MFTDump Forensic Tool
Member of the Malware-Hunters Forensic Toolkit

Quick-Start Guide

Version 1.3.0
September 12, 2012

Michael G. Spohn
mspohn@malware-hunters.net

Table of Contents

Introduction

This document describes the features of the *MFTDump* forensic tool. This tool provides a quick and easy way to extract forensic metadata from an NTFS volume $MFT file. It is designed to supplement your forensic tools such as EnCase, FTK, Hex-Ways Forensic, etc. Be sure to read the *MFTDump* FAQ document to learn more about the design of the tool.

The tool has the following features:

- Lightweight, fast, and flexible command line tool.
- Extracts NTFS file metadata from an $MFT file.
- Dumps filenames to stdout for fast searches.
- Dumps alternate data streams to stdout.
- Has three output report formats: short, standard, and long.
- Zip feature reduces size of output report on disk.
- Self-contained binary – no other dependencies.
- Runs on Windows 2000, XP, Vista, 7, Server 2003 and 2008.

The tool is used by forensic examiners and incident responders who need a quick method to extract and examine file metadata from an NTFS volume. Common uses include:

- Searching an NTFS volume for specific file name(s).
- Identifying alternate data streams (ADS).
- Identifying file attributes such as deleted, hidden, system, etc.
- Searching and sorting files based on MAC times (Modified, Accessed, and Created).
- Creating a timeline of activity on a filesystem.

Tool Use

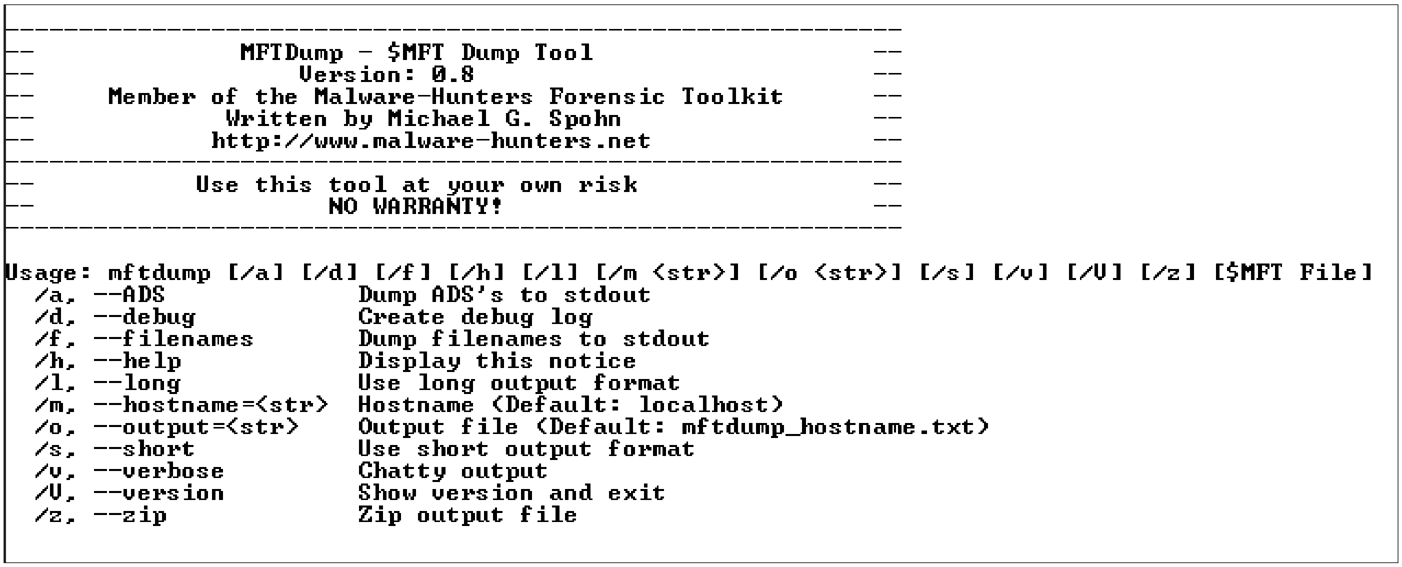
MFTDump is designed to be fast and easy to use. All you need is the tool binary and an $MFT file extracted from a forensic image or a live system. All capable forensic tools such as EnCase, FTK, Hex-Ways Forensic and the Sleuthkit can extract the $MFT from an NTFS volume forensic image.

You can extract the $MFT file from a dormant host by booting it using a live Linux bootable CD or thumb drive such as Helix, or any other live Linux distro. On live systems, my tool of choice for grabbing a copy of an $MFT is Access Data's handy FTK Imager or HBGary's free FGet tool.

Once you have the $MFT file you want to examine, simply run *MFTDump* passing the name of the $MFT file on the command line.

If you run the tool without any command line parameters, you will see a usage printout shown in Figure 1.

Table 1: MFTDump usage printout



```
--------------------------------------------------------------
--              MFTDump - $MFT Dump Tool                    --
--                  Version: 0.8                            --
--      Member of the Malware-Hunters Forensic Toolkit      --
--             Written by Michael G. Spohn                  --
--             http://www.malware-hunters.net               --
--------------------------------------------------------------
--            Use this tool at your own risk                --
--                     NO WARRANTY!                         --
--------------------------------------------------------------

Usage: mftdump [/a] [/d] [/f] [/h] [/l] [/m <str>] [/o <str>] [/s] [/v] [/V] [/z] [$MFT File]
  /a, --ADS                Dump ADS's to stdout
  /d, --debug              Create debug log
  /f, --filenames          Dump filenames to stdout
  /h, --help               Display this notice
  /l, --long               Use long output format
  /m, --hostname=<str>     Hostname (Default: localhost)
  /o, --output=<str>       Output file (Default: mftdump_hostname.txt)
  /s, --short              Use short output format
  /v, --verbose            Chatty output
  /V, --version            Show version and exit
  /z, --zip                Zip output file
```

Providing only an $MFT filename with no switches results in an output report file named ‘mftdump_localhost.txt’ using the ‘standard’ report format. This file is tab-delimited text that can be imported into Excel**. Note:** $MFT files usually have hundreds of thousands of files. Excel versions prior to 2007 have a 65k row limit.

The command line switches *MFTDump* uses are described in Table 2 below:

Table 2: MFTDump command line switches

Switch	Description
/a	Dump Alternate Data Streams (ADS) to stdout
/d	Run in debug mode - creates a log file named MFTDump.log
/f	Dump all filenames to stdout (Note: Directory names not included).
/h	Prints usage text and exits.
/l	Create an output report using long report format.
/m	Use the provided hostname string in output filename and report hostname field.
/o	Use the provided filename as the output filename.
/s	Create an output report using short report format.
/v	Verbose mode - describes application actions.
/V	Prints tool version number and exits.
/z	Zip output report file.

Report Formats

MFTDump provides three report formats; short, standard, and long. If you do not provide the /s (short) or /l (long) switches on the command line, the output report will be in the standard format. The report fields in the three report formats are shown in the below tables.

Table 3: Short Report Format

Field Name	Description
RecNo	$MFT file record number (zero based)
Deleted	Deleted flag
Directory	Directory flag
ADS	Alternate Data Stream flag
Filename	Win32/Posix file name
siCreateTime (UTC)	$STANDARD_INFORMATION attribute create time
siAccessTime (UTC)	$STANDARD_INFORMATION attribute access time
siModTime (UTC)	$STANDARD_INFORMATION attribute modified time
siMFTModTime (UTC)	$STANDARD_INFORMATION attribute MFT modified time
ActualSize	Logical size of file on disk
Ext	File extension
FullPath	Full path of file (NOTE: May not be accurate on deleted files
ReadOnly	Read-only flag
Hidden	Hidden flag
System	System flag
Hostname	Hostname (Default = 'hostname') /m parameter

Table 4: Standard Report Format

Field Name	Description
RecNo	$MFT file record number (zero based)
Deleted	Deleted flag
Directory	Directory flag
ADS	Alternate Data Stream flag
Filename	Win32/Posix file name
siCreateTime (UTC)	$STANDARD_INFORMATION attribute create time
siAccessTime (UTC)	$STANDARD_INFORMATION attribute access time
siModTime (UTC)	$STANDARD_INFORMATION attribute modified time
siMFTModTime (UTC)	$STANDARD_INFORMATION attribute MFT modified time
ActualSize	Logical size of file on disk
AllocSize	Physical size of file on disk
Ext	File extension
FullPath	Full path of file (NOTE: May not be accurate on deleted files
fnCreateTime (UTC)	$FILE_NAME attribute create time
fnModTime (UTC)	$FILE_NAME attribute access time
fnAccessTime (UTC)	$FILE_NAME attribute modified time
fnMFTModTime (UTC)	$FILE_NAME attribute MFT modified time
ReadOnly	Read-only flag
Hidden	Hidden flag
System	System flag
Hostname	Hostname (Default = 'hostname') /m parameter

Table 5: Standard Report Format

Field Name	Description
RecNo	$MFT file record number (zero based)
Deleted	Deleted flag
Directory	Directory flag
ADS	Alternate Data Stream flag
Filename	Win32/Posix file name
DOSFilename	DOS filename
siCreateTime (UTC)	$STANDARD_INFORMATION attribute create time
siAccessTime (UTC)	$STANDARD_INFORMATION attribute access time
siModTime (UTC)	$STANDARD_INFORMATION attribute modified time
siMFTModTime (UTC)	$STANDARD_INFORMATION attribute MFT modified time
ActualSize	Logical size of file on disk
AllocSize	Physical size of file on disk
Ext	File extension
FullPath	Full path of file (NOTE: May not be accurate on deleted files
fnCreateTime (UTC)	$FILE_NAME attribute create time
fnModTime (UTC)	$FILE_NAME attribute access time
fnAccessTime (UTC)	$FILE_NAME attribute modified time
fnMFTModTime (UTC)	$FILE_NAME attribute MFT modified time
ReadOnly	Read-only flag
Hidden	Hidden flag
System	System flag
Resident	Resident flag
Archive	Archive flag
Compressed	Compressed flag
Device	Device flag
Encrypted	Encrypted flag
Indexed	Indexed flag
Normal	Normal flag
Offline	Offline flag
ReparsePoint	Reparse point flag
SparseFile	Sparse flag
Temporary	Temporary flag
Hostname	Hostname (Default = 'hostname') /m parameter

Future Enhancements

MFTDump was an interesting tool to develop. You really never appreciate the design of NTFS until you dive into the on-disk structures and code data parsers. NTFS is a complex beast. I am committed to making the tool better based on your feedback.

Below is a list of future enhancements I am considering:

- Export of $MFT metadata to a SQLite database.
- Export of $MFT metadata to a SQL database import script. (MySQL, Access, Oracle, etc).
- Export of $MFT metadata to XML.
- Ability to parse an $MFT file from a live host.

Please send feedback, bug reports, and enhancement requests to me at mspohn@malware-hunters.net.