Digital Investigation (2005) 2, 201–205

ELSEVIER

Digital Investigation

www.elsevier.com/locate/diin

The Windows Registry as a forensic resource

Harlan Carvey

KEYWORDS
Windows;
Registry;
Investigation;
Forensics;
Analysis

Abstract The Windows Registry contains a wealth of information that can prove to be very valuable to the forensic investigator. The key to accessing this information is to know where the information exists within not only the file system, but also within the structure of the Registry itself.

Purpose

The purpose of this article is to briefly discuss the structure of the Windows Registry, and then delve into examples of the types of information that can be found in the Registry. Finally, methods for determining Registry ''footprints'' for arbitrary applications and user activity will be presented.

Background

Forensic investigators may use data reduction techniques, such as comparing hashes of ''known-good'' or ''known-bad'' files to the files located on the image they're examining, particularly when dealing with Windows systems. However, analysis of a Windows system can go much deeper than an examination of the file system alone. The Windows Registry provides a wealth of information that the investigator can use to make his case. Some of the information within the Registry can be correlated with the data from within the file system to produce an even more comprehensive picture of activity for the investigator.

The structure of the Registry

When most people interface with the Windows Registry, they do so with a GUI tool, the most common of which is the Registry Editor. In some very few cases, queries of or modifications to the Registry will be performed using command line interface (CLI) tools, or through a Group Policy Object. However, few are really aware of the structure of the Registry itself.

The Windows Registry[1] is a hierarchal database used to store information about the system; its configuration, devices attached to the system, and applications and users on that system. The Registry takes the place of the configuration files (config.sys, autoexec.bat, win.ini, system.ini) that most of us are familiar with from MS-DOS and older versions of Windows.

When a Windows system is running, we can see the Registry as one unified ''file system'' via the Registry Editor. Opening the Registry Editor, you see a tree view of a series of folders within the left-hand pane, as illustrated in Fig. 1. These folders are referred to as ''hives'', and hives are

[1] http://support.microsoft.com/kb/256986/EN-US/.

1742-2876/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2005.07.003

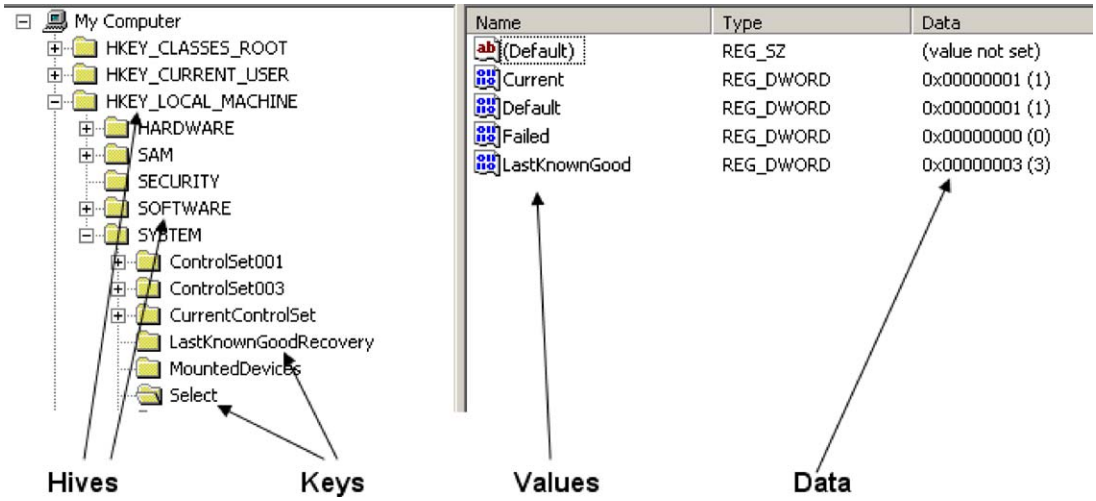

Figure 1 Structure of the Windows Registry.

made up of keys, which contain values and subkeys. Values are the names of certain items within a key, and each value has specific values associated with them that means something specific to the operating system, or to the application that depends upon the particular value.

While this may appear similar to a file system as shown in the Registry Editor, the Registry is actually maintained within several files on the system, and in fact, some of what you see in the Registry Editor is volatile, and doesn't exist until the system is powered on and booted. The various hives or sections of the Registry that are persistent on the system can be found in files located in the %SYSTEMROOT%\system32\config folder. For example, the HKEY_LOCAL_MACHINE\System hive, seen in Fig. 1, resides in the file named ''System''. The HKEY_LOCAL_MACHINE\Software hive resides in ''Software''.

There is an exception, however. The file that comprises the configuration settings for a specific user is found in that user's ''Documents and Settings'' folder. The file is named ''NTUSER.DAT'', and contains Registry settings specific to that user.

The Registry as a log file

In addition to the various keys, values, and data that can be retrieved, Registry keys have a value associated with them called the ''LastWrite'' time, which is similar to the last modification time of a file. This value is maintained as a FILETIME[2] object and indicates when the Registry key was last modified. A FILETIME object records the number of 100 nanosecond intervals that have passed since midnight, 1 January 1601.

With files, it is often difficult to determine what changes were made when the file was last modified. The forensic analyst may have a copy of the file, and the last modification time, but may not be able to determine what was changed in the file. The same holds true for Registry keys that contain several values, as it is often difficult to determine which value was changed. As we'll see, however, that this is not the case with most recently used (MRU) lists maintained in the Registry. In general, though, the forensic analyst may need to correlate the LastWrite times of a Registry key with information from other sources, such as within the file system. A comprehensive discussion of correlating data between the Registry and file system is outside the scope of this article.

What's in the Registry?

A great deal of very useful forensic information can be found within the Registry, if the investigator knows where to look. Many times, performing a string search of the Registry will reveal little, as data may be kept in binary format, or as we'll see, in some cases is subject to ROT-13 ''encryption''. Knowing where to look within the Registry can be just as valuable to an investigator as the information he/she's seeking.

Autostart locations

There are locations within the Registry known as ''autostart'' locations because they allow applications to be launched without any conscious or

[2] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/filetime_str.asp.

direct user interaction. One of the most popular is the ubiquitous ''Run'' key, used by a great many pieces of malware to remain persistent on the victim system:

HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows\CurrentVersion\Run

There are many other Registry keys that work just as well, and launch applications when the system starts up, when the user logs in, when the user takes some specific action, or when some other specific action occurs. For example, the following Registry key (found in both the HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER hives) causes an application to be launched whenever the command processor (i.e., cmd.exe) is launched:

Software\Microsoft\Command Processor\AutoRun

The next Registry key causes DLLs listed in the key to be loaded whenever a graphical user interface (GUI) application is launched:

HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows NT\CurrentVersion\Windows\
AppInit_DLLs

Finally, the following Registry key allows the administrator to designate a debugger for a specific application:

HKEY_LOCAL_MACHINE \SOFTWARE\
Microsoft\Windows NT\CurrentVersion\
Image File Execution Options

An attacker can use this key to redirect an application to a Trojaned copy of that application. For example, by adding an entry for the key for Notepad, and pointing it to a copy of Notepad that has been Trojaned with a backdoor, the attacker can cause the backdoor to be launched whenever Notepad, or any other commonly used application, is launched.

Determining and examining the autostart locations can provide the investigator with clues as to whether the activity being investigated is a result of actions performed by the user, or those performed by malware or an attacker. One of the best sources of information on autostart locations within the Registry is the AutoRuns tool from SysInternals.com. This application has been updated several times to incorporate new locations and provides perhaps the most comprehensive listing of autostart locations. Investigators should take the time to run the tool on a live system, noting empty locations.

User activity

The NTUSER.DAT file for a particular user holds all of the Registry settings specific to that user. Within the Registry, the contents of this file are mapped to the HKEY_USERS*SID* hive, and when the user logs in, the HKEY_CURRENT_USER hive is created from the HKEY_USERS*SID* hive. These hives provide a great deal of information for the forensic investigator regarding actions taken by the user.

MRU lists

There are several keys within the Registry, within each user's hives, that maintain ''most recently used'', or MRU lists. These lists consist of entries made due to specific actions taken by the user, and maintained to keep track of items the user may return to in the future. For example, the Registry key that maintains a list of commands that the user types into the Start → Run box is:

HKEY_CURRENT_USER\Software\Microsoft\
Windows\CurrentVersion\Explorer\RunMRU

When the user types something into the Run box, an entry is added to this key. Fig. 2 illustrates an example of the contents of the RunMRU key.

As illustrated in Fig. 2, there are a number of values named for letters of the alphabet; in this case, from a through g. The MRUList entry maintains a list of which value has been most recently used. For example, the first entry is 'b', indicating that the last entry entered into the Run box was ''regedit''. The LastWrite time of the RunMRU key will reflect the date and time this entry was entered.

This also applies to several other Registry keys that have values named ''MRUList''. For example, the following Registry key maintains a list of

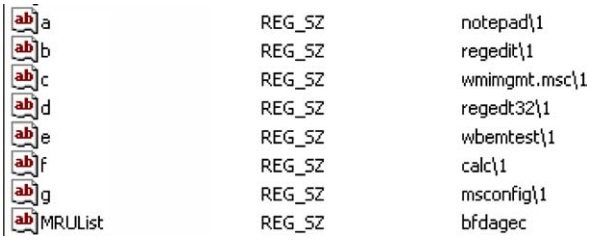

Figure 2 Contents of the Explorer\RunMRU key.

network drives the user has mapped via the Map Network Drive Wizard:

\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU

As with the other MRUList keys, the LastWrite time of the key indicates when the most recent item, denoted in the MRUList value, was accessed.

For a list of additional MRUList Registry keys, see the URL listed in Section Summary of this article.

UserAssist

The UserAssist key (i.e., Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist) contains two subkeys that look like globally unique identifiers, or GUIDs. Each of these subkeys records specific objects on the system that the user has accessed, such as Control Panel applets, shortcut files, programs, etc. However, these entries are ''encrypted'' using the algorithm known as ''ROT-13'', in which the characters are substituted with the character 13 spaces away from it in the ASCII table. These entries, once decrypted, are not definitive (i.e., they are not associated with a specific date and time), but they can illustrate that a user has accessed a particular file or object. For example, should the investigator find an entry within one of these keys that is the name of a malware file, it may indicate that the user took some action to launch the malware.

USB removable storage

As with other devices, information about USB removable storage devices (i.e., thumb drives) connected to a Windows system is maintained in the Registry. If a USB thumb drive was connected to the system, there will be an entry under the following key:

HKEY_LOCAL_MACHINE\System\ControlSet00x\Enum\USBSTOR

The ControlSet in use by an active system depends upon the data associated with the following Registry value:

HKEY_LOCAL_MACHINE\System\Select\Current

Beneath the USBSTOR key is device instance IDs for various USB storage devices that have been connected to the system. You may see device instance IDs for storage devices such as thumb drives and even digital cameras, as demonstrated in Fig. 3.

Fig. 3 also shows the device ID for a specific device identified. In this case, the device ID is for a USB thumb drive that had been connected to the system. This device ID also happens to be the serial number for the device, which is a unique value assigned by the manufacturer, similar to the MAC address of a network interface card. This serial number remains unique across Windows systems, and can be used to determine if a particular thumb drive had been connected to other Windows systems.

It should be noted that not all USB thumb drives will have a serial number. If the second character of the device ID seen above had been a ''&'' rather than ''7'', we would know that the device did not have a serial number assigned to it. In this example, the device happens to have a serial number.

The device ID key will have several values associated with it, one of which is the ParentIdPrefix. There is no vendor documentation available on how this particular value is created or used by the system, but a detailed examination of other locations within the Registry gives a clue as to how

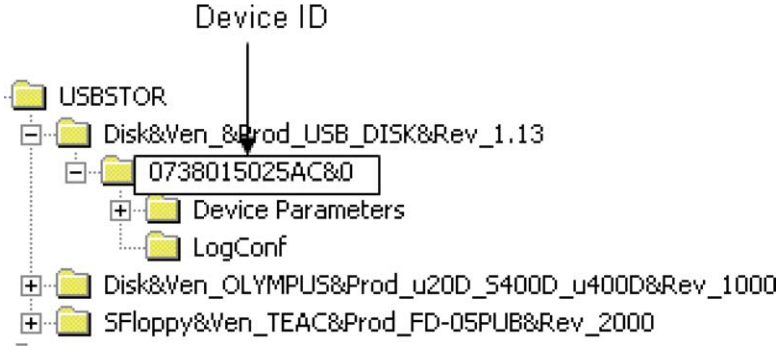

Figure 3 Example contents of USBSTOR key, showing device instance IDs.

the investigator can use this value. For example, navigate to the following Registry key:

HKEY_LOCAL_MACHINE\System\MountedDevices

This key stores a database of mounted volumes for the NTFS file system, with persistent volume names associated with unique internal identifiers for those volumes. The binary data for each \DosDevices*x*: value contains identifier information about the volume, in Unicode. Examining each of these values, we see that the data associated with the ParentIdPrefix value for our thumb drive also appear in the information about the volume that was assigned the drive letter G. Therefore, once the investigator has identified the USB removable storage device, he/she can map the ParentIdPrefix value to appropriate drive letter assignment.

Wireless SSIDs

On a running Windows XP system, you may have opened the Network Connections applet within the Control Panel, then opened the Properties for the Wireless Network Connection, and then opened the Wireless Networks tab. You may see more than a few service set identifiers (SSIDs) listed in the Preferred Networks box. This shows you which wireless networks you've connected to, and if you travel and make use of the ubiquitous wireless hotspots, you'll see quite a few entries there.

Have you ever thought about where these values are maintained? After all, some of the SSIDs may be from networks you connected to a year or more ago…where are they on your system? Better yet, what if you were investigating a case and had an image of a Windows XP system used by the suspect, and you wanted to know which wireless networks he'd/she'd have connected to? Well, open the Registry Editor and navigate to the following key:

HKEY_LOCAL_MACHINE\Software\Microsoft\WZCSVC\Parameters\Interfaces

Beneath this key, you may see one or more subkeys that (once again) look like globally unique identifiers, or GUIDs. Open one, and see if you see the value ''ActiveSettings'', and then one called ''Static#0000''. There may be more values with names that begin with ''Static#'' and are numbered sequentially. These values are binary data types, and if you right click on one and choose ''Modify'', you'll see the binary data in a dialogue box, with SSID for the connection plainly visible.

The network settings (IP address, etc.) for the interface can be found by navigating to the following Registry key:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Interfaces*GUID*

The investigator can use this information to tie a user to a particular IP address, during a particular timeframe. On imaged systems, the investigator will want to examine the specific control set that was active when the system was running.

Summary

The Windows Registry is a significant forensic resource. Knowing what information is available and where the information can be found can lead the investigator to develop a more comprehensive picture of the case. Some of that information has been presented here, but there is considerably more. For a comprehensive list of Registry keys listing autostart locations and keys listing user activity, with an explanation of the purpose or use of the keys and applicable references where appropriate, visit the following URL:

http://www.windows-ir.com/regkeys.zip

Harlan Carvey, CISSP, is a computer security engineer residing at the Metro DC/Northern Virginia area. He has conducted vulnerability assessments and incident response activities with regards to Windows systems, and currently specializes in the forensic analysis of Windows systems. He is the author of ''Windows Forensics and Incident Recovery'' (available on Amazon, etc.).