Windows Triage Forensics Methodology

Version 1.5
Date: 5-3-2017
POC: @Wired_Pulse

Date	Version	Change log	Pages	By
5-28-2016	1.0	Document creation	All	@Wired_Pulse
6-28-2016	1.1	Added vol.py timeline, common processes table, and BareMonkey script	12, 57, 58	@Wired_Pulse
6-29-2016	1.2	Added forensics report template	58	@Wired_Pulse
7-5-2016	1.3	Added PowerShell profiler script to help automate analysis and Slueth Kit Autopsy information	59	@Wired_Pulse
9-14-2016	1.4	Moved scripts from this document to my github (github.com/wiredpulse).	All	@Wired_Pulse
5-2-2017	1.5			@Wired_Pulse

Table of Contents

Methodology

The forensics methodology encompasses the following steps:

1) Note the state of the machine upon your arrival.
2) Using known good programs, output to a file the running processes, connections, and sessions.
3) Image the RAM.
4) Check for disk encryption. Edd.exe could be used but be aware that it only checks for TrueCrypt, PGP, Bitlocker, Safeboot, BestCyrpt, Checkpoint, Sophos, or Symantec.
5) If encryption is detected, get an image of the live logical drives. If no encryption exists, create a custom image for initial triage and then get a full disk image.

System Recommendations

It is recommended that you use the Windows and Linux SANS Investigative Forensics Toolkit (SIFT) distribution in order to facilitate your forensic needs. SIFT comes with tools listed in the tools section and plenty of other ones that will aid you in your endeavor.

We recommend that you set your forensic analysis machine time zone to UTC. This will provide an easy format to perform analysis in and reduce exposure to conversion errors or misinterpretation.

Tools

A lists of tools mentioned throughout this document are listed below.

Autopsy
http://www.sleuthkit.org/autopsy/
Belkasoft RAM Capturer
https://belkasoft.com/ram-capturer
Bulk Extractor
http://digitalcorpora.org/downloads/bulk_extractor/
DCode
http://www.digital-detective.net/digital-forensic-software/free-tools/

Event Log Explorer
http://eventlogxp.com/
Encrypted Disk Detector (EDD)
https://www.magnetforensics.com/free-tool-encrypted-disk-detector/
FTK Imager
http://accessdata.com/product-download/digital-forensics/ftk-imager-version-3.4.2
Recbin
http://www.mitec.cz/wfa.html
Ex: recbin –f "[root]\recycler\<SID>\INFO2"
Redline
https://www.fireeye.com/services/freeware/redline.html

RECmd
https://github.com/EricZimmerman/RECmd
Registry Explorer
https://github.com/EricZimmerman/RECmd
Registry Viewer
http://accessdata.com/product-download/digital-forensics/registry-viewer-1-8-0-5
RegRipper
https://github.com/keydet89/RegRipper2.8

1. Mount the drive containing the SAM hive or have it accessible to the system by other means
2. Open RegRipper
3. Hit Browse and navigate to the SAM hive
4. Click Ok
5. Input a name and location for a Report file
6. For plugin, select the hive in which you are parsing
7. Click Rip It
8. Once complete navigate to the output location or click View Now

SAMInside
http://saminside.findmysoft.com/

Shellbags Explorer
https://binaryforay.blogspot.com/2015/02/shellbags-explorer-v0520-released.html

Structured Storage Viewer
http://www.mitec.cz/ssv.html
SQLite Manager
https://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/

The Slueth Kit
http://www.sleuthkit.org/

Volatility
http://www.volatilityfoundation.org/
Win Prefetch View
http://www.nirsoft.net/utils/win_prefetch_view.html
Windows Jump List Parser (jmp.exe)
https://tzworks.net/prototype_page.php?proto_id=20

Ex: dir “E:\users\<some_user>\appdata\roaming\microsoft\windows\recent\customdestinations\*ions-ms” /b /s | jmp.exe –pipe –csv –timeformat hh:mm:ss –

no_whitespace

Windows LNK Parsing Utility (lp.exe)
https://www.tzworks.net/prototype_page.php?proto_id=11

Lp.exe will provide the drive letter and name the file it was executed from.
Ex: dir “E:\[root]\users\<some_user>\appdata\roaming\microsoft\windows\recent\*.lnk”

/b /s | lp.exe –pipe –csv –no_whitespace –timeformat hh:mm:ss > output_lnk.csv

Windows ShellBag Parser (sbag.exe)
https://www.tzworks.net/prototype_page.php?proto_id=14

Sbag.exe ex: sbag.exe "<path-to>\usrclass.dat" –base10 –csv –timeformat hh:mm:ss –no_whitespace > c:\sbag.csv

YARU
https://tzworks.net/prototype_page.php?proto_id=3

Memory Capture

FTK Imager

1) Open FTK Imager.
2) Click File > Capture Memory.
3) Click Browse and select a destination.
4) Click Ok.
5) Alter the destination filename, if need be.
6) Click Capture Memory.
7) Click Close when the capture is complete.
8) Create a hash of the capture using PowerShell's 'Get-FileHash' cmdlet.

Belkasoft RAM Capturer

1) Double-click the RAMCapture64.exe.
2) Take note of where the capture will be saved and hit the Capture button.
3) Click Close once the imaging is complete.
4) Create a hash of the capture using PowerShell's 'Get-FileHash' cmdlet.

Hibernation File

1) Check for encryption first using edd.exe.
2) Hibernate the system.
3) A hiberfile.sys file will be located at c:\
4) Create a hash of the capture using PowerShell's 'Get-FileHash' cmdlet.

Crash Dump

1) Open regedit.exe.
2) Navigate to HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Services\kbdhid\Parameters and add the value called namedCrashOnCtrlScroll with a REG_DWORD value of 0x01.
3) Reboot the machine and hold down the rightmost CTRL key and press the SCROLL LOCK key twice.
4) A .dmp file will be written to C:\Windows\Minidump or C:\Windows.
5) Create a hash of the capture using PowerShell's 'Get-FileHash' cmdlet.

.vmem

1) Suspend the machine.
2) A .vmem will be created in the folder the virtual machine resides.
3) Create a hash of the capture using PowerShell's 'Get-FileHash' cmdlet

Memory Analysis

Formats

Redline and Volatility has the ability to analyze multiple memory formats. A quick list of the memory capture formats, highlighted in this guide, are below with their capability of the programs we will use to analyze memory.

Converting the images to a different format can be completed using Volatility with the below syntax.

```
        Vol.py  imagecopy -f <capture_to_convert> -O <name_new_file_.img> --
profile=<profile_name>
```

Redline

- FTK Imager (.mem)
- Hibernation File (hiberfile.sys) – After conversion to .img using Volatility's imagecopy
- Crash Dump (.dmp) – After conversion to .img using Volatility's imagecopy
- Belkasoft RAM Capturer (.mem)

Volatility

- VMWare (.vmem and .vmsm)
- Belkasoft RAM Capturer (.mem)
- Hibernation File (hiberfile.sys) – After conversion to .img using Volatility's imagecopy
- Crash Dump (.dmp)

Analysis Approach

Plug-ins

Memory analysis can be generally broken up into six steps:

1) Identify Rogue Processes
2) Analyze Process DLLs and Handles
3) Review Network Artifacts
4) Look for Evidence of Code Injection
5) Check for Signs of a Rootkit
6) Dump Suspicious Processes and Drivers

Below are the recommended plugins for Volatility that cater to the above approach.

> Note: Depending on your forensics system environment path, Volatility can be called upon by vol.py or volatility. For this example, I will reference it as vol.py.

Syntax example:
Vol.py -f <mem_capture> --profile=<profile_name> <plugin>
For a more convenient way to parse a memory capture using Volatility, see the Annex A of this document to look at BareMonkey.
Recommended plugins:

1) Processes
   - Pslist - Print all running processes by following the EPROCESS lists
   - Pstree - Print process list as a tree
   - Pstree –v - Print process list as a tree
   - Psscan - Pool scanner for process objects

2) Network
   - Sockets - Print list of open sockets
   - Sockscan - Pool scanner for tcp socket objects
   - Netscan (Vista or later) - Scan a Vista (or later) image for connections and sockets
   - Connections (XP or 2003)
   - Connscan - Pool scanner for tcp connections

3) Dlls and Handles
   - Dlllist - Print list of loaded dlls for each process
   - Handles - Print list of open handles for each process

4) Code Injection
   - Malfind - Find hidden and injected code
   - Ldrmodules - Detect unlinked DLLs

5) Rootkit detection
   - Psxview - Find hidden processes with various process listings
   - Apihooks - Detect API hooks in process and kernel memory
   - Driverscan - Pool scanner for driver objects
   - Mutantscan - Pool scanner for mutex objects

6) Dump Processes
   - Procdump - Dump a process to an executable file sample
   - Dlldump - Dump DLLs from a process address space
   - Moddump - Dump a kernel driver to an executable file sample
   - Memdump - Dump the addressable memory for a process

7) Services
   - Getservicesids - Get the names of services in the Registry and return Calculated SID
   - Svcscan- Scan for Windows services

8) Miscellaneous
   - Strings - Match physical offsets to virtual addresses (may take a while, VERY verbose)
   - Sessions - List details on _MM_SESSION_SPACE (user logon sessions)
   - Shellbags - Prints ShellBags info
   - Shimcache - Parses the Application Compatibility Shim Cache registry key

- Timeliner - Creates a timeline from various artifacts in memory
- Cmdscan - Extract command history by scanning for _COMMAND_HISTORY
- Consoles - Extract command history by scanning for _CONSOLE_INFORMATION
- Filescan - Pool scanner for file objects
- Getsids - Print the SIDs owning each process
- Filescan - Pool scanner for file objects

Processes

Process	Path	Parent	Number of	User	Start Time
System	--------------------------	-----------------	1	Local System	At boot
smss.exe	System32\smss.exe	System	1 master and 1 per session	Local System	Seconds of boot
wininit.exe	System32\wininit.exe	smss.exe but it exists	1	Local System	Seconds of boot
taskhost.exe	System32\taskhost.exe	services.exe	1 or more	One or more may be owned by logged-on users and/or by local service accounts	Varies
lsass.exe	System32\lsass.exe	wininit.exe	1	Local System	Seconds of boot
winlogon.exe	System32\winlogon.exe	smss.exe but it exists	1 or more	Local System	Seconds of boot for first 1. Others occur at each logon
crss.exe	System32\crsrss.exe	smss.exe but it exists	2 or more	Local System	Seconds of boot for first 2
services.exe	Sytem32\services	wininit.exe	1	Local System	Seconds of boot
svchost.exe	System32\svchost.exe	services.exe	5 or more	Typically Local System, Network Service, or Local Service Accounts	Typically seconds of boot but could start after boot
lsm.exe	System32\lsm.exe	wininit.exe	1	Local System	Seconds of boot
explorer.exe	%SystemRoot%\explorer.exe	Userinit.exe but then exists	1 per interactive logged-on user	< Logged on user >	During interactive logon begins
Iexeplore.exe	\Program Files\Internet Explorer\iexplore.exe [or \Program Files (x86)\Internet Explorer\iexplore.exe	explorer.exe	0 to many	< Logged on user >	When user starts Internet Explorer

Image Path: \Program Files\Internet Explorer\iexplore.exe [or \Program Files (x86)\Internet Explorer\iexplore.exe] Parent Process: explorer.exe Number of Instances: 0 to many User Account: Start Time: Typically when user starts Internet Exploer. However, it can be started without explicit user interaction via the "-embedding" switch (in which case, parent may not be explorer.exe). Description: Internet Explorer (IE) is a typical desktop application launched by a user. Such applications will almost always be a child of explorer.exe. Modern versions of IE will have a sub-process for each open tab. It does this for several reasons, including enhanced security. When accessing an Internet site, IE will run the tab process with low integrity, which sandboxes the process, making it more difficult for attackers to modify sensitive areas of the registry or file system if they are able to compromise the IE child process. Attackers often name their malware iexplore.exe and place it in an alternate directory or misspell iexplore.exe as iexplorer.exe.

Disk Encryption Check

Encrypted Disk Detector (EDD)

Using Encrypted Disk Detector (EDD), we can check a system for disk encryption. The program checks for TrueCrypt, PGP, Bitlocker, Safeboot, BestCrypt, Checkpoint, Sophos, or Symantec encrypted volumes.

1) Double-click edd.exe.
2) Take note of the results it renders.

Image Creation

Custom Content (Initial Triage)

Before doing a full disk image, we should make an image of some things we can analyze while the full disk image is taking place.

1) Open FTK Imager.
2) Click File > Add Evidence Item.
3) Select "Physical Drive".
4) Click Next.
5) Click Finish.
6) In the Evidence Tree on the left, expand the newly attached drive.
7) Once you have an item you want to add to a custom image, right-click it and select "Add to Custom Content Image (AD1)". The item will then show up in the bottom left corner in the "Custom Content Sources" window.

   Note: A list of recommended data to add to a custom image is depicted in the Custom Image for Initial Triage section of this document.

   Note: To add all instances of something, do the following:

   i. Click New in the "Custom Content Sources" window.
   ii. An asterisk will appear in the window, highlight it and click Edit.
   iii. Input what you are looking for, it could be the whole name or partial names using wildcards (ex: *.evtx will look for all records ending in .evtx).
   iv. Select all three options available.
   v. Click Ok.
8) When all items are added, click "Create Image" in the "Custom Content Sources" window.
9) Click "Add".
10) Fill in the fields as needed. Examples are below.

    Case Number: YYYYMMDD-####
    Evidence Number: YYYYMMDD-####-####
    Unique Description: Triage Image
    Examiner: Fernando Tomlinson
    Notes: Triage image of <computer name>
11) Click Next.
12) Click Browse to choose the destination folder.
13) Click Ok.

14) Input an image filename and click Finish.
15) Click Start.
16) Click Close when the results screen appears.
17) Verify the image was created and that the text file with it has the hash and other pertinent information listed.

Initial Triage Recommendations

Below is a listing of recommended data to retrieve as part of an initial triage image.

- Hiberfile.sys - [root]\hiberfile.sys - A compressed image of RAM from the last time the system was placed in hibernation mode.
- Pagefile.sys - [root]\pagefile.sys - An extension of RAM (overflow).
- $MFT - [root]\$MFT – An index of every file and folder on the system.
- $Logfile - [root]\$logfile – A recording of file activity (file open, close, creation, and deletion).
- $J - [root]\$Extend\$UsnJrnl\$J – A recording of file activity (file open, close, creation, and deletion).
- $Recycle.Bin - [root]\Recycle.Bin – Location of all deleted items on the disk.
- SAM hive - [root]\Windows\System32\config\SAM – Contains last login, last failed, logon count, password policy, account creation time, and group information.
- SYSTEM hive - [root]\Windows\System32\config\SYSTEM - Contains hardware and service configuration. It lists majority of the raw device names for volumes and drives on the system including USB keys.
- SECURITY hive - [root]\Windows\System32\config\SECURITY - Contains security information that is utilized by the SAM and the operating system including policies, membership of group information, and more.
- SOFTWARE hive - [root]\Windows\System32\config\SOFTWARE - Holds the settings for applications to include Windows products and programs.
- AppData - [root]\Users\< user's name>\AppData - Contains browser history, recent files, and Jumplists.
- Prefetch - [root]\windows\prefetch – A count of how many times a program has executed and the timestamp of the last time.
- RecentFileCache - [root]\windows\Appcompat\Programs -
- Tasks - [root]\Windows\Tasks – The log location where a job executes.
- Windows Search Database - [root]\programdata\microsoft\search\data\applications\windows – Cataloged searches from a user.
- NTUSER.DAT - Contains the preferences and settings of each user.
    - ➢ XP - [root]\documents and settings\< user >\NTUSER.dat
    - ➢ Win7-Win10 - [root]\users\< user >\NTUSER.dat
- USRCLASS.DAT - Contains key information about additional program execution information and depicts which folders a user has opened or closed.
    - ➢ Win7-Win10 - [root]\users\< user >\appdata\local\microsoft\windows\USRCLASS.dat
- Setupapi.dev.log – This is the plug and play log file.
    - ➢ XP - C:\Windows\setupapi.log
    - ➢ Win7/8/10 - C:\Windows\inf\setupapi.dev.log

- .lnk - User-created shortcuts for files or programs.
  - XP - C:\%USERPROFILE%\Recent
  - Win7/8/10 - C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\
  - Win7/8/10 - C:\%USERPROFILE%\AppData\Roaming\Microsoft\Office\Recent\
- .evtx - Windows Event Logs - %systemroot%\System32\winevt\logs

Logical Drive

If a disk is encrypted, it I recommended to get a logical disk image as opposed to a full disk image. The reason being is that we would not be able to access the contents due to the encryption. By doing the logical drive, it is no longer an issue or concern.

1) Open FTK Imager.
2) Click File > Create Disk Image.
3) Select “Logical Drive”.
4) Select from the available drives.
5) Click Finish.
6) Click Add
7) Select a suitable format. E01 is recommended.
8) Click Next.
9) Fill in the fields as needed. Examples are below.

   Case Number: YYYYMMDD-####
   Evidence Number: YYYYMMDD-####-####
   Unique Description: Logical Drive Image
   Examiner: Fernando Tomlinson
   Notes: Image of <drive> from <computer name>
10) Click Next.
11) Click Browse to choose the destination folder.
12) Click Ok.
13) Input an image filename and click Finish.
14) Click Start.
15) Click Close when the results screen appears.
16) Verify the image was created and that the text file with it has the hash and pertinent other information listed.

Full Disk

Before doing a full disk image, we should make an image of some things we can analyze while the full disk image is taking place.

1) Open FTK Imager.
2) Click File > Create Disk Image.
3) Select “Physical Drive”.
4) Click Next.
5) Select the correct drive (if more than one is present) and click Finish.
6) Click Add
7) Select a suitable format. E01 is recommended.
8) Click Next.
9) Fill in the fields as needed. Examples are below.

Case Number: YYYYMMDD-####
Evidence Number: YYYYMMDD-####-####
Unique Description: Logical Drive Image
Examiner: Fernando Tomlinson
Notes: Image of <drive> from <computer name>

10) Click Next.
11) Click Browse to choose the destination folder.
12) Click Ok.
13) Input an image filename and click Finish.
14) Click Start.
15) Click Close when the results screen appears.
16) Verify the image was created and that the text file with it has the hash and pertinent other information listed.

Mount an Image

Having an image has several benefits including the following:

- Bring able to interact with the files and their native or associated application.
- Run anti-virus and malware detection applications.
- Share with remote computers.
- Copy files out of image.
- Forensically sound.

In FTK Imager (Windows)

1) Open FTK Imager.
2) Click File > Add Evidence Item.
3) Select Image File.
4) Click Next.
5) Click Browse and navigate to the image.
6) Click Open.
7) Click Finish.
8) The image is now shown in the Evidence Tree in the top left window.

As a Drive (Windows)

1) Open FTK Imager.
2) Click File > Image Mounting.
3) Click on "…" in the Image File field.
4) Navigate to the location of the image file.
5) Click Open.
6) Change the Mount Method to "File System / Read-Only".
7) Click Mount.
8) Once complete, the image is mounted as a drive and is accessible through Windows Explorer.

Ewfmount (Linux)

1) Log in as Root and open a Terminal window.
2) Create the following directories: '/mnt/ewf_mount' and '/mnt/windows_mount'.
3) Type ewfmount <path_to_.E01> /mnt/ewf_mount.
4) Change directory to /mnt/ewf_mount.
5) Type 'ls' and take note of the name of the file shown.
6) Type 'mount -o r,loop,show_sys_files,streams_interface=windows <name_of_file_from_step_5> /mnt/windows_mount'

Windows Artifacts

Everything that is done on a system leaves some form of an artifact. In the following section, we highlight those artifacts in order to aid you in your task. We have broken down the type of artifacts that are resident by categories.

File Download

Open/Save MRU

Description:
In the simplest terms, this key tracks files that have been opened or saved within a Windows shell dialog box. This happens to be a big data set, not only including web browsers like Internet Explorer and Firefox, but also a majority of commonly used applications.

Having this artifact allows us to confirm that a specific user opened a specific file. Execution history is depicted in an overall list but each file type is also grouped together in a key. With that said, we can not only depict the time the last file that was opened but also the last time the most recent file of each file type was opened.

Tool:
RegRipper, YARU, Registry Viewer, Registry Explorer

Location:

- XP

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU

- Win7/8/10

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePIDlMRU

Interpretation:
The "*" key – This subkey tracks the most recent files of any extension input in an OpenSave dialog

.??? (Three letter extension) – This subkey stores the file info from the OpenSave dialog by specific extension

Downloads

Description:
Firefox and IE have a built-in download manager application which keeps a history of every file downloaded by the user. This browser artifact can provide excellent information about what sites a user has been visiting and what kinds of files they have been downloading from them.

Having this artifact allows us to confirm that a specific user visited a specific URL.

Tool:
SQLite Manager

Location:

- Firefox:
  - XP

%userprofile%\Application Data\Mozilla\ Firefox\Profiles\<random text>.default \downloads.sqlite

- Win7/8/10

%userprofile%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\downloads.sqlite

- Internet Explorer:
  - IE8-9

%USERPROFILE%\AppData\Roaming\Microsoft\Windows\IEDownloadHistory\

- IE10-11

%USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat

Interpretation:
Downloads will include:

- Filename, Size, and Type
- Download from and Referring Page
- File Save Location
- Application Used to Open File
- Download Start and End Times

Program Execution

UserAssist

Description:
GUI-based programs launched from the desktop are tracked in the launcher on a Windows System.

Tool:
Registry Viewer, RegRipper, Registry Explorer

Location:
NTUSER.DAT\Software\Microsoft\Windows\Currentversion\Explorer\UserAssist\{GUID}\Cou

nt

Interpretation:

All values are **ROT-13 Encoded** (Registry Viewer will automatically decode them)

- GUID for XP
  - 75048700 -- Active Desktop
- GUID for Win7/8/10
  - CEBFF5CD -- Executable File Execution
  - F4E57C4B -- Shortcut File Execution

- Program Locations for Win7 UserAssist

ProgramFilesX64	6D809377-…
ProgramFilesX86	7C5A40EF-…
System	1AC14E77-…
SystemX86	D65231B0-…
Desktop	B4BFCC3A-…
Documents	FDD39AD0-…
Downloads	374DE290-…
UserProfiles	0762D272-…

Last-Visited MRU

Description:

Tracks the specific executable used by an application to open the files documented in the OpenSaveMRU key. In addition, each value also tracks the directory location for the last file that was accessed by that application.

Tool:

Registry Viewer, RegRipper, YARU (to interpret hex from the key)

Location:

- XP

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU

- Win7/8/10

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU

Interpretation:

Tracks the application executables used to open files in OpenSaveMRU and the last file path used.

RunMRU Start->Run

Description:

Whenever someone does a Start -> Run command, it will log the entry for the command they executed.

Tool:
Registry Viewer, RegRipper, YARU, Registry Explorer

Location:
NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU

Interpretation:
The order in which the commands are executed is listed in the RunMRU list value. The letters represent the order in which the commands were executed.

AppCompatCache

Description:
• Windows Application Compatibility Database is used by Windows to identify possible application compatibility challenges with executables.
• Tracks the executables file name, file size, last modified time, and in Windows XP the last update time.

Tool:
Registry Viewer, RegRipper, YARU, Registry Explorer

Location:

- XP

SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatibility

- Win7/8/10

SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache

Interpretation:
Any executable run on the Windows system could be found in this key. You can use this key to identify systems that specific malware was executed on. In addition, based on the interpretation of the time-based data you might be able to determine the last time of execution or activity on the system.

- Windows XP contains at most 96 entries

➢ LastUpdateTime is updated when the files are executed

- Windows 7 contains at most 1024 entries

➢ LastUpdateTime does not exist on Win7 systems

Jump Lists

Description:
The Windows 7 task bar (Jump List) is engineered to allow users to “jump” or access items they have frequently or recently used quickly and easily. This functionality cannot only include recent media files; it must also include recent tasks.

The data stored in the AutomaticDestinations folder will each have a unique file prepended with the AppID of the associated application.

NOTE: The AutomaticDestinations folder will not be shown if navigating from folder to folder. In order to see it, you must type in the "AutomaticDestinations" within the Explorer window once you get to the Recent folder.

Tool:
Windows Jump List Parser, Jump List View (Can be ran remotely as well on a live system)

Location:

- Win7/8/10

C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

Interpretation:

- First time of execution of application.
  - ➢ Creation Time = First time item added to the AppID file.
- Last time of execution of application w/file open.
  - ➢ Modification Time = Last time item added to the AppID file.
- List of Jump List IDs
  - ➢ http://www.forensicswiki.org/wiki/List_of_Jump_List_IDs

Prefetch

Description:
• Increases performance of a system by pre-loading code pages of commonly used applications.
• Cache Manager monitors all files and directories referenced for each application or process and map them into a .pf file. Utilized to know an application was executed on a system.
• Limited to 128 files on XP and Win7
• Limited to 1024 files on Win8
• (exe_name)-(hash).pf

Tool:
Win Prefetch View (Can be ran remotely as well on a live system)

Location:

- WinXP/7/8/10

C:\Windows\Prefetch

Interpretation:
Each .pf will include last time of execution, number of times run, and device and file handles used by the program

• Date/Time file by that name and path was first executed

- ➢ Creation Date of .pf file (-10 seconds)

• Date/Time file by that name and path was last executed

- ➢ Embedded last execution time of .pf file

- Last modification date of .pf file (-10 seconds)
- Win8+ will contain last 8 times of execution

• Disabled by default on server OS.
HKLM\system\currentcontrolset\control\session manager\memory
0 = Disabled
1 = Application launch prefetching enabled
2 = Boot prefetching enabled
3 = Application launch and boot enabled

Amacache.hve/RecentFileCache.bcf

Description:
ProgramDataUpdater (a task associated with the Application Experience Service) uses the registry file RecentFilecache.bcf to store data during process creation

Tool:
Registry Viewer, RegRipper, YARU, Registry Explorer

Location:

- Win7/8/10

C:\Windows\AppCompat\Programs\Amcache.hve
(Windows 7/8/8.1)

- Win7

C:\Windows\AppCompat\Programs\RecentFilecache.bcf

Interpretation:
• RecentFileCache.bcf – Executable PATH and FILENAME and the program is probably new to the system
• The program executed on the system since the last ProgramDataUpdated task has been run
• Amcache.hve – Keys = Amcache.hve\Root\File\{Volume GUID}\#######
• Entry for every executable run, full path information, File's $StandardInfo Last Modification Time, and Disk volume the executable was run from
• First Run Time = Last Modification Time of Key
• SHA1 hash of executable also contained in the key

File/Folder Opening

Open/Save MRU

Description:
In the simplest terms, this key tracks files that have been opened or saved within a Windows shell dialog box. This happens to be a big data set, not only including web browsers like Internet Explorer and Firefox, but also a majority of commonly used applications.

Tool:
RegRipper, Registry Viewer, YARU (to interpret hex from the key)

Location:

- XP

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU

- Win7/8/10

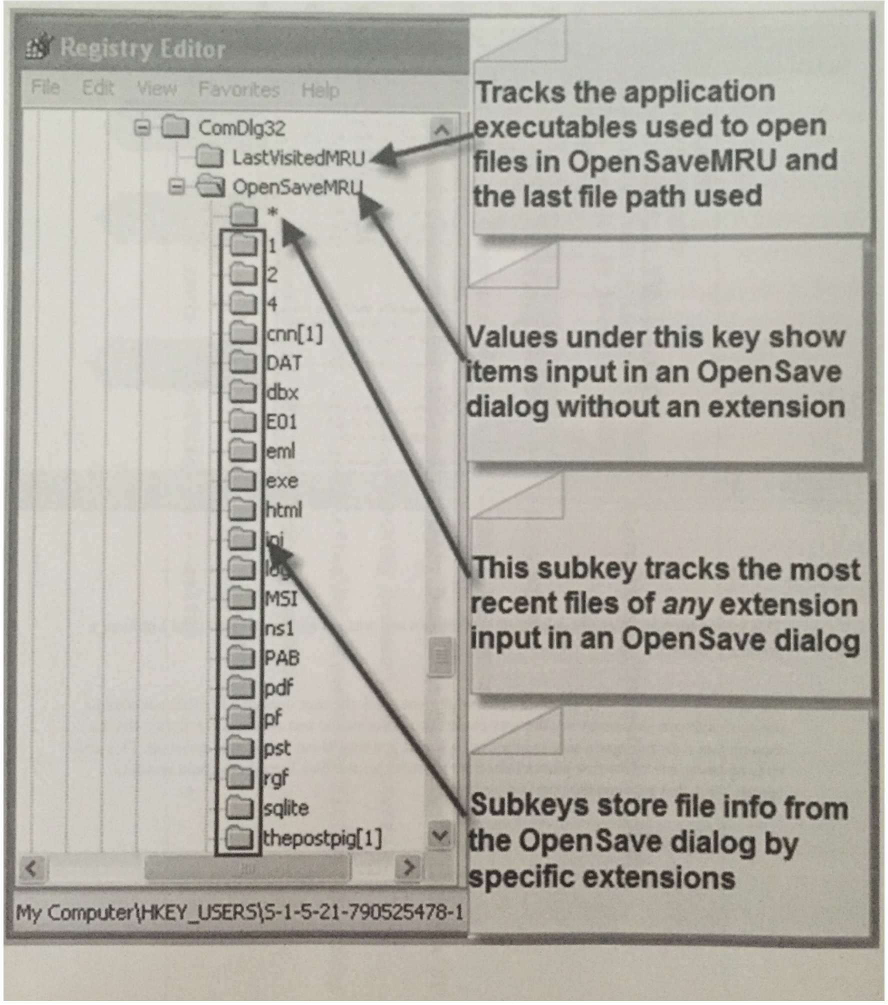
NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePIDlMRU

Interpretation:

The "*" key – This subkey tracks the most recent files of any extension input in an OpenSave dialog

.??? (Three letter extension) – This subkey stores the file info from the OpenSave dialog by specific extension

Tracks the application executables used to open files in OpenSaveMRU and the last file path used.



Tracks the application executables used to open files in OpenSaveMRU and the last file path used

Values under this key show items input in an OpenSave dialog without an extension

This subkey tracks the most recent files of *any* extension input in an OpenSave dialog

Subkeys store file info from the OpenSave dialog by specific extensions

Last-Visited MRU

Description:
Tracks the specific executable used by an application to open the files documented in the OpenSaveMRU key. In addition, each value also tracks the directory location for the last file that was accessed by that application.

Tool:
Registry Viewer, RegRipper, YARU (to interpret hex from the key)

Location:

- XP

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU

- Win7/8/10

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU

Interpretation:
Tracks the application executables used to open files in OpenSaveMRU and the last file path used.

Recent Files

Description:
Registry Key that will track the last files and folders opened and is used to populate data in "Recent" menus of the Start menu. Under the RecentDocs Key, there can be up to 150 keys with each being able to have up 10 Values. This applies to WinXP through Win10.

Tool:
Registry Viewer, RegRipper

Location:
NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

Interpretation:
• RecentDocs – Overall key will track the overall order of the last 150 files or folders opened. MRU list will keep track of the temporal order in which each file/ folder was opened. The last entry and modification time of this key will be the time and location the last file of a specific extension was opened.

• .??? – This subkey stores the last files with a specific extension that were opened. MRU list will keep track of the temporal order in which each file was opened. The last entry and modification time of this key will be the time and location of the last file of a specific extension was opened.

- Folder – This subkey stores the last folders that were opened. MRU list will keep track of the temporal order in which each folder was opened. The last entry and modification time of this key will be the time and location of the last folder opened.

Office Recent Files

Description:

MS Office programs will track their own Recent Files list to make it easier for users to remember the last file they were editing.

Tool:

Registry Viewer, DCode

Location:

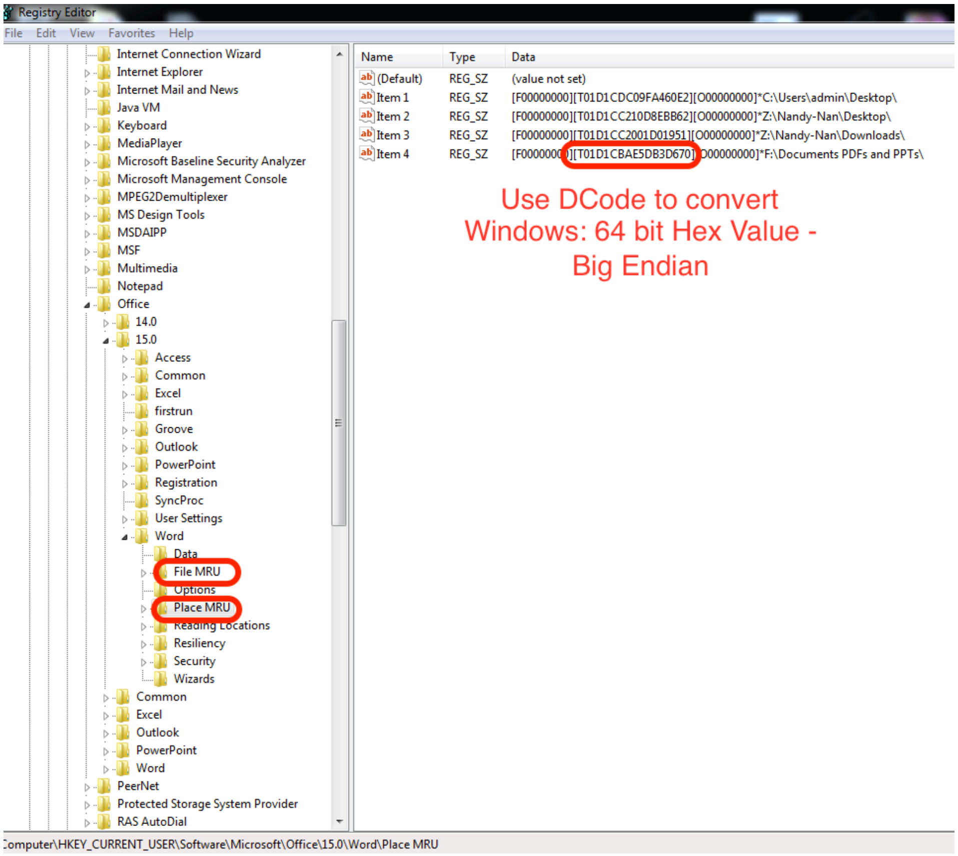
Time file last opened:

NTUSER.DAT\Software\Microsoft\Office\VERSION

- 14.0 = Office 2010
- 12.0 = Office 2007
- 11.0 = Office 2003
- 10.0 = Office XP

NTUSER.DAT\Software\Microsoft\Office\VERSION\UserMRU\LiveID_####\FileMRU

- 15.0 = Office 365/ 2013

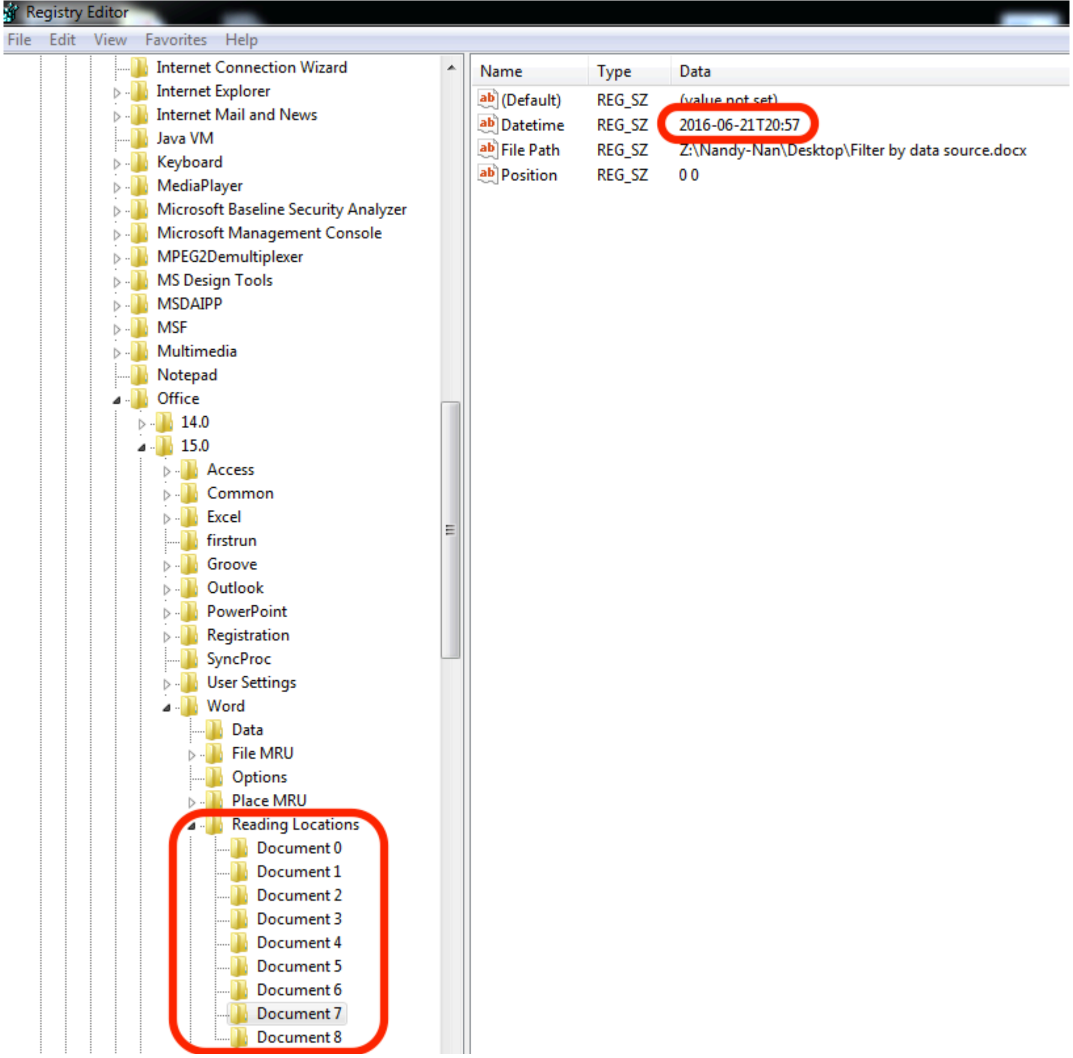


Name	Type	Data
(Default)	REG_SZ	(value not set)
Item 1	REG_SZ	[F00000000][T01D1CDC09FA460E2][O00000000]*C:\Users\admin\Desktop\
Item 2	REG_SZ	[F00000000][T01D1CC210D8EBB62][O00000000]*Z:\Nandy-Nan\Desktop\
Item 3	REG_SZ	[F00000000][T01D1CC2001D01951][O00000000]*Z:\Nandy-Nan\Downloads\
Item 4	REG_SZ	[F00000000][T01D1CBAE5DB3D670][O00000000]*F:\Documents PDFs and PPTs\

Use DCode to convert Windows: 64 bit Hex Value - Big Endian

Computer\HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Word\Place MRU

Time a file last closed:
NTUSER.DAT\Software\Microsoft\Office\VERSION\15.0 \<product>\Reading Locations\



Name	Type	Data
(Default)	REG_SZ	(value not set)
Datetime	REG_SZ	2016-06-21T20:57
File Path	REG_SZ	Z:\Nandy-Nan\Desktop\Filter by data source.docx
Position	REG_SZ	0 0

Interpretation:
Similar to the Recent Files, this will track the last files that were opened by each MS Office application. The last entry added, per the MRU, will be the time the last file was opened by a specific MS Office application.

Shell Bags

Description:
• Which folders were accessed on the local machine, the network, and/or removable devices. Evidence of previously existing folders after deletion/overwrite. When certain folders were accessed.

Tool:
Shellbags Explorer, Windows ShellBag Parser

Location:

- Explorer Access

USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\Shell\Bags
USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\Shell\BagMRU

- Desktop Access

NTUSER.DAT\Software\Microsoft\Windows\Shell\BagMRU
NTUSER.DAT\Software\Microsoft\Windows\Shell\Bags

Interpretation:
Stores information about which folders were most recently browsed by the user.

Shortcut (LNK) Files

Description:

- Shortcut Files automatically created by Windows
  - Recent Items
  - Opening local and remote data files and documents will generate a shortcut file (.lnk)
  - Any non-executable opened by Windows
- Max is 149 entries

Tool:
FTK Imager, Windows LNK Parsing Utility

Location:

- XP

➢ C:\%USERPROFILE%\Recent

- Win7/8/10

➢ C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\
➢ C:\%USERPROFILE%\AppData\Roaming\Microsoft\Office\Recent\

Note: these are primary locations of LNK files. They can also be found in other locations.

Interpretation:

- Date/Time file of that name was first opened

➢ Creation Date of Shortcut (LNK) File

- Date/Time file of that name was last opened

➢ Last Modification Date of Shortcut (LNK) File

- LNKTarget File (Internal LNK File Information) Data:

➢ Modified, Access, and Creation times of the target file

- Volume Information (Name, Type, Serial Number)
- Network Share information
- Original Location
- Name of System

Jump Lists

Description:

• The Windows 7 task bar (Jump List) is engineered to allow users to “jump” or access items have frequently or recently used quickly and easily. This functionality cannot only include recent media files; it must also include recent tasks.

• The data stored in the AutomaticDestinations folder will each have a unique file prepended with the AppID of the association application and embedded with LNK files in each stream.

Tool:

Windows Jump List Parser

Location:

- Win7/8/10

C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

Interpretation:

• Using the Structured Storage Viewer, open up one of the AutomaticDestination jumplist files.

• Each one of these files is a separate LNK file. They are also stored numerically in order from the earliest one (usually 1) to the most recent (largest integer value).

Prefetch

Description:

• Increases performance of a system by pre-loading code pages of commonly used applications. Cache Manager monitors all files and directories referenced for each application or process and maps them into a .pf file. Utilized to know an application was executed on a system.

• Limited to 128 files on XP and Win7

• Limited to 1024 files on Win8

• (exename)-(hash).pf

Tool:

Win Prefetch View

Location:

- WinXP/7/8/10
- C:\Windows\Prefetch

Interpretation:
Each .pf will include last time of execution, number of times run, and device and file handles used by the program

• Date/Time file by that name and path was first executed

- Creation Date of .pf file (-10 seconds)

• Date/Time file by that name and path was last executed

- Embedded last execution time of .pf file
- Last modification date of .pf file (-10 seconds)
- Win8+ will contain last 8 times of execution

• Disabled by default on server OS.
HKLM\system\currentcontrolset\control\session manager\memory
0 = Disabled
1 = Application launch prefetching enabled
2 = Boot prefetching enabled
3 = Application launch and boot enabled

Index.dat file://

Description:
• A little known fact about the IE History is that the information stored in the history files is not just related to Internet browsing. The history also records, local, removable, and remote (via network shares) file access giving us an excellent means for determining which files and applications were accessed on the system, day by day.

Tool:

Location:

- Internet Explorer:
  - IE6-7

%USERPROFILE%\Local Settings\History\ History.IE5

- IE8-9

%USERPROFILE%\AppData\Local\Microsoft\Windows\History\History.IE5

- IE10-11

%USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat

Interpretation:
• Stored in index.dat as: file:///C:/directory/filename.ext
• Does not mean file was opened in browser

File or File Knowledge

XP Search – ACMRU

Description:
You can search for a wide range of information through the search assistant on a Windows XP machine. The search assistant will remember a user's search terms for filenames, computers, or words that are inside a file. This is an example of where you can find the "Search History" on the Windows system.

Tool:
Registry Explorer, Registry Viewer, YARU

Location:
NTUSER.DAT HIVE
NTUSER.DAT\Software\Microsoft\SearchAssistant\ACMru\####

Interpretation:
- Search the Internet – ####=5001
- All or part of a document name – ####=5603
- A word or phrase in a file – ####=5604
- Printers, Computers and People – ####=5647

Search – WordWheelQuery

Description:
Keywords searched for from the START menu or Explorer search bar.

Tool:
RegRipper, Registry Viewer

Location:
- Win7/8/10

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery

Interpretation:
Keywords are added in Unicode and listed in temporal order in an MRUlist

Last-Visited MRU

Description:
Tracks the specific executable used by an application to open the files documented in the OpenSaveMRU key. In addition, each value also tracks the directory location for the last file that was accessed by that application.

Tool:
Registry Viewer, RegRipper, YARU (to interpret hex from the key)

Location:

- XP

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU

- Win7/8/10

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU

Interpretation:
Tracks the application executables used to open files in OpenSaveMRU and the last file path used.

OpenSave MRU

Description:
Tracks files that have been opened or saved within the Windows shell dialog box. It also tracks auto-complete terms for the same dialog box.

Tool:
RegRipper, Registry Viewer, YARU (to interpret hex from the key)

Location:

- XP

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU

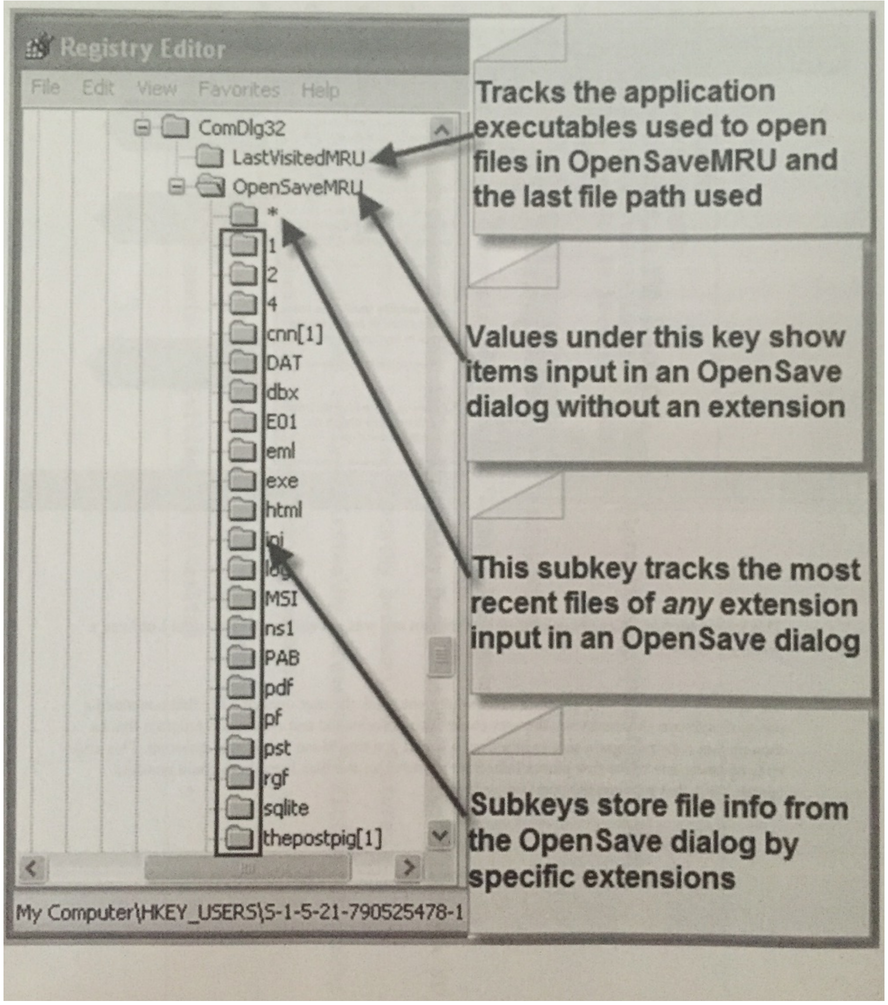
- Win7/8/10

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePIDlMRU

Interpretation:
The “*” key – This subkey tracks the most recent files of any extension input in an OpenSave dialog

.??? (Three letter extension) – This subkey stores the file info from the OpenSave dialog by specific extension

Tracks the application executables used to open files in OpenSaveMRU and the last file path used.



Tracks the application executables used to open files in OpenSaveMRU and the last file path used

Values under this key show items input in an OpenSave dialog without an extension

This subkey tracks the most recent files of *any* extension input in an OpenSave dialog

Subkeys store file info from the OpenSave dialog by specific extensions

Thumbs.db

Description:
Hidden file in directory where pictures on Windows XP machine exist. Catalogs all the pictures and stores a copy of the thumbnail even if the pictures were deleted.

Tool:
Thumbs.db Viewer

Location:
Each directory where pictures resided that were viewed in thumbnail mode. Many cameras also will auto-generate a thumbs.db file when you view the pictures on the camera itself.

Interpretation:
- Thumbnail Picture of Original
- Last Modification Time
- Original Filename

Thumbscache

Description:
On Vista/Win7 versions of Windows, thumbs.db does not exist. The data now sit under a single directory for each user of the machine located in their application data directory under their home directory.

Tool:
Thumbcache Viewer

Location:
C:\%USERPROFILE%\AppData\Local\Microsoft\Windows\Explorer

Interpretation:
• These are created when a user switches a folder to thumbnail mode or views pictures via a slide show. As it were, our thumbs are now stored in separate database files. Vista/Win7 has 4 sizes for thumbnails and the files in the cache folder reflect this:

32 -> small
96 -> medium
256 -> large
1024 -> extra large

• The thumbscache will store the thumbnail copy of the picture based on the thumbnail size in the content of the equivalent database file.

XP Recycle Bin

Description:
The recycle bin is a very important location on a Windows file system to understand. It can help you when accomplishing a forensic investigation, as every file that is deleted from a Windows recycle bin aware program is generally first put in the recycle bin.

Tool:
Recbin

Location:
Hidden System Folder

- Windows XP

➢ C:\RECYCLER" 2000/NT/XP/2003
➢ Subfolder is created with user's SID
➢ Hidden file in directory called "INFO2"
➢ INFO2 Contains Deleted Time and Original Filename
➢ Filename in both ASCII and UNICODE

Interpretation:
• SID can be mapped to user via Registry Analysis
• Maps file name to the actual name and path it was deleted from

Win7/8/10 Recycle Bin

Description:
The recycle bin is a very important location on a Windows file system to understand. It can help you when accomplishing a forensic investigation, as every file that is deleted from a Windows recycle bin aware program is generally first put in the recycle bin.

Tool:
Recbin

Location:
Hidden System Folder

- Win7/8/10
  - C:\$Recycle.bin
  - Deleted Time and Original Filename contained in separate files for each deleted recovery file

Interpretation:

- SID can be mapped to user via Registry Analysis
- Win7/8/10
  - Files Preceded by $I###### files contain
- Original PATH and name
- Deletion Date/Time
  - Files Preceded by $R###### files contain
- Recovery Data

Index.dat file://

Description:
• A little known fact about the IE History is that the information stored in the history files is not just related to Internet browsing. The history also records, local, removable, and remote (via network shares) file access giving us an excellent means for determining which files and applications were accessed on the system, day by day.

Tool:

Location:

- Internet Explorer:
  - IE6-7

%USERPROFILE%\Local Settings\History\ History.IE5

- IE8-9

%USERPROFILE%\AppData\Local\Microsoft\Windows\History\History.IE5

- IE10-11

%USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat

Interpretation:

- Stored in index.dat as: file:///C:/directory/filename.ext
- Does not mean file was opened in browser

Typed Paths

Description:
This key will show when you have manually typed a path into the Start menu or into the Explorer bar. This key would be useful in a situation where you are trying to show that the user had specific knowledge of a location. Showing that the user actually had to type of cut-and-paste the location into Explorer would show direct knowledge of the location. The key auto-sorts by most recent (with the #1 spot being the most recent).

Tool:
Registry Viewer, Registry Explorer, YARU

Location:
NTUSER.dat\Software\Microsoft\windows\currentversion\explorer\typedpaths

Physical Location

Time Zone

Description:
Identifies the current system time zone.

Tool:
Registry Viewer, Registry Explorer, YARU

Location:
SYSTEM Hive
SYSTEM\CurrentControlSet\Control\TimeZoneInformation

Interpretation:
- Time activity is incredibly useful for correlation of activity
- Internal log files and date/timestamps will be based on the system time zone information
- You might have other network devices and you will need to correlate information to the time zone information collected here.

Network Interface

Description:
- Lists the network interfaces of the machine
- Can determine whether a machine has a static IP or DHCP
- Could tie a machine to network activity that was logged
- Obtain interface GUID for additional profiling in network connections

Tool:
Registry Viewer, Registry Explorer, YARU

Location:
Key: System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces

Network History

Description:

- Identify networks that the computer has been connected to
- Network connection type could be wireless, wired, or broadband (3g)
- Identify domain name/intranet name
- Identify SSID
- Identify Gateway MAC Address

Tool:

Registry Viewer, Registry Explorer, DCode

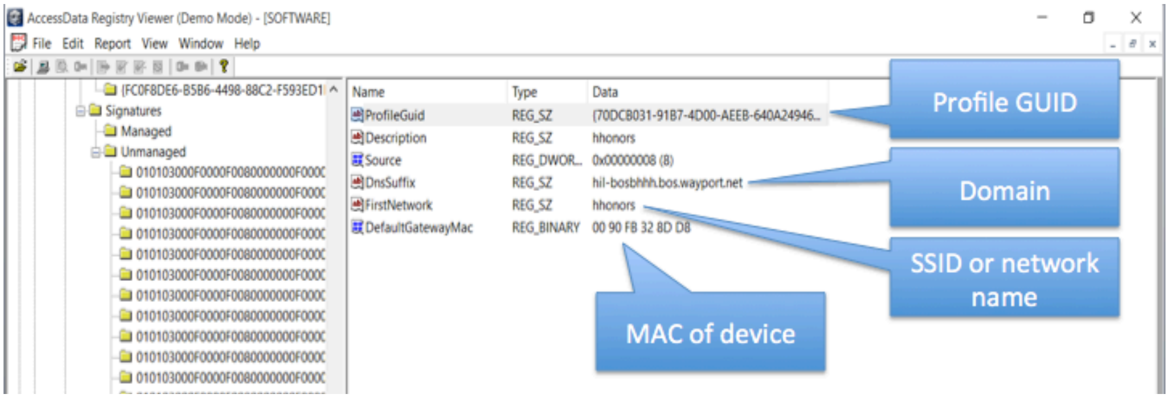
Location:

- Win7/8/10 SOFTWARE HIVE

➢ The Keys listed below will provide the information the following information:
- ○ Profile GUID
- ○ Domain
- ○ SSID or network name
- ○ MAC of device

SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\ Unmanaged

SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\ Managed



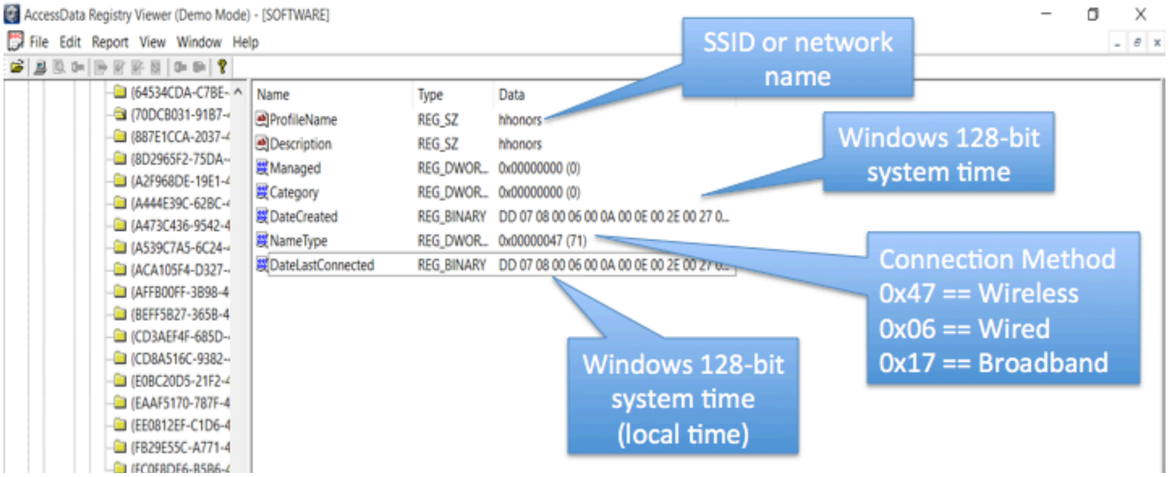
AccessData Registry Viewer (Demo Mode) - [SOFTWARE]

Name	Type	Data
ProfileGuid	REG_SZ	{70DCB031-91B7-4D00-AEEB-640A24946...
Description	REG_SZ	hhonors
Source	REG_DWOR...	0x00000008 (8)
DnsSuffix	REG_SZ	hil-bosbhhh.bos.wayport.net
FirstNetwork	REG_SZ	hhonors
DefaultGatewayMac	REG_BINARY	00 90 FB 32 8D D8

File Edit Report View Window Help
{FC0F8DE6-B5B6-4498-88C2-F593ED1
Signatures
Managed
Unmanaged
010103000F0000F0080000000F000C
Profile GUID
Domain
SSID or network name
MAC of device

➢ Using a GUID from above, match it with a GUID listed below. This Key will provide the following information:
- ○ SSID or network name
- ○ Date the connection was created (use DCode to make it human-readable)
- ○ Connection method (wired, wireless, or broadband (3G))
- ○ Date the connection was last made (use DCode to make it human-readable)

SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles\<GUID>



AccessData Registry Viewer (Demo Mode) - [SOFTWARE]
File Edit Report View Window Help
SSID or network name
Windows 128-bit system time
Connection Method
0x47 == Wireless
0x06 == Wired
0x17 == Broadband
Windows 128-bit system time (local time)

{64534CDA-C7BE-
{70DCB031-91B7-
{887E1CCA-2037-
{8D2965F2-75DA-
{A2F968DE-19E1-
{A444E39C-62BC-
{A473C436-9542-
{A539C7A5-6C24-
{ACA105F4-D327-
{AFFB00FF-3B98-
{BEFF5B27-365B-
{CD3AEF4F-685D-
{CD8A516C-9382-
{E0BC20D5-21F2-
{EAAF5170-787F-
{EE0812EF-C1D6-
{FB29E55C-A771-
{FC0F8DF6-B5B6-

Name	Type	Data
ProfileName	REG_SZ	hhonors
Description	REG_SZ	hhonors
Managed	REG_DWOR...	0x00000000 (0)
Category	REG_DWOR...	0x00000000 (0)
DateCreated	REG_BINARY	DD 07 08 00 06 00 0A 00 0E 00 2E 00 27 0...
NameType	REG_DWOR...	0x00000047 (71)
DateLastConnected	REG_BINARY	DD 07 08 00 06 00 0A 00 0E 00 2E 00 27 0...

- ➢ Network Address Awareness (NLA) is used to aid in identifying where a computer might be connected to and adjusts the firewall accordingly. A network profile (GUID) is created for every network a system is connected to. A list of all networks the system has ever connected to is listed at the below Key.

SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Nla\Cache

Interpretation:

• Identifying intranets and networks that a computer has connected to is incredibly important
• Not only can you determine the intranet name, you can determine the last time the network was connected to base on the last write time of the key
• This will also list any networks that have been connected to via a VPN
• MAC Address of SSID for Gateway could be physically triangulated

Cookies

Description:

Cookies give insight into what websites have been visited and what activities may have taken place there.

Tool:

FTK Imager

Location:

- Internet Explorer

➢ IE6-8

%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies

➢ IE10

%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies

➢ IE11

%USERPROFILE%\AppData\Local\Microsoft\Windows\INetCookies

- Firefox

➢ XP

%USERPROFILE%\Application Data\Mozilla\Firefox\Profiles\<random text>.default\cookies.sqlite

- ➢ Win7/8/10

%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext> .default\cookies.sqlite

- Chrome

- ➢ XP

%USERPROFILE%\Local Settings\ApplicationData\Google\Chrome\User Data\Default\Local Storage

- ➢ Win7/8/10

%USERPROFILE%\AppData\Local\Google\Chrome\UserData\Default\Local Storage

Browser Search Terms

Description:

Records websites visited by date and time. Details stored for each local user account. Records number of times visited (frequency). Also tracks access of local system files. This will also include the website history of search terms in search engines.

Tool:

Location:

- Internet Explorer

- ➢ IE6-7

%USERPROFILE%\Local Settings\History\History.IE5

- ➢ IE8-9

%USERPROFILE%\AppData\Local\Microsoft\Windows\History\History.IE5

- ➢ IE10-11

%USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV* .dat

- Firefox

- ➢ XP

%userprofile%\Application Data\Mozilla\Firefox\Profiles\<randomtext> .default\places.sqlite

- ➢ Win7/8/10

%userprofile%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\places.sqlite

External Device/USB Usage

Key Identification

Description:

Track USB devices plugged into a machine.

Tool:

FTK Imager, Registry Viewer, Registry Explorer

Location:

- SYSTEM\CurrentControlSet\Enum\USBSTOR
  - Vendor =
  - Product =
  - Version =
  - USB Unique Serial Number =
- SYSTEM\CurrentControlSet\Enum\USB
  - (VendorID) VID_XXXX =
  - (Product ID) PID_XXXX =

Interpretation:

- Identify vendor, product, and version of a USB device plugged into a machine
- Identify a unique USB device plugged into the machine
- Determine the time a device was plugged into the machine
- Devices that do not have a unique serial number will have an "&" in the second character of the serial number.

First/Last Times

Description:

Determine temporal usage of specific USB devices connected to a Windows Machine.

Tool:

FTK Imager, Registry Viewer, DCode

Location: ***First Time***

Plug and Play Log Files

- XP
  - C:\Windows\setupapi.log
- Win7/8/10
  - C:\Windows\inf\setupapi.dev.log

Location: ***First, Last, and Removal Times (Win7/8/10 Only)***

System\CurrentControlSet\Enum\USBSTOR\Ven_Prod_Version\USBiSerial#\Properties\{83da6326-97a6-4088-9453-a1923f573b29}\####

- 0064 = First Install (Win7/8)
- 0066 = Last Connected (Win8 only)
- 0067 = Last Removal (Win 8 only)

Use DCode to decode the 64 bit hex value timestamp of the above entries

Interpretation:

- Search for Device Serial Number
- Log File times are set to local time zone

User

Description:

Find User that used the Unique USB Device.

Tool:
Registry Viewer, Registry Explorer

Location:
- Look for GUID from SYSTEM\MountedDevices
- NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2
  user =

Interpretation:
This GUID will be used next to identify the user that plugged in the device. The last write time of this key also corresponds to the last time the device was plugged into the machine by that user. The number will be referenced in the user's personal mountpoints key in the NTUSER.DAT Hive.

Volume Serial Number

Description:
Discover the Volume Serial Number of the Filesystem Partition on the USB (NOTE: This is not the USB Unique Serial Number, that is hardcoded into the device firmware.)

Tool:
Registry Viewer, Registry Explorer

Location:
- SOFTWARE\Microsoft\WindowsNT\CurrentVersion\ ENDMgmt
  Volume Serial Number (HEX) =
- Use Volume Name and USB Unique Serial Number to find
- Last integer number in line
- Convert Decimal Serial Number into Hex Serial Number

Interpretation:
- Knowing both the Volume Serial Number and the Volume Name you can correlate the data across SHORTCUT File (LNK) analysis and the RECENTDOCs key.
- The Shortcut File (LNK) contains the Volume Serial Number and Name
- RecentDocs Registry Key, in most cases, will contain the volume name when the USB device is opened via Explorer

Drive Letter & Volume Name

Description:
Discover the last drive letter of the USB Device when it was plugged into the machine.

Tool:
Registry Viewer, Registry Explorer

Location:
- XP

- ➢ Find ParentIdPrefix

SYSTEM\CurrentControlSet\Enum\USBSTOR

- ➢ Using ParentIdPrefix Discover Last Mount Point

SYSTEM\MountedDevices

- Win7/8/10

SOFTWARE\Microsoft\Windows Portable Devices\Devices

Volume name =

Drive Letter (Vista Only) =

SYSTEM\MountedDevices

- ➢ Examine Drive Letter's looking at Value Data Looking for Serial Number

Interpretation:

Identify the USB device that was last mapped to a specific drive letter. This technique will only work for the last drive mapped. It does not contain historical records of every drive letter mapped to a removable drive.

Jump Lists

Description:

• The Windows 7 task bar (Jump List) is engineered to allow users to "jump" or access items have frequently or recently used quickly and easily. This functionality cannot only include recent media files; it must also include recent tasks.

• The data stored in the AutomaticDestinations folder will each have a unique file prepended with the AppID of the association application and embedded with LNK files in each stream.

Tool:

Windows Jump List Parser

Location:

- Win7/8/10

C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

Interpretation:

• Using the Structured Storage Viewer, open up one of the AutomaticDestination jumplist files.

• Each one of these files is a separate LNK file. They are also stored numerically in order from the earliest one (usually 1) to the most recent (largest integer value).

Shortcut (LNK) Files

Description:

• Shortcut Files automatically created by Windows

- Recent Items
- Opening local and remote data files and documents will generate a shortcut file (.lnk)
- Any non-executable opened by Windows

• Max is 149 entries

Tool:
FTK Imager, Windows LNK Parsing Utility

Lp.exe will provide the drive letter and name the file was executed from.

Ex: dir "E:\[root]\users\<some_user>\appdata\roaming\microsoft\windows\recent\*.lnk" /b /s | lp.exe –pipe –csv –no_whitespace –timeformat hh:mm:ss > output_lnk.csv

Location:

- XP
  - ➢ C:\%USERPROFILE%\Recent
- Win7/8/10
  - ➢ C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\
  - ➢ C:\%USERPROFILE%\AppData\Roaming\Microsoft\Office\Recent\

Note: these are primary locations of LNK files. They can also be found in other locations.

Interpretation:

- Date/Time file of that name was first opened
  - ➢ Creation Date of Shortcut (LNK) File
- Date/Time file of that name was last opened
  - ➢ Last Modification Date of Shortcut (LNK) File
- LNKTarget File (Internal LNK File Information) Data:
  - ➢ Modified, Access, and Creation times of the target file
  - ➢ Volume Information (Name, Type, Serial Number)
  - ➢ Network Share information
  - ➢ Original Location
  - ➢ Name of System

PnP Events

Description:
When a Plug and Play driver install is attempted, the service will log an ID 20001 event and provide a Status within the event. It is important to note that this event will trigger for any Plug and Play-capable device, including but not limited to USB, Firewire, and PCMCIA devices.

Tool:
Event Log Explorer

Location: System Log File

- Win7/8/10

%system root%\System32\winevt\logs\System.evtx

Interpretation:
• Event ID: 20001 – Plug and Play driver install attempted
• Event ID 20001
• Timestamp
• Device information

- Device serial number
- Status (0 = no errors)

System Information

OS Version and Install Date

Description:

- Determine the Microsoft Windows version, service pack level, and install data of the machine using this key
- Identify OS version

Tool:

Registry Viewer, Registry Explorer

Location:

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\
InstallDate =
ProductNameValue =

Computer Name

Description: The computer name is mainly for logging purposes and verification, but it should not go unnoticed. It can be used to ensure you are on the system you are supposed to be on.

Tool:

Registry Viewer, RegRipper, Registry Explorer

Location:

SYSTEM\currentontrolset\control\computername\
Computername =

Shutdown Information

Description:

- List the last shutdown time, which could be useful to detect certain types of activity
- The shutdown count also shows whether the user usually shuts down his machine

Tool: DCode, Registry Viewer, Registry Explorer

Location:

- Shutdown Time
  - Note: The time will need to be converted fro 64 bit Hex Little Endian using DCode

SYSTEM\currentcontrolset\control\windows

- Shutdown Count (XP only)

SYSTEM\currentcontrolset\control\watchdog\Display

Network Interface

Description:

- Lists the network interfaces of the machine
- Can determine whether a machine has a static IP or DHCP
- Could tie a machine to network activity that was logged
- Obtain interface GUID for additional profiling in network connections

Tool:

Registry Viewer, Registry Explorer

Location:

Key: System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces

Account Usage

Last Login

Description:

Lists the local accounts of the system and their equivalent security identifiers.

Tool:

SAMInside, Registry Explorer, RegRipper

Location:

- C:\windows\system32\config\SAM
- SAM\Domains\Account\Users

Interpretation:

- Only the last login time will be stored in the registry key

Last Password Change

Description:

Lists the last time the password of a specific user has been changed.

Tool:

SAMInside, Registry Explorer, RegRipper

Location:

- C:\windows\system32\config\SAM
- SAM\Domains\Account\Users

Interpretation:

- Only the last password change time will be stored in the registry key

Success/Fail Logons

Description:

Determine which accounts have been used for attempted logons. Track account usage for known compromised accounts.

Tool:

Event Log Explorer

Location:

- XP

%system root%\System32\config\SecEvent.evt

- Win7/8/10

%system root%\System32\winevt\logs\Security.evtx

Interpretation:

- XP/Win7/8/10 - Interpretation
- Event ID - 528/4624 – Successful Logon
- Event ID - 529/4625 – Failed Logon
- Event ID - 538/4634 – Successful Logoff
- Event ID - 540/4624 – Successful Network Logon (example: file shares)

Logon Types

Description:

Logon Events can give us very specific information regarding the nature of account authorizations on a system if we know where to look and how to decipher the data that we find. In addition to telling us the date, time, username, hostname, and success/failure status of a logon, Logon Events also enables us to determine by exactly what means a logon was attempted.

Tool:

Event Log Explorer

Location:

- XP
  - ➢ Event ID 528
- Win7/8/10
  - ➢ Event ID 4624

Interpretation:

Logon Type Explanation:

- 2 Logon via console
- 3 Network Logon
- 4 Batch Logon
- 5 Windows Service Logon
- 7 Credentials used to unlock screen
- 8 Network logon sending credentials (cleartext)

- 9 Different credentials used than logged on user
- 10 Remote interactive logon (RDP)
- 11 Cached credentials used to logon
- 12 Cached remote interactive (similar to Type 10)
- 13 Cached unlock (similar to Type 7)

RDP Usage

Description:
Track Remote Desktop Protocol logons to target machines.

Tool:
Event Log Explorer

Location: ***Security Log***

- XP

%SYSTEM ROOT%\System32\config\SecEvent.evt

- Win7/8/10

%SYSTEM ROOT%\System32\winevt\logs\Security.evtx

Interpretation:

- XP/Win7/8/10 - Interpretation
  - ➢ Event ID 682/4778 – Session Connected/Reconnected
  - ➢ Event ID 683/4779 – Session Disconnected
- Event log provides hostname and IP address of remote machine making the connection
- On workstations you will often see current console session disconnected (683) followed by RDP connection (682)

Services Events

Description:
• Analyze logs for suspicious services running at boot time
• Review services started or stopped around the time of a suspected compromise

Tool:
Event Log Explorer

Location:
All Event IDs reference the System Log

- 7034 – Service crashed unexpectedly
- 7035 – Service sent a Start/Stop control
- 7036 – Service started or stopped
- 7040 – Start type changed

  (Boot | On Request | Disabled)

Interpretation:
• A large amount of malware and worms in the wild utilize Services

- Services started on boot illustrate persistence (desirable in malware)
- Services can crash due to attacks like process injection

Local Users

Description:

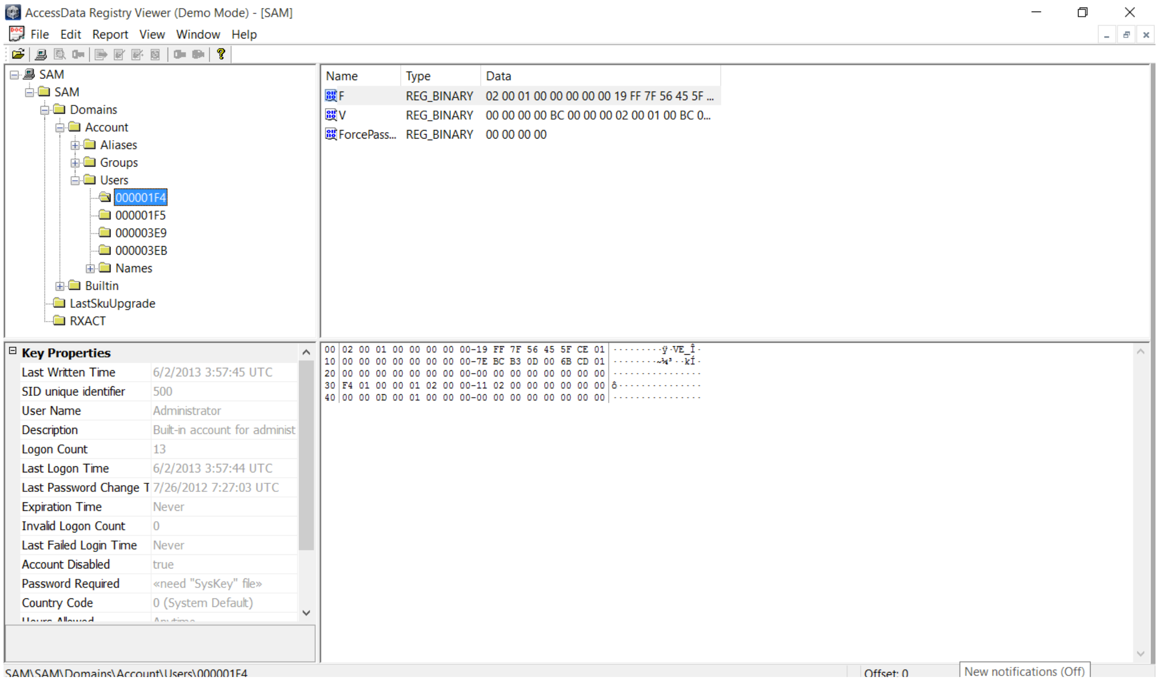
- Lists the local accounts of the system and their equivalent security identifiers (SIDs)

Tool:

Registry Viewer, RegRipper

1. Open AccessData Registry Viewer
2. Click File > Open and navigate to the saved SAM hive
3. Once open expand SAM > Domains > Account > Users
4. Click on one of the available Keys and then the ‘F’ Value
5. In the Key Properties in the lower left hand corner, useful user data is listed such as the user’s last logon time and logon count.

> **Note:** Using RegRipper will provide different information not provided by Registry Viewer such as group membership, account creation, account type, and SID.



Name	Type	Data
F	REG_BINARY	02 00 01 00 00 00 00 00 19 FF 7F 56 45 5F ...
V	REG_BINARY	00 00 00 00 BC 00 00 00 02 00 01 00 BC 0...
ForcePass...	REG_BINARY	00 00 00 00

Key Properties

Last Written Time	6/2/2013 3:57:45 UTC
SID unique identifier	500
User Name	Administrator
Description	Built-in account for administ
Logon Count	13
Last Logon Time	6/2/2013 3:57:44 UTC
Last Password Change T	7/26/2012 7:27:03 UTC
Expiration Time	Never
Invalid Logon Count	0
Last Failed Login Time	Never
Account Disabled	true
Password Required	«need "SysKey" file»
Country Code	0 (System Default)
Hours Allowed	Anytime

```
00 02 00 01 00 00 00 00 00-19 FF 7F 56 45 5F CE 01  ·········ÿ·VE_Î·
10 00 00 00 00 00 00 00 00-7E BC B3 0D 00 6B CD 01  ········~¼³··kÍ·
20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ················
30 F4 01 00 00 01 02 00 00-11 02 00 00 00 00 00 00  ô···············
40 00 00 0D 00 01 00 00 00-00 00 00 00 00 00 00 00  ················
```

SAM\SAM\Domains\Account\Users\000001F4

Offset: 0

Last Access Time On/Off

Description:

By default on Vista and up, the last access timestamp is disabled which means the last access time of files will not update.

Tool:

Registry Viewer

Location:

Path: System\CurrentControlSet\Control\FileSystem
NtfsDisableLastAccessUpdate =
0x01 == turned off
0x00 == turned on

Browser Usage

History

Description:
Records websites visited by date and time. Details stored for each local user account. Records number of times visited (frequency). Also tracks access of local system files.

Tool:
NirSoft BrowsingHistory View

Location:

- Internet Explorer
  - ➢ IE6-7

%USERPROFILE%\Local Settings\History\History.IE5

  - ➢ IE8-9

%USERPROFILE%\AppData\Local\Microsoft\Windows\History\History.IE5

  - ➢ IE10-11

%USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*
.dat

- Firefox
  - ➢ XP

%USERPROFILE%\Application Data\Mozilla\Firefox\Profiles\<random text>
.default\places.sqlite

  - ➢ Win7/8/10

%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<random
text>.default\places.sqlite

- Chrome
  - ➢ XP

%USERPROFILE%\Local Settings\Application Data\Google\Chrome\User Data\
Default\History

  - ➢ Win7/8/10

%USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\History

History span:
Location – HCU\software\Microsoft\windows\currentversion\internet settings\url history

URL Frequency

Description:
This is not directly related to "File Download" but details are stored for each local user account. It records the number of times a URL is visited (frequency).

Tool:

SQLite Manager

Location:

- Internet Explorer:
  - ➢ IE8-9

%USERPROFILE%\AppData\Roaming\Microsoft\Windows\IEDownloadHistory\index.dat

  - ➢ IE10-11

%USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat

- Firefox:
  - ➢ v3-25

%userprofile%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\downloads.sqlite

  - ➢ v26+

%userprofile%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\places.sqlite
Table:moz_annos

- Chrome:

Win7/8/10 %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default \History

Interpretation:
Many sites in history will list the files that were opened from remote sites and downloaded to the local system. History will record the access to the file on the website that was accessed via a link.

Cookies

Description:
Cookies give insight into what websites have been visited and what activities may have taken place there.

Tool:
FTK Imager

Location:

- Internet Explorer
  - ➢ IE8-9

%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies

  - ➢ IE10

%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies

  - ➢ IE11

%USERPROFILE%\AppData\Local\Microsoft\Windows\INetCookies

- Firefox
  - ➢ XP

%USERPROFILE%\Application Data\Mozilla\Firefox\Profiles\<random

text>.default\cookies.sqlite

- Win7/8/10

%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>
.default\cookies.sqlite

- Chrome
  - XP

%USERPROFILE%\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\

- Win7/8/10

%USERPROFILE%\AppData\Local\Google\Chrome\User Data\

Typed URL

Description:

• Records the last 25 web addresses typed by the user
• Increased to the last 50 addresses in IE10+
• Records the last time each URL is typed
• Auto-complete

Tool:

Registry Viewer, DCode

Use DCode to convert TypedURLTimes from Windows: 64 bit Hex

Location:

HCU\software\microsoft\internet explorer\TypedURLs
HCU\software\microsoft\internet explorer\TypedURLsTime

Cache

Description:

• The cache is where web page components can be stored locally to speed up subsequent visits
• Gives the investigator a “snapshot in time” of what a user was looking at online

- Identifies websites which were visited
- Provides the actual files the user viewed on a given website
- Cached files are tied to a specific local user account
- Timestamps show when the site was first saved and last viewed

Tool:

FTK Imager

Location:

- Internet Explorer
  - IE8-9

%USERPROFILE%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5

- IE10

%USERPROFILE%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5

- IE11

%USERPROFILE%\AppData\Local\Microsoft\Windows\INetCache\IE

- Firefox
  - XP

%USERPROFILE%\Local Settings\ApplicationData\Mozilla\Firefox\Profiles\<randomtext>.default\Cache

  - Win7/8/10

%USERPROFILE%\AppData\Local\Mozilla\Firefox\Profiles\<randomtext>.default\Cache

- Chrome
  - XP

%USERPROFILE%\Local Settings\Application Data\Google\Chrome\User Data\Default\Cache - data_# and f_######

- Win7/8/10

%USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\Cache\ - data_# and f_######

Session Restore

Description:

Automatic Crash Recovery features built into the browser.

Tool:

Structured Storage Viewer

Location:

- Internet Explorer
  - Win7/8/10

%USERPROFILE%/AppData/Local/Microsoft/Internet Explorer/Recovery

Firefox

  - Win7/8/10

%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\sessionstore.js

- Chrome
  - Win7/8/10

%USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\

Files = Current Session, Current Tabs, Last Session, Last Tabs

Interpretation:

• Historical websites viewed in each tab
• Referring websites
• Time session ended
• Modified time of .dat files in LastActive folder
• Time each tab opened (only when crash occurred)
• Creation time of .dat files in Active folder

Shares and Offline Caching

Description:

• List the open network shares on the local system

• List flags and configuration settings
• If no shares are listed, the user hasn't created any

Tool:
RegRipper, Registry Viewer

Location:
SYSTEM\currentcontrolset\services\lanmanserver\shares\

Miscellaneous

Autostart Programs

Description:
Common Keys for persistence

Tool:
RegRipper, Registry Viewer, Registry Explorer

Location:
NTUSER.dat\software\microsoft\windows\currentversion\run
NTUSER.dat\software\microsoft\windows\currentversion\RunOnce
NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\RunServices
HKLM\Software\microsoft\windows\currentversion\RunOnce
HKLM\Software\microsoft\windows\currentversion\Run
HKLM\Software\microsoft\windows\currentversion\policies\explorer\Run

HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices
HKLM\System\currencontrolset\services
Note: If service is set to 0x02 then the service will start at boot
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
userinit =

Registry

Deleted Keys/Values

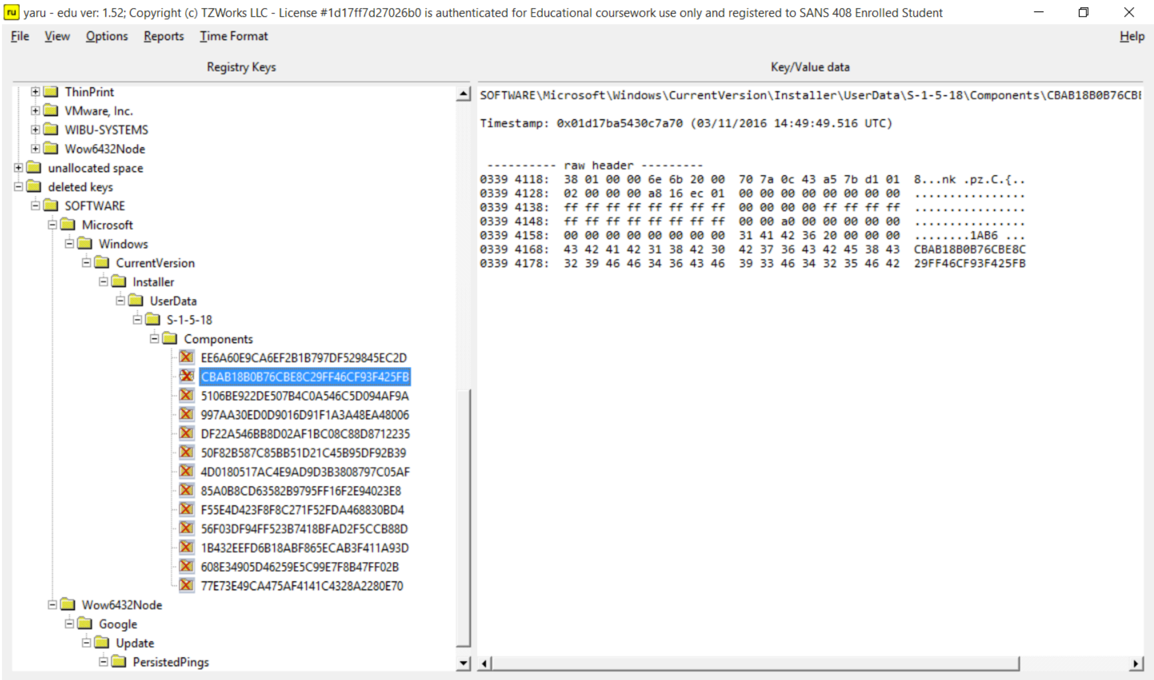
Description: Many times malware or someone trying to cover their tracks will delete registry Keys/Values. With that said useful data could be found in deleted registry keys along with many artifacts. This area is not as concrete as it may sound as the Registry is always in a state of change therefore, keys could be deleted for legitimate reasons. When keys are deleted, they will remain in the unallocated portion of the registry hive until it is overwritten.

Tool:
YARU (Yet Another Registry Utility)

Location:

1. Open YARU

2. Click File and select the suitable option to open a hive
3. Once open, scroll down the bottom
4. Expand Deleted Keys
5. Keys that have been deleted are marked with a red ‘ X ‘ as depicted below


yaru - edu ver: 1.52; Copyright (c) TZWorks LLC - License #1d17ff7d27026b0 is authenticated for Educational coursework use only and registered to SANS 408 Enrolled Student
File View Options Reports Time Format
Help
Registry Keys
Key/Value data
ThinPrint
VMware, Inc.
WIBU-SYSTEMS
Wow6432Node
unallocated space
deleted keys
SOFTWARE
Microsoft
Windows
CurrentVersion
Installer
UserData
S-1-5-18
Components
EE6A60E9CA6EF2B1B797DF529845EC2D
CBAB18B0B76CBE8C29FF46CF93F425FB
5106BE922DE507B4C0A546C5D094AF9A
997AA30ED0D9016D91F1A3A48EA48006
DF22A546BB8D02AF1BC08C88D8712235
50F82B587C85BB51D21C45B95DF92B39
4D0180517AC4E9AD9D3B3808797C05AF
85A0B8CD63582B9795FF16F2E94023E8
F55E4D423F8F8C271F52FDA468830BD4
56F03DF94FF523B7418BFAD2F5CCB88D
1B432EEFD6B18ABF865ECAB3F411A93D
608E34905D46259E5C99E7F8B47FF02B
77E73E49CA475AF4141C4328A2280E70
Wow6432Node
Google
Update
PersistedPings

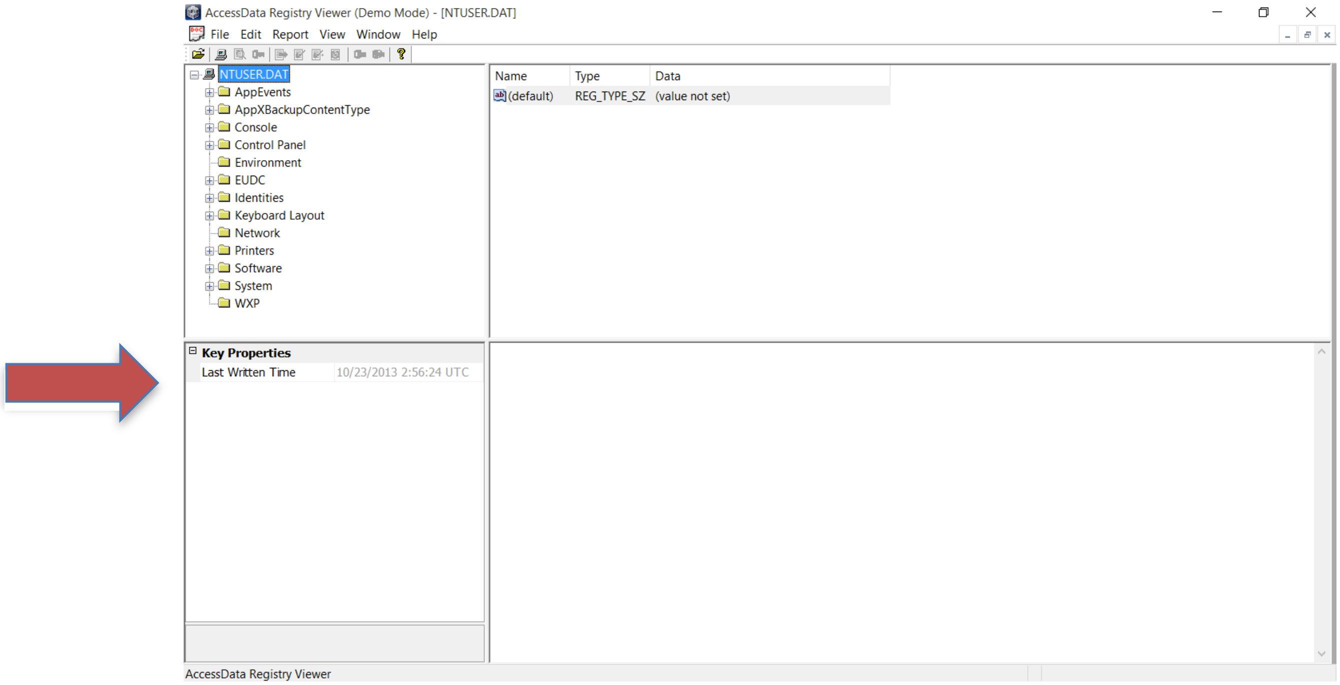
```
SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Components\CBAB18B0B76CBE

Timestamp: 0x01d17ba5430c7a70 (03/11/2016 14:49:49.516 UTC)


 ---------- raw header ---------
0339 4118:  38 01 00 00 6e 6b 20 00  70 7a 0c 43 a5 7b d1 01  8...nk .pz.C.{..
0339 4128:  02 00 00 00 a8 16 ec 01  00 00 00 00 00 00 00 00  ................
0339 4138:  ff ff ff ff ff ff ff ff  00 00 00 00 ff ff ff ff  ................
0339 4148:  ff ff ff ff ff ff ff ff  00 00 a0 00 00 00 00 00  ................
0339 4158:  00 00 00 00 00 00 00 00  31 41 42 36 20 00 00 00  ........1AB6 ...
0339 4168:  43 42 41 42 31 38 42 30  42 37 36 43 42 45 38 43  CBAB18B0B76CBE8C
0339 4178:  32 39 46 46 34 36 43 46  39 33 46 34 32 35 46 42  29FF46CF93F425FB
```

Key Last Write Time

1. Open AccessData Registry Viewer
2. Click File > Open
3. Navigate to the saved registry hives
   Note: You will not be able to use any live hives
4. Once a hive is open, click on the Key in question. The last write time will be depicted in the window in the bottom left corner as shown below.
   Note: You will only be able to tell what the Key last write time was and not any of the Values.


AccessData Registry Viewer (Demo Mode) - [NTUSER.DAT]
File Edit Report View Window Help
AccessData Registry Viewer

```
NTUSER.DAT
  AppEvents
  AppXBackupContentType
  Console
  Control Panel
  Environment
  EUDC
  Identities
  Keyboard Layout
  Network
  Printers
  Software
  System
  WXP
```

Name	Type	Data
(default)	REG_TYPE_SZ	(value not set)

Key Properties	
Last Written Time	10/23/2013 2:56:24 UTC

Backup

Description: The registry is backed up roughly every 10 days by the RegIdleBackup scheduled task. It will copy the SAM, DEFAULT, SYSTEM, SOFTWARE, and SECURITY hives. This applies to Vista, Win7, Win8, Win10, Server 2008, and Server 2016.

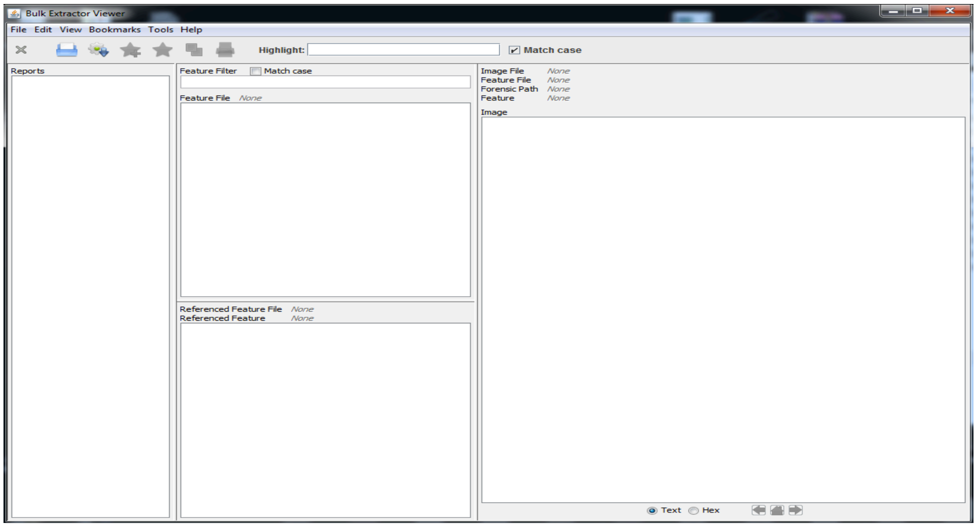
Location: %windir%\system32\config\regback

Automated Parsing and Keyword search

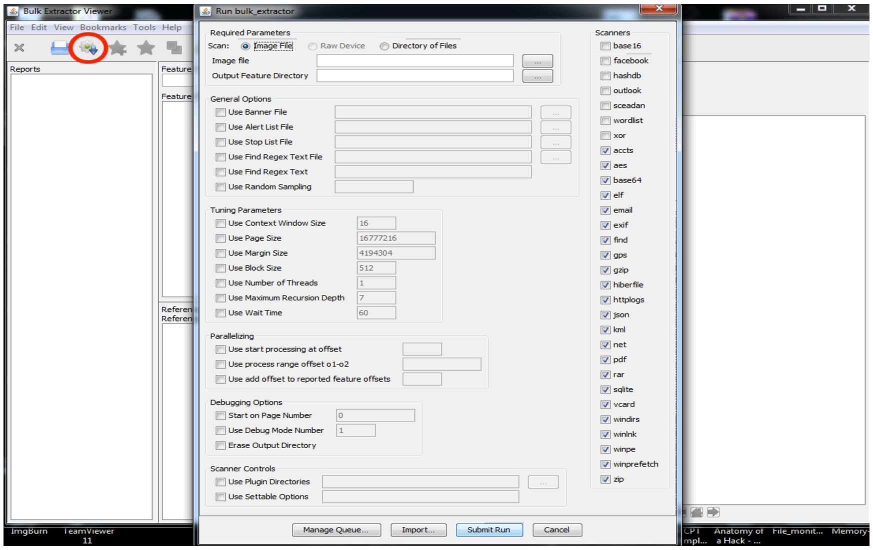
Bulk Extractor

Bulk Extractor helps streamline parsing an image. It allows us to provide a keyword list to search for and creates histograms of things that it finds.

1) Open Bulk Extractor Viewer (BEViewer). The below window will appear.



2) Click on the gear icon with the arrow on it as shown below. Another window will appear.



Required Parameters
Scan: ● Image File ○ Raw Device ○ Directory of Files
Image file
Output Feature Directory

General Options
☐ Use Banner File
☐ Use Alert List File
☐ Use Stop List File
☐ Use Find Regex Text File
☐ Use Find Regex Text
☐ Use Random Sampling

Tuning Parameters
☐ Use Context Window Size 16
☐ Use Page Size 16777216
☐ Use Margin Size 4194304
☐ Use Block Size 512
☐ Use Number of Threads 1
☐ Use Maximum Recursion Depth 7
☐ Use Wait Time 60

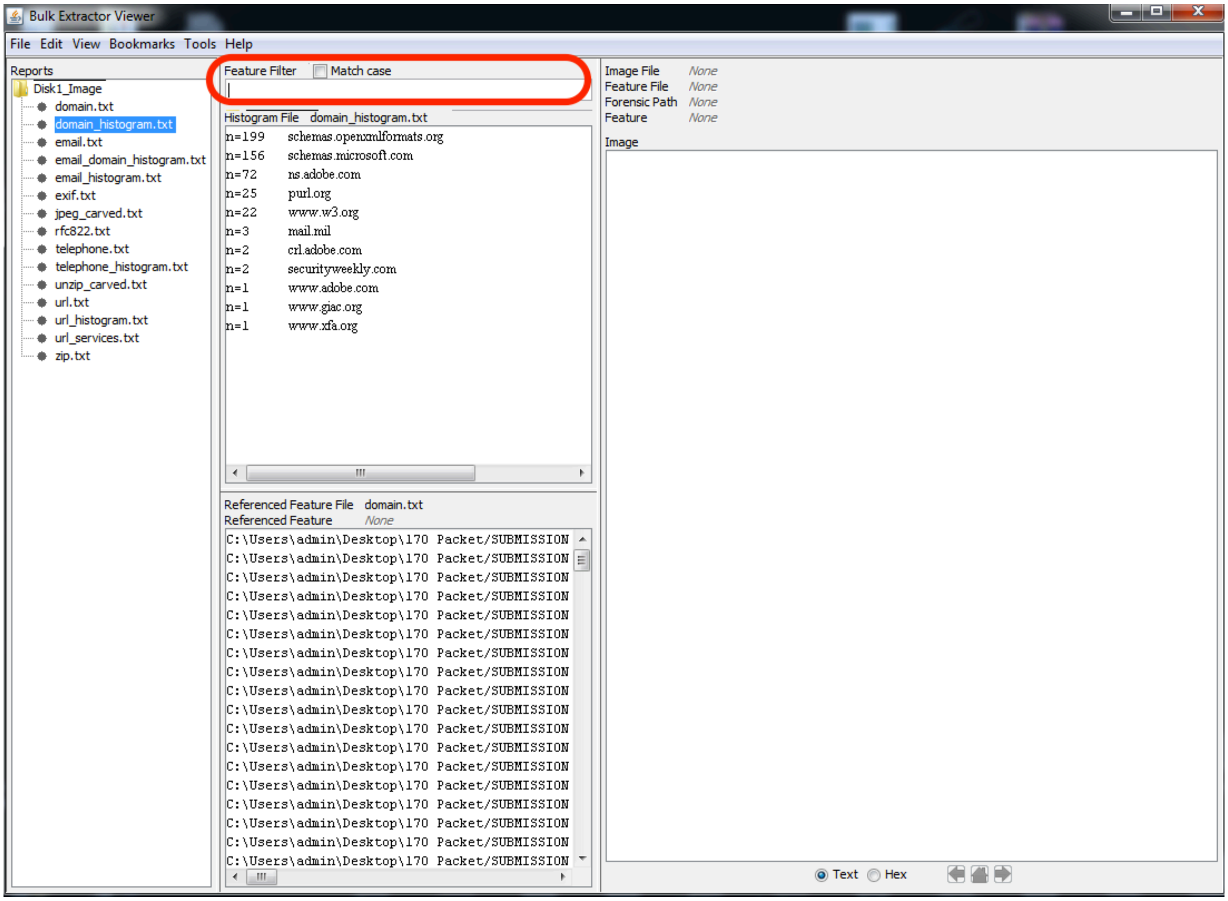
Parallelizing
☐ Use start processing at offset
☐ Use process range offset o1-o2
☐ Use add offset to reported feature offsets

Debugging Options
☐ Start on Page Number 0
☐ Use Debug Mode Number 1
☐ Erase Output Directory

Scanner Controls
☐ Use Plugin Directories
☐ Use Settable Options

Scanners
☐ base16
☐ facebook
☐ hashdb
☐ outlook
☐ sceadan
☐ wordlist
☐ xor
☑ accts
☑ aes
☑ base64
☑ elf
☑ email
☑ exif
☑ find
☑ gps
☑ gzip
☑ hiberfile
☑ httplogs
☑ json
☑ kml
☑ net
☑ pdf
☑ rar
☑ sqlite
☑ vcard
☑ windirs
☑ winlnk
☑ winpe
☑ winprefetch
☑ zip

3) Click on the "…" on the Image File line and select the image to parse.
4) Click on the "…" on the Output Feature Directory and select an output location.
5) Click Submit Run at the bottom of the screen.
6) Another window will appear showing the status.
7) Once it is complete, click close.
8) The newly created report will be visible in the top left area of the window. Results from the scanners are available as well. Clicking on a scanner will render the results from it. You can also filter and search for specific key words.



```
Reports
Disk1_Image
  domain.txt
  domain_histogram.txt
  email.txt
  email_domain_histogram.txt
  email_histogram.txt
  exif.txt
  jpeg_carved.txt
  rfc822.txt
  telephone.txt
  telephone_histogram.txt
  unzip_carved.txt
  url.txt
  url_histogram.txt
  url_services.txt
  zip.txt
```

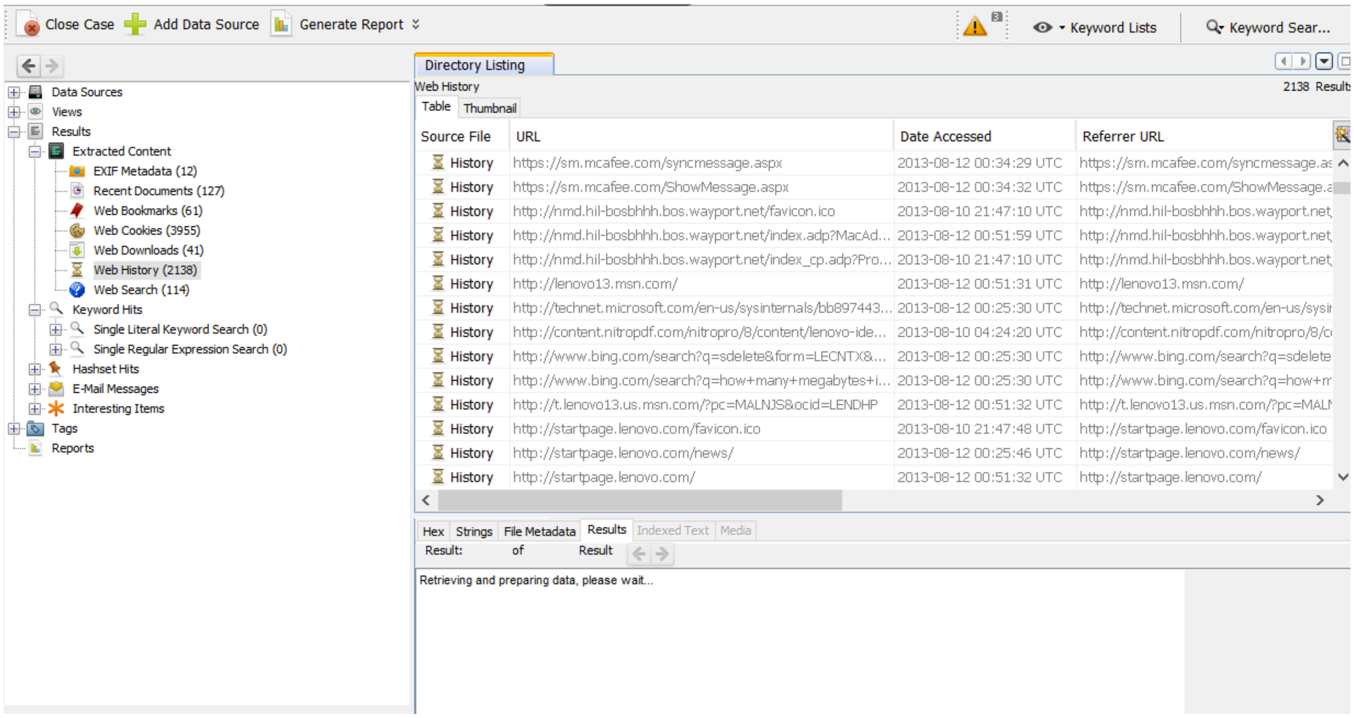
```
Histogram File   domain_histogram.txt
n=199     schemas.openxmlformats.org
n=156     schemas.microsoft.com
n=72      ns.adobe.com
n=25      purl.org
n=22      www.w3.org
n=3       mail.mil
n=2       crl.adobe.com
n=2       securityweekly.com
n=1       www.adobe.com
n=1       www.giac.org
n=1       www.xfa.org
```

```
Referenced Feature File   domain.txt
Referenced Feature        None
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
C:\Users\admin\Desktop\170 Packet/SUBMISSION
```

Autopsy

Autopsy helps streamline parsing an image. It also allows us to provide a keyword list and search for IOCs (stix format).

1. Launch Autopsy.
2. Click "Create New Case".
3. Fill in the fields regarding the case name and output directory.
4. Click Next.
5. Fill in requested information.
6. Select the image location and time zone information.
7. Click Next
8. Place a checkmark next to the items you went to parse the image for. If you click on an item, you can view advanced options regarding that option. Custom keyword searches can also be conducted as well.
9. Click Next
10. Once image is parsed, click Finish.
11. A window similar to the below will be displayed for you to sift through.



Web History — 2138 Results

Source File	URL	Date Accessed	Referrer URL
History	https://sm.mcafee.com/syncmessage.aspx	2013-08-12 00:34:29 UTC	https://sm.mcafee.com/syncmessage.as
History	https://sm.mcafee.com/ShowMessage.aspx	2013-08-12 00:34:32 UTC	https://sm.mcafee.com/ShowMessage.a
History	http://nmd.hil-bosbhhh.bos.wayport.net/favicon.ico	2013-08-10 21:47:10 UTC	http://nmd.hil-bosbhhh.bos.wayport.net
History	http://nmd.hil-bosbhhh.bos.wayport.net/index.adp?MacAd...	2013-08-12 00:51:59 UTC	http://nmd.hil-bosbhhh.bos.wayport.net
History	http://nmd.hil-bosbhhh.bos.wayport.net/index_cp.adp?Pro...	2013-08-10 21:47:10 UTC	http://nmd.hil-bosbhhh.bos.wayport.net
History	http://lenovo13.msn.com/	2013-08-12 00:51:31 UTC	http://lenovo13.msn.com/
History	http://technet.microsoft.com/en-us/sysinternals/bb897443...	2013-08-12 00:25:30 UTC	http://technet.microsoft.com/en-us/sysi
History	http://content.nitropdf.com/nitropro/8/content/lenovo-ide...	2013-08-10 04:24:20 UTC	http://content.nitropdf.com/nitropro/8/c
History	http://www.bing.com/search?q=sdelete&form=LECNTX&...	2013-08-12 00:25:30 UTC	http://www.bing.com/search?q=sdelete
History	http://www.bing.com/search?q=how+many+megabytes+i...	2013-08-12 00:25:30 UTC	http://www.bing.com/search?q=how+m
History	http://t.lenovo13.us.msn.com/?pc=MALNJS&ocid=LENDHP	2013-08-12 00:51:32 UTC	http://t.lenovo13.us.msn.com/?pc=MAL
History	http://startpage.lenovo.com/favicon.ico	2013-08-10 21:47:48 UTC	http://startpage.lenovo.com/favicon.ico
History	http://startpage.lenovo.com/news/	2013-08-12 00:25:46 UTC	http://startpage.lenovo.com/news/
History	http://startpage.lenovo.com/	2013-08-12 00:51:32 UTC	http://startpage.lenovo.com/

Result: of Result

Retrieving and preparing data, please wait...

Timeline

Once we have something that has our interest, we can develop a timeline based on the bit image and the applicable memory capture. From that, we can look for the file or time in question and then look at events leading up to it and immediately following. By doing so, we are likely to get a better understanding of what exactly happened and possibly other indicators to look for on our network.

1) Log into the Linux Sift workstation as Root
2) Type: fls -r -m {image.E01} < ./fls_bodyfile
3) Once complete, type: vol.py -f {mem_capture_location} --profile={suitable_profile} timeliner --output=body --output-file=./timeliner.body
4) Combine the files by typing: cat ./ timeliner.body >> ./fls_bodyfile
5) We will now produce the .csv file with our data using the below syntax.
   - Sets and returns the data in EST5EDT and records between 2012-04-02 and 2012-04-07.

     mactime -d -b ./fls_bodyfile -z EST5EDT 2012-04-02..2012-04-07 > ./some_file.csv

   - Sets and returns the data in EST5EDT.

     mactime -d -b ./fls_bodyfile -z EST5EDT > ./some_file.csv
6) Open the .csv using Excel
7) For optimum viewing, hide the Mode, UID, and GID column. It is also recommended to freeze the column headers by clicking Select View > Freeze Panes > Freeze Top Row. Depending on what you are searching for, you may want to adjust the sort and filter options as well.

Annex A: BareMonkey.sh (Volatility Plug-in Parser)

BareMonkey is on my github (www.github.com/wiredpulse). This BASH script will do the following for you against the memory capture you supply.

- Run all possible plug-ins for the specified OS and output to TXT
- Delete output TXT that contain no data
- Create a tarball
- Create a hash

The benefits are:

- Volatility is not needed after output
- Analyze output on any system with a text editor
- Portable

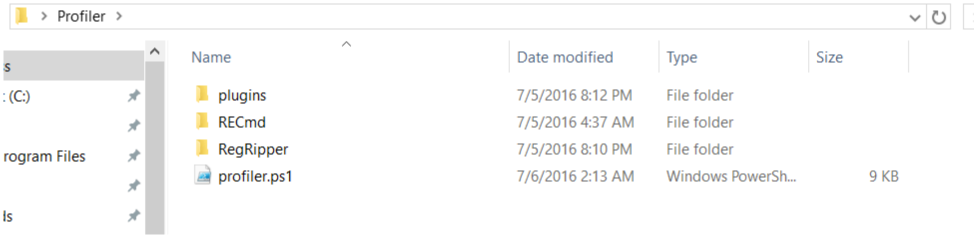
Annex B: Profiler.ps1 (Automated Process)

The profiler script will parse an image utilizing RegRipper, RECmd, and various PowerShell cmdlets. The output of the script will be in a text file called 'profiler.txt' and will contain information about said system such as system info, networking settings, firewall details, user data, autorun, service, and mru keys. The returned data will not provide you everything you need to do forensics on the image but it will present a lot of the data that you would find yourself looking for.

In order for this script to work, it will need to be in the same directory with the other supporting directories (RegRipper, RECmd, and plugins) that are included. A mounted image also needs to be available through FTK Imager.

The script and supporting files can be downloaded as a zip from my github (www.github.com/wiredpulse). To utilize the script, do the following:

1) Mount an image using FTK Imager.
2) Take note of the drive letter assigned to the mounted image.
3) Download the script and supporting files.
4) Unzip the contents of the zip.
5) Verify that a folder called ‘profiler’ is what was unzipped and inside of it should be the structure depicting in the picture.



Name	Date modified	Type	Size
plugins	7/5/2016 8:12 PM	File folder	
RECmd	7/5/2016 4:37 AM	File folder	
RegRipper	7/5/2016 8:10 PM	File folder	
profiler.ps1	7/6/2016 2:13 AM	Windows PowerSh...	9 KB

6) Open PowerShell.
7) Navigate to the Profiler directory containing the script.
8) Type profiler.ps1.
9) Input the drive letter as requested.
10) Waite for the script to finish.

Annex C: Report Template

A report template is an attachment to this document for your convenience.