SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Windows Forensic Analysis

POSTER

You Can't Protect What You Don't Know About

digital-forensics.sans.org

$25.00
DFIR-Windows_v4.2_11-17

Windows® Time Rules

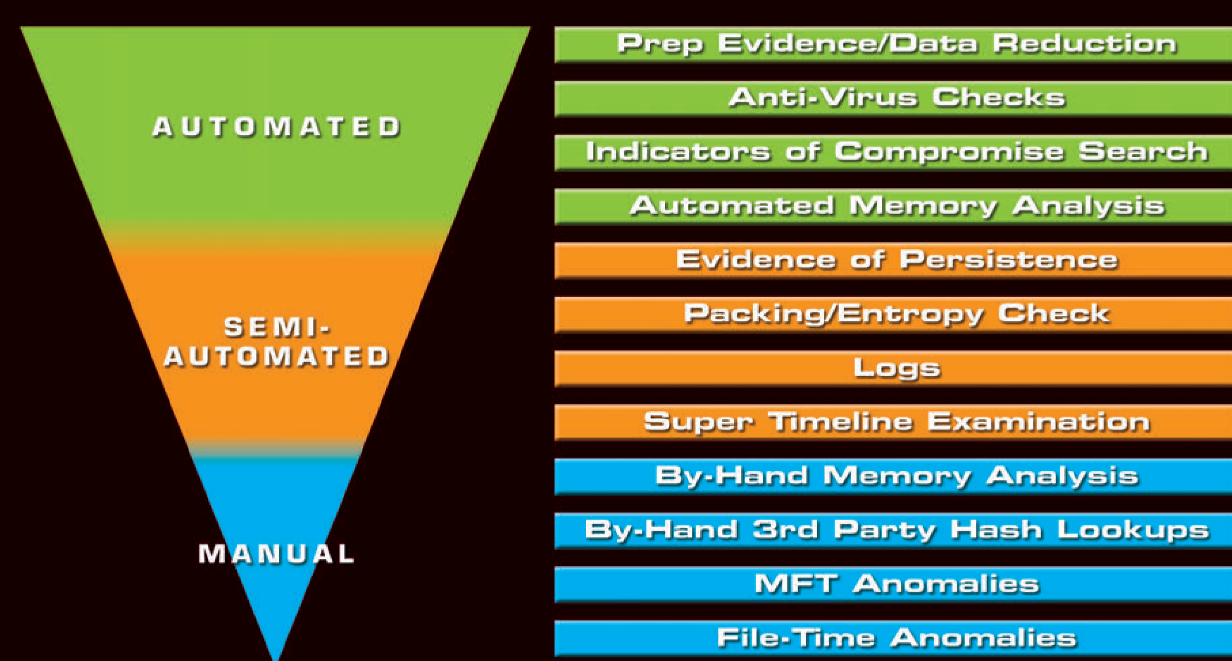
$STDINFO

	File Rename	Local File Move	Volume File Move	File Copy	File Access	File Modify	File Creation	File Deletion
Modified	No Change	No Change	No Change	No Change	No Change	Change	Change	No Change
Access	No Change	No Change	Change	Change	Change (No Change on Win7/8)	No Change	Change	No Change
Creation	No Change	No Change	Change	Change	No Change	No Change	Change	No Change
Metadata	Change	Change	Changed	Change	No Change	Change	Change	No Change

$FILENAME

	File Rename	Local File Move	Volume File Move	File Copy	File Access	File Modify	File Creation	File Deletion
Modified	No Change	Change	Change	Change	No Change	No Change	Change	No Change
Access	No Change	Change	Change	Change	No Change	No Change	Change	No Change
Creation	No Change	No Change	Change	Change	No Change	No Change	Change	No Change
Metadata	No Change	Change	Change	Change	No Change	No Change	Change	No Change

Finding Unknown Malware – Step-By-Step



Finding unknown malware is an intimidating process to many, but can be simplified by following some simple steps to help narrow your search. This is not an easy process, but using the techniques in this chart you will learn how to narrow the 80,000 files on a typical machine down to the 1-4 files that are possible malware. This process of Malware Funneling is key to your quick and efficient analysis of compromised hosts and will involve most of the skills you have learned or strengthened in FOR408 Windows Forensics and FOR508 Advanced Forensics and Incident Response

STEP 1: Prep Evidence/Data Reduction

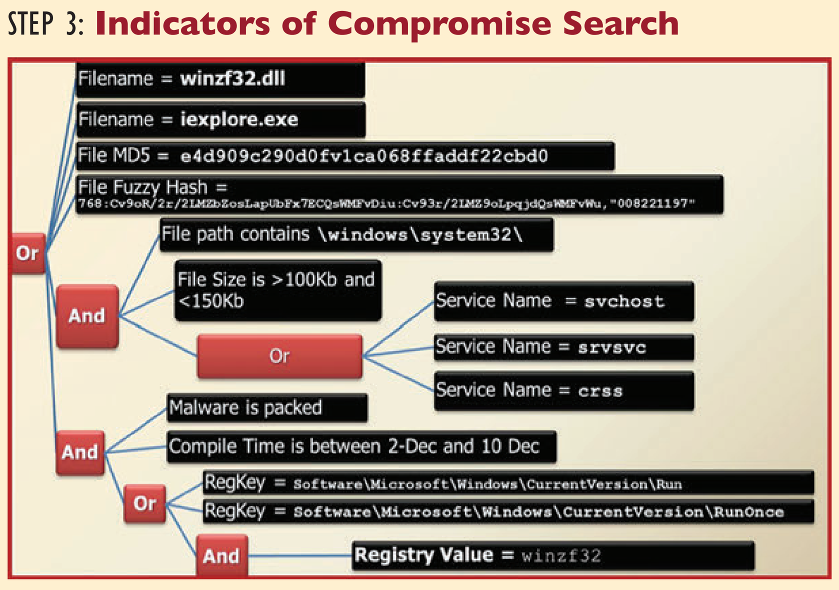
- **Carve and Reduce Evidence**
  - Gather Hash List from similar system (NSRL, md5deep)
  - Carve/Extract all `.exe` and `.dll` files from unallocated space
    - `foremost`
    - `sorter` (exe directory)
    - `bulk_extractor`
- **Prep Evidence**
  - Mount evidence image in Read-Only Mode
  - Locate memory image you collected
  - Optional: Convert `hiberfil.sys` (if it exists) to a raw image using Volatility

STEP 2: Anti-Virus Checks

Run the mounted drive through an anti-virus scanner with the latest updates.

Anti-virus scanners employ hundreds of thousands of signatures that can quickly identify well-known malware on a system. First, download the latest anti-virus signatures and mount your evidence for analysis. Use a "deep" scan when available and consider scanning your mounted drive with multiple anti-virus engines to take advantage of their scanning and signature differences. Get in the habit of scanning files exported from your images such as deleted files, data carving results, Sorter output, and email attachments. While anti-virus will not be effective on 0-day or unknown malware, it will easily find the low hanging fruit.

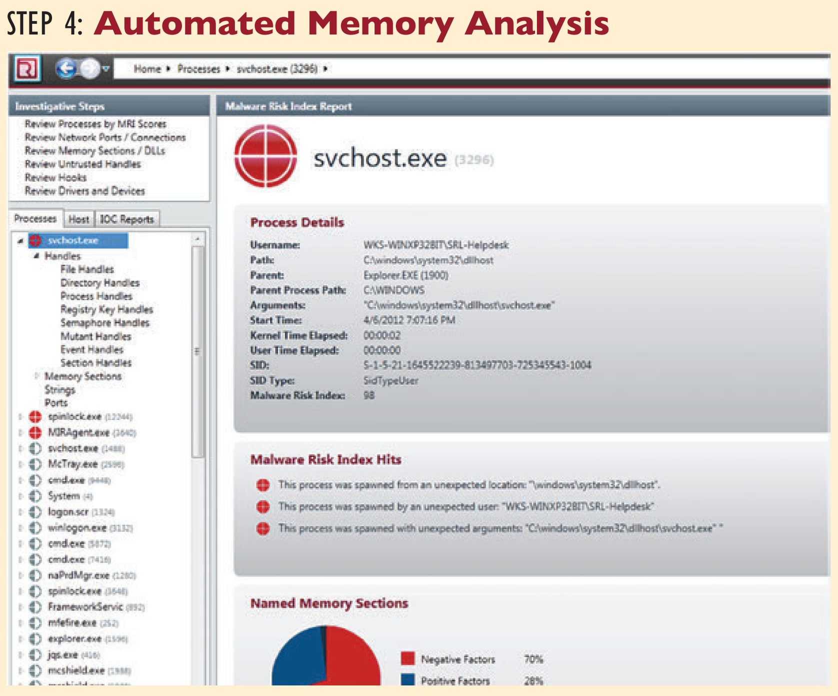
STEP 3: Indicators of Compromise Search



Using indicators of compromise (IOCs) is a very powerful technique to identify malware components on a compromised host. IOCs are implemented as a combination of boolean expressions that identify specific characteristics of malware. If these characteristics are found, then you may have a hit. An IOC should be general enough to find modified versions of the same malware, but specific enough to limit false positives. There are two types of indicators: host-based (shown above), and network-based (similar to snort signatures plus additional data). The best IOCs are usually created by reversing malware and application behavioral analysis.

What Works?
OpenIOC Framework - **`openioc.org`**
IOC Editor
Redline
STIX

STEP 4: Automated Memory Analysis



- **Behavior Ruleset**
  - Code Injection Detection
  - Process Image Path Verification
    - **`svchost`** outside **`system32`** = **Bad**
  - Process User Verification (SIDs)
    - **`dllhost`** running as **`admin`** = **Bad**
  - Process Handle Inspection
    - **`iexplore.exe`** opening **`cmd.exe`** = **Bad**
    - **`)!voqa.i4`** = known Poison Ivy mutant
- **Verify Digital Signatures**
  - Only available during live analysis
  - Executable, DLL, and driver sig checks
  - Not signed?
    - Is it found in >75% of all processes?

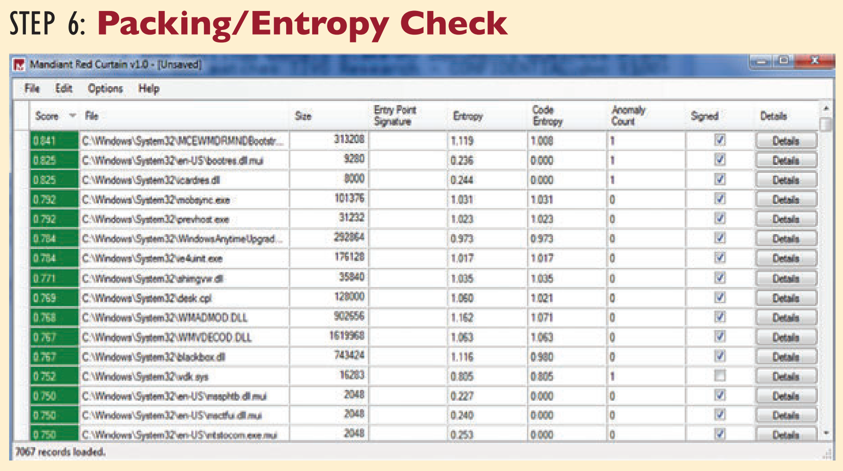
What Works?
MANDIANT Redline
https://www.mandiant.com/resources/download/redline
Volatility Malfind
https://github.com/volatilityfoundation

STEP 5: Evidence of Persistence

Malware wants to hide, but it also wants to survive a reboot. Malware persistence is extremely common and is an excellent way to find hidden malware. Persistence comes in many forms. The simplest mechanism is via scheduled tasks and the "at" command. Other popular persistence mechanisms include Windows Services and auto-start locations. Adversaries can run their malware as a new service or even replace an existing service. There are numerous Windows Registry mechanisms to auto-start an executable at boot or login. Using a tool called autorunsc.exe will easily parse the autostart locations across scheduled tasks, services, and registry keys. While these are the most common, keep in mind there are more advanced techniques. For example, the Mebromi malware even flashes the BIOS to persist. Attacks of this nature are rare because even the simplest of techniques are effective, allowing attackers to maintain persistence for long periods of time without being discovered.

What Works? Autorunsc.exe from Microsoft sysinternals
http://technet.microsoft.com/en-us/sysinternals/bb963902

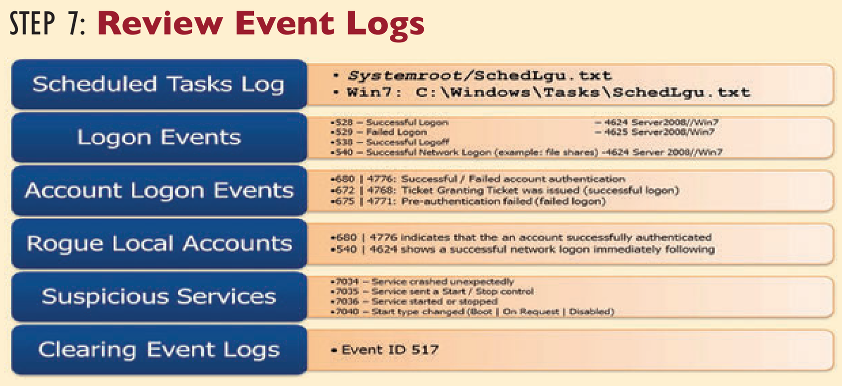
STEP 6: Packing/Entropy Check



- **Scan the file system or common locations for possible malware**
  - Indication of packing
  - Entropy test
  - Compiler and packing signatures identification
  - Digital signature or signed driver checks

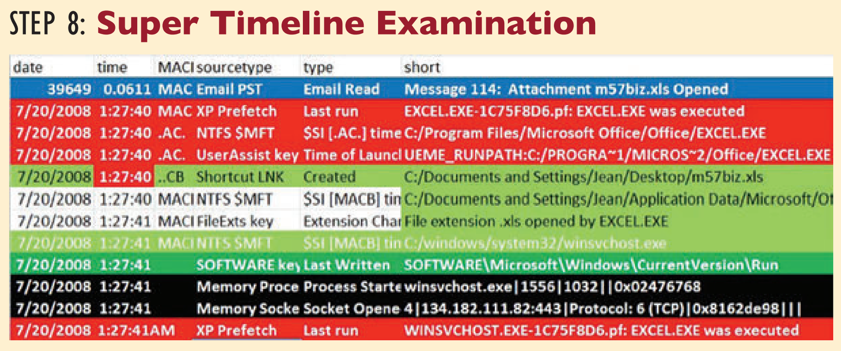
What Works?
DensityScout http://cert.at/downloads/software/densityscout_en.html
Sigcheck - http://technet.microsoft.com/en-us/sysinternals/bb897441

STEP 7: Review Event Logs



What Works?
logparser - www.microsoft.com/download/en/details.aspx?id=24659
Event Log Explorer - http://eventlogxp.com
Log Parser Lizard - www.lizard-labs.net

STEP 8: Super Timeline Examination



date	time	MACI sourcetype	type	short
39649	0.0611	MAC Email PST	Email Read	Message 114: Attachment m57biz.xls Opened
7/20/2008	1:27:40	MAC XP Prefetch	Last run	EXCEL.EXE-1C75F8D6.pf: EXCEL.EXE was executed
7/20/2008	1:27:40	.AC. NTFS $MFT	$SI [.AC.] time	C:/Program Files/Microsoft Office/Office/EXCEL.EXE
7/20/2008	1:27:40	.AC. UserAssist key	Time of Launcl	UEME_RUNPATH:C:/PROGRA~1/MICROS~2/Office/EXCEL.EXE
7/20/2008	1:27:40	..CB Shortcut LNK	Created	C:/Documents and Settings/Jean/Desktop/m57biz.xls
7/20/2008	1:27:40	MACI NTFS $MFT	$SI [MACB] tin	C:/Documents and Settings/Jean/Application Data/Microsoft/Ot
7/20/2008	1:27:41	MACI FileExts key	Extension Char	File extension .xls opened by EXCEL.EXE
7/20/2008	1:27:41	MACI NTFS $MFT	$SI [MACB] tin	C:/windows/system32/winsvchost.exe
7/20/2008	1:27:41	SOFTWARE key	Last Written	SOFTWARE\Microsoft\Windows\CurrentVersion\Run
7/20/2008	1:27:41	Memory Proce	Process Starte	winsvchost.exe|1556|1032||0x02476768
7/20/2008	1:27:41	Memory Socke	Socket Opene	4|134.182.111.82:443|Protocol: 6 (TCP)|0x8162de98|||
7/20/2008	1:27:41AM	XP Prefetch	Last run	WINSVCHOST.EXE-1C75F8D6.pf: EXCEL.EXE was executed

Once you are down to about 10-20 candidates, it is a good time to identify where those files show up in your timeline. The additional context of seeing other files in close temporal proximity to your candidates allows you to identify false positives and focus on those files most likely to be malicious. In the above example, we see the creation of the file winsvchost.exe in the C:\Windows\System32\ directory. If this were one of your candidate files, you would clearly see artifacts that indicate a spear phishing attack surrounding that file's creation time. Notably, an .XLS file was opened via email, winsvchost.exe was executed, an auto-start persistence mechanism was created, and finally, a network socket was opened. All within one second! Contextual clues in temporal proximity to the files you are examining are quite useful in your overall case.

What Works? log2timeline found in SIFT Workstation
http://computer-forensics.sans.org/community/downloads

STEP 9: By-Hand Memory Analysis

1. **Identify rogue processes**
   - Name, path, parent, command line, start time, SIDs
2. **Analyze process DLLs and handles**
3. **Review network artifacts**
   - Suspicious ports, connections, and processes
4. **Look for evidence of code injection**
   - Injected memory sections and process hollowing
5. **Check for signs of a rootkit**
   - SSDT, IDT, IRP, and inline hooks
6. **Dump suspicious processes and drivers**
   - Review strings, anti-virus scan, reverse-engineer

Memory analysis is one of the most powerful tools for finding malware. Malware has to run to be effective, creating a footprint that can often be easily discovered via memory forensics. A standard analysis can be broken down into six major steps. Some of these steps might be conducted during incident response, but using a memory image gives deeper insight and overcomes any rootkit techniques that malware uses to protect itself. Memory analysis tools are operating-system specific. Since each tool gathers and displays information differently, use multiple tools to check your results.

What Works? Volatility http://code.google.com/p/volatility
Mandiant Redline www.mandiant.com/products/free_software/redline

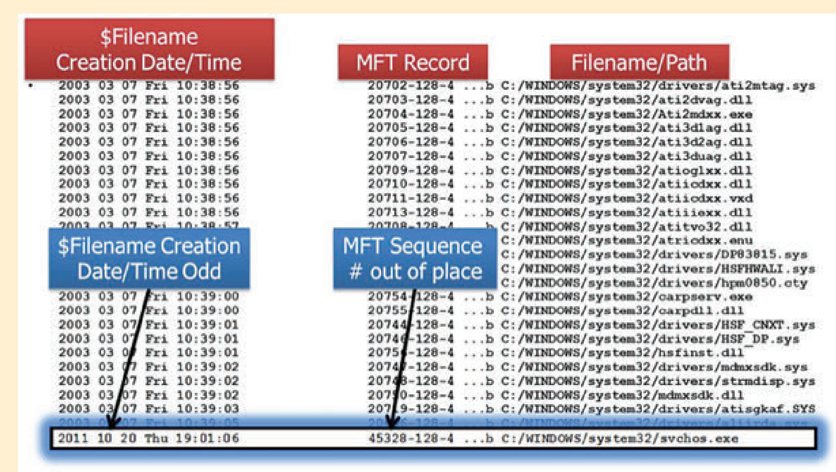
STEP 10: By-Hand Third-Party Hash Lookups

Hash lookups to eliminate known good files and identify known bad files is a useful technique when narrowing down potential malware. The National Software Reference Library also provides a robust set of known good hashes for use.

VirusTotal will scan a file through over 40 different A/V scanners to determine if any of the current signatures detect the malware. VirusTotal also allows its database to be searched via MD5 hashes, returning prior analyses for candidate files with the same MD5.

What Works?
VirusTotal www.virustotal.com
NSRL Query http://rjhansen.github.io/nsrllookup

STEP 11: Master File Table Anomalies



A typical file system has hundreds of thousands of files. Each file has its own MFT Record Number. Because of the way operating systems are installed, it's normal to see files under entire directory structures written to disk with largely sequential MFT Record Number values. For example, above is a partial directory listing from a Windows NTFS partition's %SystemRoot%\System32 directory, sorted by date. Note that the MFT Record Number values are largely sequential and, with some exceptions, tend to align with the file creation times. As file systems are used over the years and new patches are applied causing files to be backed up and replaced, the ordering of these files by MFT Record Number values can break down. Surprisingly, this ordering remains sufficiently intact on many systems, even after years of use, that we can use it to spot files of interest. This will not happen every time, as MFT entries are recycled fairly quickly, but in many cases an outlier can be identified.

STEP 12: File-Time Anomalies

H	I	M
Filename #1	Std Info Creation date	FN Info Creation date
winsvchost	8/12/2003 2:41	2/18/2007 20:41

- **Timestamp Anomalies**
  - $SI Time is before $FN Time
  - Nanosecond values are all zeroes

One of the ways to tell if file time backdating occurred on a Windows machine is to examine the NTFS $Filename times compared to the times stored in $Standard Information. Tools such as timestomp allow hackers to backdate a file to an arbitrary time of their choosing. Generally, hackers do this only to programs they are trying to hide in the system32 or similar system directories. Those directories and files would be a great place to start. Look to see if the $Filename (FN) creation time occurs after the $Standard Information creation time, as this often indicates an anomaly.

What Works?
analyzeMFT.py found on SIFT Workstation and
https://github.com/dkovar/analyzeMFT
log2timeline found on SIFT Workstation

STEP 13: You Have Malware! Now What?

- **Hand it to Malware Analyst**
  - FOR610: Reverse Engineering Malware
  - Hand over sample, relevant configuration files, memory snapshot
- **Typical Output from Malware Analyst**
  - Host-based indicators
  - Network-based indicators
  - Report on malware capabilities and purpose
- **You can now find additional systems compromised by the malware you found**

LEARN REM

SANS Windows Artifact Analysis: Evidence of...

The "Evidence of..." categories were originally created by SANS Digital Forensics and Incidence Response faculty for the SANS course FOR500: Windows Forensics. The categories map a specific artifact to the analysis questions that it will help to answer. Use this poster as a cheat-sheet to help you remember where you can discover key Windows artifacts for computer intrusion, intellectual property theft, and other common cyber crime investigations.

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.

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 file info from the OpenSave dialog by specific extension

E-mail Attachments

Description:
The e-mail industry estimates that 80% of e-mail data is stored via attachments. E-mail standards only allow text. Attachments must be encoded with MIME/base64 format.

Location:
Outlook
XP `%USERPROFILE%\Local Settings\ApplicationData\Microsoft\Outlook`
Win7/8/10 `%USERPROFILE%\AppData\Local\Microsoft\Outlook`

Interpretation:
MS Outlook data files found in these locations include OST and PST files. One should also check the OLK and Content.Outlook folder, which might roam depending on the specific version of Outlook used. For more information on where to find the OLK folder this link has a handy chart:
http://www.hancockcomputertech.com/blog/2010/01/06/find-the-microsoft-outlook-temporary-olk-folder

Skype History

Description:
- Skype history keeps a log of chat sessions and files transferred from one machine to another
- This is turned on by default in Skype installations

Location:
XP
`C:\Documents and Settings\<username>\Application\Skype\<skype-name>`
Win7/8/10
`C:\%USERPROFILE%\AppData\Roaming\Skype\<skype-name>`

Interpretation:
Each entry will have a date/time value and a Skype username associated with the action.

Browser Artifacts

Description:
Not directly related to "File Download". Details stored for each local user account. Records number of times visited (frequency).

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\<random text>.default\downloads.sqlite`
- *v26+* `%userprofile%\AppData\Roaming\Mozilla\ Firefox\Profiles\<random text>.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.

Downloads

Description:
Firefox and IE has 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.

Location:
Firefox:
- *XP* `%userprofile%\Application Data\Mozilla\ Firefox\Profiles\<random text>.default\downloads.sqlite`
- *Win7/8/10* `%userprofile%\AppData\Roaming\Mozilla\ Firefox\Profiles\<random text>.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

ADS Zone.Identifer

Description:
Starting with XP SP2 when files are downloaded from the "Internet Zone" via a browser to a NTFS volume, an alternate data stream is added to the file. The alternate data stream is named "Zone.Identifier."

Interpretation:
Files with an ADS Zone.Identifier and contains ZoneID=3 were downloaded from the Internet
- URLZONE_TRUSTED = ZoneID = 2
- URLZONE_INTERNET = ZoneID = 3
- URLZONE_UNTRUSTED = ZoneID = 4

Program Execution

UserAssist

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

Location:
NTUSER.DAT HIVE
`NTUSER.DAT\Software\Microsoft\Windows\Currentversion\Explorer\UserAssist\{GUID}\Count`

Interpretation:
All values are ROT-13 Encoded
- GUID for XP
  - **75048700** Active Desktop
- GUID for Win7/8/10
  - **CEBFF5CD** Executable File Execution
  - **F4E57C4B** Shortcut File Execution

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.

Example:
Notepad.exe was last run using the `C:\%USERPROFILE%\Desktop` folder

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.

Location:
NTUSER.DAT HIVE
`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.

RecentApps

Description:
Program execution launched on the Win10 system are tracked in the RecentApps key

Location:
Win10
`NTUSER.DAT\Software\Microsoft\Windows\Current Version\Search\RecentApps`

Interpretation:
Each GUID key points to a recent application.
AppID = Name of Application
LastAccessTime = Last execution time in UTC
LaunchCount = Number of times 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

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 1,024 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.

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 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

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)
  - Win8-10 will contain last 8 times of execution

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

Location:
Win7/8/10
`C:\Windows\AppCompat\Programs\Amcache.hve (Windows 7/8/10)`
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.

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 file info from the OpenSave dialog by specific extension

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.

Example:
Notepad.exe was last run using the `C:\Users\Rob\Desktop` folder

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.

Location:
NTUSER.DAT
`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 when and location where 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.

Location:
`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

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.

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)

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.

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 Win8-10
- (exename)-(hash).pf

Location:
WinXP/7/8/10
`C:\Windows\Prefetch`

Interpretation:
- Can examine each .pf file to look for file handles recently used
- Can examine each .pf file to look for device handles recently used

IE|Edge 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.

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

Deleted 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.

Location:
NTUSER.DAT HIVE
`NTUSER.DAT\Software\Microsoft\Search Assistant\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 bar on a Windows 7 machine.

Location:
Win7/8/10 NTUSER.DAT Hive
`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.

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.

Thumbs.db

Description:
Hidden file in directory where images on machine exist stored in a smaller thumbnail graphics. thumbs.db catalogs pictures in a folder and stores a copy of the thumbnail even if the pictures were deleted.

Location:
WinXP/Win8|8.1
Automatically created anywhere with homegroup enabled
Win7/10
Automatically created anywhere and accessed via a UNC Path (local or remote)

Interpretation:
Include:
- Thumbnail Picture of Original Picture
- Document Thumbnail – Even if Deleted
- Last Modification Time (XP Only)
- Original Filename (XP Only)

Thumbscache

Description:
Thumbnails of pictures, office documents, and folders exist in a database called the thumbcache. Each user will have their own database based on the thumbnail sizes viewed by the user (small, medium, large, and extra-large)

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. 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.

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.

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

IE|Edge 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 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.

Location:
Internet Explorer:
- *IE6-7* `%USERPROFILE%\LocalSettings\History\History.IE5`
- *IE8-9* `%USERPROFILE%\AppData\Local\Microsoft\WindowsHistory\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

Physical Location

Timezone

Description:
Identifies the current system time zone.

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 History

Description:
- Identify networks that the computer has been connected to
- Networks could be wireless or wired
- Identify domain name/intranet name
- Identify SSID
- Identify Gateway MAC Address

Location:
Win7/8/10 SOFTWARE HIVE
- `SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Unmanaged`
- `SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Managed`
- `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 it based 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.

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\User Data\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.

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`

Proper digital forensic and incident response analysis is essential to successfully solve today's complex cases. Each analyst should examine the artifacts and then analyze the activity that they describe to determine a clear picture of which user was involved, what the user was doing, when the user was doing it, and why. The data here will help you find multiple locations that can substantiate facts related to your casework.

External Device/USB Usage

Key Identification

Description:
Track USB devices plugged into a machine.

Location:
- `SYSTEM\CurrentControlSet\Enum\USBSTOR`
- `SYSTEM\CurrentControlSet\Enum\USB`

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.

Location: *First Time*
- Plug and Play Log Files
  XP `C:\Windows\setupapi.log`
  Win7/8/10 `C:\Windows\inf\setupapi.dev.log`

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

Location: *First, Last, and Removal Times (Win7/8/10 Only)*
System Hive `\CurrentControlSet\Enum\USBSTOR\Ven_Prod_Version\USB`
iSerial `#\Properties\{83da6326-97a6-4088-9453-a1923f573b29}\####`
0064 = First Install (Win7-10)
0066 = Last Connected (Win8-10)
0067 = Last Removal (Win8-10)

User

Description:
Find User that used the Unique USB Device.

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

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, which is hardcoded into the device firmware.)

Location:
- `SOFTWARE\Microsoft\WindowsNT\CurrentVersion\ENDMgmt`
- 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.

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`
- `SYSTEM\MountedDevices`
  - Examine Drive Letters 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.

Shortcut (LNK) Files

Description:
Shortcut files automatically created by Windows
- Recent Items
- Open local and remote data files and documents will generate a shortcut file (.lnk)

Location:
XP
- `%USERPROFILE%\Recent`

Win7/8/10
- `%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent`
- `%USERPROFILE%\AppData\Roaming\Microsoft\Office\Recent`

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.

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)

Account Usage

Last Login

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

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 local user has been changed.

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.

Location:
Win7/8/10
`%system root%\System32\winevt\logs\Security.evtx`

Interpretation:
- Win7/8/10 – Interpretation
- 4624 – Successful Logon
- 4625 – Failed Logon
- 4634 | 4647 – Successful Logoff
- 4648 – Logon using explicit credentials (Runas)
- 4672 – Account logon with superuser rights (Administrator)
- 4720 – An account was created

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.

Location:
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.

Location: *Security Log*
Win7/8/10
`%SYSTEM ROOT%\System32\winevt\logs\Security.evtx`

Interpretation:
- Win7/8/10 - Interpretation
  - Event ID 4778 – Session Connected/Reconnected
  - Event ID 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 (4779) followed by RDP connection (4778)

Services Events

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

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)
7045 – A service was installed on the system (Win2008R2+)
4697 – A service was installed on the system (from Security log)

Interpretation:
- All Event IDs except 4697 reference the System Log
- 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

Scheduled Tasks

Description:
Identify and audit scheduled tasks

Location:
Win7/8/10
`%system root%\System32\winevt\logs\Security.evtx`
`%system root%\System32\winevt\logs\Microsoft-Windows-TaskScheduler%4Maintenance.evtx`

Interpretation:
- 106 | 4698 – Scheduled task created (Task Scheduler/Security Log)
- 140 | 4702 – Scheduled task updated (Task Scheduler/Security Log)
- 141 | 4699 – Scheduled task deleted (Task Scheduler/Security Log)
- 200 | 201 – Scheduled task executed/completed (Task Scheduler Log)
- 4700 | 4701 – Scheduled task enabled/disabled (Security Log)

Investigative Notes
- Scheduled tasks can be executed both locally and remotely.
- Remotely scheduled tasks also cause Logon (ID 4624) Type 3 events

Authentication Events

Description:
Authentication mechanisms

Location:
Recorded on system that authenticated credentials
Local Account/Workgroup = on workstation
Domain/Active Directory = on domain controller
Win7/8/10
`%SYSTEM ROOT%\System32\winevt\logs\Security.evtx`

Interpretation:
Event ID Codes (NTLM protocol)
- 4776: Successful/Failed account authentication

Event ID Codes (Kerberos protocol)
- 4768: Ticket Granting Ticket was granted (successful logon)
- 4769: Service Ticket requested (access to server resource)
- 4771: Pre-authentication failed (failed logon)

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.

Location:
Internet Explorer
- *IE6-7* `%USERPROFILE%\Local Settings\History\History.IE5`
- *IE8-9* `%USERPROFILE%\AppData\Local\Microsoft\Windows\History\History.IE5`
- *IE10,11,Edge* `%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`

Cookies

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

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`
- *Edge* `%USERPROFILE%\AppData\Local\Packages\microsoft.microsoftedge_<APPID>\AC\MicrosoftEdge\Cookies`

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\Default\Local Storage\`

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

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`
- *Edge* `%USERPROFILE%\AppData\Local\Packages\microsoft.microsoftedge_<APPID>\AC\MicrosoftEdge\Cache`

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.

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

Flash & Super Cookies

Description:
Local Stored Objects (LSOs), or Flash Cookies, have become ubiquitous on most systems due to the extremely high penetration of Flash applications across the Internet. They tend to be much more persistent because they do not expire, and there is no built-in mechanism within the browser to remove them. In fact, many sites have begun using LSOs for their tracking mechanisms because they rarely get cleared like traditional cookies.

Location:
Win7/8/10
`%APPDATA%\Roaming\Macromedia\FlashPlayer\#SharedObjects\<randomprofileid>`

Interpretation:
- Websites visited
- User account used to visit the site
- When cookie was created and last accessed

Google Analytics Cookies

Description:
Google Analytics (GA) has developed an extremely sophisticated methodology for tracking site visits, user activity, and paid search. Since GA is largely free, it has a commanding share of the market, estimated at over 80% of sites using traffic analysis and over 50% of all sites.

__utma – *Unique visitors*
- Domain Hash
- Visitor ID
- Cookie Creation Time
- Time of 2nd most recent visit
- Time of most recent visit
- Number of visits

__utmb – *Session tracking*
- Domain hash
- Page views in current session
- Outbound link clicks
- Time current session started

__utmz – *Traffic sources*
- Domain Hash
- Last Update time
- Number of visits
- Number of different types of visits
- Source used to access site
- Google Adwords campaign name
- Access Method (organic, referral, cpc, email, direct)
- Keyword used to find site (non-SSL only)