

# Vortessence

## Automating Volatility Memory Forensics

OSDFCon 2014, Reston VA

Endre Bangerter

Beni Urech

A project of the Security Engineering Lab

<http://sel.bfh.ch>

Bern University of Applied Sciences

[info@vortessence.org](mailto:info@vortessence.org)

# Who we are...

- Beni is a security engineer at Swisscom
- Endre works at Security Engineering Lab of Bern Univ. of Applied Sciences

Thanks to Christian Bürgi, Thomas Ender and Patrick Haring

# Why Vortessence?

- **Memory forensics is key technique to detect malware / attack tools, Volatility is THE tool for doing so**
- Detection by finding anomalies
  - Need to memorize normal state of clean system (hard / impossible to memorize)
  - Looking for anomalies can be **rather mechanical and boring task**
- Vortessence **partially automates Volatility based forensics**
  - Better detection
  - Quicker
  - Lower the skills required for first cut analysis
  - Will be open sourced in Winter 2014
- Vortessence is **not rocket science: simple but quite effective**



## Agenda

- Illustrating the problem
- Vortessence basic idea
- Demo
- Release plan & outlook

```
$ vol.py -f image05.bin --profile=WinXPSP3x86 pstree
```

```
Volatility Foundation Volatility Framework 2.3.1
```

```
Name                Pid  PPid  Thds  Hnds Time
-----
0x825c8830:System          4   0   56  194 1970-01-01 00:00:00 UTC+0000
. 0x82172210:smss.exe      384   4    3   19 2013-12-06 09:48:00 UTC+0000
.. 0x82170878:csrss.exe   608  384   13  377 2013-12-06 09:48:00 UTC+0000
.. 0x8221f7e8:winlogon.exe 636  384   19  520 2013-12-06 09:48:02 UTC+0000
... 0x82162950:services.exe 680  636   15  269 2013-12-06 09:48:02 UTC+0000
.... 0x824a9c08:vmacthlp.exe 904  680    1   25 2013-12-06 09:48:02 UTC+0000
```

```
[snip]
```

```
.... 0x82499348:svchost.exe 3148  680   44  840 2013-12-20 15:40:49 UTC+0000
.... 0x82043310:msiexec.exe 1360  680    5  111 2013-12-20 15:41:04 UTC+0000
.... 0x8216f198:svchost.exe  980  680   11  268 2013-12-06 09:48:02 UTC+0000
.... 0x82452248:svchost.exe 1124  680    0  ----- 2013-12-06 09:48:02 UTC+0000
..... 0x82085020:wuauclt.exe 1068 1124    8  258 2013-12-20 15:31:34 UTC+0000
..... 0x82113da0:wuauclt.exe 1368 1124    4  123 2013-12-06 09:49:31 UTC+0000
... 0x82220da0:lsass.exe   692  636   23  353 2013-12-06 09:48:02 UTC+0000
0x821183d0:explorer.exe   1780 1760   17  609 2013-12-06 09:48:07 UTC+0000
. 0x822352c8:vmttoolsd.exe 1880 1780    3  245 2013-12-06 09:48:08 UTC+0000
. 0x82386438:ctfmon.exe   1892 1780    1   78 2013-12-06 09:48:08 UTC+0000
. 0x84f0da88:explore.exe  2400 1780    9  115 2013-12-20 15:40:44 UTC+0000
. 0x81f0da88:iexplore.exe 2280 1780   13  387 2013-12-20 15:33:44 UTC+0000
.. 0x81eccb88:iexplore.exe 2376 2280   23  648 2013-12-20 15:33:47 UTC+0000
```

## Processes - Example of an anomaly

```
$ vol.py -f stuxnet.vmem --profile=WinXPSP3x86 pstree
```

Name	Pid	PPid	Thds	Hnds	Time
0x823c8830:System	4	0	59	403	1970-01-01 00:00:00 UTC+0000
. 0x820df020:smss.exe	376	4	3	19	2010-10-29 17:08:53 UTC+0000
[SNIP]					
.... 0x81f14938:ipconfig.exe	304	968	0	-----	2011-06-03 04:31:35 UTC+0000
.... 0x822843e8:svchost.exe	1032	668	61	1169	2010-10-29 17:08:55 UTC+0000
..... 0x822b9a10:wuauclt.exe	976	1032	3	133	2010-10-29 17:12:03 UTC+0000
..... 0x820ecc10:wscntfy.exe	2040	1032	1	28	2010-10-29 17:11:49 UTC+0000
.... 0x81e61da0:svchost.exe	940	668	13	312	2010-10-29 17:08:55 UTC+0000
.... 0x81db8da0:svchost.exe	856	668	17	193	2010-10-29 17:08:55 UTC+0000
..... 0x81fa5390:wmiprvse.exe	1872	856	5	134	2011-06-03 04:25:58 UTC+0000
.... 0x821a0568:VMUpgradeHelper	1816	668	3	96	2010-10-29 17:09:08 UTC+0000
.... 0x81fee8b0:spoolsv.exe	1412	668	10	118	2010-10-29 17:08:56 UTC+0000
.... 0x81ff7020:svchost.exe	1200	668	14	197	2010-10-29 17:08:55 UTC+0000
.... 0x81c47c00:lsass.exe	1928	668	4	65	2011-06-03 04:26:55 UTC+0000
.... 0x81e18b28:svchost.exe	1080	668	5	80	2010-10-29 17:08:55 UTC+0000
.... 0x8205ada0:alg.exe	188	668	6	107	2010-10-29 17:09:09 UTC+0000
.... 0x823315d8:vmacthlp.exe	844	668	1	25	2010-10-29 17:08:55 UTC+0000
.... 0x81e0eda0:jqs.exe	1580	668	5	148	2010-10-29 17:09:05 UTC+0000
.... 0x81c498c8:lsass.exe	868	668	2	23	2011-06-03 04:26:55 UTC+0000
.... 0x82279998:imapi.exe	756	668	4	116	2010-10-29 17:11:54 UTC+0000
... 0x81e70020:lsass.exe	680	624	19	342	2010-10-29 17:08:54 UTC+0000



```
$ vol.py -f image02.vmsd --profile=Win7SP1x86 dlllist -p 544
```

```
services.exe pid: 544
```

```
Command line : C:\Windows\system32\services.exe
```

Base	Size	LoadCount	Path
0x00210000	0x41000	0xffff	C:\Windows\system32\services.exe
0x77080000	0x13c000	0xffff	C:\Windows\SYSTEM32\ntdll.dll
0x76690000	0xd4000	0xffff	C:\Windows\system32\kernel32.dll
0x753b0000	0x4b000	0xffff	C:\Windows\system32\KERNELBASE.dll
0x765e0000	0xac000	0xffff	C:\Windows\system32\msvcrt.dll
0x76d50000	0xa2000	0xffff	C:\Windows\system32\RPCRT4.dll
0x76c40000	0x1f000	0x2	C:\Windows\system32\IMM32.DLL
0x75080000	0x29000	0x1	C:\Windows\system32\WINSTA.dll
[SNIP]			
0x76d10000	0x35000	0x8	C:\Windows\system32\WS2_32.dll
0x76c60000	0x6000	0x8	C:\Windows\system32\NSI.dll
0x74af0000	0x3c000	0x4	C:\Windows\system32\mswsock.dll
0x74640000	0x5000	0x1	C:\Windows\System32\wshtcpip.dll
0x74ae0000	0x6000	0x1	C:\Windows\System32\wship6.dll
0x72bd0000	0x3000	0x1	C:\Windows\system32\lz32.DLL
0x74b30000	0x16000	0x1	C:\Windows\system32\CRYPTSP.dll
0x748d0000	0x3b000	0x1	C:\Windows\system32\rsaenh.dll

Zeroaccess

Process priorities

...

Files



Unknown drivers

Number of DLLs per process

...

Network ports

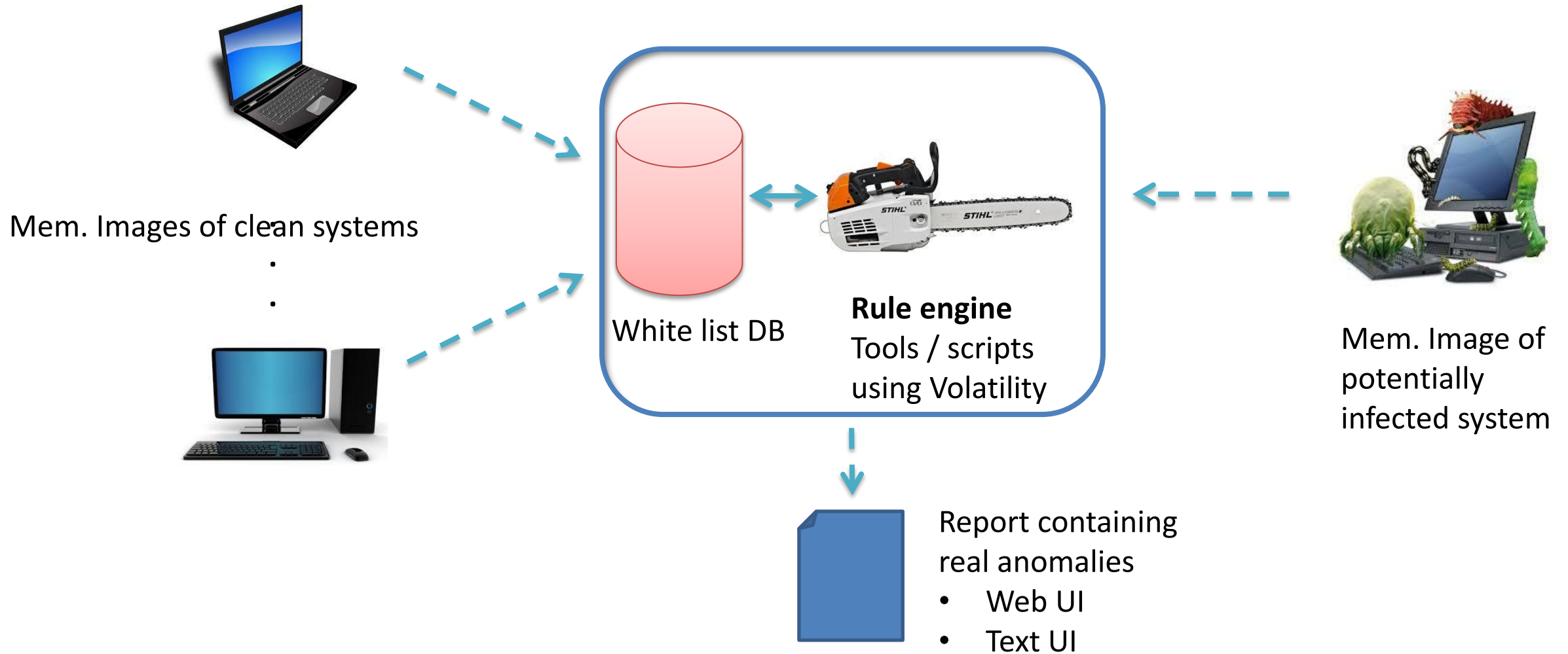
Registry



## Agenda

- Illustrating the problem
- Vortessence basic idea
- Demo
- Release plan & outlook

# Vortessence – Basic idea



# Type of detection rules

- **Simple diff:** Value found in suspect image but not in white list (“a simple diff per volatility tool”)

- **Range checks**

- **Reduction of false positives in some Volatility malware tools**

Base	0x85680550
Path	C:\Windows\system32\lsass.exe
Command line	"C:\Windows\system32\lsass.exe" <input checked="" type="checkbox"/> Whitelist
Parent Process	544 services.exe <input checked="" type="checkbox"/> Whitelist
# of instances	3   1
# Threads	6   7 - 14
# DLLs	46   60 - 63
Base priorities	8   9

# malfind has false positives

```
Process: iexplore.exe Pid: 2376 Address: 0x1b10000  
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE  
Flags: CommitCharge: 2, MemCommit: 1, PrivateMemory: 1, Protection: 6
```

```
0x01b10000 b0 00 eb 70 b0 01 eb 6c b0 02 eb 68 b0 03 eb 64 ...p...l...h...d  
0x01b10010 b0 04 eb 60 b0 05 eb 5c b0 06 eb 58 b0 07 eb 54 ...`...\...X...T  
0x01b10020 b0 08 eb 50 b0 09 eb 4c b0 0a eb 48 b0 0b eb 44 ...P...L...H...D  
0x01b10030 b0 0c eb 40 b0 0d eb 3c b0 0e eb 38 b0 0f eb 34 ...@...<...8...4
```

```
0x1b10000 b000      MOV AL, 0x0  
0x1b10002 eb70      JMP 0x1b10074  
0x1b10004 b001      MOV AL, 0x1  
0x1b10006 eb6c      JMP 0x1b10074  
0x1b10008 b002      MOV AL, 0x2  
0x1b1000a eb68      JMP 0x1b10074  
0x1b1000c b003      MOV AL, 0x3  
0x1b1000e eb64      JMP 0x1b10074  
0x1b10010 b004      MOV AL, 0x4  
0x1b10012 eb60      JMP 0x1b10074
```

False positive

Sort them out manually....

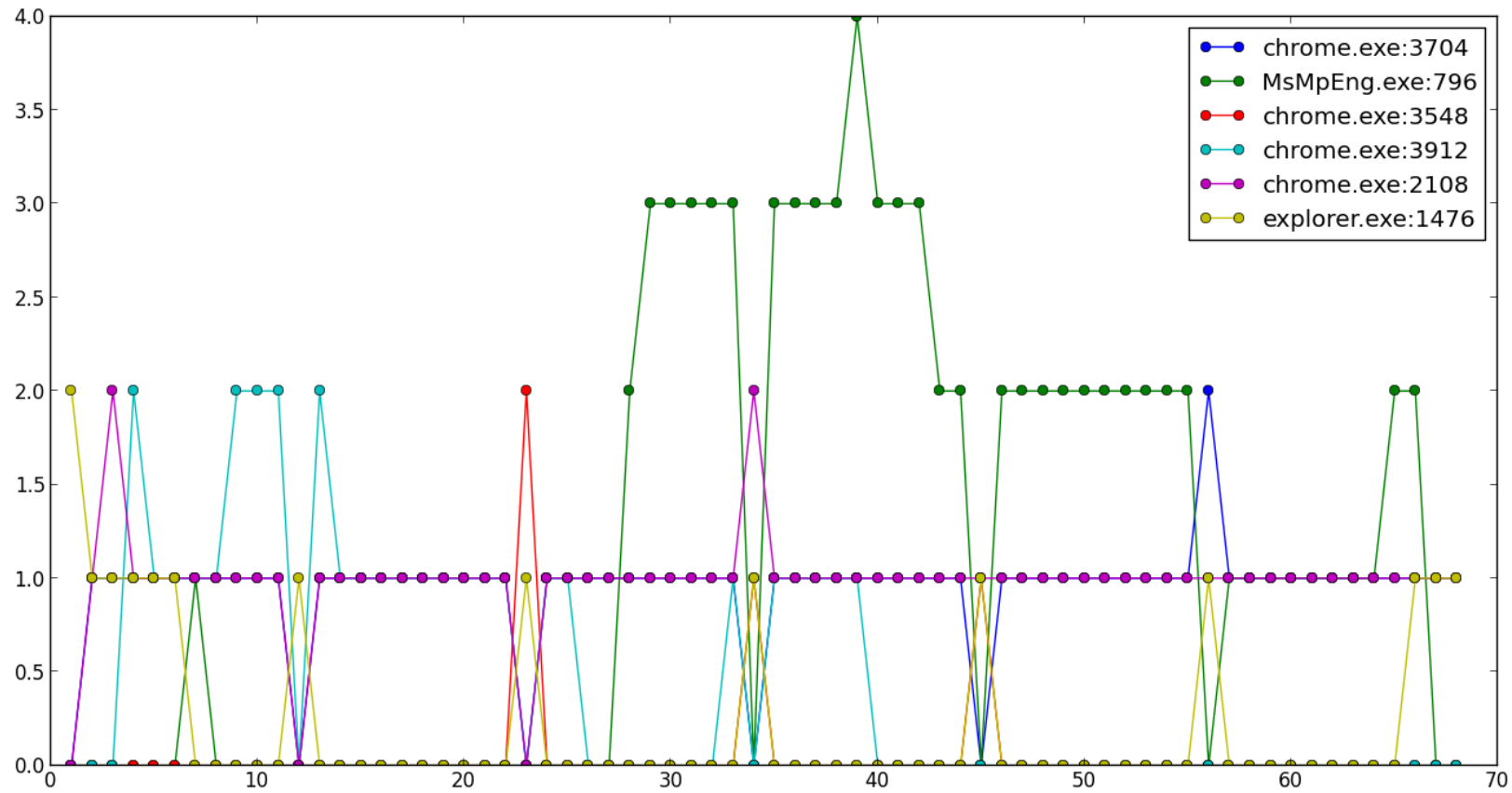
True positive

```
Process: chrome.exe Pid: 4052 Address: 0x130000  
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE  
Flags: CommitCharge: 39, MemCommit: 1, PrivateMemory: 1, Protection: 6
```

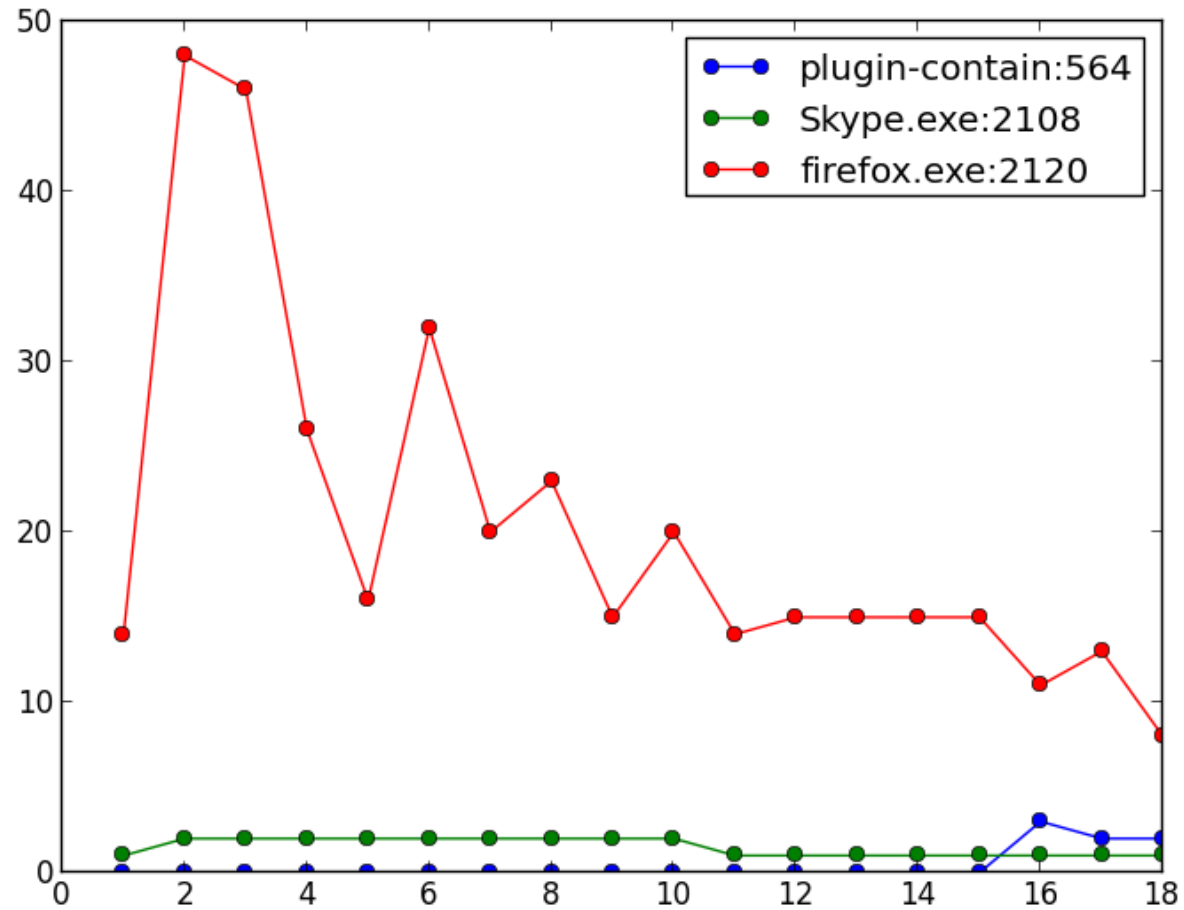
```
0x00130000 4d 5a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 MZ.....  
0x00130010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
0x00130020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
0x00130030 00 00 00 00 00 00 00 00 00 00 00 00 d0 00 00 00 .....
```

```
0x130000 4d      DEC EBP  
0x130001 5a      POP EDX  
0x130002 0000     ADD [EAX], AL  
0x130004 0000     ADD [EAX], AL  
0x130006 0000     ADD [EAX], AL  
0x130008 0000     ADD [EAX], AL  
0x13000a 0000     ADD [EAX], AL
```

# Understanding malfind false positives



# Understanding malfind false positives



## Agenda

- Illustrating the problem
- Vortessence basic idea
- Demo
- Release plan & outlook

# Technical Overview

- Python 2.7
- Volatility 2.4, extended with JSON output
- 2 interfaces, CLI and web frontend
- Django for web frontend and DB abstraction layer for the CLI
- MySQL DB for backend, any DB supported by Django will work. Even SQLite 😊
- Runs on every platform Volatility does
- Supports Win7 and newer, including 64bit



# Agenda

- Illustrating the problem
- Vortessence basic idea
- Demo
- Release plan & outlook

# Release Plan

**Winter 2014: Version 0.1** (will be available at <http://vortessence.org>)

- Based on JSON output from Volatility 2.5
- Include the following plugins:

## *Processes and DLLs*

- pslist
- dlllist
- handles
- getsids
- privs

## *Process Memory*

- vadinfo

## *Kernel Memory and Objects*

- modscan
- ssdt
- filescan
- unloadedmodules

## *Networking*

- netscan

## *Registry (own plugin)*

## *Windows Malware*

- malfind
- svcscan
- ldrmodules
- apihooks
- idt
- gdt
- threads
- callbacks
- driverirp
- devicetree
- timers

# Future releases

- **More Volatility plugins**
- **Rule framework:** hiding Vortessence internals, allowing to write rules with Python and Voaltility knowledge only
- **Rating / weighting anomalies:** Some anomalies are strong indicators, others are weak
- **PE file diffing**
- **Whitelist management:**
  - Import
  - Export --> sharing!
  - Etc.