Curs 10

Proposed agenda

- **Forensic Basics**
- **Physical and Logical Disk Structures**
- **Windows Artifacts / Evidence of …**
- **Windows Registry**

Forensic Basics

What is Forensic?

Why is it needed?

Forensic science, also known as criminalistics, is the *application of science to criminal and civil laws*, mainly on the criminal side – during criminal investigation, as governed by the legal standards of admissible evidence and criminal procedure. Forensic scientists collect, preserve, and analyze scientific evidence during the course of an investigation.

Digital forensics, also known as computer and network forensics, has many definitions. Generally, it is considered the *application of science to the identification, collection, examination, and analysis of data while preserving the integrity of the information and maintaining a strict chain of custody for the data*.
Data refers to distinct pieces of digital information that have been formatted in a specific way.

Digital forensic techniques can be used for many purposes, such as:

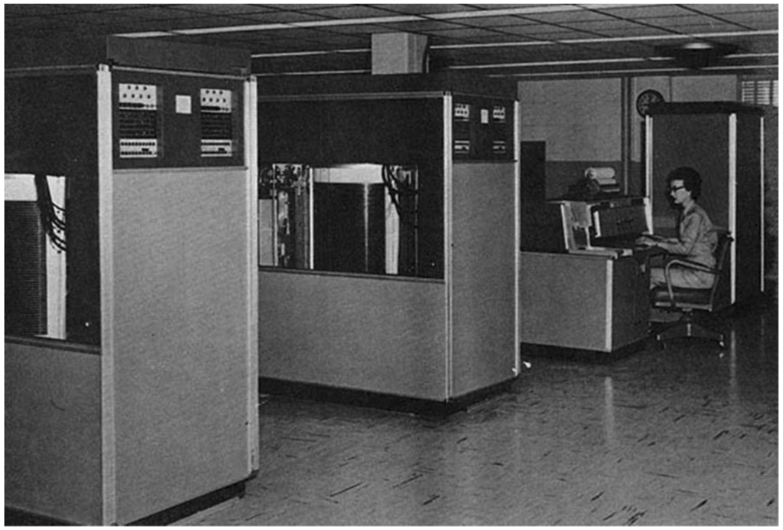
- investigating crimes and internal policy violations,
- reconstructing computer security incidents,
- troubleshooting operational problems,
- recovering from accidental system damage.

Physical and Logical Disk Structures

Back in the Day

The fist **H**ard **D**isk **D**rive (**HDD**) in the world was shipped in 1956.
The drive contained fifty 24-inch platters, stored 5 Mb of data and took more room than two refrigerators.

The price? Just 50.000 USD



Typical technologies

IDE / EIDE **E**nhanced **I**ntegrated **D**rive **E**lectronics – deprecated
SCSI **S**mall **C**omputer **S**ystems **I**nterface – found in servers and systems where high performance and data integrity is critical
SATA **S**erial **A**dvanced **T**echnology **A**rchitecture – the most commonly used type of HDD in personal computers; reached its 3rd revision (SATA III)

Physical and Logical Disk Structures

HDD Internal Characteristics

Drives will have one or more spinning platters made of rigid aluminum or glass coated with various forms of a magnetic substrate. There's an actuator arm with read/write heads attached to it. This arm positions the read-write heads over the correct area of the drive to read or write information.

The drive may need to read from multiple locations in order to launch a program or load a file, which means it may have to wait for the platters to spin and the arm to be moved into the proper position multiple times before it can complete the command.

If a drive is asleep or in a low-power state, it can take several seconds more for the disk to spin up to full power and begin operating.

Physical and Logical Disk Structures

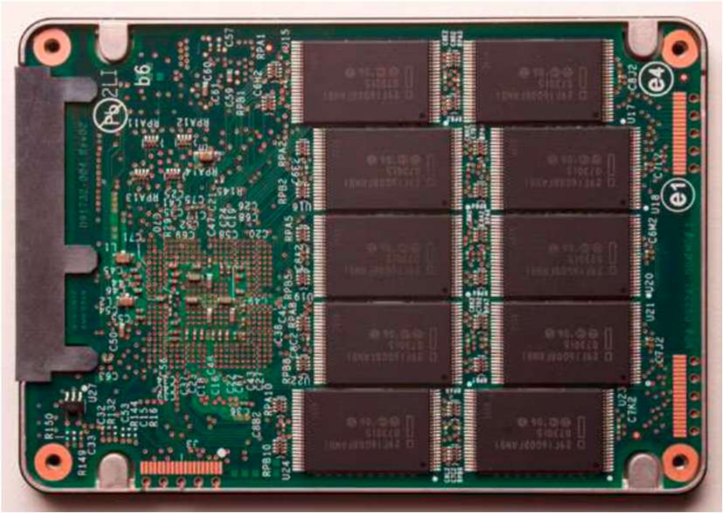
SSD Solid State Drive

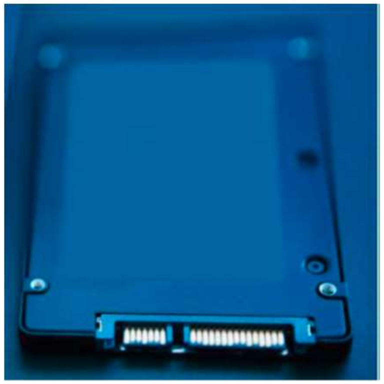
Solid state: electrical term that refers to electronic circuitry that is built entirely out of semiconductors.

In terms of an SSD: it refers to the fact that the primary storage medium is through semiconductors rather than a magnetic media such as a hard drive.

An SSD on the outside looks almost no different than a traditional hard drive. The design is to allow the SSD drive to be put in a notebook or desktop computer in place of a hard drive.

Standard dimension: 1.8, 2.5 or 3.5-inch.
Uses either the ATA or SATA drive interfaces so that there is a compatible interface (other form-factors exists).





Physical and Logical Disk Structures

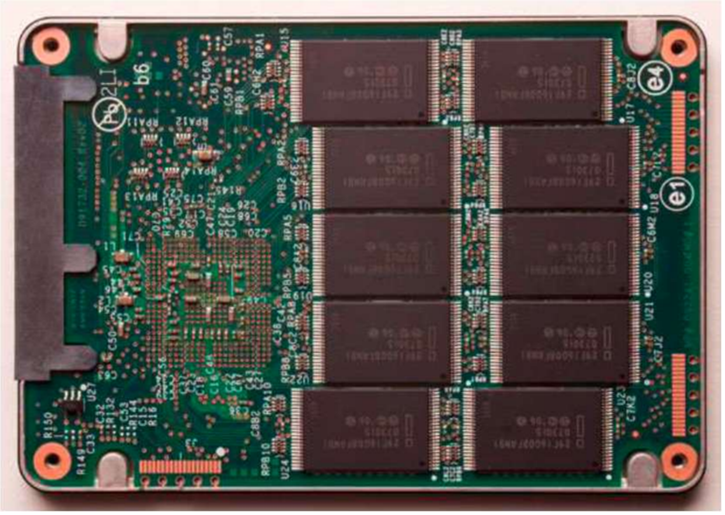
SSD Advantages

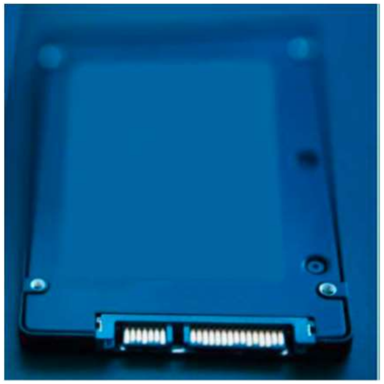
The drive does not have any moving parts.

While a traditional drive has drive motors to spin up the magnetic platters and the drive heads, all the storage on a solid state drive is handled by flash memory chips.

This provide three distinct advantages:

- Less power usage
- Faster data access
- Higher reliability





Physical and Logical Disk Structures

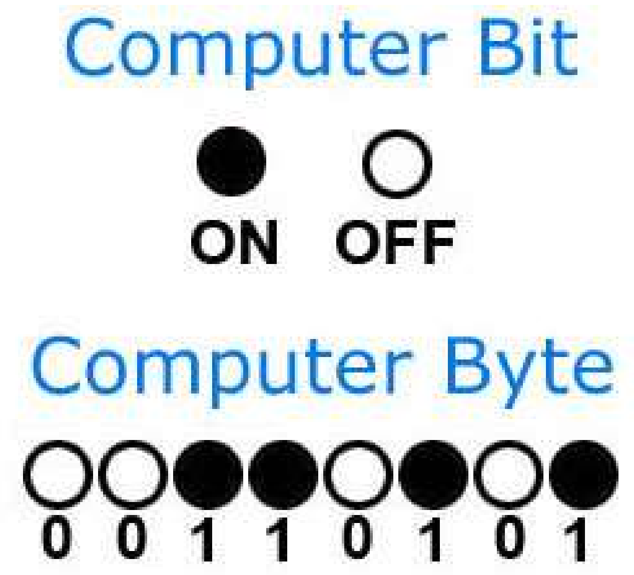
Logical Disk Structure

In every memory storage device, the data is written in the form of a magnetic field or electrical charge that represents an **on** or **off** value, which we know of as a binary digit or bit.

In order to make data storage device useable, some form of order needs to be applied to it, which refers to as the Logical Disk Structure.

For data to be written to a disk, there are three processes that must be undertaken:

- Low-level formatting
- Partitioning
- High-level formatting



Physical and Logical Disk Structures

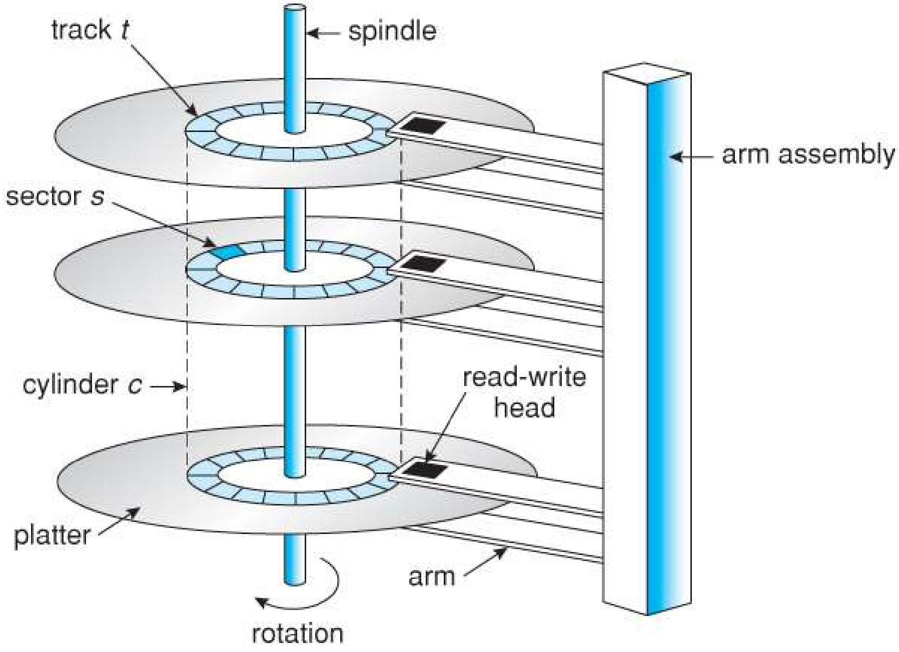
Low-level formatting / Sectors

When data is written in a circular pattern there is a problem identifying *where that data begins* and *where it ends*. To overcome this the track is divided into smaller chunks known as **sectors**.

A sector is the smallest storage unit that is writeable by an HDD; the most common physical sector size is 520 bytes, from which only 512 bytes are used for the actual storage of data and 8 bytes are used for error checking.

The process of creating sectors is called Low-level formatting and can only be done by the HDD manufacturer.

As SSD doesn’t have physical platters and tracks, the sector structure is created also by the manufacturer.



Low-level formatting / Clusters

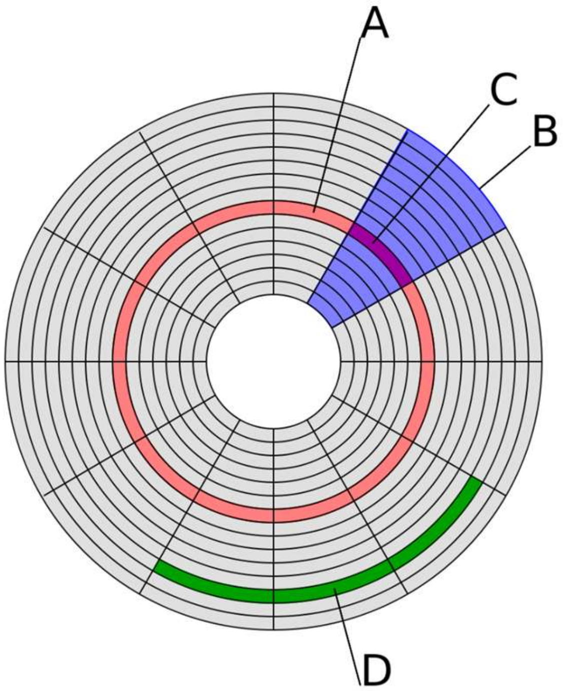
The smallest unit of disk space that can be allocated to a file is called a **cluster** (aka allocation unit).

A cluster consists of one or more consecutive sectors.

Every file must be allocated an integer number of clusters.

If a cluster size is 4096 bytes then a 4000 bytes file will use one cluster or 4096 bytes on the disk.
A 5000 bytes file will use two clusters, or 8192 bytes on the disk.

This “wasted” space is called **slack space**.



A = track
B = sector
C = sector of a track
D = cluster

Physical and Logical Disk Structures

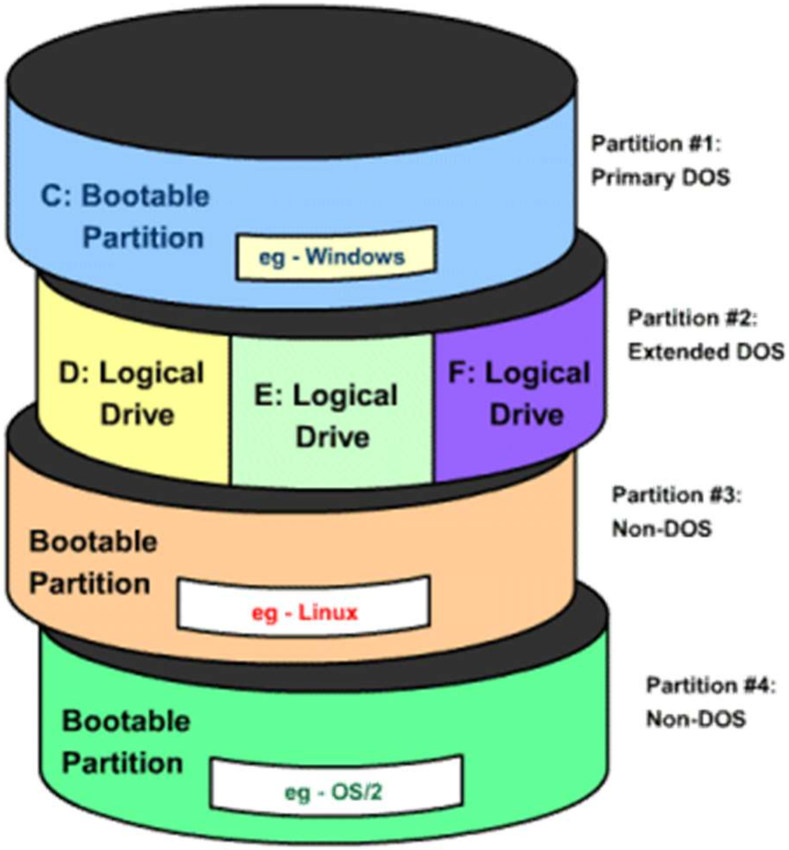
Partitioning

Involves logically dividing the hard disk up into a number of pieces, each piece being called a **partition**.

Partitioning is typically the first step of preparing a newly manufactured disk, *before* any files or directories have been created.

The disk stores the information about partitions locations and sizes in an area known as the partition table that the operating system reads before any other part of the disk.

Each partition then appears in the operating system as a distinct "logical" disk that uses part of the actual physical drive.

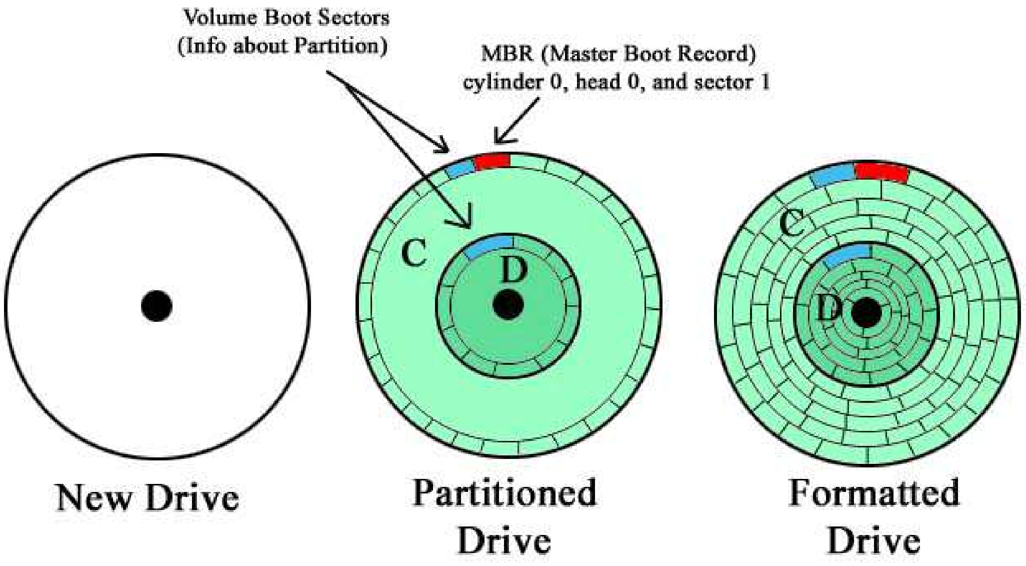


Physical and Logical Disk Structures

High-level formating

Represents the process of writing the file system structures on the disk, such as the master boot record and the file allocation tables, that let the disk be used for storing programs and data.

High-level formatting is done after the hard disk has been partitioned (even if only one partition is to be used).



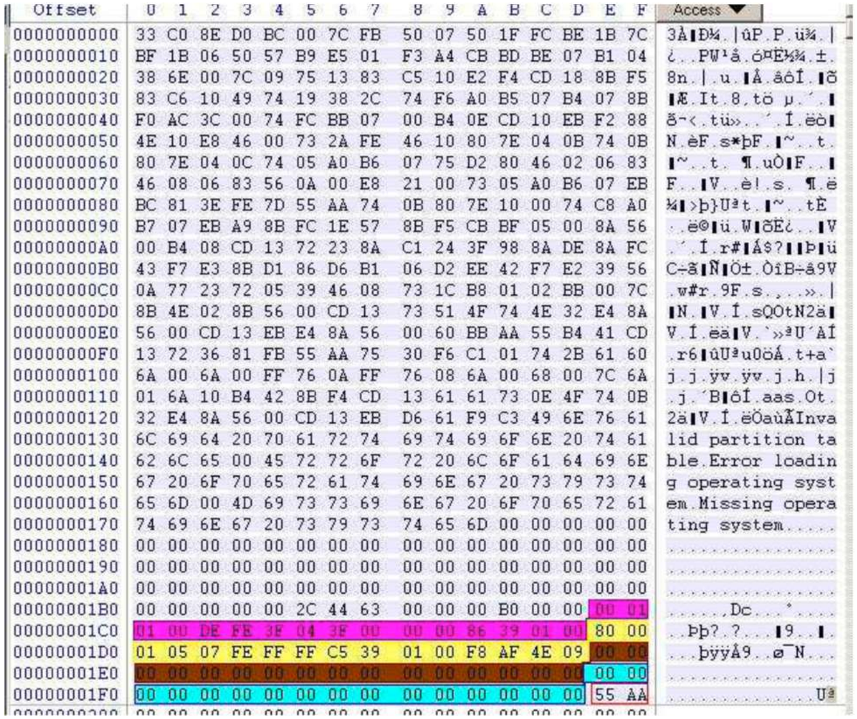
Physical and Logical Disk Structures

Master Boot Record

Begins in the very first physical sector of the hard drive, usually referred to PS0 or Physical Sector 0.
Typically the MBR will only fill one sector or 512 bytes of information; the remaining sectors in the track are reserved.

The first 446 bytes (0 to 445) of information are actual programming code or boot code. This code identifies the drive and instruct the system on the structure of the drive.
The next 64 bytes consist of the Master Partition Table (four 16 byte-entries). The last two bytes of the sector are always hexadecimal 55 AA (0x55AA) (bytes 510 and 511).

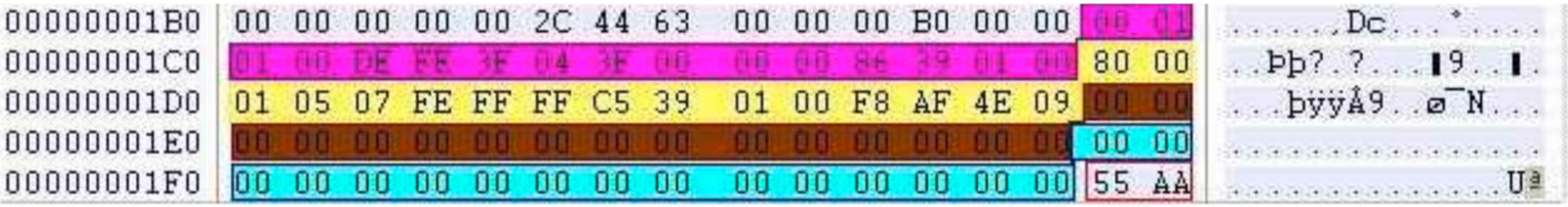
Without an MBR, the computer would stop after the BIOS finished executing its data because it wouldn’t know where to go next to find its instruction about what to load next.



```
  Offset      0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   Access ▼
0000000000   33 C0 8E D0 BC 00 7C FB  50 07 50 1F FC BE 1B 7C  3À▮Ð¼.|ûP.P.ü¾.|
0000000010   BF 1B 06 50 57 B9 E5 01  F3 A4 CB BD BE 07 B1 04  ¿..PW¹å.ó¤Ë½¾.±.
0000000020   38 6E 00 7C 09 75 13 83  C5 10 E2 F4 CD 18 8B F5  8n.|.u.▮Å.âôÍ.▮õ
0000000030   83 C6 10 49 74 19 38 2C  74 F6 A0 B5 07 B4 07 8B  ▮Æ.It.8,tö µ.´.▮
0000000040   F0 AC 3C 00 74 FC BB 07  00 B4 0E CD 10 EB F2 88  ð¬<.tü»..´.Í.ëò▮
0000000050   4E 10 E8 46 00 73 2A FE  46 10 80 7E 04 0B 74 0B  N.èF.s*þF.▮~..t.
0000000060   80 7E 04 0C 74 05 A0 B6  07 75 D2 80 46 02 06 83  ▮~..t. ¶.uÒ▮F..▮
0000000070   46 08 06 83 56 0A 00 E8  21 00 73 05 A0 B6 07 EB  F..▮V..è!.s. ¶.ë
0000000080   BC 81 3E FE 7D 55 AA 74  0B 80 7E 10 00 74 C8 A0  ¼▮>þ}Uªt.▮~..tÈ 
0000000090   B7 07 EB A9 8B FC 1E 57  8B F5 CB BF 05 00 8A 56  ·.ë©▮ü.W▮õË¿..▮V
00000000A0   00 B4 08 CD 13 72 23 8A  C1 24 3F 98 8A DE 8A FC  .´.Í.r#▮Á$?▮▮Þ▮ü
00000000B0   43 F7 E3 8B D1 86 D6 B1  06 D2 EE 42 F7 E2 39 56  C÷ã▮Ñ▮Ö±.ÒîB÷â9V
00000000C0   0A 77 23 72 05 39 46 08  73 1C B8 01 02 BB 00 7C  .w#r.9F.s.¸..».|
00000000D0   8B 4E 02 8B 56 00 CD 13  73 51 4F 74 4E 32 E4 8A  ▮N.▮V.Í.sQOtN2ä▮
00000000E0   56 00 CD 13 EB E4 8A 56  00 60 BB AA 55 B4 41 CD  V.Í.ëä▮V.`»ªU´AÍ
00000000F0   13 72 36 81 FB 55 AA 75  30 F6 C1 01 74 2B 61 60  .r6▮ûUªu0öÁ.t+a`
0000000100   6A 00 6A 00 FF 76 0A FF  76 08 6A 00 68 00 7C 6A  j.j.ÿv.ÿv.j.h.|j
0000000110   01 6A 10 B4 42 8B F4 CD  13 61 61 73 0E 4F 74 0B  .j.´B▮ôÍ.aas.Ot.
0000000120   32 E4 8A 56 00 CD 13 EB  D6 61 F9 C3 49 6E 76 61  2ä▮V.Í.ëÖaùÃInva
0000000130   6C 69 64 20 70 61 72 74  69 74 69 6F 6E 20 74 61  lid partition ta
0000000140   62 6C 65 00 45 72 72 6F  72 20 6C 6F 61 64 69 6E  ble.Error loadin
0000000150   67 20 6F 70 65 72 61 74  69 6E 67 20 73 79 73 74  g operating syst
0000000160   65 6D 00 4D 69 73 73 69  6E 67 20 6F 70 65 72 61  em.Missing opera
0000000170   74 69 6E 67 20 73 79 73  74 65 6D 00 00 00 00 00  ting system.....
0000000180   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0000000190   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000001A0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000001B0   00 00 00 00 00 2C 44 63  00 00 00 B0 00 00 00 01  .....,Dc...°....
00000001C0   01 00 DE FE 3F 04 3F 00  00 00 86 39 01 00 80 00  ..Þþ?.?...▮9..▮.
00000001D0   01 05 07 FE FF FF C5 39  01 00 F8 AF 4E 09 00 00  ...þÿÿÅ9..ø¯N...
00000001E0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000001F0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 AA  ..............Uª
0000000200   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Master Partition Table

The MPT is 64 bytes in length and contains four 16 bytes entries. Each striped line represents a single partition table entry:



```
00000001B0  00 00 00 00 00 2C 44 63  00 00 00 B0 00 00 00 01  .......Dc...*....
00000001C0  01 00 DE FE 3F 04 3F 00  00 00 86 39 01 00 80 00  ..Þþ?.?...▮9..▮.
00000001D0  01 05 07 FE FF FF C5 39  01 00 F8 AF 4E 09 00 00  ...þÿÿÅ9..ø¯N...
00000001E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000001F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 AA  ..............Uª
```

This allows for up to four "**Primary Partitions**" on a drive (a Primary Partition is a partition that can contain the computer boot files). There must be at least one Primary Partition within the MBR and *only primary partitions are* ***bootable*** (can be used to boot the computer to an operating system).

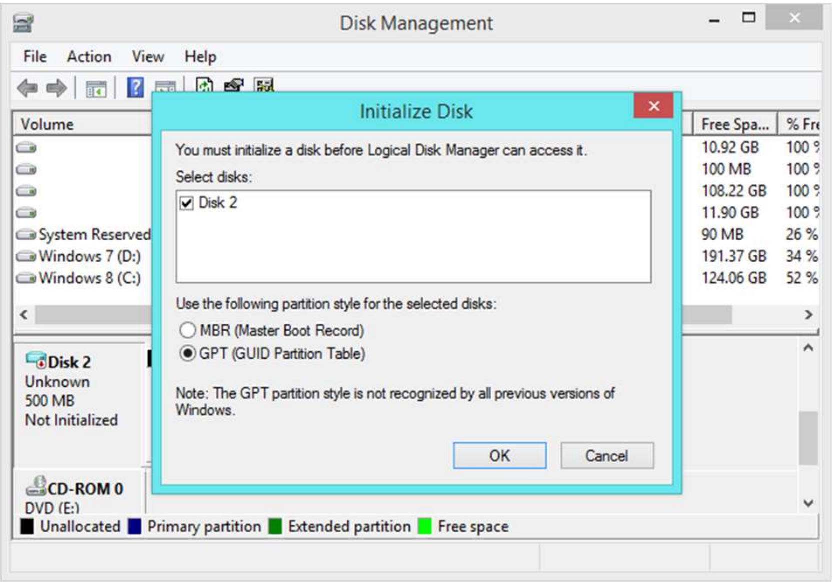
To overcome the limitations of only having four partitions, a different type of partition is allowed, this is known as an "Extended Primary Partition", that can be further split into several smaller portions ("**Logical Partitions**"), thus allowing up to a total of twenty-four partition, each partition assigned an alphabetical drive letter (drive letter A and B are reserved for floppy disk).

Physical and Logical Disk Structures

GPT Partitions

After more than 30 years of supremacy, the BIOS (Basic Input Output System) – has been replaced with a complete re-engineered boot environment, known as UEFI (Unified Extensible Firmware Interface). While the BIOS is fundamentally a solid piece of firmware, UEFI is a programmable software interface that sits on top of a computer's hardware and firmware.

UEFI uses a partition system called the GUID Partition Table (GPT). GUID stands for Globally Unique Identifier (a 128-bit number used to identify information in computer systems).



Volume	Free Spa...	% Fre
	10.92 GB	100 %
	100 MB	100 %
	108.22 GB	100 %
	11.90 GB	100 %
System Reserved	90 MB	26 %
Windows 7 (D:)	191.37 GB	34 %
Windows 8 (C:)	124.06 GB	52 %

Disk 2
Unknown
500 MB
Not Initialized

CD-ROM 0
DVD (E:)

■ Unallocated ■ Primary partition ■ Extended partition ■ Free space

Initialize Disk

You must initialize a disk before Logical Disk Manager can access it.

Select disks:

☑ Disk 2

Use the following partition style for the selected disks:

○ MBR (Master Boot Record)
● GPT (GUID Partition Table)

Note: The GPT partition style is not recognized by all previous versions of Windows.

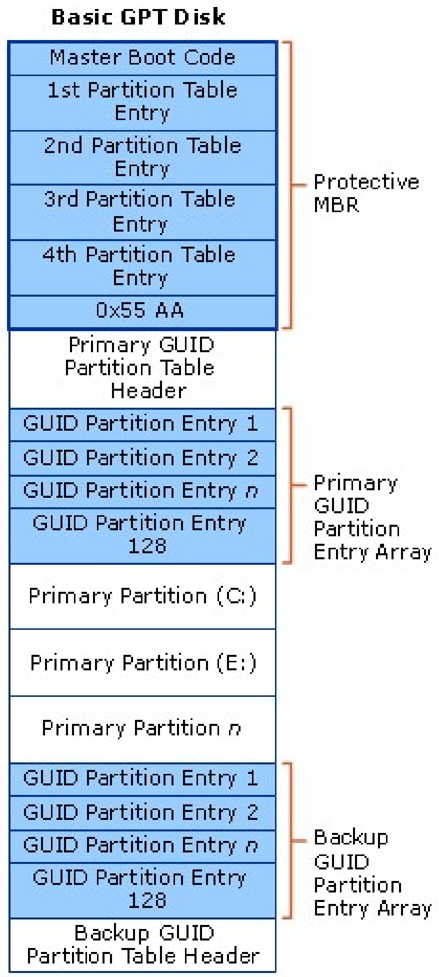
The GPT Partition table can support up to 128 partitions and uses 64-bit LBA addresses (LBA = Logical Block Addressing where every sector in the drive is given a linear address – the first sector is Sector 0 and they are numbered sequentially to the end of the drive).
The GPT system has the ability to support very large hard drives, which are becoming increasingly available and cheap to purchase.

Physical and Logical Disk Structures

GPT Partitions

There are five major parts to a GPT partitioned disk:

- The Protective MBR
- Primary GUID partition table header
- GUID partition entries
- Partition area
- Backup Area



Basic GPT Disk

Master Boot Code
1st Partition Table Entry
2nd Partition Table Entry
3rd Partition Table Entry
4th Partition Table Entry
0x55 AA
Protective MBR

Primary GUID Partition Table Header

GUID Partition Entry 1
GUID Partition Entry 2
GUID Partition Entry *n*
GUID Partition Entry 128
Primary GUID Partition Entry Array

Primary Partition (C:)
Primary Partition (E:)
Primary Partition *n*

GUID Partition Entry 1
GUID Partition Entry 2
GUID Partition Entry *n*
GUID Partition Entry 128
Backup GUID Partition Entry Array

Backup GUID Partition Table Header

GPT Partitions

Protective MBR is located at the very beginning of the disk (Physical Sector 0) and like the previous system it is usually 512 bytes in length. There will be no Boot Code present in the Protective MBR sector. This structure is for backward compatibility with disk management utilities that operate on MBR.

The **primary GUID Partition Table Header** will always immediately follow the MBR in Physical Sector 1. The GPT header always begins with the 8-byte EDI signature string: 0x45 0x46 0x49 0x20 0x50 0x41 0x52 0x54 (ASCII: “EFI PART”). Using the information contained in the GPT header it is possible to determine the layout of the disk. This includes the location of the partition table, partition data areas and backup copies.

The **partition tables** are usually located in Physical Sector 2. Each partition entry is 128 bytes in length and the entry provides much information about the partition.

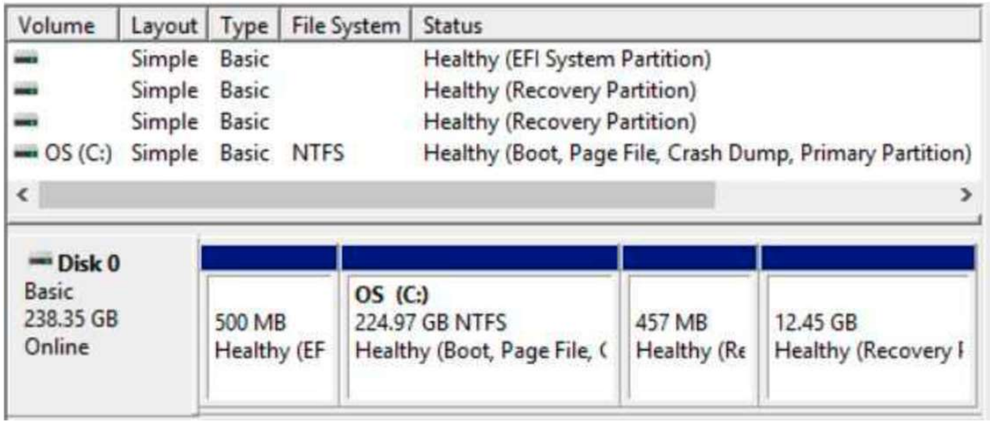
GPT Backup – one of the advantages of using a GUID partition structure is the additional resilience it provides. Located at the end of the disk is an entire backup of the Primary partition entries and the GPT Header.

NTFS (New Technology File System)

NTFS was developed in 1990's as a replacement for FAT and it has undergone four upgrades, most recent version being v3.1

The main features and advantages over FAT:

- Support for mixed case filenames;
- Support for long filenames (255 characters as opposed to FAT's 8+3 char.);
- Less fragmentation of data;
- Transaction journaling for crash recovery;
- Support for compression, encryption and quota enforcement.



Volume	Layout	Type	File System	Status
	Simple	Basic		Healthy (EFI System Partition)
	Simple	Basic		Healthy (Recovery Partition)
	Simple	Basic		Healthy (Recovery Partition)
OS (C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)

Disk 0 Basic 238.35 GB Online	500 MB Healthy (EF	OS (C:) 224.97 GB NTFS Healthy (Boot, Page File, (	457 MB Healthy (Re	12.45 GB Healthy (Recovery I

NTFS

In NTFS, all the data is stored as files (including system data).

Basically there are **regular files** (containing user data) and **“special” files** (containing system data).
These are referred to as **Metadata files** (metadata = “data about data”).
These special files are hidden to users on a live NTFS volume and are not accessible through the file system itself.

The most important metadata file in an NTFS volume is the Master File Table (**$MFT**).
NTFS uses the $MFT to track and store all information about every file within the volume – including itself.

Additional metadata files are used to track storage space allocation, security issues, accessibility permissions, journaling and encryption.

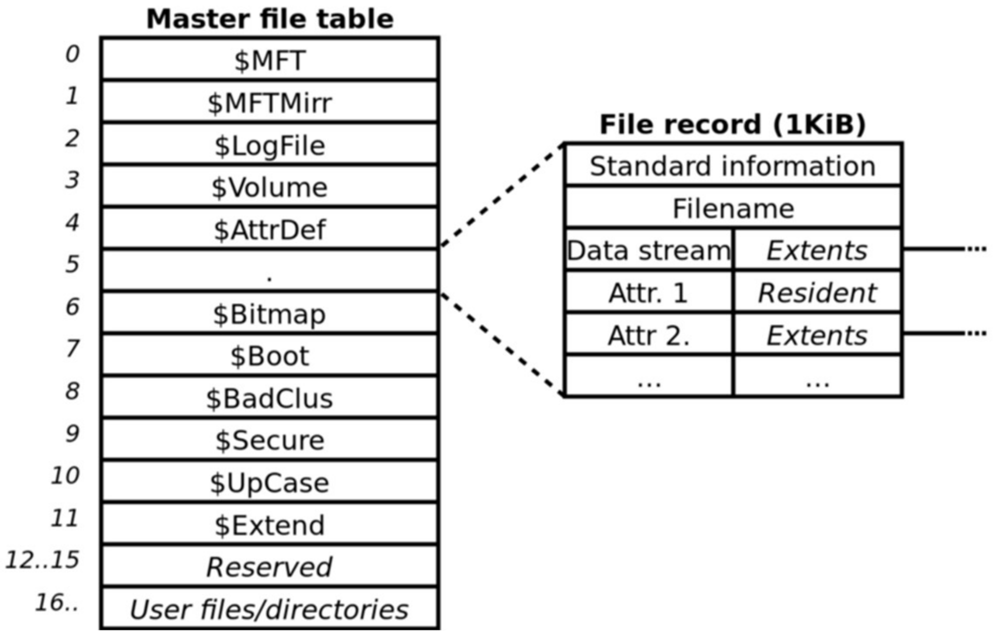
Physical and Logical Disk Structures

NTFS / MFT

NTFS uses Master File Table to track every file within the volume. It does this by having (at least) one entry for each file, called a **File Record**, which is given a unique number. The MFT is similar to a relational database table, containing various attributes about different files.

The first 16 entries in the MFT are metadata files, the first record listed being for the $MFT itself, which describes its location and size on the NTFS volume. Therefore, the $MFT needs to be processed in order to know its own size and location on the disk.

All files on an NTFS volume, including the root directory, have a record entry in the $MFT describing their size and location in the same manner.



Master file table

0	$MFT
1	$MFTMirr
2	$LogFile
3	$Volume
4	$AttrDef
5	.
6	$Bitmap
7	$Boot
8	$BadClus
9	$Secure
10	$UpCase
11	$Extend
12..15	*Reserved*
16..	*User files/directories*

File record (1KiB)

Standard information	
Filename	
Data stream	*Extents*
Attr. 1	*Resident*
Attr 2.	*Extents*
...	...

NTFS / File Records

The MFT lists each file in individual entries called **File Records**. Every file record has its own record number, which is used as an identification number for the file that the file records refers to.
As files are added to the volume, a record for each file is added in sequential order within the $MFT.

If a file is deleted, the file record that was used to track that file becomes unused; the $MFT will reuse "deleted" records before creating new records.
As a result, "deleted" MFT records can be over-written very quickly.

A file record is divided into separate blocks of data that contain information about the file record itself and the file or directory to which that file record points to. Information about the file or folder is stored in discrete blocks called "**Attributes**". Each attribute stores a certain type of information about the file.

The signature (beginning) of a file record is "**FILE**" while the end is marked with **0xFF FF FF FF**

Physical and Logical Disk Structures

NTFS / Attributes

Attributes come in different types and contain different information about the file, or the file content itself. Attributes have their own data structure, which comprises of headers and content.
The attribute headers and content are all discrete blocks of data and must be interpreted individually to determine what information the attribute holds about the file.

Attributes that are *mandatory* for all files and directories are:

- **$Standard_Information**
- **$File_Name**

If a file name is longer than eight characters or contains special characters, Windows will also create a DOS-compatible 8.3 name and save this as a second $File_Name attribute.

Files will also have a **$Data** attribute to hold their content. This attribute can be either *resident* (completely within the MFT Record – for files less than 700 bytes) or *nonresident* (header in MFT will point to the cluster the content is in).

Physical and Logical Disk Structures

NTFS / Attributes

Attribute Identifier	Attribute Name	Description
10 00 00 00	**$Standard_Information**	**Contains File permissions, time stamps, security and administrative information.**
20 00 00 00	$Attribute List	Location of all attributes that do not fit in a single file record entry.
30 00 00 00	**$File_Name**	**The name of the file.**
40 00 00 00	$Object_ID	Contains a Globally Unique Identifier for the file.
50 00 00 00	$Security_Descriptor	Access control and security properties of the file.
60 00 00 00	$Volume_Name	This contains the volume label.
70 00 00 00	$Volume_Information	This attribute contains the NTFS version information.
80 00 00 00	**$Data**	**The actual file 's data.**
90 00 00 00	**$Index_Root**	**List of directory's child files**
A0 00 00 00	$Index_Allocation	Points to the location the Index Buffers of a large directory
B0 00 00 00	$Bitmap	Tracks the allocation status
C0 00 00 00	$Symbolic_Link	Soft link information
D0 00 00 00	$Reparse_Point	Similar to a soft link (legacy)
E0 00 00 00	$EA_Information	Allows compatibility with HPFS (legacy)
00 01 00 00	$Logged_Utility_Stream	Information and keys for encryption attributes

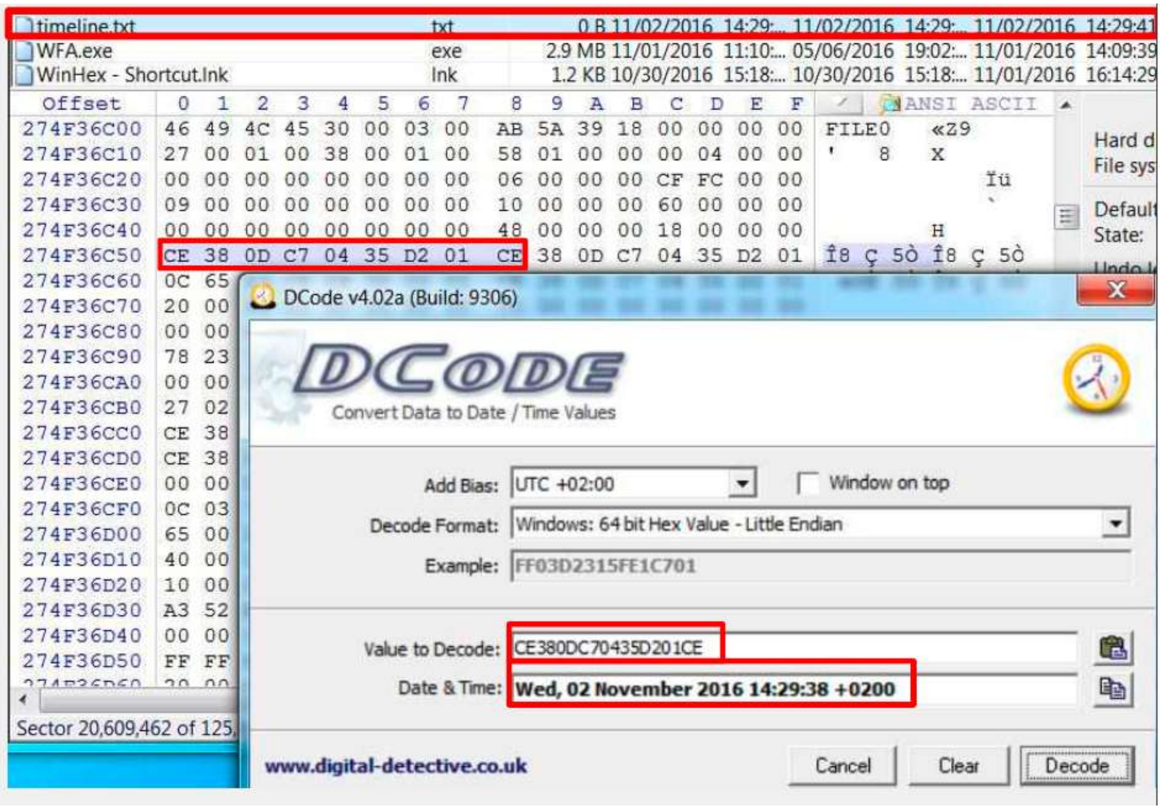
Physical and Logical Disk Structures

NTFS / Date and Time

Dates and Times are stored in $Standard_Information attribute, in what is called a FILETIME structure.

FILETIME is a 64-bit value representing the number of 100 - nanosecond intervals since January 1, 1601 in Coordinated Universal Time (UTC).

All file dates and times are written to the NTFS file system in UTC



```
timeline.txt              txt     0 B 11/02/2016 14:29:... 11/02/2016 14:29:... 11/02/2016 14:29:41
WFA.exe                   exe  2.9 MB 11/01/2016 11:10:... 05/06/2016 19:02:... 11/01/2016 14:09:39
WinHex - Shortcut.lnk     lnk  1.2 KB 10/30/2016 15:18:... 10/30/2016 15:18:... 11/01/2016 16:14:29

 Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   ANSI ASCII
274F36C00  46 49 4C 45 30 00 03 00  AB 5A 39 18 00 00 00 00  FILE0   «Z9
274F36C10  27 00 01 00 38 00 01 00  58 01 00 00 00 04 00 00  '   8   X
274F36C20  00 00 00 00 00 00 00 00  06 00 00 00 CF FC 00 00              Ïü
274F36C30  09 00 00 00 00 00 00 00  10 00 00 00 60 00 00 00              `
274F36C40  00 00 00 00 00 00 00 00  48 00 00 00 18 00 00 00          H
274F36C50  CE 38 0D C7 04 35 D2 01  CE 38 0D C7 04 35 D2 01  Î8 Ç 5Ò Î8 Ç 5Ò
274F36C60  0C 65
274F36C70  20 00
274F36C80  00 00
274F36C90  78 23
274F36CA0  00 00
274F36CB0  27 02
274F36CC0  CE 38
274F36CD0  CE 38
274F36CE0  00 00
274F36CF0  0C 03
274F36D00  65 00
274F36D10  40 00
274F36D20  10 00
274F36D30  A3 52
274F36D40  00 00
274F36D50  FF FF
274F36D60  20 00

Sector 20,609,462 of 125,
```

DCode v4.02a (Build: 9306)

DCODE
Convert Data to Date / Time Values

Add Bias: UTC +02:00 ☐ Window on top

Decode Format: Windows: 64 bit Hex Value - Little Endian

Example: FF03D2315FE1C701

Value to Decode: CE380DC70435D201CE

Date & Time: Wed, 02 November 2016 14:29:38 +0200

www.digital-detective.co.uk

Windows Artifacts / Evidence of …

- File Download
- Program Execution
- File / Folder Opening
- Physical Location
- Account Usage
- External Device / USB Usage
- Browser Usage

The categories on the left map specific Windows artifacts to the analysis questions that the forensic investigation will help to answer, be it malware infection, intellectual property theft or other cyber crime investigations.

These artifacts will help in painting a clear picture of **which** user was involved, **what** the user was doing, **when** the user was doing it, **why** and **how** the nefarious activities were performed.

Windows Artifacts

File / Folder Opening

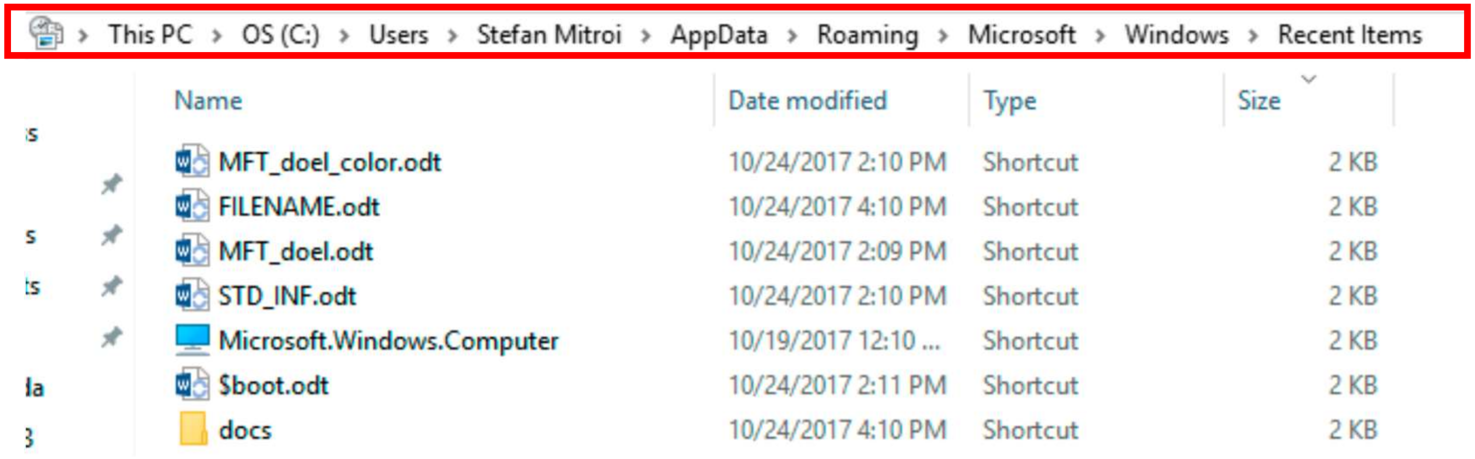
Shell Link Files

A **shell link file** is commonly referred to as a *link file* or *shortcut*. It is a special file that contains “links” or ”pointers” to other resources, for example, programs, data files and folders.

By default, when a file or document is opened in Windows, a link (.lnk) file is created in the Recent folder.

The best thing about a Link file is that it will often demonstrate a ***user’s knowledge of a file and their interaction with that file***.

In Windows 7 – 10 these files are displayed in “Recent Items”, which is a virtual folder when viewed with Windows Explorer.



This PC › OS (C:) › Users › Stefan Mitroi › AppData › Roaming › Microsoft › Windows › Recent Items

Name	Date modified	Type	Size
MFT_doel_color.odt	10/24/2017 2:10 PM	Shortcut	2 KB
FILENAME.odt	10/24/2017 4:10 PM	Shortcut	2 KB
MFT_doel.odt	10/24/2017 2:09 PM	Shortcut	2 KB
STD_INF.odt	10/24/2017 2:10 PM	Shortcut	2 KB
Microsoft.Windows.Computer	10/19/2017 12:10 ...	Shortcut	2 KB
$boot.odt	10/24/2017 2:11 PM	Shortcut	2 KB
docs	10/24/2017 4:10 PM	Shortcut	2 KB

Windows Artifacts

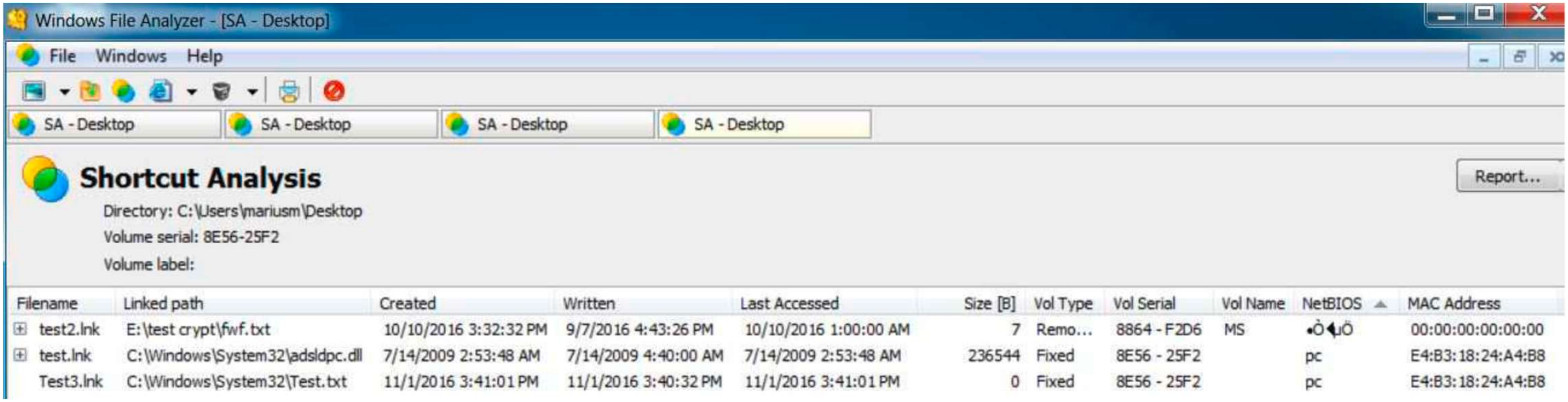
File / Folder Opening

Shell Link Files

These files contain some very useful information about the target file including:

File Attributes	MAC Times	File Size	Volume Type
Volume Serial Number	Volume Label	Original File Path	

A useful and free tool for parsing .lnk files is Windows File Analyzer (http://mitec.cz/wfa.html)



Shortcut Analysis

Directory: C:\Users\mariusm\Desktop

Volume serial: 8E56-25F2

Volume label:

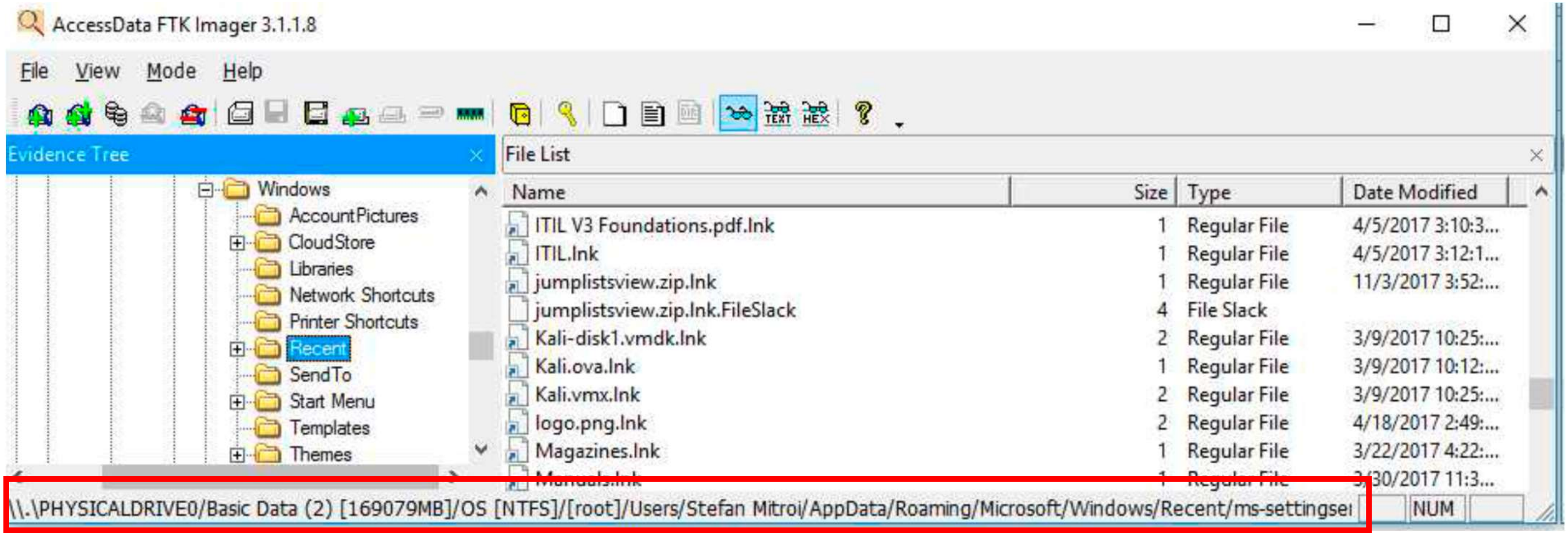
Filename	Linked path	Created	Written	Last Accessed	Size [B]	Vol Type	Vol Serial	Vol Name	NetBIOS	MAC Address
⊞ test2.lnk	E:\test crypt\fwf.txt	10/10/2016 3:32:32 PM	9/7/2016 4:43:26 PM	10/10/2016 1:00:00 AM	7	Remo...	8864 - F2D6	MS	•Ò◂µÖ	00:00:00:00:00:00
⊞ test.lnk	C:\Windows\System32\adsldpc.dll	7/14/2009 2:53:48 AM	7/14/2009 4:40:00 AM	7/14/2009 2:53:48 AM	236544	Fixed	8E56 - 25F2		pc	E4:B3:18:24:A4:B8
Test3.lnk	C:\Windows\System32\Test.txt	11/1/2016 3:41:01 PM	11/1/2016 3:40:32 PM	11/1/2016 3:41:01 PM	0	Fixed	8E56 - 25F2		pc	E4:B3:18:24:A4:B8

Windows Artifacts **File / Folder Opening**

Shell Link Files

There are many forensics implications relating to the content of these files.

The Volume Serial Number can be used to tie a specific thumb drive, USB drive, memory card or other removable media to a specific computer system.



Name	Size	Type	Date Modified
ITIL V3 Foundations.pdf.lnk	1	Regular File	4/5/2017 3:10:3...
ITIL.lnk	1	Regular File	4/5/2017 3:12:1...
jumplistsview.zip.lnk	1	Regular File	11/3/2017 3:52:...
jumplistsview.zip.lnk.FileSlack	4	File Slack	
Kali-disk1.vmdk.lnk	2	Regular File	3/9/2017 10:25:...
Kali.ova.lnk	1	Regular File	3/9/2017 10:12:...
Kali.vmx.lnk	2	Regular File	3/9/2017 10:25:...
logo.png.lnk	2	Regular File	4/18/2017 2:49:...
Magazines.lnk	1	Regular File	3/22/2017 4:22:...
Manuals.lnk	1	Regular File	3/30/2017 11:3...

\\.\PHYSICALDRIVE0/Basic Data (2) [169079MB]/OS [NTFS]/[root]/Users/Stefan Mitroi/AppData/Roaming/Microsoft/Windows/Recent/ms-settingsei

File / Folder Opening

Jump Lists

Windows 7 introduced a new feature called "**Jump Lists**", which are essentially a list of every file that has been opened (or attempted to open) by a particular application. It's similar to the "Recent" folder, except that each list only applies to one program. The Jump List feature provides the user with a graphical interface associated with each installed application which lists files that have been previously accessed by that application.

This artifact often provides significant insight to user activity and is especially beneficial if entries in the Recent folder have been deleted, or even if the application has been deleted or uninstalled.

Clearing the items in the Recent folder does not eliminate the Jump List data unless the user first reveals the hidden folders containing the Jump Lists data and manually delete them, which is not easy as they are "Super Hidden".

There are two main types of Jump Lists:

- **Automatic** – this Jump List is automatically populated by the system
- **Custom** – this Jump List is maintained by the individual application

Windows Artifacts

File / Folder Opening

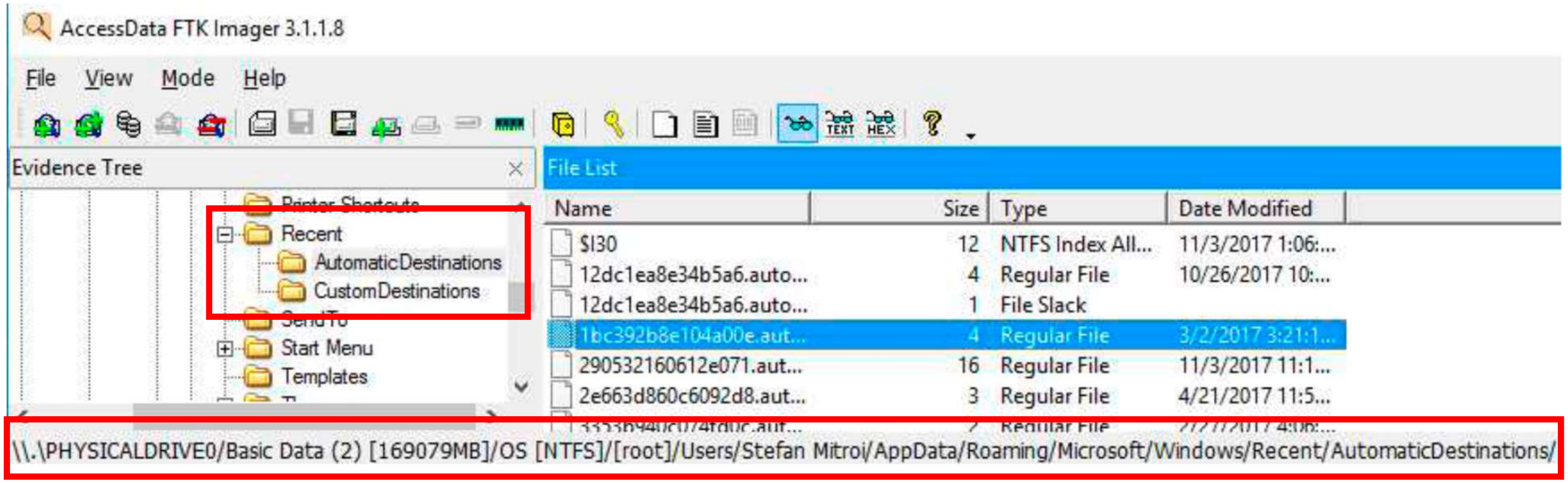
Jump Lists

Jump List data for all applications is stored in the users profile path:

%UserProfile%\AppData\Roaming\Microsoft\Windows\Recent

When this folder is viewed with a forensic tool, additional folders appear:

%UserProfile%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations
%UserProfile%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations



Name	Size	Type	Date Modified
$I30	12	NTFS Index All...	11/3/2017 1:06:...
12dc1ea8e34b5a6.auto...	4	Regular File	10/26/2017 10:...
12dc1ea8e34b5a6.auto...	1	File Slack	
1bc392b8e104a00e.aut...	4	Regular File	3/2/2017 3:21:1...
290532160612e071.aut...	16	Regular File	11/3/2017 11:1...
2e663d860c6092d8.aut...	3	Regular File	4/21/2017 11:5...
3333b94dc074fd0c.aut...	2	Regular File	2/27/2017 4:06:...

\\.\PHYSICALDRIVE0/Basic Data (2) [169079MB]/OS [NTFS]/[root]/Users/Stefan Mitroi/AppData/Roaming/Microsoft/Windows/Recent/AutomaticDestinations/

Windows Artifacts

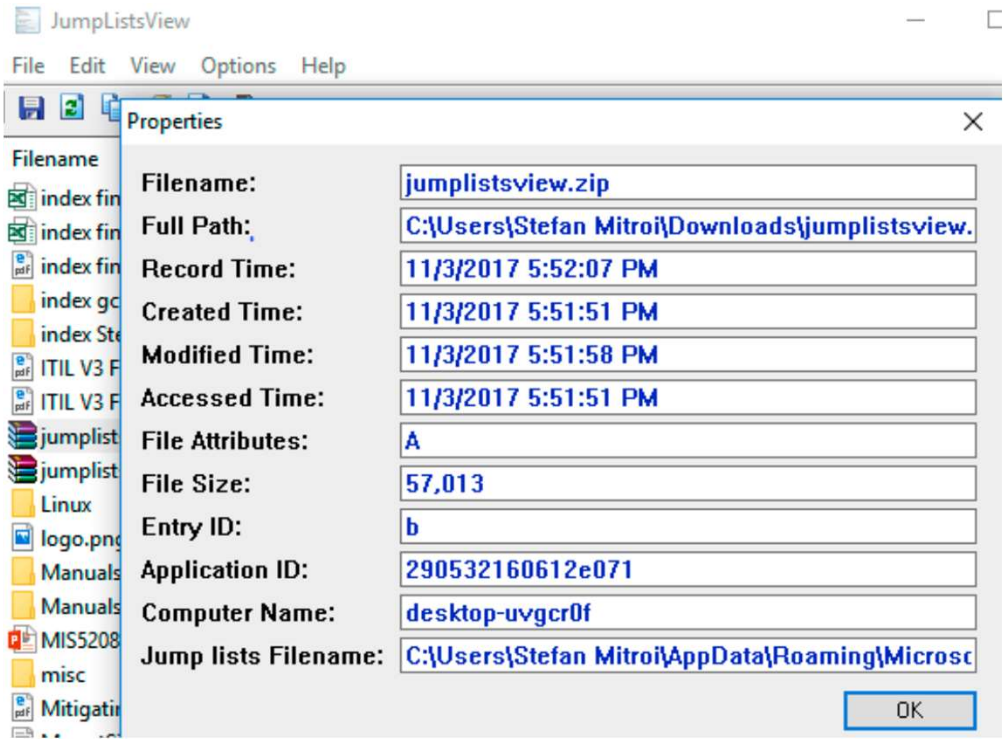
File / Folder Opening

Jump Lists

The recent item data from the Jump Lists populate those two folders. Each program will have its own file name, referred to as a ”Jump List ID”. By examining each file with a text editor, it can be determined which file links correspond to which program’s Jump List entry.
More jump list IDs can be found at:
http://www.forensicswiki.org/wiki/List_of_Jump_List_IDs

A useful tool is available for examining “lists”, called JumpListView, is available at:
http://www.nirsoft.net/utils/jump_lists_view.html

JumpListView will display a list of all items in the Jump List, their path, date and time and the entry number for each item.



Filename: jumplistsview.zip
Full Path: C:\Users\Stefan Mitroi\Downloads\jumplistsview.
Record Time: 11/3/2017 5:52:07 PM
Created Time: 11/3/2017 5:51:51 PM
Modified Time: 11/3/2017 5:51:58 PM
Accessed Time: 11/3/2017 5:51:51 PM
File Attributes: A
File Size: 57,013
Entry ID: b
Application ID: 290532160612e071
Computer Name: desktop-uvgcr0f
Jump lists Filename: C:\Users\Stefan Mitroi\AppData\Roaming\Microsc

Windows Artifacts

Account Usage

Event Logs

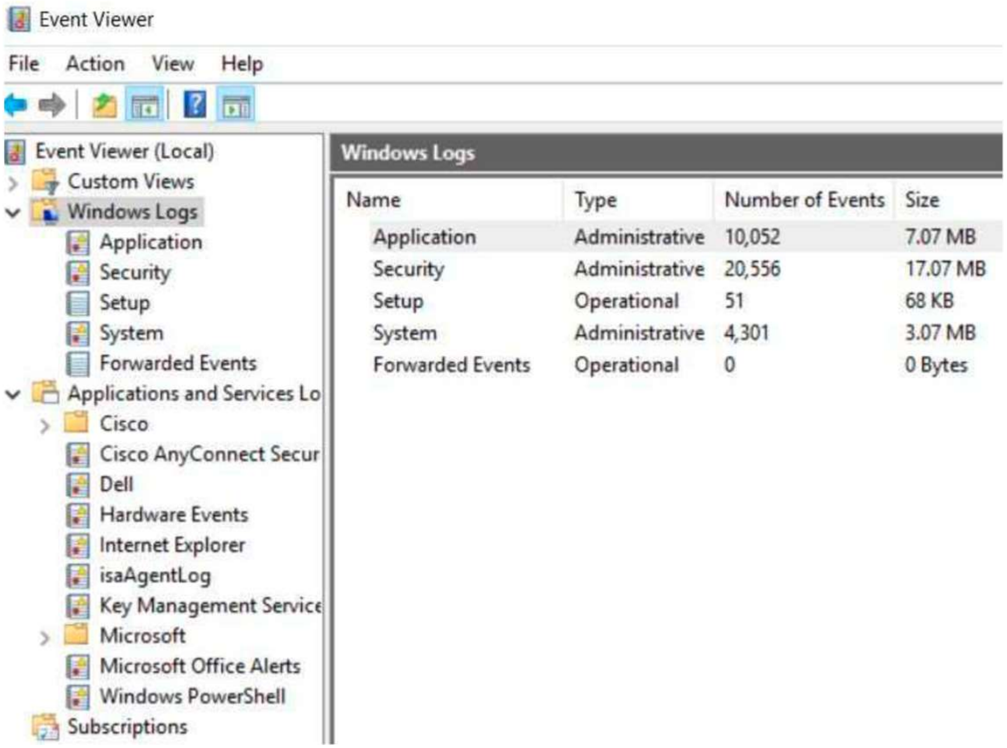
Microsoft defines an “event” as any occurrence that is potentially noteworthy either to the user, the operating system or to an application.

The logs are stored in C:\Windows\system32\winevt\logs

Event logs have the extension .evtx and utilize the .xml format

Events are categorized into 2 main classes:

- Windows Logs
- Application and Services Logs



Name	Type	Number of Events	Size
Application	Administrative	10,052	7.07 MB
Security	Administrative	20,556	17.07 MB
Setup	Operational	51	68 KB
System	Administrative	4,301	3.07 MB
Forwarded Events	Operational	0	0 Bytes

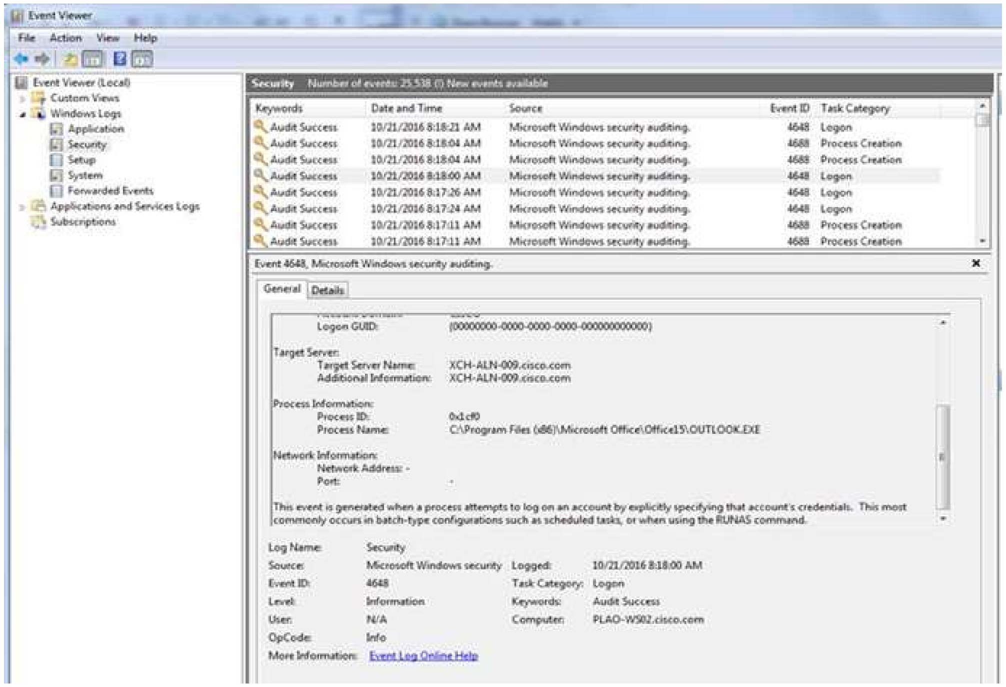
Windows Artifacts

Account Usage

Event Logs

Windows Logs consists of:

- **Security Logs** – provide details on a variety of actions like user authentication (logons, “Run as” commands, remote access etc.) and what a particular user did on a system after authentication (like Files / Folders / Share access).
- **System Logs** – contains events related to Windows services, system components, drivers, resources (ex. Service stopped; System Rebooted).
- **Application Logs** – contains software events unrelated to operating system (ex. MS Office alert or informational messages).



Security Number of events: 25,538 (!) New events available

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	10/21/2016 8:18:21 AM	Microsoft Windows security auditing.	4648	Logon
Audit Success	10/21/2016 8:18:04 AM	Microsoft Windows security auditing.	4688	Process Creation
Audit Success	10/21/2016 8:18:04 AM	Microsoft Windows security auditing.	4688	Process Creation
Audit Success	10/21/2016 8:18:00 AM	Microsoft Windows security auditing.	4648	Logon
Audit Success	10/21/2016 8:17:26 AM	Microsoft Windows security auditing.	4648	Logon
Audit Success	10/21/2016 8:17:24 AM	Microsoft Windows security auditing.	4648	Logon
Audit Success	10/21/2016 8:17:11 AM	Microsoft Windows security auditing.	4688	Process Creation
Audit Success	10/21/2016 8:17:11 AM	Microsoft Windows security auditing.	4688	Process Creation

Event 4648, Microsoft Windows security auditing.

```
        Logon GUID:                 {00000000-0000-0000-0000-000000000000}

Target Server:
        Target Server Name:         XCH-ALN-009.cisco.com
        Additional Information:     XCH-ALN-009.cisco.com

Process Information:
        Process ID:                 0x1cf0
        Process Name:               C:\Program Files (x86)\Microsoft Office\Office15\OUTLOOK.EXE

Network Information:
        Network Address: -
        Port:                       -

This event is generated when a process attempts to log on an account by explicitly specifying that account's credentials.  This most
commonly occurs in batch-type configurations such as scheduled tasks, or when using the RUNAS command.
```

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	10/21/2016 8:18:00 AM
Event ID:	4648	Task Category:	Logon
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	PLAO-WS02.cisco.com
OpCode:	Info		
More Information:	Event Log Online Help		

Windows Artifacts

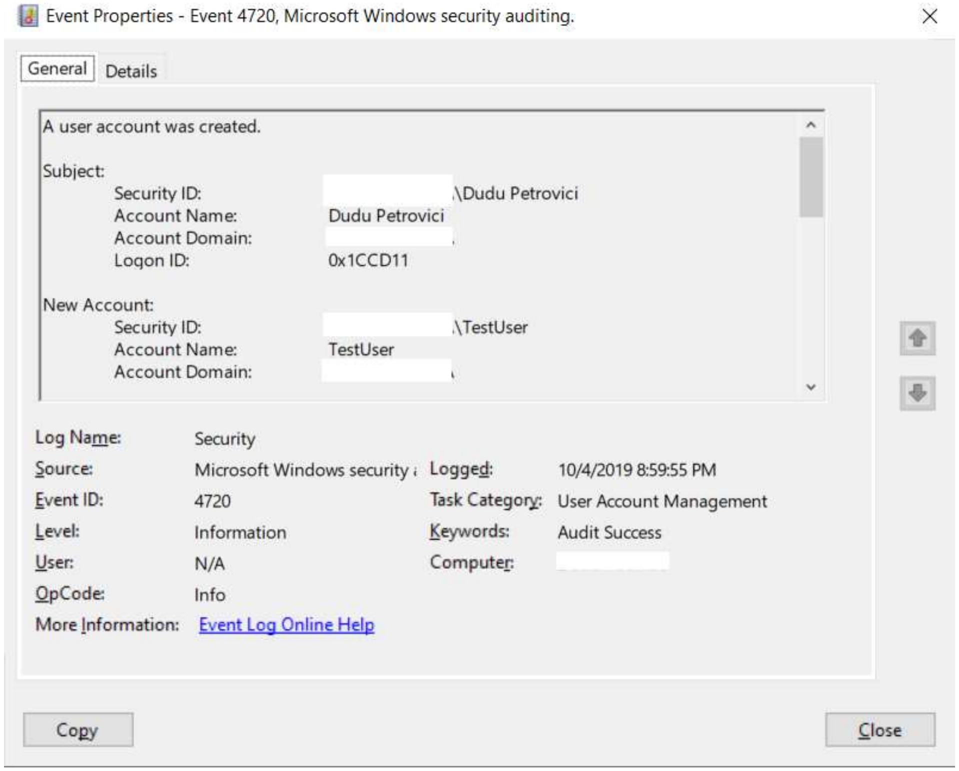
Account Usage

Event Logs

Event logs can be useful in a forensics examination to show that a user may or may not have performed a particular action at a specific time.

They can be useful for several things, such as:

- Tracing logs in the case of logging into a restricted network, proving the computer was running during a particular time;
- Proving the computer was running during a particular time;
- Showing time change events;
- USB driver installation;
- Wireless connections;
- Identify rogue accounts created by threat actors;



Event Properties - Event 4720, Microsoft Windows security auditing.

```
A user account was created.

Subject:
        Security ID:        [redacted]\Dudu Petrovici
        Account Name:       Dudu Petrovici
        Account Domain:     [redacted]
        Logon ID:           0x1CCD11

New Account:
        Security ID:        [redacted]\TestUser
        Account Name:       TestUser
        Account Domain:     [redacted]
```

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	10/4/2019 8:59:55 PM
Event ID:	4720	Task Category:	User Account Management
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	[redacted]
OpCode:	Info		
More Information:	Event Log Online Help		

Windows Artifacts

Account Usage

Event Logs

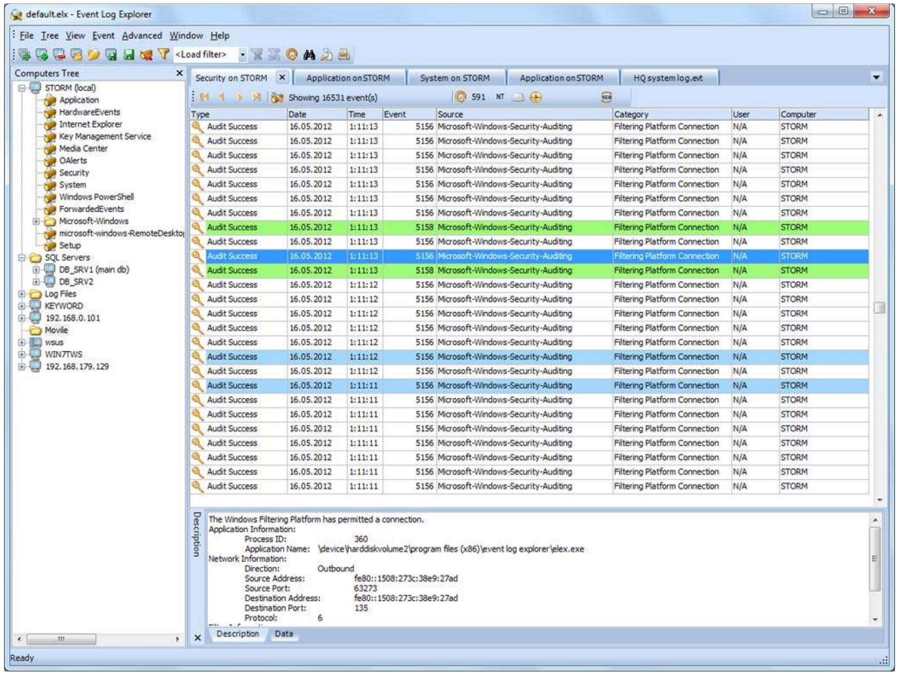
Most common Windows Event IDs of interest:

- 4626 An account successfully logged on
- 4625 Account failed to log in
- 4634 / 4647 Successful logoff
- 4648 Logon using explicit credentials (Runas)
- 4672 Special privileges assigned to new logon (Administrator)
- 4720 An account was created

A detailed description for a wide range of Event IDs can be found at

https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/

A useful tool for aggregating, searching and correlating various event logs is Event Log Explorer (https://eventlogxp.com)



Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	16.05.2012	1:11:13	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5158	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:13	5158	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:12	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:12	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:12	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:12	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:12	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:12	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:12	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:11	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:11	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:11	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:11	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:11	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:11	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:11	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM
Audit Success	16.05.2012	1:11:11	5156	Microsoft-Windows-Security-Auditing	Filtering Platform Connection	N/A	STORM

```
The Windows Filtering Platform has permitted a connection.
Application Information:
        Process ID:             360
        Application Name:   \device\harddiskvolume2\program files (x86)\event log explorer\elex.exe
Network Information:
        Direction:          Outbound
        Source Address:             fe80::1508:273c:38e9:27ad
        Source Port:                63273
        Destination Address:        fe80::1508:273c:38e9:27ad
        Destination Port:           135
        Protocol:           6
```

Windows Artifacts

Physical Location

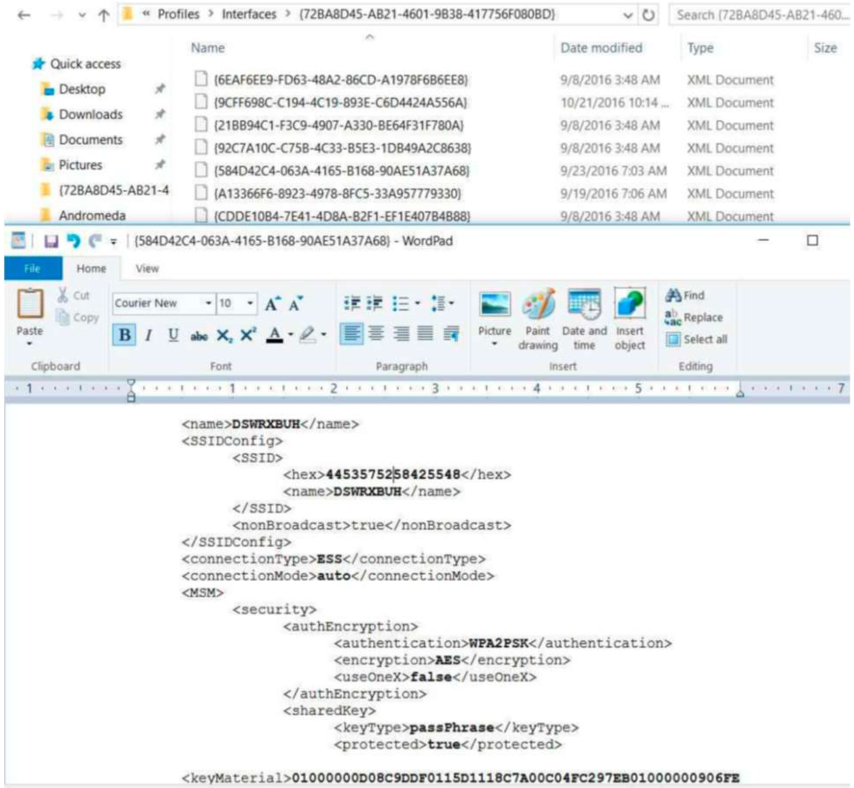
Wireless Network History

When a Windows system is connected to a wireless (WiFi) network, a record is kept in C:\ProgramData\Microsoft\Wlansvc\Profiles\Interfaces

A series of sub-folders will be created for each interface; the folder name is in GUID format and contains files named %GUID%.xml

The XML file will include the Service Set Identifier (SSID) of the network the device connected to.
The *.xml extension is used by Extensible Markup Language (XML), a programming language that is readable by both humans and computers.

Testing has shown that files relating to previously connected wireless network may be deleted



Name	Date modified	Type	Size
{6EAF6EE9-FD63-48A2-86CD-A1978F6B6EE8}	9/8/2016 3:48 AM	XML Document	
{9CFF698C-C194-4C19-893E-C6D4424A556A}	10/21/2016 10:14 ...	XML Document	
{21BB94C1-F3C9-4907-A330-BE64F31F780A}	9/8/2016 3:48 AM	XML Document	
{92C7A10C-C75B-4C33-B5E3-1DB49A2C8638}	9/8/2016 3:48 AM	XML Document	
{584D42C4-063A-4165-B168-90AE51A37A68}	9/23/2016 7:03 AM	XML Document	
{A13366F6-8923-4978-8FC5-33A957779330}	9/19/2016 7:06 AM	XML Document	
{CDDE10B4-7E41-4D8A-B2F1-EF1E407B4B88}	9/8/2016 3:48 AM	XML Document	

{584D42C4-063A-4165-B168-90AE51A37A68} - WordPad

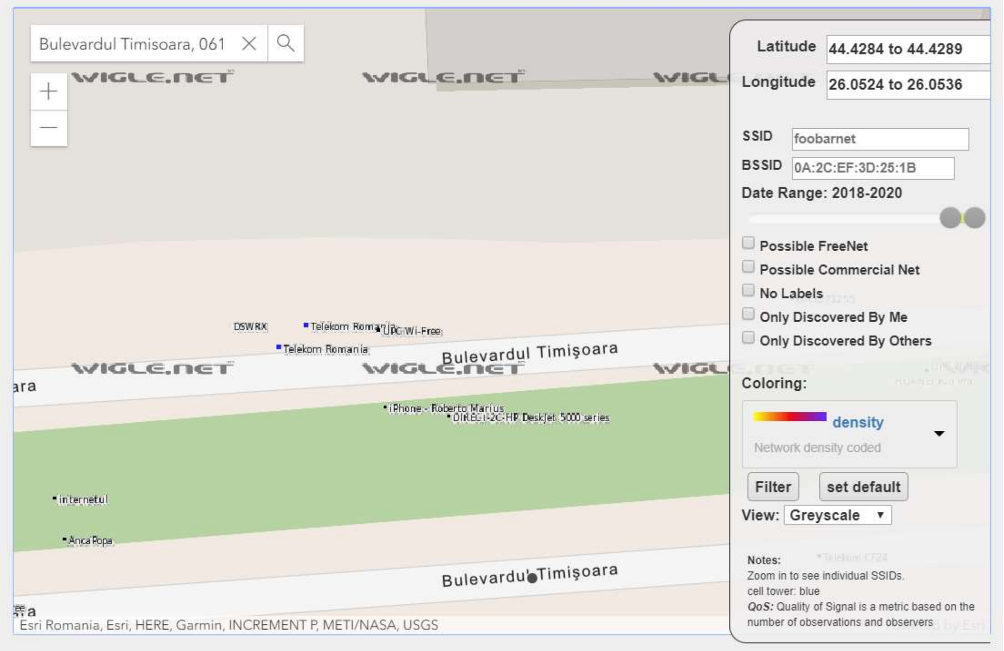
```
<name>DSWRXBUH</name>
<SSIDConfig>
        <SSID>
                <hex>4453575258425548</hex>
                <name>DSWRXBUH</name>
        </SSID>
        <nonBroadcast>true</nonBroadcast>
</SSIDConfig>
<connectionType>ESS</connectionType>
<connectionMode>auto</connectionMode>
<MSM>
        <security>
                <authEncryption>
                        <authentication>WPA2PSK</authentication>
                        <encryption>AES</encryption>
                        <useOneX>false</useOneX>
                </authEncryption>
                <sharedKey>
                        <keyType>passPhrase</keyType>
                        <protected>true</protected>

<keyMaterial>01000000D08C9DDF0115D1118C7A00C04FC297EB01000000906FE
```

Windows Artifacts

Physical Location

Wireless Network History

Using a public service such as https://wigle.net, the SSID can be used to pinpoint the geo-location of the access point used for that connection



Bulevardul Timisoara, 061

Latitude 44.4284 to 44.4289

Longitude 26.0524 to 26.0536

SSID foobarnet

BSSID 0A:2C:EF:3D:25:1B

Date Range: 2018-2020

☐ Possible FreeNet
☐ Possible Commercial Net
☐ No Labels
☐ Only Discovered By Me
☐ Only Discovered By Others

Coloring: density
Network density coded

View: Greyscale

Notes:
Zoom in to see individual SSIDs.
cell tower: blue
QoS: Quality of Signal is a metric based on the number of observations and observers

Windows Artifacts

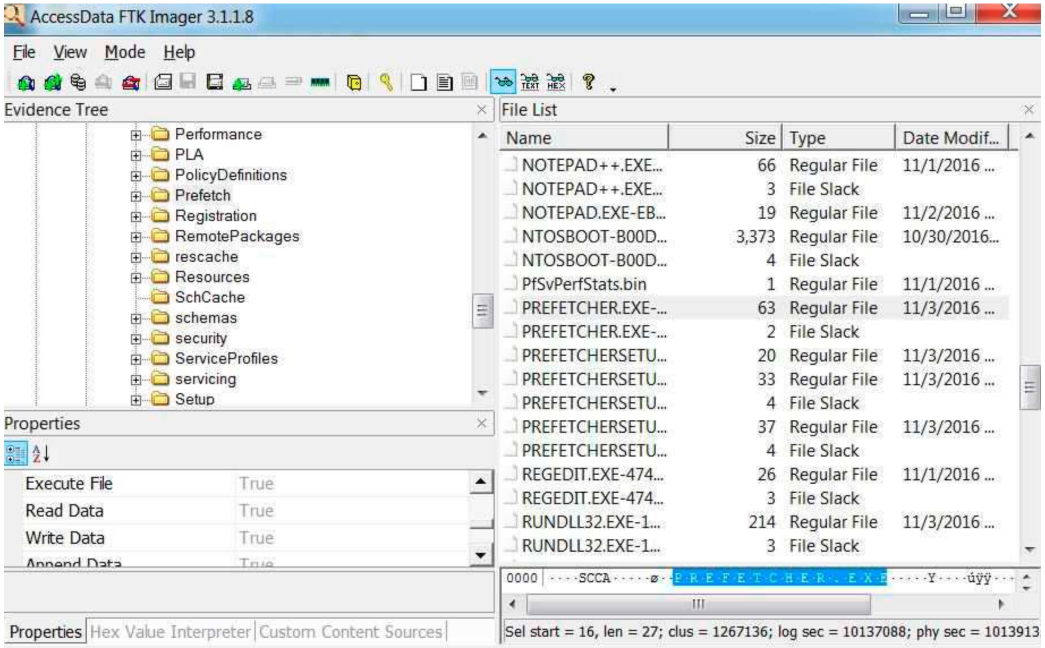
Program Execution

Prefetch / SuperFetch

Prefetching speeds up computer performance by bringing the data and code pages of programs used during boot process and in subsequent program launches into memory from the disk before that data and code is actually demanded.

The prefetch files that are created as a result of the tracing process that occurs are located in %WINDOWS%\Prefetch.

The file's name is the name of the application to which the trace applies followed by a dash and the hexadecimal representation of a hash of the file's path ending with a .PF file extension.
The prefetch folder will never grow larger than 129 entries.



Performance
PLA
PolicyDefinitions
Prefetch
Registration
RemotePackages
rescache
Resources
SchCache
schemas
security
ServiceProfiles
servicing
Setup

Execute File	True
Read Data	True
Write Data	True
Append Data	True

Name	Size	Type	Date Modif...
NOTEPAD++.EXE...	66	Regular File	11/1/2016 ...
NOTEPAD++.EXE...	3	File Slack	
NOTEPAD.EXE-EB...	19	Regular File	11/2/2016 ...
NTOSBOOT-B00D...	3,373	Regular File	10/30/2016...
NTOSBOOT-B00D...	4	File Slack	
PfSvPerfStats.bin	1	Regular File	11/1/2016 ...
PREFETCHER.EXE-...	63	Regular File	11/3/2016 ...
PREFETCHER.EXE-...	2	File Slack	
PREFETCHERSETU...	20	Regular File	11/3/2016 ...
PREFETCHERSETU...	33	Regular File	11/3/2016 ...
PREFETCHERSETU...	4	File Slack	
PREFETCHERSETU...	37	Regular File	11/3/2016 ...
PREFETCHERSETU...	4	File Slack	
REGEDIT.EXE-474...	26	Regular File	11/1/2016 ...
REGEDIT.EXE-474...	3	File Slack	
RUNDLL32.EXE-1...	214	Regular File	11/3/2016 ...
RUNDLL32.EXE-1...	3	File Slack	

```
0000 ····SCCA·····ø··P·R·E·F·E·T·C·H·E·R·.·E·X·E·····Y····úÿÿ···
```

Sel start = 16, len = 27; clus = 1267136; log sec = 10137088; phy sec = 1013913

Looking at the actual content of the .PF file, the name of the executable file being traced is located at offset 10h an is visible in plaintext. The file will also contain the run count, last run date and a list of files used by the application when it loads.

Windows Artifacts

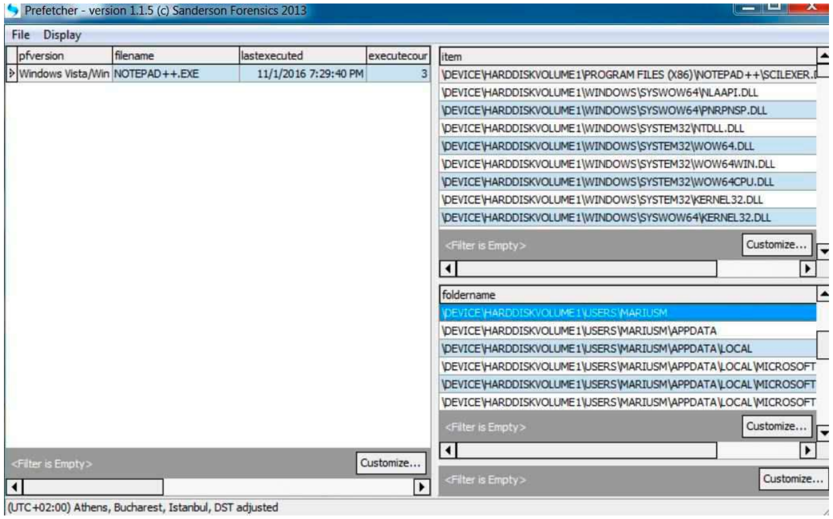
Program Execution

Prefetch / SuperFetch

SuperFetch was introduced in Windows Vista. It adds additional functionality by keeping track of when and how often a program is run.

In a forensic examination, prefetch files can be used to help determine when an application was last run. This is useful for creating a timeline of events or if attempting to determine if a virus or other exploit is active on a computer.

Examining the files and directories accessed during the launch of an application can reveal hidden directories, point to user accounts or show that an application was accessed from an external storage drive.



pfversion	filename	lastexecuted	executecour
Windows Vista/Win	NOTEPAD++.EXE	11/1/2016 7:29:40 PM	3

<Filter is Empty>

item
\DEVICE\HARDDISKVOLUME1\PROGRAM FILES (X86)\NOTEPAD++\SCILEXER.I
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\NLAAPI.DLL
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\PNRPNSP.DLL
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\NTDLL.DLL
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\WOW64.DLL
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\WOW64WIN.DLL
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\WOW64CPU.DLL
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\KERNEL32.DLL
\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSWOW64\KERNEL32.DLL

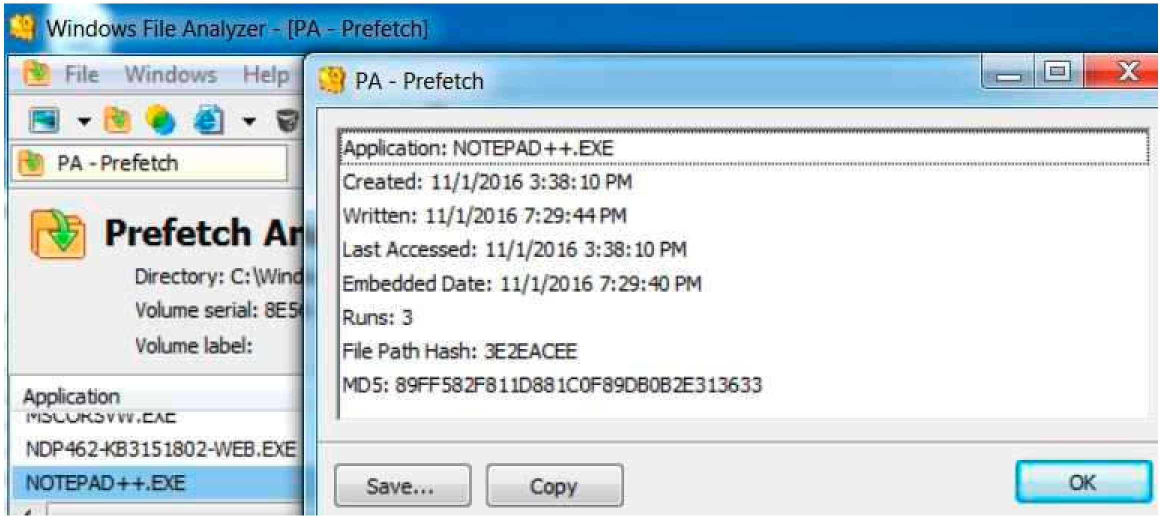
<Filter is Empty>

foldername
\DEVICE\HARDDISKVOLUME1\USERS\MARIUSM
\DEVICE\HARDDISKVOLUME1\USERS\MARIUSM\APPDATA
\DEVICE\HARDDISKVOLUME1\USERS\MARIUSM\APPDATA\LOCAL
\DEVICE\HARDDISKVOLUME1\USERS\MARIUSM\APPDATA\LOCAL\MICROSOFT
\DEVICE\HARDDISKVOLUME1\USERS\MARIUSM\APPDATA\LOCAL\MICROSOFT
\DEVICE\HARDDISKVOLUME1\USERS\MARIUSM\APPDATA\LOCAL\MICROSOFT

<Filter is Empty>

<Filter is Empty>

(UTC+02:00) Athens, Bucharest, Istanbul, DST adjusted



Directory: C:\Wind
Volume serial: 8E5
Volume label:

Application
MSCORSVW.EXE
NDP462-KB3151802-WEB.EXE
NOTEPAD++.EXE

Application: NOTEPAD++.EXE
Created: 11/1/2016 3:38:10 PM
Written: 11/1/2016 7:29:44 PM
Last Accessed: 11/1/2016 3:38:10 PM
Embedded Date: 11/1/2016 7:29:40 PM
Runs: 3
File Path Hash: 3E2EACEE
MD5: 89FF582F811D881C0F89DB0B2E313633

Windows Registry

Forensics and the Windows Registry

Windows registry is a central hierarchical database in which Microsoft Windows stores information that is necessary to configure the system for one or more users, applications, and hardware devices (profiles for each user, applications that are installed on the computer, types of documents that each can create, property sheet settings for folders and application icons, hardware that exists on the system, and the ports that are being used etc.)

The highest element of the hierarchy is known as a **hive**, which maps to one or more files in the file system that contains a binary database of registry keys and values. Hives are designed to store specific types of information.

The registry contains useful information about the computer system and its users:

System Settings
Hardware Information
USB Devices
User Names and Security Identifiers (SID)
Personal User Settings

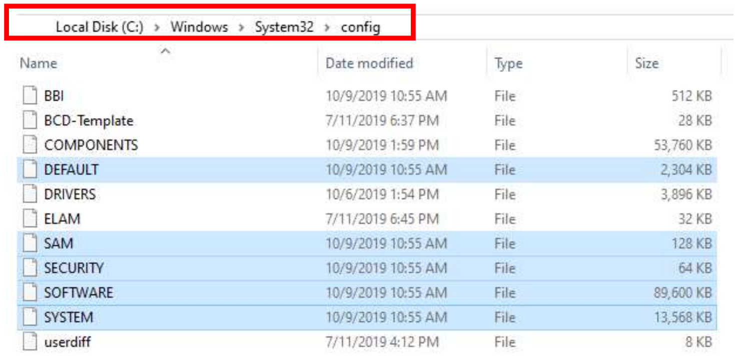
Browser Preferences
Web Browsing Activity
Recently Opened Items
Programs Execution
Windows Firewall

Windows Registry

Registry Hive Files

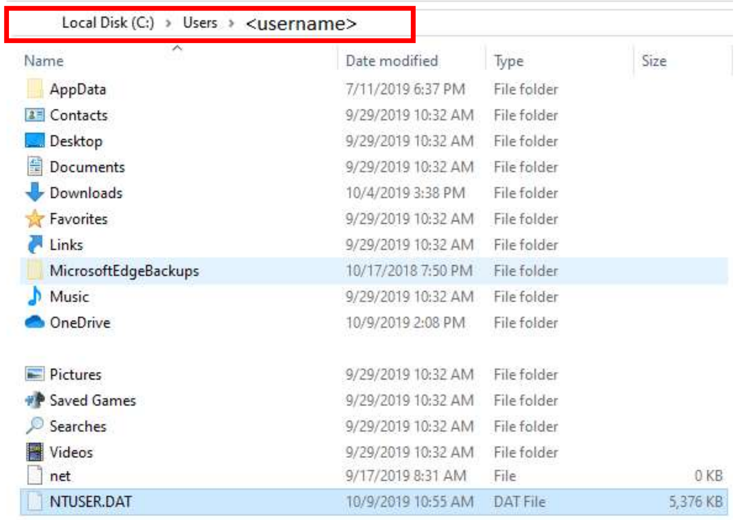
The registry is built upon a series of hive files.
A registry hive is a group of keys, subkeys, and values in the registry that has a set of supporting files.

Registry Hive	Supporting Files
HKEY_CURRENT_CONFIG	**System**, System.alt, System.log, System.sav
HKEY_CURRENT_USER	**Ntuser.dat**, Ntuser.dat.log
HKEY_LOCAL_MACHINE\SAM	**Sam**, Sam.log, Sam.sav
HKEY_LOCAL_MACHINE\Security	**Security**, Security.log, Security.sav
HKEY_LOCAL_MACHINE\Software	**Software**, Software.log, Software.sav
HKEY_LOCAL_MACHINE\System	**System**, System.alt, System.log, System.sav
HKEY_USERS\.DEFAULT	Default, Default.log, Default.sav



Local Disk (C:) › Windows › System32 › config

Name	Date modified	Type	Size
BBI	10/9/2019 10:55 AM	File	512 KB
BCD-Template	7/11/2019 6:37 PM	File	28 KB
COMPONENTS	10/9/2019 1:59 PM	File	53,760 KB
DEFAULT	10/9/2019 10:55 AM	File	2,304 KB
DRIVERS	10/6/2019 1:54 PM	File	3,896 KB
ELAM	7/11/2019 6:45 PM	File	32 KB
SAM	10/9/2019 10:55 AM	File	128 KB
SECURITY	10/9/2019 10:55 AM	File	64 KB
SOFTWARE	10/9/2019 10:55 AM	File	89,600 KB
SYSTEM	10/9/2019 10:55 AM	File	13,568 KB
userdiff	7/11/2019 4:12 PM	File	8 KB



Local Disk (C:) › Users › <username>

Name	Date modified	Type	Size
AppData	7/11/2019 6:37 PM	File folder	
Contacts	9/29/2019 10:32 AM	File folder	
Desktop	9/29/2019 10:32 AM	File folder	
Documents	9/29/2019 10:32 AM	File folder	
Downloads	10/4/2019 3:38 PM	File folder	
Favorites	9/29/2019 10:32 AM	File folder	
Links	9/29/2019 10:32 AM	File folder	
MicrosoftEdgeBackups	10/17/2018 7:50 PM	File folder	
Music	9/29/2019 10:32 AM	File folder	
OneDrive	10/9/2019 2:08 PM	File folder	
Pictures	9/29/2019 10:32 AM	File folder	
Saved Games	9/29/2019 10:32 AM	File folder	
Searches	9/29/2019 10:32 AM	File folder	
Videos	9/29/2019 10:32 AM	File folder	
net	9/17/2019 8:31 AM	File	0 KB
NTUSER.DAT	10/9/2019 10:55 AM	DAT File	5,376 KB

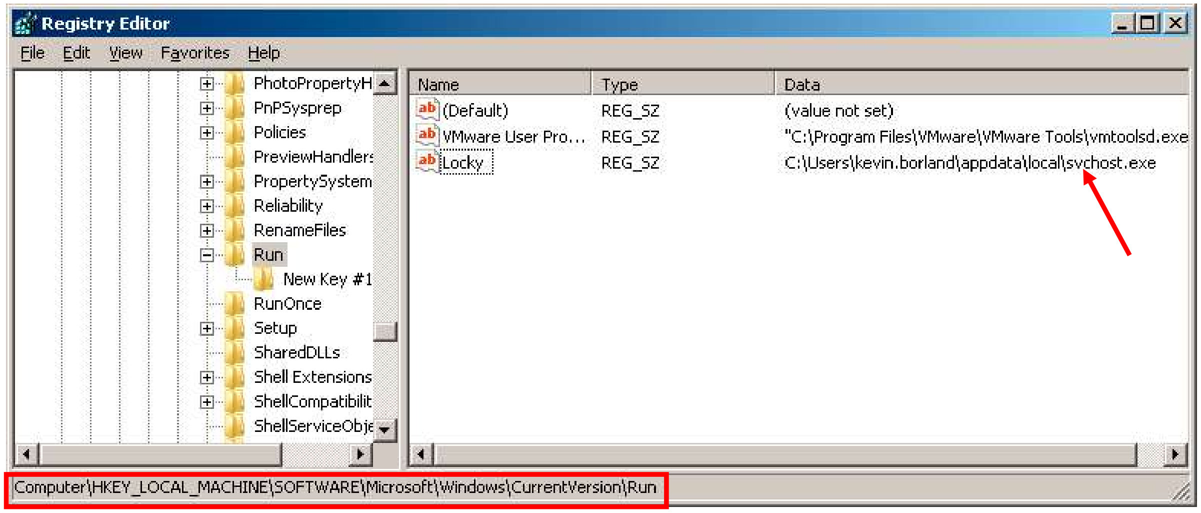
Windows Registry

Registry Autorun Locations

Often referred to as ***Autoruns*** or ***Autostart locations***, these registry entries are used to automatically start programs, either when the system boots or a user logs in.

- Indicates executables locations
- Last written time can indicate when a potential infection occurred

Can be used by both legitimate applications and by malware such as trojans, viruses, worms, spyware or adware.



Name	Type	Data
(Default)	REG_SZ	(value not set)
VMware User Pro...	REG_SZ	"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe
Locky	REG_SZ	C:\Users\kevin.borland\appdata\local\svchost.exe

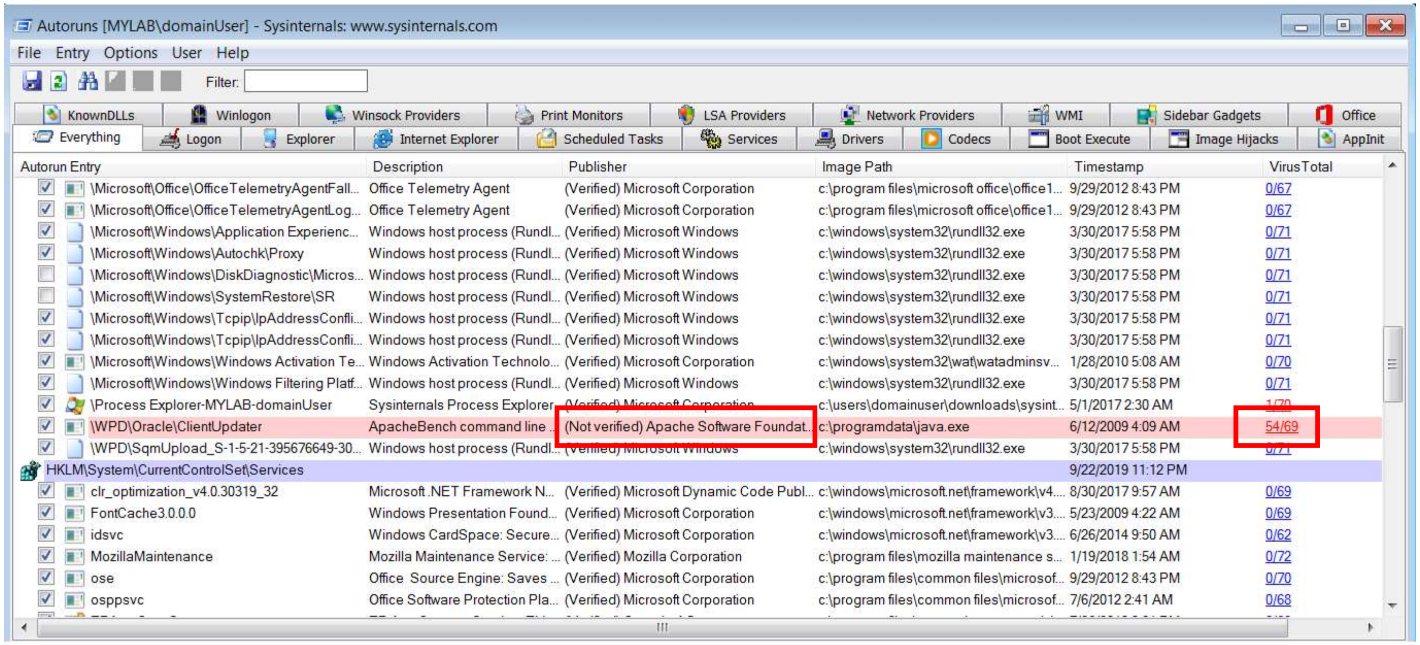
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Windows Registry

Registry Autorun Locations

Autoruns from Sysinternals (https://docs.microsoft.com/en-us/sysinternals/downloads/autoruns) is a great tool that shows you what programs are configured to run during system bootup or user login.

File hashes can be automatically submitted to VirusTotal to determine if a program is malicious or bening.



Autorun Entry	Description	Publisher	Image Path	Timestamp	VirusTotal
☑ \Microsoft\Office\OfficeTelemetryAgentFall...	Office Telemetry Agent	(Verified) Microsoft Corporation	c:\program files\microsoft office\office1...	9/29/2012 8:43 PM	0/67
☑ \Microsoft\Office\OfficeTelemetryAgentLog...	Office Telemetry Agent	(Verified) Microsoft Corporation	c:\program files\microsoft office\office1...	9/29/2012 8:43 PM	0/67
☑ \Microsoft\Windows\Application Experienc...	Windows host process (Rundl...	(Verified) Microsoft Windows	c:\windows\system32\rundll32.exe	3/30/2017 5:58 PM	0/71
☑ \Microsoft\Windows\Autochk\Proxy	Windows host process (Rundl...	(Verified) Microsoft Windows	c:\windows\system32\rundll32.exe	3/30/2017 5:58 PM	0/71
☐ \Microsoft\Windows\DiskDiagnostic\Micros...	Windows host process (Rundl...	(Verified) Microsoft Windows	c:\windows\system32\rundll32.exe	3/30/2017 5:58 PM	0/71
☐ \Microsoft\Windows\SystemRestore\SR	Windows host process (Rundl...	(Verified) Microsoft Windows	c:\windows\system32\rundll32.exe	3/30/2017 5:58 PM	0/71
☑ \Microsoft\Windows\Tcpip\IpAddressConfli...	Windows host process (Rundl...	(Verified) Microsoft Windows	c:\windows\system32\rundll32.exe	3/30/2017 5:58 PM	0/71
☑ \Microsoft\Windows\Tcpip\IpAddressConfli...	Windows host process (Rundl...	(Verified) Microsoft Windows	c:\windows\system32\rundll32.exe	3/30/2017 5:58 PM	0/71
☑ \Microsoft\Windows\Windows Activation Te...	Windows Activation Technolo...	(Verified) Microsoft Corporation	c:\windows\system32\wat\watadminsv...	1/28/2010 5:08 AM	0/70
☑ \Microsoft\Windows\Windows Filtering Platf...	Windows host process (Rundl...	(Verified) Microsoft Windows	c:\windows\system32\rundll32.exe	3/30/2017 5:58 PM	0/71
☑ \Process Explorer-MYLAB-domainUser	Sysinternals Process Explorer	(Verified) Microsoft Corporation	c:\users\domainuser\downloads\sysint...	5/1/2017 2:30 AM	1/70
☑ \WPD\Oracle\ClientUpdater	ApacheBench command line ...	(Not verified) Apache Software Foundat...	c:\programdata\java.exe	6/12/2009 4:09 AM	54/69
☑ \WPD\SqmUpload_S-1-5-21-395676649-30...	Windows host process (Rundl...	(Verified) Microsoft Windows	c:\windows\system32\rundll32.exe	3/30/2017 5:58 PM	0/71
HKLM\System\CurrentControlSet\Services				9/22/2019 11:12 PM	
☑ clr_optimization_v4.0.30319_32	Microsoft .NET Framework N...	(Verified) Microsoft Dynamic Code Publ...	c:\windows\microsoft.net\framework\v4....	8/30/2017 9:57 AM	0/69
☑ FontCache3.0.0.0	Windows Presentation Found...	(Verified) Microsoft Corporation	c:\windows\microsoft.net\framework\v3....	5/23/2009 4:22 AM	0/69
☑ idsvc	Windows CardSpace: Secure...	(Verified) Microsoft Corporation	c:\windows\microsoft.net\framework\v3....	6/26/2014 9:50 AM	0/62
☑ MozillaMaintenance	Mozilla Maintenance Service: ...	(Verified) Mozilla Corporation	c:\program files\mozilla maintenance s...	1/19/2018 1:54 AM	0/72
☑ ose	Office Source Engine: Saves ...	(Verified) Microsoft Corporation	c:\program files\common files\microsof...	9/29/2012 8:43 PM	0/70
☑ osppsvc	Office Software Protection Pla...	(Verified) Microsoft Corporation	c:\program files\common files\microsof...	7/6/2012 2:41 AM	0/68

Thank you !