BASIS TECH WEEK

4TH ANNUAL OSDF

2013 Open Source Digital Forensics Conference

Doing More with Less

Willi Ballenthin
Consultant
MANDIANT

Introduction: Willi Ballenthin

- **Professionally**
  - Incident response
  - Malware analysis
  - MANDIANT
- **Personally**:
  - Programming
  - Running
  - NYC, USA
- **Contact**:
  - @williballenthin

Enterprise-grade Incident Response

Effective enterprise-scale IR must be intelligence-driven

- You cannot keep up with an attacker if you deep-dive everything
  - Closing the door on first introductions probably won't get you anywhere
  - They can tunnel faster than you can image
- Manual review does not scale across an enterprise
  - Codify knowledge
  - Distribute jobs and queries across the environment
  - Go hunting (with a plan)

Enterprise-grade Incident Response

Funnel the initial analysis results into the intel cycle

- Analysis of a box should yield actionable intelligence
  - It *may* help paint a picture
  - It *must* help you be in a better place tomorrow
    - Understand scope of compromise
    - Know how to find evil elsewhere

Funnel the initial analysis results into the intel cycle

- Questions to answer:
  - Network, host indicators?
    - Infection vector?
    - Compromised accounts?
    - Malware & persistence?
  - Timeframe?
  - Is more analysis needed?

Enterprise-grade Incident Response

Triage analysis should not take too long

- Prepare for dozens or hundreds of compromised boxes
  - Your time is valuable, and time is of the essence
  - Answer questions and move along
- Upshot: recognize patterns across hosts
  - Bad indicators are verbatim features of a binary (but they're cheap & cost effective)
  - Great indicators describe patterns and generalize well
    - Only humans can do this
- (scripting can help)

Enterprise-grade Incident Response

Enterprise tools or preparation are required

- Have a remote forensic access route ready
  - Agent-based systems can be a great start
  - Custom solutions work too, just make sure they are be maintained
- Acquire data, analyze, and move along!
  - You don't want to wait for images
  - Respond while attackers are still on the system

Hacking (in the RMS sense)

Lets brainstorm a balanced set of data to collect for triage

- Desirables:
  - High “bang for the buck”
  - Low resource requirements
    - Bandwidth
    - Storage
    - Worry
  - Pareto principle

OSDF

Hacking (in the RMS sense)

Lets brainstorm a balanced set of data to collect for triage

Class	Data	Resources Req'd	Completeness
File System	MFT	50 MB	OK
Configuration	Registry Hives	20 MB	Good
Logs	EVT/EVTX	50 MB?	OK
Memory	???		

```
Git/INDXParse - [master●] » zip MFT.copy0.zip MFT.copy0
  adding: MFT.copy0 (deflated 89%)
Git/INDXParse - [master●] » du -sh MFT.copy0*
93M     MFT.copy0
11M     MFT.copy0.zip
```

```
python-registry/testing - [master●] » zip SYSTEM.zip SYSTEM
  adding: SYSTEM (deflated 80%)
python-registry/testing - [master●] » du -sh SYSTEM*
14M     SYSTEM
2.8M    SYSTEM.zip
```

```
python-evtx/testing-evtxs - [master●] » zip Security.evtx.zip Security.evtx
  adding: Security.evtx (deflated 92%)
python-evtx/testing-evtxs - [master●] » du -sh Security.evtx*
21M     Security.evtx
1.7M    Security.evtx.zip
```

These formats compress very well

In compressed 75MB, you can capture the triage data for a typical workstation

MFT Analysis

What to do with an MFT file?

- AnalyzeMFT.py
- Sanderson Forensics MFTView (not OSS)
- TZWorks ntfswalk (not OSS)
- MFT tools descending from INDXParse
  - Pure Python
  - Backed by module MFT.py for easy reuse
  - https://github.com/williballenthin/INDXParse

MFT Analysis

list_mft.py

- MFT timelining utility
- Concise code: around 200 lines
- Supports $SI, multiple $FN, resident INDX records
- Constant memory usage

- How?
  - Each record has an $FN attribute with a name. The $FN attribute has a link to the parent record. Iterate the records and walk up the parent links.

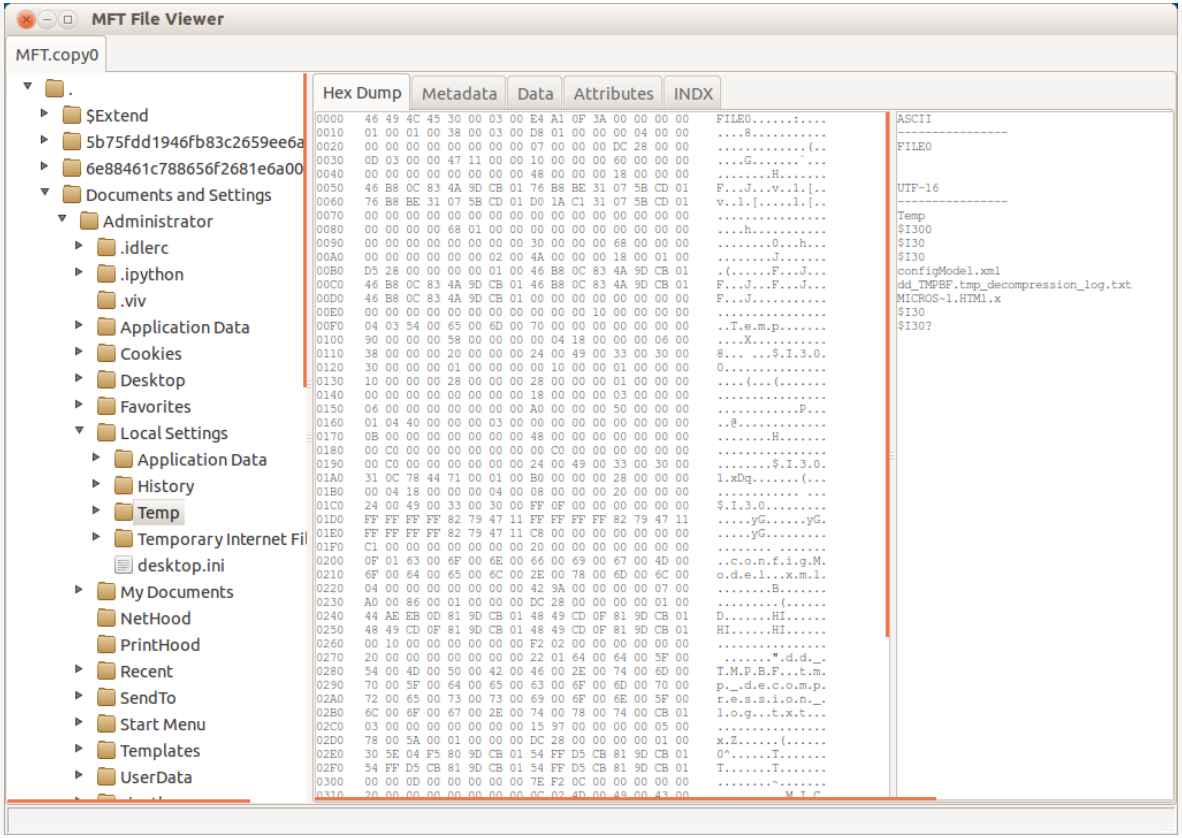
```
Git/INDXParse - [master●] » python list_mft.py  MFT.copy0 | tail | mactime -b -
Thu Dec 16 2010 12:56:07        0 ...b 0 368      0        95064    \.\\System Volume Information\_restore{B8A51974-0EE5-4C6F-AC63-
Mon Jul 02 2012 12:31:58        0 mac. 0 368      0        95064    \.\\System Volume Information\_restore{B8A51974-0EE5-4C6F-AC63-
Thu Jul 05 2012 16:52:53        0 macb 0 430      0        95061    \.\\WINDOWS\Microsoft.NET\Framework\v4.0.30319\ngenrootstoreloc
                                0 macb 0 430      0        95061    \.\\WINDOWS\Microsoft.NET\Framework\v4.0.30319\ngenrootstoreloc
                            16384 macb 0 685      0        95062    \.\\WINDOWS\Temp\Perflib_Perfdata_658.dat (filename, inactive)
                            16384 macb 0 685      0        95062    \.\\WINDOWS\Temp\Perflib_Perfdata_658.dat (inactive)
Thu Jul 05 2012 16:52:57    65536 macb 0 279      0        95063    \.\\WINDOWS\SoftwareDistribution\DataStore\Logs\tmp.edb (filena
                            65536 macb 0 279      0        95063    \.\\WINDOWS\SoftwareDistribution\DataStore\Logs\tmp.edb (inacti
                                0 macb 0 368      0        95064    \.\\System Volume Information\_restore{B8A51974-0EE5-4C6F-AC63-
Thu Jul 05 2012 16:53:04   589824 macb 0 685      0        95060    \.\\WINDOWS\Temp\tnfzggtv.TMP (filename, inactive)
                           589824 macb 0 685      0        95060    \.\\WINDOWS\Temp\tnfzggtv.TMP (inactive)
python list_mft.py MFT.copy0  266.85s user 0.31s system 99% cpu 4:28.32 total
```

list_mft.py

MFT Analysis

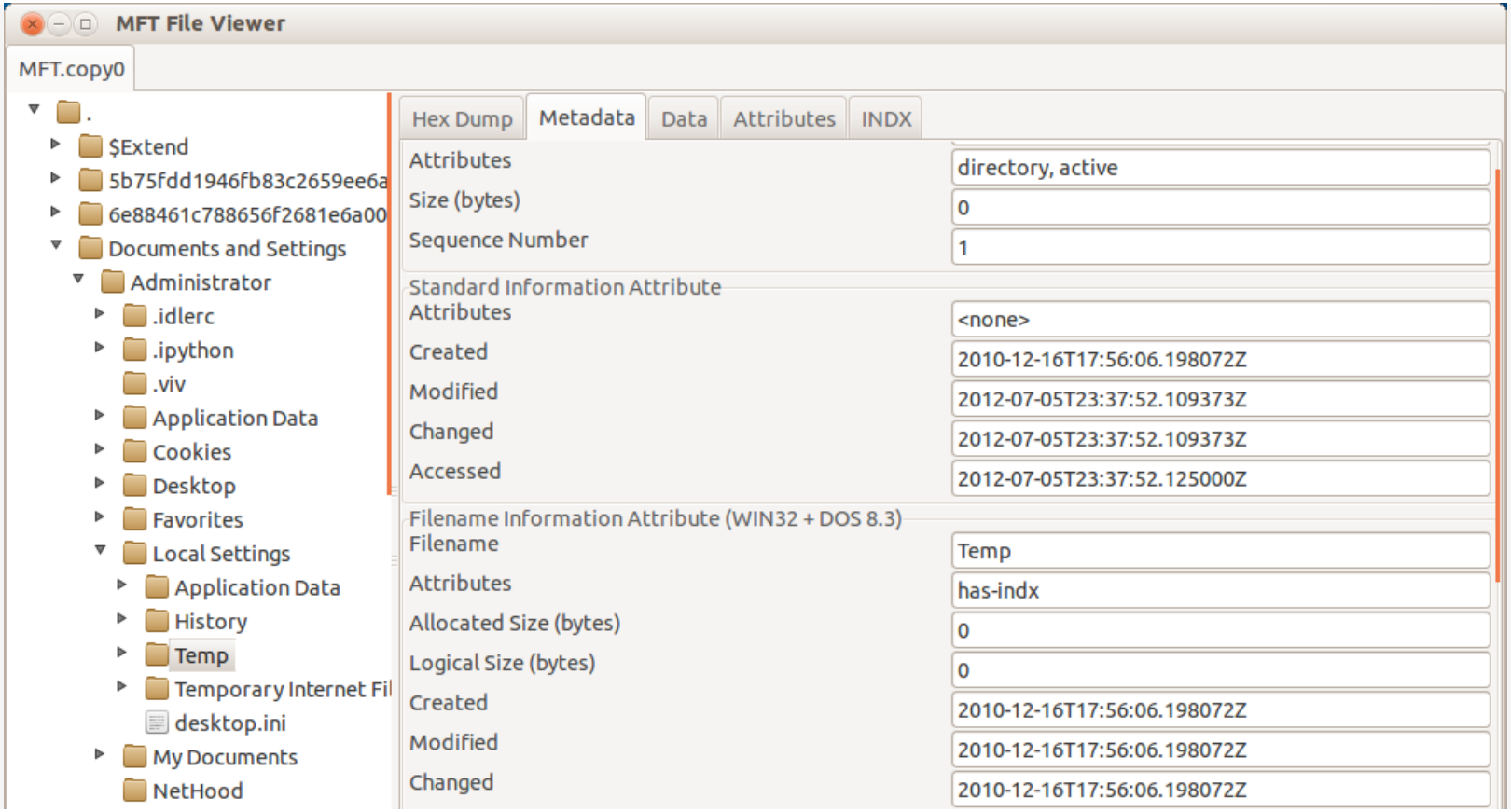
MFTView.py

- Unoriginal name, sorry.
- Pure Python
- Interactive inspection of a file system using only an MFT
- Some features
  - Integrated INDX record parsing
  - Strings, hex view
  - Data extraction and cluster run calculations

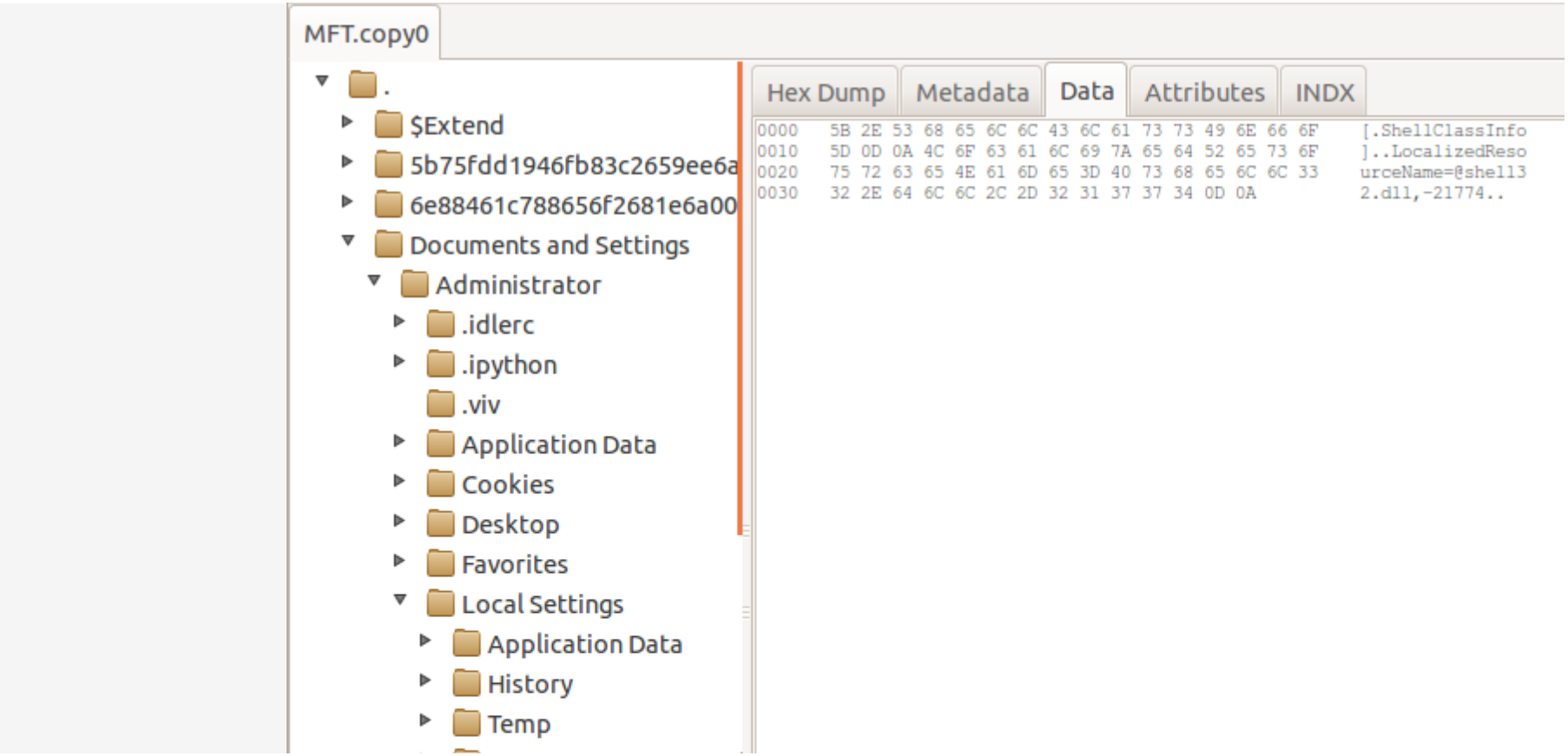


MFTView.py

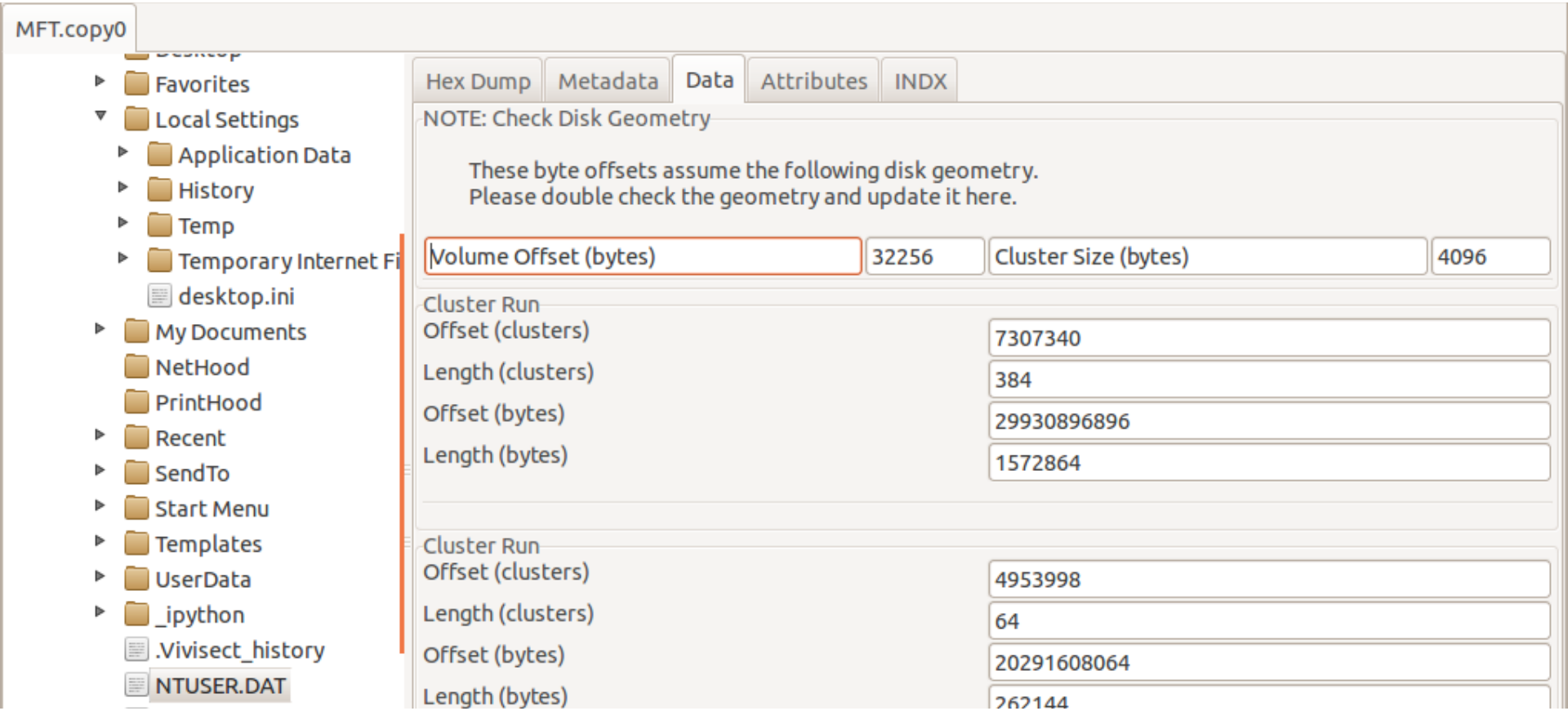
Explore a NTFS file system from just the MFT



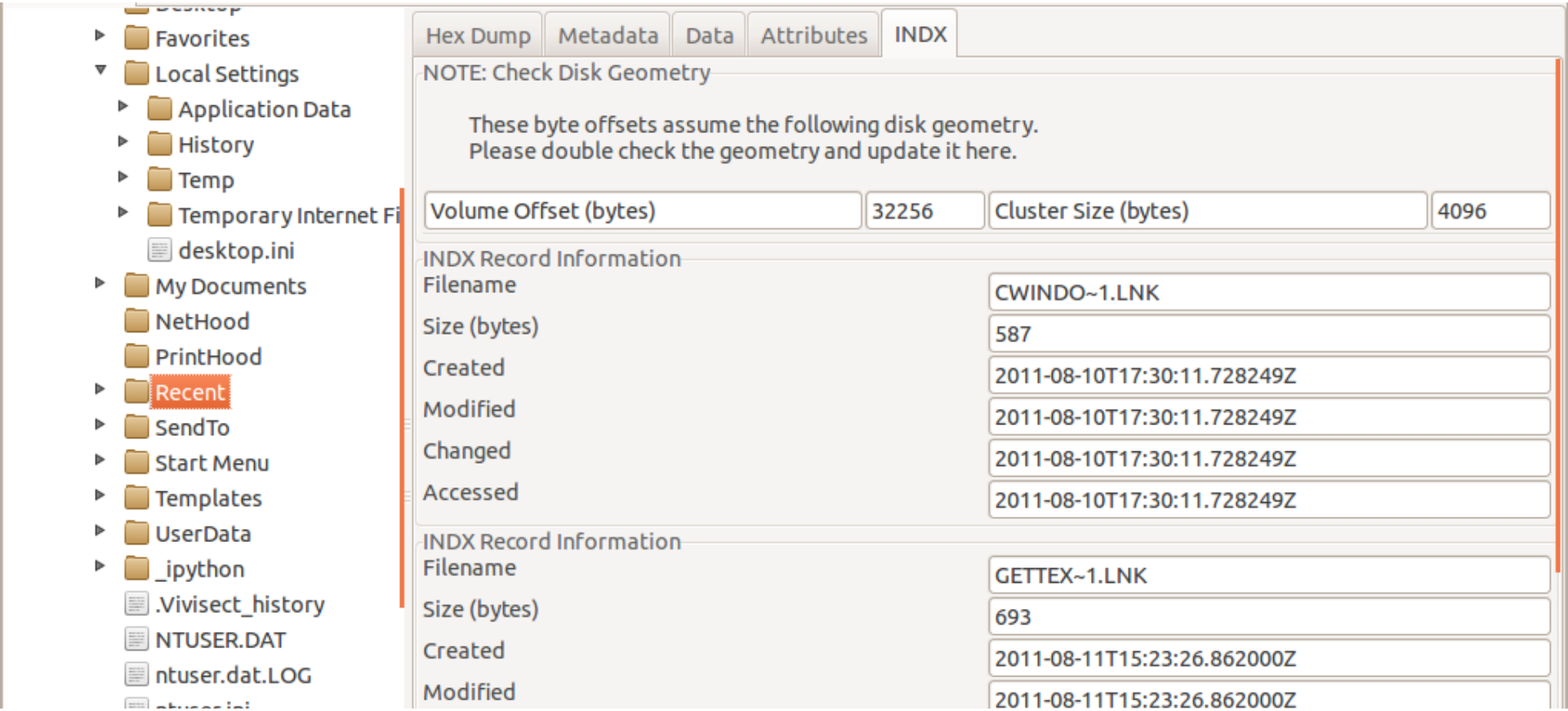
MFTView.py Metadata



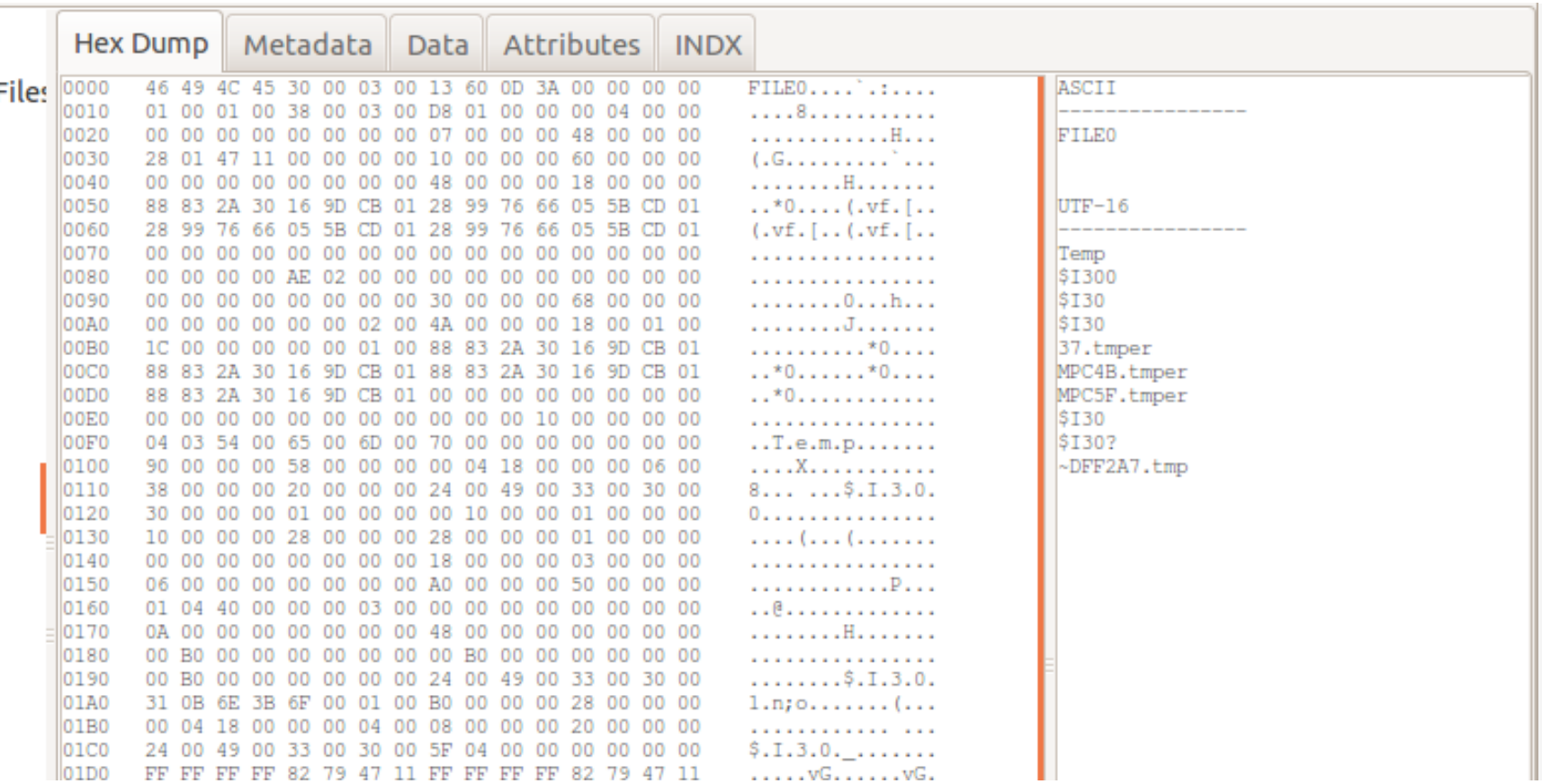
MFTView.py Resident Data



MFTView.py Non-Resident Data



MFTView.py INDX Records

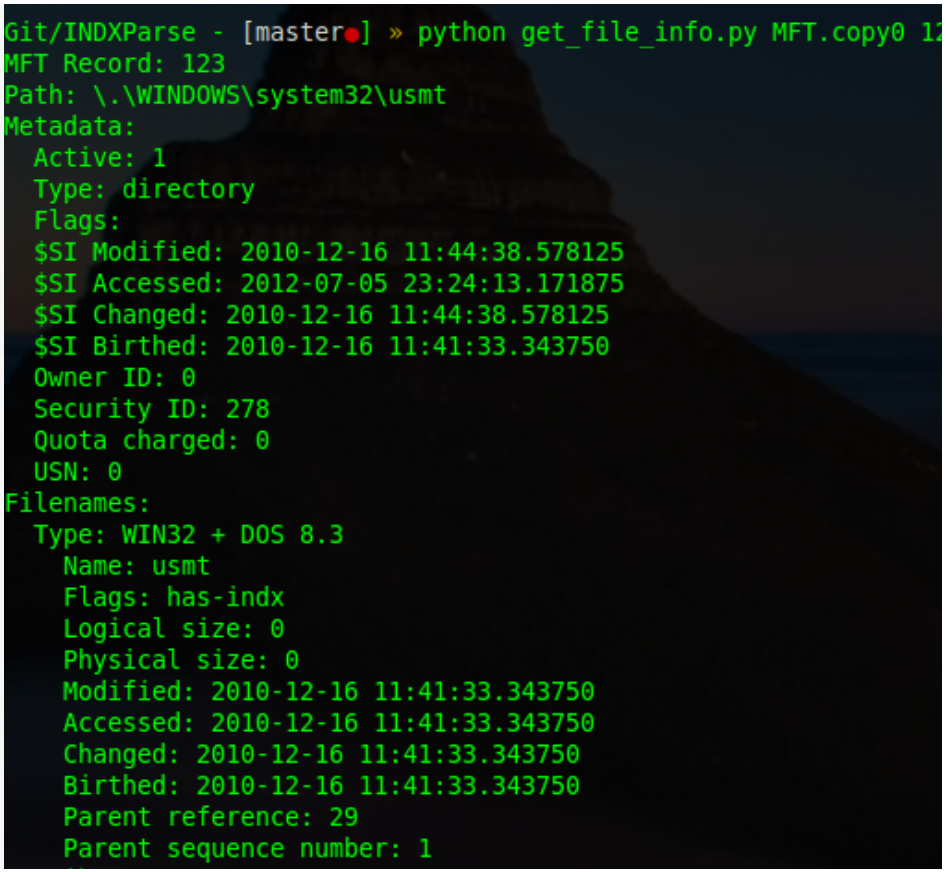


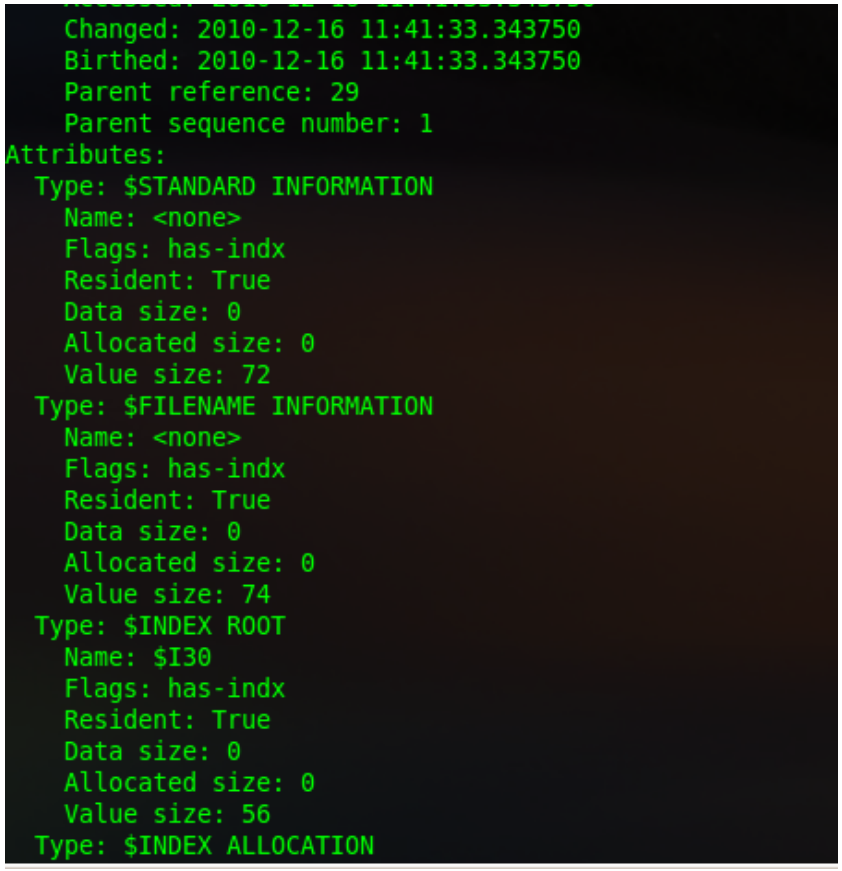
MFTView.py Strings

MFT Analysis

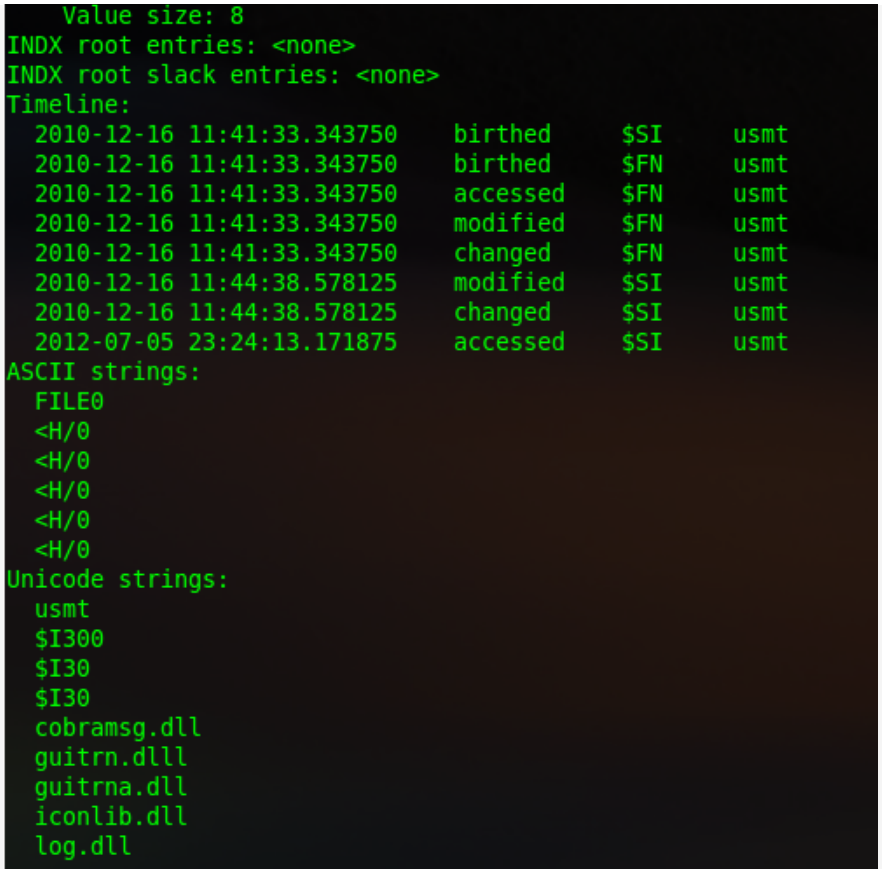
get_file_info.py

- Check the MFT structures for interesting files:
  - Timeline all embedded timestamps
  - Extract strings from MFT stack
  - Review (active & slack) INDX records
  - Pull out many other fields





get_file_info.py



get_file_info.py

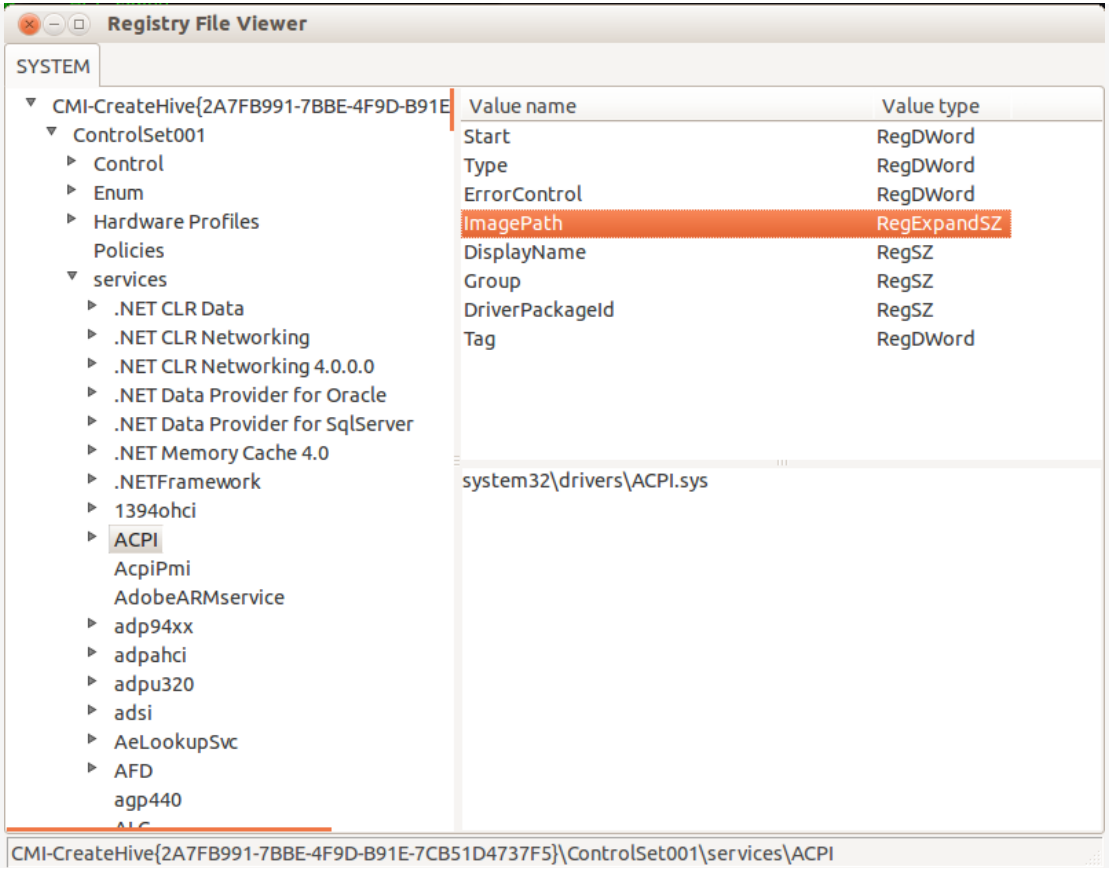
MFT Analysis Workflow

- Naturally: start with a timeline
- Manually review entries with MFTView
- Always rinse and repeat

Registry Analysis

What to do with a Registry hive?

- RegRipper
- MiTeC Windows Registry Analyzer (WRA, not OSS)
- Registry Decoder
- Registry tools included with python-registry
  - Pure Python
  - Backed by stable API in python-registry
  - https://github.com/williballenthin/python-registry



regview.py

Kinda like regedit.exe. Except read-only. And cross platform.

```
python-registry/samples - [master●] » python findkey.py ../testing/SYSTEM -i beep
nvvvpnvvvp
[Paths]
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\services\eventlog\System\beep
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet002\services\eventlog\System\beep

[Value Names]
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\Control\Print : BeepEnabled
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet002\Control\Print : BeepEnabled

[Values]
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\Enum\Root\LEGACY_BEEP\0000 : Service
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\Enum\Root\LEGACY_BEEP\0000 : DeviceDesc
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\services\Beep : DisplayName
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet002\Enum\Root\LEGACY_BEEP\0000 : Service
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet002\Enum\Root\LEGACY_BEEP\0000 : DeviceDesc
  - CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet002\services\Beep : DisplayName
```

findkey.py

Search keys, values, paths with various queries.

```
python-registry/samples - [master●] » python timeline.py ../testing/SYSTEM --bodyfile | mactime -b - | tail
Sat Aug 10 2013 22:03:06        0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
                                0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
                                0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
Sat Aug 10 2013 22:03:26        0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
Sat Aug 10 2013 22:03:27        0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
Sat Aug 10 2013 22:03:30        0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
                                0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
Sat Aug 10 2013 22:03:38        0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
Sat Aug 10 2013 22:06:02        0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
Sat Aug 10 2013 22:06:13        0 .a.. 0 0        0        0        [Registry SYSTEM] CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB5
```

timeline.py

Timeline Registry key modification timestamps

OSDF

Registry Analysis Workflow

- *Windows Registy Analysis* by H. Carvey

Event Log Analysis

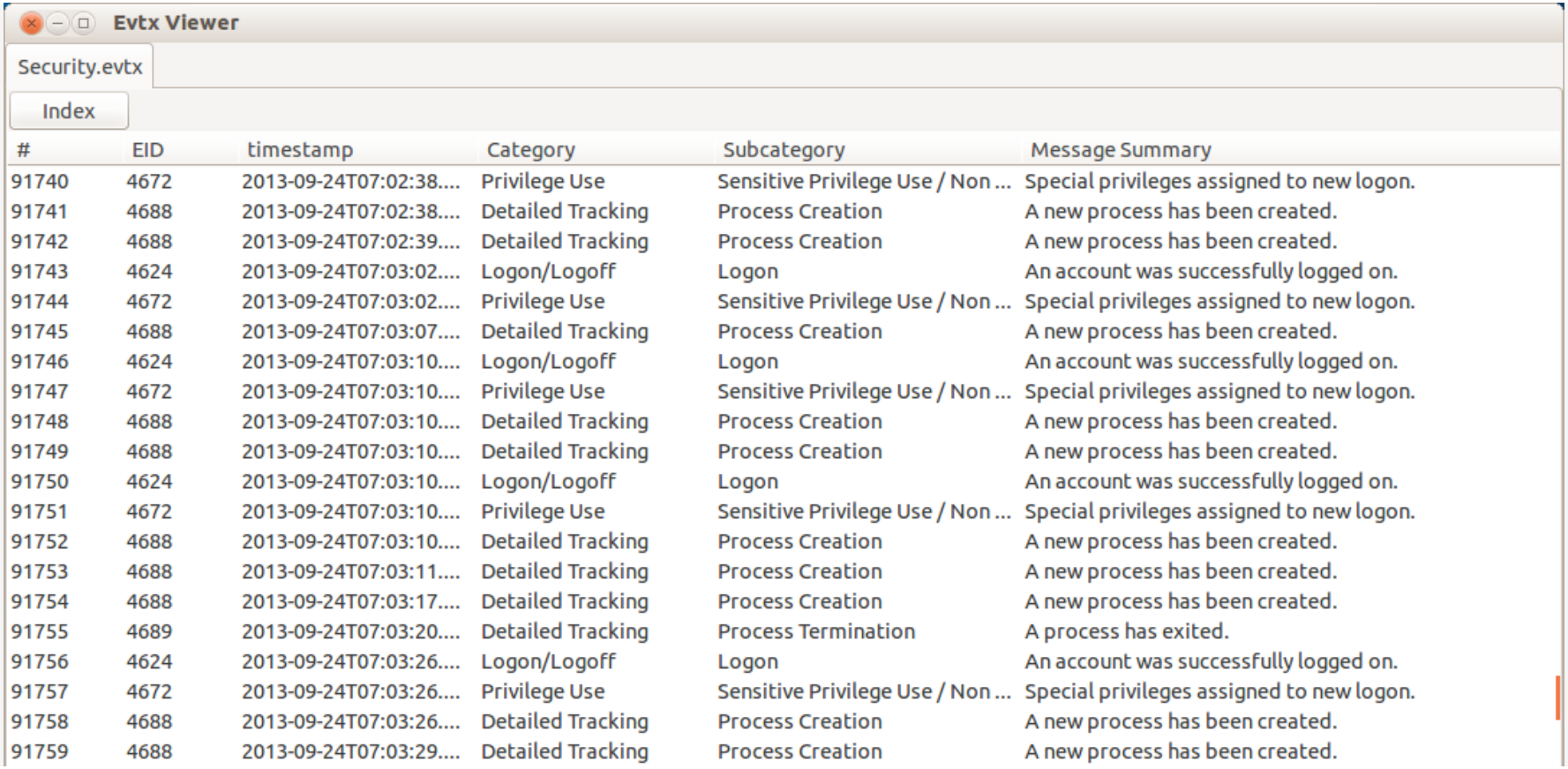
What to do with Event Log files?

- Parse-EVTX
- GrokEVT
- libevtx
- Carving library used by LfLe.py
  - *Carve for Records (not files)* – Jeff Hamm @ SANS DFIR Summit 2012
  - Pure Python module Evt.py
  - https://github.com/williballenthin/LfLe
- Parsing library python-evtx
  - Even more Python
  - https://github.com/williballenthin/python-evtx

Event Log Analysis

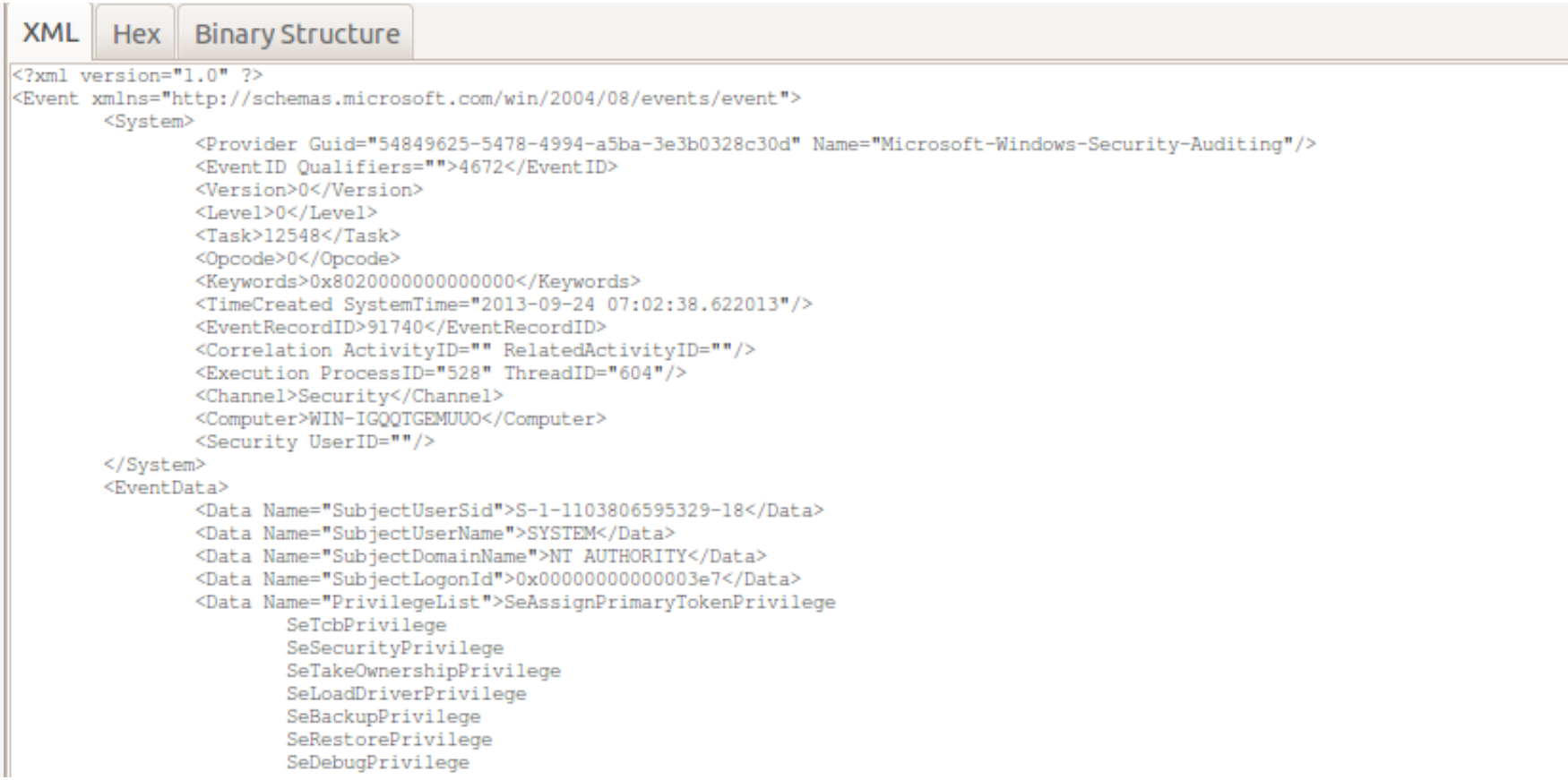
Sneak peek: Event Log Viewer

- Open source GUI for reviewing event log files
-  !
- Indexed search across events
- Multiple views of same data (searching, filtering, sorting)
- Quite slow.
- Lets collaborate!

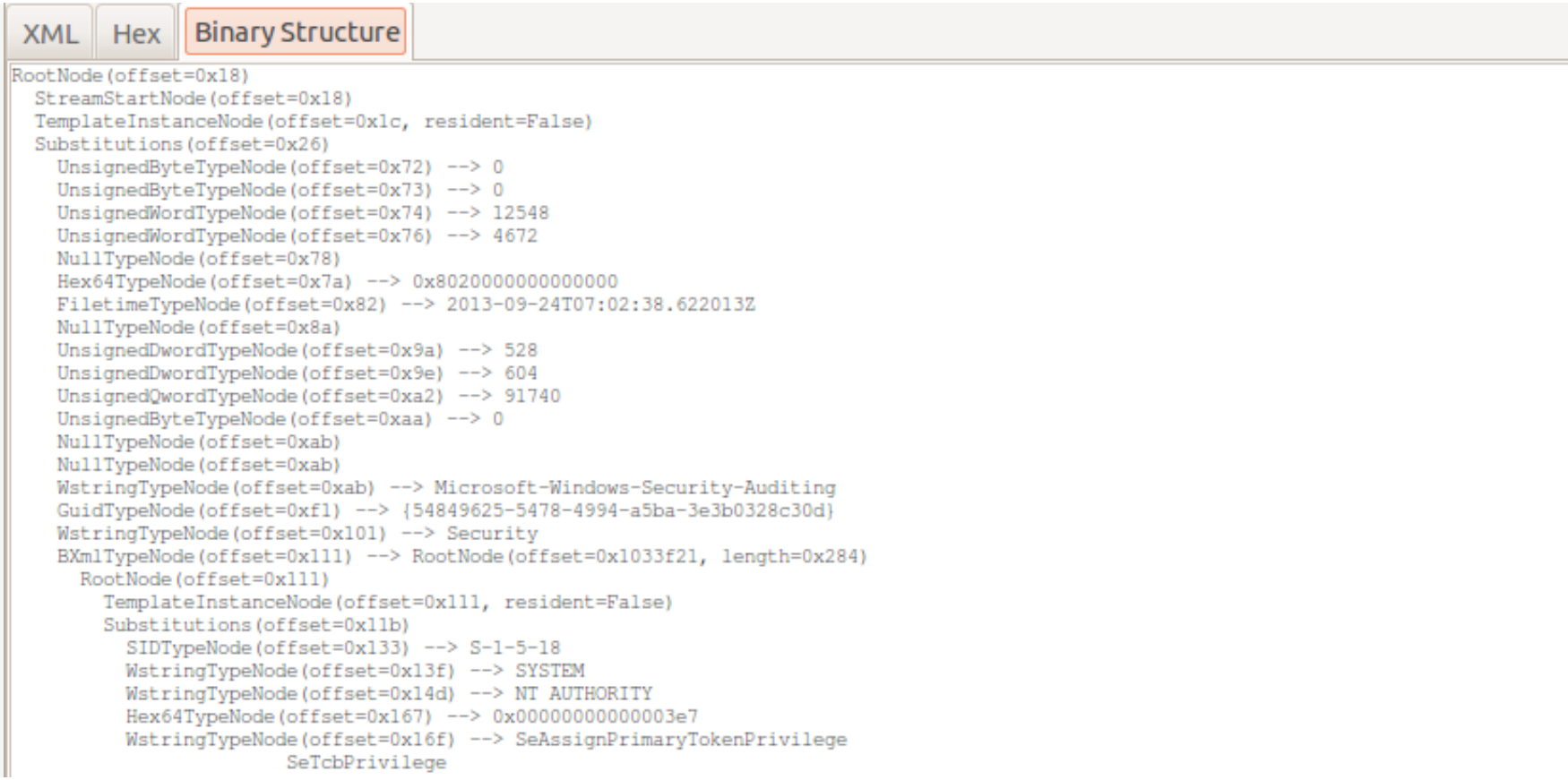


#	EID	timestamp	Category	Subcategory	Message Summary
91740	4672	2013-09-24T07:02:38....	Privilege Use	Sensitive Privilege Use / Non ...	Special privileges assigned to new logon.
91741	4688	2013-09-24T07:02:38....	Detailed Tracking	Process Creation	A new process has been created.
91742	4688	2013-09-24T07:02:39....	Detailed Tracking	Process Creation	A new process has been created.
91743	4624	2013-09-24T07:03:02....	Logon/Logoff	Logon	An account was successfully logged on.
91744	4672	2013-09-24T07:03:02....	Privilege Use	Sensitive Privilege Use / Non ...	Special privileges assigned to new logon.
91745	4688	2013-09-24T07:03:07....	Detailed Tracking	Process Creation	A new process has been created.
91746	4624	2013-09-24T07:03:10....	Logon/Logoff	Logon	An account was successfully logged on.
91747	4672	2013-09-24T07:03:10....	Privilege Use	Sensitive Privilege Use / Non ...	Special privileges assigned to new logon.
91748	4688	2013-09-24T07:03:10....	Detailed Tracking	Process Creation	A new process has been created.
91749	4688	2013-09-24T07:03:10....	Detailed Tracking	Process Creation	A new process has been created.
91750	4624	2013-09-24T07:03:10....	Logon/Logoff	Logon	An account was successfully logged on.
91751	4672	2013-09-24T07:03:10....	Privilege Use	Sensitive Privilege Use / Non ...	Special privileges assigned to new logon.
91752	4688	2013-09-24T07:03:10....	Detailed Tracking	Process Creation	A new process has been created.
91753	4688	2013-09-24T07:03:11....	Detailed Tracking	Process Creation	A new process has been created.
91754	4688	2013-09-24T07:03:17....	Detailed Tracking	Process Creation	A new process has been created.
91755	4689	2013-09-24T07:03:20....	Detailed Tracking	Process Termination	A process has exited.
91756	4624	2013-09-24T07:03:26....	Logon/Logoff	Logon	An account was successfully logged on.
91757	4672	2013-09-24T07:03:26....	Privilege Use	Sensitive Privilege Use / Non ...	Special privileges assigned to new logon.
91758	4688	2013-09-24T07:03:26....	Detailed Tracking	Process Creation	A new process has been created.
91759	4688	2013-09-24T07:03:29....	Detailed Tracking	Process Creation	A new process has been created.

Event Viewer: Overview



Event Viewer: Details



Event Viewer: Structure

XML | Hex | Binary Structure

```
0000   2A 2A 00 00 A0 03 00 00 5C 66 01 00 00 00 00 00    **.......f......
0010   73 24 F8 0D F4 B8 CE 01 0F 01 01 00 0C 01 84 30    s$.............0
0020   7C 5E 26 02 00 00 12 00 00 00 01 00 04 00 01 00    |^&.............
0030   04 00 02 00 06 00 02 00 06 00 02 00 00 00 08 00    ................
0040   15 00 08 00 11 00 10 00 00 00 04 00 08 00 04 00    ................
0050   08 00 08 00 0A 00 01 00 04 00 00 00 00 00 00 00    ................
0060   00 00 46 00 01 00 10 00 0F 00 10 00 01 00 85 02    ..F.............
0070   21 00 00 00 04 31 40 12 01 00 00 00 00 00 00 00    !....1@.........
0080   20 80 73 24 F8 0D F4 B8 CE 01 00 00 00 00 28 01     .s$..........(.
0090   00 00 EC E8 01 01 FC E7 EA 00 10 02 00 00 5C 02    ................
00A0   00 00 5C 66 01 00 00 00 00 00 00 4D 00 69 00 63    ...f.......M.i.c
00B0   00 72 00 6F 00 73 00 6F 00 66 00 74 00 2D 00 57    .r.o.s.o.f.t.-.W
00C0   00 69 00 6E 00 64 00 6F 00 77 00 73 00 2D 00 53    .i.n.d.o.w.s.-.S
00D0   00 65 00 63 00 75 00 72 00 69 00 74 00 79 00 2D    .e.c.u.r.i.t.y.-
00E0   00 41 00 75 00 64 00 69 00 74 00 69 00 6E 00 67    .A.u.d.i.t.i.n.g
00F0   00 25 96 84 54 78 54 94 49 A5 BA 3E 3B 03 28 C3    .%..TxT.I..>;.(.
0100   0D 53 00 65 00 63 00 75 00 72 00 69 00 74 00 79    .S.e.c.u.r.i.t.y
0110   00 0C 01 AE 0F 78 AB 03 1D 00 00 05 00 00 00 0C    .....x..........
0120   00 13 00 0E 00 01 00 1A 00 01 00 08 00 15 00 26    ...............&
0130   02 01 00 01 01 00 00 00 00 00 05 12 00 00 00 53    ...............S
0140   00 59 00 53 00 54 00 45 00 4D 00 00 00 4E 00 54    .Y.S.T.E.M...N.T
0150   00 20 00 41 00 55 00 54 00 48 00 4F 00 52 00 49    . .A.U.T.H.O.R.I
0160   00 54 00 59 00 00 00 E7 03 00 00 00 00 00 00 53    .T.Y...........S
0170   00 65 00 41 00 73 00 73 00 69 00 67 00 6E 00 50    .e.A.s.s.i.g.n.P
0180   00 72 00 69 00 6D 00 61 00 72 00 79 00 54 00 6F    .r.i.m.a.r.y.T.o
0190   00 6B 00 65 00 6E 00 50 00 72 00 69 00 76 00 69    .k.e.n.P.r.i.v.i
01A0   00 6C 00 65 00 67 00 65 00 0D 00 0A 00 09 00 09    .l.e.g.e........
01B0   00 09 00 53 00 65 00 54 00 63 00 62 00 50 00 72    ...S.e.T.c.b.P.r
01C0   00 69 00 76 00 69 00 6C 00 65 00 67 00 65 00 0D    .i.v.i.l.e.g.e..
01D0   00 0A 00 09 00 09 00 09 00 53 00 65 00 53 00 65    .........S.e.S.e
01E0   00 63 00 75 00 72 00 69 00 74 00 79 00 50 00 72    .c.u.r.i.t.y.P.r
```

Event Viewer: Binary

Recap

In the last 30 mins, we covered:

- GUI MFT browser
- CLI MFT timeliner and inspector
- GUI Registry explorer
- CLI Registry timeliner & utils
- GUI EVT(X) tool

…and these work great for rapid triage!

Questions?

OSDF

OSDF