Mag. iur. Dr. techn. Michael Sonntag

Windows Forensics

Institute for Information Processing and Technology (FIM)
Johannes Kepler University Linz, Austria

E-Mail: sonntag@fim.uni-linz.ac.at
http://www.fim.uni-linz.ac.at/staff/sonntag.htm

Agenda

- Recycle bin
- Print spool files
- Thumbs.db
- Prefetch
- Swap/Hibernation file
- The Windows Registry
  - → Recent files
  - → USB device history
  - → Registry traces
  - → Various elements
- Restore points

Windows forensic: Recycle bin

- When deleting files normally, they end up in the recycler
  - → Shift+DEL → Deleted immediately
    - » Special tools/file carving!
  - → Something in the recycler → Generally deleted intentionally
    - » These files are completely recoverable: Name, content etc.
- Emptying the recycle bin:
  - → The saved files are actually deleted; just like normal files
    - » Their directory entries within the recycler folder remain
    - » Their data remains on the disk until overwritten
  - → The INFO2 file (see later) is deleted and a new one created
    - » Sometimes only shortened, i.e. the record numbers continue
- Deleting a single file from the recycle bin
  - → Changes the first byte of the record in INFO2 file to '00'
    - » Typically drive letter; recreatable from drive number in record!
- Note: Removable media does not have a recycle bin!

Windows forensic: Recycle bin

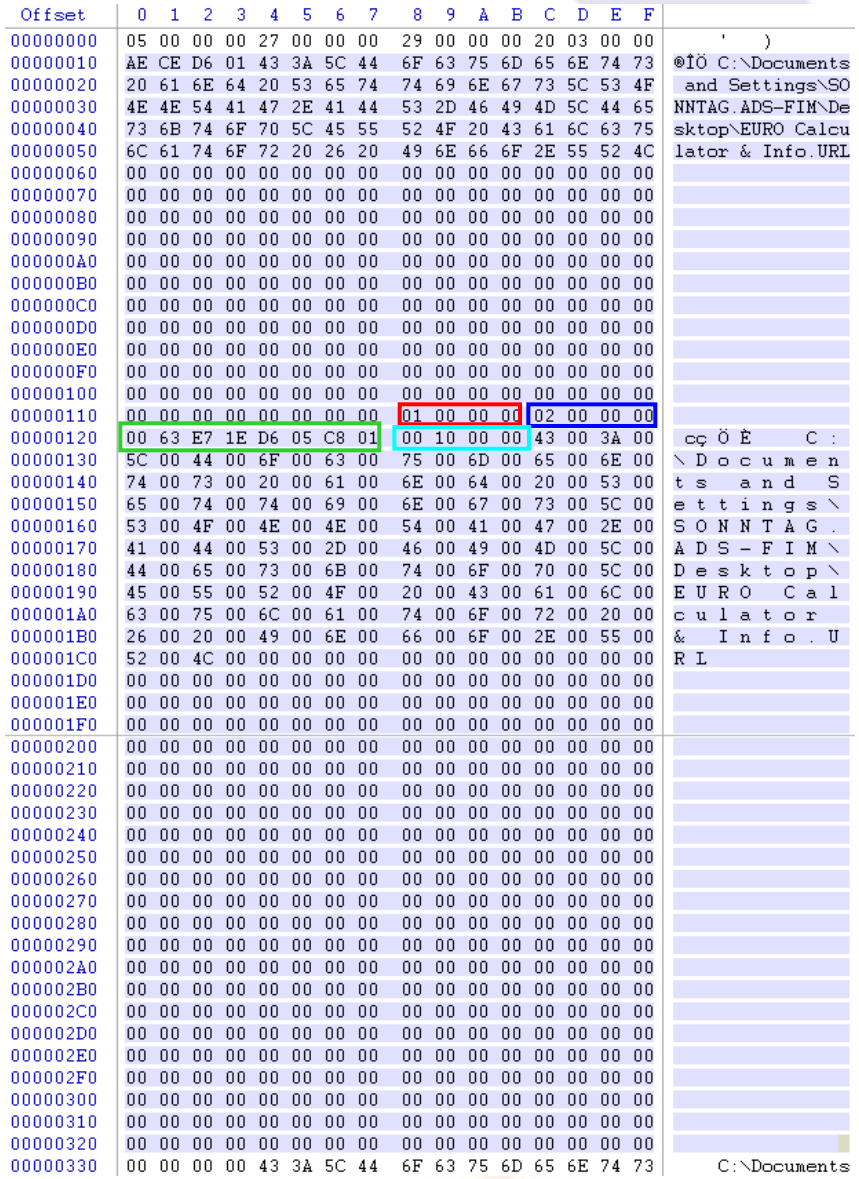
- Physical changes on deleting "into" the recycler:
  - → File entry deleted from original directory
    - » Remains there until overwritten!
  - → Modified/Last Access updated
  - → The long filename is deleted
  - → File entry created in recycler directory
    - » D<original drive letter><#>.<original extension>
      - – Dc1.txt: Second deleted file from drive C, had “txt” extension
      - – Note: In the Windows Explorer you always see only your own files and the filenames from the INFO file!
    - » Subdirectory: User-SID
  - → Information added to recycler index file ("INFO"/"INFO2" file)
    - » Includes deletion time, original location, recycle bin index
      - – Index allows discovery of deletion order!
    - » Attention: Windows Vista has replaced the INFO file with a separate file named similar as the one with the deleted data!

Windows forensic: Recycle bin

- The INFO2 file structure
  - → Binary file
  - → Contains the file name twice: ASCII and Unicode
  - → 20 Byte file header; Bytes 12-13 (-15?) are record size
    - » Record size is usually 2003 = 0x0320 = 800 Bytes
- Record structure
  - → 260 Bytes: Original file name (ASCII), including path
  - → 4 Bytes: Record number (starting at 0)
  - → 4 Bytes: Drive number (00 = A, 01 = B, 02 = C, …)
  - → 8 Bytes: Deletion time (FILETIME format, UTC)
  - → 4 Bytes: Physical file size (=Bytes on disk!)
    - » Therefore always multiples of cluster size
    - » Actual file size: See directory entry of the file itself
  - → 520 Bytes: Original file name (Unicode), including path

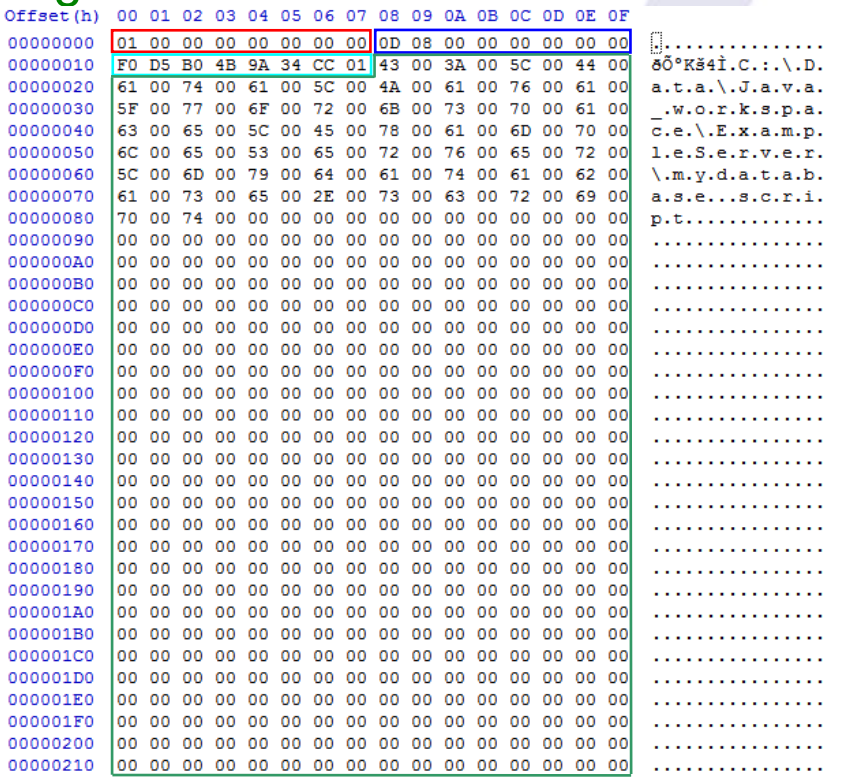
Windows forensic: Recycle bin

- Original filename: C:\Documents and Settings\SONNTAG.ADS-FIM\Desktop\EURO Calculator & Info.URL
- Record number: 1
- Drive number: 2 (= C:)
- Deletion time: 0063E71E:D605C801 (=1EE76300:01C805D6, =3.10.2007 15:57:37 UTC)
- Physical file size: 0x00100000 (=0x00001000, = 4096 Bytes



Windows forensic: Vista/7 - Recycle bin

- The directory is similar, but organisation within is different
  - → Each file is stored directly with a new filename:
    - » $R<six random characters>.<original extension>
  - → Information on “real” filename (and additional data):
    - » $I<six random characters>.<original extension>
- Additional data stored:
    - » File length: 544 Bytes
  - → Magic number
  - → Original file size
    - » Byte order is reversed!
  - → Date and time of deletion
    - » Windows 64 Bit hex value Little endian
  - → Original file name
    - » 520 Bytes = 260 characters
      - → Max. path length!



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	01	00	00	00	00	00	00	00	0D	08	00	00	00	00	00	00	.....
00000010	F0	D5	B0	4B	9A	34	CC	01	43	00	3A	00	5C	00	44	00	ðÕ°Kš4Ì.C.:.\.D.
00000020	61	00	74	00	61	00	5C	00	4A	00	61	00	76	00	61	00	a.t.a.\.J.a.v.a.
00000030	5F	00	77	00	6F	00	72	00	6B	00	73	00	70	00	61	00	_.w.o.r.k.s.p.a.
00000040	63	00	65	00	5C	00	45	00	78	00	61	00	6D	00	70	00	c.e.\.E.x.a.m.p.
00000050	6C	00	65	00	53	00	65	00	72	00	76	00	65	00	72	00	l.e.S.e.r.v.e.r.
00000060	5C	00	6D	00	79	00	64	00	61	00	74	00	61	00	62	00	\.m.y.d.a.t.a.b.
00000070	61	00	73	00	65	00	2E	00	73	00	63	00	72	00	69	00	a.s.e...s.c.r.i.
00000080	70	00	74	00	00	00	00	00	00	00	00	00	00	00	00	00	p.t.....
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Print spool files

- When printing documents, these are not immediately sent to the printer, but rather stored in a local file
  - → This is then sent to the printer
    - » And deleted after successful completion
  - → Attention: Users can configure within the printer properties that the data is sent immediately to the printer; this is rare!
  - → Note: This applies to local printers only!
    - » Network printers will have this file created on the print server
- Typical file formats for spooling are:
  - → RAW: Directly as the printer wants it, e.g. Postscript or some proprietary format
    - » Device dependent
    - » Can be re-printed simply by sending to a (similar!) printer again
  - → EMF: Enhanced Metafile Format (32 Bit version of WMF)
    - » Device independent
    - » Viewer programs available

Print spool files

- For each print job two files are created
  - → SHD: Job metadata (owner, printer, print method, …)
  - → SPL: Job data (RAW or EMF)
- Contents of the SHD file:
  - → Username, Name of user to notify
  - → Document name
  - → Printing time
    - » SYSTEMTIME structure (=UTC!)
  - → Page count
  - → Windows version
  - → Job ID
  - → Priority
  - → Printer name + driver + mode
  - → Printing processor + format
  - → Computer name

Thumbs.db

- Thumbs.db: Hidden file to store thumbnail images (previews) of the files in a folder
  - → But ONLY, if the folder was viewed in "Thumbnail view" at SOME time in the past when the file was already there
  - → Can be deactivated (Default: enabled) in Explorer properties
    - » "Do not cache thumbnails"
  - → Deleting images from the disk will not remove the thumbnail from Thumbs.db!
    - » They will never be removed!
      - – Only solution: Delete Thumbs.db file!
- File format: OLE2 Compound Document (MS Office)
- What is stored: JPEG, BMP, GIF, HTM
- Encrypted files will still have an unencrypted thumbnail!
  - » If viewed in thumbnail view when they were not yet encrypted …
  - → However, this security flaw was fixed at some time somehow

Thumbs.db

- Attention:
  - → Windows Vista does no longer have this file
  - → Win2K+NTFS: Thumbnails in ADS (FAT → Thumbs.db!)
- Before Windows XP: Contained also drive letter and path
  - → Windows ME, Win2K
- Take care when copying directories to a USB stick:
  - → When copying the directory, the Thumbs.db file is copied too
  - → When copying all files, it is not copied (unless shown anyway)
- Thumbs.db can be used to prove that images actually were on a certain computer: The Thumbs.db file is still there, and the files (including the same Thumbs.db) have been found somewhere else!

Thumbnails >=Vista

- Creates thumbnails for files on different media types
  - → Including removable and network drives
  - → Including files in encrypted containers, e.g. TrueCrypt
  - → But NOT for files encrypted with EFS, unless the thumbcache directory is also encrypted!
    - » Vista: Thumbnails are not deleted if encrypted by EFS later
- Central cache for all directories per user in his folder
  - → %USERPROFILE%\AppData\Local\Microsoft\Windows\Explorer
- Contains several files:
  - → Thumbcache_32{96,256,1024}.db
    - » Individual thumbnails in the various sizes (32-1024 pixel)
  - → Thumbcache_idx.db: Central index for thumbnails
    - » Required for finding the image in the cache files
  - → Thumbcache_sr.db: Unknown; constant content
- Win 7: Last-modified field for each thumbnail removed

Windows prefetch

- Frequently (or recently) used applications are logged in a special folder: Speed up their start by noting which sectors from the disk will be required directly upon start
  - → These are then swapped in immediately, even if not at the start of the executable file
- Stored in directory "C:\Windows\Prefetch"
  - → Naming: <Executable file name>-XXXXXXX.pf
    - » XXXXX: Hash of location from where it was run
  - → Count of executing the program: HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{GUID}\Count
    - » ROT-13 encoded!
    - » "Data": 5th byte -5 = Count of execution
  - → Maximum count (XP): 128 entries
  - → Contains also references to loaded modules

Windows prefetch

- When is it updated?
  - → XP: Boot time and application launch, 2003: Boot time (def.)
  - → Disabled for SSD disks by default
- Attention: Prefetch is system-wide
  - → You cannot tell from the file which user executed it
    - » But with MAC time this can be possible (if you know who was logged on at which time)
    - » See also the UserAssist registry entries (previous slide; HK**CU**!)
- Note the MAC times of the files:
  - → Created: Program was started for the first time
  - → Modified: Program was started for the "last" time
    - » Attention: Prefetch files will not be updated after some time
      - – Probably when windows decides it exactly knows what to do
  - → Accessed: Last run time of the program
    - » Attention: No longer updated by default on Vista or later!

Disabling prefetch: http://msdn2.microsoft.com/en-us/library/ms940847.aspx

Windows prefetch

- .pf file content
  - → Filename: Offset 0x10
  - → Timestamp: FILETIME at offset 0x78 (XP)/0x80 (>=Vista)
  - → Run count: DWORD at offset 0x90 (XP)/0x98 (>=Vista)
  - → Magic number at start: 0x11 (XP)/0x17 (>=Vista) 0x00 0x00 0x00 “SCCA”; has no end marking
  - → List of files accessed in the first 10 minutes
    - » Can be used to inspect suspected malware (what did they load) or media files launched through double-clicking them
    - » Can provide information on paths no longer existing
      - – Example: TrueCrypt volume, deleted executables etc.
- Layout.ini: List of files used on booting
  - → Used for arranging files during defragmentation
  - → Unwanted programs started on booting?

Swap/Paging file

- Contains pages from the memory
  - → Not necessarily in a "good" order!
  - → Data may remain there for a very long time as well
    - » If this sector happens to not being used
- Attention: Normal shutdown may delete, truncate, overwrite etc. the swap file!
  - → In important cases it is therefore necessary (after doing live analysis) to pull the plug, but not shutdown the system!
- Hidden file, C:\pagefile.sys
- Typical application for file carving: Assembling a file from numerous smaller parts
  - → Very difficult and unreliable, unless complete and in correct order (this is likely only for very small files)!
- Practical usage: Search for strings/regular expressions

Swap/Paging file

- Attention: Anything found in there is "suspect"!
  - → You don't know **when** this information was put in there
  - → You don't know **which user** was logged in at that time
  - → The data might already have been on the disk when the paging file was created
- The swap file need not be located in contiguous sectors
  - → There may be small "holes", which perhaps are not reused for a long time because they are so small
    - » Good location for finding “old” file fragments

Hibernation file

- Similar to the swap file: Contains memory pages
  - → But here it is a complete image of the total memory!
  - → May be smaller or larger than the swap file
- Can be used to recreate the last use of the computer
  - → Virtual machines come in handy for this
- Hidden file: C:\Hiberfil.sys
- Attention: The first block will always be overwritten with zeros after boot, so never wake up a hibernated computer without obtaining a forensic copy before!
  - → Rest of the file remains unchanged until the next hibernation!
  - → So the state at a previous point in time might be recreated partly (computer won't run; memory might be "salvaged")
- The hibernation file is compressed

The Windows Registry

- The Registry is similar to a file system:
  - → Hierarchical storage of data
  - → Keys = Directories
  - → Values = Files
    - » Name, Data and Type (BINARY, DWORD, SZ, MULTI_SZ, EXPAND_SZ)
- 5 root keys exist:
  - → HKLM: HKEY_LOCAL_MACHINE (Computer-specific data)
  - → HKU: HKEY_USERS (User-specific data)
  - → HKCR: HKEY_CLASSES_ROOT (application settings, file associations, class registrations for COM objects)
    - » Link to HKLM\Software\Classes
  - → HKCC: HKEY_CURRENT_CONFIG (Current hardware conf.)
    - » Link to HKLM\System\CurrentControlSet\Hardware Profiles\Current
  - → HKCU: HKEY_CURRENT_USER (Current user's data)
    - » Link to HKU\<SID of current user>

Physical storage of the registry

- The Registry is logically a "single file"
  - → But parts are stored in different files or created dynamically
- File locations:
  - → HKLM\SAM — %SYSTEMROOT%\System32\config\SAM
  - → HKLM\Security — %SYSTEMROOT%\System32\config\SECURITY
  - → HKLM\Software — %SYSTEMROOT%\System32\config\software
  - → HKLM\System — %SYSTEMROOT%\System32\config\system
  - → HKLM\Hardware — Stored in memory only – not on disk!
  - → HKU\.Default — %SYSTEMROOT%\System32\config\default
  - → HKU\SID — %USERPROFILE%\NTUSER.DAT
  - → HKU\SID_Classes — %USERPROFILE%\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat

User/… identifiers: SID

- SID = Security Identifier
  - → S-1-0-0 (Nobody): A group with no members
  - → S-1-1-0 (Everyone): A group that includes all users
  - → S-1-2-0 (Local): Users who logged on locally
  - → S-1-2-1 (Console Logon): Users on the phys. console
  - → S-1-3-0 (Creator Owner): The user who created a new object
  - → S-1-3-1 (Creator Group): The primary group of the user who created a new object
  - → S-1-5-2 (Logon Network): Users logging on via network
  - → S-1-5-7 (Anonymous): Anonymous logged on users
  - → S-1-5-18 (Local System): The OS itself
  - → S-1-5-19 (Local Service): Service account
  - → S-1-5-20 (Network Service): Service account
  - → S-1-5-21-?????-500: Administrator
  - → S-1-5-32-544 (Administrators): Group of all administrators

Installation dependent (unique!)

http://msdn.microsoft.com/en-us/library/windows/desktop/aa379649%28v=vs.85%29.aspx
http://support.microsoft.com/kb/243330/en-us

Obtaining access to an account

- No password for a Windows account? Login might still be possible!
- Passwords are stored as hashed values in the Registry (SAM)
- Procedure:
  - → Obtain Password Reset CD
    - » Obviously possible also manually as well, just very complex (find registry on disk, find location in registry)
  - → Boot from this CD
  - → Let it overwrite the hash in the registry with a known one
    - » Can be anything, e.g. an empty password
  - → Shutdown and reboot in Windows
  - → Enter the password and log in
- Drawback:
  - → Encrypted files will be “destroyed” by this!
    - » Why? EFS needs the password for decryption.
    - » Merely “being” the user is insufficient!
- Note: This technique is very general. It works in the same way for Linux or any other application storing the passwords as hashes where they are accessible (potentially only “offline” from a different OS!)

AutoRun

- AutoRun Programs
  - → Long list of locations in registry, e.g.
    - » HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
      - – /RunOnce
    - » HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run bzw. \RunOnce
    - » Same under HKCU!
    - » Explorer hooks, like HKLM and HKCU\Software\Classes*\SehllEx\ContextMenuHandlers
    - » Print monitors: HKLM\SYSTEM\CurrentControlSet\Print\Monitors
    - » Winlogon notifications
  - → Long list of locations outside the registry, e.g.
    - » “Startup” folder in start menu of user profile
    - » Scheduled tasks
  - → Problem: Things might be started from anywhere; no “authoritative list” from Microsoft available
- Useful tool: AutoRuns from Microsoft
  - → Will show all locations that are currently known

Installed software

- Software might be installed, although not visible as an icon on the desktop or in any start menu
  - → Registry keys are usually created during installation, but not always removed (although they should be) when the program is uninstalled
    - » HKLM\SOFTWARE\Microsoft\Windows\C.V.\App Paths
    - » HKLM\SOFTWARE\Microsoft\Windows\C.V.\Uninstall
  - → Separate registry keys for application settings might exist too
- Verification:
  - → Check for the actual executable at the contained path
  - → Check timestamp on registry key
- Cross-verification: Search for all executable files
  - → Note: Will not work for all kinds of plugins!

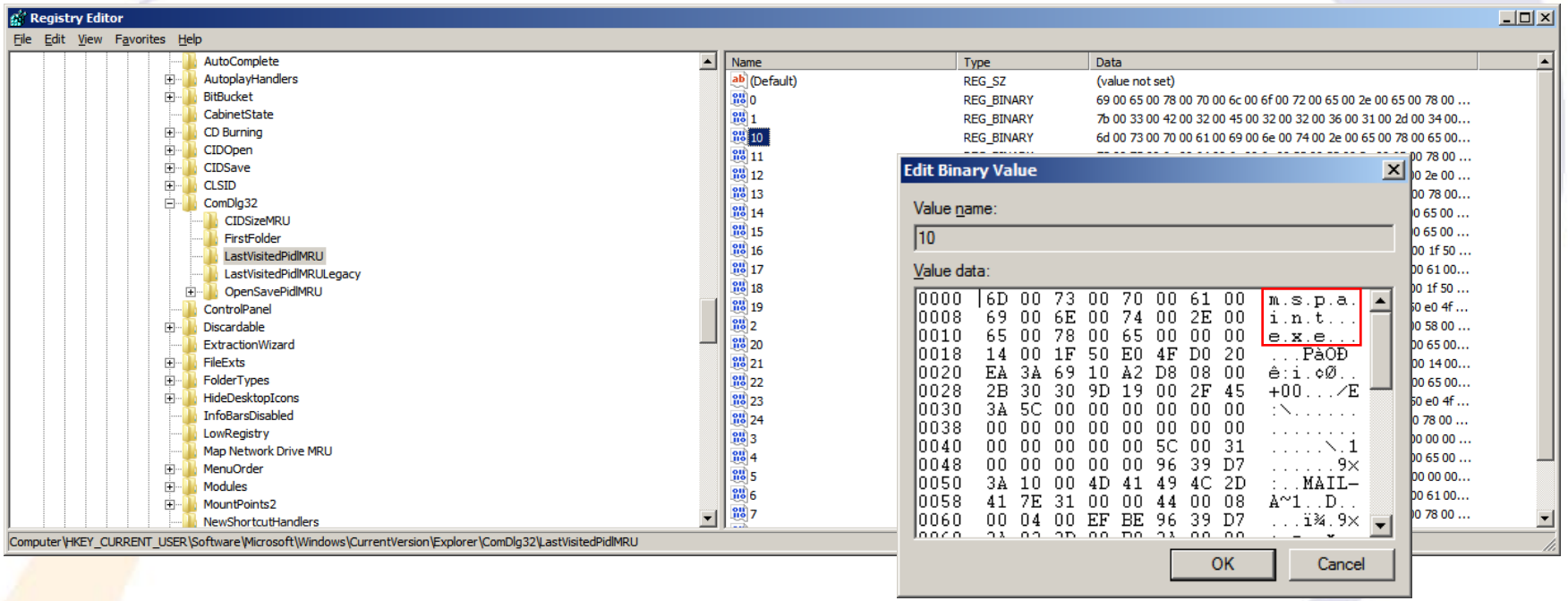
Windows forensic: Recent files MRU lists (Most Recently Used)

- These are usually stored within the registry
  - → Old windows versions: INI-files in windows/program directory
- Common lists include:
  - → Start menu: HKCU
    Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
    - » Equivalent of %USERPROFILE%\Recent
      ="My Recent Documents"
    - » Includes both local and network files!
  - → Run box: HKCU
    Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU
    - » In order of most recently added (not: Used)!
  - → Files (Common dialog box): HKCU\Software
    Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
    Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU
  - → Typed URLs in IE: HKCU
    Software\Microsoft\InternetExplorer\TypedURLs

Windows forensic: Recent files MRU lists (Most Recently Used)

- → Windows default search: HKCU Software\Microsoft\SearchAssistant\ACMru
  - » Subkey "5603": Search terms for folders and filenames
  - » Subkey "5604": Search terms for words/phrases within files
- Note: MRU list locations depend heavily on the
  - → Windows version
  - → Software version
  - → Installed patches
  - → Program configuration!
  - → Use software on a copy of the evidence when using the program to identify the registry keys modified
    - » E.g. SysInternals RegMon
- Typically key names for other software are:
  - → Settings, MRU, Recent, Opened, …

Last opened application



- HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidMRU (<Vista: LastVisitedMRU)
  - → Applications last used to access the files listed in OpenSavePidMRU (OpenSaveMRU)
  - → Contains path information as well

Windows forensic: App Compatibility

- Some applications might have problems running on newer version of Windows. These can be partially solved by replacing certain system functions by older versions (or in-memory patching, additional flags, …).
- This is a database on disk, but the registry contains a cache of a number of items from it
  - → Drawback: Only some elements; format changed several times: XP – Vista/2008 – 7/2008R2
- What is in there?
  - → Executables recently run
  - → Last modification time
  - → File was executed?
  - → File size
  - → Last time file was run

https://www.mandiant.com/blog/leveraging-application-compatibility-cache-forensic-investigations/

Windows forensic: MUI Cache

- When an application is started, the shall stores the application name in the registry
  - → Content: “FileDescription” from resource of executable
    - » Unfortunately nothing else
    - » Note: All are values in a single (few) keys → Timestamp useless
  - → XP/2000/2003:
    - » HKCU\Software\Microsoft\Windows\ShellNoRoam\MUICache
  - → Vista/7/2008
    - » HKCU\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache
    - » HKCU\Software\Classes\Local Settings\MuiCache**

https://www.mandiant.com/blog/execute/

Windows forensic: File execution hints

- Other caches whether files were executed:
  - → HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage\ProgramsCache
  - → HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage2\ProgramsCache
  - → HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage2\ProgramsCacheTBP
  - → HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage2\ProgramsCacheSMP

https://tzworks.net/prototype_page.php?proto_id=19

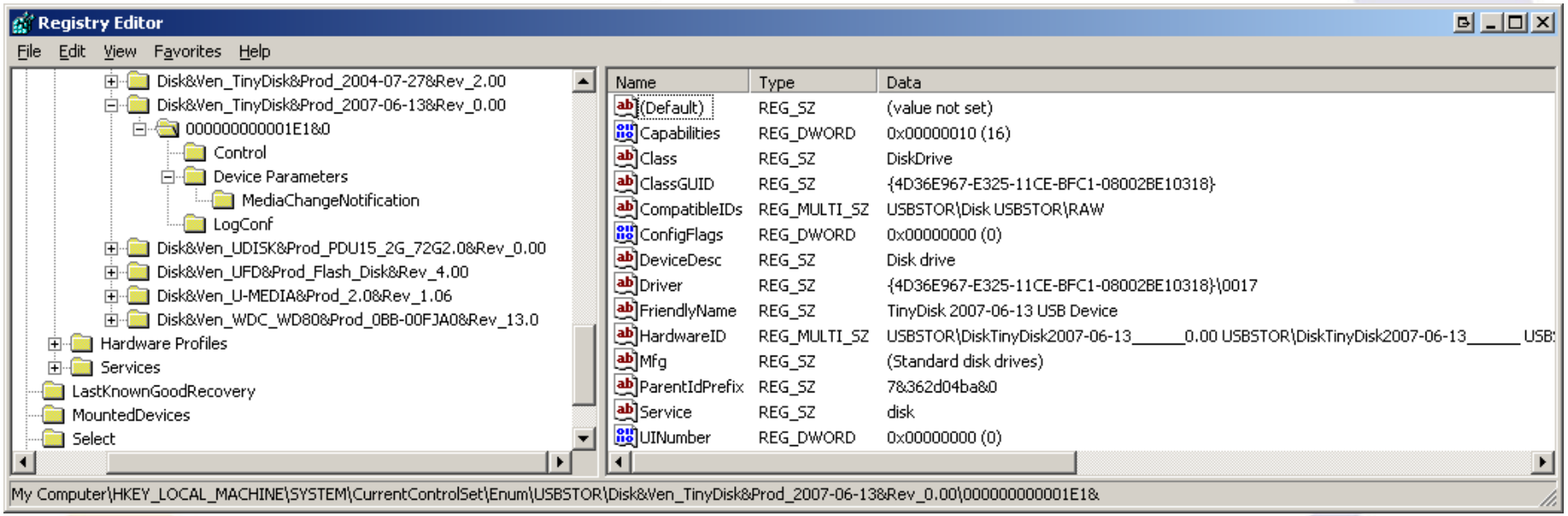
Windows forensic: Recent files

- Entries in "Recent", "Send to", "Start Menu", …
  - → Can prove the existence of files now deleted
    - » Note: Usually not produced on creating or copying files, but only on opening them!
  - → Especially useful for removable media, e.g. USB sticks
- Saved on the disk as ".LNK" files, i.e. shortcuts
- These contain a lot of information:
  - → File location
  - → Type of disk (hard disk, removable media, CD, …)
  - → File attributes, length, …
  - → Icon information
  - → MAC times
- May contain additional data
  - → Working directory, shell item ID, description, command line arguments, custom icon etc.

USB device history

- When a USB device is connected to a computer, this is "logged" within the registry
  - → I.e., configured and appropriate driver, if necessary, loaded
  - → This information remains when the device is disconnected!
- Note: Most USB storage device have unique serial numbers
  - → This means, the exact item can be recognized!
    - » Software: UVCView from Microsoft
- Registry key: HKLM\System\ControlSet00?\Enum\USBSTOR
  - → Subkey: Vendor, Producer and Revision
  - → Sub-Subkey: Serial number (if existing; else generated)
  - → ParentIdPrefix: Corresponds to HKLM\System\MountedDevices
    - » Binary value!
- In C:\Windows\setupapi.log the first installation is logged
- See also software: USBDeview!

USB device history



- Vendor: TinyDisk (Case label: "extreMEmory USB 2.0 4GB")
- Product: 2007-06-13, Rev. 0.00
- Serial number: 000000000001E1
- ParentIdPrefix: 7&362d04ba&0
  - → HKLM\System\MountedDevices: "\DosDevice\G:" \??\STORAGE#RemovableMedia#7&362d04ba&0&RM#{53f5630d-b6bf-11do-94f2-00a0c91efb8b}

RegRipper

- A tool to “rip” the registry
  - → Attention: Will not let you view open registry hives!
- Actual use: Collection of registry keys with interesting values
  - → Very large collection, and provides explanations too
- Example (“listsoft” plugin only):
  - → listsoft v.20080324
  - → (NTUSER.DAT) Lists contents of user's Software key
  - → listsoft v.20080324
  - → List the contents of the Software key in the NTUSER.DAT hive
  - → file, in order by LastWrite time.
  - → Tue May 14 14:55:56 2013Z Cygwin
  - → Thu Dec 20 16:35:24 2012Z X-Ways AG
  - → Mon Dec 17 16:35:28 2012Z Hewlett-Packard
  - → Mon Dec 17 16:26:05 2012Z Microsoft
  - → Mon Dec 17 16:06:55 2012Z AMD
  - → Mon Dec 17 16:06:55 2012Z Wow6432Node
  - → Mon Dec 17 16:06:51 2012Z ATI
  - → Mon Dec 17 15:13:03 2012Z JavaSoft
  - → Mon Dec 17 13:03:18 2012Z Realtek

Source: http://code.google.com/p/regripper/
Organizing the process: AutoRip (plugins are grouped into collections)

USB device history: setupapi.log

[**2007/07/20 11:42:13 840.8 Driver Install**]
#-019 Searching for hardware ID(s): usbstor\disktinydisk2007-06-13_____0.00,usbstor\disktinydisk2007-06-13_____,usbstor\disktinydisk,usbstor\tinydisk2007-06-13_____0,tinydisk2007-06-13_____0,usbstor\gendisk,gendisk
#-018 Searching for compatible ID(s): usbstor\disk,usbstor\raw
#-198 Command line processed: C:\WINDOWS\system32\services.exe
#I022 Found "GenDisk" in C:\WINDOWS\inf\disk.inf; Device: "Disk drive"; Driver: "Disk drive"; Provider: "Microsoft"; Mfg: "(Standard disk drives)"; Section name: "disk_install".
#I023 Actual install section: [disk_install.NT]. Rank: 0x00000006. Effective driver date: 07/01/2001.
#-166 Device install function: DIF_SELECTBESTCOMPATDRV.
#I063 Selected driver installs from section [disk_install] in "c:\windows\inf\disk.inf".
#I320 Class GUID of device remains: {4D36E967-E325-11CE-BFC1-08002BE10318}.
#I060 Set selected driver.
#I058 Selected best compatible driver.
#-166 Device install function: DIF_INSTALLDEVICEFILES.
#I124 Doing copy-only install of "USBSTOR\DISK&VEN_TINYDISK&PROD_2007-06-13&REV_0.00\000000000001E1&0".
#-166 Device install function: DIF_REGISTER_COINSTALLERS.
#I056 Coinstallers registered.
#-166 Device install function: DIF_INSTALLINTERFACES.
#-011 Installing section [disk_install.NT.Interfaces] from "c:\windows\inf\disk.inf".
#I054 Interfaces installed.
#-166 Device install function: DIF_INSTALLDEVICE.
#I123 Doing full install of "USBSTOR\DISK&VEN_TINYDISK&PROD_2007-06-13&REV_0.00\000000000001E1&0".
#I121 Device install of "USBSTOR\DISK&VEN_TINYDISK&PROD_2007-06-13&REV_0.00\000000000001E1&0" finished successfully.

USB device history

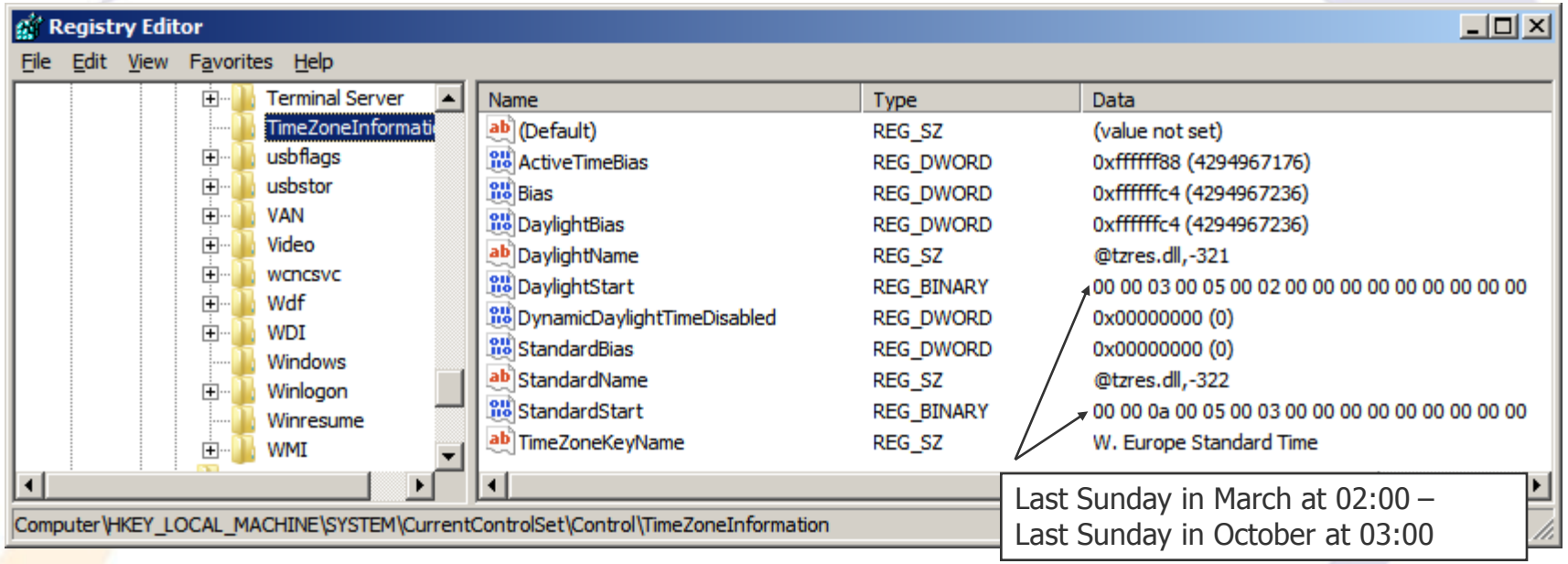
- Last connection: HKLM\System\ControlSet00?\Control\DeviceClasses
  - → Subkey "{53f56307-b6bf-11d0-94f2-00a0c91efb8b}": Disks
    - » Contains a subkey with the serial number included
  - → Subkey "{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}": Volumes
    - » Contains a subkey with the ParentIdPrefix included
  - → The LastWrite Time of these keys is the date and time the device was last connected to the computer
- Accessing the LastWrite time:
  - → Special tools, or
  - → Regedt32: Export as text and open in Notepad
  - → Key Name: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_TinyDisk&Prod_2007-06-13&Rev_0.00#000000000001E1&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
  - → Class Name: <NO CLASS>
  - → Last Write Time: **21.12.2007 - 09:12**
  - → Value 0

Registry: Owner information

- Owner/Organization: HKLM\Software\Microsoft\Windows NT\CurrentVersion
  - → RegisteredOwner: Owner name
  - → RegisteredOrganization: Organization name
  - → ProductId: Product ID
  - → DigitalProductId: Contains encr. license key (Bytes 52-66)
  - → InstallDate: Installation date (UNIX timestamp)
  - → SystemRoot: Windows installation directory
- Last user:HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
  - → DefaultUserName: Last logged in user
    - » When? → Timestamp of key!
  - → DefaultDomainName: Last domain logged into
  - → DontDisplayLastUserName: Don't store information above

http://www.dagondesign.com/articles/windows-xp-product-key-recovery/
http://geekswithblogs.net/willemf/archive/2006/04/23/76125.aspx

Registry: Timezone information



- Data from HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones is copied here
  - → ActiveTimeBias: Current Δ (0xffffff88 = -120 = 2 h E of GMT)
  - → DaylightBias: Delta during DST (= -60 = 1 h East of GMT)
  - → Daylight-/StandardStart: Should be SYSTEMTIME structure
    - » Year, Month (1=Jan), Day of week (0=Sun), Week (5=last), Hour, Min, Sec, ms (two bytes each); (doesn't match above!)

http://msdn.microsoft.com/en-us/library/ms725481%28v=vs.85%29.aspx
http://msdn.microsoft.com/en-us/library/ms724950%28v=vs.85%29.aspx

Registry: Network shares

- "My Network Shares": List of shares within the LAN HKCU\Software\Microsoft\Windows\Current Version\Explorer\ComputerDescriptions
  - → Allows reconstruction of a past view, i.e. what shares were accessed by the user
  - → Contains computers, shares (directories) and printers
  - → Value: Name = server/share/printer; Content = Description

Registry: WLANs

- When accessing a WLAN, its SSID is stored:
  HKLM\Software\Microsoft\WZCSVC\Parameters\Interfaces
  - → Subkeys look like GUIDs with values for "ActiveSettings", "Static#000?", …
  - → The values for "#Static000?" contain the SSIDs at offset 0x14
- IP address information for this connection (last only):
  HKLM\System\ControlSet00?\Services\Tcpip\Parameters\Interfaces
  - → Look for the same "GUID" key as of the WLAN!
  - → Dhcp*: Data on DHCP server, assigned address, netmask, default gateway, domain, nameservers, …
  - → LeaseObtainedTime/-TerminatesTime: Unix 23 Bit Timestamp
    - » When the Address was received and what is the definite last time it could have been used (but not: **was** used!)

Registry: Various

- Every single registry key has a "LastWrite" time value
  - → Does **NOT** exist for registry values!
  - → Format: FILETIME
  - → Updated, when a registry value directly within this key is created, modified, or deleted
- Win2K and WinXP registry editors are flawed
  - » Regedit.exe, regedt32.exe
  - → Registry values with a name length of 256-259 characters are not shown
    - » Values afterwards are suppressed as well
    - » Subkeys are inaccessible too
  - → But "reg.exe", the console registry tool, can show (and manipulate) such values!

http://search.cpan.org/~adamk/Win32-TieRegistry-0.25/TieRegistry.pm

Registry: Various

- Thinking about shutting down a suspect's computer?
  - → First check: HKLM System\CurrentControlSet\Control\SessionManager\MemoryManagement\ClearPageFileAtShutdown
    - » Value "1": Paging file NOT deleted, but **overwritten** with zeros
- HKLM\System\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate
  - → Disables writing “last access timestamp” on the file system
    - » Will **still** be changed on file modifications (i.e. writing to it), but not on reading, accessing properties etc.!
- HKEY_CURRENT_USER\Software\Microsoft\Protected Storage System Provider
  - → Contains IE auto-complete passwords (encrypted)
  - → Content not directly accessible, not even with regedt32
  - → But can be read in live systems or by parsing the registry file

Registry: Substitute executed programs

- Allows hiding what programs were ACTUALLY executed!
- HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
  - → Create subkey with name of executable, e.g. "calc.exe"
  - → Create string with name "Debugger" and value of alternate program, e.g. "C:\Windows\notepad.exe"
  - → Typing "calc.exe" will then start the text editor
    - » With "calc.exe" as parameter
      - – or whatever the "original" file was, e.g. a .lnk file!
  - → Shows only the "original" program in history lists!
- HKCR\{exe|com|bat}file\shell\open\command
  - → Default value contains command to execute ".exe" files
    - » Similar for the explorer context menu: HKCR\Drive\shell and HKCR\Folder\shell

Windows restore points

- Stored under C:\System Volume Information
  - → Not accessible through Explorer, but forensic tools!
    - » Not even for the administrator!
      - – Deleting a single one is very hard (all/all but last is easy!)!
- By default created every 24 hours and retained for 90 days
  - » Registry: HKLM\Software\Microsoft\WindowsNT\CurrentVersion\SystemRestore
  - → Also: Manually, before Windows update, SW installation, …
  - → When reverting to a previous point, another restore point is created (to allow going "forward“ again)
    - » Restoration is recorded in the system event log (Event ID 110)!
  - → Disabling them is possible
    - » Less than 200MB free on system drive → automatically disabled!
- Content:
  - → Files to restore (see change.log for original path/filename)
  - → Subfolder "snapshot": Registry files (SAM, Security, Software, System, .Default, NTUSER files, USRCLASS files)

Event log

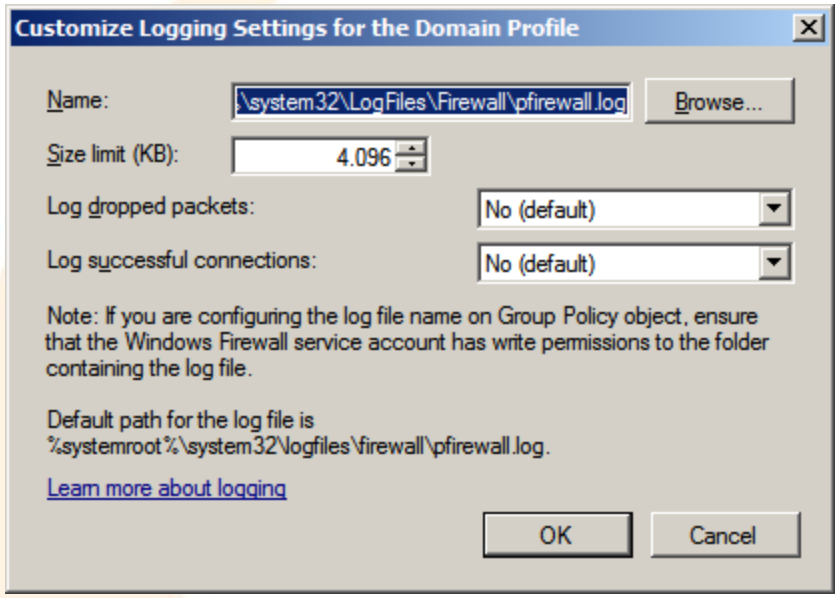
- Event logs are normal files
  - → Location: Specified in registry (→ to allow relocation)
    - » E.g. security: %SYSTEMROOT%\System32\config\SecEvent.Evt
  - → Can be recovered through file carving etc.
  - → Fixed size; configurable per log (typ. 512 kB – 16 MB)
- Problem: Security logging is off by default
  - → Application and System are on by default
- Format is not officially documented
- Clean shutdown: Offsets of oldest & newest entries written to header and a “clean” flag is set (“dirty” when running)
  - → While in use, trailer (after current last entry) has data
  - → Common source of corruption in forensics (pulling plug!)
- >=Vista: More kinds of logs (Setup, Administrative tasks, …)
  - → File format changed (*.evt → *.evtx): Binary → XML

Event log

- Content:
  - → Each Event has an EventId, specifying its type
    - » Examples: 528, 540 = Logon, 538 = Logoff
    - » Contains additional details
      - – These change with Windows versions → Use same or newer!
  - → Timestamp, …
  - → Record number: Used consecutively (→ no record deletion!)
- Inspection possible through windows
  - → Or LogParser: Allows SQL-Queries against the log!
    - » And against various other file formats as well
- Reliability: Quite good
  - → Modifications are difficult, but not impossible
  - → Deleting some elements is very hard
  - → But: Deleting the whole log is quite trivial

Internet Connection Firewall: Logs

- Location: %SYSTEMROOT%\system32\LogFiles\Firewall\pfirewall.log
- Logging is turned off by default
  - → Activating: Windows Firewall – Advanced Settings – Windows Firewall Properties – Logging



- Separate: Event log
  - → Firewall rules and connection security rules changes
  - → I.e., configuration, but not the individual "problems"!

Internet Connection Firewall: Logs

- Log file format: See file header!
  - → #Version: 1.5
    #Software: Microsoft Windows Firewall
    #Time Format: Local
    #Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsyn tcpack tcpwin icmptype icmpcode info path
- Example:
  - → 2012-08-23 11:11:09 ALLOW TCP 140.78.100.211 140.78.3.160 1735 80 0 - 0 0 0 - - - SEND
    - » Outgoing web request to JKU webserver
  - → 2012-08-23 11:11:19 ALLOW UDP fe80::a400:fe81:4022:2a12 ff02::1:2 546 547 0 - - - - - - - - SEND
    - » Outgoing DHCPv6 request ("Solicit")
  - → 2012-08-23 11:20:23 ALLOW ICMP 140.78.100.164 140.78.100.211 - - 0 - - - - 8 0 – RECEIVE
    - » Incoming ping request

Conclusions

- Especially the Registry is a treasure trove of information
  - → But the data is very well hidden under obscure names
  - → Special attention needs to be paid to how reliable the data is
    - » Windows version, source of description of keys/values etc.!
    - » Try it out on a copy and monitor the changes made
      - – E.g. using registry-diff software, sandboxing software
  - → Restore points contain previous versions of the registry
- Remnants of activity may remain on the computer for a long time: Recycle bin records, print spool files, event log, …
  - → Sometimes even forever (Thumbs.db) unless expl. removed!
- Restricting the investigation is therefore very desirable
  - → Only a subset of data need then be searched for/through!

Questions?

Thank you for your attention!

Links

- Windows Recycle Bin:
  http://www.foundstone.com/us/resources/proddesc/rifiuti.htm
- MRU locations:
  http://windowsxp.mvps.org/RegistryMRU.htm
- Windows Spool Files
  http://www.undocprint.org/winspool/spool_files
- Hurlbut, D.: Thumbs DB File Forensic Issues
  http://www.accessdata.com/media/en_US/print/papers/wp.Thumbs_DB_Files.en_us.pdf
- USBDeview
  http://www.nirsoft.net/utils/usb_devices_view.html
- System Restore Point Log Decoding
  http://www.ediscovery.co.nz/wip/srp.html

Links

- Murphey, Rick: Automated Windows event log forensics: http://www.dfrws.org/2007/proceedings/p92-murphey.pdf
- LogParser: http://www.microsoft.com/germany/technet/datenbank/articles/600371.mspx
- Rob Faber: Windows log forensics: http://www.net-security.org/dl/insecure/INSECURE-Mag-16.pdf page 86