POLI
TÉCNICO
GUARDA

Fundamentals of Digital Forensics

- Digital Fundamentals
- The Digital Forensics Process
- Volatility Tool
- Wireshark
- (Autopsy)

Cybersecurity

Interreg España - Portugal
Cofinanciado por la Unión Europea
Cofinanciado pela União Europeia
CIBERIA

Pedro Pinto | ppinto@ipg.pt

Docente

- **Pedro Pinto**
  - CISO do Instituto Politécnico da Guarda (IPG)
  - Responsável pelo Centro de Resposta a Incidentes do IPG
  - Membro da Rede Nacional de Resposta a Incidentes Informáticos
  - Membro da Metared
    - Curso CIO, CISO e Responsabilidades Jurídicas
  - Responsável e formador da C-ACADEMY
  - Membro do projeto Ciberia (POCTEP)
  - Administrador do maior site de tecnologia em Portugal (Pplware.sapo.pt)
  - Administrador do Grupo de Emergência da Guarda
  - +info: https://linktr.ee/ppinto | LinkedIn - https://www.linkedin.com/in/infopedropinto/

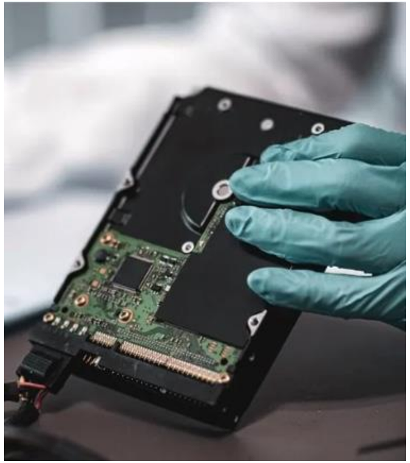
Digital forensics

- is a **forensic science branch** that focuses on **recovering material** found in digital devices during cybercrime investigation.

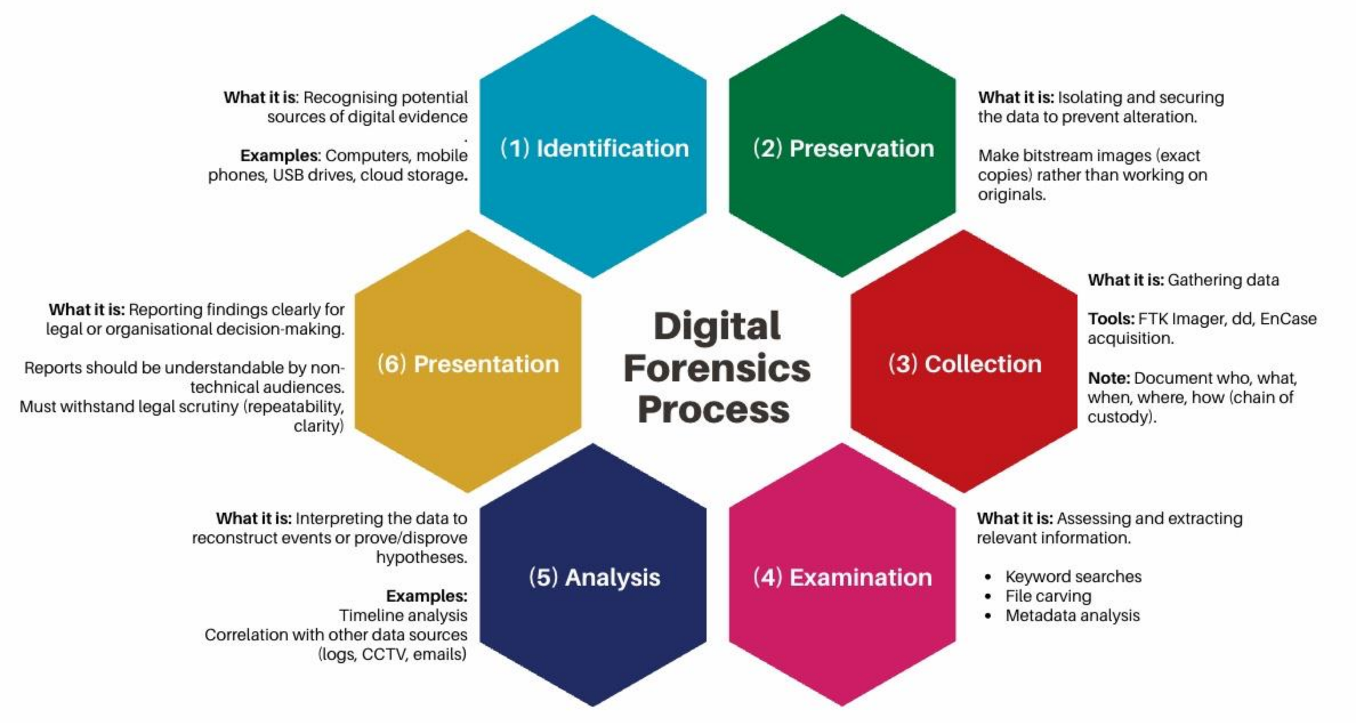
Fundamentals of Digital Forensics

- **Digital Evidence**
  - includes information on **computers**, **audio files**, **video recordings** and **digital images** (nist.gov)
- **CSIRT or Computer Security Incident Response Team**
  - group of professionals responsible for responding to an incident and assisting with analyzing evidence collected during the investigation of cybercrime.

- **Many branches in which data can be collected**, including:
  - Network Forensics
  - Computer Forensics
  - Mobile Forensics
  - Database Forensics
  - Forensic Data Analysis



The Digital Forensics Process (Framework)



Digital Forensics Process

(1) Identification

What it is: Recognising potential sources of digital evidence.

Examples: Computers, mobile phones, USB drives, cloud storage.

(2) Preservation

What it is: Isolating and securing the data to prevent alteration.

Make bitstream images (exact copies) rather than working on originals.

(3) Collection

What it is: Gathering data

Tools: FTK Imager, dd, EnCase acquisition.

Note: Document who, what, when, where, how (chain of custody).

(4) Examination

What it is: Assessing and extracting relevant information.

- Keyword searches
- File carving
- Metadata analysis

(5) Analysis

What it is: Interpreting the data to reconstruct events or prove/disprove hypotheses.

Examples:
Timeline analysis
Correlation with other data sources (logs, CCTV, emails)

(6) Presentation

What it is: Reporting findings clearly for legal or organisational decision-making.

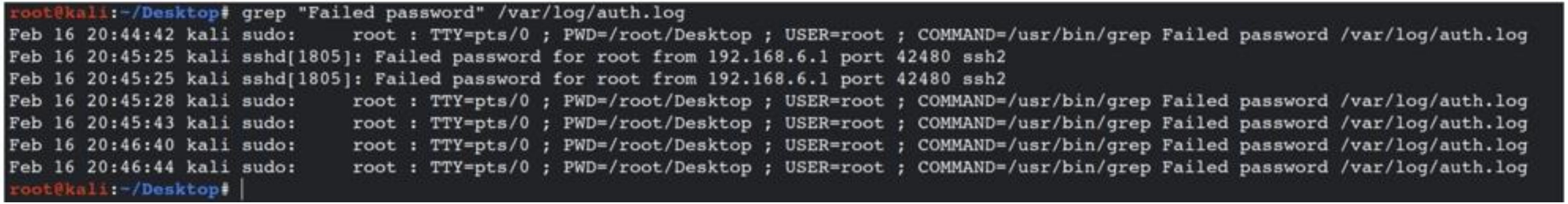
Reports should be understandable by non-technical audiences.
Must withstand legal scrutiny (repeatability, clarity)

The Digital Forensics Process: Identification

- **Locard's Exchange Principle**, a theory developed by Dr. Edmond Locard (1877–1966).
  - “Any action of an individual, and obviously the violent action constituting a crime, cannot occur without leaving a trace.”
- This theory correlates with digital forensic because cybercriminals often leave traces of their presence after an attack (**trace evidence**)

The Digital Forensics Process: Identification

- Example
  - individual tries to SSH into a system but enters an incorrect password, that attempt is logged into the **/var/log/auth.log** file, which can be used as evidence



```
root@kali:~/Desktop# grep "Failed password" /var/log/auth.log
Feb 16 20:44:42 kali sudo:     root : TTY=pts/0 ; PWD=/root/Desktop ; USER=root ; COMMAND=/usr/bin/grep Failed password /var/log/auth.log
Feb 16 20:45:25 kali sshd[1805]: Failed password for root from 192.168.6.1 port 42480 ssh2
Feb 16 20:45:25 kali sshd[1805]: Failed password for root from 192.168.6.1 port 42480 ssh2
Feb 16 20:45:28 kali sudo:     root : TTY=pts/0 ; PWD=/root/Desktop ; USER=root ; COMMAND=/usr/bin/grep Failed password /var/log/auth.log
Feb 16 20:45:43 kali sudo:     root : TTY=pts/0 ; PWD=/root/Desktop ; USER=root ; COMMAND=/usr/bin/grep Failed password /var/log/auth.log
Feb 16 20:46:40 kali sudo:     root : TTY=pts/0 ; PWD=/root/Desktop ; USER=root ; COMMAND=/usr/bin/grep Failed password /var/log/auth.log
Feb 16 20:46:44 kali sudo:     root : TTY=pts/0 ; PWD=/root/Desktop ; USER=root ; COMMAND=/usr/bin/grep Failed password /var/log/auth.log
root@kali:~/Desktop# 
```

The Digital Forensics Process: Preservation

- **Once the evidence has been identified**, the next step is to **preserve** the evidence
- Safeguarding the evidence from being manipulated or deleted.
- In some cases, controls may be set to prevent unauthorized access to a system containing evidence
  - **Example**: isolating the system on the network or even restricting physical access to the system

The Digital Forensics Process: Preservation

Techniques used to preserve evidence, some of which include:

- **Imaging drives**
  - The process of creating a forensic digital copy of a hard drive to retain evidence and to be used in an investigation.
- **Hashing values**
  - Involves generating a cryptographic hash such as MD5, SHA-1, or SHA-256 to verify the integrity of the digital evidence.
- **Following the Chain of Custody (CoC)**
  - Document all activity that occurs with the evidence.

The Digital Forensics Process: Preservation

Techniques used to preserve evidence, some of which include:

- **Do not change the current state of a device**
  - If a device is ON, do not turn it OFF and vice versa. If a device is ON, consult a forensic expert before turning the device OFF.
- **Ensure the device is physically secured**
  - Do not leave the device in an open or unsecured location; follow the CoC and keep a documented log detailing who has the device, its location, along with the date and time it was moved.
- **Do not open any files**
  - The examiner runs the risk of overwriting or losing data.

The Digital Forensics Process: Preservation

Drive Imaging (Preservation)

- process of creating a bit-by-bit copy of a hard drive
- Forensically imaging a drive plays a crucial part in preserving an exact copy of a storage device
- It is ideal for the forensic examiner to analyze the duplicate image rather than the original media
- Once the drive has been imaged, the system itself should no longer be operated on and isolated from incoming and outgoing connections
- Doing this limits the risk of the evidence being altered or destroyed if it needs to be used in court

The Digital Forensics Process: Preservation

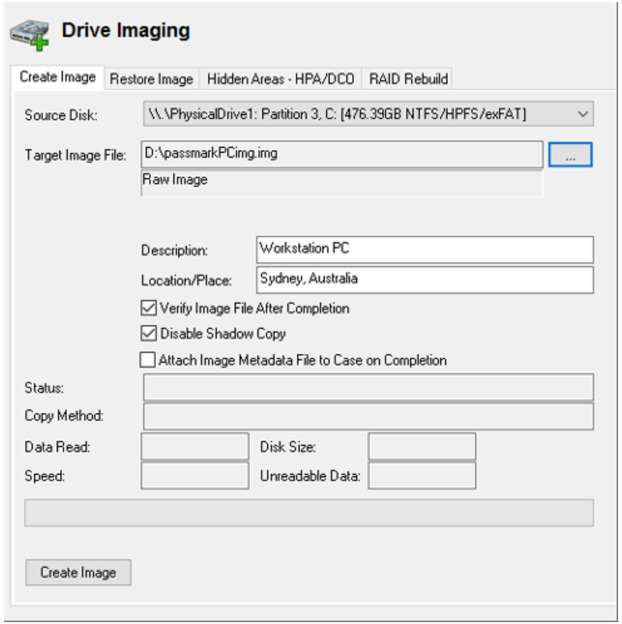
Drive Imaging (Preservation)

- Using hardware such as a **write blocker** can aid the examiner with the imaging process and prevent any data from being written to the hard drive
- PassMark's OSForensics™ software has a drive imaging function

.

The Digital Forensics Process: Preservation

Drive Imaging (Preservation) - OSForensics



Drive Imaging

Source Disk: \\.\PhysicalDrive1: Partition 3, C: [476.39GB NTFS/HPFS/exFAT]

Target Image File: D:\passmarkPCimg.img

Raw Image

Description: Workstation PC

Location/Place: Sydney, Australia

☑ Verify Image File After Completion

☑ Disable Shadow Copy

☐ Attach Image Metadata File to Case on Completion

Status:

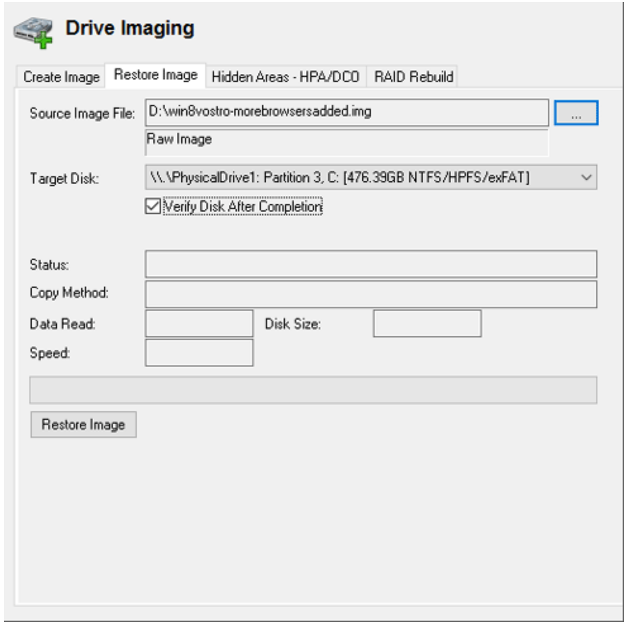
Copy Method:

Data Read:

Disk Size:

Speed:

Unreadable Data:



Drive Imaging

Source Image File: D:\win8vostro-morebrowsersadded.img

Raw Image

Target Disk: \\.\PhysicalDrive1: Partition 3, C: [476.39GB NTFS/HPFS/exFAT]

☑ Verify Disk After Completion

Status:

Copy Method:

Data Read:

Disk Size:

Speed:

The Digital Forensics Process: Preservation

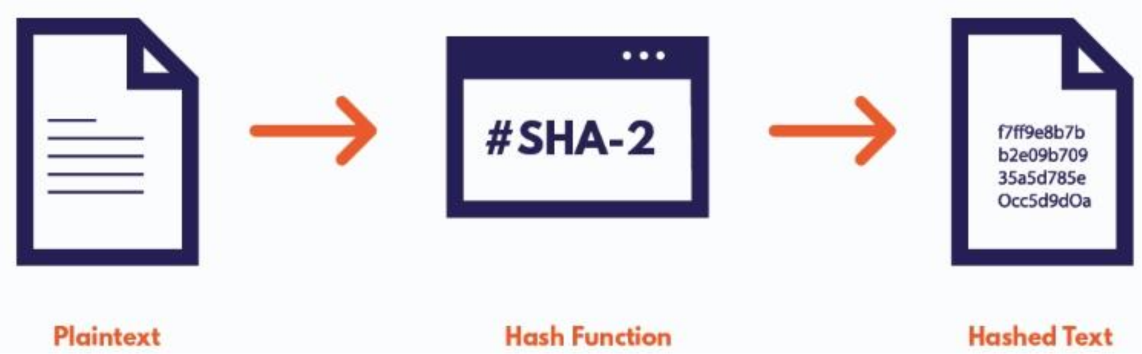
Other software that can be used for forensic drive imaging

- Sleuth Kit (+Autopsy)
- EnCase
- PALADIN
- CAINE
- SANS SIFT
- FTK Image

The Digital Forensics Process: Preservation

Hashing Values (Preservation)

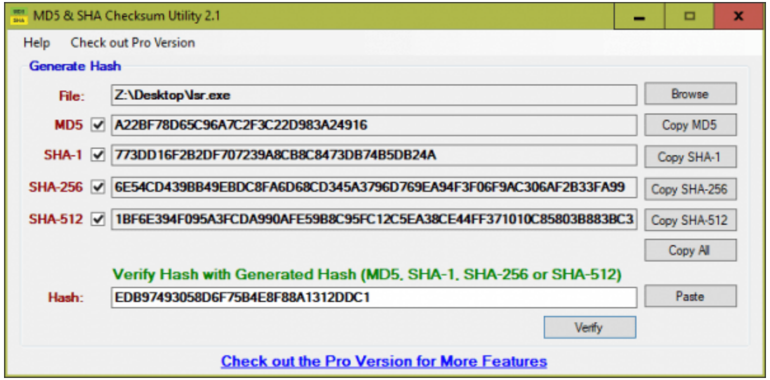
- To **generate a cryptographic hash** (MD5, SHA-1, SHA-256) of the evidence during the imaging process, specific software is used
- If any data within the evidence is altered, a new hash will be generated



The Digital Forensics Process: Preservation

Hashing Values (Preservation)

- **Hashing tools that can be used**
  - CertUtil (CMD)
  - Get-FileHash (Powershell)
  - Hash Generator (by OpenSSL Group, Paulo S. L. M. Barreto & Vincent Rijmen)
  - MD5 & SHA Checksum Utility (by Raymond Lin)
  - HashMyFiles (by Nir Sofer)



File: Z:\Desktop\lsr.exe

MD5 ☑ A22BF78D65C96A7C2F3C22D983A24916

SHA-1 ☑ 773DD16F2B2DF707239A8CB8C8473DB74B5DB24A

SHA-256 ☑ 6E54CD439BB49EBDC8FA6D68CD345A3796D769EA94F3F06F9AC306AF2B33FA99

SHA-512 ☑ 1BF6E394F095A3FCDA990AFE59B8C95FC12C5EA38CE44FF371010C85803B883BC3

Verify Hash with Generated Hash (MD5, SHA-1, SHA-256 or SHA-512)

Hash: EDB97493058D6F75B4E8F88A1312DDC1

The Digital Forensics Process: Preservation

Collection

- In the collection stage, digital forensics examiners will begin the process of acquiring volatile digital evidence
- **Volatile evidence** is evidence that can be lost when a system is powered down

- **Volatile data**
  - active connections
  - log data stored on a network device
  - running memory
  - remotely logged data
  - Address Resolution Protocol cache

The Digital Forensics Process: Preservation

Collection – Example order of volatility

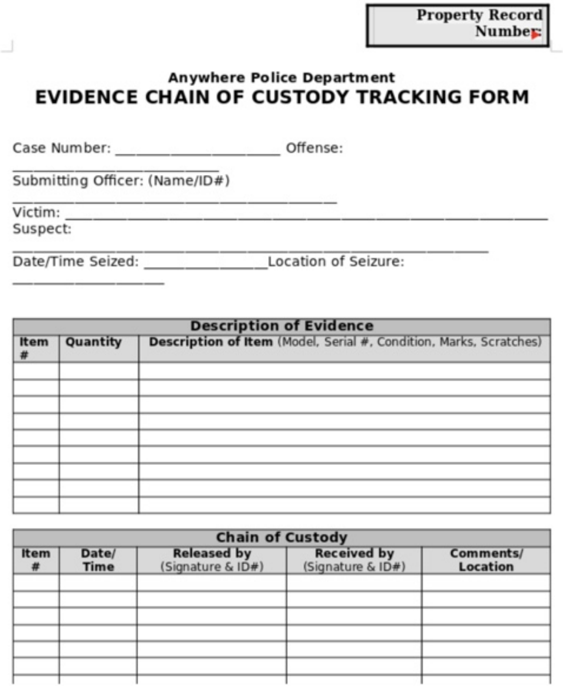
1. CPU, cache, and register content
2. Routing table, ARP cache, process table, kernel statistics
3. Memory
4. Temporary file system/swap space
5. Data on a hard disk
6. Remotely logged data
7. Data contained on archival media

Lastly, all evidence that has been collected should be documented

The Digital Forensics Process: Collecion

Chain of Custody (Collection)

- documentation of an evidence life cycle during an investigation



Property Record Number:

Anywhere Police Department

EVIDENCE CHAIN OF CUSTODY TRACKING FORM

Case Number: ____________________ Offense: ______________________________

Submitting Officer: (Name/ID#) ____

Victim: ____

Suspect: ____

Date/Time Seized: _______________Location of Seizure: ___________________

Description of Evidence		
Item #	Quantity	Description of Item (Model, Serial #, Condition, Marks, Scratches)

Chain of Custody				
Item #	Date/ Time	Released by (Signature & ID#)	Received by (Signature & ID#)	Comments/ Location

The Digital Forensics Process: Examination and Analysis

Examination and Analysis

- involves **discovering and extracting** data from the evidence using specific tools and techniques.
- The evidence is then seized as part of the incident
- The analysis process varies depending on the type of digital evidence
- The analysis process helps to determine the origin of the data

The Digital Forensics Process: Examination and Analysis

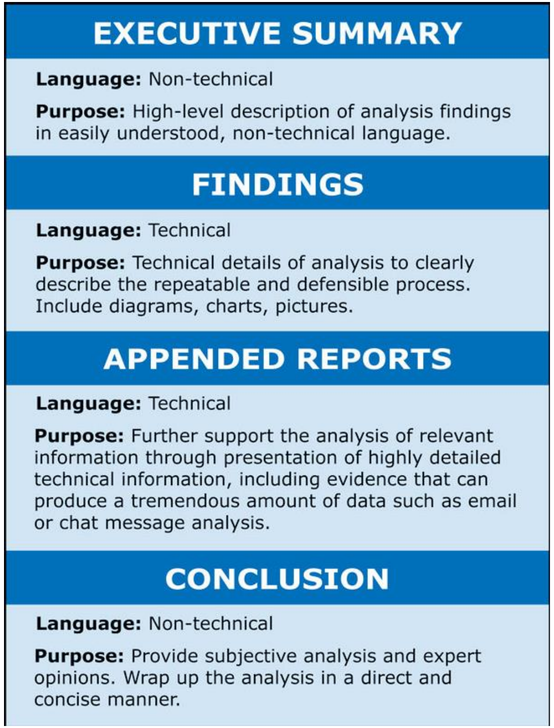
Examination and Analysis – Tools

- **Autopsy/The Sleuth Kit**
  - Designed to perform analysis of disk images, filesystems and includes a wide variety of other features.
- **AccessData FTK**
  - A toolkit that focuses on aiding examiners with a quick analysis process.
- **Paraben Suite**
  - A suite of forensics tools, some of which include smartphone and cloud analysis tools.
- **Volatility**
  - A tool used in memory forensics, it extracts information from running processes.

The Digital Forensics Process: Presentation

Presentation

- forensic examiners must prepare a detailed written report to address the actions performed to obtain the evidence, including any limitations encountered during the investigation
- This report must be clear, concise, and unbiased
- Digital forensics reports should typically be organized in this fashion
  - Executive summary
  - Findings
  - Appended reports
  - Conclusion



EXECUTIVE SUMMARY

Language: Non-technical

Purpose: High-level description of analysis findings in easily understood, non-technical language.

FINDINGS

Language: Technical

Purpose: Technical details of analysis to clearly describe the repeatable and defensible process. Include diagrams, charts, pictures.

APPENDED REPORTS

Language: Technical

Purpose: Further support the analysis of relevant information through presentation of highly detailed technical information, including evidence that can produce a tremendous amount of data such as email or chat message analysis.

CONCLUSION

Language: Non-technical

Purpose: Provide subjective analysis and expert opinions. Wrap up the analysis in a direct and concise manner.

References

- ENISA - cyberskills

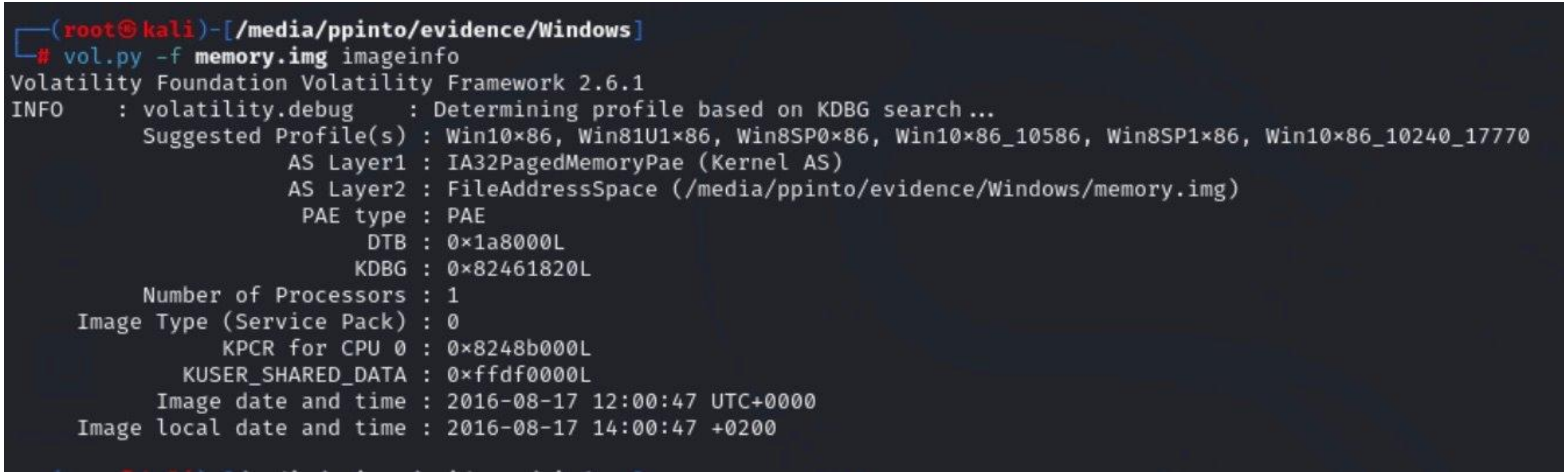
Volatility

Volatility

- is an open-source tool used for **analyzing memory dumps** (RAM captures) of computers.
- It **extracts digital artefacts** from memory images, useful for digital forensics, incident response, and malware analysis.

Volatility

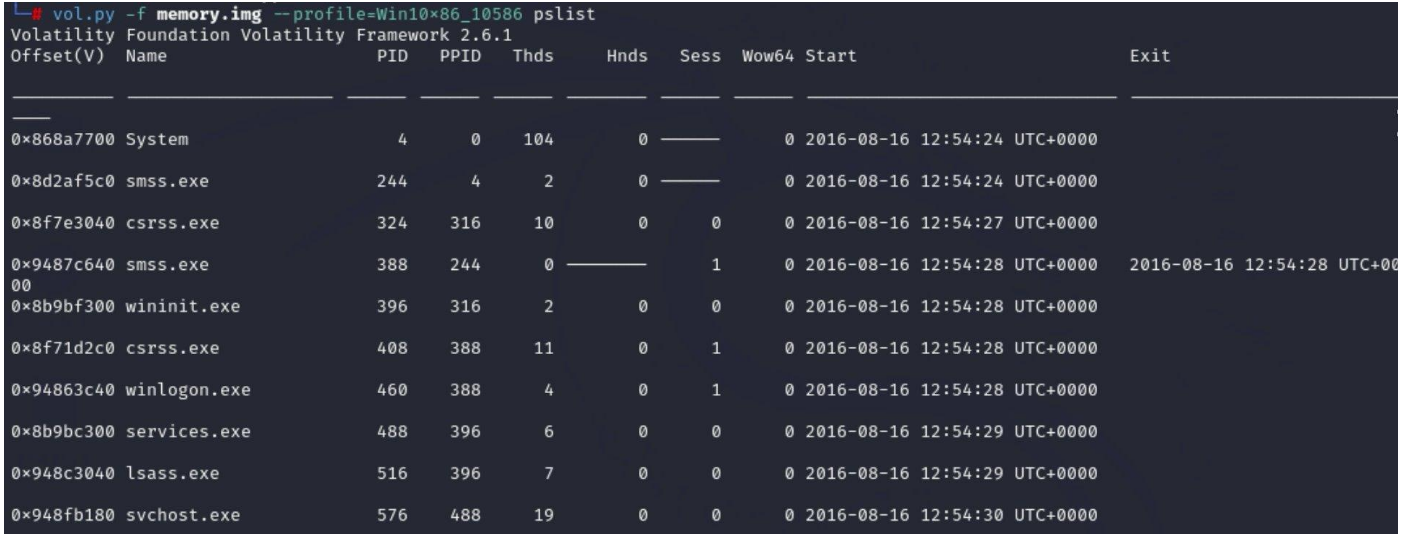
- **What type of dump am I going to analyze ?**
  - $ volatility -f MyDump.dmp imageinfo



```
┌──(root㉿kali)-[/media/ppinto/evidence/Windows]
└─# vol.py -f memory.img imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win10x86, Win81U1x86, Win8SP0x86, Win10x86_10586, Win8SP1x86, Win10x86_10240_17770
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/media/ppinto/evidence/Windows/memory.img)
                      PAE type : PAE
                           DTB : 0x1a8000L
                          KDBG : 0x82461820L
          Number of Processors : 1
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0x8248b000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2016-08-17 12:00:47 UTC+0000
     Image local date and time : 2016-08-17 14:00:47 +0200
```

Volatility

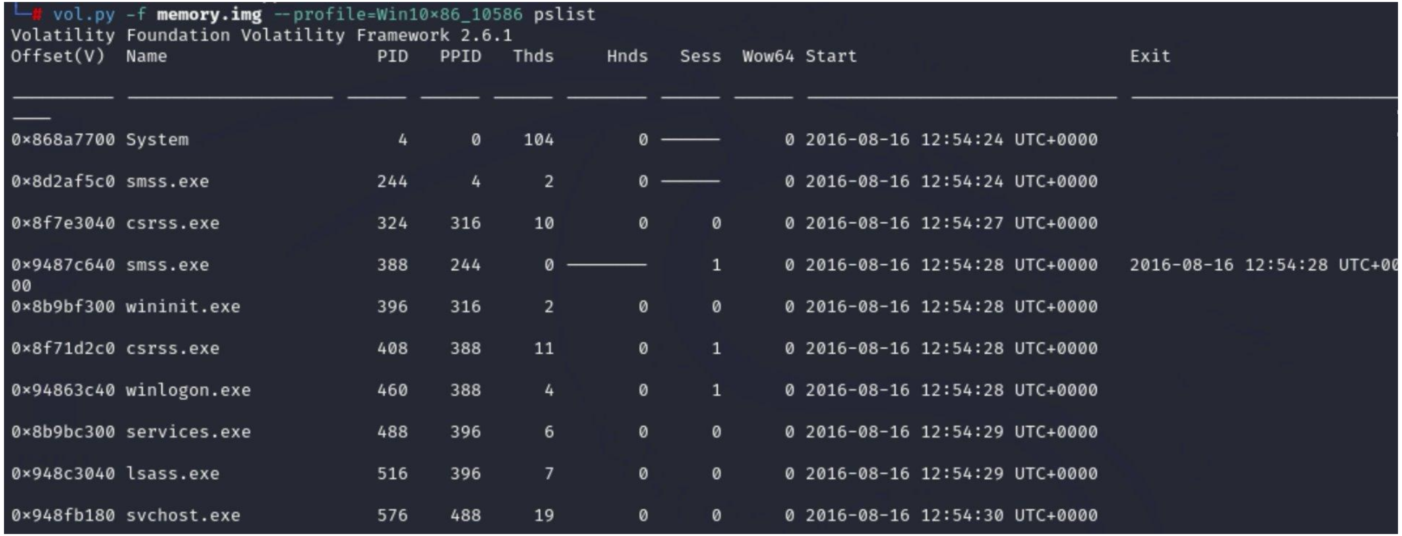
- **Which process are running?**
  - volatility -f MyDump.dmp --profile=MyProfile pslist



```
└─# vol.py -f memory.img --profile=Win10x86_10586 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit

---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ --------------------------
----
0x868a7700 System                    4      0    104        0 ------      0 2016-08-16 12:54:24 UTC+0000

0x8d2af5c0 smss.exe                244      4      2        0 ------      0 2016-08-16 12:54:24 UTC+0000

0x8f7e3040 csrss.exe               324    316     10        0      0      0 2016-08-16 12:54:27 UTC+0000

0x9487c640 smss.exe                388    244      0 --------      1      0 2016-08-16 12:54:28 UTC+0000   2016-08-16 12:54:28 UTC+00
00
0x8b9bf300 wininit.exe             396    316      2        0      0      0 2016-08-16 12:54:28 UTC+0000

0x8f71d2c0 csrss.exe               408    388     11        0      1      0 2016-08-16 12:54:28 UTC+0000

0x94863c40 winlogon.exe            460    388      4        0      1      0 2016-08-16 12:54:28 UTC+0000

0x8b9bc300 services.exe            488    396      6        0      0      0 2016-08-16 12:54:29 UTC+0000

0x948c3040 lsass.exe               516    396      7        0      0      0 2016-08-16 12:54:29 UTC+0000

0x948fb180 svchost.exe             576    488     19        0      0      0 2016-08-16 12:54:30 UTC+0000
```

Volatility

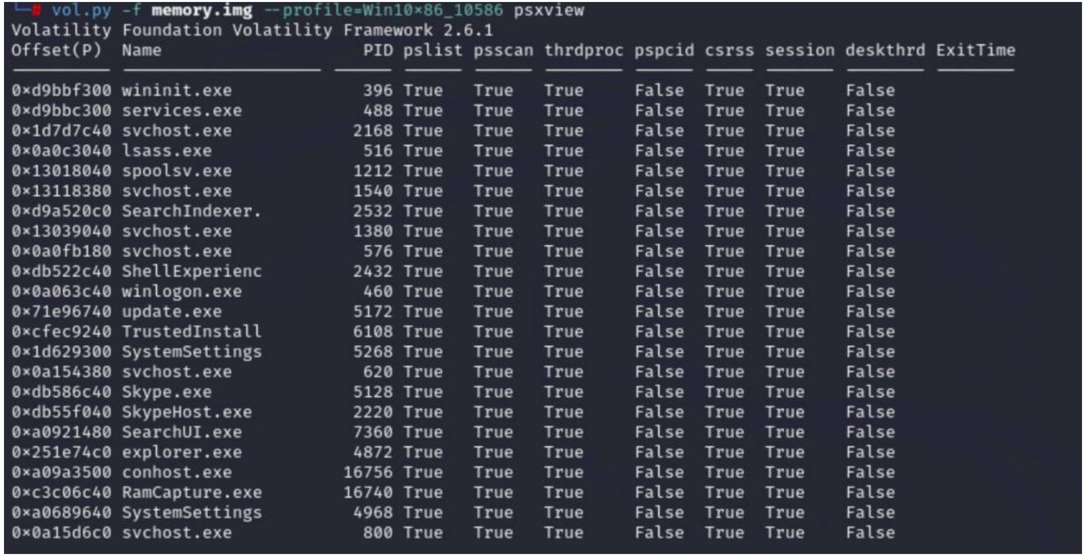
- **Which process are running?**
  - volatility -f MyDump.dmp --profile=MyProfile pstree



```
└─# vol.py -f memory.img --profile=Win10x86_10586 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit

---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ --------------------------
----
0x868a7700 System                    4      0    104        0 ------      0 2016-08-16 12:54:24 UTC+0000

0x8d2af5c0 smss.exe                244      4      2        0 ------      0 2016-08-16 12:54:24 UTC+0000

0x8f7e3040 csrss.exe               324    316     10        0      0      0 2016-08-16 12:54:27 UTC+0000

0x9487c640 smss.exe                388    244      0 --------      1      0 2016-08-16 12:54:28 UTC+0000   2016-08-16 12:54:28 UTC+00
00
0x8b9bf300 wininit.exe             396    316      2        0      0      0 2016-08-16 12:54:28 UTC+0000

0x8f71d2c0 csrss.exe               408    388     11        0      1      0 2016-08-16 12:54:28 UTC+0000

0x94863c40 winlogon.exe            460    388      4        0      1      0 2016-08-16 12:54:28 UTC+0000

0x8b9bc300 services.exe            488    396      6        0      0      0 2016-08-16 12:54:29 UTC+0000

0x948c3040 lsass.exe               516    396      7        0      0      0 2016-08-16 12:54:29 UTC+0000

0x948fb180 svchost.exe             576    488     19        0      0      0 2016-08-16 12:54:30 UTC+0000
```

Volatility

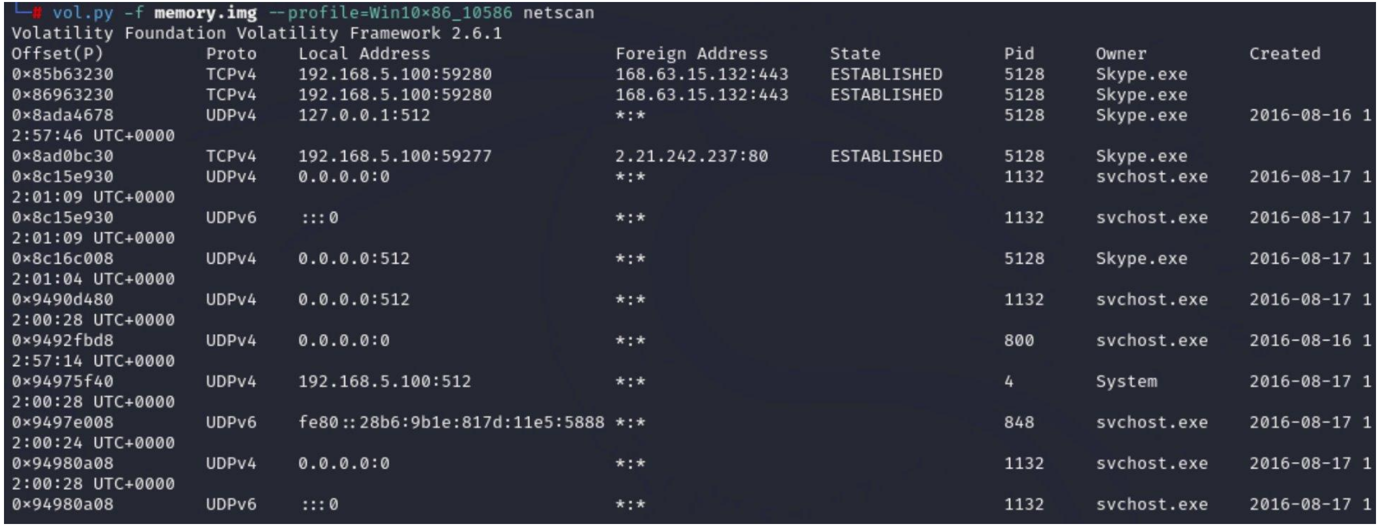
- **Which process are running?**
  - volatility -f MyDump.dmp --profile=MyProfile psxview



```
└─# vol.py -f memory.img --profile=Win10x86_10586 psxview
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd ExitTime
---------- -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --------

0xd9bbf300 wininit.exe             396 True   True   True     False  True  True    False
0xd9bbc300 services.exe            488 True   True   True     False  True  True    False
0x1d7d7c40 svchost.exe            2168 True   True   True     False  True  True    False
0x0a0c3040 lsass.exe               516 True   True   True     False  True  True    False
0x13018040 spoolsv.exe            1212 True   True   True     False  True  True    False
0x13118380 svchost.exe            1540 True   True   True     False  True  True    False
0xd9a520c0 SearchIndexer.         2532 True   True   True     False  True  True    False
0x13039040 svchost.exe            1380 True   True   True     False  True  True    False
0x0a0fb180 svchost.exe             576 True   True   True     False  True  True    False
0xdb522c40 ShellExperienc         2432 True   True   True     False  True  True    False
0x0a063c40 winlogon.exe            460 True   True   True     False  True  True    False
0x71e96740 update.exe             5172 True   True   True     False  True  True    False
0xcfec9240 TrustedInstall         6108 True   True   True     False  True  True    False
0x1d629300 SystemSettings         5268 True   True   True     False  True  True    False
0x0a154380 svchost.exe             620 True   True   True     False  True  True    False
0xdb586c40 Skype.exe              5128 True   True   True     False  True  True    False
0xdb55f040 SkypeHost.exe          2220 True   True   True     False  True  True    False
0xa0921480 SearchUI.exe           7360 True   True   True     False  True  True    False
0x251e74c0 explorer.exe           4872 True   True   True     False  True  True    False
0xa09a3500 conhost.exe           16756 True   True   True     False  True  True    False
0xc3c06c40 RamCapture.exe        16740 True   True   True     False  True  True    False
0xa0689640 SystemSettings         4968 True   True   True     False  True  True    False
0x0a15d6c0 svchost.exe             800 True   True   True     False  True  True    False
```

Volatility

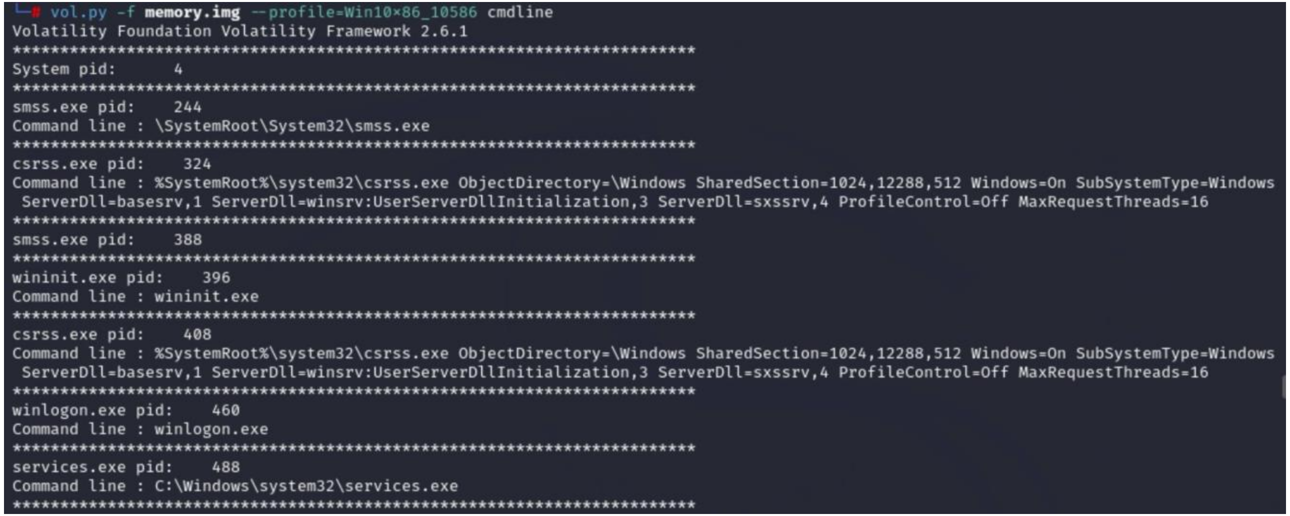
- **List open TCP/UDP connection**
  - volatility -f MyDump.dmp --profile=MyProfile netscan



```
└─# vol.py -f memory.img --profile=Win10x86_10586 netscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)          Proto    Local Address                    Foreign Address      State            Pid      Owner          Created
0x85b63230         TCPv4    192.168.5.100:59280              168.63.15.132:443    ESTABLISHED      5128     Skype.exe
0x86963230         TCPv4    192.168.5.100:59280              168.63.15.132:443    ESTABLISHED      5128     Skype.exe
0x8ada4678         UDPv4    127.0.0.1:512                    *:*                                   5128     Skype.exe      2016-08-16 1
2:57:46 UTC+0000
0x8ad0bc30         TCPv4    192.168.5.100:59277              2.21.242.237:80      ESTABLISHED      5128     Skype.exe
0x8c15e930         UDPv4    0.0.0.0:0                        *:*                                   1132     svchost.exe    2016-08-17 1
2:01:09 UTC+0000
0x8c15e930         UDPv6    :::0                             *:*                                   1132     svchost.exe    2016-08-17 1
2:01:09 UTC+0000
0x8c16c008         UDPv4    0.0.0.0:512                      *:*                                   5128     Skype.exe      2016-08-17 1
2:01:04 UTC+0000
0x9490d480         UDPv4    0.0.0.0:512                      *:*                                   1132     svchost.exe    2016-08-17 1
2:00:28 UTC+0000
0x9492fbd8         UDPv4    0.0.0.0:0                        *:*                                   800      svchost.exe    2016-08-16 1
2:57:14 UTC+0000
0x94975f40         UDPv4    192.168.5.100:512                *:*                                   4        System         2016-08-17 1
2:00:28 UTC+0000
0x9497e008         UDPv6    fe80::28b6:9b1e:817d:11e5:5888 *:*                                     848      svchost.exe    2016-08-17 1
2:00:24 UTC+0000
0x94980a08         UDPv4    0.0.0.0:0                        *:*                                   1132     svchost.exe    2016-08-17 1
2:00:28 UTC+0000
0x94980a08         UDPv6    :::0                             *:*                                   1132     svchost.exe    2016-08-17 1
```

Volatility

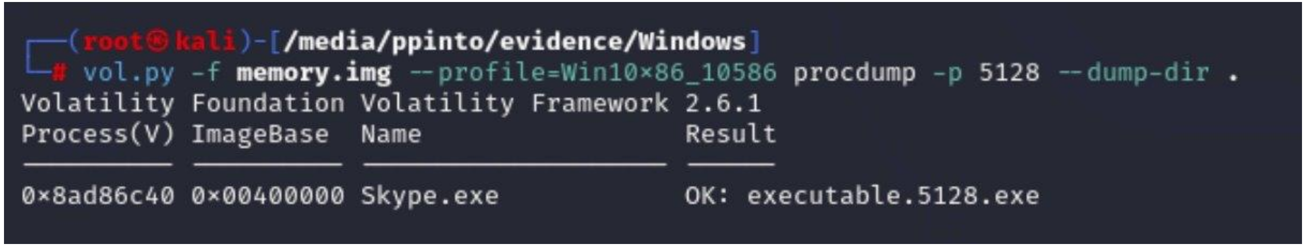
- **What commands were lastly run on the computer**
  - volatility -f MyDump.dmp --profile=MyProfile cmdline



```
└─# vol.py -f memory.img --profile=Win10x86_10586 cmdline
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
System pid:      4
************************************************************************
smss.exe pid:    244
Command line : \SystemRoot\System32\smss.exe
************************************************************************
csrss.exe pid:    324
Command line : %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,12288,512 Windows=On SubSystemType=Windows
 ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=sxssrv,4 ProfileControl=Off MaxRequestThreads=16
************************************************************************
smss.exe pid:    388
************************************************************************
wininit.exe pid:    396
Command line : wininit.exe
************************************************************************
csrss.exe pid:    408
Command line : %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,12288,512 Windows=On SubSystemType=Windows
 ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=sxssrv,4 ProfileControl=Off MaxRequestThreads=16
************************************************************************
winlogon.exe pid:    460
Command line : winlogon.exe
************************************************************************
services.exe pid:    488
Command line : C:\Windows\system32\services.exe
************************************************************************
```

Volatility

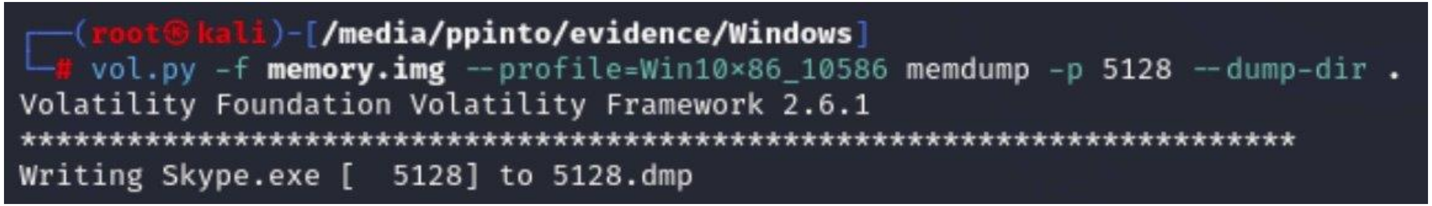
- **Dump processes exe and memory**
  - volatility -f MyDump.dmp --profile=MyProfile procdump -p MyPid --dump-dir



```
┌──(root㉿kali)-[/media/ppinto/evidence/Windows]
└─# vol.py -f memory.img --profile=Win10x86_10586 procdump -p 5128 --dump-dir .
Volatility Foundation Volatility Framework 2.6.1
Process(V) ImageBase  Name                 Result
────────── ────────── ──────────────────── ──────
0x8ad86c40 0x00400000 Skype.exe            OK: executable.5128.exe
```

Volatility

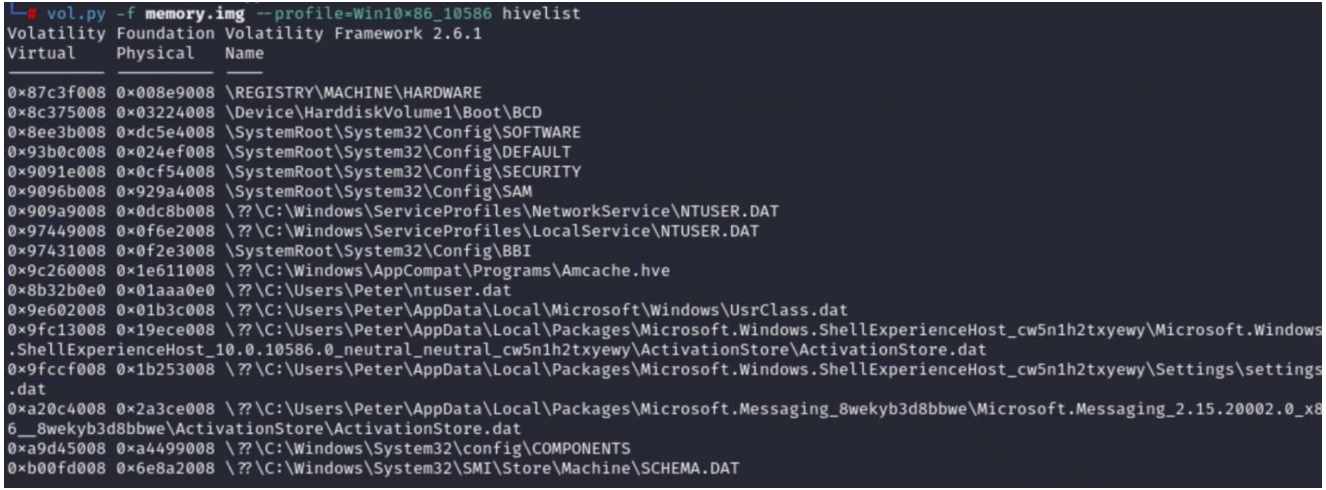
- **Mem Dump processes exe and memory**
  - volatility -f MyDump.dmp --profile=MyProfile memdump -p MyPid --dump-dir .



```
┌──(root㉿kali)-[/media/ppinto/evidence/Windows]
└─# vol.py -f memory.img --profile=Win10x86_10586 memdump -p 5128 --dump-dir .
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
Writing Skype.exe [  5128] to 5128.dmp
```

Volatility

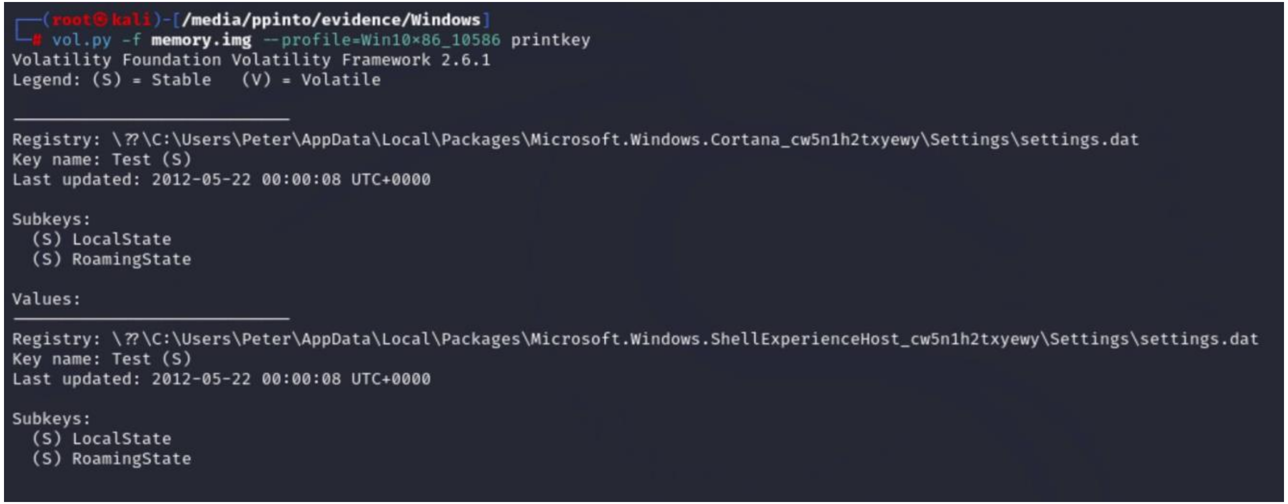
- **Hive and Registry key values**
  - volatility -f MyDump.dmp --profile=MyProfile hivelist



```
└─# vol.py -f memory.img --profile=Win10x86_10586 hivelist
Volatility Foundation Volatility Framework 2.6.1
Virtual    Physical   Name
---------- ---------- ----

0x87c3f008 0x008e9008 \REGISTRY\MACHINE\HARDWARE
0x8c375008 0x03224008 \Device\HarddiskVolume1\Boot\BCD
0x8ee3b008 0xdc5e4008 \SystemRoot\System32\Config\SOFTWARE
0x93b0c008 0x024ef008 \SystemRoot\System32\Config\DEFAULT
0x9091e008 0x0cf54008 \SystemRoot\System32\Config\SECURITY
0x9096b008 0x929a4008 \SystemRoot\System32\Config\SAM
0x909a9008 0x0dc8b008 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0x97449008 0x0f6e2008 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0x97431008 0x0f2e3008 \SystemRoot\System32\Config\BBI
0x9c260008 0x1e611008 \??\C:\Windows\AppCompat\Programs\Amcache.hve
0x8b32b0e0 0x01aaa0e0 \??\C:\Users\Peter\ntuser.dat
0x9e602008 0x01b3c008 \??\C:\Users\Peter\AppData\Local\Microsoft\Windows\UsrClass.dat
0x9fc13008 0x19ece008 \??\C:\Users\Peter\AppData\Local\Packages\Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy\Microsoft.Windows
.ShellExperienceHost_10.0.10586.0_neutral_neutral_cw5n1h2txyewy\ActivationStore\ActivationStore.dat
0x9fccf008 0x1b253008 \??\C:\Users\Peter\AppData\Local\Packages\Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy\Settings\settings
.dat
0xa20c4008 0x2a3ce008 \??\C:\Users\Peter\AppData\Local\Packages\Microsoft.Messaging_8wekyb3d8bbwe\Microsoft.Messaging_2.15.20002.0_x8
6__8wekyb3d8bbwe\ActivationStore\ActivationStore.dat
0xa9d45008 0xa4499008 \??\C:\Windows\System32\config\COMPONENTS
0xb00fd008 0x6e8a2008 \??\C:\Windows\System32\SMI\Store\Machine\SCHEMA.DAT
```

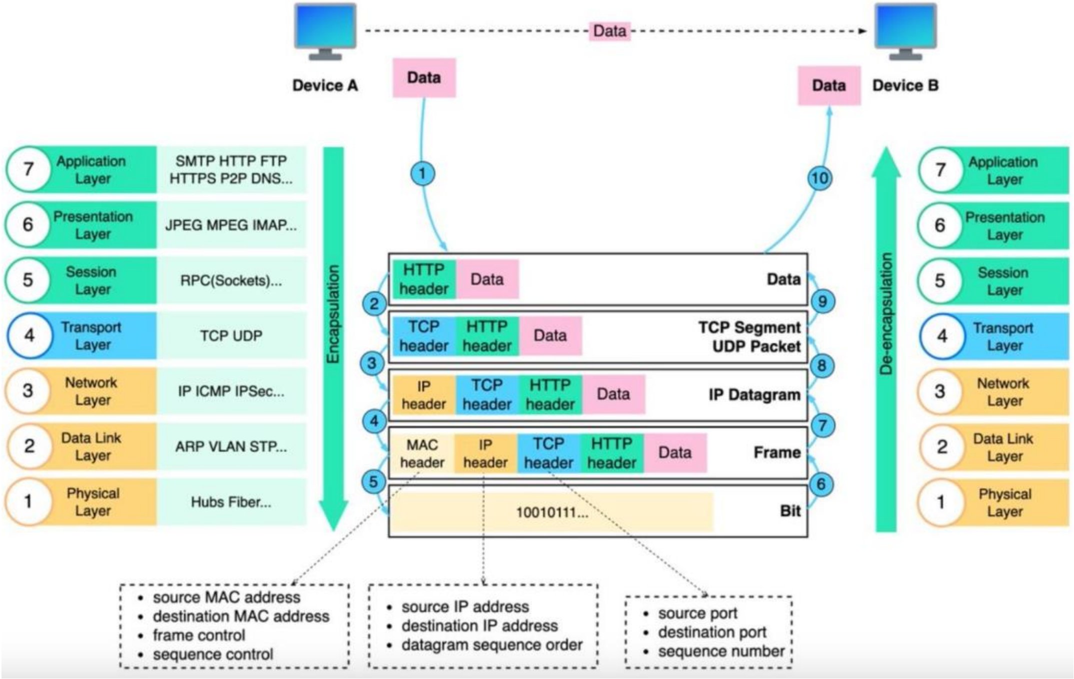
Volatility

- **Hive and Registry key values (printkey)**
  - volatility -f MyDump.dmp --profile=MyProfile printkey



```
┌──(root㉿kali)-[/media/ppinto/evidence/Windows]
└─# vol.py -f memory.img --profile=Win10x86_10586 printkey
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \??\C:\Users\Peter\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\Settings\settings.dat
Key name: Test (S)
Last updated: 2012-05-22 00:00:08 UTC+0000

Subkeys:
  (S) LocalState
  (S) RoamingState

Values:
----------------------------
Registry: \??\C:\Users\Peter\AppData\Local\Packages\Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy\Settings\settings.dat
Key name: Test (S)
Last updated: 2012-05-22 00:00:08 UTC+0000

Subkeys:
  (S) LocalState
  (S) RoamingState
```

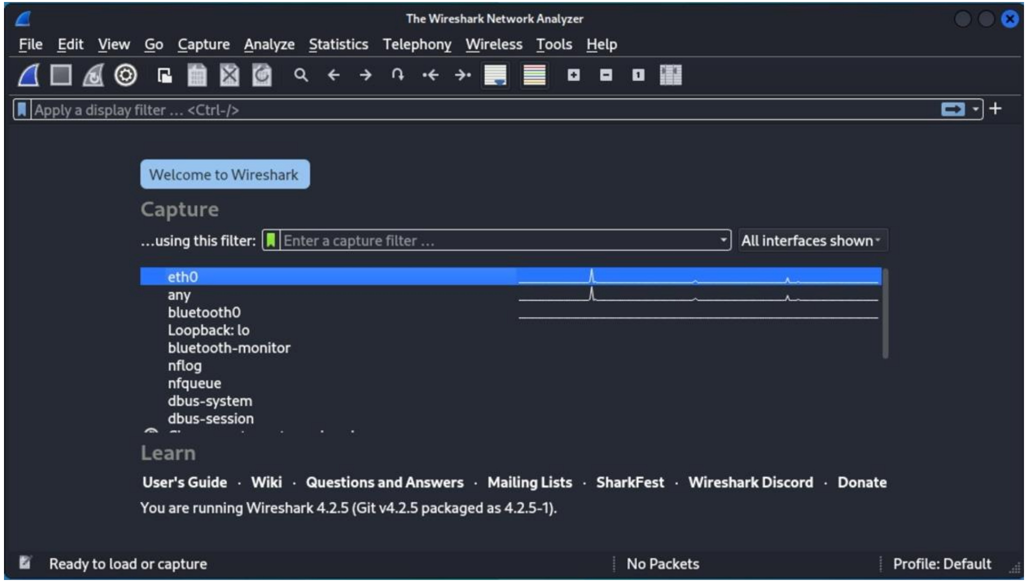
TCP Encapsulation



Wireshark

Wireshark

- Protocol analysis tool that allows real-time capture of network traffic



The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Welcome to Wireshark

Capture

…using this filter: Enter a capture filter … All interfaces shown

eth0
any
bluetooth0
Loopback: lo
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session

Learn

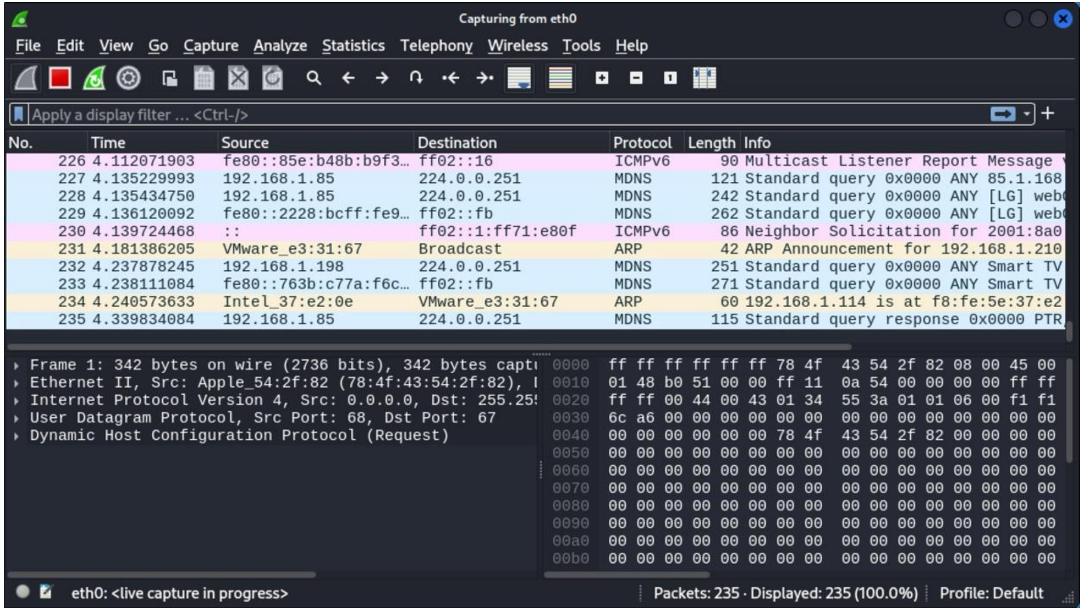
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

You are running Wireshark 4.2.5 (Git v4.2.5 packaged as 4.2.5-1).

Ready to load or capture | No Packets | Profile: Default

Wireshark

▪ Wireshark – Capturing Packets


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.     Time         Source                 Destination        Protocol  Length Info
    226 4.112071903  fe80::85e:b48b:b9f3…   ff02::16           ICMPv6        90 Multicast Listener Report Message
    227 4.135229993  192.168.1.85           224.0.0.251        MDNS         121 Standard query 0x0000 ANY 85.1.168
    228 4.135434750  192.168.1.85           224.0.0.251        MDNS         242 Standard query 0x0000 ANY [LG] web
    229 4.136120092  fe80::2228:bcff:fe9…   ff02::fb           MDNS         262 Standard query 0x0000 ANY [LG] web
    230 4.139724468  ::                     ff02::1:ff71:e80f  ICMPv6        86 Neighbor Solicitation for 2001:8a0
    231 4.181386205  VMware_e3:31:67        Broadcast          ARP           42 ARP Announcement for 192.168.1.210
    232 4.237878245  192.168.1.198          224.0.0.251        MDNS         251 Standard query 0x0000 ANY Smart TV
    233 4.238111084  fe80::763b:c77a:f6c…   ff02::fb           MDNS         271 Standard query 0x0000 ANY Smart TV
    234 4.240573633  Intel_37:e2:0e         VMware_e3:31:67    ARP           60 192.168.1.114 is at f8:fe:5e:37:e2
    235 4.339834084  192.168.1.85           224.0.0.251        MDNS         115 Standard query response 0x0000 PTR
```

```
› Frame 1: 342 bytes on wire (2736 bits), 342 bytes capt
› Ethernet II, Src: Apple_54:2f:82 (78:4f:43:54:2f:82),
› Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.25
› User Datagram Protocol, Src Port: 68, Dst Port: 67
› Dynamic Host Configuration Protocol (Request)
```

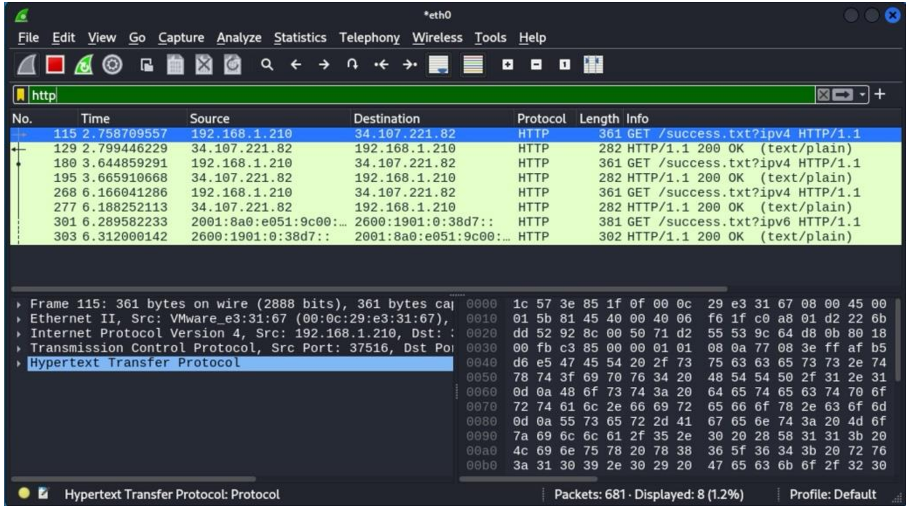
```
0000  ff ff ff ff ff ff 78 4f  43 54 2f 82 08 00 45 00
0010  01 48 b0 51 00 00 ff 11  0a 54 00 00 00 00 ff ff
0020  ff ff 00 44 00 43 01 34  55 3a 01 01 06 00 f1 f1
0030  6c a6 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0040  00 00 00 00 00 00 78 4f  43 54 2f 82 00 00 00 00
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

eth0: <live capture in progress>

Packets: 235 · Displayed: 235 (100.0%) | Profile: Default

Wireshark

- **Filters**
  - HTTP
  - DNS
  - ip.address == 192.168.0.1
  - tcp.port == 22
  - tcp.port == 80 || udp.port == 80
  - ip.src == 192.168.1.1
  - ip.dst == 192.168.1.1
  - ip.addr == 192.168.1.1 && http
  - tcp contains "GET



http

```
No.   Time         Source               Destination          Protocol Length Info
  115 2.758709557  192.168.1.210        34.107.221.82        HTTP        361 GET /success.txt?ipv4 HTTP/1.1
  129 2.799446229  34.107.221.82        192.168.1.210        HTTP        282 HTTP/1.1 200 OK  (text/plain)
  180 3.644859291  192.168.1.210        34.107.221.82        HTTP        361 GET /success.txt?ipv4 HTTP/1.1
  195 3.665910668  34.107.221.82        192.168.1.210        HTTP        282 HTTP/1.1 200 OK  (text/plain)
  268 6.166041286  192.168.1.210        34.107.221.82        HTTP        361 GET /success.txt?ipv4 HTTP/1.1
  277 6.188252113  34.107.221.82        192.168.1.210        HTTP        282 HTTP/1.1 200 OK  (text/plain)
  301 6.289582233  2001:8a0:e051:9c00:… 2600:1901:0:38d7::   HTTP        381 GET /success.txt?ipv6 HTTP/1.1
  303 6.312000142  2600:1901:0:38d7::   2001:8a0:e051:9c00:… HTTP        302 HTTP/1.1 200 OK  (text/plain)
```

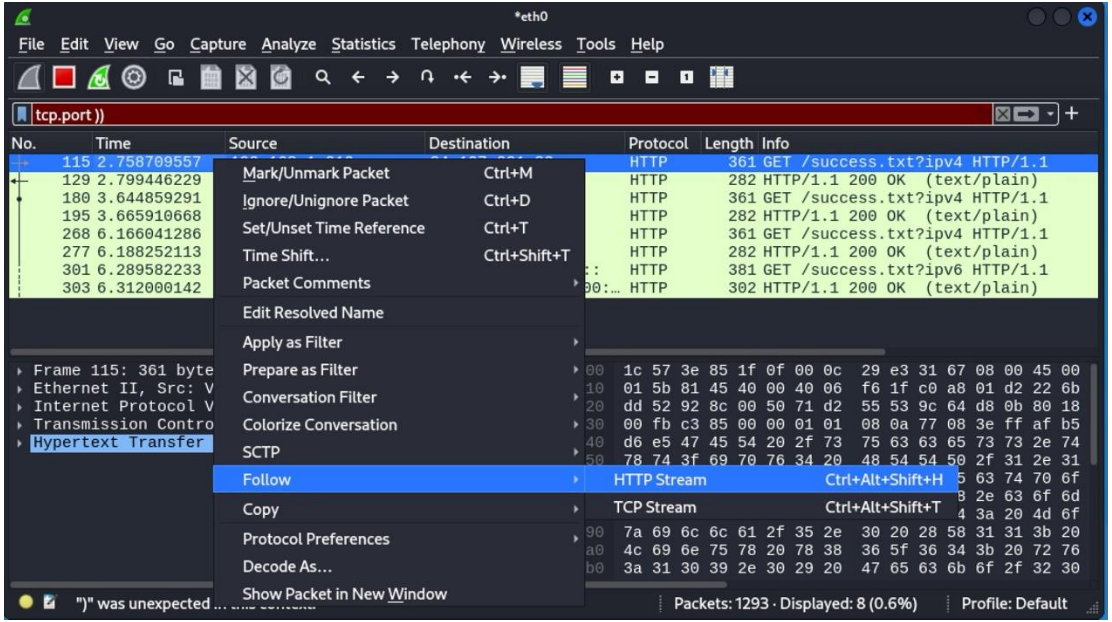
```
› Frame 115: 361 bytes on wire (2888 bits), 361 bytes ca
› Ethernet II, Src: VMware_e3:31:67 (00:0c:29:e3:31:67),
› Internet Protocol Version 4, Src: 192.168.1.210, Dst: :
› Transmission Control Protocol, Src Port: 37516, Dst Po
› Hypertext Transfer Protocol
```

```
0000  1c 57 3e 85 1f 0f 00 0c  29 e3 31 67 08 00 45 00
0010  01 5b 81 45 40 00 40 06  f6 1f c0 a8 01 d2 22 6b
0020  dd 52 92 8c 00 50 71 d2  55 53 9c 64 d8 0b 80 18
0030  00 fb c3 85 00 00 01 01  08 0a 77 08 3e ff af b5
0040  d6 e5 47 45 54 20 2f 73  75 63 63 65 73 73 2e 74
0050  78 74 3f 69 70 76 34 20  48 54 54 50 2f 31 2e 31
0060  0d 0a 48 6f 73 74 3a 20  64 65 74 65 63 74 70 6f
0070  72 74 61 6c 2e 66 69 72  65 66 6f 78 2e 63 6f 6d
0080  0d 0a 55 73 65 72 2d 41  67 65 6e 74 3a 20 4d 6f
0090  7a 69 6c 6c 61 2f 35 2e  30 20 28 58 31 31 3b 20
00a0  4c 69 6e 75 78 20 78 38  36 5f 36 34 3b 20 72 76
00b0  3a 31 30 39 2e 30 29 20  47 65 63 6b 6f 2f 32 30
```

Hypertext Transfer Protocol: Protocol | Packets: 681 · Displayed: 8 (1.2%) | Profile: Default

Wireshark

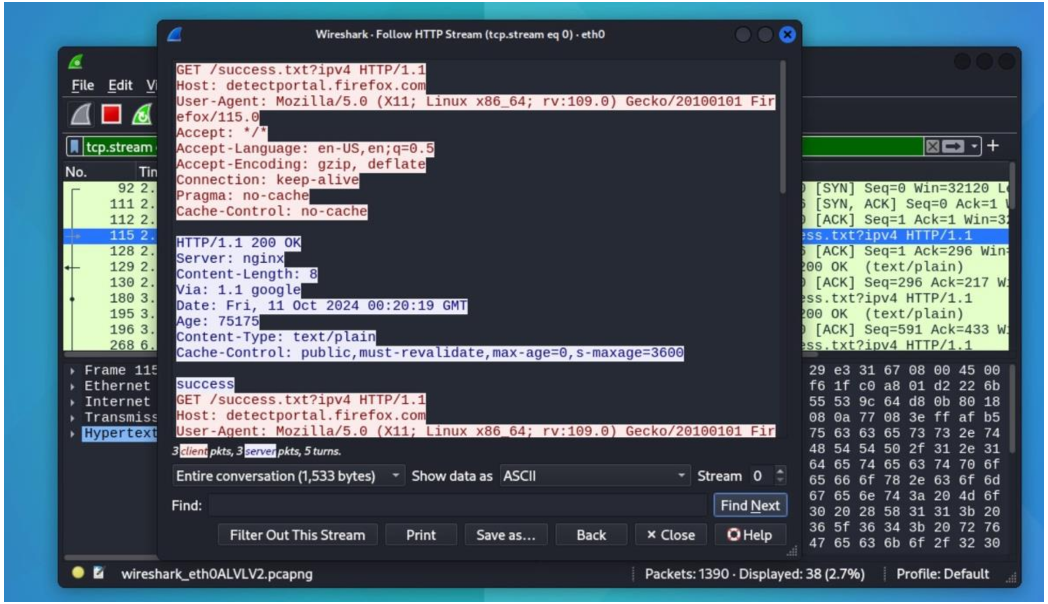
- ## Follow HTTP Stream



```
tcp.port ))

No.   Time          Source    Destination   Protocol  Length  Info
  115 2.758709557                           HTTP         361  GET /success.txt?ipv4 HTTP/1.1
  129 2.799446229                           HTTP         282  HTTP/1.1 200 OK  (text/plain)
  180 3.644859291                           HTTP         361  GET /success.txt?ipv4 HTTP/1.1
  195 3.665910668                           HTTP         282  HTTP/1.1 200 OK  (text/plain)
  268 6.166041286                           HTTP         361  GET /success.txt?ipv4 HTTP/1.1
  277 6.188252113                           HTTP         282  HTTP/1.1 200 OK  (text/plain)
  301 6.289582233               ::          HTTP         381  GET /success.txt?ipv6 HTTP/1.1
  303 6.312000142               00:…        HTTP         302  HTTP/1.1 200 OK  (text/plain)

▸ Frame 115: 361 byte      00  1c 57 3e 85 1f 0f 00 0c  29 e3 31 67 08 00 45 00
▸ Ethernet II, Src: V      10  01 5b 81 45 40 00 40 06  f6 1f c0 a8 01 d2 22 6b
▸ Internet Protocol V      20  dd 52 92 8c 00 50 71 d2  55 53 9c 64 d8 0b 80 18
▸ Transmission Contro      30  00 fb c3 85 00 00 01 01  08 0a 77 08 3e ff af b5
▸ Hypertext Transfer       40  d6 e5 47 45 54 20 2f 73  75 63 63 65 73 73 2e 74
                           50  78 74 3f 69 70 76 34 20  48 54 54 50 2f 31 2e 31
                                                        5 63 74 70 6f
                                                        3 2e 63 6f 6d
                                                        4 3a 20 4d 6f
                           90  7a 69 6c 6c 61 2f 35 2e  30 20 28 58 31 31 3b 20
                           a0  4c 69 6e 75 78 20 78 38  36 5f 36 34 3b 20 72 76
                           b0  3a 31 30 39 2e 30 29 20  47 65 63 6b 6f 2f 32 30

")" was unexpected …          Packets: 1293 · Displayed: 8 (0.6%)     Profile: Default
```

Wireshark

- ## Follow HTTP Stream (2)



Wireshark · Follow HTTP Stream (tcp.stream eq 0) · eth0

```
GET /success.txt?ipv4 HTTP/1.1
Host: detectportal.firefox.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Fir
efox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

HTTP/1.1 200 OK
Server: nginx
Content-Length: 8
Via: 1.1 google
Date: Fri, 11 Oct 2024 00:20:19 GMT
Age: 75175
Content-Type: text/plain
Cache-Control: public,must-revalidate,max-age=0,s-maxage=3600

success
GET /success.txt?ipv4 HTTP/1.1
Host: detectportal.firefox.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Fir
```

3 client pkts, 3 server pkts, 5 turns.

Entire conversation (1,533 bytes) · Show data as ASCII · Stream 0

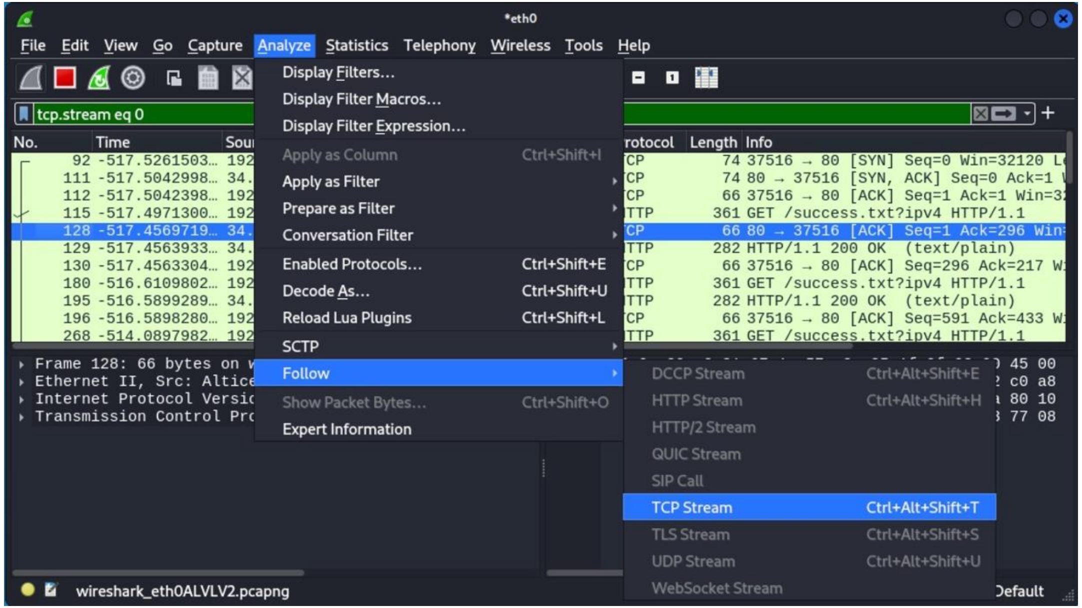
Find:

```
 92 2.      ) [SYN] Seq=0 Win=32120 L
111 2.      5 [SYN, ACK] Seq=0 Ack=1 W
112 2.      ) [ACK] Seq=1 Ack=1 Win=3
115 2.      ess.txt?ipv4 HTTP/1.1
128 2.      5 [ACK] Seq=1 Ack=296 Win=
129 2.      200 OK  (text/plain)
130 2.      ) [ACK] Seq=296 Ack=217 W
180 3.      ess.txt?ipv4 HTTP/1.1
195 3.      200 OK  (text/plain)
196 3.      ) [ACK] Seq=591 Ack=433 W
268 6.      ess.txt?ipv4 HTTP/1.1

▸ Frame 115       29 e3 31 67 08 00 45 00
▸ Ethernet        f6 1f c0 a8 01 d2 22 6b
▸ Internet        55 53 9c 64 d8 0b 80 18
▸ Transmiss       08 0a 77 08 3e ff af b5
▸ Hypertext       75 63 63 65 73 73 2e 74
                  48 54 54 50 2f 31 2e 31
                  64 65 74 65 63 74 70 6f
                  65 66 6f 78 2e 63 6f 6d
                  67 65 6e 74 3a 20 4d 6f
                  30 20 28 58 31 31 3b 20
                  36 5f 36 34 3b 20 72 76
                  47 65 63 6b 6f 2f 32 30
```

wireshark_eth0ALVLV2.pcapng · Packets: 1390 · Displayed: 38 (2.7%) · Profile: Default

Wireshark

- **Follow TCP Stream**



```
tcp.stream eq 0
```

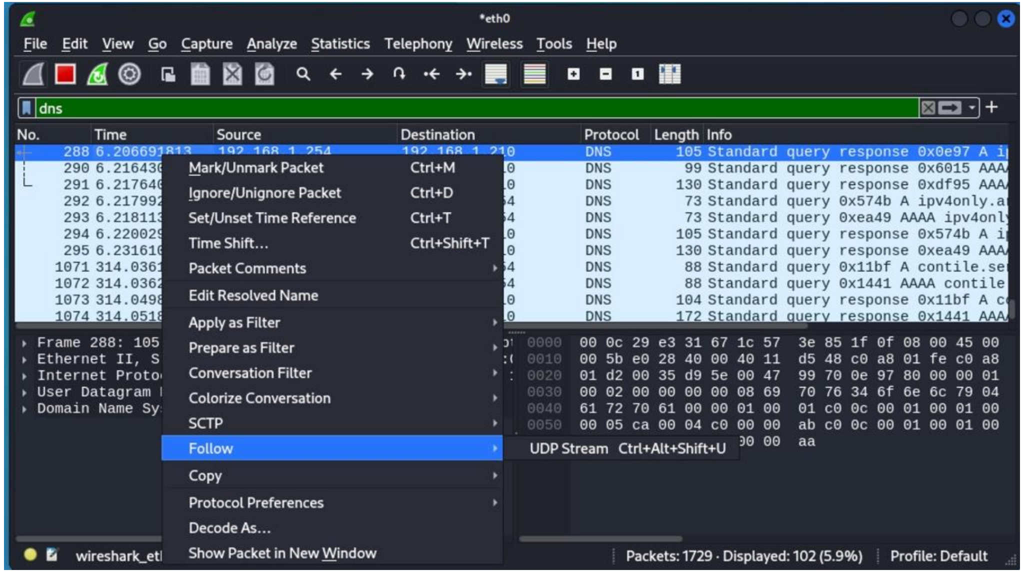
```
No.    Time            Sou           rotocol  Length  Info
    92 -517.5261503…  192            CP           74  37516 → 80 [SYN] Seq=0 Win=32120 L
   111 -517.5042998…  34.            CP           74  80 → 37516 [SYN, ACK] Seq=0 Ack=1 W
   112 -517.5042398…  192            CP           66  37516 → 80 [ACK] Seq=1 Ack=1 Win=3
   115 -517.4971300…  192            TTP         361  GET /success.txt?ipv4 HTTP/1.1
   128 -517.4569719…  34.            CP           66  80 → 37516 [ACK] Seq=1 Ack=296 Win
   129 -517.4563933…  34.            TTP         282  HTTP/1.1 200 OK  (text/plain)
   130 -517.4563304…  192            CP           66  37516 → 80 [ACK] Seq=296 Ack=217 W
   180 -516.6109802…  192            TTP         361  GET /success.txt?ipv4 HTTP/1.1
   195 -516.5899289…  34.            TTP         282  HTTP/1.1 200 OK  (text/plain)
   196 -516.5898280…  192            CP           66  37516 → 80 [ACK] Seq=591 Ack=433 W
   268 -514.0897982…  192            TTP         361  GET /success.txt?ipv4 HTTP/1.1
```

```
› Frame 128: 66 bytes on w                          45 00
› Ethernet II, Src: Altic                           c0 a8
› Internet Protocol Versic                          80 10
› Transmission Control Prc                          77 08
```

*eth0
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Display Filters…
Display Filter Macros…
Display Filter Expression…
Apply as Column Ctrl+Shift+I
Apply as Filter
Prepare as Filter
Conversation Filter
Enabled Protocols… Ctrl+Shift+E
Decode As… Ctrl+Shift+U
Reload Lua Plugins Ctrl+Shift+L
SCTP
Follow
Show Packet Bytes… Ctrl+Shift+O
Expert Information
DCCP Stream Ctrl+Alt+Shift+E
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream
SIP Call
TCP Stream Ctrl+Alt+Shift+T
TLS Stream Ctrl+Alt+Shift+S
UDP Stream Ctrl+Alt+Shift+U
WebSocket Stream
wireshark_eth0ALVLV2.pcapng
Default

Wireshark

- **Follow UDP STream**



dns

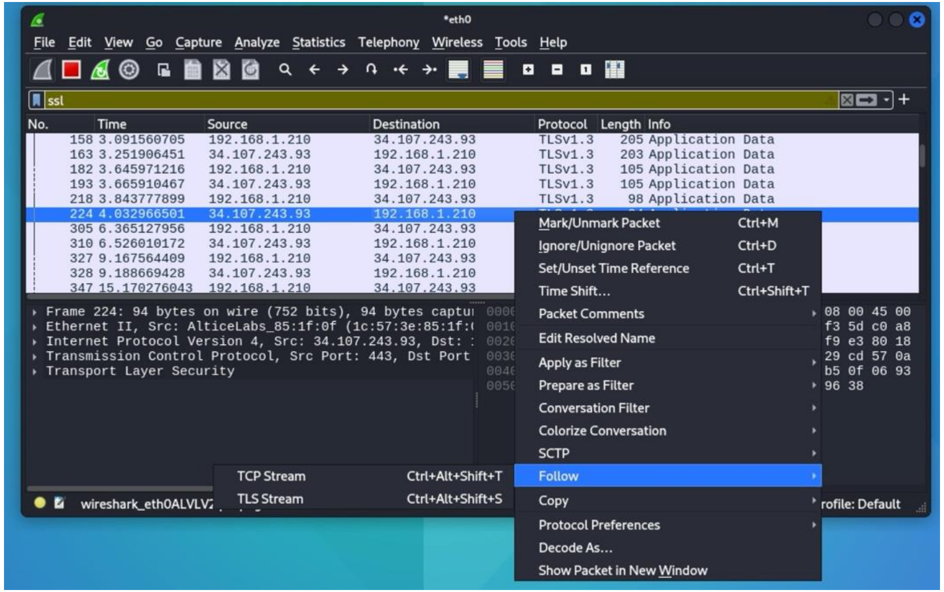
```
No.    Time          Source          Destination     Protocol  Length Info
  288 6.206691813   192.168.1.254   192.168.1.210   DNS        105 Standard query response 0x0e97 A i
  290 6.21643                                   0   DNS         99 Standard query response 0x6015 AAA
  291 6.21764                                   0   DNS        130 Standard query response 0xdf95 AAA
  292 6.21799                                   4   DNS         73 Standard query 0x574b A ipv4only.a
  293 6.21811                                   4   DNS         73 Standard query 0xea49 AAAA ipv4only
  294 6.22002                                   0   DNS        105 Standard query response 0x574b A i
  295 6.23161                                   0   DNS        130 Standard query response 0xea49 AAA
 1071 314.0361                                  4   DNS         88 Standard query 0x11bf A contile.se
 1072 314.0362                                  4   DNS         88 Standard query 0x1441 AAAA contile
 1073 314.0498                                  0   DNS        104 Standard query response 0x11bf A c
 1074 314.0518                                  0   DNS        172 Standard query response 0x1441 AAA
```

```
› Frame 288: 105              0000  00 0c 29 e3 31 67 1c 57  3e 85 1f 0f 08 00 45 00
› Ethernet II, S              0010  00 5b e0 28 40 00 40 11  d5 48 c0 a8 01 fe c0 a8
› Internet Proto              0020  01 d2 00 35 d9 5e 00 47  99 70 0e 97 80 00 00 01
› User Datagram               0030  00 02 00 00 00 00 08 69  70 76 34 6f 6e 6c 79 04
› Domain Name Sy              0040  61 72 70 61 00 00 01 00  01 c0 0c 00 01 00 01 00
                              0050  00 05 ca 00 04 c0 00 00  ab c0 0c 00 01 00 01 00
                                                       00 00  aa
```

wireshark_etl | Packets: 1729 · Displayed: 102 (5.9%) | Profile: Default

Wireshark

- ## Follow TLS STream



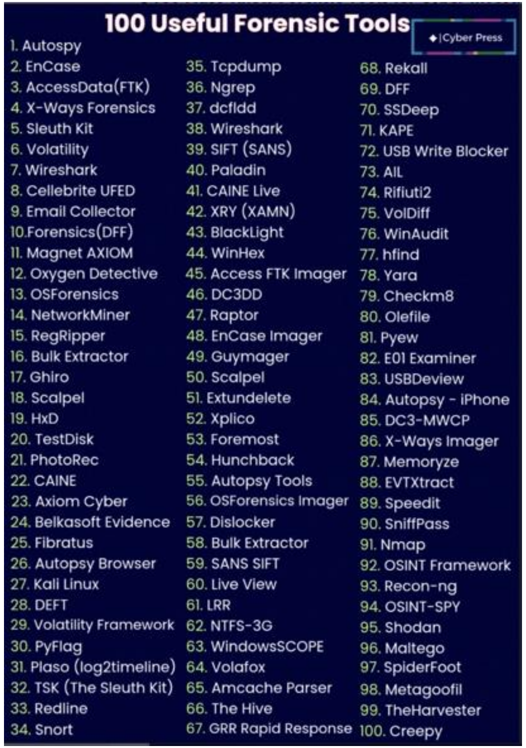
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ssl

```
No.    Time          Source          Destination     Protocol  Length Info
   158 3.091560705   192.168.1.210   34.107.243.93   TLSv1.3   205 Application Data
   163 3.251906451   34.107.243.93   192.168.1.210   TLSv1.3   203 Application Data
   182 3.645971216   192.168.1.210   34.107.243.93   TLSv1.3   105 Application Data
   193 3.665910467   34.107.243.93   192.168.1.210   TLSv1.3   105 Application Data
   218 3.843777899   192.168.1.210   34.107.243.93   TLSv1.3    98 Application Data
   224 4.032966501   34.107.243.93   192.168.1.210
   305 6.365127956   192.168.1.210   34.107.243.93
   310 6.526010172   34.107.243.93   192.168.1.210
   327 9.167564409   192.168.1.210   34.107.243.93
   328 9.188669428   34.107.243.93   192.168.1.210
   347 15.170276043  192.168.1.210   34.107.243.93
```

```
▸ Frame 224: 94 bytes on wire (752 bits), 94 bytes captu
▸ Ethernet II, Src: AlticeLabs_85:1f:0f (1c:57:3e:85:1f:(
▸ Internet Protocol Version 4, Src: 34.107.243.93, Dst: :
▸ Transmission Control Protocol, Src Port: 443, Dst Port
▸ Transport Layer Security
```

```
000  08 00 45 00
001  f3 5d c0 a8
002  f9 e3 80 18
003  29 cd 57 0a
004  b5 0f 06 93
005  96 38
```

Forensic Tools



100 Useful Forensic Tools

Cyber Press

1. Autospy
2. EnCase
3. AccessData(FTK)
4. X-Ways Forensics
5. Sleuth Kit
6. Volatility
7. Wireshark
8. Cellebrite UFED
9. Email Collector
10.Forensics(DFF)
11. Magnet AXIOM
12. Oxygen Detective
13. OSForensics
14. NetworkMiner
15. RegRipper
16. Bulk Extractor
17. Ghiro
18. Scalpel
19. HxD
20. TestDisk
21. PhotoRec
22. CAINE
23. Axiom Cyber
24. Belkasoft Evidence
25. Fibratus
26. Autopsy Browser
27. Kali Linux
28. DEFT
29. Volatility Framework
30. PyFlag
31. Plaso (log2timeline)
32. TSK (The Sleuth Kit)
33. Redline
34. Snort
35. Tcpdump
36. Ngrep
37. dcfldd
38. Wireshark
39. SIFT (SANS)
40. Paladin
41. CAINE Live
42. XRY (XAMN)
43. BlackLight
44. WinHex
45. Access FTK Imager
46. DC3DD
47. Raptor
48. EnCase Imager
49. Guymager
50. Scalpel
51. Extundelete
52. Xplico
53. Foremost
54. Hunchback
55. Autopsy Tools
56. OSForensics Imager
57. Dislocker
58. Bulk Extractor
59. SANS SIFT
60. Live View
61. LRR
62. NTFS-3G
63. WindowsSCOPE
64. Volafox
65. Amcache Parser
66. The Hive
67. GRR Rapid Response
68. Rekall
69. DFF
70. SSDeep
71. KAPE
72. USB Write Blocker
73. AIL
74. Rifiuti2
75. VolDiff
76. WinAudit
77. hfind
78. Yara
79. Checkm8
80. Olefile
81. Pyew
82. E01 Examiner
83. USBDeview
84. Autopsy - iPhone
85. DC3-MWCP
86. X-Ways Imager
87. Memoryze
88. EVTXtract
89. Speedit
90. SniffPass
91. Nmap
92. OSINT Framework
93. Recon-ng
94. OSINT-SPY
95. Shodan
96. Maltego
97. SpiderFoot
98. Metagoofil
99. TheHarvester
100. Creepy