**Windows Forensics**

When performing a forensic analysis on a windows machine, various artifacts should be observed post intrusion. When performing registry forensics, a free software can be used called Registry Explorer. [RegRipper](https://www.sans.org/blog/regripper-ripping-registries-with-ease/) can also be added to your forensic workstation to extract registry keys

**Some important acronyms\information to know:**

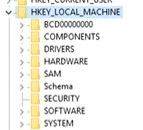
HKCU – HKEY\_Current\_User

HKLM – HKEY\_Local\_Machine

MRU – Most Recently Used

Windows Registry is located in C:\Windows\System32\config

Default, SAM, SECURITY, SOFTWARE, and SYSTEM are the most important registry hives to obtain when performing a forensic analysis of a windows system. A copy of these files can be found in C:\Windows\System32\config\RegBack. We can open these registry files in Registry Explorer later for analysis.

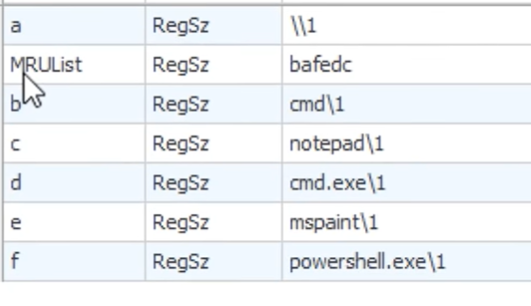


**NTUSER.DAT**

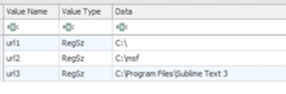
Every user profile (e.g. C:\users\Dennis) has an NTUSER.DAT file (invisible file), which plugs into the register as HKCU. We want to grab this file as well, because HKCU is being read from NTUSER.DAT. Typically when we talk about HKCU we are talking about analysis of a live system, and when we’re talking about NTUSER.DAT we are talking about analysis of a dead system.

Open NTUSER.DAT and navigate to HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer

* \ComDig32
  + \LastVisitedPidlMRU – Binaries that have been used to open/save files
  + \OpenSavedPidlMRU – The last file path you were browsing (e.g. saving a file to a location in MS Word)
    - These registries can provide information about files that have been opened or browsed, and then deleted in the system.
* \RecentDocs – can show recent things that have been interacted with in the system (e.g. open or saved).
* \RunMRU – Most recently run from run.exe, ran in the order of MRUList (in the example below: cmd\1, [\\1](about:blank), powershell.exe\1, ect….



* \TypedPaths – Paths ran in windows explorer. Can be used as artifact for recently deleted files.



* \UserAssist – Shows us evidence of an application execution that utilizes a GUI. Values shown in this key are ROT13 encoded

**Run and Run Once**

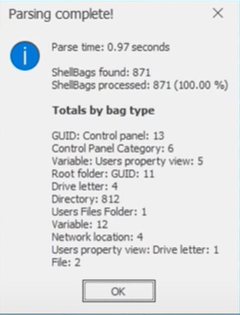
HKCU and HKLM – Navigate back to HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\

* Run – Programs that are specified to start upon logging into Windows.
* Run Once - Programs that are specified to start upon logging into Windows.

**Shellbags**

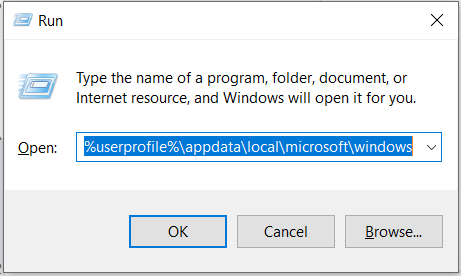
When you open a file in Windows explorer and you customize the icons, change the sorting methods, or modify the general look and feel of the window, this information is stored in [ShellBags](https://www.sans.org/blog/computer-forensic-artifacts-windows-7-shellbags/). ShellBags persist for objects that have not long existed on the device. This can give the investigator insight into deleted files and folders.

Under HKCU\SOFTWARE\Microsoft\Windows\Shell – use a software called [ShellBags Explorer](https://ericzimmerman.github.io/#!index.md) to parse this key.



**UsrClass.dat**

UsrClass.dat was introduced in Windows 7 and provides segmentation from low integrity processes that do not have access to read and write to the registry. UsrClass.dat records configuration information from user processes that do not have access to write to the registry hives. In order to get all Shellbags information, we need to parse both NTUSER.dat and UsrClass.dat. These registry hives are found in %user profile% and %user profile%\AppData\Local\Microsoft\Windows respectively. A tool called [sbag](https://tzworks.com/prototype_page.php?proto_id=14) from TZWorks can be used to parse shellbags from NTUSER.dat and UsrClass.dat.



**USB Enumeration**

The two keys below are used to observe USB information.

HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR

HKLM\SYSTEM\CurrentControlSet\Enum\USB

In a forensically obtained image, \CurrentControlSet\ might not exist. In that case we will observe ControlSet00X, where X being a variable derived from the data in key HKLM\SYSTEM\Select\Current



In this case we will observe HKLM\SYSTEM\ControlSet001\Enum\USBSTOR. This hive will show us the USB, it’s serial, and it’s last write time in UTC. If the & is towards the end of the serial, we know this is a globally unique serial. Otherwise, the manufacturer did not follow microsoft guidelines, and the value is not unique.

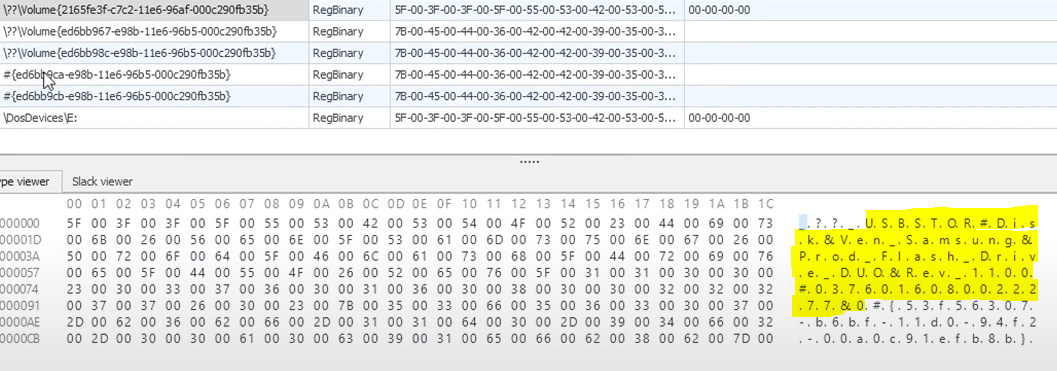


In HKLM\SYSTEM\ControlSet001\Enum\USB we can see the VID & PID information, which we can lookup in well known databases and find more information on the make and model of the flashdrive.

Load the SOFTWARE hive and navigate to HKLM\SOFTWARE\Windows Portable Devices\Devices. Here we can get information about the name of the Volume of USB drives. In this case, the volume name was Sticky.

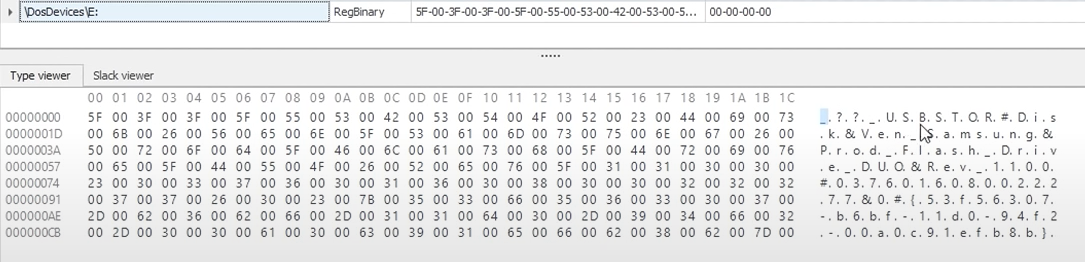


Load the SYSTEM hive and navigate to HKLM\SYSTEM\MountedDevices, and match the USB serial to the MountedDevices key values to find the volume GUID. The volume GUID (globally unique identifier, specific per system) is important because any reference to the GUID will let us know that that specific USB was being used.



216fe3f-c7c2-11e6-96af-000c290fb35b in this case is the Volume GUID

We can also use the serial number to obtain the drive letter that was assigned to the device.

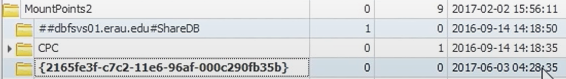


We now know the make and model, the VID and PID, the serial number, the volume name, and the first and last mount time of the USB.

Using the Volume GUID found in SYSTEM\MountedDevice, we can find the user that mounted the USB Device - load the NTUSER.DAT hive and navigate to

NTUSERS.DAT\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Mountpoints2

The last write timestamp will show what time the device was mounted, and we know which user mounted the device because we are analyzing that user’s NTUSER.DAT file.



**Timezone Information -**

Reload the SYSTEM hive and navigate to SYSTEM\ControlSet001\Control\TimeZoneInformation and obtain the time zone. That way we can ensure even correlation is accurate in terms of timezones.



**Computer Name -**

Navigate to SYSTEM\ControlSet001\Control\ComputerName\ComputerName and obtain the computer name key -



**File System -**

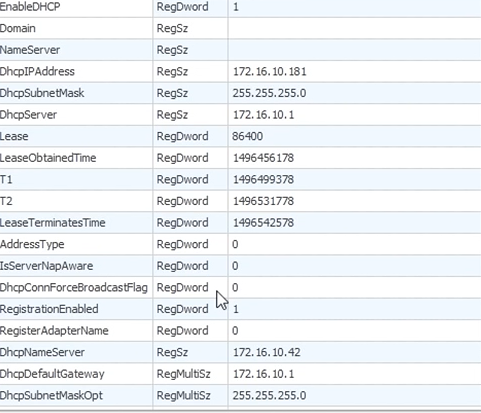
Navigate to SYSTEM\ControlSet001\Control\FileSystem and grab the NtfsDisableLastAccessUpdate and take note if the Data is 1 or a 0. If the value is 1, that means that accessing a file will not update the file access timestamp.

**Shares Drives -**

Navigate to SYSTEM\ControlSet001\Services\LanmanServer and grab any shared drives that are configured.

**Services -**

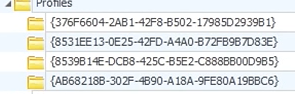
Navigate to SYSTEM\ControlSet001\Services\Tcpip. This directory will give us different network information. Drilling down to SYSTEM\ControlSet001\Services\Tcpip\Interfaces can tell us the DHCP IP address, subnet mask, as well as DHCP server, lease time, and other information:



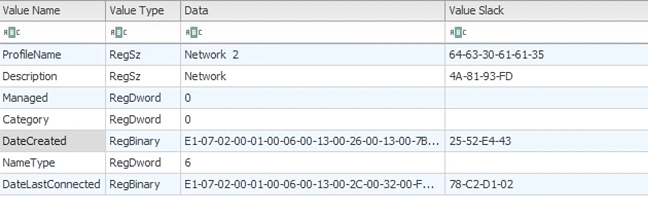
**Network Location Awareness (NLA) -**

NLA aggregates the network information for a PC and generates a GUID to identify each network. The Windows Firewall will use this information to apply firewall rules to the appropriate profile. Using the NLA registry, you can find evidence of every network a machine has connected to.

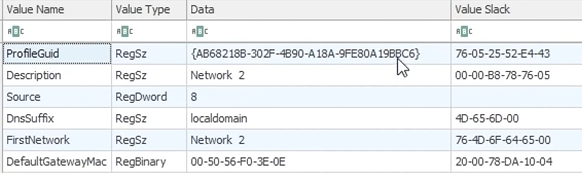
Load the SOFTWARE hive and navigate to HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles to obtain a GUID for each network.



In the Profiles sub directory you can get information about if the network was wireless or wired from the NameType key (6 for wired network, 47 for wireless network, and 17 for broadband network). Finally, you can see the DateCreated and DateLastConnected. This time is recorded in Windows 128 bit System Structure, UTC time. You will need to use [DCode](https://www.digital-detective.net/dcode/) to decode this time.

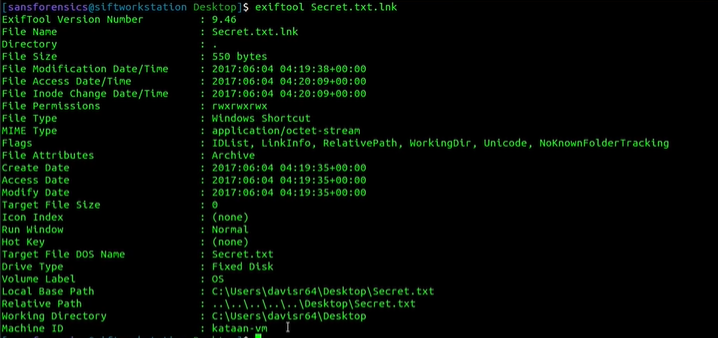


Under HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Unmanaged you can see information like the Default Gateway MAC, which you can then use to see the OUI to determine the manufacturer of the Gateway (first 3 octets or first 6 number/letter combination). You can also obtain the DnsSuffix which can tell you information about the domain, as well as the SSID of the network from the FirstNetwork key.



**Link File Analysis -**

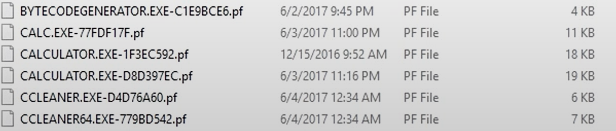
.lnk files contain a [wealth of metadata](http://forensicswiki.org/wiki/LNK). If carving for .lnk files, the file signature is 0x4C (4C 00 00 00). By using tools like [exiftool](https://exiftool.org/), you can obtain metadata that .lnk file includes e.g. the MAC times (modified, accessed, created), size of the file when it was last accessed, serial number of volume where it was stored, network volume share name, and sometimes the MAC address of the host computer. This information can help in proving that a file in question once existed on a device.



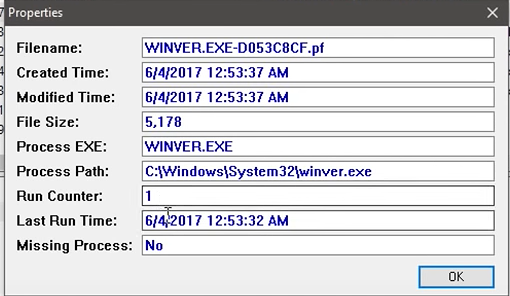
**Prefetch and SuperFetch**

Prefetch and SuperFetch attempts to make the Windows user experience better by making sure often-accessed data can be read from RAM instead of HDD. This can speed up boot times and shorten the amount of time to start a program. By observing the Prefetch folder, C:\Windows\PreFetch we can obtain evidence on global application execution, but GUI and apps run from CMD.

When observing files in the PreFetch folder, the first portion of the file is the application name, followed by a hash of the file’s location.



These PreFetch files (.pf) can be analyzed using a tool called [WinPrefetchView](https://www.nirsoft.net/utils/win_prefetch_view.html).



If you do not have a Prefetch folder or there is nothing in the Prefetch folder, load the SYSTEM hive and navigating to HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters, and observe the EnablePrefetcher Key. If the Data key is not set to 3, then Prefetch is not turned on (not on by default in Windows Server).