Forensic Analysis of the Windows Registry

Lih Wern Wong
School of Computer and Information Science, Edith Cowan University
lihwern@yahoo.com

Abstract

Windows registry contains lots of information that are of potential evidential value or helpful in aiding forensic examiners on other aspects of forensic analysis. This paper discusses the basics of Windows XP registry and its structure, data hiding techniques in registry, and analysis on potential Windows XP registry entries that are of forensic values.

Keywords:

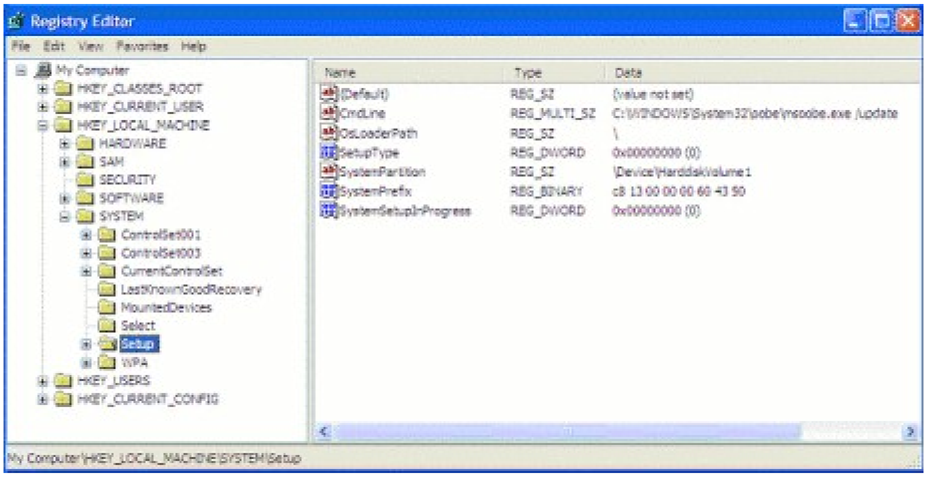
Windows registry, forensic analysis, data hiding

INTRODUCTION

Windows 9x/ME, Windows CE, Windows NT/2000/XP/2003 store configuration data in registry. It is a central repository for configuration data that is stored in a hierarchical manner. System, users, applications and hardware in Windows make use of the registry to store their configuration and it is constantly accessed for reference during their operation. The registry is introduced to replace most text-based configuration files used in Windows 3.x and MS-DOS, such as .ini files, autoexec.bat and config.sys. Due to the vast amount of information stored in Windows registry, the registry can be an excellent source for potential evidential data. For instance, windows registry contains information on user accounts, typed URLs, network shared, and Run command history. Aspects discussed in this paper are based solely on Windows XP (Service Pack 2) registry.

REGISTRY STRUCTURE

Figure 1 shows Windows registry logical view from Register Editor (Windows default register editor). Each folder in the left key pane is a registry key. The right panes show the key's value. Subkey is used to show the relationship between a key and the keys nested below it. Branch refers to a key and all its subkeys. Windows uses symbolic link (i.e. similar to file system's shortcut) to link a key to a different path which allows the same key and its values to appear at two different paths (Russinovich, 1999).



Name	Type	Data
(Default)	REG_SZ	(value not set)
CmdLine	REG_MULTI_SZ	C:\WINDOWS\System32\oobe\msoobe.exe /update
OsLoaderPath	REG_SZ	\
SetupType	REG_DWORD	0x00000000 (0)
SystemPartition	REG_SZ	\Device\HarddiskVolume1
SystemPrefix	REG_BINARY	c8 13 00 00 00 60 43 50
SystemSetupInProgress	REG_DWORD	0x00000000 (0)

My Computer\HKEY_LOCAL_MACHINE\SYSTEM\Setup

Figure 1: Windows Registry Logical View Key

Forensic Analysis of the Windows Registry

There are 5 root keys (i.e. starting point) in Windows registry. Table 1 shows the root keys and the abbreviation normally used.

Name	Abbreviation
HKEY_CLASSES_ROOT	HKCR
HKEY_CURRENT_USER	HKCU
HKEY_LOCAL_MACHINE	HKLM
HKEY_USERS	HKU
HKEY_CURRENT_CONFIG	HKCC

Table 1: Root Keys

Value

Each key has one or more values. There are 3 parts in value, which are Name, Type and Data, as shown in Table 2.

Value Parts	Description
Name	Every value has a unique name in that particular key.
Type	Value’s type determines the type of data value contains. The common value types in registry for instance are: REG_BINARY type contains binary data; REG_DWORD type contains double-word (32-bit) data; REG_SZ type contains fix-length string data.
Data	Value’s data contains data which usually relates to the value’s type.

Table 2: Value Parts

When an application read value’s data in REG_BINARY from the registry, the application decides on how to decode the value. Application can store data in binary (using REG_BINARY type) using their own data structure, hence only the application knows how to interpret it. For instance, interpreting REG_BINARY data as 8-bit ASCII or 16-bit Unicode could result in two different values. This technique could be used to hide data or at least confuse forensic examiner. Alternatively, some applications store REG_SZ and REG_DWORD data in REG_BINARY value, decoding and finding them can be difficult (Honeycutt, 2003, p. 25). Offender can use this technique to hide data. Program can use four-byte REG_BINARY and REG_DWORD values (32-bit) interchangeably. Since Intel x86-based system uses little endian architecture, REG_BINARY 0x01 0x02 0x03 0x04 is equivalent to REG_DWORD 0x04030201.

Regardless of value’s type, the registry actually stores all values in binary format in the actual file. Since all values are stored alongside with their corresponding type, it allows the Registry Editor to interpret the value’s data correctly (Honeycutt, 2003, p. 25).

REGISTRY ROOT KEYS ORGANIZATION

HKLM and HKU are the only root keys that Windows physically stores on files. HKCU is a symbolic link to subkey in HKU. HKCR and HKCC are symbolic links to subkeys in HKLM. Below are the brief descriptions of each 5 root keys (Honeycutt, 2003, p. 26).

HKEY_USER

HKU contains per-user (user-specific) information. HKU contains at least these 3 subkeys:
- .DEFAULT
- SID, SID is the security identifier for console user (user currently using the keyboard).
- SID_CLASSES contains per-user class registration and file association.

HKU has other well-known SID in Windows XP.
- S-1-5-18 refers to LocalSystem account.
- S-1-5-19 refers to LocalService account. It is used to run local services that do not require LocalSystem account.
- S-1-5-20 refers to NetworkService account. It is used to run network services that do not require LocalSystem account.

Any other subkeys in HKU are associated to secondary users. Windows XP has a feature called Secondary Logon, which allows user to run a program as a different user, usually with elevated privileged. Thus, user can logon to a limited account for daily routines and uses elevated privileged for occasional administrative task. The secondary user SID (usually administrative account SID) will only present in the HKU subkeys if the user performs a secondary logon during the user's session. If an offender performs a secondary logon on any other accounts, the secondary user subkey will exist in HKU until secondary user logoff, or the program running in the elevated privileged is closed.

HKEY_CURRENT_USER

HKCU contains the computer users' per-user settings. HKCU is actually a symbolic link to HKU/SID, the current console user's SID (Russinovich, 1997). This branch contains information on environmental variables, desktop settings, mapped network drive settings, and application settings. Table 3 briefly describes some HKCU subkeys that are of potential forensic values (Honeycutt, 2003, p. 28).

Subkeys	Descriptions
Environment	Each subkey corresponds to an environmental variable user has set.
Identities	Each Identities subkey corresponds to an identity in Microsoft Outlook Express. Outlook Express allows multiples identities (users) to use a single mail client. However, since Windows XP supports multiple user profiles, users rarely have to share their mail client.
Network	Each Network subkey corresponds to a mapped drive Windows connects during user system logon. Subkey name is the drive letter to which the network drive is mapped. The subkey contains configuration to connect the network drive.
Software	Contains user-specific application settings. Programs store their settings in a standard way, HKCU\Software\Vendor\Program\Version\. Vendor is program's publisher; Program is the program's name; and Version is program's version.
Volatile Environment	Contains environmental variables that are defined when user logon to Windows XP.

Table 3: Partial HKCU Subkeys

HKEY_LOCAL_MACHINE

HKLM contains per-computer (computer-specific) settings which apply to all users logging into that particular computer. Table 4 shows all HKLM subkeys (Honeycutt, 2003, p. 29).

Subkeys	Descriptions
HARDWARE	Stores information regarding hardware Windows XP detects during startup. The subkeys are dynamically created during system startup. They include information on device driver and associated resources.
SAM	Security Accounts Manager (SAM) is a local security database which contains local users and groups information. ACL prevents Administrator from viewing this subkey.
SECURITY	Contains Windows local security database in the SAM subkey. ACL prevents Administrator from viewing this subkey.
SOFTWARE	Stores per-computer application settings. Programs store their settings in this standard form, HKLM\Software\Vendor\Program\Version.
SYSTEM	Contains control set, which contains device driver and service configurations. HKLM\SYSTEM\CurrentControlSet is a symbolic link to ControlSetXXX, and the key HKLM\SYSTEM\Select indicates which ControlSetXXX is in use.

Table 4: HKLM Subkeys

HKEY_CLASSES_ROOT

HKCR contains two types of per-user settings, file associations, and class registration for Component Object Model (COM) object. File associations describes the file types and associated programs that open and edit them. HKCR consumes most of the space in registry (Russinovich, 1997). Windows merges two keys HKLM\SOFTWARE\Classes (contains default file associations and class registration) and HKCU\Software\Classes (contains per-user file associations and class registration) to obtain HKCR. In fact, HKCU\Sofware\Classes is a link to HKU\SID_Classes. By merging the two keys, program can register per-computer and per-user file associations and program classes (Honeycutt, 2003, p. 29).

HKEY_CURRENT_CONFIG

HKCC is a symbolic link to current hardware profile configurations subkey, HKLM\SYSTEM \CurrentControlSet\Hardware Profiles\Current. Current is a link to the key HKLM \SYSTEM\CurrentcontrolSet\Hardware Profiles\XXXX (Honeycutt, 2003, p. 30).

REGISTRY HIVES

Registry Editor only shows the logical structure of the registry. Physically, registry is not stored in a single file in the hard drive. Windows stores registry in a few separated binary files called hives (Microsoft, 2005a). For each hives file, Windows creates additional supporting files that contain backup copy of the respective hives to restore the hives during failed system boot. Only HKLM and HKU has corresponding hives (since the rest are symbolic links). However, none of 5 root keys are directly associated to a hive file.

Table 5 shows registry path and their corresponding hives on disk. All hives in HKLM are stored in %SYSTEMROOT%\System32\config\ (%SYSTEMROOT% usually refers to C:\WINDOWS). HKLM\HARDWARE is a dynamic hive that is created each time the system boots and it is created and managed entirely in memory (Russinovich, 1999). HKU\.DEFAULT hive file correspond to %SYSTEMROOT%\System32\config\default. HKU\SID hive file is stored in user home directory, which is %USERPROFILE%\NTUSER.DAT, while HKU\SID_CLASSES hive file correspond to %USERPROFILE%\Local Settings \Application Data\Microsoft\Windows\UsrClass.dat. Table 6 describes the actual hive files and the supporting files extension (Honeycutt, 2003, p. 31).

Registry Path	Hive and Supporting Files
HKLM\SAM	SAM, SAM.LOG
HKLM\SECURITY	SECURITY, SECURITY.LOG
HKLM\SOFTWARE	software, software.LOG, software.sav
HKLM\SYSTEM	system, system.LOG, system.sav
HKLM\HARDWARE	(Dynamic/Volatile Hive)
HKU\.DEFAULT	default, default.LOG, default.sav
HKU\SID	NTUSER.DAT
HKU\SID_CLASSES	UsrClass.dat, UsrClass.dat.LOG

Table 5: Registry Hives

File Extension	Description
No extension	Actual Hive File
.alt extension	Backup copy of hive, used in Windows 2000, not XP
.log extension	Transaction log of changes to a hive
.sav extension	Backup copy of hive created at the end of text-mode (console) phrase during Windows XP setup

Table 6: Hive Files Extension

REGISTRY LASTWRITE TIME

All registry key has a value called “LastWrite” time, which is similar to file’s last modification time. In fact, this value is a FILETIME structure, which is the same as file’s MAC (Modified, Accessed, Created) time (Tan, 2001). The FILETIME structure is a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 UTC (MSDN, 2005c). However, investigator could only obtain the registry key LastWrite time, but not the registry value LastWrite time. The LastWrite time will be updated whenever a registry value in the key is created, modified or deleted. Tool such as Keytime.exe (Carvey, 2005a) allows examiner to retrieve LastWrite time of a specific key. Knowing the time of a key is modified or created allows forensic investigator to infer the approximate time an event or activity occurred. For instance, if a suspicious registry value is found in the registry’s Run key, investigator could query the LastWrite time of the key and compare it to the MAC time of the file to which the registry value is pointing. If there is a match between the key LastWrite time and the MAC time of the file to which the registry value is pointing, investigator will know the time the registry value was created.

DATA HIDING IN REGISTRY

Suspect can hides all sorts of data including password, text information, and binary files in registry. Suspect can effectively hide data in registry keys’ value entries. By using different encoding techniques, suspect could obfuscate or hide data from forensic examiner. Furthermore, Register Editor has an implementation flaw that allows suspect to hide data.

Registry Keys' Values

Since registry's value supports binary data type, suspect can store segments of program or the entire binary in the registry. These segments of program can be placed in several dispersed keys. Unless forensic examiner knows the relevant keywords to search in the registry, finding hiding data in tens of thousands of registry keys can be a tedious task.

An example of a place to hide data is in the time zone information key, HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation (Carvey, 2004). This key contains time zone information, including the difference in minutes between UTC and local time, and reference information during daylight saving time. Windows reads this registry key into TIME_ZONE_INFORMATION structure during system startup.

There are two strings in TIME_ZONE_INFORMATION structure, StandardName and DaylightName, of which can legally be an empty string (MSDN, 2005b). Any information written to them using SetTimeZoneInformation() function is returned unchanged by the GetTimeZoneInformation() function (MSDN, 2005a). Since Windows does not utilize these registry values which are nested somewhere in some registry keys, and they are merely used for storing string information, suspect can hides information such as passwords or passphrases in these values effectively. Suspect merely modifies registry values StandardName and DaylightName manually using Registry Editor to store information. Suspect can retrieve this information using a piece of benign code by calling GetTimeZoneInformation() function which is loaded in Windows kernel32.dll without raising much suspicion (Carvey, 2004).

Different Encoding

Suspect may store text-based information using value type REG_BINARY. This technique however does not hide data, as tool like hex editors automatically interpret binary data into readable format (usually ASCII). Using different encoding technique to store data, such as using Unicode instead of ASCII does not improve stealthiness, if suspect only uses common English characters. For instance ASNI ASCII for "pass" is 0x70 0x61 0x73 0x73. While Unicode (16-bit) encoding translate into 0x70 0x00 0x61 0x00 0x73 0x00 0x73 0x00 (Windows stores 16-bit characters in little-endian format). Examiner could easily find the word "pass" using tools that features text finding using different encoding format. Suspect may substitute the 0x00 with random binary numbers to improve stealthiness. However, forensic examiner could still analyse the suspicious text at different intervals (e.g. even or odd characters position) and derive possible meaningful information from the incident context.

A better way to hide data is to encode text-based information into binary format in hexadecimal notation and stored the binary form in registry values as string using type REG_SZ. For instance, storing string 70 61 73 73 (hexadecimal notation for "pass" in ASCII) in the REG_SZ registry value. Thus, only the suspect knows how to decode it. However, this technique requires a simple piece of code to encode the text before storing it into the registry, and to decode the binary data to its readable form when retrieving it. It is not-trivial for forensic examiner to find such hidden data as the binary data (encoded text in hexadecimal form) is stored as it is in the registry, and binary data is common in registry.

Registry Editor Implementation Flaw

Windows 2000 and XP Registry Editor (regedit.exe or regedt32.exe) have an implementation flaw that allows hiding of registry information from viewing and editing, regardless of users access privilege (Secunia, 2005). The flaw involves any registry values with name from 256 to 259 (maximum value name) characters long. The overly long registry value (regardless of type) not only hides its own presence, but also subsequently created values (regardless of type) in the same key (Franchuk, 2005). The editor stops displaying the remaining of the values thinking the overly long value as the last value in that key. Suspect could exploit such Registry Editor flaw to hide information.

This vulnerability allows malware to hide malicious code in “autorun” entries such as the infamous HKLM\Software\Microsoft \Windows\CurrentVersion\Run. Any program or components specified in this key will be automatically run during system startup. Windows will still execute these hidden entries successfully at startup (Wesemann, 2005).

Some common malware scanners are not able to detect such maliciously crafted registry values (Gregg, 2005). Fortunately, Windows console registry tool (reg.exe) can display overly long registry values. For instance, to detect values in registry key HKLM\Software\Microsoft\Windows\CurrentVersion\Run, the instruction is reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run.

REGISTRY KEYS OF FORENSIC VALUE

The following section highlights some of the important registry keys in Windows XP (Service Pack 2) and how they can be of benefit to help describing suspect activities on the computer.

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU

MRU is the abbreviation for most-recently-used. This key maintains a list of recently opened or saved files via typical Windows Explorer-style common dialog boxes (i.e. Open dialog box and Save dialog box) (Microsoft, 2002). For instance, files (e.g. .txt, .pdf, htm, .jpg) that are recently opened or saved files from within a web browser (including IE and Firefox) are maintained. However, documents that are opened or saved via Microsoft Office programs are not maintained. Subkey * contains the full file path to the 10 most recently opened/saved files. Other subkeys in OpenSaveMRU contain far more entries related to previously opened or saved files (including the 10 most recent ones), which are grouped accordingly to file extension.

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU

This key correlates to the previous OpenSaveMRU key to provide extra information. Whenever a new entry is added to the previous OpenSaveMRU key, registry value is created or updated in this key. Each binary registry value under this key contains a recently used program executable filename, and the folder path of a file to which the program has been used to open or save it. If a file is saved, the folder path refers to the saved file destination path; if a file is opened, the folder path refers to the file source path. New registry value will only be created to this key, if no existing registry values contain the program executable filename. However, if there is a matching executable filename in the existing values, only the folder path section of the related registry value is updated.

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

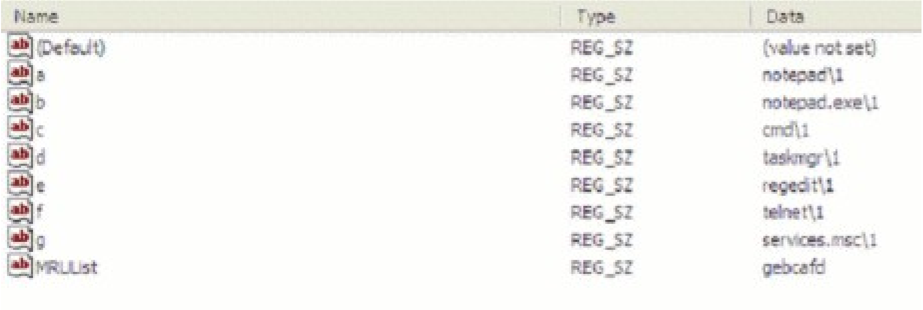
This key also maintains list of files recently executed or opened through Windows Explorer. This key corresponds to %USERPROFILE%\Recent (My Recent Documents). The key contains local or network files that are recently opened and only the filename in binary form is stored. It has similar grouping as the previous OpenSaveMRU key, opened files are organized according to file extension under respective subkeys. In addition, the Subkey Folder contains the folder (without drive letter and parent folder) of the recently open files. Subkey NetHood which corresponds to %USERPROFILE%\NetHood , contains only LAN shared folder path (server and folder name) which the file was opened. However, deleting this RentDocs key does not removed the content in both folders %USERPROFILE%\Recent and %USERPROFILE%\NetHood (Honeycutt, 2003, p. 102).

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU

This key maintains a list of entries (e.g. full file path or commands like cmd, regedit, compmgmt.msc) executed using the Start>Run commands, as shown in Figure 2. The MRUList value maintains a list of alphabets which refer to the respective values. The alphabets are arranged according to the order the entries is being added, In Figure 2, “services.msc” which correlates to “g’ is the most recently added entry, while “taskmgr” is the earliest.

However, most recently added entry does not imply most recently used command as suspect may have re-executed previous commands. Windows does not modify the key LastWrite time or MRUList if there is an existing entry in the key. If a file is executed via Run command, it will leaves traces in the previous two keys OpenSaveMRU and RecentDocs. Deleting the subkeys in RunMRU does not remove the history list in Run command box immediately. However, when either button Start>Log Off or Turn Off Computer is clicked (without actually logging off or shutdown), the respective entries in Run history list are then removed. By using Windows "Recent Opened Documents" Clear List feature via Control Panel>Taskbar and Start Menu, suspect can remove the Run command history list. In fact, executing the Clear List function will remove the following registry keys and their subkeys:

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU\
HKCU\Software\Microsoft\Internet Explorer\TypedURLs\
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU



Name	Type	Data
(Default)	REG_SZ	(value not set)
a	REG_SZ	notepad\1
b	REG_SZ	notepad.exe\1
c	REG_SZ	cmd\1
d	REG_SZ	taskmgr\1
e	REG_SZ	regedit\1
f	REG_SZ	telnet\1
g	REG_SZ	services.msc\1
MRUList	REG_SZ	gebcafd

Figure 2: Content of RunMRU Key

HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management

This key maintains Windows virtual memory (paging file) configuration. The paging file (usually C:\pagefile.sys) may contain evidential information that could be removed once the suspect computer is shutdown. This key contains a registry value called ClearPagefileAtShutdown which specify whether Windows should clear off the paging file when the computer shutdowns. By default, windows will not clear the paging file. However, suspect may modify this registry value to 1 to signify paging file clearing during system shutdown (Microsoft, 2003). Forensic investigator should check this value before shutting down a suspect computer during evidence collection process.

HKCU \Software\Microsoft\Search Assistant\ACMru

This key contains recent search terms using Windows default search. Subkey 5603 contains search terms for finding folders and filenames, while subkey 5604 contains search terms for finding words or phrases in a file (Honeycutt, 2003, p. 102).

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

Each subkey in this key represent an installed program in the computer. All programs listed in Control Panel>Add/Remove Programs correspond to one of the listed subkeys. However, they are other installed programs (e.g. device driver, Windows patch) that are not listed in Add/Remove Programs. Each subkey usually contains these two common registry values – DisplayName (program name) and UninstallString (application Uninstall component's file path, which indirectly refers to application installation path). Other possible useful registry values may exist, which include information on install date, install source and application version.

HKLM \SYSTEM\MountedDevices

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\CPC\Volume\

The first key contains a list of mounted devices, with associated persistent volume name and unique internal identifier for respective devices (Carvey, 2004). This key lists any volume that is mounted and assigned a drive letter, including USB storage devices and external DVD/CDROM drives. From the listed registry values, value's name that starts with "\DosDevices\" and ends with the associated drive letter, contains information regarding that particular mounted device. For instance, if the binary data for registry value "\DosDevices\F" contains "\??\Storage#RemoveableMedia" at the beginning of the value, it signifies a USB removable disk was connected to the system USB port. By correlating the entry with registry key LastWrite time, investigator would know when the removable device is connected. The second key also contains similar information as MountedDevices key, which is located under the respective device GUID (Globally Unique Identifiers) subkey and in the binary registry value named Data.

HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR

This key contains addition information about list of mounted USB storage devices, including external memory cards. This key when used in conjunction with two previous keys will provide evidential information. To illustrate the method, assume a USB thumb drive named "USB Card IntelligentStick" with serial number "20000101061325-00" was connected to a suspect system. USB storage device unique serial number can be acquired via UVCView program, under the field "iSerialNumber" (UVCView, 2005). However, not every USB thumb drive has a serial number (Carvey, 2005e). This key will have a subkey containing device name, such as "Disk&Ven_USB_Card&Prod_IntelligentStick&Rev_1.00 ". Under this subkey is the device ID subkey which contains the device serial number; "20000101061325-00&0". The latter subkey has a ParentIdPrefix value (data="7&1064d032&0") which corresponds to the binary registry value in HKLM \System\MountedDevices ; \DosDevices\F for instance. The latter value will contain binary data similar to "\??\Storage#RemoveableMedia#7&1064d032&0……". By mapping this two key, forensic examiner will know which USB device (using device serial number) is mounted to which drive letter. Apple iPod devices leave similar trace (Carvey, 2005d).

HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\Run
HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\RunOnce
HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\RunOnceEx
HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\RunServices
HKLM\ SOFTWARE \Microsoft\Windows\CurrentVersion\RunServicesOnce

This first key usually contains programs or components paths that are automatically run during system startup without requiring user interaction. Malware usually leaves trace in this key to be persistent whenever system reboots. Subsequent four subkeys may also contain suspicious entries. Similar 5 sets of "Run" registry keys may exist under root key HKCU, pertaining to the logged on user configuration (Carvey, 2004).

HKLM\SOFTWARE\Microsoft\Command Processor

HKCU\Software\Microsoft\Command Processor

This key has a registry value named Autorun, which could contain command that is automatically executed each time cmd.exe is run (Microsoft, 2005b). However, modification to this key requires administrative privilege. Malware exploits this feature to load itself without user's knowledge (Symantec, 2004). Suspect could also covertly run a malicious program under the cover of cmd.exe, by setting the Autorun data to the executable file path.

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

This key has a registry value named Shell with default data Explorer.exe. Malware such as Kwbot appends the malware executable file to the default value's data, modifying it into Shell=Explorer.exe %system%\System32.exe to stay persistence across system reboots and logins (Symantec, 2003). Suspect could append executable file path to this registry value to run program covertly as done by Trojan Watson (Symantec, 2004). Furthermore, there is another registry value in this key named TaskMan which allows user to run an alternate task manager (Microsoft, 2005c). Though by default it is not created in Windows XP, suspect can create it and point it to an executable file. Both registry values are executed automatically whenever the system boots. Suspect can utilize these two registry values to run program secretly. However, modification to this key requires administrative privilege.

HKLM\SYSTEM\CurrentControlSet\Services\

This key contains list of Windows services. Each subkey represents a service and contains service's information such as startup configuration and executable image path. Some malware such as BackOrifice2K will install itself as service. Thus, it leaves trace in this key (Carvey, 2001).

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

This key allows administrator to map an executable filename to a different debugger source, allowing user to debug a program using a different program. Modification to this key requires administrative privilege. Suspect could exploit this feature to launch a completely different program under the cover of the initial program (Epp, 2005). First, suspect creates a subkey named for example, notepad.exe (taskmgr.exe, compmgmt.msc or any benign looking executable). Then under the subkey notepad.exe, suspect creates a new string (REG_SZ) value named Debugger, and directs it to an undercover program (e.g. C:\Windows\system32\telnet.exe). When the suspect executes notepad.exe, telnet client is launched instead of Notepad. If the suspect runs notepad.exe through Windows Run for instance, its history list will only shows notepad.exe. Thus, suspect could use this technique to deceive forensic examiner. Suspect could also redirect the initial program to a Trojan version of the program which launches a backdoor whenever the initial program is run. Malware exploits this feature to load itself without user's knowledge (Symantec, 2005).

HKCR\exefile\shell\open\command\

This key contains instruction to execute any .exe extension file. Normally, this key contains one default value with data "%1" %* (ShaolinTiger, 2003). However, if the value's data is changed to something similar to somefilename.exe "%1" %* , investigator should suspect some other hidden program is invoked automatically when the actual .exe file is executed. Malware normally modify this value to load itself covertly (File Extensions, n.d.). This technique apply to other similar keys (Carvey, 2004), including

HKEY_CLASSES_ROOT\batfile\shell\open\command
HKEY_CLASSES_ROOT\comfile\shell\open\command

HKCR\Drive\shell\
HKCR\Folder\shell\

These two key contains subkeys that refer to menu items in Windows context menu. The first key points to the context menu when right clicking on Windows drive letter, while the second key refers to folder's context menu. Suspect could create a key to launch command prompt from the drive letter context menu, through key HKCR\Drive\shell\cmd\command\. It is a very helpful feature especially if users need to open command prompt at folder level, via HKCR\Folder\shell\cmd\command. By default, Windows does not have this key. The default registry value has data cmd.exe /k "cd %L". Suspect could append for instance && notepad.exe to this value to launch both programs at once (Carvey, 2005b). However, the second program (notepad.exe) is loaded within the same cmd.exe window (cmd.exe is not fully loaded until notepad.exe is closed). By modifying the default registry value's data to cmd.exe /k "cd %L" && start notepad.exe , the two programs are launched separated under different windows. Thus, the second program can be loaded covertly.

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\GUID

This key contains network adapter recent settings such as system IP address and default gateway for the respective network adapters. Each GUID subkey refers to a network adapter (AccessData, 2005a). The data is retained even though the network connection is disconnected.

HKLM\SOFTWARE\Microsoft\WZCSVC\Parameters\Interfaces\GUID

This key contains wireless network information for adapter using Windows Wireless Zero Configuration Service. Under the GUID subkey, there are binary registry values named Static#0000, Static#0001, etc. (depending on the number of listed SSID) which correspond to the respective list of SSID in "Preferred Networks" box in Wireless Network Connection configuration (Carvey, 2005e). The registry value contains the SSID name in binary form. If registry value ActiveSettings contains an SSID name, it may signify last connected SSID. However, the result is not consistent when tested. If suspect connect to wireless networks using other 3rd party program that is usually bundled with the network adapter, instead of using Wireless Zero Configuration, no trace is left on this key. Forensic examiner can use this key with the previous network adapter GUID key to determine the last assigned IP address (Carvey, 2005e).

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2

The first key maintains a list of mapped network drive, including the server name and shared folder (Shannon, 2004). The value in this key is still retained even though the mapped network drive has been permanently removed or disconnected. In addition, permanent subkey (unless manually removed from registry) regarding mapped network drive is also created in the second key, and the subkey is named in the form of ##servername#sharedfolder.

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist

This key contains two GUID subkeys. Each subkey maintains a list of system objects such as program, shortcut, and control panel applets that a user has accessed. The GUID subkey beginning with "5E6" corresponds to IE toolbar, while subkey starting with "750" pertains to Active Desktop (Carvey, 2005c). However, registry values under these subkeys are weakly encrypted using "ROT-13" algorithm which basically substitutes a character with another character 13 position away from it in the ASCII table (Carvey, 2005e). Even though each registry value is not associated with specific time and date the event occurred, it could imply suspect has accessed certain file or object. For instance, the existence of an attack tool's filename on the entries could indicate suspect is trying to execute the malicious tool.

HKCU\Software\Microsoft\Protected Storage System Provider

Windows Protected Storage is maintained under this key. Protected Storage is a service used by Microsoft products to provide a secure area to store private information (Carvey, 2004). Information that could be stored in Protected Storage includes MSN Explorer and Internet Explorer AutoComplete strings and passwords, Microsoft Outlook and Outlook Express accounts' passwords, and MSN Messenger password. Registry Editor hides these registry keys from users viewing, including administrator. There are tools that allow examiner to view the decrypted Protected Storage on a live system, such as Protected Storage PassView (NirSoft, 2004) and PStoreView (PStoreView, 2005). AccessData Registry Viewer is capable of accessing and decrypting the subkeys in an offline manner (AccessData, 2005b).

HKCU\Software\Microsoft\Internet Explorer\TypedURLs

This key contains a listing of 25 recent URLs (or file path) that is typed in the Internet Explorer (IE) or Windows Explorer address bar. It shows websites suspect has recently been surfing. However, the key will only show links that are fully typed, automatically completed while typing, or links that are selected from the list of stored URLs in IE address bar. Websites that are accessed via IE Favorites are not recorded. IE will only write all the typed URLs during that session to the key when IE is closed (AccessData, 2005a). If suspect clears the URL history using Clear History via IE Internet Options menu, this key will be completely removed.

CONCLUSION

Windows registry is an excellent source for potential evidential data. Knowing the type of information that could possible exist in registry and location to it gives forensic examiner the edge in the forensic analysis process. Investigator will get a better picture of the whole case. This paper illustrates some of techniques to hides data in registry and registry keys of evidential value. The fact that Microsoft and other organizations treat the registry settings as in-house information without providing sufficient and comprehensive documentation about the registry keys used causes registry analysis difficult, which undermines the resourcefulness of registry. Thus, there is a need to unveil and publish evidentiary registry keys to assist forensic investigation on Windows system.

REFERENCES

AccessData (2005a). AccessData – Registry Quick Find Chart. Retrieved October 8, 2005, from http://www.accessdata.com/files/whitepapers/rqfc-8-8-05.pdf

AccessData (2005b). Registry Viewer – Overview. Retrieved October 8, 2005, from http://www.accessdata.com/Product01_Overview.htm?ProductNum=01

Carvey, H. (2001, August 15). NT/2K Incident Response Tools. Retrieved September 26, 2005, from http://www.securityfocus.com/infocus/1294

Carvey, H. (2004).Windows Forensics and Incident Recovery. United State of America: Addison Wesley.

Carvey, H. (2005a). Windows Forensics and Incident Recovery – Tools. Retrieved September 29, 2005, from http://www.windows-ir.com/tools.html

Carvey, H. (2005b, April 8). Windows Incident Response - Interesting stuff on Windows. Retrieved October 1, 2005, from http://windowsir.blogspot.com/2005_04_01_windowsir_archive.html

Carvey, H. (2005c, April 25). Windows Incident Response - Registry keys posted on the web. Retrieved October 9, 2005, from http://www.windows-ir.com/regkeys.zip

Carvey, H. (2005d, August 2). iPod Info. Retrieved October 1, 2005, from http://windowsir.blogspot.com/2005_08_01_windowsir_archive.html

Carvey, H. (2005e, September). The Windows Registry as a forensic resource. Digital Investigation, 2(3), 201-205.

Dodd, A. (2005, April).Using areas of the Microsoft Windows registry to mine data. Retrieved September 27, 2005, from http://www.forensicfocus.com/computer-forensics-newsletter-april-2005

Epp, D. (2005, March 21).Using Image File Execution options as an Attack Vector on Windows. Retrieved September 30, from http://silverstr.ufies.org/blog/archives/000809.html

File Extensions (n.d.). Potentially harmful extensions. Retrieved September 30, 2005, from http://www.file-ext.com/harmful.html

Franchuk, I. (2005, August 24).[Full-disclosure] Miscrosoft Registry Editor 5.1/XP/2K long string key vulnerability. Retrieved September 26, 2005, from http://lists.grok.org.uk/pipermail/full-disclosure/2005-August/036448.html

Gregg, K. (2005, August 26). Windows Flaw May Let Hackers Hide Code From AV Scanners. Retrieved September 26, 2005, from http://www.techweb.com/showArticle.jhtml?articleID=170100835

Honeycutt, J. (2003). Microsoft Windows XP Registry Guide. United State of America: Microsoft Press.

Microsoft (2002, June 11). Names of Previously Opened and Saved Files Appear in Typical Windows Explorer-Style Dialog Boxes in Windows XP. Retrieved October 8, 2005, from http://support.microsoft.com/default.aspx?scid=kb;en-us;322948#XSLTH3132121124120121120120

Microsoft (2003, June 3). How to Clear the Windows Paging File at Shutdown. Retrieved September 27, 2005, from http://support.microsoft.com/kb/q182086/

Microsoft (2005a, August 12). Description of the Microsoft Windows registry. Retrieved September 27, 2005, from http://support.microsoft.com/default.aspx?scid=kb;EN-US;256986

Microsoft (2005b). Command Processor – Autorun. Retrieved October 1, 2005, from http://www.microsoft.com/resources/documentation/Windows/2000/server/reskit/en-us/ Default.asp?url=/resources/documentation/Windows/2000/server/reskit/en-us/regentry/942.asp

Microsoft (2005c). Winlogon – TaskMan. Retrieved October 1, 2005 from http://www.microsoft.com/resources/documentation/Windows/2000/server/reskit/en-us/ Default.asp?url=/resources/documentation/Windows/2000/server/reskit/en-us/regentry/ 58543.asp

MSDN (2005a, September). GetTimeZoneInformation. Retrieved September 27, 2005, from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/ gettimezoneinformation.asp

MSDN (2005b, September). TIME_ZONE_INFORMATION. Retrieved September 27, 2005, from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/time_zone_information_str.asp

MSDN (2005c, September). FILETIME. Retrieved September 30, 2005, from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/sysinfo/base/filetime_str.asp

NirSoft (2004). Protected Storage PassView v1.62 – Recover Protected Storage Passwords. Retrieved October 9, 2005, from http://www.nirsoft.net/utils/pspv.html

PStoreView (2004). Retrieved October 9, 2005, from http://www.ntsecurity.nu/toolbox/pstoreview/

Russinovich, M. (1999, May). Inside The Registry. Retrieved September 27, 2005, from http://www.windowsitpro.com/Articles/Print.cfm?ArticleID=5195

Russinovich, M. (1997, April). Inside the Windows NT Registry. Retrieved September 27, 2005, from http://www.windowsitpro.com/Articles/Print.cfm?ArticleID=122

Secunia (2005, August 29). Windows Registry Editor Utility String Concealment Weakness. Retrieved September 26, 2005, from http://secunia.com/advisories/16560/

Wesemann, D. (2005, August 29). Nasty Games of Hide and Seek in the Registry; Nepenthes. Retrieved September 26, 2005, from http://isc.sans.org/diary.php?date=2005-08-24

Shannon, M. (2004, March 26). Accessing and Analyzing the Windows Registry. Retrieved September 25, 2005, from http://www.agilerm.net/linux2.html

ShaolinTiger (2003, August 17). Places That Viruses And Trojans Hide On Start Up. Retrieved September 30, from http://www.governmentsecurity.org/forum/index.php?showtopic=1467

Symantec (2003, April 7). Symantec Security Response - W32.Kwbot.F.Worm. Retrieved September 30, 2005, from http://securityresponse.symantec.com/avcenter/venc/data/w32.kwbot.f.worm.html

Symantec (2004, October 19). Symantec Security Response - Trojan.Watsoon.A. Retrieved September 30, 2005, from http://securityresponse.symantec.com/avcenter/venc/data/trojan.watsoon.a.html

Symantec (2005, February 27). Symantec Security Response - W32.Zellome@m. Retrieved September 30, 2005, from http://securityresponse.symantec.com/avcenter/venc/data/w32.zellome@m.html

Tan, J. (2001, July 17). Forensic Readiness. Retrieved September 26, 2005, from http://www.atstake.com/research/reports/acrobat/atstake_forensic_readiness.pdf

UVCView (2005, June 17). UVCView – Diagnostic Tool for USB Video Class Hardware. Retrieved October 1, 2005, from http://www.microsoft.com/whdc/device/stream/vidcap/UVCView.mspx