Chapter 4 - Artifact Analysis & Anti Forensics

Introduction

When conducting incident response and digital forensics on operating systems one of the sources of evidence that is normally part of every investigation is the analysis of artifacts. The user is unaware that the operating system registers traces of their activity, specific to their usage. This stored information contains probative information known as “Artifacts”. By knowing where these artifacts are stored can assist crime scene reconstruction in forensic analysis. These artifacts find great use in forensics analysis. Digital forensic scientists can employ these in incident response scenarios or lab analysis.

Like any operating system, Windows systems offer a wealth of artifacts. In the case of the Windows operating system, there may be a greater wealth of artifacts than on some other operating systems. As such, those artifacts are unique to the operating system.

Modern Linux systems have come a long way from their humble roots as a free Unix-like system for home computers. Over the past 20 years, Linux has found its way into everything to children’s toys and networking devices to the most powerful supercomputing clusters in the world. The first version of Mac OS X was released 10 years ago, and in the subsequent 10 years Apple has seen its fortune rise considerably. While not nearly as prevalent as Windows desktops and laptops, it is important that an examiner be prepared to deal with an OS X system if necessary. This chapter addresses and discusses a number of artifacts unique and specific to Windows systems, Linux file system artifacts, artifacts of system and user activity, MAC OS artifacts related to user and system activity.Also covers the different types of log file analysis and some of the anti-forensic techniques.

4.1 OS Artifacts

Every organization will deal with cyber-crime occurring on the latest operating systems. Analysts will investigate crimes including fraud, insider threats, industrial espionage, traditional crimes, and computer hacking. Government agencies use media exploitation of Windows systems to recover key intelligence available on adversary systems. To help solve these cases, organizations are hiring digital forensic professionals, investigators, and agents to uncover what happened on a system. Forensic Analysis focuses on a comprehensive and deep analysis of the latest Operating Systems. There are numbers of artifacts can be found from windows O.S. Which will helpful to the cyber forensics examiner to identify the original threat and clear the

path to the criminal.

1. Link (Shortcut) file:

LNK files are valuable artifacts for the forensics investigator. They are shortcut files that link to an application or file commonly found on a user's desktop, or throughout a system and end with an .LNK extension. LNK files are excellent artifacts for forensic investigators who are trying to find files that may no longer exist on the system they're examining. The files might have been wiped or deleted, stored on a USB or network share, so although the file might no longer be there, the LNK files associated with the original file will still exist and reveal valuable information as to what was executed on the system.

LNK files typically contain the following items of evidentiary value:

- The original path of the file.
- MAC times of the original file, not only will a LNK file contain timestamps for the LNK file itself, it will also contain MAC times for the linked file within its metadata as well.
- Information about the volume and system where the LNK file is stored. This will include volume name, serial number, NetBIOS name, and MAC address of the host where the linked file is stored.
- Network details if the file was stored on a network share or remote computer.
- File size of the linked file.

2. Thumbnails:

Thumbs.db is a hidden file used by Windows to store thumbnail images of the files in a folder. It is then used to display thumbnails when a folder is in Thumbnails view. The caching of the thumbnails in thumbs.db is intended to speed up the displaying of thumbnails as new thumbnails need not be created every time the folder is viewed. Forensic experts are interested in thumbs.db because deleting images from a folder does not remove the thumbnail from the thumbs.db database cache.

Law-enforcement agencies have used this file to prove that illicit photos were previously stored on the hard drive. There are some caveats here. On windows 8, Thumbs.db will only be created in folders under a user profile folder, so anything created in C:\ or C:\program files or C:\program data or any other folder not under a user profile, ie, C:\Users\<USER>* will not have thumbs.db files.

But this has got nothing to do with a particular logged in user. A thumbs.db file will be created

even when the logged in user browses folders of another user under their profile (as long as file permissions allow that user to write files to the other users' folder). This behaviour is different from Windows 7 thumbs.db where the location does not matter for creation of thumbs.db files. There is another oddity noted. Sometimes a thumbs.db is created immediately upon folder being opened in explorer, on other occasions, it has be triggered by changing the 'view' of the folder to 'Large icons'.

3. Jumplist Artifacts:

From a forensics standpoint, Jump Lists are a good indicator of which files were recently opened or which websites were visited frequently. The Windows operating system derives the Jump List content from two sets of Destination files:

(a) the first set is the automaticDestinations type files. These are the ones the operating system creates and maintains. They store information about data file usage. Items are sorted either by MRU or by MFU, depending on the application.

(b) The second set is the customDestinations type file. The content contained within, and the tasks specified by this category of file, are maintained by the specific application responsible for that specific Destination file.

These two categories of files are located in the following areas on the file system.

%APPDATA%\Microsoft\Windows\Recent\AutomaticDestinations\[AppID].automaticDestinations-ms

%APPDATA%\Microsoft\Windows\Recent\CustomDestinations\[AppID].customDestinations-ms

Note: These directories are hidden.

The variable named %APPDATA% is used to resolve to the C:\Users\<user account>\AppData\Roaming location.

Note : For list of AppId refer : http://forensicartifacts.com/tag/jump-lists/

4. Spooler file

When computers print files in a Windows environment, a print spooling process occurs. Spooling involves writing the print job to a couple of files so that the print job can run in the background while the user continues working with their applications. The print job is placed in a queue and prints as the printer is available to do the job. When the print job is done, the spooling files are deleted by the operating system.

When you set up your printer definition, you have the option of choosing either RAW or EMF mode. The RAW mode is a straight graphic dump of the print job. With the EMF mode, the graphics are converted to the EMF (Microsoft Enhanced Metafile) image format and each printed page is represented by individual EMF files.
For each print job, two files are created.

On Windows NT and 2000, these files will be located in Winnt\system32\spool\printers.

On Windows XP/2003/Vista/2008/7 these files will be located in Windows\system32\spool\printers.

The location is configurable by the user in the registry key ***HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Print\Printers\DefaultSpoolDirectory.***

Two temporary files created for each print job, one is called a shadow file, with the extension .shd, and the second is called the spool file, with the extension .spl. The shadow file (SHD) is analogous to a job ticket. It contains information about the print job that includes the username, the printer, the name of the file being sent to print, and the print mode (RAW or EMF).

The spool file (SPL) contains the same kind of print job information, although formatted differently, and it contains the actual print job graphical data. When the mode is RAW, the graphic content is just that a raw output of the graphical stream needed by the printer to output the job.

5. Prefetch File

Windows Prefetch files first appeared in Windows XP, and their purpose is to boost the startup process of launched applications. The information in the Prefetch file is used for optimizing the loading time of the application in the next time that you run it. Each time that you run an application in your system, a Prefetch file contains information about the files loaded by that application.

Point of interest in Prefetch files :

- The name of the executable which they accelerate and Unicode itemizations of the DLLs that the executable requires to function.

- Timestamps which pinpoint when the application was last launched.
- A counter that keeps track of the times that the executable has been launched.
- The Volume's ID (Serial Number).

Prefetch files can reveal that an application was actually installed and launched by the suspect at some point in time.

Prefetch file's naming convention :
{exename}-{hash}.pf
Exename is the name of the executable, hash is an eight character hexadecimal hash of the path from which the executable was launched, and .pf is the file extension. Filename ought to be made up of only uppercase characters with the exception of the file extension. If an application is started from three separate locations on the drive then three distinct prefetch files will be created, each corresponding to one of the locations from which the application was run.

Prefetch file is created at the same time when an application is launched first. On subsequent execution of an application, Windows will find existing prefetch file, and it will use it to make the application load up faster. The file will also be updated after it is accessed, and the application launch count will be maintained. Those files are stored in the \%windir%\prefetch directory. One important note is that the process depends on the Task Scheduler service. If the Task Scheduler service isn't running, the prefetch mechanism isn't used and the files won't be read or updated.

WinPrefetchView is a utility from NirSoft which decrypt the .pf file structure gives details about each prefetch file such as created and modified time of a prefetch file as well as last run time along with executable full path and most important one is run counter because from which user interaction to specific application could be examined from run counter.

6. Windows Shellbags Artifacts

Windows uses the Shellbag keys to store user preferences for GUI folder display within Windows Explorer. Everything from visible columns to display mode (icons, details, list, etc.) to sort order are tracked. If you have ever made changes to a folder and returned to that folder to find your new preferences intact, then in this regard Shellbags plays a major role. Shellbag information is available only for folders that have been opened and closed in Windows Explorer at least once. So that the simple existence of a Shellbag sub-key for a given directory indicates that the specific user account once visited that folder. Sometimes historical file listings are

available, which is the indication for deleted filed tracking.

Since the ShellBag subkeys store various metadata on how Windows Explorer items were arranged, and since they are recorded for each user, from a computer forensics standpoint, one can parse the data and pull out various pieces of information that relate to user interaction. When combining with detailed investigation, It could provide a more complete picture of what files were accessed or deleted by the user.

Forensic Value of Shellbags:

- Which folders were accessed (via Explorer) on the local machine, the network, and/or removable devices.
- Evidence of previously existing folders after deletion/overwrite.

  E.g: if a folder is deleted, but it was accessed via Explorer before being deleted, the shellbag artifact (and thus, the full path + initial MAC times) for that folder will remain.
- Historical MAC times of folders corresponding to the time that the folders were first accessed via Explorer.

  - Useful for determining whether or not folders were copied/moved to a new volume.
- Which user(s) accessed certain folders.
- When certain folders were accessed.

These Registry hives are located in the %user profile% and %user profile%\AppData\Local\Microsoft\Windows folders respectively. Windows uses the following Registry keys to save the folders information:

- NTUSER.DAT\Software\Microsoft\Windows\Shell\BagMRU
- NTUSER.DAT\Software\Microsoft\Windows\Shell\Bags
- NTUSER.DAT\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags
- UsrClass.DAT\Local Settings\Software\Microsoft\Windows\Shell\BagMRU
- UsrClass.DAT\Local Settings\Software\Microsoft\Windows\Shell\Bags

Shellbagsview is a utility from NirSoft which gives the details about folder accessed through windows explorer with the name of that folder followed by the full path, slot number, slot type and slot key registry path, slot modified time.

7. Recycle Bin

Forensic investigators are aware of the old adage that when a file is deleted, it is not really gone. This is especially true with the advent of the Recycle Bin on the Windows desktop. The Recycle

Bin allows user to retrieve and restore files that have been deleted. A user can open the Recycle Bin, select files that have been thrown away, and restore them to their previous locations.When a user selects a file on the desktop or through Windows Explorer and deletes it. It is not really gone. The file is simply moved to the Recycle Bin, which appears by default in the file structure as the Recycler directory at the root of each drive. In many cases, this directory can provide a significant amount of information relevant to an investigation. As a user on a system begins to delete files through the shell, a subdirectory is created for that user within the Recycler directory; that subdirectory is named with the user's security identifier, or SID. For example, the subdirectory will look something like this:

C:\RECYCLER\S-1-5-21-1454471165-630328440-725345543-1003

When an investigator opens the Recycle Bin from the desktop, the current user's subdirectory is automatically opened for view. Files sent to the Recycle Bin are maintained according to a specific naming convention. When a file is moved to the Recycle Bin, it is renamed using the following convention: **D<original drive letter of file><#>.<original extension>**

8. UserAssist

It is a registry key. It has values in subkeys that relate to each item executed on the system.

Location: ***NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist***

Contains standard subkeys related to program/LNK file execution and toolbar interaction.

Typical layout under UserAssist key is as follows:

Windows 7

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

Each count subkey contains ROT-13 encoded values; each value is a separate UserAssist entry.

Forensic Value of UserAssist:

- Frequency of program execution per user.
- Last time a program was launched.
- From whence items were being launched most often.
- System date/time changes.
- Evidence of programs after deletion/uninstall.
- How long a user has interacted with a given program (Win7).
- Evidence of absence. (i.e. "items were in a specific location at one time." e.g. "My Documents" folder is empty...but was launched 224 times).

4.2 Internet Artifacts

Internet is the main part of the public life and the analysis of the internet artifacts is very crucial task this includes mostly web browser artifacts. A browser is a software application used to locate, retrieve and display content on the World Wide Web, including Web pages, images, video and other files. As a client/server model, the browser is the client run on a computer that contacts the Web server and requests information. The Web server sends the information back to the Web browser which displays the results on the computer or other Internet-enabled device that supports a browser. Browsers can also be used to access information provided by web servers in private networks or files in file systems.

Today's browsers are fully-functional software suites that can interpret and display HTML Web pages, applications, JavaScript, AJAX and other content hosted on Web servers. Many browsers offer plug-ins which extend the capabilities of the software so it can display multimedia information, or the browser can be used to perform tasks such as video conferencing, to design web pages or add anti-phishing filters and other security features to the browser.

The two most popular browsers are Microsoft Internet Explorer and Firefox. Other major browsers include Google Chrome, Apple Safari and Opera. There are lots of artifacts we can get from the browser database files like history of searched url, search string in the google search, Download files list, stored login credentials, Bookmarks, caches, session histories, cookies , etc.

A. Cookies

A cookie is simply a small piece of textual data which is sent from the server to the client, and from the client back to the server. The data is often a string which represents a unique identifier for the client. The data is sent as part of the HTTP request and response headers.

Cookies are stored (for a period) on the client. Each cookie is associated with the domain name of the server from where it originated. When the client makes a request to a server, it checks if it has stored any cookies from that server. Cookies have expiry dates.Cookies are often used to give identities to clients. When a new cookie is created, a unique identity must be created. Cookies can store more things, such as path information, and domain information.

B. Cache

A cache is a place to store something temporarily. The files a user requests by looking at a Web page are stored on the hard disk in a cache sub directory under the browser directory. When the

user returns to a page he or she recently looked at, the browser can get it from the cache rather than the original server, saving time and decreasing network traffic. A cookie is information that a Web site puts on a user's hard disk so that the user's system can remember something about the site at a later time. Typically, a cookie records user preferences when using a particular site. Using the Web's Hypertext Transfer Protocol (HTTP), each request for a Web page is independent of all other requests. For this reason, the Web server has no memory of what pages it has sent to a user previously or anything about a user's previous visits.

Web cache is a collection of locally stored web pages, embedded images, and other types of digital objects retrieved over the Internet using the http. A web cache proxy server is essentially a shared web cache. It is a web cache program running on a dedicated server that archives and returns documents frequently requested by a group of web clients.

C. History

Web history is a common and important part of a digital investigation. It involves examining and analyzing what users of a computer system have been viewing on the World Wide Web through their internet web browsers.Critical evidence can be found in the suspect's browser history, including websites visited, searches conducted and web-based e-mail.

The history files produced by the browsers are not in a human readable format and parsing them requires external tools. For example, IE stores usage history in several index.dat files, located in various different places depending on the version of Windows. There are several forensic tools that currently analyse browser history.

D. Bookmark

A bookmark is a saved shortcut that directs your browser to a specific webpage. It stores the title and URL of the corresponding page. Saving bookmarks allows you to easily access your favourite locations on the Web. All major web browsers allow you to create bookmarks.

E. Downloads

This is the process in which data is sent to your computer. Whenever you receive information from the Internet, you are downloading it to your computer. The term downloading is distinguished from the streaming, which indicates the receiving of data that is used near immediately as it is received, while the transmission is still in progress , whereas in downloading, this would imply that the data is only usable when it has been received in its

entirety.

F. Session ID:

HTTP is a stateless protocol, which means that it provides no integrated way for a web server to maintain states throughout users subsequent requests. In order to overcome this problem, web servers or sometimes web applications implement various kinds of session management.

The basic idea behind web session management is that the server generates a session identifier (ID) at some early point in user interaction, sends this ID to the users browser and makes sure that this same ID will be sent back by the browser along with each subsequent request. Session IDs thereby become identification tokens for users, and servers can use them to maintain session data.

4.2.1 Forensic Artifact Locations

1. Internet Explorer

Since Internet Explorer (IE) is installed by default on most Windows installations, it's likely the most commonly used in corporate environments. Depending on the version of Windows and IE installed, the evidence will be stored in different locations. The locations are listed below:

WinXP – %root%/Documents and Settings/%userprofile%/Local Settings/Temporary Internet Files/Content.IE5

Win Vista/7 – %root%/Users/%userprofile%/AppData/Local/Microsoft/Windows/Temporary Internet Files/Content.IE5

Win Vista/7 – %root%/Users/%userprofile%/AppData/Local/Microsoft/Windows/Temporary Internet Files/Low/Content.IE5

Win8/IE10 – %root%/Users/%userprofile%/AppData/Local/Microsoft/Windows/History

2. Mozilla Firefox

Firefox is a very popular browser and also stores its data in various locations based on the operating system installed. It's installed as the default browser on many Linux distributions and is available for Mac OS-X as well.

WinXP – %root%/Documents and Settings/%userprofile%/Local Settings/Application Data/Mozilla/Firefox/Profiles/*.default

Win7/8 – %root%/Users/%userprofile%/AppData/Local/Mozilla/Firefox/Profiles/*.default

Linux – /home/%userprofile%/.Mozilla/Firefox/$PROFILE.default

Mac OS-X – /Users/%userprofile%/Library/Caches/Firefox/Profiles/$PROFILE.default

3. Google Chrome

Google Chrome is also one of the top 3 browsers used today. It is available for Windows, Linux, and Mac OS-X. Google also makes the Chromium open source project available to Linux users and runs similar to the regular Chrome package with some minor changes.

WinXP – %root%/Documents and Settings/%userprofile%/Local Settings/Application Data/Google/Chrome/User Data/Default

Win7/8 – %root%/Users/%userprofile%/AppData/Local/Google/Chrome/User Data/Default

Linux – /home/%userprofile%/.config/Google-chrome/Default

Mac OS-X – /Users/%userprofile%/Caches/Google/Chrome/Default

4.3 Registry Artifacts

Windows Registry is a hierarchical database which stores information about the system configuration for single or multiple users, applications installed in the system and services that run on system. The applications and operating system itself use this Registry to store values or data about user and configuration of the current system. The Registry are available all time until the system is running. All programs keep data persistently in registry so as to refer for any future use.

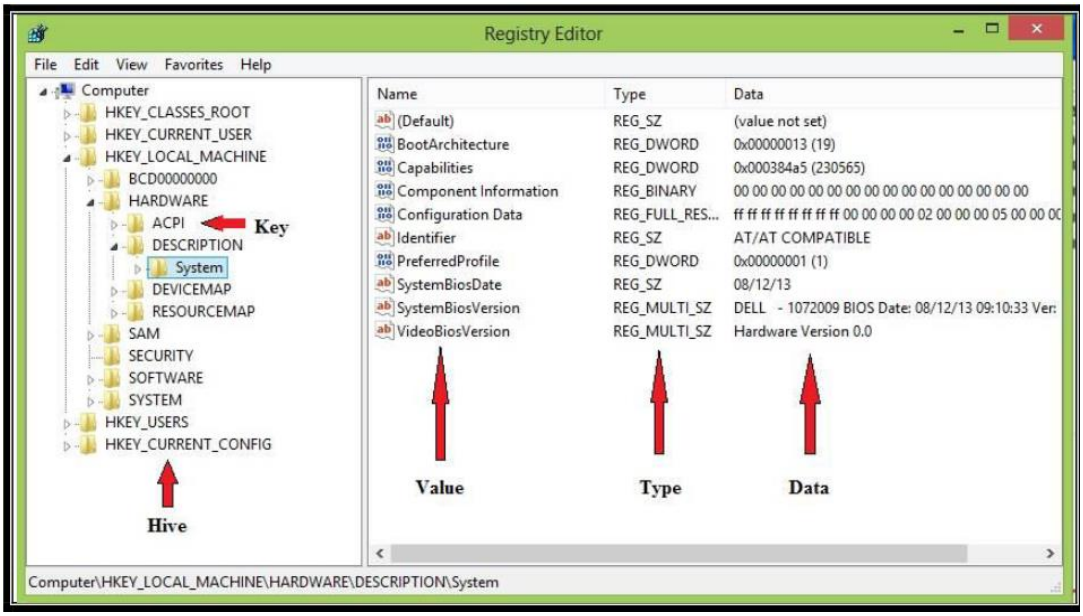
A forensic investigator discovers lots of information from the Registries such as when software has been installed, when it is updated and deleted. Also a forensic analyst can check which hardware devices are connected and at what time these devices are connected and disconnected. The investigator has a log of internet files viewed and the analyst can use that history for any further investigation.

Apart from that Registry also contains valuable data and settings for users of the software that has been installed, for instance if any changes made to control panel settings, file associations, system policies or installed software the changes are reflected and stored in software.

4.4.1 Structure of the Windows Registry:-

The structure of the Registry is of nested kind which is called as subtrees, keys and subkeys which are similar to that of files and folders in the system. Registry entries are used to store actual data, which are lowest level Registries. There are series of continuous links between each entry until the lowest entry. Each entry has a unique name with which individual entries are identified. Whenever necessary the entries are referenced by their entire path and with their names.

For opening the Registry Editor, type regedit in the run window, the Registry can be seen as one unified file system. The Windows Registry Editor is divided into two panels, the left one is key panel and the right one is value panel. In the left panel, there are five root keys, HKEY_CLASSES_ROOT [HKCR], HKEY_CURRENT_USER [HKCU], HKEY_LOCAL_MACHINE [HKLM], HKEY_USERS [HKU], and HKEY_CURRENT_CONFIG [HKCC]. However, this structure is just a logical structure. Among these five root keys, only two root keys, HKEY_LOCAL_MACHINE[HKLM] and HKEY_USERS[HKU], have physical files or hives. These two keys are called master keys. The other three keys are derived keys since they are derived from the two master keys and their subkeys.



The five root keys and their subkeys are described below.

1. HKEY_LOCAL_MACHINE (HKLM):-

HKLM is the first master key. It contains all of the configuration settings of a computer. When a computer startups, the local machine settings will boot before the individual user settings. If we double-click this entry in Windows Registry Editor, five subkeys will be listed: HARDWARE, SAM, SECURITY, SOFTWARE and SYSTEM. The information contained by these subkeys are listed below:

- **HARDWARE** is used to store the information of hardware devices that a computer detects when the computer starts up. So, the subkeys in HARDWARE are also created during the booting process.
- **SAM** is the abbreviation of Security Account Manager which is a local security database. Subkeys in SAM contain the setting data of users and work groups.
- **SECURITY** includes a local security database in SAM and a strict ACL (Access Control List) is used to manage the users who could access the database.
- **SOFTWARE** includes all of the configuration settings of programs. Information on the programs is stored in a standard format: HKLM\Software\Vendor\Program\Version.
- **SYSTEM** contains the configuration settings of hardware drivers and services. The key path is HKEY_LOCAL_MACHINE \SYSTEM \ ControlSetXXX, where XXX is a three digital number from 000.

2. HKEY_USERS (HKU):-

HKU is another master key. It contains all of the per-user settings such as current console user and other users who logged on this computer before. Double-click this entry, we can see at least three kinds of subkeys listed: DEFAULT, SID, and SID_CLASS. SID is security identifier which refers to the current console. SID-CLASSES contains per user class registration and file association. Usually, we could see S-1-5-18, S-1-5- 19, and S-1-5-20, which represents Local System Account, Local Service Account and Network Service Account respectively.

Unlike the above two keys, HKEY_CLASSES_ROOT (HKCR), HKEY_CURRENT_USER (HKCU), and HKEY_CURRENT_CONFIG (HKCC) are derived keys and they only link to the two master keys and their subkeys.

3. HKEY_CLASSES_ROOT (HKCR):-

HKCR contains two keys: HKLM\SOFTWARE\Classes and HKCU\Software\Classes. The first one refers to the default registration classes, and the second one refers to per user registration

classes and file associations.

4. HKEY_CURRENT_USER (HKCU):-

HKCU links to a subkey of HKU, HKU\SID. This key allows all of the Windows programs and applications to create, access, modify, and store the information of current console user without determining which user is logging in. Under the root key HKCU, there are also five subkeys: Environment, Identities, Network, Software, and Volatile Environment.

- Environment is about the environmental configurations.
- Identities are related to Outlook Express.
- Network contains settings to connect the mapped network drive.
- Software refers to the user application settings.
- Volatile Environment is used to define the environmental variables according to different users who logon a computer.

5. HKEY_CURRENT_CONFIG (HKCC):-

HKCC is an image of the hardware configuration profiles. HKLM\SYSTEM\CurrentControlSet\HardwareProfiles\Current, is also a link to HKLM\SYSTEM\ControlSet\HardwareProfiles\XXXX, where XXXX is a four digital number from 0000.

4.4.2 HIVE:-

Registry Editor only shows the logical structure of the registry. Physically, registry is not stored in a single file in the hard drive. Windows stores registry in a few separated binary files called hives. For each hives file, Windows creates additional supporting files that contain backup copy of the respective hives to restore the hives during failed system boot. Only HKLM and HKU has corresponding hives. However, none of 5 root keys are directly associated to a hive file. Table 1 shows registry path and their corresponding hives on disk.

Registry Path	Hive and Supporting Files
HKLM\SAM	SAM, SAM.LOG

HKLM\SECURITY	SECURITY, SECURITY.LOG
HKLM\SOFTWARE	software, software.LOG
HKLM\SYSTEM	system, system.LOG
HKU\.DEFAULT	default, default.LOG
HKU\SID	NTUSER.DAT
HKU\SID_CLASSES	UsrClass.dat, UsrClass.dat.LOG

Table: Registry Hives

All hives in HKLM are stored in C:\Windows\System32\config\. HKLM\HARDWARE is a dynamic hive that is created each time the system boots and it is created and managed entirely in memory. HKU\.DEFAULT hive file correspond to C:\Windows\System32\config\default. HKU\SID hive file is stored in user home directory, which is C:\Users\<username>\NTUSER.DAT, while HKU\SID_CLASSES hive file correspond to: C:\Users\<username>\AppData\Local\Microsoft\Windows\UsrClass.dat.

4.4.3 FORENSIC IMPORTANCE OF REGISTRY HIVE

1. NTUSER.DAT

NTUSER.DAT stores information that is specific to the user. If there are multiple user accounts on the computer, there are also multiple NTUSER.DAT files – one for each user. NTUSER.DAT stores data that is specific to the user, such as which files they opened, which applications they used, and which websites they visited. All of this data can be found here:

C:\Users\Username\NTUSER.DAT

For opening the NTUSER.DAT file :-

1] The YARU tool is opened in the Run As Administrator mode
2] Click on File tab then Open user hive (local)
3] Load the hive ntuser.dat which has the path given below
   C:\user\username\ ntuser.dat
4] Then Click on the Accept button.

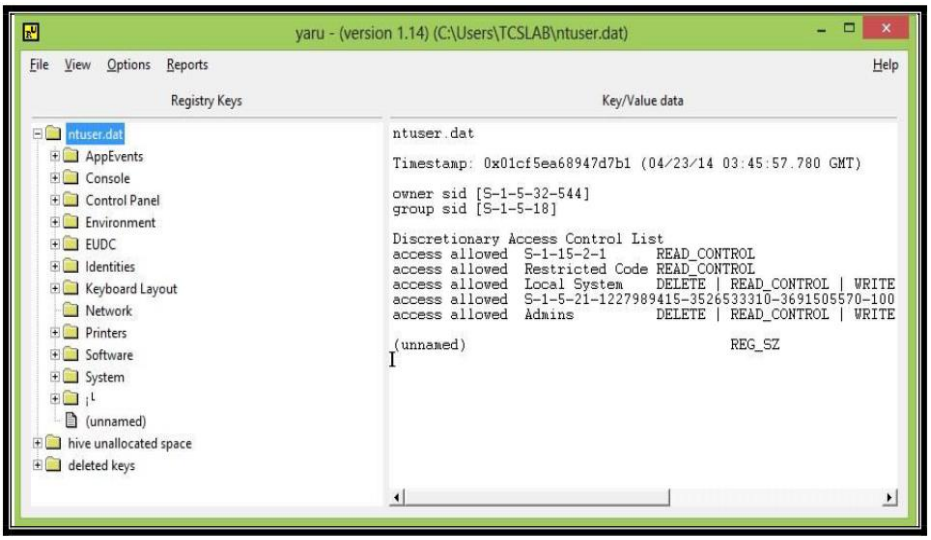


Figure: Registry in Yaru Parser tool

A] Computer name and Volume Serial Number

In this registry key, we get the Computer name which is here TCSLAB19 and Volume

Serial Number of the C drive in Hexadecimal form.

ntuser.dat\Software\Microsoft\Windows Media\WMSDK\General

B] OpenSavePidlMRU -- Recently Opened or Saved File

This key tracks files that have been opened or saved within a Windows shell dialog box. There is a possibility for a large number of subkeys present in the OpenSaveMRU key. It stores full path information of that files with extension that have been opened or saved. The * subkey tracks the last twenty files of any extension that have been input into the OpenSave dialog box in Windows 8. Each subkey keeps its own Most Recently Used (MRU) list and last write time.

ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU

C] LastVisitedPidlMRU -- Recently Opened or Saved Folders

It tracks the specific executable used by an application to open the files documented in the OpenSaveMRU key. It also tracks the directory location for the last file that was accessed by that application. Data is stored in binary format. This key is used by the OpenSave dialog box to show the last directory used by that application when you are attempting to open or save a file. It keeps its own MRU list and last write time.

ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\La

stVisitedPidlMRU

D] Typed URLs

When a user types a URL directly into the Internet Explorer browser, the TypedURLs subkey is updated. The list of URLs is sorted by number like url1, url2 etc. The lowest number is the first or most recently typed URL. Every new URL will contain a separate entry and the contents of the URL are in UTF-16 format. The names are indexed based on the most recently used as the lowest index. Conversely, the highest index equates to the oldest entry.

ntuser.dat\Software\Microsoft\Internet Explorer\TypedURLs

The key TypedURLs is an artifact which contains a listing of the most recent URLs that have been typed into the Internet Explorer address bar. It is useful key in an investigation because it shows that the website has been visited by a specific user account on the computer.

2. System Information

The system information is stored in the registry in the CurrentControlSet key. The active key is numbered 1. Within the Registry Viewer are two ControlSets: ControlSet001 and ControlSet002. An investigator can find the computer name in the following key, in the ComputerName value:

SYSTEM\CurrentControlSet\Control\ComputerName\ActiveComputerName

An investigator can find the time that the system was last shut down in the following key:

SYSTEM\ControlSet00x\Control\Windows

The ShutdownTime value beneath this key is a FILETIME object that can be correlated with other times on the system, such as Event Log entries and the like, to assist in developing a timeline of activity and system use.

The following key could also be of value during an investigation:

SOFTWARE\Microsoft\Windows NT\CurrentVersion

This key holds several values that provide information about the system. The ProductName, CurrentBuild- Number, and CSDVersion values tell about the operating system and version.

3. Time Zone Information

An investigator can find information about the time zone settings in the following key:

SYSTEM\CurrentControlSet\Control\TimeZoneInformation

This information can be important for establishing a timeline of activity on the system. Many tools are available to extract information regarding times and dates in UTC/GMT time. The investigator can use the ActiveTimeBias value from the TimeZoneInformation key to translate or normalize the times to other sources from the system, such as entries in log files.

4. Autostart Locations

Autostart locations within the registry are locations that allow applications to be launched without any interaction from the user. These locations were originally provided for the user's convenience. Some applications, such as touch-pad drivers and applications on laptops, as well as antivirus and firewall applications, are most useful when they are started automatically.

Attackers often use malware that automatically starts when the system boots or when a user logs in because a user does not have to run a particular program to cause the malware to run in these situations.

On a live Windows XP system, the msconfig command launches the System Configuration Utility. Users can run this command by clicking the **Start** button, choosing **Run**, typing **msconfig** into the textbox, and then pressing Enter. The System Configuration Utility allows an administrator to choose which applications and services automatically start.

5. User Login

When a user logs into a system, the following registry keys are accessed and parsed so that the listed applications can be executed:

1.HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Runonce
2.HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
3. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
4. HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run
5. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
6. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce

These keys are ignored if the system is started in Safe Mode.

On Windows XP and 2003 systems, if an administrator prefaces the RunOnce values (keys 1 and 6) with an asterisk (*), the associated programs can be run even if the system is started in Safe Mode.

6. MRU Lists

Many applications maintain an MRU list, which is a list of files that have been most recently used. Within the running application, these file names generally appear at the bottom of the drop-down menu when File on the menu bar is selected.
The MRU list registry key is the following key:

\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

This key can contain a number of values, all of which are binary data types. The values investigators are interested in are the ones that have names that are numbers and the one named MRUListEx. The numbered value names contain the names of the files accessed (in Unicode), and the MRUListEx value maintains the order in which they were accessed (as DWORDs).

The RecentDocs key also has a number of subkeys, each one being the extension of a file that was opened (.doc, .txt, .html, etc.). The values within these subkeys are maintained in the same way as in the Recent- Docs key: the value names are numbered, and their data contains the name of the file accessed as a binary data type.

7. Tracking User Activity

We can use a number of Registry keys to track user activity. This type of Registry key is different from the autostart/user activity Registry keys, which are keys accessed when a user performs a specific action. You can find these Registry keys in the NTUSER.DAT file for the user and they are updated (i.e., entries are added). when a user performs specific actions. When this happens, the key's LastWrite time is updated, which brings us back to the concept of the Registry as a log file. Also, there are keys that track user activity and add or modify time-stamp information associated with the Registry values. This time-stamp information is maintained in the value data.

The HKEY_CURRENT_USER hive contains the most information about user activities in locations referred to as most recently used, or MRU, lists.

8. USB Removable Storage Devices

When a USB removable storage device, such as a thumb drive, is connected to a Windows system, footprints or artifacts are left in the registry. When the device is plugged in, the Plug and Play (PnP) Manager receives the event and queries the device descriptor in the firmware for information about the device, such as the manufacturer.

The PnP Manager then uses this information to locate the appropriate driver for the device and, if necessary, loads that driver. Once the device has been identified, a registry key is created beneath the following key:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Enum\USBSTOR

The subkey that is created beneath this key is named with the following form:
Disk&Ven_###&Prod_###&Rev_###

4.4 File System Analysis

4.4.1 File System

With all the millions or billions of files floating around inside our computers, there has to be some way to keep things neat and tidy. This indispensible function is the responsibility of the file system. The file system tracks the drive's free space as well as the location of each file. The free space, also known as unallocated space, is either empty or the file that previously occupied that location has been deleted.

There are many different types of file systems. Some of the most commonly encountered by forensic examiners include FAT, NTFS, ext, and HFS+. Let's take a closer look:

File Allocation Table (FAT) is the oldest of the common files system. It comes in four flavors: FAT12, FAT16, FAT32, and FATX. Although not used in the latest operating systems, it can often be found in flash media and the like.

The New Technology File System (NTFS) is the system used currently by Windows 7, Vista, XP, and Windows Server. It's much more powerful than FAT and capable of performing many more functions. For example, "NTFS can automatically recover some disk-related errors, which FAT32 cannot," it provides better support for larger hard drives, and better security through permissions and encryption (Microsoft Corporation).

Hierarchical File System (HFS+) and its relatives HFS and HFSX are used in Apple products. HFS+ is the upgraded successor to HFS. This newer version offers several improvements including improved use of disk space, cross-platform compatibility, and international-friendly file names.

The Ext file system is based on the Unix file system (UFS) but removed many components of it. In addition, the Ext file system was designed to be fast and reliable to store data. Therefore, backups of important data structures of the file system are made. Ext3 is the successor of Ext2. The main difference between the Ext2 and Ext3 file system is that Ext3 supports journaling. the

Ext4 file system added besides support for bigger files and directories features like multiblock allocation, delayed allocation, journal checksum and an option to turn journaling off. Note that these features mainly have increased the performance and reliability when compared to Ext3. For instance, the fsck command, i.e., the file system check runs faster on an Ext4 file system.

4.4.2 File Signatures

A file signature is a unique identifying number located at the beginning of a file. This number identifies the type of file, giving information about the data contained within the actual file. This information can be used to determine what type of file is being read when the file extension or user error has misidentified the file as an incorrect type. The file signature also can contain information that ensures the original data that was stored in the file is still intact and has not been modified. The combination of these elements allows a file signature to serve as an important form of verification, especially against computer viruses.

The concept of a file signature emerged because of the need for a file header, a block of data at the beginning of a file that defines the parameters of how information is stored in the file. Part of the header information is a sequence of bytes that defines the file type that was originally created. This can be an image file, a document from a specific program, or even a protocol type when a file stream is being used as a communications method between a client and server. The file header does not use a defined standard. It, instead, is proprietary to each different format, meaning a program or operating system needs a file signature database to determine the type of an unknown file.

Examples of Some File Signatures:

S.NO	File Extension	File (Hex) Signature
1.	JFIF, JPE, JPEG, JPG	FF D8 FF E0 xx xx 4A 46 49 46 00
2.	PNG	89 50 4E 47 0D 0A 1A 0A
3.	PDF	25 50 44 46
4.	MP4	00 00 00 14 66 74 79 70 69 73 6F 6D

4.4.3 Carving

Basically, carving is the process of searching a data stream for file headers and magic values and trying to find the file end point. This part of data is then ”carved” out and saved into a file. *Magic values* are basically *signatures* determining the start and end of known file types. These are generally file type specific hexadecimal values located at specific offsets relative to the beginning of the file. Usually, carving is performed on unallocated space and therefore also allows recovering of (intentionally) deleted files whose metadata has been deleted. What is more, it can be used to recover data from a damaged or corrupted file system because not all information usually given by the file system is needed.

Note that fragmentation can be minimized by first extracting the unallocated space of the volume into a contiguous file using blkls. Extracting the right information of unstructured data is more of an art than science. In addition, carving is still an open problem . For example the tool foremost can be used to carve files based on file headers or internal structures of the data. It can operate on raw file systems or disk images. Originally, it was developed by the United States *Air Force Office of Special Investigation* and later updated by Nick Mikus at the *Center for Information Systems Security Studies and Research, Naval Postgraduate School*. The tool can be downloaded from http://foremost.sourceforge.net/.

See below for an example using JPEG:

Basically, the tool finds files based on signatures saved in a configuration file. For each signature there is an entry in the configuration file. Note that the configuration file uses \x[0-f][0-f] to specify a value in hexadecimal which is decoded before use. For instance, the entry for a JPEG looks as follows:

jpg	y	200000	\xff\xd8	\xff\xd9

where the following information is stored. Note that we label the fields of an entry from left to right.

1. Typical file extension
2. Whether the header and footer value is case sensitive indicated by *y* (yes) or *n* (no).
3. Maximum size of a file in bytes
4. Header value
5. Footer value

Note that the *maximum size* of the file indicates the maximum number of bytes between the header and footer value. Therefore, if foremost finds no footer value after reading the number of bytes given by the *maximum size* starting at the header value, the carving will stop for this file.

It is important to see that files bigger than the *maximum size* given in the configuration file cannot be detected and in our example, there might be a lot of important JPEGs being missed as many are bigger than 200000 bytes.

This simple technique can be supplemented with content analysis which can reduce false positives when combined with signature searching. What is more, this technique can be used solely to find data. This can be very useful, e.g., when the signatures are corrupted, i.e., if there are no file headers or footers which can be searched for. However, preventing false positives in this case can be quite a challenge depending on the file type.

4.4.4 File Slack

A block is the smallest addressable amount of data storage. Therefore, if a file is smaller than the block or in general not a multiple of the data unit size, the rest of the last block is unused. This space behind the file footer is called *slack space*. The slack space can now be divided into two regions where data not belonging to the actual file can reside. Note that a block consists of multiple physical disk sectors. See Fig. for a schematic overview. In the following, we will explain these two regions in more detail.

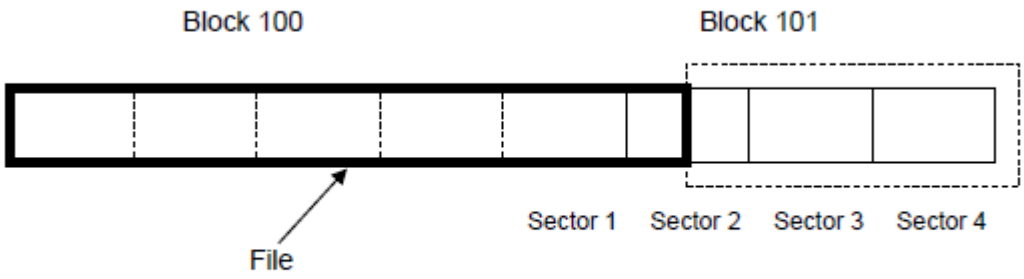


Fig. File Slack. The file allocates 2 blocks but its real size is 5.5 sectors. The sectors are labeled relative to the block 101. The dotted rectangle indicates the slack space.

The first area is the space behind the file footer until the end of the current sector,i.e., the sector which contains the file footer. This first area is of particular interest since the whole sector has to be written and it is therefore not possible for the operating system not to write any data after the file footer. Therefore, depending on the operating system, this particular region of slack space might be padded with zeros before being written. However, it used to be padded with RAM content. This later special case occurred, for instance, in some older operating systems such as MSDOS and early DOS based MS Windows. This particular piece of the slack space was called *RAM slack* and could even contain passwords or cryptographic keys, respectively. Nowadays, modern operating systems do pad with zeros.

The second region of the slack space is the remaining area from the end of the sector where the file footer resides in to the last sector contained in the block. It does not have to be written and is either again padded with zeros or ignored depending on the operating system. Obviously, if the area is ignored, it could contain data from a deleted file which previously allocated it. Note that slack space belongs to the allocated space of a file and does not reside in unallocated space. The term *file slack* describes slack space belonging to a particular file.

Another kind of slack is *volume slack*. It describes the part of a volume which is not included in the file system but follows the actual file system. First, note that a file system does not have to use the whole size of the volume. In particular, a volume could contain more than one file system. Therefore, if the file system is smaller than the volume, the space behind it can be used to hide data.

4.5 File System Artifacts

4.5.1 NTFS (New Technology File System) analysis:

NTFS is the standard file system of Windows series of Microsoft with better file security (Encrypting File System), disk compression, logging features, reliability and stability. NTFS volumes can not be accessed by DOS, or Windows 95 or Windows 98. Formatting a volume with the NTFS file system results in the creation of several system (metadata) files such as $MFT - Master File Table, $Bitmap, $LogFile and others, which contains information about all the files and folders on the NTFS volume.

NTFS - Metadata files:

Metadata file	Description
$MFT	Store MFT Record
$MFTMirr	Contain partial backup of MFT
$LogFile	Transaction logging file
$Volume	Contain volume information such as label, identifier and version
$AttrDef	Attribute definition
.	Root directory of file system
$Bitmap	Contain the allocation status of all clusters

$Boot	Contain the boot record
$BadClus	Mark clusters as bad clusters
$Secure	Contain information about the security and access control information
$Upcase	Contain information for converting file names to the Unicode (16-bit) file

NTFS Partition Boot Sector:

The first sector on an NTFS volume is a boot sector which starts at sector 0.

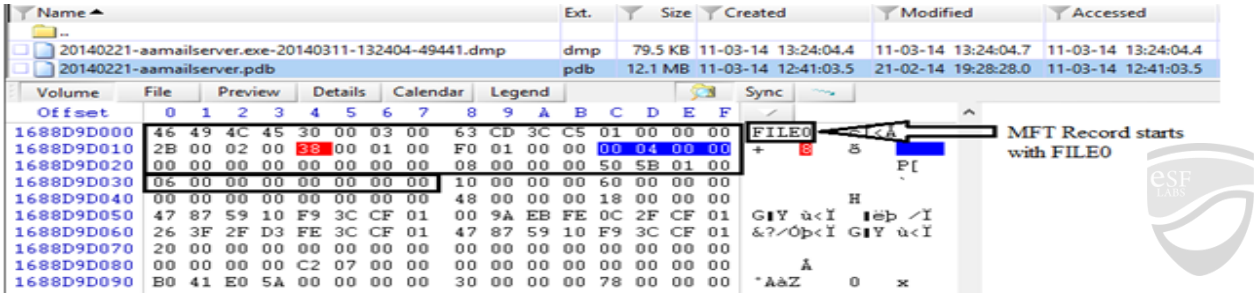
Byte Offset	Value	Field Name
0x03-0x0A	NTFS	OEM ID
0x0B-0x0C	0x0002	Bytes per Sector-512
0x0D	0x08 = 8	Sectors per Cluster - 8
0x0E-0x0F	0x0000	Reserved Sectors
0x15	0xF8 = F8	Media Descriptor - Fixed Disk
0x18	0x3F00	Sectors per Track - 63
0x1A-0x1B	0xFF00	Number of Heads - 255
0x1C-0x1F	0x00F80A00	Hidden Sectors
0x28-0x2F	0xFF05B51200000000	Total Sectors = 313853439 Bytes = 313853439*512 Size = 149GB
0x30-0x37	0x00000C0000000000	Logical Cluster Number for the $MFT file

0x38-0x3F	0x0200000000000000	Logical Cluster Number for the $MFTMirror file
0x48-0x4B	0xCC85BE50	Volume Serial Number - 50BE85CC

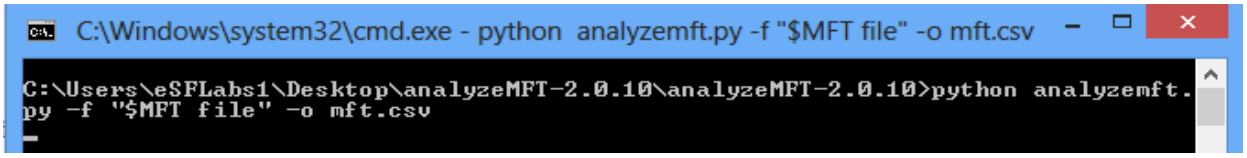
In NTFS, everything is file. Every file or directory has at least one entry in MFT (Master File Table). Microsoft calls each entry in MFT as file record and its default size is 1024 bytes. The first 56 bytes is fixed for MFT entry header and remaining bytes stores attributes. Example of attributes are $STANDARD_INFORMATION(0x10), $FILE_NAME(0x30) and $DATA(0x80). In $DATA(0x80) attribute, content can be either resident or nonresident.

As files are added to an NTFS file system volume, more entries are added to the MFT and the MFT increases in size. When files are deleted from an NTFS file system volume, their MFT entries are marked as free and may be reused. However, disk space that has been allocated for these entries is not reallocated, and the size of the MFT does not decrease.

The first 56 bytes is fixed for MFT entry header and remaining bytes stores attributes.



Mft file can be viewd in mft parser and the location of the mft file is in the root. Below is the image of how mft parser can parse the mft file. Here one python tool called analyzemft is used to parse the mft file. It will parsed the mft file in .csv format.



4.5.2 FAT (File Allocation Table) analysis:

FAT12: The oldest type of FAT uses a 12-bit binary number to hold the cluster number. A volume formatted using FAT12 can hold a maximum of 4,086 clusters, which is 2^12 minus a few values. FAT12 is therefore most suitable for very small volumes and is used on floppy disks and hard disk partitions smaller than 16 MB.

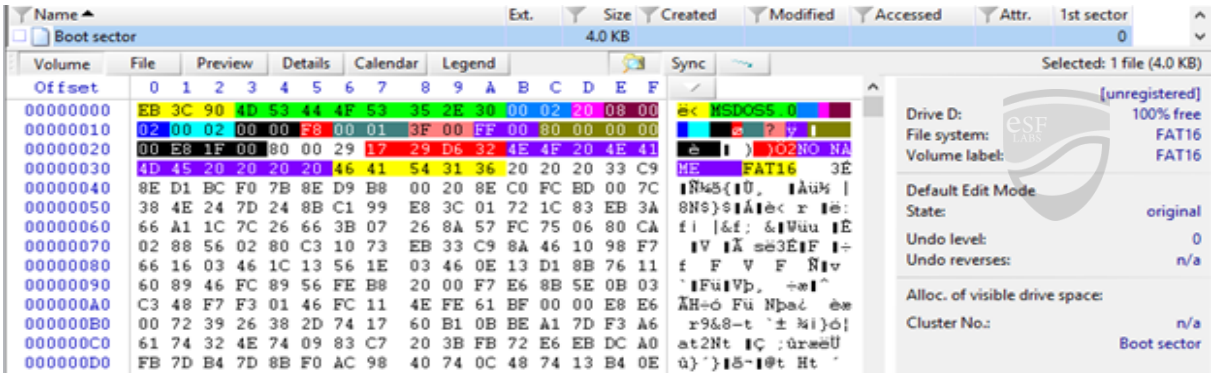
FAT16: The FAT used for most older systems, and for small partitions on modern systems, uses a 16-bit binary number to hold cluster numbers. A volume using FAT16 can hold a maximum of 65,526 clusters, which is 2^16 minus a few values. FAT16 is used for hard disk volumes ranging in size from 16 MB to 2,048 MB.

FAT32: The newest FAT type, FAT32 is supported by newer versions of Windows, including Windows 98, Windows ME and Windows 2000. FAT32 uses a 28-bit binary cluster number because 4 of the 32 bits are "reserved". 28 bits is still enough to permit huge volumes. FAT32 can handle volumes with over 268 million clusters, and will support (theoretically) drives up to 2 TB in size.

Layout of FAT Formatted Volume:

Partition Boot Sector	FAT 1	FAT 2 (duplicate)	Root folder	Other folders and files

FAT Partition Boot Sector:

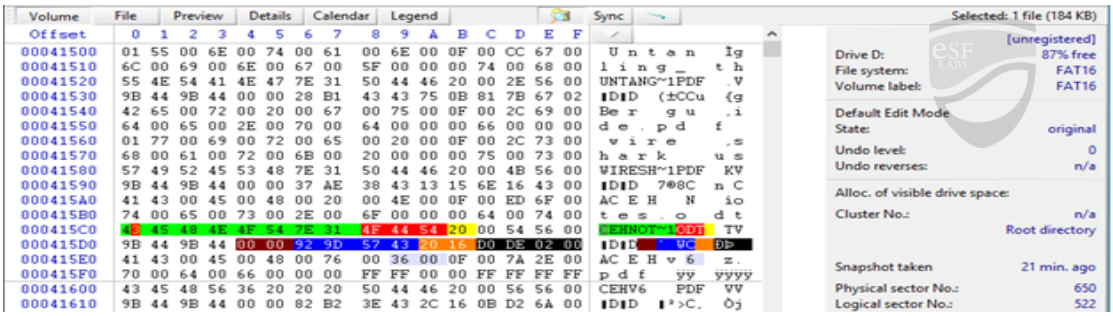


Byte Offset	Value	Field Name
0x00	EB 3C 90	Jump instruction
0x03	MSDOS5.0	OEM Name in text
0x0B-0x0C	0x0002	**Bytes per Sector - 512 bytes**
0x0D	0x20	**Sectors Per Cluster - 32** **32*512 bytes = 16384 bytes = 16KB = Allocation unit size**
0x0E-0x0F	0x0800	Reserved Sectors - 8
0x10	0x02	**Number of file allocation tables(FATs) - 2**
0x11-0x12	0x0002	Root Entries - 512
0x13-0x14	0x0000	Small Sectors - 0

0x15	0xF8	Media Type - F8 - i.e. Hard Disk
0x16-0x17	0x0001	Sectors per file allocation table(FAT) - 256 sectors
0x18-0x19	0x3F00	Sectors per Track - 63
0x1A-0x1B	0xFF00	Number of Heads - 255
0x1C-0x1F	0x80000000	Hidden Sectors - 128
0x20-0x23	0x00E81F00	Large Sectors - 2091008
0x24	0x80	Physical Disk Number - Hard Disk
0x27-0x2A	0x1729D632	Volume Serial Number - 1729D632
0x2B-0x35	NO NAME	Volume Label
0x36	FAT16	System ID

Directory entry of a file:

In the fig. Shows the file system is FAT 16 and the color highlighted string is the entry of the file in the FAT file system.



4.6 Metadata Analysis

Metadata is a structured description of objects, which contains certain properties useful to the user as well as the program on which the document was created. More succinctly, Metadata is data about data. Metadata plays a number of important roles in computer forensics:

- It can provide corroborating information about the document data itself.
- It can reveal information that someone tried to hide, delete, or obscure.
- It can be used to automatically correlate documents from different sources.

It is embedded and provides additional information, including when and by whom it was created, accessed, or modified which is called summary of a file.

4.6.1 Kinds of Metadata

There are different kinds of metadata and out of that forensically important metadata are kind of below :

- Digital image metadata such as the image size, color combinations and EXIF info such

as device name which had used to take image, GPS info.

- File system metadata such as MACB times, ACLs (Access Control List).
- Document metadata such as author of a document,last write time, etc.

Metadata is also used by relational databases in order to describe the tables, columns, and fields. In addition, it is used by newer filesystems to store information about the data that is saved on the disk. Linux ext3 filesystem logs all filesystem data and metadata changes in a journal. Microsoft filesystems, such as NTFS, also store this type of information. In addition, older filesystems, such as Linux ext2 and Windows FAT16, store metadata file information such as timestamps, size, and data location. Filesystem forensics and investigation can help to fetch details about files which can be fetched from journal files of newer filesystems and also through manual file metadata analysis.

Metadata is used extensively by office applications such as Microsoft Office, OpenOffice.org applications, PDF documents in order to store information such as the document's creator, date of creation, date and last time it was saved, the location the document to which the document was saved.

4.6.2 Microsoft Office Metadata

According to Microsoft's Knowledge Base [17, 18, 19, 20], the following data can be saved as hidden information inside of Microsoft Office documents, spreadsheets and presentations:

- Hidden Information (PowerPoint)
- Your name (All)
- Your initials (All)
- Your company or organization name (All)
- The name of your computer (All)
- The name of the network server or hard disk where you saved the document (All)
- Other file properties and summary information (All)
- Non-visible portions of embedded OLE objects (All)
- The names of previous authors (All)
- Document revisions (Word, Excel)
- Document versions (Word)
- Template information (Word, PowerPoint)
- Hidden text (Word, Excel)
- Comments (All)

- Globally Unique Identifiers (GUIDs) (All)

4.6.3 Metadata Formats

After Office 2003, the XML format will be the standard file format. When saving documents in the new XML format, the user or investigator can access the metadata by viewing the source XML file with a text editor. This is an example of how the metadata is stored near the beginning of a test Microsoft Office 2003 Word XML file:

```
<o:DocumentProperties>
    <o:Title>This is a test</o:Title>
    <o:Author>JJ</o:Author>
    <o:LastAuthor>JJ</o:LastAuthor>
    <o:Revision>2</o:Revision>
    <o:TotalTime>0</o:TotalTime>
    <o:Created>2006-02-26T02:0600z</o:Created>
    <o:LastSaved>2006-02-26T202:06:00z</o:LastSaved>
    <o:Pages>1</o:Pages>
    <o:Words>9</o:Words>
    <o:Characters>55</o:Characters>
    <o:Company>Self</o:Company>
    <o:Lines>1</o:Lines>
    <o:Paragraphs>1</o:Paragraphs>
    <o:Version>11.6568</o:Version>
</o:DocumentProperties>
```

Microsoft begins the metadata section with the **<o:DocumentProperties>** and continues with fairly self explanatory tags describing the document, as well as the authors, and company information.

4.6.4 File Metadata Analysis

Regarding to file metadata, we will take different scenarios such as copy-paste operation, cut-paste operation and will note down file timestamps after each operation and in conclusion will describe how it would be important in forensic investigations.

Step 1) "FileAlyzer 2.0.5.57" software used to analyze what changes reflects in MACB of a document if it has been copied or moved from one location to another.

- Birth Time – Time when the file was created.
- Created Time – Time when the file was created on the current system.
- Last Accessed Time – Time when the file was accessed for the last time.
- Last Write Time – Time when the file was written for the last time.

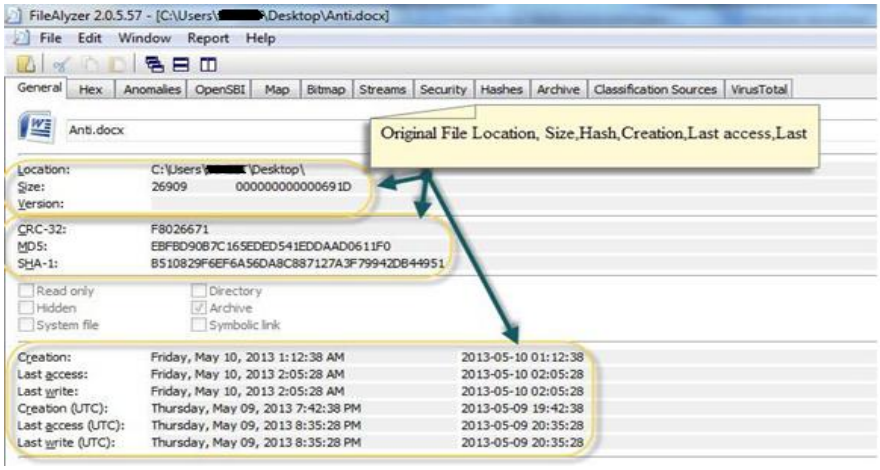


Figure 1: File metadata

Step 2) Analyze file after copy paste operation.

After doing Copy (Copy + Paste) operation, Creation & Last access time of a document would be changed.

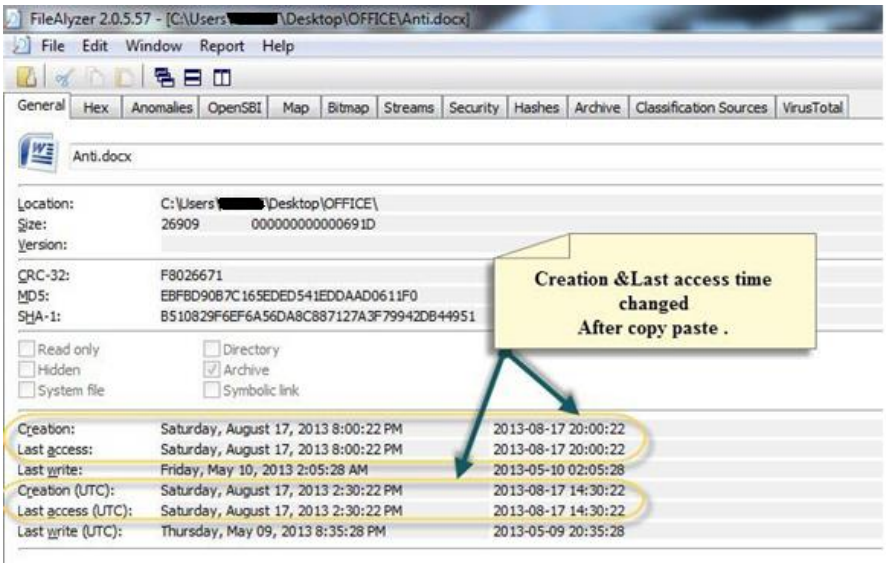


Figure 2: File metadata after copy - paste operation

Step 3) Analyze file after cut paste operation.

After doing Move (Cut + Paste) operation only Creation time of a document would be changed.

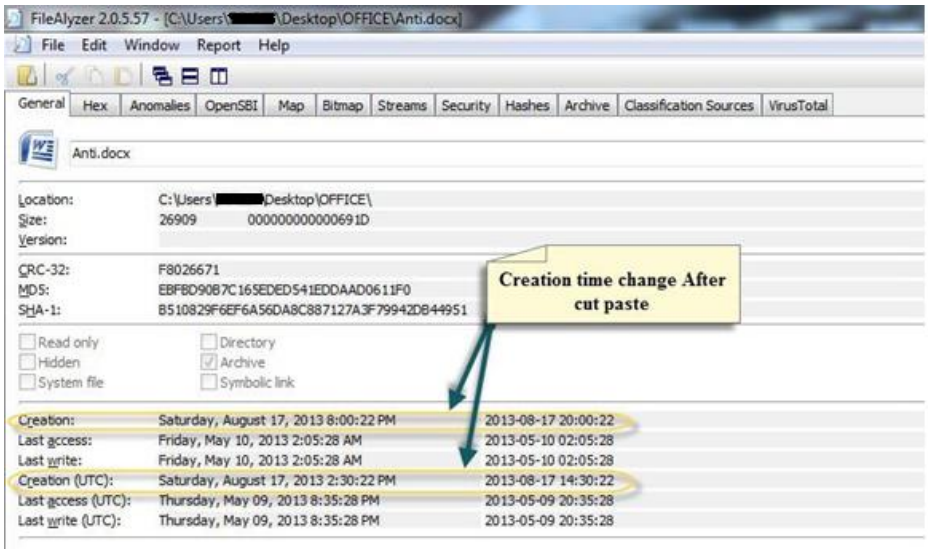


Figure 3 : File metadata after cut - paste operation

Conclusion: All operation with it affect the MACB is shown in below Table.

Operation	Birth Time	Created Time	Last Accessed Time	Last Write Time
Copy - Paste		Yes	Yes	
Cut - Paste		Yes		

4.6.5 Forensic Investigation

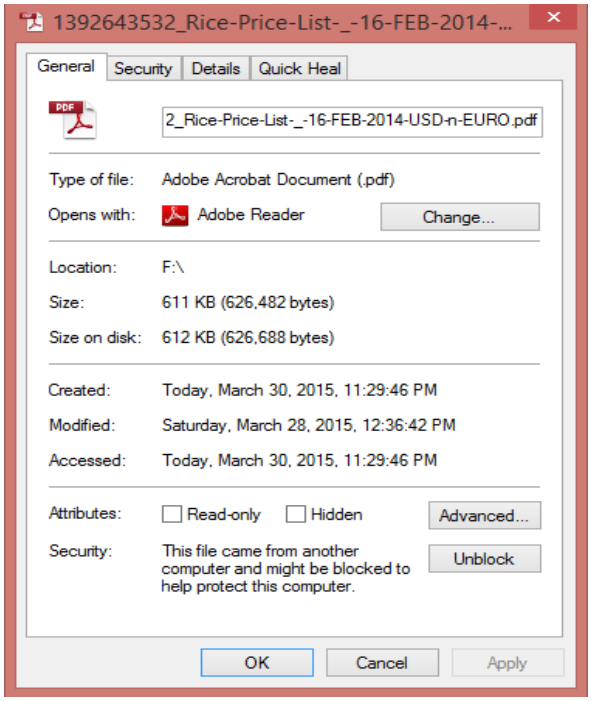


Figure 4 : File Metadata Investigation

Carefully, see the created time and modified time of a subject file in above figure. By looking at

above timestamps one would think that how modified time be earlier than created time.Its possible when file is copied from another location and pasted or moved at different location so at that time created time of a file would be changed to current time when it is moved or pasted so we can say this file is not created over here and might came from different location.

4.7 Application Artifacts

Applications (Apps) that utilise the Metro Modern UI are treated differently to programs that work in desktop mode. Apps are installed in the following directory:

\Program Files\WindowsApps

Settings and configuration DBs are located in following directories:

\Users\user_name\AppData\Local\Packages\package_name\LocalState

Two DB formats:

1. SQLite DBs (.SQL)
2. Jet DBs (.EDB)

4.7 Log Analysis

Log files are excellent sources for determining the health status of a system and are used to capture the events happened within an organization‟s system and networks. Logs are a collection of log entries and each entry contains information related to a specific event that has taken place within a system or network. Many logs within an association contain records associated with computer security which are generated by many sources, including operating systems on servers, workstations, networking equipment and other security software‟s, such as antivirus software, firewalls, intrusion detection and prevention systems and many other applications. Routine log analysis is beneficial for identifying security incidents, policy violations, fraudulent activity, and operational problems. Logs are also useful for performing auditing and forensic analysis, supporting internal investigations, establishing baselines, and identifying operational trends and long-term problems.

Initially, logs were used for troubleshooting problems, but nowadays they are used for many functions within most organizations and associations, such as optimizing system and network performance, recording the actions of users, and providing data useful for investigating

malicious activity. Logs have evolved to contain information related to many different types of events occurring within networks and systems. Within an organization, many logs contain records related to computer security; common examples of these computer security logs are audit logs that track user authentication attempts and security device logs that record possible attacks.

With the world wide deployment of network servers, service station and other computing devices, the number of threats against networks and systems have greatly increased in number, volume, and variety of computer security logs and with the revolution of computer security logs, computer security log management are required. Log management is essential to ensure that computer security records are stored in sufficient detail for an appropriate period of time. Log management is the process for generating, transmitting, storing, analyzing, and disposing of computer security log data. The fundamental problem with log management is effectively balancing a limited quantity of log management resources with a continuous supply of log data. Log generation and storage can be complicated by several factors, including a high number of log sources; inconsistent log content, formats, and timestamps among sources and increasingly large volumes of log data. Log management also involves protecting the confidentiality, integrity, and availability of logs. Another problem with log management is ensuring that security, system, and network administrators regularly perform effective analysis of log data.

4.8 Windows Logs

Windows systems maintain log files for a number of events and actions that may be important to an analyst. Besides application log files, which maintain logs of events with respect to specific applications, the Windows operating system also maintains a number of logs.

The Event Logs are perhaps the most well-known logs on Windows systems. The information present in the event logs is useful, especially when it comes to gathering evidence for forensic investigations related to malicious attacks intrusive actions of fraudulent behaviour. The windows event logs are considered as important sources of forensic information as they relate certain events to particular point in time. The windows operating system is built on a complex architecture with which to handle events like logging on requires proper security measures. It is also possible that the windows event logs are targeted to specific kind of events. The system logs and application logs can be used in a number of ways of writing specific events to the log.

The Event Logs are the most well-known logs on Windows systems, the most equivalent of

syslog on Linux systems. The Event Logs record a variety of day-to-day events that occur on Windows systems and are configurable to record a range of additional events. These events are split into categories such as Security, System, Setup and Application Event Logs. The Event Logs can provide a good value of information that's useful for troubleshooting issues as well as for understanding events and actions during forensic analysis.

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

The Windows Logs category includes the logs that were available on previous versions of Windows: the Application, Security, and System logs. It also includes two new logs: the Setup log and the ForwardedEvents log. Windows logs are intended to store events from legacy applications and events that apply to the entire system.

1. Application log

The Application log contains events logged by applications or programs. For example, a database program might record a file error in the application log. Program developers decide which events to log.

2. Security log

The Security log contains events such as valid and invalid logon attempts, as well as events related to resource use, such as creating, opening, or deleting files. Administrators can specify what events are recorded in the security log. For example, if you have enabled logon auditing, attempts to log on to the system are recorded in the security log.

3. Setup log

The Setup log contains events related to application setup.

4. System log

The System log contains events logged by Windows system components. For example, the failure of a driver or other system component to load during startup is recorded in the system log. The event types logged by system components are predetermined by Windows.

5. ForwardedEvents log

The ForwardedEvents log is used to store events collected from remote computers.

4.8.1 Types of Event Logs

Information - An information event indicates an infrequent but significant successful operation. For example, when Microsoft SQL Server successfully loads, it may be appropriate to log an

information event stating that "SQL Server has started." Note that while this is appropriate behavior for major server services, it is generally inappropriate for a desktop application to log an event each time it starts.

Warning - A warning event indicates a problem that is not immediately significant but that may cause future complications. Resource consumption is a good candidate for a warning event. For example, an application can log a warning event if disk space is low. If an application can recover from an event without loss of functionality or data, it can generally classify the event as a warning event.

Error - An error event indicates a significant problem the user should know about, such as a loss of functionality or data. For example, if a service cannot be loaded as the system boots, it can log an error event.

Success audit (Security log) - A success audit event is a security event that occurs when an audited access attempt is successful. For example, a successful logon attempt is a success audit event.

Failure audit (Security log) - A failure audit event is a security event that occurs when an audited access attempt fails. For example, a failed attempt to open a file is a failure audit event.

Understanding an Event

Date - The date the event occurred

Time - The time the event occurred

User - The user who has logged onto the computer when the event occurred

Computer - The computer where the event occurred

Event ID - An event number that identifies the event type. Helps to know more about the event

Source - The source which generated the event. It could be an application or system component

Type - Type of event (Information, Warning, Error, Success Audit and Failure Audit)

Right click on event and click Event Properties :

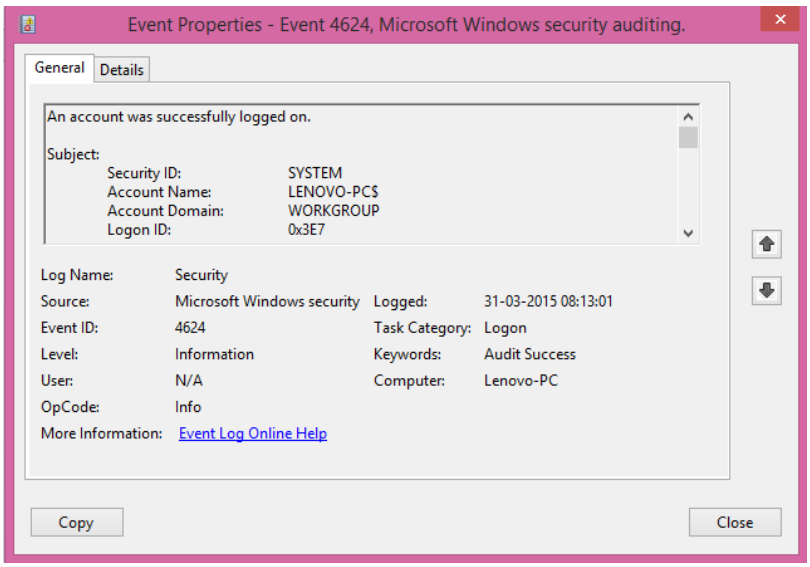


Figure : Event Properties

4.8.2 Understanding Event Logs

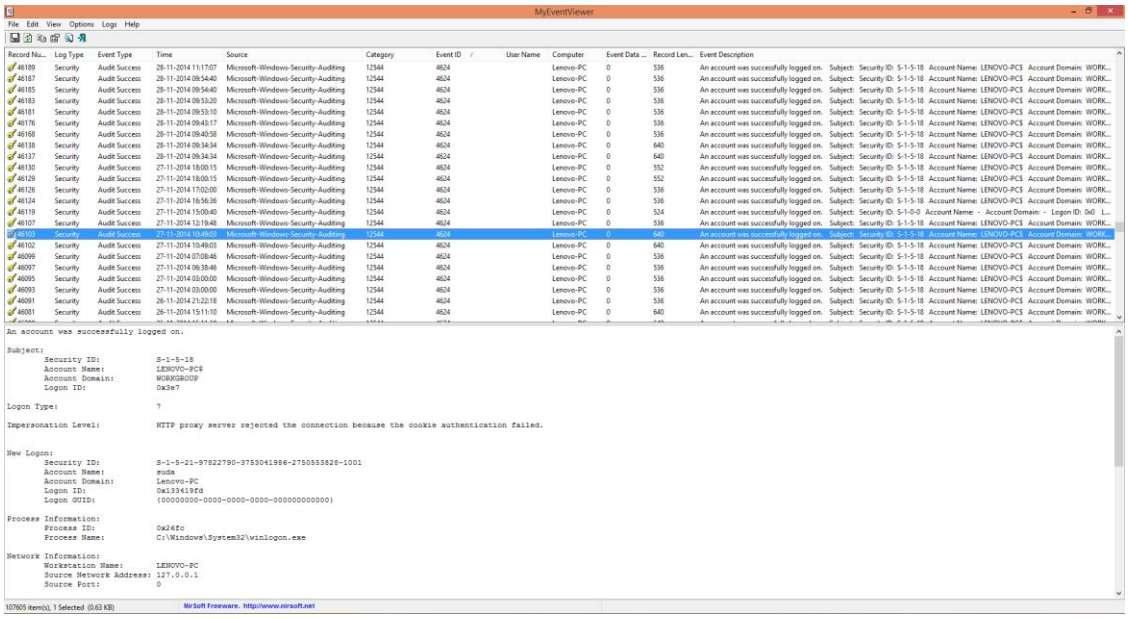


Figure: Event Log Description

Event highlighted in above figure is a Security (Audit success event) which has event id 4624 on computer name Lenovo-PC that describes successful account logon event on specific date time for specific account SID.

4.9 UNIX Logs

There are a wide variety of logging functions and services on UNIX. Some of these, such as the Solaris audit facility, are limited to a particular variety of UNIX. It is important that the digital forensics analyst become familiar with the logging deployed on the UNIX system that they are

reviewing. In particular, look at the syslog configuration file, the "/var/log" and "/var/run" directories and check if there are any remote log servers. Syslog is a network service that is most commonly run locally. This allows for the capability of sharing logs to a remote system.

Syslog and Other Standard Logs

There are five primary log files that will exist on nearly any UNIX system (the location may vary slightly). These have been listed in below.

The 5 primary Unix Log files:

- /var/log/btmp btmp contains the failed login history
- /var/log/messages is the default location for messages from the syslog facility
- /var/log/secure is the default log for access and authentication
- /var/run/utmp utmp contains summary of currently logged on users
- /var/log/wtmp wtmp details the history of logins and logouts on the system

/var/log/btmp: The bad logon attempt file ("/var/log/btmp") is a semi-permanent log (such as wtmp) that tracks failed login attempts. This file is a binary format and is read using the "*lastb*" command. In many systems, the btmp file will not be created by default. If this file does not exist the system will not log to it. Any review of a UNIX system should validate the existence of this file and ensure that it is functioning correctly. One way to validate that this file is working correctly is to attempt to log into the system using a set of invalid credentials. If the log is working correctly, an entry should be recorded noting the analyst's failed attempt. It is important that this file is restricted so only root can access or change it. General users have no reason to see failed attempts and should never change or delete this file.

/var/log/messages: The messages log ("/var/log/messages") or at times also the default syslog (on some systems this file will be named "/var/log/syslog") contains by default the sum of the system messages. Depending on the consideration of the syslog configuration file (commonly "/etc/syslog.conf"), this may contain failed drivers, debug information and many other messages associated with the running of a UNIX system.

/var/log/secure: The "secure" log ("/var/log/secure") is designed to record this security and authentication events that occur on the system. By default, applications such as TCPwrappers will log to this file. In addition, the PAM system and "login" facilities will write to this file on most UNIX systems.

/var/run/utmp: The utmp file ("/var/run/utmp") contains a point in time view of the users that logged on to the system. This file is used by a number of applications and utilities (such as the "finger" and "who" commands). This file is volatile in that it will not survive a system boot. Further, when the user logs out of the system their entry is removed. This file does not contain historical data. It is possible to gain a snapshot of user information at a point in time through this file. This information includes the username, terminal identifier, the time that the user logged in to the system and also where they logged in from (which may be a local TTY or remote network host). Many rootkits will change the contents of this file in an attempt to hide themselves.

/var/log/wtmp: The wtmp file ("/var/log/wtmp") is a binary file similar to "utmp". This file is also utilized by applications such as "finger", "last" and "who" and contains much of the same information as "utmp". The primary difference however is that it is more permanent in nature. This file provides a formal audit trail of user access and will also record system boots and other events. This file is commonly used when investigating an incident. The "last" command uses this file to display a list of accesses to the system. It will display a historic list as well as listing any user who was still logged onto the system. Like many other UNIX logging facilities it must be activated.

Most UNIX systems (and any that are configured correctly) will rotate logs periodically. This may be done through an automated facility such as "cron" or through some other application. It is important to both verify and validate how the log files are being rotated, whether they are being stored in an offline facility, are they being backed up and lastly that they are maintained online for an adequate period of time. Regulatory standards such as PCI-DSS version 1.1 require that system logs are not only maintained, but are accessible online for a minimum period of time (in this case 90 days). The analyst should ensure that all log files meet the minimum requirements for storage. In addition, always consider long-term data retention needs and the capability to restore logs after an extended period of time. Such log recovery may require that hardware and software associated with the previous system are maintained for a fair number of years.

4.10 Application Logs

Most applications generate logs of important events, including records of network-based access, debugging messages, and routine startup/shutdown logs. Application servers include:

- Web servers
- Database servers
- Mail servers
- DNS servers
- VoIP servers
- Firewalls
- Logging servers
- Authentication servers
- Filesharing servers

Application servers are constantly evolving new features and changing output formats to keep up with the latest developments in hardware, software, and protocol specifications. As a result, application log contents and formats are also constantly changing. Although some applications create logs in well-documented, published formats, network forensic analysts constantly run into new log formats, or formats that are simply not well documented. In some cases, application logs in real life are just different from the documented format or outdated. Furthermore, many applications allow local system administrators to customize log contents and formatting, in which case there may be no documentation at all as to the meaning of the output (other than what is listed in the application configuration).

Example : SMTP Logs

Here is an example of logs from a Postfix/SMTPD mail server. The logs begin by recording a message that was sent from "username@example.com" on the local system to "sn34kyg33k @gmail.com." You can see that it was successfully sent through Google's mail server, 209.85.222.47. Subsequently, an unknown system (201.250.45.83, registered at the time to the ISP "Telefonica de Argentina") unsuccessfully attempted to connect to the SMTP daemon on "bigserver" and relay a message from "onotherplap3@mail15.com" to "tofasllls@mail15.com."

(It is likely that this was an attempt to send a SPAM message.)

Sep 23 14:33:45 bigserver postfix/pickup [25480]: 9612218 C069: uid =1001 from=< username@example.com >

Sep 23 14:33:45 bigserver postfix/cleanup [26011]: 9612218 C069: message -id = <20090923203345.9612218 C069@bigserver@example.com >

Sep 23 14:33:45 bigserver postfix/qmgr [24702]: 9612218 C069: from=< username@example.com >, size =1060 , nrcpt =1 (queue active)

Sep 23 14:33:45 bigserver postfix/pickup [25480]: A501018C062: uid =1001 from=<
username@example.com >
Sep 23 14:33:45 bigserver postfix/cleanup [26011]: A501018C062: message -id =
<20090923203345. A501018C062@bigserver@example.com >
Sep 23 14:33:45 bigserver postfix/qmgr [24702]: A501018C062: from=<
username@example.com >, size =1064 , nrcpt =1 (queue active)
Sep 23 14:33:45 bigserver postfix/pickup [25480]: B3A3718C06B: uid =1001 from=<
username@example.com >
Sep 23 14:33:45 bigserver postfix/cleanup [26011]: B3A3718C06B: message -id
= <20090923203345. B3A3718C06B@bigserver@example.com >
Sep 23 14:33:45 bigserver postfix/qmgr [24702]: B3A3718C06B: from=<
username@example.com >, size =1084 , nrcpt =1 (queue active)
Sep 23 14:33:46 bigserver postfix/smtp [26064]: B3A3718C06B: to=<
sn34kyg33k@gmail.com >, relay=aspmx.l.google.com [209.85.222.47]:25 , delay =0.57 ,
delays =0.04/0.01/0.14/0
.37, dsn =2.0.0 , status=sent (250 2.0.0 OK 1253738039 13 si3213003pzk .59)
Sep 23 14:33:46 bigserver postfix/qmgr [24702]: B3A3718C06B: removed
Sep 23 15:19:55 bigserver postfix/smtpd [26160]: warning: 201.250.45.83:
hostname 201 -250 -45 -83. speedy.com.ar verification failed: Name or service not known
Sep 23 15:19:55 bigserver postfix/smtpd [26160]: connect from unknown [201.250.45.83]
Sep 23 15:19:56 bigserver postfix/smtpd [26160]: NOQUEUE: reject: RCPT from unknown
[201.250.45.83]: 554 5.7.1 <tofasllls@mail15.com >: Relay access denied;
from=<onotherplap3@mail15.com > to=<tofasllls@mail15.com > proto= SMTP helo=<none >
Sep 23 15:19:57 bigserver postfix/smtpd [26160]: disconnect from unknown
[201.250.45.83]
Sep 23 15:23:17 bigserver postfix/anvil [26163]: statistics: max connection rate 1/60s for
(smtp :201.250.45.83) at Sep 23 15:19:55
Sep 23 15:23:17 bigserver postfix/anvil [26163]: statistics: max connection count 1 for
(smtp :201.250.45.83) at Sep 23 15:19:55
Sep 23 15:23:17 bigserver postfix/anvil [26163]: statistics: max cache size 1 at Sep 23 15:19:55
Sep 23 15:19:57 bigserver postfix/smtpd [26160]: disconnect from unknown [201.250.45.83]

Below is a snippet of the mail server's error log, which contains error messages generated by

the Postfix service. These include records caused by mistakes in the local “mail” command usage, permission errors, and more.

Sep 20 21:53:09 bigserver postfix/sendmail [10815]: fatal: usage: sendmail [options]
Sep 20 22:27:48 bigserver postfix/sendmail [10961]: fatal: Recipient addresses must be specified on the command line or via the -t option
Sep 20 22:27:48 bigserver postfix/sendmail [10963]: fatal: Recipient addresses must be specified on the command line or via the -t option
Sep 20 22:28:29 bigserver postfix/sendmail [10979]: fatal: Recipient addresses must be specified on the command line or via the -t option
Sep 22 13:04:31 bigserver postfix/sendmail [24424]: fatal: usage: sendmail [options]
Sep 22 15:32:07 bigserver postfix/postmap [25785]: fatal: open database /etc/ postfix/generic.db: Permission denied
Sep 22 15:55:40 bigserver postfix/postmap [26209]: fatal: open database /etc/ postfix/virtual.db: Permission denied
Sep 22 17:01:33 bigserver postfix [27072]: error: to submit mail , use the Postfix sendmail command
Sep 22 17:01:33 bigserver postfix [27072]: fatal: the postfix command is reserved for the superuser

4.11 Network Logs

Many devices and computers in a network generate logs of events and activities. As such, log files serve as a primary source of evidence in network investigations. There are several different types of log files. Some of the logs of interest include authentication, application, operating system, and the firewall log. An **authentication log** identifies the account (and IP address) connected to a particular event. **Application logs** record the date and time as well as the application identifier. The date/time stamps indicate when the application was started and how long it was used. **Operating system logs** track system reboots as well as the use of different devices. The operating system logs are useful in recognizing patterns of activity as well as anomalies (unusual occurrences) in the network. **Device logs** such as those generated by routers and firewalls are also worth examining.

There are some things to keep in mind with log files. Log files can change or disappear pretty rapidly. They can be purged at regular intervals to help keep storage space free. There’s also a good chance that not all of the relevant logs will be in your possession. Attacks that originate

outside of your organization will pass through devices under the control of a third party, such as an Internet Service Provider (ISP). These logs may have to be subpoenaed, which can take some time. ISPs won't likely hang onto these logs forever. They likely have document retention and destruction policies in place controlling what gets kept and for how long. Lacking a clear need or reason to keep it, those logs will be destroyed.

The router logs can contain much information of interest. Some of the things we can uncover are:

Requested Uniform Resource Locators (URLs)

Server Name

Server IP Address

Client's URL

Client IP Address

Who logged in and when

When attempting to collect evidence from a router, it's very important to minimize any interaction. Instead of accessing the router through the network itself, it's a better option to go through the router's console. Remember, our objective is to observe and record what we find, not to alter or change anything. To that end, we should avoid any command that could potentially modify any of the data. A configuration command, for example, is one that should be avoided. The "show" command is a much better option. Here are a couple of examples of "show" commands:

>(router name)#show clock detail—Displays the system time

>(router name)#show users—Displays the users that have access to the router

4.11.1 Network Investigative Tools

The actual traffic (packets) moving on the network can hold some valuable clues. There are several tools, called "sniffers," available that can capture and analyze network traffic. Some of these tools include:

Wireshark (www.wireshark.org)

NetIntercept (http://www.niksun.com/product.php?id=16)

Netwitness Investigator (http://www.netwitness.com/products-services/ investigator)

Snort (http://www.snort.org/)

Capturing network traffic can yield some great clues. For instance, we can determine what files have been stolen, what commands were executed, as well as any malicious payload that was

delivered. From a legal perspective, it's important to realize that monitoring network traffic in certain instances can be considered wiretapping.

4.12 Linux OS Artifact Analysis

1. Linux Boot Processes and Services

The Linux boot process with which you should be familiar is a valuable target for malicious modification. The main reason is to obtain persistent access to the system. Altering or setting up a script modifying the boot process is not hard if sufficient privileges are available. Therefore, when starting an investigation, reviewing all scripts involved in the boot process is advisable.

2. User Accounts

As partly explained in the book, the user management files used in a Linux system are passwd, shadow and groups which can all be found in /etc/. In the following, we will explain the passwd and shadow files in more detail. The passwd file contains the following information as shown using Alice's entry on her VM.

```
...
alice:x:1000:1000:Alice,,,:/home/alice:/bin/bash
...
```

where the following information is stored. Note that we label the fields of an entry separated by a colon from left to the right.

1. Username
2. Hashed password field
3. User id
4. Primary group id
5. Comma-separated "GECOS" comment field8
6. Path of the user's home directory
7. Program run upon initial login (mostly default shell)

Note that non user accounts such as mysql do not have a valid password hash stored, e.g., "!", "!!" or "*" which indicates that this "user" should not be able to login. Furthermore, they usually have an invalid initial program, e.g., /bin/false.

When investigating an incident, it should be checked whether a non user account has a valid hashed password field or a valid initial program, respectively. Historically, the passwd file was also used to save the hashed passwords. However, it was later decided to move the hashed

passwords into the shadow file since the passwd file needs to be readable by all users. Indeed when reading Alice's shadow file we see the entry:

...

alice:6elkQzD...wLqUEriJ.:14792:0:99999:7:::

...

where the following information is stored

1. Username
2. Hashed password
3. Number of days since the Unix epoch (1 Jan 1970 UTC) until the password was last changed.
4. Minimal amount of days between password changes
5. Maximum time the password is valid
6. Number of days prior to expiration to warn users
7. Expiration date
8. Reserved

Note that some Linux distributions generate backup copies of these files when a user or group is either added or modified. The backup files will have a dash sign or a minus appended. Those backups can be useful during an investigation if a user has been added to the system as using *diff* on different versions of those files immediately provides a list of users added.

3. Shell History

Usually, commands typed in any shell session will be stored in the user's home directory in .bash history. Obviously, the bash history provides valuable information. However, the bash history only contains a pure list of issued commands and no additional information such as timestamps whatsoever. Therefore, it is essential to correlate the information gained from the history with other information such as file system or log file time information. In particular, this is true for commands where the issue time is important to the investigation. What is more, if the history file does not exist or if it has been replaced by a link to /dev/null, these are valuable clues that the system might have been compromised. The history can be shown using the command history. Note that deleting the whole history is not possible by simply deleting the .bash history file.

4. SSH

The .ssh folder in the home directory contains the known hosts file.Whenever ssh is used to

connect to a new host the host's name or IP address and the host's public key are stored in this file.

5. GNOMEWindow Manager Artifacts

There are a lot of different files created on a Linux System with a default GNOME installation. For instance, subdirectories for any media handled by the GNOME automounter can be found in ***/home/$USERNAME/.gconf/apps/nautilus/desktop-metadata/.***

Each of these subdirectories contains a %gonf.xml file. As you see, the xml files contain an entry for the *nautilus icon position*, the *icon scale* and a timestamp. Note that all entries have a timestamp themselves (*mtime*).Of particular interest is the changed timestamp of the file itself as it shows the time a device has been connected to the system.

Use date -d @TIMESTAMP to convert a timestamp given in UNIX time into human readable format.Make sure to take into account the timezone of a given time when comparing.

6. Log Files

The information available from the logs stored in /var/log/ is very valuable. However, depending on the kind of the investigated incident, the logs might have been altered in a way which either destroyed artifacts or which even produced faked or misleading artifacts. Note that the user login log files located in /var/log/wtmp and /var/log/lastlog from a different filesystem can be shown using the command last with the option -f and passing the path to the log file. Other important log files already mentioned are syslog, messages and auth.log.

7. Syslog

The bulk of system logs on a Linux system are stored under the "/var/log" directory, either in the root of this directory or in various subdirectories specific to the application generating the logs. Syslog operates on client/server model, which enables events to be recorded to a remote, dedicated syslog server. However, on a standalone Linux system, events are usually written directly to the files on the local host. Syslog uses a "facility/priority" system to classify logged events. The "facility" is the application or class of application that generated the event.

Syslog events are a single line containing made up of five fields:

1. Date of message creation

2. Time of message creation

3. Host name of the system creating the log entry

4. Process name creating the log entry

5. Text of the log entry

This uniform logging format makes searching for log entries of note on a Linux system relatively easy. It is important to note that most Linux systems implement some level of log rotation. For example, a default Ubuntu Linux desktop installation rotates logs every month, compressing the older log file with GZip for archival. Server systems will likely archive logs more rapidly and are more likely to delete logs from active systems after a shorter retention period.

4.13 MAC OS Artifact Analysis

Like Linux systems, OS X places all volumes under a single unified namespace below the root directory "/". Immediately beneath the root directory are the following directories:

Applications—This directory is the standard location for all installed OS X applications. Generally this directory will hold applications designed to be launched interactively via the OS X GUI.

Library—Library directories hold supporting data that may need to be modified during execution of a program. This generally includes things such as preferences, recent items, and other similar data. The root Library directory contains system-wide configuration data. User-specific data are stored under user Library directories.

Network—Directory for items in the Network domain generally empty.

System—This directory contains operating system-specific data, somewhat analogous to contents of the system32 directory on a Windows system.

Users—The parent directory for user home directories.

Volumes—The parent directory for mounted volumes similar to /mnt or /media on Linux systems.

bin and **sbin**—These directories contain command-line utilities included in the OS X system.

private—This directory contains (among other things) OS X versions of **/var, /tmp**, and **/etc.**

1. Property Lists

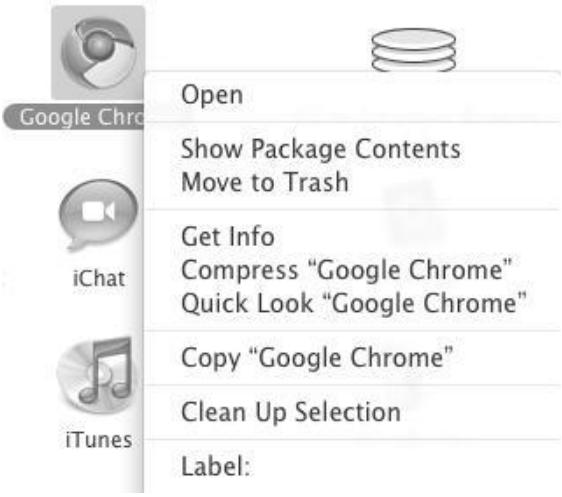
Many artifacts of interest on an OS X system are stored as *property lists* or *.plist* files. There are two types of property lists: plain text, XML property lists and binary property lists. Plain text plist files can be examined directly or viewed in any XML display program. Binary plists need to be converted to plain text prior to analysis. Due to being more compact than their plain text

equivalent, binary plists are becoming used more and more commonly. A typical OS X install will have thousands of plist files, so knowing which will be relevant to your examination is key.

Many OS X-native, closed source utilities interact with plist files, including a basic command line utility called plutil. Fortunately, Pete M. Wilson has developed a Perl script that can convert binary plist files to their plain text equivalent. His script, plutil.pl, is quite simple to use. When provided with the name of a binary plist file, plutil.pl parses the file and outputs a plain text version in the same directory.

2. Bundles

On an OS X system, the "applications" users generally interact with are not monolithic files at all, but are in fact special directories known as *bundles*. Bundles are directories that have their contents hidden from the end user's view by the operating system. Opening or double-clicking on an application bundle is enough to execute the application, but the actual executable code is several directories down from the bundle itself. Drilling down into the bundle via the OS X GUI can be performed via the right-click menu as shown in Figure below:



When examining an OS X system via the Sleuth Kit, the bundle will just be treated as a standard directory. Application bundles can be identified in a file listing by the ".app" extension in the directory name, and a set of subdirectories similar to the following:

Contents

Contents/Info.plist

Contents/MacOS

Contents/PkgInfo

Contents/Resources

Contents/Versions

The actual executable code is normally stored in the MacOS directory, with the other directories holding supporting data such as icons, text files, and the like.

3. System Startup and Services

On system boot, the bootloader boots the OS X kernel (/mach_kernel), which then runs the launchd process. Launchd serves as a replacement for init and the init scripts process found on Linux systems. The launchd takes its tasking from four directories. System tasks that run in the background are read from */System/Library/LaunchDaemons* and */Library/LaunchDaemons*, while user-interactive launch tasks are read from */System/Library/LaunchAgents* and */Library/LaunchAgents*. Launchd will read and process plists in these directories, launching the appropriate applications. Note that all of the plist files in these directories should be in the plain XML format and thus do not require conversion before examination.

4. Kexts

OS X has the capability to load additional functionality into the kernel via *kernel extensions*. Kernel extensions are bundles with the extension *kext* and can be found in the /System/Library/Extensions directory. There are many Apple-provided kernel extensions in this directory, and there may be extensions for third-party hardware devices or programs that require low-level access, such as disk encryption software.

5. Network Configuration

The bulk of local system network configuration information on an OS X system is stored in various plist files under /Library/Preferences/SystemConfiguration. The *preferences.plist* file contains general settings for all the network interfaces on the system, as well as location-specific network profile information if this feature is in use. In addition, this file also shows the hostname of the computer, which may be important in a network-related examination.

```
<dict>
<key>ComputerName</key>
<string>forensic-macpro</string>
<key>ComputerNameEncoding</key>
```

<integer>1536</integer>
</dict>

The *com.apple.network.identification.plist* file is a rich source of historic network information. Among other data, this file contains a running list of previously assigned network addresses with time stamps. This information is obviously invaluable during investigation of a mobile laptop.

6. Hidden Directories

In addition to special bundle directory types that hide their contents from the user by default, OS X honors traditional Unix-style “dotfile” hiding. Files and directories named with a leading dot will be hidden from the user’s view by default. There are not as many of these hidden files on OS X systems as there are on a standard Linux installation but some are present.

7. Installed Applications

/Library/Receipts contains information about applications installed via the OS X Installer system. This directory contains various “pkg” bundles, which contain information about the installed package. The creation times of these directories should correspond with the date that the software was installed.

8. Swap and Hibernation Data

OS X stores swap files and hibernation data under the /private/var/vm directory. Depending on how heavily used (and short on resources) the system is, there will be anywhere from 1 to 10 numbered *swapfile* items in this directory. These contain paged out sections of memory and can persist on disk for some time. Additionally, if hibernation is enabled, a *sleepimage* file will be present. This file will be the same size as the available RAM on the system and will contain a copy of memory as it existed the last time the system was put to sleep. Any techniques used for processing unstructured data streams are applicable to these files, including string extraction and file carving.

9. System Logs

OS X shares many logs with other Unix-like operating systems, such as Linux. In general, BSD and Linux-derived applications for OS X will store logs under /private/ var/log. OS X will generally have a syslog daemon running and will generate many of the same logs (including syslog-derived logs) that a standard Linux system does. Of the most interest to the examiner are

the logfiles shown in Table, all found under /private/var/log.

System-wide software created specifically for OS X will generally store logs under /Library/Logs. Examples include the "*Software Update.log*" file, which tracks when system software updates are applied, and the *CrashReporter*, *HangReporter*, and *PanicReporter* logs, which record application and kernel errors and make contain system state information related to the time they were generated.

fsck_hfs.log	Record of all HFS/HFS+/HFSX volumes attached to the system
system.log	A catch-all log file, equivalent to messages on Linux
secure.log	Records all system authentication, including screensaver unlocks and SSH access.

4.14 Anti - Forensic Techniques

4.14.1 Introduction

Computers have become an integral part of our day to day life. They are being used in every walk of life and have made life convenient for everyone. However, different capabilities of computer have also equipped criminals with technology that can be misused. Now a crime can be surfaced either without leaving any evidences or by removing them such that they are untraceable. Hence, computer crime is any crime which involves usage of computer or any other digital device making it a subject of investigation.

Web definition of forensics explains it as methods and techniques used in investigation of crime. Therefore computer forensics refers to a computer system which has been seized from a crime scene and undergoes forensic analysis in search of substantial evidence. Computer forensics is necessary not only to collect evidences of crime from computer but also to validate them so that they are admissible to the court of law.

With most of the crimes taking place using digital devices, courts of law all around the world are giving equal importance to digital evidences as eye witnesses. Therefore, it becomes crucial for both criminals and investigators to handle digital evidences carefully. While investigators take measures to retrieve evidence from digital devices intact, criminals, on the other hand, take countermeasures trying to hide or even destroy the evidences making investigation even harder. For instance, a case of murder presented as a suicide case. The methods used by the criminal to

hide or tampering the evidences are anti-forensic techniques.

Hence, anti-forensics are the counter-measures taken to frustrate forensic investigation and evade from it. The main aim of anti- forensic technique is to prevent any crime evidence from getting caught. Application of these techniques is further encouraged by the availability of open source tools that have been designed keeping anti-forensics in mind.

4.14.2 Anti - Forensic Techniques

1. Artifact Wiping

This technique has been in use for long time now. Artifact wiping is used to attempt data sanitization, where data sanitization is the process of deliberately, and irreversibly removing or destroying the data stored on memory. It can be achieved in any of the following ways. It may seem that permanently deleting (shift+del) any file makes it irretrievable, but the fact is that all such files can be traced down and recovered during forensic analysis. It is because when files are deleted this way, they don‟t disappear from the hard drive but get de-allocated by the operating system.

Thus, unallocated or slack space is a space which is not allocated to any file by the operating system and may contain deleted files which can be overwritten by other files. Aware of this fact, tools have been developed which delete or wipe off the files and their traces. Various open source tools such as BC Wipe, Eraser, PGP Wipe are available for the purpose of data sanitization, and wiping slack and unallocated spaces. These tools destroy data files by using repeated overwrites which makes their retrieval very difficult.

Two sample files, sample1.pdf and sample2.pdf, were created to test artifact wiping. Eraser version 6.0.10 was used as counter-forensic tool. It is an open source tool and can be downloaded from http://eraser.heidi.ie/. Sample1 was permanently deleted (shift+del) while sample2 was deleted using Eraser. After running forensic analysis on the drive that contained these files, it was discovered that sample1 was still there on the drive and could be recovered whereas there was no trace of sample2 on disk image created by FTK Imager.

Artifact wiping is preferred as it is less time consuming and efficient, yet there are certain limitations to it. Some data wiping tasks are very difficult to achieve, for example erasing file data that is wholly contained in the master file table. Similarly deleting data from other file system structures such as journal file and page file is also challenging. Hence these tools are not able to successfully overwrite or delete the complete file, thereby leaving traces of the existence

of the file. It gives way for data recovery of the residual data. Artifact wiping tools when used even leave traces of wiping, which if identified may cause problem for the owner of the

seized system and question his claim to be innocent.

2. Data hiding

Data hiding is, perhaps one of the oldest methods around. This technique exploits the belief that a limited amount of time and resources can be devoted to one case, by the investigators with huge amount of material to search through given the size of hard drives in modern computers which is capable of storing huge amount of data. There are multiple ways of hiding data which ensure that data is undetectable while it is still present on the device. Some of the data hiding techniques are discussed as follows:

One way of hiding data is relocation of data. Target data is stored at a location, of which user is sure will not be examined by the investigator. Another way relocating data is to transfer data to any other portable storage device, and the wiping it off from the computer. This is a convenient technique, but relocating data leaves behind a record of data transfer in automated logs, while usage of portable device can be detected since computers typically update system logs whenever a data storage device is connected to it.

Second way is making data “invisible”. Data is made to be “invisible”, concealing the fact that the hidden data still exists. It can be achieved by either steganography or streaming. Steganography is a technique where information or files are hidden in another files. Generic steganography was used to hide data in images which could be detected easily. However, many advance techniques have been developed in steganography which embed text in various file formats such as graphic files, audio and video files.

W.Bender has discussed various techniques of steganography such as one in which data is hidden within the random texture pattern of a picture, another technique for embedding data in audio files using low-bit encoding and hiding data in text files using white space encoding. All these techniques are applied carefully keeping in mind the fact that, the changes made go unnoticed to human senses. Softwares for steganography are openly available. Steghide is one such program and is available at http://steghide.sourceforge.net/download.php.

Streaming files, on the other hand is a process in which user can associate more than one file with table entry of a single primary file. Therefore, when forensic tools are run, they will display the

primary file associated with each file table entry, whereas any other files, if exists, are hidden in the unallocated space of hard drive and remain undetected to the investigator. However, streamed files can still be spotted by keyword or other searches, if they are not compressed or encoded before streaming.

Finally, data can also be hidden by altering file extensions. It is known that different files in computer are identified by their file extensions. File extension is a suffix to the computer file name indicating the file format of its content or usage. For example, file extension .doc identifies the file as a word document, or .xls is used for an excel file. A text file can be hidden by altering its extension from .txt to .exe to make it appear to be an executable program file. It relies on the fact that windows identify files by their file extension and does not open the file unless its extension identifies it as such. Therefore, even if the investigator is able to access the file, he will not be able to open it. Although this is an old technique for hiding data from being caught, it may still effective and can bypass forensic analysis.

However, altered file extensions gain attention of investigators who make sure that all the file extensions on a seized computer match their actual file types. This process is also called signature analysis. Error message was displayed when the file was tried to open. It indicated that signature of the file didn‟t match its extension thereby raising suspicions. Signature analysis plays a significant role and is a routine part of any investigation for most of the forensic investigators.

3. Trail obfuscation

This technique is more popularly known as counterfeiting‟. Dictionary meaning of counterfeiting is, to imitate usually with the intent to forge. Therefore trail obfuscation or evidence counterfeiting techniques are practiced with purpose of confusing and disorientate the investigation. This could be achieved in a number of ways.

One way to counterfeit evidences is Defragmentation. This technique uses the concept of artifact wiping with the aim of confusing investigation. Fragmentation of files become necessary when hard drive of computer gets almost full. In such case, it becomes very difficult to store large files in one contiguous space. Consequently, file is then stored at various locations part by part . This process, however, slows down the computer to a great extent because to load one file in memory, hard drive is required to be searched at different locations and assemble it. It becomes problematic when the process has to be repeated on a routine basis for a number of system files which are fragmented similarly. Hence defragmentation is carried out to reorganize and rearrange

the hard drive so that all the files with all parts are stored in contiguous space. This process involves rewriting and erasing of files all over the disk. This causes disruption to data in allocated spaces which is likely to destroy any evidential remains written in those locations. Hence defragmentation doesn‟t delete files unlike artifact wiping but attempts to confuse the investigation by destroying any residual data that may be present. Many defragmenting tools are available which could be downloaded for free from internet. TuneUp Drive Defrag from AVG is one such example. This technique gives desired output only in case when most of the hard disk is full.

Applying defragmentation is very typical, but raises suspicions when it is discovered that it was carried out after the system became subject of forensic analysis. Moreover, excessive use of defragmenting tool may also lead to strengthen the suspicion since it is described as a healthy practice by Microsoft technical documents, but only when it is used annually or semi-annually. Hence frequent usage of defragmenter on subject computer is treated as an attempt to destroy evidences and mislead investigation.

Another way which is also more effective and popular is to attack at the credibility of the evidence in question and render it useless to be presented in the front of the legal system. One most common way of doing this is modification of metadata. Metadata is data about data that is generated when a file is created. Time stamps are a type of metadata which indicates when a document was created, last accessed or last modified.

This information is also called as MACE values such that M stands for last write time, A stands for last access time, C stands for creation time and E stands for change time.The ability to modify time stamps is a technique which can be troubling and make investigation even harder. Now let us consider a case of suicide where the suicide note retrieved is a hard copy generated from subject‟s computer and it outlines the date and time when the letter was supposedly written. If file's MACE values are not altered, the letter must be written before the estimated time of the questionable death. If, after digital forensics investigation, metadata obtained from the computer indicates that file was created after the time indicated in printed copy, it would certainly raise questions.

Investigators can sense hints of crime which was committed and then tried to be covered. Time stamp metadata can be easily altered using open source tools available namely Metasploit Timestomp (http://www.rapid7.com/products/metasploit/download.jsp) and File Touch. Both of

these tools are used to modify time stamp metadata of NTFS files and disrupt the forensics investigation right from the very initial stages.

Before using Timestomp utility, meterpreter session is required to be opened. Timestomp is available in the meterpreter session. All the files in the current directory are listed using –ls command. Then target file is selected and its MACE values are checked using –v command. These values can then be altered using any command options available after typing –h command.

Although the tools discussed above and other that are available are able change time stamps of NTFS file system and are very effective from anti-forensics point of view, there are yet other ways to detect any kind of tampering with temporal evidences. Time based data forensics can detect evidence tampering by analyzing both the discrepancies and similarities among various temporal evidences associated with file metadata and registry entries. It is called crossreference time-based forensic scheme which not only identifies malicious activities such as illegal access and modification of file but also detect tampering of file timestamps.

4.15 Forensic Tool Development

The fourth and final unaddressed research theme is the design and implementation of digital forensic tools. Many of the respondents believed that current tools were somewhat limited in terms of their ease of use and software engineering. Ease of use is a major issue. Tools must not be too technical and must have intuitive interfaces, but, at the same time, they should be customizable for use by skilled practitioners. Furthermore, the goal should be to provide information and knowledge, not merely data. This might be accomplished through data visualization, automated link analysis, cross-correlation and features for “zooming in” on information to reduce information overhead.

Another approach is to shift from the tradition of presenting data hierarchically based on file system relationships to presenting data temporally. The digital forensic research community should consider, extend and adapt approaches devised by graphics and visualization and human-computer interaction researchers. The respondents also suggested several improvements with respect software engineering. Software development must take advantage of hardware advances, including massive parallelism and streaming. Increased interoperability via standardized data (i.e., tool input/output) and API formats is needed. More operating-system-independent tools (e.g., PyFlag) are required. All-in-one tools with respect to data types are needed. Such tools would intelligently leverage data from static media, volatile

memory/devices, network dumps, etc. It is also important to increase the automation of forensic processes.

These suggestions beg the fundamental question: Why are digital forensic tools not there yet? Is it a symptom of the relative nascence of the field and, thus, the tools? Or, are digital forensic analytical tasks fundamentally different from tasks in other domains where similar technologies work? If so, are the differences regarding the human analytical sub-task, the computational sub-task, or both? Is this even a valid research stream for digital forensic researchers, or should it be left to commercial software developers? Do these questions necessitate research, or simply awareness of the problem on the part of tool developers? In any case, these questions must be considered and collectively answered by the digital forensic community.