Frequently Asked Questions

What is MFTDump?

MFTDump is a command line forensic tool that extracts NTFS volume metadata from an $MFT file.

What problem are you trying to solve?

I have been unable to find a robust, reliable, easy to use, fast, and free $MFT forensic analysis tool – so I wrote one. Forensic tools such as EnCase, FTK Imager, and Hex-Ways Forensics all have the ability to extract $MFT metadata and I use them in my everyday work.

The two main drawbacks of these tools are they are overkill for simple MFT analysis and they are expensive. MFTDump is lightweight, making it perfect for incident response (IR) work. Best feature? It does not require a dongle!

What does the tool do?

The main purpose of this tool is to provide NTFS metadata in a flexible format. Forensic examiners and incident responders understand the value of filesystem timeline analysis. *MFTDump* extracts the critical metadata from an $MFT file to a tab delimited text file. This file can be viewed and sorted in Excel or imported into a database. The tool also dumps $MFT file names and alternate data streams (ADS) to stdout for quick searches.

What operating systems are supported?

I have tested the tool with $MFT files from Windows 2000, XP, Vista, 7, and Windows Server 2003 and 2008. It also runs on all these platforms. If you need to run it on a Windows 2000 system you will need the Microsoft redistributable DLL file msvcr90.dll.

What programming language is the tool written in?

The tool is written in C++ and compiled with Microsoft Visual Studio 2008. It is object oriented (e.g. C++ classes) and contains optimized C code for speed. It uses the open source cross-platform *wxWidgets* framework. It is also linked with Brain Carrier’s excellent SleuthKit forensic library written in C.

How do I use the tool?

The tool is very simple to use. Simply extract the $MFT file from a forensic image and provide the filename as a parameter when you run the tool. You can also extract the $MFT from a live system using Access Data’s free FTK Imager tool or HBGary’s excellent and free FGet tool. Depending on the command line switches you choose, the tool will either dump filename/ADS information to stdout or create a tab delimited file containing the $MFT metadata.

Is there anything I need to be aware of when using this tool?

Yes. You *cannot* rely on the file path information field for deleted files. That is why the path information provided for deleted files have “?” marks at the beginning and end.

An NTFS volume has a number of metadata structures in addition to the $MFT file. For example, the volume boot record (VBR) contains important information about the $MFT structure and location. File metadata is maintained in non-resident $INDEX_ALLOC attributes.

Since *MFTDump* only has access to the $MFT file, it is not possible to 'chase' down deleted files through the $INDEX_ALLOC structures to determine if the file is an orphan. Instead, the tool uses the resident $FILE_NAME attribute to determine its parent folder, and follows the folder path to the root folder. In the case of deleted files, this information may or may not be accurate. To determine the exact status of a deleted file, you need to analyze the file system in a forensic tool.

Despite the fact the path to a deleted file cannot be accurately determined; rest assured the file *did* exist on the filesystem.

A third thing to realize about the tool is there is no way to search for deleted $MFT file records in unallocated space similar to way FTK Imager does.

Is there a fee for using this tool?

No. I build a lot of tools for incident response work. In the spirit of giving back to the digital community that provides us so many incredible free tools, I provide this tool at no cost to anyone that finds it useful.

This is a pretty cool tool. Are there any plans for enhancements?

Yes. The next release of the tool will provide the ability to extract the $MFT data to a SQLite database. This will provide a more robust way to index/sort the data. I have analyzed $MFT files from NTFS volumes that contain close to a million files. That is a lot of data to be slinging around in Excel.

I also plan on enhancing the tool to dump $MFT files from live systems. This will provide an incredibly efficient way to search live hosts for files during an incident response.

Since the tool uses the cross-platform wxWidgets platform, it should be a straightforward port to Linux. I will consider creating a Linux version of the tool is there is a demand for it.

Any other suggestions for improvement are most welcome.

Who created this tool?

My name is Michael Spohn. I am a professional incident response consultant based in the Los Angeles area. You can email me at mspohn@malware-hunters.net.

Where can I learn more about this project?

The project is well documented. You can find the project docs at www.malware-hunters.net.