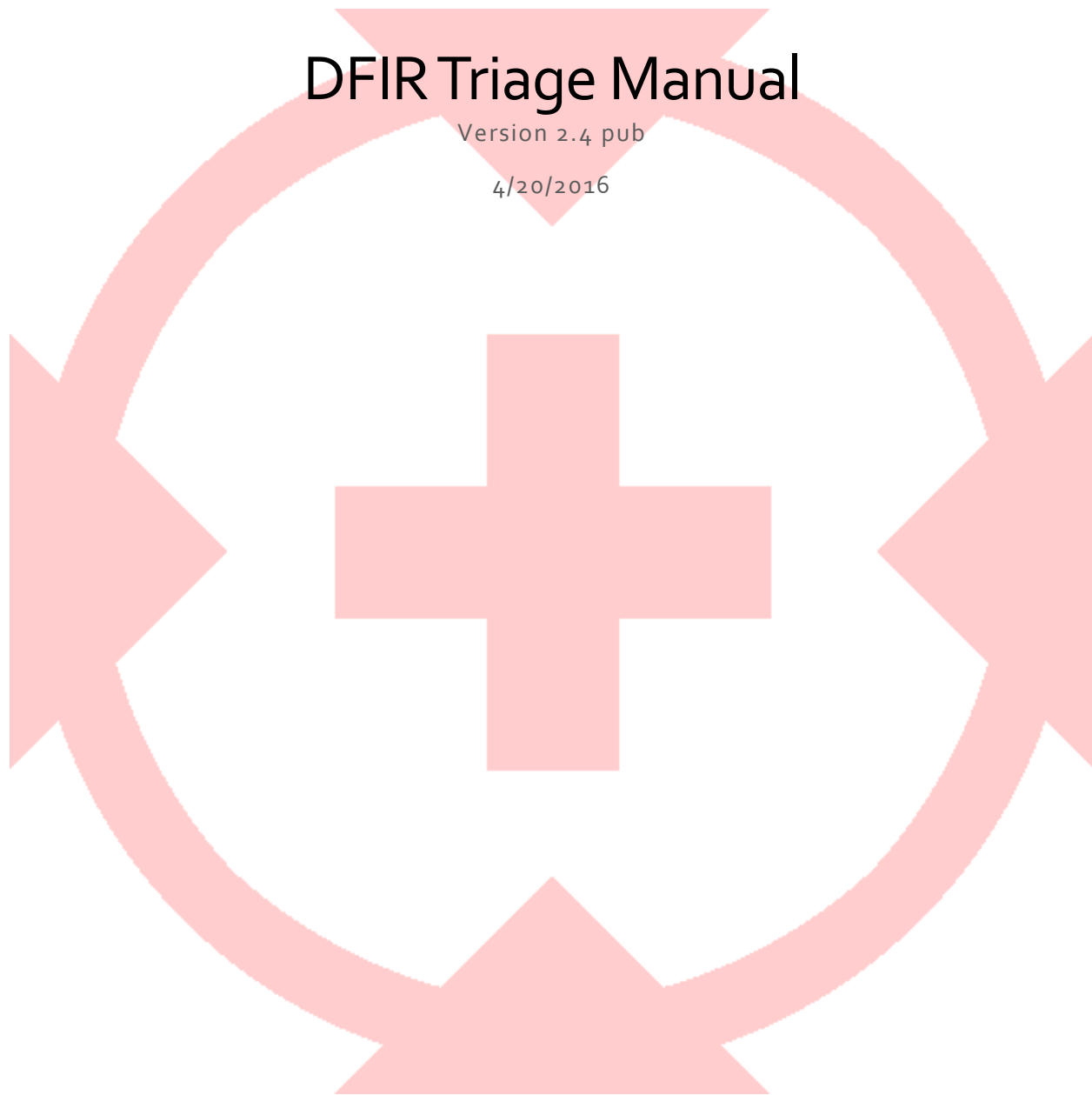


DFIR Triage Manual

Version 2.4 pub

4/20/2016



Contents

Description..... 3

About 3

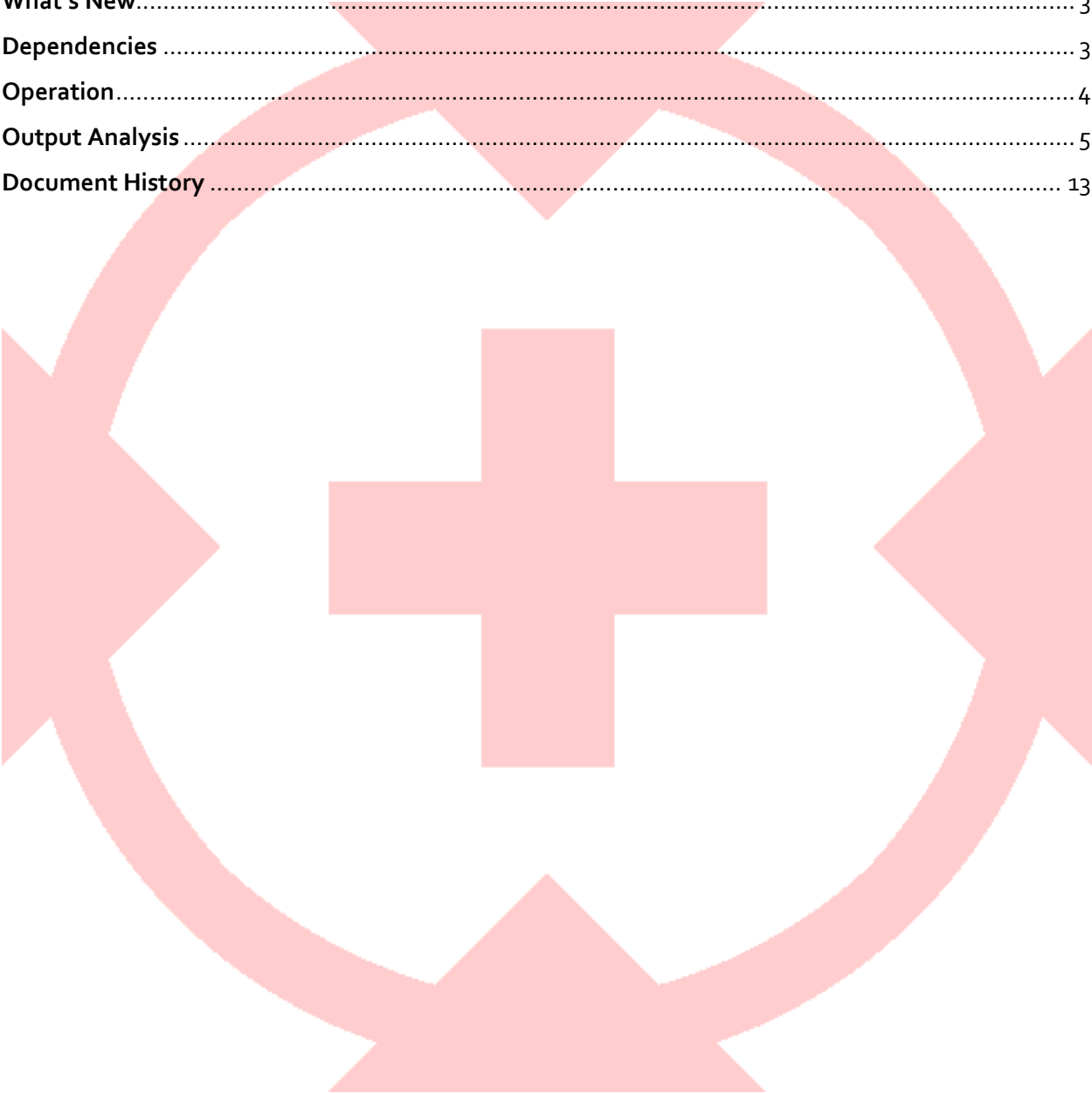
What’s New..... 3

Dependencies 3

Operation..... 4

Output Analysis 5

Document History 13



Description

This document outlines the functionality and proper use of the DFIRTriage tool. Also included is detailed information to help with analysis of the output.

About

DFIRTriage is a python script intended to provide Incident Responders with rapid host data. The python code has been compiled to eliminate the dependency of python on the target host. The tool will run a variety of commands automatically upon execution. The acquired data will reside in the root of the execution directory. DFIRTriage may be ran from a USB drive or executed in remote shell on the target. Windows-only support.

What's New

- MD5 integrity check
- Debug mode to bypass code blocks that perform lengthier analysis to speed up script testing.
- Parsing out specific event log entries. This code has been written so it is easy to add or remove event IDs for future modifications. We are now parsing out 15 specific events from the Event logs. These events are indicate remote connections, possible lateral movement, privilege escalation, service creation, scheduled tasks for possible persistence, and program installation.
- Console color has been added to help differentiate between normal, warning, and error states. (Applies to local execution only)
- Corrected issues with alternate data stream location code
- Parsing registry hives

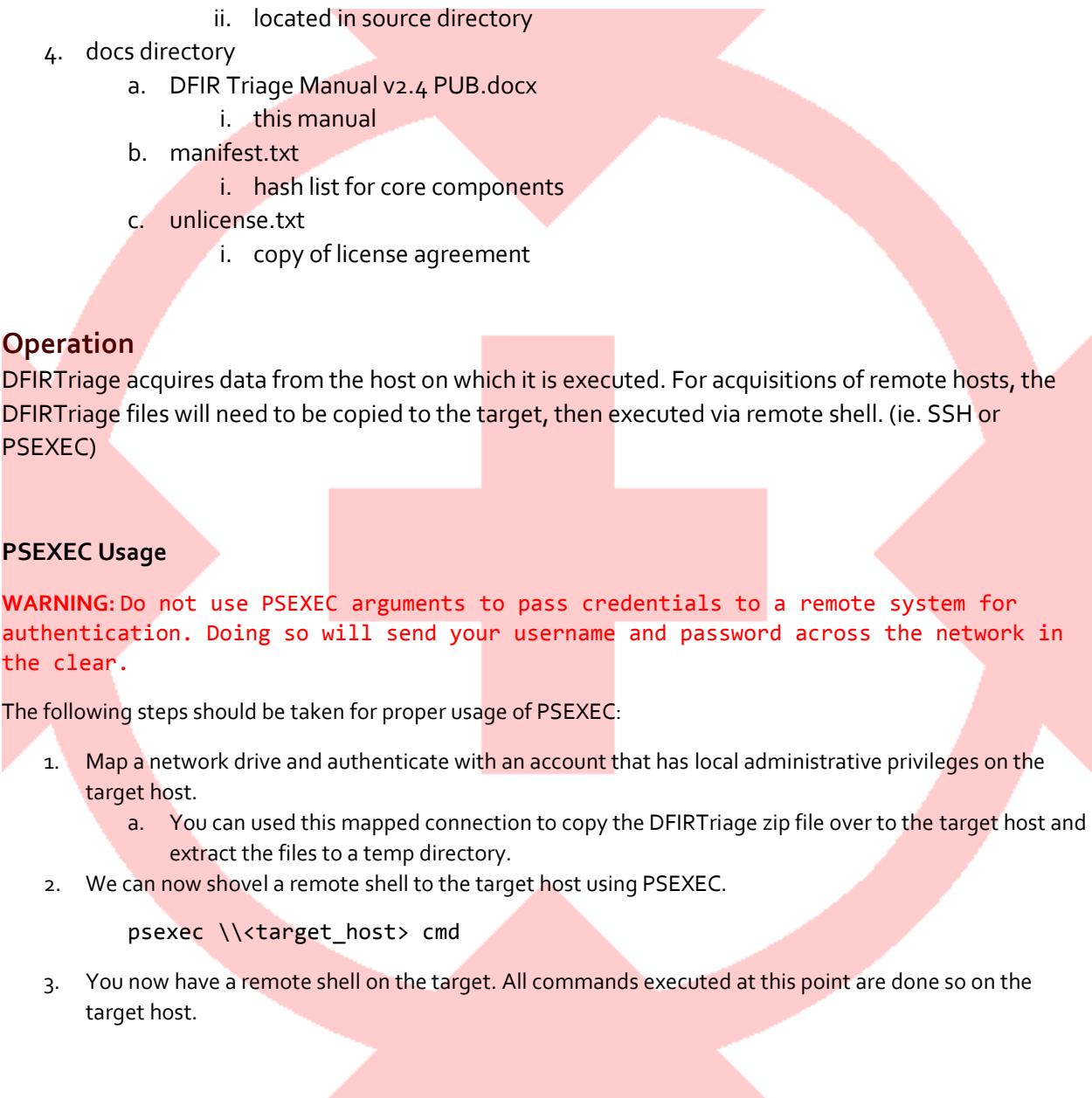
Dependencies

The tool repository contains the full toolset required for proper execution and is packed into a single a single file named "core.ir". This ".ir" file is the only required dependency of DFIRTriage. DFIRTriage is packaged in a zip archived with the following naming convention – "DFIRTriage-pub_2.4.zip", which contains all of the files required for normal operation. Please note that the demo version of select TZWorks tools are used in the public release of DFIRTriage. Licensed copies may be purchased at www.tzworks.com.

The contents of the folder should look like this:

Name	Date modified	Type	Size
DFIRTriage.exe	4/20/2016 12:27 PM	Application	3,122 KB
core.ir	4/12/2016 8:12 AM	IR File	13,410 KB
source	4/20/2016 12:30 PM	File folder	
docs	4/20/2016 12:59 PM	File folder	

1. DFIRTriage.exe

- 
- a. compiled executable
 2. core.ir
 - a. tool set repository (required)
 3. source directory
 - a. DFIRTriage-pub_2.4.py
 - i. source code (not required for execution, dev purposes only)
 - ii. located in source directory
 4. docs directory
 - a. DFIR Triage Manual v2.4 PUB.docx
 - i. this manual
 - b. manifest.txt
 - i. hash list for core components
 - c. unlicense.txt
 - i. copy of license agreement

Operation

DFIRTriage acquires data from the host on which it is executed. For acquisitions of remote hosts, the DFIRTriage files will need to be copied to the target, then executed via remote shell. (ie. SSH or PSEXEC)

PSEXEC Usage

WARNING: Do not use PSEXEC arguments to pass credentials to a remote system for authentication. Doing so will send your username and password across the network in the clear.

The following steps should be taken for proper usage of PSEXEC:

1. Map a network drive and authenticate with an account that has local administrative privileges on the target host.
 - a. You can use this mapped connection to copy the DFIRTriage zip file over to the target host and extract the files to a temp directory.
2. We can now shovel a remote shell to the target host using PSEXEC.

```
psexec \\<target_host> cmd
```
3. You now have a remote shell on the target. All commands executed at this point are done so on the target host.

DFIRTriage Usage

1. Once the remote shell has been established on the target you can change directory to the location of the extracted DFIRTriage.exe file and execute.

NOTE: If running locally and physically at the console of a workstation, DFIRTriage must be executed with Administrative privileges as seen below:




2. Immediately after execution, you will be prompted for memory acquisition as seen below:

```
# (      (      #  
# )\ )   \ ) \ )\ )\ )    *   )          #  
# (( )/( ( )/( (( )/( ( )/( ` ) /(( (      ) ( (      (      #  
# /( _ ) )/( _ ))/( _ ))( _ )   ( _ )( _ )\ ( /( _ ))( _ )\   #  
# ( _ )_ ( _ )_ ( _ )_( _ )   ( _ ( _ )| ( ( _ )_( _ )| ( _ )\ /( _ ) #  
# | _ \ | _ | _ | _ | _ | _ | _ |( _ | _ |( _ )_ ( ( _ | _ ) ) #  
# | | ) | _ | | | | | | | | | | | ' _ | / _ _ / _ _ | / _ _ ) #  
# | _ _ / | _ | _ _ | _ | _ | _ | | _ | _ | _ _ , _ _ , _ \ _ | #  
#                                     | _ _ /          #  
#                               Version 2.4              #  
#                                                         #  
#####  
  
[+] Detecting OS and System Architecture... [64bit system, forcing 32bit]  
  
[+] Verifying core integrity..  
  
[+] Core integrity... [OK]  
  
[+] Has Local Admin rights... [Yes]  
  
[+] Done.  
  
Do you want to acquire memory? (y | n)
```

3. Press **"y"** or **"n"** and then hit ENTER to continue.

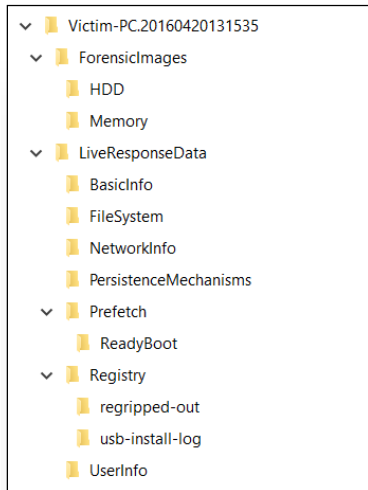
Output Analysis

Once the script has completed, you should find DFIRTriage, core.ir, and an output directory beginning with the hostname of the target.

Name	Date modified	Type	Size
 DFIRTriage2.3	2/10/2016 7:58 AM	Application	3,941 KB
 core.ir	1/7/2016 4:15 PM	IR File	8,913 KB
 VICTIM-PC.20160210083749	2/10/2016 8:38 AM	File folder	

Output Folder

The output folder name includes the target hostname and a date/time code indicating when DFIRTriage was executed. The date/time code format is YYYYMMDDHHMMSS.



The above graphic shows the complete output folder structure created by DFIRTriage. All information and artifacts gathered by the script will reside in this folder structure.

Artifacts List

The following is a general listing of the information and artifacts gathered.

Artifacts	Description
Memory	Raw image acquisition (optional)
Prefetch	Collects all prefetch files and parses into a report
User activity	HTML report of recent user activity
System32 file hash	MD5 hash of all files in root of System32
Network information	Network configuration, routing tables, etc
Extended process list	Processes, PID, and image path
Windows character code page information	Character set that Windows is using
Complete file listing	Full list of all files on the system partition
List of hidden directories	List of all hidden directories on the system partition
Current user information	User running DFIRTriage script
System information	Build, service pack level, installed patches, etc
Windows version	Logs the version number of the target OS
Current date and time	Current system date and time
List of scheduled tasks	List of all configured scheduled tasks
Loaded processes and DLLs	List of all running processes and loaded DLLs
Running processes	Additional information on running processes
Network configuration	Network adaptor configuration
Network connections	Established network connections
Open TCP/UDP ports	Active open TCP or UDP ports
DNS cache entries	List of complete DNS cache contents

ARP table information	List of complete ARP cache contents
Local user account names	List of local user accounts
NetBIOS information	Active NetBIOS sessions, transferred files, etc
Installed software	List of all installed software through WMI
Autorun information	All autorun locations and content
List of remotely opened files	Files on target system opened by remote hosts
Logged on users	All users currently logged on to target system
Alternate Data Streams	List of files containing alternate data streams
Registry hives	Copy of all registry hives
USB artifacts	Collects data needed to parse USB usage info
Hash of all collected triage data	MD5 hash of all data collected by DFIRTriage

Analysis

This section of the manual is provided to offer guidance during analysis of the DFIRTriage script output. There are select output files listed below with guidance on analyzing the contents. These are only guidelines as it is not practical to detail every possible use of this data. The bulk of analysis will depend on context and the analysis skills of the Responder.

Output Directory Root	Analysis Notes
ForensicImages\	See information below for content details.
LiveResponseData\	See information below for content details.
Triage_File_Collection_Hashlist.csv	This file contains the calculated hash value for all data collected by DFIRTriage. This information can be used to verify integrity of the output data.

ForensicImages \ HDD	Analysis Notes
.E01, .dd, etc	The triage script does not acquire a file system image. This folder is here for organizational purposes should one be acquired by another means.

ForensicImages \ Memory	Analysis Notes
memdump.raw	If the option to acquire memory was chosen, this file will be present. Otherwise the folder will be empty. Memdump.raw is a full raw image of volatile memory which should have been acquired before the infected machine was powered off. Memory analysis is beyond the scope of this section. Memory analysis tools such as Redline and Volatility should be used to help identify suspicious processes.

	Multiple tools should be used to validate findings.
--	---

LiveResponseData \ BasicInfo	Analysis Notes
Alternate_data_streams.txt	Use this to review all files on the target system that contain alternate data streams. Alternate data streams can be used to easily hide information, or even entire files while remaining undetected by the user.
current_date_time.txt	Simply used verify that the system date and time on the target system is correct and has not been modified.
eventlogs.csv	This file contains specific event information from the target system. These events identify possible lateral movement, privilege escalation, service creation & scheduled tasks for possible persistence, as well as program installation. New events are easily added to the function code as needed.
Full_file_listing.txt	This report is very helpful in determining if a known folder or file is present on the target system.
Hashes_md5_System32_WindowsPE_and_Dates.txt	MD5 value of suspicious files can be used to search IOC lists.
Hashes_md5_System_TEMP_WindowsPE_and_Dates.txt	MD5 value of suspicious files can be used to search IOC lists.
Hashes_md5_User_TEMP_WindowsPE_and_Dates.txt	MD5 value of suspicious files can be used to search IOC lists.
LastActivityView.html	This report does not collect all activity accurately at times and all findings of interest should be validated through additional analysis.
List_hidden_directories.txt	Log of all directories that have been hidden from the User. This log should be reviewed for suspicious hidden directories in unusual locations (eg. in user temp folders)
PrcView_extended.txt	Log of all running processes on the target system. Includes the process name, PID, priority, and executable path. If bad process name is known, this list can be used to locate. Also, look for processes that are misspelled or running from unusual locations. (eg. explorer.exe running from "system32"). Compare process list to processes running on a known good machine to eliminate normal processes.
PrcView_extended_long.txt	This contains the same information as "prcview_extended.txt", but includes the process arguments that were passed during execution. One use of this information is to look for incorrect or suspicious arguments.

	(eg. svchost.exe running without the "-k" parameter).
psfile.txt	Review information to determine if there are any files opened remotely on the target host.
psinfo.txt	This file contains information on the target host system such as uptime, kernel version, service pack level, processor and memory information, etc.
pslist.txt	Additional information on processes running on the target host. The thread and handle count columns should be reviewed to identify anomalies (eg. an instance of csrss.exe with only 12 handles) as well as CPU time to determine when a process actually started. (eg. csrss.exe, smss.exe, and wininit.exe should start seconds after boot while "iexplore.exe" should start when launched by the user. If "csrss.exe" started 2 minutes after boot, it is a red flag and should be investigated further.)
PsLoggedon.txt	Use this information to help identify any users (local or remote) who are authenticated to target system.
Running_processes.txt	Additional process information to cross-check with previously gathered process data.
system_info.txt	Target system information. Similar to the information gathered in "psinfo.txt", but more verbose.
Windows_codepage.txt	This file contains the active code page identifier on the target system. The typical North American build should have a code page value of "437". This is typically not an issue, but modifying this value will cause data corruption.
Windows_Version.txt	Contains the version of Windows running on the target system.

LiveResponseData \ NetworkInfo	Analysis Notes
ARP.txt	This file contains the ARP cache from the target system. While the ARP protocol is not routable to the internet, it can help to identify additional hosts on a network that may have been compromised or that may have been used to launch the internal attack.
cports.html	This is a very detail report showing TCP/UDP connections on the target host. Additionally, you have information on the process that created the connection (name, PID, etc), the Window Title (if exists), and more.
DNS_cache.txt	This is a log file of the target system DNS cache. Malware generally has the ability to

	connect to the network in order to do things like gathering additional exploits, join a command & control infrastructure, wait for more commands, etc. It is common for malware to be coded with domain names which must be queried and resolved before it can connect. This information can be found in the DNS cache.
Internet_settings.txt	This is a log of the local network adapter configuration on the target host. This log should be reviewed to ensure the settings are correct and have not been altered. (Eg. Suspicious domains added to the DNS Suffix Search List)
nbtstat.txt	This file contains NetBIOS names cache from the target system. This can be very useful if native Windows networking components have been utilized to transmit data over the network (eg. Windows File Sharing).
NetBIOS_sessions.txt	This file will contain information on any current NetBIOS sessions to the target host.
NetBIOS_transferred_files.txt	This log will show if any files were transferred over the network from the target host using the "net file" command.
netstat_anb_results.txt	This file contains network connection information for processes running on the target host. This information can be correlated with the "cports.html" found in the same output directory.
Open_network_connections.txt	This file also contains TCP/UDP connection information. The process PID and connection state information is also available. While it may seem redundant, it is absolutely essential to identify current and recent network activity. Some of these tools may capture information that the others miss. All findings should be validated.
routing_table.txt	This file contains the routing table of the target host. This information should be reviewed to ensure it has not been modified with additional routes or a modified gateway. Comparing this information to the routing table from a known good machine may be helpful.
Tcpvcon.txt	Additional information on network connections from target host. Contains protocol type (TCP/UDP), process name, PID, state, local address, and remote address.

autorunsc.txt	This information will show all of the programs that Windows will automatically execute when starting up. This is a very common method used by malware to maintain persistence on a system. This data can be reviewed for suspicious file names and paths.
Loaded_dlls.txt	This file contains a process listing which includes all loaded DLLs for each running process. Persistence can be gained by injecting a malicious DLL into a normal Windows process. This data should be examined for suspicious DLLs. It is very helpful to have a list of loaded DLLs from a known good system to use for comparison.
scheduled_tasks.txt	This file contains all scheduled tasks found on the target system. Inserting a scheduled task into the target host is a common method used by malware to maintain persistence on the victim machine. This information should be reviewed for suspicious tasks.
services_aw_processes.txt	This file provides a list of services running on the target system, with the associated process name and PID. Rogue services are another persistence mechanism that can be utilized by malware.

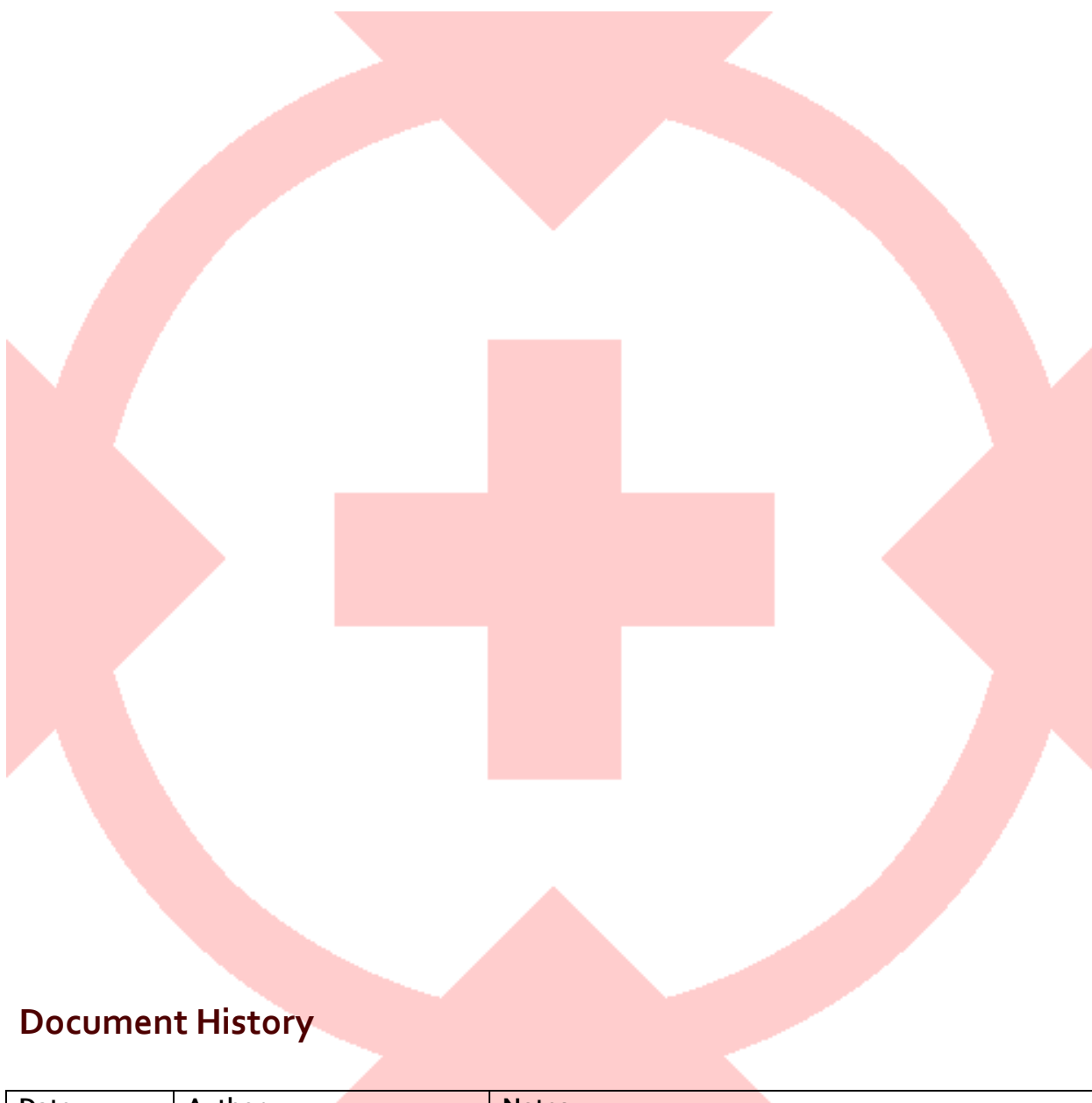
LiveResponseData \ Prefetch	Analysis Notes
.pf	The Prefetch directory contains the actual prefetch files found on the target system. This data is collected and then parsed later in the DFIRTriage process. The filenames of the prefetch files will give you an indication of which programs were recently executed. Especially useful if you already have a binary name from an external source.

LiveResponseData \ Registry	Analysis Notes
NTUSER	A copy of the user registry hive (NTUSER.dat) for the logged in user on the target system. NTUSER.dat contains information on general user behavior such as recently viewed documents, typed URLs, mount points, mapped drives, local search terms, uninstalled software, and more. This file can be parsed with Regripper for easier analysis.
SAM	A copy of the Security Accounts Manager registry hive (SAM) from the target system. The SAM registry file contains local user and group information such as Security Identifiers (SID) for local accounts and groups, account

	and group creation and deletion information. This file can be parsed with Regripper for easier analysis.
SECURITY	A copy of the Security registry hive (SECURITY) from the target system. The SECURITY registry hive contains account and system security information such as local security policies, user rights assignments, password policies, and more. The SECURITY hive is linked to the SAM hive for update accuracy. This file can be parsed with Regripper for easier analysis.
setupapi.dev.log	This is a copy of the device installation log from the target system. This log, in correlation with the SYSTEM registry hive, can be used to determine the first time a removable device (eg. USB drive) was plugged into the system.
SOFTWARE	A copy of the Software registry hive (SOFTWARE) from the target system. The SOFTWARE registry hive contains information about installed software, uninstalled software, file extension associations, last logged on user, and more. This file can be parsed with Regripper for easier analysis.
SYSTEM	A copy of the System registry hive (SYSTEM) from the target system. The SYSTEM registry hive contains information specific to the software and hardware configuration of the target system. For example, the SYSTEM registry contains system startup parameters, device driver configurations, hardware configurations, timezone settings, computer names, USB connections and pointers, and more. This file can be parsed with Regripper for easier analysis.
rr.<registry>-out.txt	The "regripped-out" directory contains Regripper output files for each of the registry hives. Eg. rr.sam-out.txt contains the Regripper output for the SAM registry hive.

LiveResponseData \ PersistenceMechanisms	Analysis Notes
List_users.txt	This file simply contains a list of all local users on the target system. This file can be reviewed for suspicious local accounts.
prefetch-out.txt	This file contains parsed data from the prefetch files collected from the target system. Information such as file name, modified, accessed, and created times, number of times executed, last run time, and

	all loaded DLLs and other dependant files used during execution.
whoami.txt	This file contains the user account that is executing the DFIRTriage script.



Document History

Date	Author	Notes
2-10-16	Travis Foley	Public release v2.3
4-14.16	Travis Foley	Public release v2.4