Comparing Forensic Tools on an Image File

By Jake Edom, Ahmed Alzahrani, Pranav Sarma

The table of contents

Summary

There exists a large range of forensic tools available ranging from free open source products including PECmd from Erik Zimmerman, to paid for programs developed by companies such as FTK from Exterro. Knowing what's good from the large array of possibilities is difficult without knowing how they all compare to one another. With that in mind we wanted to do exactly that. Starting with the NIST hacking case image. Investigators look through an image of Greg Schardt's computer who had been accused of war-driving and stealing user information under the alias of Mr. Evil. War-driving is the act of physically searching for wireless networks with vulnerabilities by driving around and then submitting this information to a third party.[1] This then led us to investigate what programs they had run to find proof of war-driving programs. What information they had stolen from users while doing this. Proof that Greg Schardt also goes by the name of Mr. Evil. In addition to presenting any other evidence that comes up. All while using various tools that acquire evidence of these activities then comparing how these tools work in comparison to each other. These can be in terms of how easy it is to use, how good the information it gives is and how easy to read it is. These will all be looked into further in the various sections of the report.

Problem

To investigate an incident, the investigator has a variety of tools available to aid in finding the missing puzzle. As mentioned, the tools can be available as open-source tools or paid tools. Some licensed software are expensive for independent investigators, or students who are learning the digital forensics concepts. To make the comparison between the tools, we need to have an image for analysis; therefore, we selected the NIST hacking case image to demonstrate the comparison between the selected tools in every area of analysis with the available licensed tools from the RIT virtual environment. In some areas, we only compared open-source tools with an attempt to compare it with licensed tools.

Primarily, we focused on Web Browsing/Network, prefetch files, registry files, and timeline analysis using the NIST hacking case. For the completion of the project, we analyzed the obtained artifacts to support evidence against Mr. Evil.

With the investigation of Web Browsing/Network, Autopsy and Index.dat Analyzer both free tools were used. These helped to obtain the browsing history evidence specifically and other options were investigated to find more evidence on the network but did not yield good results, this is the “netsh wlan show profile” command and where it gets its information.

Then for the prefetch files, three tools were used: PECmd. prefetch-parser which are open source terminal tools and Prefetch Parser a free GUI based tool. All of which were compared based on information presented, how intuitive they were and other metrics.

For registry files, we selected Forensics Registry EDitor (Fred), and RegRipper, that represent the open-source tools, and Registry Viewer, which is a licensed tool from AccessData [8][9][10]. We gleaned some values from the registry of the acquired image using the three mentioned tools, and compared them to demonstrate the difference between them [11]. Although our main goal was to compare the tools, we attempted to find evidence from the suspect machine to support the accusation against Greg Schardt. We found a hiberfil.sys file, we could dump it to a disk image, and we analyzed its content [12].

For timeline analysis, we have used two tools- log2timeline/plaso and FLS Sleuthkit. These tools are used for investigation purposes to collect timestamps and computer artifacts from a computer system. It gives clear information through the specific year, month, date and time. Those results are then analyzed to find any digital evidence which can corroborate if allegations pointed on the criminal in the investigated case was true or not and can hence serve as a very effective tool for examining people. We attempted to get timeline results using the two tools and compared which one is better than the other

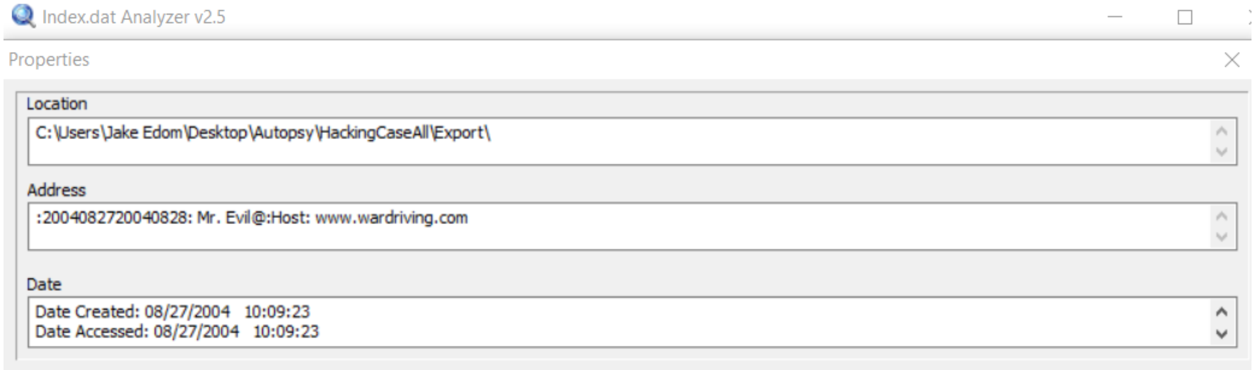
Procedure and demonstration

In this section, we present demonstrations for comparing the tools and analyzing the evidence along the way. This section is divided into various sections to show the related tools and comparison.

Web Browsing/Network

The main goal of this section was to find evidence on the file of internet connections that Greg Schardt had done to prove that they had made connections to other networks. Since this would be important to prove that they were doing war-driving, as with this information one could figure out where they were trying to connect to and this could lead to the vulnerabile networks they were looking for. Starting off with Autopsy, a free digital forensics tool to look for possible files and evidence to look for this.[2] Which lead to the sections based on web cookies and web history to try and find information. Web Cookies lead to evidence of where the user had been on websites and so did Web History. With the main difference being that Web History needed an additional tool of Index.dat Analyzer.[4] Which gave information in the index.dat file which mainly had to do with the URL address and when the file was created. What would be better evidence would be something like the “netsh wlan show profile” which has the ability to show the connection profiles of a Windows machine.[3] Which in our investigation would show these war-driving connections. Although I could not find where this information is stored.

Web artifacts



Location

C:\Users\Jake Edom\Desktop\Autopsy\HackingCaseAll\Export\

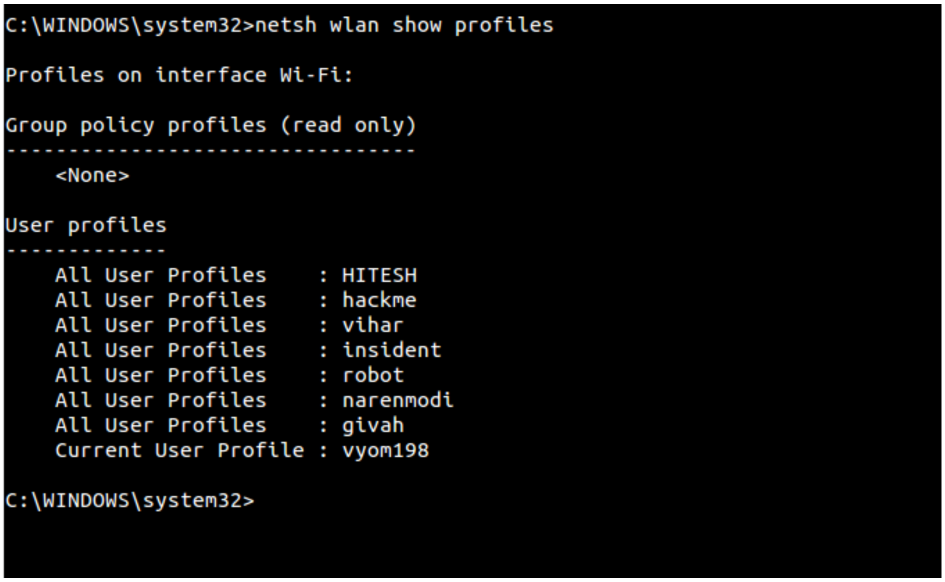
Address

:2004082720040828: Mr. Evil@:Host: www.wardriving.com

Date

Date Created: 08/27/2004   10:09:23
Date Accessed: 08/27/2004   10:09:23

Index.dat Analyzer

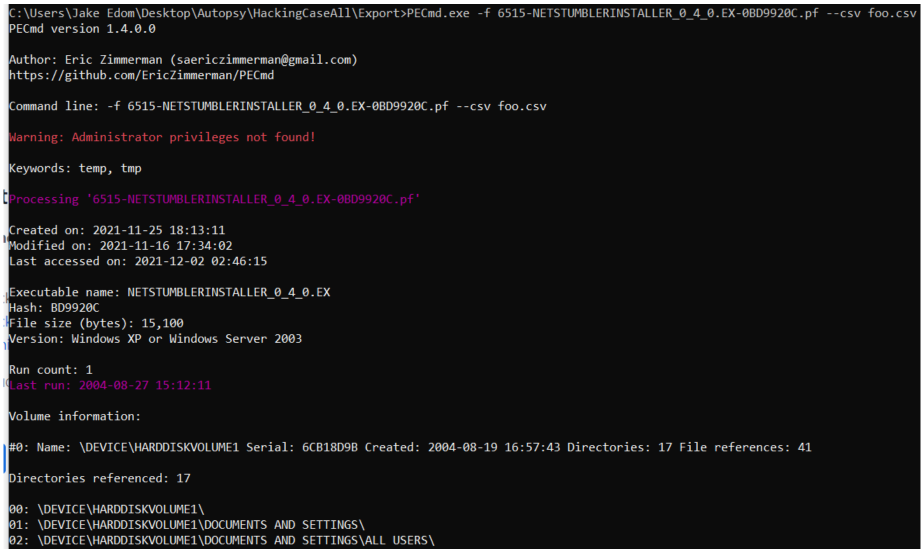


```
C:\WINDOWS\system32>netsh wlan show profiles

Profiles on interface Wi-Fi:

Group policy profiles (read only)
---------------------------------
    <None>

User profiles
-------------
    All User Profiles    : HITESH
    All User Profiles    : hackme
    All User Profiles    : vihar
    All User Profiles    : insident
    All User Profiles    : robot
    All User Profiles    : narenmodi
    All User Profiles    : givah
    Current User Profile : vyom198

C:\WINDOWS\system32>
```

netsh wlan show profiles

Prefetch

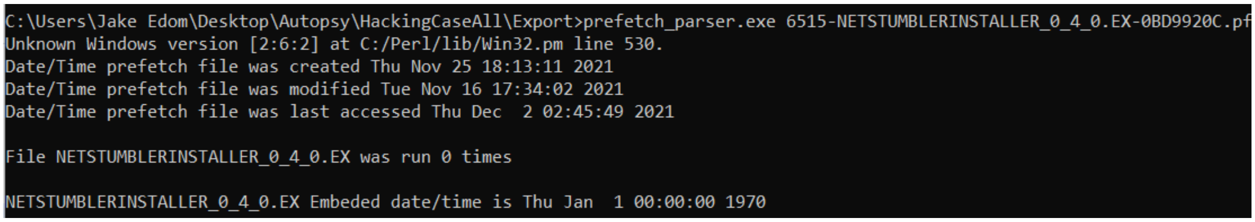
Prefetch files are automatically created files in Windows that normally act to accelerate applications startup processes. This sets up a record of the application having been run and at a specific location. For forensic analysis this means that these files can be used as evidence that a certain file was run on the system. This isn't the only information that a prefetch file can contain. There is information on when the file was last accessed, how many times the file was run, what the explicit exe this prefetch file is connected to and what files and directories it references. With that background information accounted for, we'll talk about the tools used and how they compare. Three different tools were selected, PECmd.exe developed by Erik Zimmerman, Prefetch_Parser developed by Mark Mckinnon on GitHub and prefetch-parser created by Paul Hutelmyer on threatfix.com. Starting with PECmd, this is a terminal based tool that requires some knowledge of the terminal although the README on the download website gives examples on how to run it.[5] As well as some of the other options including how it is able to accept prefetch files including individual files and folders that contain prefetch files. Then the information gathered can then be saved in various ways, including, json and csv.



```
C:\Users\Jake Edom\Desktop\Autopsy\HackingCaseAll\Export>PECmd.exe -f 6515-NETSTUMBLERINSTALLER_0_4_0.EX-0BD9920C.pf --csv foo.csv
PECmd version 1.4.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/PECmd

Command line: -f 6515-NETSTUMBLERINSTALLER_0_4_0.EX-0BD9920C.pf --csv foo.csv

Warning: Administrator privileges not found!

Keywords: temp, tmp

Processing '6515-NETSTUMBLERINSTALLER_0_4_0.EX-0BD9920C.pf'

Created on: 2021-11-25 18:13:11
Modified on: 2021-11-16 17:34:02
Last accessed on: 2021-12-02 02:46:15

Executable name: NETSTUMBLERINSTALLER_0_4_0.EX
Hash: BD9920C
File size (bytes): 15,100
Version: Windows XP or Windows Server 2003

Run count: 1
Last run: 2004-08-27 15:12:11

Volume information:

#0: Name: \DEVICE\HARDDISKVOLUME1 Serial: 6CB18D9B Created: 2004-08-19 16:57:43 Directories: 17 File references: 41

Directories referenced: 17

00: \DEVICE\HARDDISKVOLUME1\
01: \DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\
02: \DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\ALL USERS\
```

PECmd demonstration

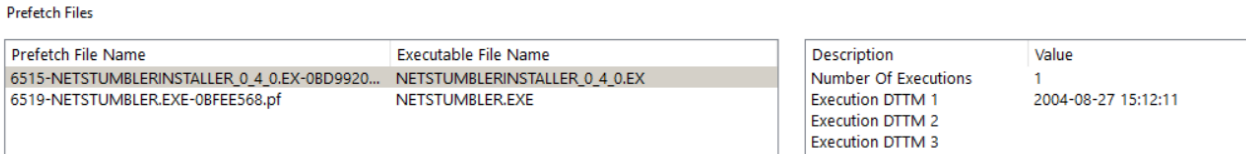
Next will be the other terminal based tool prefetch-parser, unfortunately there isn’t much to say with this one as it is much more simplistic. Only able to work on individual files with no other options available. It does show the critical information for a prefetch file, including MAC times and the number of times the executable has been run. Although the problem is there are some problems with the tool when tested. When using the same prefetch files as PECmd it was unable to find the program had been run before.[6]



```
C:\Users\Jake Edom\Desktop\Autopsy\HackingCaseAll\Export>prefetch_parser.exe 6515-NETSTUMBLERINSTALLER_0_4_0.EX-0BD9920C.pf
Unknown Windows version [2:6:2] at C:/Perl/lib/Win32.pm line 530.
Date/Time prefetch file was created Thu Nov 25 18:13:11 2021
Date/Time prefetch file was modified Tue Nov 16 17:34:02 2021
Date/Time prefetch file was last accessed Thu Dec  2 02:45:49 2021

File NETSTUMBLERINSTALLER_0_4_0.EX was run 0 times

NETSTUMBLERINSTALLER_0_4_0.EX Embeded date/time is Thu Jan  1 00:00:00 1970
```

prefetch-parser failing

The final tool that was investigated was Prefetch Parser which is a GUI based tool, which is more intuitive than the terminal tools, allowing the user to select the prefetch files that exist in a folder and importing them to the tool.[7] From there each file that gets imported can be selected and the information about that particular file is shown. This is what we’ve come to expect, the number of executions, when it was executed with the added bonus of showing the times and dates of the various executions.



Prefetch Files

Prefetch File Name	Executable File Name
6515-NETSTUMBLERINSTALLER_0_4_0.EX-0BD9920...	NETSTUMBLERINSTALLER_0_4_0.EX
6519-NETSTUMBLER.EXE-0BFEE568.pf	NETSTUMBLER.EXE

Description	Value
Number Of Executions	1
Execution DTTM 1	2004-08-27 15:12:11
Execution DTTM 2	
Execution DTTM 3	

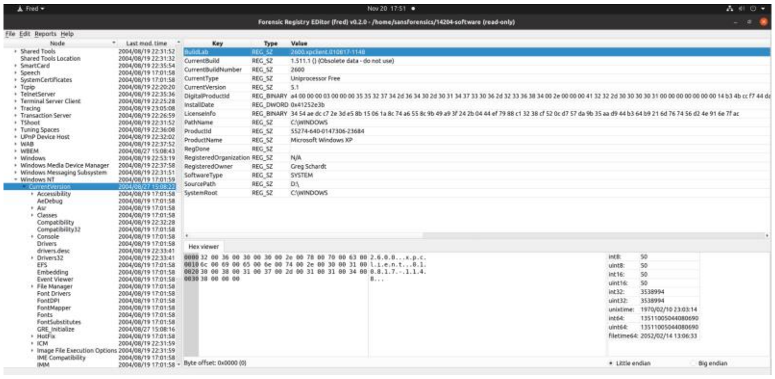
Prefetch Parser

Overall looking at these tools, the best tool is the PECmd for someone who can work with a terminal due to the additional options that make the results more flexible in how the data can be saved. Although with Prefetch Parser is more then adequate for an investigator who may not be as technologically savvy due to the much more intuitive use that a GUI program would have.

Registry Analysis Tools Comparison

The Windows Registries are an essential element of the Windows Operating System since many valuable configurations are stored on them. There are various tools used to glean these configurations from the registry. One of the well-known tools is Registry Viewer from AccessData [*]. Although Registry Viewer is a good tool for analyzing Windows Registry, investigators must pay the license for using the software. There are similar open-source tools that can be used instead of Registry Viewer, such as Forensics Registry EDitor (Fred), RegRipper, and Registry Explorer/RECmd [*][*][*]. In this project, we explored Fred and RegRipper to analyze the registry files from the Hacking Case, and we compared the outputs of the open-source tools with Registry Viewer. Fred is a graphical user interface, and it has a report generation feature. RegRipper is a command-line tool that has plugins for analyzing the registry file.

Starting with Fred, we gleaned the suspected Operating System Version, Installed Data, product ID, and Registered Owner from the “\Microsoft\Windows N\CurrentVersion” from the Software Hive file. We found that the operating system version is Windows XP, and the registered owner goes by the name Greg Schardt.



Forensic Registry EDitor (fred) v0.2.0 - /home/sansforensics/14204-software (read-only)

Node	Last mod. time
Shared Tools	2004/08/19 22:31:52
Shared Tools Location	2004/08/19 22:31:32
SmartCard	2004/08/19 22:35:54
Speech	2004/08/19 17:01:58
SystemCertificates	2004/08/19 17:01:58
Tcpip	2004/08/19 22:20:20
TelnetServer	2004/08/19 22:35:36
Terminal Server Client	2004/08/19 22:25:28
Tracing	2004/08/19 23:05:08
Transaction Server	2004/08/19 22:26:59
TShoot	2004/08/19 22:31:52
Tuning Spaces	2004/08/19 22:36:08
UPnP Device Host	2004/08/19 22:32:02
WAB	2004/08/19 22:37:52
WBEM	2004/08/27 15:08:43
Windows	2004/08/19 22:53:19
Windows Media Device Manager	2004/08/19 22:37:58
Windows Messaging Subsystem	2004/08/19 22:31:51
Windows NT	2004/08/19 17:01:59
CurrentVersion	2004/08/27 15:08:22
Accessibility	2004/08/19 17:01:58
AeDebug	2004/08/19 17:01:58
Asr	2004/08/19 17:01:58
Classes	2004/08/19 17:01:58
Compatibility	2004/08/19 22:32:28
Compatibility32	2004/08/19 17:01:58
Console	2004/08/19 17:01:58
Drivers	2004/08/19 17:01:58
drivers.desc	2004/08/19 22:33:41
Drivers32	2004/08/19 22:33:41
EFS	2004/08/19 17:01:58
Embedding	2004/08/19 17:01:58
Event Viewer	2004/08/19 17:01:58
File Manager	2004/08/19 17:01:58
Font Drivers	2004/08/19 17:01:58
FontDPI	2004/08/19 17:01:58
FontMapper	2004/08/19 17:01:58
Fonts	2004/08/19 17:01:58
FontSubstitutes	2004/08/19 17:01:58
GRE_Initialize	2004/08/27 15:08:16
HotFix	2004/08/19 17:01:58
ICM	2004/08/19 22:31:59
Image File Execution Options	2004/08/19 22:31:59
IME Compatibility	2004/08/19 17:01:58
IMM	2004/08/19 17:01:58

Key	Type	Value
BuildLab	REG_SZ	2600.xpclient.010817-1148
CurrentBuild	REG_SZ	1.511.1 () (Obsolete data - do not use)
CurrentBuildNumber	REG_SZ	2600
CurrentType	REG_SZ	Uniprocessor Free
CurrentVersion	REG_SZ	5.1
DigitalProductId	REG_BINARY	a4 00 00 00 03 00 00 00 35 35 32 37 34 2d 36 34 30 2d 30 31 34 37 33 30 36 2d 32 33 36 38 34 00 2e 00 00 00 41 32 32 2d 30 30 30 30 31 00 00 00 00 00 00 00 14 b3 4b cc f7 44 d
InstallDate	REG_DWORD	0x41252e3b
LicenseInfo	REG_BINARY	34 54 ae dc c7 2e 3d e5 8b 15 06 1a 8c 74 a6 55 8c 9b 49 a9 3f 24 2b 04 44 ef 79 88 c1 32 38 cf 52 0c d7 57 da 9b 35 aa d9 44 b3 64 b9 21 6d 76 74 56 d2 4e 91 6e 7f ac
PathName	REG_SZ	C:\WINDOWS
ProductId	REG_SZ	55274-640-0147306-23684
ProductName	REG_SZ	Microsoft Windows XP
RegDone	REG_SZ	
RegisteredOrganization	REG_SZ	N/A
RegisteredOwner	REG_SZ	Greg Schardt
SoftwareType	REG_SZ	SYSTEM
SourcePath	REG_SZ	D:\
SystemRoot	REG_SZ	C:\WINDOWS

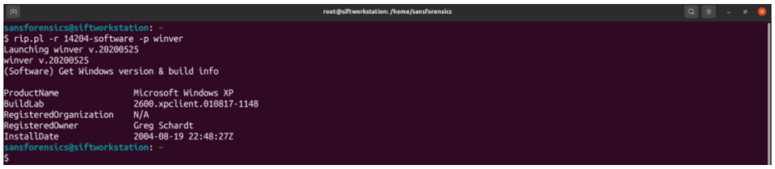
```
0000 32 00 36 00 30 00 30 00 2e 00 78 00 70 00 63 00 2.6.0.0...x.p.c.
0010 6c 00 69 00 65 00 6e 00 74 00 2e 00 30 00 31 00 l.i.e.n.t...0.1.
0020 30 00 38 00 31 00 37 00 2d 00 31 00 31 00 34 00 0.8.1.7.-.1.1.4.
0030 38 00 00 00                                     8...
```

int8:	50
uint8:	50
int16:	50
uint16:	50
int32:	3538994
uint32:	3538994
unixtime:	1970/02/10 23:03:14
int64:	13511005044080690
uint64:	13511005044080690
filetime64:	2052/02/14 13:06:33

Byte offset: 0x0000 (0)

Fred

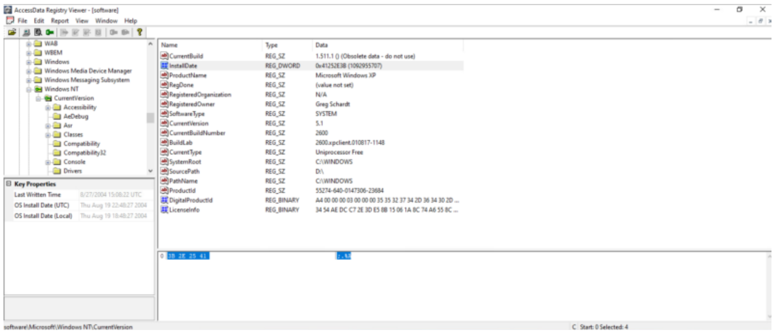
In RegRipper, we ran the tool from the terminal by typing the tool script name “rip.pl”, “-r” for registry hive file parser, the hive path, and name, “-p” followed by plugin name. Note that RegRipper displayed fewer details compared to Fred.



```
sansforensics@siftworkstation: ~
$ rip.pl -r 14204-software -p winver
Launching winver v.20200525
winver v.20200525
(Software) Get Windows version & build info

ProductName               Microsoft Windows XP
BuildLab                  2600.xpclient.010817-1148
RegisteredOrganization    N/A
RegisteredOwner           Greg Schardt
InstallDate               2004-08-19 22:48:27Z
sansforensics@siftworkstation: ~
$
```

RegRipper

By comparing the result of the open-source tools with the Registry Viewer, we almost have the same output. We noted that the Installed Date in the Registry Viewer is displayed in hexadecimal value. We had to highlight the value in the lower pane to interpret it into a readable format.



Name	Type	Data
CurrentBuild	REG_SZ	1.511.1 () (Obsolete data - do not use)
InstallDate	REG_DWORD	0x41252E3B (1092955707)
ProductName	REG_SZ	Microsoft Windows XP
RegDone	REG_SZ	(value not set)
RegisteredOrganization	REG_SZ	N/A
RegisteredOwner	REG_SZ	Greg Schardt
SoftwareType	REG_SZ	SYSTEM
CurrentVersion	REG_SZ	5.1
CurrentBuildNumber	REG_SZ	2600
BuildLab	REG_SZ	2600.xpclient.010817-1148
CurrentType	REG_SZ	Uniprocessor Free
SystemRoot	REG_SZ	C:\WINDOWS
SourcePath	REG_SZ	D:\
PathName	REG_SZ	C:\WINDOWS
ProductId	REG_SZ	55274-640-0147306-23684
DigitalProductId	REG_BINARY	A4 00 00 00 03 00 00 00 35 35 32 37 34 2D 36 34 30 2D ...
LicenseInfo	REG_BINARY	34 54 AE DC C7 2E 3D E5 8B 15 06 1A 8C 74 A6 55 8C ...

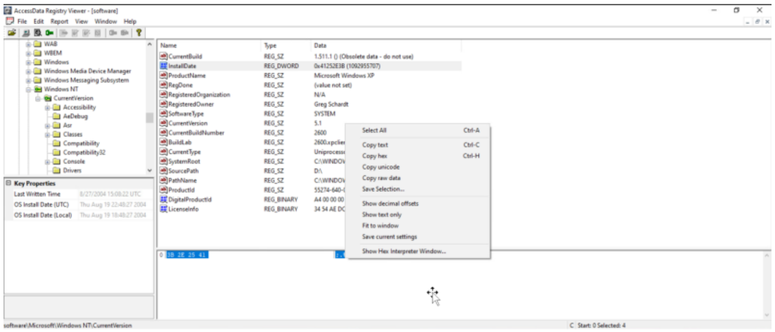
Key Properties	
Last Written Time	8/27/2004 15:08:22 UTC
OS Install Date (UTC)	Thu Aug 19 22:48:27 2004
OS Install Date (Local)	Thu Aug 19 18:48:27 2004

```
0  3B 2E 25 41                    ;.%A
```

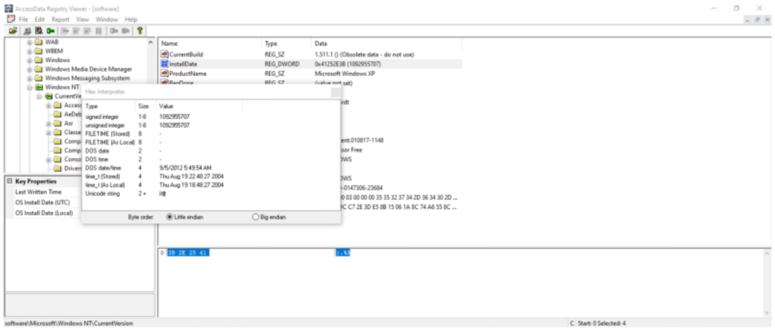
software\Microsoft\Windows NT\CurrentVersion

C Start: 0 Selected: 4

Registry Viewer



Registry Viewer highlights the hex value


AccessData Registry Viewer - [software]
File Edit Report View Window Help
WAB
WBEM
Windows
Windows Media Device Manager
Windows Messaging Subsystem
Windows NT
CurrentVe
Access
AeDeb
Asr
Classe
Comp
Comp
Conso
Driver
Key Properties
Last Written Time
OS Install Date (UTC)
OS Install Date (Local)
Hex Interpreter
Byte order: ● Little endian ○ Big endian

Name	Type	Data
CurrentBuild	REG_SZ	1.511.1 () (Obsolete data - do not use)
InstallDate	REG_DWORD	0x41252E3B (1092955707)
ProductName	REG_SZ	Microsoft Windows XP
RegDone	REG_SZ	(value not set)

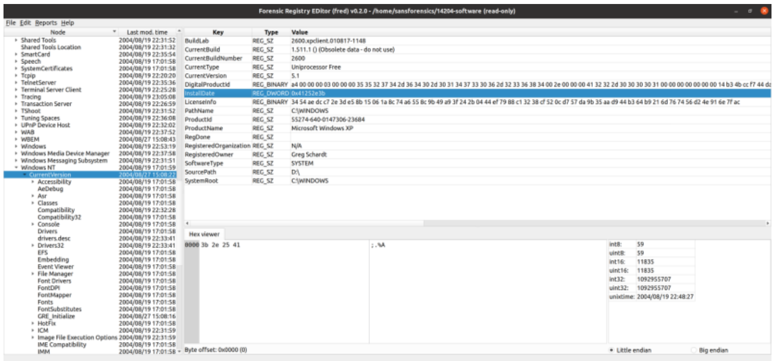
Type	Size	Value
signed integer	1-8	1092955707
unsigned integer	1-8	1092955707
FILETIME (Stored)	8	-
FILETIME (As Local)	8	-
DOS date	2	-
DOS time	2	-
DOS date/time	4	9/5/2012 5:49:54 AM
time_t (Stored)	4	Thu Aug 19 22:48:27 2004
time_t (As Local)	4	Thu Aug 19 18:48:27 2004
Unicode string	2+	㬮䄥

```
0 3B 2E 25 41                    ;.%A
```

software\Microsoft\Windows NT\CurrentVersion   C Start: 0 Selected: 4

Registry Viewer interprets the hex value

Although Fred displayed the value of Installed Date in hexadecimal, it presents a table in the right of the lower pane to interpret the hex values. Thus, we did not have to highlight the value and select the interpretation option.


Forensic Registry EDitor (fred) v0.2.0 - /home/sansforensics/14204-software (read-only)
File Edit Reports Help
Hex viewer

Node	Last mod. time
Shared Tools	2004/08/19 22:31:52
Shared Tools Location	2004/08/19 22:31:32
SmartCard	2004/08/19 22:35:54
Speech	2004/08/19 17:01:58
SystemCertificates	2004/08/19 17:01:58
Tcpip	2004/08/19 22:20:20
TelnetServer	2004/08/19 22:35:36
Terminal Server Client	2004/08/19 22:25:28
Tracing	2004/08/19 23:05:08
Transaction Server	2004/08/19 22:26:59
TShoot	2004/08/19 22:31:52
Tuning Spaces	2004/08/19 22:36:08
UPnP Device Host	2004/08/19 22:32:02
WAB	2004/08/19 22:37:52
WBEM	2004/08/27 15:08:43
Windows	2004/08/19 22:53:19
Windows Media Device Manager	2004/08/19 22:37:58
Windows Messaging Subsystem	2004/08/19 22:31:51
Windows NT	2004/08/19 17:01:59
CurrentVersion	2004/08/27 15:08:22
Accessibility	2004/08/19 17:01:58
AeDebug	2004/08/19 17:01:58
Asr	2004/08/19 17:01:58
Classes	2004/08/19 17:01:58
Compatibility	2004/08/19 22:32:28
Compatibility32	2004/08/19 17:01:58
Console	2004/08/19 17:01:58
Drivers	2004/08/19 17:01:58
drivers.desc	2004/08/19 22:33:41
Drivers32	2004/08/19 22:33:41
EFS	2004/08/19 17:01:58
Embedding	2004/08/19 17:01:58
Event Viewer	2004/08/19 17:01:58
File Manager	2004/08/19 17:01:58
Font Drivers	2004/08/19 17:01:58
FontDPI	2004/08/19 17:01:58
FontMapper	2004/08/19 17:01:58
Fonts	2004/08/19 17:01:58
FontSubstitutes	2004/08/19 17:01:58
GRE_Initialize	2004/08/27 15:08:16
HotFix	2004/08/19 17:01:58
ICM	2004/08/19 22:31:59
Image File Execution Options	2004/08/19 22:31:59
IME Compatibility	2004/08/19 17:01:58
IMM	2004/08/19 17:01:58

Key	Type	Value
BuildLab	REG_SZ	2600.xpclient.010817-1148
CurrentBuild	REG_SZ	1.511.1 () (Obsolete data - do not use)
CurrentBuildNumber	REG_SZ	2600
CurrentType	REG_SZ	Uniprocessor Free
CurrentVersion	REG_SZ	5.1
DigitalProductId	REG_BINARY	a4 00 00 00 03 00 00 00 35 35 32 37 34 2d 36 34 30 2d 30 31 34 37 33 30 36 2d 32 33 36 38 34 00 2e 00 00 00 41 32 32 2d 30 30 30 30 31 00 00 00 00 00 00 00 14 b3 4b cc f7 44 da
InstallDate	REG_DWORD	0x41252e3b
LicenseInfo	REG_BINARY	34 54 ae dc c7 2e 3d e5 8b 15 06 1a 8c 74 a6 55 8c 9b 49 a9 3f 24 2b 04 44 ef 79 88 c1 32 38 cf 52 0c d7 57 da 9b 35 aa d9 44 b3 64 b9 21 6d 76 74 56 d2 4e 91 6e 7f ac
PathName	REG_SZ	C:\WINDOWS
ProductId	REG_SZ	55274-640-0147306-23684
ProductName	REG_SZ	Microsoft Windows XP
RegDone	REG_SZ	
RegisteredOrganization	REG_SZ	N/A
RegisteredOwner	REG_SZ	Greg Schardt
SoftwareType	REG_SZ	SYSTEM
SourcePath	REG_SZ	D:\
SystemRoot	REG_SZ	C:\WINDOWS

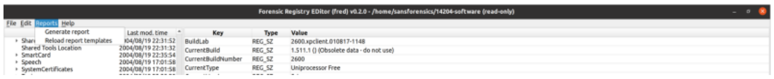
```
0000 3b 2e 25 41                    ;.%A
```

int8:	59
uint8:	59
int16:	11835
uint16:	11835
int32:	1092955707
uint32:	1092955707
unixtime:	2004/08/19 22:48:27

Byte offset: 0x0000 (0)   ● Little endian ○ Big endian

Fred

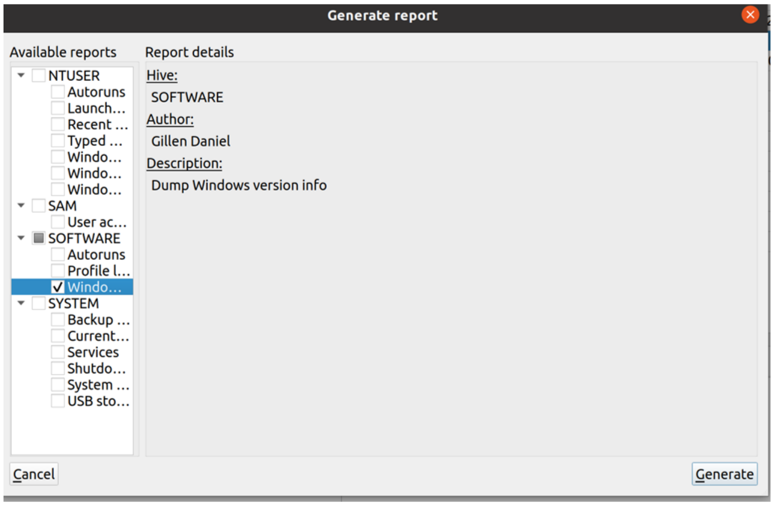
Fred has a feature for generating a report in which the investigator can check the most common artifacts in the registry to be included in a report template. The following screenshots show the Generate Report feature in Fred:


Forensic Registry EDitor (fred) v0.2.0 - /home/sansforensics/14204-software (read-only)
File Edit Reports Help
Generate report
Reload report templates

Node	Last mod. time
Shar	04/08/19 22:31:52
Shared Tools Location	2004/08/19 22:31:32
SmartCard	2004/08/19 22:35:54
Speech	2004/08/19 17:01:58
SystemCertificates	2004/08/19 17:01:58

Key	Type	Value
BuildLab	REG_SZ	2600.xpclient.010817-1148
CurrentBuild	REG_SZ	1.511.1 () (Obsolete data - do not use)
CurrentBuildNumber	REG_SZ	2600
CurrentType	REG_SZ	Uniprocessor Free

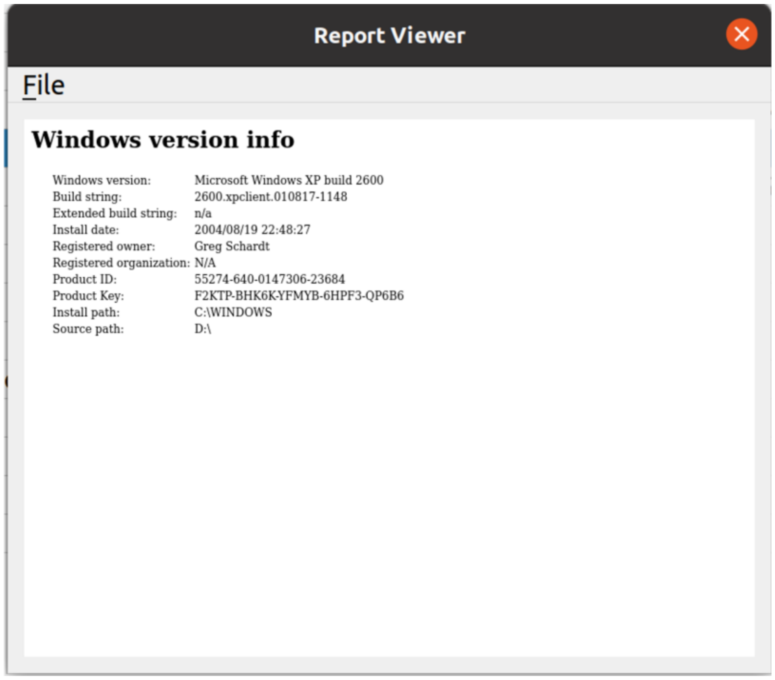
Select the generate report feature in Fred



☑ Windo…

Hive:
SOFTWARE
Author:
Gillen Daniel
Description:
Dump Windows version info

Generating Report in Fred

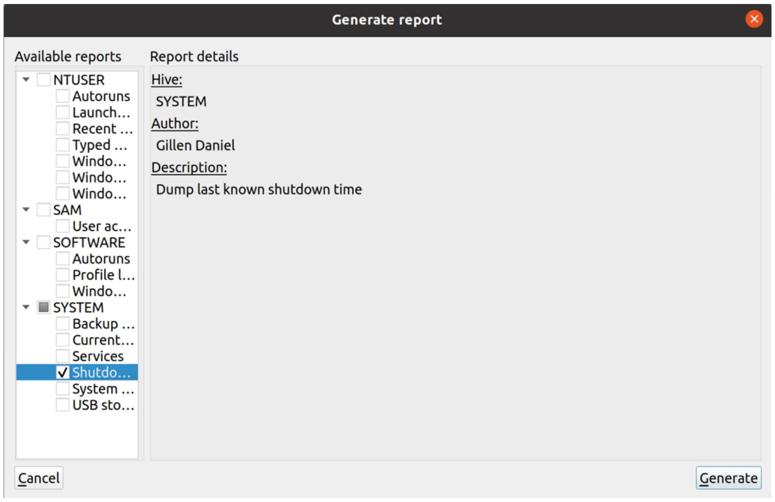


Windows version info

```
Windows version:          Microsoft Windows XP build 2600
Build string:             2600.xpclient.010817-1148
Extended build string:    n/a
Install date:             2004/08/19 22:48:27
Registered owner:         Greg Schardt
Registered organization: N/A
Product ID:               55274-640-0147306-23684
Product Key:              F2KTP-BHK6K-YFMYB-6HPF3-QP6B6
Install path:             C:\WINDOWS
Source path:              D:\
```

The report in Fred

The Last shutdown time is essential in determining when the suspect machine was active. The path to this value is stored in "\ControlSet001\Control\Windows" from the System Hive file. We used the Generate Report feature in Fred to display the Last shutdown by checking the shutdown option. We already opened the System Hive file in Fred, and it would not show any values if the file was not opened.

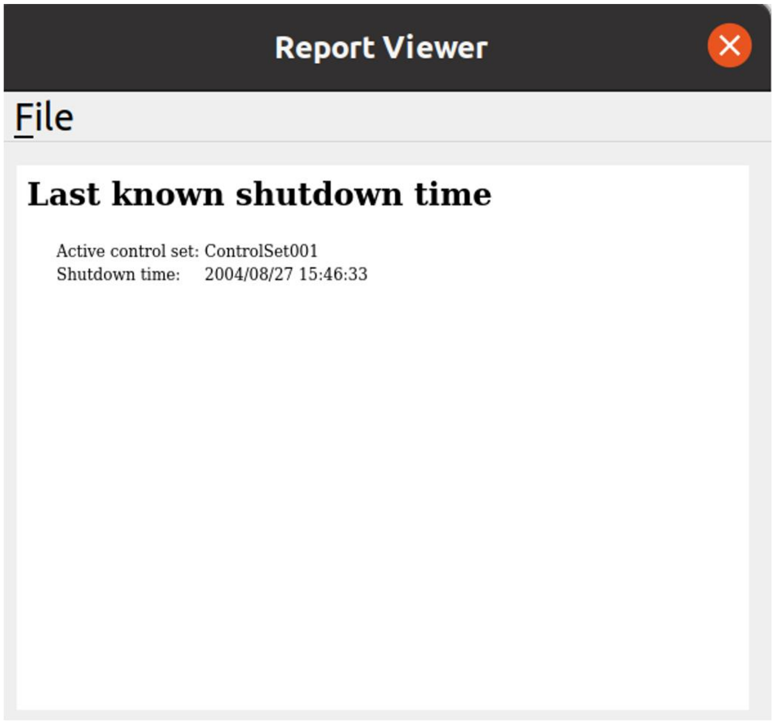


- NTUSER
  - □ Autoruns
  - □ Launch...
  - □ Recent ...
  - □ Typed ...
  - □ Windo...
  - □ Windo...
  - □ Windo...
- SAM
  - □ User ac...
- SOFTWARE
  - □ Autoruns
  - □ Profile l...
  - □ Windo...
- SYSTEM
  - □ Backup ...
  - □ Current...
  - □ Services
  - ☑ Shutdo...
  - □ System ...
  - □ USB sto...

Hive:
SYSTEM
Author:
Gillen Daniel
Description:
Dump last known shutdown time

Last Shutdown in Fred

We noticed that the last time Mr. Evil touched his device was 15:46:33 in 2004/08/27.

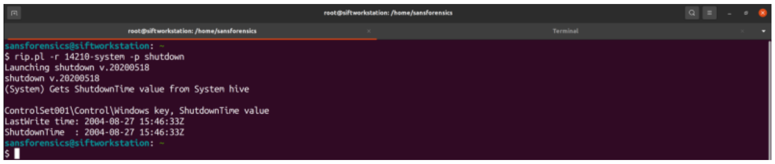


Last known shutdown time

```
Active control set: ControlSet001
Shutdown time:      2004/08/27 15:46:33
```

Last Shutdown in Fred's report

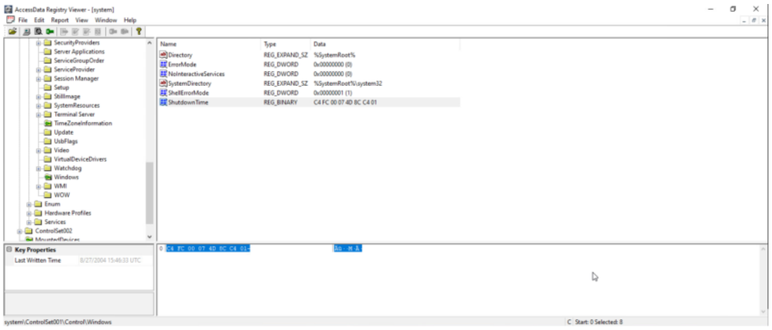
To view the Last Shutdown using RegRipper, we provided the System Hive file along with the plugin "shutdown"



```
sansforensics@siftworkstation: ~
$ rip.pl -r 14210-system -p shutdown
Launching shutdown v.20200518
shutdown v.20200518
(System) Gets ShutdownTime value from System hive

ControlSet001\Control\Windows key, ShutdownTime value
LastWrite time: 2004-08-27 15:46:33Z
ShutdownTime  : 2004-08-27 15:46:33Z
sansforensics@siftworkstation: ~
$ 
```

RegRipper shows the last shutdown

We compared the Shutdown value using the Registry Viewer, and we noticed that the last shutdown display was in Hexadecimal again. Therefore, the investigators have to use the interpretation option to view the shutdown time.

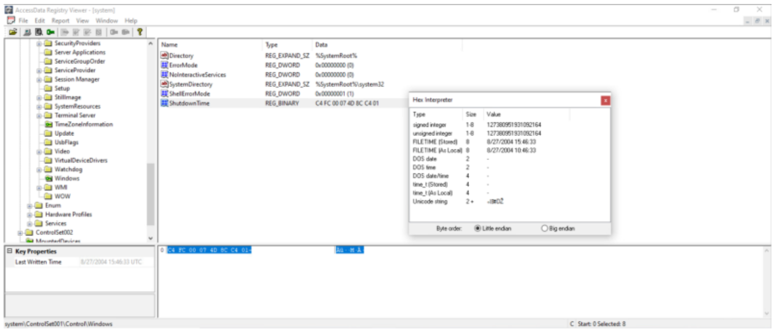


Name	Type	Data
Directory	REG_EXPAND_SZ	%SystemRoot%
ErrorMode	REG_DWORD	0x00000000 (0)
NoInteractiveServices	REG_DWORD	0x00000000 (0)
SystemDirectory	REG_EXPAND_SZ	%SystemRoot%\system32
ShellErrorMode	REG_DWORD	0x00000001 (1)
ShutdownTime	REG_BINARY	C4 FC 00 07 4D 8C C4 01

Key Properties
Last Written Time 8/27/2004 15:46:33 UTC

system\ControlSet001\Control\Windows

Last shutdown in Registry Viewer



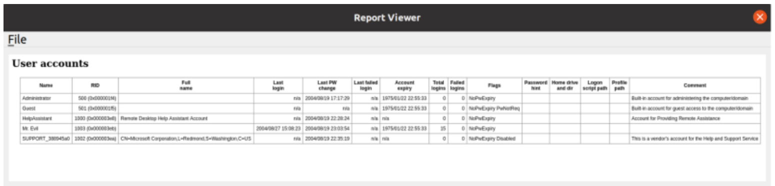
Hex Interpreter

Type	Size	Value
signed integer	1-8	127380951931092164
unsigned integer	1-8	127380951931092164
FILETIME (Stored)	8	8/27/2004 15:46:33
FILETIME (As Local)	8	8/27/2004 10:46:33
DOS date	2	-
DOS time	2	-
DOS date/time	4	-
time_t (Stored)	4	-
time_t (As Local)	4	-
Unicode string	2 +	-ꙮ뒌DŽ

Byte order: ● Little endian ○ Big endian

Interpretation of last shutdown in Registry Viewer

We glean the account users on the suspect machine using the report generation feature from Fred. We noticed that only one user account was active on the system. This can be seen in the Total logins, in which only Mr. Evil had 15 times of logons.

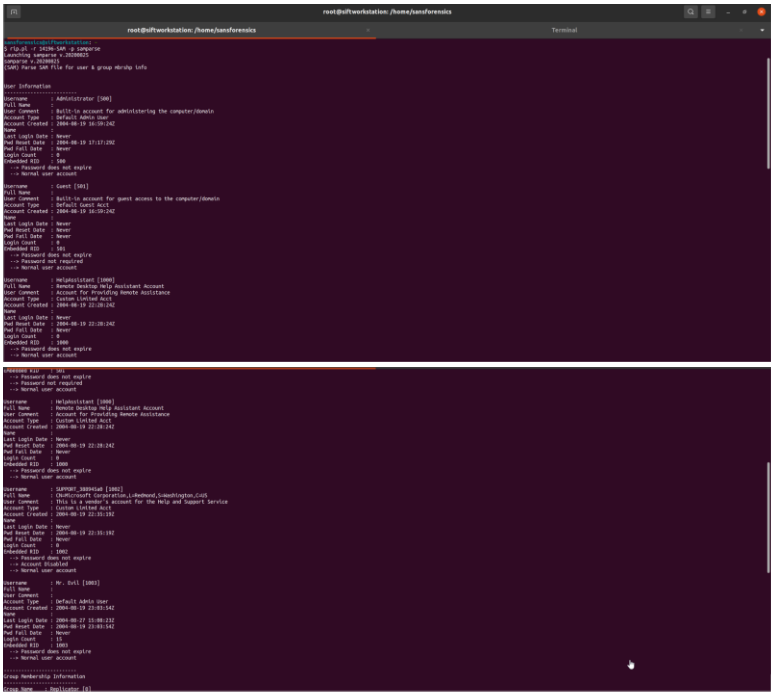


User accounts

Name	RID	Full name	Last login	Last PW change	Last failed login	Account expiry	Total logins	Failed logins	Flags	Password hint	Home drive and dir	Logon script path	Profile path	Comment
Administrator	500 (0x000001f4)		n/a	2004/08/19 17:17:29	n/a	1975/01/22 22:55:33	0	0	NoPwExpiry					Built-in account for administering the computer/domain
Guest	501 (0x000001f5)		n/a	n/a	n/a	1975/01/22 22:55:33	0	0	NoPwExpiry PwNotReq					Built-in account for guest access to the computer/domain
HelpAssistant	1000 (0x000003e8)	Remote Desktop Help Assistant Account	n/a	2004/08/19 22:28:24	n/a	n/a	0	0	NoPwExpiry					Account for Providing Remote Assistance
Mr. Evil	1003 (0x000003eb)		2004/08/27 15:08:23	2004/08/19 23:03:54	n/a	1975/01/22 22:55:33	15	0	NoPwExpiry					
SUPPORT_388945a0	1002 (0x000003ea)	CN=Microsoft Corporation,L=Redmond,S=Washington,C=US	n/a	2004/08/19 22:35:19	n/a	n/a	0	0	NoPwExpiry Disabled					This is a vendor's account for the Help and Support Service

User accounts shown in Fred report

In the RegRipper, we only needed to provide the appropriate plugin and the Hive file to list the user accounts on the suspected system.

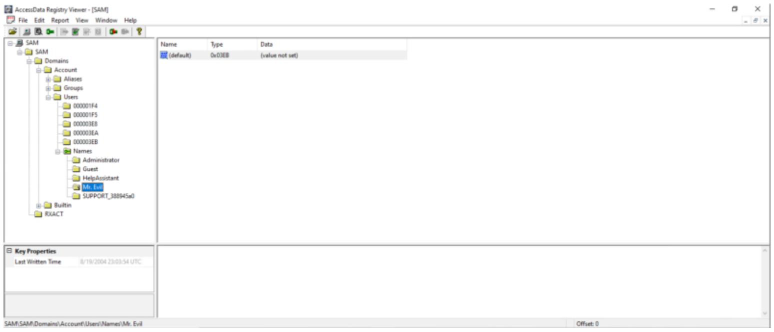

root@siftworkstation: /home/sansforensics
root@siftworkstation: /home/sansforensics
Terminal

```
sansforensics@siftworkstation: ~
$ rip.pl -r 14196-SAM -p samparse
Launching samparse v.20200825
samparse v.20200825
(SAM) Parse SAM file for user & group mbrshp info


User Information
-------------------------
Username        : Administrator [500]
Full Name       :
User Comment    : Built-in account for administering the computer/domain
Account Type    : Default Admin User
Account Created : 2004-08-19 16:59:24Z
Name            :
Last Login Date : Never
Pwd Reset Date  : 2004-08-19 17:17:29Z
Pwd Fail Date   : Never
Login Count     : 0
Embedded RID    : 500
  --> Password does not expire
  --> Normal user account

Username        : Guest [501]
Full Name       :
User Comment    : Built-in account for guest access to the computer/domain
Account Type    : Default Guest Acct
Account Created : 2004-08-19 16:59:24Z
Name            :
Last Login Date : Never
Pwd Reset Date  : Never
Pwd Fail Date   : Never
Login Count     : 0
Embedded RID    : 501
  --> Password does not expire
  --> Password not required
  --> Normal user account

Username        : HelpAssistant [1000]
Full Name       : Remote Desktop Help Assistant Account
User Comment    : Account for Providing Remote Assistance
Account Type    : Custom Limited Acct
Account Created : 2004-08-19 22:28:24Z
Name            :
Last Login Date : Never
Pwd Reset Date  : 2004-08-19 22:28:24Z
Pwd Fail Date   : Never
Login Count     : 0
Embedded RID    : 1000
  --> Password does not expire
  --> Normal user account
```

```
Embedded RID    : 501
  --> Password does not expire
  --> Password not required
  --> Normal user account

Username        : HelpAssistant [1000]
Full Name       : Remote Desktop Help Assistant Account
User Comment    : Account for Providing Remote Assistance
Account Type    : Custom Limited Acct
Account Created : 2004-08-19 22:28:24Z
Name            :
Last Login Date : Never
Pwd Reset Date  : 2004-08-19 22:28:24Z
Pwd Fail Date   : Never
Login Count     : 0
Embedded RID    : 1000
  --> Password does not expire
  --> Normal user account

Username        : SUPPORT_388945a0 [1002]
Full Name       : CN=Microsoft Corporation,L=Redmond,S=Washington,C=US
User Comment    : This is a vendor's account for the Help and Support Service
Account Type    : Custom Limited Acct
Account Created : 2004-08-19 22:35:19Z
Name            :
Last Login Date : Never
Pwd Reset Date  : 2004-08-19 22:35:19Z
Pwd Fail Date   : Never
Login Count     : 0
Embedded RID    : 1002
  --> Password does not expire
  --> Account Disabled
  --> Normal user account

Username        : Mr. Evil [1003]
Full Name       :
User Comment    :
Account Type    : Default Admin User
Account Created : 2004-08-19 23:03:54Z
Name            :
Last Login Date : 2004-08-27 15:08:23Z
Pwd Reset Date  : 2004-08-19 23:03:54Z
Pwd Fail Date   : Never
Login Count     : 15
Embedded RID    : 1003
  --> Password does not expire
  --> Normal user account

-------------------------
Group Membership Information
-------------------------
Group Name     : Replicator [0]
```

Account users shown using RegRipper

Although Registry Viewer has the features of the common areas, it does not show all the details in one place. For example, they could include the “Users” in the common area feature, so all the details can be viewed quickly in one place.



```
SAM
  SAM
    Domains
      Account
        Aliases
        Groups
        Users
          000001F4
          000001F5
          000003E8
          000003EA
          000003EB
          Names
            Administrator
            Guest
            HelpAssistant
            Mr. Evil
            SUPPORT_388945a0
      Builtin
    RXACT
```

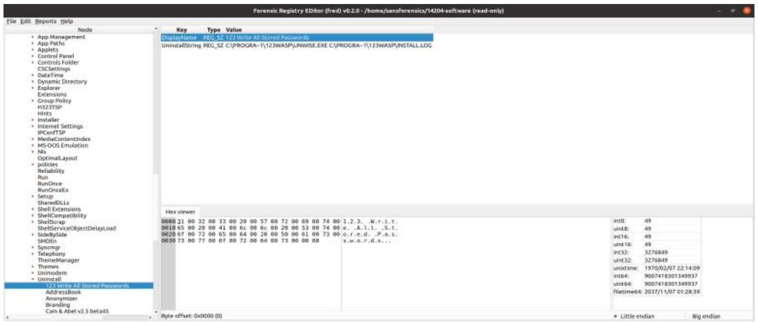
Name	Type	Data
(default)	0x03EB	(value not set)

Key Properties
Last Written Time 8/19/2004 23:03:54 UTC

SAM\SAM\Domains\Account\Users\Names\Mr. Evil   Offset: 0

Registry Viewer shows the user accounts

Although open-source tools can be used to analyze a case, they might have some limitations or flaws. For example, Fred could not accurately interpret the date of uninstalled tools or programs.



Forensic Registry EDitor (fred) v0.2.0 - /home/sansforensics/14204-software (read-only)
File Edit Reports Help
Node
Hex viewer

```
App Management
App Paths
Applets
Control Panel
Controls Folder
CSCSettings
DateTime
Dynamic Directory
Explorer
Extensions
Group Policy
H323TSP
Hints
Installer
Internet Settings
IPConfTSP
MediaContentIndex
MS-DOS Emulation
Nls
OptimalLayout
policies
Reliability
Run
RunOnce
RunOnceEx
Setup
SharedDLLs
Shell Extensions
ShellCompatibility
ShellScrap
ShellServiceObjectDelayLoad
SideBySide
SMDEn
Syncmgr
Telephony
ThemeManager
Themes
Unimodem
Uninstall
  123 Write All Stored Passwords
  AddressBook
  Anonymizer
  Branding
  Cain & Abel v2.5 beta45
```

Key	Type	Value
DisplayName	REG_SZ	123 Write All Stored Passwords
UninstallString	REG_SZ	C:\PROGRA~1\123WASP\UNWISE.EXE C:\PROGRA~1\123WASP\INSTALL.LOG

```
0000 31 00 32 00 33 00 20 00 57 00 72 00 69 00 74 00 1.2.3. .W.r.i.t.
0010 65 00 20 00 41 00 6c 00 6c 00 20 00 53 00 74 00 e. .A.l.l. .S.t.
0020 6f 00 72 00 65 00 64 00 20 00 50 00 61 00 73 00 o.r.e.d. .P.a.s.
0030 73 00 77 00 6f 00 72 00 64 00 73 00 00 00       s.w.o.r.d.s...
```

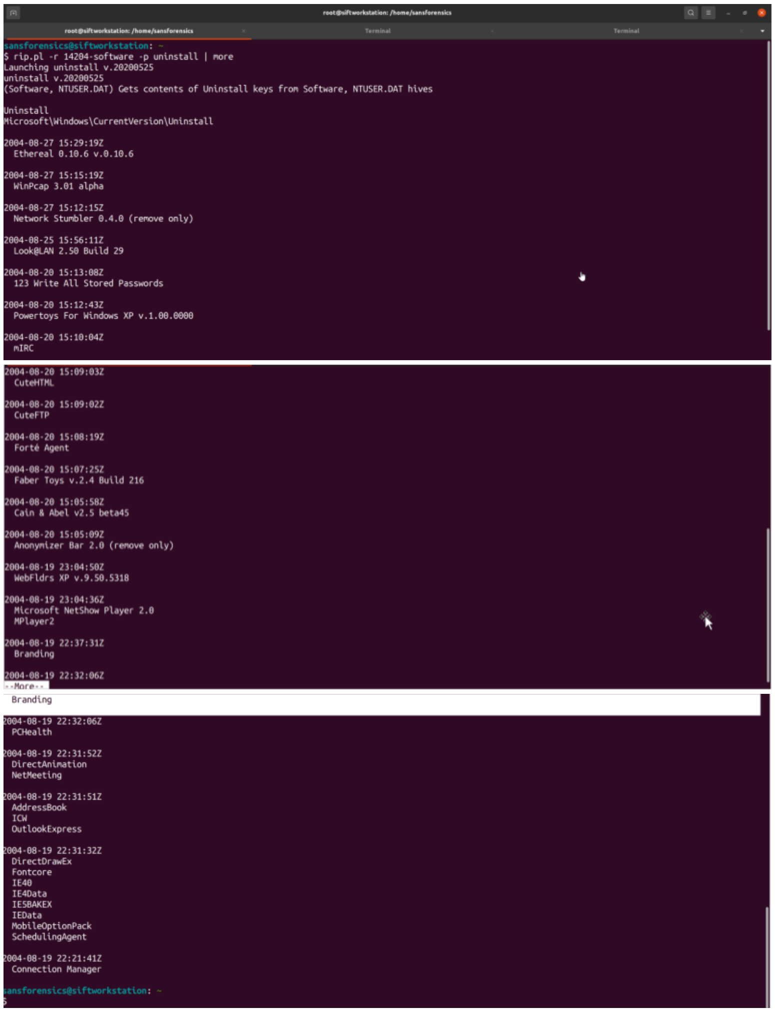
int8:	49
uint8:	49
int16:	49
uint16:	49
int32:	3276849
uint32:	3276849
unixtime:	1970/02/07 22:14:09
int64:	9007418301349937
uint64:	9007418301349937
filetime64:	2037/11/07 01:28:39

Byte offset: 0x0000 (0)

● Little endian ○ Big endian

Fred could not interpret the date

By gleaning the same information using RegRipper, we recommend using more than one tool to ensure the output's accuracy. For example, the following screenshots display how the uninstalled program along with their uninstalled date:



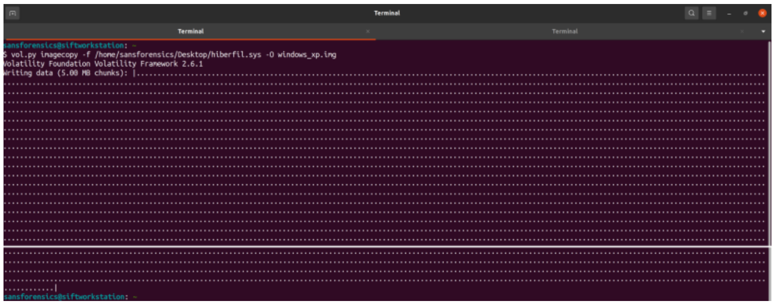
root@siftworkstation: /home/sansforensics
root@siftworkstation: /home/sansforensics
Terminal
Terminal

```
sansforensics@siftworkstation: ~
$ rip.pl -r 14204-software -p uninstall | more
Launching uninstall v.20200525
uninstall v.20200525
(Software, NTUSER.DAT) Gets contents of Uninstall keys from Software, NTUSER.DAT hives

Uninstall
Microsoft\Windows\CurrentVersion\Uninstall

2004-08-27 15:29:19Z
  Ethereal 0.10.6 v.0.10.6

2004-08-27 15:15:19Z
  WinPcap 3.01 alpha

2004-08-27 15:12:15Z
  Network Stumbler 0.4.0 (remove only)

2004-08-25 15:56:11Z
  Look@LAN 2.50 Build 29

2004-08-20 15:13:08Z
  123 Write All Stored Passwords

2004-08-20 15:12:43Z
  Powertoys For Windows XP v.1.00.0000

2004-08-20 15:10:04Z
  mIRC
2004-08-20 15:09:03Z
  CuteHTML

2004-08-20 15:09:02Z
  CuteFTP

2004-08-20 15:08:19Z
  Forté Agent

2004-08-20 15:07:25Z
  Faber Toys v.2.4 Build 216

2004-08-20 15:05:58Z
  Cain & Abel v2.5 beta45

2004-08-20 15:05:09Z
  Anonymizer Bar 2.0 (remove only)

2004-08-19 23:04:50Z
  WebFldrs XP v.9.50.5318

2004-08-19 23:04:36Z
  Microsoft NetShow Player 2.0
  MPlayer2

2004-08-19 22:37:31Z
  Branding

2004-08-19 22:32:06Z
--More--
  Branding

2004-08-19 22:32:06Z
  PCHealth

2004-08-19 22:31:52Z
  DirectAnimation
  NetMeeting

2004-08-19 22:31:51Z
  AddressBook
  ICW
  OutlookExpress

2004-08-19 22:31:32Z
  DirectDrawEx
  Fontcore
  IE40
  IE4Data
  IE5BAKEX
  IEData
  MobileOptionPack
  SchedulingAgent

2004-08-19 22:21:41Z
  Connection Manager

sansforensics@siftworkstation: ~
$
```

RegRipper display the uninstalled programs

Analyze the hiberfil.sys file

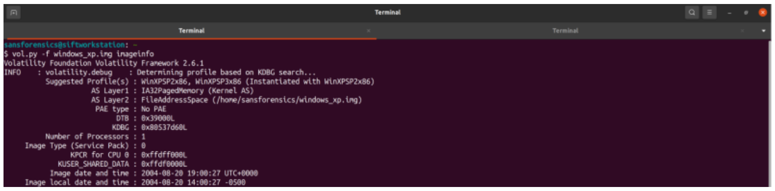
For the sake of completion, we attempted to find as much evidence as possible in the suspect machine. We found a hiberfil.sys file and exported it for analysis. The hiberfil.sys file is used to save the state of a Windows Operating System in the hard drive when the system goes to hibernate mode. The saved state of the system might have some volatile data from memory. We dumped the hiberfil.sys into disk image using a plugin from Volatility called “imagecopy” and the output file will be “windows_xp.img”



```
sansforensics@siftworkstation: ~
$ vol.py imagecopy -f /home/sansforensics/Desktop/hiberfil.sys -O windows_xp.img
Volatility Foundation Volatility Framework 2.6.1
Writing data (5.00 MB chunks): |...............................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...............................................................................................
...........|
sansforensics@siftworkstation: ~
```

Volatility dump the hiberfil.sys into disk image

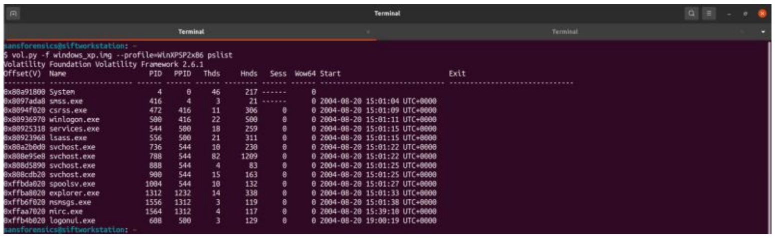
After dumping the file, we needed to specify the image's profile using the “imageinfo.”



```
sansforensics@siftworkstation: ~
$ vol.py -f windows_xp.img imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/sansforensics/windows_xp.img)
                      PAE type : No PAE
                           DTB : 0x39000L
                          KDBG : 0x80537d60L
          Number of Processors : 1
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2004-08-20 19:00:27 UTC+0000
     Image local date and time : 2004-08-20 14:00:27 -0500
```

Finding the profile

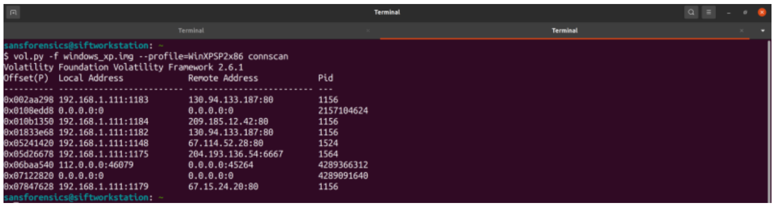
We could view the running process by providing the “pslist” plugin. We noticed that “mirc.exe” is associated with one of the uninstalled programs used as an Internet Relay Chat client. Another interesting process is “msmsgs.exe” which is used as part of Windows Messenger.



```
sansforensics@siftworkstation: ~
$ vol.py -f windows_xp.img --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x80a91800 System                    4      0     46      217 ------      0
0x8097ada8 smss.exe                416      4      3       21 ------      0 2004-08-20 15:01:04 UTC+0000
0x8094f020 csrss.exe               472    416     11      306      0      0 2004-08-20 15:01:09 UTC+0000
0x80936970 winlogon.exe            500    416     22      500      0      0 2004-08-20 15:01:11 UTC+0000
0x80925318 services.exe            544    500     18      259      0      0 2004-08-20 15:01:15 UTC+0000
0x80923968 lsass.exe               556    500     21      311      0      0 2004-08-20 15:01:15 UTC+0000
0x80a2b0d0 svchost.exe             736    544     10      230      0      0 2004-08-20 15:01:22 UTC+0000
0x808e95e8 svchost.exe             788    544     82     1209      0      0 2004-08-20 15:01:22 UTC+0000
0x808d5890 svchost.exe             888    544      4       83      0      0 2004-08-20 15:01:25 UTC+0000
0x808cdb20 svchost.exe             900    544     15      163      0      0 2004-08-20 15:01:25 UTC+0000
0xffbda020 spoolsv.exe            1004    544     10      132      0      0 2004-08-20 15:01:27 UTC+0000
0xffba8020 explorer.exe           1312   1232     14      338      0      0 2004-08-20 15:01:33 UTC+0000
0xffb6f020 msmsgs.exe             1556   1312      3      119      0      0 2004-08-20 15:01:38 UTC+0000
0xffaa7020 mirc.exe               1564   1312      4      117      0      0 2004-08-20 15:39:10 UTC+0000
0xffb4b020 logonui.exe             608    500      3      129      0      0 2004-08-20 19:00:19 UTC+0000
sansforensics@siftworkstation: ~
```

The running process shown from the hiberfil.sys

Since the suspect is accused of wardriving, we considered looking for any network connection. We could glean the TCP connection and the remote connection addresses can be a proof of the locations he attempted to exploit.



```
sansforensics@siftworkstation: ~
$ vol.py -f windows_xp.img --profile=WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
0x002aa298 192.168.1.111:1183        130.94.133.187:80         1156
0x0108edd8 0.0.0.0:0                 0.0.0.0:0                 2157104624
0x010b1350 192.168.1.111:1184        209.185.12.42:80          1156
0x01833e68 192.168.1.111:1182        130.94.133.187:80         1156
0x05241420 192.168.1.111:1148        67.114.52.28:80           1524
0x05d26678 192.168.1.111:1175        204.193.136.54:6667       1564
0x06baa540 112.0.0.0:46079           0.0.0.0:45264             4289366312
0x07122820 0.0.0.0:0                 0.0.0.0:0                 4289091640
0x07847628 192.168.1.111:1179        67.15.24.20:80            1156
sansforensics@siftworkstation: ~
```

Remote connection from the suspect machine

Timeline analysis tools

Timeline analysis has become a mainstay of digital forensic analysis in both public and private sectors. They help to explain what was happening on a given device or set of devices during a cybersecurity incident, a crime, a collision, or other event. Through the timeline analysis, an analyst can easily find out when a particular event or transaction happened. It also helps to figure out the other events which took place during the same time interval along with their interconnection to one another. The result of this timeline analysis helps to frame the situation to explain to customers, attorneys, juries, and other stakeholders.

Here we have used 2 tools for getting the result of timeline analysis- Log2Timeline/plaso and FLS Sleuthkit.

log2timeline is a command line tool to extract events from individual files, recursing a directory, for example a mount point, or storage media image or device. log2timeline creates a plaso storage file, which contains the extracted events and various metadata about the collection process alongside information collected from the source data.

The Sleuth Kit as we know is a library and collection of Unix- and Windows-based utilities for extracting data from disk drives and other storage. The FLS Sleuthkit is a tool in sleuthkit specifically used to list the files and directory names in a file system. It will process the contents of a given directory and can display information on deleted files

Whatever tool we use, we had to first download the image, enter proper unix commands compatible to each tool’s working and that way the result comes in the form of a CSV file. The contents of the file are then analyzed and a complete timeline of events are presented as results

Firstly, we had entered the necessary unix commands to get the CSV file required for the timeline analysis

Once the commands were processed, we found the CSV file in the same path and opened it to observe the computer artifacts found in them

From the analysis, we got the following artifacts which have strong evidence against Mr. Evil. For some events, there are multiple instances found, hence here I have given a single instance of it:

1. Logging into/logging off Mr. Evil account now and then:

TIMESTAMP	2004-08-27T15:08:23.000000+00:00
TIMESTAMP TYPE	Last Login Time
SOURCE	REG
MESSAGE	[HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users] Username: Mr. Evil Full name: Comments: RID: 1003 Login count: 15

TIMESTAMP	2004-08-26T16:04:08.000000+00:00
TIMESTAMP TYPE	Last Shutdown Time
SOURCE	REG
MESSAGE	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Windows] Description: ShutdownTime

2. Doing password resets to Mr. Evil account:

TIMESTAMP	2004-08-19T23:03:54.000000+00:00
TIMESTAMP TYPE	Last Password Reset

SOURCE	REG
MESSAGE	[HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users] Username: Mr. Evil Full name: Comments: RID: 1003 Login count: 15

3. Searching the internet for articles about hacking:

TIMESTAMP	0000-00-00T00:00:00.000000+00:00
TIMESTAMP TYPE	Not a time
SOURCE	WEBHIST
MESSAGE	Location: http://g.msn.com/0USHP/1016?!hacking

4. Visiting websites which has blogs and articles about hacking:

TIMESTAMP	0000-00-00T00:00:00.000000+00:00
TIMESTAMP TYPE	Not a time
SOURCE	WEBHIST
MESSAGE	Location: http://www.elitehackers.com/cgi-bin/ref/referers.cgi?

5. Using rootkits, spywares, packet sniffing tools, etc

TIMESTAMP	2004-08-20T15:09:18.000000+00:00
TIMESTAMP TYPE	Last Checked Time
SOURCE	WEBHIST
MESSAGE	Location: Visited: Mr. Evil@file:///D:/Drivers/GhostWare/Receipt.rtf Number of hits: 1 Cached file size: 0 HTTP headers:

TIMESTAMP	0000-00-00T00:00:00.000000+00:00
TIMESTAMP TYPE	Content Modification Time
SOURCE	WEBHIST
MESSAGE	Location: http://cdn1.tribalfusion.com/media/191336/flash_wrap_v4.js Number of hits: 1 Cached file: HYU1BON0\flash_wrap_v4[1].js Cached file size: 3936 HTTP headers: HTTP/1.1 200 OK - P3P: CP="NOI DEVa TAIa OUR BUS" - Content-Length: 3936 - Content-Type: application/x-javascript - - ~U:mr. evil -

TIMESTAMP	2004-08-20T15:04:51.000000+00:00

TIMESTAMP TYPE	Last Visited Time
SOURCE	WEBHIST
MESSAGE	Location: Visited: Mr. Evil@file:///D:/Drivers/Anonyymizer/keys.txt Number of hits: 1 Cached file size: 0 HTTP headers:

Comparison between both tools:

More than FLS Sleuthkit, Log2Timeline/Plaso has more information which includes Windows event logs, link files, prefetch, shellbags, etc. These would be very useful in a case where the data stolen is of very sensitive nature, involves high number of users and large size of data

On the other hand, Log2Timeline/Plaso takes too much time to create the CSV file whereas FLS Sleuthkit takes very less time to do the same. This factor can be very useful if data stolen was not too sensitive, involves handful of users and size of data stolen was less

References

1. Fortinet. 2021. *What Is Wardriving? | Fortinet*. [online] Available at: <https://www.fortinet.com/resources/cyberglossary/wardriving> [Accessed 2 December 2021].

2. Autopsy. 2021. *opsy* | *Digital Forensics*. [online] Available at: <https://www.autopsy.com/> [Accessed 2 December 2021].

3. J, H., 2021. *Netsh WLAN Commands for Windows 10 - Find Wifi Key & More!*. [online] WebServerTalk.com. Available at: <https://www.webservertalk.com/netsh-wlan-commands> [Accessed 2 December 2021].

4. Majorgeeks.com. 2021. *Index.dat Analyzer*. [online] Available at: <https://www.majorgeeks.com/files/details/index_dat_analyzer.html> [Accessed 2 December 2021].

5. GitHub. 2021. *GitHub - EricZimmerman/PECmd: Prefetch Explorer Command Line*. [online] Available at: <https://github.com/EricZimmerman/PECmd> [Accessed 2 December 2021].

6. GitHub. 2021. *Prefetch_Parser/Prefetch_Parser_2_0_0_9_Setup.exe at master · markmckinnon/Prefetch_Parser*. [online] Available at: <https://github.com/markmckinnon/Prefetch_Parser/blob/master/Installation%20Files/Prefetch_Parser_2_0_0_9_Setup.exe> [Accessed 2 December 2021].

7. ThreatFix. 2021. *Prefetch Parser*. [online] Available at: <http://www.threatfix.com/prefetch-parser.html> [Accessed 2 December 2021].

8. AccessData. 2021. Registry Viewer 2.0.0. [online] Available at: <https://accessdata.com/product-download/registry-viewer-2-0-0> [Accessed 3 December 2021].

9. Pinguin.lu. 2021. FRED | www.pinguin.lu. [online] Available at: <https://www.pinguin.lu/fred> [Accessed 3 December 2021].

10. GitHub. 2021. GitHub - keydet89/RegRipper3.0: RegRipper3.0. [online] Available at: <https://github.com/keydet89/RegRipper3.0> [Accessed 3 December 2021].

11. Betweentwodfirns.blogspot.com. 2021. NIST Hacking Case Tutorial: Wrap up an Old-School Badguy by Happy Hour. [online] Available at: <https://betweentwodfirns.blogspot.com/2016/04/nist-hacking-case-tutorial-wrap-up-old.html> [Accessed 3 December 2021].

12. @Forensicxs. 2021. Computer Forensics : Hacking Case using Autopsy > @Forensicxs. [online] Available at: <https://www.forensicxs.com/computer-forensics-hacking-case-using-autopsy/> [Accessed 3 December 2021].

13. GitHub. GitHub - *log2timeline/plaso: Super timeline all the things* [online] Available at: <https://github.com/log2timeline/plaso> [Accessed 3 December 2021]

14. Plaso.readthe docs.io - *Welcome to the Plaso documentation* [online] Available at: <https://plaso.readthedocs.io/en/latest/index.html> [Accessed 3 December 2021]

15. May 4th 2020 - *Timeline Analysis In Digital Forensics Investigation & Link Analysis Feature* [online] Available at: <https://www.mailxaminer.com/blog/link-analysis-timeline-analysis-in-digital-forensic/> [Accessed 3 December 2021]

16. September 10th 2020 - Timelines In Digital Forensic Investigation: From Investigation To Court [online] Available at: <https://www.forensicfocus.com/articles/timelines-in-digital-forensic-investigation-from-investigation-to-court/> [Accessed 3 December 2021]

17. Youtube - September 8th 2020 - *Getting Started with Plaso and Log2Timeline - Forensic Timeline Creation* [online] Available at: <https://www.youtube.com/watch?v=sAvyRwOmE10> [Accessed 3 December 2021]