

Forensic Toolkit (FTK) 7.0.0.163, Registry Viewer 2.0.0.7

Test Results for Windows Registry Forensic Tool

April 07, 2019



**Homeland
Security**

Science and Technology

This report was prepared for the Department of Homeland Security (DHS) Science and Technology Directorate (S&T) by the Office of Law Enforcement Standards of the National Institute of Standards and Technology.

For additional information about ongoing DHS S&T cybersecurity projects, please visit <https://www.dhs.gov/science-and-technology/cyber-security-division>.

April 2019

**Test Results for Windows Registry Forensic Tool:
Forensic Toolkit (FTK) 7.0.0.163, Registry Viewer
2.0.0.7**

Table of Contents

Introduction.....	1
How to Read This Report.....	1
Test Results for Windows Registry Forensic Tool	2
1. Results Summary.....	2
2. Test Environment and Selected Cases.....	4
2.1. Execution Environment.....	4
2.2. Test Dataset.....	4
2.3. Test Case Selection.....	8
3. Test Results	11
3.1. Results on Core Features.....	12
3.2. Results on Recovering Deleted Registry Objects.....	17
3.3. Results on Extracting Windows Forensic Artifacts	18

Introduction

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.

Test results provide the information necessary for developers to improve tools, users to make informed choices, and the legal community and others to understand the tools' capabilities. The CFTT approach to testing computer forensics tools is based on well recognized methodologies for conformance and quality testing. Interested parties in the computer forensics community can review and comment on the specifications and test methods posted on the [CFTT Website](#).

This document reports the results from testing FTK 7.0.0.163 against a registry dataset that consists of various Windows NT registry hive files. The dataset is available at the [CFReDS website](#).

Test results from other tools can be found on the DHS S&T-sponsored [digital forensics web page](#).

How to Read This Report

This report is divided into four sections. Section 1 identifies and provides a summary of any significant anomalies observed in the test runs. This section is sufficient for most readers to assess the suitability of the tool for the intended use. Section 2.1 and 2.2 list a testing environment and test data (a registry dataset) prepared for measuring the success of each test. Section 2.3 identifies the test cases that were selected. The test cases are selected, in general, based on features offered by the tool. Section 3 provides an overview of the test case results reported by the tool.

¹NIST 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.

Test Results for Windows Registry Forensic Tool

Tool Tested: Forensic Toolkit (FTK)
Software Version: FTK 7.0.0.163, Registry Viewer 2.0.0.7
Supplier: AccessData
Address: 588 West 400 South Suite 350
Lindon, UT 84042
USA
Tel: 1-801-377-5410
WWW: <https://www.accessdata.com>

1. Results Summary

Below is a comprehensive summary on how FTK 7.0.0.163 and Registry Viewer 2.0.0.7 performed when processing hive files of a Windows registry dataset. Except for the following anomalies, the tool processed and extracted all supported data completely and accurately for all registry hive files tested.

Results on Core Features:

- The tool incorrectly reported a QWORD value.
- The tool did not process hive files generated by *hivex* library.
- The tool did not report several big-data values in a v1.5 hive file.

Results on Extracting Windows Forensic Artifacts:

- Account
 - Partial account related data was reported. [Windows: 10RS1]
 - The FTK reported user names only. It did not identify hash values.
- Application
 - Partial application related data was reported. [Windows: 10, 10RS1]
 - 32-bit (x86) application related data in 64-bit Windows systems was not reported.
- Auto Run
 - Partial auto run related data was reported. [Windows: 10, 10RS1]
 - 32-bit (x86) application related data in 64-bit Windows systems was not reported.

- Dialog Usage
 - Dialog usage related data was not reported. [Windows: 10, 10RS1]
- Network Drive
 - Network drive related data was not reported. [Windows: 10, 10RS1]
- Recently Opened File and Directory
 - Opened file and directory related data was not reported. [Windows: 10, 10RS1]
- Run Command
 - Run command related data was not reported. [Windows: 10, 10RS1]
- ShellBag
 - Partial ShellBag related data was reported. [Windows: ALL²]
 - The FTK reported ShellBag related data in USRCLASS hives, but those data in NTUSER hives were not reported. [Windows: 10, 10RS1]
 - The FTK did not report MRU (Most Recently Used) timestamps.
 - The FTK incorrectly reported directory paths relating to network drives.
 - The Registry Viewer reported invalid parsing results.
- UserAssist
 - UserAssist related data was not reported. [Windows: 10, 10RS1]
 - Partial UserAssist related data was reported. [Windows: 7, 8, 8.1]
 - The FTK only reported values having a 'run count' greater than 1.

For more test results, see section 3.

² 'Windows: ALL' includes Vista, 7, 8, 8.1, 10 and 10RS1. (refer to Section 2.2)

2. Test Environment and Selected Cases

This section describes the test execution environment, test dataset and test cases.

2.1. Execution Environment

FTK 7.0.0.163 was installed on Windows 7 Enterprise v6.1.7601 (x64, English).

2.2. Test Dataset

The tool was measured by analyzing interpreted and extracted data from various registry hive files developed as a reference dataset. Table 1, Table 2 and Table 3 list data codes that are linked to registry files for testing core features and an optional feature relating to recovering deleted registry objects. In addition, well-known registry hive files from reference Windows systems with ground truth data were used to test an optional feature on extracting Windows registry forensic artifacts. In that regard, Table 4 defines several artifact groups considered for populating the reference Windows systems (Vista, 7, 8, 8.1, 10 and 10RS1) to limit the scope of tool testing.

For this tool testing, registry hive files from the last volume shadow copy (VSC) of each Windows system were processed by the tool, since the last VSC includes the greatest number of artifacts before performing anti-forensic activities to remove usage history. For more information, the dataset and related documents can be obtained from: www.cfreds.nist.gov.

Table 1. Dataset for Testing Core Features

Category	Code	Description	Generation method
Normal Registry Hive File	NR-01-1	Possible data types	[Windows] API (.REG)
	NR-01-2	Possible data types	[Linux] <i>hivex</i> library
	NR-02-1	Simple tree structure	[Windows] API (.REG)
	NR-02-2	Simple tree structure	[Linux] <i>hivex</i> library
	NR-03-1	Tree structure with the maximum levels (512)	[Windows] API (.REG)
	NR-03-2	Tree structure with the maximum levels (512)	[Linux] <i>hivex</i> library
	NR-03-3	Tree structure with abnormal levels (1 million)	[Linux] <i>hivex</i> library
	NR-04-1	Maximum key name length (255 and 256)	[Windows] API (.REG)
	NR-04-2	Maximum key name length (255 and 256)	[Linux] <i>hivex</i> library
	NR-04-3	Maximum key name length beyond limitation	[Linux] <i>hivex</i> library
	NR-05-1	Maximum value name length (16,383)	[Windows] API (.REG)
	NR-05-2	Maximum value name length (16,383)	[Linux] <i>hivex</i> library
	NR-05-3	Maximum value name length beyond limitation	[Linux] <i>hivex</i> library
	NR-06-1	Big data (16,344 or more)	[Windows] API (.REG)
	NR-06-2	Big data (16,344 or more)	[Linux] <i>hivex</i> library
	NR-07-1	Non-ASCII characters	[Windows] API (.REG)
NR-07-2	Non-ASCII characters	[Linux] <i>hivex</i> library	

Category	Code	Description	Generation method
	NR-08	Naming convention	[Windows] API (Python)
Corrupted Registry Hive File	CR-01	A hive bin with Root key	Python script
	CR-02	A hive bin	Python script
	CR-03	Last half	Python script
	CR-04	Multiple fragments with hbin header	Python script
	CR-05	Base block	Python script

Table 2. Dataset for Testing Core Features (continue)

Category	Code	Description	Manipulation point
Manipulated Registry Hive File	MR-01-1	Hide a root key (invalid checksum)	'root cell offset' in the base block
	MR-01-2	Hide a root key (valid checksum)	'root cell offset' in the base block
	MR-02-1	Hide key names	'key name size' in the key (nk) cell
	MR-02-2	Hide key names	'key cell size' in the key (nk) cell
	MR-03-1	Hide subkeys of a key	'number of subkeys' in the key (nk) cell
	MR-03-2	Hide subkeys of a key	'subkey-list cell size' in the key (nk) cell
	MR-03-3	Hide subkeys of a key	'number of subkeys' in the subkey-list cell
	MR-03-4	Hide subkeys of a key	'subkey offset' items in the subkey-list cell
	MR-04-1	Hide values of a key	'number of values' in the key (nk) cell
	MR-04-2	Hide values of a key	'value-list cell size' in the value-list cell
	MR-04-3	Hide values of a key	'value offset' items in the value-list cell
	MR-05-1	Hide value names	'value name size' in the value (vk) cell
	MR-05-2	Hide value names	'value cell size' in the value (vk) cell
	MR-06-1	Hide data of a value	'data size' in the value (vk) cell
	MR-06-2	Hide data of a value	'data cell size' in the data cell
	MR-06-3	Hide data of a value	'data offset' in the value (vk) cell
	MR-06-4	Hide data of a value	'data type' in the value (vk) cell
	MR-07	Hide big data of a value	'data size' in the value (vk) cell
	MR-08	Infinite key loop	'subkey offset' in the subkey-list cell
	MR-09	Invalid integer data size	'data size' in the value (vk) cell
MR-10	Invalid binary data size	'data size' in the value (vk) cell	
MR-11	Invalid string data size	'data size' in the value (vk) cell	
MR-12	Version mismatch (big data processing)	'minor version value' in the base block	
MR-13	Ambiguous key name	'encoding flag' in the key (nk) cell	
MR-14	Ambiguous value name	'encoding flag' in the value (vk) cell	
MR-15	Ambiguous encodings	text encoded by various encoding standards	

Table 3. Dataset for Testing an Optional Feature: Recovering Deleted Registry Objects

Category	Code	Description	Generation method
Normal Registry Hive File with Deleted Registry Data	NRD-01-1	Delete keys with values, but without subkeys	[Windows] API (.REG)
	NRD-01-2	Delete keys with values, but without subkeys	[Linux] <i>hivex</i> library
	NRD-02-1	Delete a key with values and subkeys	[Windows] API (.REG)
	NRD-02-2	Delete a key with values and subkeys	[Linux] <i>hivex</i> library
	NRD-03-1	Delete a key without values and subkeys	[Windows] API (.REG)
	NRD-03-2	Delete a key without values and subkeys	[Linux] <i>hivex</i> library
	NRD-04	Delete a value with normal data	[Windows] API (.REG)
	NRD-05	Delete a value with big data	[Windows] API (.REG)
	NRD-06	Delete multiple values in a key	[Windows] API (.REG)

Table 4. Artifacts considered for Testing an Optional Feature: Extracting Forensic Artifacts

Windows	Artifact group	Details (D: description, C: check points, R: related paths) * The paths (R) show a few examples although there may exist other paths. Note that 'Wow6432Node' key should be also considered on 64-bit Windows systems.	
Vista+ The '+' symbol signifies later versions.	Account	D	Accounts
		C	Name, type, login count, timestamps (login, pw reset, failed), etc.
		R	SAM\SAM\Domains\Account\Users\ SAM\SAM\Domains\Builtin\Aliases\ SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\ SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\
	Application	D	Installed programs
		C	Name, vendor, version, installed path, timestamp, etc.
		R	SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\ SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\?SID?\Products\ SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\ SOFTWARE\Classes\Installer\Products\ USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\Repository\
	Application Experience & Compatibility (Shimcache)	D	Windows Application Compatibility related data
		C	File name, file size, timestamp, etc.
		R	SYSTEM\?ControlSet?\Control\Session Manager\AppCompatCache\
	Auto Run	D	Programs that start automatically when a user logs on
		C	Name, executable path, timestamp, etc.
		R	NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Run\ NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\RunOnce\ SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\
	Dialog Usage	D	Dialog box related user actions
		C	Name, timestamps, etc.
		R	NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidIMRU\ NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidIMRU\
	External Device	D	External devices (like USB storages) attached into the system
		C	Vendor, product, serial number, connected date, drive letter, etc.
		R	SYSTEM\MountedDevices\ SYSTEM\?ControlSet?\Control\DeviceClasses\ SYSTEM\?ControlSet?\Control\DeviceContainers\ SYSTEM\?ControlSet?\Enum\ SOFTWARE\Microsoft\WindowsNT\CurrentVersion\EMDMgmt\ SOFTWARE\Microsoft\Windows Portable Devices\Devices\

Windows	Artifact group	Details (D: description, C: check points, R: related paths) * The paths (R) show a few examples although there may exist other paths. Note that 'Wow6432Node' key should be also considered on 64-bit Windows systems.	
			NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\
	Network Connection	D	Configurations of interface cards and network connection history
		C	Name, IP, gateway, MAC, SSID, DNS, etc.
		R	SYSTEM\?ControlSet?\Services\Tcpip\Parameters\Interfaces\ SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkCards\ SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\ SOFTWARE\Microsoft\WZCSVC\Parameters\Interfaces\
	Network Drive	D	Network connection history to external systems
		C	Name, IP, account drive letter, type, timestamp, etc.
		R	NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU\ NTUSER.DAT\Network\
	OS Information	D	Installed OS (Windows) information
		C	Version, install date, computer name, owner, shutdown time, etc.
		R	SOFTWARE\Microsoft\Windows NT\CurrentVersion\ SYSTEM\?ControlSet?\Control\Windows\ SYSTEM\?ControlSet?\Control\ComputerName\
	Recently Opened File and Directory	D	Recently opened files and directories
		C	Name, timestamp, etc.
		R	NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\ NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Applets\?APP_NAME?\Recent File List\ NTUSER.DAT\Software\Microsoft\MediaPlayer\Player\RecentFileList\ NTUSER.DAT\Software\Microsoft\Office\?VERSION?\?APP_NAME?\User MRU\ NTUSER.DAT\Software\Adobe\Acrobat Reader\?VERSION?\AVGeneral\cRecentFiles\ NTUSER.DAT\Software\Adobe\Acrobat Reader\?VERSION?\AVGeneral\cRecentFolders\
	Remote Desktop	D	Network connection history to external systems
		C	IP, account ID, timestamp, etc.
		R	NTUSER.DAT\Software\Microsoft\Terminal Server Client\Default\ NTUSER.DAT\Software\Microsoft\Terminal Server Client\Servers?\IP?\
	Run Command History	D	Recently used commands from Windows Run
		C	Command, timestamp, etc.
		R	NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU\
	Service and Driver	D	Service and driver list
		C	Display name, description, type, start, image path, etc.
		R	SYSTEM\?ControlSet?\Services\?NAME?\
	Shared Directory	D	Shared directory list
		C	Name, directory path, type, timestamp, etc.
		R	SYSTEM\?ControlSet?\Services\LanmanServer\Shares\
	ShellBag	D	Directories or files accessed by each user account (Database to track user's window viewing preferences)
		C	Directory or file path, timestamp, etc.
		R	NTUSER.DAT\Software\Microsoft\Windows\Shell\Bags\ NTUSER.DAT\Software\Microsoft\Windows\Shell\BagMRU\ NTUSER.DAT\Software\Microsoft\Windows\ShellNoRoam\Bags\ NTUSER.DAT\Software\Microsoft\Windows\ShellNoRoam\BagMRU\ USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\Shell\Bags\ USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\ USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\ShellNoRoam\Bags\ USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\ShellNoRoam\BagMRU\

Windows	Artifact group	Details (D: description, C: check points, R: related paths) * The paths (R) show a few examples although there may exist other paths. Note that 'Wow6432Node' key should be also considered on 64-bit Windows systems.	
	Timezone	D	Timezone information
		C	Timezone name, time offset, etc.
		R	SYSTEM\?ControlSet?\Control\TimeZoneInformation\
	UserAssist	D	Programs executed by each user account (executable and link files)
		C	Account, file name, run count, timestamp, etc.
		R	NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\
Win 7 and Win 8	Search	D	Search history using Windows Search feature
		C	Search keyword, timestamp, etc.
		R	Win 7: NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery\ Win 8: NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\SearchHistory\Microsoft.Windows.FileSearchApp\ (Vista, 8.1 and 10 does not save search keywords into the registry.)
Win 7+	Application Experience & Compatibility (Amcache)	D	Windows Application Compatibility related data
		C	App name, executable path, hash value, timestamp, etc.
		R	Amcache.hve\Root\File\?VOLUME_GUID?\ Amcache.hve\Root\Programs\?PROGRAM_ID?\

2.3. Test Case Selection

FTK 7.0.0.163 as an integrated multi-functional forensic toolkit does support built-in features to process the Windows registry hive format. The tool provides a registry hive file handler through an independent tool known as 'Registry Viewer'. In addition, the 'Evidence Processing' feature of the tool also provides several options for extracting various forensic artifacts from well-known registry hive files. Therefore, the selected test cases are:

- Core features
 - Processing normal registry hive files
 - Processing corrupted registry hive files
 - Processing manipulated registry hive files
- Optional features
 - Extracting Windows registry forensic artifacts

The following tables give a brief description of available test cases in the dataset. Not all test cases are used for this tool testing.

Table 5. Test Cases for Testing Core Features

Test case ID	Case description
WRT-TC-NR-01-1	◦ Process a primary file containing values with various data types (total 12) ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NR-01-2	
WRT-TC-NR-02-1	◦ Process a primary file containing a simple tree structure ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NR-02-2	
WRT-TC-NR-03-1	◦ Process a primary file containing an experimental tree structure that is 512 or more levels deep ◦ Note that 3 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NR-03-2	
WRT-TC-NR-03-3	

Test case ID	Case description
WRT-TC-NR-04-1	<ul style="list-style-type: none"> ◦ Process a primary file containing keys with long names (255 or more bytes) ◦ Note that 3 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NR-04-2	
WRT-TC-NR-04-3	
WRT-TC-NR-05-1	<ul style="list-style-type: none"> ◦ Process a primary file containing values with long names (16,383 or more bytes) ◦ Note that 3 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NR-05-2	
WRT-TC-NR-05-3	
WRT-TC-NR-06-1	<ul style="list-style-type: none"> ◦ Process a primary file containing values with big data (> 16,344 bytes) ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NR-06-2	
WRT-TC-NR-07-1	<ul style="list-style-type: none"> ◦ Process a primary file containing keys and values with non-ASCII characters ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NR-07-2	
WRT-TC-NR-08	<ul style="list-style-type: none"> ◦ Process a primary file containing keys and values with unusual (but valid) names
WRT-TC-CR-01	<ul style="list-style-type: none"> ◦ Process a corrupted primary file that contains a wiped hive bin (having root key)
WRT-TC-CR-02	<ul style="list-style-type: none"> ◦ Process a corrupted primary file that contains a wiped hive bin (randomly selected)
WRT-TC-CR-03	<ul style="list-style-type: none"> ◦ Process a corrupted primary file that contains wiped hive bins (last half)
WRT-TC-CR-04	<ul style="list-style-type: none"> ◦ Process a corrupted primary file that contains wiped multiple blocks (randomly selected among blocks having the hbin header structure)
WRT-TC-CR-05	<ul style="list-style-type: none"> ◦ Process a corrupted primary file that contains a wiped base block
WRT-TC-MR-01-1	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains hidden keys ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-MR-01-2	
WRT-TC-MR-02-1	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains hidden key names ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-MR-02-2	
WRT-TC-MR-03-1	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains hidden subkeys ◦ Note that 4 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-MR-03-2	
WRT-TC-MR-03-3	
WRT-TC-MR-03-4	
WRT-TC-MR-04-1	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains hidden values ◦ Note that 3 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-MR-04-2	
WRT-TC-MR-04-3	
WRT-TC-MR-05-1	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains hidden value names ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-MR-05-2	
WRT-TC-MR-06-1	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains hidden data ◦ Note that 4 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-MR-06-2	
WRT-TC-MR-06-3	
WRT-TC-MR-06-4	
WRT-TC-MR-07	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains hidden big data
WRT-TC-MR-08	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains an infinite key loop
WRT-TC-MR-09	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains an invalid integer data size
WRT-TC-MR-10	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains an invalid binary data size
WRT-TC-MR-11	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains an invalid string data size
WRT-TC-MR-12	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains a mismatched version indicator (focusing on big data processing)
WRT-TC-MR-13	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains a mismatched key name encoding flag
WRT-TC-MR-14	<ul style="list-style-type: none"> ◦ Process a manipulated primary file that contains a mismatched value name encoding flag

Test case ID	Case description
WRT-TC-MR-15	◦ Process a manipulated primary file that contains key names, value names and data encoded by unsupported encoding standards

Table 6. Test Cases for Testing Optional Features: Recovering Deleted Registry

Test case ID	Case description
WRT-TC-NRD-01-1	◦ Process a primary file that contains deleted keys with values but without subkeys ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NRD-01-2	
WRT-TC-NRD-02-1	◦ Process a primary file that contains a deleted key with values and subkeys ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NRD-02-2	
WRT-TC-NRD-03-1	◦ Process a primary file that contains a deleted key without values and subkeys ◦ Note that 2 subcases were defined based on the dataset (see Section 2.2)
WRT-TC-NRD-03-2	
WRT-TC-NRD-04	◦ Process a primary file that contains a deleted value with data
WRT-TC-NRD-05	◦ Process a primary file that contains a deleted value with big data
WRT-TC-NRD-06	◦ Process a primary file that contains deleted multiple values in a key

Table 7. Test Cases for Testing Optional Features: Extracting Forensic Artifacts

Test case ID	Case description
WRT-TC-FA-01	◦ Process primary files containing Account related data
WRT-TC-FA-02	◦ Process primary files containing Application related data
WRT-TC-FA-03	◦ Process primary files containing Application Compatibility (Amcache) data
WRT-TC-FA-04	◦ Process primary files containing Application Compatibility (Shimcache) data
WRT-TC-FA-05	◦ Process primary files containing Auto Run related data
WRT-TC-FA-06	◦ Process primary files containing Dialog Usage related data
WRT-TC-FA-07	◦ Process primary files containing External Device related data
WRT-TC-FA-08	◦ Process primary files containing Network Connection related data
WRT-TC-FA-09	◦ Process primary files containing Network Drive related data
WRT-TC-FA-10	◦ Process primary files containing OS Information related data
WRT-TC-FA-11	◦ Process primary files containing Recently Opened File and Directory related data
WRT-TC-FA-12	◦ Process primary files containing Remote Desktop related data
WRT-TC-FA-13	◦ Process primary files containing Run Command related data
WRT-TC-FA-14	◦ Process primary files containing Search related data
WRT-TC-FA-15	◦ Process primary files containing Service and Driver related data
WRT-TC-FA-16	◦ Process primary files containing Shared Directory related data
WRT-TC-FA-17	◦ Process primary files containing ShellBag related data
WRT-TC-FA-18	◦ Process primary files containing Timezone related data
WRT-TC-FA-19	◦ Process primary files containing UserAssist related data

NOTES:

- Some test cases are for specific tool features.
- Test cases ‘WRT-TC-NRD-*’ were not used, because FTK 7.0.0.163 and Registry Viewer 2.0.0.7 did not support recovering deleted registry objects.
- The ‘WRT-TC’ prefix (that means Windows Registry Tool – Test Case) will be omitted for simplicity in the remainder of this report.

3. Test Results

This section provides the test results reported by FTK 7.0.0.163 and Registry Viewer 2.0.0.7. The results are as follows:

AS Expected	The Windows registry forensic tool returned expected test results – the tool processed and reported data from registry hive files successfully.
Partial	The Windows registry forensic tool returned some of data from registry hive files.
Observed	<p>The Windows registry forensic tool returned some of data from ‘corrupted’ or ‘manipulated’ registry hive files without any error or crash. This type of test results is not subject to selection between ‘As Expected’ and ‘Partial’. Instead, each result describes detailed behaviors when processing abnormal (or experimental) hive files.</p> <p><i>Note: The observed results may be useful to understand and improve each tool’s registry processing algorithms, which include format validation, data interpretation, anomaly detection and exception handling.</i></p>
Not As Expected	The Windows registry forensic tool failed to return expected test results – the tool did not process or report supported data from registry hive files properly.
Not Applicable (NA)	The Windows registry forensic tool does not provide support for a particular test case.

3.1. Results on Core Features

The reference registry hive files were processed with Registry Viewer 2.0.0.7.

All test cases were successful except for the following.

- The tool incorrectly reported a QWORD value. (NR-01-1)
- The tool did not process hive files generated by *hivex* library. (NR-01-2, NR-02-2, NR-03-2, NR-04-2, NR-04-3, NR-05-2, NR-05-3, NR-06-2 and NR-07-2)
- The tool did not report several big-data values in a v1.5 hive file of NR-06-1.

NOTES:

- Detailed observation results when the tool processed the ‘corrupted’ and ‘manipulated’ hive files are available in Table 8.

See Table 8 below for more details.

Table 8. Test Results on Core Features

Registry Viewer 2.0.0.7		
Test case	Test result	Note
NR-01-1	<i>Partial</i>	The tool incorrectly reported a QWORD value. For example, a QWORD value stores ‘8’, but the tool interpreted the data as ‘34359738368’.
NR-01-2	<i>Not As Expected</i>	The tool didn’t return any result.
NR-02-1	<i>As Expected</i>	-
NR-02-2	<i>Not As Expected</i>	The tool didn’t return any result.
NR-03-1	<i>As Expected</i>	-
NR-03-2	<i>Not As Expected</i>	The tool didn’t return any result.
NR-03-3	<i>Not As Expected</i>	The tool didn’t return any result. Note that this reference file is not a normal but experimental hive file.
NR-04-1	<i>As Expected</i>	-
NR-04-2	<i>Not As Expected</i>	The tool didn’t return any result.
NR-04-3	<i>Not As Expected</i>	The tool didn’t return any result.
NR-05-1	<i>As Expected</i>	-

Registry Viewer 2.0.0.7		
Test case	Test result	Note
NR-05-2	<i>Not As Expected</i>	The tool didn't return any result.
NR-05-3	<i>Not As Expected</i>	The tool didn't return any result.
NR-06-1	<i>Partial</i>	When processing a v1.3 hive file, the tool identified all valid values and their raw data. However, when a v1.5 hive file, the tool did not identify several values including 'BINARY 16345', 'BINARY 20440' and 'BINARY 32688'. In addition, the tool identified 'BINARY 1MB-4' and 'BINARY 1MB-3' values, but it didn't report their raw data.
NR-06-2	<i>Not As Expected</i>	The tool didn't return any result.
NR-07-1	<i>As Expected</i>	-
NR-07-2	<i>Not As Expected</i>	The tool didn't return any result.
NR-08	<i>As Expected</i>	Note that this test case was tested through comparing with outputs from Windows (RegEdit.exe).
CR-01	<i>Observed</i>	The tool didn't return any result with the following message: "ERROR: Could not open file: ..."
CR-02	<i>Observed</i>	The tool processed (probably ignored) corrupted registry objects without error, and it reported recoverable (accessible) keys and values. However, there were no notifications about abnormal data.
CR-03	<i>Observed</i>	When processing a v1.3 hive file, the tool didn't return any result. On the other hand, when processing a v1.5 hive file, the tool processed (probably ignored) corrupted registry objects without error, and it reported recoverable (accessible) keys and values. However, there were no notifications about abnormal data.
CR-04	<i>Observed</i>	The tool processed (probably ignored) corrupted registry objects without error, and it reported recoverable (accessible) keys and values. However, there were no notifications about abnormal data.
CR-05	<i>NA</i>	The tool does not support this kind of corruption.
MR-01-1	<i>Observed</i>	The tool identified a key (the 1st subkey of the original root key) due to the edited 'root cell offset' in a base block. Although many allocated cells still exist, the tool didn't identify them. In addition, the tool does not validate a checksum value in the base block.
MR-01-2	<i>Observed</i>	The tool identified a key (the 1st subkey of the original root key) due to the edited 'root cell offset' in a base block. Although many allocated cells still exist, the tool didn't identify them.
MR-02-1	<i>Observed</i>	The tool identified a partial key name due to the edited 'key name size' in a key (nk) cell.
MR-02-2	<i>Observed</i>	The tool did not identify a key (0x01_TYPE1_DATA-TYPES) due to the edited 'key cell size' in a key (nk) cell. That is, the tool considers 'cell size' as a key factor for parsing.

Registry Viewer 2.0.0.7		
Test case	Test result	Note
MR-03-1	<i>Observed</i>	The tool did not identify keys (0 of 7) due to the adjusted 'number of subkeys' in a key (nk) cell. That is, the tool considers 'number of subkeys' in the key cell as a key factor for parsing.
MR-03-2	<i>Observed</i>	The tool did not identify keys (0 of 7) due to the edited 'subkey-list cell size' in a subkey-list cell. That is, the tool considers 'cell size' as a key factor for parsing.
MR-03-3	<i>Observed</i>	The tool did not identify keys (0 of 7) due to the adjusted 'number of subkeys' in a subkey-list cell. That is, the tool considers 'number of subkeys' in the subkey-list cell as a key factor for parsing.
MR-03-4	<i>Observed</i>	The tool identified partial keys (3 of 7) due to zeroized 'subkey offset' items in a subkey-list cell.
MR-04-1	<i>Observed</i>	The tool identified partial values (8 of 12) due to the adjusted 'number of values' in a key (nk) cell.
MR-04-2	<i>Observed</i>	The tool did not identify values (0 of 12) due to the edited 'value-list cell size' in a value-list cell. That is, the tool considers 'cell size' as a key factor for parsing.
MR-04-3	<i>Observed</i>	The tool identified partial values (8 of 12) due to zeroized 'value offset' items in a value-list cell.
MR-05-1	<i>Observed</i>	The tool identified a partial value name (VALUE 0x0) due to the edited 'value name size' (a half of the original size) in a value (vk) cell.
MR-05-2	<i>Observed</i>	The tool did not identify a value (VALUE 0x00 (NONE)) due to the edited 'value cell size' in a value (vk) cell. That is, the tool considers 'cell size' as a key factor for parsing. In addition, the tool failed to report types and data of other valid values.
MR-06-1	<i>Observed</i>	The tool didn't identify a data stream due to the NULL 'data size' in a value (vk) cell.
MR-06-2	<i>Observed</i>	The tool did not identify a value (VALUE 0x03 (BINARY)) due to the edited 'data cell size' in a data cell. That is, the tool considers 'cell size' as a key factor for parsing. In addition, the tool failed to report types and data of other valid values.

Registry Viewer 2.0.0.7		
Test case	Test result	Note
MR-06-3	<i>Observed</i>	The tool identified a wrong data stream due to the NULL 'data offset' in a value (vk) cell. Having the NULL offset is abnormal, but there was no notification from the tool.
MR-06-4	<i>Observed</i>	The tool identified a BINARY value as a string (SZ) due to the edited 'data type' in a value (vk) cell.
MR-07	<i>Observed</i>	The tool didn't identify a big-data stream due to the NULL 'data size' in a value (vk) cell. In addition, the tool failed to report types and data of other valid values in a v1.5 hive file.
MR-08	<i>Observed</i>	The tool processed a key loop without error. That is, it allowed users to access keys within a closed loop until the tool was terminated. In addition, the tool was terminated without any notification when generating a report on hive files having a key loop. Infinite loops are abnormal, but there was no notification from the tool.
MR-09	<i>Observed</i>	The tool did not identify an integer value (VALUE 0x04 (DWORD-LE)) due to the invalid 'data size' in a value (vk) cell. The 'data size' item beyond a cell's boundary is abnormal, but there was no notification from the tool. In addition, the tool failed to report types and data of other valid values.
MR-10	<i>Observed</i>	The tool did not identify a binary value (VALUE 0x03 (BINARY)) due to the invalid 'data size' in a value (vk) cell. The 'data size' item beyond a cell's boundary is abnormal, but there was no notification from the tool. In addition, the tool failed to report types and data of other valid values.
MR-11	<i>Observed</i>	The tool did not identify a string value (VALUE 0x01 (SZ)) due to the invalid 'data size' in a value (vk) cell. The 'data size' item beyond a cell's boundary is abnormal, but there was no notification from the tool. In addition, the tool failed to report types and data of other valid values.
MR-12	<i>Observed</i>	The tool didn't return any results. That is, the tool considers the version indicator in the base block as a key factor for parsing.
MR-13	<i>Observed</i>	The tool returned printable ASCII values due to the edited 'encoding flag' in a key (nk) cell.
MR-14	<i>Observed</i>	The tool returned printable ASCII values due to the edited 'encoding flag' in a value (vk) cell.
MR-15	<i>Observed</i>	The tool tried to decode key/value names and data as UTF-16LE, so names encoded by other encoding standards could not be identified properly.

3.2. Results on Recovering Deleted Registry Objects

FTK 7.0.0.163 and Registry Viewer 2.0.0.7 did not support recovering deleted registry objects. See Table 9 below for more details.

Table 9. Test Results on Recovering Deleted Registry Objects

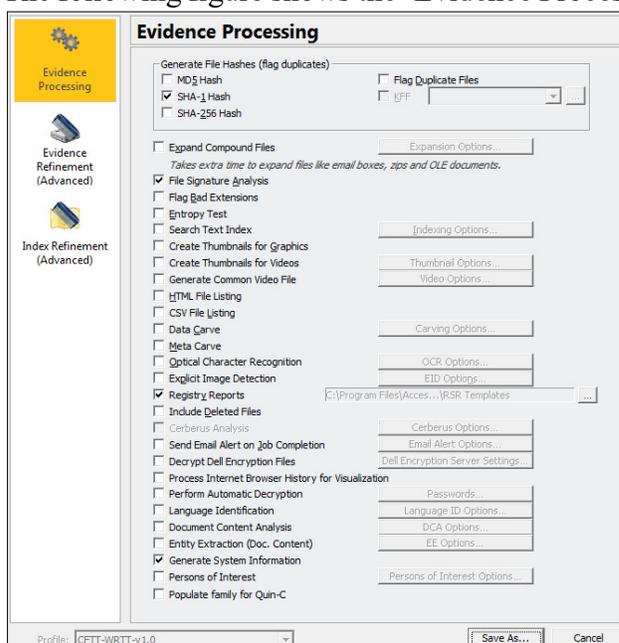
FTK 7.0.0.163 and Registry Viewer 2.0.0.7		
Test case	Test result	Note
NRD-01-1	<i>NA</i>	-
NRD-01-2	<i>NA</i>	-
NRD-02-1	<i>NA</i>	-
NRD-02-2	<i>NA</i>	-
NRD-03-1	<i>NA</i>	-
NRD-03-2	<i>NA</i>	-
NRD-04	<i>NA</i>	-
NRD-05	<i>NA</i>	-
NRD-06	<i>NA</i>	-

3.3. Results on Extracting Windows Forensic Artifacts

The reference registry hive files were processed with FTK 7.0.0.163³ and Registry Viewer 2.0.0.7.

DETAILS:

- This optional feature was not measured strictly by a detailed criterion. In the light of the fact that forensic tools may produce different forms of results although they process identical data to extract Windows registry forensic artifacts, this test was performed through comparative analysis with ground truth data and registry-related knowledge by CFTT staff.
- To perform test on FTK 7.0.0.163, six cases for each Windows system (Vista, 7, 8, 8.1, 10, 10RS1) were created for processing reference hive files. For each case, a disk image (containing hive files of the last VSC) was added as ‘Evidence’ of the tool, and then it was processed along with the following ‘Evidence Processing’ options.
 - ‘Registry Reports’ option
 - This option creates Registry Summary Reports⁴ (RSR).
 - RSR requires that the ‘File Signature Analysis’ option also be enabled.
 - ‘Generate System Information’ option
 - This option extracts well-known system information (including registry artifacts) and populates the ‘System Information Tab’.
 - The system information extracted by the tool includes but not limited to Applications (Prefetch, UserAssist, Installed), Browsers (Downloads, URLs), Network Information (Network Connections, Wireless Profiles), Owner Information, Recent Files, SAM Users and USB Devices.
 - The following figure shows the ‘Evidence Processing’ profile used for this test:



³The ‘Time Zone’ option for each case was configured as ‘Eastern Time with Daylight Saving’.

⁴The Registry Viewer supports Registry Summary Report (RSR) generation as a part of case processing. It should be noted that this test only utilized the default RSR templates provided by the tool.

- In addition, this tool testing also considered built-in data parsers of an independent tool, Registry Viewer 2.0.0.7. The built-in parsers of the tool provide information presented in an easy to read format on well-known data structures such as UserAssist, ShellBag and RecentDocs.
 - Note that the Registry Viewer provides a special feature called ‘Common Areas’ view for quickly accessing forensically interesting registry keys. However, this work didn’t consider the feature because it simply provided shortcuts to pre-defined known registry key paths without any data processing or normalization.

- As a result, the registry-related results processed by both FTK and Registry Viewer were identified through the following three different ways.
 - System Information Tab of FTK
 - The System Information Tab was populated by the ‘Generate System Information’ option.
 - Registry Summary Reports (RSRs)
 - HTML reports were created by the ‘Registry Reports’ option, and they could be accessed at ‘[Case-Directory]\registry_viewer\AutoRSR\’.
 - Built-in Parsers in Registry Viewer
 - The ‘Key Properties’ and ‘Value Properties’ viewers of Registry Viewer provide parsed information for well-known data structures.

- For reference, the below table shows a relationship between the above three different ways and each test case.

Test case	Artifact types in System Information Tab of 'FTK'	Names of RSR reports	Built-in parsers in 'Registry Viewer'
FA-01	◦ SAM Users	◦ SAM – Users ◦ SOFTWARE – Profile List	◦ SAM Users ⁵
FA-02	◦ Applications – Installed	◦ SOFTWARE – Installed Software	-
FA-03	-	-	-
FA-04	-	-	-
FA-05	-	◦ NTUSER – Startup Software by User ◦ SOFTWARE – Startup Software	-
FA-06	-	◦ NTUSER – Vista ComDlg32	-
FA-07	◦ USB Devices	◦ SOFTWARE – Installed Devices	-
FA-08	◦ Networks - Network Connections	-	-
FA-09	◦ Networks – Network Shares	◦ NTUSER – Map Network Drive MRU	-
FA-10	◦ Owner Information	◦ SOFTWARE – OS Version ◦ SOFTWARE – User and OS ◦ SYSTEM – Last Shutdown Time ◦ SYSTEM – Computer Name	◦ OS Install Date
FA-11	◦ Recent Files – NTUser	◦ NTUser – Recent Docs	◦ RecentDocs ⁶
FA-12	-	-	-
FA-13	-	◦ NTUSER – RunMRUs	-
FA-14	-	-	-
FA-15	-	-	-
FA-16	-	-	-
FA-17	◦ Shell Bags	-	◦ BagMRU ⁷
FA-18	-	◦ SYSTEM – Time Zone Settings	-
FA-19	◦ Application – User Assist	-	◦ UserAssist ⁸

⁵ The User entry parser for SAM hive does report the following fields: RID unique identifier, User Name, Description, Logon Count, Last Logon Time, Last Password Change Time, Expiration Time, Invalid Logon Count, Last Failed Login Time, Account Disabled, Password Required, Country Code, Hours Allowed, NT Hash and LM Hash.

⁶ The RecentDocs entry parser for NTUSER hive does report the following fields: Shortcut Target Name (Unicode) and Shortcut Target Name (ASCII).

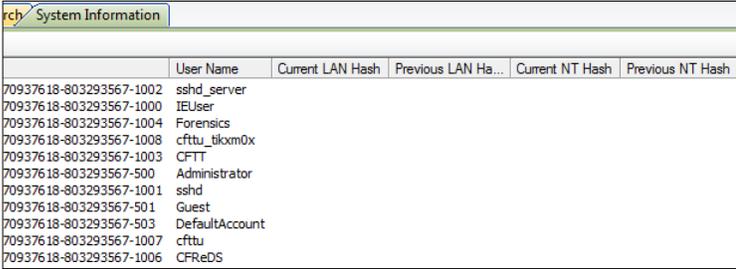
⁷ The BagMRU entry parser for NTUSER and USRCLASS hives does report the following fields: Resource Type, DOS 8.3 Name, Name, Full Path, Create Time, Modify Time and Access Time.

⁸ The UserAssist entry parser for NTUSER hive does report the following fields: Value Name ROT13, Time and Times Executed.

All test cases were successful except for the following. It should be noted that the remainder of this report mentions three different ways for identifying outputs as follows: ‘System Information Tab of FTK’ as *FTK*, ‘Registry Summary Report’ as *RSR* and ‘Built-in Parser in Registry Viewer’ as *RV*. In addition, unless otherwise noted, RSR reports mentioned here are composed of multiple tables containing keys and values without any data normalization.

- Account

- Partial account related data was reported. [Windows: 10RS1]
 - The FTK reported user names only. It did not identify NT Hash values.

Description	Related screenshots
<ul style="list-style-type: none"> • The FTK failed to identify NT Hash values. 	<ul style="list-style-type: none"> • Account list identified by the FTK [Windows 10RS1]:  <ul style="list-style-type: none"> ◦ The tool did not report hash values for each account.

- Note that the FTK reported user names and NT Hash values only. That is, it did not support extracting several well-known account related fields such as ‘Logon Count’, ‘Last Logon Time’, ‘Last Password Change Time’, ‘Invalid Logon Count’ and ‘Last Failed Login Time’. On the other hand, in the RSR and RV, account related data including those well-known fields was reported properly. [Windows: ALL⁹]

- Application

- Partial application related data was reported. [Windows: 10, 10RS1]
 - In the FTK and RSR, 32-bit (x86) application related data in 64-bit Windows systems was not reported. That is, they ignored `Wow6432Node` key on 64-bit Windows systems.
 - All three ways (FTK, RSR and RV) did not support to extract Windows Store app¹⁰ related data.

- Auto Run

- Partial auto run related data was reported. [Windows: 10, 10RS1]
 - In the RSR, 32-bit (x86) application related data in 64-bit Windows systems was not reported. That is, the tool ignored `Wow6432Node` key on 64-bit Windows systems.

⁹ ‘Windows: ALL’ includes Vista, 7, 8, 8.1, 10 and 10RS1. (refer to Section 2.2)

¹⁰ Artifacts on installed app packages can be identified from ‘StateRepository-Machine.srd’ SQLite file (in Windows 10) and ‘PackageRepository.edb’ ESE file (until Windows 8.1), as well as USRCLASS hives as shown in Table4.

- Dialog Usage
 - Dialog usage related data was not reported. [Windows: 10, 10RS1]
 - The RSR did not identify dialog usage history from well-known keys, such as `LastVisitedPidMRU` and `OpenSavePidMRU` as listed in Table 4.

- External Device
 - All USB storage devices were reported as expected, but the following findings need to be taken into consideration. [Windows: ALL]
 - The FTK identified USB storage devices and associated timestamps from `Enum`, and `DeviceClasses` keys of the SYSTEM hive. That is, the tool did not identify those data from other well-known keys, such as `DeviceContainers`, `EMDMgmt` and `Windows Portable Devices` as listed in Table 4.
 - Since Windows 8, it has been known that there exist additional timestamps such as ‘First Attached Time’, ‘Last Attached Time’ and ‘Last Detached Time’. However, all three ways (FTK, RSR and RV) did not support reporting those additional data. [Windows: 8, 8.1, 10, 10RS1]

- Network Drive
 - Network drive related data was not reported. [Windows: 10, 10RS1]
 - The FTK and RSR did not report network drive history.
 - Network drive related data was reported as expected, but the following findings need to be taken into consideration. [Windows: Vista, 7, 8, 8.1]
 - The FTK and RSR identified network drives and associated timestamps from `Map Network Drive MRU` and `MountPoints2` keys of the NTUSER hive. That is, the tool did not identify those data from another well-known key, `Network` as listed in Table 4.

- OS Information
 - OS Information related data was reported as expected, but the following findings need to be taken into consideration. [Windows: ALL]
 - The FTK simply reported raw integer data of the ‘InstallDate’ value, instead of converting it into a human-readable string.
 - Note that the RSR and RV correctly identified ‘InstallDate’ value as a human-readable string.

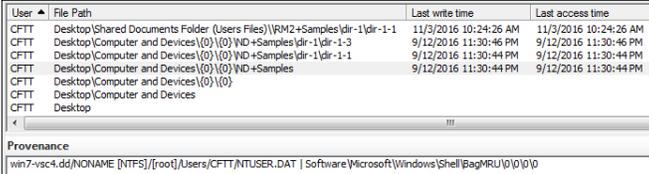
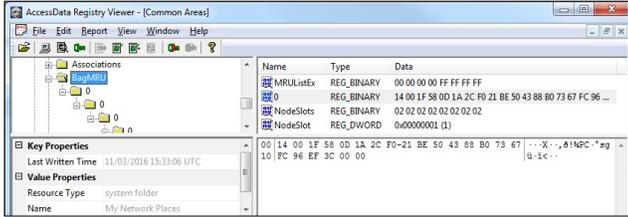
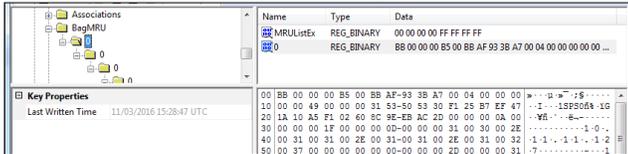
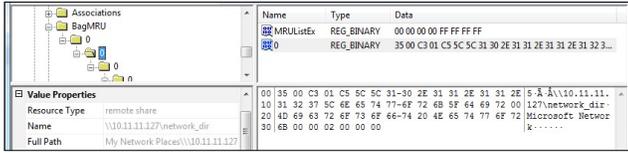
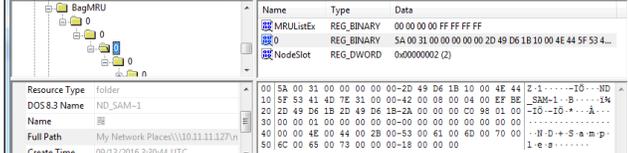
- Recently Opened File and Directory
 - Partial opened file and directory related data was reported. [Windows: Vista, 7, 8, 8.1]
 - The FTK identified recent file history on famous user applications such as Adobe Reader and MS Office (Excel, PowerPoint, Word). That is, the tool did not report those data from another well-known key, `RecentDocs` as listed in Table 4.
 - Opened file and directory related data was not reported. [Windows: 10, 10RS1]
 - The FTK and RSR did not report opened file and directory related data.

- Run Command
 - Run command related data was not reported. [Windows: 10, 10RS1]
 - The RSR failed to generate an RSR report related to run command history.

- ShellBag

- Partial ShellBag related data was reported. [Windows: ALL]

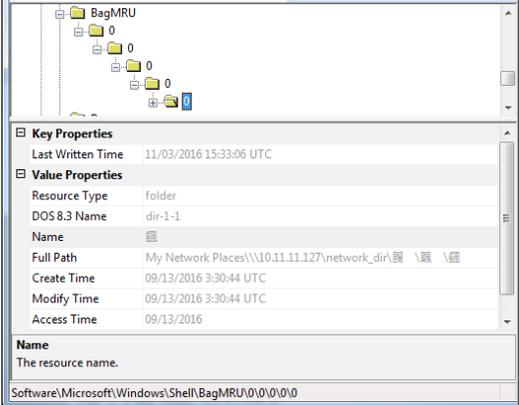
- The FTK reported ShellBag related data in USRCLASS hives, but those data in NTUSER hives were not reported. [Windows: 10, 10RS1]
 - The FTK did not report MRU timestamps¹¹. That is, the tool only reported MAC (Modified/Accessed/Created) timestamps stored within each ShellBag entry.
 - The FTK incorrectly reported directory paths relating to network drives.

Description	Related screenshots
<ul style="list-style-type: none"> Directory paths were reported inaccurately. 	<ul style="list-style-type: none"> Sample ShellBag entries identified by the FTK [Windows 7]:  <ul style="list-style-type: none"> The tool reported wrong paths including specific characters '{0}'. <ul style="list-style-type: none"> The following shows manual verification results: <ul style="list-style-type: none"> Each value of 'BagMRU' key contains a Known Folder ID¹²:  <ul style="list-style-type: none"> → F02C1A0D-BE21-4350-88B0-7367FC96EF3C means 'Network' Each value of sub-keys contains a string name:  <ul style="list-style-type: none"> → '10.11.11.127'  <ul style="list-style-type: none"> → '\\10.11.11.127\network_dir'  <ul style="list-style-type: none"> → 'ND+Samples' As a result, a full path can be constructed as the following: <u>[Network]\10.11.11.127\10.11.11.127\network_dir\ND+Samples\</u>

¹¹ The MRU (Most Recently Used) timestamp can be identified from each key cell. Analyzing MRU timestamps of ShellBag entries is one of important parts to understand users' behaviors relating to handling files and folders.

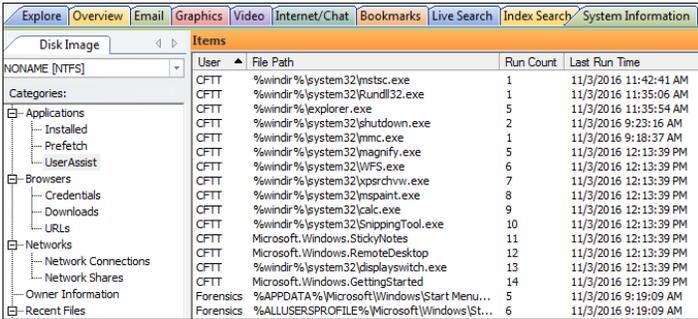
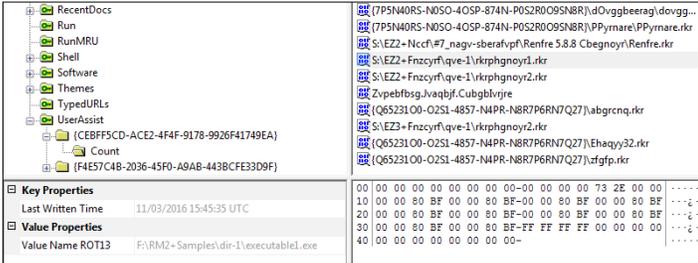
¹² In modern Windows systems, standard folders are referenced by a set of GUID (globally unique identifier) values called Known Folder IDs. (<https://docs.microsoft.com/en-us/windows/desktop/shell/known-folders>)

- The RV reported invalid parsing results.

Description	Related screenshots
<ul style="list-style-type: none"> • Invalid results on 'Name' and 'Full Path' fields were reported. 	<ul style="list-style-type: none"> • Sample ShellBag entries identified by the RV [Windows 7]:  <ul style="list-style-type: none"> ◦ The tool reported invalid characters on 'Name' and 'Full Path' fields.

- UserAssist

- UserAssist related data was not reported. [Windows: 10, 10RS1]
 - The FTK did not report UserAssist entries.
- Partial UserAssist related data was reported. [Windows: 7, 8, 8.1]
 - The FTK only reported values having a 'run count' greater than 1. That is, the tool ignored other UserAssist values, which do not store 'run count' information as shown in the below example.

Description	Related screenshots
<ul style="list-style-type: none"> • The FTK reported UserAssist data partially. 	<ul style="list-style-type: none"> • Sample UserAssist entries identified by the FTK [Windows 7]:  <ul style="list-style-type: none"> ◦ The FTK only identified 15 values from the following path: NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count\ • The following shows manual verification results:  <ul style="list-style-type: none"> ◦ As shown in the above figure, there exist other values that do not have 'run count' information, but the FTK ignored them.

- Note that since Windows 7 most UserAssist entries relating to user applications do not store 'run count' and 'last run date/time' information. Instead, it has been known that there exists additional information such as 'focus count' and 'focus time'. However, both the FTK and RV did not support reporting those additional data.

NOTES:

- The following test cases were not supported by the tool:
 - Application (Windows Store app related data)
 - Application Compatibility (Amcache)
 - Application Compatibility (Shimcache)
 - Network Connection
 - It should be noted that the FTK properly reported network profiles related data, including network name, gateway MAC address, first/last connect time and profile name. However, regarding this test case, the tool did not support extracting network interface card related data, such as assigned IP, Gateway and DNS addresses.
 - Remote Desktop
 - Search
 - Service and Driver
 - Shared Directory

See Table 10 below for more details.

Table 10. Test Results on Extracting Windows Registry Forensic Artifacts
 (The hyphen (-) symbol means that reference hive files do not include any associated entries)

FTK 7.0.0.163 and Registry Viewer 2.0.0.7						
Test cases - registry artifacts	Test result					
	Vista	7	8	8.1	10	10 RS1
FA-01 Account	<i>As Expected</i>	<i>Partial</i>				
FA-02 Application	<i>As Expected</i>	<i>As Expected</i>	<i>As Expected</i>	<i>As Expected</i>	<i>Partial</i>	<i>Partial</i>
FA-03 App Compatibility (Amcache)	-	-	NA	NA	NA	NA
FA-04 App Compatibility (Shimcache)	NA	NA	NA	NA	NA	NA
FA-05 Auto Run	<i>As Expected</i>	<i>As Expected</i>	<i>As Expected</i>	<i>As Expected</i>	<i>Partial</i>	<i>Partial</i>
FA-06 Dialog Usage	-	-	-	-	<i>Not As Expected</i>	<i>Not As Expected</i>
FA-07 External Device	<i>As Expected</i>	<i>As Expected</i>				
FA-08 Network Connection	NA	NA	NA	NA	NA	NA
FA-09 Network Drive	<i>As Expected</i>	<i>As Expected</i>	<i>As Expected</i>	<i>As Expected</i>	<i>Not As Expected</i>	<i>Not As Expected</i>
FA-10 OS Information	<i>As Expected</i>	<i>As Expected</i>				
FA-11 Recently Opened F&D	<i>Partial</i>	<i>Partial</i>	<i>Partial</i>	<i>Partial</i>	<i>Not As Expected</i>	<i>Not As Expected</i>
FA-12 Remote Desktop	NA	NA	NA	NA	NA	NA
FA-13 Run Command	<i>As Expected</i>	<i>As Expected</i>	<i>As Expected</i>	<i>As Expected</i>	<i>Not As Expected</i>	<i>Not As Expected</i>
FA-14 Search	-	NA	NA	-	-	-
FA-15 Service and Driver	NA	NA	NA	NA	NA	NA
FA-16 Shared Directory	NA	NA	NA	NA	NA	NA
FA-17 ShellBag	<i>Partial</i>	<i>Partial</i>	<i>Partial</i>	<i>Partial</i>	<i>Partial</i>	<i>Partial</i>
FA-18 Timezone	<i>As Expected</i>	<i>As Expected</i>				
FA-19 UserAssist	<i>As Expected</i>	<i>Partial</i>	<i>Partial</i>	<i>Partial</i>	<i>Not As Expected</i>	<i>Not As Expected</i>