Public Beta: Insider's Preview

Mike Auty, Andrew Case | 16 October 2019 | #OSDFCon

Looking Back

December 2011 Volatility Technology Preview (Internal Project)

October 2012 OMFW: Volatility 3.0 on Roadmap (Technology Preview)

November 2013 "Volatility Past & Present" (Volatility 3.0: Python 3/Pagefile/Performance)

December 2013 Volatility Technology Preview → Rekall

November 2014 "Restructuring Memory" (Unified output) → 2.5

Volatility 3

October 2019 Volatility 3.0 Public Beta

| 2006 | 2007 | 2008 | 2009 | | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | | | 2019 |

Volatility 2.x

December, 2016 Volatility 2.6 (Windows 10 / Server 2016)

October 2015 Volatility 2.5 (Unified Output / Community)

September 2014 Volatility Foundation created

August 2014 Volatility 2.4 (Art of Memory Forensics)

October 2013 Volatility 2.3.1 (Mac OSX and Android ARM)

August 2013 Volatility Plugin Contest

May 2013 Month of Volatility Plugins II

October 2012 Volatility 2.2 (Linux Support)

September 2012 Month of Volatility Plugins I

August 2012 Volatility 2.1 (Malware and 64-bits)

August 2011 Volatility 2.0 (Beyond XP)

January 2009 Malfind 1.0

August 2008 Volatility 1.3 (DFRWS Contest, OMFW, and Plugins)

August 2007 Volatility 1.1.1 (Scanning and VAD)

February 2007 VolaTools (Black Hat - Initial Public Release)

February 2006 FATKit (Digital Investigation Journal)

Memory Forensics: 2006 vs. 2019
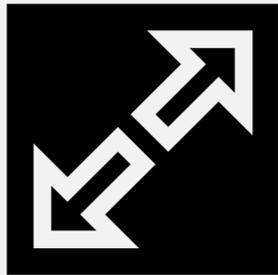
SAMPLE SIZE

THEN	NOW
<= 4GB	>= 16GB

>= 128GB common
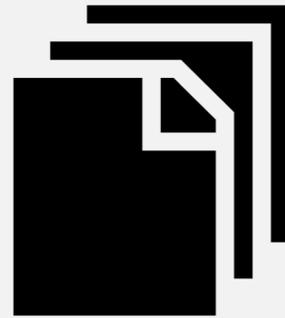
SAMPLE QUANTITY

| THEN	NOW
1	5+

VOLATILITY ANALYSIS TASKS

THEN	NOW
<= 10	>= 50

Operating System Release Cycles in 2019 [3, 4]

- **Feature updates** will be released twice per year, around March and September. As the name suggests, these will add new features to Windows 10, delivered in bite-sized chunks compared to the previous practice of Windows releases every 3-5 years.

- Linux_5.3 Released Sun, 15 September 2019 (70 days)
- Linux_5.2 Released Sun, 7 July 2019 (63 days)
- Linux_5.1 Released Sun, 5 May 2019 (63 days)
- Linux_5.0 Released Sun, 3 March 2019 (70 days)

The History of Vol3
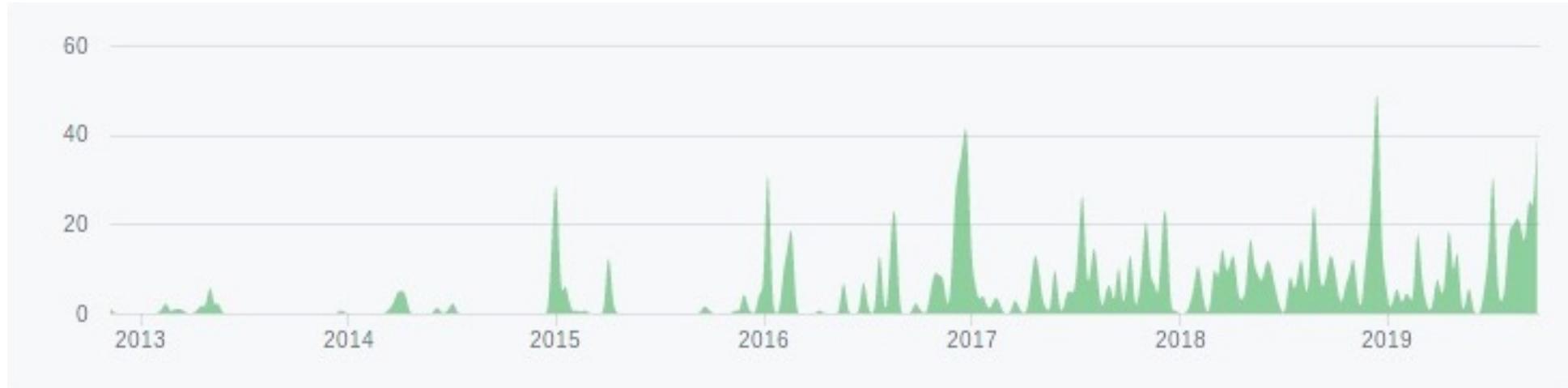
- Many novel ideas attempted and refined before being put into the stable code base

- The goal: Meet the needs of the next decade of memory analysis

What is New in Volatility 3?

- **All** of it

- **Every** line of code

- **Entire framework** (backend, plugins, etc.) was completely rewritten and redesigned

What is New in Volatility 3? Cont.

- Written in Python 3

- Major performance boost!
  - Natively supports multi-processing and memory caches

- Much simpler integration into other libraries and user interfaces

What is New in Volatility 3? Cont.

- No more --profile for any OS!
  - Automatic detection of profiles
  - Extraction of known-good data from debug info vs hardcoded

- 32bit apps on 64bit kernels natively supported
 - Proper Wow64 analysis!

- Automated evaluation of in-memory code

What is New for Developers?

- Extensive API documentation

- Plugins can directly call other plugins

- Plugins are versioned

- Much easier to use custom data structures and symbols
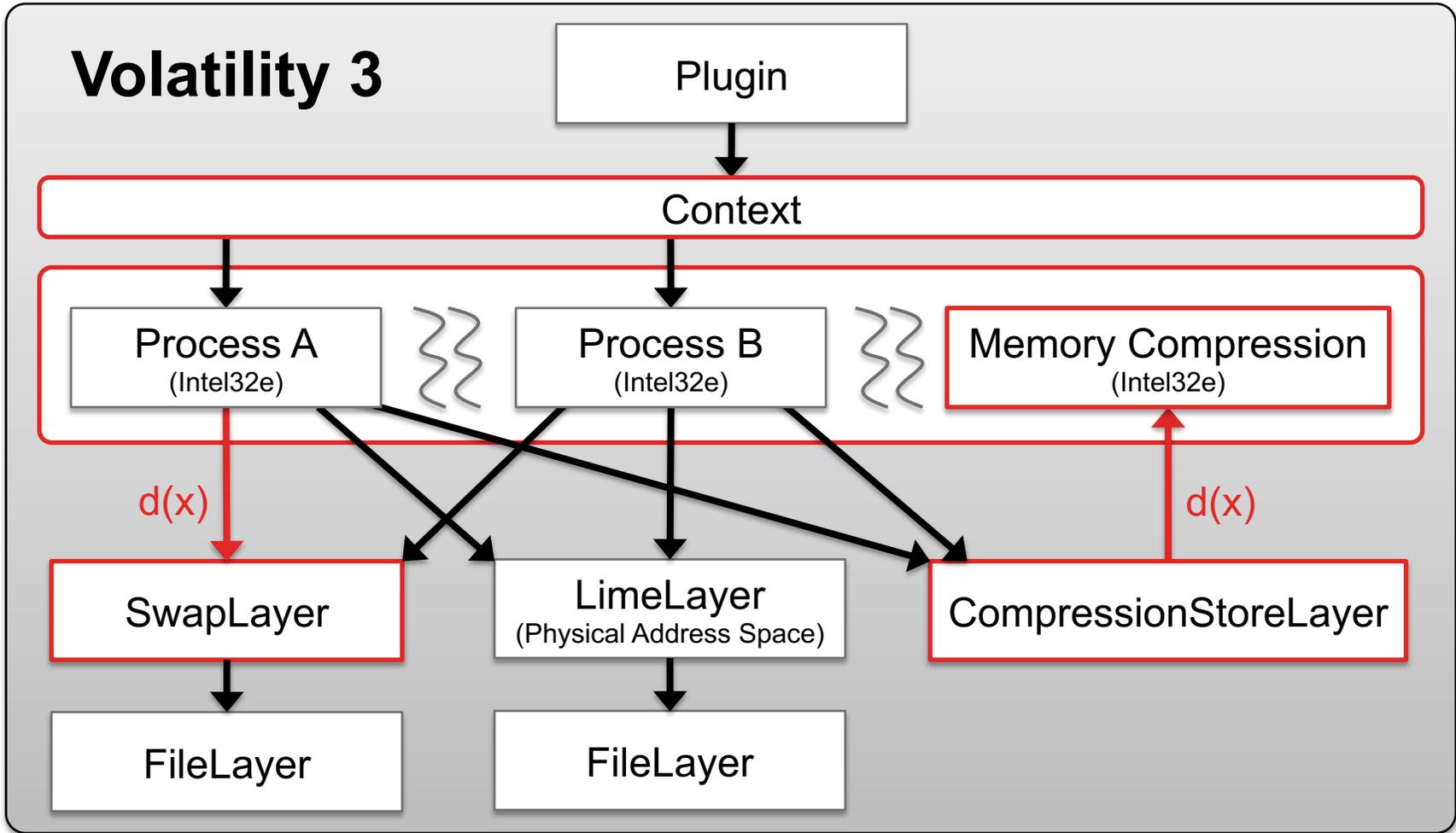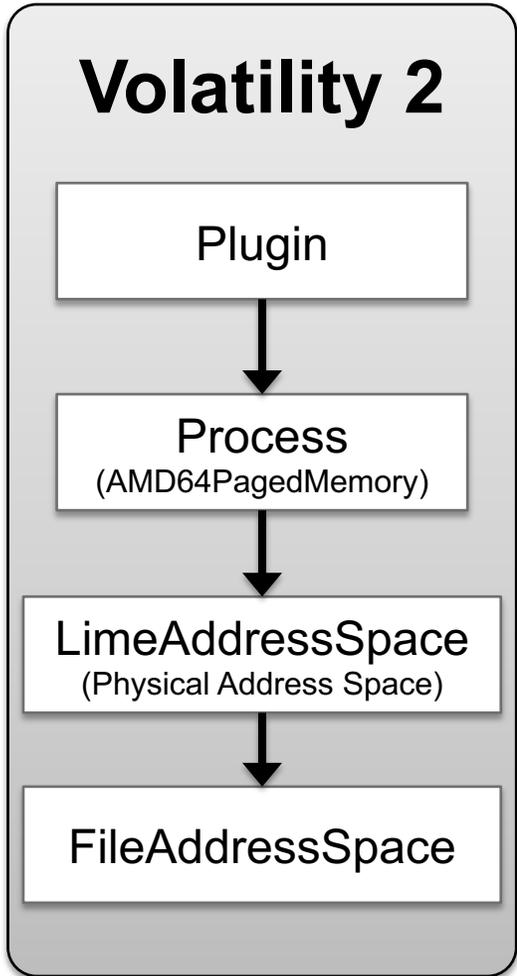
PLUGIN TIMINGS

Rekall   Volatility 2   Volatility 3
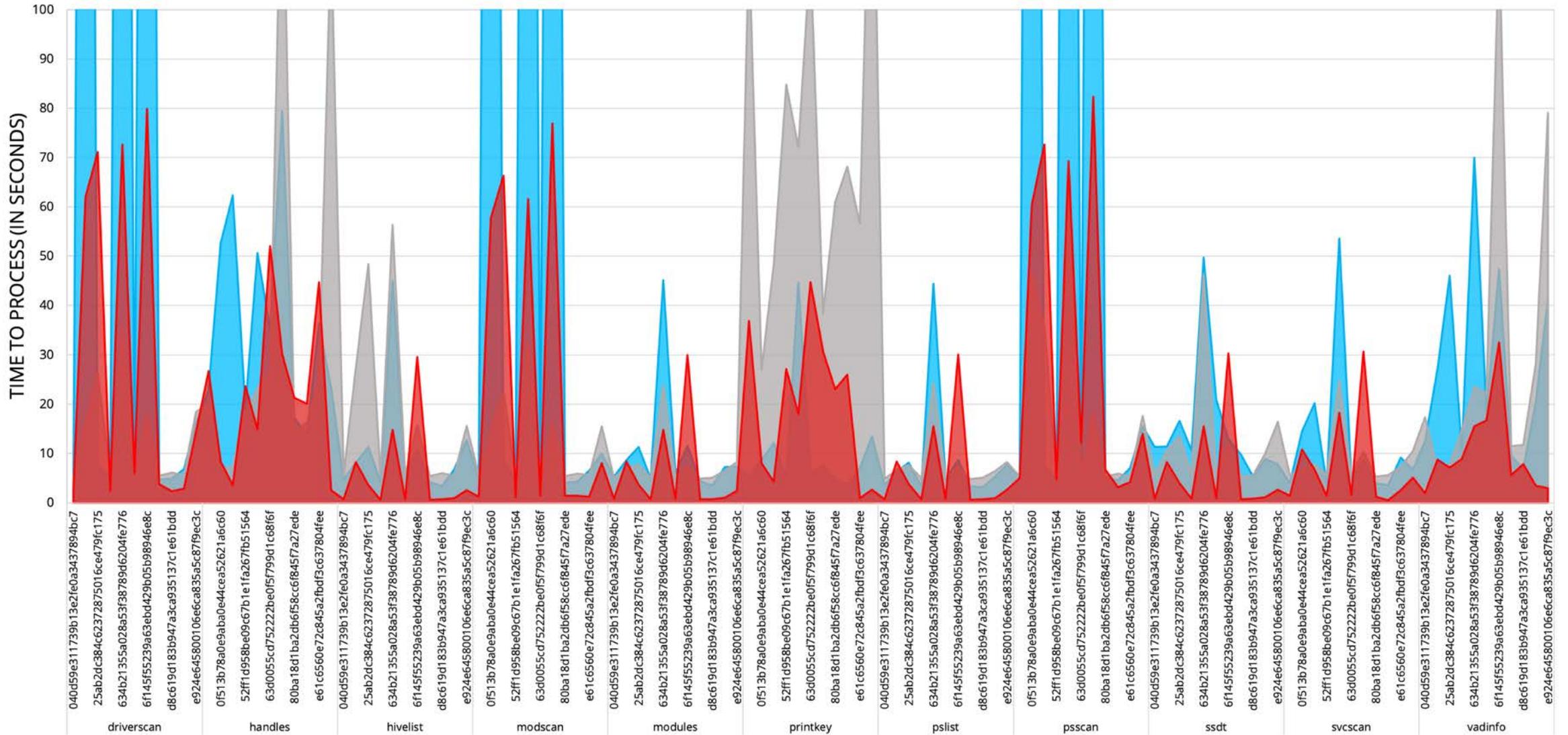
```
$ time python3 vol.py -f turlanew.raw -r pretty windows.pslist
Volatility 3 Framework 1.0.0-beta.1
```

	PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64		CreateTime		ExitTime
*	4	0	System	0xfa8030eb5840	94	413	N/A	False		2018-07-31 17:39:31.000000		N/A
*	276	4	smss.exe	0xfa80325e9700	2	30	N/A	False		2018-07-31 17:39:31.000000		N/A
*	372	360	csrss.exe	0xfa8033185710	10	553	0	False		2018-07-31 17:39:32.000000		N/A
*	428	360	wininit.exe	0xfa803351e060	3	79	0	False		2018-07-31 17:39:32.000000		N/A
*	512	428	services.exe	0xfa80335a6060	9	239	0	False		2018-07-31 17:39:32.000000		N/A
*	544	428	lsass.exe	0xfa80335d3710	8	585	0	False		2018-07-31 17:39:32.000000		N/A
*	552	428	lsm.exe	0xfa80335d42d0	11	149	0	False		2018-07-31 17:39:32.000000		N/A
*	648	512	svchost.exe	0xfa8033821a70	13	372	0	False		2018-07-31 17:39:32.000000		N/A
*	708	512	vmacthlp.exe	0xfa8033856910	3	56	0	False		2018-07-31 17:39:32.000000		N/A
*	1004	812	audiodg.exe	0xfa8033951630	6	136	0	False		2018-07-31 17:39:32.000000		N/A
*	1632	2568	notepad.exe	0xfa803113e7d0	2	75	2	False		2019-04-11 19:35:28.000000		N/A
*	2920	2568	wordpad.exe	0xfa8031a66060	5	132	2	False		2019-04-11 19:35:31.000000		N/A
*	3384	1448	cmd.exe	0xfa80312b9480	0	-	0	False		2019-04-11 19:35:36.000000		2019-04-11 19:35:36.000000
*	3960	372	conhost.exe	0xfa803104cb30	0	-	0	False		2019-04-11 19:35:36.000000		2019-04-11 19:35:36.000000
*	3956	3384	ipconfig.exe	0xfa80312c9630	0	-	0	False		2019-04-11 19:35:36.000000		2019-04-11 19:35:36.000000

```
real    0m1.116s
user    0m0.760s
sys     0m0.076s
```

```
$ python3 volshell.py -f turlanew.raw
Volshell (Volatility 3 Framework) 1.0.0-beta.1
Readline imported successfully  Building linux caches

    Call help() to see available functions

    Volshell mode: Generic
    Current Layer: primary2


(primary2) >>> hh()


Methods:
* dt, display_type
    Display Type describes the members of a particular object in alphabetical order
* db, display_bytes
    Displays byte values and ASCII characters
* dw, display_words
    Displays word values (2 bytes) and corresponding ASCII characters
* dd, display_doublewords
    Displays double-word values (4 bytes) and corresponding ASCII characters
* dq, display_quadwords
    Displays quad-word values (8 bytes) and corresponding ASCII characters
* dis, disassemble
    Disassembles a number of instructions from the code at offset
* cl, change_layer
    Changes the current default layer
* dpo, display_plugin_output
    Displays the output for a particular plugin (with keyword arguments)
* gt, generate_treegrid
    Generates a TreeGrid based on a specific plugin passing in kwarg configuration values
* rt, render_treegrid
    Renders a treegrid as produced by generate_treegrid
* ds, display_symbols
    Prints an alphabetical list of symbols for a symbol table
```

Supporting Modern and Advanced Analytics

- Automating (where possible) operating system and application support

- Automating analysis decisions beyond simply presenting data structures and raw disassembly listings

- Automating analysis of multiple samples at once

Automated Kernel Module Analysis – NDIS & Netfilter [5, 6]

With the hooks installed, whenever the network adapter driver attempts to register to NDIS, or whenever there is an attempt to install NDIS intermediate driver or NDIS filter driver, the hook handlers will register Snake's own *MiniportXxx* functions with the NDIS library.

With its own miniport handler functions, it can send/receive data by using a private TCP/IP stack, bypassing all firewall hooks, and making its open ports invisible to scanners.

```
1007        magic_packet_hook_options.hook = (void *)magic_packet_hook;
1008        magic_packet_hook_options.hooknum = 0;
1009        magic_packet_hook_options.pf = PF_INET;
1010        magic_packet_hook_options.priority = NF_IP_PRI_FIRST;
```

Automated Version Analysis – TrueCrypt vs VeraCrypt [7, 8]

Tuesday, January 14, 2014

TrueCrypt Master Key Extraction And Volume Identification

- 🪟 *Windows*:
  - Installer: <u>VeraCrypt Setup 1.24.exe</u> (34.2 MB) <u>(PGP Signature)</u>
  - Portable version: <u>VeraCrypt Portable 1.24.exe</u> (34 MB) <u>(PGP Signature)</u>
  - Debugging Symbols: <u>VeraCrypt_1.24_Windows_Symbols.zip</u> (9.45 MB) <u>(PGP Signature)</u>

Automatic Symbol Inclusion

```
$ python3 vol.py -f sample.vmem windows.ssdt
Volatility 3 Framework 1.0.0-beta.1
Index   Address         Module      Symbol
0       0xf800034dea50  ntoskrnl    NtMapUserPhysicalPagesScatter
1       0xf800033c40a0  ntoskrnl    NtWaitForSingleObject
2       0xf800030c68a0  ntoskrnl    NtCallbackReturn
3       0xf800033b7210  ntoskrnl    NtReadFile
<snip>


$ python3 vol.py -f sample.vmem windows.callbacks
Volatility 3 Framework 1.0.0-beta.1
Type                            Callback        Module      Symbol                  Detail
PspLoadImageNotifyRoutine       0xf800033ea520  ntoskrnl    EtwpTraceLoadImage      N/A
PspCreateProcessNotifyRoutine   0xf80003099590  ntoskrnl    ViCreateProcessCallback N/A
<snip>
```

Automated Emulation of In-Memory Hooks [9]

```
Hook mode: Usermode
Hook type: Inline/Trampoline
Process: 3068 (iexplore.exe)
Victim module: ntdll.dll (0x77640000 - 0x7777c000)
Function: ntdll.dll!LdrLoadDll at 0x776a22b8
Hook address: 0x74c601f8
Hooking module: <unknown>
Disassembly(0):
0x776a22b8 e93bdf5bfd        JMP 0x74c601f8
<cut>
Disassembly(1):
0x74c601f8 e9c3daabeb        JMP 0x6071dcc0
<cut>
```

```
3068 iexplore.exe      ntdll.dll!LdrLoadDll at 0x776a22b8
PAGE_EXECUTE_READWRITE <Non-File Backed Region: 0x74c60000
0x74c6afff>
PAGE_EXECUTE_WRITECOPY \Device\HarddiskVolume1\
Program Files\AVG\Antivirus\snxhk.dll (2)
PAGE_EXECUTE_READWRITE <Non-File Backed Region: 0x74c60000
0x74c6afff> (46)
PAGE_EXECUTE_WRITECOPY \Device\HarddiskVolume1\
Program Files\AVG\Antivirus\aswhookx.dll (2)
PAGE_EXECUTE_READWRITE <Non-File Backed Region: 0x6f670000
0x6f67ffff> (4)
PAGE_EXECUTE_WRITECOPY \Device\HarddiskVolume1\Windows\
System32\ntdll.dll (2)
```

Automatically Analyzing Multiple Samples

Volatility 2

1. Run kdbgscan (or imageinfo)
2. <wait>
3. Set --profile
4. Run plugin

Volatility 3

1. Run plugin

```
$ for sample in $(ls samples/*.mem); do python3 vol.py -f $sample windows.pslist | grep System; done
4    0    System    0x8a13f280    120    -    N/A    False    2019-09-30 03:16:14.000000    N/A
4    0    System    0x86569040    109    -    N/A    False    2019-01-05 02:29:56.000000    N/A
4    0    System    0x8954f040    121    -    N/A    False    2019-07-15 21:29:58.000000    N/A
4    0    System    0x85f6c040    118    -    N/A    False    2019-08-02 00:39:33.000000    N/A
```

Volatility 3.0 Licensing - Volatility Software License 1.0

- The Volatility Foundation will no longer release code under the GPL
- Short, simple, plain language copyleft license
  - No philosophical agendas or drama!
- Volatility Software License 1.0
  - Grants all rights of copyright ("Freedom Zero")
  - Requires changes to the software are shared
  - Covers content and data
- Legal Advisor: Heather Meeker
- License Inspiration
  - API Copyleft License (Kyle E. Mitchell)
  - Nmap Public Source License

Looking Forward

Volatility 3

Feature parity/new & unique capabilities

August 2020 Volatility 3.0 Official Release

August 2021 – future Volatility development & support ONLY for 3.x

2020

2021

Plugin & operating system updates

Volatility 2.x

August 2021 2.x: Development & support for 2.x Ends
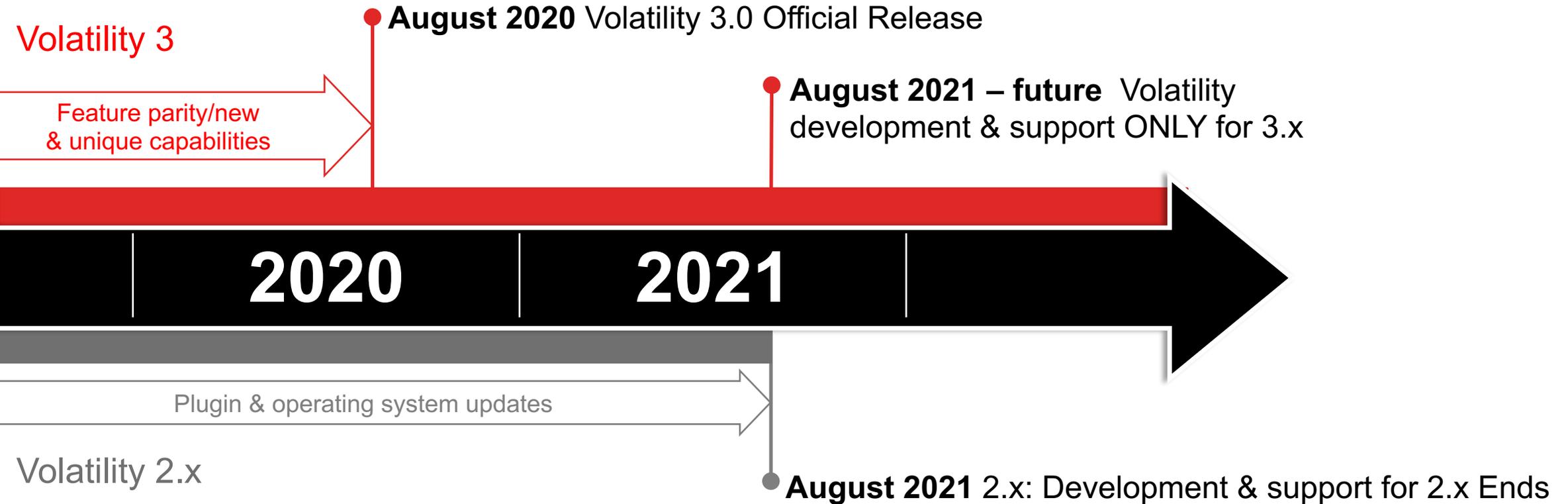
Start Using It and Get Involved!

- https://www.github.com/volatilityfoundation/volatility3

- https://volatility3.rtfd.io/

- https://www.volatilityfoundation.org/slack

- https://lists.volatilityfoundation.org/pipermail/vol-users/

References

[1] https://www.blackhat.com/presentations/bh-dc-07/Walters/Paper/bh-dc-07-Walters-WP.pdf
[2] https://www.volatilityfoundation.org/20
[3] https://docs.microsoft.com/en-us/windows/deployment/update/waas-quick-start
[4] https://www.kernelnewbies.org
[5] https://artemonsecurity.com/snake_whitepaper.pdf
[6] https://github.com/f0rb1dd3n/Reptile/
[7] https://volatility-labs.blogspot.com/2014/01/truecrypt-master-key-extraction-and.html
[8] https://www.veracrypt.fr/en/Downloads.html
[9] http://dfrws.org/conferences/dfrws-usa-2019/sessions/hooktracer-system-automated-and-accessible-api-hooks-analysis