Introduction to Digital Forensics - Overview

Brian Olliff

Defensive Engineering Instructor

Key Concepts

- Forensics basics
- Evidence and its collection
- Endpoint forensics

MAJOR TOPICS

- Digital forensics basics
- Evidence collection & preservation
- Windows forensics
  - Memory management
  - Registry
  - File system
- Linux forensics

LEARNING OUTCOMES

- Understand the fundamentals of digital forensics
- Describe different types of evidence collection and information that can be gathered
- Be able to explain how to properly preserve evidence using a chain of custody
- Understand components of Windows processes, Registry, and file system
- Be familiar with Linux processes and file system

PREREQUISITES

- Basic understanding of cybersecurity fundamentals
- Understanding of Windows systems

Let’s Get Started!

Introduction to Digital Forensics

+ Aligned with Cisco CyberOps Associate
+ Starting with digital forensics basics
+ Evidence collection and preservation
+ Windows & Linux forensics

Digital Forensics

Digital Forensics

- Identification, collection, examination, and analysis of data
  - Evidence
  - Data can come from any type of device
  - Frequently used for legal purposes (in court of law)
- Requires preservation of evidence integrity
  - Strict chain of custody
- Often included in incident response activities
  - DFIR - digital forensics and incident response
- Attribution - identifying origin or source of attack
  - Can be very difficult to accomplish during investigation
  - Evidence-led investigation

Categories of Forensics Investigations

Digital Evidence

- Information in digital form used to support investigation
- In court - used to prove/disprove facts
  - Provides implication and extrapolations of information

Best evidence	Corroborating evidence	Circumstantial (indirect) evidence
Original form of data System/disk images Exact copies of data	Supports theory Confirms proposition	Relies on extrapolation Supports other evidence

Evidence Collection

Digital Evidence

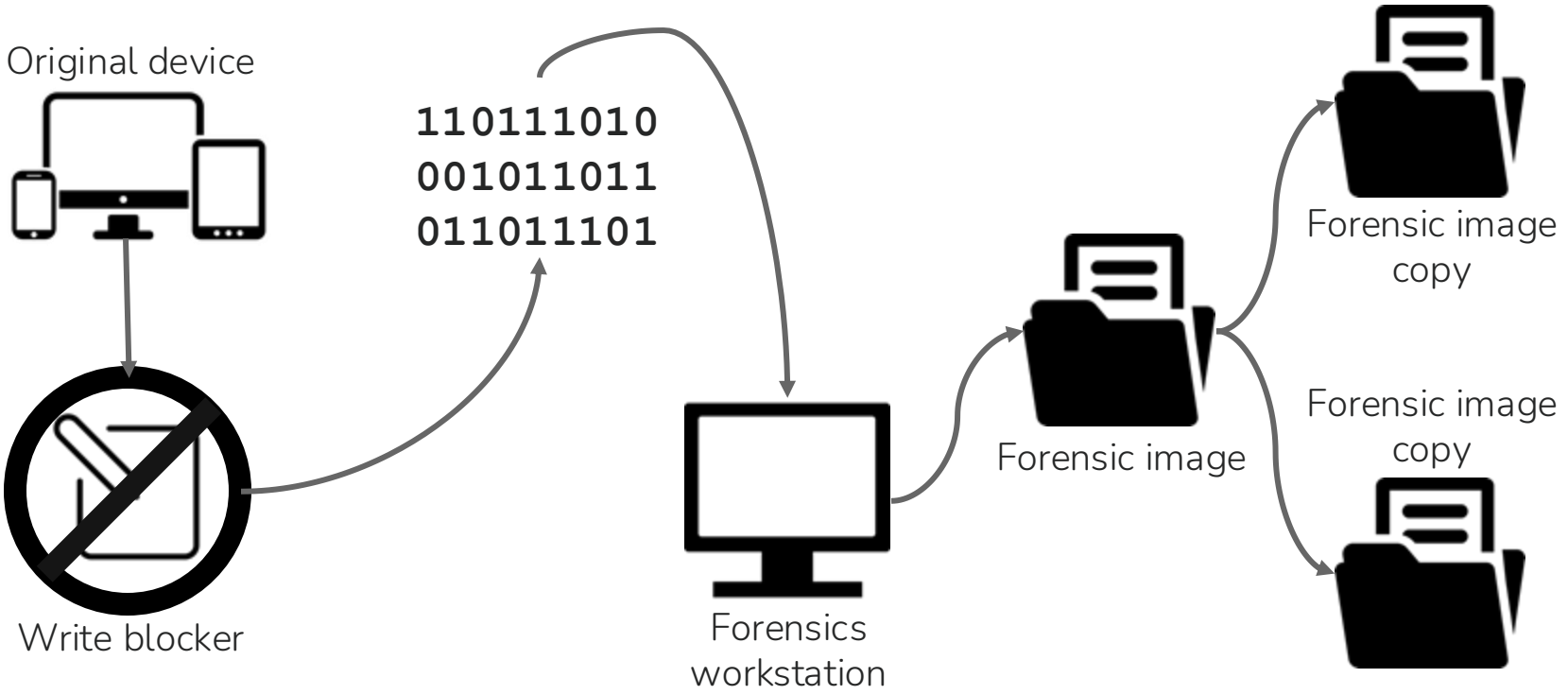
- Takes many forms
- Can originate from almost any device/system
- Integrity of evidence is critical

- Specialized tools used to collect and protect integrity
  - Prevent contamination of device and evidence
  - Write blockers often used

Disk Imaging

- Exact duplicate of source disk
  - All data, including deleted files and free space
  - “Physical copy”, bit-for-bit copy
- Created target file is *forensic image file*
  - “Forensically sound” copy of data
- Original device can be stored or returned to use
  - Reduces need to reexamine device, reduces possible drive failure
- Forensic image then used for analysis (or copy)
  - Should be securely stored when not in use

Disk Imaging



Metadata

- Data about the data
- Can provide important information
  - File creation, modification, deletion, etc dates
  - Location and device information (EXIF data)
  - Will vary depending on file type
- Internal vs external storage
  - Internal - stored with file/data itself
  - External - stored in operating system
- Often requires careful interpretation
- Can provide valuable information about user/attacker activity

Deleted Data

- Files deleted on most devices aren't actually deleted
  - Reference in file allocation tables is removed, data may still remain
    - In "unallocated" drive space
  - Files may still be recoverable until overwritten by operating system
- Attackers commonly delete files needed for investigation
  - Logs, malware, etc
- Recovery through various methods
  - Data carving - recovering files from unallocated space
- Importance of forensically sound disk image

Network Infrastructure Devices

- Data extensively used for monitoring network health and security
  - Not frequently stored on device long term
- Forensics analysis often requires centralized logging system
  - Devices configured to send log data to central system
  - Log system indexes, maintains data
  - Should have capacity to store long term
- Potential for compromise by attackers
  - Log data may be compromised
  - Multiple methods to verify integrity of systems

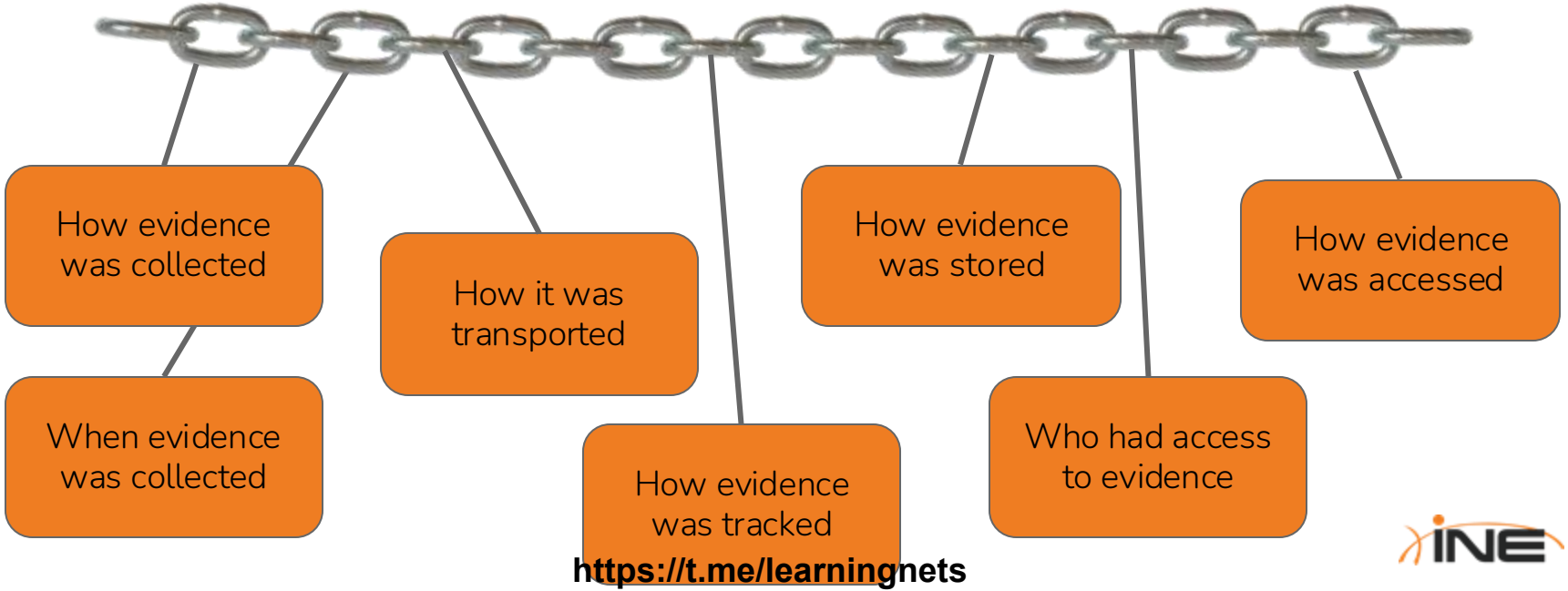
Evidence Collection Challenges

- Encryption
  - When used by attackers, may not be possible to retrieve data
    - Memory data retrieval, exposed encryption key recovery
  - If used by organization, much easier to retrieve
    - Keys often available
- Mobile devices and IoT
  - Logs may not be available depending on device
  - Some mobile devices may be inaccessible without unlocking
  - Sensor data may be encrypted by manufacturer
  - MDM systems can have useful evidence

Evidence Preservation

Chain of Custody

- Documentation of evidence collection
  - From moment forensics investigation begins until presented in court



Preservation Guidelines

- Evidence integrity must be protected
  - Not changed, nothing added, removed, or destroyed (*evidence preservation*)
- Never work with original evidence, always copy (or copy of copy)
  - Disk images, memory dumps, etc
- Evidence should always be labeled
  - Investigator's name, date created, case/ticket number, etc
- Physical media protected and secured
  - ESD bags, specialized evidence collection bags, Faraday cages
  - Transported in secure, lockable containers
    - Responsible person remain with evidence at all times during transport

Reverse Engineering Introduction

Reverse Engineering

- Used for both defensive and malicious purposes
  - Stealing technology, reverse cryptographic algorithms, exploit development
  - Malware analysis, vulnerability research, software development
- Malware analysis
  - Frequently used by researchers, forensics experts
  - Ability to trace steps & actions malware takes
    - Assess damage, determine removal methods, create countermeasures
  - Can assist with forensics investigations
    - Lead to further discovery of infected/compromised systems (IOCs)
    - Help with attribution

Malware Analysis Tools

- System monitoring tools
  - Sniffers, disk/file monitors, process explorers
- Disassemblers
  - Use executable binary to generate assembly language code
  - Shows what software is doing at low machine level
- Debuggers
  - Allow observation of individual steps in program while running
  - Breakpoints - points to pause running for analysis and tracing
- Decompilers
  - Attempt to take binary file (program), produce original high-level code
  - file.exe -> C+ code

Tools

Windows Processes, Services, & Memory

Windows Processes

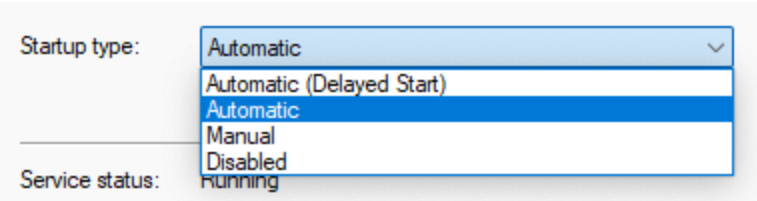
- Process
  - Program running on system, contains all resources needed to execute
  - Made up of one or more threads
- Thread
  - Basic unit that OS allocates processor time to
  - Can be executed at any time, and by other threads
  - Every process starts with *primary thread*
- Job
  - Grouped processes managed as a unit
  - Used to control attributes of associated processes

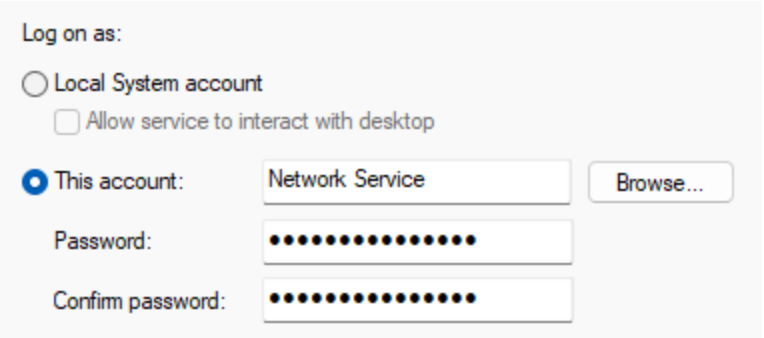
Processes and Services

- Thread pool
  - Group of worker threads to increase efficiency of execution
  - Help to reduce number of application threads & manage worker threads
- Fiber
  - Single unit of execution, manually scheduled by application
- Service
  - Executable applications, usually long-running
  - Run in the background, without user interaction (normally)

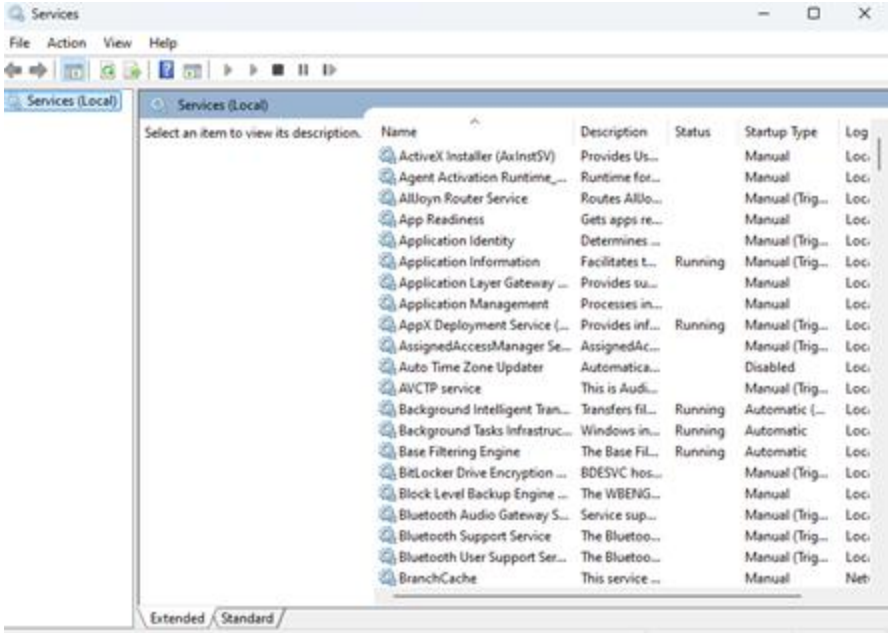
Windows Services

- Multiple startup methods
  - On startup, manually, triggered by another service, disabled
- Security contexts
  - As local system account, or specific account (user, service, etc)





Managing Services




```
PS C:\Windows\System32> get-service winmgmt

Status   Name               DisplayName
------   ----               -----------
Running  winmgmt            Windows Management Instrumentation
```

```
C:\Windows\System32>sc start winmgmt

SERVICE_NAME: winmgmt
        TYPE               : 30  WIN32
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 1272
        FLAGS              :
```

Memory Management

- Volatile memory and nonvolatile memory (NVRAM)
- Static memory allocation
  - Memory allocated by program at compile time
- Dynamic memory allocation
  - Memory allocated at run time
- Heap vs stack
  - Heap - memory set aside for dynamic allocation
  - Stack - memory set aside as “scratch” space for a thread

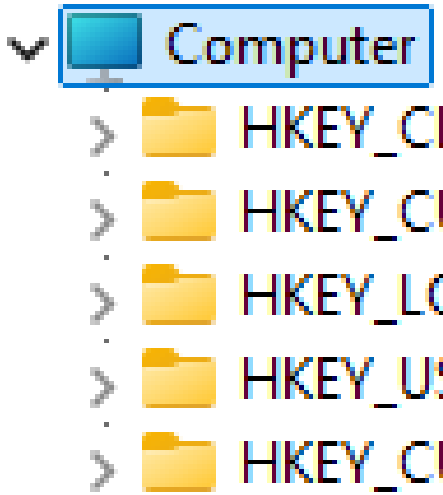
Windows Registry

Windows Registry

- Database that stores information necessary for Windows
  - System configuration, application information, user data, hardware, etc
- Functionality
  - Loads device drivers
  - Runs startup applications/programs
  - Configures and stores environment variables
  - Stores user settings, OS parameters
  - Holds configuration data/information about connected hardware
- Most activities on Windows reference Registry
  - Excellent source of data for forensics investigation

Registry Hives

- Organized by “folder” hierarchy (hives)



→ Windows Explorer configuration

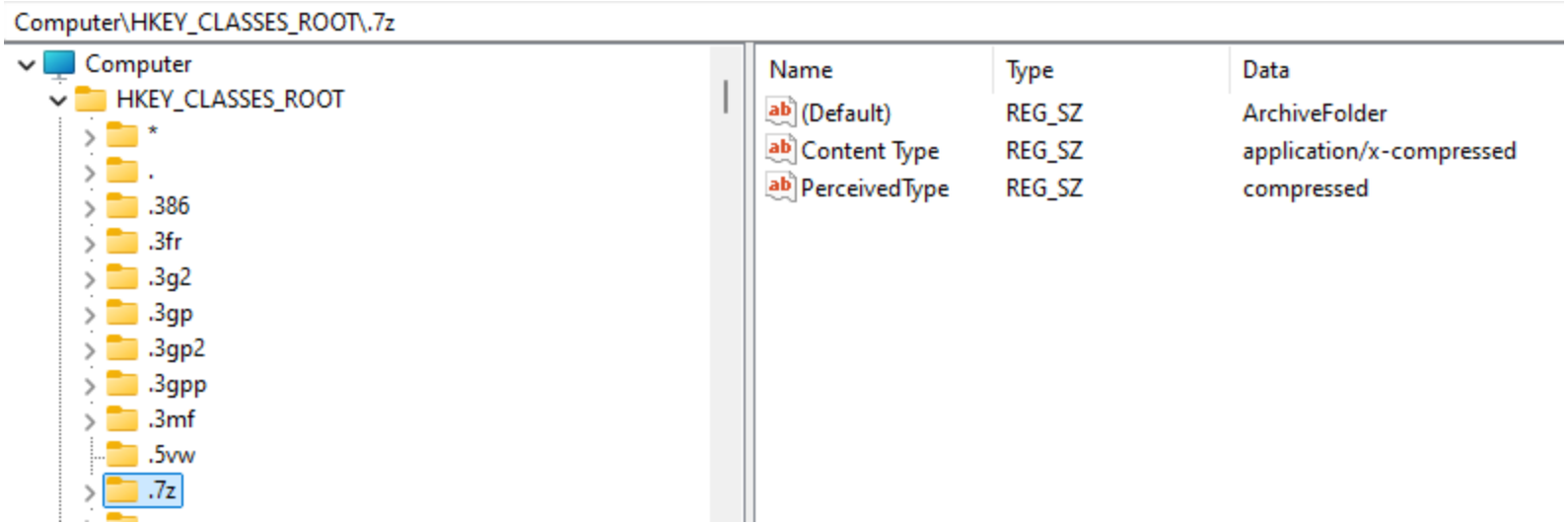
→ Currently logged in user config info

→ Hardware-specific info for OS

→ Info on all user profiles

→ General config info (current)

Keys and Values

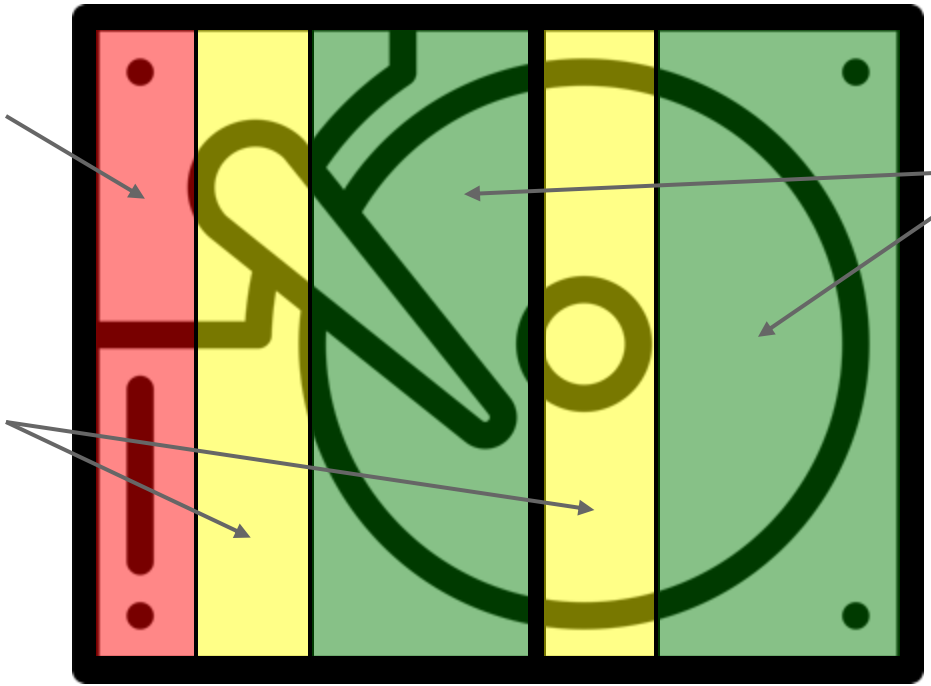


Forensics Use

- Most system-level activities have information in Registry
- All keys/values contain LastWrite time
  - Can be used to approximate date/time event occurred
- Autorun keys
  - Launch processes and programs at boot/login
  - Frequently used by malware for persistence
- MRU (most recently used) list
  - Contains list of entries made due to user actions
- Hardware information
  - USB device usage, mounted devices, hardware changes, etc

Windows
File System

Partitioned Drives



File Allocation Table (FAT)

- FAT, FAT12, FAT16, FAT32, exFAT

- FAT - 8 bits assigned to address clusters
  - Maximum file size 8 MB, no sub-directories
- FAT32 - 32 bits (4 reserved, 28 usable bits)
  - Max file up to 256 GB
- exFAT - 32 usable bits
- Multiple limitations over newer file systems
  - No modern properties added to file
    - Encryption, compression, permissions, etc

NTFS

- Default file system in Windows since Windows NT
- More secure, more features
  - Tracks multiple timestamps
    - MACE - modify, access, create, entry modified
  - ADS - Alternate Data Streams
- Boot sector
  - First sector of partition, contains information about file system
- Master file table ($MFT)
  - Part of file system
  - Metadata entry for every file in partition, each 1,024 bytes
  - 42 byte header (with signature), 982 bytes for metadata
  - If space left, file's data is stored within entry

EFI System Partition (ESP)

- Used by UEFI
  - Unified Extensible Firmware Interface
  - Loads files to start OS and other utilities
- Formatted with file system based on FAT & specified by UEFI
- Includes multiple files
  - Boot loaders/kernel images for installed operating systems
  - Driver files for installed hardware needed at boot
  - System utilities run before OS loaded
  - Data and error log files

Linux Forensics Fundamentals

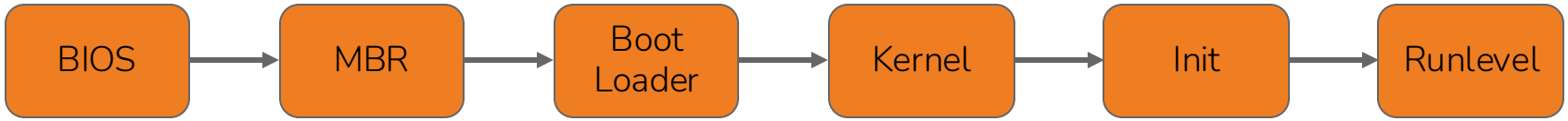
Linux Processes

- Typically also apply to macOS systems
- Processes can run in foreground or background
  - Foreground usually visible on screen (terminal or GUI)
  - Background require other methods to monitor
- **`ps`** command
  - Shows information about running processes
- **`top`** and **`htop`** commands
  - Show additional resource consumption information
    - CPU, memory, network, etc
- Look for suspicious, orphaned or zombie processes

Linux File System

- Two main partitions
  - Data partition - Linux system data, root partition
  - Swap partition
- ***ext4*** - most commonly used file system
  - Maximum file system size of 1 EB (exabyte) = ~1 million TB
  - Supports journaling and improves performance and reliability over ext3/2
- Journaling
  - Maintains record of changes not yet committed (*journal*)
    - Journal - type of circular logging
  - Primary benefit - improved recovery after system crash/power failure
  - ext4 includes checksums of journal data for reliability/integrity
    - Journal is most used part of disk - increased risk of failure

Boot Process



1. Basic Input/Output executes MBR processes
2. MBR executes boot loader
   a. GRUB, LILO, LOADLIN
3. Boot loader executes kernel code
4. Kernel executes /sbin/init
5. Init starts runlevel programs
   a. Coordinates boot process & configures user env
6. Runlevel programs start

Swap Space

- Expansion of system memory, on disk
  - Virtual memory
  - Typically 2x amount of physical memory
    - 16GB of RAM = 32GB swap
- Can only be accessed/view by Linux system
- Slower than physical memory, but important to OS operations
- For forensic purposes
  - Everything in RAM has potential to be stored in swap space at some point
  - Interesting data could be found in swap partition
    - Encryption keys, credentials
    - Sensitive information, emails, etc

Introduction to Digital Forensics - Summary

Key Concepts - Recap

+ Forensics basics
+ Evidence and its collection
+ System forensics

Learning Outcomes Recap

+ Understand the fundamentals of digital forensics
+ Describe different types of evidence collection and information that can be gathered
+ Be able to explain how to properly preserve evidence using a chain of custody
+ Understand components of Windows processes, Registry, and file system
+ Be familiar with Linux processes and file system

Next Steps

- Additional Resources: Skill Dive labs for digital forensics
- Continue the Cisco CyberOps Associate learning path
- Dive deeper into DFIR with additional courses at INE

Thank you!

Introduction to Digital Forensics

+ Aligned with Cisco CyberOps Associate
+ Thank you for your time!

EXPERTS AT MAKING YOU AN EXPERT