Sardar Vallabhbhai Patel
National Police Academy,
Hyderabad

Cyber Crime Investigation Manual

Volume - IV

CYBER X

Foreword

Cybercrime is one of the biggest challenges we face today. In the past decade, as technology has grown at an incredible pace, so has our dependence on the internet. While this has improved our lives in countless ways, it has also created new opportunities for criminals. From disrupting critical infrastructure to stealing financial assets and sensitive data, cybercrimes can cause serious harm. What makes it even more alarming is how easy and rewarding these crimes can be, often happening across borders without much cost.

Technology has brought great opportunities but also increased our vulnerability to cyber threats. As cybercrimes grow more frequent and complex, the lack of trained professionals to handle such cases effectively is a major challenge. The shortage of skilled officers leads to delays and unresolved cases, highlighting the need for stronger efforts to build a capable workforce to combat these threats efficiently and on time.

At the Sardar Vallabhbhai Patel National Police Academy (SVPNPA), we've been working hard to bridge this gap. Through our CyberX unit (previously NDCRTC), we've trained over 15,000 officers and staff since 2015. These officers are now better equipped to handle the complexities of cybercrime investigations.

To further support our investigators, the CyberX unit has developed five comprehensive manuals. These manuals are designed to be practical, user-friendly guides to help officers navigate the often-complicated process of cybercrime investigations. They focus on bridging the knowledge and skill gaps, offering clear and actionable insights.

I strongly encourage all investigators to use these manuals to their full advantage. They cover the latest tools and techniques, providing the confidence and clarity needed to take on even the most challenging cases. Together, we can make significant progress in the fight against cybercrime and ensure justice in this ever-changing digital world.

Amit Garg, IPS
Director
Sardar Vallabhbhai Patel
National Police Academy

Contributors:

Mohammed Arif Ali Khan:

Mohammed Arif Ali Khan is working as Chief Forensic Analyst at SVPNPA. He has a decade long experience in capacity building in cyber-crime investigation and digital forensics. He has also worked with the Cyber Crimes Cell, CID Hyderabad and specializes in solving cases related to online harassment, job frauds, fake websites, etc. His interest in Cyber Security was rewarded by companies like Indeed.com, AT&T, Mail.ru for finding security vulnerabilities in their services.

Parmesh Naik:

Parmesh Naik is Senior Forensic Analyst at SVPNPA with over eight years of experience in training law enforcement personnel, specializing in OSINT, Linux forensics, and Malware analysis. His profound understanding of digital forensics is demonstrated through the innovative software tools he has developed, which have become essential in law enforcement investigations.

Shaik Ghousal Mubarak:

Shaik Ghousal Mubarak is working as a Senior Forensic Analyst at SVPNPA. He holds a vast experience of 10 years in the domain of cybercrime investigation.
He previously worked as a cyber-crime consultant at CID Cyber Crimes Hyderabad. He is holding a PG-Diploma in Advance Computing and a B-Tech in Computer Science. His area of interest is Financial Fraud Investigations. Additionally, he is a regular guest speaker at various Police academies, Central Agencies, and other institutions.

Nitin Sharma:

Nitin Sharma is working as the Lead Forensic Analyst at SVPNPA, he imparts training to law enforcement officers and specializes in areas such as Internet Crime Investigation, Cryptocurrency Investigation & Digital Forensics. He holds a PG diploma in Cyber Law & Cyber Forensics from NLSIU Bangalore and an M-Tech in Cyber Security from Gujarat Forensic Sciences University. His extensive experience includes assisting field officers in cases ranging from Internet crimes to Dark Web & Cryptocurrency investigations for agencies like NIA, NCRB, Punjab Police, and others.

Aishwarya Tiwari:

Aishwarya Tiwari is a Forensic Analyst in NDCRTC with four years of specialized experience in training law enforcement agencies and conducting research in cryptocurrency investigation. Aishwarya's expertise is further solidified by a CHFI Certification, a CEH Certification from EC Council, and a Blockchain and Cryptocurrency Diploma from Oxford, London. Aishwarya, continues to make

significant contributions to cyber forensics and security, driven by a steadfast commitment to innovation and excellence in protecting digital assets and mitigating cyber threats.

Priya Ghurde:

Priya Ghurde currently holds the position of 'Cyber Investigation and Forensic Specialist' at the Indian Cyber Crime Coordination Centre (I4C), cryptocurrency-related offenses. Prior to her tenure at I4C, she served as Lead Forensic Analyst at SVPNPA. She has total experience of six years in the field of Cyber Crime Investigation and Cyber Security. She imparts training to law enforcement officers and specializes in areas such as Internet Crime Investigation, Dark web Monitoring & Digital Forensics. She holds B-Tech Degree in Information Technology along with certifications including Cyber Shiksha from Microsoft and CHFI from EC-Council. Her extensive experience includes assisting field officers in cases ranging from Dark Web related investigations to Digital Forensic Investigations for agencies like NIA, NCRB, Punjab Police, and others.

Ashmit Sharma:

Ashmit Sharma, presently serving as Scientist 'B' (Forensic Electronics) at CFSL, DFSS, MHA, GoI (Bhopal) previously as Lead Forensic Analyst at SVPNPA. He is a seasoned professional with expertise in digital forensics. Armed with B-Tech in ECE and an MSc in Forensic Science, Ashmit has honed his skills across various prestigious organizations including RFSL (NR, Dharamshala, HP), CFL (State Crime Branch, Haryana), and CFDML(SFIO). His dedication to continuous learning is evident through his publication of two international papers, focusing on smartphone and WhatsApp vulnerabilities, further establishing his reputation as an avid learner in the field

Mohammed Nazim:

Mohammed Nazim is working as a Forensic Analyst at SVPNPA, equipped with a Computer Science Engineering background and accreditation as an Information Security Management Systems Auditor (ISO 27001). Specializing in CDR/IPDR analysis and fueled by a fervour for Internet Governance, Nazim extends his expertise generously to esteemed institutions such as police academies, NIA, Central Detective Training Institute, and ESCI

Contents

1. Malware and Network Forensics

2. Dark Web and Cryptocurrency

3. Advance Digital Forensics

1.Malware and Network Forensics

1. Introduction to Malware Analysis

The number of cyber-attacks is undoubtedly on the rise, targeting government, military, public and private sectors. These cyber-attacks focus on targeting individuals or organizations with an effort to extract valuable information. Sometimes, these cyber-attacks are allegedly linked to cybercrime or state-sponsored groups, but may also be carried out by individual groups to achieve their goals. Most of these cyber-attacks use malicious software (also called malware) to infect their targets. Knowledge, skills, and tools required to analyze malicious software are essential to detect, investigate and defend against such attacks.

What Is Malware?

Malware is a code that performs malicious actions; it can take the form of an executable, script, code, or any other software. Attackers use malware to steal sensitive information, spy on the infected system, or take control of the system. It typically gets into your system without your consent and can be delivered via various communication channels such as email, web, or USB drives.

The following are some of the malicious actions performed by malware:

- Disrupting computer operations
- Stealing sensitive information, including personal, business, and financial data
- Unauthorized access to the victim's system
- Spying on the victims
- Sending spam emails
- Engaging in distributed-denial-of-service attacks (DDOS)
- Locking up the files on the computer and holding them for ransom

Malware is a broad term that refers to different types of malicious programs such as trojans, viruses, worms, and rootkits. While performing malware analysis, you will often come across various types of malicious programs; some of these malicious programs are categorized based on their functionality and attack vectors as mentioned here:

- **Virus or Worm**: Malware that is capable of copying itself and spreading to other computers. A virus needs user intervention, whereas a worm can spread without user intervention.
- **Trojan**: Malware that disguises itself as a regular program to trick users to install it on their systems. Once installed, it can perform malicious actions such as stealing sensitive data, uploading files to the attacker's server, or monitoring webcams.
- **Backdoor / Remote Access Trojan (RAT):** This is a type of Trojan that enables the attacker to gain access to and execute commands on the compromised system.
- **Adware**: Malware that presents unwanted advertisements (ads) to the user. They usually get delivered via free downloads and can forcibly install software on your system.
- **Botnet:** This is a group of computers infected with the same malware (called bots), waiting to receive instructions from the command-and-control server controlled by the attacker. The attacker can then issue a command to these bots, which can perform malicious activities such as DDOS attacks or sending spam emails.

- **Information stealer:** Malware designed to steal sensitive data such as banking credentials or typed keystrokes from the infected system. Some examples of these malicious programs include key loggers, spyware, sniffers, and form grabbers.
- **Ransomware:** Malware that holds the system for ransom by locking users out of their computer or by encrypting their files.
- **Rootkit:** Malware that provides the attacker with privileged access to the infected system and conceals its presence or the presence of other software.
- **Downloader or dropper**: Malware designed to download or install additional malware components.

What Is Malware Analysis?

Malware analysis is the study of malware's behaviour. The objective of malware analysis is to understand the working of malware and how to detect and eliminate it. It involves analyzing the suspect file in a safe environment to identify its characteristics and functionalities so that better defenses can be built to protect an organization's network.

Why Malware Analysis?

The primary motive behind performing malware analysis is to extract information from the malware sample, which can help in responding to a malware incident. The goal of malware analysis is to determine the capability of malware, detect it, and contain it. It also helps in determining identifiable patterns that can be used to cure and prevent future infections. The following are some of the reasons why you will perform malware analysis:

- To determine the nature and purpose of the malware. For example, it can help you determine whether malware is an information stealer, HTTP bot, spam bot, rootkit, keylogger, or RAT, and so on.
- To gain an understanding of how the system was compromised and its impact.
- To identify the network indicators associated with the malware, which can then be used to detect similar infections using network monitoring. For example, during your analysis, if you determine that a malware contacts a particular domain/IP address, then you can use this domain/IP address to create a signature and monitor the network traffic to identify all the hosts contacting that domain/IP address.
- To extract host-based indicators such as filenames, and registry keys, which, in turn, can be used to determine similar infection using host-based monitoring. For instance, if you learn that a malware creates a registry key, you can use this registry key as an indicator to create a signature, or scan your network to identify the hosts that have the same registry key.
- To determine the attacker's intention and motive. For instance, during your analysis, if you find that the malware is stealing banking credentials, then you can deduce that the motive of the attacker is monetary gain.

Types of Malware Analysis?

To understand the working and the characteristics of malware and to assess its impact on the system, you will often use different analysis techniques. The following is the classification of these analysis techniques:

- **Fully Automated analysis**: The easiest way to begin learning about a particular malware specimen is to examine it using fully automated tools, some of which are

available as commercial products and some as free ones. These utilities are designed to quickly assess what the specimen might do if it ran on a system. They typically produce reports with details such as the registry keys used by the malicious program, its mutex values, network traffic, and so on. They might not provide as much insight as a human analyst would obtain when examining the specimen in a more manual fashion. However, they contribute to the incident response process by rapidly handling vast amounts of malware, allowing the analyst (whose time is relatively expensive) to focus on the specimens that truly require her attention.

- **Static analysis**: This is the process of analysing a file without executing it. It is easiest to perform and allows you to extract the metadata associated with the suspect binary. Static analysis might not reveal all the required information, but it can sometimes provide interesting information that helps in determining where to focus your subsequent analysis efforts.
- **Dynamic analysis (Behavioural Analysis):** This is the process of executing the suspect file in an isolated environment and monitoring its behaviour. This analysis technique is easy to perform and gives valuable insights into the activity of the binary during its execution. This analysis technique is useful but does not reveal all the functionalities of the hostile program.
- **Code analysis:** It is an advanced technique that focuses on analysing the code to understand the inner workings of the binary. This technique reveals information that is not possible to determine just from static and dynamic analysis. Code analysis is further divided into Static code analysis and Dynamic code analysis. Static code analysis involves disassembling the suspect binary and looking at the code to understand the program's behaviour, whereas Dynamic code analysis involves debugging the suspect binary in a controlled manner to understand its functionality. Code analysis requires an understanding of the programming language and operating system concepts.
- **Memory analysis (Memory forensics):** This is the technique of analysing the computer's RAM for forensic artefacts. It is typically a forensic technique, but integrating it into malware analysis will assist in gaining an understanding of the malware's behaviour after infection. Memory analysis is especially useful to determine the stealth and evasive capabilities of the malware.

2. Setting Up the Lab Environment (Windows Malware Analysis)

Analysis of a hostile program requires a safe and secure lab environment, as we do not want to infect our system or the production system. A malware lab can be very simple or complex depending on the resources available to us (hardware, virtualization software, Windows license, and so on). This section will guide us to set up a simple personal lab on a single physical system consisting of virtual machines (VMs).

Lab Requirements:

Before setting up a lab, there is a need of few components: a physical system running a base operating system of Linux, Windows, or macOS X, and installed with virtualization software (such as VMware or VirtualBox). When analysing the malware, we will be executing the malware on a Windows-based virtual machine (Windows VM). The advantage of using a virtual machine is that after we finish analysing the malware, we can revert it to a clean state.

VMware Workstation for Windows and Linux is available for download from:

https://www.vmware.com/products/workstation/workstation-evaluation.html

VMware Fusion for macOS X is available for download from:

https://www.vmware.com/products/fusion/fusion-evaluation.html

VirtualBox for different flavours of operating systems is available for download from:

https://www.virtualbox.org/wiki/Downloads

To create a safe lab environment, we should take the necessary precautions to avoid malware from escaping the virtualized environment and infecting our physical (host) system. The following are a few points to remember when setting up the virtualized lab.

- Keep virtualization software up to date. This is necessary because it might be possible for malware to exploit a vulnerability in the virtualization software, escape from the virtual environment, and infect the host system.
- Install a fresh copy of the operating system inside the virtual machine (VM), and do not keep any sensitive information in the virtual machine.
- While analyzing a malware, if we don't want the malware to reach out to the internet, then we should consider using host-only network configuration mode or restrict our network traffic within our lab environment using simulated services.
- Do not connect any removable media that might later be used on the physical machines, such as USB drives.
- For analyzing Windows malware (typically Executable or DLL), it is recommended to choose a base operating system such as Linux or macOS X for our host machine instead of Windows. This is because, even if a Windows malware escapes from the virtual machine, it will still not be able to infect your host machine.

Overview of Lab Architecture:

The lab architecture consists of a physical machine (called host machine) running Ubuntu Linux with instances of Linux virtual machine (Ubuntu Linux VM) and Windows virtual machine (Windows VM). These virtual machines will be configured to be part of the same network and use Host-only network configuration mode so that the malware is not allowed to contact the Internet and network traffic is contained in the isolated lab environment.

Windows VM is where the malware will be executed during analysis, and the Linux VM is used to monitor the network traffic and will be configured to simulate Internet services (DNS, HTTP, and so on) to provide an appropriate response when the malware requests for

these services. For example, the Linux VM will be configured such that when the malware requests a service such as DNS, the Linux VM will provide the proper DNS response.

The following figure shows an example of a simple lab architecture. In this setup, the Linux VM will be preconfigured to IP address 192.168.1.100, and the IP address of the Windows VM will be set to 192.168.1.x (where x is any number from 1 to 254 except 100). The default gateway and the DNS of the Windows VM will be set to the IP address of the Linux VM (that is, 192.168.1.100) so that all the Windows network traffic is routed through the Linux VM.

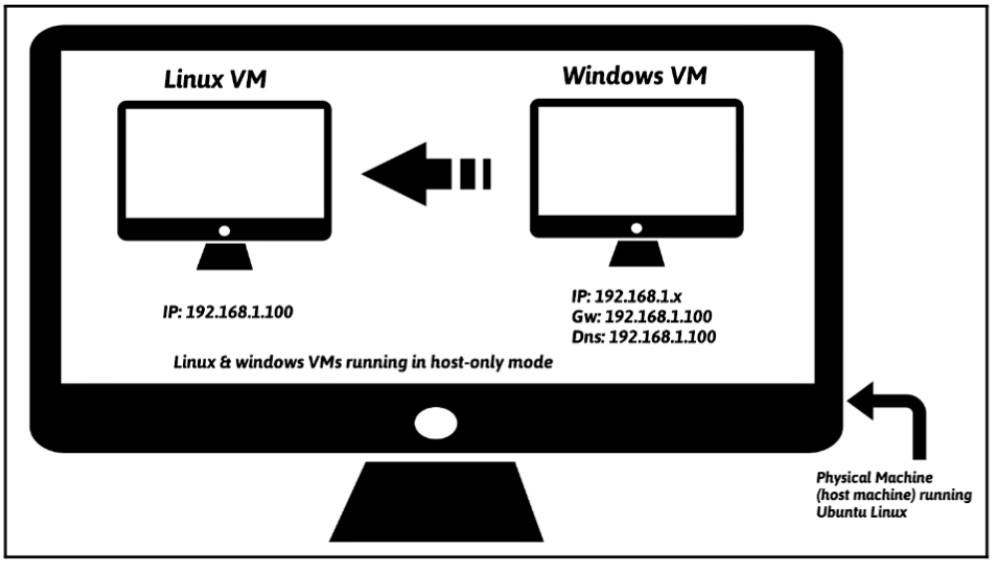


Figure 1 Virtual machine

It is also possible to set up a lab consisting of multiple VMs running different versions of Windows; this will allow us to analyse the malware specimen on various versions of Windows operating systems. An example configuration containing multiple Windows VMs will look similar to the one shown in the following diagram:

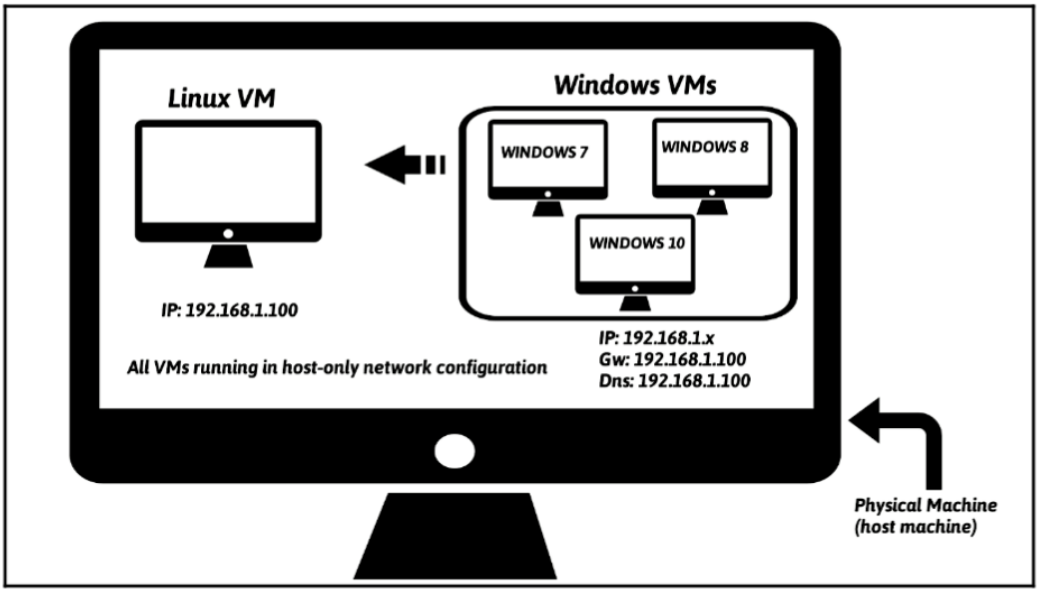


Figure 2 Multiple VMs

Setting Up and Configuring Linux VM:

To set up the Linux VM, we can use Ubuntu 16.04.2 LTS Linux distribution (http://releases.ubuntu.com/16.04/). The reason for choosing Ubuntu is that most of the tools are either preinstalled or available through the apt-get package manager. The following is a step-by-step procedure to configure Ubuntu 16.04.2 LTS on VMware and VirtualBox.

- Download Ubuntu 16.04.2 LTS from:
  http://releases.ubuntu.com/16.04/
  install it in VMware Workstation/Fusion or VirtualBox.
- Install the Virtualization Tools on Ubuntu; this will allow Ubuntu's screen resolution to automatically adjust to match our monitor's geometry and provide additional enhancements, such as the ability to share clipboard content and to copy/paste or drag and drop files across our underlying host machine and the Linux virtual machine. To install virtualization tools on VMware Workstation or VMware Fusion, follow the procedure mentioned at:
  https://kb.vmware.com/selfservice/microsites/search.do?language=en_UScmd=displayKCexternalId=1022525
  or watch the video at:
  https://youtu.be/ueM1dCk3o58
  Once installed, reboot the system
- If VirtualBox is being used, there is need to install Guest Additions software. To accomplish this, from the VirtualBox menu, select Devices | Insert guest additions CD image. This will bring up the Guest Additions Dialog Window. Then click on Run to invoke the installer from the virtual CD. Authenticate with password when prompted and reboot.
- Once the Ubuntu operating system and the virtualization tools are installed, start the Ubuntu VM and install the following tools and packages.
- Install pip; pip is a package management system used to install and manage packages written in Python. In this course, we will be running a few Python scripts; some of them rely on third-party libraries. To automate the installation of third-party packages, we need to install pip. Run the following command in the terminal to install and upgrade pip:

```
$ sudo apt-get update
$ sudo apt-get install python-pip or ($ sudo apt-get install python3-pip)
$ pip install --upgrade pip or ($ pip3 install --upgrade pip)
```

The following are some of the tools and Python packages that will be used in malware analysis. To install these tools and Python packages, run these commands in the terminal:

```
$sudo apt-get install python-magic

$sudo apt-get install upx

$sudo pip install pefile  or ($ sudo python3.6 -m pip install pefile)

$sudo apt-get install yara
```

$sudo pip install yara-python or ($ sudo python3.6 -m pip install yara-python)

$sudo apt-get install ssdeep

$sudo apt-get install build-essential libffi-dev python python-devlibfuzzy-dev

$sudo pip install ssdeep or ($ sudo python3.6 -m pip install ssdeep)

$sudo apt-get install wireshark

$sudo apt-get install tshark

- INetSim (http://www.inetsim.org/index.html) is a powerful utility that allows simulating various Internet services (such as DNS, and HTTP) that malware frequently expects to interact with. To install INetSim, use the following commands.

```
$ sudo su
# echo "deb http://www.inetsim.org/debian/ binary/" > \
  /etc/apt/sources.list.d/inetsim.list
# wget -O - http://www.inetsim.org/inetsim-archive-signing-key.asc
| \
apt-key add -
# apt update
# apt-get install inetsim
```

- Isolate Ubuntu VM within our lab by configuring the virtual appliance to use Host-only network mode. On VMware, bring up the Network Adapter Settings and choose Host-only mode as shown in the following Figure. Save the settings and reboot.

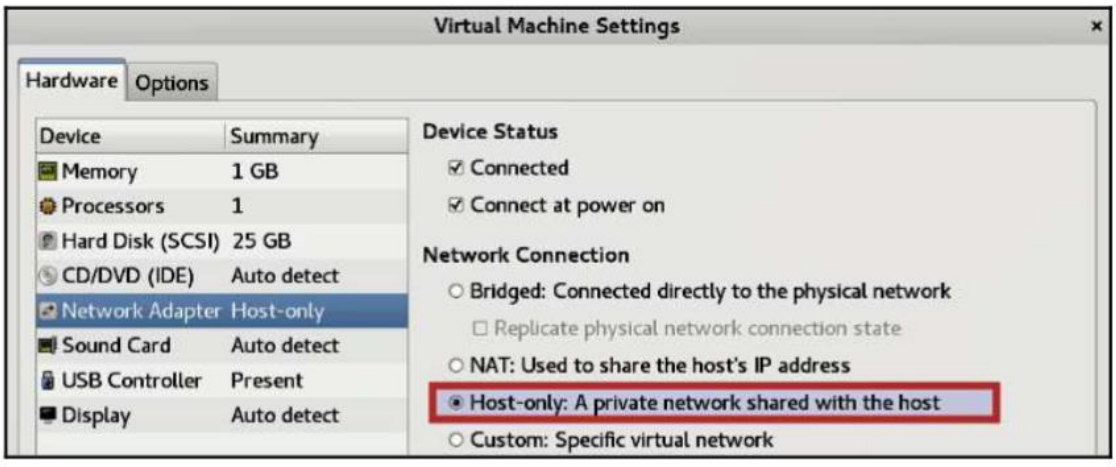


Figure 3 VM isolation

In VirtualBox, shut down Ubuntu VM and then bring up Settings. Select Network and change the adapter settings to Host-only Adapter as shown in the following diagram; click on OK. On VirtualBox, sometimes when we choose the Host-only adapter option, the interface name might

appear as Not selected. In that case, we need to first create at least one host-only interface by navigating to File|Preferences | Network | Host-only networks | Add host-only network. Click on OK; then bring up the Settings. Select Network and change the adapter settings to Host-only Adapter, as shown in the following screenshot. Click on OK.

Figure 4Changing Network settings in VM

- Assign a static IP address of 192.168.1.100 to the Ubuntu Linux VM. To do that, power on the Linux VM, open the terminal window, type the command ifconfig, and note down the interface name. For example, the interface name is ens33. Open the file /etc/network/interfaces using the following command:

  $ sudo gedit /etc/network/interfaces

  Add the following entries at the end of the file (make sure you replace ens33 with the interface name on your system) and save it:

  ```
  auto ens33
  iface ens33 inet static
  address 192.168.1.100
  netmask 255.255.255.0
  ```

  The /etc/network/interfaces file should now look like the one shown here.

  ```
  # interfaces(5) file used by ifup(8) and ifdown(8)
  auto lo
  iface lo inet loopback

  auto ens33
  iface ens33 inet static
  address 192.168.1.100
  netmask 255.255.255.0
  ```

Then restart the Ubuntu Linux VM. At this point, the IP address of the Ubuntu VM should be set to 192.168.1.100. You can verify that by running the following command:

```
$ ifconfig
ens33 Link encap:Ethernet HWaddr 00:0c:29:a8:28:0d
inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
```

```
inet6 addr: fe80::20c:29ff:fea8:280d/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:21 errors:0 dropped:0 overruns:0 frame:0
TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:5187 (5.1 KB) TX bytes:5590 (5.5 KB)
```

- The next step is to configure INetSim so that it can listen to and simulate all the services on the configured IP address 192.168.1.100. By default, it listens on the local interface (127.0.0.1), which needs to be changed to 192.168.1.100.
  To do that, open the configuration file located at /etc/inetsim/inetsim.conf using the following command:

```
$ sudo gedit /etc/inetsim/inetsim.conf
```

Go to the service_bind_address section in the configuration file and add the entry shown here:

```
service_bind_address 192.168.1.100
```

The added entry (highlighted) in the configuration file should look like this:

```
# service_bind_address
#
# IP address to bind services to
#
   # Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
#service_bind_address 10.10.10.1
service_bind_address 192.168.1.100
```

By default, INetSim's DNS server will resolve all the domain names to 127.0.0.1. Instead of that, we want the domain name to resolve to 192.168.1.100 (the IP address of Linux VM). To do that, go to the dns_default_ip section in the configuration file and add an entry as shown here:

```
dns_default_ip 192.168.1.100
```

added entry (highlighted in the following code) in the configuration file should look like this:

```
# dns_default_ip
#
# Default IP address to return with DNS replies
```

```
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
#dns_default_ip 10.10.10.1
dns_default_ip 192.168.1.100
```

Once the configuration changes are done, Save the configuration file and launch the INetSim main program. Verify that all the services are running and also check whether the inetsim is listening on 192.168.1.100, as highlighted in the following code. We can stop the service by pressing CTRL+C:

```
$ sudo inetsim

INetSim 1.2.6 (2016-08-29) by Matthias Eckert & Thomas Hungenberg

Using log directory: /var/log/inetsim/

Using data directory: /var/lib/inetsim/

Using report directory: /var/log/inetsim/report/

Using configuration file: /etc/inetsim/inetsim.conf

=== INetSim main process started (PID 2640) ===

Session ID: 2640

Listening on: 192.168.1.100

Real Date/Time: 2017-07-08 07:26:02

Fake Date/Time: 2017-07-08 07:26:02 (Delta: 0 seconds)

Forking services...

irc_6667_tcp - started (PID 2652)

ntp_123_udp - started (PID 2653)

ident_113_tcp - started (PID 2655)

time_37_tcp - started (PID 2657)

daytime_13_tcp - started (PID 2659)

discard_9_tcp - started (PID 2663)

echo_7_tcp - started (PID 2661)

dns_53_tcp_udp - started (PID 2642)

[.....REMOVED.....]

http_80_tcp - started (PID 2643)
```

https_443_tcp - started (PID 2644)

done.

Simulation running.

- At some point, we need the ability to transfer files between the host and the virtual machine. To enable that on *VMware*, power off the virtual machine and bring up the **Settings**. Select **Options** | **Guest Isolation** and check both **Enable drag and drop** and **Enable copy and paste**. **Save** the settings.

  On *Virtualbox*, while the virtual machine is powered off, bring up **Settings |General | Advanced** and make sure that both **Shared Clipboard and Drag 'n' Drop** are set to **Bidirectional**. Click on **OK**.

- At this point, the Linux VM is configured to use **Host-only** mode, and INetSim is set up to simulate all the services. The last step is to take a snapshot (clean snapshot) and give it a name of your choice so that you can revert it back to the clean state when required. To take a snapshot on **VMware workstation**, click on **VM | Snapshot | Take Snapshot**. On **Virtualbox**, the same can be done by clicking on **Machine | Take Snapshot**.
  Apart from the drag and drop feature, it is also possible to transfer files from the host machine to the virtual machine using shared folders; refer to the following for VirtualBox (*https://www.virtualbox.org/manual/ch04.html#sharedfolders)* and to the following for VMware (*https://docs.vmware.com/en/VMware-Workstation-Pro/14.0/com.vmware.ws.using.doc/GUID-AACE0935-4B43-43BA-A935-FC71ABA17803.html*).

Setting Up and Configuring Windows VM:

Before setting up the Windows VM, we need to install a Windows operating system (Windows 7, Window 8, and so on) of our choice in the virtualization software (such as VMware or VirtualBox). Once we have Windows installed, follow these steps:

- Download Python from https://www.python.org/downloads/, be sure to download Python 2.7.x (such as 2.7.13); most of the scripts that run on the Python 2.7 version may not run correctly on Python 3. After downloading the file, run the installer. Make sure to check the option to install pip and add python.exe to Path, as shown in the following screenshot. Installing pip will make it easier to install any third-party Python libraries, and adding Python to the path will make it easier to run Python from any location.

Figure 5 pyhton installation

- Configure Windows VM to run in **Host-only** network configuration mode. To do that in **VMware** or **VirtualBox**, bring up the **Network Settings** and choose the **Host-only mode**; save the settings and reboot (this step is similar to the one covered in the *Setting Up and Configuring Linux VM* section).
- Configure the IP address of the Windows VM to 192.168.1.x (choose any IP address except 192.168.1.100 because the Linux VM is set to use that IP) and set up your Default gateway and the DNS server to the IP address of Linux VM (that is, 192.168.1.100), as shown in the following screenshot. This configuration is required so that when we execute the hostile program on the Windows VM, all of the network traffic will be routed through the Linux VM.

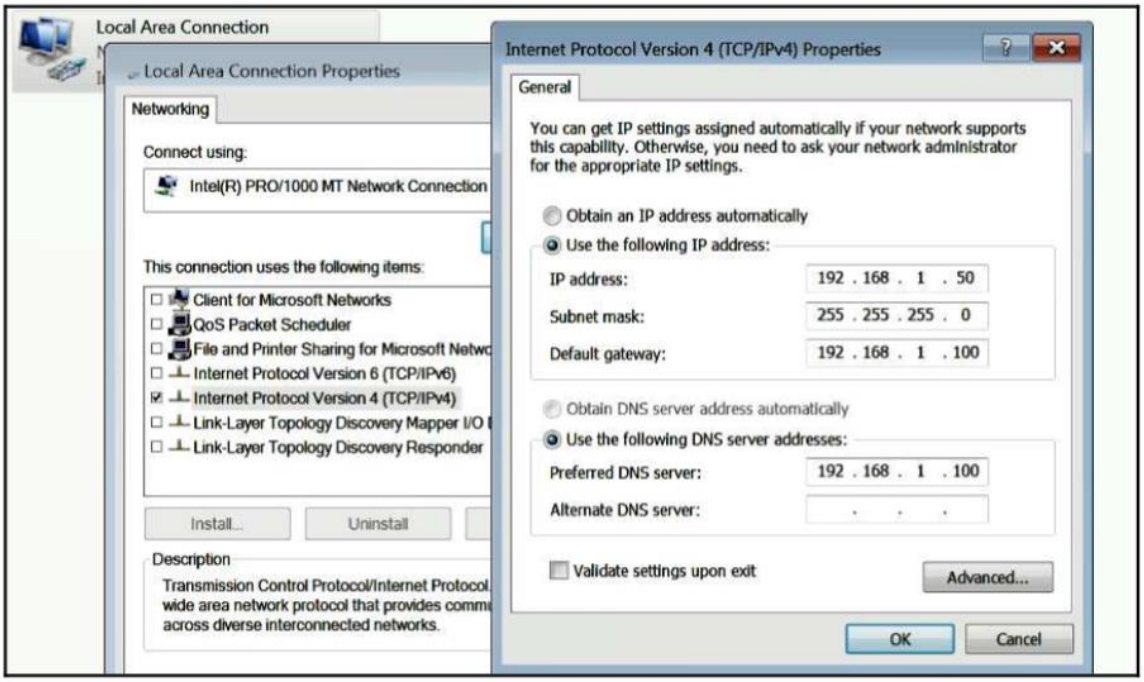


Figure 6 IP configuration

- Power on both the Linux VM and the Window VM, and make sure they can communicate with each other. You can check for the connectivity by running the ping command, as shown in this screenshot:

```
C:\Users\test>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
```

Figure 7 ping command

- Windows Defender Service needs to be disabled on your Windows VM as it may interfere when we are executing the malware sample. To do that, press the Windows key + R to open the Run menu, enter gpedit.msc, and hit **Enter** to launch the **Local Group Policy Editor**. In the left-hand pane **of Local Group Policy Editor**, navigate to **Computer Configuration | Administrative Templates| Windows Components | Windows Defender**. In the right-hand pane, double-click on the **Turn off Windows Defender policy** to edit it; then select **Enabled** and click on **OK.**

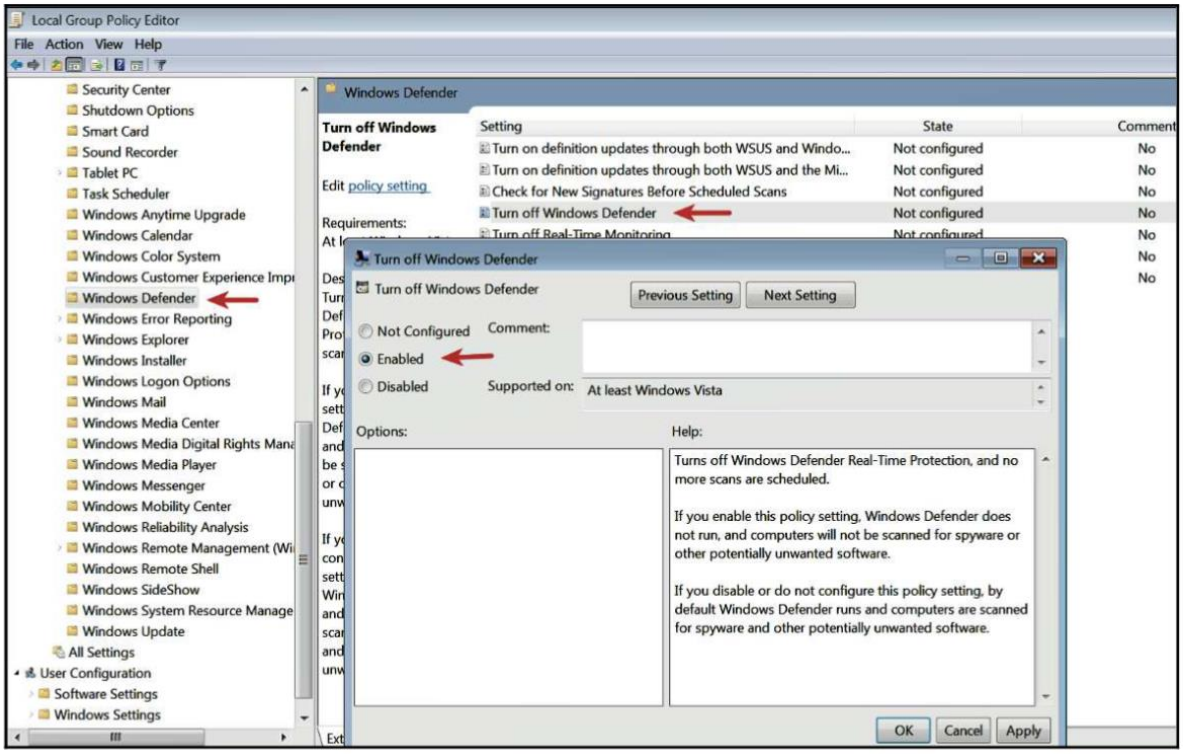


Figure 8 Local Group Policy Editor

- To be able to transfer files (drag and drop) and to copy clipboard content between the host machine and the Windows VM, follow the instructions as mentioned in *Setting Up and Configuring Linux VM* section.
- Take a clean snapshot so that we can revert to the pristine/clean state after every analysis. The procedure to take a snapshot was covered in *Setting Up and Configuring Linux VM* section.

  At this point, our lab environment should be ready. The Linux and Windows VMs in our clean snapshot should be in Host-only network mode and should be able to communicate with each other.

Malware Sources

Once we have a lab set up, we will need malware samples for performing analysis. The following are some of the sources from where we can get malware samples for our analysis. Some of these sources allow us to download malware samples for free (or after free registration), and some require us to contact the owner to set up an account, after which we will be able to obtain the samples:

- *Hybrid Analysis*: https://www.hybrid-analysis.com/
- *VirusBay*: https://beta.virusbay.io/
- *Contagio malware dump*: http://contagiodump.blogspot.com/
- *VirusShare*: https://virusshare.com/
- *theZoo*: http://thezoo.morirt.com/
- Lenny Zeltser's blog post: https://zeltser.com/malware-sample-sources/

2. Static Analysis

Static analysis is the technique of analysing the suspect file without executing it. It is an initial analysis method that involves extracting useful information from the suspect binary to make an informed decision on how to classify or analyse it and where to focus our subsequent analysis efforts. Static analysis involves tools and techniques to extract valuable information from the suspect file. Following are covered in case of static analysis of **Windows malware** sample.

- Identifying the malware's target architecture
- Fingerprinting the malware
- Scanning the suspect binary with anti-virus engines
- Extracting strings, functions, and metadata associated with the file
- Identifying the obfuscation techniques used to thwart analysis
- Classifying and comparing the malware samples

I. Determining file type of malware

- If the suspect binary has a file type of Portable Executable (PE), which is the file format for Windows executable files (.exe , .dll , .sys , .drv , .com ,.ocx and so on), then we can deduce that the file is designed to target the Windows operating system.
- Most Windows-based malware are executable files ending with extensions such as .exe, .dll, .sys, and so on. But relying on file extensions alone is not recommended. File extension is not the sole indicator of file type. Attackers use different tricks to hide their file by modifying the file extension and changing its appearance to trick users into executing it. Instead of relying on file extension, File signature can be used to determine the file type.
- A file signature is a unique sequence of bytes that is written to the file's header. Different files have different signatures, which can be used to identify the type of file. The Windows executable files, also called PE files (such as the files ending with .exe , .dll , .com , .drv ,.sys , and so on), have a file signature of MZ or hexadecimal characters 4D 5A in the first two bytes of the file.

- A handy resource for determining the file signatures of different file types based on their extension is available at http://www.filesignatures.net/.

(a) Identifying File Type Using Manual Method:

Use hex editors such as **hxd** (https://mh-nexus.de/en/hxd/). On Linux systems, to look for the file signature, the **xxd** command "**xxd -g l filename**(with or without extension**) | more**" can be used, which generates a hex dump of the file. Look for magic numbers in hex code. For eg., 'MZ' or '4D 5A' in output indicates the format used for .EXE files.

(b) Identifying File Type Using Tools:

Linux: $ file filename(with extension)
Windows: CFF Explorer (http://www.ntcore.com/exsuite.php)

(c) Identifying File Type Using Python:

Linux: $ sudo apt-get install python-magic
Windows: To install the python-magic module, we can follow the procedure mentioned at https://github.com/ahupp/python-magic.Once the python-magic is installed, the following commands can be used in the script to determine the file type:

```
$ python3
  >>> import magic
  >>> m = magic.open(magic.MAGIC_NONE)
 >>> m.load()
 >>> ftype = m.file(r'filename with extension')
 >>> print ftype
output : PE32 executable (GUI) Intel 80386 or x86-64, for MS
 Windows
```

II. Fingerprinting the malware

- Fingerprinting involves generating the cryptographic hash values for the suspect binary based on its file content.
- Identifying a malware specimen based on filename is ineffective because the same malware sample can use different filenames, but the cryptographic hash that is calculated based on the file content will remain the same. Hence, a cryptographic hash for suspect file serves as a unique identifier.
- During dynamic analysis, when malware is executed, it can copy itself to a different location or drop another piece of malware. Having the cryptographic hash of the sample can help in identifying whether the newly dropped/copied sample is the same as the original sample or a different one. This information can assist us in deciding whether the analysis needs to be performed on a single sample or multiple samples.
- File hash is frequently used as an indicator to share with other security researchers/investigators to help them identify the sample.

- File hash can be used to determine whether the sample has been previously detected by searching online or searching the database of multi Anti-virus scanning service like VirusTotal.

(a) Generating Cryptographic Hash Using Tools:

Linux:

```
$md5sum filename
$sha256sum filename
$sha1sum filename
```

Windows: HashMyFiles (http://www.nirsoft.net/utils/hash_my_files.html)

(b) Determining Cryptographic Hash in Python:

It is possible to generate file hashes using the hashlib module, as shown here in example:

```
$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
>>> import hashlib
>>> content = open(r"log.exe","rb").read()
>>> print hashlib.md5(content).hexdigest()
6e4e030fbd2ee786e1b6b758d5897316
>>> print hashlib.sha256(content).hexdigest()
01636faaae739655bf88b39d21834b7dac923386d2b52efb4142cb278061f97f
>>> print hashlib.sha1(content).hexdigest()
625644bacf83a889038e4a283d29204edc0e9b65
```

III. Multiple Anti-Virus Scanning

- There are a few factors/risks to consider when scanning a binary with Anti-Virus scanners or when submitting a binary to online anti-virus scanning services. If a suspect binary does not get detected by the Anti-Virus scanning engines, it does not necessarily mean that the suspect binary is safe. These anti-virus engines rely on signatures and heuristics to detect malicious files. The malware authors can easily modify their code and use obfuscation techniques to bypass these detections, because of which some of the anti-virus engines might fail to detect the binary as malicious.
- When you upload a binary to a public site, the binary you submit may be shared with third parties and vendors. The suspect binary may contain sensitive, personal, or proprietary information specific to your organization, so it is not advisable to submit a binary that is part of a confidential investigation to public anti-virus scanning services. Most web-based anti-virus scanning services allow you to search their existing database of scanned files using cryptographic hash values (MD5,

SHA1, or SHA256); so an alternative to submitting the binary is to search based on the cryptographic hash of the binary.

- When you submit a binary to the online antivirus scanning engines, the scan results are stored in their database, and most of the scan data is publicly available and can be queried later. Attackers can use the search feature to query the hash of their sample to check whether their binary has been detected. Detection of their sample may cause the attackers to change their tactics to avoid detection.

(a) Scanning the Suspect Binary with VirusTotal

VirusTotal web interface provides you the ability to search their database using hash, URL, domain, or IP address.VirusTotal offers another useful feature called VirusTotal Graph, built on top of the VirusTotal dataset.For more information on VirusTotal Graph, refer to the documentation: https://support.virustotal.com/hc/en-us/articles/115005002585-VirusTotal-Graph . VirusTotal offers different private (paid) services (https://support.virustotal.com/hc/en-us/articles/115003886005-Private-Services),which allow you to perform threat hunting and download samples submitted to it.

(b) Querying Hash Values Using VirusTotal Public API

VirusTotal provides scripting capabilities via its public API (https://www. virustotal.com/en/documentation/public-api/); it allows us to automate file submission, retrieve file/URL scan reports, and retrieve domain/IP reports. We need to connect to the internet and must have a VirusTotal public API key (which can be obtained by signing up for a VirusTotal account).

(c) Alternate tools

pestudio (https://www.winitor.com/)
PPEE (https://www.mzrst.com/)
Jotti Malware Scan (https://virusscan.jotti.org/)
Metadefender (https://www.metadefender.com/#!/scan-file)

IV. Extracting Strings

- Strings are ASCII and Unicode-printable sequences of characters embedded within a file.
- Extracting strings can give clues about the program functionality and indicators associated with a suspect binary. For example, if a malware creates a file, the filename is stored as a string in the binary. Or, if a malware resolves a domain name controlled by the attacker, then the domain name is stored as a string. Strings extracted from the binary can contain references to filenames, URLs, domain names, IP addresses, attack commands, registry keys, and so on.

(a) String Extraction Using Tools

Linux: $ strings -a filename

'strings' by default extracts strings of atleast four characters length. With the -a option it is possible to extract strings from the entire file.

$ strings -a -el filename

Malware specimens also use Unicode (2 bytes per character) strings. To get useful information from the binary, sometimes we need to extract both ASCII and Unicode strings.To extract Unicode strings using the strings command, use the -el option

Windows:

-Pestudio

-PPEE

-The strings utility ported to Windows by Mark Russinovich (https://technet.microsoft.com/en-us/sysinternals/strings.aspx)

(b) Decoding Obfuscated Strings

Obfuscated strings will not show up in the strings utility and other string extraction tools. FireEye Labs Obfuscated String Solver (FLOSS) is a tool designed to identify and extract obfuscated strings from malware automatically. FLOSS can also be used just like the strings utility to extract human-readable strings (ASCII and Unicode). We can download FLOSS for Windows or Linux from:
https://github.com/fireeye/flare-floss

On windows,

copy the floss file to folder of malware binary then run following in cmd prompt

>floss64.exe filename

or

floss64.exe --no-static-strings filename (do avoid displaying static strings, displays only decoded/stack strings)

On Linux,

download floss then copy it into folder of malware sample,

$ chmod +x floss

$ ./floss filename

V. Determining File Obfuscation

Often malware authors obfuscate or armor their malware binary.The obfuscation techniques make it difficult to detect/analyze the binary; extracting the strings from such binary results in very fewer strings, and most of the strings are obscured. Malware authors often use programs suchas Packers and Cryptors to obfuscate their file to evade detection from security products suchas anti-virus and to thwart analysis.

(a) Packers and Cryptors

A Packer is a program that takes the executable as input, and it uses compression to obfuscate the executable's content. This obfuscated content is then stored within the structure of a new executable file; the result is a new executable file (packed program) with obfuscated content on the disk. Upon execution of the packed program, it executes a decompression routine, which extracts the original binary in memory during runtime and triggers the execution.

A Cryptor is similar to a Packer, but instead of using compression, it uses encryption to obfuscate the executable's content, and the encrypted content is stored in the new executable file. Upon execution of the encrypted program, it runs a decryption routine to extract the original binary in the memory and then triggers the execution.

UPX (https://upx.github.io/),is a popular packer used for file obfuscation

Eg:

$ upx -o spybot_packed.exe spybot.exe

command to unpack

$ upx -d -o spybot_unpacked.exe spybot_packed.exe

$ls -al command gives output of files including file size,file permissions

(b) Determining File Obfuscation Using Exeinfo PE

To detect packers on Windows, we can use a freeware tool such as Exeinfo PE (http://exeinfo.atwebpages.com/); it has an easy-to-use GUI. It uses more than 4,500 signatures (stored in userdb.txt in the same directory) to detect various compilers, packers, or cryptors utilized to build the program. In addition to detecting Packers, another interesting feature of Exeinfo PE is that it gives information/references on how to unpack the sample.

Other tools for packer detections:

a. TrID (http://mark0.net/soft-trid-e.html)
b. TRIDNet (http://mark0.net/soft-tridnet-e.html)
c. Detect It Easy (http://ntinfo.biz/)
d. RDG Packer Detector (http://www.rdgsoft.net/)
e. packerid.py (https://github.com/sooshie/packerid)
f. PEiD (http://www.softpedia.com/get/Programming/Packers-Crypters-Protectors/PEiD-updated.shtml)

VI. Inspecting PE Header Information

PE file is a series of structures and sub-components that contain the information required by the operating system to load it into memory. Windows executables must conform to the PE/COFF (Portable Executable/Common Object File Format). The PE file format is used by the Windows executable files (such as .exe , .dll ,.sys , .ocx , and .drv) and such files are generally called Portable Executable (PE) files.

When an executable is compiled, it includes a header (PE header), which describes its structure. When the binary is executed, the operating system loader reads the information from the PE header and then loads the binary content from the file into the memory. The PE header contains information such as where the executable needs to be loaded into memory, the address where the execution starts, the list of libraries/functions on which the application relies on, and the resources used by the binary. Examining the PE header yields a wealth of information about the binary, and its functionalities.

The following are some of the great resources for understanding the PE file structure:

- An In-Depth Look into the Win32 Portable Executable File Format - Part 1:

http://www.delphibasics.info/home/delphibasicsarticles/anin-depthlookintothewin32portableexecutablefileformat-part1

- An In-Depth Look into the Win32 Portable Executable File Format - Part 2: http://www.delphibasics.info/home/delphibasicsarticles/anin-depthlookintothewin32portableexecutablefileformat-part2

- PE Headers and structures:

http://www.openrce.org/reference_library/files/reference/PE%20Format.pdf

- PE101 - A Windows Executable Walkthrough: https://github.com/corkami/pics/blob/master/binary/pe101/pe101.pdf

The following are some of the tools that allow you to examine and modify the PE structure and its sub-components:

a) Pestudio: https://www.winitor.com
b) CFF Explorer: http://www.ntcore.com/exsuite.php
c) PE Internals: http://www.andreybazhan.com/pe-internals.html
d) PPEE(puppy): https://www.mzrst.com/
e) PEBrowse Professional: http://www.smidgeonsoft.prohosting.com/pebrowse-pro-file-viewer.html

(a) Inspecting File Dependencies and Imports

Usually, malware interacts with the file, registry, network, and so on. To perform such interactions, malware frequently depends on the functions exposed by the operating system. Windows exports most of its functions, called Application Programming Interfaces (API), required for these interactions in Dynamic Link Libary (DLL) files. Executables import and call these functions typically from various DLLs that provide different functionality. The functions that an executable imports from other files (mostly DLLs) are called imported functions (or imports).

Inspecting the DLLs that a malware relies upon and the API functions that it imports from the DLLs can give an idea about the functionality and capability of malware and what to anticipate during its execution.

The file dependencies in Windows executables are stored in the import table of the PE file structure.In PEstudio,'libraries' shows the DLLs used by program ,'imports' shows imported APIs from DLLs by the malware

(b) Inspecting Exports

The executable and DLL can export functions, which can be used by other programs. Typically, a DLL exports functions (exports) that are imported by the executable. A DLL cannot run on its own and depends on a host process for executing its code. An attacker often creates a DLL that exports functions containing malicious functionality. To run the malicious functions within the DLL, it is somehow made to be loaded by a process that calls these malicious functions. DLLs can also import functions from other libraries (DLLs) to perform system operations.

Inspecting the exported functions can give us a quick idea of the DLL's capabilities. Export function names may not always give an idea of a malware's capabilities. An attacker may use random or fake export names to mislead our analysis or to throw off track.

(c) Examining PE Section Table and Sections

PE file is divided into 5 sections followed by PE Header Sections represent either code or data and they have in memory attributes (read-execute/read-write etc).If it is code, then section contains instructions(that are to be executed by processor).If it is data, section contains different types of data, such as read/write program data (global variables), import/export tables, resources, and so on.

During the compilation of the executable, consistent section names are added by the compilers.

.text or CODE- contains executable code
.data or DATA- contains read/write data and global variables
.rdata-contains read only data,sometimes import and export information too
.idata-if present,contains import table else import information is stored in .rdata
.edata-if present,contains export table else export information is stored in .rdata
.rsrc-contains resources used by executable such as icons,menus,dialogs,strings etc

These section names are mainly for humans and are not used by the operating system, which means it is possible for an attacker or an obfuscation software to create sections with different names. If you come across section names that are not common, then you should treat them with suspicion, and further analysis is required to confirm maliciousness.

Information about these sections (such as section name, where to find the section, and its characteristics) is present in the section table in the PE header. Examining a section table will give information about the section and its characteristics.

- Sections in pestudio:
- some of Field - Description:

  Names -section names
  virtual_size -size of the section when loaded into memory
  virtual_address-relative virtual address (RVA) i.e., offset from base address of the executable, where section can be found in memory
  raw_size -size of the section on the disk
  raw_data -offset in the file where section can be found
  entry_point -RVA where code starts executing

(d) Examining the Compilation Timestamp

Compilation Timestamp gives an idea of when the malware was first created, information can be useful in building a timeline of the attack campaign. It is also possible that an attacker modifies the timestamp to prevent an analyst from knowing the actual timestamp. A compile timestamp can sometimes be used to classify suspicious samples (like timestamp modified to future date).

In Pestudio, the path is fileheader>compiler-stamp

All Delphi binaries have a compile timestamp set to June 19, 1992, making it hard to detect the actual compile timestamp. If we are investigating a malware binary set to this date, there is a high possibility that we are looking at Delphi binary. The blog post at a http://www.hexacorn.com/blog/2014/12/05/the-not-so-boring-land-of-borland-executables part-1/ gives information on how it may be possible to get the compilation timestamp from a Delphi binary.

(e) Examining PE Resources

.rsrc section contains information of :

- icons
- menus
- dialogs
- strings
- version information (origin,company name,program author,copyright information)

Attackers can store:

- additional binary
- decoy documents
- configuration data

Tools:

Resource Hacker (http://www.angusj.com/resourcehacker/) is a great tool to examine, view, and extract the resource from a suspect binary.

When loaded the binary in resourcehacker, analyze the content in 'Binary','Icon','Icon Group' and right click then save required resource to disk.

VII. Comparing and Classifying the Malware

Comparing the suspect binary with previously analyzed samples or the samples stored in a public or private repository can give an understanding of the malware family, its characteristics, and the similarity with the previously analyzed samples. cryptographic hashing (MD5/SHA1/SHA256) is a great technique to detect identical samples, it does not help in identifying similar samples.

(a) Classifying Malware Using Fuzzy Hashing

Fuzzy hashing is a great method to compare files for similarity. ssdeep (http://ssdeep.sourceforge.net) is a useful tool to generate the fuzzy hash for a sample, and it also helps in determining percentage similarity between the samples

In Ubuntu, install ssdeep ($sudo apt-get install ssdeep,$sudo pip install ssdeep or ($ sudo python3.6 -m pip install ssdeep)) and run the following command to determine fuzzy hash of a sample

$ssdeep filename

For comapring fuzzyhashes of two or more samples in the same directory,follow the below procedure

```
$ls
$md5sum *       (* is used to select all)
$ssdeep -pb *   (-p option is pretty matching mode)
```

For comapring fuzzyhashes of two or more samples in the directory and subdirectories, follow the below procedure

```
$ssdeep -lrpa directoryname/     (-r is recursive mode)
```

Technique to compare any new file with the hashes of previously analyzed samples:

```
$ssdeep * > outputfilename.txt
$ssdeep -m outputfilename.txt newmalwarefilename
```

(b) Classifying Malware Using Import Hash

Import hash (or imphash) is a technique in which hash values are calculated based on the library/imported function (API) names and their particular order within the executable.If the files were compiled from the same source and in the same manner, those files would tend to have the same imphash value.

If we come across samples that have the same imphash values, it means that they have the same import address table and are probably related.

We should also take a look at http://blog.jpcert.or.jp/2016/05/classifying-mal-a988.html which covers details of using import API and the fuzzy hashing technique (impfuzzy) to classify malware samples.

In Pestudio,filename>imphash is the path

Files having the same imphash does not necessarily mean they are from the same threat group; we might have to correlate information from various sources to classify our malware. For example, it is possible that the malware samples were generated using a common builder kit that is shared among groups; in such cases, samples might have the same imphash.

(c) Classifying Malware Using Section Hash

In Pestudio, filename>sections>MD5sum is the path. When we are analyzing a malware sample, we should consider generating the fuzzy hash, imphash, and section hashes for the malicious binary and store them in a repository; that way, when you come across a new sample, it can be compared with these hashes to determine similarity.

(d) Classifying Malware Using YARA

Security researchers classify malware based on the unique strings and the binary indicators present in the binary. Sometimes, malware can also be classified based on general characteristics. YARA (http://virustotal.github.io/yara/) is a powerful malware identification and classification tool. Malware researchers can create YARA rules based on textual or binary information contained within the malware specimen. These

YARA rules consist of a set of strings and a Boolean expression, which determines its logic. Once the rule is written, we can use those rules to scan files using the YARA utility or we can use yara-python to integrate with our tools.

3. Dynamic Analysis

Dynamic analysis (behavioral analysis) involves analyzing a sample by executing it in an isolated environment and monitoring its activities, interaction, and effect on the system

Lab Environment Overview

When performing dynamic analysis, we will be executing the malware specimen, so we need to have a safe and secure lab environment to prevent our production system from being infected. The following diagram shows the lab environment that can be used to perform dynamic analysis.

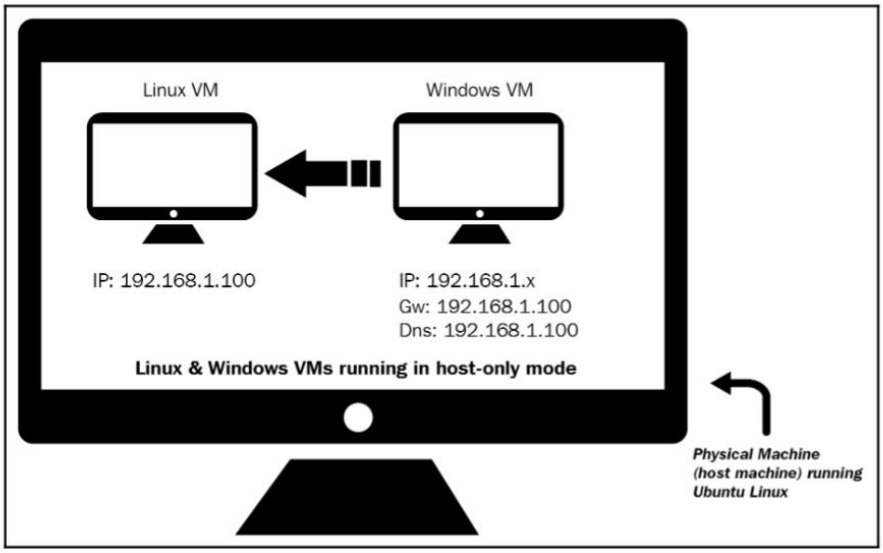


Figure 9 Host Only VM

In this setup, both the Linux and Windows VM were configured to use the host-only network configuration mode. The Linux VM was preconfigured to an IP address of 192.168.1.100, and the IP address of the Windows VM was set to 192.168.1.50. The default gateway and the DNS of the Windows VM were set to the IP address of the Linux VM (192.168.1.100), so all the Windows network traffic is routed through the Linux VM. The Windows VM will be used to execute the malware sample during analysis, and the Linux VM will be used to monitor the network traffic and will be configured to simulate internet services (such as DNS, HTTP, and so on) to provide the appropriate response when malware requests these services.

I. System and Network Monitor Tools

When malware is executed, it can interact with system and network in various ways and perform different activities. Examples:

-spawning a child process
-drop additional files on filesystem
-create registry keys and values for persistence
-download other components
-take commands from C2 server etc

The following list outlines different types of monitoring carried out during dynamic analysis:

a. Process monitoring: monitoring the process activity and examining the properties of the result process during malware execution.
b. File system monitoring: monitoring the real-time file system activity during malware execution.
c. Registry monitoring: monitoring the registry keys accessed/modified and registry data that is being read/written by the malicious binary.
d. Network monitoring: monitoring the live traffic to and from the system during malware execution

II. Dynamic Analysis (Monitoring) Tools:

Before performing dynamic analysis, it is essential to understand the tools that we use to monitor the malware's behaviour. After setup of lab environment as we can download these tools to our host machine and then transfer/install those tools to our virtual machines and take a new, clean snapshot. We need to run these tools with administrator privileges; this can be done by right-clicking on the executable and selecting Run as administrator.

(a) Process Inspection with Process Hacker

Process Hacker (http://processhacker.sourceforge.net/) is an open source, multi-purpose tool that helps in monitoring system resources. It is a great tool for examining the processes running on the system and to inspect the process attributes. It can also be used to explore services, network connections, disk activity, and so on.
Once the malware specimen is executed, this tool can help us identify the newly created malware process (its process name and process ID), and by right-clicking on a process name and selecting Properties, you will be able to examine various process attributes. You can also right-click on a process and terminate it. The following screenshot shows Process Hacker listing all the processes running on the system, and the properties of wininit.exe

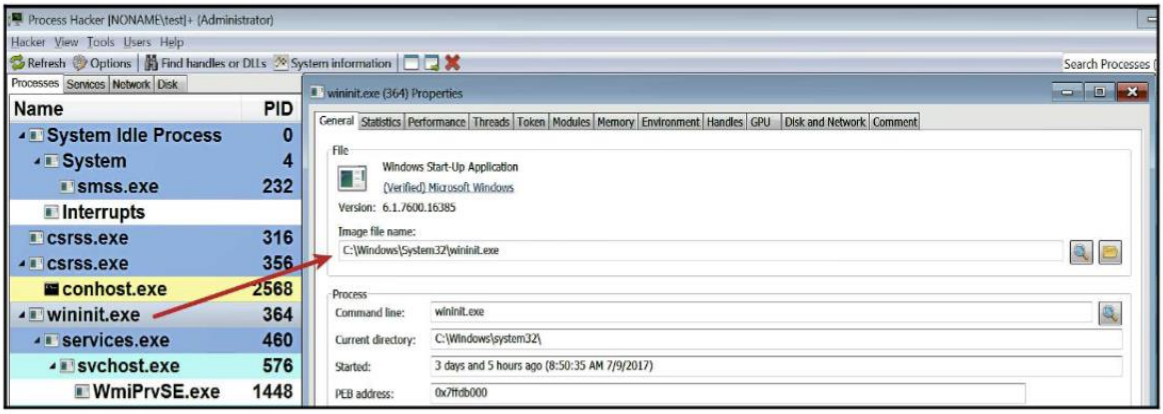


Figure 10 properties of wininit.exe

(b) Determining System Interaction with Process Monitor

Process Monitor (https://technet.microsoft.com/en-us/sysinternals/processmonitor.aspx) is an advanced monitoring tool that shows the real-time interaction of the processes with the filesystem, registry, and process/thread

activity. When we run this tool (run as Administrator), you will immediately notice that it captures all the system events, as shown in the following screenshot. To stop capturing the events, you can press Ctrl + E, and to clear all the events you can press Ctrl+ X. The following screenshot shows the activities captured by Process Monitor on a clean system:

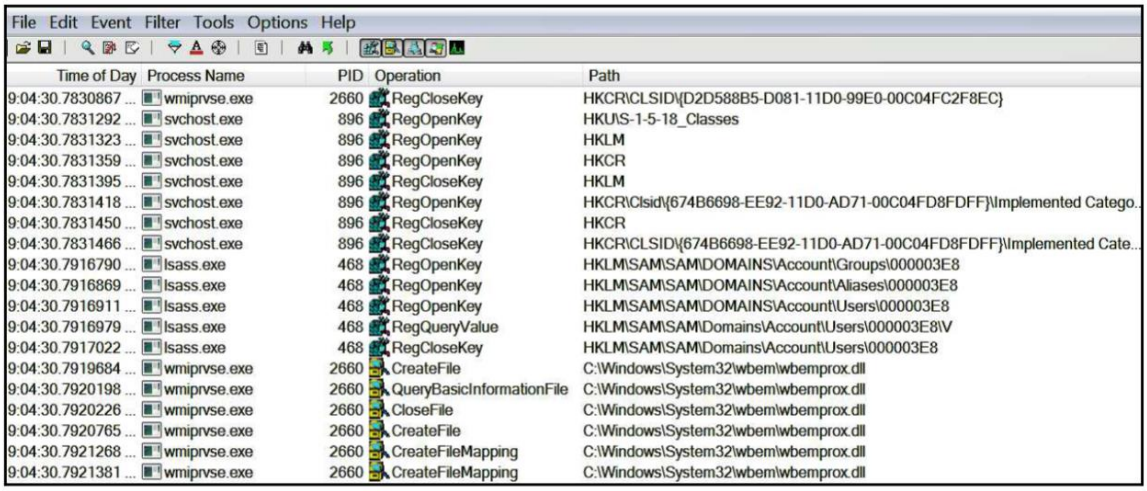


Figure 11 activities captured by Process Monitor

From the events captured by the Process Monitor, we can see that lots of activity gets generated on a clean system. When performing malware analysis, we will only be interested in the activities produced by the malware. To reduce noise, we can use the filtering features which hides unwanted entries and allows you to filter on specific attributes. To access this feature, select the Filter menu and then click on Filter (or press Ctrl + L). In the following screenshot, the filter is configured to display events only related to the process, svchost.exe

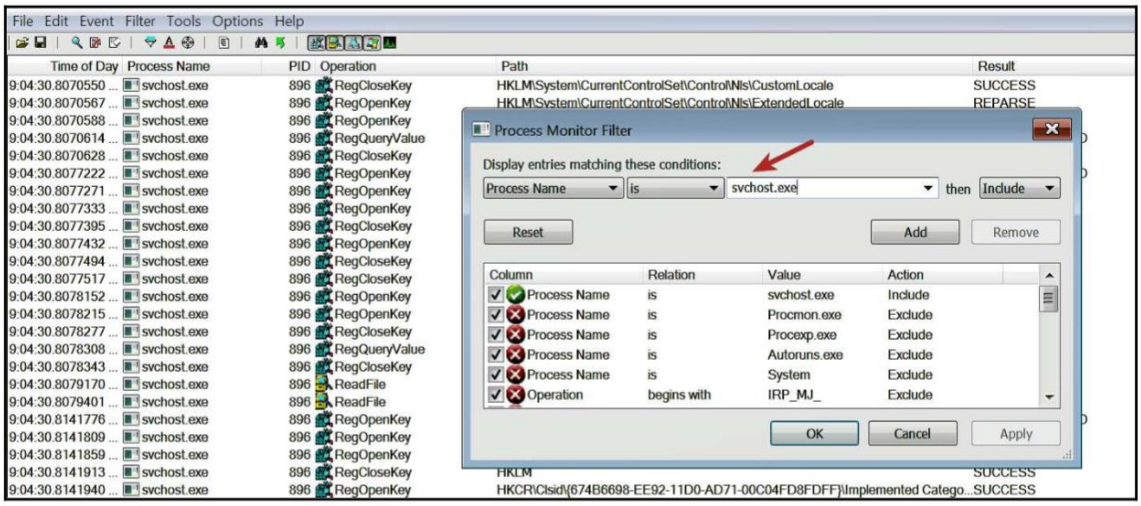


Figure 12 svchost.exe

(c) Logging System Activities Using Noriben

Even though Process Monitor is a great tool to monitor a malware's interaction with the system, it can be very noisy, and manual effort is required to filter the noise. Noriben (https://github.com/Rurik/Noriben) is a Python script that works in conjunction with Process Monitor and helps in collecting, analyzing, and reporting

runtime indicators of the malware. The advantage of using Noriben is that it comes with pre-defined filters that assist in reducing noise and allow you to focus on the malware-related events. To use Noriben, download it to our Windows VM, extract it to a folder, and copy Process Monitor (Procmon.exe) into the same folder before running the Noriben.py Python script, as shown in the following screenshot.

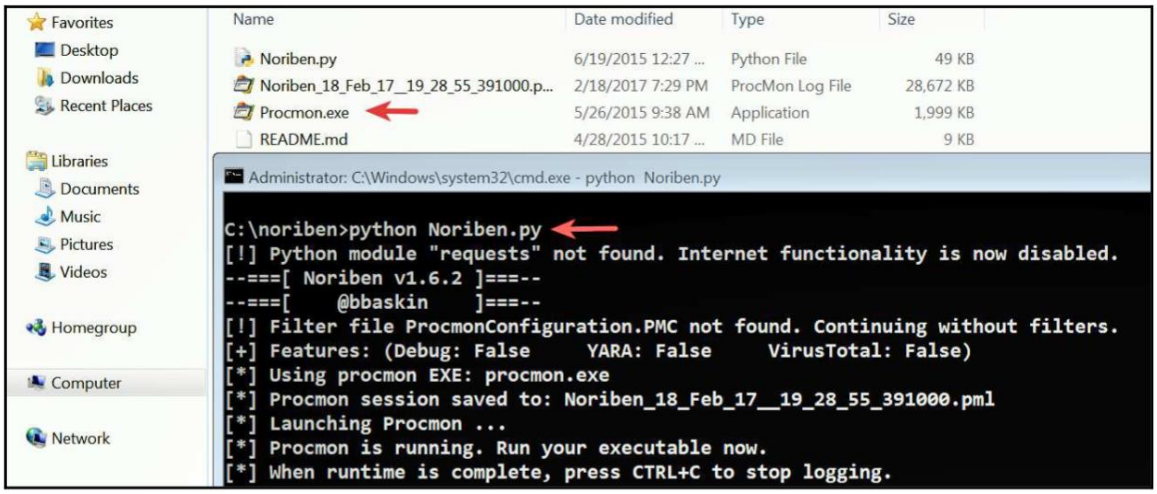


Figure 13 Procmon.exe

When we run Noriben, it launches Process Monitor. Once we are done with the monitoring, we can stop Noriben by pressing Ctrl + C, which will terminate Process Monitor. Once terminated, Noriben stores the results in a text file (.txt) and a CSV file (.csv) in the same directory. The text file contains events segregated based on the categories (like process, file, registry, and network activity) in separate sections, as shown in the following screenshot. Also, note that the number of events is much less because it applied predefined filters that reduced most of the unwanted noise:

Figure 14 Noriben results text format

The CSV file contains all the events (process, file, registry, and network activity) sorted by the timeline (the order in which the events occurred), as shown in the following screenshot.

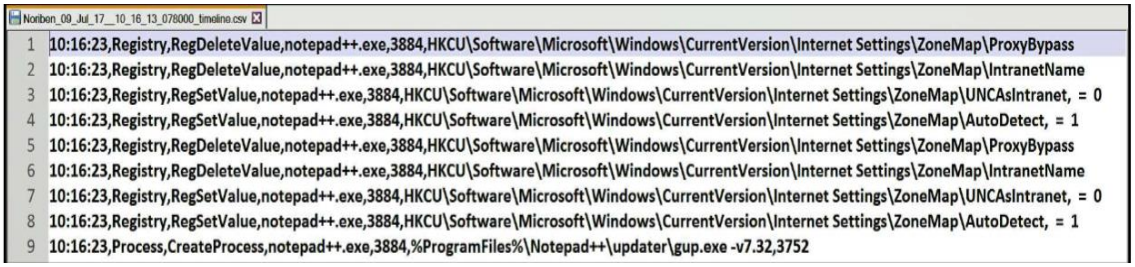


Figure 15 Noriben results excel format

The text file and the CSV file can give different perspectives. If we are interested in the summary of events based on the category then we can look at the text file; if we are interested in the sequence of events in the order in which it occurred then we can view the CSV file.

(d) Capturing Network Traffic With Wireshark

When the malware is executed, we will want to capture the network traffic generated as a result of running the malware; this will help us understand the communication channel used by the malware and will also help in determining network-based indicators. Wireshark (https://www.wireshark.org/) is a packet sniffer that allows you to capture the network traffic.

To invoke Wireshark on Linux, run the following command:

$ sudo wireshark

To start capturing the traffic on a network interface, click on Capture | Options (Or press Ctrl + K), select the network interface, and click on Start:

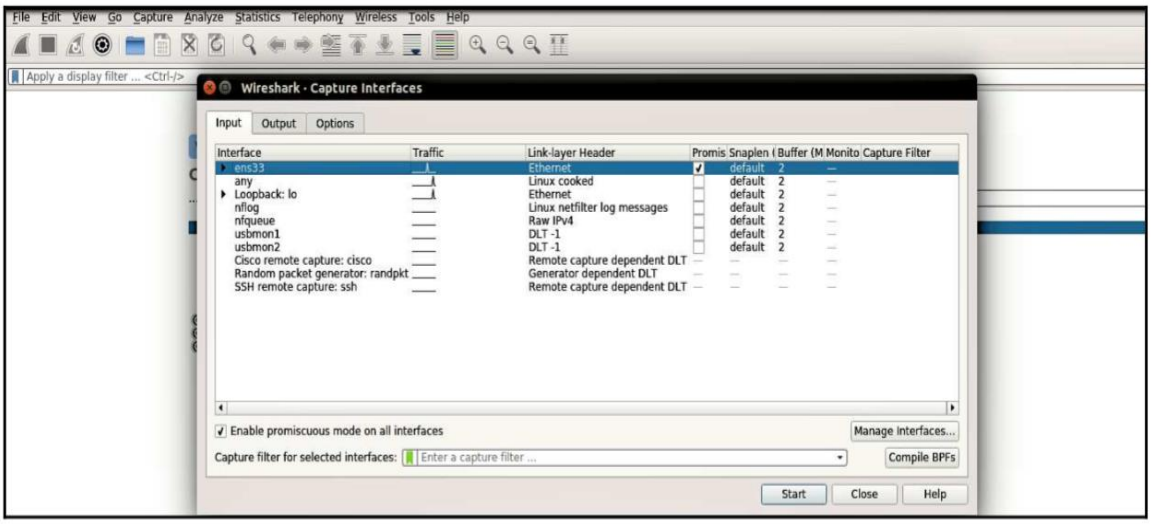


Figure 16 Wireshark Interface

(e) Simulating Services with INetSim

Most malware, when executed, reach out to the internet (command and control server), and it is not a good idea to allow the malware to connect to its C2 server, and also sometimes these servers may be unavailable. During malware analysis, we need to determine the behaviour of the malware without allowing it to contact the actual

command and control (C2) server, but at the same time, we need to provide all the services required by the malware so that it can continue its operation.

INetSim is a free Linux-based software suite for simulating standard internet services (such as DNS, HTTP/HTTPS, and so on). Once INetSim is launched, it simulates various services, as shown in the following output, and it also runs a dummy service that handles connections directed at nonstandard ports:

```
->installation
$ sudo su
# echo "deb http://www.inetsim.org/debian/ binary/" > \
/etc/apt/sources.list.d/inetsim.list
# wget -O - http://www.inetsim.org/inetsim-archive-signing-key.asc
| \
apt-key add -
# apt update
# apt-get install inetsim

 $ sudo inetsim
INetSim 1.2.6 (2016-08-29) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 2758) ===
Session ID: 2758
Listening on: 192.168.1.100
Real Date/Time: 2017-07-09 20:56:44
Fake Date/Time: 2017-07-09 20:56:44 (Delta: 0 seconds)


Forking services...
* irc_6667_tcp - started (PID 2770)
* dns_53_tcp_udp - started (PID 2760)
* time_37_udp - started (PID 2776)
* time_37_tcp - started (PID 2775)
* dummy_1_udp - started (PID 2788)
* smtps_465_tcp - started (PID 2764)
* dummy_1_tcp - started (PID 2787)
* pop3s_995_tcp - started (PID 2766)
* ftp_21_tcp - started (PID 2767)
* smtp_25_tcp - started (PID 2763)
* ftps_990_tcp - started (PID 2768)
* pop3_110_tcp - started (PID 2765)
[.....REMOVED.
```

.....]
* http_80_tcp - started (PID 2761)
* https_443_tcp - started (PID 2762)
done.
Simulation running.

Apart from simulating services, INetSim can log communications, and it can also be configured to respond to HTTP/HTTPS requests and return any files based on the extensions. For example, if malware requests an executable (.exe) file from the C2 server, INetSim can return a dummy executable file to the malware. That way, we get to know what malware does with the executable file after downloading it from the C2 server.

The following example demonstrates the use of INetSim. In this example, a malware sample was executed on the Windows VM, and the network traffic was captured using Wireshark on the Linux VM without invoking INetSim. The following screenshot displays the traffic captured by Wireshark. It shows that the infected Windows system (192.168.1.50) is trying to communicate with the C2 server by first resolving the C2 domain, but because our Linux VM does not have a DNS server running, that domain could not be resolved (as indicated by the Port Unreachable message).

Figure 17 Data captured in wireshark

This time, the malware was executed, and the network traffic was captured on the Linux VM with INetSim running (simulating services). From the following screenshot, it can be seen that the malware first resolves the C2 domain, which is resolved to the Linux VM's IP address of 192.168.1.100. Once resolved, it then makes an HTTP communication to download a file (settings.ini):

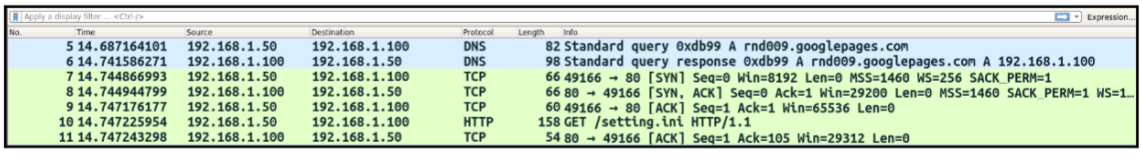


Figure 18 Malware Behaviour

From the following screenshot, it can be seen that the HTTP response was given by the HTTP server simulated by INetSim. In this case, the User-Agent field in the HTTP request suggests that the standard browser did not initiate the communication and such an indicator can be used to create network signatures:

```
GET /setting.ini HTTP/1.1
User-Agent: AutoIt
Host: rnd009.googlepages.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Tue, 11 Jul 2017 05:18:16 GMT
Content-Length: 258
Content-Type: text/html
Connection: Close
Server: INetSim HTTP Server
```

Figure 19 User-Agent field in the HTTP

Another alternative to INetSim is FakeNet-NG (https://github.com/fireeye/flare-fakenet-ng), which allows us to intercept and redirect all or specific network traffic by simulating network services.

III. Dynamic Analysis Steps

During dynamic analysis (behavioral analysis), you will follow a sequence of steps to determine the functionality of the malware. The following list outlines the steps involved in the dynamic analysis:

- Reverting to the clean snapshot: This includes reverting our virtual machines to a clean state.
- Running the monitoring/dynamic analysis tools: In this step, we will run the monitoring tools before executing the malware specimen. We need to run them with administrator privileges.
- Executing the malware specimen: In this step, we will run the malware sample with administrator privileges.
- Stopping the monitoring tools: This involves terminating the monitoring tools after the malware binary is executed for a specified time.
- Analyzing the results: This involves collecting the data/reports from the monitoring tools and analyzing them to determine the malware's behavior and functionality.

IV. Other Malware Analysis Tools:

Code Analysis	IDA Pro x64dbg/x32dbg OllyDbg etc
Mobile Malware Analysis	MobSF Android Studio MVT (tool developed by Amnesty International Security Lab after Pegasus news breakout) Jd GUI Cucko Xapkdetector etc
Fully Automated Analysis	Any Run Hybrid Analysis Cuckoo etc

3. Overview of Network Forensics

Network Forensics is a process of collecting and analyzing raw network data and tracking network traffic systematically to ascertain how an attack was carried out or how an event occured on a network. Because network attacks are on the rise, there's more focus on this field and increasing demand for skilled technicians. Labor forecasts predict a shortfall of 50,000 network forensics specialists in Law Enforcement, Legal firms, corporations, and universities.

You might hear the terms Cyber Forensics or Digital Forensics; they usually refer to network forensics, not computer forensics.

When intruders break into a network, they leave a trail behind. Being able to spot variations in network traffic can help you track intrusions, so knowing your network's typical traffic patterns is important. For example, the primary ISP of SVP National Police Academy, has peak hours of use between 9:30 a.m. to 6:30 p.m. because that is the duration of standard working hours at SVP National Police Academy. If a usage spike occured during the night, the network administrator on duty would recognize it as unusual activity and could take steps to investigate it.

Network forensics can also help you determine whether a network is truly under attack or a user has unknowingly installed a custom program.

Network forensics examiners must establish standard procedures for how to acquire data after an attack or intrusion incident. Typically, network administrators want to find compromised machines, get them offline, and restore them as quickly as possible to minimise downtime.

However, taking the time to follow standard procedures is essential to ensure that all compromised systems have been found and to ascertain attack methods in an effort to prevent them from happening again.

Securing a Network

Network Forensics is used to determine how a security breach occured; however, steps must be taken to harden networks before a security breach happens, particularly with recent increase in network attacks, viruses, and other security incidents. Hardening includes a range of tasks, from applying the latest patches to using a layered network defense strategy, which sets up layers of protection to hide the most valuable data at the innermost part of the network. It also ensures that the deeper into the network an attacker gets, the more difficult access becomes and the more safeguards are in the place.

Testing networks is as iportant as testing servers. You need to be up to date on the latest methods intruders use to infiltrate networks as well as methods internal employees use to sabotage networks. In the early and mid-1990s, approximately 70% of network attacks were caused by internal employees. Since then, this problem has been compounded by contract employees, who often have the same level of network privileges as full-time employees.

In addition, small companies of fewer than 10 employees often don't consider security precautions against internal threats necessary, so the can be more susceptible to problems caused by empoyees revealing proprietary information to competitors. However, increasing use of the internet has caused a sharp rise in external threats, so internal and external threats are currently about 50-50.

I. Scope of Network Forensics

Network forensics plays a critical role in the cloud computing environment but with limitations that tie the network forensics deeply to systems and computer forensics. Network forensics is best applied where the network is owned by the company at the boundary and into the desktops or systems that access cloud resources. Network forensics works in the cloud environment when the company has addressed many of the limitations of network forensics in the cloud when the company is still building out their cloud infrastructure. It is possible to go back and retrofit an in-built forensics capability, and it should be done if the capability to conduct forensics was not part of

the original business plan of moving information and systems into the cloud.

Network forensics can also have an influence on the outcome of an investigation into an event as long as data was collected at the box and at the entry and exit points of the company network. The use of in-built firewall logs, system logs, and other logs will generally point to an entry time, place, and IP address that can be used to help determine how the event was propagated through the network and what steps can be taken to help minimize any future event by providing solid data on the event. A large part of network forensics is being able to monitor the network traffic in order to isolate the number of servers that need to be taken down for the traditional forensics process.

II. The Importance of DHCP Logs

If the network for which you are performing network forensics uses Dynamic Host Configuration Protocol (DHCP), then it is vitally important that the organization records and preserves the DHCP logs for the period of time being examined. Without the DHCP logs, an IT-savvy attorney can

challenge the link between the Internet Protocol (IP) address and the computer and, ultimately, to the user of that computer. If DHCP logs are not available, you will need to find other ways

to establish the link between a computer and an IP address. If you have access to the suspect's computer or the computer of interest, you may find logged records of the IP address in the security event log and the firewall log. Although it is still part of the network, you might be able to query the DHCP server or perform ipconfig/all on the suspect's computer.

The DHCP log entry also provides you a way to physically locate the computer within the network. These logs describe which device issued the IP address to a computer with a specific Mac address. The switch logs can divulge which switch port was used. The switch port connects by cable to your cable infrastructure. Following this cable leads to a specific data jack in a specific building and room. If your network or facilities team has maintained a good database of these associations, then you can find the physical location of the suspected computer. Otherwise, you will need to physically

locate the suspected computer by going room to room and checking the identifiers on each data jack. If the jacks aren't labeled, you are left to pulling on wires and following the cable, which may or may not be possible with walls and floors in place.

III. Standard Operating Procedure of Network Data

Search and seizure of Network Data basically includes collection of data from Network devices such as Routers, Firewall, Modems, and computers which are connected with the network.

Criminals now a days are heavily using Network based servers in committing their crimes apart from using Computer, Mobile and Laptops etc.

For example – If a death threat email is sent by using an email server which is hosted by the criminal then the email server will essentially have to be seized in order to preserve the evidence.

Similarly, in a case of Denial of Service (DoS) attack on an organization or network, the traces are found on the network devices.

The recent case of "Bangladesh Bank heist" further emphasizes the importance of having an ability to investigate network device in order to find relevant evidences of the crime.

Hence, it becomes pertinent to have a Standard Operating Procedure (SOP) to seize such Network devices. Electronic records such as Network Logs, Firewall logs, FTP logs and File Server Logs etc. play an important role in gathering the evidences in network based crimes.

a. Search & Seizure of Digital Evidence from Network

Step1: Before You Twitch

Consent search or Search warrant	• Understand the nature of the crime • Read the search warrant
Concerns	• Safety – It is a crime scene • Destruction of potential evidence
Plan, Plan, Plan	• The seizure • The collection techniques • The order of events

Step2: - What to Take Along

1) Evidence Tape
2) Chain of Custody forms
3) Reading Glasses
4) Inventory forms
5) Camera (battery, memory)
6) Backup disposable camera
7) Tool kit. Jewelers set. Needle nose pliers.
8) Sharpies, pens
9) Adhesive tape
10) New, wiped and verified Hard Drives in Pelican, w/lock
11) Gloves
12) Static wrist bands
13) Tableau Pelican (ATA, SCSI, eSATA, Firefly) with power supplies and line cords. Firewire I/F cables, laptop adaptor. Small laptop adaptor.
14) Firewire I/F board.
15) Several USB mouse. Two PS mouse.
16) Laptop with X-Ways and FTK (crossover tested)
17) eSATA interface
18) USB-small USB cable
19) PS2/USB converter
20) Small flatscreen monitor
21) UPS
22) Extension cord
23) Power strip (2)
24) Digital Media Flash reader
25) DOS Boot w/Firewire USB.
26) DOS Boot with utilities
27) 1GB NIC
28) ATA interface with cable
29) CDs with WinHex, FTK, Linen,
30) Boot CD with Helix/Lenin, Boot USB
31) F-Response CD
32) Dongles – FTK, X-Ways, F-Response
33) Flashlight
34) Powered USB Hub
35) Magnifying Glass
36) Blank Labels
37) Bottle water
38) RFID readers/writers
39) Credit card readers/writers
40) Smart card readers/writers
41) Bar code readers/writers

Step3: Think About Potential Evidence

- Probable cause to seize HW?
- Probable cause to seize SW?
- Probable cause to seize Data? Such as – network logs, firewall logs, Pcap Files, FTP logs, File system logs
- Where will the search of the seized evidence be conducted?
- Careful of business interruption issues and proprietary information.
- Depends on the role of the computers in the crime.

Step4: Prior to Serving the Warrant

- Start your investigation report
- Understand the nature of the crime
- Describe the role of the computer/digital device in the crime
- Describe the limits of your investigation
- Probable cause for seizure
- What can be seized
- What can be looked at
- Where is the search to be conducted

Step5: Seize what

- HW
- SW
- Data - Such as – network logs, firewall logs, Pcap Files, FTP logs, File system logs
- All things digital
- All things related to digital
- Media, notes, documentation
- Stay within the bounds of the search warrant

Step 6: Search or Seizure Where

- Secure the scene, restrict access
- Preserve the area, no more fingerprints
- Insure the safety of all concerned
- Nobody touch nothing!
- Usually the forensic specialist will not be a first responder. However, often they are.
- On site, in the field office, in a lab
- Disposal of seized items
- Consider the size of the seizure
- Suspects:
    - Interview
    - Passwords
    - Location of data
    - Installed software
    - Network

Step 7: Tag & Bag

- Tape every drive slot shut
- Photograph, diagram and label all components
- Photograph, diagram and label all connections
- Photograph, diagram and label all cables – both ends
    - ✓ You will have to reconstruct
- Pack it for transport
- Keep it away from EM
- Collect all printed material
    - ✓ Docs, records, notes

Step8: Seizure

- If the network is active
    - ✓ Do not power down any networking gear
    - ✓ They have no hard drives
    - ✓ All evidence is volatile
    - ✓ If no significant network traffic disconnect from the ISP
- Using the USB device harvest the routers and switches
- Then disassemble the network
    - ✓ Seize the servers and work stations
- Get the network admin to help
    - ✓ They could corrupt the data, SO be careful
- If OFF leave it off.
    - ✓ Tag and bag

- If ON
  - ✓ Photograph and document especially comm connections
  - ✓ An attempt may be made to access memory and capture the most recently printed document.
  - ✓ If the device is a scan first and then dispatch, everything is stored on the hard drive.
  - ✓ Disconnect the comms interfaces
  - ✓ Tag and bag
- Determine phone connections
  - ✓ Subpoena service provider

Step9: Other Stuff

- Docs, notes, documentation, etc.
- Credit cards, smart cards, RFIDs, etc
- CDs, DVDs – all media

Step 10: Security Systems

- Ingress/egress logs – time line, IDs
- Service provider
- System info
- Photograph and document location of all devices
- Text, video
- Tag and bag all stored data and recorded data.
- Detailed documentation – you can't tag and bag

Step11: After Pictures are taken from a "switched on" Network Device

- If the computer is a standalone PC
  - ✓ pull the plug
  - ✓ Vista is different
  - ✓ Do not turn it off
- If it is a laptop
  - ✓ Pull the plug
  - If it is still on, it has a functioning battery
  - Pull the battery
  - Keep the battery separate

Step12: Photos Method

- Items and placement
- Each Item
  - ✓ Placement
  - ✓ Model numbers, Serial numbers
  - ✓ Front
  - ✓ Back
  - ✓ Cables
  - ✓ Anything that might be of interest.
- You only get one chance to record the original evidence
- Floor plan
  - ✓ Locate all equipment
  - ✓ Number all equipment on the floor plan

- ✓ You will have to reconstruct
- Photograph/Video graph
  - ✓ The entire area containing HW & cables
  - ✓ The screen of each computer that is on.

Steps14: Notes

- Keep a very detailed log of every operation action
  - ✓ Details
  - ✓ Time
  - ✓ Order
- They can cover a lot of mistakes during the seizure and search
- What did you do?
- What reasons for doing it.
- Itemize potential harm versus another way of doing it.

b. Crime Scene Scenarios (Network based Server)

- **When Network Server is in Power ON condition**

Step I	Isolating the accused working on the Computer if any and his/her Interrogation without allowing him/her to touch the Computer
Step II	Visible inspection of Scene of Crime in front of technically qualified independent witnesses without touching anything
Step III	Photography of the Scene of Crime (SoC) • Close shot of the MONITOR • Long shot and close shot of the SoC from various angles showing all the devices connected with the Computer • Long and close shot of the system from different angles identifying all externally connected devices to the system • System in power on condition (Contd..)
Step IV	Collection of finger print if required
Step V	Search for any kind of external memory devices like Pen Drive, Hard Disk, etc.
Step VI	Collection of RAM dump and system information, encrypted files if any by the IO/ Cyber Forensic Expert
Step VII	Remove the power plug without shutting down the system
Step VIII	Open the CPU and take a photograph of the inside view showing all peripherals like Hard Disk, RAM, Motherboard etc. Remove the Hard Disk
Step IX	• Photography of the Hard Disk showing:

	• Unique S. No. of the Hard Disk • Connector Ports • Jumper Position • Logic Board
Step X	• Creation of 3 Images of Hard Disk and other external memory devices seized with Write Blocker • Hash calculation
Step XI	Preparation of seizure list mentioning all details like Unique S.No. of External Drives, Hard Disk and Hash value of the Hard Disk and other external memory devices
Step XII	• Dispatching of HardDisk: • 1st image to be sent to the Forensic Lab along with seizure list and questionnaire with permission of the court as per regular procedure • 2nd image to be kept with IO for analysis • 3rd image to be handed over to the accused party • Original Hard Disk and external memory devices along with seizure list to be sent to the Court along with other original documents at the time of submission of Final Report.

Crime Scene Scenarios (MODEM)

Step I	Take Photographs of MODEM indicating its make and model along with serial numbers
Step II	- IF the MODEM is ON, then browse into the MODEM through any of the computer devices which is connected to the MODEM. The password for the same should be available with the Network Administrator - Note down various details which are available in the MODEM. Such as – o MAC addresses connected with the MODEM o Logs of websites o The static IP assigned to the router
Step III	- IF the MODEM is switched off, then seize the Modem with proper chain of custody

4. Log Analysis

I. What is log?

Log is a systematic daily or hourly record of activities, events, and/or occurrences in a computer.

Logging is the act of keeping a log. In the simplest case, messages are written to a single logfile.

II. Types of logs

- **Computer or System Logs :** Detailed list of an application information, system performance, or user activities. A log can be useful for keeping track of computer use, emergency recovery, and application improvement.
- **Computer Log file** is a file that records either events that occur in an operating system or other software runs, or messages between different users of a communication software.

III. Why do we perform Log Analysis?

- Compliance with security policies
- Compliance with audit or regulation
- System troubleshooting
- Forensics (during investigations, or in response to Summons)
- Security incident response

IV. Benefits of Logs

- Logs provide clues about performance issues, application function problems, intrusion and attack attempt etc.
- The logs provide vital inputs for managing the computer security incidents, both for Incident Prevention and Incident Response Benefits
- When responding to computer security incident, logs provide leads to the activities performed over the system
- Facilitates cyber-crime investigation
    - Determine the activity
    - Determine the origin of attack

V. What are the sources of Logs?

- System Logs
- Web Server Logs
- Firewall logs
- Mail Server Logs
- Database Server Logs
- FTP Logs

VI. Windows Event Logs

The windows operating system is built on a complex architecture with which to handle events like logging on requires proper security measures.

The system logs and application logs can be used in a number of ways of writing specific events to the log. Windows also has a specific type of logging, the security logging system, which can only be written by the Local Security Authority Sub-System Service or LSASS.

The windows event logging system logs events like account logon, account management, directory service access, object access, policy change, privilege use, process tracking, and system events.

Windows Operating System maintains primarily three types of logs, which are as follows:

- **Security Logs**
  - valid and invalid login attempts
  - resource use such as creating, opening, or deleting files or other objects
- **Application Logs**
  - events logged by applications or programs
  - Depends on developer
- **System Logs**
  - events logged by system components
    Example: Functioning of drivers

How do we view the logs of a Windows Operating System?

Event Viewer, a component of Microsoft's Windows NT line of operating systems, lets administrators and users view the event logs on a local or remote machine.

To access Event viewer, follow these steps:

Goto **Start** > **Control Panel > Administrative Tools** > **Event Viewer** (or)

Goto **Start** > **Run** and type "eventvwr.msc**"**

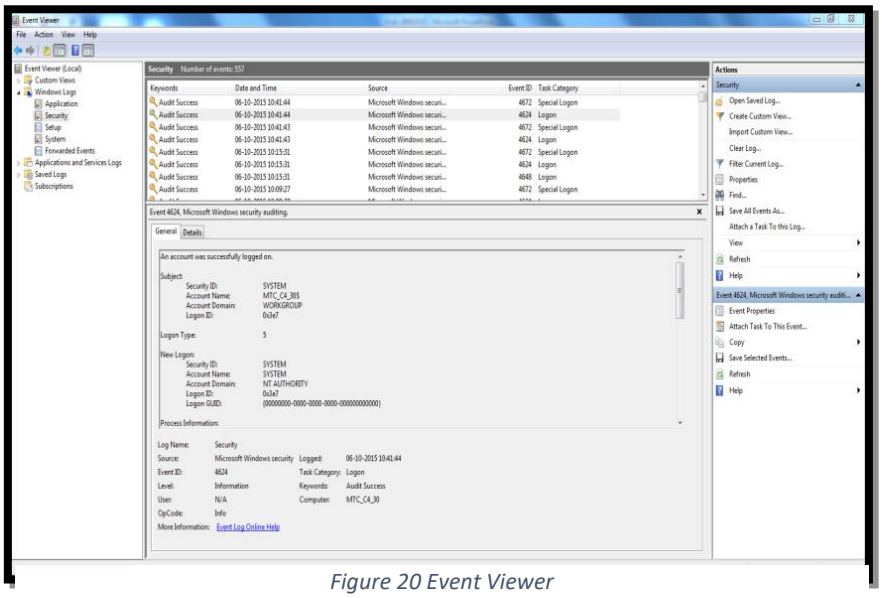


Figure 20 Event Viewer

In the mage: Windows Event Viewer, You can find all the events in the Event Viewer window.

You can also access them manually from the following location: C:\Windows\System32\Winevt

They are stored with these file extensions: *.evt, *.evtx

Some important Events which can be found in Logs, are as follows

Local logon attempt failures

- Event IDs 529, 530, 531, 532, 533, 534, and 537

Account Misuse

- Events IDs 530, 531, 532, and 533

Account Lockouts

- Event ID 539

Terminal Services attacks

- Terminal Services sessions can be left in a connected state that allows processes to continue running after the session is ended. Event ID 683 indicates when a user does not log out from the Terminal Services session, and Event ID 682 indicates when a connection to a previously disconnected session has occurred.

Policy Change

- Event ID 608: User right assigned
- Event ID 609: User right removed

Application Server Logs

- **Web Server logs**
  - Error Logs
  - Access Logs
- **Mail Server logs**
  - Connection Status
  - SMTP queues
  - Protocol Status (IMAP, POP3, SMTP)
- **FTP Server Logs**
  - Current logins
  - Commands executed
  - File uploaded and downloaded
- **Database Server Logs**
  - User activity
  - Objects accessed
  - Creation of new tables, databases, etc..

Note: Windows Operating System has a default firewall of its own and the logs generated by that even can be very helpful in analysis.

Firewall logs

- Firewall logs provide useful information about
  - The inbound and outbound packets
  - Information about particular servers e.g. Web Server
  - Packets which have been dropped
  - Alerts to the SA
  - Probing the system

By default, the Windows Firewall Logs are stored at: C:\Windows\System32\LogFiles\pfirewall.log (if 'C:' is your drive where Windows is installed)

Why and when is Firewall Logging useful?

To verify if newly added firewall rules work properly or to debug them if they do not work as expected.

To determine if Windows Firewall is the cause of application failures — With the Firewall logging feature you can check for disabled port openings, dynamic port openings, analyze dropped packets with push and urgent flags and analyze dropped packets on the send path.

To help and identify malicious activity — With the Firewall logging feature you can check if any malicious activity is occurring within your network or not, although you must remember it does not provide the information needed to track down the source of the activity.

If you notice repeated unsuccessful attempts to access your firewall and/or other high profile systems from one IP address (or group of IP addresses), then you might want to write a rule to drop all connections from that IP space (making sure that the IP address isn't being spoofed).

Outgoing connections coming from internal servers such as Web servers could be an indication that someone is using your system to launch attacks against computers located on other networks.

A **Windows Firewall log** looks as follows:

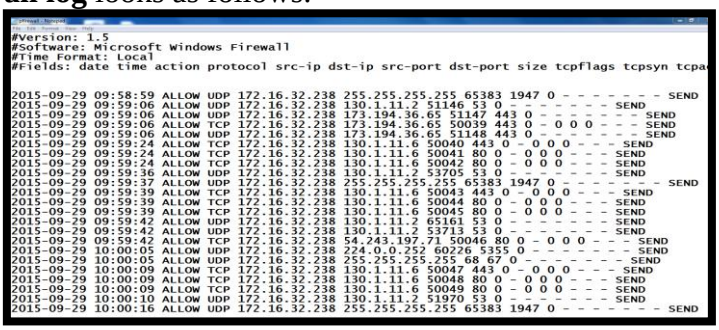


Figure 21 Windows Firewall Log

It can be explained as follows:

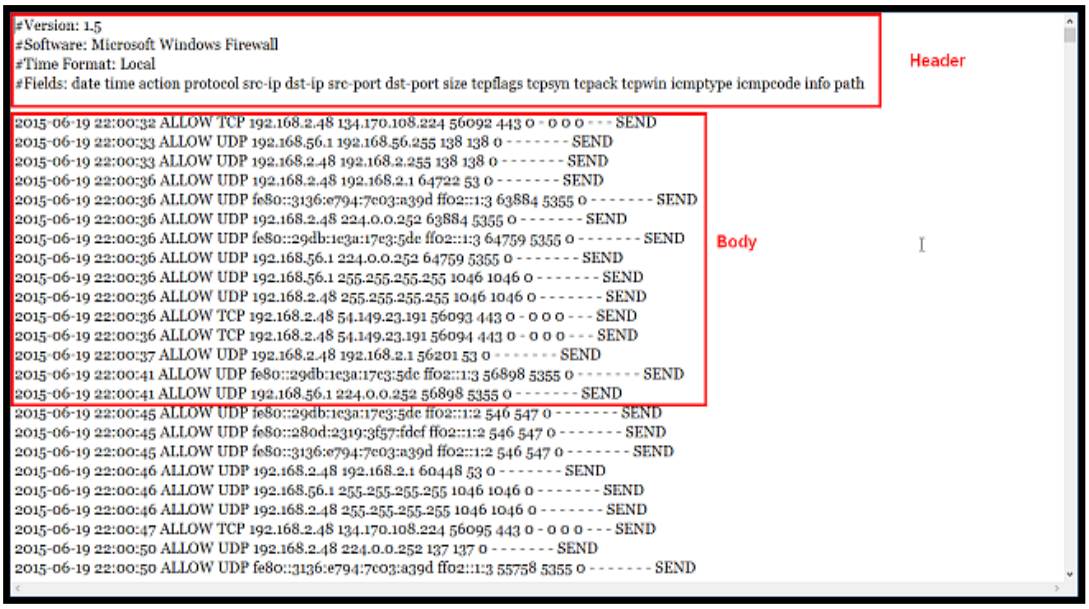


Figure 22 Firewall Logs

While dealing with Logs, there are a few problems associated with it too, such as:

- Non availability of proper logs
  - No auditing
  - Insufficient security
  - Poor management of Logs
- With logs available
  - Volume
  - Storage space, portability
  - Skills

5. Capturing Network Traffic

To identify the activites going on within a network, we need to monitor the traffic in & out of our network. Ideally, the organisations do not capture complete content of data flowing in & out of the network, but maintain something called as NetFlow records, which is probably the meta-data of the network activities. Technically 2 GB of a full traffic capture (PCAP) is equivalent to 2 MB of NetFlow records.

In this book, we will discuss about capturing **full traffic** (pcap). Let us begin by using TCPDUMP.

I. TCPDUMP

TCPDUMP is the pioneer among all open source packet sniffers. It was written in 1987 by Van Jacobson, Craig Leres, and Steven McCanne, all from the Lawrence Berkeley Laboratory. It is a command-line tool designed to operate under most versions of Unix including Linux, Solaris, AIX, Mac OS X, BSD, and HP UX. WinDump is a port of tcpdump for use in Windows systems. Most open source sniffers, today, are wrappers for libpcap (or something similar). The libpcap contains a set of system-independent functions for packet capture and network analysis. The tcpdump provides the user interface to communicate with libpcap, which talks with the network device driver, which talks to the network device. tcpdump, WinDump, and Wireshark rely on the Berkeley Packet Filter (BPF) in order to limit the output from libpcap or to specify which fields of information libpcap should record.

Using TCPDUMP

All applications of tcpdump should be done with root privileges. The Advanced Packaging Tool apt-get utility can be used to retrieve and install tcpdump in most Unix implementations.

You may install tcpdump using the following command(s):

```
sudo apt-get install tcpdump
```

For WinDump, you will need to download the WinDump binaries for WinPcap and WinDump from www.winpcap.org. Because WinDump is a port of tcpdump, most of the description of tcpdump also applies to WinDump. To simplify reading the material, I will only refer to tcpdump from this point on.

The most basic scenario is to use tcpdump to print all packets that are sent to or from a specific host. The host can be specified by name or IP address. For example, to print all packets arriving at or departing from the host *itsmerif*:

```
tcpdump host itsmerif
```

Suppose you don't want to see all traffic to or from *itsmerif*, and you want only traffic between *itsmerif* and two other hosts, *npalab1* and *npalab2*. Note the escape literal "\" that has to appear before the left and right parentheses. To print traffic between *itsmerif* and either *npalab1* or *npalab2*:

```
tcpdump host itsmerif and \( npalab1 or npalab2 \)
```

To print only the UDP packets between *npalab2* and any host except 172.16.21.226:

```
tcpdump udp host npalab2 and not 172.16.21.226
```

If your sniffer host were on the NPA campus, you could print all traffic between local hosts and hosts at NPA using the following:

```
tcpdump net npa-ether
```

or

```
tcpdump net 172.16.0.0/16
```

In this next example, the expression is surrounded by single quotes to

explicitly describe the order of execution. The example would capture all

File Transfer Protocol (FTP) traffic through Internet gateway npa-internal, which was seen by the tcpdump host. FTP uses different ports for control and data transfer, hence the ftp and ftp-data parameters.

```
tcpdump 'gateway  npa-internal and (port ftp or ftp-data)'
```

Limitations of tcpdump
tcpdump can be used for any general packet-monitoring mechanism in promiscuous mode. However, there are a few limitations to tcpdump.

1. tcpdump is a command-line utility. The user is required to know all the options for screening specific packets, so there is no easy user interface.

2. Packets blocked by a gateway firewall, router, or switch may not be seen. Modern switches may prevent hosts from seeing any traffic that is not destined for the host even when in promiscuous mode. In a switched network, tcpdump will only see traffic addressed to itself and broadcast traffic. In order to see more than that, the sniffer device will need to be connected to a Switched Port ANalyzer (SPAN) port.

3. To replay recorded traffic or perform additional analysis, use of other tools like tcppreplay or tcpopera is required

Alternative solution? Use Wireshark – It has been discussed later in this book.

II. NeSA (Network Session Analyzer)

Network Session Analyzer (NeSA) is a network forensics tool for analysing network packets. NeSA accepts packet dump file in the pcap format, generated using any third-party packet capturing tool.
NeSA provides an easy filter expression building facility to help novice users. Time zone-based analysis is incorporated into NeSA to make it capable of analysing dumps collected from different time zones. NeSA can reconstruct files from sessions that can then be exported for future references.

NeSA has a good Hex Viewer, which can indicate the data communication direction colours. Regular expression-based search is available to locate the evidence and evidence related items. The analysis state of a file can be saved and the analysis can be resumed from that point at a later time.

Features

- Data Reconstruction
- Searching and Filtering
- Data reconstruction of HTTP, SMTP, POP3 and FTP
- Regular expression based searching and filtering at data level and packet level
- Export data and packets
- Hexview, FIleview, Mailview and Thumbnail view for easy analysis
- Statistics view

Creating a dump file

Before we can analyze a file, it has to be created. To create a "packet dump" file, we have to capture packets passing through the network. To capture packets either you can use the packet capture feature in NeSA or you can use third party tools like Wireshark or tcpdump. To capture packets using NeSA select the Capture dialog by selecting Tools → Capture menu.

Preliminary Settings for NeSA

Before doing an analysis some preliminary settings has to be made in NeSA. For that select the settings menu. If you captured packets behind a proxy, you have to set the proxy port in the HTTP port settings. For "Squid" default HTTP port is 3128. Add this number or your proxy port in the HTTP port settings. The default value here is 80, you can add new values by separating the values by a comma.

By default the time zone is set to zero. If you are from a different time zone select your time zone from the drop down combo box. In the case of India it is +0530. New settings will take effect when we click the Ok button.

Loading a dump File

To load a dump file select the File → Open Menu. Make sure that files of type field in the dialog as "*.dump" or "all files". Select the dump file that you are captured or select the "demo.dump" file available with the NetForce suite. At this phase we have finished loading the dump file. Please note that, "demo.dump" file is captured from behind a Squid proxy, so please add "3128" to the "HTTP Ports" field in the "Settings" dialog (Tools → Settings)

Analyzing a dump file

After loading the dump file, a rich window with different controls are shown. At top left we have two tabs Rebuild and Analyse. In the "Rebuild pane" we have an "IP tree" in the left, a "Session list" in the right middle, "Hex view", "Thumbnail view", "File view" and "Mail view" in the bottom right. In the "Analyse pane" we have "Packet list" in the top, "Packet tree" in the middle and "Packet hex view" in the bottom. Here after we refer to all these windows in these names.

When you expand the IP tree you can see all the IPs that took part in the communication. By selecting a particular IP you can see all its communication sessions in the Session list. It is not mandatory that all these IPs have a communication session. Some IPs may have, some may not. Please select the IP "172.16.29.25" to see meaningful communications. Now you got sub-nodes for the IP "172.16.29.25", they are 172.16.50.1, 172.16.50.74. This means the system "172.16.29.25" has communicated with both 172.16.50.1, 172.16.50.74.

Each entry in session list is a communication session between two systems. That is, when we load a web page, the entire content of the web page may come in a single session or it may use different sessions for each file in the web page. Note that a web page is made with a group of files. When we select a particular entry in the session list, the data communicated in the session is shown in different format in the bottom panes.

Hex view lists the entire data in a hexadecimal form and in text form. As the data are communicated in two directions, in order to distinguish them we have used a coloring scheme. The data colored in bluish is the data coming from server, and the data colored in greenish is the data transferred from client to server.

The Thumbnail view lists the thumbnail form of the images available in the particular session. A session may contain more than one image. You can select one or more images and export them for future reference or to view in a native viewer.

The file view lists all files present the particular session in a very detailed way in a list control. By double clicking you can open the file in its native viewer or a right click will give more options to export or open in a different viewer. A check box is present in the left of each entry

in the File view. You can make multiple selections by selecting or deselecting it. A usual "Shift Key", "Control Key" combinations will also work in this for fast selection of multiple files. These selected items can be exported. To sort items in the list you just have to click on the header of a particular field. A simple click sorts the items in ascending order and a "Ctrl+Click" sorts the items in descending order.
If the selected session in Session view is a mail session (SMTP, POP3) then the mails present in it are listed in the Mail view. If the mail contains any images it will be shown in the Thumbnail view and all the files present in the mail session is shown in the File view. The Mail view lists each mail in a very detailed form in a list control. Selecting the particular entry in the list shows the mail content in bottom pane. This just works as a usual mail client like Outlook express or Evolution. This list can also be sorted as described in the case of File view. In this also you can select multiple mails and can exported in commonly used "mbox" format. This mbox-formatted file can later be loaded in to a mail client that supports mbox format. Right clicking on the particular mail in the list and selecting the Show Header menu will show the mail header.
Now we are back to session list. Items in session list can also be sorted by clicking the particular header. A simple click sorts the items in ascending order and a "Ctrl+Click" sorts the items in descending order. In session list also multiple items can be selected by checking/un-checking the check box available. You can use "Shift+Click" and "Ctrl+Click" combinations for fast selection. The multiple selected items can be updated in Thumbnail view, File view and Mail view. All the files, mail and images available in all these sessions will be updated in their corresponding views. To update the bottom view with multiple sessions, select the sessions you want update, right click on any of the session and select the "Update Bottom Views" menu. As stated above multiple items in the IP tree can also be selected by checking, un-checking the checkbox available in the left. Then right click on the IP tree, and select the update menu. All the sessions corresponding to these selected IPs will be added in the session list.

Session Filtering

We may not be interested in the whole sessions available in the dump file. We may be suspecting only a particular person, or the activities of users using a particular protocol like mail (POP3, SMTP), or on the data traffic in a particular time period. Using NeSA's filtering feature you can accomplish any of these, all of these or the combinations of these. To prepare a filter select tools Filter dialog or if you are familiar with the filter language of NeSA you can simply type the filter on the Combo box available on the toolbar of NeSA.
Lets assume that you want to filter out all "TCP" sessions. Type "tcp" in filter combo and press enter or click "Apply" button. All TCP session will be filtered out and all the windows will be updated to reflect the result. The present filter is displayed in the top right window of NeSA.

Searching

The next important feature of NeSA is searching. This too will aid you in analyzing a dump file like harvesting a password, or filtering out the sessions based on the search keyword. NeSA supports both normal and regex based search. Before applying a search, it is a good idea to filter out the data to be searched. i.e. if you give a search term NeSA will not search in whole dump file, instead it searches only in the filtered sessions. If you want to search in whole dump file sessions, clear the filter (clear the filter combo and press enter).

To start searching, select the tools → Search menu. Add the terms you what to search in this dialog using the add button. To start searching, select the find button in the search dialog.

Search hits are indicated by checkbox in red to the left of IP tree as well as the session list. When you select a search hit entry in the session list, the search hits will be indicated in the Hex view by marking the hit text in green. By pressing the F3 key and Ctrl+F3 keys you can navigate through the search hits in both hex view and Session list.

Packet Level Analysis

Next we are moving to packet level analysis. Select Analyse pane. Each Packet is represented in this as an entry with a short detailed description. When we select each entry in the packet list the bottom windows (Packet tree and Packet hex view) are updated. Packet tree helps us to see the packet in a very detailed manner. It lists each protocol present in the packet including its fields. When we select each field in the packet tree that area is marked in the packet hex view with a blue color. This is applicable only to the protocols known to NeSA. In this too we can perform the search using the search dialog. The search hits are marked as in the Packet hex view. Packet filtering is also supported in this, but the filtering language is different from the session filter. The filtering language used in this is the well-known language used by tcpdump. For example, if you want to filter out only UDP packets enter the filter text "udp" in filter combo and press enter. It will filter out all the UDP packets present in the dump file.

In this tool you can select multiple packets from Packet list and export it to view in other packet analysis tools. The combined use of these features should help a person to analyse and extract evidence from a dump file.

III. Wireshark

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

You could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course).

In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, all that has changed.

Wireshark is perhaps one of the best open source packet analyzers available today.

Features

The following are some of the many features Wireshark provides:

- Available for *UNIX* and *Windows*.
- *Capture* live packet data from a network interface.
- *Open* files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
- *Import* packets from text files containing hex dumps of packet data.
- Display packets with *very detailed protocol information*.
- *Save* packet data captured.
- *Export* some or all packets in a number of capture file formats.
- *Filter packets* on many criteria.
- *Search* for packets on many criteria.
- *Colorize* packet display based on filters.
- Create various *statistics*.

Using Wireshark

1. To run Wireshark in Linux, in monitor mode, we need to be a super user. andRun Wireshark with the following command:

sudo wireshark

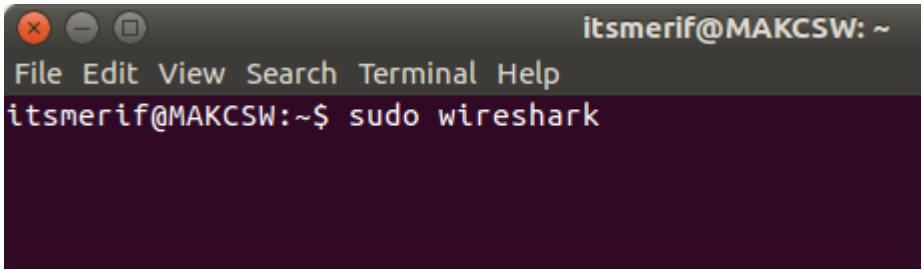


Figure 23 opening wireshark

Microsoft
Windows' Users can simply right-click on the Wireshark executable file and select "Run as Admin"

2. Select interface to start monitoring on; Here we are selecting our Wireless interface (wlan0)

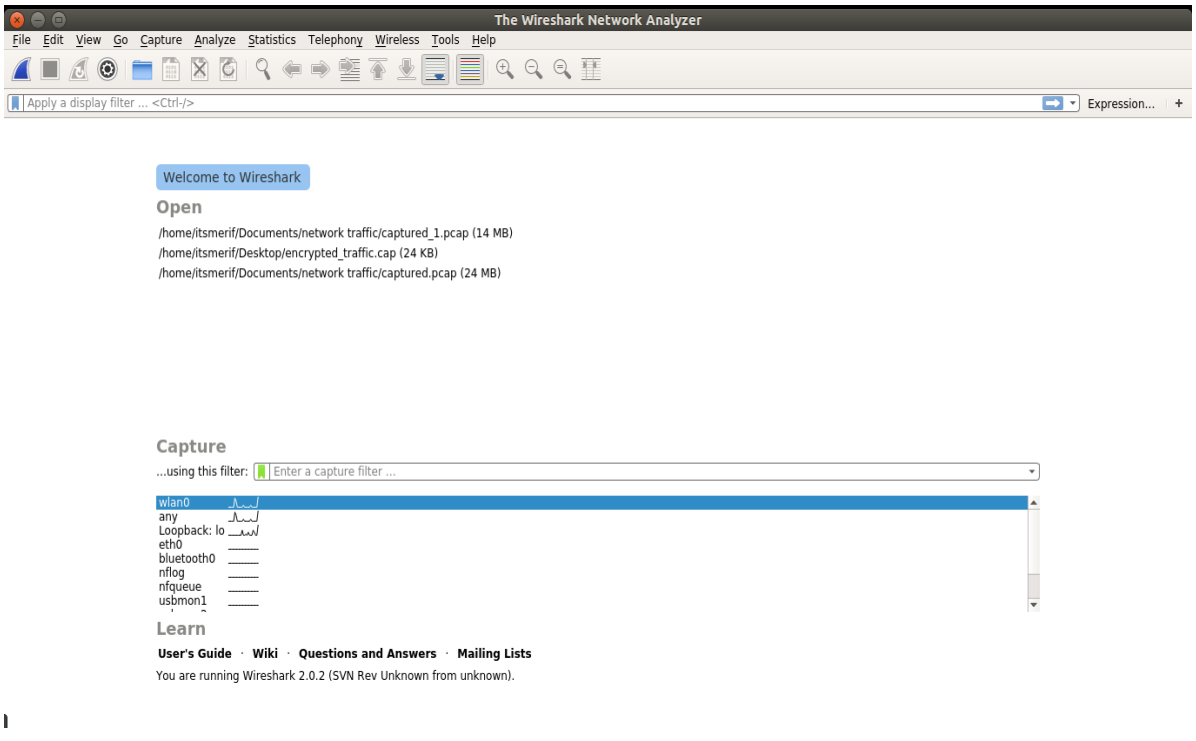


Figure 24 Wireshark

3. We can see capture in progress

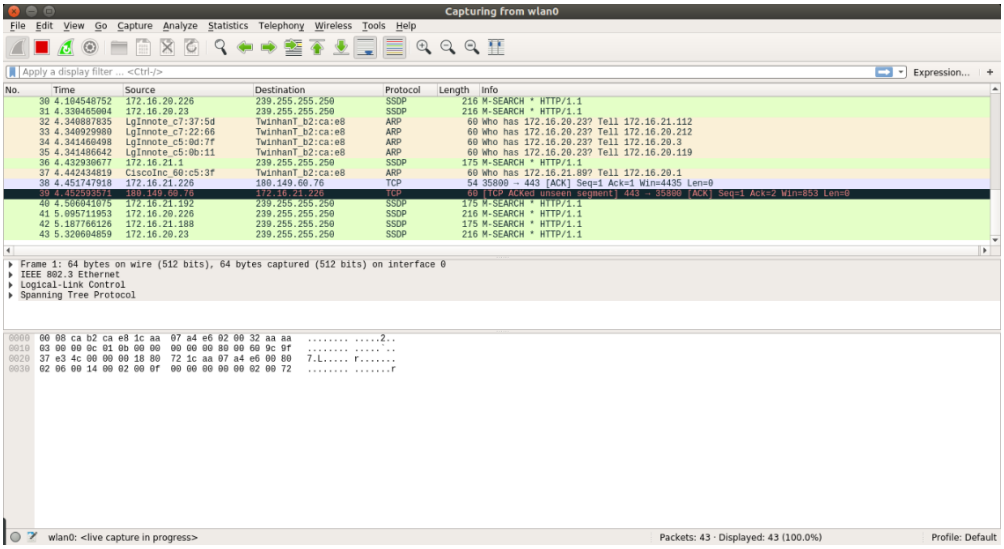


Figure 25 Capturing traffic in wireshark

4. To stop capturing traffic, click on the stop button (in red).

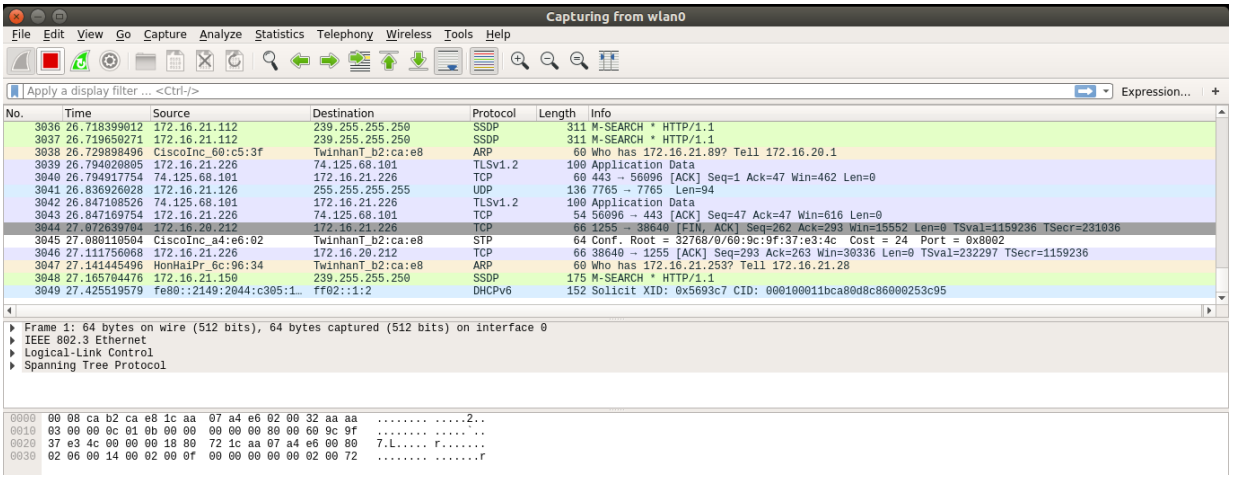


Figure 26 Capturing traffic in wireshark

5. To save the traffic dump, click on “File” in Menu bar, and select “Save as”.

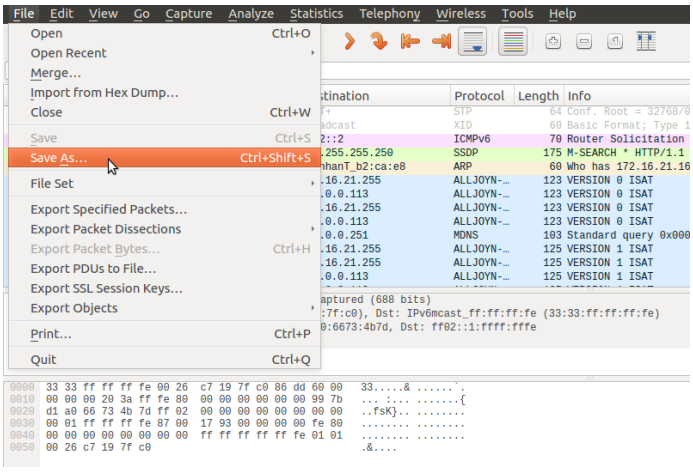
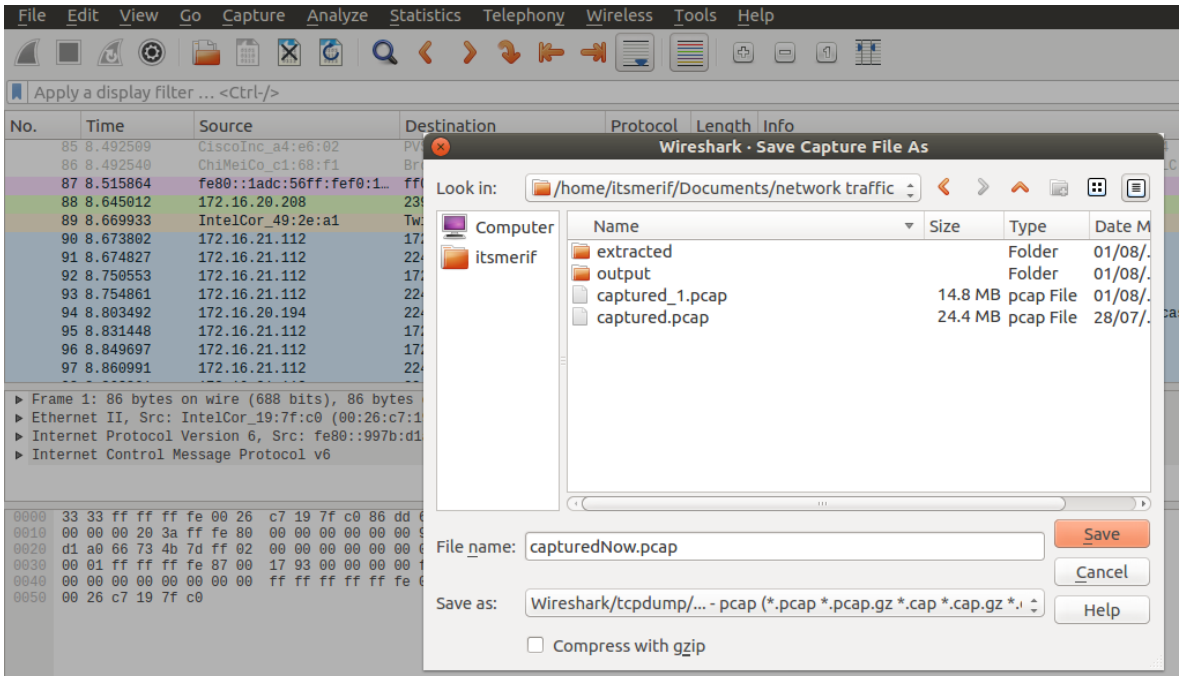


Figure 27 save captured traffic

6. Save the file with any name, and in any available formats shown below:



Now, we can also use these pcap file(s) to analyse using various other tools as this format is supported by almost all the network forensics tools.

Some basic filters in Wireshark

7. To filter out all traffic and show only ‘http’ traffic, type “http” in the filter tab as shown below:

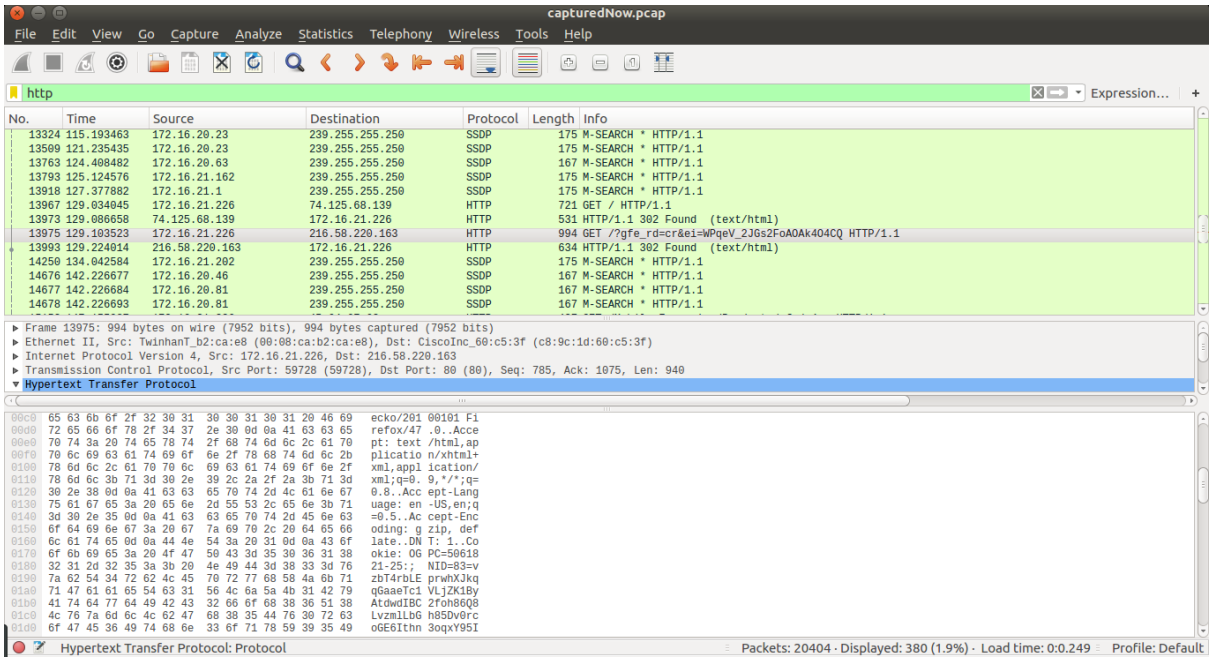


Figure 28 Filtering http traffic

Similarly, we can also use ftp, smtp, smb, telnet, etc to display respective traffic.

8. To find out if any user in the network had visited a website containing the keyword “google” in the domain, use the filter as follows:

http.request.uri contains "google"

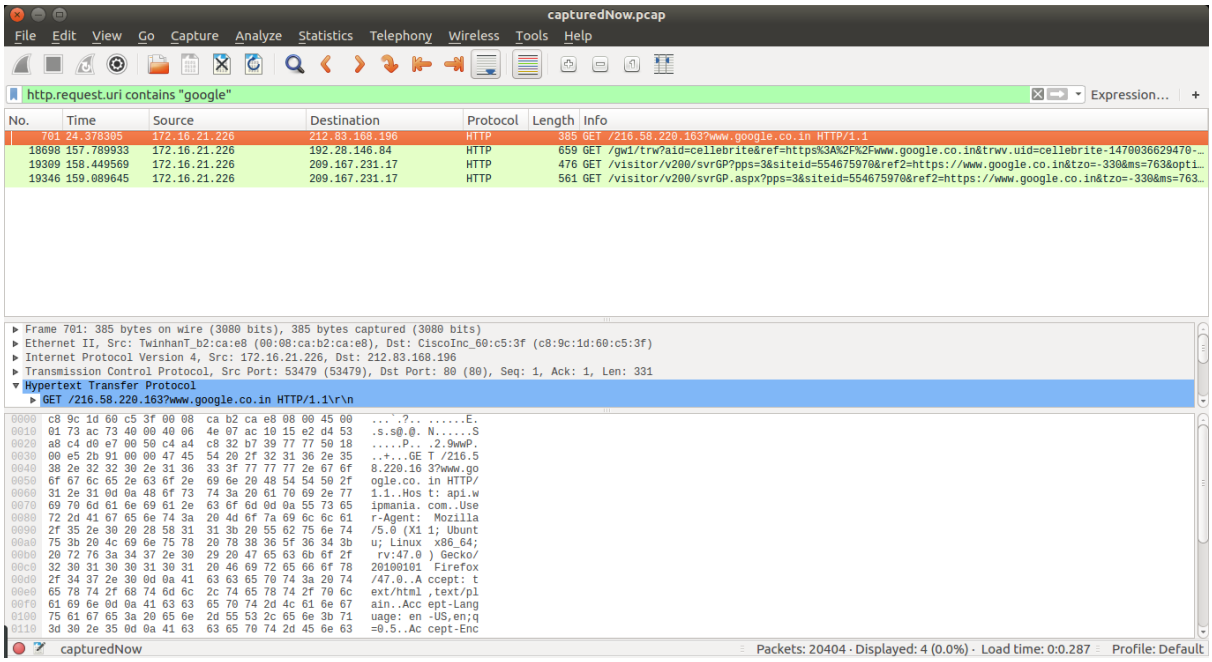


Figure 29 Filter with keyword

9. Similarly, to find out if any user in the network had visited YouTube website, or any website containing the keyword “youtube”, apply the following filter:

http.request.uri contains "youtube"

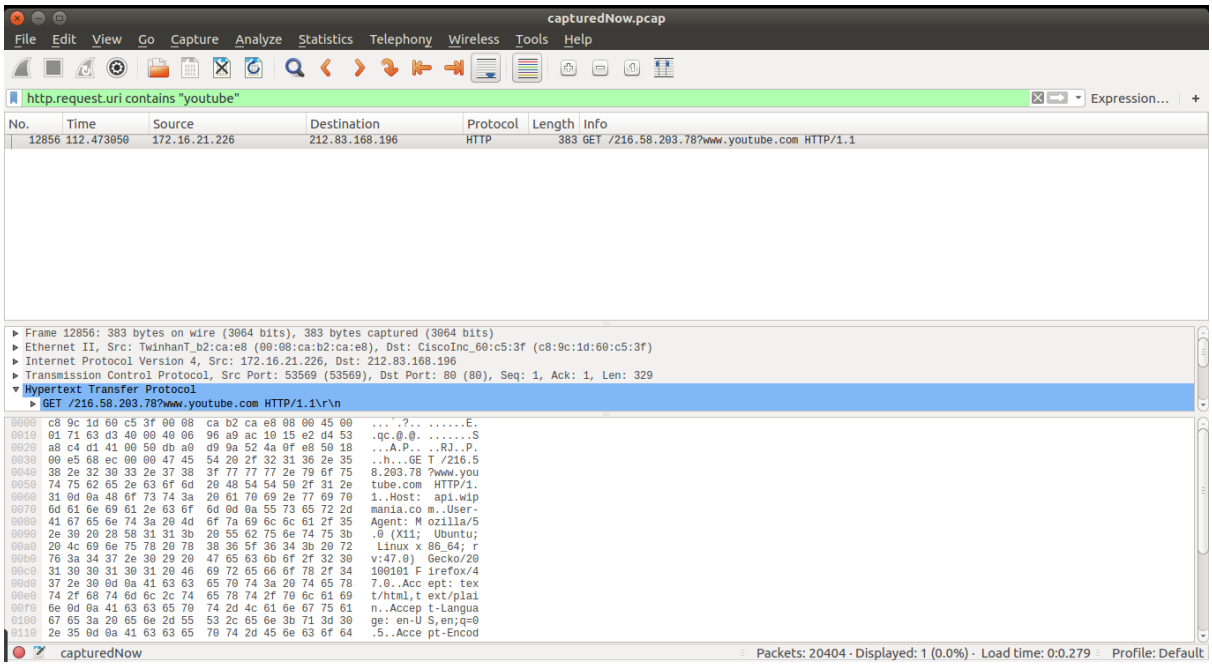


Figure 30 Filter with keyword

10.To find out if any network activity was done from a device having a specific MAC address, apply the following filter:

eth.addr==<MAC address of target device>

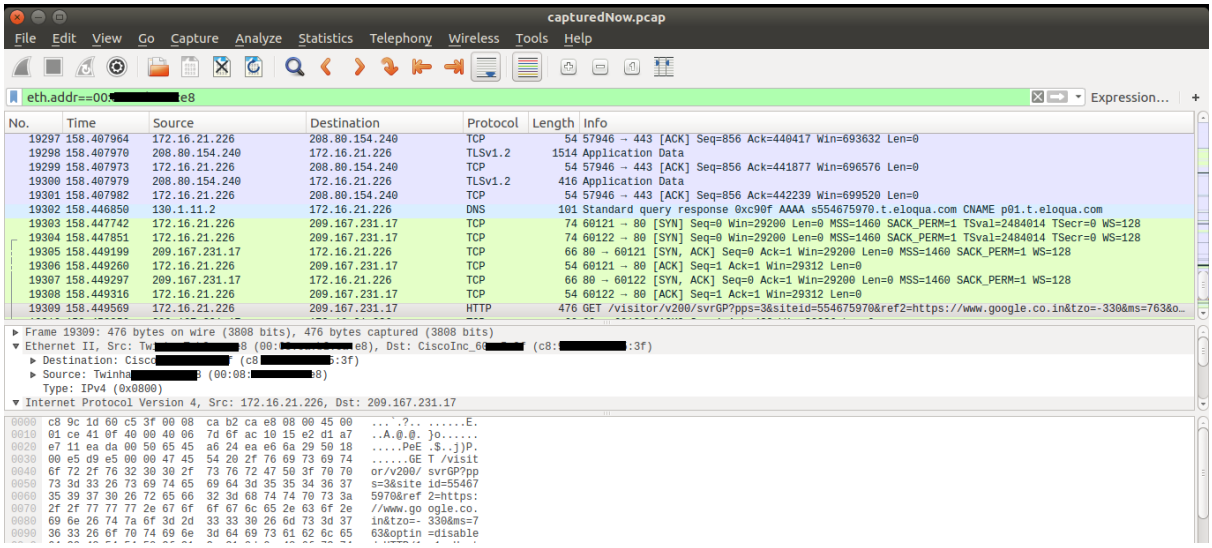


Figure 31 MAC address filter

FTP Analysis using Wireshark

1. Open the pcap file in Wireshark

2. To find out if there is any FTP traffic in it, just apply the filter '**ftp**' in the display filter.

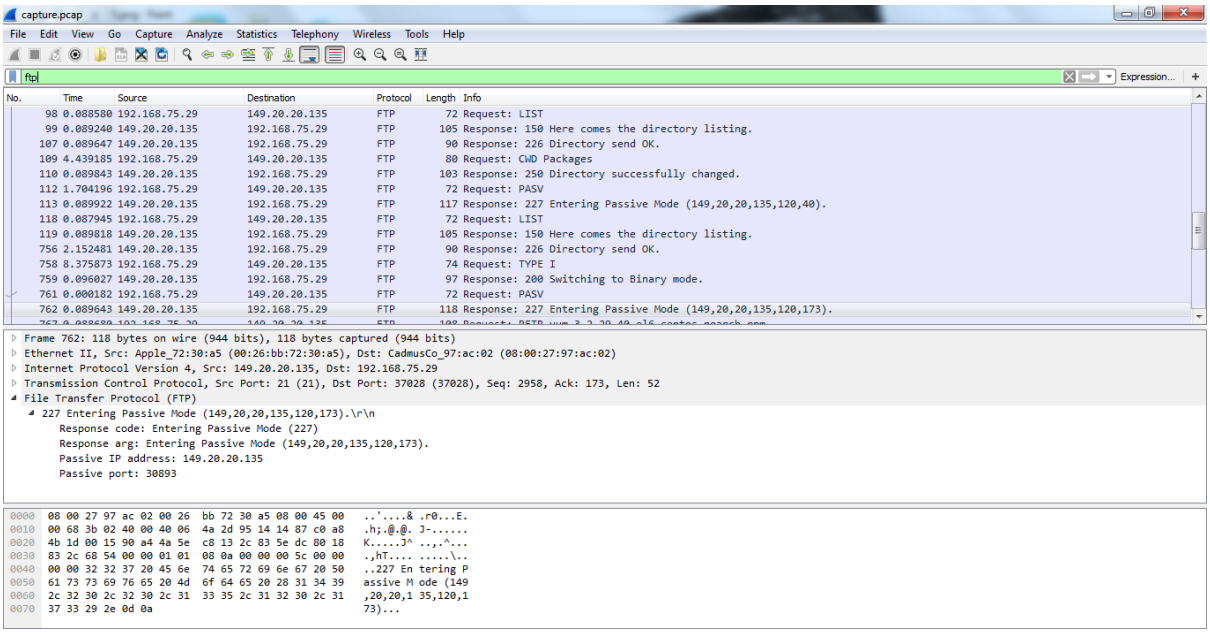


Figure 32 FTP filter

3. Now when you see FTP traffic, find out if any data was downloaded by a user. To find out, just apply the display filter **ftp.request.command=="RETR"**

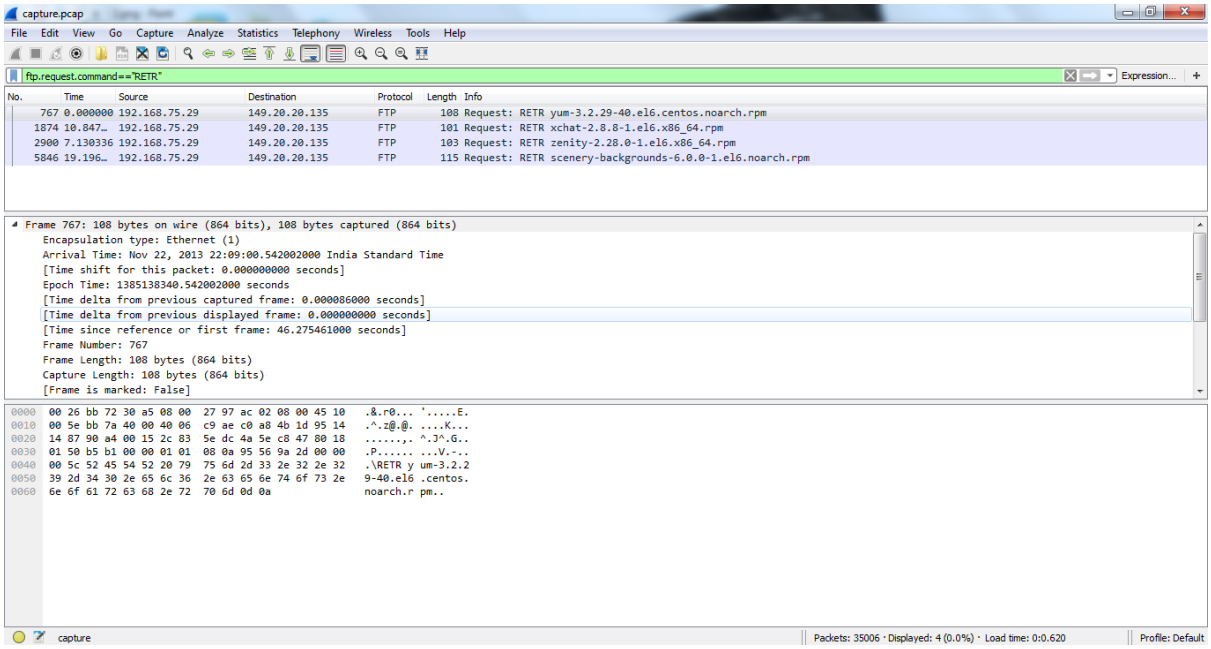


Figure 33 Filter FTP download

4. You can see a retrieve (RETR) request by a few packets, let us see the packet number 767 which was the first instance.

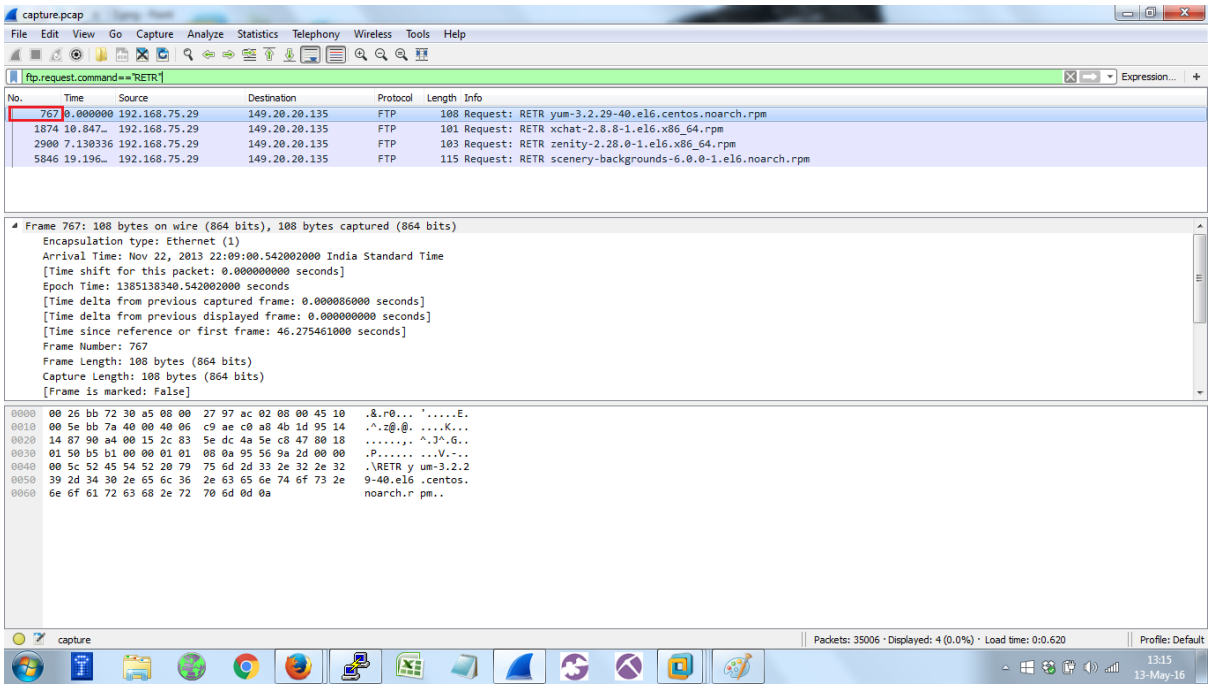


Figure 34 FTP filtering

5. Now clear the filters and scroll down to a couple of packets before 767. If you look at packet number 762, the "info" tab gives some information. If you click on the "packet details" pane and on "File Transfer Protocol (FTP)" as shown in the picture below, you will find a field called "passive port". Note down the port number of that field.

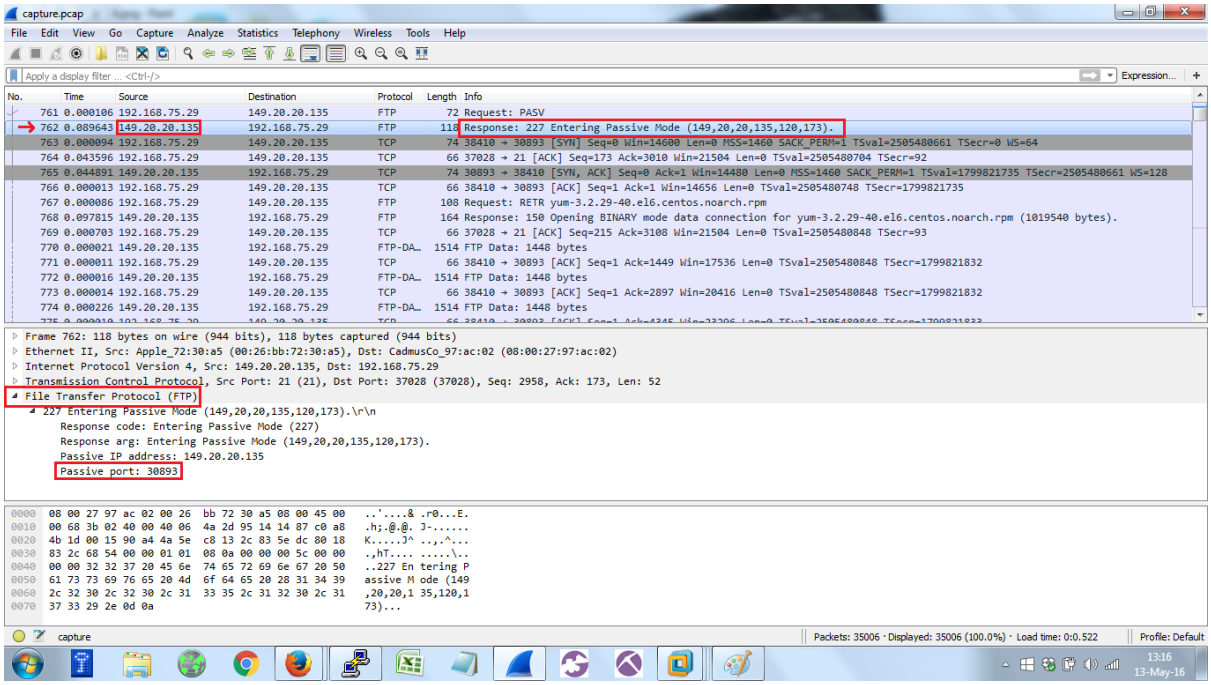


Figure 35 FTP Filtering

SMTP Analysis using Wireshark

1. Load the captured pcap file in Wireshark.

2. Type “smtp” in filter to show only SMTP traffic.

3. Click on the frame having the protocol 'Internet Message Format' (IMF) to see the content of the mail.

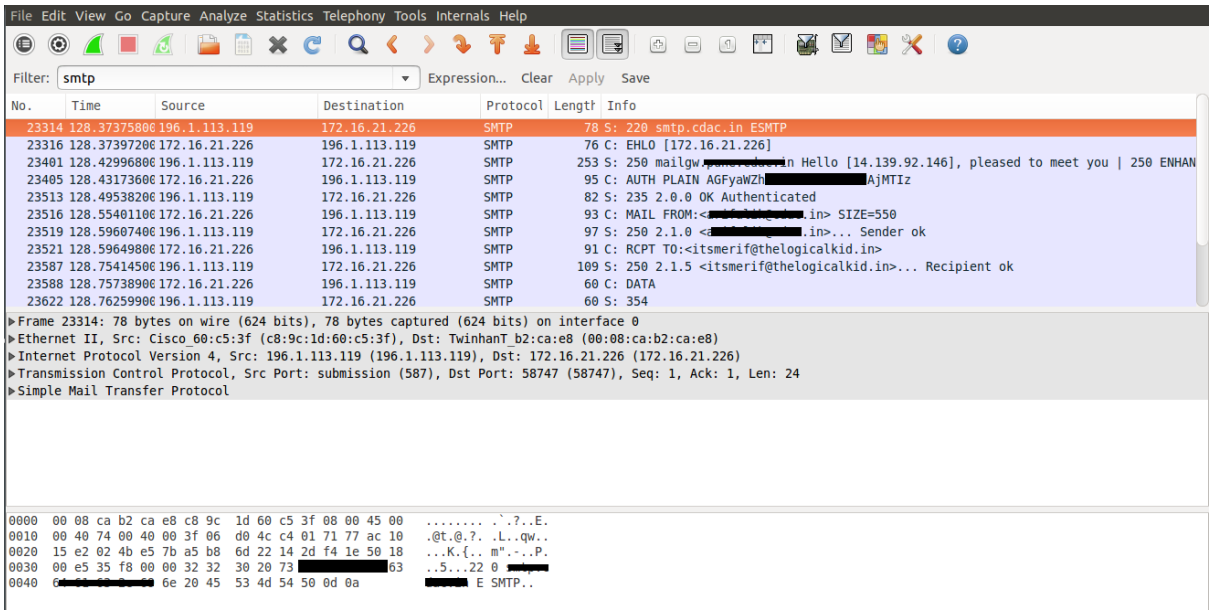


Figure 36 SMPT filter

4. Notice the frames/packets which contain info as “AUTH”

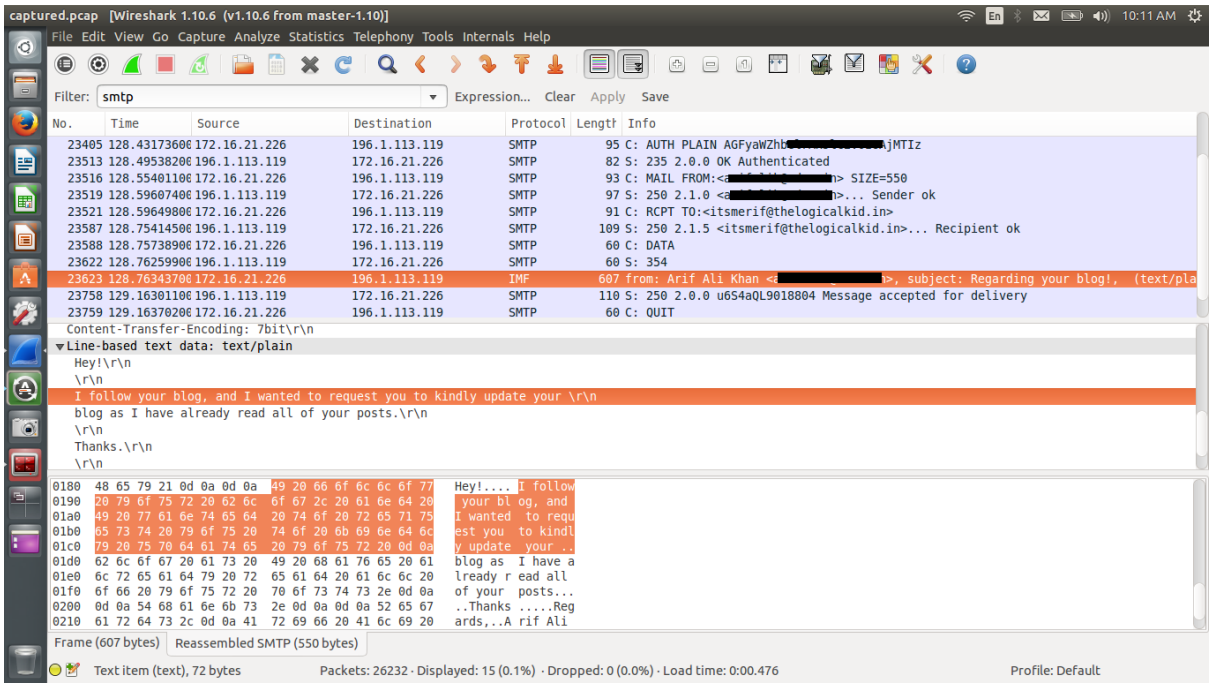


Figure 37 auth

5. Right click on the packet containing “AUTH” in info, and select “**Protocol Preferences**” > “**Decrypt AUTH parameters**”

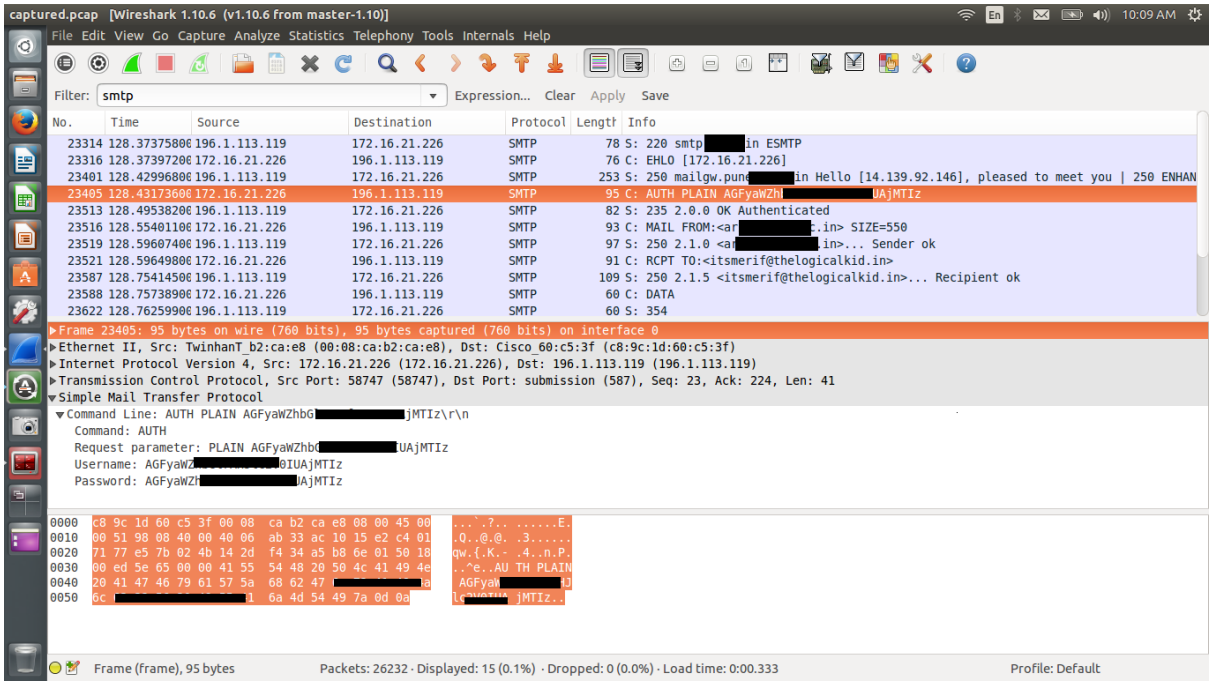


Figure 38 Decrypt AUTH parameters

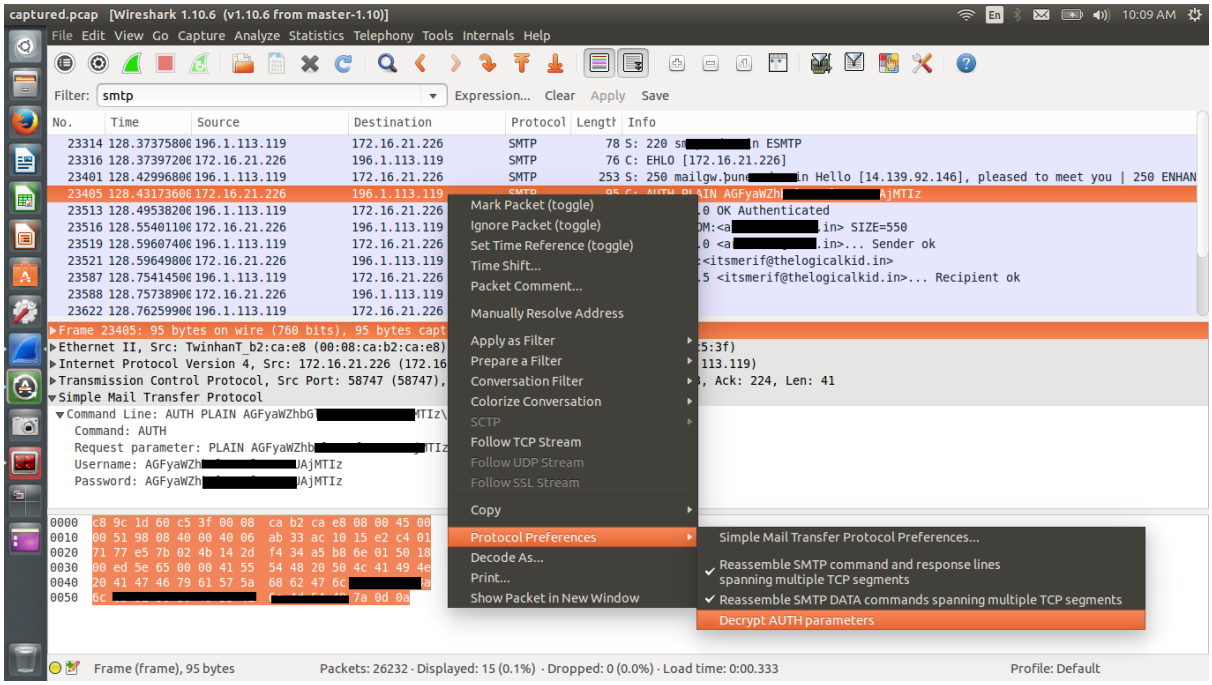


Figure 39 Protocol Preferences

6. You can now see the credentials in clear text.

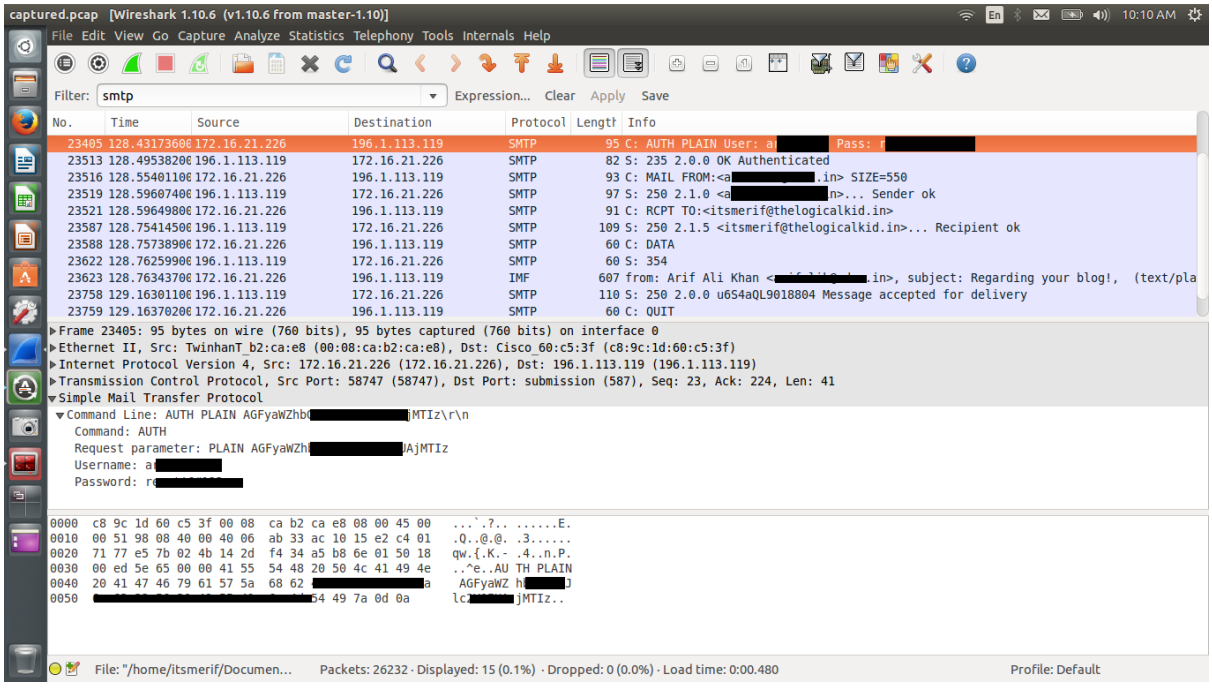


Figure 40 Credentials in clear text

SSL Decryption using Wireshark

What is SSL?

SSL (Secure Sockets Layer) is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral. SSL is an industry standard and is used by millions of websites in the protection of their online transactions with their customers.
To be able to create an SSL connection a web server requires an SSL Certificate. When you choose to activate SSL on your web server you will be prompted to complete a number of questions about the identity of your website and your company. Your web server then creates two cryptographic keys - a Private Key and a Public Key.
The Public Key does not need to be secret and is placed into a Certificate Signing Request (CSR) - a data file also containing your details. You should then submit the CSR. During the SSL Certificate application process, the Certification Authority will validate your details and issue an SSL Certificate containing your details and allowing you to use SSL. Your web server will match your issued SSL Certificate to your Private Key. Your web server will then be able to establish an encrypted link between the website and your customer's web browser.
The complexities of the SSL protocol remain invisible to your customers. Instead their browsers provide them with a key indicator to let them know they are currently protected by an SSL encrypted session - the lock icon in the lower right-hand corner, clicking on the lock icon displays your SSL Certificate and the details about it. All SSL Certificates are issued to either companies or legally accountable individuals.

What is an SSL Certificate and How Does it Work?

SSL Certificates have a key pair: a public and a private key. These keys work together to establish an encrypted connection. The certificate also contains what is called the “subject,” which is the identity of the certificate/website owner.

How Does the SSL Certificate Create a Secure Connection?

When a browser attempts to access a website that is secured by SSL, the browser and the web server establish an SSL connection using a process called an “SSL Handshake” (diagram 1). Note that the SSL Handshake is invisible to the user and happens instantaneously.
Essentially, three keys are used to set up the SSL connection: the public, private, and session keys. Anything encrypted with the public key can only be decrypted with the private key, and vice versa.
Because encrypting and decrypting with private and public key takes a lot of processing power, they are only used during the SSL Handshake to create a symmetric session key. After the secure connection is made, the session key is used to encrypt all transmitted data.

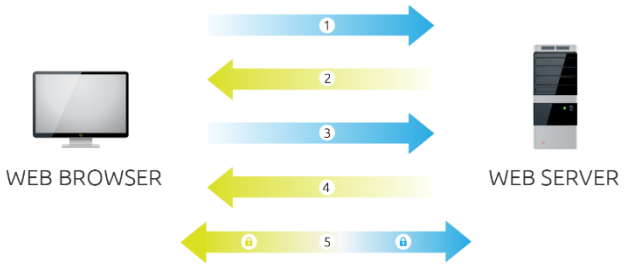


Figure 41 Web browser - server

Browser connects to a web server (website) secured with SSL (https). Browser requests that the server identify itself.
The sequence of steps is as follows:

A. Server sends a copy of its SSL Certificate, including the server’s public key.

B. Browser checks the certificate root against a list of trusted CAs and that the certificate is unexpired, unrevoked, and that its common name is valid for the website that it is connecting to. If the browser trusts the certificate, it creates, encrypts, and sends back a symmetric session key using the server's public key.
C. Server decrypts the symmetric session key using its private key and sends back an acknowledgement encrypted with the session key to start the encrypted session.
D. Server and Browser now encrypt all transmitted data with the session key.

Agenda:

You might have heard quite often that the Government is able to 'read' all our 'secure' traffic too, and they do it by using some secret keys, or so. Did you hear the same?

Well, thats the way they do it! Now to understand how they do it, read on! First things first. To make this happen, the Government needs to have the decryption key, probably the private key. We will emulate the same here.

Things we require:
i). SSL Traffic dump
ii). Decryption key (private key)

iii). Wireshark (Here we have used version 2.0.0rc3)

1. Run Wireshark, and open the sample SSL traffic dump file (.cap/.pcap).

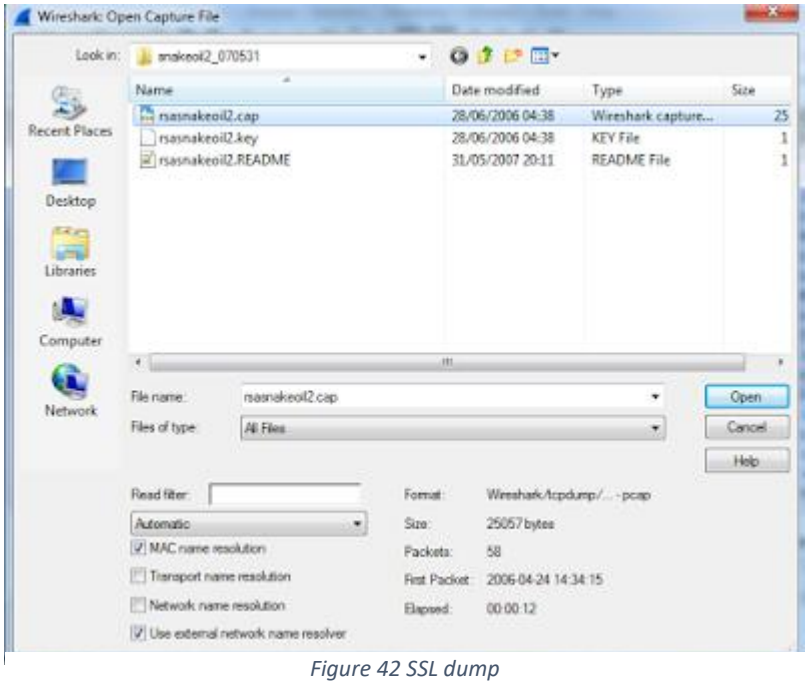
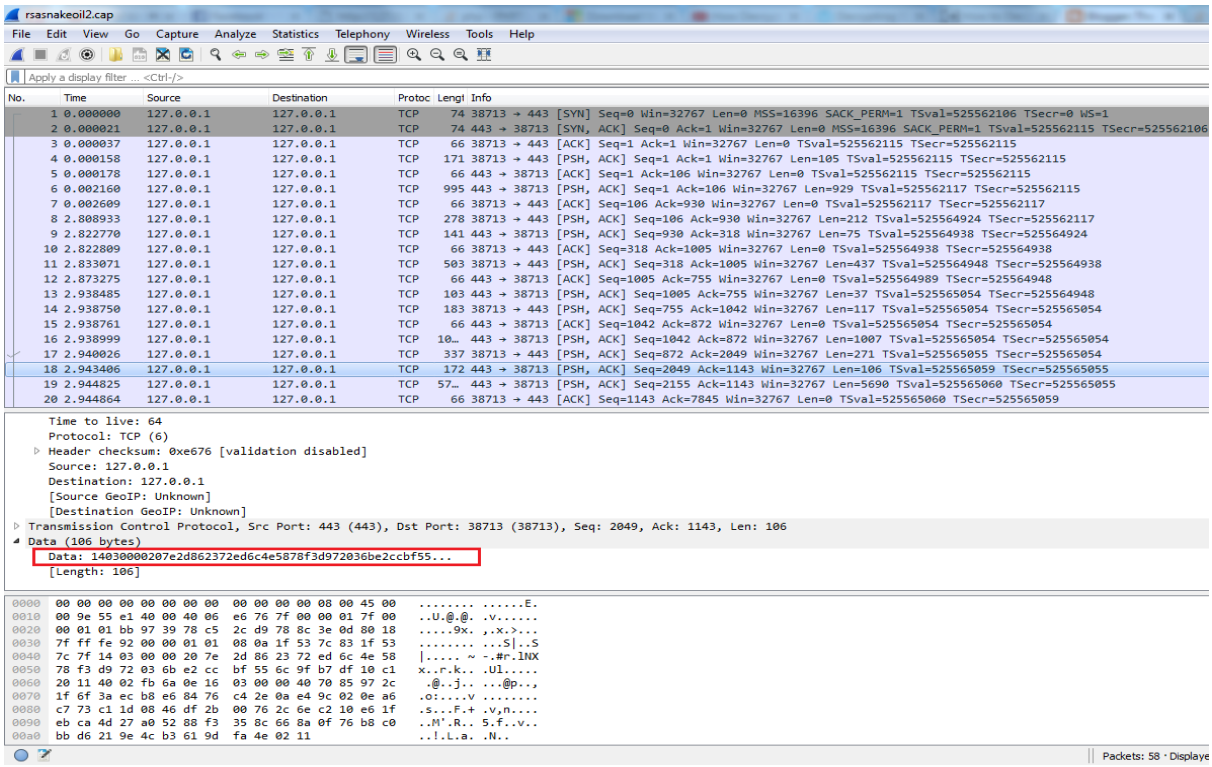


Figure 42 SSL dump

The data gets loaded in Wireshark. you can see some (encrypted) content.



In the above picture you can see (under **data**) that the data is encrypted. So, what next?

2. Click on **Edit** -> **Preferences** -> **Protocols** -> **SSL** and click on 'Edit' beside "RSA keys list" as follows:

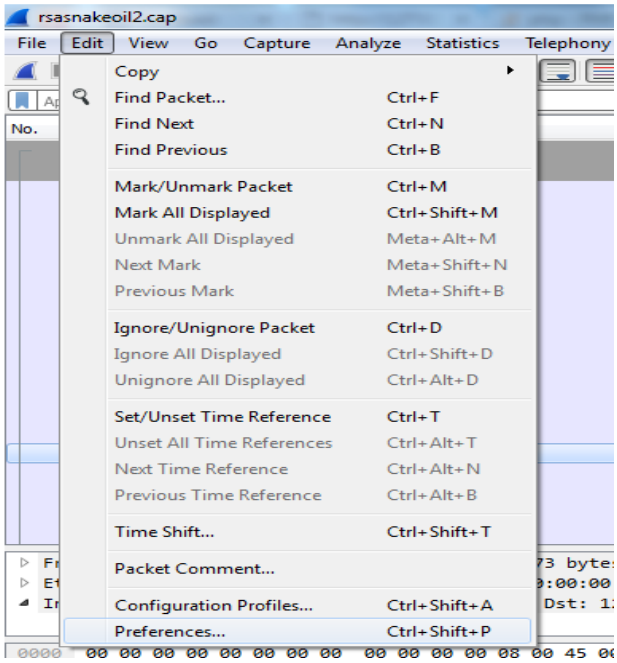


Figure 43 RSA key list step 1

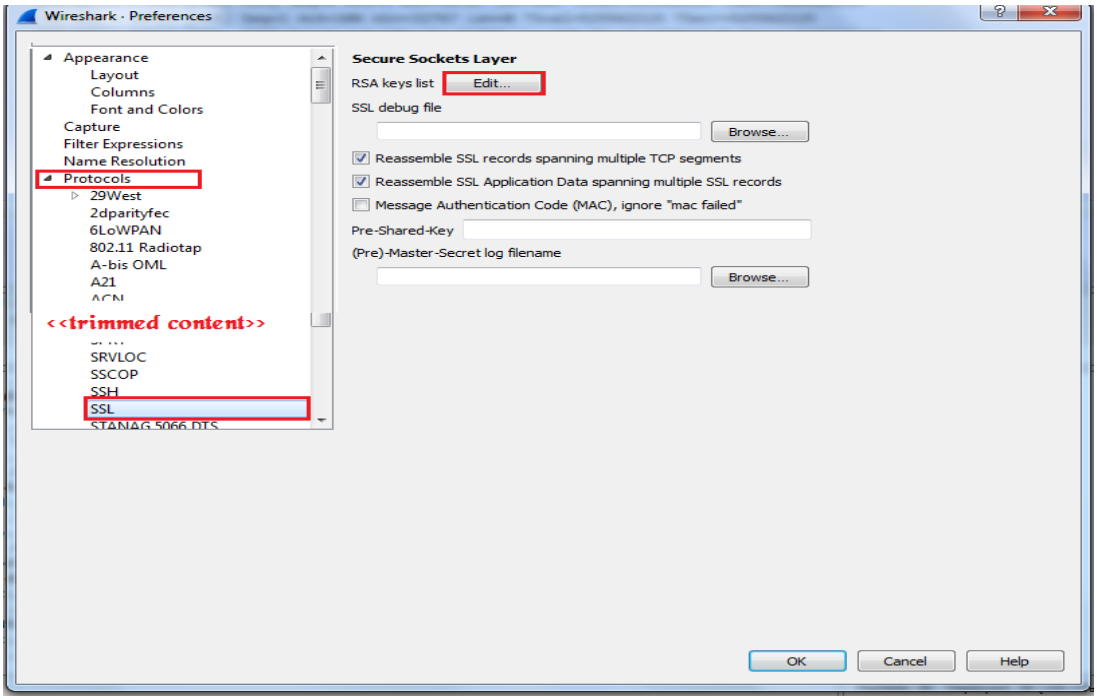


Figure 44 RSA key list Step 2

3.
Click on "+" and fill the fields *IP Address*, *Port*, *protocol* with **127.0.0.1**, **443** and **http** respectively.

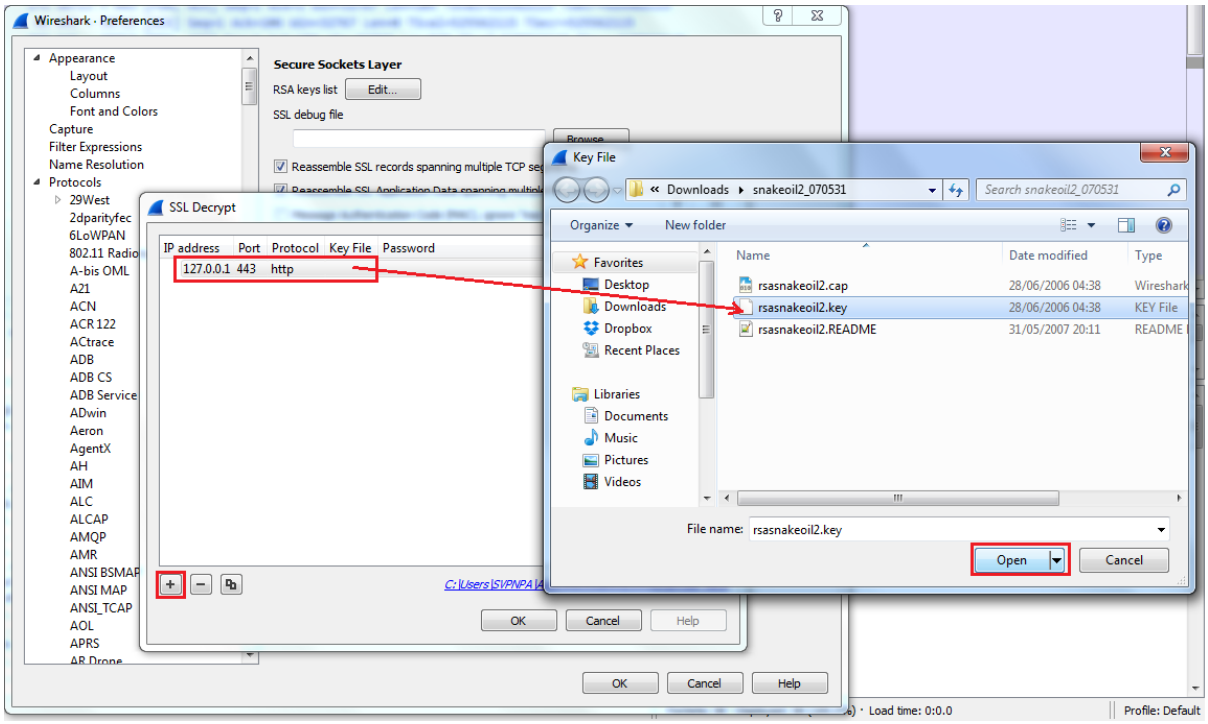


Figure 45 Key file

4. Now double click on the 'Key file' tab, browse and select the private key file.

Now you will see some extra tabs in the Wireshark window, as follows:

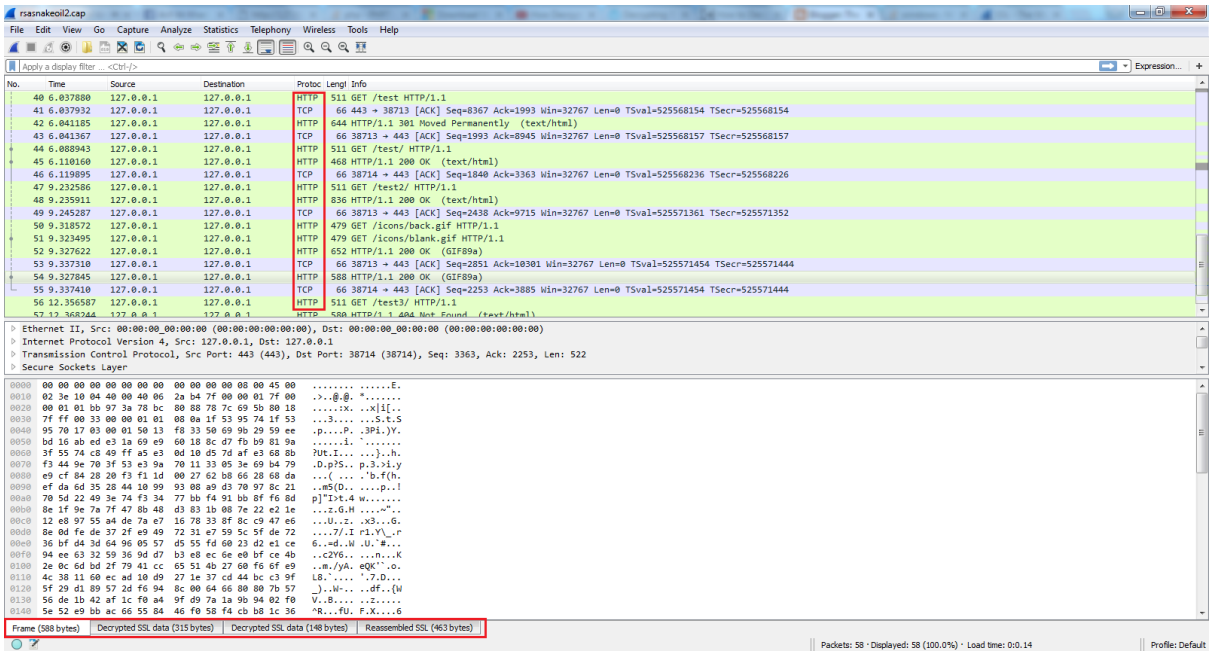


Figure 47 SSL stripping

You can see that the SSL traffic was decrypted and *explained* in clear HTTP. Lets compare a couple of frames as to how they looked when encrypted, and also after decryption.

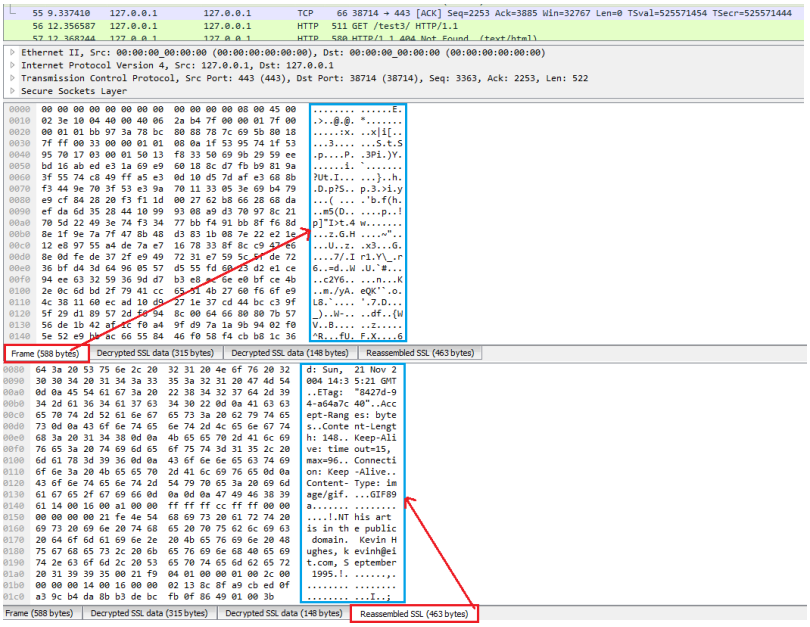


Figure 46 Decrypted content

Now whenever somebody says that Government is able to decrypt all your 'secure' data sent to an *xyz* website, you now know that they probably have the private key used by that website.

Those who have captured encrypted traffic, and also have the decryption (private) key associated with that website, can use the above steps to decrypt data and find relevance in your investigation. The rest, can practise from the sample dump provided at:

https://wiki.wireshark.org/SampleCaptures?action=AttachFile&do=view&target=snakeoil2_070531.tgz

6. Tools for retrieving content from network traffic

I. NetworkMiner

NetworkMiner is a Network Forensic Analysis Tool (NFAT) for Windows (but also works in Linux / Mac OS X / FreeBSD). NetworkMiner can be used as a passive network sniffer/packet capturing tool in order to detect operating systems, sessions, hostnames, open ports etc. without putting any traffic on the network. NetworkMiner can also parse PCAP files for off-line analysis and to regenerate/reassemble transmitted files and certificates from PCAP files.
Ths tool makes it easy to perform advanced Network Traffic Analysis (NTA) by providing extracted artifacts in an intuitive user interface. The way data is presented not only makes the analysis simpler, it also saves valuable time for the analyst or forensic investigator.
It has, since the first release in 2007, become a popular tool among incident response teams as well as law enforcement. NetworkMiner is today used by companies and organizations all over the world.
This tool can extract files and certificates transferred over the network by parsing a PCAP file or by sniffing traffic directly from the network. This functionality can be used to extract and save media files (such as audio or video files) which are streamed across a network from websites such as YouTube. Supported protocols for file extraction are FTP, TFTP, HTTP, SMB, SMB2 and SMTP.

You can download the tool from: **http://www.netresec.com/?page=NetworkMiner**

Steps for using NetworkMiner for extracting multimedia files from Network Traffic

1. Double-click on the NetworkMiner.exe file

2. The Interface of the tool will be displayed as follows.

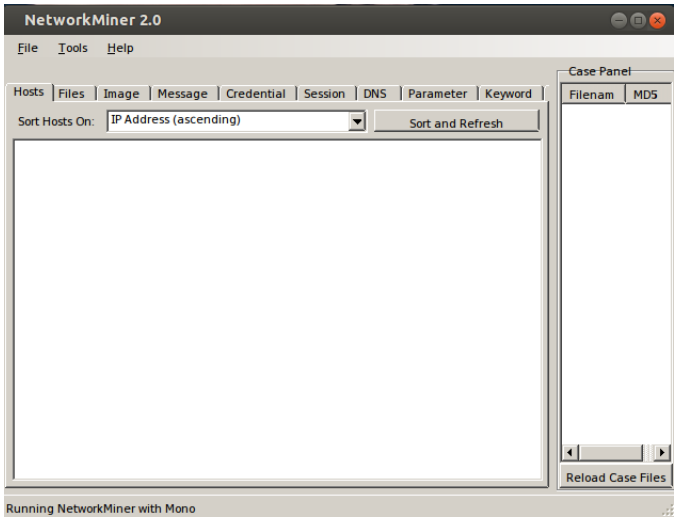


Figure 48 Network Miner Interface

3. Click on “File”in Menu bar and select “Open”.

4. Select the appropriate file. NetworkMiner supports all popular packet capture file formats. Let us analyse the .pcap file which we have – i.e captured_1.pcap

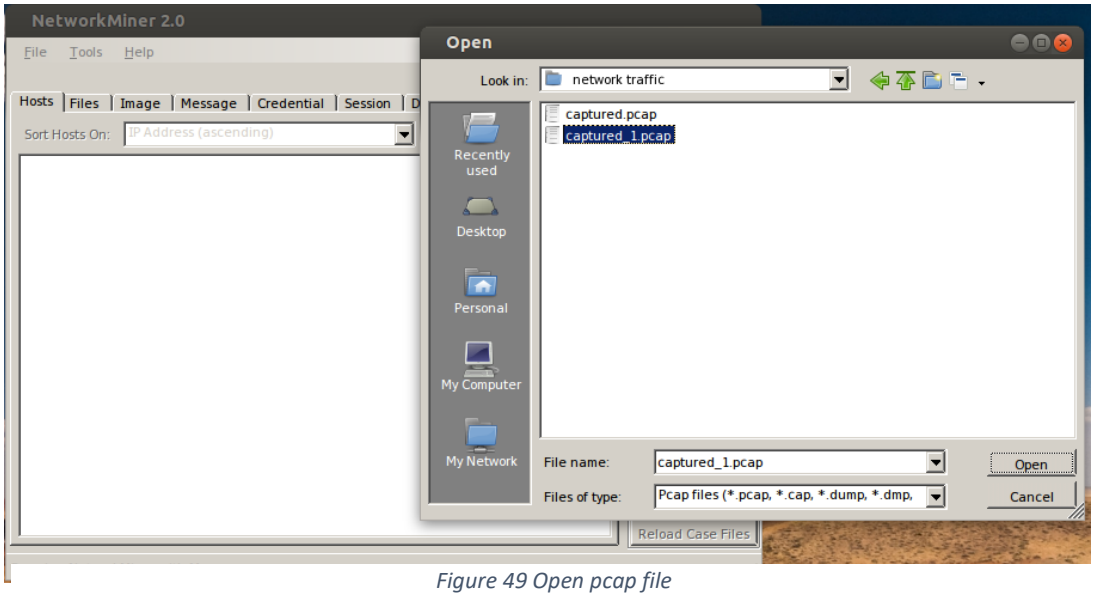


Figure 49 Open pcap file

5. Wait until the pcap file gets loaded and parsed by NetworkMiner.

6. Once the file gets loaded, we can see a variety of data available in the various tabs of the interface, like “Hosts”, “Files”, “Images”, “Credentials”, etc.

Let us explore important tabs first.

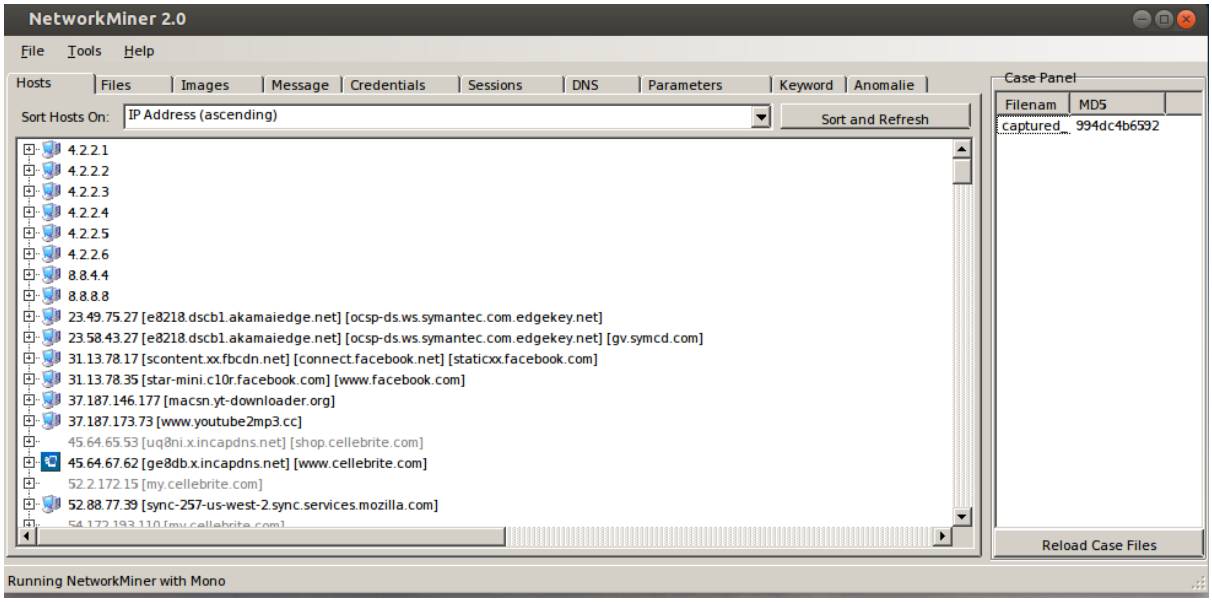


Figure 50 Data analysis in NetworkMiner

7. The “Files” tab shows us if any files were created/downloaded/uploaded during a session.

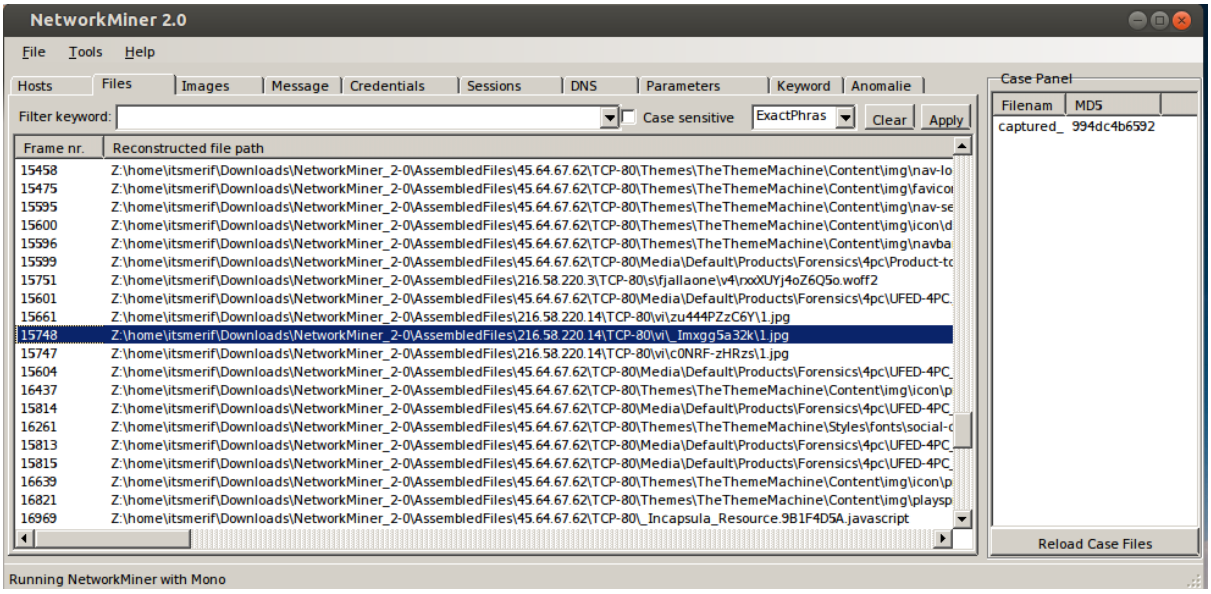


Figure 51 Files Tab-Networkminer

8. The "Images" tab shows the thumbnails of the images which were downloaded/uploaded during a session. When we hover our mouse over any image, it will show the source of the image from where it was downloaded, and the destination to which the image was served.

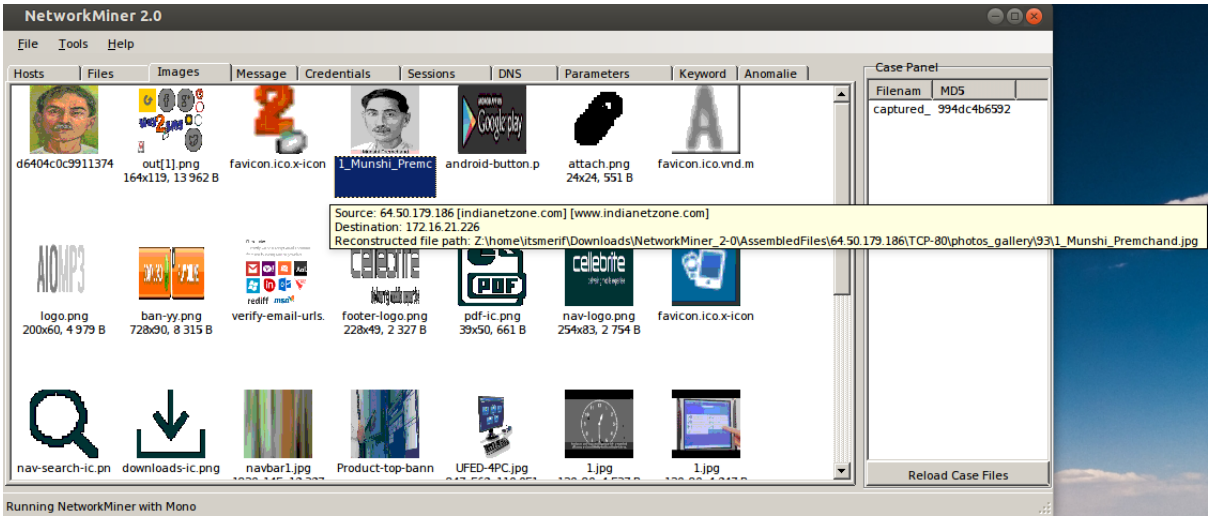


Figure 52 Image tab - NetworkMiner

In the above screenshot, we can see that the image was fetched/downloaded from a source 64.50.179.186 which apparently belongs to a website indianetzone.com, and was requested by the destination IP 172.16.21.226 which belongs to a host within the internal network of the organisation from which the image was requested.

To view the complete image, right-click on any image and select "Open Image" to open with a default image-viewer or a web browser.

9. This tool has the ability to even identify authentication sessions and/or credentials from the pcap file. This will help us find out details of any website logins by the devices in our Network.

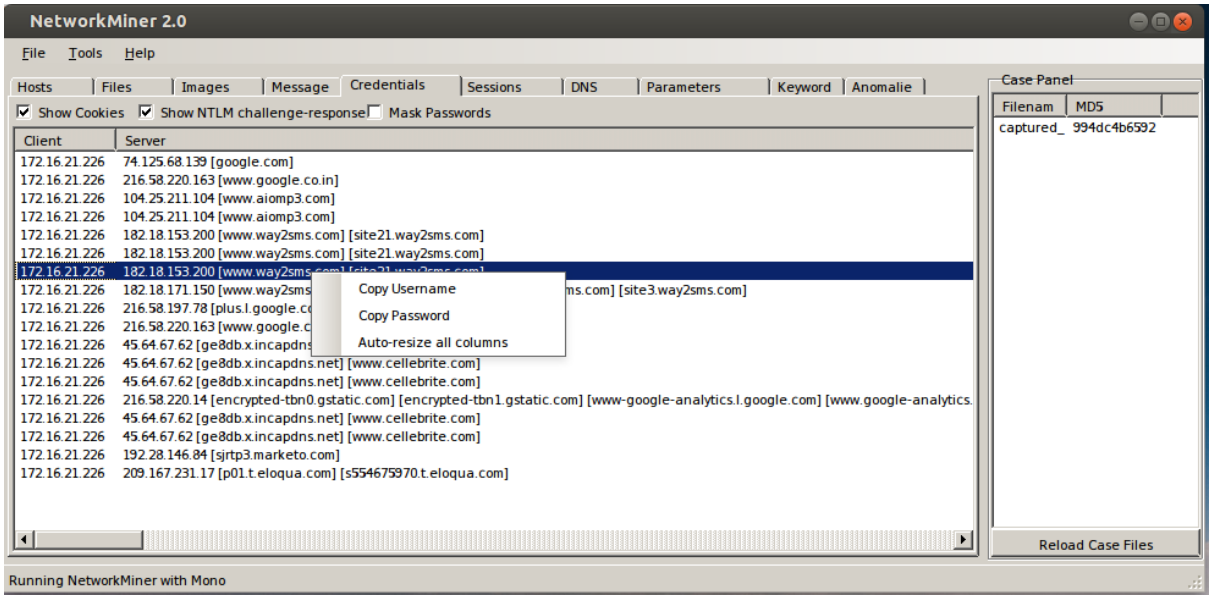


Figure 53 identification of authentication sessions and/or credentials

Although NetworkMiner is good as it quickly parses the data, but the free version is limited to analysis of a lesser size of traffic. This can be used if the network dump (pcap) is of a lesser size.

2.Dark Web and Cryptocurrency

1. Dark Web Introduction

Internet is nothing but huge library of websites which people use for various purposes. We use search engines like Google, Bing, DuckDuckGo for accessing these websites. But not all websites are accessible by these search engines or standard browsers like Edge, Chrome or Firefox. This restriction on accessing websites creates different layers of Internet. The hidden layer which needs special browser and search engine is known as Dark Web.

I. Layers of the Internet

There are basically three layers of Internet depending on how they can be accessed by users.

Surface Web

Surface web is that part of the World Wide Web which is easily available to the general public and accessible through standard web search engines. It is section of the internet that is being indexed by search engines.

The websites, webpages and information that user find using web search engine like Google, Yahoo, Bing, etc. only portray that user are exploring just the surface of the web. Search Engines use the crawling technique to index the webpages. Thus, the we generally access only surface web. Only 4% of the online content is available for the general public in the entire ocean of the web

Deep Web

Deep web is that content of World Wide Web which is not indexed by standard web search engines. The content of the deep web is behind HTTP forms, and it used for many general purposes such as web mail, online banking, and services for which users has to pay and which is protected by a paywall. Search engines won't provide access to the links that are deep inside the website even if the search is specific. Content of the deep web can be accessed by a direct URL or IP address using normal browsers, but it will require some type of authentication like password or other security measures to get access to its database.

Dark Web

The Dark Web is defined as a layer of information and pages that one can only get access to through so-called "overlay networks", which run on top of the normal internet. one need special software to access the Dark Web because a lot of it is encrypted, and most of the dark web pages are hosted anonymously.

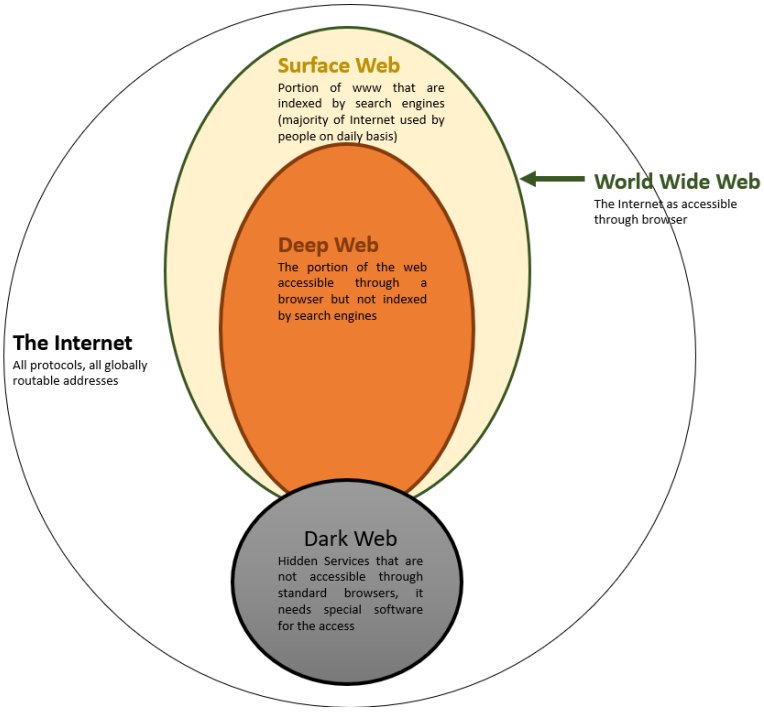


Figure 54 Layers of Internet

II. Different Darkwebs

I2P

The Invisible Internet Project (I2P) is a private network layer that is entirely encrypted. It safeguards both your activities and your location.

I2P shields the user from the server and the server from the user. I2P traffic is entirely internal to the network. I2P traffic does not directly interface with the Internet. It's a layer that sits on top of the web. Between you and your peers, encrypted unidirectional tunnels are used. Nobody can see where the traffic comes from, where it goes, or what it contains. I2P also has a high level of resilience to censorship and pattern recognition. Location blocking is also decreased because the network relies on peers to route data.

The I2P network is run entirely by volunteers. Peers share a portion of their resources, especially bandwidth, with other network members. This eliminates the need for centralised servers to run the network.

Freenet

Freenet is a piece of free software that allows you to share files anonymously, browse and publish "freesites" (web sites available exclusively on Freenet), and discuss on forums without fear of being censored. Freenet is decentralised to make it less vulnerable to attack, and it is highly difficult to identify when used in "darknet" mode, when users only connect with their friends.

Freenet node communications are encrypted and routed through other nodes, making it incredibly difficult to figure out who is seeking information and what it is about.

Users help the network by donating bandwidth and a piece of their hard drive (referred to as the "data store") for file storage. Files are retained or removed automatically according on their popularity, with the least popular being purged to create room for newer or more popular

content. Because files are encrypted, the user can't easily see what's in his datastore, and hence can't be held responsible for it. This distributed data storage underpins chat forums, websites, and search functionality.

TOR

Tor (The Onion Router) is an advancement of a program that was originally developed by the US Navy in the mid-1990s. It provides user greater anonymity online by encrypting internet traffic and passing it through a series of nodes.

When a user connects to Tor, his outgoing internet traffic is rerouted through a random series of at least three nodes (called relays) before routed its destination i.e., the website the user wants to visit. User's computer is connected to an entry node, and the traffic passes through is the exit node, after which it reaches its destination. Even incoming traffic is rerouted in the same way.

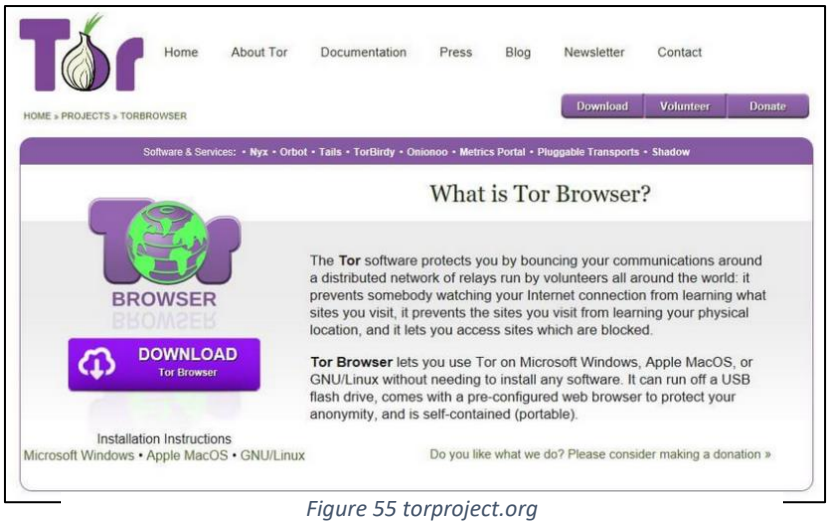


Figure 55 torproject.org

Along with the traffic passing through several nodes, the traffic is also encrypted multiple times. It might lose a level of encryption at each node, but it is never fully decrypted until it leaves the exit node for its destination.

Each node has an identifying IP address, which is encrypted. Only the IP address of final / exit node is visible to the destination website.

Currently Tor network is made up of about 7,000 relays (nodes) and 800 bridges. Bridges are similar to relays, but they are not recorded in the Tor directory. These are typically used by anyone who is unable to access the Tor network by regular means.

- **IP address privacy**

  On Tor network, user activity will never be traceable back to user IP address. Internet Service Provider (ISP) will not be able to view information about the contents of user traffic, including which website user is visiting. ISP would see only a Tor entry node, and the IP address of the Tor exit node.

- **Using Tor**

  The easiest way to use Tor is through the Tor browser. This is a Firefox-based application which can be downloaded and installed on computer. Its versions are

available for MacOS, Windows, and Linux. Tor browser enables user to access Clearnet and ". onion" sites through the browser.

If the use of the Tor browser is blocked, user may opt for a tor bridge. User first just needs to locate a bridge then configure it with the Tor browser.

- **Anonymity using Tor**

  Since all the traffic arriving at destination will appear to come from a Tor exit node, so will have the IP address of that node assigned to it. Because the traffic has passed through several additional nodes while encrypted, it can't be traced back to user.

  It must be noted that using the Tor browser only protects traffic going through that particular connection and would not anonymize other apps on the computer. Also, user's ISP can see that user is using Tor. For improved privacy, user can use a VPN alongside the Tor browser.

- **Tor and the Darknet**

  Clear net websites can be accessed using Tor, but it can also access darknet websites, specifically. onion sites. These are sites which only people using the Tor browser can access, and have. onion as part of their URL. They are also referred to as "Tor hidden services." These websites are not indexed by search engines and can be difficult to find if users don't know where to look. Tor protects the anonymity of the operators of. onion sites, so it would be difficult to find out who is hosting the site. Hence the combination of both operator and user anonymity makes the darknet ideal for criminal activity.

- **Why to use Tor?**

  It is always presumed that Tor is used for illegal activities or to access dark web but it is not true. Tor can simply be used by any privacy-conscious user for day-to-day browsing on Clearnet sites, to help maintain user anonymity and privacy while online. There are also some professions where anonymity is necessary for various reasons and using Tor include helps them to achieve it:

  - Journalists
  - Law enforcement officers
  - Activists
  - Whistle blowers
  - Business executives
  - Bloggers
  - Militaries
  - IT professionals

- **Legality of Tor**

  Tor is completely legal, even if it has been or is now restricted in some countries. Though ISPs have been reported to throttle the bandwidth of Tor users and have even contacted customers to tell them to stop using the Tor browser. Users might be questioned by ISPs regarding which websites they are connecting to through Tor.

Authorities themselves at times become suspicious of Tor users and conduct investigations into their activities on those grounds alone. However, there haven't been any reports of penalties or prosecutions relating to Tor use.

III. Illegitimate Activities on The Dark Web

The Dark Web allows its customers to anonymously reach to its websites utilizing applications like TOR. This way they can escape from monitoring and their identities remain concealed. Notwithstanding the way that online anonymity makes way for customer privacy, it in like manner opens approaches to a lot of criminal activities. Some of these are mentioned below:

1. **Counterfeit Currency:**
   Many counterfeit currency distributors are active on Dark Web, who sells fake currency with a guarantee of surpassing standard ultraviolet light checks successfully.
2. **Forged Documents:**
   Several sites on the Dark Web provides fake passports, immigration papers, driving licences and other identity documents for any country in the world. These services allow notorious people to acquire fake citizenship as per their needs. Other forged documents that are readily available include citizenship papers, fake IDs, college diplomas and even diplomatic identity cards.
3. **Drugs:**
   On the Dark Web, you may buy a variety of illicit substances of various types and quality. On Dark Web marketplaces, even illegal medications and pharmaceuticals like Ritalin and Xanax can be found. Silk Road is an example of a Dark Web marketplace which became famous for the wide range of drugs that were sold through it in huge amounts.
4. **Stolen Confidential Information:**
   This involves the purchase and sale of stolen credit card numbers, bank account numbers, and even personal data such as social security numbers. Apart from physical credit or debit cards, bank accounts can also be purchased at different prices in this Dark world.
5. **Hackers:**
   Hackers can easily buy sophisticated malwares and even get paid by interested parties to carry out any kind of online hacking attacks against specific governments, organizations or individuals.
6. **Arms and Ammunitions:**
   Illegal trade of explosives, weapons and firearms is also carried out openly. These services ensure that the specified products are delivered to the customer in special packaging that is easily scanned and security checked.
7. **Human Organ Trafficking:**
   Human organ trafficking is another business which has its roots deeply penetrated in the Dark Web. Organs such as kidneys, liver, heart, and eyeballs are routinely purchased in these underground markets.

8. **Terrorist Activities:**
   From secret communication and propaganda to recruitment and training for terrorist, everything that cannot be openly done on the Visible Web, is carried out through Dark Web.
9. **Child Pornography:**
   The Dark Web has also become famous destination for hosting child abuse videos as well as child pornography.

The list of illicit activities carried out using Dark Web is endless. Criminals thrive in this secret world because procedures to trace their footprints are complex and tracking them is much more difficult. The adoption of Bitcoin and other cryptocurrencies as a means of payment is a crucial element driving the proliferation of Dark Web-based crimes.

IV. Gathering information about TOR nodes/ relays:

- To ensure whether given IP address belongs to TOR network, the IP addresses can be searched on TOR directory if IP is present in list means it is part of TOR network. link for tor directory is:
  https://www.dan.me.uk/torlist/
- To verify if given IP address was used as Tor Relay Node on Specific Date:
  https://metrics.torproject.org/exonerator.html
- To gather information about TOR nodes:
  https://torstatus.rueckgr.at/
- To find repository of TOR Relays in India:
  https://metrics.torproject.org/rs.html

V. Crawling websites of Tor

The technique of indexing data on online sites using a software or automated script is known as web crawling. Web crawlers, spiders, spider bots, and crawler are all names for automated scripts or programmes that crawl the website. Web crawlers save pages to be processed by a search engine, which indexes the pages so that users may find information more quickly. A crawler's job is to figure out what's on each page. This allows users to quickly access any information on one or more pages.

On the dark web, website URLs are generally made up of a randomized string of letters and digits, followed by the ". onion" subdomain. Standard browsers such as Chrome and Safari are unable to resolve these websites, which need the use of the TOR browser.

Crawling Tor website will index links to websites, email addresses, crypto currency addresses present on the websites also it will gather information about server where website might be hosted.

Some tools are enlisted below which can be used to crawl ". onion" websites:

- OnionScan
- OnionOff
- Onion-nmap

- TorBot
- TorCrawl
- Onion Ingestor

VI. Evidences related to TOR in windows system

- RAMDUMP:

RAMDUMP is nothing but copy of the volatile memory of the system taken during live acquisition of the system. When it comes to investigation related to darkweb majority of the evidences can be found in RAM, hence taking RAMDUMP will help course of investigation in multiple ways.

Following evidences can be expected to find in RAMDUMP:

- URL of the websites visited by the user
- Emails addresses on which user communicated
- All recent activities along with recently composed mails, personally identifiable details, file names that are attached with mail etc.
- Crypto currency addresses
- Passwords of some accounts which user might have accessed recently

- TOR browser

When user install TOR browser it creates folder with name “Tor Browser” in the location which user has selected for installation. This folder contains multiple sub folders using which we can gather some evidences.

Bookmarks of the TOR browser can be extracted from “places.sqlite” file which is present in the folder” Profile.default”. Path of this folder is:

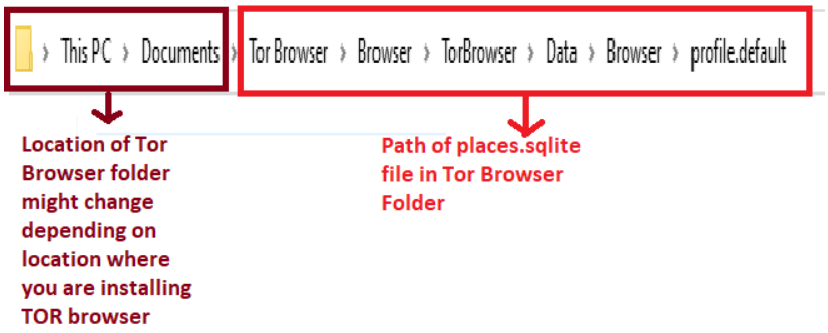


Figure 56 path for places.sqlite file

Date & Time of last use of TOR can be found in “state” file which is present in following location:

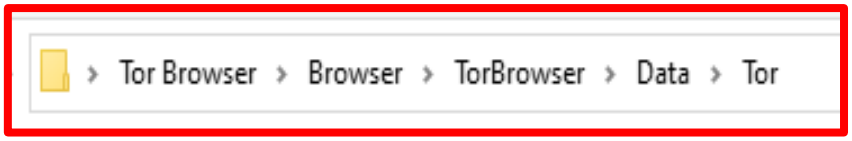


Figure 57 path to "state" file

State file should be opened using notepad to see the timestamp:

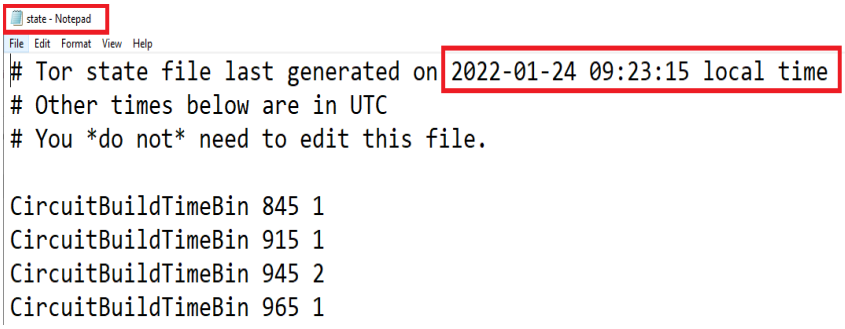


Figure 58 state file in notepad

If user has configured any restricted Entry or Exit node then that information can be gathered from “torrc” file present at following location:

Figure 59 PATH TO "TORRC" FILE

Open “torrc” file using notepad:

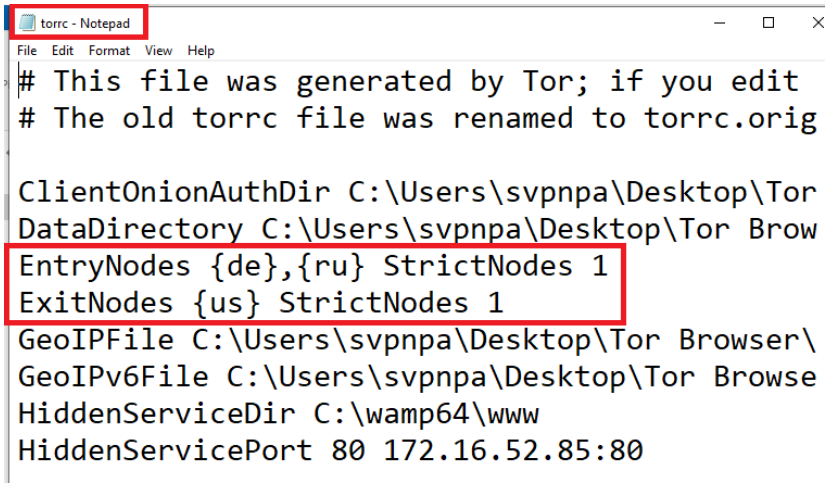


Figure 60 torrc file in notepad

VII. Taking archive of Tor Websites:

Tor websites are very unstable in nature they might go down at any point of time considering the security reasons. Hence in this situation if investigation related to some website is ongoing and that website goes down it might create hindrance for further investigation. To avoid this circumstances IO can take archive of the TOR website so that even if the website goes down in future, he can refer

to the offline copy of the website to continue with investigation. There are few tools which can be used for the same purpose.

- archive.today:

You can just open “archive.today” website on any browser. Paste the url of the “.onion” site of which you want to take archive. And add “.pet” extension after “.onion” and click on save.

Figure 61 archive.today

- Hunchly

It is the paid tool which can keep record of all your investigation activities on darkweb once you enable capture of the browser which you are using for investigation.

Dashboard of Hunchly tool:

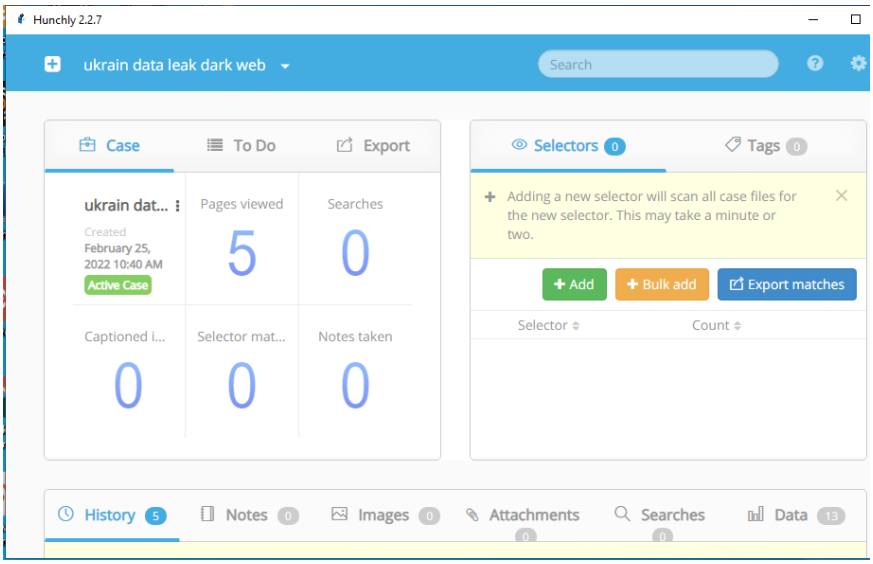


Figure 62 Hunchly

VIII. Configuring Entry/Exit Nodes:

To configure entry/exit nodes, first open ‘torrc’ file from the Tor Browser folder, Path for the same is as follows:

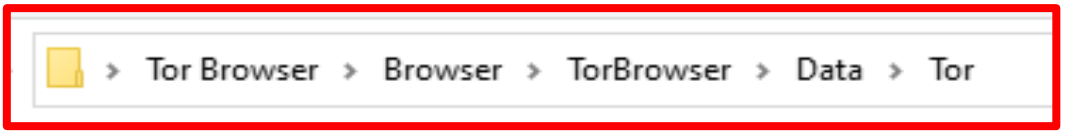


Figure 63 path to torrc file

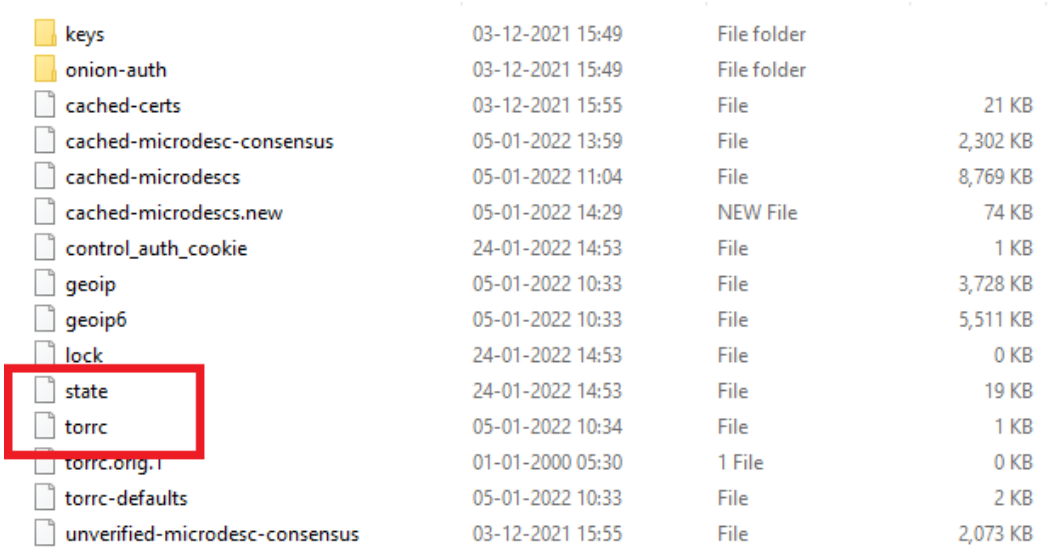


Figure 64 torrc file

Add the following lines in **torrc** file of Tor Browser:

- To use entry nodes only from specific country

  EntryNodes {in} StrictNodes 1

  EntryNodes {in},{au},{us} StrictNodes 1

- To use exit nodes only from specific country

  ExitNodes {in} StrictNodes 1

  ExitNodes {in},{au},{us} StrictNodes 1

- If you do NOT want to use nodes from specific country

  ExcludeEntryNodes {cn},{pk} StrictNodes 1

  ExcludeExitNodes {us},{cn},{pk} StrictNodes 1

  ExcludeNodes {cn},{pk} StrictNodes 1

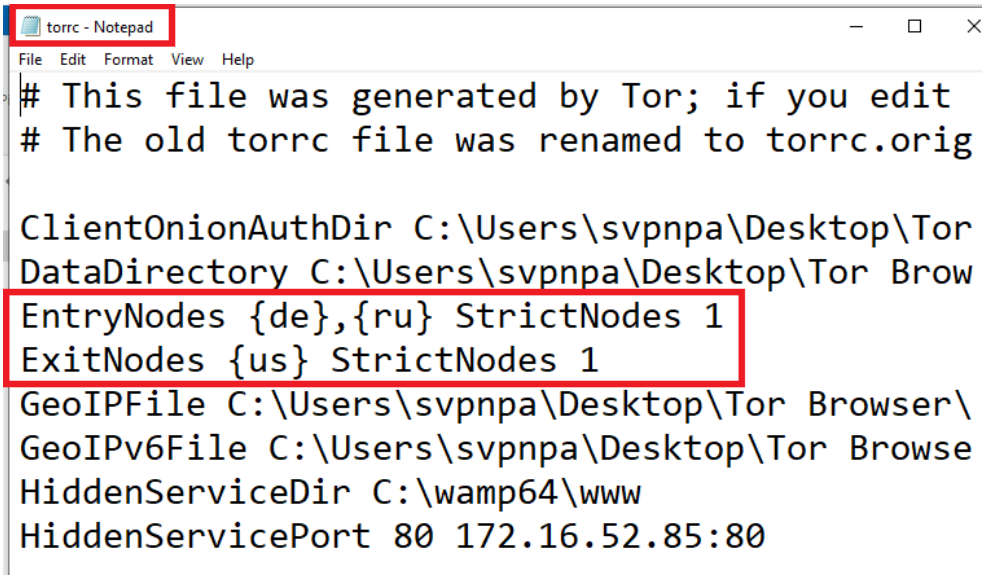


Figure 65 Tor node customization

2. Introduction to Blockchain

I. Concept of Blockchain Technology

Blockchains are tamper evident and tamper resistant digital ledgers implemented in a distributed fashion, not having any central authority. In simple terms, it stores Data, and records its movements in a distributed environment.

The concept of blockchain was first proposed by a group of researchers in 1991. The intent behind concept was time-stamping digital documents so that predating documents will not be possible thereafter. But the whole concept remained unused for a long period.

In 2009 Satoshi Nakamoto (anonymous name) in his white paper "Bitcoin: A Peer-to-Peer Electronic Cash System" used the concept of blockchain. Hence the first blockchain 'Bitcoin' came into existence.

II. Blockchain Architecture

a) Understanding Architecture of Blockchain

Blockchain is a distributed database or open vault that stores details of assets and its movements/ transactions across a peer to peer network. Each transaction is secured using cryptography and later all the transaction history is grouped and stored as blocks of data. Then the blocks are linked together with cryptography and secured from modification. The whole procedure creates an unforgeable and immutable record of the transactions that happened across the network. Additionally, this block of records is also stored to every participating computer in the network, so everyone has access to it, which is greatest advantage of using blockchain technology.

The data in blockchain is stored in the form of individual blocks, that's why it is called as Blockchain (i.e. a chain of blocks).

Each block in a blockchain will have the following fields.

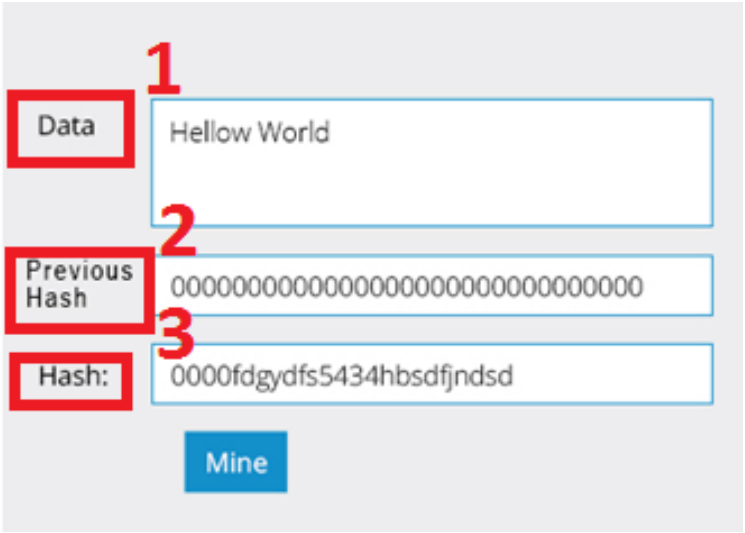


Figure 66 Fields in Blockchain

1) **Data:** Stores the data. As far as the user is concerned the Data field is the most important. The actual data (like transaction details, asset details etc.) are stored in this field.
2) **Previous Hash:** Stores the hash of the previous block (consider it as a link to the previous block). The blocks are connected through this value.
3) **Hash:** Hash value for the current block which can be used to refer this block

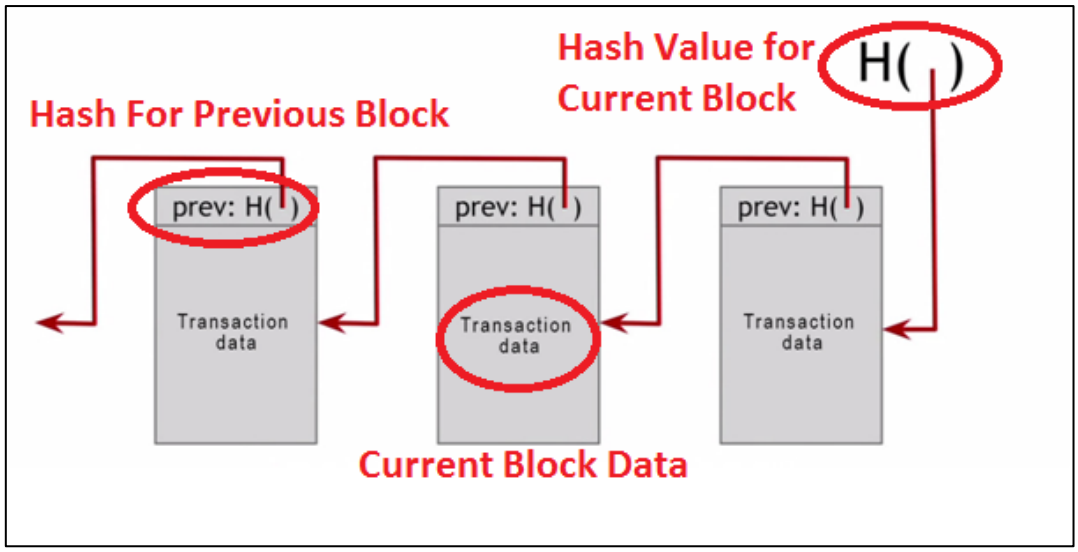


Figure 67 Blockchain Mechanism

b) Data Distribution in Blockchain

Blockchain follow the Peer to Peer model for distribution of data as oppose to widely adopted client server model. The peer to peer data distribution approach is the reason behind unrestricted nature of Blockchain; there is no central authority to control**.**

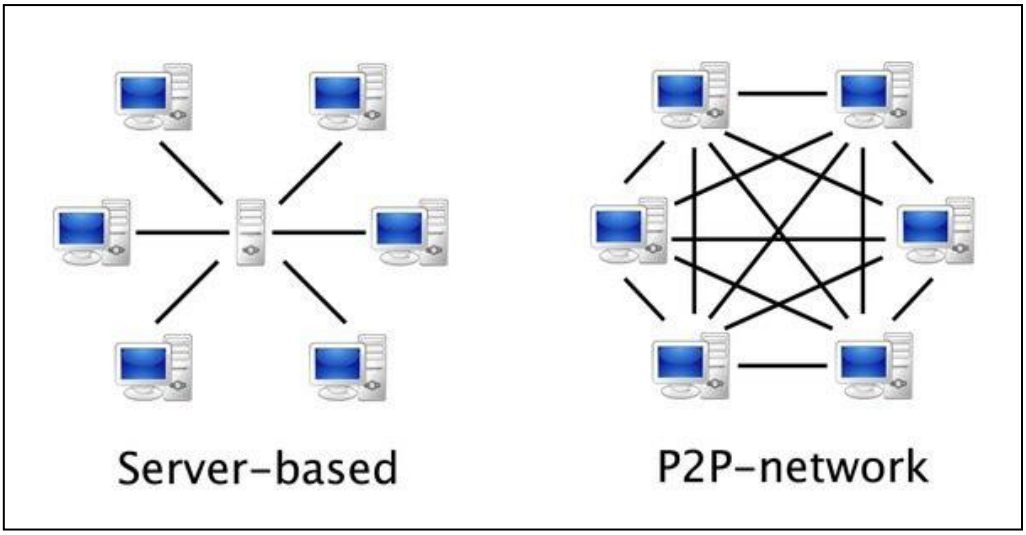


Figure 68 Server Based Network Vs Peer to Peer Network

Data is stored in all the participant nodes in the network. All the individual nodes have the copy of the entire 'Blocks' and a single change in a particular block is updated in all the nodes.

c) Block Validation

Unlike Client-Server model there is no central authority to validate database in the blockchain so it uses consensus mechanism of the blockchain network.

The asset and its transactions are stored as connected blocks in blockchain. Only the valid transactions are added to the blockchain. Blockchain validation is simply the process of finding the block hash. In a blockchain, all the blocks are added to the blockchain after validation only.

Whenever a transaction takes place in the blockchain it is stored to a block; sometimes one transaction per block and at times multiple transactions per block. It depends on the block size and the nature of the network. Before a transaction is stored to the block, it has to go through a validation process. The hash value for the block can be calculated using some algorithms (like SHA 256). The use of hash value is justified by its certain properties. The main thing is that the hash value will be collision-free i.e. no two blocks will have the same hash value. Since each block is represented using the hash value it will have similarity. The second property is that the hash is irreversible. This means the block data could not be retrieved from the hash value.

d) Consensus Mechanism

Consensus is a distributed computing concept that is used in blockchain in order to provide a means of agreeing to a single version of the truth by all peers on the blockchain network. A consensus mechanism is a set of steps that are taken by most or all nodes in a blockchain to agree on a proposed state value. Consensus mechanisms have most recently come into the limelight and gained considerable popularity with the advent of blockchain and Bitcoin.

There are various requirements that must be met to provide the desired results in a consensus mechanism. The following describes these requirements:

- **Agreement**: All honest nodes decide on the same value
- **Termination**: All honest nodes terminate execution of the consensus process and eventually reach a decision
- **Validity:** The value agreed upon by all honest nodes must be the same as the initial value proposed by at least one honest node
- **Fault tolerant:** The consensus algorithm should be able to run in the presence of faulty or malicious nodes (Byzantine nodes)

- **Integrity:** This is a requirement that no node can make the decision more than once in a single consensus cycle

Block validators are the nodes which participate in the process of block validation. Different blockchain protocols adopt different methodologies for selecting the validator from available pool of nodes. Some of the methods are described below. The following is not an exhaustive list, but it includes all notable algorithms.

- **PoW (Proof of Work)**

In PoW, the mining challenge is open to all. All the miners compete with each other to add the next block. A fixed reward is given to the miner who finds the solution first. In fact, the node with more computational power usually wins the race. Bitcoin and Litecoin use the PoW algorithm.

- **PoS (Proof of Stake)**

It is a common alternative of PoW. Here, the validators are chosen based on the fraction of coins they own in the system. The nodes with more number of coins have more chance to be selected than the node with lesser number of coins so that any malicious attempt by that user would outweigh the benefits of performing such an attack on the network. In PoS the reward is in the form of transaction fee, new coins are not created for paying the validators. Presently, Blackcoin, NXT and Peercoin blockchains uses the PoS algorithm. Ethereum is also planning to shift to this method by 2018.

- **Delegated Proof of Stake (DPoS)**

This is modification over standard PoS, whereby each node that has a stake in the system can delegate the validation of a transaction to other nodes by voting. It is used in the BitShares blockchain.

- **Proof of Activity**

PoA is a combination of PoS and PoW, though it is introduced to overcome some of the problems in PoS and PoW. In this method, the mining begins with PoW later the process is switched PoS. Presently, 'Decred' is the only coin that is using a variation of proof of activity.

- **Proof of Elapsed Time**

In this method, the network uses a lottery functions for implementing consensus. A lottery algorithm 'Trusted Execution Environment' (TEE) is used for finding the leaders from a set of nodes. So the validators are selected randomly from the pool. Hyperledger Sawtooth blockchain uses PoET method. .

- **Proof of Burn**

In this method, the aspiring validators increase their stake in the system by sending their coins to an irretrievable location (thus the name burn). The validators are selected randomly, but those who has more stake in the system has high probability to get selected. Over the time the earned stake decays and the nodes has to burn more currency to increase their stake. The only coin that uses proof of burn mechanism is slimcoin.

- **Proof of Deposit (PoD)**

In this mechanism, nodes that wish to participate in the network have to make a security deposit before they can mine and propose blocks. This mechanism is used in the Tendermint blockchain.

- **Proof of Importance (PoI)**

This idea is significant and different from PoS. PoI not only relies on how large a stake a user has in the system, but it also monitors the usage and movement of tokens by the user in order to establish a level of trust and importance. It is used in the NEM coin blockchain.

- **Federated consensus or federated Byzantine consensus**

This mechanism is used in the stellar consensus protocol. Nodes in this protocol retain a group of publicly-trusted peers and propagate only those transactions that have been validated by the majority of trusted nodes.

- **Reputation-based mechanisms**

In this mechanism, a leader is elected by the reputation it has built over time on the network. It is based on the votes of other members.

- **PBFT**

This mechanism achieves state machine replication, which provides tolerance against Byzantine nodes. Various other protocols including PBFT, PAXOS, RAFT, and **Federated Byzantine Agreement** (**FBA**) are also being used or have been proposed for use in many different implementations of distributed systems and blockchains.

- **Proof of Capacity (PoC)**

This scheme uses hard disk space as a resource to mine the blocks. This is different from PoW, where CPU resources are used. In in PoC, hard disk space is utilized for mining and as such is also known as *hard drive mining*. This concept was first introduced in the Burstcoin cryptocurrency.

- **Proof of Storage**

This scheme allows for the outsourcing of storage capacity. This scheme is based on the concept that a particular piece of data is probably stored by a node *which* serves as a means to participate in the consensus mechanism. Several variations of this scheme have been proposed, such as Proof of Replication, Proof of Data Possession, Proof of Space, and Proof of Space-Time.

At this stage we can't say which method is more efficient. Each method has its own advantages and disadvantages. Many other methods are also being introduced to attain maximum productivity on a blockchain.

III. Characteristics of Blockchain

- **Decentralization**

A blockchain is stored in a file that can be accessed and copied by any node on the network. This creates decentralization. Blockchain is Decentralized as well as an open ledger. Ledger is the record of the transactions done and because it is visible to everyone, therefore is called an open ledger. No individual or any organisation is incharge of the transactions. Each and every connection in the blockchain network has a same copy of the ledger.

- **Persistency**

A blockchain is a permanent record of transactions. Once a block is added, it cannot be altered. This creates trust in the transaction record. Data stored in blockchain is immutable and cannot be changed easily as explained above. Also the data is added to the block after it is approved by everyone in the network and thus allowing secure transactions. Those who validate the transactions and add them in block are called miners.

- **Anonymity**

In blockchain anonymity is achieved by not binding the key pairs to their owners' true identity. Bitcoin is designed to allow its users to send and receive payments with an acceptable level of privacy as well as any other form of money.

But the point should be noted thatc Bitcoin cannot be more anonymous than cash and it is not likely to prevent criminal investigations from being conducted. Additionally, Bitcoin is also designed to prevent a large range of financial crimes.

- **Auditability**

Since the blockchain is an open file, any party can access it and audit transactions. This creates provenance under which asset lifetimes can be tracked.

IV. Types of Blockchain

Based on the evolution of blockchain, it can be divided into multiple categories with some distinct though partially-overlapping attributes. These blockchain types can occur on any blockchain tier, as there is no direct relationship between those tiers and the various types of blockchain.

a) Distributed ledgers

A distributed ledger is a broad term used for shared databases; therefore, all blockchains are technically distributed ledgers (i.e. shared databases). But all distributed ledgers are not necessarily a blockchain.

Difference between a distributed ledger and blockchain is that a distributed ledger does not necessarily consist of blocks of transactions to keep the ledger growing, whereas a blockchain is a special type of distributed ledger that is comprised of blocks of transactions.

For example, R3's Corda is a distributed ledger that does not use blocks of transactions, which is developed to record and manage agreements and is especially focused on financial services industry. On the other hand, more popular blockchains like Bitcoin and Ethereum use blocks to update the shared database.

As the name suggests, a distributed ledger is distributed among its participants and spread across multiple sites or organizations. This type of ledger can be either private or public. The fundamental idea here is that, unlike many other blockchains, the records are stored contiguously instead of being sorted into blocks. This concept is used in Ripple which is a blockchain and cryptocurrency based global payment network.

b) Distributed Ledger Technology

Nowadays the terms **distributed ledger** or **Distributed Ledger Technology (DLT)** are commonly used to describe blockchain in **finance industry**. DLT is now a very thriving area of research in the financial sector. From a financial sector point of view, DLTs are permissioned blockchains that are shared and used between known participants. DLTs usually serve as a shared database, with all participants known and verified. They do not have a cryptocurrency or do not require mining to secure the ledger.

c) Public Blockchain

Public blockchains are not owned by anyone. They are open to the public, and anyone can participate as a node in the decision-making process. Users may or may not be rewarded for their participation.

All users of these permission less ledgers maintain a copy of the ledger on their local nodes and use a distributed consensus mechanism to decide the eventual state of the ledger. Bitcoin and Ethereum are both considered public blockchains.

d) Private Blockchain

Private blockchains are meant for private use. That is, they are open only to a consortium or group of individuals or organizations who have decided to share the ledger among them. There are various blockchains available in this category, such as HydraChain and Quorum. Optionally, both of these blockchains can also run in public mode if required, but their primary purpose is to provide a private blockchain.

- **Semiprivate Blockchains**

Semiprivate blockchains is a very new concept which just exist theoretically and no real world POCs have yet been developed. Here part of the blockchain is private and part of it is public. With a semi-private blockchain, the private part is controlled by a group of individuals, while the public part is open for participation by anyone.

This hybrid model can be used in scenarios where the private part of the blockchain remains internal and shared among known participants only, while the public part of the blockchain can still be used by anyone, optionally allowing mining to secure the blockchain. This way, the blockchain as a whole can be secured using PoW, thus providing consistency and validity for both the private and public parts. This type of blockchain can also be called a semi-decentralized model, where it is controlled by a single entity but still allows for multiple users to join the network by following appropriate procedures.

- **Sidechains**

Sidechains are more precisely known as pegged sidechains, where coins can be moved from one blockchain to another and moved back again. It can be used for the creation of new altcoins (alternative cryptocurrencies) where coins are burnt as a proof of an adequate stake. Burnt or burning the coins in this context suggest that the coins are sent to an address which is unspendable and this process makes the burnt coins irrecoverable. This mechanism is used to bootstrap a new currency or introduce scarcity which results in increased value of the coin. This mechanism is also called Proof of Burn (PoB) and is used as an alternative method for distributed consensus to PoW and Proof of Stake (PoS).

Above example for burning coins applies to oneway pegged sidechain. The second type is called a two-way pegged sidechain, which allows the movement of coins from the main chain to the sidechain and back to the main chain when required. This process enables the building of smart contracts for the Bitcoin network. Rootstock is one of the leading examples of a sidechain, which enables smart contract development for Bitcoin using this paradigm. It works by allowing a two-way peg for the Bitcoin blockchain, and this result in much faster throughput.

- **Permissioned ledger**

A permissioned ledger is a blockchain where participants of the network are already known and trusted. Permissioned ledgers do not use a distributed consensus mechanism; instead, it uses an agreement protocol to maintain a shared version of the truth about the state of the records on the blockchain. For verification of transactions on the chain, all verifiers are already preselected by a central authority and typically there is no need for a mining mechanism.

Permissioned blockchain is not private, rather it is public blockchain but with regulated access control. For example, Bitcoin can become a permissioned ledger if an access control layer is introduced on top of it that verifies the identity of a user and then allows access to the blockchain.

e) Shared Ledger

Any application or database that is shared by the public or a consortium is a shared ledger. In general, all blockchains fall into the category of a shared ledger.

f) Fully Private and Proprietary Blockchains

There is no mainstream application of these types of blockchains, as they deviate from the core concept of decentralization in blockchain technology. This type of blockchains can be used in some specific private settings within an organization, where there is need to share data and provide some level of guarantee of the authenticity of the data.

An example of this type of blockchain might be to allow for collaboration and the sharing data between various governments departments. In that case, no complex consensus mechanism is required, apart from simple state machine replication and an agreement protocol with known central validators.

g) Tokenized Blockchains

These are the blockchains that generate cryptocurrency as a result of a consensus process using mining or initial distribution. Bitcoin and Ethereum are prime examples of this type of blockchain.

h) Tokenless Blockchains

These blockchains are designed in such a way that they do not have the basic unit for the transfer of value. However, they are still valuable in situations where there is no need to transfer value between nodes and only the sharing of data among various trusted parties is required. This is similar to full private blockchains, the only difference being that use of tokens is not required. This can also be thought of as a shared distributed ledger used for storing data. It does have its benefits when it comes to immutability, security, and consensus driven updates but is not used for common blockchain application of value transfer or cryptocurrency.

V. Hashing

An important feature of blockchain technology is the use of cryptographic hash functions for many operations.

Hashing is a method of applying a cryptographic hash function to data, which calculates a relatively unique fixed size output called a *message digest* or *digest* for an input of nearly any size (e.g., a file, text, or image). Even the smallest change to the input (e.g., changing a single bit) will result in a completely different output digest.

- **Cryptographic hash functions have these important security properties**:

Preimage resistant: Hash functions are one-way; it is not feasible to compute the correct input value for any given output value (e.g., given a digest, find *x* such that hash(*x*) = digest).

Second preimage resistant: This means that it is not possible to find an input that will result into a specific output. Cryptographic hash functions are designed so that given a specific input, it is computationally infeasible to find a second input which produces the same output (e.g., given *x*, find *y* such that hash(*x*) = hash(*y*)). The only approach available is to exhaustively search the input space, but this is computationally infeasible to do with any chance of success.

Collision resistant: This means that it is infeasible to find two inputs that hash to the same output. (e.g., find an *x* and *y* which hash(*x*) = hash(*y*)).

A more popular cryptographic hash function used in many blockchain implementations is the **Secure Hash Algorithm (SHA)** with an output size of 256 bits (SHA-256). SHA-256 has an output of 32 bytes (1 byte = 8 bits, 32 bytes = 256 bits), displayed as a 64-character hexadecimal string (see Table 1 below). This means that there are $2^{256} \approx 10^{77}$, or 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936 possible digest values.

Input Text	SHA-256 Digest Value
1	0x6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
2	0xd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35
Hello, World!	0xdffd6021bb2bd5b0af676290809ec3a53191dd81c7f70a4b28688a362182986f

Table 1 SHA-256

Since there are an infinite number of possible input values and a finite number of possible output digest values, it is possible but highly unlikely to have a collision where hash(*x*) = hash(*y*) (i.e., the hash of two different inputs produces the same digest). SHA-256 is said to be collision resistant, since to find a collision in SHA-256, one would have to execute the algorithm, on average, about 2^{128} times (which is 340 undecillions, or more precisely 340,282,366,920,938,463,463,374,607,431,768,211,456; roughly 3.402 x 10^{38}).

To put this into perspective, the hash rate (hashes per second) of the entire Bitcoin network in 2015 was 300 quadrillion hashes per second (300,000,000,000,000,000/s). At this rate, it would

take the entire Bitcoin network roughly 35,942,991,748,521 (roughly 3.6×10^{13}) years to manufacture a collision (note that the universe is estimated to be 1.37×10^{10} years old). Even if any such input *x* and *y* that produce the same digest, it would be also very unlikely for both inputs to be valid in the context of the blockchain network (i.e., *x* and *y* are both valid transactions).

Within a blockchain network, cryptographic hash functions are used for many tasks, such as:

- Address derivation
- Creating unique identifiers
- Securing the block data – a publishing node will hash the block data, creating a digest that will be stored within the block header.
- Securing the block header – a publishing node will hash the block header. If the blockchain network utilizes a proof of work consensus model, the publishing node will need to hash the block header with different values until the puzzle requirements have been fulfilled. The current block header's hash digest will be included within the next block's header, where it will secure the current block header data.

Because the block header includes a hash representation of the block data, the block data itself is also secured when the block header digest is stored in the next block. There are many families of cryptographic hash functions utilized in blockchain technology (SHA-256 is not the only one).

3. Blockchain Technology Concepts

I. Distributed Ledger

A ledger is a collection of transactions. Throughout history, pen and paper ledgers have been used to keep record of the exchange of goods and services. In modern times, ledgers are stored digitally, often in databases owned and operated by a centralized trusted third party on behalf of a community of users. These ledgers with centralized ownership can be implemented in a centralized or distributed fashion (i.e., just one server or a coordinating cluster of servers).

There is increasing interest in exploring having distributed ownership of the ledger. Blockchain technology enables such an approach using both distributed ownership as well as a distributed physical architecture. The distributed physical architecture of blockchain networks often involve a much larger set of computers than is typical for centrally managed distributed physical architecture. The growing interest in distributed ownership of ledgers is because of trust, security, and reliability concerns related to ledgers with centralized ownership.

- **Centrally owned ledgers have a chance of getting lost or destroyed; a user must trust that the owner does have proper back up of the system.**

A blockchain network is distributed, because of which multiple copies are created automatically of all updates and same are synced to the ledger data between peers. A key benefit to blockchain technology is that every user can maintain own copy of the ledger. Whenever new full nodes join the blockchain network, they reach out to discover other full nodes and request a full copy of the blockchain network's ledger, making loss or destruction of the ledger difficult. There are some private transactions also that facilitate the delivery of information only to those nodes participating in a transaction and not the entire network.

- **Centrally owned ledgers are in general developed on a homogeneous network, where all software, hardware and network infrastructure may be the same. Because of which it creates possibility of single point of failure, hence the overall system resiliency may be reduced since an attack on one part of the network will work on everywhere.**

A blockchain network is a heterogeneous or distributed, where the software, hardware and network infrastructure are all different. Because of the many differences between nodes on the blockchain network, an attack on one node is not guaranteed to work on other nodes.

- **Centrally owned ledgers may be located entirely in specific geographic locations (e.g., all in one country). If network outages were to occur in that location, the ledger and services which depend on it may not be available.**

A blockchain network can be comprised of geographically diverse nodes which may be found around the world. Because of this, and the blockchain network working in a peer-to-peer fashion, it is resilient to the loss of any node, or even an entire region of nodes.

- **The transactions on a centrally owned ledger are not made transparently and may not be valid; a user must trust that the owner is validating each received transaction.**

A blockchain network must check that all transactions are valid; if a malicious node was transmitting invalid transactions, others would detect and ignore them, preventing the invalid transactions from propagating throughout the blockchain network.

- **The transaction list on a centrally owned ledger may not be complete; a user must trust that the owner is including all valid transactions that have been received.**

A blockchain network holds all accepted transactions within its distributed ledger. To build a new block, a reference must be made to a previous block – therefore building on top of it. If a publishing node did not include a reference to the latest block, other nodes would reject it.

- **The transaction data on a centrally owned ledger may have been altered; a user must trust that the owner is not altering past transactions.**

A blockchain network utilizes cryptographic mechanisms such as digital signatures and cryptographic hash functions to provide tamper evident and tamper resistant ledgers.

- **The centrally owned system may be insecure; a user must trust that the associated computer systems and networks are receiving critical security patches and have implemented best practices for security. The system may be breached and have had personal information stolen because of insecurities.**

A blockchain network, due to the distributed nature, provides no centralized point of attack. Generally, information on a blockchain network is publicly viewable, and offers nothing to steal. To attack blockchain network users, an attacker would need to individually target them. Targeting the blockchain itself would be met with the resistance of the honest nodes present in the system. If an individual node was not patched, it would only affect that node – not the system overall.

II. Advantages and Disadvantage of Blockchain

a) Advantages

- **Transaction Speed**
  Cryptocurrencies offer exceptionally high speed transaction which is far more superior to the current banking transaction speed. Bitcoin takes a maximum of 10 minutes for validating a transaction and it is about 10 seconds in Ethereum.
- **Anonymity**
  Cryptocurrency transactions are completely anonymous and it is impossible to identify who had done this transaction or to whom this transaction is made. The members will use only the network address of the sender and receiver. No identity of those participants will be published in the shared ledger.
- **No restriction on payments**
  A limitless transaction is the most important advantage of cryptocurrency. The client can send the money at anytime from anyplace to all over the world. That implies no time boundaries like bank holidays.

- **Less /No transaction fees**
  The cryptocurrency transactions in general do not include any processing fee, or the fee will be much less than present financial transaction charges. In bitcoin, anybody can do transactions without paying any transaction fees. The user also has the option to offer transaction fees for speeding up their transaction. That is if a person is providing a transaction fee, more miners will come to validate the transaction; hence the transaction gets validated fast.
- **Immutable transactions**
  Cryptocurrencies are one of the most secure currency systems available today. It has the 'immutable' property; i.e. If one transaction had occurred in the blockchain based cryptocurrency, it is irreversible. So the chances of fraudulent transactions are nearly impossible.
- **Government can't De-monetize**
  Cryptocurrencies work as a decentralized system and its exchange rate is fixed dynamically according to the demand-supply factors. Government regulation can't stop such independent cryptocurrencies. The only thing that a government can do is restrict the conversion of it to normal currency. However, they can't stop the transactions in cryptocurrencies.
- **Secure Payment information**
  Cryptocurrency transactions are completely anonymous and don't use any identity of the users. They will only use the wallet address of the sender and receiver, all other information is securely hashed and no one can retrieve it back. During transaction among two entities, none of the personal information will be shared with them. Only the particular amount of bitcoin will be transferred from one account to another account.
- **No Inflation**
  Since cryptocurrencies generally have a fixed number of currencies in their exchequer. E.g., bitcoin have 21 million in its exchequer. Once the entire thing has mined there won't be any more new bitcoins. So there is no chance for inflation.

b) <u>Disadvantages</u>

- **Less Acceptance**
  Despite of increasing popularity and demand for 'cryptocurrency' day by day, yet many governments have not given any official approval for 'cryptocurrency' transaction. Beacuase of which, its usage is still limited to some specific domains only. Hence cryptocurrency is still away from common people.
- **Inconsistent rate**
  Although cryptocurrency follow a strict demand supply rule to define the exchange rate, at present market trends indicate an uncommon surge in the exchange rate of cryptocurrencies, especially that of Bitcoin.
- **Government Ban**
  As discussed above government can't control cryptocurrencies, but they can ban it and illegalize its transaction.
- **Key recovery is impossible**
  There is no central authority to regulate cryptocurrencies, hence every individual is responsible for keeping their account safe. If anyone loses the wallet key, no one can help them to recover the key.
- **Supports Money Laundering/Black Market**
  Cryptocurrency are popular among black market and money launders because of its anonymity. Since the user identity is not revealed anywhere, its misuses are reported

many times. Famous two are the “silk road” website which provides illegal drugs and other illegal items payable by bitcoin is recent ‘Wannacry’ cyber-attack.

III. Blockchain use cases

It was Bitcoin which started advent of blockchain in real world; following the trend, many other cryptocurrencies also came into the market. However, soon the actual potential of blockchain technology was realized and possibilities of use of blockchain in many other unprecedented domains explored. Healthcare Industry, Enterprise software development, financial domains like Banking, Insurance and so on; today the blockchain is drastically changing existing technology frameworks of almost all domains.

a) In IoT

Internet of Things is defined as a network of intelligent physical objects (any object such as cars, fridges, industrial sensors, watches and so on) that are capable of connecting to the internet, sensing real-world events or environments, reacting to those events, collecting relevant data, and communicating it over the internet. The exciting concepts, such as wearables, smart homes, smart grids, smart connected cars, and smart cities all are based on basic concept of an IoT device. These devices perform basic functions that are **sensing**, **reacting**, **collecting**, and **communicating**. All these functions are performed by using various components on the IoT device.

A typical IoT can consist of many physical objects connecting with each other and to a centralized cloud server. This is shown in the following diagram:

Blockchain can prove very useful for IoT devices:

- To build trust, reduce costs, and accelerate transactions.
- Decentralization, which is at the core of blockchain technology, can eliminate single points of failure in an IoT network. For example, if a central server is not able to cope with the amount of data that billions of IoT devices are producing at high frequency.
- The peer-to-peer communication model provided by blockchain can help to reduce costs because it eliminates need to build high-cost centralized data centers and implementation of complex public key infrastructure for security. Devices can communicate with each other directly or via routers.
- It is estimated that, by 2020 there will be roughly 22 billion devices connected to the internet. With this explosion of billions of devices connecting to the internet, it is hard to imagine that centralized infrastructures will be able to cope with the high demands of bandwidth, services, and availability without incurring excessive expenditure. Blockchain-based IoT will be able to solve scalability, privacy, and reliability issues in the current IoT model.
- Blockchain enables things to communicate and transact with each other directly and with the availability of smart contracts, negotiation, and financial transactions can also occur directly between the devices instead of requiring an intermediary, authority, or human intervention. For example, if a room in a hotel is vacant, it can rent itself out, negotiate the rent, and can open the door lock for a human who has paid the right amount of funds. Another example could be that if a washing machine runs out of detergent, it could order it online after finding the best price and value based on the logic programmed in its smart contract.
- Blockchain-based IoT can also thwart denial of service attacks where hackers can target a centralized server or data center more efficiently, but with blockchain's distributed and decentralized nature, such attacks are no longer possible.

The five-layer IoT model can be adapted to a blockchain-based model by adding a blockchain layer on top of the network layer. This layer will run smart contracts, and provide security,

privacy, integrity, autonomy, scalability, and decentralization services to the IoT ecosystem. The management layer, in this case, can consist of only software related to analytics and processing, and security and control can be moved to the blockchain layer. This model can be visualized in the following diagram:

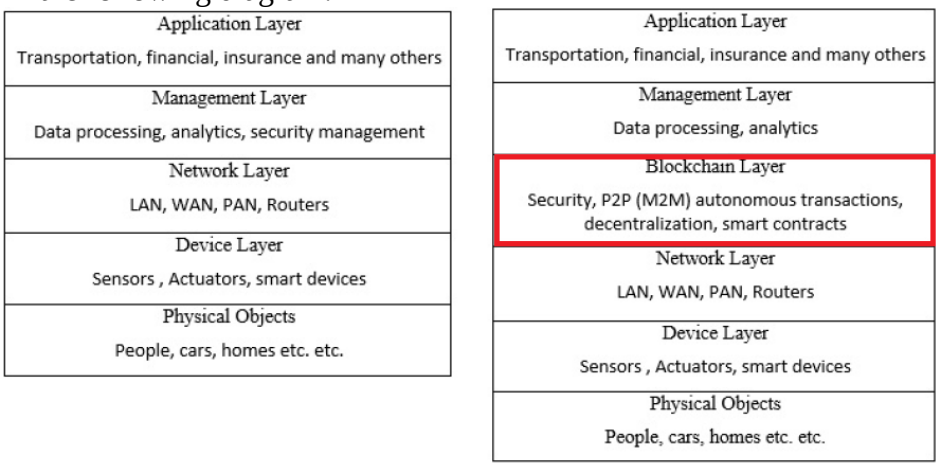


Figure 69 Blockchain Based IoT Model

b) In Financial Services

Blockchain in finance is the hottest topic in the industry currently, and major banks and financial organizations are researching to find ways to adapt blockchain technology primarily due to its highly-desired potential to save cost.

Insurance:

- In the insurance industry, blockchain technology can help to stop fraudulent claims, increase the speed of claim processing, and enable transparency.
- A shared ledger between all insurers can provide a quick and efficient mechanism for handling intercompany claims.
- Blockchain can reduce the cost and effort required to a great extent for claims. Claims can be automatically verified and paid via smart contracts and the associated identity of the insurance policyholder.
- Several start-ups such as Dynamis have proposed smart contract-based peer-to-peer insurance platforms that run on Ethereum blockchain. This is initially proposed to be used for unemployment insurance and does not require underwriters in the model.

Financial crime prevention:

- In the process of KYC, currently, each institution maintains their own copy of customer data and performs verification via centralized data providers. This can be a time-consuming process and can result in delays in onboarding a new client.
- Blockchain can provide a solution to this problem by securely sharing a distributed ledger between all financial institutions that contain verified and true identities of customers. This distributed ledger can only be updated by consensus between the participants thus providing transparency and auditability. This can not only reduce costs but also enable meeting regulatory and compliance requirements in a better and consistent manner.
- In the process of Anit Money Laudering, due to the immutable, shared, and transparent nature of blockchain, regulators, can easily be granted access to a private blockchain where they can fetch data for relevant regulatory reporting. This will also result in reducing complexity and costs related to the current regulatory reporting paradigm

where data is fetched from various legacy and disparate systems and aggregated and formatted together for reporting purposes. Blockchain can provide a single shared view of all financial transactions in the system that are cryptographically secure, authentic, and auditable, thus reducing the costs and complexity associated with the currently employed regulatory reporting methods.

c) In Governance

Blockchains have potential to support to take current e-government model to next level. E-government is a paradigm where information and communication technology are used to deliver public services to citizens. Blockchain attributes like transparency, auditability, and integrity can be helpful for managing and delivering public services including but not limited to identity cards, driving licenses, secure data sharing among various government departments and contract management.

- Automated Border Control System:

One key issue with current border control systems is data sharing where the systems are controlled by a single entity and data is not readily shared among law enforcement agencies. This lack of the ability to share data makes it challenging to track suspected travel documents or individuals. Another issue is related to the immediate implementation of blacklisting of a travel document, for example, when there is an immediate need to track and control suspected travel documents. Currently, there is no mechanism available to blacklist or revoke a suspected passport immediately and broadcast it to the border control ports worldwide.

Blockchain can provide a solution for this problem by maintaining a blacklist in a smart contract which can be updated as required and any changes will be immediately visible to all agencies and border control points thus enabling immediate control over the movement of a suspected travel document. Blockchain based systems will provide cryptographically guaranteed immutability which helps with auditing and discourages any fraudulent activity.

- **Voting:**

Voting is a key function of government which allows citizens to participate in the democratic election process. While voting has evolved over the years as a secure process, it still has limitations that need to be addressed to achieve a desired level of maturity.

The limitations in current voting systems revolve around fraud, weaknesses in operational processes, and inadequate transparency. Over the years, secure voting mechanisms (machines) are built which make use of specialized voting machines that promises security and privacy, but they still have vulnerabilities that could be exploited to sabotage the voting process which may result in mistrust in the government by the public.

Blockchain-based voting systems can resolve these issues:

- Blockchain can introduce end-to-end security and transparency in the voting process. Security is provided in the form of integrity and authenticity of votes by using public key cryptography which comes as standard in a blockchain.
- Immutability guaranteed by blockchain ensures that votes cast once cannot be cast again. This can be achieved through a combination of biometric features and a smart contract maintaining a list of votes already cast. For example, a smart contract can maintain a list of already casted votes with the biometric ID (for example a fingerprint) and can use that to detect and prevent double casting.
- **Zero-Knowledge Proofs** (**ZKPs**) can be used on the blockchain to protect voters' privacy on the blockchain.

- **Citizen Identification (ID Cards):**

Though there are robust mechanisms already existing in almost all countries regarding issuing and securing ID cards. However using blockchain technology several improvements can be made to this technology. Digital identity is not only limited to just government-issued ID cards; it is a concept that applies to online social networks and forums too.

Benefits of combining blockchain technology with digital identities:

- A blockchain-based online digital identity allows control over personal information sharing.
- Users can see who used their data and for what purpose and can control access to it. This is not possible with the current infrastructures which are centrally controlled.
- The key benefit is that a single identity issued by the government can be used easily and in a transparent manner for multiple services via a single government blockchain. E.g., the blockchain can serve as a platform where a government is providing various services such as pensions, taxation, or benefits and a single ID is being used for accessing all these services. Blockchain, in this case, provides a permanent record of every change and transaction made by a digital ID, thus ensuring integrity and transparency of the system.
- Citizens can notarize birth certificates, marriages, deeds, and many other documents on the blockchain tied with their digital ID as a proof of existence.

d) <u>In Healthcare</u>

The health industry can also get advances by adapting blockchain technology.

- Blockchain provides an immutable, auditable, and transparent system. Also, blockchain provides a cost-effective, simpler infrastructure as compared to traditional complex PKI networks.
- Major issues of healthcare system are privacy compromise; data breaches, high costs, and frauds which can arise from lack of interoperability, overly complex processes, transparency, auditability. Another burning issue is counterfeit medicines; especially in developing countries.
- With the adaptability of blockchain in the health sector, several benefits can be realized:
- Cost savings, increased trust, faster processing of claims, high availability, and no operational errors due to complexity in the operational procedures.
- Preventing distribution of counterfeit medicines.

e) <u>In Policing</u>

As the nature of crime in the world has expanded and evolved, so has the burden on law enforcement officers, who must now consider violent drug cartels, terrorism, cybercrime, and other relatively new threats under their jurisdiction.

By using blockchain technology to monitor, flag, and analyze transactions that may be directly connected to violent criminality, the risk of surveillance will become more affordable and less dangerous. There are also several administrative and bookkeeping roles that the technology can play — from preserving the chain of custody to interagency data sharing and more — that will free up funds to augment the more life-preserving resources that law enforcement officers require.

Securing the Chain of Custody for Evidence:

Though it's not yet possible to physically store a glove on the blockchain, it *is* possible to document who has handled any evidence, in addition to important information regarding

storage methods and other critical information. When it comes to evidence, tamper-proofing records of origination and chain of custody is imperative. Currently, the legacy evidence locker sign-in sheet means of storing evidence is not tamper-proof. Greater observation over where, when, and how evidence is stored and handled would put the onus on its handlers to do so with care — or face the wrath of accountability that comes with potentially blowing a case.

- Standardizing Distributed Crime Reports:

If an interoperable system exist by which victims could log onto a platform to see the progress of their case, whether it is the collection of security camera footage or important case notes, they could be better assured that the taxes they pay to support law enforcement yield results on a personal level.

- Tracing Criminality More Efficiently:

The immutable record of the blockchain can provide a historical cache of records and documents by which law enforcement can analyze potential criminal activity. Instead of having to expend resources engaging in issues of whether cell records, internet history, transactions, and more are admissible in an investigation, the blockchain's public nature ensures that law enforcement has access to those who would use the technology to commit crimes.

- Secure, Interoperable, Interagency Data Sharing:

Blockchain as not being controlled by any one party, being tamper-proof and interoperable, and allowing for sharing on a private or public network can be used to create a central data repository to coordinate the exchange of information on a national level for the benefit of all law enforcement agencies and departments

- Decentralizing Emergency Alert/Response Infrastructure:

Blockchain can be thought of as a means to create a unified platform for emergency responses across emergency response agencies and, on a broader spectrum, neighboring counties in case of widespread malfunctions. Whether in a singular or widespread emergency event, better coordination among first responders and their dispatchers is well worth looking further into.

4. Crypto Currencies

I. Basic understanding of Crypto Currency

The idea of 'cryptocurrencies' started its advent from 1998 itself. The first known attempt for creating a digital cryptocurrency was B-Money and Bit Gold, but both never came into reality. Cryptocurrencies are the digital or virtual currencies working on the cryptographic principles. It doesn't have any physical existence or they are not tangible. They only exist as a set of programming codes. Yet they are capable of providing high security and usability than many existing currencies.

Cryptocurrency works on blockchain technology; we have already seen how blockchain works. In the case of cryptocurrency, the ledger keeps the track of cryptocurrency that is generated and transacted across the network. Every individual in a particular blockchain will have a unique account Id/address. The cryptocurrency is always associated with this account (Currency is Debited and Credited to this account).

People can manage their account through the application called wallets. Through the wallets, anyone can make the transaction to anyone on the network (both the sender and receiver must have an account). The transactions are verified by nodes and added to the blockchain ledger. So, the immutable and encrypted ledger of blockchain is the backbone of cryptocurrency.

All features of blockchain are also applicable to cryptocurrency; the encryption mechanism, peer to peer network, and no central authority/central server to control. Each cryptocurrency will work on a blockchain protocol. One of the most famous cryptocurrencies is bitcoin which works on the bitcoin blockchain. And ether is another fast-growing cryptocurrency which runs on Ethereum protocol. While comparing with the traditional currencies, the cryptocurrencies

provide highly anonymous nature for participants. The only visible identity of a user will be his account ID, rest everything will be encrypted. The participants will not have any idea about the real identity of a user.

II. Evolution of Cryptocurrency

Digital currencies have always been an active area of research for many decades. Early proposals to create digital cash go as far back as the early 1980s.

- In **1982**, **David Chaum**, a computer scientist, and cryptographer proposed a scheme that used blind signatures to build untraceable digital currency. This research was published in a research paper, Blind **Signatures for Untraceable Payments**.

In this scheme, a bank would issue digital money by signing a blind and random serial number presented to it by the user. The user could then use the digital token signed by the bank as currency. The limitation of this scheme was that the bank had to keep track of all used serial numbers. This was a central system by design and required to be trusted by the users.

- Later on, in **1988**, **David Chaum and others** proposed a refined version named **e-cash** that not only used a blinded signature, but also some private identification data to craft a message that was then sent to the bank.

This scheme allowed the detection of double spending but did not prevent it. If the same token was used at two different locations, then the identity of the double spender would be revealed. e-cash could only represent a fixed amount of money.

- **Adam Back,** a cryptographer and now CEO of Blockstream, who is involved in blockchain development, introduced **hashcash** in **1997**. It was originally proposed to control email spam. The idea behind hashcash was to solve a computational puzzle that was easy to verify but comparatively difficult to compute. The idea was that for a single user and a single email, the extra computational effort was negligible, but someone sending a large number of spam emails would be discouraged as the time and resources required to run the spam campaign would increase substantially.
- In **1998**, **B-money** was proposed by **Wei Dai,** a computer engineer who used to work for Microsoft, introduced the idea of using Proof of Work (PoW) to create money. The term Proof of Work emerged and got popular later with Bitcoin, but in Wei Dai's B-money an idea of creating money was introduced by providing a solution to a previously unsolved computational problem. This concept is similar to PoW, where money is created by broadcasting a solution to a previously unsolved computational problem.

A major weakness in the given system was that an adversary with higher computational power could generate unsolicited money without allowing the network to adjust to an appropriate difficulty level. The system did not have detailed account on the consensus mechanism between nodes and several security issues such as Sybil attacks were not addressed.

- In contemporary time with Wei Dai, **Nick Szabo**, a computer scientist introduced the concept of **BitGold**, which was also based on the PoW mechanism but had the same problems as B-money with the exception that the network difficulty level was adjustable.
- **Tomas Sander and Amnon Ta-Shma** from the International Computer Science Institute (ICSI), Berkley introduced an **e-cash scheme** under a research paper named **Auditable, Anonymous Electronic Cash** in 1999. It was first time used in Merkle trees to represent coins and Zero-Knowledge Proofs (ZKPs) to prove the possession of coins.

This scheme required a central bank to keep record of all used serial numbers. This scheme allowed users to be fully anonymous. This was a theoretical design which was not practical to implement due to inefficient proof mechanisms.

- **Reusable Proof of Work (RPoW)** was introduced in **2004** by **Hal Finney**, a computer scientist, developer and first person to receive Bitcoin from Satoshi Nakamoto. It used the

hashcash scheme by Adam Back as a proof of computational resources spent to create the money. This was also a central system that kept a central database to keep track of all used PoW tokens. This was an online system that used remote attestation made possible by a trusted computing platform (TPM hardware).

All the above mentioned schemes were intelligently designed but were weak from one aspect or another. Specifically, all the schemes which rely on a central server required to be trusted by the users.

- In **2008**, **Bitcoin** was introduced through a paper called, Bitcoin: A Peer-to-Peer Electronic Cash System. It was written by **Satoshi Nakamoto**, which is believed to be a anonymous name, hence the true identity of Bitcoin inventor is unknown and subject of much speculation.

The first key idea introduced in the paper was of a purely peer-to-peer electronic cash that does need an intermediary bank to transfer payments between peers.

Bitcoin is built on decades of cryptographic research such as the research in Merkle trees, hash functions, public key cryptography, and digital signatures. Moreover, ideas such as BitGold, B-money, hashcash, and cryptographic time stamping provided the foundations for bitcoin invention. All these technologies are cleverly combined in Bitcoin to create the world's first decentralized currency.

The key issue that has been addressed in Bitcoin is an elegant solution to the Byzantine Generals' Problem along with a practical solution of the double spend problem.

The below figure shows analysis of the selected cryptocurrencies is based on the information available to the public via the internet.

Name	Symbol		Market Cap[122]	Supply limit[123]
Bitcoin		BTC	$124.969.093.161	21 million
Ethereum		ETH	$57.462.517.858	TBD[124]
Ripple		XRP	$23.790.387.789	100 billion
Bitcoin Cash		BCH	$17.159.025.225	21 million
Litecoin		LTC	$6.704.709.572	84 million
Stellar		XLM	$5.128.373.973	100 billion
Cardano		ADA	$5.034.129.651	45 billion
IOTA		MIOTA	$4.038.240.572	2,779,530,283,277,761
NEO		NEO	$3.386.383.000	100 million
Monero		XMR	$2.626.586.260	18,4 million
Dash		DASH	$2.592.894.544	17.74 – 18.92 million[125]

Table 2 Cryptocurrencies

III. Types of Wallets

The wallet is a software application which is used to store private or public keys and Bitcoin address. It performs multiple functions, such as receiving and sending bitcoins. Nowadays, software usually offers both functionalities: Bitcoin client and wallet. On the disk, the Bitcoin core client wallets are stored as the Berkeley DB file.

Private keys are generated by randomly choosing a 256-bit number by wallet software. Private keys are used by wallets to sign the outgoing transactions. Wallets do not store any coins, and there is no concept of wallets storing balance or coins for a user. In fact, in the Bitcoin network, coins do not exist; instead, only transaction information is stored on the blockchain which are then used to calculate the number of bitcoins.

There are different types of wallets that can be used to store private keys. As a software program, they also provide some functions to the users to manage and carry out transactions on the Bitcoin network.

- **Non-deterministic wallets**

It contain randomly generated private keys and are also called *just a bunch of key wallets*. The Bitcoin core client generates some keys when first started and generates keys as and when required.

- **Deterministic wallets**

Here keys are derived out of a seed value via hash functions. This seed number is generated randomly and is commonly represented by human-readable mnemonic code words.

- **Hierarchical Deterministic wallets**

Hierarchical Deterministic (HD) wallets store keys in a tree structure derived from a seed. The seed generates the parent key (master key), which is used to generate child keys and, subsequently, grandchild keys. Key generation in HD wallets does not generate keys directly; instead, it produces some information (private key generation information) that can be used to generate a sequence of private keys. The complete hierarchy of private keys in an HD wallet is easily recoverable if the master private key is known.

- **Brain wallets**

The master private key can also be derived from the hash of passwords that are memorized. The key idea is that this passphrase is used to derive the private key and if used in HD wallets, this can result in a full HD wallet that is derived from a single memorized password. This is known as a brain wallet.

- **Paper wallets**

A paper-based wallet with the required key material printed on it. It requires physical security to be stored.

- **Hardware wallets**

Hardware wallet suggests the use of tamper-resistant device to store keys. It is the most secure way of storing any amount of cryptocurrency. There have been no verifiable incidents of money being stolen from a hardware wallet. Unlike paper wallets, which must be imported to software at some point, hardware wallets can be used securely and interactively. Moreover, they are immune to computer viruses, the funds stored cannot be transferred out of the device in plaintext, and in most instances, their software is open source.

Figure 70 Hardware Wallet

Name	Price	Features
Ledger Nano S	58 €	Screen; two buttons that you need to press simultaneously to confirm a transaction, which prevents hackers from hacking into it and confirming payments; PIN code; box ships with an anti-tampering seal
TREZOR	$99	Screen; two buttons; wallet can be backed up with up to 24 words + passphrase; PIN code

KeepKey	$99	Screen; digital screen and metal body; PIN code; number randomization; can be backed with up to 24 words; recovery can be done with

Table 3 Hardware Wallets

- **Online wallets**

Online wallets are stored entirely online and are provided as a service usually via the cloud. They provide a web interface to the users to manage their wallets and perform various functions such as making and receiving payments.

Service	Features
Coinbase	One-stop solution, an exchange integrated with a wallet
Lumi Wallet	Free, easy, client-side interface to generate one wallet that supports BTC, ETH and plenty of ERC20 tokens
Circle	Users can store, send, receive and buy Bitcoins
Blockchain	One of the most popular web-based wallets
Strongcoin	Offers a hybrid wallet, which lets you encrypt your private address keys before sending them to its servers
Xapo	A simple Bitcoin wallet, with the added security of a cold-storage vault

Table 4 Online Wallets

- **Mobile Wallets**

Mobile wallets are applications installed on mobile devices. They can provide various methods to make payments, most significantly the ability to use smartphone cameras to scan QR codes quickly and make payments. Mobile wallets are available for the Android platform and iOS, for example, Blockchain, breadwallet, Copay, and Jaxx.

Name	Operating System	Features
FreeWallet	iOS, Android	Cold storage, withdraw from and to any cryptocurrency
Edge	iOS, Android	Zero-knowledge, single sign-on, one-touch 2 factor authentication

Atomic Wallet	iOS, Android	Very user-friendly, 500+ assets, instant exchange, buy crypto option, custody-free app.
Lumi Wallet	iOS, Android	Secure and easy crypto wallet & exchange for mobile
Blockchain Wallet	iOS, Android	Hierarchical deterministic, enable to browse Bitcoin merchants in your area, open source software
Copay	iOS, Android, Windows Mobile	Can have multiple users, so the group approves each transaction to send money, open source software
Jaxx	iOS, Android	Cold storage, no verification required
Mycelium	iOS, Android	Cold storage, hierarchical deterministic, open source softwar

Table 5 Mobile Wallets

- **Desktop Wallets**

Desktop wallets are downloaded and installed onto your computer, storing your private keys on your hard drive. By definition, they are more secure than online and mobile wallets, as they don't rely on third parties for their data and are harder to steal. They are still connected to the internet, which makes them inherently less secure. However, desktop wallets are a great solution for those who trade small amounts of Bitcoin from their computers.

There is a variety of different options of desktop wallets that cater to different needs. Some focus on security, some on anonymity and so on.

Name	Operating system	Features
Electrum	MacOS, Windows, Linux	One of the most popular, robust, effective and secure desktop wallets; open source; allows you to replace a transaction fee on an already broadcasted transaction, which speeds up the confirmation process; address tagging; encryption
Exodus	MacOS, Windows, Linux	Very user-friendly and easy to understand, reliable wallet

Atomic Wallet	MacOS, Windows, Linux	Atomic is available for Mac OS, Windows and Linux, 500+ assets, keys are encrypted on your device, instant exchange, buy crypto option, custody-free app, 24/7 help center.
Bitcoin Core	MacOS, Windows, Linux	Full node wallet, you need to download the entire blockchain to use it. It allows you to independently verify transactions and not rely on anyone else in the system
Copay	MacOS, Windows, Linux	Multisignature wallet; mobile and desktop; open source
Armory	MacOS, Windows, Linux, Ubuntu, RaspberriPi	Prioritizes safety and security; features a variety of encryption and cold-storage opti

Table 6 Desktop Wallets

IV. Bitcoin

Bitcoin is the first application of blockchain technology. Bitcoin has begun a revolution with the introduction of the very first completely decentralized digital currency, and the one that has proven to be extremely secure and stable from a network and protocol perspective. As a currency bitcoin is quite unstable and highly volatile, although valuable.

Bitcoin can be defined in various ways; it's a protocol, a digital currency, and a platform. It is a combination of peer-to-peer network, protocols, software that facilitates the creation and usage of the digital currency named bitcoin. Nodes in this peer-to-peer network talk to each other using the Bitcoin protocol.

Since its introduction in 2008 by Satoshi Nakamoto, Bitcoin has gained massive popularity, and at present it is currently the most successful digital currency in the world with billions of dollars invested in it. Its popularity is also evident from the high number of users and investors, increasing bitcoin price, everyday news related to Bitcoin, and the number of start-ups and companies that are offering bitcoin-based online exchanges, and it's now also traded as Bitcoin Futures on Chicago Mercantile Exchange (CME).

The name of the Bitcoin inventor Satoshi Nakamoto is believed to be anonymous, as the true identity of Bitcoin inventor is unknown. It is built on tremendous research in the field of cryptography, digital cash, and distributed computing.

a) Bitcoin Transaction Using a Blockchain Wallet

For demonstration we are using Blockchain wallet for mobile devices.

1. To initiate transaction sender needs the address of beneficiary, which can be obtained from payment request sent from a user, via any appropriate communication mechanism. The sender can also initiate a transfer to send money to another user. In any case, the address of beneficiary is required.
   Here we begin with creating request

Figure 71 Transaction Request

2. The sender either enters the receiver's address or scans the QR code that has the Bitcoin address, amount and optional description encoded in it.

Figure 72 Receivers QR code

3. In the wallet of the sender, this transaction is constructed by following some rules and broadcasted to the Bitcoin network. From a user's point of view, once the QR code is decoded the transaction will appear similar to what is shown in the following screenshot. (How the transaction is created, digitally signed, broadcasted, validated and added to the block is explained in following sections.)

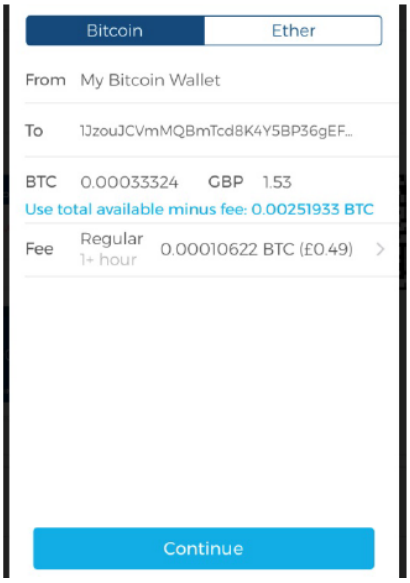


Figure 73 Transaction Initiation

Fee is calculated based on the size of the transaction and a fee rate is a value that depends on the volume of the transaction in the network. This is represented in Satoshis/byte. Fee in Bitcoin network ensures that your transaction will be included by miners in the block.

4. Once the transaction is sent it will appear as shown here in the Blockchain wallet software:

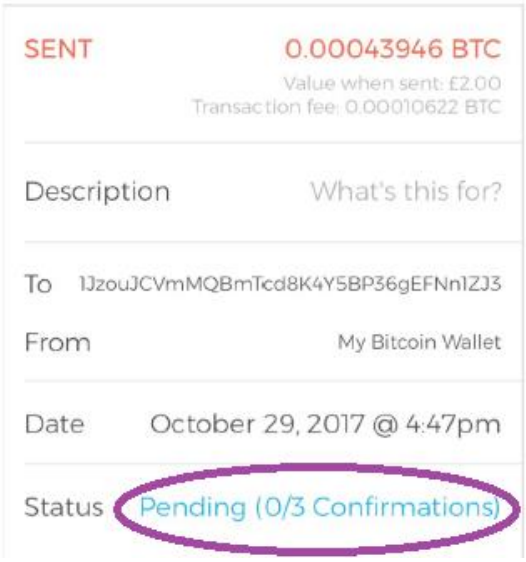


Figure 74 Transaction Initiated but Confirmation Pending

After the transaction has been constructed, signed and sent out to the Bitcoin network. This transaction will be picked up by miners to be verified and included in the block.
In the preceding screenshot, confirmation is pending for this transaction. These confirmations will start to appear as soon as the transaction is verified, included in the block, and mined.

5. The following screenshot visually shows how the transaction flowed on the network from origin (sender) to receivers on the right-hand side.

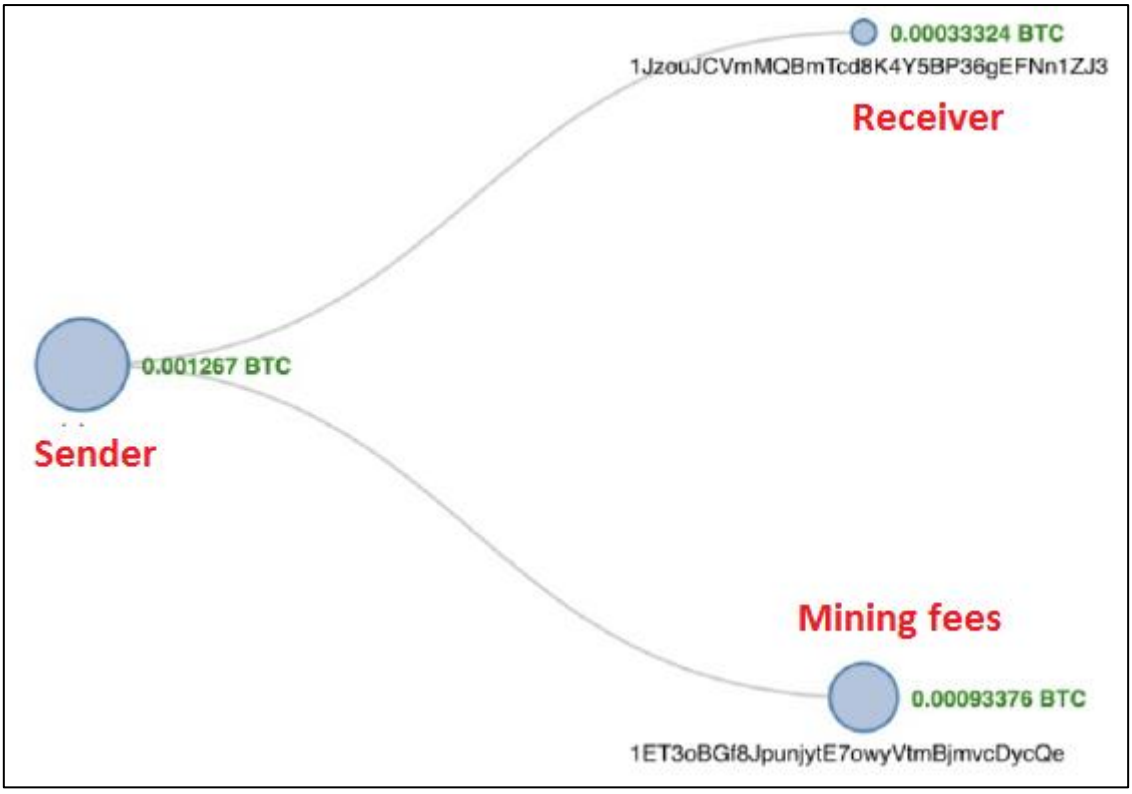


Figure 75 Transaction Flow

Here a payment of 0.001267 BTC (approximately 11 USD) is originated from the sender's address and been paid to receiver's address (starting with 1Jz). The fee of 0.00010622 (approximately 95 cents) is also deducted from the transaction as mining fee.

6. A summary view of various attributes of the transaction:

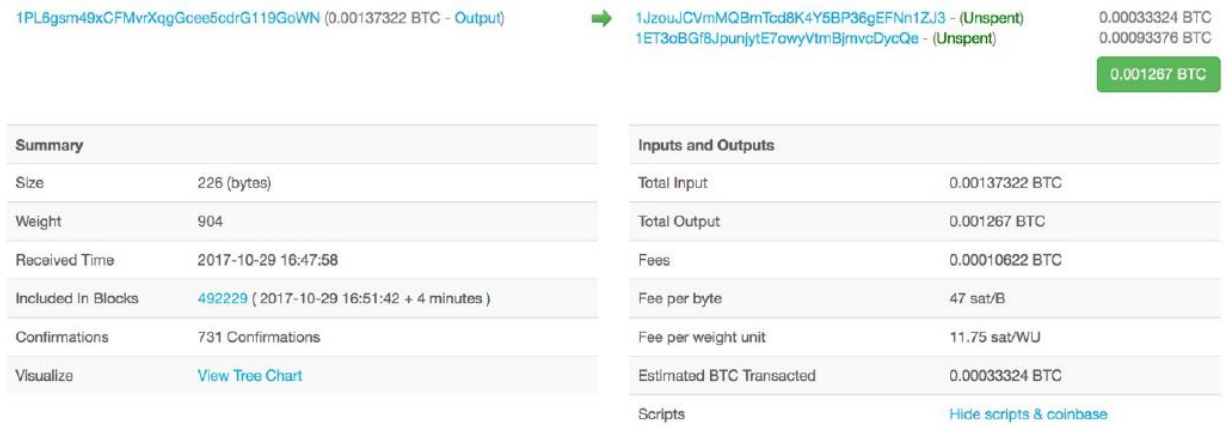


Figure 76 Attributes of Transaction

There numbers of fields involved in transaction that contain various values are listed here with their purpose and explanation:

- **Size:** This is the size of the transaction in bytes.
- **Weight:** This is the new metric given for block and transaction size since the introduction of **Segregated**
- **Witness** (**SegWit**) version of Bitcoin.
- **Received Time:** This is the time when the transaction is received.

- **Included In Blocks:** This shows the block number on the blockchain in which the transaction is included.
- **Confirmations:** This is the number of confirmations by miners for this transaction.
- **Total Input:** This is the number of total inputs in the transaction.
- **Total Output:** This is the number of total outputs in the transaction.
- **Fees:** This is the total fees charged.
- **Fee per byte:** This field represents the total fee divided by the number of bytes in a transaction. For example, 10 Satoshis per byte.
- **Fee per weight unit:** For legacy transaction it is calculated using *total number of bytes * 4*. For SegWit
- transactions it is calculated by combining SegWit marker, flag, and witness field as one weight unit and each byte of other fields as four weight units.

The bitcoin currency, being digital has various denominations which are shown in the following table. A sender or receiver can request any amount. The smallest bitcoin denomination is the Satoshi. The bitcoin currency units are described as follows:

DENOMINATION	ABBREVIATION	FAMILIAR NAME	VALUE IN BTC
Satoshi	SAT	Satoshi	0.00000001 BTC
Microbit	µBTC (uBTC)	Microbitcoin or Bit	0.000001 BTC
Millibit	mBTC	Millibitcoin	0.001 BTC
Centibit	cBTC	Centibitcoin	0.01 BTC
Decibit	dBTC	Decibitcoin	0.1 BTC
Bitcoin	BTC	Bitcoin	1 BTC
DecaBit	daBTC	Decabitcoin	10 BTC
Hectobit	hBTC	Hectobitcoin	100 BTC
Kilobit	kBTC	Kilobitcoin	1000 BTC
Megabit	MBTC	Megabitcoin	1000000 BTC

Table 7 Bitcoin Denomination

b) Key Elements of Bitcoin

- Digital keys

Elliptic Curve Cryptography (**ECC**) is used to generate public and private key pairs in the Bitcoin network.

Private keys are needed to be kept safe and it normally resides only on the owner's side. Private keys are used to digitally sign the transactions proving the ownership of the bitcoins.
Private keys are usually encoded in **Wallet Import Format** (**WIF**) form in order to make them easier to copy and use. It is a way to represent the full size private key in a different format. WIF can be converted into a private key and vice versa.
An example of a private key:
"A3ED7EC8A03667180D01FB4251A546C2B9F2FE33507C68B7D9D4E1FA5714195201"
Private key converted into WIF format looks like this:
"L2iN7umV7kbr6LuCmgM27rBnptGbDVc8g4ZBm6EbgTPQXnj1RCZP"
Public keys exist on the blockchain and all network members have access to it. Public keys are derived from private keys due to their special mathematical relationship with the private keys. Once a transaction signed with the private key is broadcasted on the Bitcoin network, public keys are used by the nodes to verify that the transaction has indeed been signed with the corresponding private key. This process of verification proves the ownership of the bitcoin.

- Addresses

A bitcoin address is created by taking the corresponding public key of a private key and hashing it twice, first with the SHA-256 algorithm and then with RIPEMD-160. The resultant 160-bit hash is then prefixed with a version number and finally encoded with a Base58Check encoding scheme. The bitcoin addresses are 26-35 characters long and begin with digit 1 or 3. Currently, there are two types of addresses, the commonly used P2PKH and another P2SH type, starting with number 1 and 3, respectively.

A typical bitcoin address looks like a string shown below:
"1ANAguGG8bikEv2fYsTBnRUmx7QUcK58wt"

- Transactions

Transactions are the core functions of the bitcoin ecosystem. Transactions can be as simple as just sending some bitcoins to a bitcoin address, or it can be complex depending on the requirements. Each transaction is composed of at least one input and output. Inputs can be thought of as coins being spent that have been created in a previous transaction and outputs as coins being created. If a transaction is minting new coins, then there is no input and therefore no signature is needed. If a transaction is to send coins to some other user (a bitcoin address), then it needs to be signed by the sender with their private key and a reference is also required to the previous transaction in order to show the origin of the coins. Coins are, in fact, unspent transaction outputs represented in Satoshis.
Transactions are not encrypted and are publicly visible in the blockchain. Blocks are made up of transactions and these can be viewed using any online blockchain explorer.

- Blockchain

Blockchain is a distributed ledger of a timestamped, ordered, and immutable list of all transactions on the Bitcoin network. Each block is identified by a hash in the chain and is linked to its previous block by referencing the previous block's hash.

- Mining

Mining is a process by which new blocks are added to the blockchain. Blocks contain transactions that are validated via the mining process by mining nodes on the Bitcoin network. Blocks, once mined and verified are added to the blockchain which keeps the blockchain growing. This process is resource-intensive due to the requirements of PoW where miners compete in order to find a number which is less than the difficulty target of the network. This difficulty in finding the correct value (also called sometimes the mathematical puzzle) is there

to ensure that the required resources have been spent by miners before a new proposed block can be accepted.

New coins are minted by the miners by solving the PoW problem, also known as partial hash inversion problem. This process consumes a high amount of resources including computing power and electricity. This process also secures the system against frauds and double spending attacks while adding more virtual currency to the Bitcoin ecosystem.

Once a node connects to the bitcoin network, there are several tasks that a bitcoin miner performs:

- **Synching up with the network**: Once a new node joins the bitcoin network, it downloads the blockchain by requesting historical blocks from other nodes. This is mentioned here in the context of the bitcoin miner; however, this not necessarily a task only for a miner.
- **Transaction validation**: Transactions broadcasted on the network are validated by full nodes by verifying and validating signatures and outputs.
- **Block validation**: Miners and full nodes can start validating blocks received by them by evaluating them against certain rules. This includes the verification of each transaction in the block along with verification of the nonce value.
- **Create a new block**: Miners propose a new block by combining transactions broadcasted on the network after validating them.
- **Perform Proof of Work**: This task is the core of the mining process and this is where miners find a valid block by solving a computational puzzle. The block header contains a 32-bit nonce field and miners are required to repeatedly vary the nonce until the resultant hash is less than a predetermined target.
- **Fetch reward**: Once a node solves the hash puzzle (PoW), it immediately broadcasts the results, and other nodes verify it and accept the block. There is a slight chance that the newly minted block will not be accepted by other miners on the network due to a clash with another block found at roughly the same time, but once accepted, the miner is rewarded with 12.5 bitcoins and any associated transaction fees.

- The Bitcoin networks

The Bitcoin network is a peer-to-peer network where nodes exchange transactions and blocks. There are different types of nodes on the network. There are two main types of nodes, **full nodes** and **SPV nodes**.

Full nodes are implementations of Bitcoin core clients performing the wallet, miner, full blockchain storage, and network routing functions. However, it is not necessary to perform all these functions.

Simple Payment Verification (SPV) nodes or lightweight clients perform only wallet and network routing functionality.

- Wallets (client software)

The wallet is a software application which is used to store private or public keys and Bitcoin address. It performs multiple functions, such as receiving and sending bitcoins. Nowadays, software usually offers both functionalities: Bitcoin client and wallet. On the disk, the Bitcoin core client wallets are stored as the Berkeley DB file.

Private keys are generated by randomly choosing a 256-bit number by wallet software. Private keys are used by wallets to sign the outgoing transactions. Wallets do not store any coins, and there is no concept of wallets storing balance or coins for a user. In fact, in the Bitcoin network, coins do not exist; instead, only transaction information is stored on the blockchain which are then used to calculate the number of bitcoins.

V. Ethereum

Vitalik Buterin conceptualized Ethereum in November, **2013.** The core idea proposed was the development of a Turing-complete language that allows the implemntation of arbitrary programs (smart contracts) for blockchain and decentralized applications. This concept is in contrast to Bitcoin, where the scripting language is limited in nature and allows necessary operations only.

The following table shows the timeline for releases of Ethereum starting from the first release to the planned final release:

Version	**Release date**
Olympic	May, 2015
Frontier	July 30, 2015
Homestead	March 14, 2016
Byzantium (first phase of Metropolis)	October 16, 2017
Metropolis	To be released
Serenity (final version of Ethereum)	To be released

Table 8 Versions of Ethereum

The core idea in Ethereum blockchain is, executing transactions incrementally from a genesis state is into a final state. The final transformation is then accepted as the absolute undisputed version of the state. In the following diagram, the Ethereum state transition function is shown, where a transaction execution has resulted in a state transition:

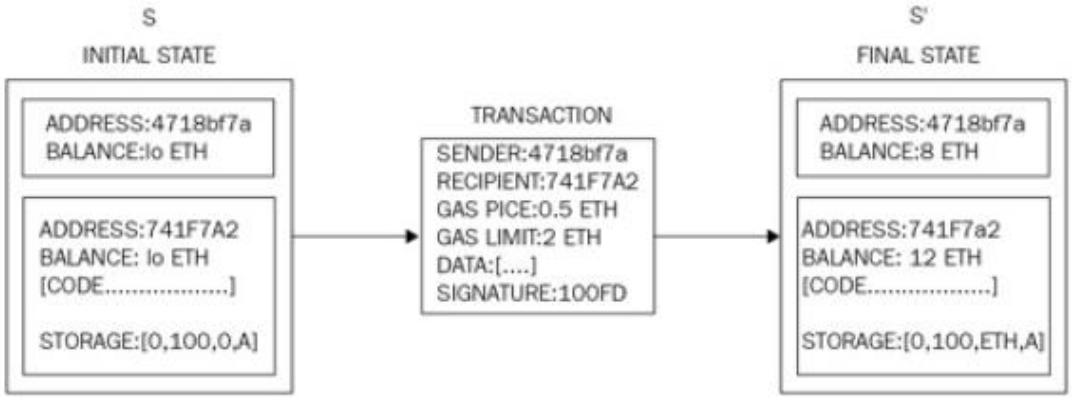


Figure 77 Ethereum State Transition Function

a) The Ethereum network

The Ethereum network is a peer-to-peer network where nodes participate in order to maintain the blockchain and contribute to the consensus mechanism. Networks can be divided into three types, based on requirements and usage. These types are described in the following subsections.

- **Mainnet**

Mainnet is the current live network of Ethereum. The current version of mainnet is Byzantium (Metropolis) and its chain ID is 1. Chain ID is used to identify the network.

- **Testnet**

Testnet is also called Ropsten and is the widely used network for the Ethereum blockchain. This test blockchain is used to test smart contracts and DApps before being deployed to the production live blockchain. Moreover, being a test network, it allows experimentation and research. The main testnet is called Ropsten which contains all features of other smaller and special purpose testnets that were created for specific releases.

Other testnets include Kovan and Rinkeby which were developed for testing Byzantium releases. The changes that were implemented on these smaller testnets also been implemented on Ropsten. Now the Ropsten test network contains all properties of Kovan and Rinkeby.

- **Private net**

This is the private network that can be created by generating a new genesis block. This is usually the case in private blockchain distributed ledger networks, where a private group of entities start their blockchain and use it as a permissioned blockchain.

The following table shows the list of Ethereum network with their network IDs. These network IDs are used to identify the network by Ethereum clients.

Network name	Network ID / Chain ID
Ethereum mainnet	1
Morden	2
Ropsten	3
Rinkeby	4
Kovan	42
Ethereum Classic mainnet	61

Table 9 Ethereum Network IDs

b) Components of the Ethereum Blockchain

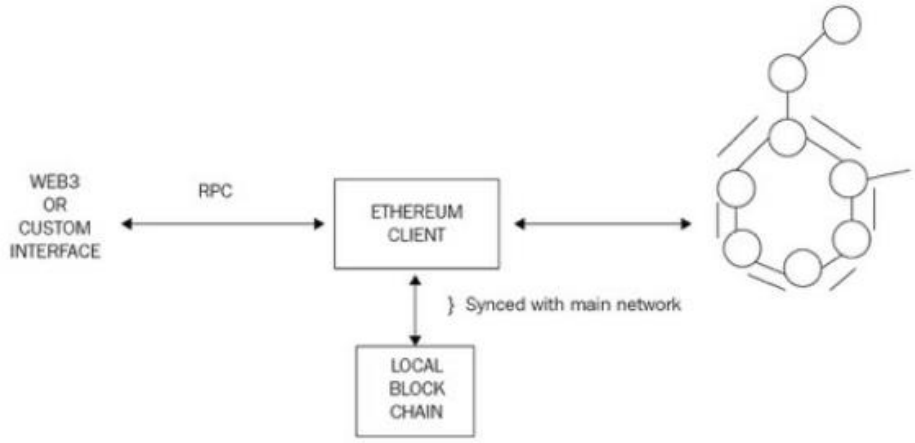


Figure 78 Ethereum Architecture

The Ethereum blockchain stack consists of various components.

- Ethereum blockchain running on the peer-to-peer **Ethereum network.**
- **Ethereum client** (usually Geth) that runs on the nodes and connects to the peer-to-peer Ethereum network from where blockchain is downloaded and stored locally. It provides various functions, such as mining and account management. The local copy of the blockchain is synchronized regularly with the network.
- **web3.js library** that allows interaction with the Geth client via the Remote Procedure Call (RPC) interface.
- **Keys and addresses**
  Keys and addresses are used in Ethereum blockchain to represent ownership and transfer of Ether. Keys are used in pairs of private and public type. The private key is generated randomly and is kept secret whereas a public key is derived from the private key. Addresses are derived from the public keys which are a 20-bytes code used to identify accounts.
  The process of key generation and address derivation is described here:
  1. A private key is randomly chosen (256 bits positive integer) under the rules defined by elliptic curve secp256k1 specification (in the range [1, secp256k1n − 1]).
  2. The public key is then derived from this private key using ECDSA recovery function. We will discuss this later in the next section, Accounts in the context of digital signatures.
  3. An address is derived from the public key which is the right most 160 bits of the Keccak hash of the public key.
- **Accounts**

Accounts are one of the main building blocks of the Ethereum blockchain. Ethereum is a transaction driven state mechanism, the state is created or updated as a result of the interaction between accounts and transaction execution. Operations performed between and on the accounts, represent state transitions.

Two kinds of accounts exist in Ethereum:

Externally Owned Accounts (EOAs)

- EOAs are similar to accounts that are controlled by a private key in Bitcoin.
- EOAs has ether balance
- They are capable of sending transactions
- They have no associated code
- They are controlled by private keys
- Accounts contain a key-value store
- They are associated with a human user

Contract Accounts (CAs)

- CAs are the accounts that have code associated with them along with the private key.
- CAs have Ether balance.
- They have associated code that is kept in memory/storage on the blockchain.
- They can get triggered and execute code in response to a transaction or a message from other contracts. It is worth noting that due to the Turing-completeness property of the Ethereum blockchain, the code within contract accounts can be of any level of complexity. The code is executed by Ethereum Virtual Machine (EVM) by each mining node on the Ethereum network.
- CAs can maintain their permanent state and can call other contracts.
- They are not intrinsically associated with any user or actor on the blockchain.
- CAs contain a key-value store.

- **Transactions and messages**

A transaction in Ethereum is a digitally signed data packet using a private key that contains the instructions, upon completion of which, either result in a message call or contract creation. Transactions can be divided into two types based on the output they produce:

Message call transactions: This transaction just produce a message call that is used to pass messages from one contract account to another.

Contract creation transactions: These transactions result in the creation of a new contract account. This means that when this transaction is executed successfully, it creates an account with the associated code.

- **Ether cryptocurrency/tokens**

As an incentive to the miners, Ethereum rewards its own native currency called Ether (ETH), as an incentive to the miners.

There are two Ethereum blockchains:

1. Ethereum Classic, and its currency is represented by ETC

2. ETH it is a the hard-forked version, which continues to grow and on which active development is being carried out.

Ether is minted by miners as currency rewards for their computational effort spend to secure the network by verifying and with validation transactions and blocks. Ether is used within the Ethereum blockchain to pay for the execution of contracts on the EVM. Ether is used to purchase gas as crypto fuel, which is required to perform computation on the Ethereum blockchain.

- **The EVM**

EVM is a simple stack-based execution machine that runs bytecode instructions to transform the system state from one state to another. The word size of the virtual machine is set to 256-bit. The stack size is limited to 1024 elements and is based on the Last In, First Out (LIFO) queue.

EVM is a Turing-complete machine but is limited by the amount of gas that is required to run any instruction. This means that infinite loops that can result in denial of service attacks are not possible due to gas requirements.
EVM also supports exception handling, in case exceptions occur, such as not having enough gas or invalid instructions, in which case the machine would immediately halt and return the error to the executing agent.
EVM is an entirely isolated and sandboxed runtime environment. The code that runs on the EVM does not have access to any external resources, such as a network or filesystem. This results in increased security, deterministic execution and allows untrusted code (anyone can run code) to be run on Ethereum blockchain.

- **Smart contracts**

A smart contract is a secure and unstoppable computer program representing an agreement that is automatically executable and enforceable.
A smart contract is nothing but a computer program that is written in a language that a computer or target machine can understand. It comprehends agreements between parties in the form of business logic. Smart contracts are automatically executed when certain conditions are met. They are enforceable, which means all contractual terms are executed as defined and expected, even in the presence of adversaries.

VI. Ripple

Ripple is an open-source, P2P decentalized digital payment platform that allows for near instantaneous transfers of currency regardless of their form (e.g. US Dollar, Yen, Bitcoin, …) It was launched in 2012 by the private company Ripple (Labs), and it is responsible for the further development of the Ripple protocol.
It is first ever company to receive a "BitLicense" for an institutional use case of digital assets from New York's Department of Financial Services. It is also getting support from a number of big players in the financial services industry, such as Bank of America Merill Lynch, Santander, etc.
Ripple's inventors launched the cryptocurrency XRP. XRP was built to become a bridge currency to allow financial institutions to settle cross-border payments a lot faster and cheaper than they can using the global payment networks that are in place today, which can be slow and involve multiple middlemen (i.e. banks). However, in practice, Ripple's payment platform does not need a bridge currency to actually work. According to Ripple, XRP can handle more than 1,500 transactions per second. While it was initially developed and intended for enterprise use, it has meanwhile been adopted by a large number of cryptocurrency users. Ripple (XRP) is not based on a PoW or a PoS mechanism to validate transactions, but it makes use of its own specific consensus protocol. The total supply of XRP has been fully "pre-mined" by its inventors.
Unlike Ethereum's inventors, Ripple's inventors did not sell a portion of XRP via a crowdsale upon XRP's creation to fund Ripple (Labs), Inc. The company was privately funded. At present, it is not fully transparent how XRP, mainly held by Ripple (Labs), Inc.) is or will be further distributed in the future.

- **Ripple runs on a public permissioned blockchain**
  Unlike Bitcoin and Ethereum, Ripple runs on a permissioned blockchain. This is because Ripple (Labs) Inc., the company behind Ripple (XRP), determines who may act as a transaction validator on its network. The blockchain itself is considered public, as it can be accessed and viewed by anyone.
- **Ripple (XRP) is directly convertible into fiat currency**
  Like Bitcoin, XRP can be directly converted into fiat currency on various crytocurrency exchanges (e.g. Kraken, LiteBit, Anycoin Direct, Bitsane, …).

- **Ripple (XRP) is a medium of exchange**
  Ripple (XRP) is being accepted as a means of payment by a growing number of (online) merchants for various goods and services (e.g. e-cigarettes, honey, coffee, …). There is recently even speculation on the internet that Amazon might be looking to adopt Ripple in the near future.

- **Ripple (XRP) is a pseudo-anoymous coin**
  Like Bitcoin, Ripple (XRP) can be qualified as a pseudo-anonymous coin.

VII. Litecoin

Like Bitcoin, Litecoin (LTC) is an open-source decentralized P2P cryptocurrency. It was launched in October 2011 and is based on what is known as the Scrypt PoW algorithm, which utilises Bitcoin's original SHA-256 PoW algorithm. Litecoin is often described as the 'silver' to Bitcoin's gold.

- **Apart from the fact that it uses a different algorithm, it is different from Bitcoin in two ways**.
  - Litecoin offers a much faster transaction speed than Bitcoin. The time needed to generate a block on the Bitcoin BC is about ten minutes, while the average block creation time on the Litecoin blockchain is approximately 2.5 minutes.
  - The total supply limit of Litecoin is with 84 million coins, much higher than the 21 million supply limit of Bitcoin.
- **Litecoin runs on an open, permissionless blockchain:**
  Just like Bitcoin, Litecoin runs on on an open, permissionless blockchain. All that is needed to join the network is a download of the open-source software code.
- **Litecoin is directly convertible into fiat currency:**
  Litecoin can be bought with fiat currency on a number of cryptocurrency exchanges (e.g. BTCDirect, LiteBit, Coinbase, Anycoin Direct, …) and on those exchanges, it can also easily be exchanged for fiat currency.
- **Litecoin is a medium of exchange:**
  Litecoin is accepted as a means of payment by a gradually growing number of online merchants. Like Bitcoin, it thus also constitutes a medium of exchange.
- **Litecoin is a pseudo-anonymous coin:**
  Just like Bitcoin, Litecoin is a pseudo-anonymous coin. Everyone can verify the chain of LTC transactions on the basis of the public ledger, which would make it technically possible to identify the coins sender and/or receiver.
- **Litecoin and the case of "Atomic Swaps":**
  Litecoin community recently introduced a new technology into the crypto-world which is referred to as the "atomic swap". In simple terms, an atomic swap enables a P2P cross-chain exchange or trade of one cryptocurrency for another cryptocurrency, without the need of a third-party. For example, if Anuj has one Bitcoin and he wants 100 Litecoins in return, he would normally have to go through an exchange (i.e. a third-party) and pay certain fees to get this trade done. Suppose that Shri owns 100 Litecoins and she instead wants one Bitcoin, then with an atomic swap Anuj and Shri could simply trade their Coins with one another.
  Now, in practice an atomic swap is not so easy. First of all, since it is still in its initial phase, the implementation of the atomic swap technology requires a lot of IT-knowledge. For example, a link has to be made between the two cryptocurrency blockchains, which requires the implementation of an IT-protocol known in the crypto-community as the "Lightning Protocol". In addition, both blockchains have to share the

same cryptographic function (for example the SHA-256 function) in order to perform atomic swap to be possible.

VIII. Components in Cryptocurrency system

a) Cryptocurrency user

A cryptocurrency user is a natural person or legal entity who obtains coins to use them for various purposes like

(i) to purchase real or virtual goods or services
(ii) to make P2P payments, or
(iii) to hold them for investment purposes

Cryptocurrency users can obtain coin through multiple ways:

- User can simply buy his coins on a cryptocurrency exchange using fiat money or another cryptocurrency;
- User can buy his coins directly from another cryptocurrency user (i.e. through a trading platform, referred to as a "P2P exchange");
- If a cryptocurrency is based on a PoW consensus mechanism, user can mine a new coin (i.e. participate in the validation of transactions by solving of a "cryptographic puzzle" and be rewarded a new coin);
- In some cases user can obtain his coins directly from the coin offeror, either as part of a free initial offering of coins or in the framework of a crowd sale set-up by the coin offeror (e.g. a large bulk of ether (cf. Ethereum) was sold in a crowdsale to cover certain development costs);
- If user sells goods or services in exchange for cryptocurrency, he can also receive coins as a payment for those goods or services;
- In case of a "hard fork" of a coin's blockchain, user will automatically obtain an amount of the newly created coin;
- User can receive coins as a gift or donation from another cryptocurrency user

b) Cryptocurrency Miners

"Miners" are important entity of cryptocurrency since they participate in validating transactions on the blockchain by solving a "cryptographic puzzle". The process of mining relates to cryptocurrencies that are based on a PoW consensus mechanism. A miner supports the network by harnessing computing power to validate transactions and is rewarded with newly mined coins. Miners can be cryptocurrency users, or, more commonly, parties who have made a new business out of mining coins to sell them for fiat currency (such as US Dollar or Euro).

c) Cryptocurrency Exchanges

Cryptocurrency exchanges are entities that facilitate exchange services to cryptocurrency users, usually against payment of a certain fee (i.e. a commission). They allow cryptocurrency users to sell their coins for fiat currency or buy new coins with fiat currency. They generally function both as a bourse and as a form of exchange office. Examples of well-known cryptocurrency exchanges are: Bitfinex, HitBTC, Kraken and Coinbase GDAX. Some exchanges are pure cryptocurrency exchanges, which means that they only accept payments in other cryptocurrencies, usually Bitcoin (for example Binance), whereas others also accept payments in fiat currencies such as US dollar or Euro (for example Coinbase).

Moreover, many cryptocurrency exchanges only allow their users to buy a particular selection of coins. Many cryptocurrency exchanges (i.e. both regular and pure cryptocurrency exchanges) operate as custodian wallet providers (for example Bitfinex). In general cryptocurrency exchanges offer their users a wide array of payment options, such as wire transfers, PayPal transfers, credit cards and other coins. Some cryptocurrency exchanges also

provide statistics on the cryptocurrency market (like trading volumes and volatility of the coins traded) and offer conversion services to merchants who accept payments in cryptocurrencies.

d) Wallet Providers

Wallet providers are those entities that provide cryptocurrency users digital wallets or e-wallets which are used for holding, storing and transferring coins. A wallet holds a cryptocurrency user's cryptographic keys. A wallet provider simply translates a cryptocurrency user's transaction history into human readable format, which looks much like a regular bank account transaction detail.

There are several types of wallet providers:

- **Hardware wallet providers** that provide cryptocurrency users with specific hardware solutions to privately store their cryptographic keys (e.g. Ledger Wallet;
- **Software wallet providers** that provide cryptocurrency users with software applications which allow them to access the network, send and receive coins and locally save their cryptographic keys (e.g. Jaxx);
- **Custodian wallet providers** that take (online) custody of a cryptocurrency user's cryptographic keys (e.g. Coinbase).

e) Coin Inventotrs

Coin inventors are individuals or organizations who have developed the technical foundations of a cryptocurrency and set the initial rules for its use. In some cases, their identity is known (e.g. Ripple, Litecoin, Cardano), but more often they remain unidentified (eg. Bitcoin, Monero). Some remain involved in maintaining and improving the cryptocurrency's code and underlying algorithm, while others simply disappear (e.g. Bitcoin).

f) Coin offerors

Coin offerors are individuals or organizations that offer coins to cryptocurrency users upon the coin's initial release, either against payment (i.e. through a crowdsale) or at no charge (i.e. in the framework of a specific (sign-up) program (e.g. Stellar)), normally to fund the coin's further development or boost its initial popularity. The coins issued by coin offerors offer to cryptocurrency users are created or pre-mined prior to the coin's official release / the coin's inception. Coins that are distributed this way are either partially pre-mined or pre-created (i.e. cryptocurrency users can still generate more coins after the release), or are fully pre-mined or pre-created. In the latter case the coin offeror usually retains a large portion of the coins (e.g. this is the case with Stellar). It is important to note that not all coins have an identifiable coin offeror, nor are all coins pre-mined or is its full supply pre-created. A coin offeror can be the same person as the coin inventor, or another individual or organization.

IX. Challenges in Cryptocurrency

Cryptocurrencies despite of its huge advantages and multiple future perspective, is not free from financial problems and security concerns. The main problems and impacts of cryptocurrency can include:

a. <u>Security threats:</u>

Hackers and malicious users if successful in breaking cryptocurrency system, they can create as much as they want from virtual currency. This will lead to the ability to create fake virtual currency or steal virtual currency by just seating in front of computer. For example, selling in-game virtual items and virtual currency is against World of Warcraft (WoW) game policies. Therefore, many users log into WoW gold selling websites to buy virtual gold in order to pay for virtual items that they need. Many of WoW gold selling websites are not reliable and they

are vulnerable to hacking and many users are complaining about paying real money for nothing or for fake virtual currency

b. Collapse concerns in cryptocurrency systems:

Unlimited issuing of cryptocurrency in the variety crypto communities will lead to economic problems since its issuing is not based on the demand and supply. It is possible for some providers such as Second Life to issue unlimited Linden Dollars and increase their virtual items prices in order to gain more real revenues. On the other hand, it will suffer from inflation and economic issues leading to collapse in the cryptocurrency system.

c. Impact on real economy:

Since some crypto currency systems are connected with real world monetary systems, they may affect the demands and supply facilities of main stream economy. For example, enabling users to purchase virtual and real goods and services with cryptocurrency in some platforms may reduce the demands on fiat money. Users will no longer depend on fiat money for their purchase. On the other hand, some platforms enable users to exchange their virtual currency with real currency and this will increase the demands on fiat currency. This fluctuation will affect on the main stream economy.

d. Gold farming risks:

Gold farmers are players who play in social games such as World of Warcraft in order to gain gold, which is virtual currency of the game, and then sell it for real money. The targeted buyers are the players who do not have enough time to play and compete for gaining virtual currency. In fact, huge cash flow is generated from gold farming process and it is not controlled and regulated. This will increase fraud and financial risks where virtual currency is exchanged with real money in unreliable environment.

e. Money laundering:

Money laundering is one risk that is very likely to rise with the use of cryptocurrency especially with platforms that enable users to exchange cryptocurrency with fiat money.

f. Unknown identity risks:

Since most of the cryptocurrencies keep identity of user anonymous, financial transactions cannot be monitored very well. There is no way to recognize the source of creating or cashing out the virtual currencies. This leads to inability to track the transactions in case of money laundering suspicion. Moreover, unknown identity will enable criminals to get paid with virtual currency for their crimes.

g. Black market for cryptocurrency:

The increasing popularity of virtual currency in online environment has led to a thriving black market for trading virtual currency with real money. By observing several social games' forums, some fraud cases have been raised and discussed between users. For example, when a gamer decides to quit from a game, he/she may want to sell the owned virtual currency by offering them in the game's forums. The way of receiving the payments is risky since many malicious users may not complete the payment or they dispute after paying. In this case, they will get their money back plus the virtual currency.

3.Advance Digital Forensics

1. Introduction to Disk Forensics

Disk forensics is the science of extracting forensic information from digital storage media like Hard disk, USB devices, Firewire devices, CD, DVD, Flash drives, Floppy disks etc. The process of Disk Forensics includes Identification of digital evidence, Seizure & Acquisition of the evidence and authentication of the evidence.

Main process of Disk Forensics involves the following actions:

- **Identification** (recognize incident, requirement for action, intelligence for investigation)
- **Authentication**
- **Preparation** (intelligence for search, adequate toolkits, operational briefing, task allocation)
- **Securing and Evaluating the Scene** (ensure safety, confirm computer equipment present and recognize further possibilities, secure equipment, identify and protect evidence, conduct interviews)
- **Documenting the Scene** (create a permanent record of the scene by means of photography and note taking, document condition and location of computers and related components whether these are to be removed or not, mark and label artefacts, use seals and sealable containers, evidence bags)
- **Evidence Collection** (cater for computer devices found to be switched on or off, attending to order of volatility (see Glossary), collect computer hardware and media while preserving evidential value, obtain analogue evidence such as passwords, handwritten notes, computer manuals, printouts)
- **Packaging, Transportation and Storage** (protect equipment and media during transfer avoiding extreme temperatures, physical impact and vibration, static electricity and magnetic sources, establish procedures for reception and storage of machines and media, maintain chain of custody, inventory for storage in secure area free of contaminants)
- **Initial Inspection** (identification of devices, external and internal physical examination of computers, tool selection and expectations)
- **Forensic Imaging and Copying** (e.g. for hard drive – removal of physical disk from computer, digital preview and capture using physical or logical disk acquisition, with write blockers, followed by return of original media to evidence custodian)
- **Forensic Examination and Analysis** (use forensic techniques and tools for analysis and processing including: creation of cryptographic hash values and filtering with hash libraries, file viewing, file exporting and expansion of compound files (e.g. email), extraction of metadata, searching and indexing)
- **Presentation and Report** (document procedures, analysis and findings, use log files, bookmarks and notes made during the examination, make conclusions, prepare exhibits suitable for court)

I. Digital Evidence

Computer forensics involves acquiring digital evidence from a computer hard drive, a mobile phone, a tablet or PDA, or other storage media (like CD/DVD, USB thumb drive) among other places, in a systematic way.

'Digital evidence' is any kind of file or data/metadata that is presented in digital format and could be used for trial in court of law.

a) Digital Evidence Types

There are two main types of digital evidences with respect to who has created them. i.e. User-created data and Machine/ Network Created Data.

User-created data includes anything created by a user using a digital device. It includes the following and more:

- Text files (e.g. MS Office documents, IM chat, bookmarks), spreadsheets, database, and any text stored in digital format,
- Audio and video files,
- Digital images,
- Webcam recordings (digital photos and videos),
- Address book and calendar,
- Hidden and encrypted files (including zipped folders) created by the computer user,
- Previous backups (including both cloud storage backups and offline backups like CD/DVDs and tapes)
- Account details (username, picture, password),
- E-mail messages and attachments (both online and client e-mails as Outlook),
- Web pages, social media accounts, cloud storage, and any online accounts created by the user.

Machine/network-created data includes any data which is auto generated by a digital device. It includes the following and more:

- Computer logs. These include the following logs under Windows OS: Application, Security, Setup, System, Forward Events, Applications, and Services Logs.
- Router logs, including third-party service provider (e.g., Internet service providers (ISPs) commonly store users' account web browsing history logs)
- Configuration files and audit trails
- Browser data (browser history, cookies, download history)
- Instant messenger history and buddy list (Skype, WhatsApp)
- GPS tracking info history (from devices with GPS capability)
- Device Internet protocol (IP) and MAC addresses in addition to the IP addresses associated with a LAN network and the broadcast settings,
- Applications history (e.g., recently opened file on MS Office) and Windows history,
- Restore points under Windows machines
- Temporary files
- E-mail header information
- Registry files in Windows OS
- System files (both hidden and ordinary)
- Printer spooler files
- Hidden partition and slack space (can also contain hidden user information)
- Bad cluster
- Paging and hibernation files
- Memory dump files
- Virtual machines
- Surveillance video recordings

II. File system and data storage

File systems provide a mechanism for the operating system to keep track of files in a partition. Before user can use a storage device to store data and install applications and OS, user need to

initialize it first through writing the data structures of the file system to the drive. Windows OS uses either the FAT or the NTFS file system to install itself on hard drives.
File system analysis examines data in a volume (i.e., a partition or disk) and interprets them as a file system. There are many end results from this process, but examples include listing the files in a directory, recovering deleted content, and viewing the contents of a sector.

a) FAT

The *File Allocation Table* (FAT) file system is one of the simplest file systems found in common operating systems. FAT is the primary file system of the Microsoft DOS and Windows 9x operating systems, but the NT, 2000, and XP line has defaulted to the New Technologies File System (NTFS), which is discussed later in the book. FAT is supported by all Windows and most Unix operating systems and will be encountered by investigators for years to come, even if it is not the default file system of desktop Windows systems. FAT is frequently found in compact flash cards for digital cameras and USB "thumb drives."Many people are familiar with the basic concepts of the FAT file system but may not be aware of data hiding locations, addressing issues, and its more subtle behaviors.
The layout of the data is slightly different in FAT12/ 16 and FAT32. In FAT12/ 16 the beginning of the data are reserved for the root directory, but in FAT32 the root directory can be anywhere in the data area (although it is rare for it to not be in the beginning of the data area). The dynamic size and location of the root directory allows FAT32 to adapt to bad sectors in the beginning of the data are and allows the directory to grow as large as it needs to. The FAT12/ 16 root directory has a fixed size that is given in the boot sector. The starting address for the FAT32 root directory is given in the boot sector, and the FAT structure is used to determine its size. Figure 1.1 shows how the various boot sector values are used to determine the layout of FAT12/16andFAT32 file systems.

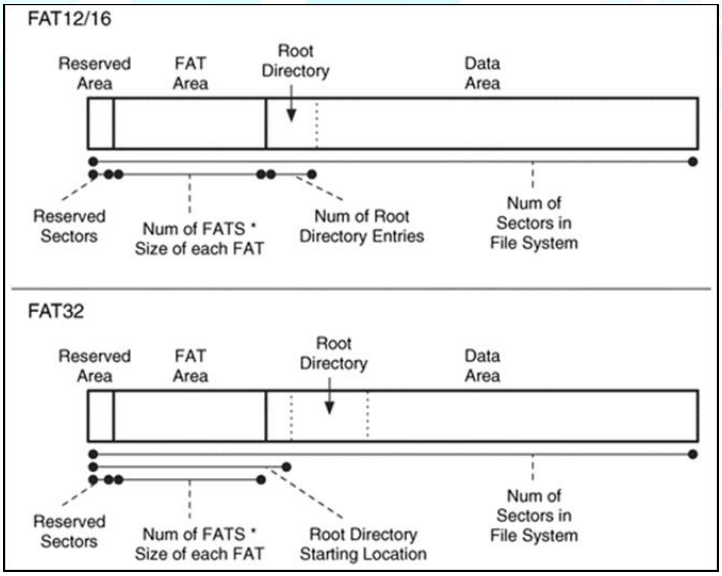


Figure 79 FAT file system layout and data from the boot sector that is used to calculate the locations and sizes

b) NTFS

NTFS is a proprietary file system developed by Microsoft for its modern Windows operating system; formatting a volume with NTFS results in the creation of several metadata files such as the master file table ($MFT), $Bitmap, $LogFile and others, which contain information about all the files and folders on the NTFS volume.

Figure 80 Sample formatted new technology file system volume

In the NTFS file system, each file stored within it is composed of a set of data streams: the primary stream is the one that holds the actual data a user sees when opening a file. The other stream is called the alternative data stream (ADS). Digital forensics examiners should search within data streams of all files stored on an
NTFS partition, as they can contain hidden data.

2. Introduction and importance of live forensics

Live forensics considers the value of the data that may be lost by powering down a system and collects it while the system is still running. The other objective of live forensics is to minimize impacts to the integrity of data while collecting evidence from the suspect system. Live Forensics is a methodology for extracting forensically sound evidence from "live" system. According to traditional forensics procedure, power plug is pulled to switch off the system when the system is in the running mode. But in live forensics, before pulling the cord we collect information such as details in memory, running process, network connection etc. In Live Forensic volatile data that may be lost by a power down is collected from a running system.

I. Why Live Forensics

Live forensics considers the value of the data that may be lost by powering down a system and collects it while the system is still running. It is a methodology, which advocates extracting "live" system data before pulling the cord to preserve memory, process, and network information that would be lost with traditional forensic approach

- Extract volatile forensic data that would be lost on power off
- Will have minor impacts to the underlying machine's operating state
- Often used in incident handling to determine if an event has occurred
- May or may not proceed a full traditional forensic analysis

Data on a system has an order of volatility. Live Forensics focuses on extracting and examination of the volatile forensic data that would be lost on power off. Data from the memory, swap space, network processes, and running systems processes is the most volatile and will be lost on system reboot. The goal live forensics is to extract and preserve the volatile data on a system while, to the extent possible, otherwise preserving the state of the system. When powering down a computer system, a lot of volatile information is lost. So, Live Forensics has become an important part of digital forensic investigations. This information will not be available for further analysis, although it may contain valuable clues regarding the incident, and there is sometimes no other way to obtain it other than collecting it on the running system.

a) Goal of the Live Forensic

The goal of any live forensics task should be to extract and preserve the volatile data on a system while, to the extent possible, otherwise preserving the state of the system. Additionally, this is often the first step of an incident response scenario where a handler is simply trying to determine if an event has occurred.
Nowadays hard disks are of terabytes size. So, the investigator needs a hard disk of terabytes size in order to acquire the image of source hard disk to continue with offline forensic analysis.

But in case of live forensics this is not needed since the investigator collects only volatile information from the system.

b) Live / Volatile Vs Non-volatile Data

Live / Volatile Data	Non-volatile Data
• System time • Logged-on user(s) • Open files • Network information • Network connections • Process information • Process to port mapping • Process memory • Network Status • Clipboard contents • Service/ driver information • Command history • Mapped drives • Shares	• Event logs, • security log of the system, • system information • Windows Registry • Installed software's, • network details • Internet history

c) Live Forensics before Pulling the Power Plug

Before pulling the Power Plug if the machine is in running mode at the scene of crime, then first live forensics must be performed. Otherwise all crucial information available in the suspect's system will be lost forever. And the information collection by live forensic is very crucial as it can provide useful hints in the offline forensics.

II. Acquisition of RAM Dump (Acquiring Volatile Memory)

Data is volatile when it will be lost when a device is switched off or rebooted. Volatile data may be overwritten due to normal use (e.g., when closing a particular application on a PC, the reserved data space will vanish from RAM memory, permitting other applications to utilize its space for activity). The process of capturing data from volatile memory is known as dumping.

Capturing and analysing volatile memory is more difficult than the traditional acquisition of hard drives because capturing a live memory requires specialized software tools. As well as analysing volatile data forensic image files needs specialized software, as RAM does not store data in the similar way as hard drives do.

Volatile memory can be found in other devices along with computers; for example, networking devices like routers and switches also have volatile data stored in their logs.
Following is the list of information which can be found in RAM:

- Cryptographic keys
- Processes running
- Executed console commands
- Clipboard contents
- Network information
- Decrypted contents
- Registry hives
- Text files and images
- Deleted files

- Web browsing logs
- Open/active registry keys
- Internet account passwords (e.g., e-mail, social media, and cloud storage)
- Instant messages
- Exploit-related information
- Malware (rootkits and Trojan horses)
- Evidence of activity not typically stored on the local hard disk

a) Step Action of Taking RAM Dump Using FTK Imager

FTK Imager is a Windows acquisition tool and it can be downloaded directly from Access Data web site free of cost. FTK Imager available in two types “FTK Imager” and “FTK Imager Lite”. Both the software has same features and functions. Only difference is lite version can be run from a pendrive or External Source, so Setup is not required for this version.

Step 1
Run FTK Imager, we will get the AccessDataFTK Imager window.

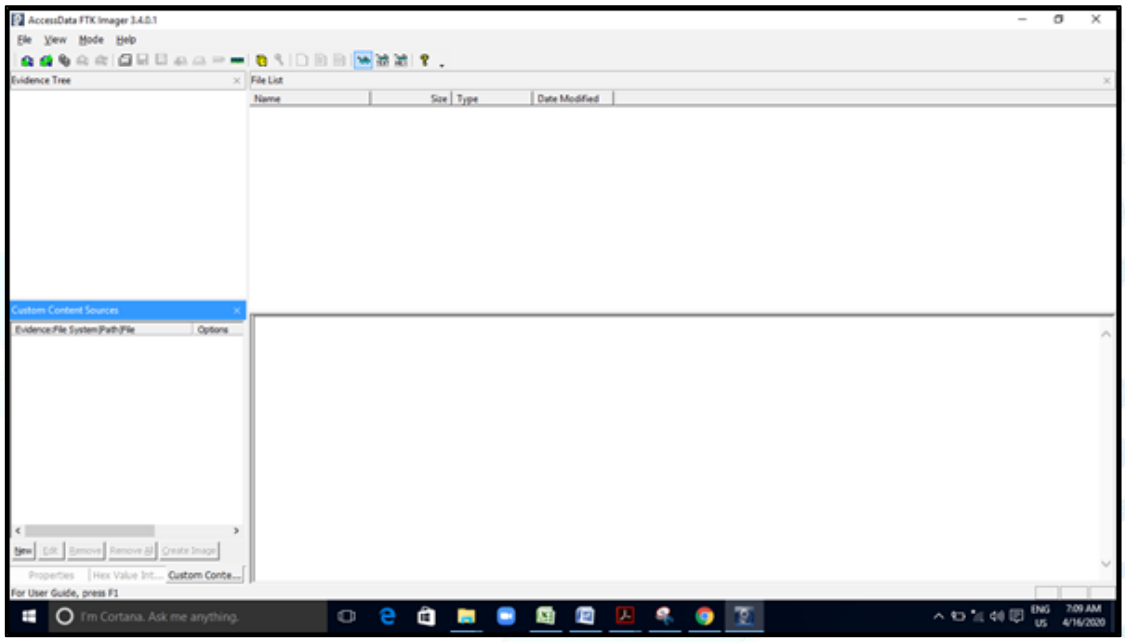


Figure 81 FTK Imager Home Screen

Step 2
To capture RAM Dump i.e. Volatile Memory, go to file menu and click on Capture Memory.

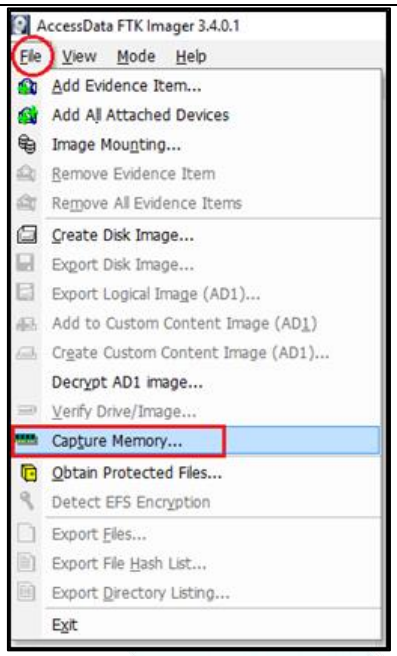


Figure 82 Capture Memory Option

Step 3

We will get memory capture window:

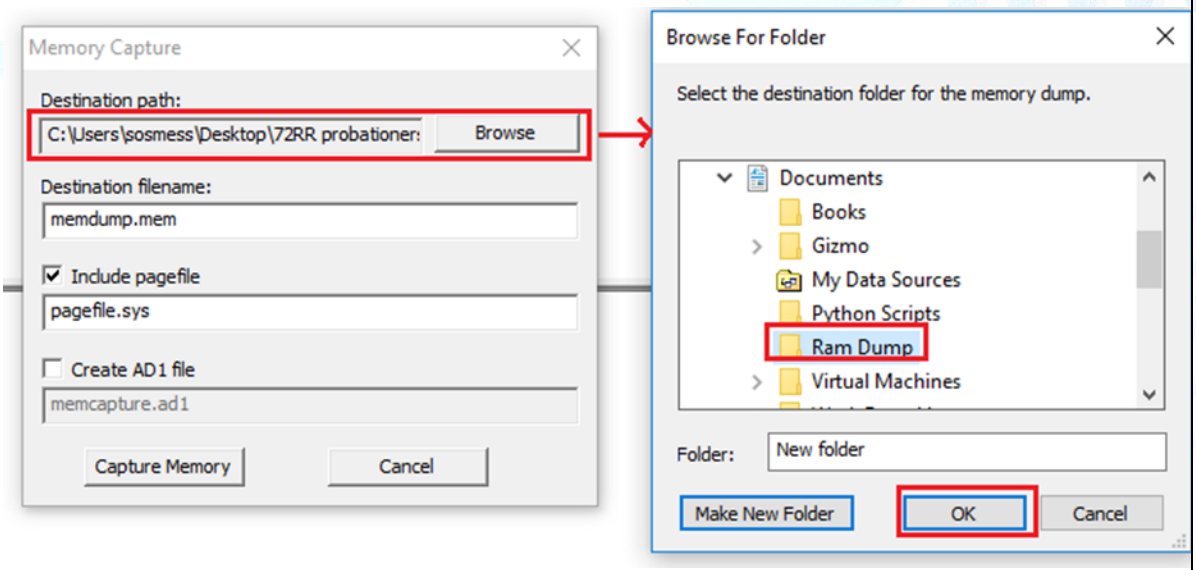


Figure 83 Capture Memory WIndow

Click on Browse Button to choose the location, where RAM Dump will be saved
Note: Always choose External Storage Media to Store any Evidence File like External Hard Disk. Never store RAM dump in suspects computer.
Enter the Memory Dump file name by default file name will be memdump.mem. We can change it as per case requirement.
If we want to take backup of Page file enable check box Include Page file.
Click on Capture Memory to start the process

Step 4

If we observe this progress window, we found total memory installed in the system. Here total memory we can see is 5GB.

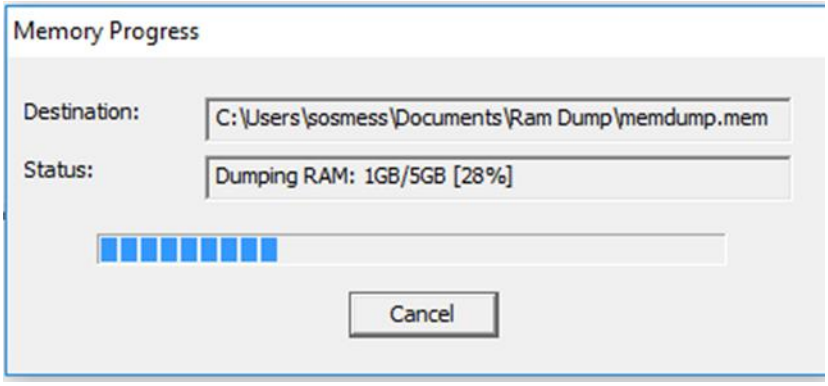


Figure 84 Memory Capture in Progress

When memory capture finished successfully click on close button.

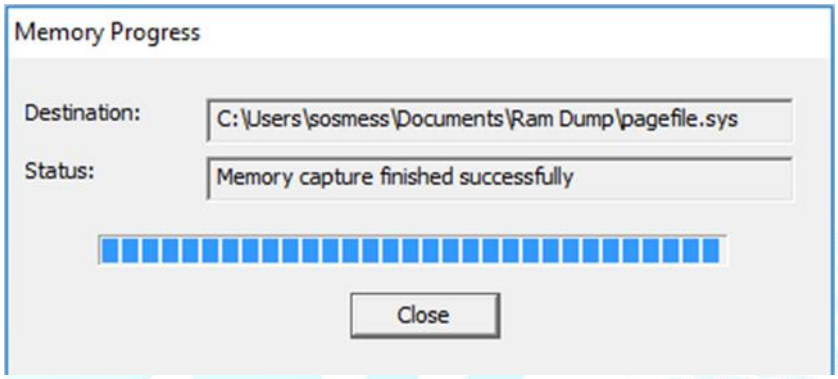


Figure 85 Memory Capture Finished

Go to the location where memdump.mem file is saved.

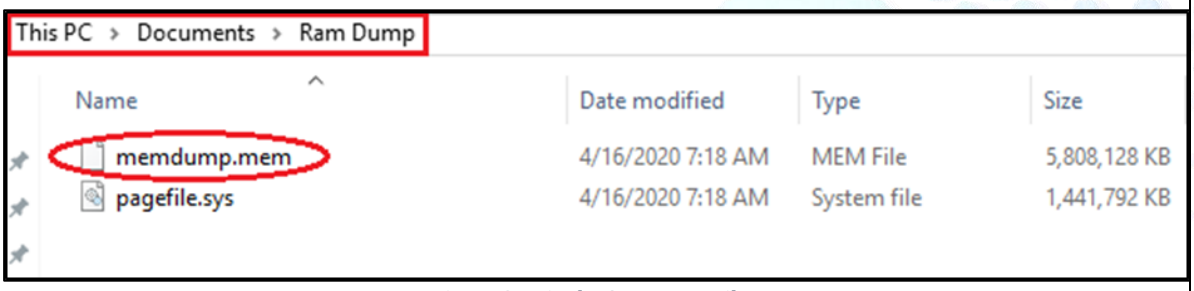


Figure 86 Final RAM Dump File

b) To collect Windows Protected Files from Live System

Step 1
To capture Windows Protected files, go to obtain protected files in file menu.

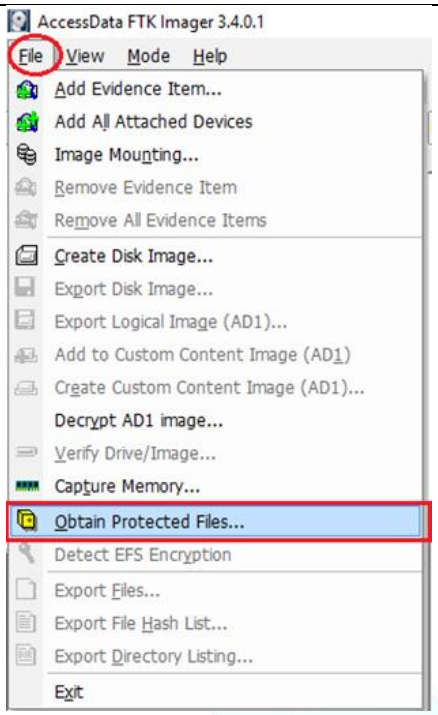


Figure 87 Obtain Protected Files

Step 2

Obtain System Files window will appear:

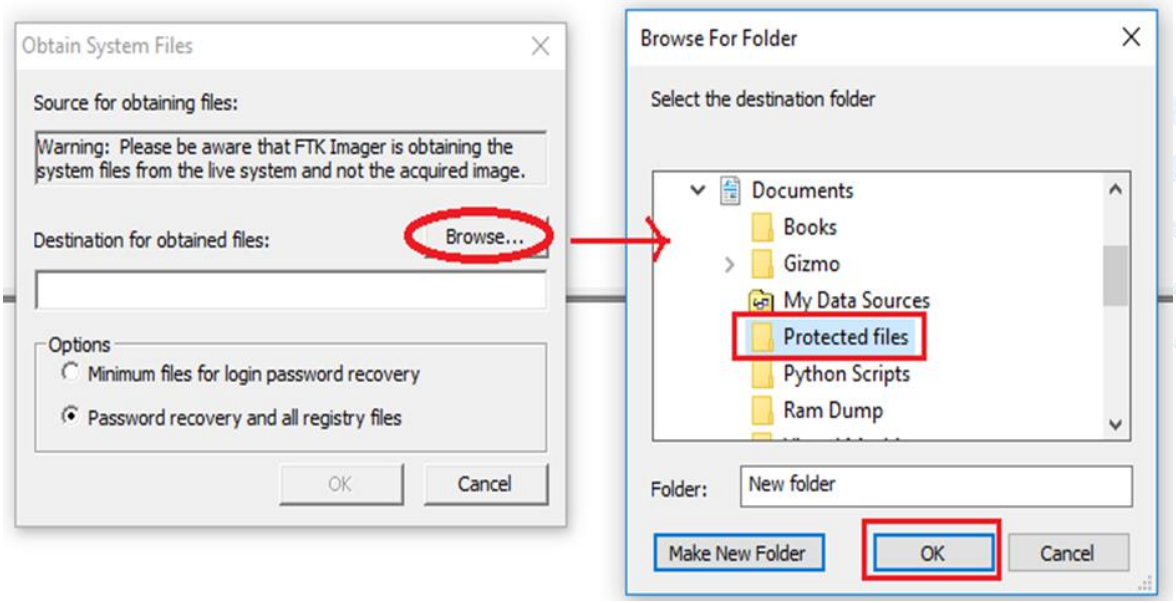


Figure 88 Obtain System Files Window

- Click on Browse Button to choose the location, where these files will be saved.
- Note: Before doing this create a folder "Windows Protected files" on External Hard Disk and choose this folder to save the Evidence file.
- Click on Password recovery and all registry files option and press OK.

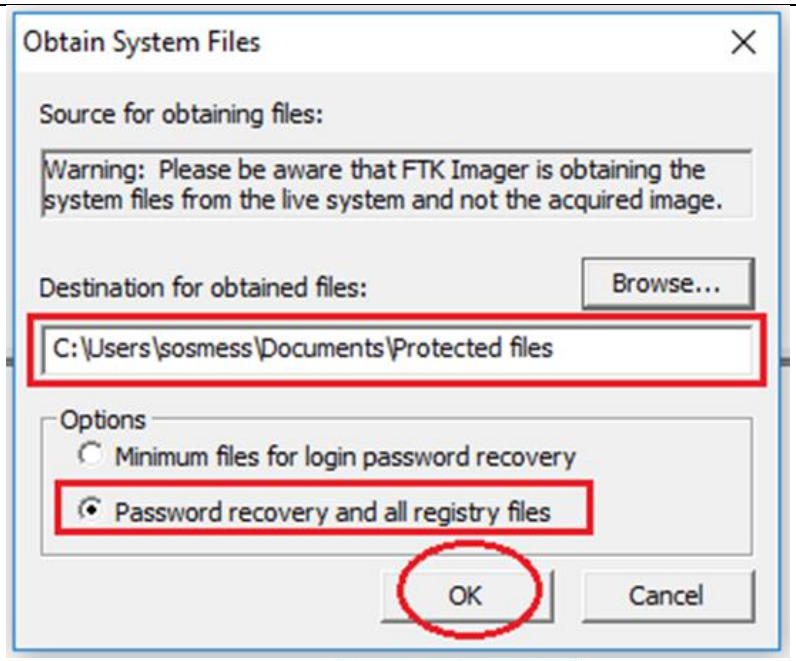


Figure 89 Click OK

Step 3

Export Files progress windows will appear.

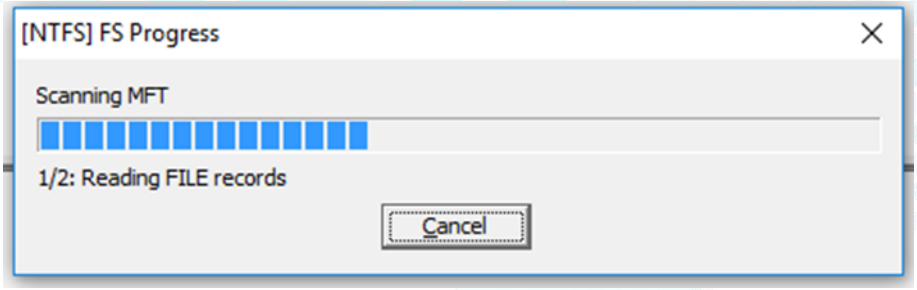


Figure 90 File Export in Progress

Step 4

After completion of Process, go to the location where these files are saved.

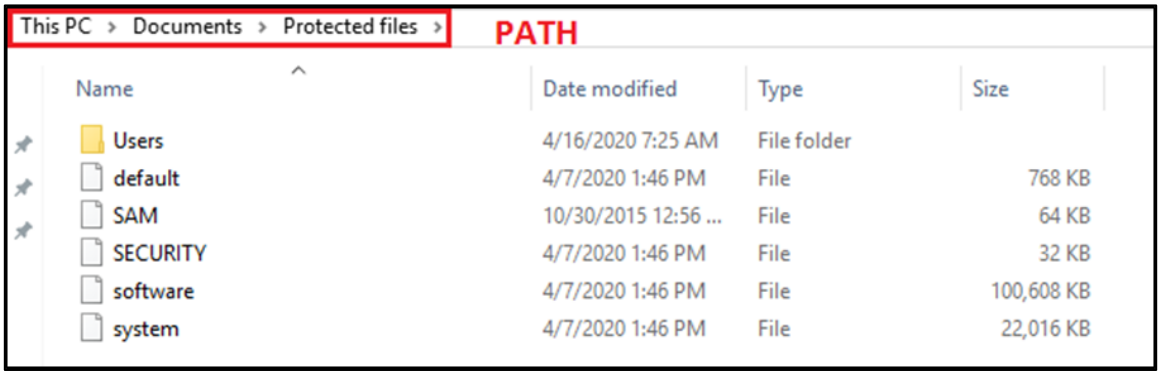


Figure 91 Files Exported at Destination Folder

c) Challenges faced during RAM Dump Acquisition

Acquiring volatile memory might face a couple of challenges by the forensic examiners. The following are some scenarios.

i) **Windows Is Locked**
We might come across a running computer with a login screen (locked computer). It is advisable to perform a hard shutdown. However, we can bypass the Windows login page with no reboot by using some tools/techniques to avoid losing RAM contents:
- Put computer in Hibernation mode, it will allow hibernate file to store in Disk.

ii) **Administrative Privileges**
Most of the RAM Dump capturing software tools needs administrative privileges in order to work. If target PC is running with limited user permission (e.g., user account) it would be challenging to capture RAM Dump.

iii) **Capturing Tool Footprint**
The RAM Dump capturing tool will also leave some traces on the target machine. This means some data may be overwritten as a result of acquiring live memory. Traces of capture tool can be justified, if documented properly.

d) Importance of Pulling the Plug

A live system will have numerous background processes running which are continually reading and writing data to and from hard drive. When forensic examiner decides to shutdown the system, pulling the plug might be best suitable option. Because normal shutdown will start a chain of actions, that involve writing a lot of data to the hard drive, possibly overwriting important evidence.

Removing power plug will "freeze" the system in its current state, making sure that the data on the system is no longer modified. But before pulling the plug one must ensure to capture the RAM since all the data stored in RAM will be lost once we pull out the plug.

III. Analysis of RAM Dump

The most important principle to remember for a forensic examiner is **"Whatever works is always in the memory and whatever happens is always in the memory".** Hence it is very necessary to capture and analyse memory dumps. It gives an examiner clearer picture about the current state of the system. There are many tools available in market to perform RAM Dump analysis which extracts certain type of information from the dump. Here we are going to explain use of '**Bulk Extractor'** for RAM Dump analysis.

a) Analysis RAM dump using Bulk Extractor

Bulk extractor is a computer forensics tool that scans a disk image, a file or a directory of files and extracts useful information without parsing the file system or file system structures. The results can be easily inspected, parsed or processed with automated tools.

Bulk Extractor also creates a histogram of features that it finds, as features that are more common tend to be more important. The program can be used for law enforcement, defense, intelligence and cyber-investigation applications.

Bulk Extractor now creates an output directory that includes:

- **ccn.txt** -- Credit card numbers
- **ccn_track2.txt** -- Credit card "track 2″ information
- **domain.txt** – Internet domains found on the drive, including dotted-quad addresses found in text.
- **email.txt**–Email addresses

- **ether.txt** – Ethernet MAC address found through IP packet carving of swap files and compressed system hibernation files and file fragments.
- **exitf.txt** – EXIFs from JPEGs and video segments. This feature file contains all of the EXIF fields, expanded as XML records.
- **find.txt** – The results of specific regular expression search requests.
- **ip.txt** – IP addresses found through IP packet carving.
- **telephone.txt** – US and international telephone numbers.
- **url.txt** – URLs, typically found in browser cache, email messages and pre-compiled into executables.
- **url_searches.txt** – A histogram of terms used in Internet searches from services such as Google, Bing, Yahoo and others.
- **wordlist.txt** – A list of all "words" extracted from the disk, useful for password cracking.
- **wordlist_*.txt** – The wordlist with duplicates removed, formatted in a form that can be easily imported into a popular password-cracking program.
- **zip.txt** – A file containing information regarding every ZIP file component found on the media. This is exceptionally useful as ZIP files contain internal structure and ZIP is increasingly the compound file format of choice for a variety of products such as Microsoft Office.

For each of the above, two additional files may be created:

- ***_stopped.txt** – Bulk Extractor supports a stop list, or a list of items that do not need to be brought to the user's attention. However rather than simply suppressing this information, which might cause something critical to be hidden, stopped entries are stored in the stopped files.
- ***_histogram.txt** – Bulk Extractor can also create histogram of feature. This is important, as experience has shown that email addresses, domain names and other information that appear more frequently on a hard drive or in a cell phone's memory can be used to rapidly create a pattern of life report.

Step 1

After Launching Bulk Extractor, Following window will be displayed.

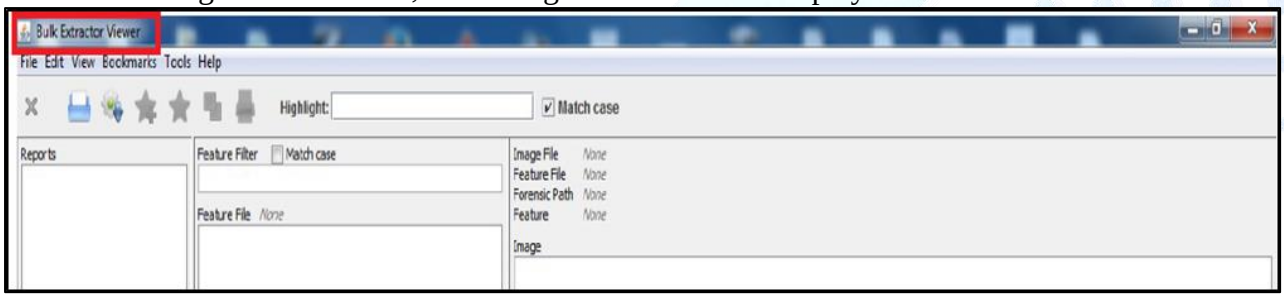


Figure 92 Bulk Extractor Home Screen

Step 2

Go to **Tools** option, select '**Run Bulk Extractor'**

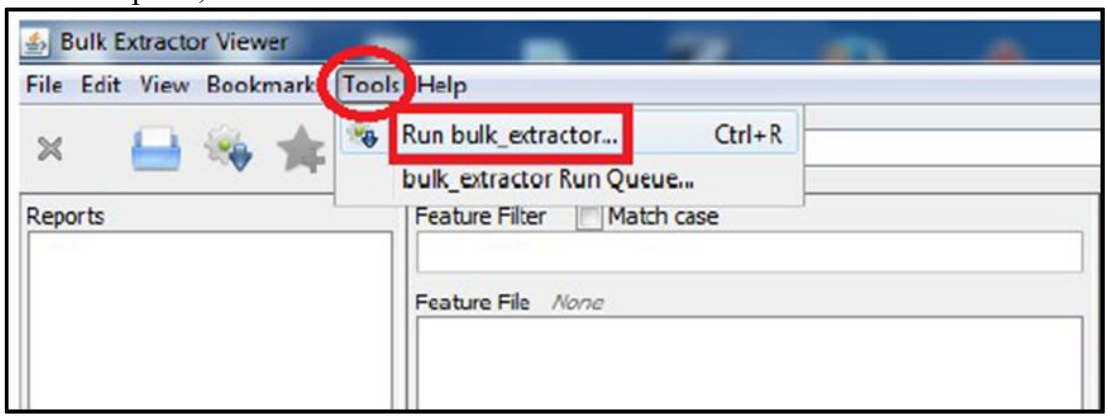


Figure 93 Tools - Run Bulk Extractor

Following Window will be displayed, select 'Image File' option and browse target image file as shown below.

After that, select target folder where you want to save the final analysis report.

Finally click on '**Submit Run**' button to start the process

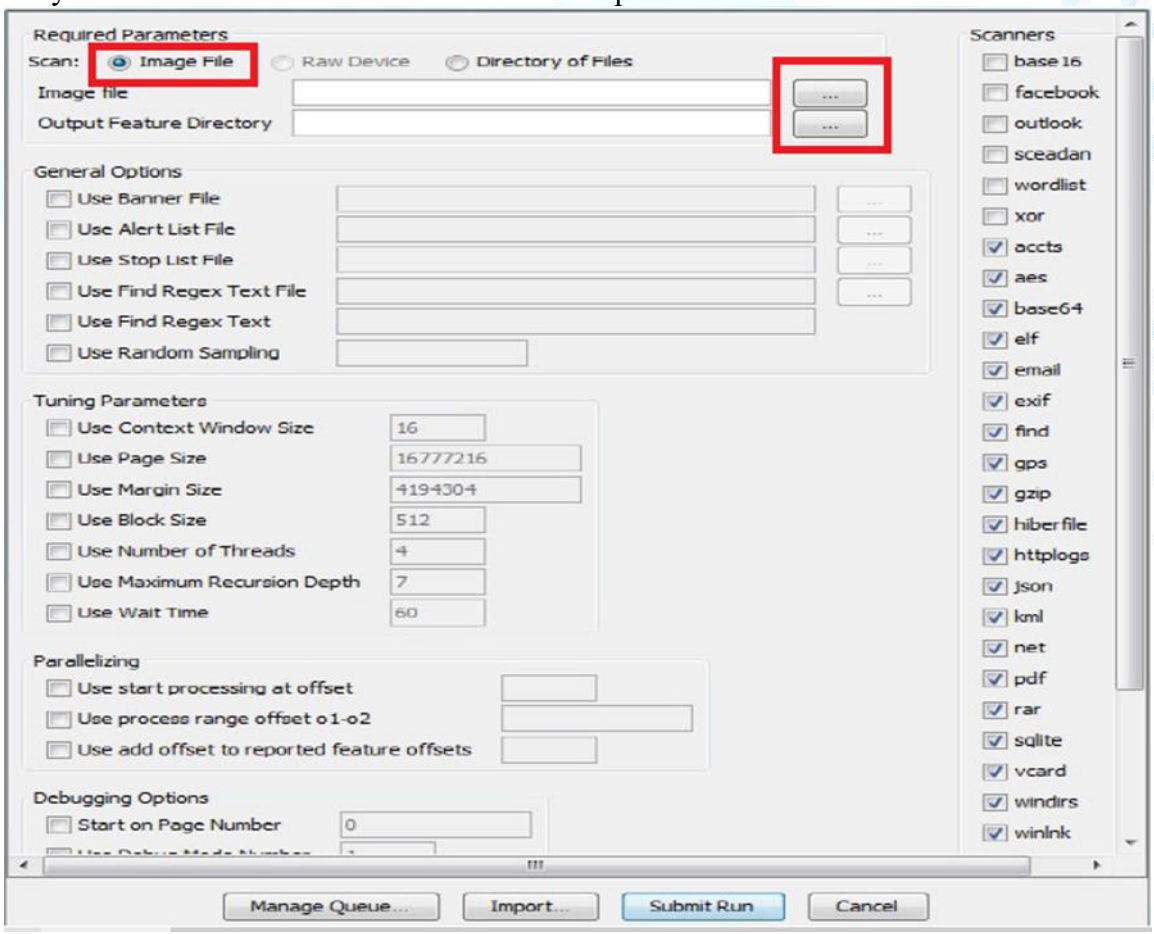


Figure 94 Select Image File for RAM Dump Analysis

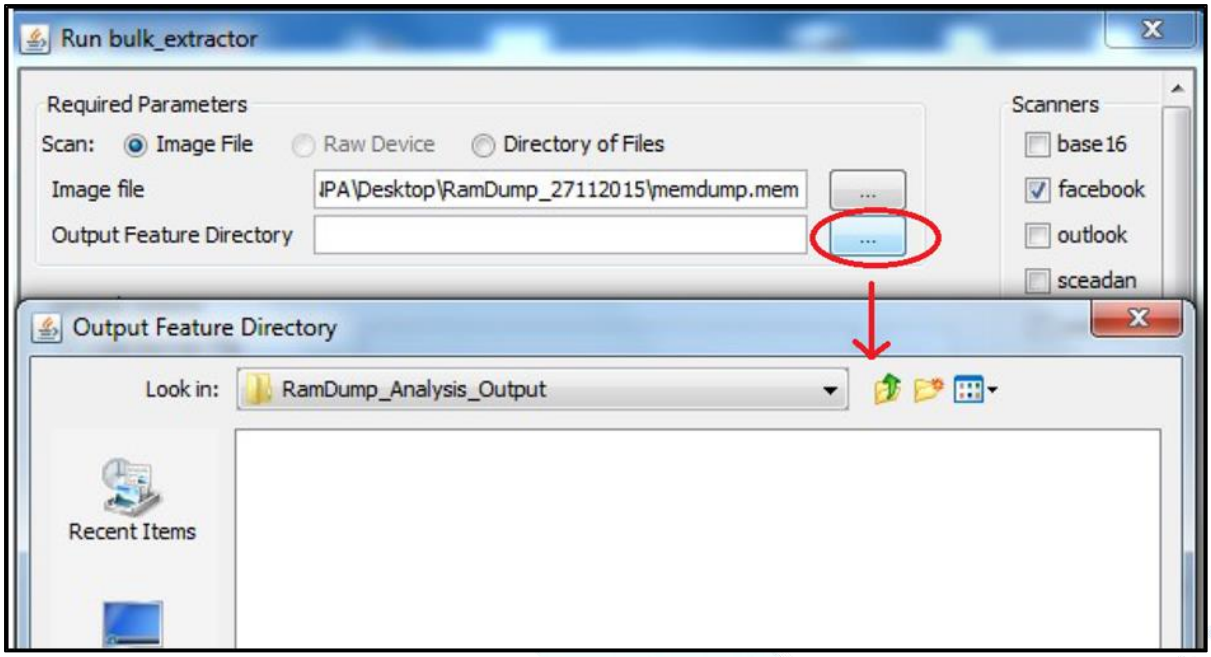


Figure 95 Selecting Destination Folder

Step 3

This window shows that file extraction process is in progress.

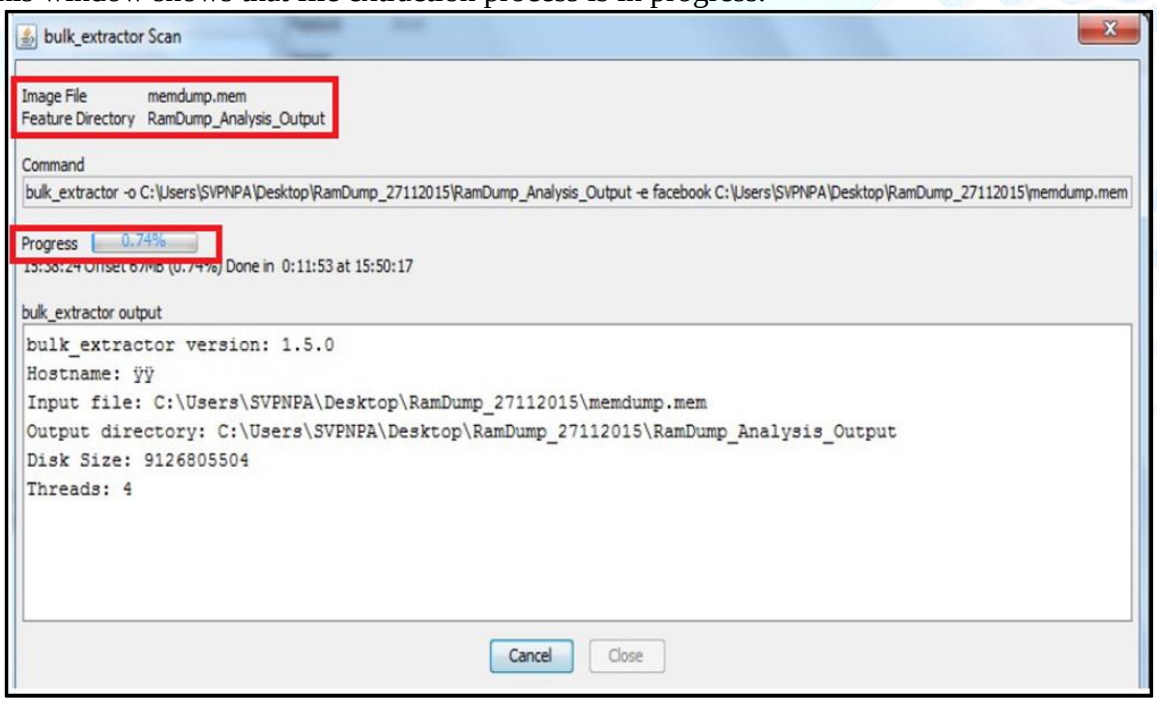


Figure 96 Files Extraction in Progress

Step 4

When files extraction finishes follwing window appears with all extracted files listed in the left panel.

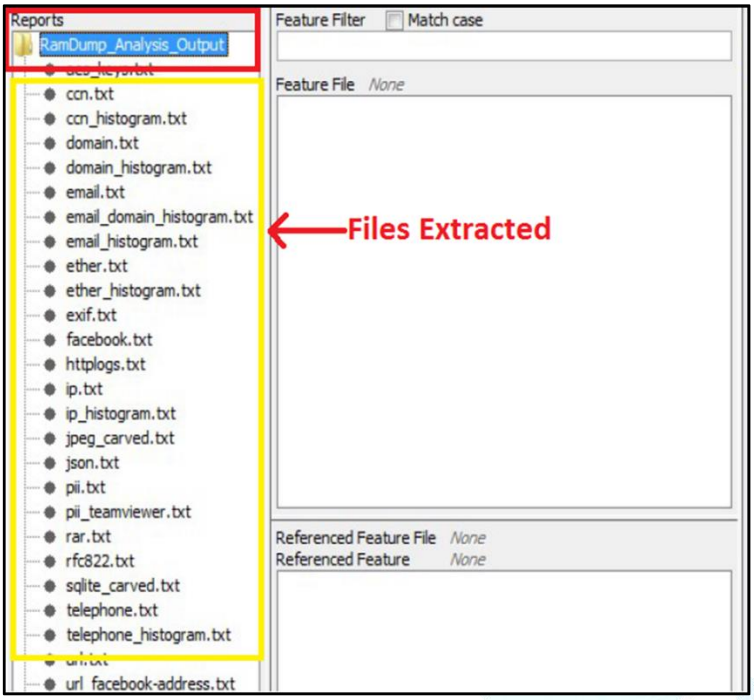


Figure 97 Extracted Report - List of Files Present in Report

Step 5

Analysing credit card details from ccn.txt file. All content of the file will be displayed in middle upper block.

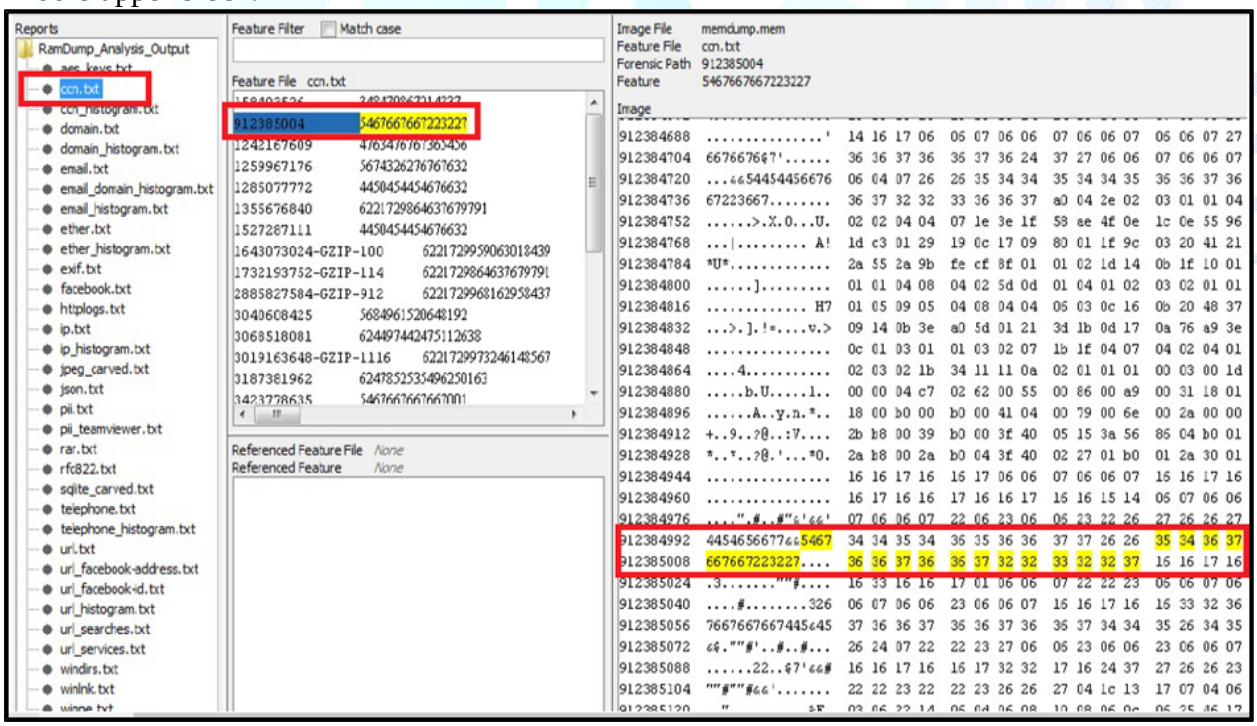


Figure 98 ccn.txt - Credit Card Details

Step 6

domain.txt file enlist all the domains access by target PC.

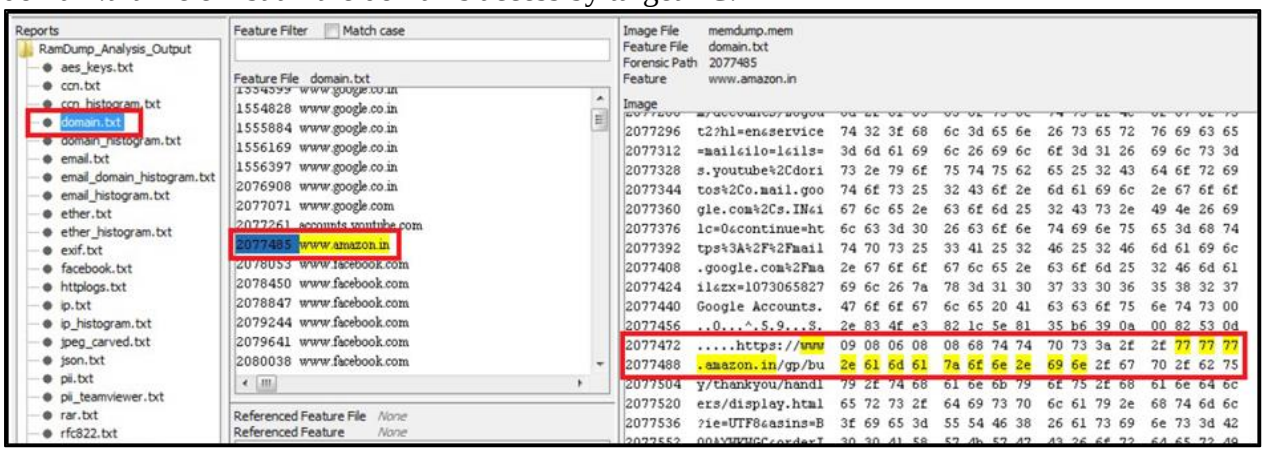


Figure 99 domain.txt

Step 7

All email addresses access by target PC will be extracted in email.txt file.

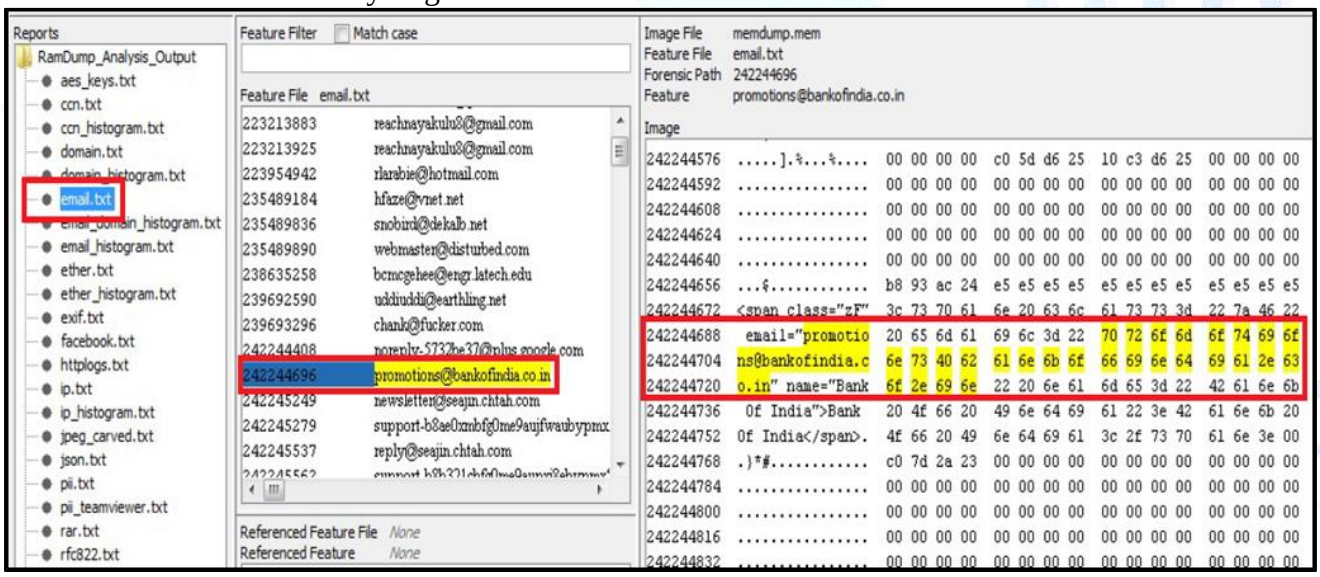


Figure 100 email.txt - email addresses

Step 8

MAC address of the PC will be captured in ether_hostgram.txt file.

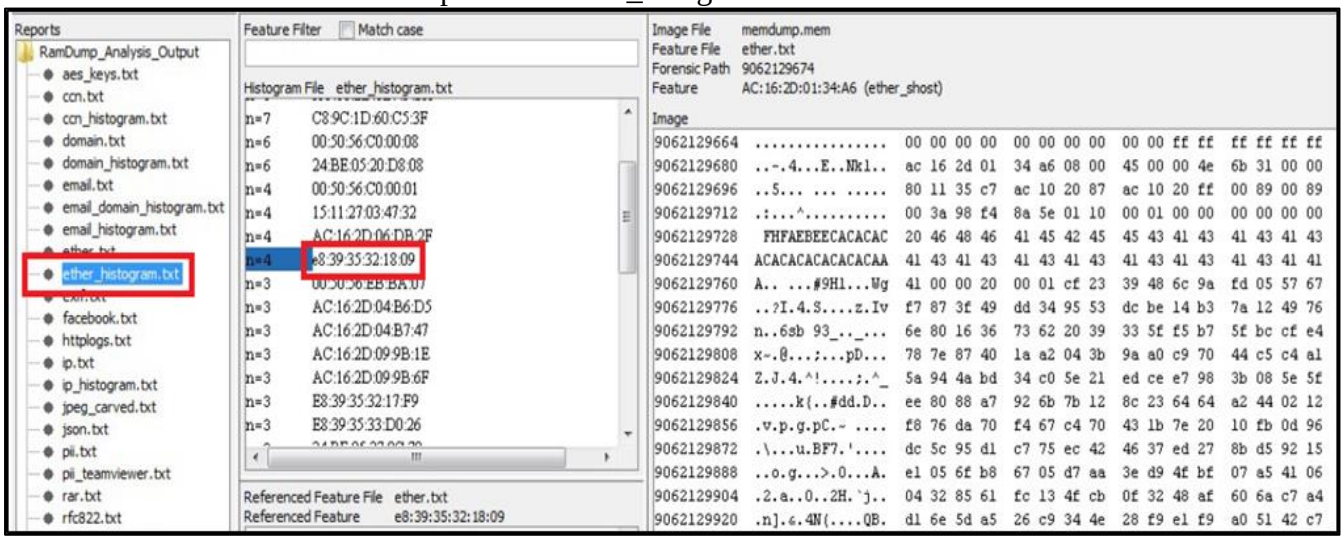


Figure 101 MAC address

Step 9

telephone.txt file will contain all the telephone numbers.

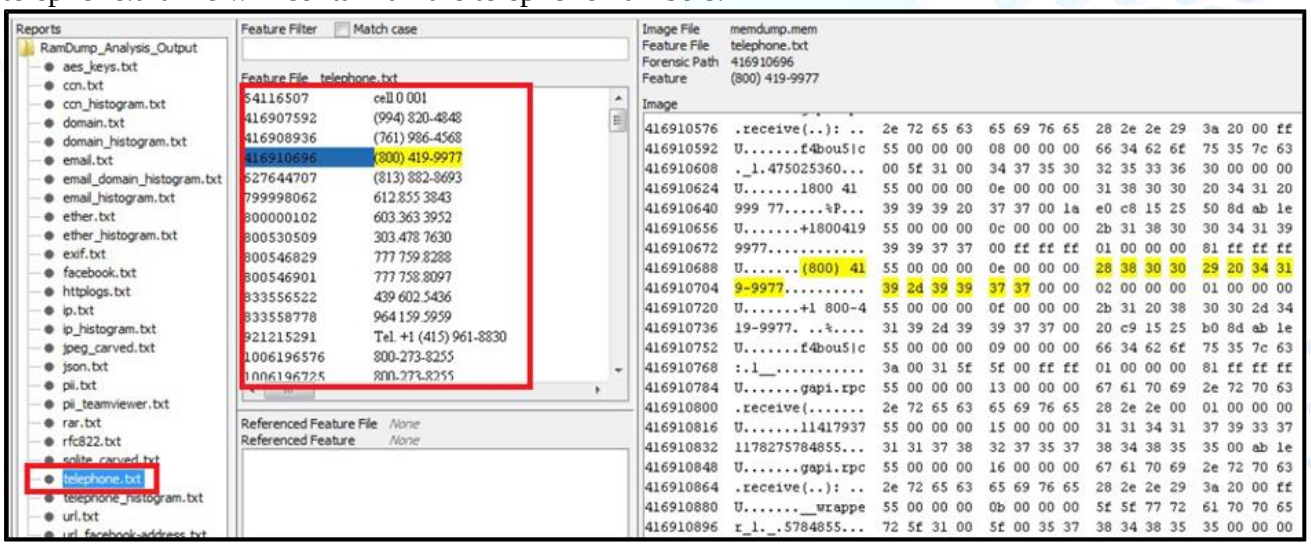


Figure 102 Telephone Numbers

Step 10

All the facebook address accessed by target PC will be listed in url_facebook-address.txt file.

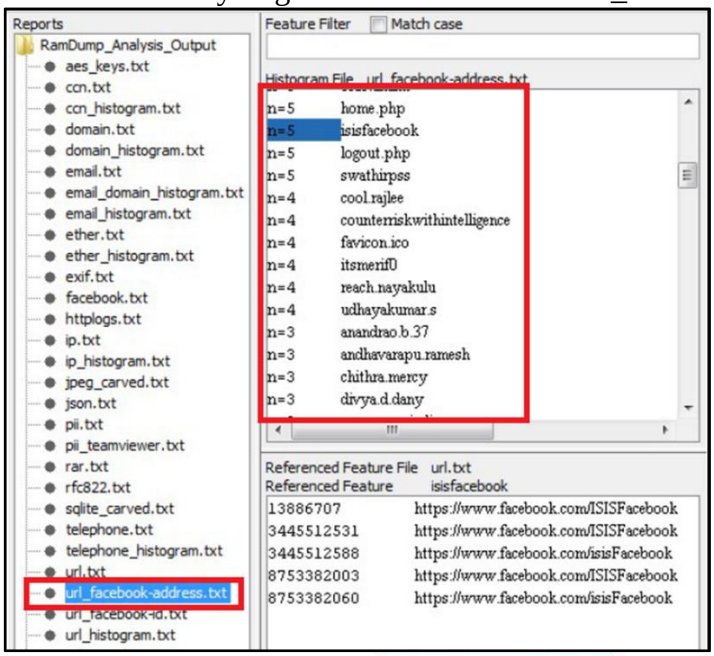


Figure 103 Facebook Addresses

Step 11

Search history of the target PC will be saved in url_searches.txt file

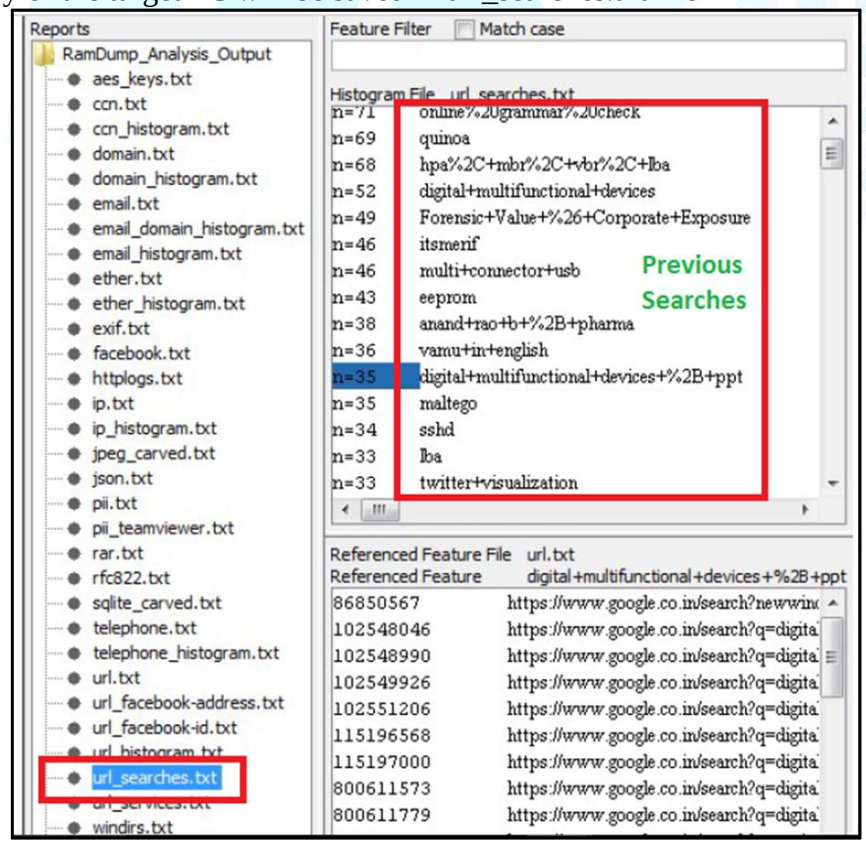


Figure 104 Search History

IV. Hiberfile, Swapfile and Pagefile

Swapfile.sys, Hiberfil.sys, and Pagefile.sys are three important system files required for the proper functioning of the Windows OS. The default location of these files is in system drive (usually the C:\ drive). All three files will be always hidden in the system.

a) How to access Hiberfile, Swapfile and Pagefile in Windows

Step 1

Open ‘C drive’ of the system. Here we see all system folders and files but not hidden and protected files.

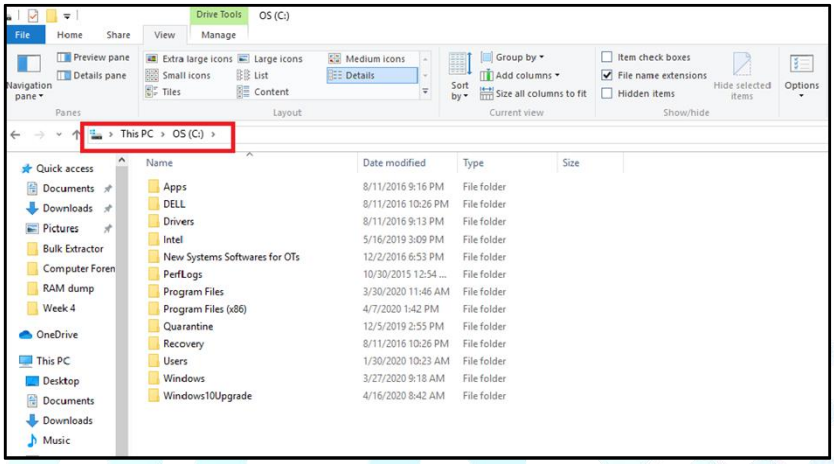


Figure 105 Open C drive

Step 2

Next go to view option as shown in figure below, and follow the procedure illustrated in diagram below to view hidden files of the system.

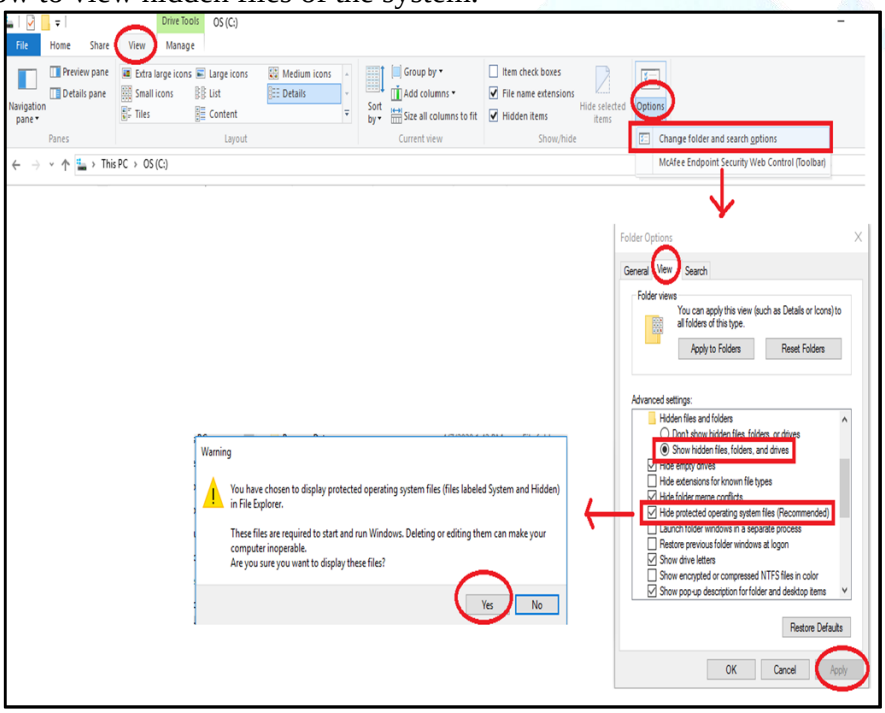


Figure 106 show Hidden Files

Step 3

In below figure we can see that now hidden system files are visible in c drive.

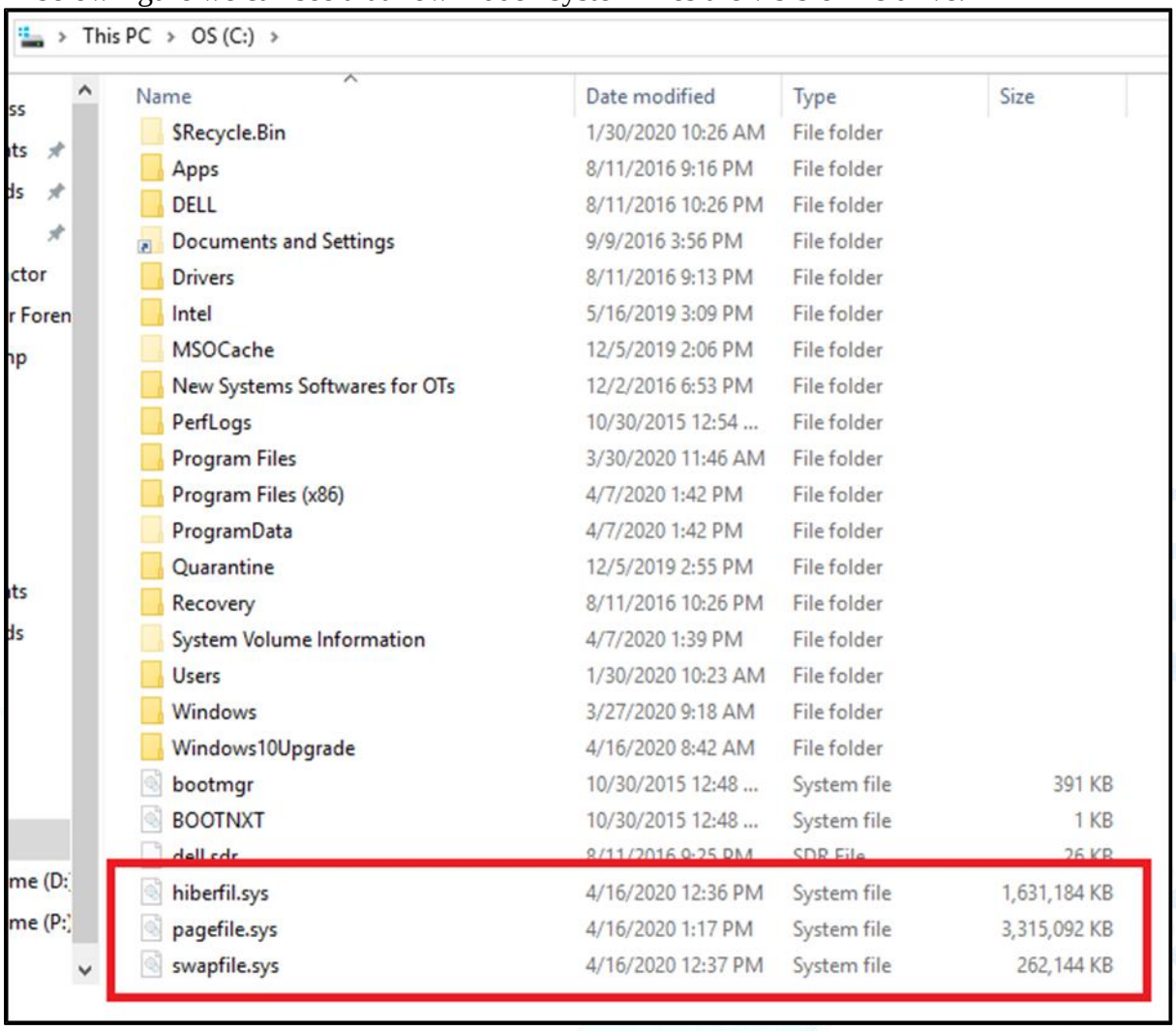


Figure 107 Hiberfile, Pagefile, Swapfile

a) Hiberfil.sys

This is a file used by Windows to enable the hibernation feature; the approximate size of this file is about 3/4th of system RAM. The hibernation file in earlier versions of Windows (e.g., 7 and Vista) stored kernel session, device drivers, and application data while in modern versions of Windows (like 8 and 10) it stores only the kernel session and device drivers, making it notably less in size.

hiberfil.sys can store a pile of important information about the running machine. Following tool can investigate the hiberfil.sys file:

- **Volatility**, free and open source tool: *www.volatilityfoundation.org*

b) Pagefile.sys

Forensic examiners should not ignore the importance of virtual memory, as this file can hold important information shifted from RAM. For example, fragments of decrypted files can still reside there, and encryption keys or passwords (or a fragment of it) can also be found here. The pagefile.sys is a hidden system file, it resides by default at %SystemDrive%\pagefile.sys; however, a user can change its default location.

Nowadays capacity of the physical memory is increasing with the continual advance of computing power (for example, it is common these days to buy a laptop with 16 GB of RAM memory). This effectively limits the need to swap any files to the virtual memory, which results in low expectations of computer forensic investigators when investigating pagefile.sys.

c) Swapfil.sys

Swapfile is used to store the idle and other non-active objects transferred from the RAM, whenever a user tries to access an idle process again, its information will be shifted to the RAM. In modern Windows versions (like 8 and 10) we can see that both Pagefile and Swapfile exist together on a system drive; we can consider that these two files form together what is known now as virtual memory in Windows OS. Swapfile has a fixed size in modern Windows versions (8, 10), which is 256 MB.

3. Windows Forensics

I. Importance of Windows forensics

Microsoft Windows is a graphical user interface (GUI) operating system that has been distributed in various forms since 1985. Overtime, windows became the most common operating system that was installed on a computer system. Each version has brought changes to the user interface, though not all have been popular. Each version has had different ways of storing data that is of forensics value to the examiner.

An artefact refers to anything man-made – a Windows artefact, for the purpose of computer forensics, is evidential data that is automatically saved by the windows operating system as a result of a person interacting with or using the computer. It does not refer to the default files that saved to the computer on install – most of which may have no bearing on an investigation.

II. Artifacts in windows PC

a) Shell Link Files

A “shell link file” is more 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, folders, and printers, They provide a powerful and convenient way for users to gain quick access to frequently used programs and files, They are most often implemented via icons on the desktop or items presented from a menu such as Windows Start Menu.

During an examination of a Windows system many Link files (lnk) will be found. These files contain some very useful information about the target file including:

- File MAC Times.
- File Size.
- Volume Details - Serial Number, Label.
- Original File Path.

The information within a link file can vary depending on:

- Version of Windows was in use.
- If the link file was created by an application / user / 0S.
- File system of target file.

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. The file structure is quite complex, but well documented online. Tool useful for parsing Link files are:
https://4discovery.com/our-tools/link-parser/

It is important to understand that the times stored within a Link file relate to the actual target file - and they are stored in FILETIME format and are UTC.

A link file's embedded time becomes very powerful when the examiner can cross check the MAC times of the target within the file system to those within the link file ~ any file system date and time entries that are after those embedded within the link file show that a user has interacted with the file.

There are many forensic 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. The MAC address can be used to identify a single computer. Even if MAC spoofing is used, the original MAC will still be inside the Link File.

By default, when a file or document is opened in either by double clicking, or using a programs File > Open dialog, a link (ink) file is created in the Recent folder. Table below provides the location in Windows XP, Vista,7,8 and 10 operating systems.

Location found	Windows version
C:\Documents and Settings\ %Username%\Recent	XP
C:\Users\%Username%\Appdata\Roaming\Microsoft\Windows\ Recent	Vista, 7, 8, 10

Table 10 Link File Location

In a Windows 10 system, the files are displayed in the Recommended item on the Start Menu,

Figure below shows the recently accessed files. If clicked on more the actual recent folder will contain a link file for every user file that has been recently opened.

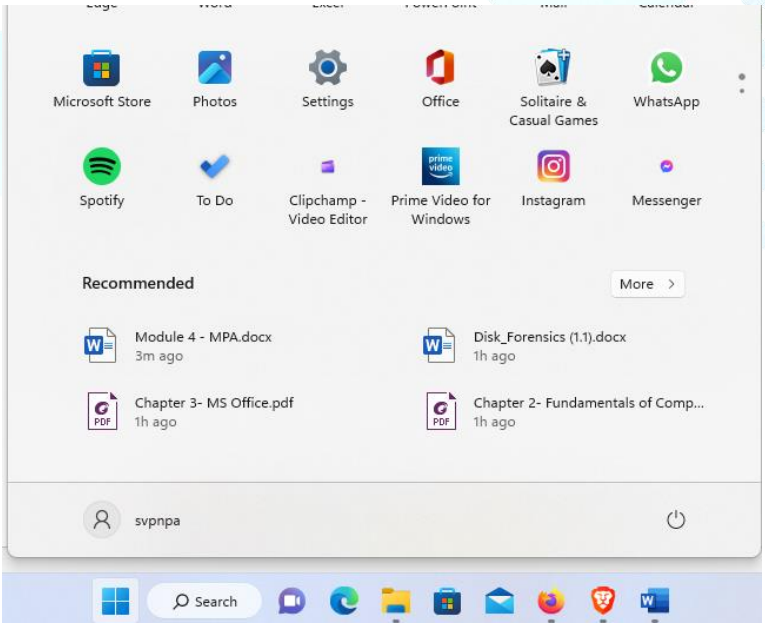


Figure 108 Recent items as seen in Windows 10

In Windows 10, the Explorer also gives users the option to exclude recent files from view in “quick Access". This option turns off the view within Windows Explorer Quick access, but the Operating System still collects the Link files in the recent folder.

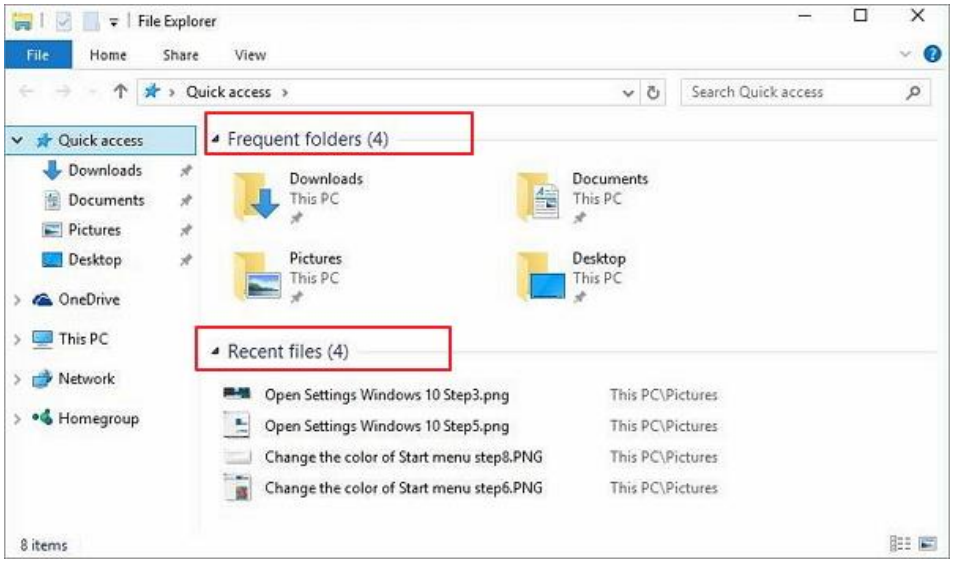


Figure 109 Windows 10 default Quick access view with recent files and frequent folders

The standard folder options window in the General tab allows users to exclude files and frequent folders independently. It also gives users a very quick way to clear all link files from the Recent Items folder as seen in Figure below.

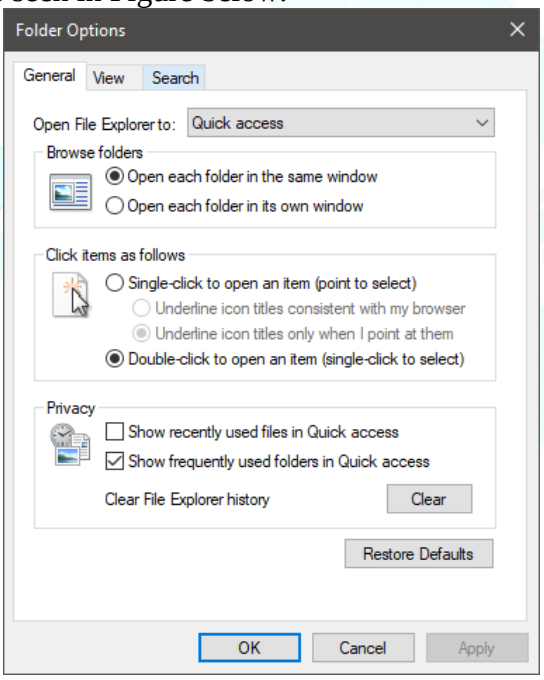


Figure 110 Windows Explorer Folder Option

b) Jump Lists

Windows 7 introduced a new feature called “Jump Lists”, which are essentially a list of recent files that have been opened (or attempted to open) by a particular application. It is like the “Recent” folder, except that each list only applies to one program.

This artefact often provides significant insight to user activity and be especially beneficial if Link files in the Recent folder have been deleted, or even if the application has been deleted. It is possible the most under-utilized and yet most valuable artefact to the forensic investigator, as it proves user knowledge of a file through their interaction with it.

Observe in Figure, the start menu on the left shows that there are two documents that are in the Recent history for Microsoft Word 2013. However, the same start menu shown on the right shows that Recent Items is empty, and indeed, the Recent folder displayed at the

bottom of Figure is indeed empty. This is because the shortcuts for an individual program are stored in Jump List files, rather than shortcuts in the Recent folder.

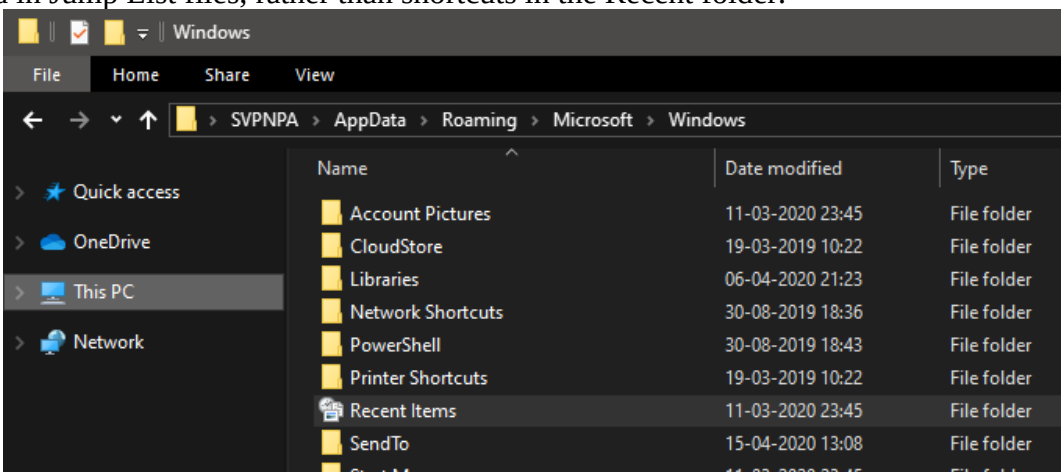


Figure 111 Windows Jump list and Recent Items

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 List data and manually deletes them, which is not easy as they are "Super Hidden".

'There are two main types of Jump Lists:

1. Automatic - this Jump List is automatically populated by the system. It records information about file usage and stores that information in destination file associated to the program used to open the file.
2. Custom - this Jump List is maintained by the individual application and can provide a list of tasks specific to the program menu along with custom defined categories.

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

%User Profile%\ AppData\ Roaming Microsoft\Windows\Recent

When this folder is viewed in Windows Explorer, nothing unusual is noted, as shown in Figure. However, 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.

Figure below shows the recent folder with the folders AutomaticDestinations and CustomDestinations visible.

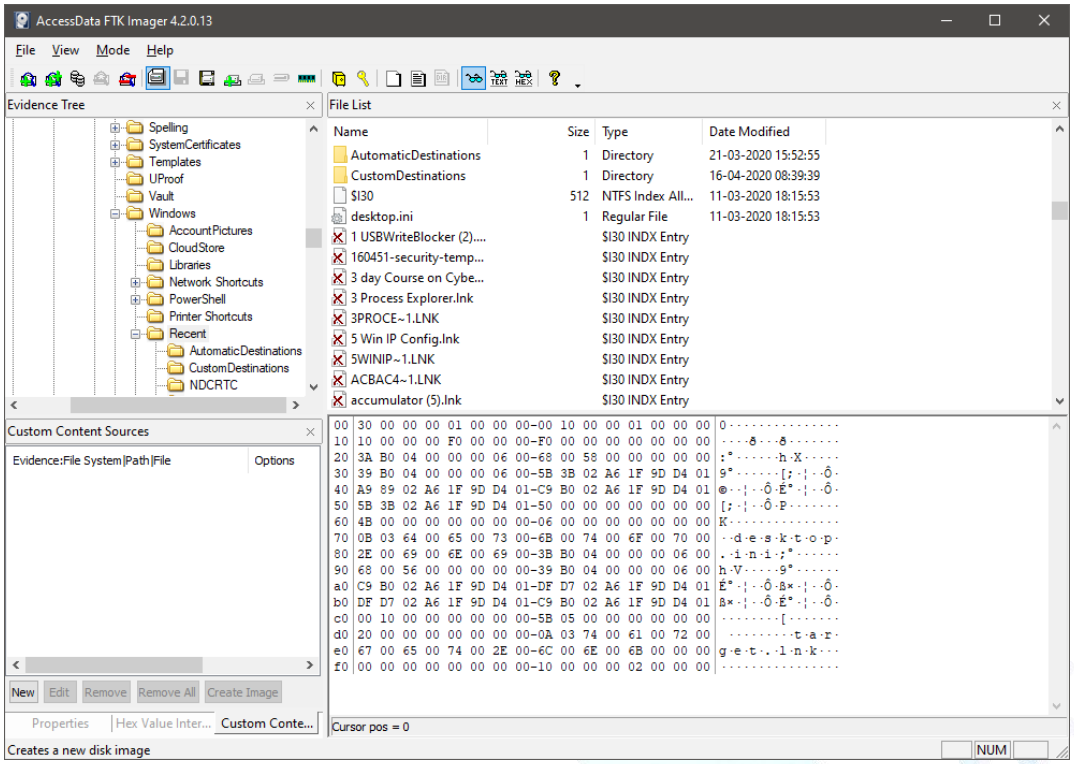


Figure 112 Jump list folder

The recent item data from the Jump Lists populate these two folders. Each program will have its own file name, referred to by its "Jump List ID" and may have both an AutomaticDestinations-ms and CustomDestinations-ms file. By examining each file with a text editor, it can be determined which file links to which program's Jump List entry.
More jump list IDs can be found at http://www.forensicswikiorg/wiki/List of Jump List IDs. This is not an exhaustive list, and a program's Jump List Id will often change each (program) version update.

A Jump List is an Object Linking and Embedding (OLE) compound file. It is a container that defines blocks that are assigned to a stream using multiple allocation tables, and is not unlike the FAT file system. There are two main parts to a Jump List - the Destination List (Dest List) and the Link Files themselves. 'The Dest List is a brief listing of all items in the Jump List, their path, date and time and the entry number for each item. This view is useful to try and find a specific item. There will also be a more detailed entry for each file in the Jump List, which contains the same information as a Link File.

Most major forensic tools will parse Jump List content; however, a useful free tool is available for examining automatic custom lists, called Jump Lister: http://www.woanware.co.uk/forensics/jumplister, Figure 5. 14 shows Jump Lister displaying the Destination List.

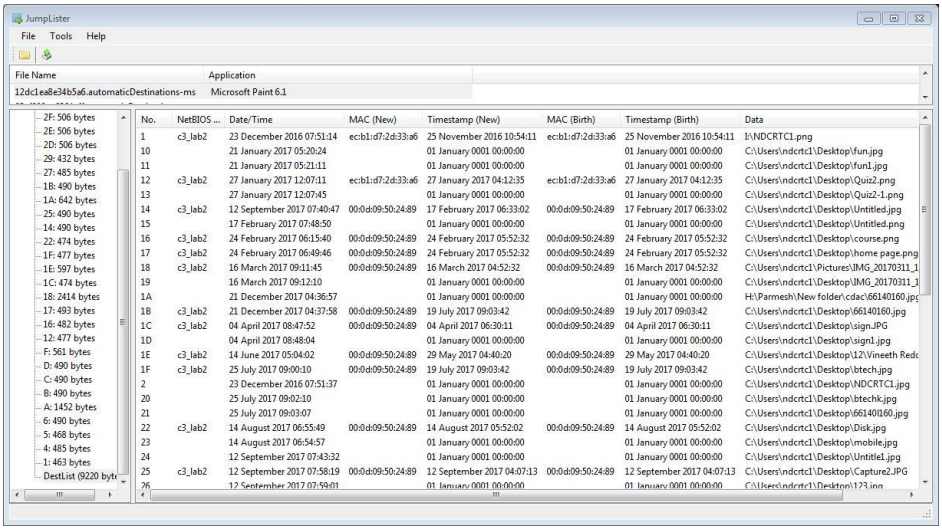


Figure 113 Jump list content

Within each of the data streams is a Link file, and as such, the same information can be found including: The Created and Modified Timestamps, the Serial Number of the volume the item was located on, the type of volume and the full path. This information is extremely useful to tie a piece of removable media that contains evidence not only to the exhibit computer, but also to a specific user account. Figure below shows the details stored in Jump List entry 13 when viewed by Jump Lister.

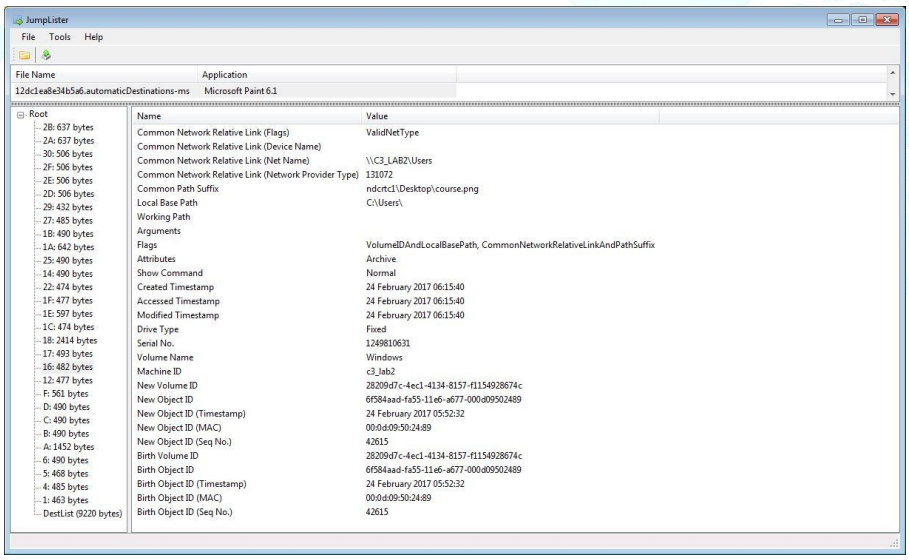


Figure 114 Jump list entry content

Windows 7

By default, Jump Lists are enabled in Windows 7, and will display up to the last ten files opened by an application. It can be disabled by the user in the Task Bar and Start Menu Properties, as shown in Figure. Although it only shows the last ten items, many more are recorded in both the Recent Folder and the Jump Lists,

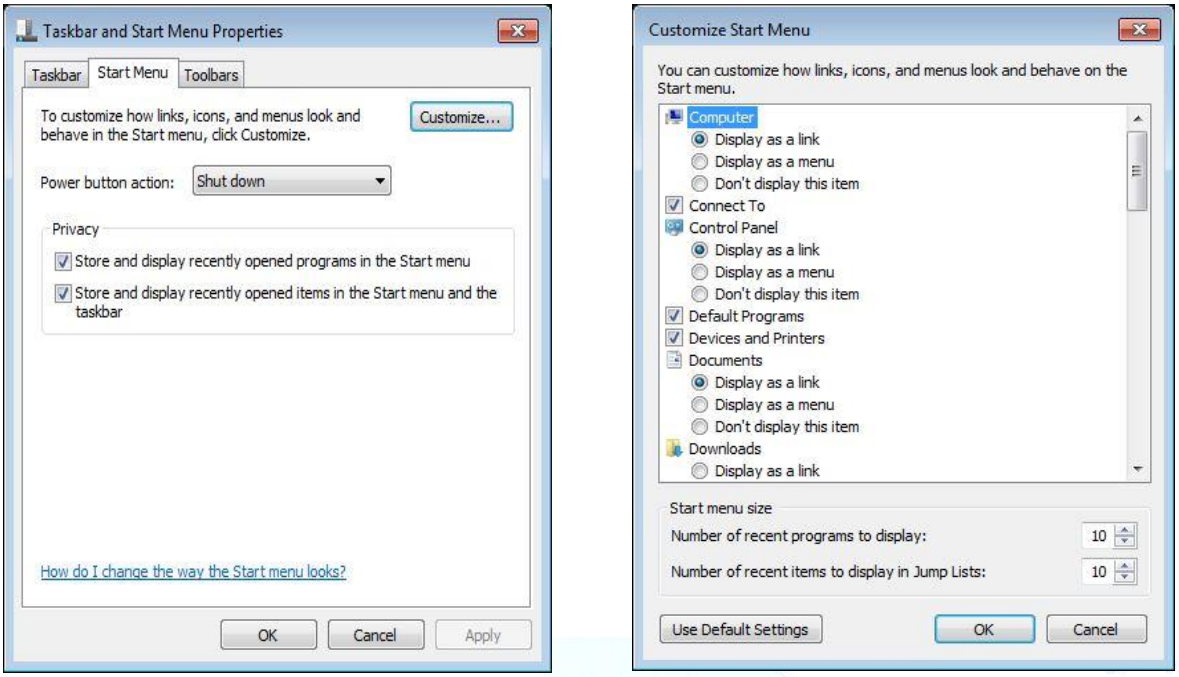


Figure 115 Jump list entry content

Windows 8

Jump Lists remain enabled by default in Windows 8, and have been evolved for the Start Experience. A feature called Secondary Tiles also utilizes Jump Lists, where a user can pin a new tile to the Start Experience for a part of a metro app. An example may be a pinned tile for a specific web site, The Taskbar properties has a new Jump Lists tab for all settings in Windows 8, as seen in Figure.

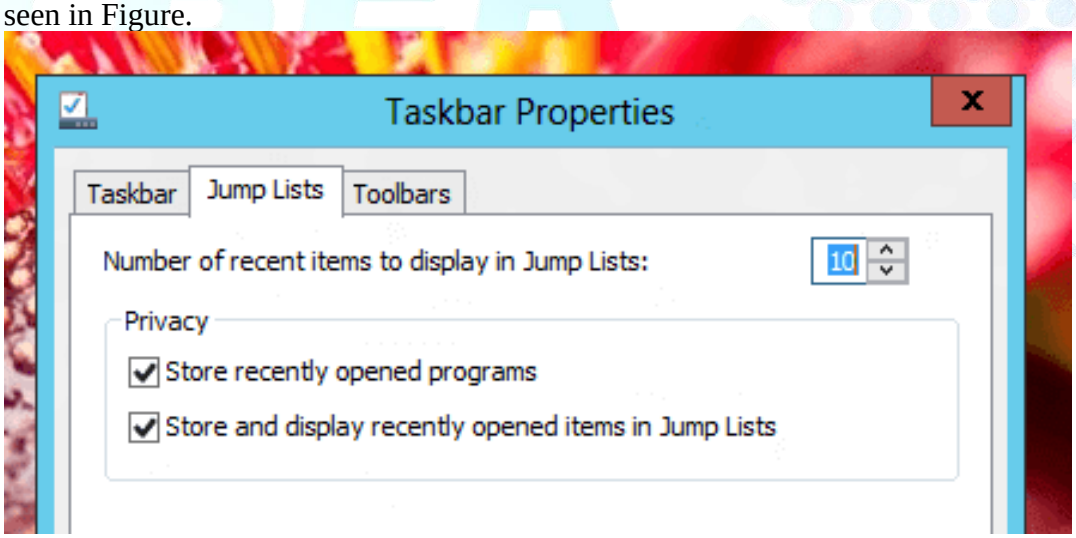


Figure 116 Windows 8 Jump list setting

Windows 10

Jump lists are enabled by default in Windows 10 and users still have the ability to disable their use in the Start Menu and on the Taskbar. In Windows 10, the binary structure of the destination list has changed slightly which has created issues for some of the tools described above. The destination list and Link files are still viewable with a compound file viewer and with most major forensic tools.

Also, in Windows 10, Microsoft has added functionality with Windows Explorer tying into the Jump Lists, specifically the “Frequent Folders” view in “Quick Access”. The file with the Jump List ID "01b4d95cf55d32a”, is the location utilized by Windows Explorer to populate the Frequent Folders viewed. Within the Frequent Folders Jump List a user can pin folders to Quick Access, manually adding data to this Jump List, and/or the Operating system will track 4 folders automatically which the user frequently interacts with.
When a folder is manually pinned to the Frequent Folders view, it is given a pinned status and location. The order is logically first at the top left then reading to the right. Shown in Figure is the pin order, 1 will begin at the top left of the Windows Explorer Frequent Folders.

#	App ID	Linked Path	Volume Name	Volume Serial Number	Target NetBIOS Na...	Accessed Count	Pin Status
6	f01b4d95cf55d32a	C:\Users\Student\Desktop\One more folder	TI10657300E	AC12CDF7	laptop012	10	Not Pinned
2	f01b4d95cf55d32a	C:\Users\Student\Desktop\Registry Practical	TI10657300E	AC12CDF7	laptop012	57	Not Pinned
21	f01b4d95cf55d32a	C:\Users\Student\Videos	TI10657300E	AC12CDF7	laptop012	1	Pin order 1
17	f01b4d95cf55d32a	C:\Users\Student\Downloads	TI10657300E	AC12CDF7	laptop012	15	Pin order 2
22	f01b4d95cf55d32a					0	Pin order 4
18	f01b4d95cf55d32a	C:\Users\Student\Desktop	TI10657300E	AC12CDF7	laptop012	17	Pin order 4
20	f01b4d95cf55d32a	C:\Users\Student\Documents	TI10657300E	AC12CDF7	laptop012	1	Pin order 5
19	f01b4d95cf55d32a	C:\Users\Student\Pictures	TI10657300E	AC12CDF7	laptop012	1	Pin order 6
16	f01b4d95cf55d32a	C:\Users\Student\Desktop\bjcavsfan	TI10657300E	AC12CDF7	laptop012	3	Pin order 7
15	f01b4d95cf55d32a	C:\Users\Student\Desktop\New folder	TI10657300E	AC12CDF7	laptop012	0	Pin order 8
14	f01b4d95cf55d32a	C:\Users\Student\Desktop\FAT Practical	TI10657300E	AC12CDF7	laptop012	0	Pin order 9

Figure 117 Windows 10 frequent folders pinned status

The automatically pinned folders displayed are populated using the access count in the Jump List. The access count has not proven to be completely reliable for an actual count of access by a user, however it does show interaction with folders and the access count typically increases with more frequent use, but this is not always 100% accurate.
During testing it was also determined that the highest Access Count was not always automatically pinned, however the folders automatically pinned always came from the highest part of the list. The Jump List also tracks additional folders accessed, which are not pinned nor automatically displayed by Windows Explorer. The “Frequent Folders" view may also continue to show folders to the user that he/she has already deleted.

c) Recycle Bin

The Recycle Bin was designed by Microsoft to protect user's data from accidental deletion, and to protect users from themselves, by “recycling” unwanted files instead of deleting them, as shown in the Delete File confirmation dialogue - “..move this file to the Recycle Bin?”, in Figure.

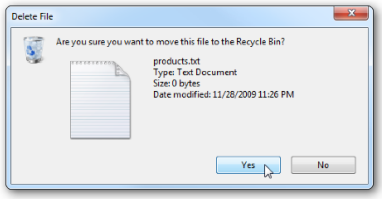


Figure 118 “Delete” File warning

The Recycle Bin is a series of hidden system folders that contain the unwanted files. When a user “deletes” a file in Microsoft Windows, the file content is not deleted or moved. The directory entry is moved (FAT) or the $MFT entry is changed (NTFS) to show that the file is now located in the Recycle Bin.

A Recycle Bin is only created on volume marked as a Fixed Disk (non-removable). The on-disk format and operation of the Recycle Bin is affected by two things - the version of

Microsoft Windows and the file system in use. The effect of the files system is the same on all versions of the Microsoft Windows Operating System:

- On a FAT volume, all Recycle Bin files are stored under the root folder.
- On an NTFS volume, the first time a user deletes a file, a folder is created under the Recycle Bin, and is given a file name which is the user's Security Identifier (SID). Everything that is deleted by this user account is then placed under this folder.

The SID is the same as used in Registry, and can be matched to the user's name in the Registry Key:
HKLM\SOFTWARE \ Microsoft\WindowsNT\CurrentVersion\ProfileList.
A Recycle Bin that Contains multiple SID's with different domain or computer identifiers indicates the volume may have been connected to multiple Computers, or accessed by users from different domains.
In all versions of Windows, pressing Shift+delete will bypass the Recycle Bin, and the files will be marked as deleted in the file system, as shown in the Delete File confirmation dialogue "...permanently delete this file?" in Figure - note that there is no Recycle Bin icon in this window.

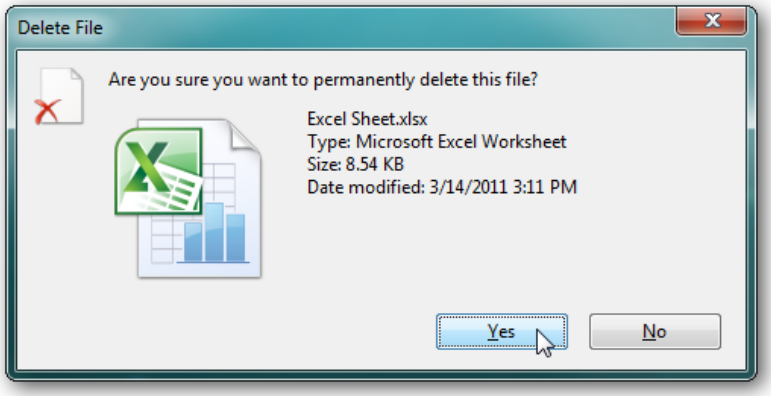


Figure 119 "Delete" File warning

In Windows XP, the Recycle Bin is actually a hidden system folder in the root of the volume. The folder name is dependent on the file system:

- "Recycler" on NTFS volumes - %drive letter%\Recycler\%SID%\
- "Recycled" on FAT volumes - %drive letter%\Recycled\

By default, the Recycle Bin can grow in size up to ten percent of the volume size. Any files/folders that are too large for the Recycle Bin will be deleted. Figure shows the Recycle Bin on an NTFS volume, with two SID's.

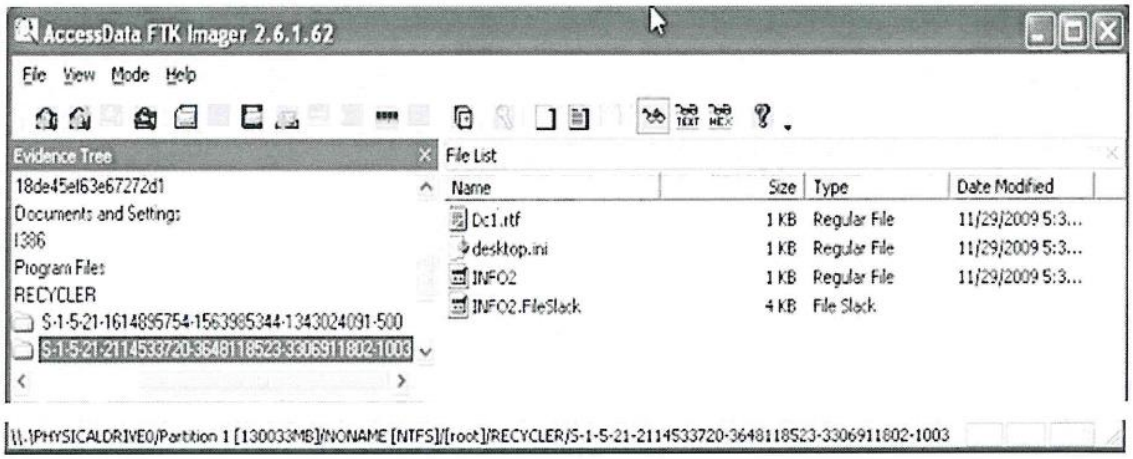


Figure 120 user-1003 Recycle Bin Files

The SID's in the Recycler shown in Figure have different domain/computer identities. This can mean that the volume has been connected to more than one computer or that the volume has been connected to one computer, where a user with a domain account has recycled files and a local computer account have recycled files.

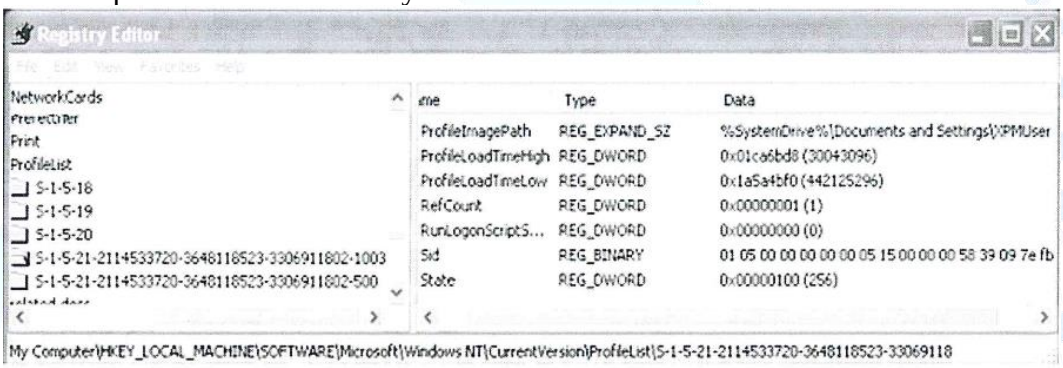


Figure 121 user – 1003 User Profile Path

By default, Windows creates a Recycle Bin folder on each non-removable drive it connected to the system. The recycle bin located on the Desktop shows files deleted from all non-removable volumes on the system. Microsoft maintains an INFO2 file within each \Recycler folder, which contains the original path of the deleted files.

When a user or Windows compliant application deletes a file, three things occur - with slight differences for each file system:

1. **NTFS** - The File Record is changed to show a new parent $MFT record number for the folder with the account SID - \\Recycler\%SID%.

FAT - The first character of the file name is replaced with 0xE5 to show the deletion.

When a directory is deleted on a FAT volume, the parent directory entry is marked as deleted by replacing the first character in the directory name with OxE5. The file names within the deleted directory are NOT changed.

2. NTFS - the file name is changed.

FAT - A new directory entry is created for the file in the \Recycler folder. With the exception of the file name, the new entry in the \Recycler folder contains the same file attributes, file dates/times, and starting cluster number as the original entry.

3. A new entry is added to the INFO2 file located in the \Recycled (or \Recycler\%SID%) folder. 800 bytes are allocated for each entry in the INFO2, which contains:

- Date and time of deletion.
- File Size.

- It's index number in the recycle bin (its order in the recycle bin)
  - 1is assigned to the first file,
  - 2 to the next,
  - etc.
- The Path and original file name of the file deleted to the recycle bin.

XP Recycle Bin File Name Rules:

The file name for the new entry in the \Recycler folder follows a very specific file naming convention. This naming convention is:

- Drive letter,
- File number,
- Original file extension.

Take for example, the file C:\My Pictures\bomb diagram.jpg. If this was deleted to an empty Recycle Bin, the new file name would be:

DC1JPG - Drive C, first file deleted, JPG extension retained.

If the file D:\Secret\Payments.xls was Recycled and the user's SID folder already contained four files, then new file name would be:

DDOS.XLS - Drive D fifth deleted file.

When the Recycle Bin is emptied, if the last file in the recycle bin was entry number 10 and the bin was emptied, the next deleted file sent to the recycle bin would be 11. If the user empties the bin and logs off, the count is reset to 1.

INFO2 FILE STRUCTURE:

The INFO2 file begins with a 16-byte header. The header includes the size of each entry within the INFO2 file, at offset 13 (0x0C) for four bytes - the entry size has always been 800 bytes.

After the header, each 800-byte entry is written to the INFO2 file when a file or folder is added to the Recycle Bin. The data structure for an INFO2 record is given in the Table 5.2.

Offset	Length (bytes)	Description
0x04	260 (0x104)	Original File Name and Full Path - ASCII
0x108	4	Recycled Record Number (incremental count)
0x10C	4	Drive number for Recycled File – 0=A:\ 4=E:\ 8=I:\ 12=M:\ 1= B:\ 5=F:\ 9=J:\ 13=N:\ 2=C:\ 6=G:\ 10=K:\ 14=O:\ 3=D:\ 7=H:\ 11=L:\ etc...
0x110	8	File Recycled Time (FILETIME - UTC)
0x118	4	Deleted file physical size
0X11C	520 (0x208)	Original File Name and Full Path - Unicode

Table 11 INFO2 record structure

An INFO2 Record is shown in Figure. The INFO2 header is shown in the first line, immediately followed by the first entry. Each entry is written sequentially to the INFO2 file after the previous.

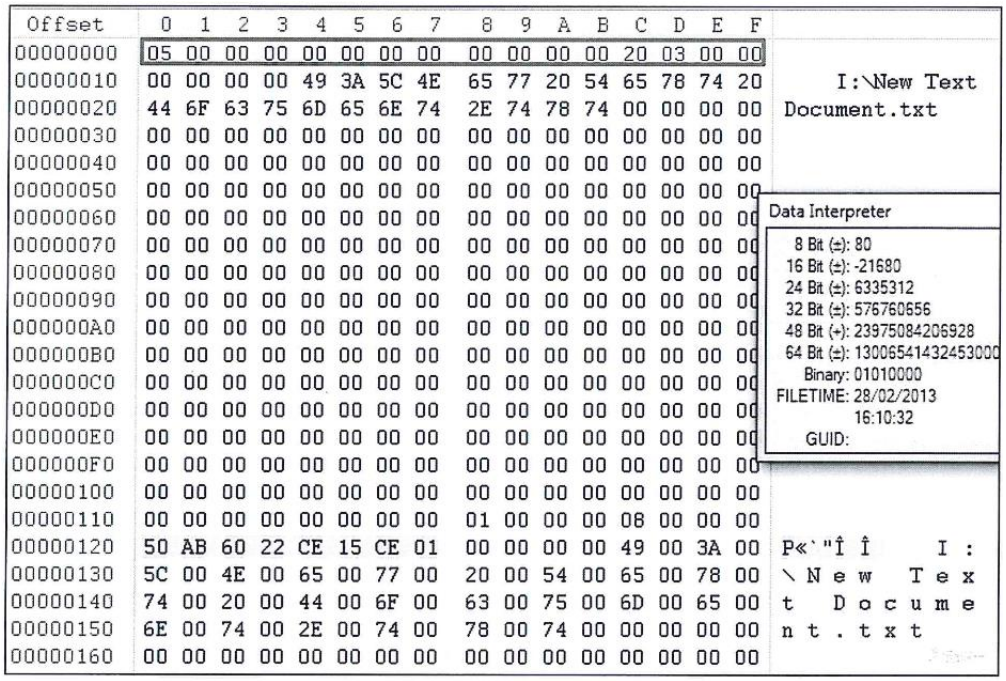


Figure 122 INFO2 record entry

The original file was I:\New Text Document.txt, it was sent to the Recycle Bin at 28 February 2013 16:10:32 UTC, and was zero bytes long.
The record number for this file is 01, and the drive number is 08 (I:\) – the name for this file in the Recycle Bin will be Di01.txt.

EMPTYING THE RECYCLE BIN

If an individual file is removed from the Recycle Bin, (it is restored, or deleted) the first character of the ASCII path in the INFO2 file is changed to 0x00.

If the entire Recycle Bin is emptied, Windows, resizes the INFO2 file to 20 bytes and modifies the Desktop.ini file. It should be noted that even though the INFO2 file has been resized, the previous entries, or portions thereof, may be recoverable.

The modified date of the file desktop.ini is a good indication of then the Recycle Bin was emptied.

$RECYCLE.BIN — VisTA,7,8 AND 10

There has been a significant change to the way Windows Vista\7\8\10 stores deleted files. The Recycle Bin is still a hidden system folder but the name has been changed to $Recycle.Bin.
As with Windows XP, it is configured for each logical drive and is not created on drives marked "removable". The folder name for the Recycle Bin has been renamed, and is the same for both FAT and NTFS file systems:

- FAT - %drive letter%\$Recycle.Bin\.
- NTFS - %drive letter%\Recycle.Bin\%SID%\.

The first notable change is that the individual SID (user folder) is created when a user logs on for the first time. The most significant change is that the $Recycle.Bin uses a set of paired files to track one deleted item.

FILE DELETION

When a file is deleted, the original directory entry is still marked as deleted in the same way in FAT, or if it is NTFS, the $MFT is changed in the same way as described during an XP Recycle function.

This original file becomes the first of the paired Recycle.Bin files, and is renamed "$Rxxxxxx" - $R followed by a random six alphanumeric character file name. The extension from the deleted file remains the same. This file contains the content of the original file.

Example >>> MyPicture.jpg – will become - - - - **$R123456.jpg**

The second paired file is an administrative file and the file name is starts with $I, then has the same random six characters as the $R and the extension - - - **$1123456.jpg**.

This file is 544 bytes long, in Windows Vista,7 and 8, but in Windows 10 the file uses additional values to determine the length of the file path and name, only using what is necessary. The $I file tracks the deleted file's time of deletion (offset 0x10) and the full path to the original location. There are few differences with regard to Windows Vista, 7,8 and Windows 10, however there are two main differences to keep in mind. The first change is that the header of the $I file changed from 0x01 00 00 00 00 00 00 00 to 0x02 00 00 00 00 00 00 00. The second change is that the $I file now tracks the length of the file path and name so that it doesn't have to assign 544 bytes to each $1 file. Figure 5.24 shows the $R and $1 files.

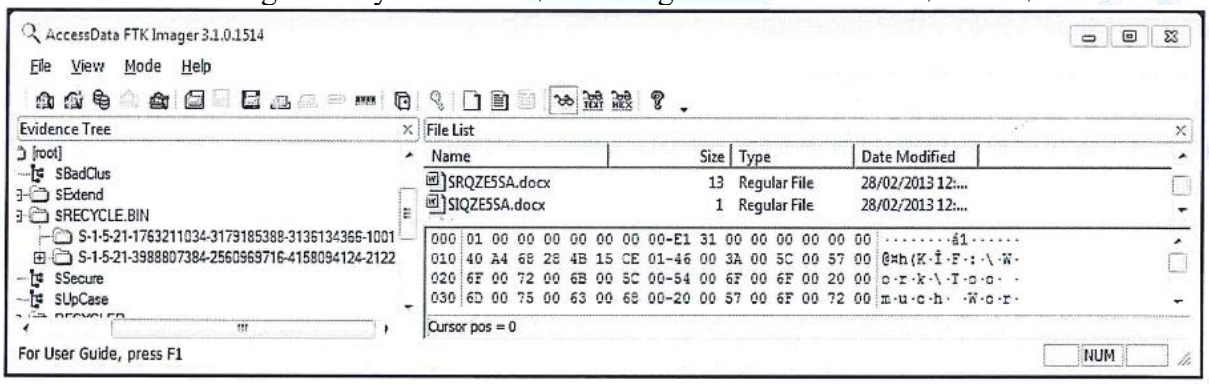


Figure 123 $Recycle.Bin $I file content

The data structure for the $I file in Windows Vista,7 and 8 is given in the table 5.3

Offset	Length	Description
0x00	8 bytes	File Header
0x08	8 bytes	Original File Size in bytes
0x10	8 bytes	File Recycled Time (FILETIME – UTC)
0x2018	~	Original File Name and Full Path - Unicode

Table 12 $I file structure

A $I file is shown figure

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	01	00	00	00	00	00	00	00	E1	31	00	00	00	00	00	00	á1
00000010	40	A4	68	28	4B	15	CE	01	46	00	3A	00	5C	00	57	00	@¤h(K Î F : \ W
00000020	6F	00	72	00	6B	00	5C	00	54	00	6F	00	6F	00	20	00	o r k \ T o o
00000030	6D	00	75	00	63	00	68	00	20	00	57	00	6F	00	72	00	m u c h W o r
00000040	6B	00	2E	00	64	00	6F	00	63	00	78	00	00	00	00	00	k . d o c x

Figure 124 $I file content

The original file was F:\Work\Too much Work.docx. It was 0x31E1 (12,679) bytes long and was recycled on 28 February 2013 00:32:58 UTC.

The data structure for the $1 files in Windows 10 is given in the Table 5.4.

Offset	Length	Description
0x00	8 bytes	File Header
0x08	8 bytes	Original File Size in bytes
0x10	8 bytes	File Recycled Time (FILETIME – UTC)
0x18	4 bytes	File path length (in ASCII characters)
0x1C	~	Original File Name and Full Path - Unicode

Table 13 $I file structure

DIRECTORY DELETION

If a directory is deleted, then there will be a $R and $I entry in the Recycle Bin for that directory. The $R will still contain all the sub-directories and files with their original (unaltered) names. The directory (FAT) or INDX (NTFS) content remain unchanged.

EMPTY RECYCLE BIN

On removal from $Recycle Bin, the files are deleted but may still be recovered.

There has been a lot of speculation that Windows 7and 8 may deliberately over-write (wipe) files that have been deleted after the Recycle Bin is emptied, or deleted files that have bypassed the Recycle Bin. Before passing judgment on this theory, two points need to be taken into consideration.

- While a Windows operating system is in a powered-on state, there is a lot of disk activity on the system drive (C:\), even though there is no user interaction and it appears there may be no applications open. But Windows Defender will be running, Windows Update may be downloading patches, Windows Search is Indexing, in addition to all the system files that Windows interacts with in the background.
- Due to the nature of NTFS, a $I record will be resident in the $MFT. Combined with the constant disk activity, the $MFT entry is often reassigned quite quickly, making the $1 content unrecoverable. The same applies with the $MFT entry for the $R record.

Microsoft Windows 7,8 and 10 do not deliberately wipe deleted files - regardless if they have been through the Recycle Bin or bypassed it. Normal Windows activity on a running system will over-write deleted files and make it appear that wiping is occurring. This is proven by running the test on a non-system volume, or by using a large number of test files.

Once the file records have been re-used, remnants of the file content can be carved from unallocated space.

d) Ram Files

Random Access Memory (RAM) is volatile (data not retained when power is removed), high speed memory that is used by a computer system to store data for quicker access by the processor than if the data had been retrieved from the hard drive. RAM is allocated in blocks, called pages, which are typically 4096 bytes. The operating system manages the RAM, and allocates the RAM pages to it, to running programs, and to files that are currently in use.

Windows uses special files on the hard disk for to help manage RAM, called virtual memory. This virtual memory is used to enhance performance, by creating making the RAM appear larger or by speeding up the shut-down / start-up process.

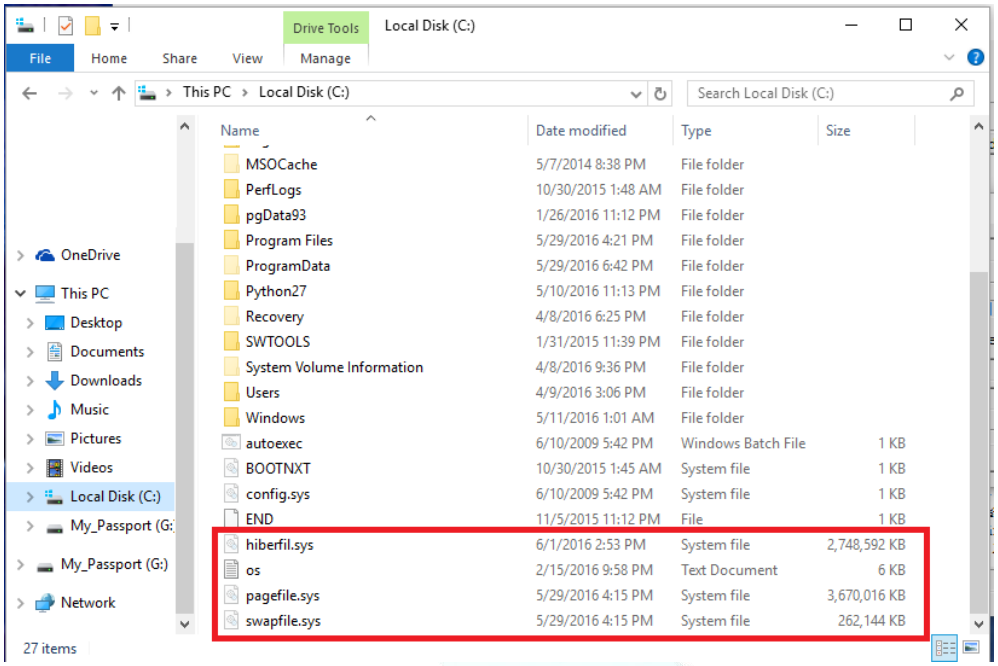


Figure 125 Picture of Pagefile.sys

Pagefile

Windows uses a pagefile file(s) to hold parts of programs and data that do not fit in memory. The operating system moves data from the pagefile to memory as needed and moves data out of memory to the pagefile to make room for new data. On XP, Vista and 7 systems it is named pagefile.sys. Windows 8 and 10 added a second virtual memory file named Swapfile.sys.

The pagefile is potentially a dynamite location to find data that the user does not know is still on the disk. In fact, while running, the contents of pagefile are not accessible to the user. Although the content is not formatted for easy reading, passwords, graphics, text files, file names, URLs and other valuable information is often found within this file.

Figure 37 shows a .jpg image that was a screen shot of a Windows 7 start menu found in a pagefile. A file does not have to be saved to the internal hard drive to be found in the page file - if it was created on the computer and not saved, it may appear in the pagefile.sys.

By default, pagefile.sys is created in the root folder of the drive that holds the Windows system files. The user can change the size of the pagefile, move it to a separate physical drive, spread it across multiple disk drives or even disable it. The default file size of pagefile.sys will range between 1.5 to 3 times the sizes of the physical RAM; however, the pagefile.sys file size will not decrease.

When the system is shutdown, the paging file remains intact. However, the Registry key below can be set to 1, in which case Windows will fill inactive pages in the paging file with zeros whenever you shut down the system. HKLM\System\CurrentControlSet\Control\Session_Manager\MemoryManagement\ClearPageFileAtShutdown.

Windows 8 and 10

In Windows 8 and 10, Microsoft utilizes two swap files, Pagefile.sys and Swapfile.sys to handle the operating system demand on the RAM, as shown in Figure 44. According to Microsoft, the Pagefile is utilized for RAM, and the Swapfile is utilized for swapping out applications - more specifically for metro mode applications. Therefore, it increases the chance to finding important artefacts from the swap file.

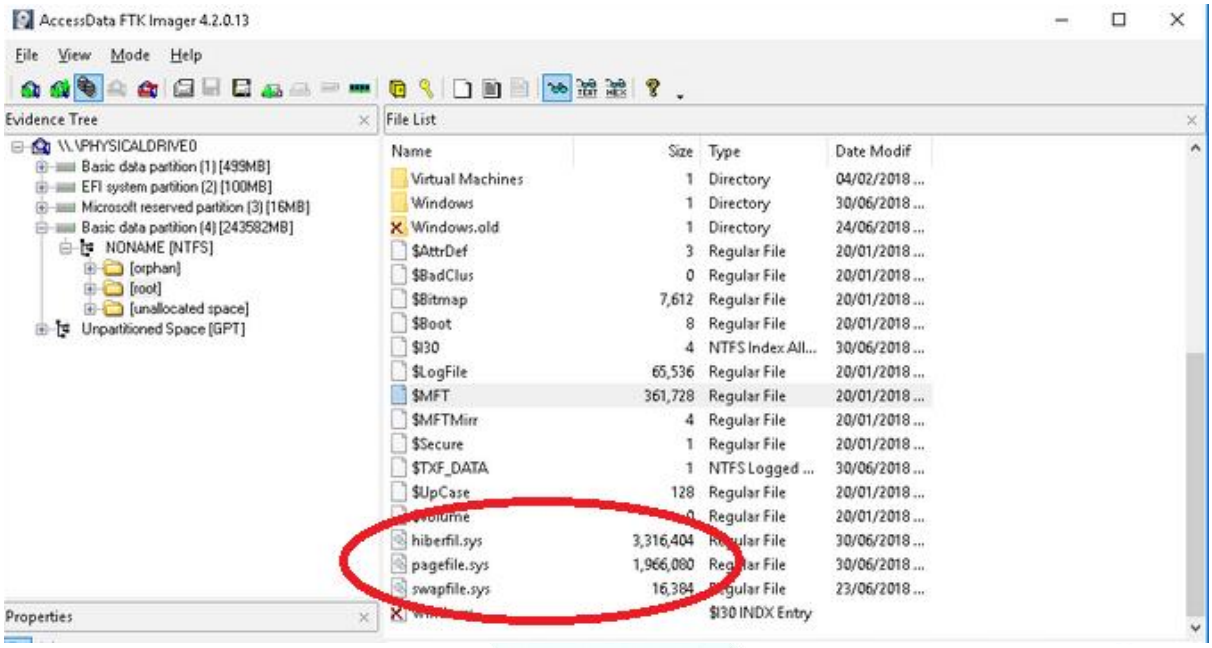


Figure 126 Windows 8 Ram swap files

Hibernation

Hibernation is a power mode that allows the computer to power completely off, but retain its current state. All open programs are retained by saving the content of the Random-Access Memory to the hard disk drive, in a C:\Hiberfil.sys. When the computer is powered back on, the contents of Hiberfil.sys is copied into RAM, and the computer returns to the same state as when the computer was shut down.

Just like the RAM and swapfiles, Hiberfil.sys will often contain files or parts of files that the user has been working on, even if the user didn't save those files to the internal hard drive. The hiberfil.sys files is compressed and can be read using MoonSols Windows Memory Toolkit,

http://www.moonsols.com/windows-memory-toolkit/.

Analyzing RAM Files

A RAM file and a RAM dump both contain live data which can include unencrypted passwords, and running program data. The latter is very important when investigating a computer that may be infected by a virus or other exploit. In depth RAM analysis is beyond the scope of this course, and requires specialized tools to be done effectively. Some RAM analysis tools are discussed below.

Strings

Strings.exe (http://technet.microsoft.com/en-us/sysinternals/bb897439) is a command line utility for extracting Unicode and/or ASCII strings, which may include plaintext passwords. It will work on any type of file and makes a dictionary file that may assist cracking encrypted files. To use:

1. Copy stings.exe in the same folder as the RAM dump or the RAM file - use 8.3 file names.
2. Open a command window and navigate to the folder containing the RAM image and "strings.exe"
3. Run the command specifying:
   a. Input file - which file(s) should strings.exe scan for text; and,
   b. Output file - where the output should go.

strings.exe [input file name] > [output file name] Excluding Square brackets.

Bulk Extractor

Bulk Extractor (http://www.forensicswiki.org/wiki/Bulk_extractor#Download) is a simple Graphical User Interface (GUI) program, which requires Java to be installed.

Like strings, it will get ASCII and Unicode strings, but also refine the results into credit card numbers (ccn), e-mail, MAC addresses, Uniform Resource Locators (URL) and telephone numbers, and much more, as shown below.

To run the tool > select tools > Run Bulk Extractor. There is a list of scanners that can selected to find selected information.

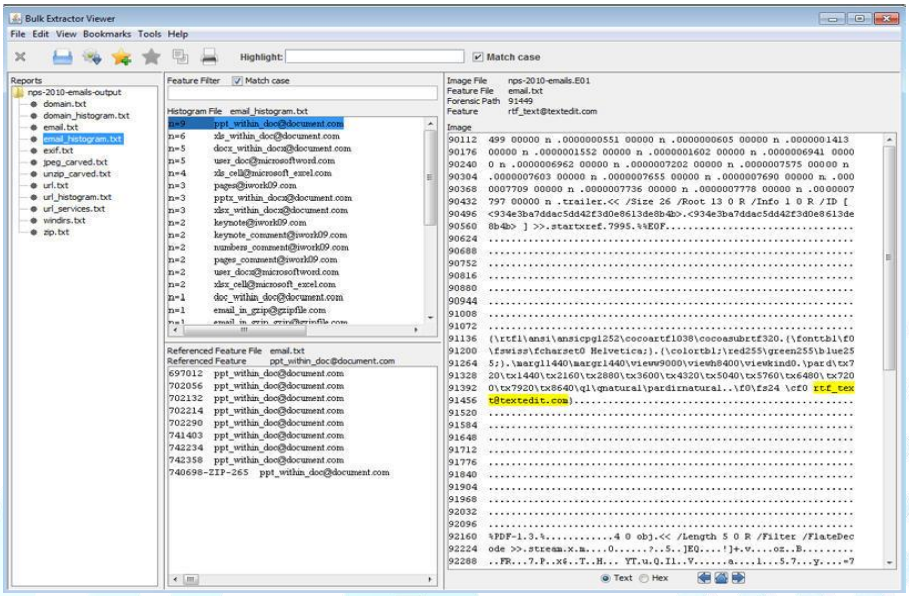


Figure 127 Bulk Extractor

Mandiant Redline

Redline (https://www.mandiant.com/resources/download/redline) is probably the most powerful GUI RAM application, when you get past the default scripts or configure a series of capture and analysis scripts that work! The drawbacks are that it is still experimental for Windows 8 and it will only analyze RAM image files in Data Dump (DD) format, and the analysis computer must have sufficient RAM to load the entire RAM image.

SIFT Workstation

The SIFT workstation (http://digital-forensics.sans.org/community/downloads) is an advanced and completely virtual machine based forensic investigation tool, with powerful utilities tool for RAM analysis. It must be run in VMWare, and is predominantly command line based (for RAM tools).

Volatility

The Volatility Foundation (http://www.volatilityfoundation.org/) released Volatility in 2007 as an Open Source Project to support memory forensic analysis. Volatility is multiplatform tool which integrates structured analysis of memory samples. This tool is a command line-based tool which incorporates plugins for various Operating Systems, applications, and platforms (i.e. Windows, Linux, and Mac).

e) Thumbnail Cache

The concept of a thumbnail cache is to help users find their desired files quickly by allowing Windows Explorer to pre-view the content of a picture of graphic file as a "thumbnail". A thumbnail image is simply a miniature view of a graphic (picture) file.

Windows XP - THUMBS.DB

If the user sets Windows Explorer to "Thumbnail" or "Filmstrip" view, Windows Explorer will display files as thumbnail images without having to open a separate viewer.

Windows accomplishes this by creating a hidden system file named thumbs.db in each folder that contains a graphics image file when the user selects to view that folder in thumbnail mode. A thumbnail sized image is then placed (cached) in the thumbs.db file for each image file within the folder.

The thumbs.db file gets updated if an image file is, or has been, added to the folder. Windows Explorer then reads this file and displays the thumbnail images contained within the file. There will not, however, be a thumbs.db file created on removable media.

The image formats JPEG, BMP, GIF, TIF, PDF and HTM are some of the file formats stored as thumbnails. Each thumbnail is then represented in the thumbs.db as a JPEG, regardless of the original format. The thumbs.db file contains a listing of the filename and the Last Modification date of the original file.

Thumbnail caching is on by default, however; the user can choose to disable it using the View Tab of the Folders Option Dialog. The setting is stored in the user's registry, NTUser.dat\Software\ Microsoft\Windows\CurrentVersion\Explorer\Advanced under the value DisableThumbnailCache. A data value of 0 for this key indicates that Thumbnail caching is on, a data value of 1 indicates that Thumbnail Caching is off.

An excellent tool or viewing thumbs.db was created by Eric Kutcher called Thumbs Viewer, and is available under General Public License for download from: https://thumbsviewer.github.io/. Figure 5.29 shows a thumbs.db file in Thumbs Viewer - note that there is one picture called "autumn.jpg".

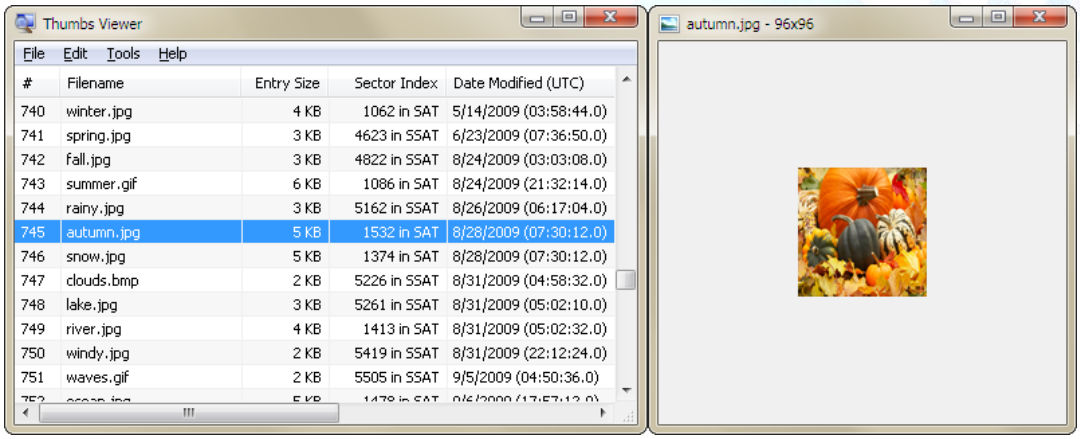


Figure 128 Thumbcache Viewer

Synchronization between the thumbs.db file and the folder content is not perfectly maintained. The thumbs.db is updated whenever a graphics file is added to the folder - if the folder is being viewed in thumbnail or filmstrip mode.

However, when a user deletes a file or files from the folder, the thumbnail image and related data in the thumbs.db is not removed and will remain unless the entire thumbs.db file or the entire folder containing the thumbs.db is deleted. Figure 5.30 demonstrates this, by showing the content of the folder that the thumbs.db file in Figure 5.29 was taken from. The

folder is empty - the file “autumn.jpg” has been deleted, but remnants can be found in thumbs.db.

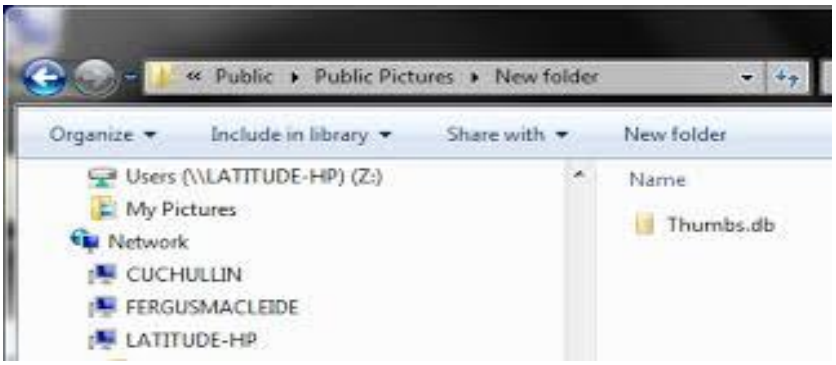


Figure 129 Picture deleted from folder

In this situation, the examiner can use file name from thumbs.db as a keyword search to try and identify any other remnant information about the file. This is very beneficial for the forensic examiner as the user assumed that by deleting the images, all traces of them were gone. Also, with regards to encryption, it may be possible to see the thumbnail while the actual file remains encrypted. As the thumbs.db file is a hidden system file and many users are totally unaware of its existence. Obviously, in Child Sexual Abuse and other cases that commonly involve image files the thumbs.db files can contain significant investigative information and assist in proving knowledgeable possession. The picture may be deleted but the jpg in Thumbs.db remains.

WINDOWS VISTA — 10 - THUMBCACHE

Windows Vista,7,8 and 10 thumbs.db was replaced with thumbcache. However, a thumbs.db file can still be found in folders that have been shared - it is created when that folder is mapped to and the pictures are viewed in thumbnail mode.

With thumbcache, there is a _ centralized cache for each user located in C:\Users\ [user] \AppData\Local\Microsoft\Windows\Explorer. This may be helpful to link a particular image with a particular user.

The registry keys for thumbnail caching can be located for each user in the NTUser.dat file. A datavalue of (0) means that thumbs are displayed and a data value of (1) means that only icons are displayed.

NTUSER.dat

- Software\Microsoft\Windows\CurrentVersion\ Policies\Explorer\
  - NoThumbnailCache = 00000001
  - DisableThumbnailCache = 00000001
- Software\Policies\Microsoft\Windows\Explorer\,
  - DisableThumbsDBOnNetworkFolders = 00000001
- Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\
  - DisableThumbnailCache = 00000001
  - NoThumbnailCache = 00000001

HKEY Local Machine

- Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\
  - NoThumbnailCache = 00000001
  - DisableThumbnailCache = 00000001
- SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\

- DisableThumbnailCache = 00000001
- NoThumbnailCache = 00000001

Windows Vista and 7

In Windows 7, there are four different cache sizes, set in the folder view options as shown in Figure 32:

- Thumbcache _32.db
Thumbnails up to 32x32 pixels
BMP format for both files and folders

-Thumbcache_96.db
Thumbnails up to 96x96 pixels
BMP format for both files and folders

-Thumbcache_256.db
Thumbnails up to 256x256 pixels
JPG format for files, PNG format for folders

-Thumbcache_1024.db.
Thumbnails up to 1024x1024
JPG for files, no folders.

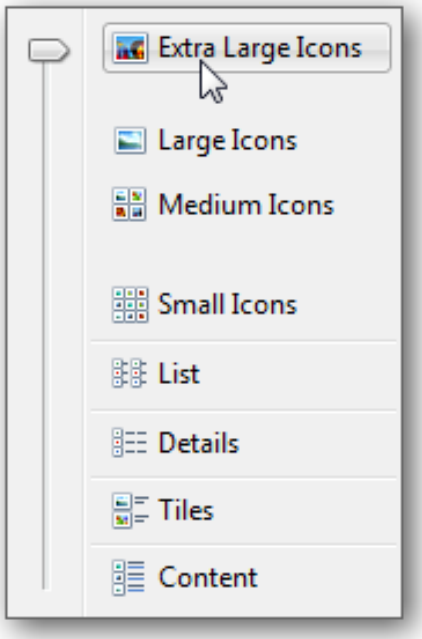


Figure 3.130 Icon Settings

Each of these files holds the different resolutions of the thumbnails on a system based on the users viewing option. The Picture folder defaults to large icons but can be changed by the user - Figure shows the settings available in Windows Vista and 7.

WINDOWS 8

During testing of Windows 8, the large icons were the _96 and the Extra-large icons were the _256 and the cache files did not populate unless the size was adjusted by the user. Scalable thumbnails of a file or folder are also visible rather than generic program icons. Figure 5.32 shows the view settings available in Windows 8.

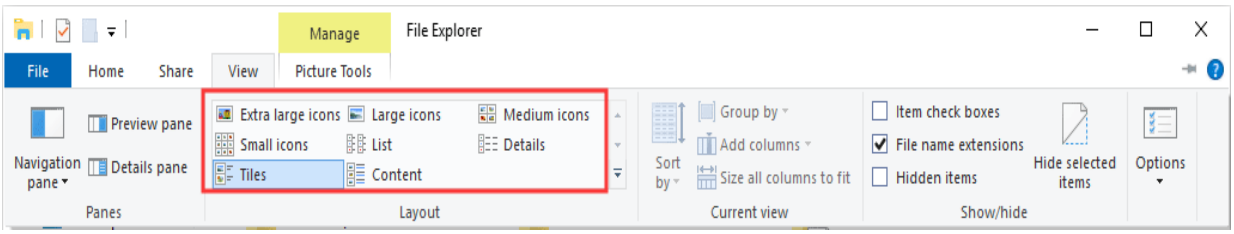


Figure 131 Icon Sizes in Windows 8

Windows 8 added an additional thumbcache size file and two other thumbcache files that seem to have a specific purpose.
- Thumbcache - 48.db
Thumbnails up to 96x96 pixels
BMP format for both files and folders

- Thumbcache_exifidb

JPG format

Holds the thumbnail of the first picture in a folder also known as an album cover.

- Thumbcache_wide.db

A Thumbcache file has an easily recognizable header of 0x43 0x4d Ox4d 0x4d (CMMM), whichallows an examiner to easily search for any deleted thumbcaches.

Windows 10

Windows 10 has changed some of the filenames/resolutions that are collected as thumbcache.

Windows 10 got rid of 1024 and 1600, but added 768, 1280, 1920, 2560, and a file titled thumbcache_custom_stream.db. The file headers for thumbcache remain the same and Windows 10 follows the same testing as described above with Windows 8 for size and storage. During testing of Windows 10, the thumbcache_768.db is populated automatically without the thumbnails being viewed. The file thumbcache_2560.db is used to cache images seen in the preview pane of the windows explorer. The file thumbcache_1280.db is used to cache images from the new Microsoft Photos application. On all Windows 10 systems that have been tested thus far both files thumbcache_1920.db and thumbcache_custom_stream.db have been unpopulated.

THUMBCACHE ANALYSIS

A major drawback of thumbcache is that the thumb files are renamed within the cache. The new file name is saved in hex as a 64-bit hash generated from the Volume GUID and MFT record.

Generally, the tools that can resolve the thumbnail to the original image require that image to be present on the machine and not deleted.

Eric Kutcher has also developed an excellent tool for thumbcache files called Thumbcache Viewer, shown in Figure 5.33. It is available under General Public License from:https://thumbcacheviewer.github.io/

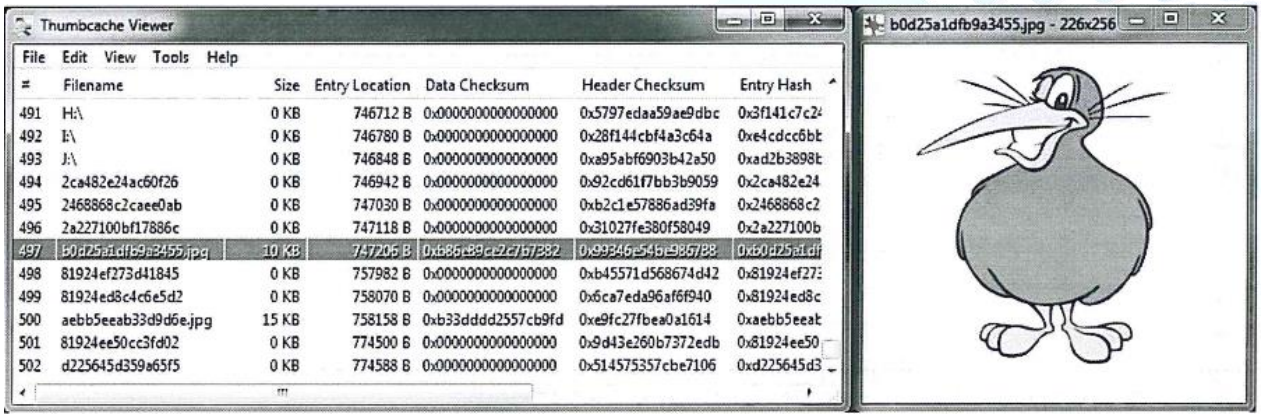


Figure 132 Thumbcache content

This thumbnail has been renamed to b0d25a1dfb9a3455.jpg, as shown in Figure 5.34.

This unusual name is called the ThumbNailCache Id, and is generated from the Volume GUID, the MFT record number and MFT Sequence Number of the picture.

Figure 133 Renaming Thumbnail

There is also an online = ThumbNailCache Id generator available at http://www.dmthumbs.com/test_platform/thumbnailcache_id.php. This is an interesting tool, but it could be useful if you only want to generate the ThumbNailCache Id of a single picture.

Windows needs to be able to relate that back to the original file in order to display the correct thumbnail to the user. An examiner can do the same thing, using the Windows Search Indexing database file, called windows.edb. This is located at:

C:\ProgramData\Microsoft\Search\Data\Applications\Windows\Windows.edb.

A database file is a structured set of data, and will be discussed during the Compound File block of the course. Windows.edb is a database that uses the Extensible Storage Engine (ESE) data storage technology.

This file cannot be copied when the computer is running, and is very sensitive to improper shutdowns, as it is continuously updated.

Woanware have an excellent tool called EseDbViewer freely available from http://www.woanware.co.uk/forensics/esedbviewer.html, which can be used to analyze *.edb files.

1. Copy the file Windows.edb from the evidential computer to a Windows 7, 8 or 10 laboratory computers.
   - C:\ProgramData\Microsoft\Search\Data\Applications\Windows\Windows.edb

2. Use EseDbViewer to open the file Windows.edb.
   - This often results in a database dirty shutdown error, which means the database requires a repair or recovery. If a repair is required, use the following steps:
     - In an Administrative command prompt, change directory to the folder that the copy of the Windows.edb file is located in.
     - Type the command: esentutl /p windows.edb (/p is repair. /r would be recovery). It is likely that to have an error prompt similar to that shown in Figure 5.35.

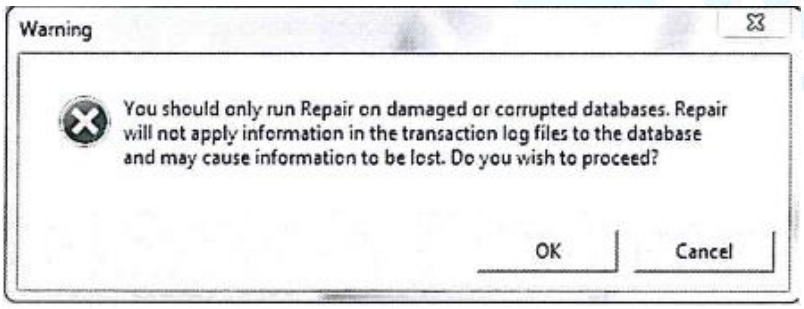


Figure 134 Recovery prompt

It is also possible that the command won't complete, and will need to be run additional times. Figure 5.36 shows the result of a successful repair.

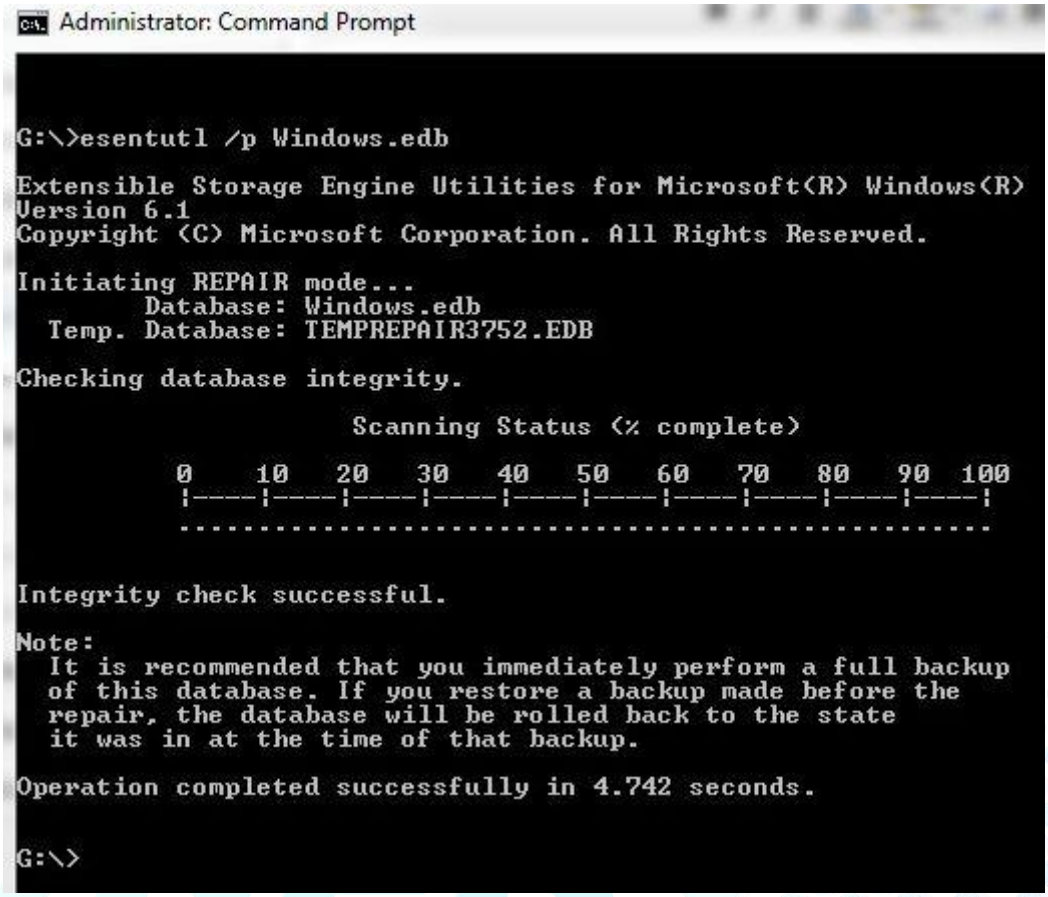


Figure 135 Repair Completed successfully

Use EseDbViewer to open the repaired file Windows.edb.

3. Select the correct table based on the exhibit computer's operating system:
   - On a Windows 7 computer, there will be seven tables - the table for this process is called SystemIndex_0A.
   - On a Windows 8 computer, there are seventeen tables - the table for this process is called SystemIndex_PropertyStore.

4. Export the current table to *.csv., and then open it with a spread sheet application.
   - This results in a very large spread sheet, which contains a lot of extraneous information, and can make navigation a bit cumbersome.

The final step is to search for the ThumbCacheId name of any thumbnails that have evidential value.

f) Prefetch

With all of the improvement to computers over the last fifteen years, the biggest consumer complaint to Microsoft from users is the speed at which systems boot, recover from hibernation, and launch applications.

To combat these complaints, Microsoft introduced prefetching. Prefetching speeds up computer performance by bringing the data and code pages of programs used during the 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 what files are called for during system boot and application launch are located in the folder %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 the .PF file extension. The file signature(header) for .PF files varies between versions:

- Windows XP 11 00 00 00.53 43 43 41 OF 00 00 00
- Windows Vista/7 170000 0053 43 41 11 00 00 00 00
- Windows 8 1A 00 00 00 53 43 41 11 00 00 00 00
- Windows 10 4D 41 4D 04

Bytes 5-8, in XP through 8, (53 43 43 41 hex) are "SCCA" in ASCII which is easier to remember as search string for examiners who need to recover deleted .PF files.

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

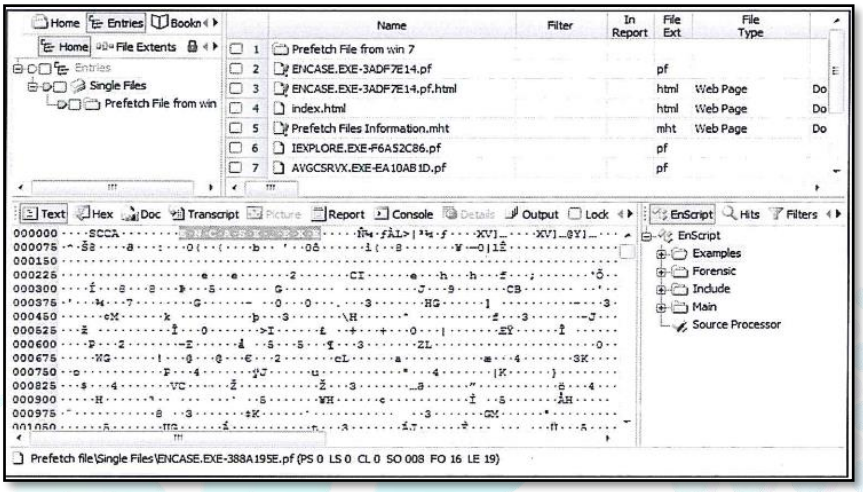


Figure 136 Prefetch Executable filename

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. Table 5.5 below gives the offsets for XP prefetch files.

Bytes		Data	Format
Offset	Length		
0x04	4	Header	SCCA
0x10	60	Application Name	Unicode
0x78	8	Last Run Date	FILETIME
0x90	4	Execution Run Count	Hex

Table 14 XP Prefetch format

The execution count is the number of times the application has been run. It is of note that if the application runs as part of the boot process the run count is NOT updated - this only applies to application launches that are not part of the startup or auto run processes.

The content of the .PF file also includes a record of the files and directories accessed during the first 10 seconds of application launch.

Even launching the most benign program can cause the operating system to access a number of files. Not all these are necessarily used by the application proper but as applications are launched and traces are done that data are used in subsequent launches.

Examining the files and directories accessed during the launch of an application can be very beneficial because it can reveal hidden directories, point to user accounts or show an application was accessed from an external storage drive. Simple File Parser by Chris Mayhew is available free from https://code.google.com/p/simple-file-parser/, and can parse XP and Vista/7 prefetch files.

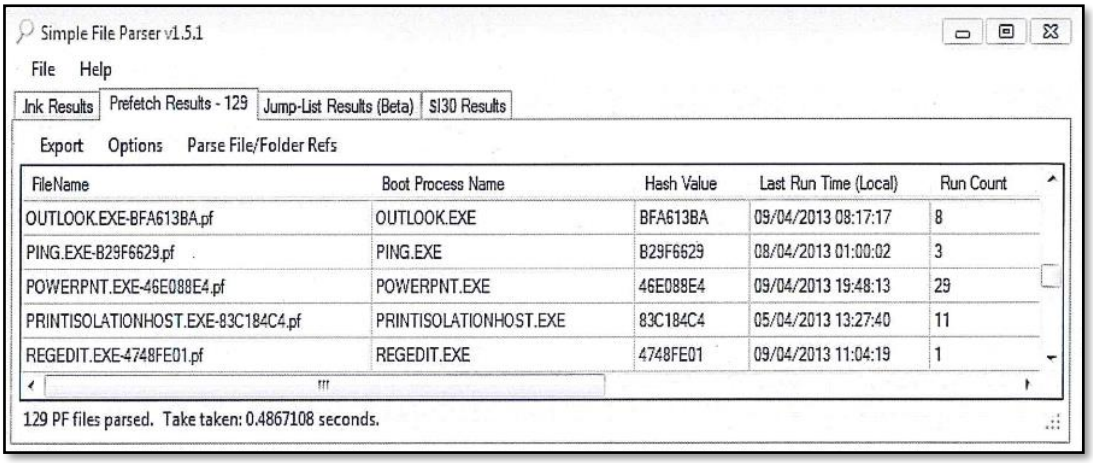


Figure 137 XP Prefetch Report using Simple File Parser

Windows Vista and 7

In Windows Vista and 7, there is a minor change to the data structure of a prefetch file, as given in Table below.

Bytes		Data	Format
Offset	Length		
0x04	4	Header	SCCA
0x10	60	Application Name	Unicode
0x80	8	Last Run Date	FILETIME
0x90	4	Execution Run Count	Hex

Table 15 Vista/7 Prefetch file structure

Windows 8

Windows 8 + has up to 1024 individual prefetch files. An individual application can have multiple prefetch files, for example, if the application is moved to a new directory a new prefetch file would be created as the. There have also been some changes to the file format, which allow a prefetch file to store the last eight runtimes of an application.

There are some anomalies with this edition to take note of. The run count only increments when a new date and time entry is added to the prefetch file. Once the run count reaches 10, Windows 8 only periodically updates the date and time and therefore also the run count. Testing has shown that once this occurs, the prefetch file is periodically updated when the program is open for a length of time and an "open" or "save" occurs within the program. Table gives the data structure for a Windows 8 prefetch file.

Bytes		Data	Format
Offset	Length		
0x04	4	Header	SCCA
0x10	60	Application Name	Unicode
0x80	8	Last Run Date	FILETIME
0x88	8	2nd Last Run Date	FILETIME
0x90	8	3rd Last Run Date	FILETIME
0x98	8	4th Last Run Date	FILETIME
0xA0	8	5th Last Run Date	FILETIME
0xA8	8	6th Last Run Date	FILETIME
0xB0	8	7th Last Run Date	FILETIME
0xB8	8	8th Last Run Date	FILETIME
0x90	4	Execution Run Count	Hex

Table 16 Vista/7 Prefetch

A free tool is available to parse.all versions of Prefetch Files, called WinPrefetchView. It is available from, https://www.nirsoft.net/utils/win_prefetch_view.html.An example of a Windows 8 Prefetch file is given in Figure below. Note that there are multiple programs.

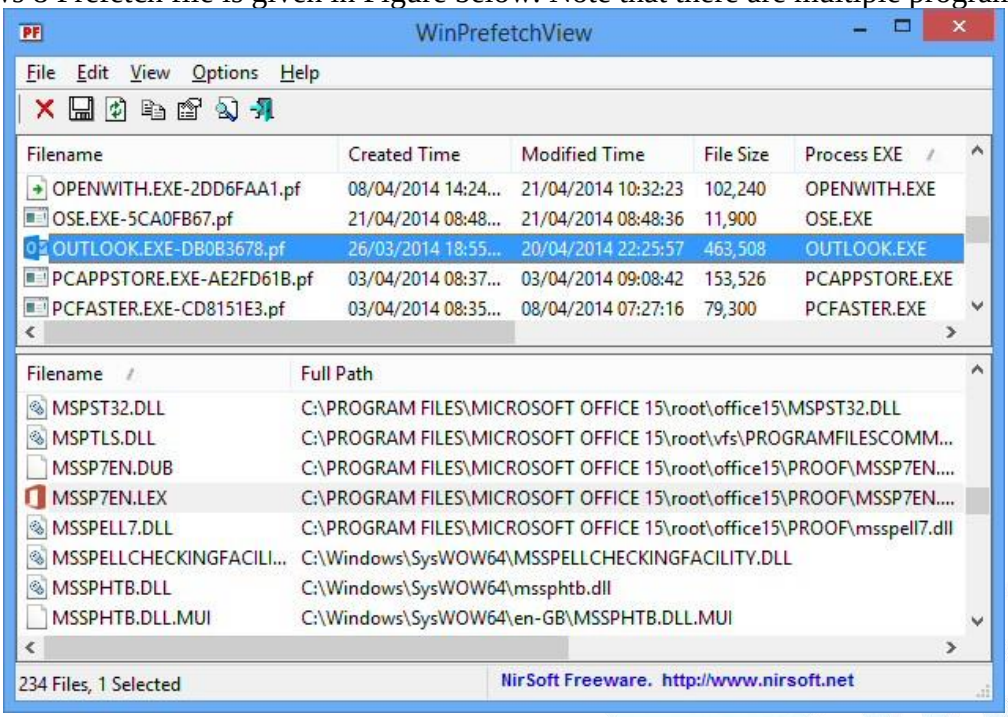


Figure 138 Windows 8 Prefetch in WinPrefetchView

Windows 10

Windows 10 has changed Prefetch only in the format in which the file is stored on disk. Windows 10 is now using the same compression as Windows 8.1 utilized in the compression of Superfetch files. Forensically speaking this doesn't change the role or functionality of Prefetch as described above, but it does change the tools used to view the contents of a prefetch file. Because the file needs to be uncompressed the data structure given above is not feasible for analysing the binary file. Also, Simple File Parser and Prefetcher, at the time this manual is written do not support the decompression of the Windows 10 prefetch files. The prefetch files are still located in the same directory, with the same extension, however the file header is now 0x4D 41 4D 04.

g) Sticky Notes

Sticky Notes is an application native to Windows Vista, 7, 8 and 10 that allows users to set digital notes and reminders for themselves that can reside anywhere on the desktop. Users have the ability to have single notes, multiple notes, add bullets, change the font, text size or even change the color of the notes. Figure shows three sticky notes posted to the desktop.

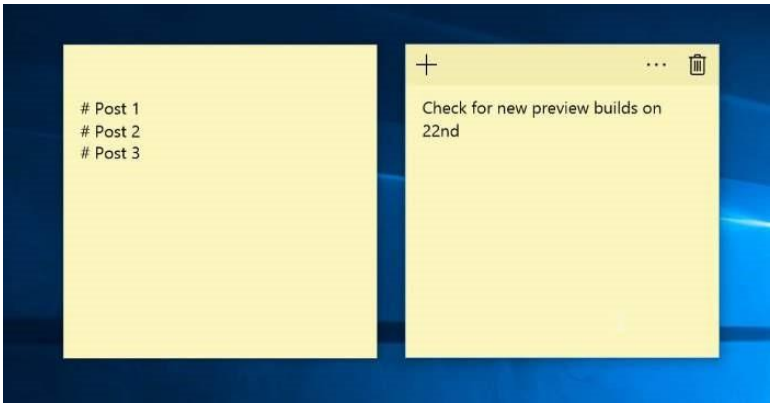


Figure 139 Sticky Note Desktop display

Sticky Notes are stored in %user_profile$\AppData\Roaming\Microsoft\Sticky Notes as StickyNotes.snt. This folder only exists if sticky notes have been used.

The header for this file, DO CF 11 EO A1 B1 1A E1, identifies it as a MS compound document. This file can be easily viewed with a compound file viewer such as CFX - Compound File Explorer (http://www.coco.co.uk/developers/CFX.html).

Figure shows that are three different storages in the file, 69495598-de13-11de-9, 72a62b60-de13-11de-9, and 83bd1d50-de13-11de-9.

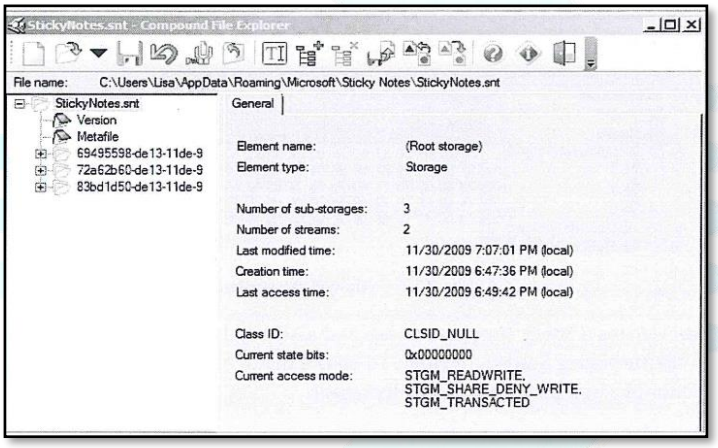


Figure 140 Sticky Notes in Compound file explorer

Figure below shows the content of storage 69495598-de13-11de-9 which contains three data streams. Notice the last modified time and created time are supposed to be reported as local time, but testing has shown issues with this, and it appears that the time is not being translated to local time. The time is showing as local, but is always UTC.

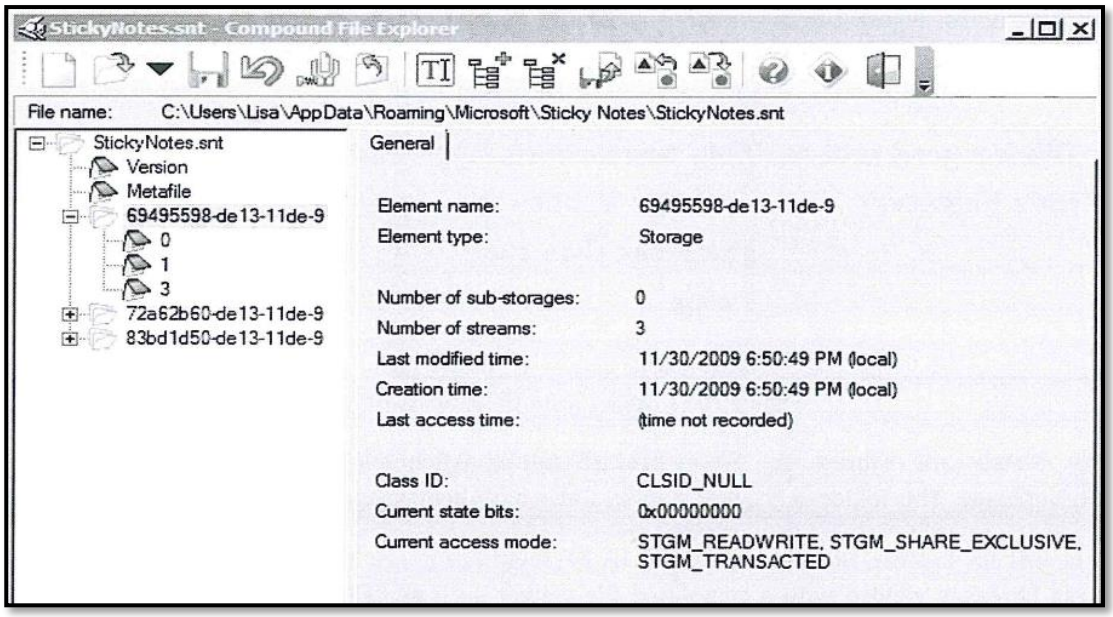


Figure 141 Sticky Notes Data Streams

Figure below shows the text entered on the actual Sticky Note by the user is found in stream 3.

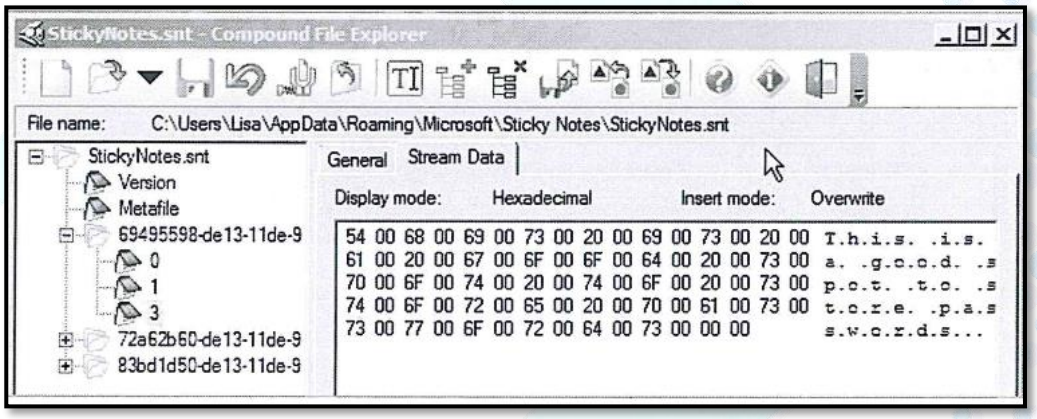


Figure 142 Sticky Note Content

When the user deletes a Sticky Note, its storage and associated data streams still exist but are not displayed to the Operating System. Figure below shows that if the Sticky Note .snt file is examined with a text editor like Notepad, the data is still present.

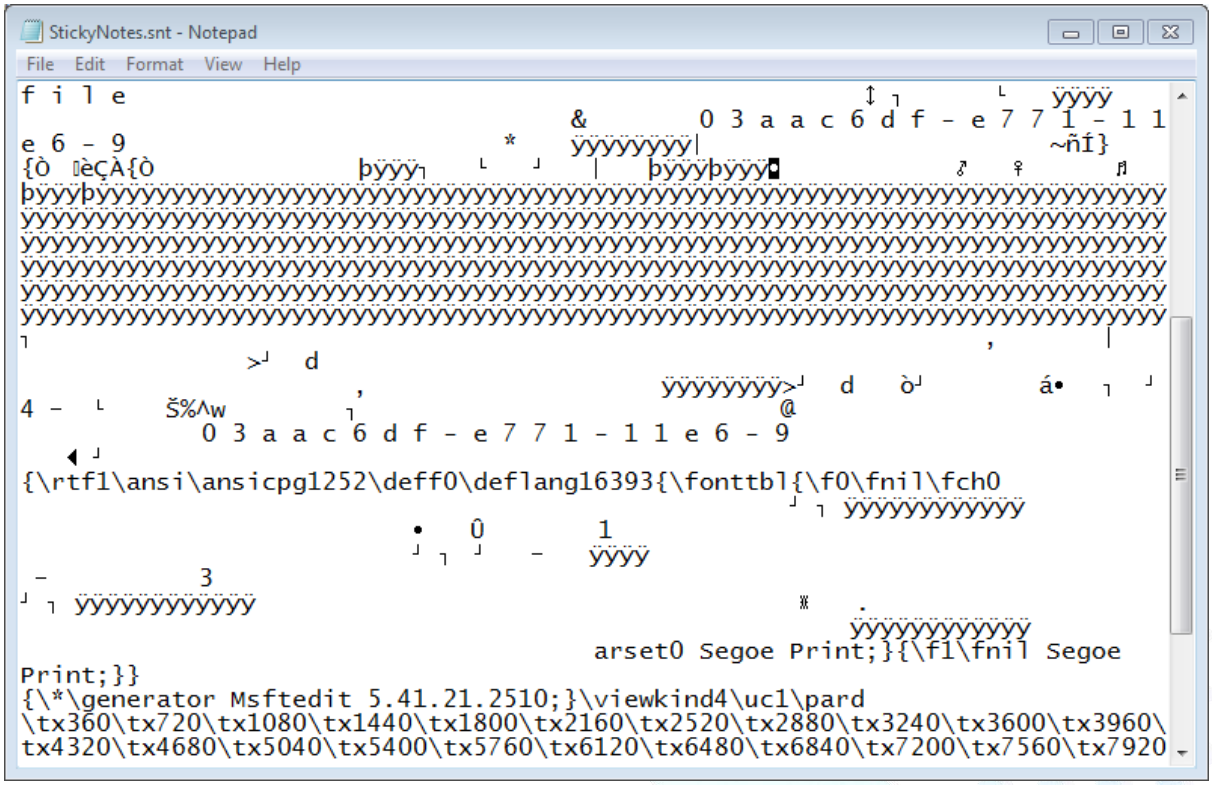


Figure 143 Remnant StickyNote content

This Sticky Note data will remain in the file until it is over written by a new Sticky Note. When viewing Sticky notes, it is important to look at the storage which has the RTF format, as this will have the entire note.

III. Event logs

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

Events are categorized into 3 classes: System, Application and Security. By default, in XP, all three logs are stored in the C:\Windows\system32\config folder and are called SecEvent.Evt, AppEvent.Evt and SecEvent.Evt respectively. These logs record errors, failures, successes, information and warnings.

- **SecEvent.Evt** - Contains log records of system processes and device drive activity. Event logs include things such as device drivers that fail to start or stop properly, hardware failures, duplicate IP addresses, and the starting/stopping/pausing of system processes.
- **AppEvent.Evt** - Contains log records of events related to the application software installed on the system. The events logged include errors, warnings, and any other information an application is designed to report. The developers of the application determine what gets logged.z
- **SecEvent.Evt** - Contains the events of the security processes used by NT, 2K and XP. Some of the security events that can be logged include changes in user privileges, logins and logouts, file and directory access, and printer activity.

Windows Vista, 7,8 AND 10

There are significant changes to Event Logs with Vista and 7 including their format and the restructuring into two main categories of Event Logs.

Event Logs now have the extension .EVTX and utilize the XML format. They are now stored in C:\Windows\System32\winevt\logs. The two main categories of Event Logs are now Windows Logs and Applications and Services Logs. Windows Logs still contain the logs that were available in XP but now include the logs Setup.EVTX and ForwardedEvents.EVTX.

- Windows Logs are much the same as those in Windows XP
- The Applications and Services Logs store events from a single application or component rather than events that might have system wide impact. The category subtypes found in this log include: Admin, Operational, Analytic and Debug logs. These logs are designed to aid IT Professionals using the Event Viewer to troubleshoot problems and are beyond the scope of this paper.
- Setup.EVTX - logs events that are related to application setup.
- ForwardedEvents.EVTX - stores events collected from remote computers with a created event subscription.

The Vista, 7, 8 and 10 event viewer is much more robust and has built in support that provides the definition of each log function as well as advanced options including filtering, sorting and custom view options.

Event Logs can be useful in a forensic examination to show that a user may or may not have performed a particular action at a particular time. They can be useful for a number of things from tracing logins in the case of logging into a restricted network, proving the computer was running during a particular time, showing time change/time change synchronization events, USB driver installation and wireless connections just to name a few monitored actions.

Figure below shows a manually time changed event. The time was changed by the user. Notice the values are recorded in UTC time.

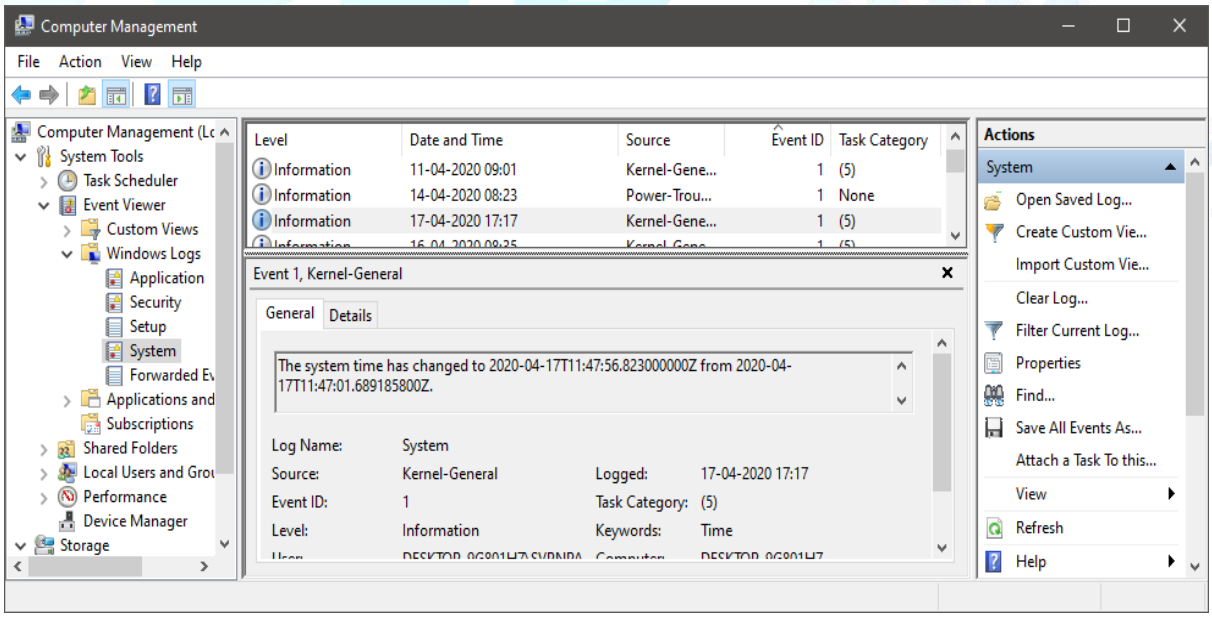


Figure 144 System Time change in System log

Some events may be shown in more than one log. Figure below shows the same time change event also captured in the Security log.

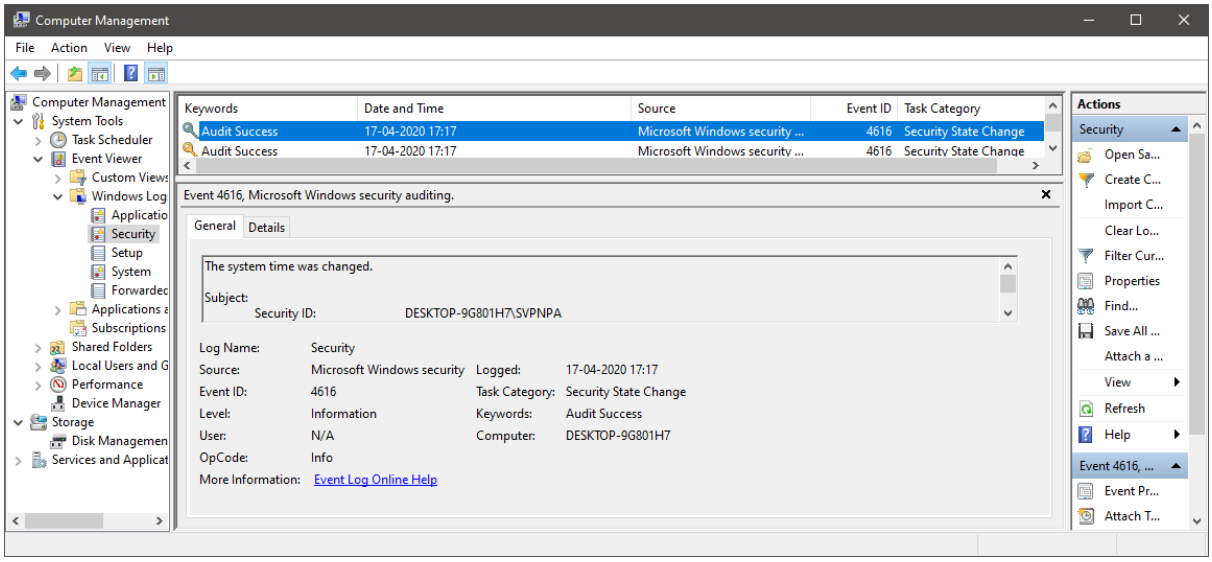


Figure 145 System Time change in Security log

To examine a Windows Vista + event log, copy the content of the folder C:\Windows\System32\winevt\logs to the forensic machine. The event logs can be accessed through Windows built in Event Viewer by selecting Action > Open Saved Log. The same precaution exists as Windows XP, the examiner's computer must be set to the same time zone as thesuspect computer as the Event Log is written in UTC. Event Log explorer is useful for *.evtx files and can export to spreadsheet for additional manipulation.

IV. Windows Registry

The Windows Registry can be an excellent source of potential evidence. Unfortunately, the Registry is still widely underutilized in day-to-day computer examinations. The registry can store system settings, hardware information, passwords, application cache and much more.

The Microsoft Computer Dictionary, Fifth Edition, defines the registry as:

A central hierarchical database used in Microsoft Windows 9x, Windows CE, Windows NT, and Windows 2000 used to store information necessary to configure the system for one or more users, applications and hardware devices.

The Registry contains information that Windows continually references during operation, such as profiles for each user, the applications installed on the computer and the types of documents that each can create property sheet settings for folders and application icons, what hardware exists on the system, and which ports are being used.

The Registry replaces most of the text-based .ini files used in Windows 3.x and MS-DOS configuration files, Such as the Autoexec.bat and Config.sys. Although the Registry is common for several Windows 'platforms, there are some differences among them.

See the Recommended Resources for relevant Knowledge Base articles from Microsoft.

The Registry contains three (3) major categories of data may be of evidentiary value.

1) **User Specific Information** — such as Desktop Preferences, Typed URLs, Messenger Contacts, MRU Lists and Passwords.
2) **System Specific Information** — such as Network Settings, Time Zone Information, Registered Owner details, Last Shutdown Date/Time and Hardware information.
3) **Application Specific Information** — such as File Associations, Application Registration Information etc.

The information stored within the registry is often difficult to find.

There are two distinct types of registries in use throughout all recent distributions of Windows, referred to in this document as 95 and NT respectively:

1) 95/98/98SE/ME
2) NT/2000/XP/VISTA/WINDOWS 7/WINDOWS 8/WINDOWS 10

In order to examine a suspect's registry, the first step is to identify which type of registry is in use. If you already know which operating system is being used, you can identify the registry type instantly. If not, the quickest way to identify the registry type is by understanding the files used by each registry type and their locations within the folder hierarchy.

The core registry files for both types of registries can be found in the Windows directory. The Windows directory is normally very easy to find but it is important to remember that this directory is customizable.

With that said, in 99.9% of cases, the Windows directory will be called either “WINDOWS” or “WINNT” andreside in the root directory of C drive. We shall refer to this directory as <windir> throughout this document.

Windows 95, 98, ME, XP, VISTA and WINDOWS 7, WINDOWS 8, and WINDOWS 10 use “WINDOWS” as the default Windows directory. Windows NT and 2000 use “WINNT” as the default directory. Another important directory is the profiles directory referred to as <profiles> from this point forward. This directory is also customizable; however, by default this directory will be called “<windir>/profiles” under Windows 95, 98 and NT. Under Windows ME, 2000, and XP, it will be called “Documents and Settings”. Windows Vista and higher has the profiles in the root directory in the “Users” subdirectory.

I. **Logical Layout**

We have established that Windows 95 and NT registries store their data in different files and these files have a different internal structure. This is the registry’s physical layout. The logical layout however, appears the same under both 95 and NT registries.

The layout of a Windows registry can be compared to a file system. The registry contains “keys” which can be compared to directories and “values” that can be compared with files. A key can contain any number of subkeys and any number of values. There are typically five (5) root level keys in a Windows registry:

- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CLASSES ROOT
- HKEY_CURRENT_USER
- HKEY_CURRENT_CONFIG

HKEY_LOCAL_MACHINE, referred to hereafter as HKLM, is by far the most significant key in the registry. HKLM has subkeys named Software, System, Security, Sam and Hardware. In Windows registries, these subkeys are each stored in a separate registry file with the exception of the Hardware subkey. This key is dynamically created when the system boots and is not written to disk at shutdown.

Each user of the system has their own user file containing all user-based settings and data. These files are stored in the user’s profile directory and are named NTUSER.DAT and USER.DAT under NT and 95 registries respectively. The HKEY_USERS key is the point where each of these user files is grafted to the logical registry layout.

HKEY_CLASSES_ROOT can be ignored because it is merely alias for the HKEY_LOCAL_MACHINE/Software/Classes subkey. The Classes subkey stores file extension associations.

HKEY_CURRENT_USER, referred to hereafter as HKCU is merely an alias for a subkey of HKEY USERS for the user currently logged on. When examining a registry of a system that has been shutdown there is no current user and therefore this key does not exist. For a running system, an application does not need to know how many users are on the system or which one is logged on. It can reference HKCU and the correct settings and data will be provided to it.

HKEY_CURRENT_CONFIG is an alias to the current hardware profile, which is stored at HKLM\System\CurrentControlSet\Hardware Profiles\Current. Please note, CurrentControlSet is also an alias and it points to the control set currently in use eg. HKLM\System\ControlSet001. By visiting HKLM\System\Select you can determine which is the current control set, default control set, faulty control set, and the last known good control set. These values are named Current, Default, Failed and LastKnown-Good, respectively.

Code	Type	Description
0	REG_NONE	Data with no particular type. This data is written to the registry by the system or applications and is displayed in hexadecimal format as a Binary Value.
1	REG_SZ	A fixed-length text string.
2	REG_EXPAND_SZ	A variable-length data string. This data type includes variables that are resolved when a program or service uses the data.
3	REG_BINARY	Raw binary data. Most hardware component information is stored as binary data and is
4	REG_DWORD or REG_DWORD LITTLE_ENDIAN	Data represented by a number that Is 4 bytes long (a 32-bit integer). Many parameters for device drivers and services are this type and are displayed in binary, hexadecimal, or decimal format.
5	REG_DWORD_BIG ENDIAN	The SameasREG_ DWORD_LITTLE_ENDIANexcept that DWORD_LITTLE_ENDIAN has the least significant byte at the lowest address and DWORD_BIG_ENDIAN has the least significant byte at the highest address.
6	REG_LINK	A Unicode string naming a symbolic link.
7	REG_MULTI_SZ	A multiple string. Values that contain lists or multiple values in a form that people can read are usually this type. Entries are separated by spaces, commas, or other marks.
8	REG_RESOURCE_LIST	A series of nested arrays designed to store are source list used by a hardware device driver or one of the physical devices it controls. This data is detected and written into the \ResourceMap tree by the system and is displayed in hexadecimal format.
9	REG_FULL_RESOURCE_ DESCRIPTOR	A series of nested arrays designed to store are source list used by a physical hardware device. This data is detected and written into the\HardwareDescription tree by the system and is displayed in hexadecimal format.
10	REG_RESOURCE_ REQUIREMENTS	A series of nested arrays designed to store a device driver's list of possible hardware resources it or one of the physical devices it controls can use, from which the system writes a subset into the\ResourceMap tree. This data is detected by the system and is displayed in hexadecimal format.

11	REG_QWORD	Data represented by a number that is a 64-bitinteger. This data is displayed as a Binary Value. It was first introduced in Windows 2000.

Table 17 Windows Registry

Physical Layout

The file hierarchy for 95/98/98SE/ME style registries is as follows:

File Location	File Description
<windir>/SYSTEM.DAT	The System file
<windir>/USER.DAT	The User file
<profiles>/<username>/USER.DAT	Zero or more User files

Table 18 File hierarchy for 95/98/98SE/ME style registries

A typical single user system will not have a profiles directory and all user settings will be stored in the mainUSER.DAT file. In this case, the only files necessary for a registry examination are SYSTEM.DAT andUSER.DAT from <windir>.

The file hierarchy for NT/2000/XP/VISTA/WINDOWS7/8/10 style registries is as follows:

File Location	File Description
<windir>/system32/config/SYSTEM	The System file
<windir>/system32/config/SOFTWARE	The Software file
<windir>/system32/config/SECURITY	The Security file
<windir>/system32/config/SAM	The Sam file
<windir>/system32/config/systemprofile/NTUSER.DAT	The User file
<profiles>/<username>/NTUSER. DAT	One or more User files

Table 19 File hierarchy for NT/2000/XP/VISTA/WINDOWS7/8/10 style registries

Window NT/2000/XP/VISTA/WINDOWS7-10 requires at least one user, the Administrator account. Even a single user system will typically have six user accounts, each with an NTUSER.DAT file, e.g., Darren Freestone, Administrator, Default User, All Users, LocalService, NetworkService, Software Tools

Before selecting an appropriate tool for exploring the Registry it's important to understand the distinction between Live and Non-Live Registries. A live registry is available only when Windows is up and running and contains volatile information that will be lost upon shutdown. Regedit can be used to explore a live registry but access to the **SAM** and **SECURITY** keys will not be allowed. When the machine is shut down and a copy of the hard disk is made, it is possible to locate and examine the contents of the individual registry files. This process is referred to as examining a Non-Live registry.

a) Regedit

All versions of Windows are distributed with a version of Regedit (regedit.exe or regedt32.exe), which allows the user to interface directly with the system registry. Regedit is provided for advanced users, and Microsoft warns that making changes can cause serious damage to your system. Regedit displays the keys in the left-hand pane in a hierarchical tree-view. The right-hand pane displays each value present in the currently selected key. For each value the user sees the value name, value type and value data.

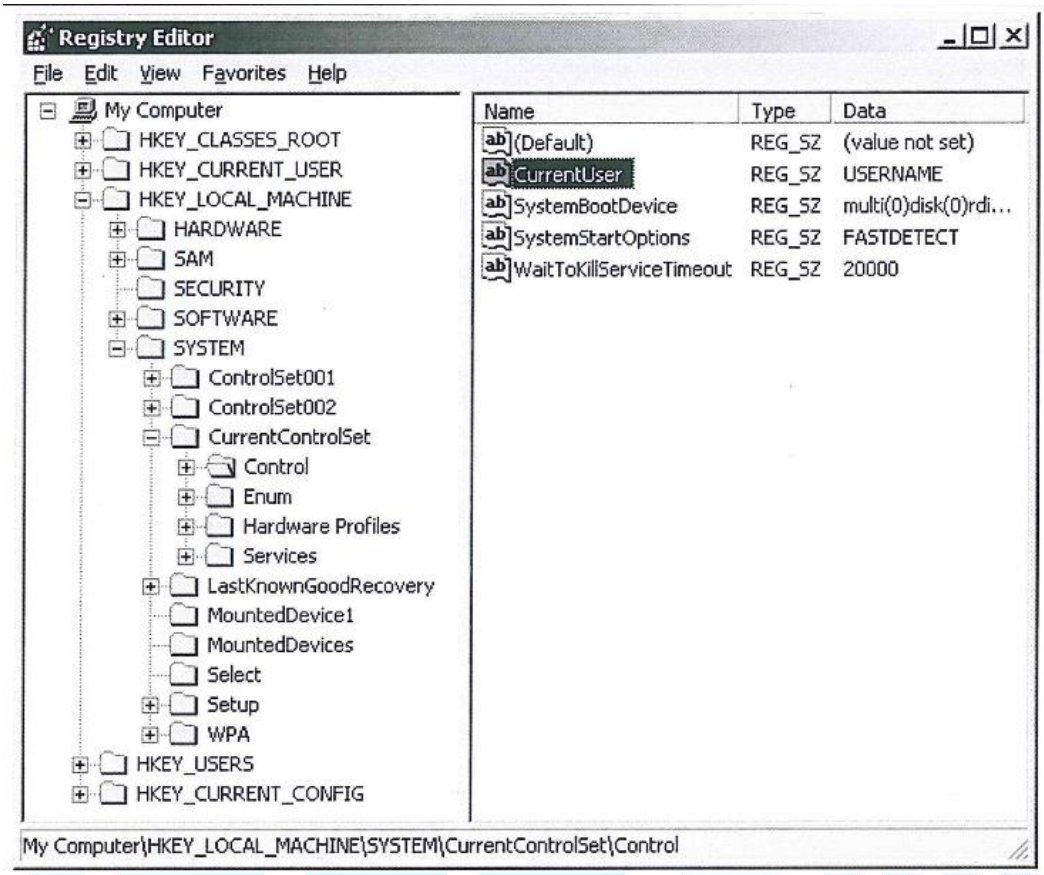


Figure 146 Regedit Home Screen

If you wish to conduct an examination of the registry of a live Windows system, you could use Regedit. This technique presents a few problems. The first problem is that you will not have access to the HKLM/Sam and HKLM/Security keys. This is for security reasons. Another problem with using Regedit is you will have difficulty extracting important information into report form.

Another major reason for not using Regedit to examine the system registry is that in most cases the registry you are examining will NOT be live, i.e., you will have a disk image of the Windows file system, including the registry files, but you will not actually boot the system. If you perform a restore operation and boot the disk image, then using Regedit would be a possibility.

b) Registry Browser

Registry Browser is a WIN32 application, designed to work a NON-LIVE windows registry. It should be installed onto your lab machine. It supports both 95 and NT registries. RB requires to the Windows directory <windir> of the target system in order to function.

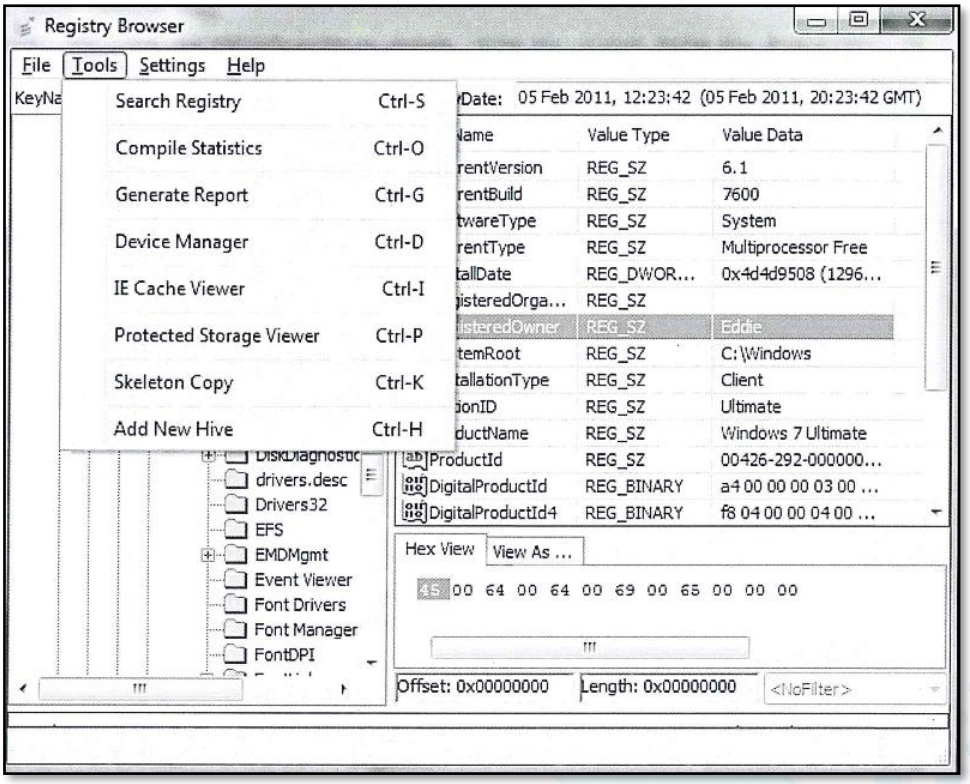


Figure 147 Registry Browser Home Screen

If your forensic software does not allow third party applications like RB to directly access the files and folders of the imaged file system, then you should familiarize yourself with the process of exporting registry files so that they can be made available to RB or other third-party applications.

During the exporting process, you must ensure the original directory structure is maintained so that the result is a skeleton of the <windir> and <profiles> directories. See pictured Windows 10 example

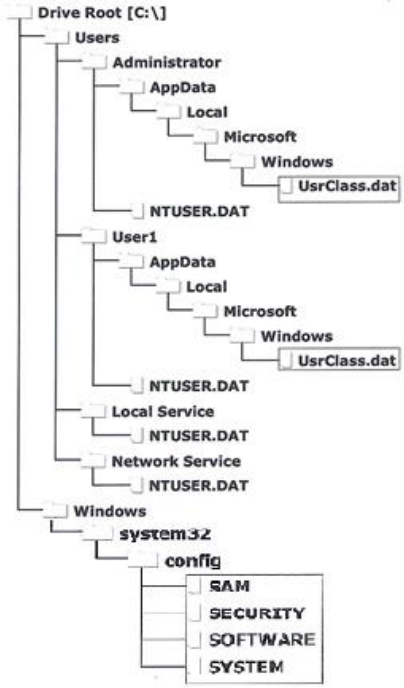


Figure 148 Registry

Registry Browser has a similar screen layout to Regedit and is therefore simple to use. It has superior searching options and a predefined yet customizable reporting function. The report uses a template file, which can be customized by the user to include new keys and values without needed an updated version of the application.

c) RegRipper

RegRipper is created and maintained by Harlan Carvey. RegRipper is a Windows Registry data extraction and correlation tool. RegRipper uses plugins (similar to Nessus) to access specific Registry hive files in order to access and extract specific keys, values, and data, and does so by bypassing the Win32API. RegRipper isn't a registry Viewer, but a collection of scripts used to extract certain items of interest from thehive files. RegRipper, being written in Pearl, makes it easy to add new items to extract from the hives by writing a short script. Examiners can write custom scripts to locate specific item during examinations.

```
----------------------------------------
RecentDocs - recentdocs
**All values printed in MRUList\MRUListEx order.
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
LastWrite Time Sun Apr  3 00:38:25 2011 (UTC)
  14 = RegRipper032911
  13 = ripxp_instructions.txt
  6 = Practical
  9 = System_Rip.txt
  12 = Security_Rip.txt
  10 = Security_Rip.log
  3 = Downloads
  8 = agenda5.doc
  7 = Software_Rip.txt
  2 = RegRipper032911.zip
  1 = 1033
  0 = Blog.dotx
  11 = Pictures
  48 = captain_america_first_avenger_by_delta_seb-d39e1ps.jpg
  19 = IAAS 335
  21 = ahrendtHW4.docx
  47 = My Kindle Content
  46 = B003ODIZL6_EBOK.azw
  45 = SIFT Workstation 2.0 Distro Version
  44 = SIFT Workstation 2.0.vmdk
  43 = ahrendtHW5.docx
  27 = FreeBSD-8.2-RELEASE-i386-dvd1
  23 = FreeBSD-8.2-RELEASE-i386-dvd1.iso
  18 = ahrendtHW3.docx
  5 = agenda4.doc
  4 = readme.txt
  4294967295 =
```

Figure 149 Sample RegRipper Output 1

```
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.avi
LastWrite Time Sat Apr  2 23:40:19 2011 (UTC)
MRUListEx = 4294967295
  4294967295 =

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.azw
LastWrite Time Sat Apr  2 23:22:28 2011 (UTC)
MRUListEx = 0,4294967295
  0 = B003ODIZL6_EBOK.azw
  4294967295 =

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.doc
LastWrite Time Sun Apr  3 00:22:50 2011 (UTC)
MRUListEx = 1,0,4294967295
  1 = agenda5.doc
  0 = agenda4.doc
  4294967295 =

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.docx
LastWrite Time Sat Apr  2 23:24:14 2011 (UTC)
MRUListEx = 1,2,0,4294967295
  1 = ahrendtHW4.docx
```

Figure 150 Sample RegRipper Output 2

d) Timezones

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformtion

Windows NT based registries store a modified date for each registry key. These dates, like many of the dates encountered, are stored in Greenwich Mean Time (GMT). In order to convert these GMT times tolocal time, the registry's Time Zone Information needs to be determined.

TimeZone Values	Explanation
Bias	signed 4-byte value, standard offset to GMT (minutes)
StandardBias	signed 4-byte value, standard time adjustment (minutes)
StandardName	string value, time zone description — standard time
Standardstart	16-byte structure defining start date for standard time
DaylightBias	signed 4-byte value, daylight time adjustment (minutes)
DaylightName	string value, time zone description — daylight time
DaylightStart	16-byte structure defining start date for daylight time
ActiveTimeBias	signed 4-byte value, offset currently in effect (minutes)

Table 20 Timezone Values

The values ***Bias, Standard Bias*** and ***ActiveTimeBias*** are all 4 byte signed values measuring minutes behind GMT time. For example, the time zone -5 GMT would be stored [+300](minutes). The time zone +10 would be stored as [-600] minutes.

To calculate the local time, during standard time, use the formula:

Local Time = GMT - Bias - StandardBias

To calculate time local time, during daylight time, use the formula:

Local Time = GMT - Bias - DaylightBias

The **ActiveTimeBias** value is updated to reflect the current offset to GMT (eg.**Bias + DaylightBias**). This value will change after the trigger dates **StandardStart** and **DaylightStart**. These trigger dates are stored as a 16-byte structure, made up of six, two-byte values as follows: Year, Month, Day of Week, Day, Hour, Minute, Second and Milliseconds. These dates do not explicitly express a given date. Insteadthey express a rule for determining the date, e.g., Last Sunday in October at 2am.

e) Hardware Devices

There is a lot more to the Windows Registry than just program settings, cache and data. A lot of information about the system's hardware is stored in the registry as well, particularly for NT based registries. These hardware keys are located at HKLM\System\CurrentControlSet\Enum and therefore can be viewed or searched with Regedit or other registry viewing software. This can be a little problematic as the information is not laid out in a human friendly way.
A more effective method of viewing these keys is RB's Registered Device Manager, or RDM. The RDM Interface is just like the Device Manager function of Windows except that it shows all hardware registered with the system, not just the devices currently attached (Figure 5.69). With this feature this user can identify hardware devices such as a thumb-drive and determine when it was first attached to the system.

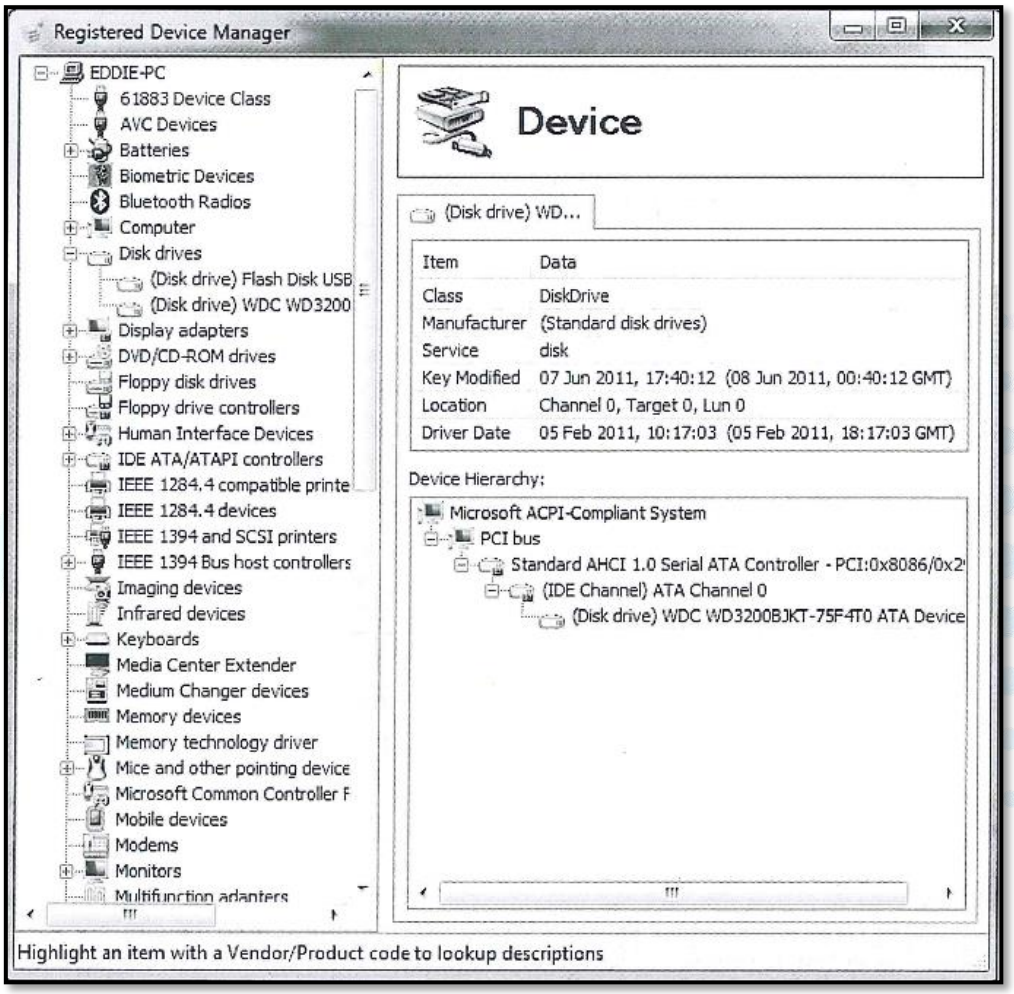


Figure 151 Registered Device Manager

f) Security Identifiers (SIDs)

When examining an NT style registry, or even the file system it resides on, you will encounter Security Identifiers (commonly abbreviated SIDs). SIDs is an alphanumeric character string which is by a Windows Domain Controller during the log-on process. The SID uniquely identifies the user or group from all other users and groups on the network.
The following is an example of the SID S-1-5-12-4626982574-2357336844-2345214626-11114 broken down into its components:

Denotes a SID	Revision Level	Authority Valve	Domain or Computer Identifier	Relative ID RID
S	1	5	12-4626082574-2357336844-2345214626	1014

There are many fixed SIDs common to all systems. A few examples are listed below:

Security Identifier	Username	Account Description
S-1-5-18	System	Powerful service account used by the OS
S-1-5-19	Local Service	For running services locally
S-1-5-20	Network Service	For running services over the network.
S-1-5-21-domain-500	Administrator	System Admin's account with full control.
S-1-5-21-domain-501	Guest	A guest account - disabled by default

Table 21 SIDs common to all systems

When examining a regular system, the most important part of the SID to look at is the last portion of digits, the Relative ID (RID) It is important to be able to relate an identified SID/RID back to a user account name. The easiest way is as follows:

- Navigate to:
  HKEY_LOCAL_MACHINE\Sam\SAM\Domains\Account \Users \Names
- Scroll through the usernames within this key. eg. Administrator, Guest whilst looking at the “Value Type" column.
- The value in the value type column is the Relative ID (RID) of that user.
- So, once the RID has been found the account name has been identified.

The subfolders in the Recycle Bin folder are named by SID with the respective RID. In the scenario where a deleted file of particular relevance is located within one of those sub folders, the first step will be to identify which user account the SID refers.

g) Windows 8 Registry Changes

The release of Windows 8 additional artifacts added to the Registry which contain items of interest to an examiner, which are carried over in Windows 10. The location and structure of the hive files remains the same but additional keys have been added in support of the User Interface.

Within each user's NTUSER.DAT file contained within their profile is a key which can identify typed URLs for Internet Explorer, along with the time this information was entered.

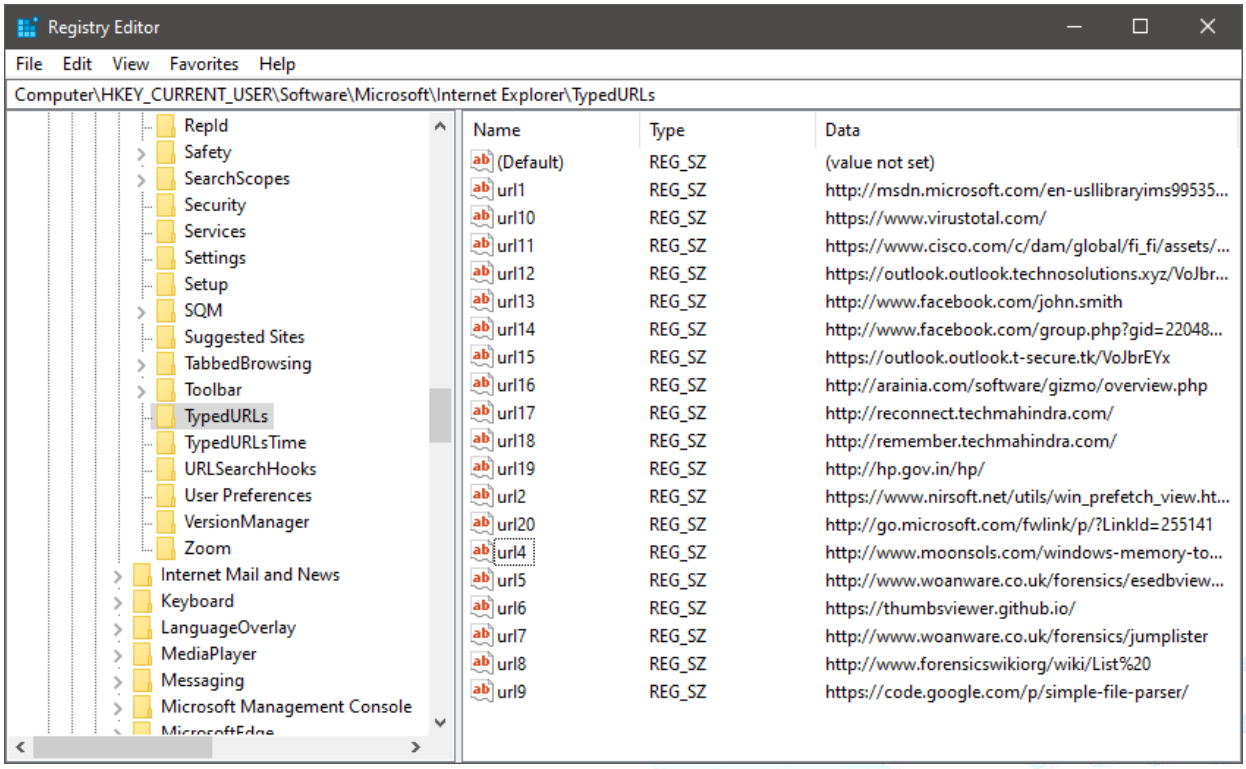


Figure 152 SID of User\Software\Microsoft\Internet Explorer\TypedURLs

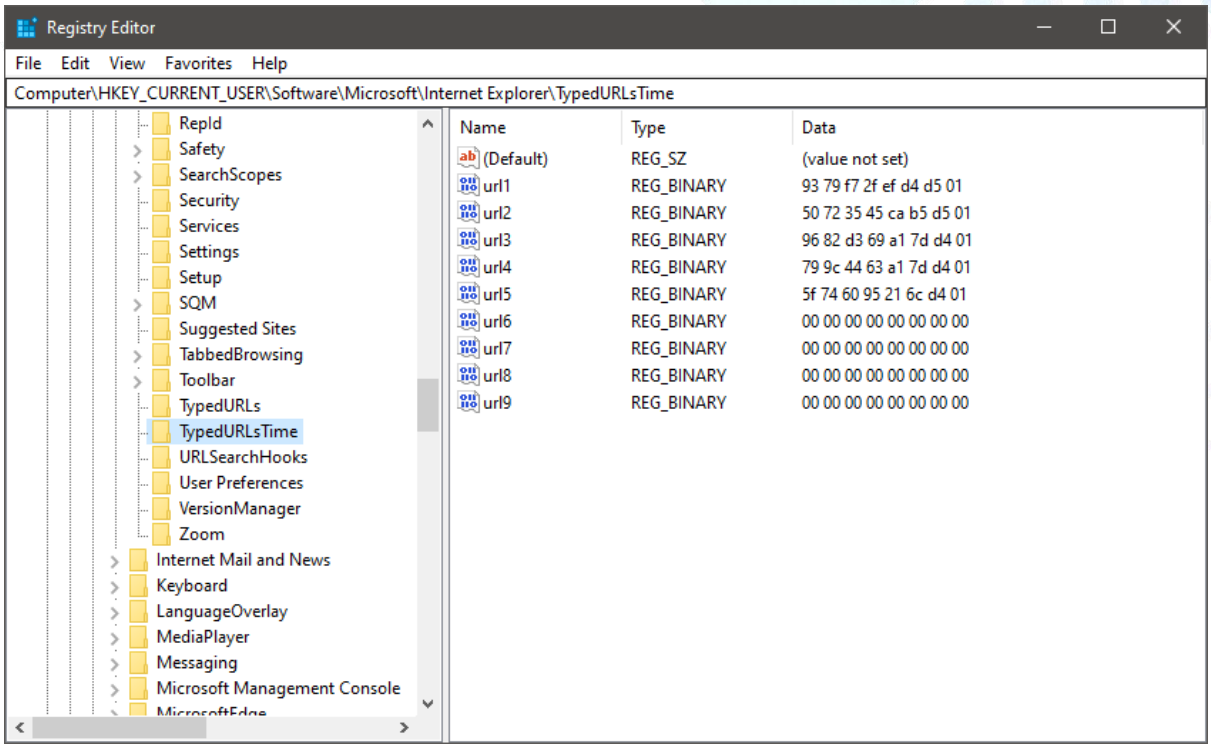


Figure 153 SID of User\Software\Microsoft\Internet Explorer\TypedURLs

This information is stored in Windows FILETIME format.
The SAM file contains the hive which lists each user's Internet User Name associated with their Windows Live account.
If the user has a "local" type of account, their Microsoft account used for the App store is found in the User's NTUSER.DAT file.

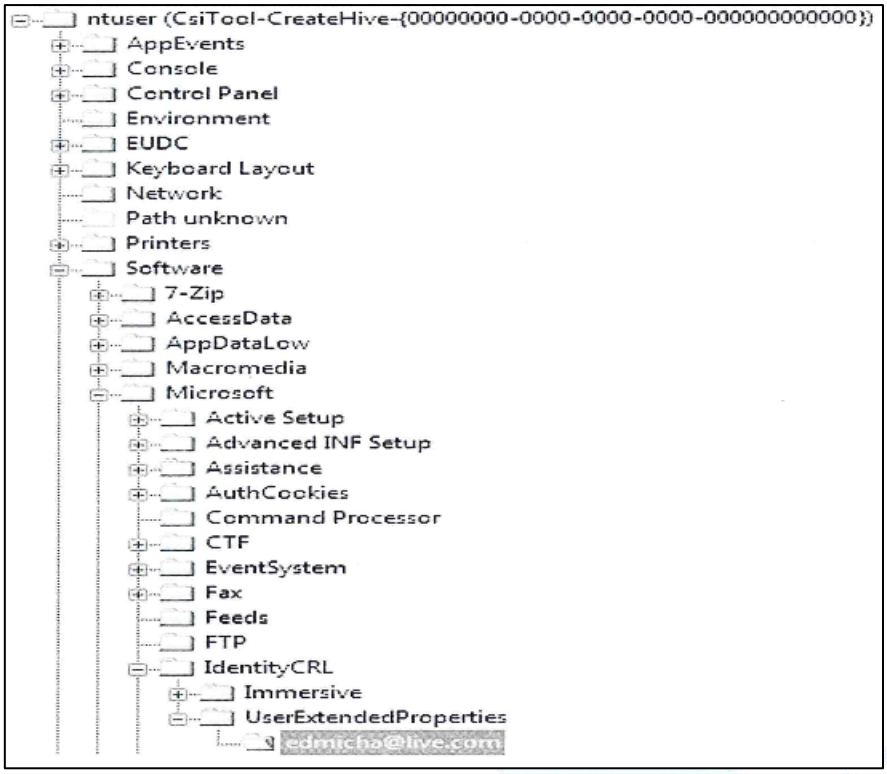


Figure 154 NTUSER.DAT file

The software key contains additional information about Metro applications installed on the system through the App store Microsoft has made available to Windows 8 users.

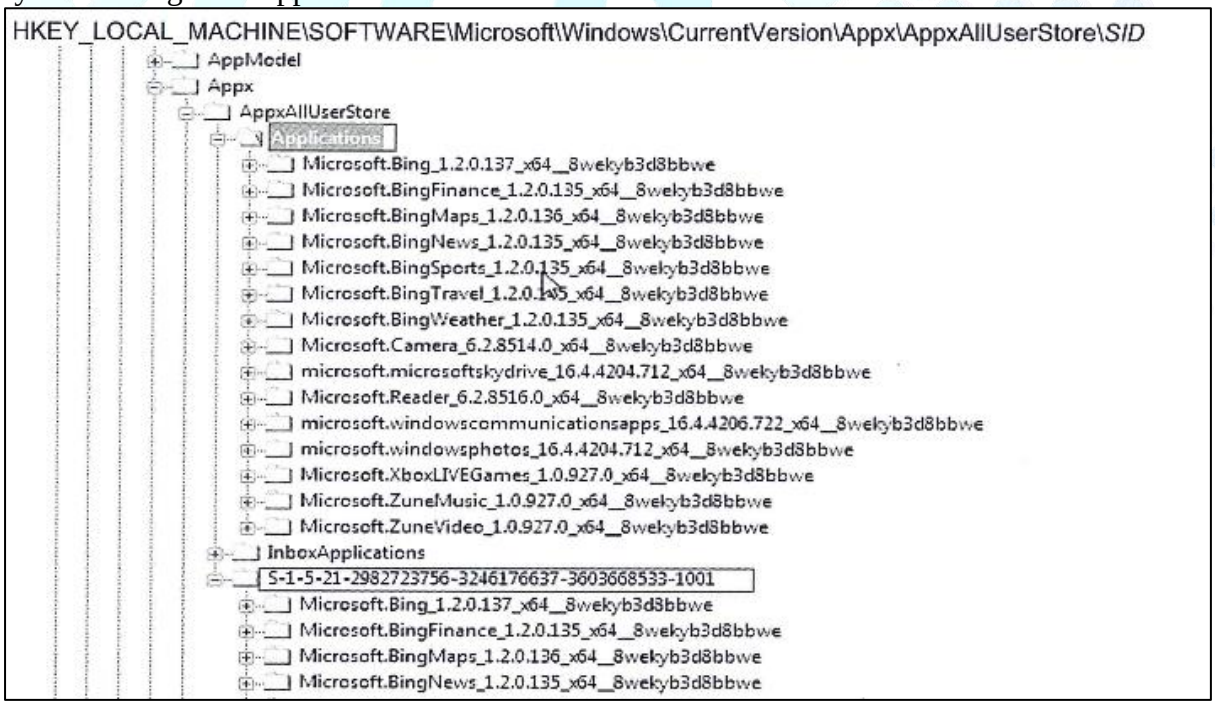


Figure 155 Information about Metro applications

Registry Keys & Values — Examples

HKLM\Software\Microsoft\Windows NT\currentVersion

- Registered Owner
- RegisteredOrganization

- Productid
- CurrentVersion
- CurrentBuildNumber
- ProductName
- InstallDate (UNIX time_t date format)

HKLM\System\CurrentControlset\control\Windows

- ShutdownTime (FILETIME date format)

HKLM\system\CurrentControlSet\Control\TimeZoneInformation

- ActiveTimeBias
- Bias
- DisableAutoDay1ightTimeset
- StandardName
- StandardBias
- standardstart
- DaylightName
- DaylightBias
- Daylightstart

HKCU\Software\Microsoft\Windows\CurrentVersion\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnceEx

These keys provide a listing of applications that start at boot

HKLM\Software\Microsoft\Windows\CurrentVersion\internet Settings\cache\Paths
HKLM\Software\Microsoft\Windows\CurrentVersion\internet Settings\Cache\Content
HKLM\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Cookies
HKLM\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\special Paths\Cookies
HKLM\Software\Microsoft\Windows\CurrentVersion\InternetSettings\Cache\History
HKLM\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\special Paths\History
HKLM\Software\Microsoft\Windows\CurrentVersion\internet Settings\URL History

- Daystokeep

HKLM\system\CurrentControlset\Services\Tcpip\Parameters\Interfaces\ [keyname]

- EnableDHCP (yes/no)
- IPAddress
- SubnetMask
- DhcpIPAddress
- DhcpSubnetMask
- DhcpServer
- LeaseObtainedTime (UNIX time_t date format)

HKLM\System\CurrentControlset\Control\Print\Printers\ [keyname]

- Name
- Port
- Printer Driver

HKLM\Software\Microsoft\Windows\CurcentVersion\Uninstal1\ [keyname]

- Add/Remove programs section - stores a record of each program installed on the computer.
- DisplayName
- Uninstallstring
- DisplayVersion
- InstallDate

- Publisher
- EstimatedSize
- InstallSourcenstallsource
- ModifyPath

HKCU\Control Panel\Desktop

- ScreenSaveActive
- ScreenSaverIsSecure
- ScreenSaveTimeOut
- wallpaper
- OriginalWallpaper
- SCRNSAVE.EXE

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

Shows folder redirection e.g: A change of location for the My Documents folder.

HKCU\Software\Microsoft\MediaPlayer\Player\Settings

- SaveAsDir
- OpenDir

HKCU\Software\Microsoft\MediaPlayer\Player\RecentFileList

List of recent files played in Media Player

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU

Most recent commands entered in the Run box

HKCU\Software\Microsoft\Windows\CurrentVersion\Applets\Regedit

LastKey May show if the user may have or was trying to tweak the registry

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

List of the most recent files opened by file extension

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePIDMRU (Vista/Win7)

List of the most recent files opened or saved listed by file name and extension.

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComD1g32\LastVisitedMRU

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComD1g32\LastVisitedPIDMRU (Vista/Win7)

Most recent files opened in Windows: includes filename and application

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Wallpaper\MRU

HKCU\Software\Microsoft\Internet Account Manager

Stores the user's Internet account lists

Stores information for Outlook/Outlook Express mail

HKCU\Software\Microsoft\Internet Account Manager\Accounts

Stores the user's POP3 saved passwords.

HKCU\Software\Microsoft\Internet Explorer\TypedURLs

Listing of URL sites manually entered or accessed via a link

HKLM\System\CurrentContolSet\Enum\USB

USB devices connected to the computer

HKLM\System\CurrentControlSet\Enum\USBSTOR

USB storage devices connected to the computer

HKLM\System\MountedDevices

Listing of current and past mounted devices connected that used a drive letter

HKCU\Software\ Computing\Winzip\filemenu

HKCU\Software\ Computing\Winzip\extract
HKCU\Software\NComputing\Winzip\rrs

Opened
Date
Days

- xdyx
- xmox
- xyrx

4. Decryption of BitLocked Device

For performing decryption of the bitlocked drive we need two tools:

- john-1.9.0-jumbo-1-win64
- hashcat-6.2.2

However successful decryption will be subject to the password length, system resources available and the password dictionary that is being used for decryption purpose. Following steps can be followed to decrypt the bitlocker encrypted drive.

a) Create Image of the bitlocked pendrive using FTK Imager
b) Download the too "john-1.9.0-jumbo-1-win64" and open the following path:
c) "Bitlocker Decryption\john-1.9.0-jumbo-1-win64\run" and type 'cmd' on address bar

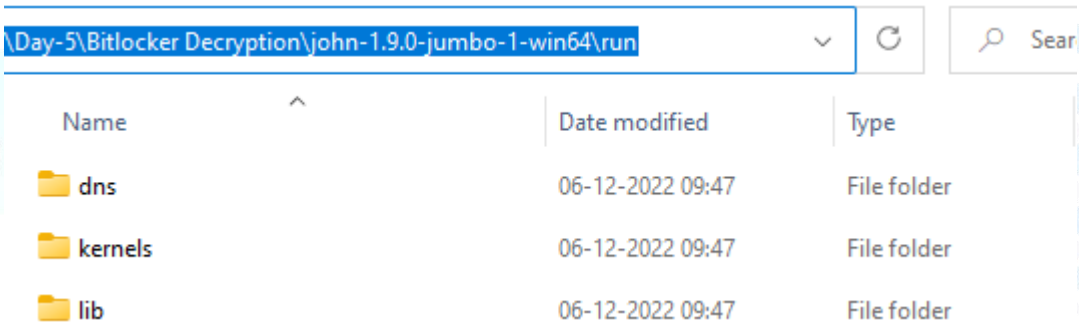


Figure 156 Path to john tool

d) Type "bitlocker2john.exe -i "<path-to-file>" on cmd prompt.
(The above step would fetch the hash values of this bit locked drive)

bitlocker2john.exe -i "C:\Users\svpnpa\Desktop\Day-5\Bitlocker Decryption\Bitlocked_Drive_4008.001"

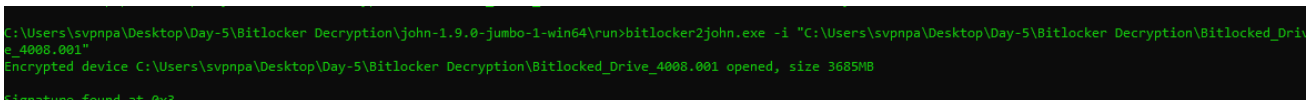


Figure 157 bitclocked disk

e) Store the hashes in a file named *allhashes.txt*.

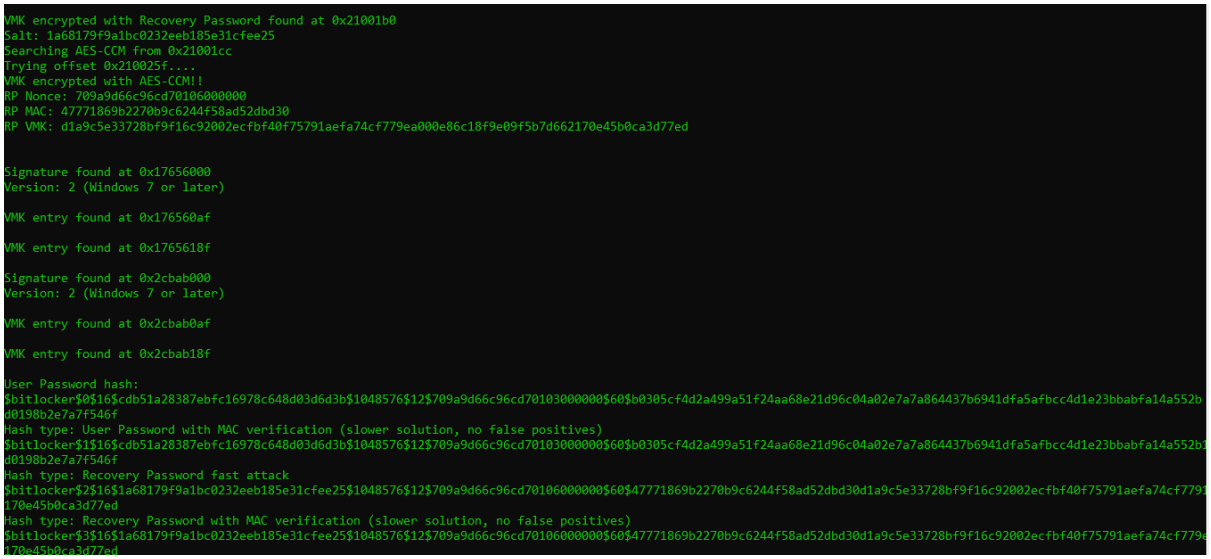


Figure 158 hash of bitlocked disk

f) Select hash value starting with $bitlocker$1$ and save it in file named as *hash.txt* at this path. **C:\Users\svpnpa\Desktop\Day-5\Bitlocker Decryption\hashcat-6.2.2**

$bitlocker$1$16$cdb51a28387ebfc16978c648d03d6d3b$1048576$12$709a9d66c96cd70103000000$60$b0305cf4d2a499a51f24aa68e21d96c04a02e7a7a864437b6941dfa5afbcc4d1e23bbabfa14a552b1b4cd0bb34fbeb8145b21958d0198b2e7a7f546f

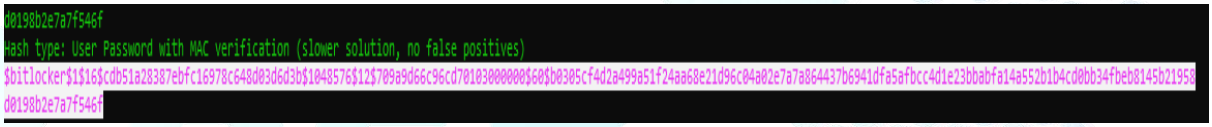


Figure 159 $bitlocker$1$

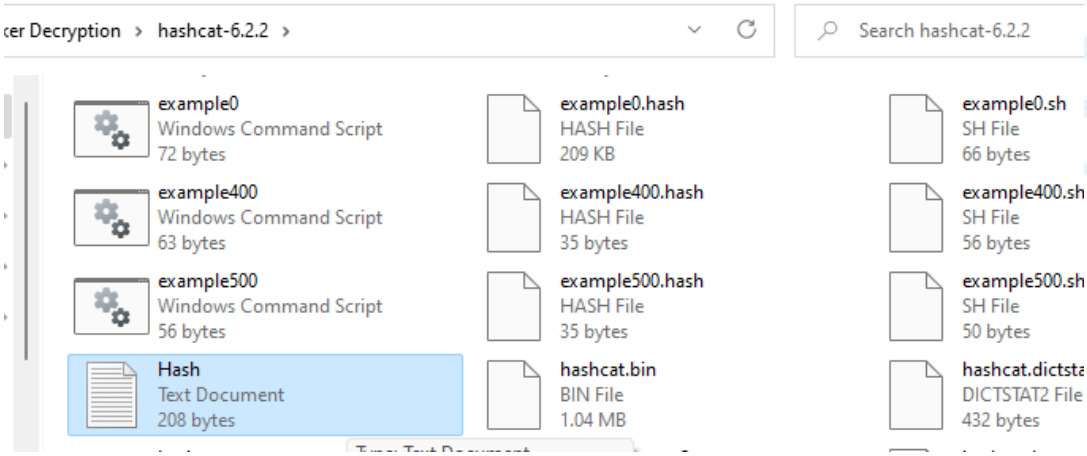


Figure 160 hash.txt

g) Navigate to hashcat-6.2.2 folder and type ‘cmd’ on it.

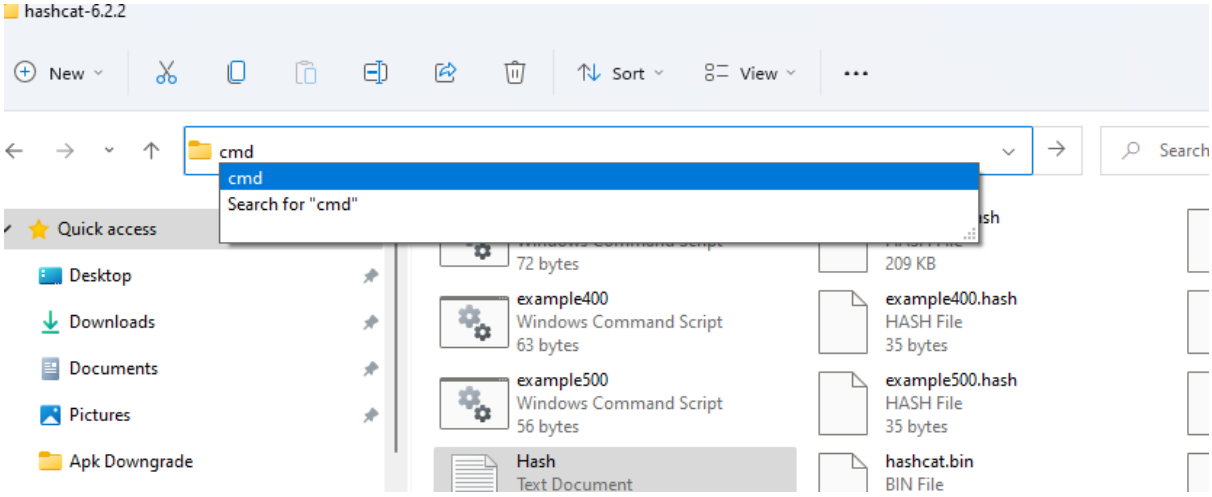


Figure 161 Open Command Prompt

h) Command Prompt will open up.

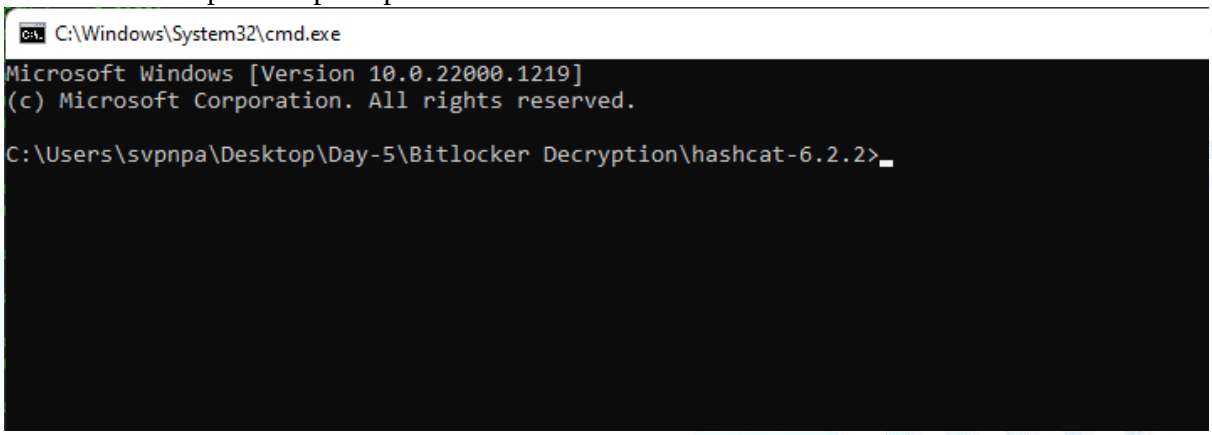


Figure 162 Command prompt

i) You have also been provided with a dictionary file of passwords named as *passes.txt* that may be used to perform dictionary attack on the hash values identified in earlier step.

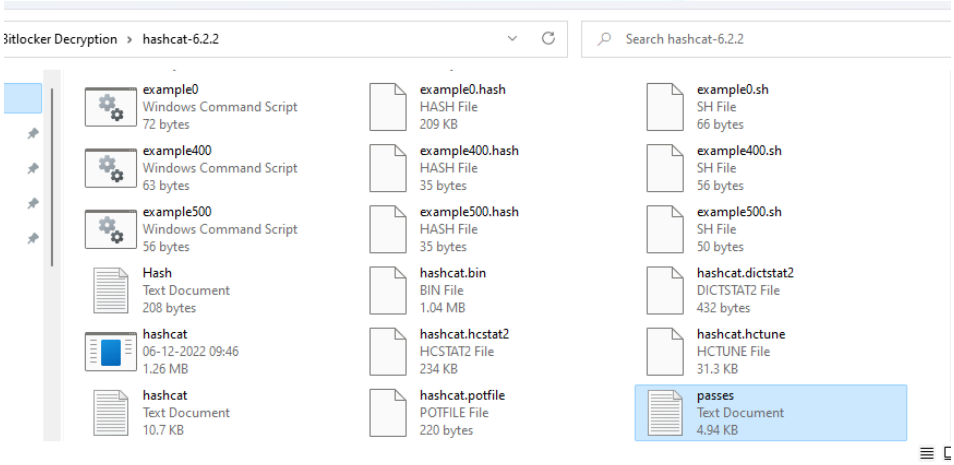


Figure 163 passes.txt

j) On the Command Prompt type “hashcat.exe -m 22100 hash.txt passes.txt --show”

```
C:\Users\svpnpa\Desktop\Day-5\Bitlocker Decryption\hashcat-6.2.2>hashcat.exe -m 22100 hash.txt passes.txt --show
$bitlocker$1$16$cdb51a28387ebfc16978c648d03d6d3b$1048576$12$709a9d66c96cd70103000000$60$b0305cf4d2a499a51f24aa68e21d96c0
1b4cd0bb34fbeb8145b21958d0198b2e7a7f546f:kill2hack
```

Figure 164 Hash Matching with password hash

Password of the Encrypted PenDrive is kill2hack.

k) Verify whether the password is correct or not. First Install OSFMount Tool and mount the image

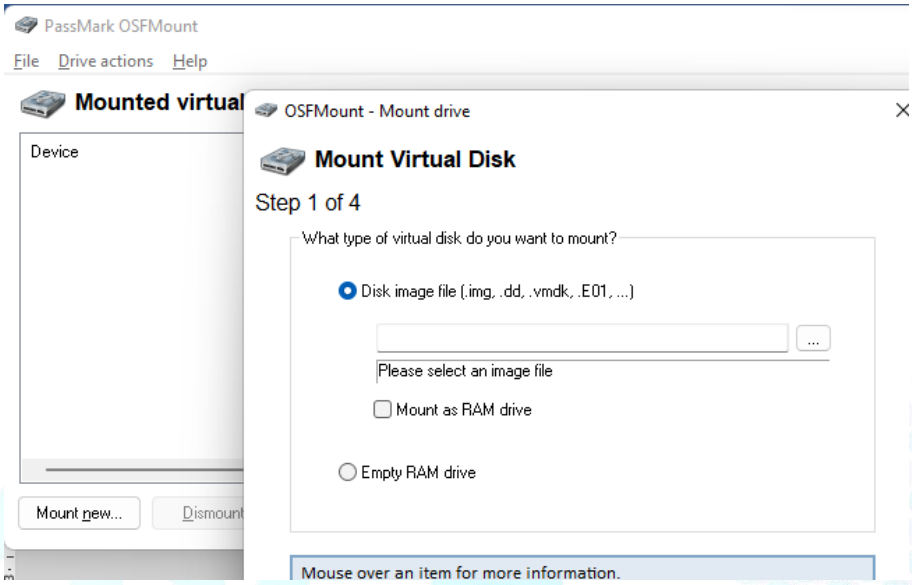


Figure 165 Verifying Password

l) And mount the image as Physical Drive

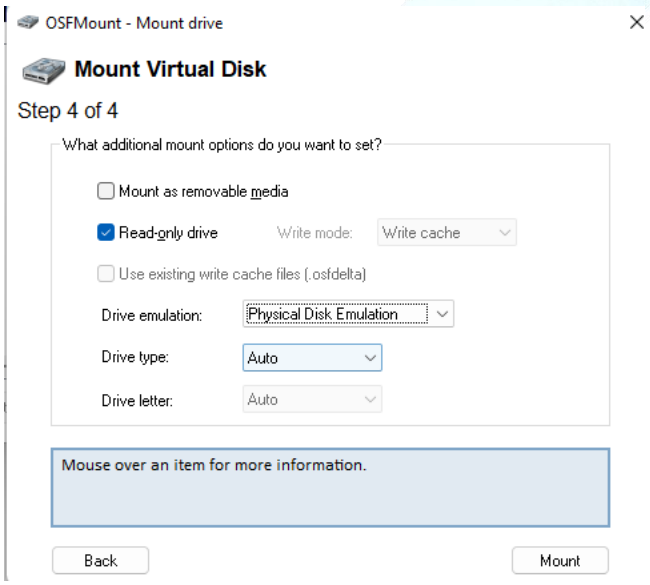


Figure 166 mount drive

m) Physical Drive will appear

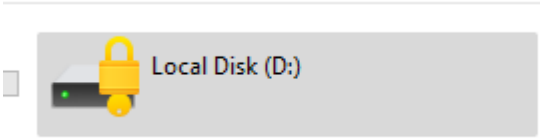


Figure 167 Mounted drive

n) Enter the password

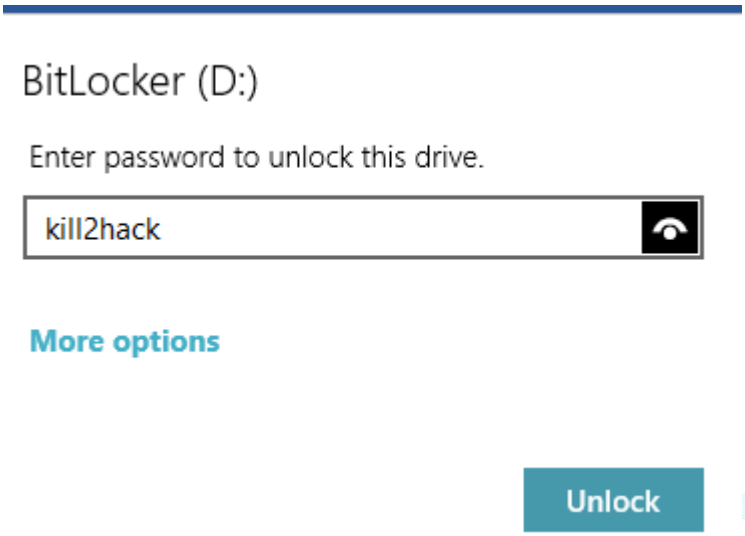


Figure 168 Enter password

o) The Drive will be decrypted

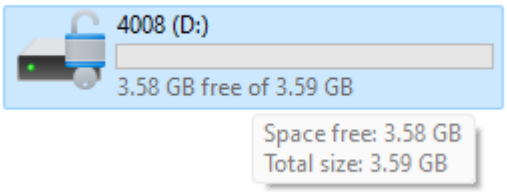


Figure 169 Decrypted drive

5. Pattern Bypass of the Android Phone

Problem Statement:

You receive an android-based mobile device that is in ON and locked condition, specifically protected by a pattern lock.

Possible solutions:

1. Request the accused for password. (*Good luck!*)
2. Use password-reset mechanisms
   (*Beware!* These may also result in complete wipe of data.)
3. Forensic Tools to the rescue!

We tried connecting the device to UFED, a mobile forensic tool that supports various types of acquisitions.

At first, we tried physical acquisition so that in case we aren't able to bypass password, at least we get the entire data that may be used as evidence.

Unfortunately, this type of acquisition required root access of the device, and the device that we had was not rooted.

Next option, please!

We proceeded with the second most significant option of data acquisition, i.e File System Acquisition. This type of acquisition claims to acquire some important data from the device even while the system is in locked condition.

We then followed the sequence of connection as directed by the tool, and proceeded further, as follows:

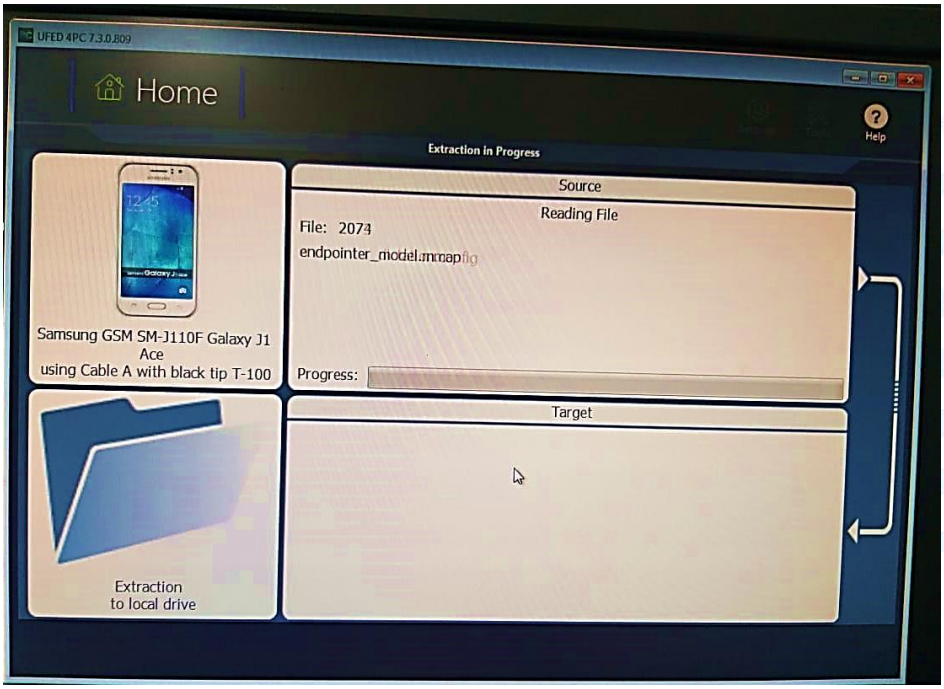


Figure 170 File System Acquisition

Incidentally, out of so many random files which were seemingly of no use to us, we found a file gesture.key also acquired from the device. We had information that android stores the pattern information in a file named as gesture.key

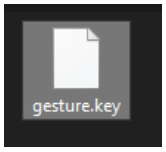


Figure 171 gesture key found in File System Acquisition

Next, we tried opening the file using a generic text editor hoping to find some useful information.

All we could see is some gibberish content.

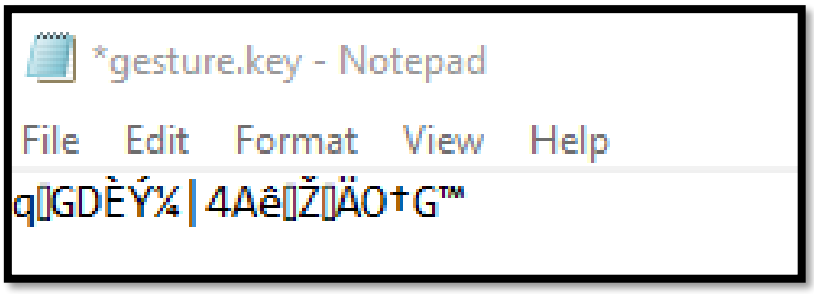


Figure 172 content of gesture key if opened in notepad

After some research, we found that the lock sequence/pattern information is encrypted with **SHA1** hashing algorithm. Now, we tried figuring out the approach towards identifying the decrypted string and discovered a java program ***DecodeAndroidGesture***, available on GitHub for the same.

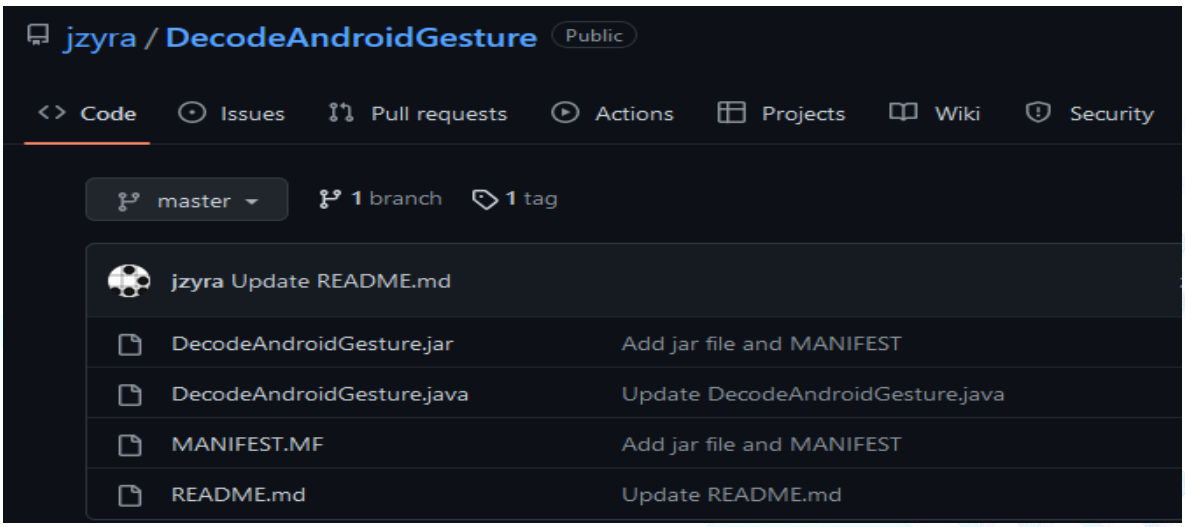


Figure 173 DecodeAndroidGesture - Github

The above tool was downloaded, and stored at the path where we had the gesture.key file, and was executed.

Figure 174 Execute the tool in path where gesture key is stored

Upon execution, an output containing a sequence was displayed as follows:

```
D:\Arif Cases\Telangana\Mobile Forensics>java -jar DecodeAndroidGesture.jar gesture.key
[+] Searching...
[+] Sequence: 21034785
```

Figure 175 Sequence of the pattern decoded

Next, we checked the dialpad of one of our phones to see the sequence of numbers.

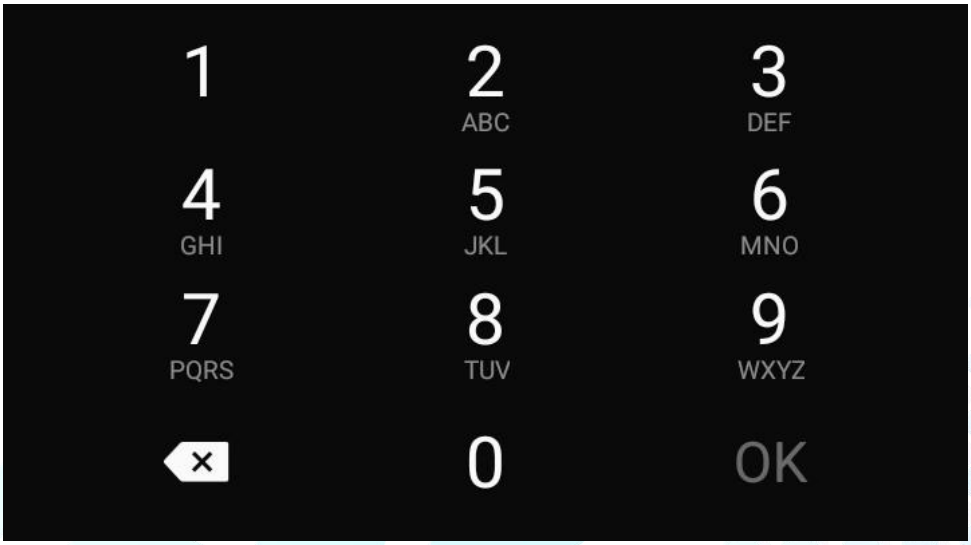


Figure 176 try matching the sequence on the keypad

If you notice, the sequence/pattern identified by using the program was not possible using this layout, as 2-1-0 was not a feasible approach without having to traverse through either 4->7, or 5->8.

What next? Having realized that the above approach wasn't possible, we tried making possible approaches with a change in layout.

A layout beginning with 0 was designed to see if the same combination was feasible this way.

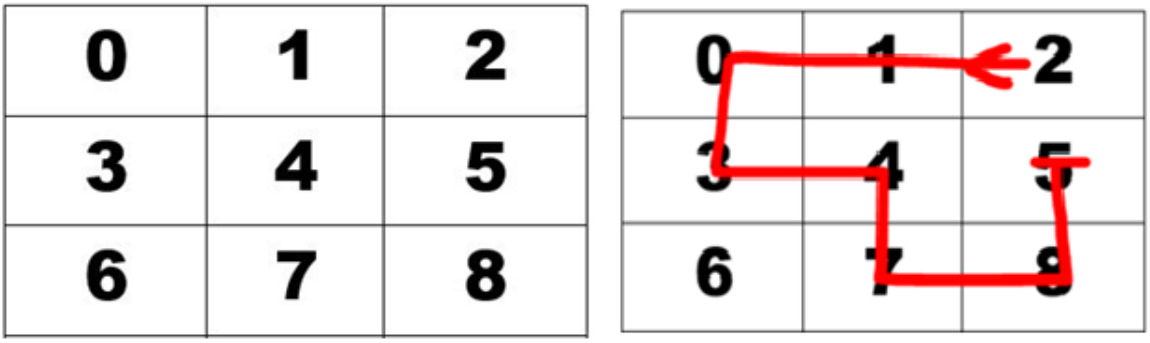


Figure 177 Pattern Found

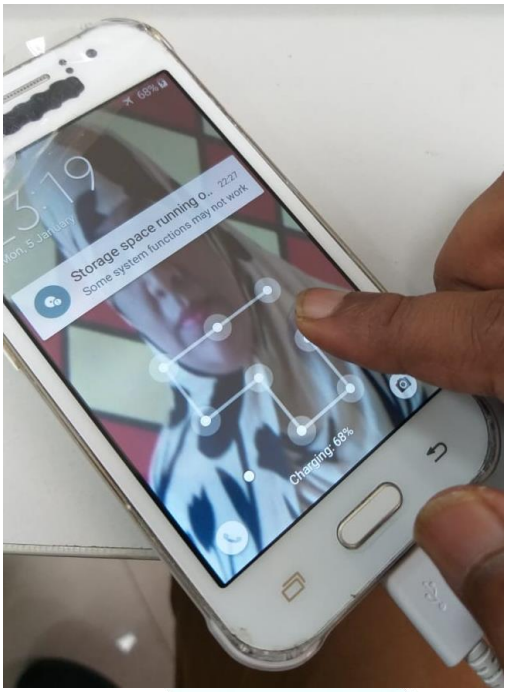


Figure 178 Pattern applied on phone and phone was unlocked

Volume - I

- Overview of Cybercrimes
- Information Gathering
- Crime Scene Management
- IP, Website and E-mail Investigation
- Communication Device Based Investigation
- Investigation of Financial Frauds
- Social Media Investigation
- Windows & Network Forensics

Volume - II

- Mobile Phone Investigation & Forensics
- IPDR and VoIP Investigation
- Cyber Security & Framework

Volume - III

- Disk Forensics
- Operating System Forensics (Windows, Linux & Mac)
- Browser Forensics
- Servers and RAID configuration
- Investigation of Digital Payment Frauds
- Virtual currencies and Crypto currencies
- Open-Source Intelligence

Volume - IV

- Malware and network forensics
- Dark web and cryptocurrency
- Advance Digital Forensics

Volume - V

- Trending Modus Operandi of Cybercrimes
- Acquaintance to Web Server and technology
- Investigation of E-Mails
- Cyber Law and Admissibility of Digital Evidence
- Digital crime Scene management
- Social media Monitoring and Sentiment Analysis
- Dark Web & Cryptocurrency Investigation
- New Technologies (Cloud, Metaverse, IoT) Investigation & Challenges