ECE 443/518 – Computer Cyber Security
Lecture 22 File System Forensic Analysis and Disk Encryption

Professor Jia Wang
Department of Electrical and Computer Engineering
Illinois Institute of Technology

November 14, 2022

Outline

Reading Assignment

- Please refer to the books below for more details
  - "File System Forensic Analysis", Brian Carrier
  - "Digital Forensics and Incident Response", Gerard Johansen
- Next lecture: ICS 19

Outline

File System Forensic Analysis

Disk Encryption

Data Acquisition and Analysis in Storage Devices

- A layered approach.
  - As identified by tools and APIs to access data.
  - Need more resource as we move toward lower layers.
  - More data are lost as we move toward higher layers.
- Sector: physical medium, every bytes.
- Volume: logical organization of sectors.
  - E.g. partition and RAID arrays
  - But may miss sectors not allocated to volumes.
- File system: name, metadata, and content.
  - E.g. documents and photos.
  - But may miss deleted files.
- OS/Application: interpretation of content.
  - E.g. browsing history.
  - But may miss information hidden in other files.
- Special systems like swap and databases may skip one or more layers.

Disk Data Acquisition

- BIOS vs Direct access
  - BIOS may provide unified interface to facilitate disk data acquisition.
  - However, discrepancy exists, e.g. due to misconfiguration or compromised BIOS.
  - Prefer direct access of disk data via disk firmware.
- Dead vs Live acquisition
  - Live acquisition: from within the suspect operating system.
  - But the suspect operating system may be compromized to hidden information.
  - Prefer dead acquisition using a trusted OS.
- Write blockers: prevent accidental writes using either software or hardware.
- What should you trust?
  - Application, OS, firmware, and even hardware could have bugs or be compromised already.

Volume Analysis

- Assemble sector data into volumes.
- Volumes provide OS an abstraction of physical disks.
  - Divide a disk into partitions for different purposes.
  - Combine multiple disks into a single logical one to simplify management.
  - Support advanced features like error-correction, encryption, snapshots, and backups that are transparent to OS.

Master Boot Record (MBR)

- Widely used on PC systems since DOS
  - And later on Windows and Linux.
- Stored in the first 512-byte sector.
  - Boot code (446 bytes): instructions to boot from the disk.
  - Partition table (4 entries of 16 bytes each): locations and sizes of at most 4 primary partitions.
  - Signature (2 bytes): 55, AA
- Sector 1-62 are reserved – potential locations to hide data.
- Additional (> 4) partitions can be supported via logical partition within primary partitions.
- 32-bit sector addresses limit MBR to 2TB and smaller disks.

GUID Partition Table (GPT)

- To replace MBR on larger disks and newer systems.
  - Use 64-bit logical block addresses (LBA) of sectors.
- Typical layout with 512-byte sectors.
  - LBA 0: reserved for limited backward compatibility with MBR.
  - LBA 1: GPT header.
  - LBA 2-33: Partition Entry Array, 128 entries of 128 bytes each.
- A secondary GPT is located toward the end of the disk for backup purposes.

Redundant Arrays of Inexpensive Disks (RAID)

- Use multiple cheap disks together to provide redundancy and/or performance that are only available for expensive disks.
- RAID levels (with N disks)
  - RAID 0: striping, no capacity loss, no redundancy
  - RAID 1: mirroring, 1/N capacity, survive N-1 disk failures
  - RAID 5: parity, lose 1/N capacity, survive 1 disk failure
  - RAID 6: double parity, lose 2/N capacity, survive 2 disk failure
- Read throughput usually grows with N.
- Write throughput depends on parity calculation and number of data copies.
- Nested RAID levels may help to organize more disks better.

Hardware vs. Software RAID

- Hardware RAID
  - Via a special piece of hardware called RAID controller.
  - Usually has better performance than software RAID.
  - May need specific driver to acquire data.
  - Data layout on disks is usually not published – difficult to acquire data if the RIAD controller is missing, while hiding data could be possible.
- Software RAID
  - As part of OS, w/ or w/o hardware support.
  - Possible to acquire data without the suspect OS – data layout is usually known.
- It remains the safest to acquire contents of disks in addition to the RAID volume.
  - Nevertheless, the presence of RAID system already indicates that you need a large storage system to store the acquired data.

File System Analysis

- Structured and organized data.
  - Build on top of fixed size blocks.
- Data categories
  - File system: general file system information
  - File name: usually as part of directory content
  - Metadata: content location, size, dates, ACL etc.
  - Content: actual data, or file names and metadata locations for directories.
  - Application: file system or application features like journals or file headers.
- Specific OS may have special ways to deal with specific file systems.
  - E.g. if a file is truly erased or just marked for deletion.

General Analysis Guideline I

- File system category
  - Root metadata location.
  - Volume ID etc. to help decide when and where the file system was created.
  - Reserved spaces that may contain hidden data.
- File name category
  - In many cases, file names together with the directory structure already reveal a huge amount of information.
  - Perform a directory walk from the root.
- Content category
  - Blocks allocated to files and directories.
  - Find interesting keywords in specific files.
  - Find interesting keywords w/o file name or in deleted files from all blocks (allocated and unallocated).

General Analysis Guideline II

- Metadata category
  - Location of content blocks.
  - Slack space for hidden data if file size is not multiple of block size.
  - Information regarding things like compression and encryption.
- Application category
  - Journals help to maintain file system consistency when OS crashes. They are essentially logs of recent file system events.
  - Pattern in file headers may help to reveal actual file type.

Recover Deleted Files

- OS may speed-up file deletion by simply unallocating metadata and content blocks.
  - Disks are slow.
- Recover using metadata
  - Find an unallocated block that looks like metadata.
  - Assume content block locations are not erased.
  - Assume content blocks remain unallocated.
- Recover using application data
  - Find an unallocated block with specific pattern.
  - Assume content blocks are allocated consecutively.

Wiping

- A.k.a. secure delete
- Write zeros or random data to allocated locations before completing a file deletion.
  - Prefered.
  - But this will slow down the deletion process.
- Write zeros or random data to unallocated locations.
  - On a regular basis, or at special occasions.
  - This may be even slower if a lot of locations are unallocated.
- Wiping will make digital forensics harder.
  - But we have good reason to practice wiping – what if someone steals your phone right after you delete a sensitive file?
  - There are data erasure standards requiring wiping disks multiple times.
- Any better method to address the "slow" issue with wiping?

Challenges with Solid State Drives (SSD)

- SSDs operate in a completely different way than HDD.
- Wear leveling: a controller inside a SSD constantly copies blocks around to avoid excessive writes to a single block.
- There should be enough blocks that are not in use – the OS must tell the controller when a block is no longer in use.
- As a consequence, unallocated blocks are “erased” immediately upon file deletion.
  - Impossible to recover.
- The SSD controller may also support self-erasing, allowing the SSD to erase itself upon the next power-on event.
  - You need this feature to remotely erase a lost device.

Outline

File System Forensic Analysis

Disk Encryption

The Need for Disk Encryption

- Access control cannot provide any protection to data on a disk if adversaries obtain physical access to the disk and apply file system forensic analysis.
  - Lost cellphones, laptops, and external drives.
  - Liquidated servers.
  - Repair and warranty services.
- Data erasure standard exist, but
  - Only apply to a system at end-of-life.
  - Very time consuming: usually multiple full disk writes are required, may take days to complete for a single large disk.

Usability

- Advances in cryptography make it possible to provide very strong encryption of data.
- The remaining problem is how to implement them in the production system so that
  - People use them on a daily basis to protect their data.
  - People are less likely to make any mistake to leak sensitive information.

File Encryption

- Encrypt individual files as requested.
  - Use a tool, e.g. any zip program.
  - Use functionality provided by a specific file system.
- Concerns
  - Unencrypted Metadata of the file, e.g. file name, may leak certain information.
  - One may accidentally remove encryption from the file.
  - People tend to forget strong passwords if not used on a regular basis.

Transparent Encryption

- Encrypt on-the-fly so users won't notice it.
- Usually apply to the whole disk.
  - User is required to "unlock" the disk when the system starts, then use the system as usually without worry about encryption.
- Concerns
  - Performance: faster disks require encryption to be done faster so users won't notice it – hardware accelerations and multicore processors do improve performance but fast SSDs introduce new challenges.
  - Digital forensics: how to protect user data while allowing law enforcement to locate evidences remains an open problem.

The Case of TrueCrypt

- Was a very popular software supporting transparent encryption before 2015.
  - Support majority of operating systems.
  - Source code available.
- Plausible deniability: to deny the use of TrueCrypt, or file encryption tools in general.
- On 5/28/2014, the TrueCrypt official website warned that the software may contain unfixed security issues, and that development of TrueCrypt was ended.

The Case of BitLocker

- BitLocker: full volume encryption supported by Windows
- Self-encrypting drives
  - A drive with built-in accelerators for full-disk encryption – especially useful for encrypting/decrypting fast SSDs.
  - As standardized by Opal Storage Specification.
  - BitLocker was replying on such accelerators whenever presented.
- On 11/5/2018, a research work indicated that some SSDs are equiped with flawed accelerators.
  - The key is not tied to user password.
  - Key bits are not random enough.
  - Other design and implementation issues.
- On 9/24/2019, a Windows update changes BitLocker so it no longer trusts any accelerators by default any more.

Summary

- File system forensic analysis help to recover and investigate material found in storage devices.
- Disk encryption help to protect user data if one losts physical control over storage devices.