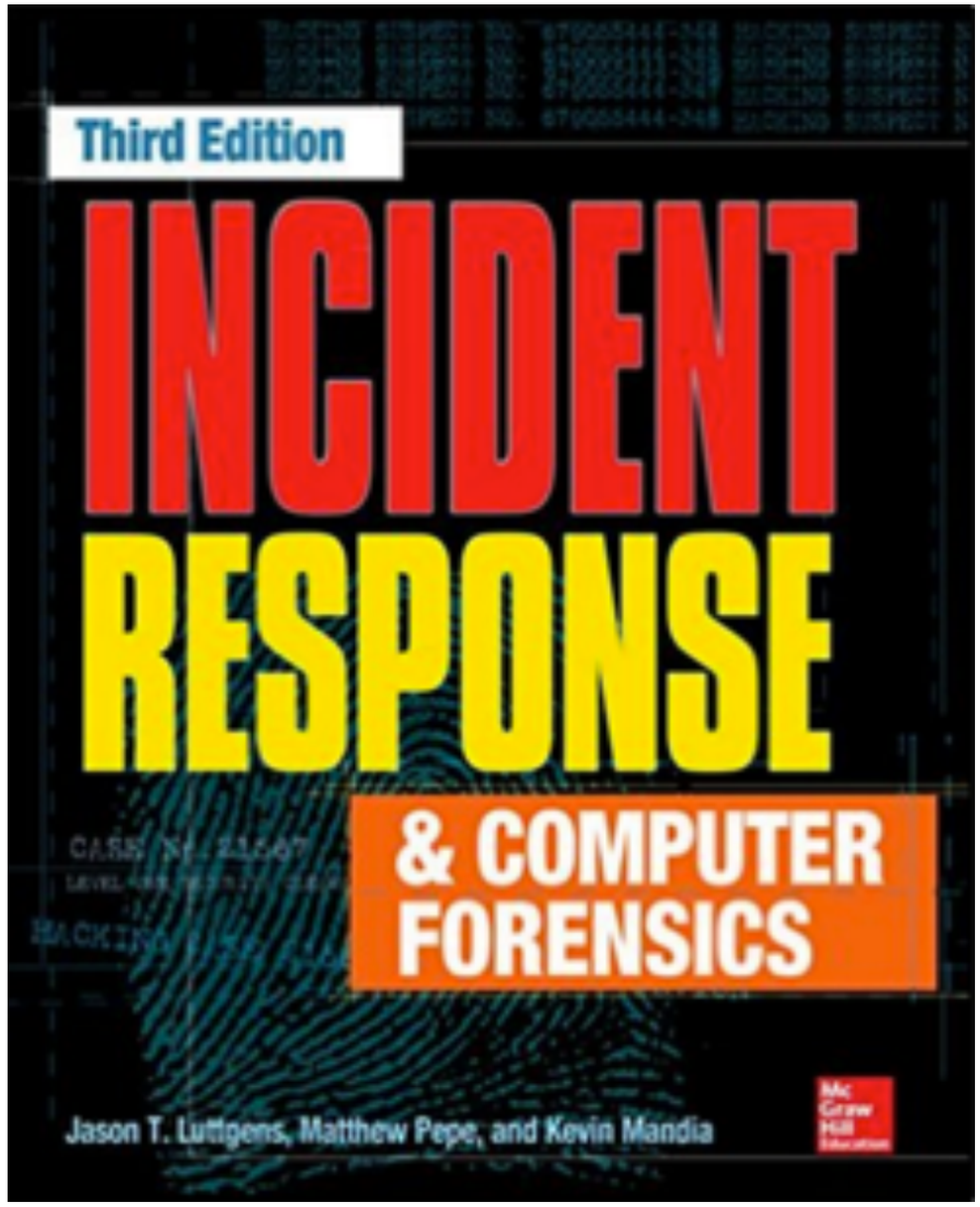
CNIT 152: Incident Response

64



12 Investigating Windows Systems (Part 1)

Updated 11-3-22

Ch 12 Part 1

- **NTFS and file system analysis**
- **Windows prefetch**
- **Event logs**
- **Scheduled tasks**

Ch 12 Part 2

- **The registry**

Ch 12 Part 3

- **Other artifacts of interactive sessions**
- **Memory forensics**
- **Alternative persistence mechanisms**

NTFS and File System Analysis

NTFS and FAT

- **FAT was the old file system used by MS-DOS, Windows 95, Windows 98**
- **NTFS was the replacement**

Master File Table
(MFT)

- **Defines how disk space is allocated and utilized**
- **How files are created and deleted**
- **How metadata is stored and updated**

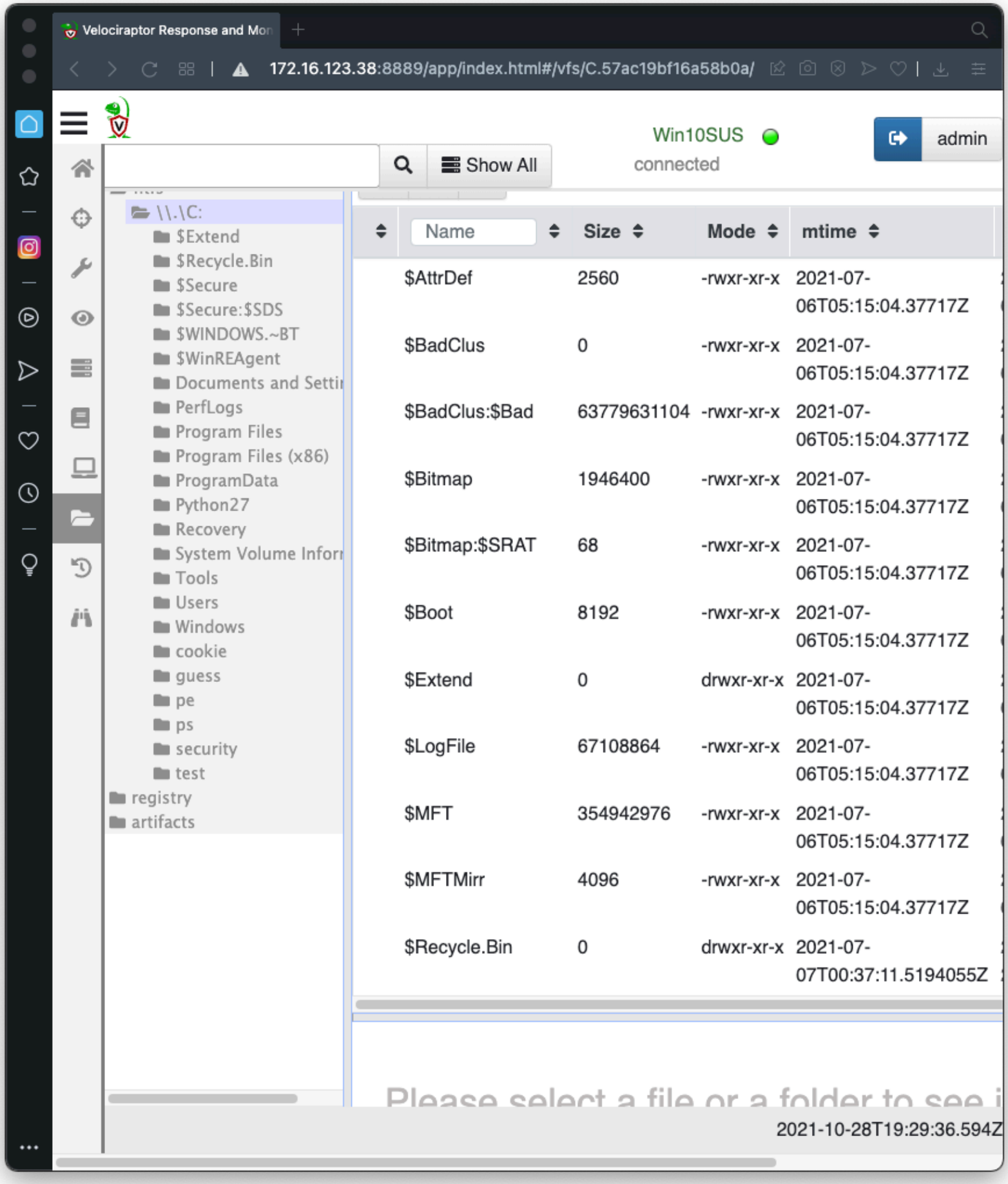
MFT Contents

- **Primary source of metadata in NTFS**
- **Contains or references everything about a file**
  - **Timestamps**
  - **Size**
  - **Attributes (such as permissions)**
  - **Parent directory**
  - **Contents**

The Evidence

- **Each NTFS volume has its own MFT**
- **Stored in the volume root as a file named $MFT**
- **You need raw disk access to acquire $MFT**
  - **It's not accessible through Windows Explorer or standard API calls**

$MFT in Velociraptor



MFT Structure

- **On a standard hard drive with 512-byte sectors**
- **A series of 1024-byte records or "entries"**
- **One for each file and directory on a volume**
- **First 16 entries are reserved for essential NTFS artifacts**
  - **$MFT itself, $LogFile, and more**

MFT in WinHex

Drive E:

\ 2 min. ago

Name	Ext.	Size	Created	Modified	Accessed	Attr.	1st sector
$Extend		448 B	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	68,288
(Root directory)		4.1 KB	03/06/2014 06:55:...	03/06/2014 07:04:...	03/06/2014 07:04:...	SH	102,478
$AttrDef		2.5 KB	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	68,250
$BadClus		0 B	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	
$Bitmap		25.0 KB	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	102,486
$Boot		8.0 KB	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	0
$LogFile		2.0 MB	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	64,154
$MFT		64.0 KB	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	68,266
$MFTMirr		4.0 KB	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	102,399
$Secure		0 B	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	
$UpCase		128 KB	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	102,536
$Volume		0 B	03/06/2014 06:55:...	03/06/2014 06:55:...	03/06/2014 06:55:...	SH	
FILE1.TXT	TXT	1.0 KB	03/06/2014 07:04:...	03/06/2014 06:56:...	03/06/2014 07:04:...	A	96,585
FILE2.TXT	TXT	1.0 KB	03/06/2014 07:04:...	03/06/2014 06:56:...	03/06/2014 07:04:...	A	96,587

MFT Entry Contents

- **Record type (file or directory)**
- **Record # (integer)**
- **Parent record #**
- **Active/Inactive flag**
  - **Deleted files are inactive**
- **Attributes (metadata)**

Attributes

- **$STANDARD_INFORMATION**
- **$FILE_NAME**
- **$DATA**

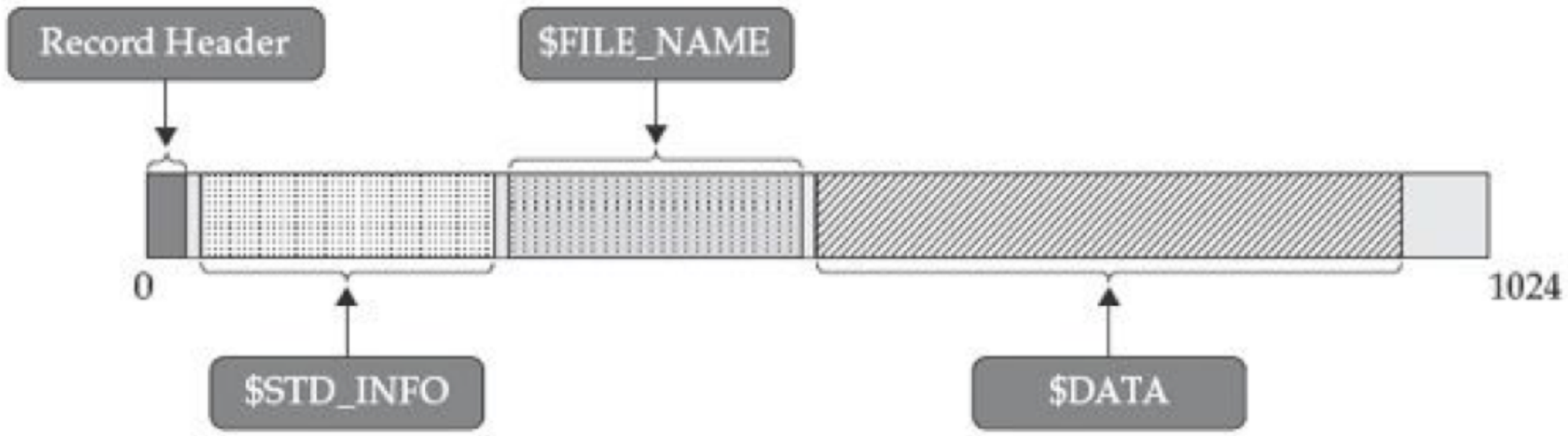
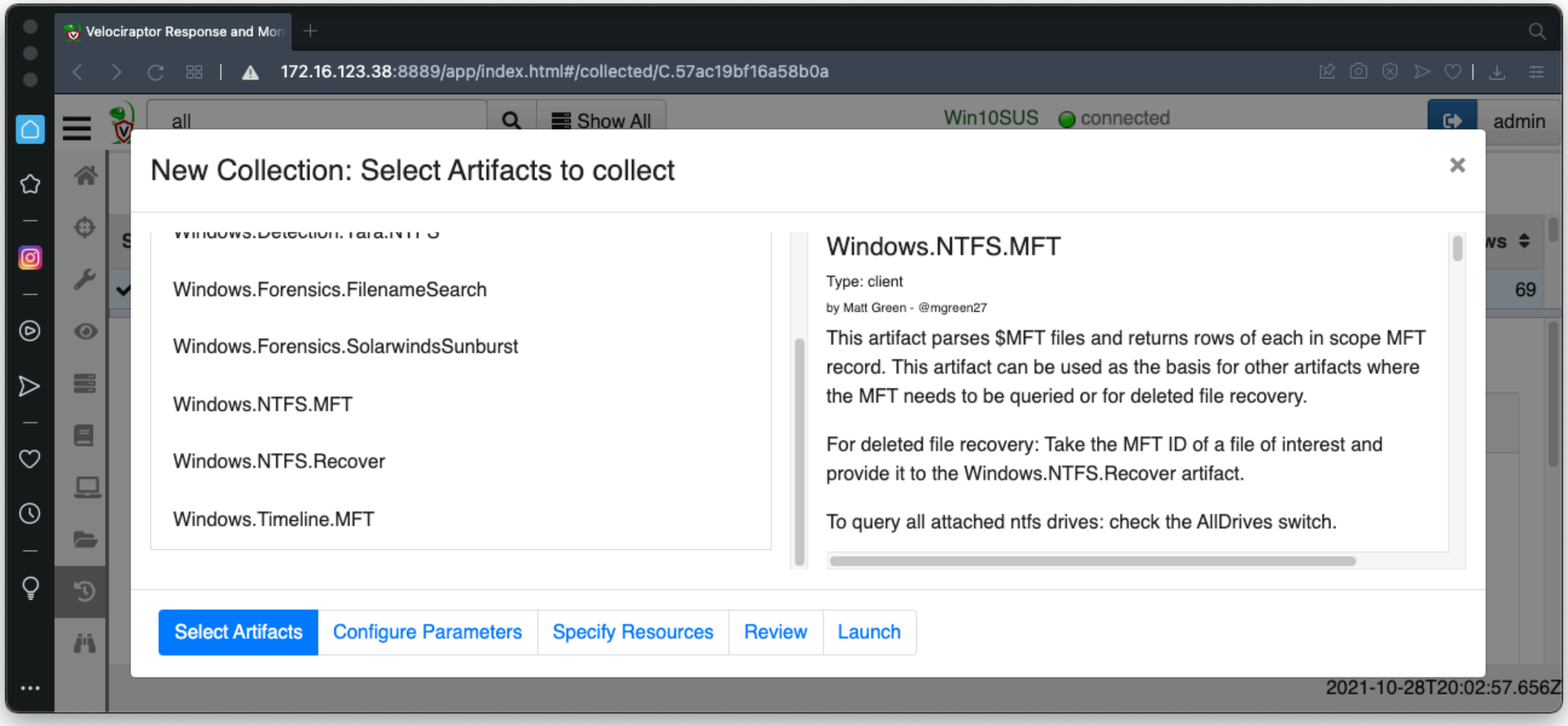


Figure 12-1. Overview of the structure of an MFT file record

MFT Records in Velociraptor and Deleted File Recovery



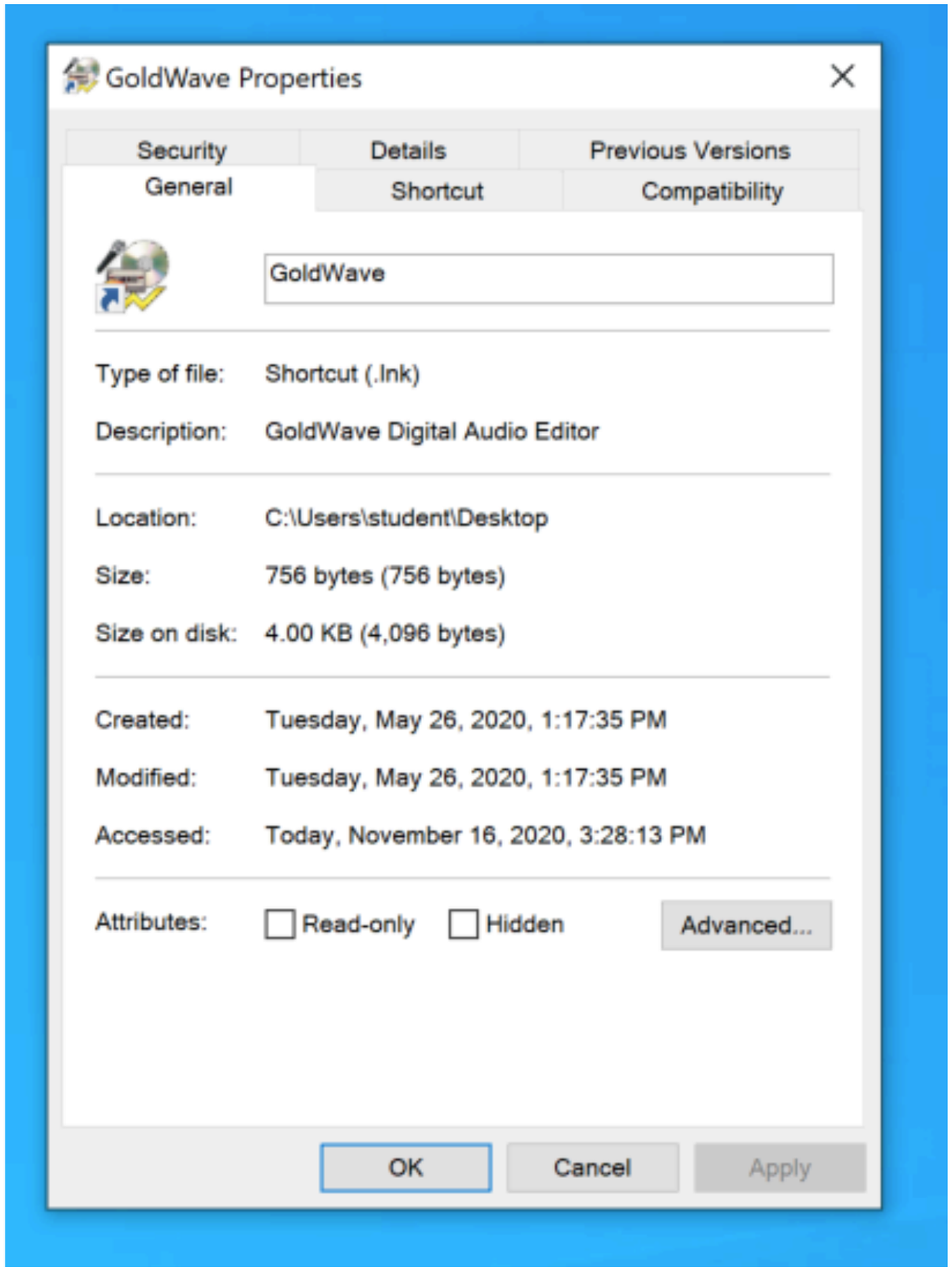
Deleted Files

- **Deleting a file causes its MFT record to be marked "inactive"**
- **Nothing else is changed, until this record is re-used**
- **The file's contents and its metadata can be recovered**
- **But NTFS will always re-use an existing MFT entry before creating a new one**
- **So inactive entries only last for seconds or minutes on the operating system volume**

Timestamps

- **MACE timestamps**
  - **Modified, Accessed, Created, Entry Modified**
- **An MFT entry will always have at least two sets of attributes containing MACE timestamps**
  - **STANDARD_INFORMATION (also known as $SIA or $SI)**
  - **FileName (also known as FNA, FILE_NAME, or $FN)**

- **These are Standard Information ($SI) timestamps**
  - **Created**
  - **Accessed**
  - **Modified**
- **Entry Modified timestamp not visible in Windows Explorer**
- **Forensic tools like SleuthKit, EnCase, and FTK show it**



MACE Timestamps

- **Modified** When the contents of a file were last changed
- **Accessed** When the contents of a file were last read
- **Created** When the file was “born”
- **Entry Modified** When the MFT entry associated with a file, rather than the contents of the file, was changed

Accessed Timestamp

- **Versions of Windows after Windows XP no longer update the Accessed timestamp by default**
- **It can be enabled with a registry change, but even when it's enabled, NTFS may delay updates by up to an hour**
  - **Link Ch 12a**

$FN Timestamps

- **Refer to the MFT entry for the filename itself**
- **NTFS actually maintains multiple sets of file name attributes**
  - **Full, case-sensitive long filename**
  - **MS-DOS 8.3 short file name**

Time-Stomping

- **Only the $SI timestamps are available to user applications through the Windows API**
- **Programs can only alter those timestamps**
  - **A processes called "time-stomping"**
- **Setmace can alter all the timestamps (link Ch 12b)**
- **Malware droppers and installers often automate this process, inserting timestamps from system files to hide in the timeline**

$SI and $FN Timestamps

- **$SI timestamps are easily altered**
- **$FN timestamps require a complex and indirect process to modify**
- **Inconsistencies may remain between the $SI and $FN timestamps**

	SIA	SIA	SIA	SIA	FN	FN	FN	FN
Name	Created	Modified	Accessed	Entry Modified	Created	Modified	Accessed	Entry Modified
Rasmon.dll	02/28/2006 12:00:00	04/13/2008 21:42:10	01/15/2010 05:29:55	07/28/2009 10:12:38	05/18/2009 08:04:51	05/18/2009 08:04:51	05/18/2009 08:04:51	05/18/2009 08:04:51
Wmiprop.dll	02/28/2006 12:00:00	02/28/2006 12:00:00	01/08/2009 18:12:21	01/08/2009 18:12:21				
Msscp.dll	02/28/2006 12:00:00	12/04/2006 08:21:50	01/14/2010 10:40:45	07/28/2009 10:24:24				
Msscp.dll	02/28/2006 12:00:00	12/04/2006 08:21:50	05/15/2009 01:03:24	05/15/2009 01:03:24				

Stomped

Correct

Figure 12-3. Timestamp manipulation example

SANS Windows Artifact Analysis: Evidence of...

Created for FOR408 – Windows Forensics – SANS Digital Forensics and Incident Response faculty created the "Evidence of..." categories to map a specific artifact to the analysis question that it will help to answer. Use this poster as a cheat-sheet to help you remember where you can discover key items to an activity for Microsoft Windows systems for intrusions, intellectual property theft, or common cyber-crimes.

File Download

Open/Save MRU

Description:

In simplest terms, this key tracks files that have been opened or saved within a Windows shell dialog box. This happens to be a big data set, not only including web browsers like Internet Explorer and Firefox, but also a majority of commonly used applications.

Location:

- **XP** NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
- **Win7** NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePIDlMRU

Interpretation:

- The "*" key – This subkey tracks the most recent files of any extension input in an OpenSave dialog
- .??? (Three letter extension) – This subkey stores file info from the OpenSave dialog by specific extension

E-mail Attachments

Description:

The e-mail industry estimates that 80% of e-mail data is stored via attachments. E-mail standards only allow text. Attachments must be encoded with MIME / base64 format.

Location: Outlook

- **XP** %USERPROFILE%\Local Settings\Application Data\Microsoft\Outlook
- **Win7** %USERPROFILE%\AppData\Local\Microsoft\Outlook

Interpretation:

MS Outlook data files found in these locations include OST and PST files. One should also check the OLK and Content.Outlook folder which might roam depending on the specific version of Outlook used. For more information on where to find the OLK folder this link has a handy chart: http://www.hancockcomputertech.com/blog/2010/01/06/find-the-microsoft-outlook-temporary-olk-folder

Skype History

Description:

- Skype history keeps a log of chat sessions and files transferred from one machine to another
- This is turned on by default in Skype installations

Location:

- **XP** C:\Documents and Settings\<username>\Application\Skype\<skype-name>
- **Win7** C:\Users\<username>\AppData\Roaming\Skype\<skype-name>

Interpretation:

Each entry will have a date/time value and a Skype username associated with the action.

Index.dat/ Places.sqlite

Description:

Not directly related to "File Download". Details stored for each local user account. Records number of times visited (frequency).

Location: Internet Explorer

- **XP** %userprofile%\Local Settings\History\ History.IE5
- **Win7** %userprofile%\AppData\Local\Microsoft\Windows\History\History.IE5
- **Win7** %userprofile%\AppData\Local\Microsoft\Windows\History\Low\History.IE5

Location: Firefox

- **IE** %userprofile%\Application Data\Mozilla\ Firefox\Profiles\<random text>.default\places.sqlite
- **Win7** %userprofile%\AppData\Roaming\Mozilla\ Firefox\Profiles\<random text>.default\places.sqlite

Interpretation:

Many sites in history will list the files that were opened from remote sites and downloaded to the local system. History will record the access to the file on the website that was access via a link.

Downloads.sqlite

Description:

Firefox has a built-in download manager application which keeps a history of every file downloaded by the user. This browser artifact can provide excellent information about what sites a user has been visiting and what kinds of files they have been downloading from them.

Location: Firefox

- **IE** %userprofile%\Application Data\Mozilla\ Firefox\Profiles\<random text>.default\downloads.sqlite
- **Win7** %userprofile%\AppData\Roaming\Mozilla\ Firefox\Profiles\<random text>.default\downloads.sqlite

Interpretation:

Downloads.sqlite will include:

- Filename, Size, and Type
- Download from and Referring Page
- File Save Location
- Application Used to Open File
- Download Start and End Times

Program Execution

UserAssist

Description:

GUI-based programs launched from the desktop are tracked in the launcher on a Windows System.

Location: NTUSER.DAT HIVE

NTUSER.DAT\Software\Microsoft\Windows\Currentversion\Explorer\UserAssist\{GUID}\Count

Interpretation:

All values are ROT-13 Encoded

- GUID for XP
  - **75048700** Active Desktop
- GUID for Win7
  - **CEBFF5CD** Executable File Execution
  - **F4E57C4B** Shortcut File Execution
- Program Locations for Win7 Userassist
  - **ProgramFilesX64** 6D809377-...
  - **ProgramFilesX86** 7C5A40EF-...
  - **System** 1AC14E77-...
  - **SystemX86** D65231B0-...
  - **Desktop** B4BFCC3A-...
  - **Documents** FDD39AD0-...
  - **Downloads** 374DE290-...
  - **UserProfiles** 0762D272-...

Last Visited MRU

Description:

Tracks the specific executable used by an application to open the files documented in the OpenSaveMRU key. In addition, each value also tracks the directory location for the last file that was accessed by that application.

Example: Notepad.exe was last run using the C:\Users\<Username>\Desktop folder

Location:

- **XP** NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU
- **Win7** NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU

Interpretation:

Tracks the application executables used to open files in OpenSaveMRU and the last file path used.

RunMRU Start->Run

Description:

Whenever someone does a Start -> Run command, it will log the entry for the command they executed.

Location: NTUSER.DAT HIVE

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU

Interpretation:

The order in which the commands are executed is listed in the RunMRU list value. The letters represent the order in which the commands were executed.

Application Compatibility Cache

Description:

- Windows Application Compatibility Database is used by Windows to identify possible application compatably challenges with executables.
- Tracks the executables file name, file size, last modified time, and in Windows XP the last update time

Location:

- **XP** SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatibility\
- **Win7** SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache

Interpretation:

Any executable run on the Windows system could be found in this key. You can use this key to identify systems that specific malware was executed on. In addition, based on the interpretation of the time based data you might be able to determine the last time of execution or activity on the system.

- Windows XP contains at most 96 entries
  - LastUpdateTime is updated when the files are executed
- Windows 7 contains at most 1024 entries
  - LastUpdateTime does not exist on Win7 systems

Tool to parse:

MANDIANT's ShimCacheParser

Win7 Jump Lists

Description:

- The Windows 7 task bar (Jump List) is engineered to allow users to "jump" or access items they frequently or have recently used quickly and easily. This functionality cannot only be recent media files, but recent tasks as well.
- The data stored in the AutomaticDestinations folder will each have a unique file prepended with the AppID of the associated application.

Location:

- **Win7** C:\Users\<user>\AppData\Roaming\Microsoft\Windows\Recent\ AutomaticDestinations

Interpretation:

- First time of execution of application.
  - Creation Time = First time item added to the AppID file.
- Last time of execution of application w/file open.
  - Modification Time = Last time item added to the AppID file.
- List of Jump List IDs -> http://www.forensicswiki.org/wiki/List_of_Jump_List_IDs

Prefetch

Description:

- Increases performance of a system by pre-loading code pages of commonly used applications. Cache Manager monitors all files and directories referenced for each application or process and maps them into a .pf file. Utilized to know an application was executed on a system.
- Limited to 128 files on XP and Win7
- (exename)-(hash).pf

Location:

- **Win7/XP** C:\Windows\Prefetch

Interpretation:

- Each .pf will include last time of execution, # of times run, and device and file handles used by the program
- Date/Time File by that name & path was first executed
  - Creation Date of .pf file (-10 seconds)
- Date/Time File by that name & path was last executed
  - Embedded last execution time of .pf file
  - Last Modification Date of .pf file (-10 seconds)

Services Events

Description:

- Analyze logs for suspicious services running at boot time
- Review services started or stopped around the time of a suspected compromise

Location:

All Event IDs reference the System Log

- **7034** – Service crashed unexpectedly
- **7035** – Service sent a Start / Stop control
- **7036** – Service started or stopped
- **7040** – Start type changed (Boot | On Request | Disabled)

Interpretation:

- A large amount of malware and worms in the wild utilize Services
- Services started on boot illustrate persistence (desirable in malware)
- Services can crash due to attacks like process injection

File Opening / Creation

Open/Save MRU

Description:

In simplest terms, this key tracks files that have been opened or saved within a Windows shell dialog box. This happens to be a big data set, not only including web browsers like Internet Explorer and Firefox, but also a majority of commonly used applications.

Location:

- **XP** NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU
- **Win7** NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePIDlMRU

Interpretation:

- The "*" key – This subkey tracks the most recent files of any extension input in an OpenSave dialog

Last Visited MRU

Description:

Tracks the specific executable used by an application to open the files documented in the OpenSaveMRU key. In addition, each value also tracks the directory location for the last file that was accessed by that application.

Example: Notepad.exe was last run using the C:\Users\Rob\Desktop folder

Location:

- **XP** NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\ LastVisitedMRU
- **Win7** NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\ LastVisitedPidlMRU

Recent Files

Description:

Registry Key that will track the last files and folders opened and is used to populate data in "Recent" menus of the Start menu.

Location: NTUSER.DAT

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

Interpretation:

- RecentDocs – Overall key will track the overall order of the last 150 files or folders opened. MRU list will keep track of the temporal order in which each file/folder was opened. The last entry and modification time of this key will be time and location of the last file of a specific extension was opened.
- .??? – This subkey stores the last files with a specific extension that were opened. MRU list will keep track of the temporal order in which each file was opened. The last entry and modification time of this key will be time and location of the last file of a specific extension was opened.
- Folder – This subkey stores the last folders that were

Office Recent Files

Description:

MS Office programs will track their own Recent Files list to make it easier for the user to remember the last file they were editing.

Location:

NTUSER.DAT\Software\Microsoft\Office\VERSION

- 14.0 = Office 2010
- 12.0 = Office 2007
- 11.0 = Office 2003
- 10.0 = Office XP

Interpretation:

Similar to the Recent Files, this will track the last files that were opened by each MS Office

Shell bags

Description:

- Can track user window viewing preferences to Windows Explorer
- Can be utilized to tell if activity occurred in a folder
- In some cases, you can see the files from a specific folder as well

Location:

- **XP** NTUSER.DAT\Software\Microsoft\Windows\Shell\Bags
- **XP** NTUSER.DAT\Software\Microsoft\Windows\Shell\BagMRU
- **XP** NTUSER.DAT\Software\Microsoft\Windows\ShellNoRoam\Bags
- **XP** NTUSER.DAT\Software\Microsoft\Windows\ShellNoRoam\BagMRU
- **Win7** USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\Shell\Bags
- **Win7** USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\Shell\BagMRU
- **Win7** NTUSER.DAT\Software\Microsoft\Windows\Shell\BagMRU

Shortcut (LNK) Files

Description:

- Shortcut Files automatically created by Windows
  - Recent Items
  - Opening local and remote data files and documents will generate a shortcut file (.lnk)

Location:

- **XP** C:\Documents and Settings\<username>\Recent\
- **Win7** C:\Users\<user>\AppData\Roaming\Microsoft\Windows\Recent\
- **Win7** C:\Users\<user>\AppData\Roaming\Microsoft\Office\Recent\

Note these are primary locations of LNK files. They can also be found in other locations.

Interpretation:

- Date/Time File of that name was first opened
  - Creation Date of Shortcut (LNK) File
- Date/Time File of that name was last opened
  - Last Modification Date of Shortcut (LNK) File
- LNKTarget File (Internal LNK File Information) Data:

Win7 Jump Lists

Description:

- The Windows 7 task bar (Jump List) is engineered to allow users to "jump" or access items they frequently or have recently used quickly and easily. This functionality cannot only be recent media files, but recent tasks as well.
- The data stored in the AutomaticDestinations folder will each have a unique file prepended with the AppID of the association application and embedded with LNK files in each stream.

Location:

- **Win7** C:\Users\<user>\AppData\Roaming\Microsoft\Windows\Recent\ AutomaticDestinations

Interpretation:

- Using the Structured Storage Viewer open up one of the AutomaticDestination jumplist files.

Prefetch

Description:

- Increases performance of system by pre-loading code pages of commonly used applications. Cache Manger monitors all files and directories referenced for each application or process and maps them into a .pf file. Utilized to know an application was executed on a system.
- Limited to 128 files on XP and Vista/Win7
- (exename)-(hash).pf

Location:

- **Win7/XP** C:\Windows\Prefetch

Interpretation:

- Can examine each .pf file to look for file handles recently

Index.dat file://

Description:

- A little known fact about the IE History is that the information stored in the history files is not just related to Internet browsing. The history also records local and remote (via network shares) file access, giving us an excellent means for determining which files and applications were accessed on the system, day by day.

Location: Internet Explorer

- **XP** %userprofile%\Local Settings\History\History.IE5
- **Win7** %userprofile%\AppData\Local\Microsoft\Windows\History\History.IE5
- **Win7** %userprofile%\AppData\Local\Microsoft\Windows\History\Low\History.IE5

Interpretation:

- Link Ch 12c

Ch 12a-1

Data Runs

- **$DATA attribute lists all clusters with the file's contents**
- **May not be contiguous (fragmented file)**
  - **Lists "data runs" that must be assembled together to get the complete file**

Resident Data

- **MFT entry contains 1024 bytes**
- **That's enough room to store complete data for small files (up to 700 or 800 bytes) in the MFT**
- **These are called "Resident files"**
- **Set the Resident flag in the MFT entry**

MFT Slack Space

- **MFT may contain leftovers from previously resident data**
- **This happens if a file was small enough to be resident and then expanded to be too large to remain resident**

Alternate Data Streams

- **Additional named $DATA attributes in a file's MFT entry**
- **Each can point to an unique set of cluster runs**
- **All the data streams share the same Standard Information and Filename attributes**
  - **So they all share the same timestamps**

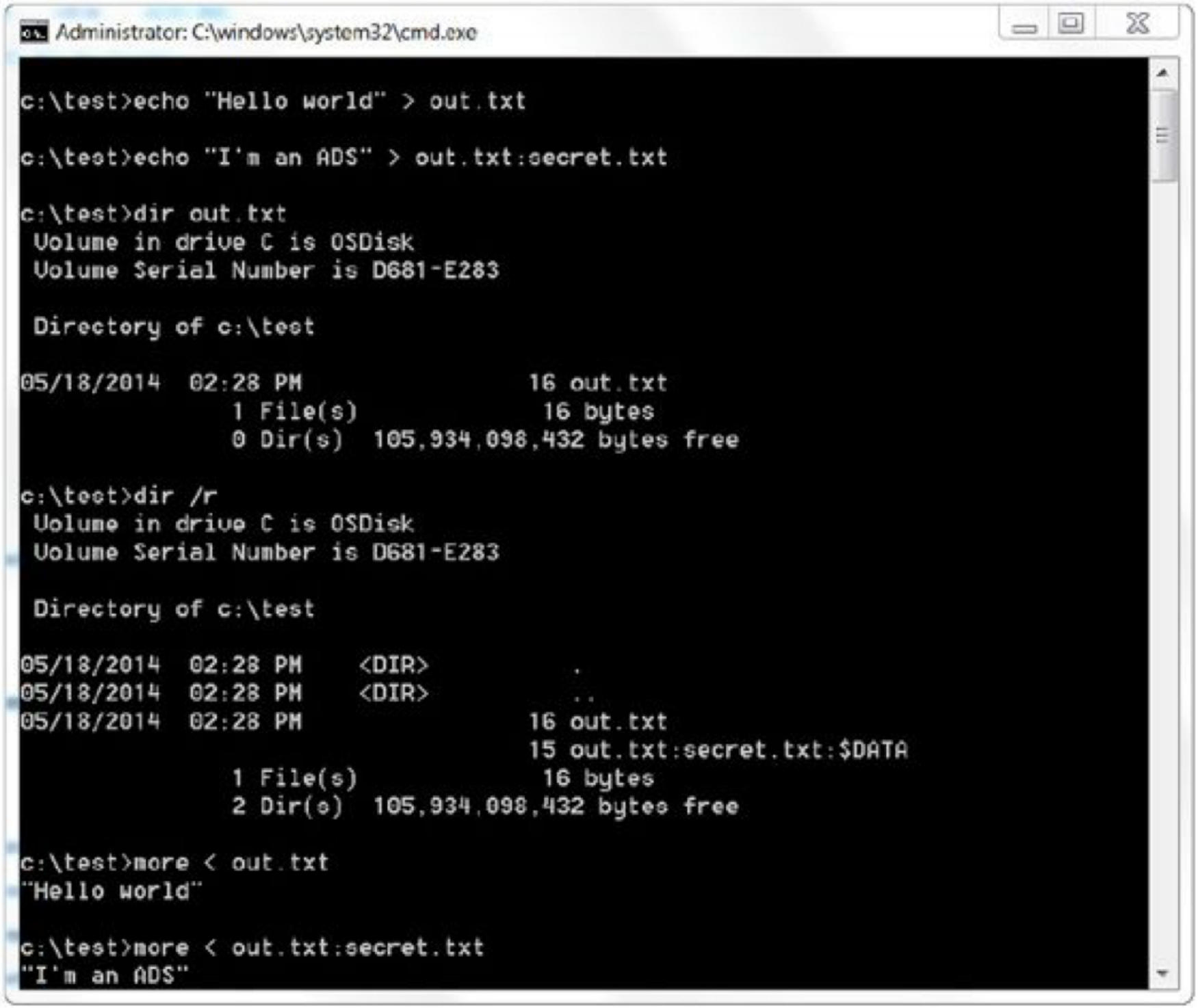


Figure 12-4. Creation and display of an alternate data stream

Known Alternate Stream Names

- **Browsers append a stream to downloaded files**
  - **Named Zone.Identifier**
- **Windows Explorer uses this data to determine the origin of a file and enforce security controls on it**
  - **Link Ch 12c**

```
C:\Users\Administrator\Downloads>dir
 Volume in drive C has no label.
 Volume Serial Number is C6E7-CFDE

 Directory of C:\Users\Administrator\Downloads

10/19/2016  09:30 AM    <DIR>          .
10/19/2016  09:30 AM    <DIR>          ..
09/27/2016  11:56 AM       108,771,096 iTunesSetup.exe
10/19/2016  09:30 AM           143,873 Streams.zip
               2 File(s)    108,914,969 bytes
               2 Dir(s)  13,337,886,720 bytes free

C:\Users\Administrator\Downloads>dir /r
 Volume in drive C has no label.
 Volume Serial Number is C6E7-CFDE

 Directory of C:\Users\Administrator\Downloads

10/19/2016  09:30 AM    <DIR>          .
10/19/2016  09:30 AM    <DIR>          ..
09/27/2016  11:56 AM       108,771,096 iTunesSetup.exe
                                    26 iTunesSetup.exe:Zone.Identifier:$DATA
10/19/2016  09:30 AM           143,873 Streams.zip
                                    26 Streams.zip:Zone.Identifier:$DATA
               2 File(s)    108,914,969 bytes
               2 Dir(s)  13,337,952,256 bytes free
```

```
C:\Users\Administrator\Downloads>streams -d iTunesSetup.exe

streams v1.60 - Reveal NTFS alternate streams.
Copyright (C) 2005-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Users\Administrator\Downloads\iTunesSetup.exe:
   Deleted :Zone.Identifier:$DATA

C:\Users\Administrator\Downloads>dir /r
 Volume in drive C has no label.
 Volume Serial Number is C6E7-CFDE

 Directory of C:\Users\Administrator\Downloads

10/19/2016  09:36 AM    <DIR>          .
10/19/2016  09:36 AM    <DIR>          ..
09/27/2016  11:56 AM       108,771,096 iTunesSetup.exe
10/19/2016  09:30 AM           143,873 Streams.zip
                                    26 Streams.zip:Zone.Identifier:$DATA
               2 File(s)    108,914,969 bytes
               2 Dir(s)  13,337,956,352 bytes free
```

MFT Analysis Tools

- **The Sleuth Kit** www.sleuthkit.org/sleuthkit
  Comprehensive open source toolkit for analyzing disk images and file system metadata.
- **mft2csv** code.google.com/p/mft2csv Suite of tools for converting the MFT to a CSV file and dumping single MFT entries to console for a specified file/path.
- **analyzeMFT** github.com/dkovar/analyzeMFT
  Another MFT parsing utility, capable of converting entries to CSV and Sleuthkit body file formats. If mft2csv fails to convert a given MFT successfully, try using this tool (and vice versa).
- **plaso** plaso.kiddaland.net A powerful timeline analysis engine that can incorporate evidence from Sleuth Kit and numerous other sources of metadata. This tool was designed to supersede the popular log2timeline utility.

INDX Attributes

- **Used to make file searches faster**
- **Often contains metadata from deleted files**
  - **Links Ch 12h, 12i**

- File name
- Parent directory MFT record number
- All four MACE timestamps
- Physical and logical file size

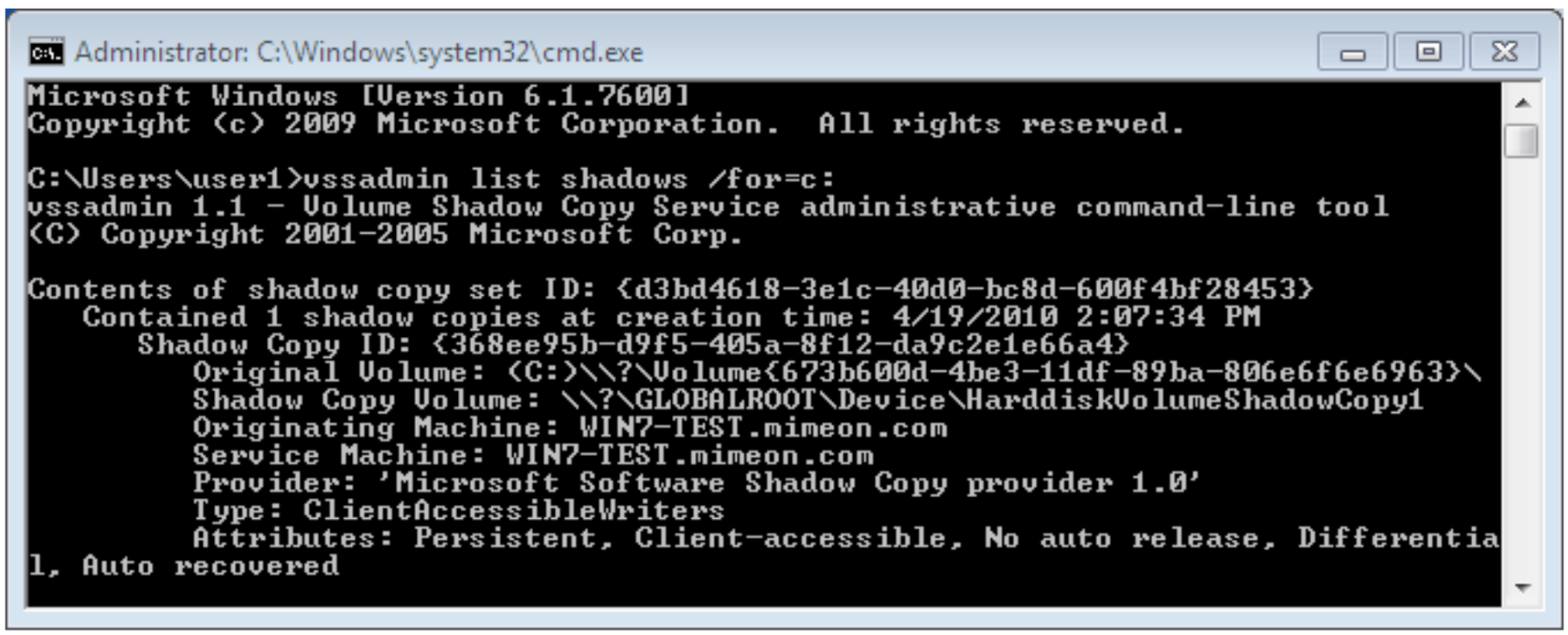
Ch 12a-2

Change Logs

- **$LogFile tracks all transactions that change the structure of a volume**
  - **File or directory creation/copy/delete**
  - **Changes to file metadata or INDX records**
- **$UsnJrnl (Update Sequence Number) journal**
  - **Tracks less data but has a longer history**

Volume Shadow Copies

- **Automatically generated backups of Windows files**
- **Manage with the vssadmin and mklink command-line tools (link Ch 12k)**



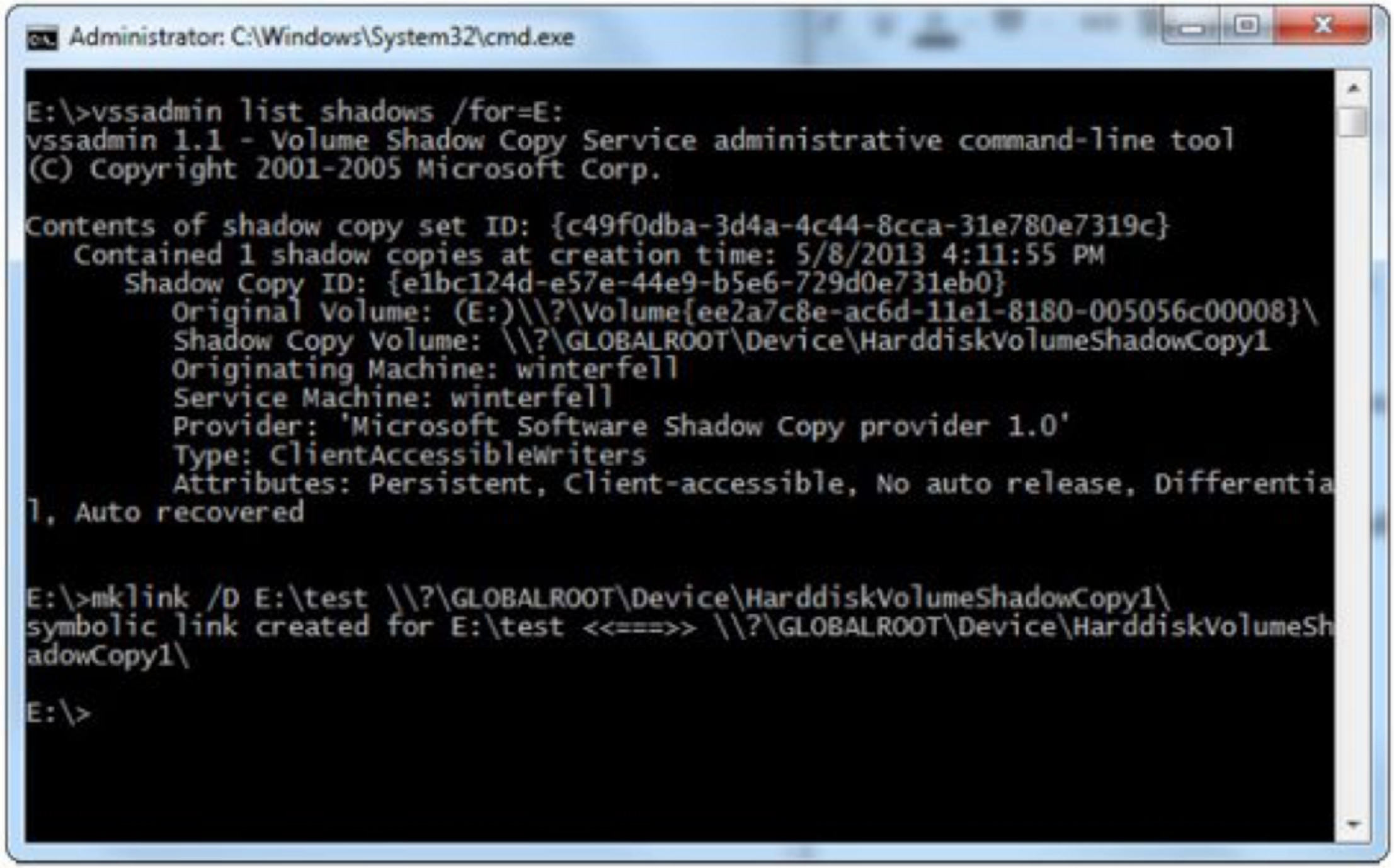


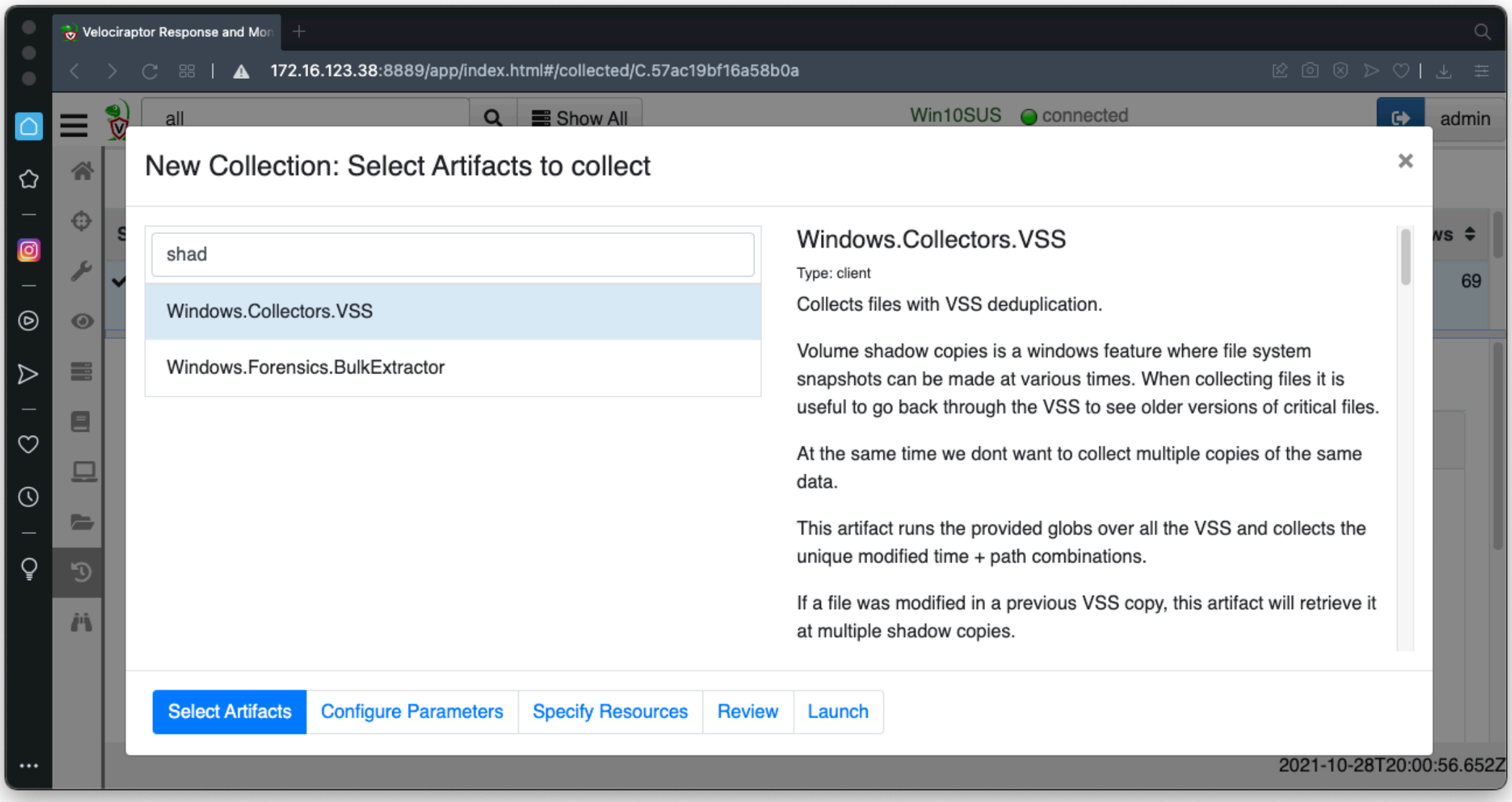
Figure 12-7. Syntax to list shadow copies for a volume and mount via symbolic link

Shadow Copy

- **A mirror of the volume's entire file system at the time of the snapshot**
- **Available within the linked directory**
- **Other tools:**

libvshadow code.google.com/p/libvshadow
Multiplatform library and tools for interacting with volume shadow snapshot data.
Shadow Explorer www.shadowexplorer.com
An easy-to-use user interface for exploring the contents of shadow copy snapshots.
VSC Toolset dfstream.blogspot.com/p/vsc-toolset.html
A user interface through which you can mount shadow copies, browse their contents, and execute batch scripts against them.

Shadow Copies in Velociraptor



File System Redirector

- **Windows 32-bit on Windows 64-bit (WoW64)**
- **Redirects some folders elsewhere when 32-bit programs run on 64-bit Windows, like**
- **%SYSTEMROOT%\system32 redirects to C:\Windows\SysWOW64**
- **32-bit tools may not see the whole file system**

Windows Prefetch

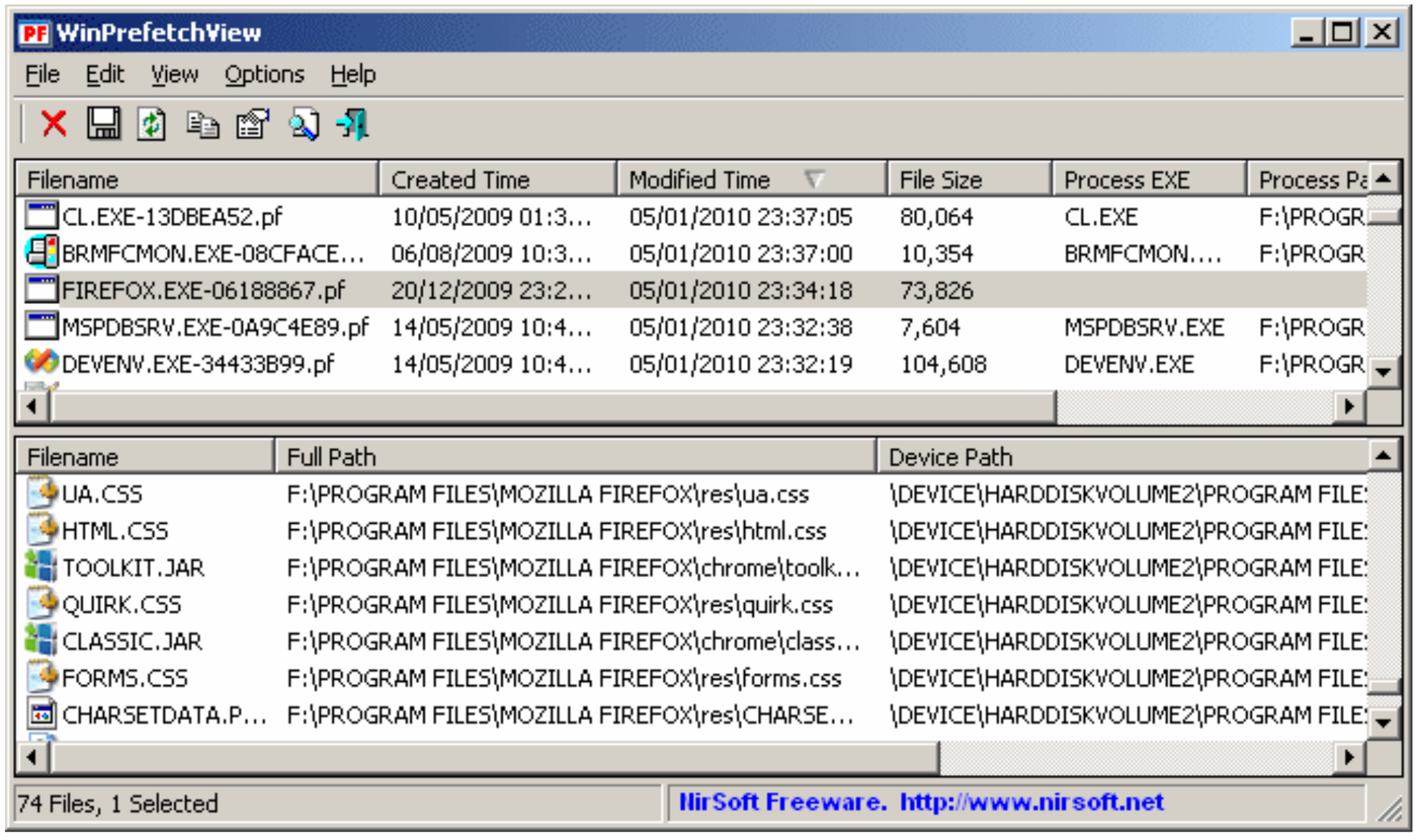
C:\Windows\Prefetch Contains

- **NTOSBOOT-BooDFAAD.pf (system boot prefetch) -- only file existing on Windows Server by default**
- **Layout.ini (for disk defragmenter)**
- **Appname-#########.pf (up to 128 application-specific prefetch files)**

Value

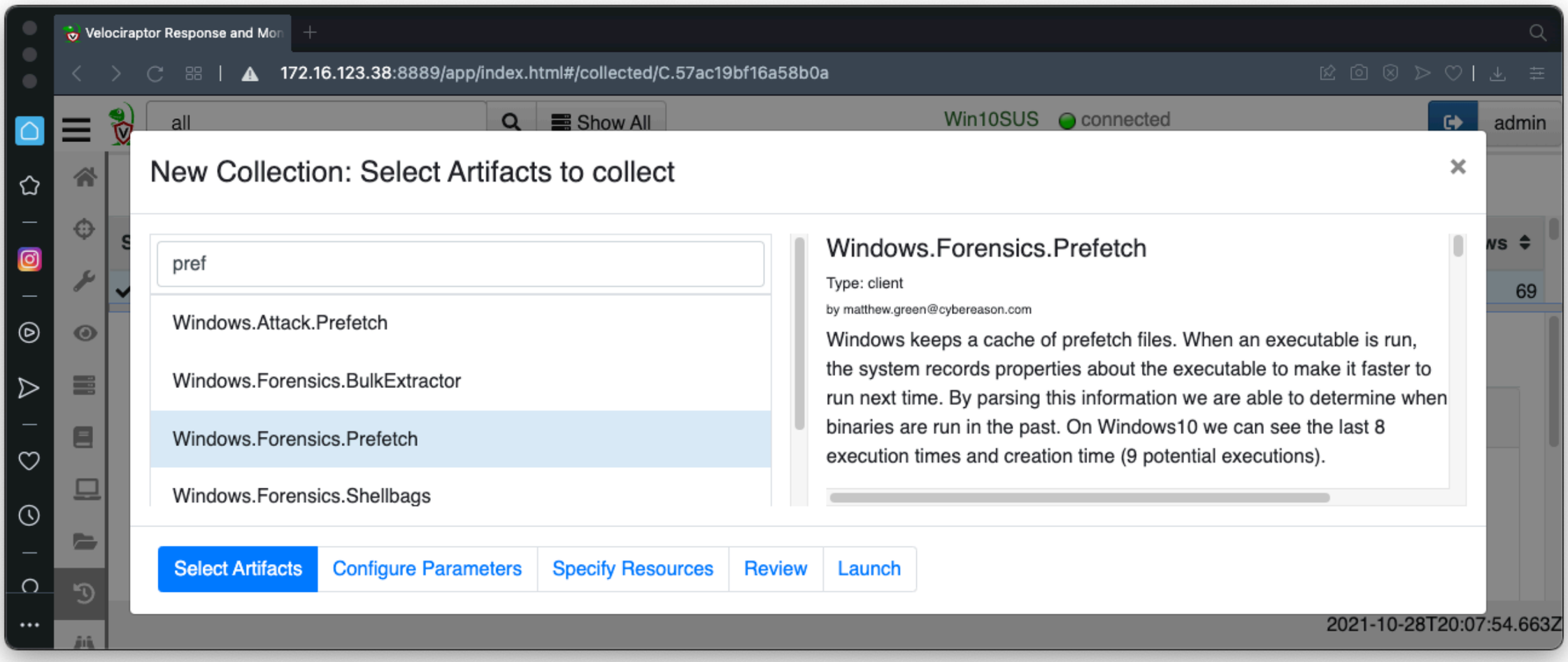
- **A record of programs executed on a system**
- **Even if the executable has been deleted**
- **Shows when application was first run, when it most recently ran, and how many times it was run**
- **Also shows each component loaded**

WinPrefetchView



- **Link Ch 12l**

Prefetch in Velociraptor



Event Logs

Event Logs Enable these Tasks

- Identify successful and failed logon attempts and determine their origin
- Track the creation, start, and stop of system services
- Track usage of specific applications
- Track alterations to the audit policy
- Track changes to user permissions
- Monitor events generated by installed applications (such as antivirus, database, and web server services)

Types of Logs

- **Core event logs in all Windows versions**
  - **Application**
    - **Errors and info from apps; antivirus and host-based IPS logs**
  - **System**
    - **Events from core Windows services; changes in time, driver loads, network configuration issues**
  - **Security**
    - **Login and logoff attempts, changes to audit policy**

Acquiring Logs

- **Log file locations are specified in this Registry key: HKLM\SYSTEM\CurrentControlSet\Services\Eventlog**
- **For Vista and later, the logs are in these XML files:**

  - **Application** %SYSTEMROOT%\System32\Winevt\Logs\Application.evtx
  - **System** %SYSTEMROOT%\System32\Winevt\Logs\System.evtx
  - **Security** %SYSTEMROOT%\System32\Winevt\Logs\Security.evtx

Applications and Services Logs

- **EVTX files in %SYSTEMROOT%\System32\Winevt\Logs**
- **Logs for Task scheduler, Windows Firewall, AppLocker, Terminal Services, User Access Control**

Event ID

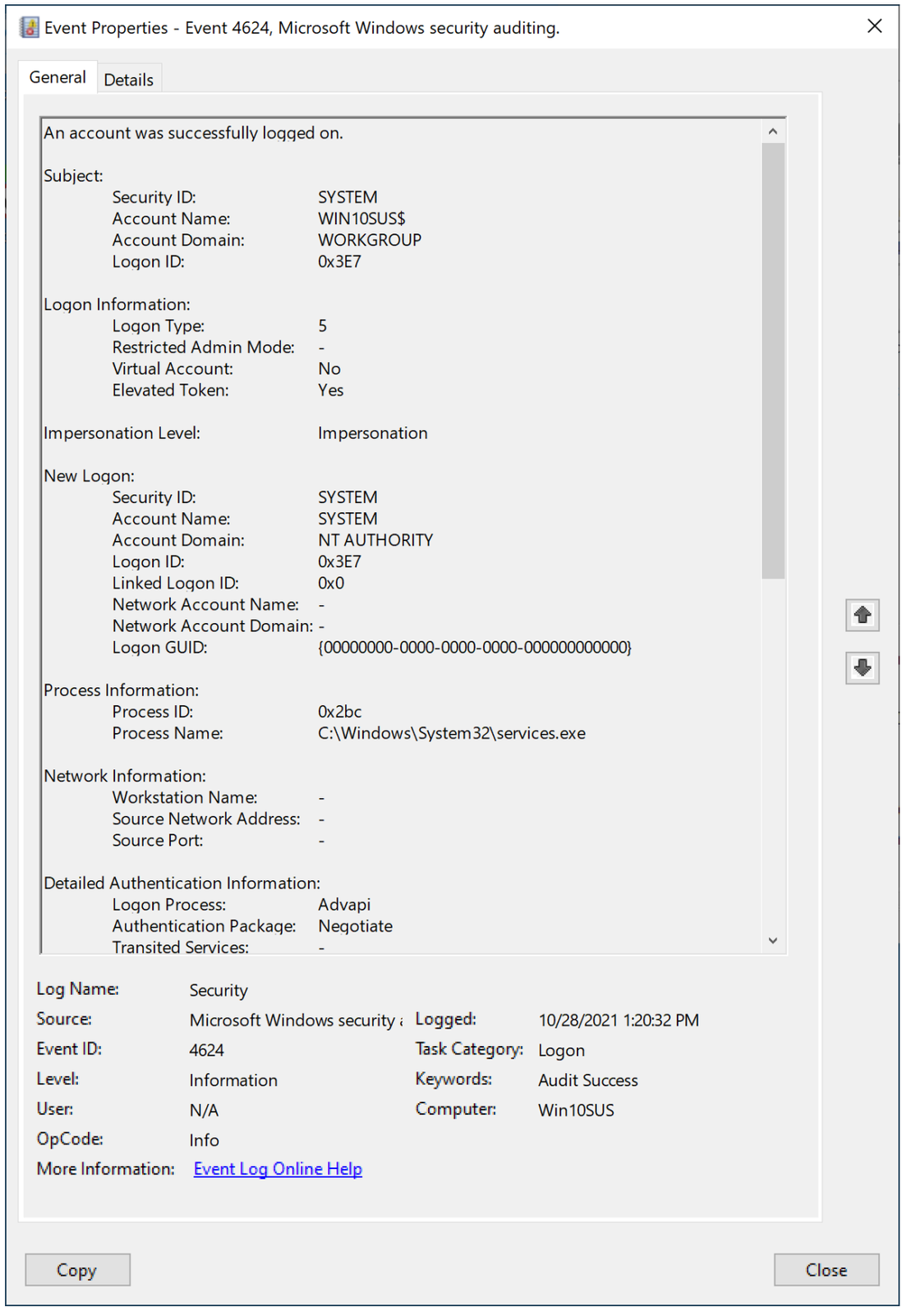
- **Each event is labelled with its Source and Event ID number**
- **Vista and later often have EventIDs that are 4096 larger than the EventID from Windows XP**

Logon Events

```
Event ID: 540

Successful Network Logon:
User Name: Administrator
Domain:  CORPDOMAIN
Logon ID:  (0x0,0x3E2C4E73)
Logon Type: 3
Logon Process: NtLmSsp
Authentication Package: NTLM
Workstation Name: laptop1022
Source Network Address: 10.0.1.13
```

From Event Viewer



- **Logon ID** A unique session identifier. You can use this value as a search term or filter to find all event log entries associated with this specific logon session.
- **Logon Type** A code referencing the type of logon initiated by the user. The following table provides further detail on the Logon Type field and its possible values:

Type	Code	Type	Code
Interactive	2	Unlock	7
Network	3	NetworkCleartext	8
Batch	4	NewCredentials	9
Service	5	RemoteInteractive	10
Proxy	6	CacheInteractive	11

Fields

- **Logon Process** The process that initiated the logon event. Common options include NtLmSsp, Kerberos, User32, and Advapi.
- **Authentication Package** The security and authentication protocol used to process the logon event. Common options include NTLM, Kerberos, and Negotiate.
- **Workstation Name** The source system from which the logon originated. This is not always captured in the event log entry.
- **Source Network Address** The source IP address from which the logon originated. This is not always captured in the event log entry.

Lateral Movement

- **Attackers use stolen credentials to move from system to system**
- **Often use a common administrator account**
- **Or a domain or domain administrator account**

Example

- Our attacker, Bob, has interactive access to a Windows 7 workstation, alpha, through a persistent backdoor.
- Alpha is joined to a corporate domain, ACME.
- The backdoor runs under the context of the domain user who owns alpha, ACME\Eve.
- Through password dumping and other intrusion activities, the attacker has obtained credentials for two accounts:
  - A local administrator, localAdmin, that is configured with an identical password on each workstation in the ACME domain
  - A domain administrator, ACME\domainAdmin, who has full access to all workstations and servers in the environment

In Command Shell as ACME\Eve

1. He mounts the C$ share for workstation beta, from source system alpha, to transfer malware and tools, using the following command:

```
net use \\beta\c$ /u:localAdmin "badPassword"
```

2. He uses the SysInternals PSExec utility to remotely execute a command on workstation gamma, once again from source system alpha, using the following command:

```
psexec.exe  \\gamma -u ACME\domainAdmin -p worsePassword "C:\
path\to\malware.exe"
```

3. He establishes a remote desktop connection to server zeta, once again from source system alpha, using the Windows built-in RDP client (username ACME\domainAdmin, password worsePassword).
4. He browses to an IIS intranet web server, delta, that requires NTLM authentication. Bob uses ACME\domainAdmin credentials.

Events Logged

- Action 1 will generate a logon type 3 (network) recorded on beta because a local account was used.
- Action 2 will generate a logon type 3 recorded on beta, as well as on the ACME domain controller, because a domain account was used. In addition, a “Logon attempt using explicit credentials” event (EID 4648) will be recorded on alpha and reference both the attacker’s use of the credentials ACME\domainAdmin and the target system beta. This event is generated due to the use of PsExec under a different set of domain credentials than the attacker’s current session (ACME\Eve).
- Action 3 will generate a logon type 10 (RemoteInteractive) recorded on zeta as well as on the ACME domain controller.
- Action 4 will generate a logon type 3 (due to using IIS authentication) recorded on delta as well as on the ACME domain controller.

Changes to Accounts and Security Settings: Security Logs

- Account management events indicate whether a user account has been created, deleted, enabled, or disabled, as well as similar changes to account groups.
- Policy change events capture changes to system security settings, including the audit policies that specify what is recorded in event logs.
- An event noting “The audit log was cleared” is recorded whenever a user clears the event logs, irrespective of audit settings. This message includes the username responsible for the change.

Process Auditing

- **Not on by default**
- **Turn it on in local audit policy or Group Policy**
- **Puts an event in the Security log every time a process is executed or terminated**
- **Generates a lot of log events**

Service Events

- **System logs record every time a service starts or stops**
- **A common persistence mechanism for malware**

Logs for PsExec

Date	Event ID	Event Description	User
10/20/2013 21:12:59	7035	The PsExec service was successfully sent a start control.	CORPDOMAIN\Jane
10/20/2013 21:12:59	7036	The PsExec service entered the running state.	N/A
10/20/2013 21:19:53	7035	The PsExec service was successfully sent a stop control.	CORPDOMAIN\Jane
10/20/2013 21:19:53	7036	The PsExec service entered the stopped state.	N/A

Suspicious Things

- **Abnormal usernames using PsExec**
- **Known-bad service names**
- **Errors from malicious binaries that were deleted, but still referenced by a service**

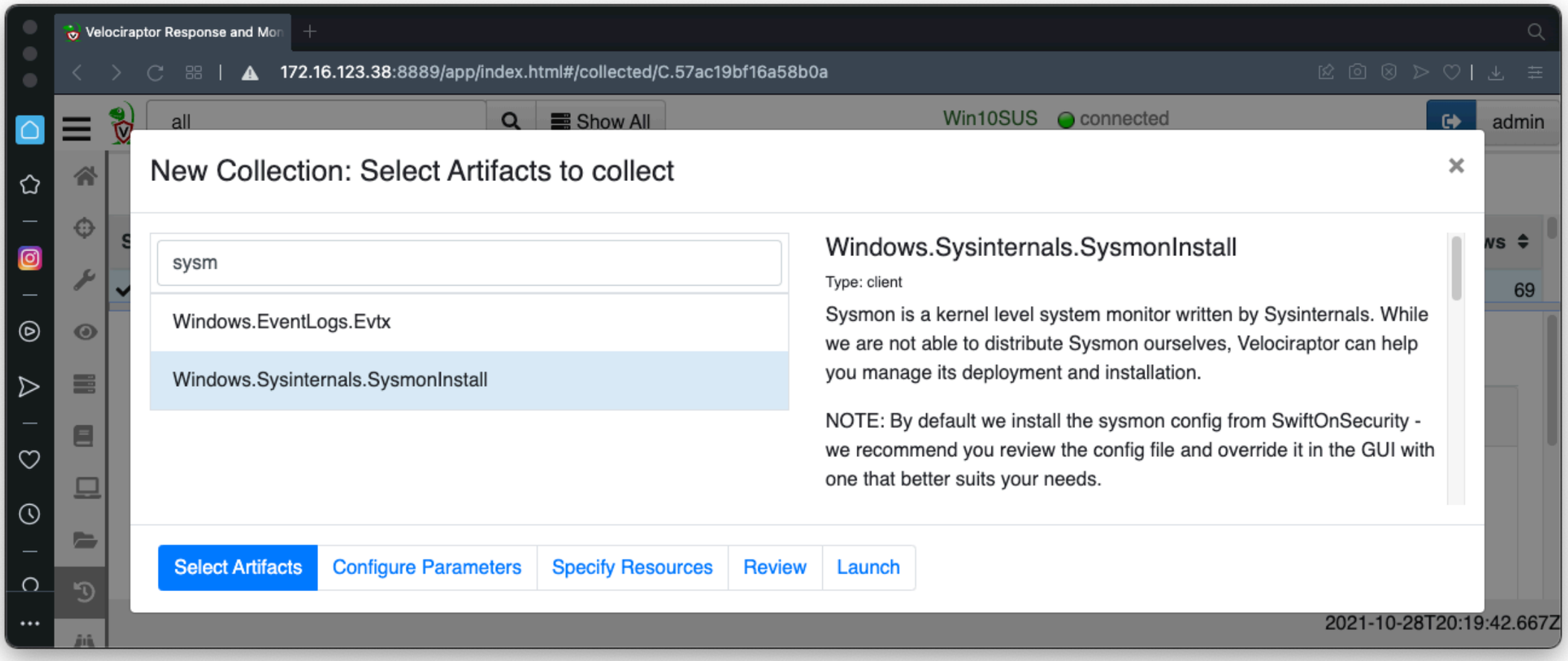
Log Analysis Tips

- **Check Application log for AV alert during period of interest**
- **Increase log file sizes to retain a longer history**
- **If log files in the old binary format are corrupt, use FixEVT (link Ch 12m)**

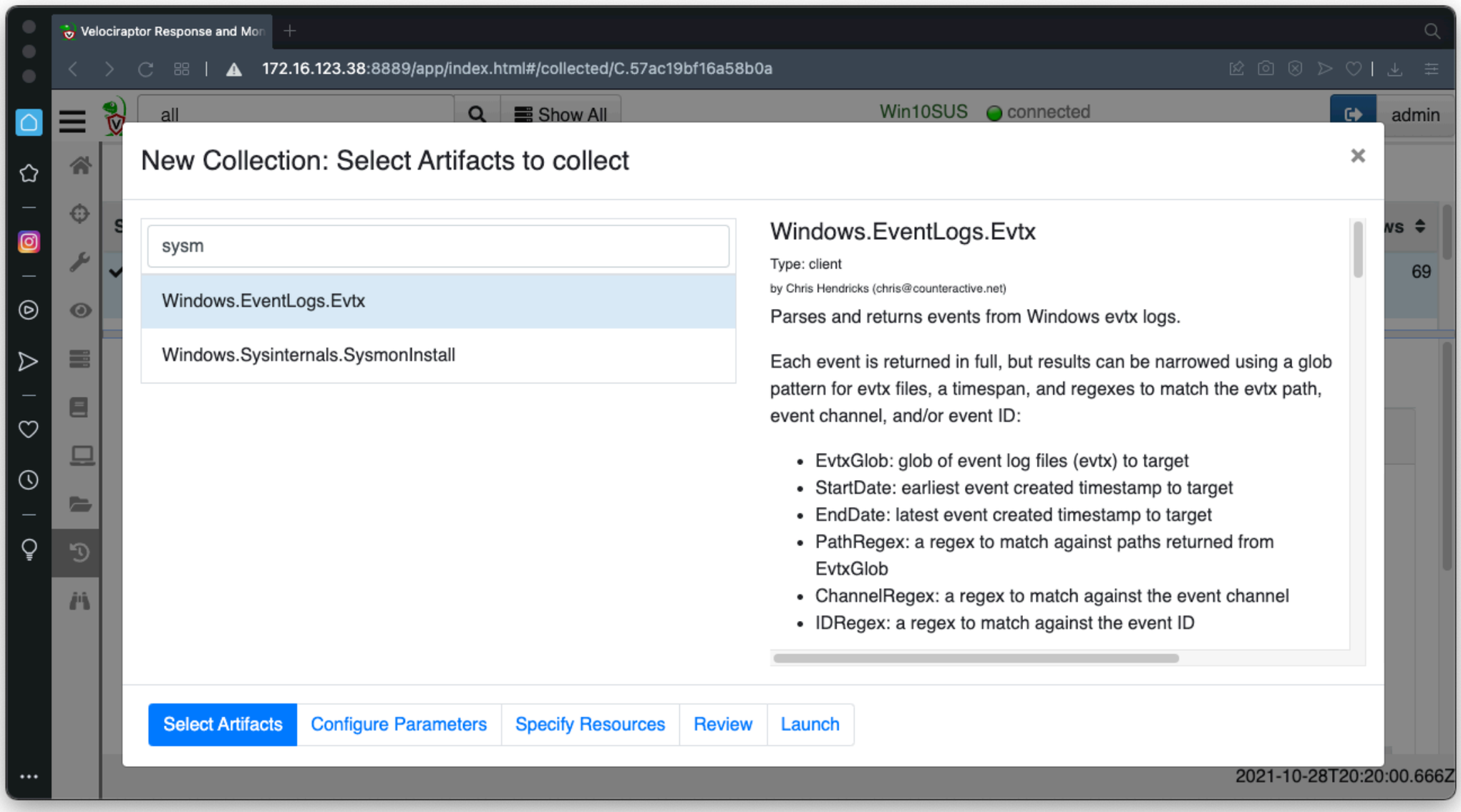
Tools

Tool Name	Capabilities	Free/ Paid	URL
Event Viewer	Allows you to open acquired event log files as well as search/ sort/filter via keyword or XPath.	Free	Built in to Windows
PSLogList	Dumps event logs to plain-text delimited files from a local or remote running system.	Free	technet.microsoft.com/ en-us/sysinternals/ bb897544.aspx
Log Parser	Allows you to issue SQL queries against local event logs.	Free	www.microsoft.com/ en-us/download/ details.aspx?id=24659
Event Log Explorer	Allows you to load, consolidate, filter, and search event logs. High performance on large log files.	Paid	www.eventlogxp.com
LfLe	Recovers Windows Event entries heuristically from a disk image.	Free	github.com/ williballenthin/LfLe
Python-Evtx	Python parser for EVTX format event logs.	Free	www.williballenthin .com/evtx/index.html
Plaso	Evidence-parsing engine designed to facilitate timeline development—supports EVT and EVTX files.	Free	code.google.com/p/ plaso

Sysmon in Velociraptor



Event Logs in Velociraptor



Scheduled Tasks

The "at" Command

- **Requires administrator privileges**
- **Uses local time**
- **Run as SYSTEM**

- `at 16:25 "C:\WINDOWS\evil.exe"`

  Run “evil.exe” once at the next time the clock is 16:25.
- `at 10:25 "C:\temp\beacon.exe" /every:m,t,w`

  Run “beacon.exe” at 10:25 on Monday, Tuesday, and Wednesday on a recurring basis.
- `at \\alpha 08:00 "C:\RECYCLER\passdump.bat"`

  Run “C:\RECYCLER\passdump.bat” on “alpha” the next time its local system time is 08:00.

The "schtasks" Command

- **More complex format**
- **Rarely used by attackers**

.job Files

- **Configuration data for scheduled tasks**
- **One file per task**
- **In %SYSTEMROOT%\Tasks**
- **Files persist until shutdown or reboot of system**

Task Scheduler Logs

- **%SYSTEMROOT%\Tasks\SchedLgU.txt**
- **Records start time and completion of tasks**
- **Also Event Logs, including**
  - **Microsoft-Windows-TaskScheduler%4Operational.evtx**
  - **Security log**

Analyzing .job Files

- **A binary file**
- **Strings will show user information and file path**

Job File Parser

- **Link Ch 12n**

```
$ python jobparser.py -d Tasks/
**************************************************
File: Tasks/At1.job
Product Info: Windows Vista
File Version: 1
UUID: {CE14B659-4115-4263-BFAD-A8318428AB68}
Maximum Run Time: 72:00:00.0 (HH:MM:SS.MS)
Exit Code: 0
Status: Properties not set
Flags: TASK_FLAG_DONT_START_IF_ON_BATTERIES
Date Run: Task not yet run
Running Instances: 0
Application: notepad.exe
Working Directory: Working Directory not set
User: SYSTEM
Comment: Created by NetScheduleJobAdd.
Scheduled Date: Jul 17 02:20:00.0 2012
```

Scheduled Tasks Log

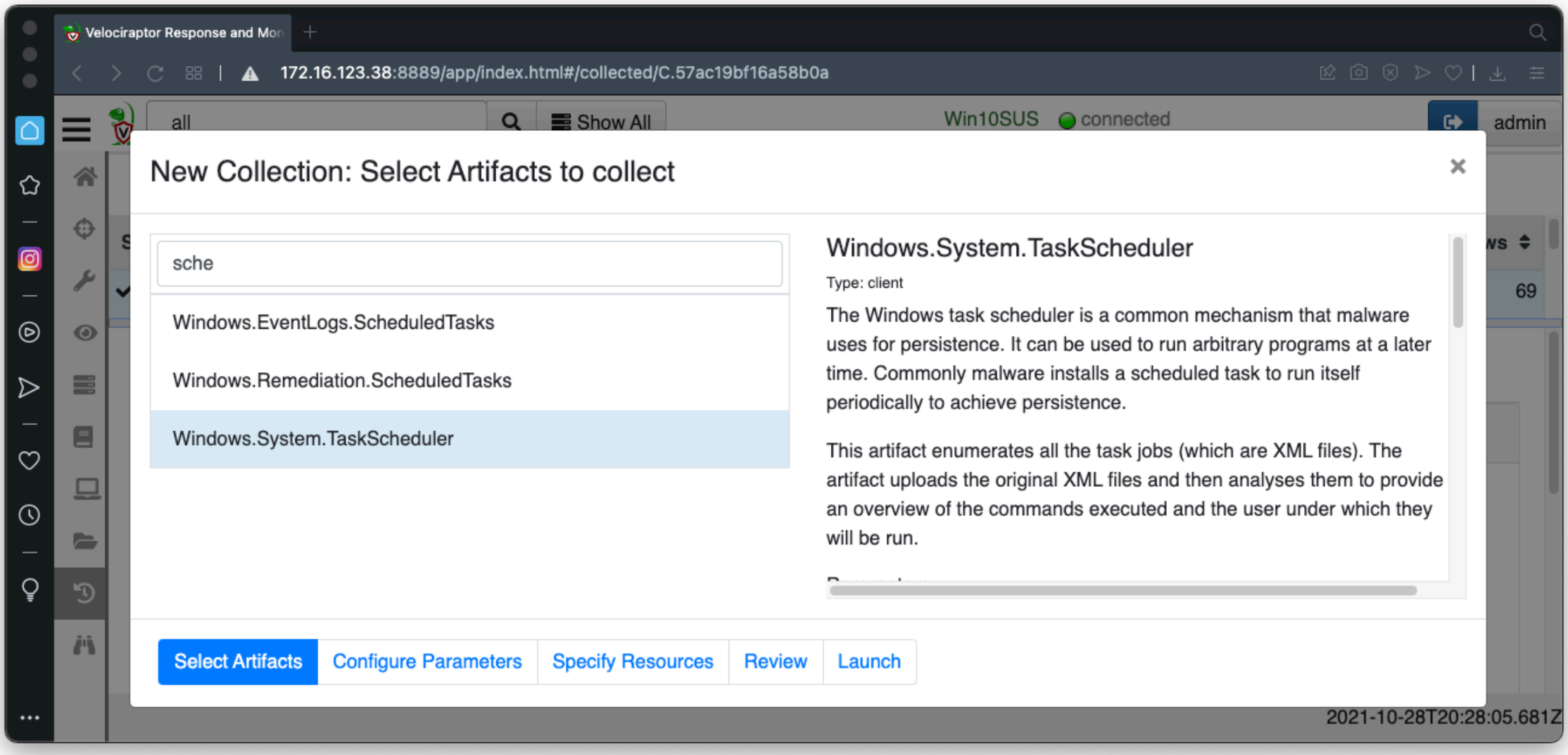
```
"Task Scheduler Service"
    Started at 9/16/2009 4:01:46 PM
"Task Scheduler Service"
5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
"At2.job" (a.bat)
    Started 9/25/2009 2:26:00 AM
"At2.job" (a.bat)
    Finished 9/25/2009 2:34:13 AM
    Result: The task completed with an exit code of (0).
"Task Scheduler Service"
    Started at 9/26/2009 11:12:10 AM
"SCOM 2007 Agent Resume Task.job" (sc.exe)
    Started 9/14/2010 2:55:00 PM
"SCOM 2007 Agent Resume Task.job" (sc.exe)
    Finished 9/14/2010 2:55:00 PM
    Result: The task completed with an exit code of (0).
```

Figure 12-13. Excerpt of a sample Scheduled Task log, SchedLgU.txt

Windows Task Scheduler Operational Log in Event Viewer

#	Date and Time UTC	Event Message	Event ID
1	03/01/2012 10:03:40	User "CORPDOMAIN\superuser" registered Task Scheduler task "\At1"	106
2	03/01/2012 10:03:40	User "CORPDOMAIN\superuser" updated Task Scheduler task "\At1"	140
3	03/01/2012 10:05:00	Task Scheduler launched "{3843A931-B021-98DC-2F3F-940C4EB09011}" instance of task "\At1" due to a time trigger condition	107
4	03/01/2012 10:05:00	Task Engine "S-1-5-18:NT AUTHORITY\System:Service:" received a message from Task Scheduler service requesting to launch task "\At1"	319
5	03/01/2012 10:05:00	Task Scheduler started "{3843A931-B021-98DC-2F3F-940C4EB09011}" instance of the "\At1" task for user "CORPLOCAL\DCSERVER2008$"	100
6	03/01/2012 10:05:00	Task Scheduler launched action "c:\windows\system32\drop.bat" in instance "{3843A931-B021-98DC-2F3F-940C4EB09011}" of task "\At1"	200
7	03/01/2012 10:05:00	Task Scheduler launch task "\At1", instance "C:\Windows\SYSTEM32\cmd.exe" with process ID 8192.	129
8	03/01/2012 10:05:00	Task Scheduler successfully completed task "\At1", instance "C:\Windows\SYSTEM32\cmd.exe" , action "{3843A931-B021-98DC-2F3F-940C4EB09011}"	201
9	03/01/2012 10:05:00	Task Scheduler successfully finished "{3843A931-B021-98DC-2F3F-940C4EB09011}" instance of the "\At1" task for user "CORPLOCAL\DCSERVER2008$"	102

Scheduled Tasks in Velociraptor



Ch 12a-3