June 2018

Windows Registry Forensic Tool Specification

Draft 2 of Version 1.0 for Public Comment

National Institute of
Standards and Technology
U.S. Department of Commerce

31

Abstract

This specification defines requirements for Windows registry forensic tools that parse the registry hive file format as well as extract interpretable data from registry hive files, and test methods used to determine whether a specific tool meets the requirements for producing accurate results. These requirements are statements used to derive test assertions that define expectations of a tool or application. Test cases describe the combination of test parameters required to test each assertion. Test assertions are described as general statements of conditions that can be checked after a test is executed. Each assertion appears in one or more test cases consisting of a test protocol and the expected test results. The test protocol specifies detailed procedures for setting up the test, executing the test, and measuring the test results. The associated assertions and test cases are defined in the test plan document entitled: *Windows Registry Forensic Tool Test Assertions and Test Plan*, located on the CFTT web site, www.cftt.nist.gov.

As this document evolves updated versions will be posted at www.cftt.nist.gov.

[*] NIST does not endorse nor recommend products or trade names identified in this paper. All products used in this paper are mentioned for use in research and testing by NIST.

48

Table of Contents

1. Introduction

There is a critical need in the law enforcement community to ensure the reliability of digital forensic tools. A capability is required to ensure that forensic software tools consistently produce accurate and objective results. The goal of the Computer Forensic Tool Testing (CFTT) project at the National Institute of Standards and Technology (NIST) is to establish a methodology for testing forensic software tools. We adhere to a disciplined testing procedure, established test criteria, test sets, and test hardware requirements, that result in providing necessary feedback information to toolmakers so they can improve their tool's effectiveness; end users benefit in that they gain vital information making them more informed about choices for acquiring and using computer forensic tools, and lastly, we impart knowledge to interested parties by increasing their understanding of a specific tool's capability. Our approach for testing forensic tools is based on established, well recognized international methodologies for conformance testing and quality testing. For more information on this project, please visit us at: www.cftt.nist.gov.

The Computer Forensics Tool Testing (CFTT) program is a joint project of the Department of Homeland Security (DHS), and the National Institute of Standards and Technology Special Program Office (SPO) and Information Technology Laboratory (ITL). CFTT is supported by other organizations, including the Federal Bureau of Investigation, the U.S. Department of Defense Cyber Crime Center, U.S. Internal Revenue Service Criminal Investigation Division Electronic Crimes Program, the National Institute of Justice (NIJ), and the U.S. Department of Homeland Security's Bureau of Immigration and Customs Enforcement, U.S. Customs and Border Protection and U.S. Secret Service. The objective of the CFTT program is to provide measurable assurance to practitioners, researchers, and other applicable users that the tools used in computer forensics investigations provide accurate results. Accomplishing this requires the development of specifications and test methods for computer forensic tools and subsequent testing of specific tools against those specifications.

The Windows registry is a system-defined database in which applications and system components store and retrieve configuration data. The Windows operating system provides registry APIs to retrieve, modify, or delete registry objects such as keys, values and data. Note that the Windows registry in this specification means Windows NT registry (i.e. not Windows 3.1 or Windows 95/98/ME).

From digital forensics point of view, the Windows registry is one of primary targets for Windows forensics as a treasure box including not only configurations of the operating system and user installed applications, but also meaningful data that can be useful for identifying users' behaviors and reconstructing their past events. Although Windows registry analysis techniques are already generally being used in Windows forensics, there is a lack of objective and scientific evaluation efforts on digital forensic tools (dedicated registry forensic tools as well as digital forensic suites having registry-related features), which can parse and interpret Windows registry internals and various traces stored within the registry.

2. Purpose

This specification defines requirements for Windows registry forensic tools that parse the registry hive file format as well as extract interpretable data from registry hive files, and test methods used to determine whether a specific tool meets the requirements for producing measurable results. These requirements were developed through a combination of processes including but not limited to Windows forensics research, personal interviews with forensic investigators, and informal discussions with individuals who are experts in the field of forensic investigation.

The Windows registry forensic tool requirements are used to derive test assertions. The test assertions are described as general statements of conditions that can be checked after a test is executed. Each assertion generates one or more test cases consisting of a test protocol and the expected test results. The test protocol specifies detailed procedures for setting up the test, executing the test, and measuring the test results.

3. Scope

The scope of this specification is limited to software tools capable of handling the Windows NT registry hive format v1.3 and v1.5 generally used in modern Windows operating systems. The Windows registry forensic tool specification is general and capable of being adapted to digital forensic suites having registry-related features as well as dedicated registry forensic tools.

The type of input data for registry-related tools may be one of the follows: hive file(s), hive set(s), and disk image file(s) containing at least one Windows system partition.

4. Definitions

This glossary provides context in the absence of definitions recognized by the digital forensics community.

Analysis – The examination of acquired data for its significance and probative value.

Artifact – An object created as a result of the use of a digital device or software that shows usage history by users and includes potential digital evidence. Thus, digital forensic activities usually handle a multitude of forensic artifacts stored within various digital data storages including volatile and non-volatile storage devices.

ASCII – American Standard Code for Information Interchange.

Examination – A technical review that makes the evidence visible and suitable for analysis; as well as tests performed on the evidence to determine the presence or absence of specific data.

Extraction – A process by which potential digital evidence is parsed, processed, or interpreted for the examination and analysis.

File system – A software mechanism that defines the way that files are named, stored, organized, and accessed on logical volumes of partitioned memory.

FILETIME – A time structure that contains a 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 (UTC).

Hive file – An offline registry file that physically stores registry objects including keys, values and data. A primary hive file may exist along with multiple transaction log files.

Hive set – A hive set consists of primary hives and their transaction log files generally including (but not limited to) SAM, SYSTEM, SOFTWARE, SECURITY and pairs of [NTUSER, USRCLASS] for each Windows account. Multiple hive sets can be found from Restore Points (Windows XP and earlier) as well as Volume Shadow Copies (Windows Vista and later) stored within a Windows system partition if relevant features are turned on.

Registry – A hierarchical database that contains data that is critical for the operation of Windows and the applications and services running on Windows.

Registry Key – An object within the registry that contains values and additional subkeys like a directory (folder) in a hierarchical file system.

Registry Value – Registry name/value pair associated with a registry key analogous to a file in a hierarchical file system.

Unicode – A standard for the consistent encoding, representation, and handling of text expressed in most of writing systems in the world (e.g., UTF-8 and UTF-16).

Volume Shadow Copy – A technology included in modern Microsoft Windows that allows taking manual or automatic backup copies of volumes, even when they are in use.

5. Background

5.1. Windows NT Registry File Format

In modern Windows systems, the registry is composed of multiple registry hives, and each registry hive that is a group of keys, subkeys and values is stored into a Windows NT registry file (also known as a hive file) as a backup container. The followings are commonly identified registry hives used in a running Windows OS[1]:

- ✓ HKEY_LOCAL_MACHINE\SAM
- ✓ HKEY_LOCAL_MACHINE\SECURITY
- ✓ HKEY_LOCAL_MACHINE\SOFTWARE
- ✓ HKEY_LOCAL_MACHINE\SYSTEM
- ✓ HKEY_CURRENT_CONFIG
- ✓ HKEY_USERS*

The Figure 1 shows the internal structure of a registry file. As depicted in the figure, a registry file consists of a base block (a header area starting with '`regf`' signature) and multiple hive bins, and more specifically each hive bin has a hive bin header (starting with '`hbin`' signature) and a multitude of hive cells. We should note, that for registry formats version 1.3 and 1.5, a hive block of 0x1000 (4,096) bytes is used as the basic unit of allocation to expand the size of a hive file.

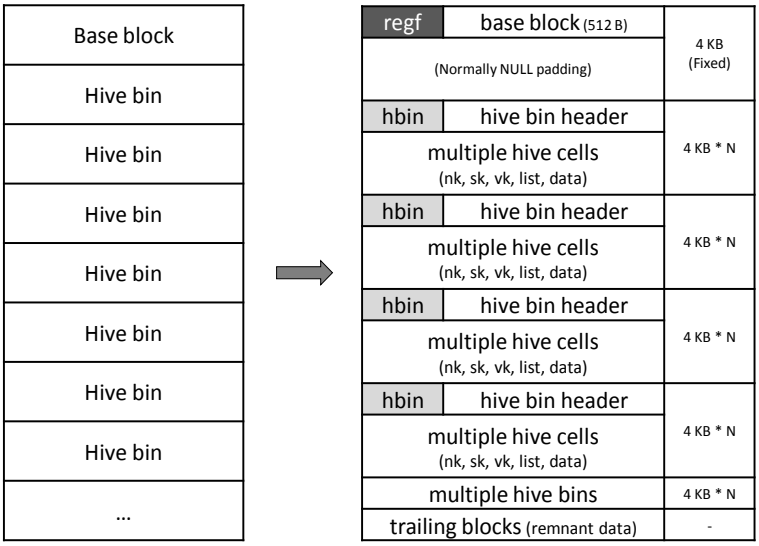

Figure 1. Windows registry file format internals

In this storage format, the hive cell structure consists of a 4-byte cell size (this value is negative if the cell is allocated or positive if it is unallocated by the deletion operation) and cell data that is one of the key node (nk), subkey list (lf, lh, ri, li), key value (vk), value list, key security (sk), and big data (db). More details about the registry file format are available in literature (Section 5.5).

[1] It should be noted that there are application hives, which do not have a specific visible mount point. In Table 1, the 'Amcache.hve' hive is an example of such a registry hive.

Forensic tools tailored for registry data extraction and analysis should minimally be able to parse registry objects (e.g., key, value and data) stored in hive files and provide reports of the data in a human-readable format. Because registry hive files as one of important investigative targets, specifically generated by modern Windows OSes, include a variety of forensically meaningful data (potential digital evidence) created during the usage of the operating systems, tools that possess Windows forensics-related features are generally required to provide examiners with the ability to perform proper interpretation of well-known registry files (e.g., hive files having accounts, applications and devices-related registry data) and generate reports in a meaningful format.

5.2. Fundamental Characteristics of Registry File Format

This specification considers the following characteristics of the registry file format. Note that there may of course exist more properties about the file format, but the following list is considered as fundamental conditions to define testing strategies for Windows registry forensic tools.

- The format uses little-endian byte ordering.
- The date and time value is stored in a FILETIME (UTC) structure[2].
- A key name has a limit of 255 characters[3].
- A value name has a limit of 16,383 characters.
- A registry tree can be 512 levels deep.
- Key and value names are case insensitive.
- Key and value names are stored either in ASCII (ISO/IEC 8859-1: Latin-1) or Unicode (UTF-16LE without the byte order mark). Note that the null (0x00) and backslash ('\', 0x5C) characters are not allowed for naming keys.

5.3. Well-known Registry Files on Windows Forensics

Tools that provide Windows forensics-related features may have the ability to recover and extract forensically meaningful artifacts stored in well-known registry files like Table 1 from Windows forensics point of view. The following list shows some examples of those kind of artifacts:

- ✓ User accounts (local and live accounts) and their activities
- ✓ System configurations
- ✓ Directories and files related traces
- ✓ System or third-party application related data
- ✓ External device usage traces
- ✓ Miscellaneous features including search, shared directory, network drive, system backup, etc.

[2] It should be noted that the last two bits of the 'last reorganized' (FILETIME) timestamp in the base block are used to encode the reorganization type.

[3] However, it is possible to store 256 characters in a key name using a Windows registry API.

Given that a Windows system partition has a set of common registry files as listed in Table 1, we should also note that multiple sets can be found from Restore Points (XP and earlier) as well as volume shadow copies (Vista and later).

Table 1. Common registry files stored in modern Windows operating systems

Hive Path (considering only Vista and later)	Description and linked paths (RegEdit.exe)
{Boot Partition}\Boot\BCD	- BCD (Boot Configuration Data) - HKEY_LOCAL_MACHINE\BCD00000000
%UserProfile%\NTUSER.DAT	- User specific data - HKEY_USERS\<SID>
%UserProfile%\AppData\Local\Microsoft\Windows\UsrClass.dat	- File associations and COM registry entries - HKEY_USERS\<SID>_Classes
%SystemRoot%\AppCompat\Programs\Amcache.hve	- Application experience and compatibility data - Windows 7 and later[4]
%SystemRoot%\ServiceProfiles\LocalService\NTUSER.DAT	- 'Local Service' account (SID: S-1-5-19) - HKEY_USERS\S-1-5-19
%SystemRoot%\ServiceProfiles\NetworkService\NTUSER.DAT	- 'Network Service' account (SID: S-1-5-20) - HKEY_USERS\S-1-5-20
%SystemRoot%\System32\Config\BBI	- BBI (Browser-Based Interface) - Windows 8 and later
%SystemRoot%\System32\Config\BCD-Template	- Template file for BCD registry - Windows 8 and later
%SystemRoot%\System32\Config\COMPONENTS	- Windows optional components related data - HKEY_LOCAL_MACHINE\COMPONENTS
%SystemRoot%\System32\Config\DEFAULT	- 'Local System' account (SID: S-1-5-18) - HKEY_USERS\S-1-5-18 - HKEY_USERS\.DEFAULT
%SystemRoot%\System32\Config\DRIVER	- Driver database - Windows 8 and later
%SystemRoot%\System32\Config\ELAM	- ELAM (Early Launch Anti-Malware) - Windows 8 and later
%SystemRoot%\System32\Config\SAM	- SAM (Security Account Manager) part - HKEY_LOCAL_MACHINE\SAM
%SystemRoot%\System32\Config\SECURITY	- Security specific data - HKEY_LOCAL_MACHINE\SECURITY
%SystemRoot%\System32\Config\SOFTWARE	- Software specific data - HKEY_LOCAL_MACHINE\SOFTWARE
%SystemRoot%\System32\Config\SYSTEM	- System specific data - HKEY_LOCAL_MACHINE\SYSTEM
%SystemRoot%\System32\Config\RegBack\	- A directory containing backup copies of some primary hive files
%SystemRoot%\System32\SMI\Store\Machine\SCHEMA.DAT	- SMI (Settings Management Infrastructure) - HKEY_LOCAL_MACHINE\SCHEMA
%SystemDrive%\System Volume Information\Syscache.hve	- volume shadow copies related data - Windows 7 and later

[4] The 'Amcache.hve' is a hive file introduced in Windows 8, but this file is also available in Windows 7 updated with latest patches.

5.4. Transaction Log

Registry hives can consist of primary hive files, transaction log files and transactional registry (TxR) files[5]. The transaction log files (.LOG, .LOG1 and .LOG2) are used to perform fault-tolerant write operations to primary files. Before writing modified (dirty) blocks to a primary file, the Windows registry handler will write those data into a transaction log file. With this transaction log file, if an exception occurs when writing to a primary file, the log file will be used to recover it.

In modern Windows systems, there are two formats for storing registry transaction logs: a legacy (old) format and an incremental (new) format. According to literature, when the incremental log is used, a kernel may delay writing to a primary file up to an hour. In addition, because the kernel timer is paused when a system is hibernated, modifications to a primary file may only remain in transaction log files for multiple days.

5.5. References

It is important to note that these references are primarily informative:

H. Carvey – Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry.

J. Metz – Windows NT Registry File format specification. [Online].
Available: https://github.com/libyal/libregf/tree/master/documentation

J. Thomassen – Forensic Analysis of Unallocated Space in Windows Registry Hive Files.
Available: http://www.sentinelchicken.com/research/thomassen_registry_unallocated_space/

M. Suhanov – Windows registry file format specification. [Online].
Available: https://github.com/msuhanov/regf

Microsoft – Windows registry information for advanced users. [Online].
Available: https://support.microsoft.com/en-us/kb/256986

P. Norris – The Internal Structure of the Windows Registry.
Available: http://amnesia.gtisc.gatech.edu/~moyix/suzibandit.ltd.uk/MSc/

T. D. Morgan – The Windows NT Registry File Format (Version 0.4).
Available: http://www.sentinelchicken.com/data/TheWindowsNTRegistryFileFormat.pdf

[5] The transactional registry (TxR) is a feature that allows an application to accumulate multiple modifications within a transaction, which can be committed or rolled back. The TxR is similar to the transactional NTFS (TxF) and also uses the Common Log File System (CLFS) as its format. The TxR logs will be created when an application uses specific registry APIs for a transacted operation, such as `RegOpenKeyTransacted`, `RegCreateKeyTransacted` and `RegDeleteKeyTransacted`.

6. Test Methodology

To provide repeatable test results, the following test methodology is strictly followed. Each forensic application under evaluation is installed on a host workstation operating with the required platform as specified by the application. Additionally, a Windows registry dataset developed by the Computer Forensic Reference Data Sets (CFReDS) project at the NIST is used as a common reference dataset with ground truth data during the tool testing procedure. Briefly, the dataset used here consists of two different classes: *user-generated data* that is specially crafted based on the Windows NT registry file format, and *system-generated data* that is generated naturally by Windows operating systems populated along with a multitude of known user actions. The data objects and characteristics described in Section 5 were considered in developing the Windows registry dataset. For more information on this test dataset, please visit us at: www.cfreds.nist.gov.

7. Requirements

The Windows registry tool requirements[6] are in two sections: 7.1 and 7.2. The first Section 7.1 lists requirements, i.e., Windows Registry Tool-Core Requirement-01, WRT-CR-01 through WRT-CR-03 that all tools shall meet. Section 7.2 lists requirements i.e., Windows Registry Tool-Requirement Optional-01, WRT-RO-01 through WRT-RO-02 that the tool shall meet on the condition that specified features or options are offered by the tool. If a feature is not present, then requirements for those features will not be tested.

7.1. Requirements for Core Features

All Windows registry forensic tools shall meet the following core requirements.

WRT-CR-01 A Windows registry forensic tool shall support at least one of possible input data types, which include an independent hive file, a set of hive files, and a disk image containing Windows system partitions.

WRT-CR-02 A Windows registry forensic tool shall have the ability to notify the user of abnormal information (that can usually be found in corrupted or manipulated registry hive files) detected during data processing without application crash.

WRT-CR-03 A Windows registry forensic tool shall have the ability to perform an interpretation of supported registry objects without modification to the objects.

[6] It should be noted that the transaction log file (Section 5.4) is not considered in this version of tool testing. Given the fact that there is a proliferation of Windows 10 as well as forensic tools for supporting transaction log files have appeared, it will be included in the next phase of this work.

7.2. Requirements for Optional Features

The following Windows registry forensic tool requirements define optional tool features. If a tool provides the capability defined, the tool is tested for conformance to these requirements. If the tool does not provide the capability defined, the requirement does not apply.

The following optional features are identified:

- Deleted registry object recovery
- Registry forensic artifact extraction

WRT-RO-01 A Windows registry forensic tool shall have the ability to identify and recover deleted registry objects such as keys, values and their data from supported registry hive files.

WRT-RO-02 A Windows registry forensic tool shall have the ability to extract registry forensic artifacts.

8. History

Rev	Issue Date	Section	History
1.0 draft 1	2018-03-14	All	- The first release for public comments
1.0 draft 2	2018-06-25	4	- Updated several definitions
		5	- Updated and corrected information - Added 'Transaction Log' section - Updated 'References' section
		7	- Added a footnote about the transaction log file