**YOUR LOGO HERE**

**Digital Forensics and Incident Response**

**Validation Manual**

**DOCUMENT CONTROL #**

**CLASSIFICATION LEVEL HERE**

May be exempt from public release under the Freedom of Information Act (5 U.S.C. 552) exemption number and category: **7, Law Enforcement**

**Department of Name of Agency review required before public release**

Name/Org: Your name/org Date:

Guidance (if applicable):

**Record of Changes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Pages Affected | Description | Author |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**TABLE OF CONTENTS**

1. [Purpose](#purpose)4

2. [Scope](#Scope)4

3. [Roles and Responsibilities](#Roles_and_Responsibilities)4

4. [Requirements](#Requirements)5

[Prefetch Parser version 1.5](#prefetch_parser)6

[FTK Imager version 3.11](#ftk_imager)9

[FTK Imager Lite version 3.11](#ftk_imager_lite)19

[RegRipper version 2.5](#regripper)27

[SDelete version 1.6](#sdelete)30

[WinMD5Free version 1.20](#winmd5free)33

[List of Approved Cyber Forensic Tools](#list_of_approved_tools)39

[Forensic Hardware Validations](#forensic_hardware_beginning)40

[List of Approved Forensic Hardware Devices](#approved_forensic_hardware_list)42

**1. Purpose**

The purpose of the Cyber Incident Response Team (CIRT) Validation Manual is to document the testing and validation of forensic hardware and software tools. Validation is performed by [Agency Name] CIRT personnel anytime new, revised, or reconfigured tools are introduced to the forensic process. Tools are validated in accordance with the [Agency Name] CIRT Quality Manual. This manual also serves as a reference guide for personnel using the tools listed in this document.

**2. Scope**

This document is applicable to all Cyber Security personnel who have CIRT responsibilities including, but not limited to; collection and preservation of digital evidence, incident response, cyber investigations, fraud, waste, and abuse investigations, or the forensic analysis of digital evidence. The individuals most impacted by this technical manual are those assigned to the [Agency Name] Monitor and Control Team.

**3. Roles and Responsibilities**

|  |  |
| --- | --- |
| Key Role / Position Title | Responsibility |
| Cyber Security Personnel assigned to CIRT | * Test and validate tools as necessary to perform forensic analysis of digital evidence. * Provide input into making updates or changes to this manual as needed. * Notify the [Agency Name] Monitor and Control Team when new tools or updates to tools are distributed. |
| CIRT Supervisor | * Maintain document control, including revision tracking. * Ensure the document is reviewed at least annually to account for technological changes. * Incorporate changes requested by CIRT personnel. * Ensure document meets all legal, technical, and administrative requirements. * Approve individual validation papers written by CIRT personnel. |
| Cyber Security Manager | * Provide input as necessary to document. * Review as necessary. |
| Information System Security Manager (ISSM) | * Ensure software used is in compliance with [Agency Name] Commercial off the shelf (COTS) and freeware/shareware policies. * Final approval of software once validated by CIRT. |

**4. Requirements**

As outlined in the Quality Manual, only tools that have been validated shall be utilized by CIRT personnel in the scope of their duties. It is the practice of [Agency Name] to validate all forensic hardware and software used in cyber investigations. Major releases of all software will be validated (e.g., version 1.x to version 2.x), however minor releases and patches will not require a new validation study (e.g., version 1.1 to version 1.5). Forensic hardware write-blocking devices shall be validated annually, or anytime they sustain physical damage.

**5. References**

Refer to the Organizational Procedure (OP) titled “Cyber Forensic Tools” for further direction and information about the validation and testing of software.

|  |
| --- |
| **Forensic Validation – Prefetch Parser v 1.5** |

**1. Test Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number: | 2012 – 001 | Test Title: | Prefetch Parser v. 1.5 | | |
| Test Date: | 08/13/2012 | Person Conducting Test: | | | J. Moulin |
| Test Result: | **PASS** |  | |  | |

**2. Tool Being Tested**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Tool: | Prefetch Parser | | | |
| Manufacturer: | Redwolf Computer Forensics | | | |
| Website: | <http://redwolfcomputerforensics.com/index.php> | | | |
| Version (or Download Date if no Version): | | | 1.5 | |
| Approved by Cyber (Date): | | 11/15/2012 | | |
| IT Tracking Log # (If Freeware / Shareware): | | | | 129791 |

**3. Test Platform**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | [removed] | | OS Name and Version: | | | Windows XP, SP 3 | |
| Hard Drive Capacity: | | | | 1 TB | RAM Capacity: | 3 GB | | | |
| Processor: | Intel Core 2 Quad Q9650 | | | | S/N or Asset Tag: | | [removed] | | |
| Make / Model: | | Dell Optiplex 780 | | |  | | | |  |

**4. Purpose and Scope**

This test procedure will test the functionality of Prefetch Parser and its ability to interpret information contained in the Windows prefetch directory.

**5. Description of Methodology**

Prefetch parser version was placed on the test machine’s local hard drive and executed. The tool was expected to correctly identify the location of prefetch files and provide an accurate report of the contents. Several programs were launched prior to testing the software and the exact date, time, and count of each program launch was recorded to check the accuracy of prefetch parser.

A second test was conducted to check the software’s ability to parse exported prefetch data that is contained within a local directory. This was accomplished by exporting the prefetch file directory from another Windows machine and pointing Prefetch Parser at the exported directory.

**6. Test Notes**

None.

**7. Test procedures**

1. Turn on the test system and boot into Windows with administrative privileges.
2. Place the folder containing Prefetch Parser and its related system files on the hard drive of the test system.
3. Double click the file titled “prefetch\_parser\_gui”.
4. Click “No” if it prompts you to remember the previous case information and file paths.
5. Enter case number.
6. Browse to where the prefetch directory is located that is to be examined.
7. Specify where to place the results of the prefetch file parsing. Remember to use proper naming conventions (e.g., ~Documents and Settings\User\Desktop\12-001\Item1\Prefetch\).
8. Specify the correct version of Windows in the dropdown box.
9. Specify the output type (generally to be HTML).
10. Click “Parse prefetch files”.
11. At the conclusion of the program running, open up the directory that contains your report information.
12. Find the index.html file in this directory and launch it in your browser.
13. Review the information obtained. Remember that times reports are in UTC 24-hour time and may need to be converted to local time by subtracting 7 or 8 hours depending on the time of year.

**8. Expected Results**

1. The software correctly parses Windows prefetch files and outputs the results in a HTML report.
2. The software correctly interprets the number of times a program has been executed.
3. The software correctly interprets the date and time of the last execution of a program.
4. The software correctly interprets the file paths of files needed to run the executable.
5. The software is able to parse Windows prefetch files within any directory it is pointed to.

**9. Test Results**

The actual results were the expected results.

**10. Observations / Concerns**

None.

**11. Limitations**

This software is currently limited only to the Windows operating system.

**12. Recommendations**

This software should be part of the [Agency Name]’s CIRT forensic tool set.

|  |
| --- |
| **Forensic Validation – FTK Imager version 3.11** |

**1. Test Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number: | 2012 – 002 | Test Title: | FTK Imager version 3.11 | | |
| Test Date: | 012/03/2012 | Person Conducting Test: | | | J. Moulin |
| Test Result: | **PASS** |  | |  | |

**2. Tool Being Tested**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Tool: | FTK Imager | | | |
| Manufacturer: | AccessData Corporation | | | |
| Website: | [www.accessdata.com](http://www.accessdata.com) | | | |
| Version (or Download Date if no Version): | | | 3.11 | |
| Approved by Cyber (Date): | | 7/10/2012 | | |
| IT Tracking Log # (If Freeware / Shareware): | | | | N/A - COTS |

**3. Test Platform**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | [removed] | | OS Name and Version: | | | Windows XP, SP 3 | |
| Hard Drive Capacity: | | | | 1 TB | RAM Capacity: | 3 GB | | | |
| Processor: | Intel Core 2 Quad Q9650 | | | | S/N or Asset Tag: | | [removed] | | |
| Make / Model: | | Dell Optiplex 780 | | |  | | | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | N/A | | OS Name and Version: | | | N/A | |
| Hard Drive Capacity: | | | | 1 TB | RAM Capacity: | N/A | | | |
| Processor: | N/A | | | | S/N or Asset Tag: | | N/A | | |
| Make / Model: | | Lexar 2 GB USB thumb drive | | |  | | | |  |

**4. Purpose and Scope**

This test procedure will test the functionality of FTK Imager in the following categories:

1. Forensic imaging of physical devices.
2. Forensic imaging of logical files.
3. Hashing.
4. Mounting forensic images in a forensically sound manner.
5. Obtaining protected files from host computer.
6. Hex value interpretation.
7. Capturing volatile memory (RAM).

**5. Description of Methodology**

FTK Imager was installed on the test machine’s local hard drive and executed. The tool was expected to correctly perform the following functions:

1. Acquire an image of a physical disk without making any changes to the data on the disk.
2. Acquire an image of logical files without making any changes to the files.
3. Creating an accurate hash value of the data being hashed.
4. Mounting a forensic image and allowing the image to be previewed without making any changes to the forensic image.
5. Capturing RAM from a live system and creating a forensic image of the RAM contents.
6. Obtaining protected files (e.g., registry hives) of a running computer system.
7. Correctly interpreting hex values in a file.

**6. Test Notes**

None.

**7. Test Procedure – Acquire an image of a physical disk without making any changes to the data on the disk**

1. Launch FTK Imager on forensic machine.
2. Connect test device with known files on it to the forensic machine.
3. Hash the original device by clicking “File” and then “Add Evidence Item”.
4. Click “Physical Drive”.
5. Click “Next”.
6. Select the physical disk that is the known test media.
7. Right click the physical drive represented in the left hand column titled “Evidence Tree”.
8. Select “Verify Drive/Image…”.
9. Maintain screenshot of this MD5 and SHA1 hash as a baseline for the rest of the test procedure.
10. Remove this as an evidence item after the hash is obtained by clicking “File” and then “Remove all Items” or alternatively click the icon in the toolbar for this command.
11. Click “File” and “Create Disk Image”, or alternatively click the gray disk image icon in the toolbar.
12. Select “Physical Drive”.
13. Select the physical disk to be imaged from the drop down menu.
14. Click “Finish”.
15. In the Create Image dialog box, ensure that the boxes for “verify images after they are created”, “precalculate progress statistics”, and create directory listings of all files in the image after they are created” are checked.
16. Select the type of disk image to be created (e.g., dd, EO1, SMART, etc.).
17. Click “Next”
18. Enter [Agency Name] unique case number (mandatory), evidence number (mandatory), unique description (optional), examiner (mandatory), notes (optional).
19. Click “Next”.
20. Click “Browse” and select the destination for the forensic image. This should be placed on an appropriately prepared piece of media.
21. Provide the image file a name according to appropriate naming conventions.
22. Provide a fragment size (if necessary).
23. Select the compression ratio (if applicable).
24. Click “Finish”
25. The image process should begin and will end with a window showing the pre and post MD5 and SHA1 hash values of the network resource.

**8. Expected Results**

1. The software correctly creates a physical disk image of the known device and contains all known files in addition to unallocated space.
2. The software correctly hashes the physical disk and the MD5 and SHA1 hash of the forensic image matches the original MD5 and SHA1 hash of the test media.

**9. Test Results**

The actual results were the expected results.

**10. Observations / Concerns**

None.

**11. Limitations**

None.

**12. Test Procedure – Acquire a forensic image of logical files without making changes to the files**

1. Place a folder on the forensic computer containing known good files.
2. Create a hash of the entire folder using a known good utility.
3. Launch FTK Imager.
4. Click “File”.
5. Click “Create Disk Image” or alternatively use the toolbar icon.
6. Click “Contents of a Folder”.
7. Click “Next”.
8. If a warning appears about the compatibility of logical image files, click “Yes” to continue.
9. Select the source of the evidence by clicking “Browse” and then navigate to the known good test folder. Click “Ok”.
10. Click “Finish”.
11. In the Create Image dialog box, ensure that the boxes for “verify images after they are created”, “precalculate progress statistics”, and create directory listings of all files in the image after they are created” are checked.
12. Click “Add”.
13. Enter [Agency Name] unique case number (mandatory), evidence number (mandatory), unique description (optional), examiner (mandatory), notes (optional).
14. Click “Next”.
15. Click “Browse” and select the destination for the forensic image. This should be placed on an appropriately prepared piece of media.
16. Provide the image file a name according to appropriate naming conventions.
17. Provide a fragment size.
18. Select the compression ratio.
19. Select whether or not to use AccessData (AD) encryption.
20. Click “Finish”.
21. The image process should begin and will end with a window showing the pre and post MD5 and SHA1 hash values of the network resource.
22. Compare the MD5 and SHA1 hash values of the forensic logical image to that of the original folder.

**13. Expected Results**

1. The software correctly creates a logical disk image of the known folder and contains all known files.
2. The software correctly hashes the logical folder and the MD5 and SHA1 hash of the forensic image matches the original MD5 and SHA1 hash of the test files.

**14. Test Results**

The actual results were the expected results.

**15. Observations / Concerns**

None.

**16. Limitations**

None.

**17. Test Procedure – Test the hashing functionality of FTK Imager**

1. Place a folder on the forensic computer that contains several files. Each file has been hashed previously by a known good utility.
2. Launch FTK Imager.
3. Click “File” and “Add Evidence Item” or alternatively, use the toolbar icon.
4. Select “Contents of a Folder”.
5. Click “Browse” and navigate to the test folder.
6. Click “Finish”.
7. Expand the Evidence Tree to reveal the file path of the test folder.
8. Right click the file path and select “Export Hash List”.
9. Determine where to save the .csv file and provide a name for the document.
10. Click “Save”.
11. Open the .csv file to locate the MD5, SHA1, and filenames (including file path) of each file that was present in test folder.
12. Compare the MD5 and SHA1 hash values in the .csv file with the original hash values obtained by the known good utility.

**18. Expected Results**

1. The software correctly adds the contents of the folder selected.
2. The software correctly creates a .csv file containing the MD5, SHA1 and filename (with original file path) of each file in the test folder.
3. The software correctly hashes the logical folder and the MD5 and SHA1 hash of the forensic image matches the original MD5 and SHA1 hash of the test files.

**19. Test Results**

The actual results were the expected results.

**20. Observations / Concerns**

None.

**21. Limitations**

None.

**22. Test Procedure – Mounting a forensic image and allowing the image to be previewed without making any changes to the forensic image**

1. Connect a physical disk to the forensic computer using write protection.
2. Launch FTK Imager.
3. Add the physical disk to FTK Imager as an evidence item and verify the disk by creating a MD5 and SHA1 hash of the entire physical disk.
4. Create a physical disk image of the disk connected to the forensic computer using FTK Imager. Store the disk image on the forensic computer.
5. At the conclusion of the forensic imaging process (using the steps outlined in this document), compare the MD5 and SHA1 hash values of the forensic image with the original MD5 and SHA1 of the physical disk.
6. Click “File” and then select “Image Mounting”, or alternatively the small gray icon in the toolbar that represents this same command.
7. Click the ellipsis under the “Image File” dialog box and navigate to where the forensic image is currently stored and click “open”.
8. Select whether to mount this as a logical or physical drive.
9. Assign the drive letter.
10. Select File System / Read Only as the mount method.
11. Click “Mount”.
12. Go to “My Computer” and ensure the drive mounted appropriately (note, if the image had multiple partitions, you should see each partition mounted with its own drive letter).
13. Try to add files into the mounted volume, attempt to change files, and do other actions in an attempt to alter files.
14. Switch back to FTK Imager and click on the “Mapped Images” area. Click on the Image file that is currently mounted and then click “Unmount”.
15. In FTK Imager click “Add Evidence Item”.
16. Click “Image File”.
17. Click “Next”.
18. Browse to the image file of the known disk used in this test procedure and click “Open”.
19. Click “Finish”.
20. Right click the image file shown on the left side under “Evidence Tree”.
21. Select “Verify Drive/Image”.
22. At the completion of the hashing of the image file, compare the MD5 and SHA1 hash values of this verification to the original hash and the hash of the image prior to mounting it.

**23. Expected Results**

1. The software correctly mounts an image file as a logical volume on the local machine.
2. The software allows the user to preview the logical files using forensic software tools or built-in tools such as Windows Explorer.
3. The software blocks any write attempts to the mounted image file.
4. The MD5 and SHA1 hash values of the forensic image file does not change after being mounted and deliberate attempts made to change data on the volume.

**24. Test Results**

The actual results were the expected results.

**25. Observations / Concerns**

None.

**26. Limitations**

None.

**27. Test Procedure – Obtaining protected files from the host computer**

1. On a live Windows computer system install FTK Imager (Note, this can be done using FTK Imager Lite from USB device, however that is detailed in the validation for FTK Imager Lite).
2. Click “File” and then “Obtain Protected Files”, or alternatively click the icon in the toolbar for this function.
3. Browse for the destination of the files (generally a removable storage device).
4. Select the appropriate options (either minimum files for login password recovery, or recover all registry files). In this validation the selection of “recover all registry files was chosen”.
5. Click “OK”.
6. FTK Imager will then export all the registry hives from the live system to the target location selected by the examiner.
7. Open the target location and ensure the files exist.
8. Open the registry hives with a known good utility (such as regedit) to confirm they are readable.
9. Close FTK Imager.

**28. Expected Results**

1. The software will correctly identify all registry hives and export them to the target media.
2. The registry hives will be readable.

**29. Test Results**

The actual results were the expected results.

**30. Observations / Concerns**

None.

**31. Limitations**

None.

**32. Test Procedure – Hex value interpretation**

1. Launch FTK Imager on the forensic machine.
2. Place a file(s) in a folder on the forensic machine that contains information in hex which can be interpreted by FTK Imager.
3. In this scenario, multiple prefetch files were placed in a folder labeled “prefetch” on the desktop of the forensic computer.
4. Add the folder “prefetch” as an evidence item in FTK Imager (already discussed in this document).
5. Select one prefetch file and ensure the view window is showing the hex view.
6. Ensure the bottom left window (properties window) is selected for Hex Value Interpreter) option is available at the bottom of the window. Also select Big Endian or Little Endian (in this case it is Little Endian).
7. For this scenario, several prefetch files were looked at in hex view, specifically at decimal offset 144. There is a DWORD value of 4 bytes at this location which represents the number of times an application has been run.
8. By using another validated tool, Prefetch Parser, the results were confirmed that the hex value interpreter was working correctly for DWORD values.
9. At decimal offset 120 is a 64-bit value that is FILETIME, showing the last time an application was run in UTC. This was confirmed as accurate with both prefetch parser and independent recollection of the last time the application was run.

**33. Expected Results**

1. The software will correctly view data in hex.
2. The software’s hex value interpreter will correctly interpret different hex values.

**34. Test Results**

The actual results were the expected results.

**35. Observations / Concerns**

None.

**36. Limitations**

None.

**37. Test Procedure – Capturing volatile memory (RAM)**

1. Install FTK Imager on suspect machine (Note, this function is available on FTK Imager Lite and would not require an installation, however that is covered under a different validation).
2. Launch FTK Imager.
3. Click “File” and then “Capture Memory”, or alternatively click on the toolbar icon.
4. Click “Browse” and select the target location where the RAM dump will be saved to.
5. Enter a filename for the RAM dump.
6. Click “Capture Memory”.
7. At the completion of the memory capture, check the target disk to ensure there is a memory dump file there.

**38. Expected Results**

1. The software will obtain the memory of the computer.

**39. Test Results**

The actual results were the expected results.

**40. Observations / Concerns**

None.

**41. Limitations**

This functionality is only for a computer using the Windows operating system.

**42. Recommendations**

This software should be part of the [Agency Name]’s CIRT forensic tool set.

|  |
| --- |
| **Forensic Validation – FTK Imager Lite version 3.11** |

**1. Test Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number: | 2012 – 003 | Test Title: | FTK Imager Lite version 3.11 | | |
| Test Date: | 12/03/2012 | Person Conducting Test: | | | J. Moulin |
| Test Result: | **PASS** |  | |  | |

**2. Tool Being Tested**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Tool: | FTK Imager Lite | | | |
| Manufacturer: | AccessData Corporation | | | |
| Website: | [www.accessdata.com](http://www.accessdata.com) | | | |
| Version (or Download Date if no Version): | | | 3.11 | |
| Approved by Cyber (Date): | | 7/10/2012 | | |
| IT Tracking Log # (If Freeware / Shareware): | | | | Freeware - |

**3. Test Platform**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | [removed] | | OS Name and Version: | | | Windows XP, SP 3 | |
| Hard Drive Capacity: | | | | 1 TB | RAM Capacity: | 3 GB | | | |
| Processor: | Intel Core 2 Quad Q9650 | | | | S/N or Asset Tag: | | [removed] | | |
| Make / Model: | | Dell Optiplex 780 | | |  | | | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | N/A | | OS Name and Version: | | | N/A | |
| Hard Drive Capacity: | | | | 1 TB | RAM Capacity: | N/A | | | |
| Processor: | N/A | | | | S/N or Asset Tag: | | N/A | | |
| Make / Model: | | Lexar 2 GB USB thumb drive | | |  | | | |  |

**4. Purpose and Scope**

This test procedure will test the functionality of FTK Imager Lite in the following categories:

1. Forensic imaging of physical devices.
2. Forensic imaging of logical files.
3. Hashing.
4. Obtaining protected files from host computer.
5. Hex value interpretation.
6. Capturing volatile memory (RAM).

**5. Description of Methodology**

FTK Imager Lite was placed onto a USB thumb drive. The tool was expected to correctly perform the following functions:

1. Acquire an image of a physical disk without making any changes to the data on the disk.
2. Acquire an image of logical files without making any changes to the files.
3. Creating an accurate hash value of the data being hashed.
4. Mounting a forensic image and allowing the image to be previewed without making any changes to the forensic image.
5. Capturing RAM from a live system and creating a forensic image of the RAM contents.
6. Obtaining protected files (e.g., registry hives) of a running computer system.
7. Correctly interpreting hex values in a file.

**6. Test Notes**

FTK Imager Lite does not require any files to be installed on the host computer. Several files are required to run off of the USB device in order to launch FTK Imager Lite from the USB device.

**7. Test Procedure – Acquire an image of a physical disk without making any changes to the data on the disk**

1. Launch FTK Imager Lite on host machine.
2. Connect test device with known files on it to the forensic machine.
3. Hash the original device by clicking “File” and then “Add Evidence Item”.
4. Click “Physical Drive”.
5. Click “Next”.
6. Select the physical disk that is the known test media.
7. Right click the physical drive represented in the left hand column titled “Evidence Tree”.
8. Select “Verify Drive/Image…”.
9. Maintain screenshot of this MD5 and SHA1 hash as a baseline for the rest of the test procedure.
10. Remove this as an evidence item after the hash is obtained by clicking “File” and then “Remove all Items” or alternatively click the icon in the toolbar for this command.
11. Click “File” and “Create Disk Image”, or alternatively click the gray disk image icon in the toolbar.
12. Select “Physical Drive”.
13. Select the physical disk to be imaged from the drop down menu.
14. Click “Finish”.
15. In the Create Image dialog box, ensure that the boxes for “verify images after they are created”, “precalculate progress statistics”, and create directory listings of all files in the image after they are created” are checked.
16. Select the type of disk image to be created (e.g., dd, EO1, SMART, etc.).
17. Click “Next”
18. Enter [Agency Name] unique case number (mandatory), evidence number (mandatory), unique description (optional), examiner (mandatory), notes (optional).
19. Click “Next”.
20. Click “Browse” and select the destination for the forensic image. This should be placed on an appropriately prepared piece of media.
21. Provide the image file a name according to appropriate naming conventions.
22. Provide a fragment size (if necessary).
23. Select the compression ratio (if applicable).
24. Click “Finish”
25. The image process should begin and will end with a window showing the pre and post MD5 and SHA1 hash values of the network resource.

**8. Expected Results**

1. The software correctly creates a physical disk image of the known device and contains all known files in addition to unallocated space.
2. The software correctly hashes the physical disk and the MD5 and SHA1 hash of the forensic image matches the original MD5 and SHA1 hash of the test media.

**9. Test Results**

The actual results were the expected results.

**10. Observations / Concerns**

None.

**11. Limitations**

None.

**12. Test Procedure – Acquire a forensic image of logical files without making changes to the files**

1. Place a folder on the forensic computer containing known good files.
2. Create a hash of the entire folder using a known good utility.
3. Launch FTK Imager Lite.
4. Click “File”.
5. Click “Create Disk Image” or alternatively use the toolbar icon.
6. Click “Contents of a Folder”.
7. Click “Next”.
8. If a warning appears about the compatibility of logical image files, click “Yes” to continue.
9. Select the source of the evidence by clicking “Browse” and then navigate to the known good test folder. Click “Ok”.
10. Click “Finish”.
11. In the Create Image dialog box, ensure that the boxes for “verify images after they are created”, “precalculate progress statistics”, and create directory listings of all files in the image after they are created” are checked.
12. Click “Add”.
13. Enter [Agency Name] unique case number (mandatory), evidence number (mandatory), unique description (optional), examiner (mandatory), notes (optional).
14. Click “Next”.
15. Click “Browse” and select the destination for the forensic image. This should be placed on an appropriately prepared piece of media.
16. Provide the image file a name according to appropriate naming conventions.
17. Provide a fragment size.
18. Select the compression ratio.
19. Select whether or not to use AccessData (AD) encryption.
20. Click “Finish”.
21. The image process should begin and will end with a window showing the pre and post MD5 and SHA1 hash values of the network resource.
22. Compare the MD5 and SHA1 hash values of the forensic logical image to that of the original folder.

**13. Expected Results**

1. The software correctly creates a logical disk image of the known folder and contains all known files.
2. The software correctly hashes the logical folder and the MD5 and SHA1 hash of the forensic image matches the original MD5 and SHA1 hash of the test files.

**14. Test Results**

The actual results were the expected results.

**15. Observations / Concerns**

None.

**16. Limitations**

None.

**17. Test Procedure – Test the hashing functionality of FTK Imager Lite**

1. Place a folder on the forensic computer that contains several files. Each file has been hashed previously by a known good utility.
2. Launch FTK Imager Lite.
3. Click “File” and “Add Evidence Item” or alternatively, use the toolbar icon.
4. Select “Contents of a Folder”.
5. Click “Browse” and navigate to the test folder.
6. Click “Finish”.
7. Expand the Evidence Tree to reveal the file path of the test folder.
8. Right click the file path and select “Export Hash List”.
9. Determine where to save the .csv file and provide a name for the document.
10. Click “Save”.
11. Open the .csv file to locate the MD5, SHA1, and filenames (including file path) of each file that was present in test folder.
12. Compare the MD5 and SHA1 hash values in the .csv file with the original hash values obtained by the known good utility.

**18. Expected Results**

1. The software correctly adds the contents of the folder selected.
2. The software correctly creates a .csv file containing the MD5, SHA1 and filename (with original file path) of each file in the test folder.
3. The software correctly hashes the logical folder and the MD5 and SHA1 hash of the forensic image matches the original MD5 and SHA1 hash of the test files.

**19. Test Results**

The actual results were the expected results.

**20. Observations / Concerns**

None.

**21. Limitations**

None.

**22. Test Procedure – Obtaining protected files from the host computer**

1. On a live Windows computer system launch FTK Imager Lite from a USB device.
2. Click “File” and then “Obtain Protected Files”, or alternatively click the icon in the toolbar for this function.
3. Browse for the destination of the files (generally a removable storage device).
4. Select the appropriate options (either minimum files for login password recovery, or recover all registry files). In this validation the selection of “recover all registry files was chosen”.
5. Click “OK”.
6. FTK Imager Lite will then export all the registry hives from the live system to the target location selected by the examiner.
7. Open the target location and ensure the files exist.
8. Open the registry hives with a known good utility (such as regedit) to confirm they are readable.
9. Close FTK Imager Lite.

**23. Expected Results**

1. The software will correctly identify all registry hives and export them to the target media.
2. The registry hives will be readable.

**24. Test Results**

The actual results were the expected results.

**25. Observations / Concerns**

None.

**26. Limitations**

None.

**27. Test Procedure – Hex value interpretation**

1. Launch FTK Imager Lite on the forensic machine.
2. Place a file(s) in a folder on the forensic machine that contains information in hex which can be interpreted by FTK Imager Lite.
3. In this scenario, multiple prefetch files were placed in a folder labeled “prefetch” on the desktop of the forensic computer.
4. Add the folder “prefetch” as an evidence item in FTK Imager Lite (already discussed in this document).
5. Select one prefetch file and ensure the view window is showing the hex view.
6. Ensure the bottom left window (properties window) is selected for Hex Value Interpreter) option is available at the bottom of the window. Also select Big Endian or Little Endian (in this case it is Little Endian).
7. For this scenario, several prefetch files were looked at in hex view, specifically at decimal offset 144. There is a DWORD value of 4 bytes at this location which represents the number of times an application has been run.
8. By using another validated tool, Prefetch Parser, the results were confirmed that the hex value interpreter was working correctly for DWORD values.
9. At decimal offset 120 is a 64-bit value that is FILETIME, showing the last time an application was run in UTC. This was confirmed as accurate with both prefetch parser and independent recollection of the last time the application was run.

**28. Expected Results**

1. The software will correctly view data in hex.
2. The software’s hex value interpreter will correctly interpret different hex values.

**29. Test Results**

The actual results were the expected results.

**30. Observations / Concerns**

None.

**31. Limitations**

None.

**32. Test Procedure – Capturing volatile memory (RAM)**

1. Run FTK Imager Lite from a USB drive connected to the host computer.
2. Click “File” and then “Capture Memory”, or alternatively click on the toolbar icon.
3. Click “Browse” and select the target location where the RAM dump will be saved to.
4. Enter a filename for the RAM dump.
5. Click “Capture Memory”.
6. At the completion of the memory capture, check the target disk to ensure there is a memory dump file there.

**33. Expected Results**

1. The software will obtain the memory of the computer.

**34. Test Results**

The actual results were the expected results.

**35. Observations / Concerns**

None.

**36. Limitations**

This functionality is only for a computer using the Windows operating system.

**37. Recommendations**

This software should be part of the [Agency Name]’s CIRT forensic tool set.

|  |
| --- |
| **Forensic Validation – RegRipper v. 2.5** |

**1. Test Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number: | 2012 – 004 | Test Title: | RegRipper | | |
| Test Date: | 08/21/2012 | Person Conducting Test: | | | J. Moulin |
| Test Result: | **PASS** |  | |  | |

**2. Tool Being Tested**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Tool: | RegRipper | | | |
| Manufacturer: | Harlan Carvey | | | |
| Website: | <http://regripper.wordpress.com/> | | | |
| Version (or Download Date if no Version): | | | 2.5 | |
| Approved by Cyber (Date): | | 09/05/2012 | | |
| IT Tracking Log # (If Freeware / Shareware): | | | | Freeware – Tracking # 129787 |

**3. Test Platform**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | [removed] | | OS Name and Version: | | | Windows XP, SP 3 | |
| Hard Drive Capacity: | | | | 1 TB | RAM Capacity: | 3 GB | | | |
| Processor: | Intel Core 2 Quad Q9650 | | | | S/N or Asset Tag: | | [removed] | | |
| Make / Model: | | Dell Optiplex 780 | | |  | | | |  |

**4. Purpose and Scope**

This test procedure will test the functionality of RegRipper in the following categories:

1. Ability to properly parse selected registry hives.
2. Ability to report registry information using plugins.
3. Correctly interpret registry findings.

**5. Description of Methodology**

RegRipper does not have an installation file. The files required to run RegRipper can be ran from the local machine, or via a USB device. The tool was expected to correctly perform the following functions:

1. Ability to properly parse selected registry hives.
2. Ability to report registry information using plugins.
3. Correctly interpret registry findings.
4. Correctly produce a report with registry artifacts documented correctly.

**6. Test Notes**

RegRipper is not installed on a system and may be ran on a forensic computer against a mounted image, from a thumb drive, or by exporting registry hives out of the host computer and running RegRipper against the exported files.

**7. Test Procedure – Setup of RegRipper**

1. Place required files on the forensic computer. For RegRipper, place the two application files, .dll, and Perl files. For optional plugins to work, create a subdirectory within the directory containing the application and Perl files named “plugins”.
2. Ensure plugins are functioning by double clicking the “rr” application file and dropping down the “plugin file” menu. If setup properly, there should be several options available.

**8. Test Procedure – Ability of RegRipper to properly identify and parse registry hives, apply proper plugins, correctly interpret findings, accurately report findings.**

1. Open RegRipper by launching the “rr” application file.
2. Select the registry hive to analyze. In this test, the local machine registry files were examined.
3. Select the “SAM” registry hive. In this case it is located in C:\Windows\System32\config\
4. Select the report file. In this case, a new folder was created on the desktop titled “Registry Reports” and the file name of “SAM Report” was created.
5. Select the “sam-all” plugin.
6. Click “Rip It”.
7. Navigate to the “SAM Report.txt” file that was created.
8. Review report and log file that was created, it should complete with 0 errors.
9. Compare information from report with known good information by running regedit or another known good utility.
10. Launch the “rr” application file.
11. Select the “System” registry hive.
12. Select the report file and save it as SYSTEM Report.txt.
13. Select the “system-all” plugin.
14. Click “Rip It”.
15. Navigate to the “SYSTEM Report.txt” report file and review the contents.
16. Compare report with known good information.
17. Launch the “rr” application file.
18. Select the “Software” registry hive.
19. Select the report file and save it as SOFTWARE Report.txt.
20. Select the “software-all” plugin.
21. Click “Rip It”.
22. Navigate to the “SOFTWARE Report.txt” report file and review the contents.
23. Compare report with known good information.
24. Launch the “rr” application file.
25. Select the “NTUSER.dat” registry hive for a user with known activity.
26. Select the report file and save it as NTUSER Report.txt.
27. Select the “ntuser-all” plugin.
28. Click “Rip It”.
29. Navigate to the “NTUSER Report.txt” report file and review the contents.
30. Compare report with known good information.

**9. Test Results**

The actual results were the expected results.

**10. Observations / Concerns**

None.

**11. Limitations**

None.

**12. Recommendations**

This software should be part of the [Agency Name]’s CIRT forensic tool set.

|  |
| --- |
| **Forensic Validation – SDelete v 1.6** |

**1. Test Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number: | 2012 – 005 | Test Title: | SDelete | | |
| Test Date: | 08/21/2012 | Person Conducting Test: | | | J. Moulin |
| Test Result: | **PASS** |  | |  | |

**2. Tool Being Tested**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Tool: | SDelete | | | |
| Manufacturer: | Microsoft | | | |
| Website: | <http://technet.microsoft.com/en-us/sysinternals/bb897443.aspx> | | | |
| Version (or Download Date if no Version): | | | 1.6 | |
| Approved by Cyber (Date): | | Approved vendor | | |
| IT Tracking Log # (If Freeware / Shareware): | | | | Freeware |

**3. Test Platform**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | [removed] | | OS Name and Version: | | | Windows XP, SP 3 | |
| Hard Drive Capacity: | | | | 1 TB | RAM Capacity: | 3 GB | | | |
| Processor: | Intel Core 2 Quad Q9650 | | | | S/N or Asset Tag: | | [REMOVED] | | |
| Make / Model: | | Dell Optiplex 780 | | |  | | | |  |

**4. Purpose and Scope**

This test procedure will test the functionality of SDelete in the following categories:

1. Ability to securely delete files.

**5. Description of Methodology**

SDelete was installed on the test platform. Files were saved on a removable USB thumb drive and then deleted with SDelete. The USB thumb drive was then opened in a hex editor to confirm the files were overwritten securely. The tool was expected to correctly perform the following functions:

1. Securely delete files by overwriting the sector(s) it occupied.

**6. Test Notes**

SDelete is part of an entire toolset created by Microsoft, commonly referred to as Windows Sysinternals. SDelete could be used by installing it on a host machine to securely delete files on that machine (such as in a spillage event), or it can be installed on a forensic machine that has access to a device or share location and then used to delete those files.

**7. Test Procedure – Secure Overwrite of Specific File**

1. Place the SDelete.exe executable file on the forensic machine.
2. Click “Start” and “Run” and type “cmd” in the run line.
3. Navigate to the file path of the executable (e.g., C:\Documents and Settings\%user%\Desktop\SDelete.exe and press enter, or simply drag the SDelete icon into the command window and press enter. Ensure SDelete runs appropriately.
4. There are multiple options in this command line interface, including wiping free space, overwriting specific files and directories, etc.
5. Place a text file on a removable USB thumb drive.
6. Open the physical disk (USB thumb drive) in FTK Imager or other hex editor and locate the starting sector for the text file and view the information in hex view.
7. Launch SDelete again. Type the command to overwrite the new text file with 1 pass. In this case the command was C:\Documents and Settings\%user%\Desktop\SDelete.exe –p 1 E:\Test.txt. SDelete should advise 1 file was located and it was deleted.
8. Reopen the physical USB thumb drive with FTK Imager or other hex editor.
9. Go back to the original starting sector of the text document.
10. The original filename should be replaced with new characters, as SDelete overwrites a filename 26 times by default.
11. The data in the sector that once had readable text should now be unreadable random characters.

**8. Expected Results**

1. The original filename is overwritten by new characters.
2. The data area on the disk is overwritten by random characters.

**9. Test Results**

The actual results were the expected results.

**10. Observations / Concerns**

None.

**11. Limitations**

None.

**12. Test Procedure – Secure overwriting Entire Free Space on Disk**

1. Salt a drive with known data and review the data in a hex viewer.
2. Delete all data and reformat thumb drive.
3. Open thumb drive in hex viewer and confirm that old deleted data is still visible in the sectors.
4. Determine the drive letter of the disk to be wiped.
5. Launch SDelete with the syntax of: [filepath]sdelete.exe –p 1 –z E:\. The –z switch actually replaces all of the data in free space with zeros, alternatively the –c switch could be used to clean the free space with random characters. In this test the –z was selected as it provides immediate confirmation in a hex viewer that the sectors were overwritten.
6. At the completion of the overwriting, reopen the physical disk in the hex viewer.
7. Confirm that the old data that was still remaining in sectors on the disk have been overwritten with zeros.

**13. Test Results**

The actual results were the expected results.

**14. Observations / Concerns**

None.

**15. Limitations**

It should be noted that this software will not forensically wipe an entire drive. The disk being wiped must be formatted and recognized by the Windows operating system in order for this tool to work. This tool is a quick and simple solution to rapidly and securely overwrite logical files or free space on a drive.

**16. Recommendations**

This software should be part of the [Agency Name]’s CIRT forensic tool set.

|  |
| --- |
| **Forensic Validation – WinMD5Free** |

**1. Test Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number: | 2012 – 006 | Test Title: | WindMD5Free | | |
| Test Date: | 08/22/2012 | Person Conducting Test: | | | J. Moulin |
| Test Result: | **PASS** |  | |  | |

**2. Tool Being Tested**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Tool: | WinMD5Free | | | |
| Manufacturer: | WinMD5.com | | | |
| Website: | <http://www.winmd5.com> | | | |
| Version (or Download Date if no Version): | | | 1.20 | |
| Approved by Cyber (Date): | | 11/19/2012 | | |
| IT Tracking Log # (If Freeware / Shareware): | | | | Freeware |

**3. Test Platform**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | [REMOVED] | | OS Name and Version: | | | Windows XP, SP 3 | |
| Hard Drive Capacity: | | | | 1 TB | RAM Capacity: | 3 GB | | | |
| Processor: | Intel Core 2 Quad Q9650 | | | | S/N or Asset Tag: | | [REMOVED] | | |
| Make / Model: | | Dell Optiplex 780 | | |  | | | |  |

**4. Purpose and Scope**

This test procedure will test the functionality of WinMD5Free in the following categories:

1. Ability to correctly hash files using the MD5 hashing algorithm.

**5. Description of Methodology**

WinMD5Free does not require installation and is a standalone executable file. A file was created and hashed using a known good hashing utility (in this case FTK Imager) and was dropped into WinMD5Free. The hash value from WinMD5Free were compared to the hash value from FTK Imager. The tool was expected to correctly perform the following functions:

1. Create an accurate MD5 hash of a file.

**6. Test Notes**

None.

**7. Test Procedure – Creating a MD5 hash value of a file**

1. Create a new text document on the desktop of the test platform.
2. Place the new text document inside of a new folder.
3. Launch FTK Imager.
4. Add “contents of a folder” as an evidence item.
5. Expand the new folder and click on the new text document created.
6. Right click the text document and then select “export file hash list”.
7. Export the hash list to the desktop of the computer in a .csv file.
8. Open the .csv file and expand the cell containing the MD5 hash.
9. Launch WinMD5Free.
10. Drag and drop the new text document into WinMD5Free.
11. Review the MD5 hash value in the box titled “Current file MD5 checksum value”.
12. Compare the MD5 hash from WinMD5Free with that of FTK Imager.

**8. Expected Results**

1. The hash value created by WindMD5Free is correct and matches the hash value created by a known good utility.

**9. Test Results**

The actual results were the expected results.

**10. Observations / Concerns**

None.

**11. Limitations**

This program is only for the Windows operating system and only creates MD5 hash values of single files.

**12. Recommendations**

This software should be part of the [Agency Name]’s CIRT forensic tool set.

|  |
| --- |
| **Forensic Validation – CelleBrite UFED Touch** |

**1. Test Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number: | 2012 – 026 | Test Title: | CelleBrite UFED Touch | | |
| Test Date: | 12/27/2012 | Person Conducting Test: | | | J. Moulin |
| Test Result: | **PASS** |  | |  | |

**2. Tool Being Tested**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Tool: | CelleBrite UFED Touch | | | |
| Manufacturer: | CelleBrite | | | |
| Website: | <http://www.cellebrite.com> | | | |
| Version (or Download Date if no Version): | | | 1.8.1, image | |
| Approved by Cyber (Date): | | COTS | | |
| IT Tracking Log # (If Freeware / Shareware): | | | |  |

**3. Test Platform**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | CelleBrite UFED Touch | | OS Name and Version: | | | Windows Mobile | |
| Hard Drive Capacity: | | | | N/A | RAM Capacity: | N/A | | | |
| Processor: | N/A | | | | S/N or Asset Tag: | | [removed] | | |
| Make / Model: | | UFED Touch | | |  | | | |  |

**4. Purpose and Scope**

This test procedure will test the functionality of the CelleBrite UFED Touch:

1. Obtain the file system from a mobile device.
2. Obtain the phonebook from a mobile device.
3. Obtain the contents of a memory card from a mobile device.
4. Obtain SMS (text messages) from a mobile device.
5. Obtain the call history from a mobile device.
6. Obtain pictures from a mobile device.
7. Obtain videos from a mobile device.
8. Obtain Email from a mobile device.
9. Obtain contents from a Global System for Mobile Communication (GSM) Subscriber Identity Module (SIM) card.

**5. Description of Methodology**

Salt a mobile device with known data and then connect it to the CelleBrite UFED Touch. Confirm the results of the CelleBrite device with the known data on the device.

1. Extract SIM data.
2. Extract phone data – logical.
3. Extract phone data – physical.
4. Review results report.
5. Compare results report with known information on salted device.

**6. Test Notes**

Each make and model of phone is different and will produce different results. The UFED Touch will not work through a Virtual USB bus. It must be connected to a physical PC or a Mac that is fully bootcamped into Windows.

**7. Test Procedures – Extracting SIM card data**

1. Turn on the CelleBrite Device.
2. Remove SIM from mobile device and insert it into the front slot of the CelleBrite.
3. Connect an export USB drive to the ***target*** side of the CelleBrite, or connect the CelleBrite to a forensic workstation via a USB cable.
4. Using the touch stylus, click “Start” and then click “UFED Touch.”
5. On the “Select Extraction Type” window, click “SIM Data Extraction.”
6. Select whether the SIM is from an Iden phone or a regular SIM.
7. Select the extraction location as either a removable drive or a PC.
8. Select what the CelleBrite should extract. In this scenario Phonebook, SMS, and Call Logs were selected. Make the selections by touching the categories which should show a check mark if they are selected.
9. Click “Next.”
10. Ensure the SIM card is inserted into the CelleBrite correctly. Press “continue.”
11. If prompted, select which partition to read (USIM or SIM).
12. Once the extraction has completed successfully, remove the SIM and the USB export drive. Click OK to return to the main menu.

**8. Test Procedures – Physical Extraction**

1. Turn on the CelleBrite Device.
2. Using the touch stylus, click “Start” and then click “UFED Touch.”
3. Select “Physical Extraction” from the menu.
4. Select the device make. In this test it was a GSM Blackberry.
5. Select the model of the device. In this test it was a 9810 Tourch.
6. Select the extraction location. In this test “PC” was selected.
7. Select the type of data to extract (e.g call logs, SMS, pictures, etc.).
8. Connect USB cable #441 from the target side of the CelleBrite to the forensic computer. Note – the forensic computer must have UFED Physical Analyzer installed and the dongle must be inserted into the computer.
9. Connect the mobile device to the CelleBrite unit using the cable indicated on the screen. In this case it was cable T-100. Follow all of the instructions on the screen.
10. Open UFED Physical Analyzer on the forensic computer.
11. Click the “read” icon in the Physical Analyzer software when prompted by the UFED Touch.
12. Once the extraction has completed, review the data within each separate container (Images, Device Info., Text, etc.) in Physical Analyzer.
13. Compare the data in Physical Analyzer with the known salted data on the mobile device.
14. Disconnect the mobile device from the UFED and disconnected the UFED from the forensic computer.

**9. Test Procedures – Logical Extraction**

1. Turn on the CelleBrite Device.
2. Using the touch stylus, click “Start” and then click “UFED Touch.”
3. Select “Logical Extraction” from the menu.
4. Select the device make. In this test it was a Samsung.
5. Select the model of the device. In this test it was a SCH-1500.
6. Select the extraction location. In this test “Removable Device” was selected.
7. Select the type of data to extract (e.g call logs, SMS, pictures, etc.).
8. Launch CelleBrite Logical Analyzer software on the forensic workstation.
9. Follow the instructions on the screen of the UFED.
10. When prompted connect the mobile device to the UFED.
11. If any additional prompts show up to make changes to the mobile device (e.g., allowing untrusted applications), follow those instructions.
12. Wait for the extraction to complete.
13. Remove the USB thumb drive and review the data for accuracy.

**10. Expected Results**

1. The CelleBrite unit would properly extract information from a SIM card, from a Blackberry during a physical analysis and from a Samsung CDMA phone during a logical analysis.

**11. Test Results**

The actual results were validated without errors.

**12. Observations / Concerns**

None.

**13. Limitations**

Some makes and models of phones may not be supported or have limited support.

|  |
| --- |
| **List of Approved Cyber Forensic Tools** |

|  |  |  |
| --- | --- | --- |
| Tool Name | Version | Date Approved |
| Prefetch Parser | 1.5 | 12/04/2012 |
| FTK Imager | 3.11 | 12/04/2012 |
| FTK Imager Lite | 3.11 | 12/04/2012 |
| RegRipper | 2.5 | 12/04/2012 |
| SDelete | 1.6 | 12/04/2012 |
| CelleBrite | 1.8.1, Image 2.121 | 12/27/2012 |
| UFED Physical Analyzer | 3.1.0.96 | 12/27/2012 |
| UFED Logical Analyzer | 3.1.0.122 | 12/27/2012 |
| --The rest removed for public document -- |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Forensic Hardware Validation – Tableau Forensic USB Bridge** |

**1. Test Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number: | 2012 – 012 | Test Title: | Tableau Forensic USB Bridge | | |
| Test Date: | 09/14/2012 | Person Conducting Test: | | | J. Moulin |
| Test Result: | **PASS** |  | |  | |

**2. Hardware Being Tested**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Tool: | Forensic USB Bridge | | | |
| Manufacturer: | Tableau | | | |
| Website: | http://www.tableau.com/index.php?pageid=products&category=forensic\_bridges | | | |
| Model (or Download Date if no Version): | | | T8-R2 | |
| Serial Number: | | [removed] | | |
| Approved by Cyber (Date): | |  | | |
| IT Tracking Log # (If Freeware / Shareware): | | | | N/A |

**3. Test Platform**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer Name: | | | [removed] | | OS Name and Version: | | | Mac OS X | |
| Hard Drive Capacity: | | | | 320 GB | RAM Capacity: | 8 GB | | | |
| Processor: | Intel Core i5 2.4GHz | | | | S/N or Asset Tag: | | [removed] | | |
| Make / Model: | | MacBook Pro | | |  | | | |  |

**4. Purpose and Scope**

This procedure will test the ability of the Tableau Forensic USB Bridge to write-protect a USB Thumb Drive:

**5. Description of Methodology**

A USB drive containing various files will be connected to the Tableau Write Blocker. Attempts to copy and delete files to/from the USB drive will be made.

**6. Test Notes**

The test drive used during this procedure was seeded with various files, write-protected, and FTK imager was used to create an MD5 hash of the entire drive. This MD5 hash will be compared to a new hash after the testing is complete. Both hashes should be identical, indicating no data was written to or deleted from the drive.

**7. Test Procedure**

1. Add various seed files to the test drive.
2. Connect drive the write blocker and use FTK Imager to get an MD5 hash of the entire drive.
3. Use Windows Explorer to add and delete files to the drive.
   1. Windows will indicate that the “Drive is Write-Protected. Try again.”
4. Disconnect the drive from the write-blocker then reconnect it again.
5. Use FTK Imager to create a new MD5 hash of the drive.
6. Compare the pre and post hashes. They should be identical.

**8. Expected Results**

1. The write-blocker should have prevented all writes to and from the drive.

**9. Test Results**

The actual results were the expected results.

**10. Observations / Concerns**

None.

**11. Limitations**

None.

|  |
| --- |
| **List of Approved Cyber Forensic Write Blockers & Forensic Duplicators** |

|  |  |  |  |
| --- | --- | --- | --- |
| Device | Model | S/N | Date Approved |
| Tableau | Forensic USB Bridge T8-R2 | Removed | 12/04/2012 |
| Tableau | Forensic USB Bridge T8-R2 | Removed | 12/04/2012 |
| Tableau | eSATA Forensic Bridge T35es-R2 | Removed | 12/04/2012 |
| Tableau | Forensic USB Bridge T8-R2 | Removed | 12/04/2012 |
| Tableau | eSATA Forensic Bridge T35es-R2 | Removed | 12/04/2012 |
| Tableau | Forensic Duplicator TD2 | Removed | 12/04/2012 |
| Tableau | Forensic USB Bridge T8-R2 | Removed | 12/04/2012 |
| Tableau | Forensic Duplicator TD2 | Removed | 12/04/2012 |
| Tableau | eSATA Forensic Bridge T35es-R2 | Removed | 12/04/2012 |
| Tableau | Forensic Firewire Bridge T9 | Removed | 12/04/2012 |
| Tableau | Forensic SCSI Bridge T4es | Removed | 12/04/2012 |
| Tableau | Forensic Firewire Bridge T9 | Removed | 12/04/2012 |
| Tableau | Forensic SCSI Bridge T4es | Removed | 12/04/2012 |
| WiebeTech | USB WriteBlocker | Removed | 12/04/2012 |
| WiebeTech | USB WriteBlocker | Removed | 12/04/2012 |
| Tableau | eSATA Forensic Bridge T35es-R2 | Removed | 12/04/2012 |
| Tableau | eSATA Forensic Bridge T35es-R2 | Removed | 12/04/2012 |
|  | -- rest removed for public document -- |  |  |