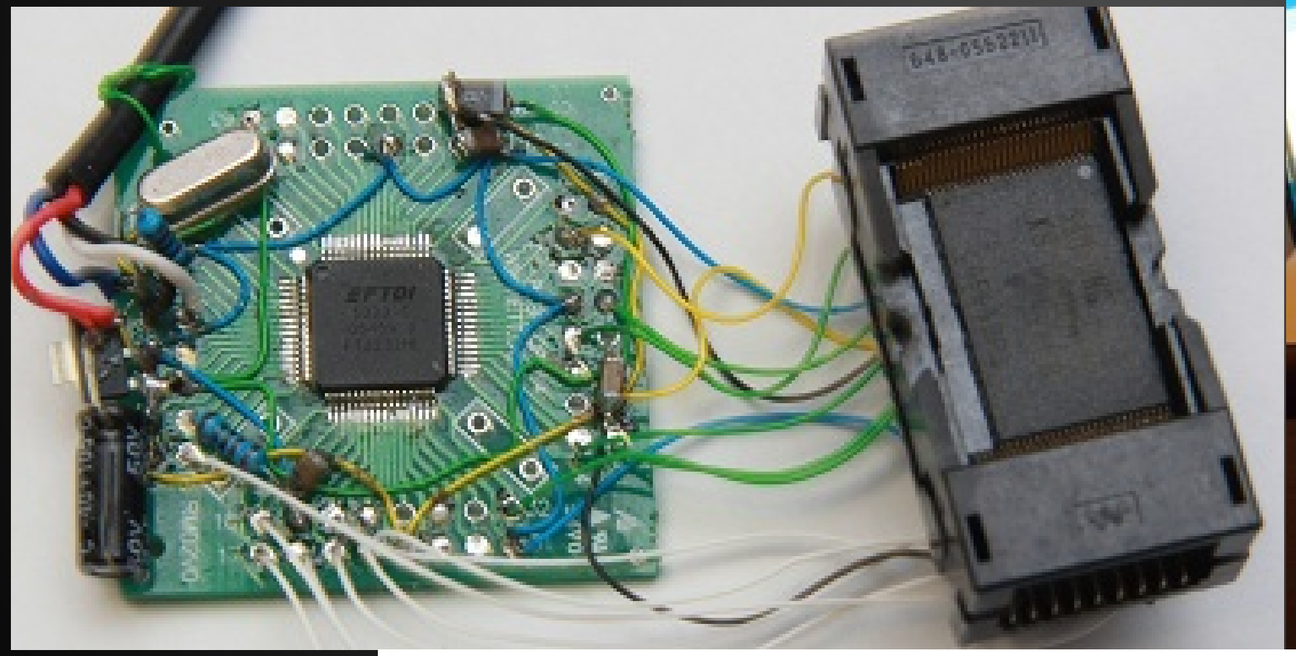
Sanitization in data centers



Future Works

- The tool can be improved with low wear-inducing methods.
- Completely automate the tool for IoT devices with limited access and explore options when no shell access is available.
- Possibility of sanitizing mobile storage using adb interface.
- Tools specific to widely used OS, device interface, or type.
- Hardware-based operations for sanitization and verification.
- Researching into Factory Access Mode, Flash Transition Layer, mtd-devices.

Hardware methods and complexity



Conclusion

- Importance of disk storage sanitization for IoT devices
- How NAND flash storage affected the game of sanitization.
- Discussed sanitization techniques targeting NAND storage.
- The most compatible and feasible techniques were converted into a command line tool open for public use.
- The need for hardware methods to gain better control.