Rekall Memory Forensics

Michael Cohen
mic@google.com

SANS DFIR Summit - Prague 2014

Rekall - our first user!

What is Rekall?

- An advanced memory analysis solution.
    - Historically a fork of the Volatility memory analysis framework
        - Most code rewritten/updated.
    - Fully open source and GPL - all commits are public.
    - Focus on:
        - code quality - public code reviews.
        - performance.
        - ease of use as a library - Integrated into other tools.

Rekall project goals

- To make memory analysis:
    - Accessible
        - GUI, Data exports, Library API.
    - Fast and efficient
        - Reduce scanning/guessing algorithms.
    - Accurate - works everywhere!
        - Profile repository gives accurate, specific information.
    - Powerful
        - Research novel analysis techniques.

How is it different from X?

- Other memory forensic frameworks rely on guessing global symbols through signature scanning.
- Rekall uses a different design philosophy:
    - Exact symbol information for the analyzed system
        - e.g. Fetch from Microsoft Symbol Server.
    - Store profiles in a public profile repository
        - Rekall fetches the required profile at runtime.
            - We have an index of kernel profiles.
        - We have over 200 different kernels in the public repository.

How is it different from X?

- This means we do not need to guess or try to deduce global symbols.
    - This makes Rekall much faster, more efficient and more accurate.
    - For example, Rekall does not use the Kernel Debugger Block
        - This can easily be overwritten by malware. Or newer versions of Windows.
    - This is similar to the way the kernel debugger works - much more reliable.

How is it different from X?

- Rekall distributes and supports a complete memory acquisition solution.
  - We have synergy between acquisition and analysis.
  - Support all major operating systems:
    - Windows - Winpmem tool.
    - Linux - pmem tool + LMAP tool (No need to precompile on target system).
    - OSX - OSXPmem tool (supports 10.9.4+).
  - Rekall acquisition tools allow for live system analysis (Triaging etc).

Basic research that you can use!

DFRWS 2013: Anti-Forensic Resilient Memory Acquisition, Johannes Stuettgen and Michael Cohen.

- Incorporated into WinPmem, LMAP and OSXPmem
  - OSXPmem is only memory acquisition software that works reliably since OSX 10.9.3 due to tightening of OSX API.
  - This technology enables implementation of LMAP

DFRWS 2014EU: Robust Linux Memory Acquisition with Minimal Target Impact, Johannes Stuettgen and Michael Cohen

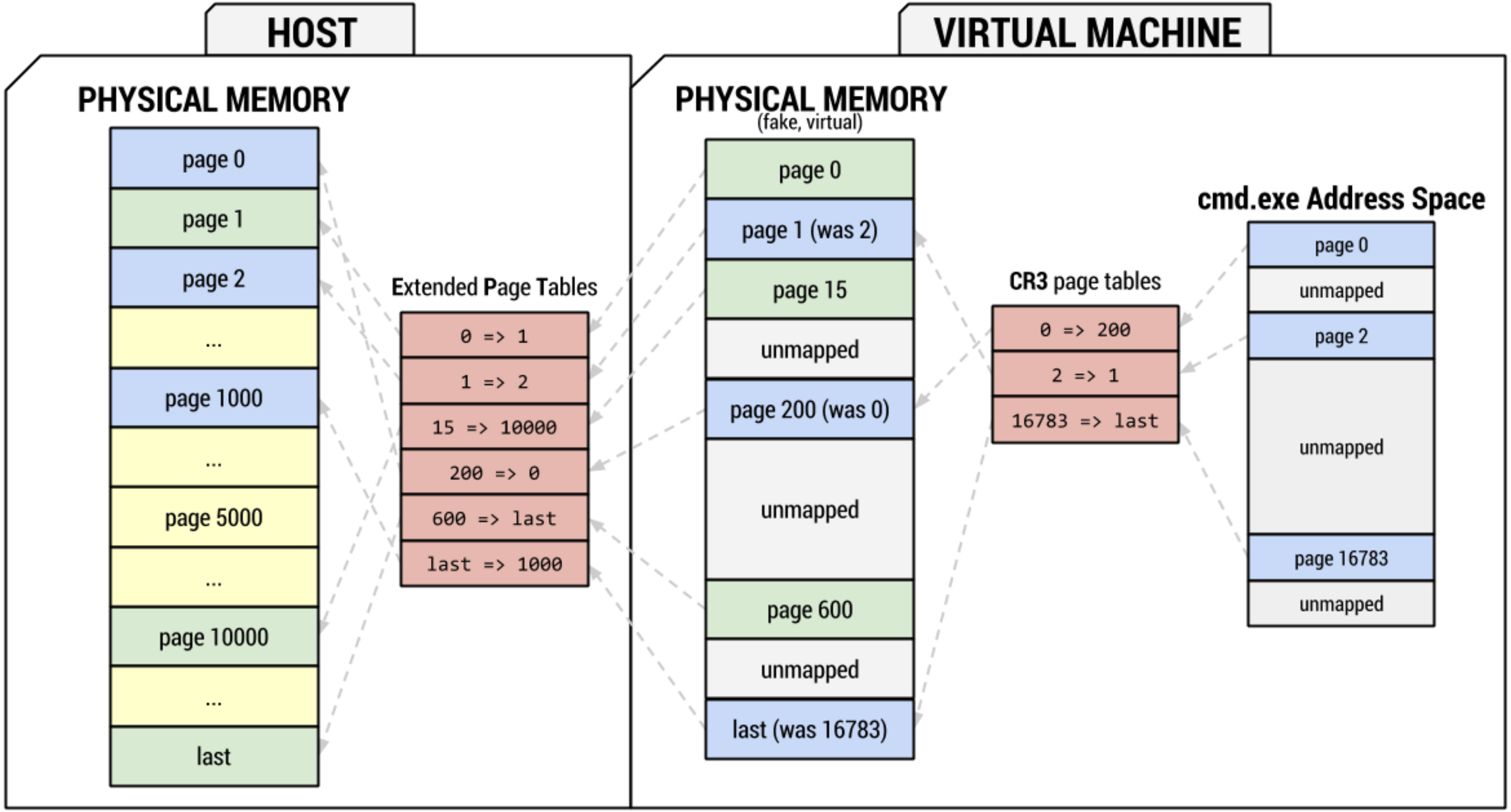
- Only Linux memory acquisition software that works, precompiled, on any kernel out of the box. Regardless of kernel version or configuration!
- Really big deal for incident response!

Virtual Machine introspection

By Jordi Sanchez

- Virtual Machines used everywhere!
  - Lots of different virtualization technologies (vmware, KVM, Virtual Box, MS Virtual Server etc).
  - For efficiency it turns out they all use the hardware to map memory into the guest OS.
- Rekall can analyze memory of guest OS from the host memory image!
  - Rekall also supports VM nesting.

http://www.rekall-forensic.com/posts/2014-10-03-vms.html



HOST

PHYSICAL MEMORY

- page 0
- page 1
- page 2
- ...
- page 1000
- ...
- page 5000
- ...
- page 10000
- ...
- last

Extended Page Tables

```
0 => 1
1 => 2
15 => 10000
200 => 0
600 => last
last => 1000
```

VIRTUAL MACHINE

PHYSICAL MEMORY
(fake, virtual)

- page 0
- page 1 (was 2)
- page 15
- unmapped
- page 200 (was 0)
- unmapped
- page 600
- unmapped
- last (was 16783)

CR3 page tables

```
0 => 200
2 => 1
16783 => last
```

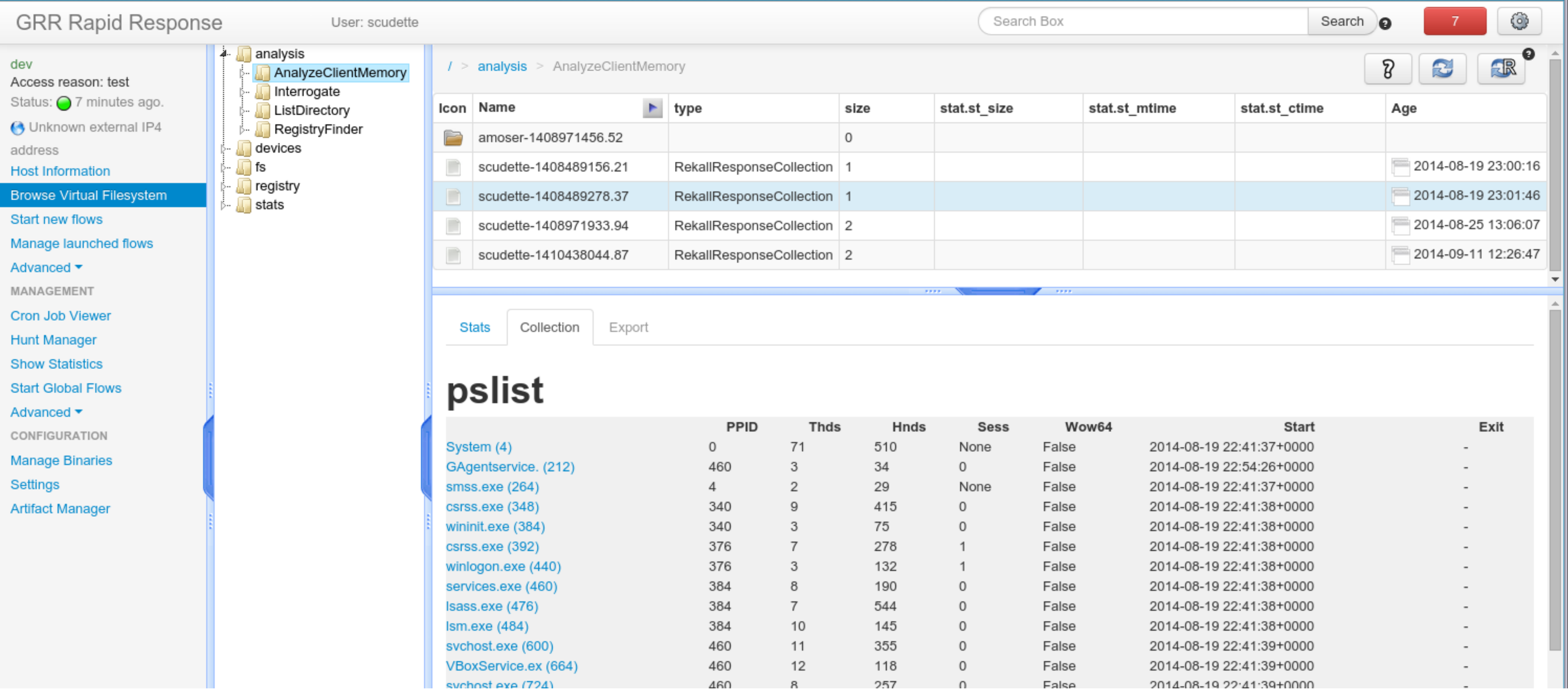
cmd.exe Address Space

- page 0
- unmapped
- page 2
- unmapped
- page 16783
- unmapped

Page used by cmd.exe in the VM	🟦
Physical page of the VM	🟩
Host physical page, valid but not used	🟨
Unmapped page	⬜

Rekall as a library

- Rekall was specifically designed to be included into a large project as a library.
  - Non-global objects - thread safe.
  - Progress reporting/suspension points for event driven loops. (Particularly useful for GUI apps).
- Rekall is fully integrated into the GRR project
  - Can remotely analyze systems.
  - Data shipped back to the GRR server as JSON objects.



Icon	Name	type	size	stat.st_size	stat.st_mtime	stat.st_ctime	Age
	amoser-1408971456.52		0				
	scudette-1408489156.21	RekallResponseCollection	1				2014-08-19 23:00:16
	scudette-1408489278.37	RekallResponseCollection	1				2014-08-19 23:01:46
	scudette-1408971933.94	RekallResponseCollection	2				2014-08-25 13:06:07
	scudette-1410438044.87	RekallResponseCollection	2				2014-09-11 12:26:47

pslist

	PPID	Thds	Hnds	Sess	Wow64	Start	Exit
System (4)	0	71	510	None	False	2014-08-19 22:41:37+0000	-
GAgentservice. (212)	460	3	34	0	False	2014-08-19 22:54:26+0000	-
smss.exe (264)	4	2	29	None	False	2014-08-19 22:41:37+0000	-
csrss.exe (348)	340	9	415	0	False	2014-08-19 22:41:38+0000	-
wininit.exe (384)	340	3	75	0	False	2014-08-19 22:41:38+0000	-
csrss.exe (392)	376	7	278	1	False	2014-08-19 22:41:38+0000	-
winlogon.exe (440)	376	3	132	1	False	2014-08-19 22:41:38+0000	-
services.exe (460)	384	8	190	0	False	2014-08-19 22:41:38+0000	-
lsass.exe (476)	384	7	544	0	False	2014-08-19 22:41:38+0000	-
lsm.exe (484)	384	10	145	0	False	2014-08-19 22:41:38+0000	-
svchost.exe (600)	460	11	355	0	False	2014-08-19 22:41:39+0000	-
VBoxService.ex (664)	460	12	118	0	False	2014-08-19 22:41:39+0000	-
svchost.exe (724)	460	8	257	0	False	2014-08-19 22:41:39+0000	-

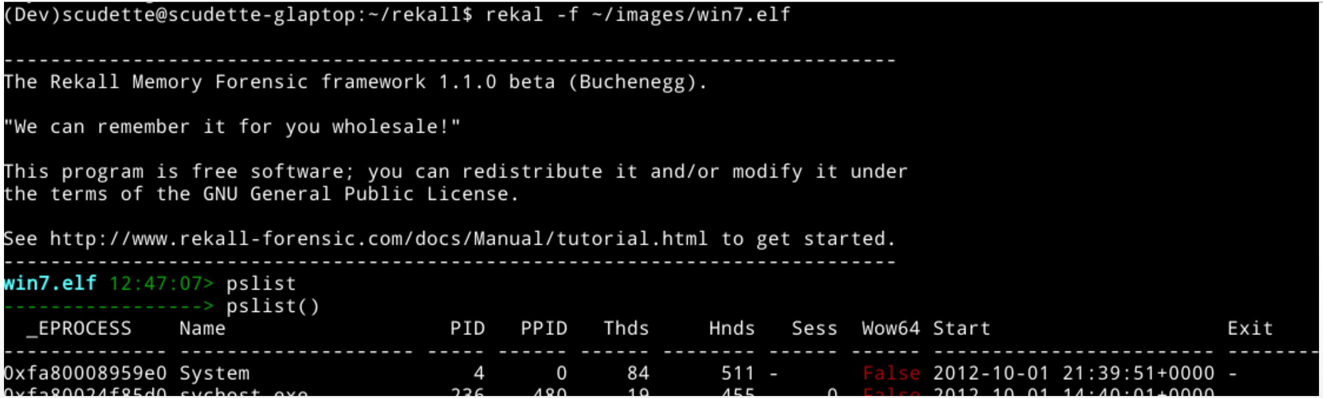
Rekall is integrated in GRR: Remote memory forensics at scale.

Google | Rekall

The Rekall User interfaces

- Rekall has 3 user interfaces:
  - Command line - single shot, run and exit.
  - Interactive console - IPython based (text only).
  - Webconsole - most powerful.
- The same plugin works in all environments!
  - Writing a plugin is easy
    - You do not need to think about output formatting - the framework does it all!

Text Interactive Console



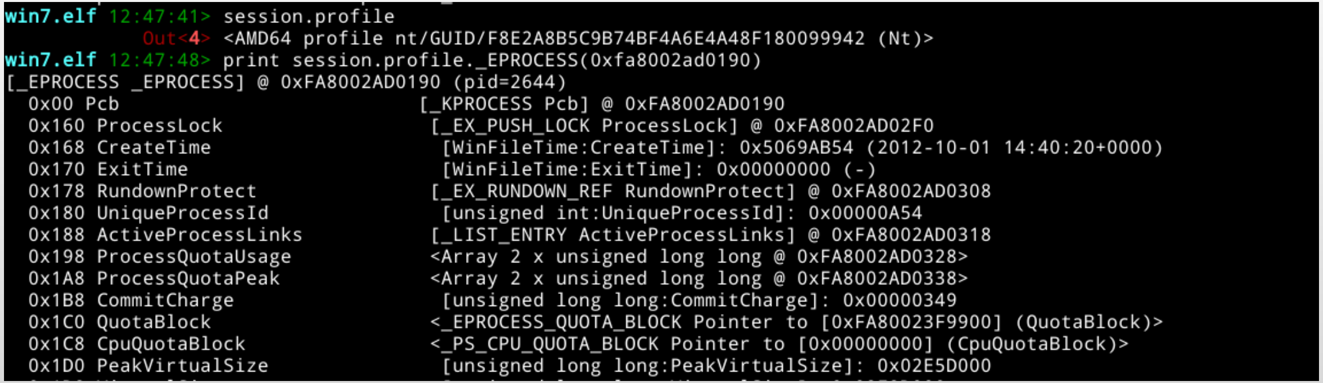
```
(Dev)scudette@scudette-glaptop:~/rekall$ rekal -f ~/images/win7.elf

----------------------------------------------------------------------------
The Rekall Memory Forensic framework 1.1.0 beta (Buchenegg).

"We can remember it for you wholesale!"

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License.

See http://www.rekall-forensic.com/docs/Manual/tutorial.html to get started.
----------------------------------------------------------------------------
win7.elf 12:47:07> pslist
----------------> pslist()
  _EPROCESS      Name                    PID    PPID    Thds      Hnds    Sess   Wow64  Start                        Exit
-------------- -------------------- ----- ------ ------ -------- ------ ------ ------------------------ --------
0xfa80008959e0 System                      4       0      84       511 -        False  2012-10-01 21:39:51+0000 -
0xfa80024f85d0 svchost.exe               236     480      19       455      0  False  2012-10-01 14:40:01+0000
```



```
win7.elf 12:47:41> session.profile
           Out<4> <AMD64 profile nt/GUID/F8E2A8B5C9B74BF4A6E4A48F180099942 (Nt)>
win7.elf 12:47:48> print session.profile._EPROCESS(0xfa8002ad0190)
[_EPROCESS _EPROCESS] @ 0xFA8002AD0190 (pid=2644)
  0x00 Pcb                          [_KPROCESS Pcb] @ 0xFA8002AD0190
  0x160 ProcessLock                  [_EX_PUSH_LOCK ProcessLock] @ 0xFA8002AD02F0
  0x168 CreateTime                    [WinFileTime:CreateTime]: 0x5069AB54 (2012-10-01 14:40:20+0000)
  0x170 ExitTime                      [WinFileTime:ExitTime]: 0x00000000 (-)
  0x178 RundownProtect               [_EX_RUNDOWN_REF RundownProtect] @ 0xFA8002AD0308
  0x180 UniqueProcessId               [unsigned int:UniqueProcessId]: 0x00000A54
  0x188 ActiveProcessLinks           [_LIST_ENTRY ActiveProcessLinks] @ 0xFA8002AD0318
  0x198 ProcessQuotaUsage            <Array 2 x unsigned long long @ 0xFA8002AD0328>
  0x1A8 ProcessQuotaPeak             <Array 2 x unsigned long long @ 0xFA8002AD0338>
  0x1B8 CommitCharge                  [unsigned long long:CommitCharge]: 0x00000349
  0x1C0 QuotaBlock                   <_EPROCESS_QUOTA_BLOCK Pointer to [0xFA80023F9900] (QuotaBlock)>
  0x1C8 CpuQuotaBlock                <_PS_CPU_QUOTA_BLOCK Pointer to [0x00000000] (CpuQuotaBlock)>
  0x1D0 PeakVirtualSize               [unsigned long long:PeakVirtualSize]: 0x02E5D000
```

- Fast and efficient
- Great for interactively exploring data types.
- Great for scripting complex analysis (no need to write plugin).

The Rekall Web Console interface

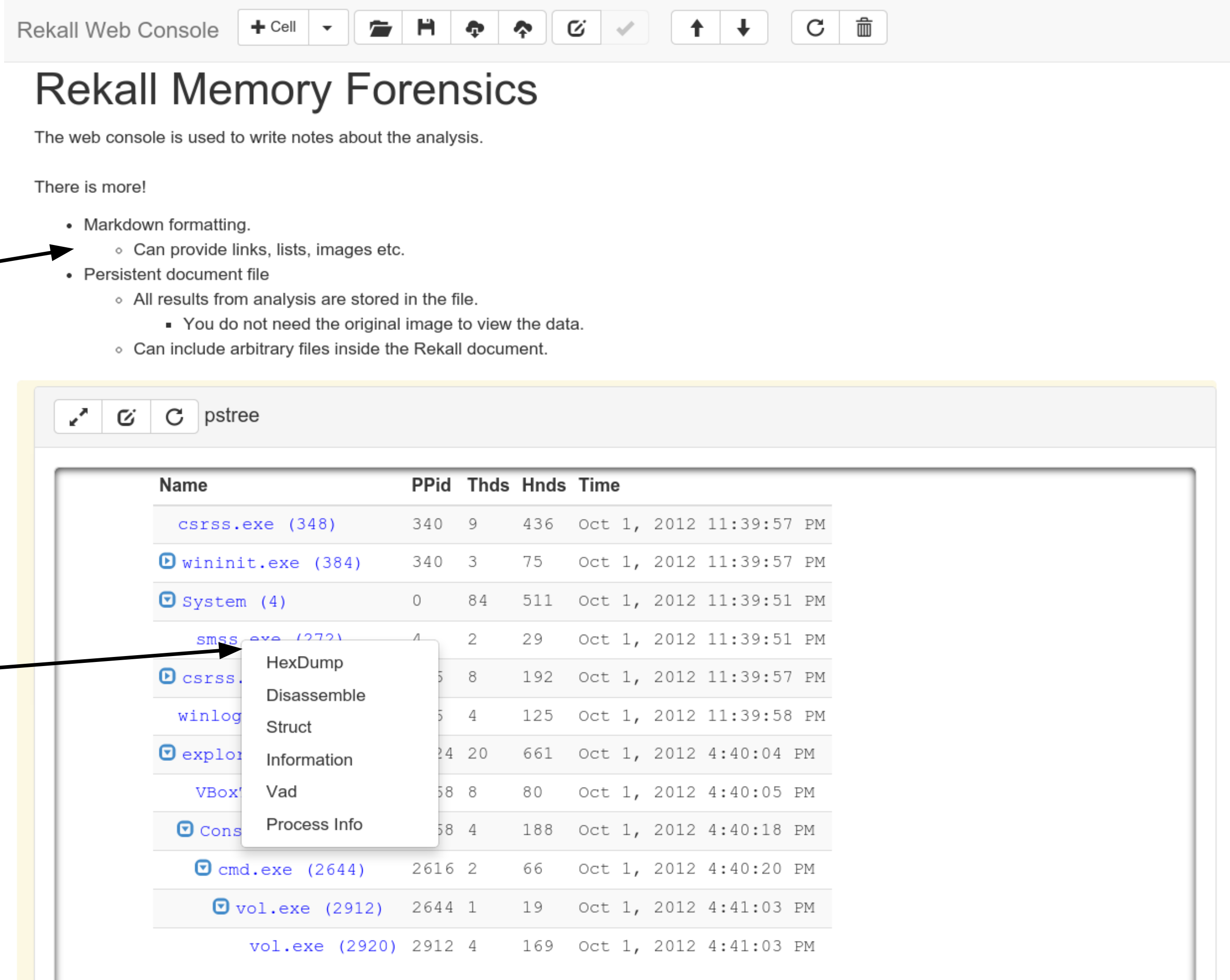
By Misha Bushkov

A GUI is not just a pretty thing!

- The Rekall Web console GUI helps drive analysis by:
    - Allowing the user to annotate her analysis
        - Notebook interface creates a mini "report" format.
        - Hides/Compacts long analysis to improve document flow.
    - Persistent file storage allows results to be managed and shared (based on Zip files).
    - Rekall files contain plugin output in JSON format
        - Machine readable - can be exported

Markdown Comments intersperse analysis.

Context Sensitive Actions



Rekall Memory Forensics

The web console is used to write notes about the analysis.

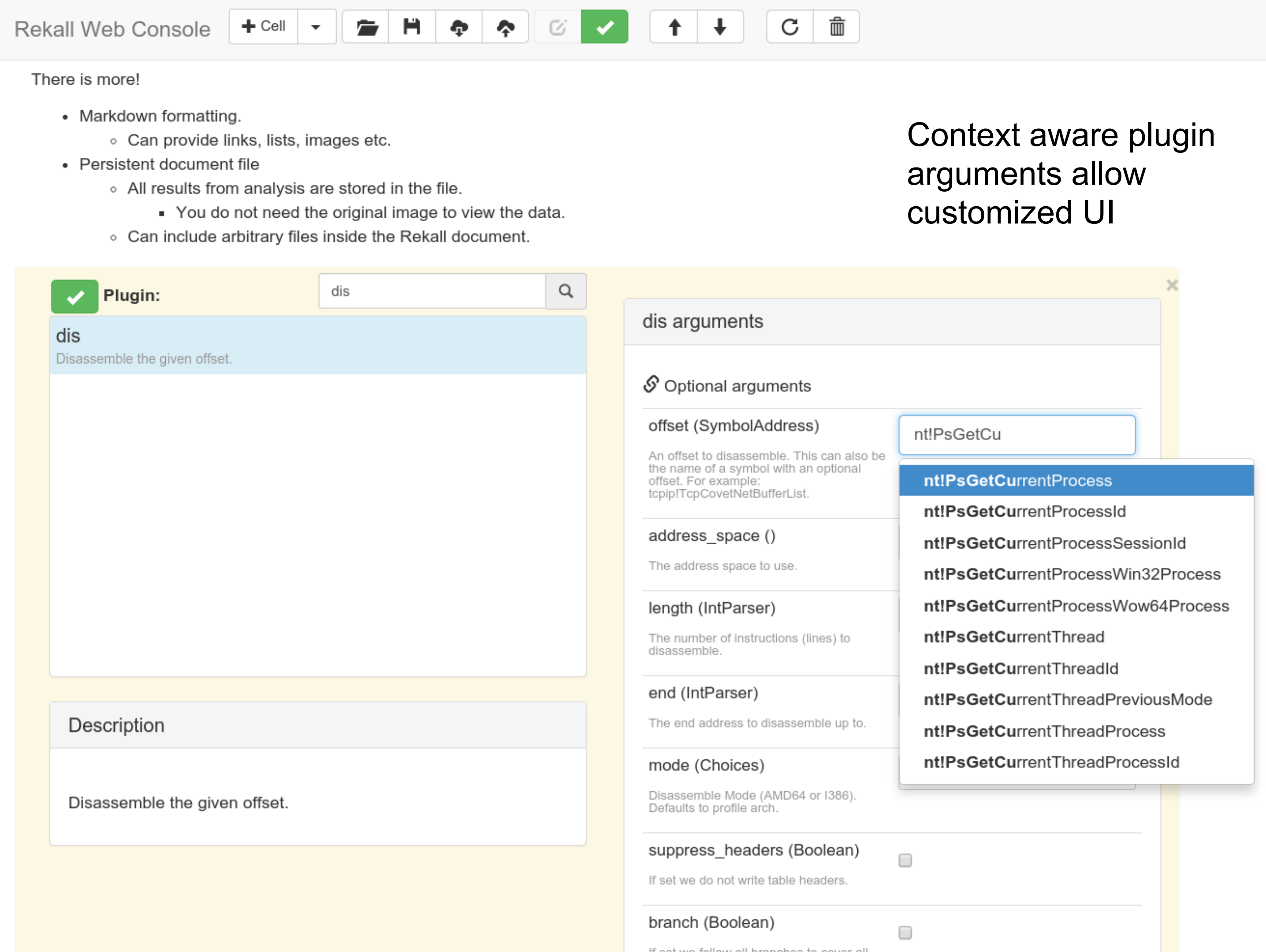
There is more!

- Markdown formatting.
  - Can provide links, lists, images etc.
- Persistent document file
  - All results from analysis are stored in the file.
    - You do not need the original image to view the data.
  - Can include arbitrary files inside the Rekall document.

Name	PPid	Thds	Hnds	Time
csrss.exe (348)	340	9	436	Oct 1, 2012 11:39:57 PM
wininit.exe (384)	340	3	75	Oct 1, 2012 11:39:57 PM
System (4)	0	84	511	Oct 1, 2012 11:39:51 PM
smss.exe (272)	4	2	29	Oct 1, 2012 11:39:51 PM
csrss.	5	8	192	Oct 1, 2012 11:39:57 PM
winlog	5	4	125	Oct 1, 2012 11:39:58 PM
explor	24	20	661	Oct 1, 2012 4:40:04 PM
VBox	58	8	80	Oct 1, 2012 4:40:05 PM
Cons	58	4	188	Oct 1, 2012 4:40:18 PM
cmd.exe (2644)	2616	2	66	Oct 1, 2012 4:40:20 PM
vol.exe (2912)	2644	1	19	Oct 1, 2012 4:41:03 PM
vol.exe (2920)	2912	4	169	Oct 1, 2012 4:41:03 PM

Can Launch any Rekall plugins from UI.

Context aware plugin arguments allow customized UI



There is more!

- Markdown formatting.
  - Can provide links, lists, images etc.
- Persistent document file
  - All results from analysis are stored in the file.
    - You do not need the original image to view the data.
  - Can include arbitrary files inside the Rekall document.

dis
Disassemble the given offset.

Description

Disassemble the given offset.

dis arguments

Optional arguments

offset (SymbolAddress): nt!PsGetCu
An offset to disassemble. This can also be the name of a symbol with an optional offset. For example: tcpip!TcpCovetNetBufferList.

- nt!PsGetCurrentProcess
- nt!PsGetCurrentProcessId
- nt!PsGetCurrentProcessSessionId
- nt!PsGetCurrentProcessWin32Process
- nt!PsGetCurrentProcessWow64Process
- nt!PsGetCurrentThread
- nt!PsGetCurrentThreadId
- nt!PsGetCurrentThreadPreviousMode
- nt!PsGetCurrentThreadProcess
- nt!PsGetCurrentThreadProcessId

address_space ()
The address space to use.

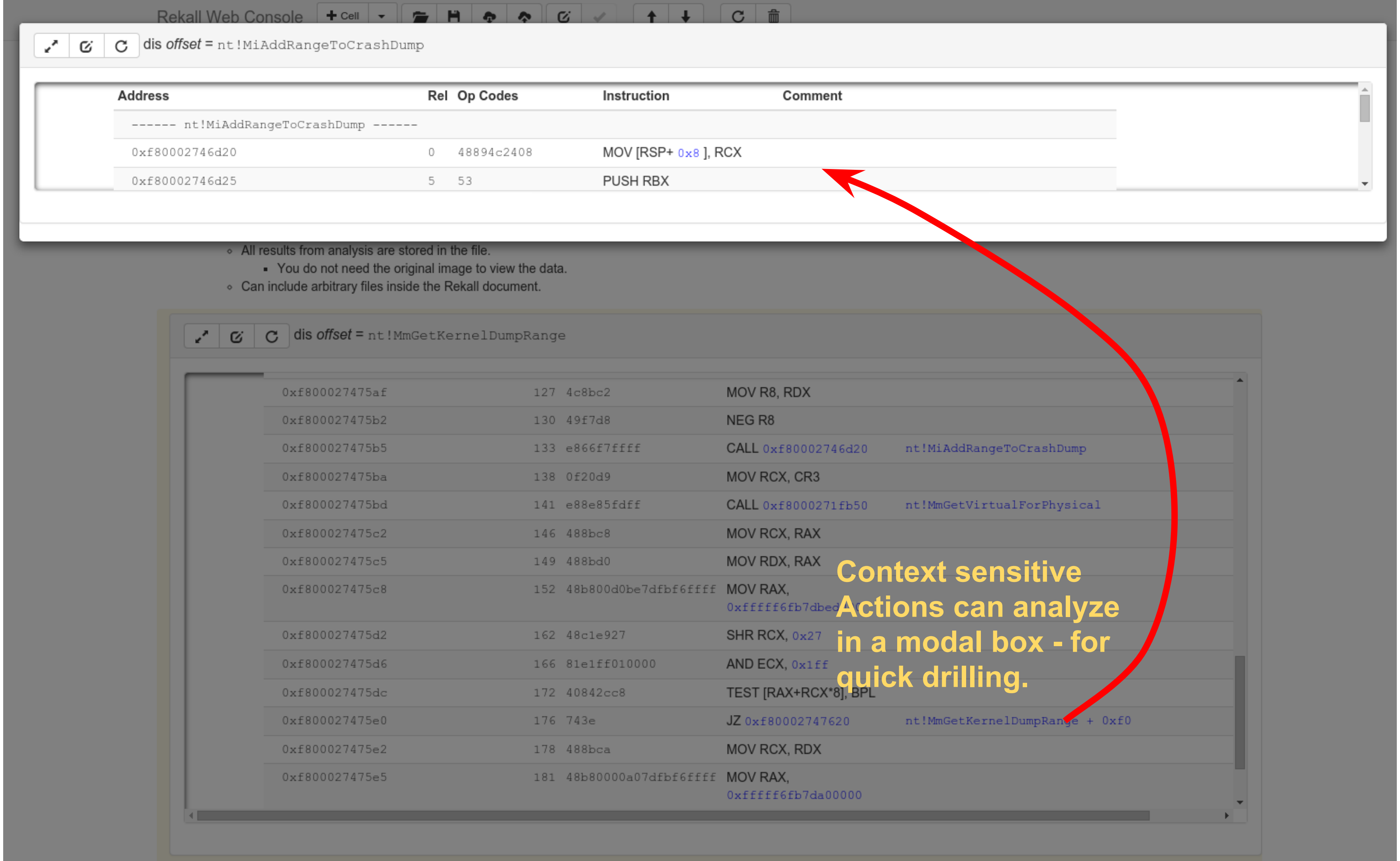
length (IntParser)
The number of instructions (lines) to disassemble.

end (IntParser)
The end address to disassemble up to.

mode (Choices)
Disassemble Mode (AMD64 or I386). Defaults to profile arch.

suppress_headers (Boolean) □
If set we do not write table headers.

branch (Boolean) □
If set we follow all branches to cover all



dis *offset* = nt!MiAddRangeToCrashDump

Address	Rel	Op Codes	Instruction	Comment
------ nt!MiAddRangeToCrashDump ------				
0xf80002746d20	0	48894c2408	MOV [RSP+ 0x8], RCX	
0xf80002746d25	5	53	PUSH RBX	

- All results from analysis are stored in the file.
  - You do not need the original image to view the data.
- Can include arbitrary files inside the Rekall document.

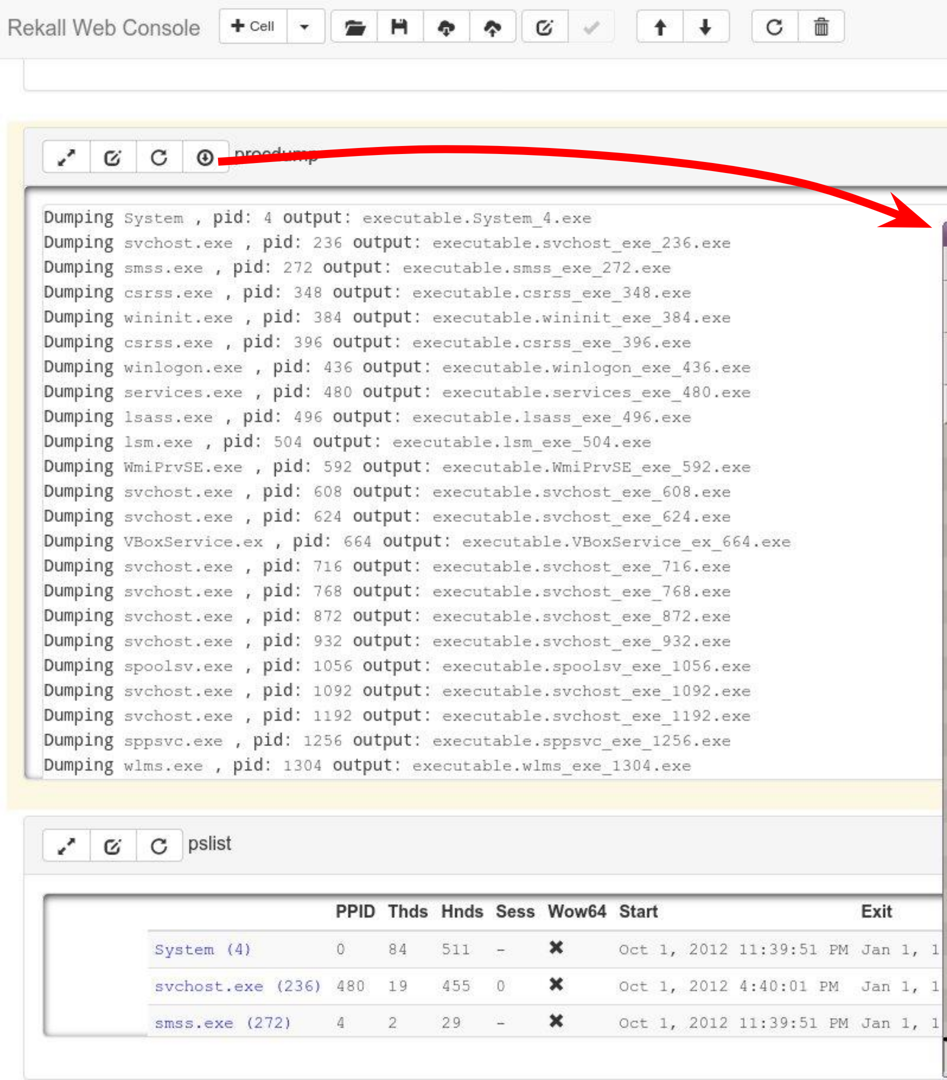
dis *offset* = nt!MmGetKernelDumpRange

```
0xf800027475af    127  4c8bc2                MOV R8, RDX
0xf800027475b2    130  49f7d8                NEG R8
0xf800027475b5    133  e866f7ffff            CALL 0xf80002746d20      nt!MiAddRangeToCrashDump
0xf800027475ba    138  0f20d9                MOV RCX, CR3
0xf800027475bd    141  e88e85fdff            CALL 0xf8000271fb50      nt!MmGetVirtualForPhysical
0xf800027475c2    146  488bc8                MOV RCX, RAX
0xf800027475c5    149  488bd0                MOV RDX, RAX
0xf800027475c8    152  48b800d0be7dfbf6ffff  MOV RAX,
                                             0xfffff6fb7dbed
0xf800027475d2    162  48c1e927              SHR RCX, 0x27
0xf800027475d6    166  81e1ff010000          AND ECX, 0x1ff
0xf800027475dc    172  40842cc8              TEST [RAX+RCX*8], BPL
0xf800027475e0    176  743e                  JZ 0xf80002747620        nt!MmGetKernelDumpRange + 0xf0
0xf800027475e2    178  488bca                MOV RCX, RDX
0xf800027475e5    181  48b80000a07dfbf6ffff  MOV RAX,
                                             0xfffff6fb7da00000
```

Context sensitive Actions can analyze in a modal box - for quick drilling.

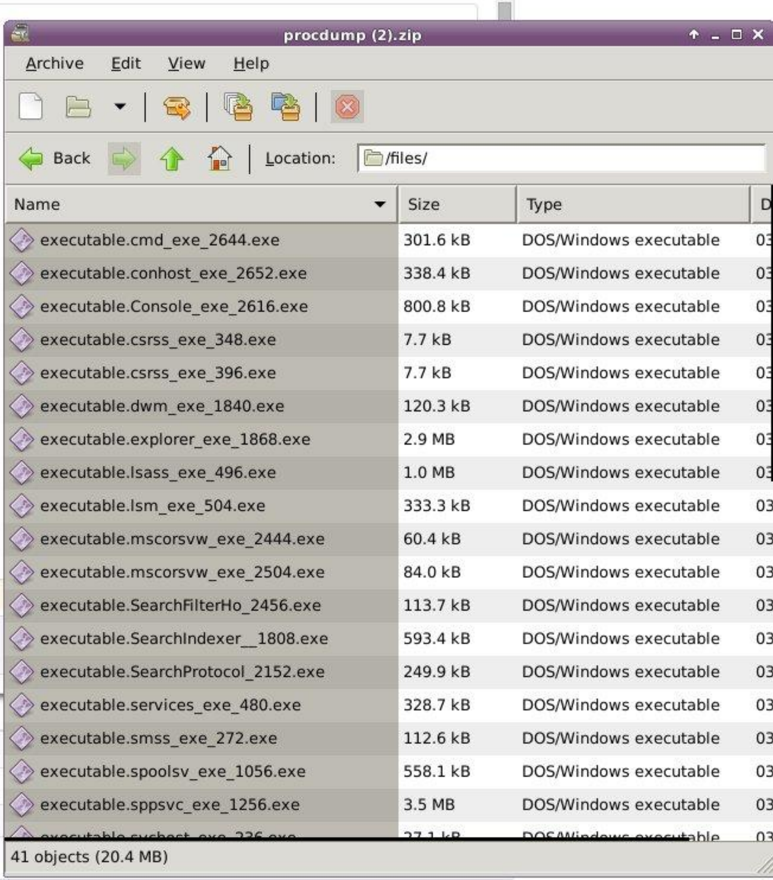
Plugins that dump files can produce a zip file.

Dumped files are also persistent in the Rekall file.



```
Dumping System , pid: 4 output: executable.System_4.exe
Dumping svchost.exe , pid: 236 output: executable.svchost_exe_236.exe
Dumping smss.exe , pid: 272 output: executable.smss_exe_272.exe
Dumping csrss.exe , pid: 348 output: executable.csrss_exe_348.exe
Dumping wininit.exe , pid: 384 output: executable.wininit_exe_384.exe
Dumping csrss.exe , pid: 396 output: executable.csrss_exe_396.exe
Dumping winlogon.exe , pid: 436 output: executable.winlogon_exe_436.exe
Dumping services.exe , pid: 480 output: executable.services_exe_480.exe
Dumping lsass.exe , pid: 496 output: executable.lsass_exe_496.exe
Dumping lsm.exe , pid: 504 output: executable.lsm_exe_504.exe
Dumping WmiPrvSE.exe , pid: 592 output: executable.WmiPrvSE_exe_592.exe
Dumping svchost.exe , pid: 608 output: executable.svchost_exe_608.exe
Dumping svchost.exe , pid: 624 output: executable.svchost_exe_624.exe
Dumping VBoxService.ex , pid: 664 output: executable.VBoxService_ex_664.exe
Dumping svchost.exe , pid: 716 output: executable.svchost_exe_716.exe
Dumping svchost.exe , pid: 768 output: executable.svchost_exe_768.exe
Dumping svchost.exe , pid: 872 output: executable.svchost_exe_872.exe
Dumping svchost.exe , pid: 932 output: executable.svchost_exe_932.exe
Dumping spoolsv.exe , pid: 1056 output: executable.spoolsv_exe_1056.exe
Dumping svchost.exe , pid: 1092 output: executable.svchost_exe_1092.exe
Dumping svchost.exe , pid: 1192 output: executable.svchost_exe_1192.exe
Dumping sppsvc.exe , pid: 1256 output: executable.sppsvc_exe_1256.exe
Dumping wlms.exe , pid: 1304 output: executable.wlms_exe_1304.exe
```

	PPID	Thds	Hnds	Sess	Wow64	Start	Exit
System (4)	0	84	511	-	✖	Oct 1, 2012 11:39:51 PM	Jan 1, 1
svchost.exe (236)	480	19	455	0	✖	Oct 1, 2012 4:40:01 PM	Jan 1, 1
smss.exe (272)	4	2	29	-	✖	Oct 1, 2012 11:39:51 PM	Jan 1, 1



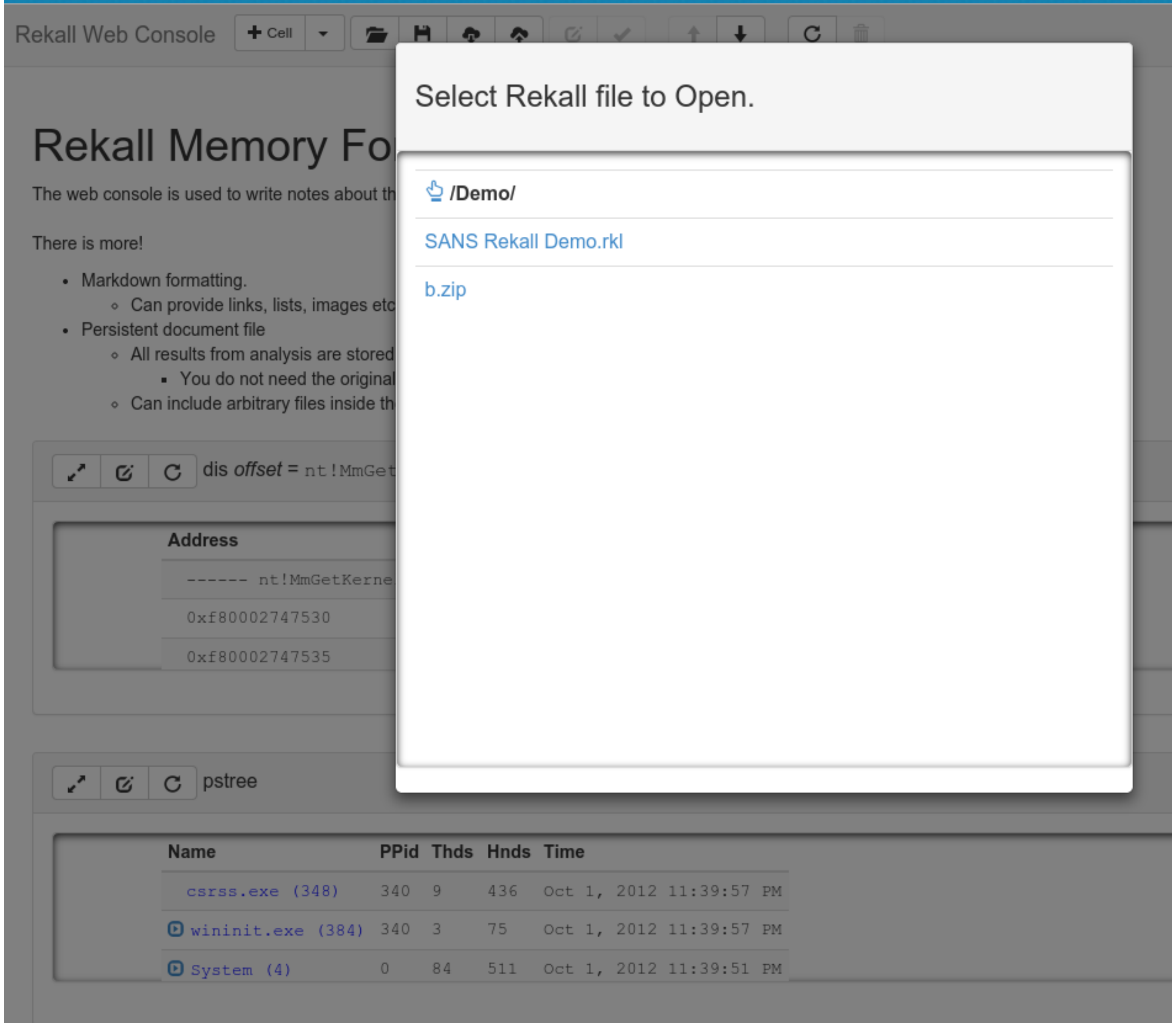
Name	Size	Type	D
executable.cmd_exe_2644.exe	301.6 kB	DOS/Windows executable	03
executable.conhost_exe_2652.exe	338.4 kB	DOS/Windows executable	03
executable.Console_exe_2616.exe	800.8 kB	DOS/Windows executable	03
executable.csrss_exe_348.exe	7.7 kB	DOS/Windows executable	03
executable.csrss_exe_396.exe	7.7 kB	DOS/Windows executable	03
executable.dwm_exe_1840.exe	120.3 kB	DOS/Windows executable	03
executable.explorer_exe_1868.exe	2.9 MB	DOS/Windows executable	03
executable.lsass_exe_496.exe	1.0 MB	DOS/Windows executable	03
executable.lsm_exe_504.exe	333.3 kB	DOS/Windows executable	03
executable.mscorsvw_exe_2444.exe	60.4 kB	DOS/Windows executable	03
executable.mscorsvw_exe_2504.exe	84.0 kB	DOS/Windows executable	03
executable.SearchFilterHo_2456.exe	113.7 kB	DOS/Windows executable	03
executable.SearchIndexer__1808.exe	593.4 kB	DOS/Windows executable	03
executable.SearchProtocol_2152.exe	249.9 kB	DOS/Windows executable	03
executable.services_exe_480.exe	328.7 kB	DOS/Windows executable	03
executable.smss_exe_272.exe	112.6 kB	DOS/Windows executable	03
executable.spoolsv_exe_1056.exe	558.1 kB	DOS/Windows executable	03
executable.sppsvc_exe_1256.exe	3.5 MB	DOS/Windows executable	03
executable.svchost_exe_236.exe	27.1 kB	DOS/Windows executable	03

41 objects (20.4 MB)

Analysis files can be restored at will.

The UI works directly with the analysis file so no need to "save" the document.

It is possible to download a current snapshot of the document at any time (e.g. for backup).



Select Rekall file to Open.

/Demo/

SANS Rekall Demo.rkl

b.zip

Rekall Memory Fo

The web console is used to write notes about th

There is more!

- Markdown formatting.
  - Can provide links, lists, images etc
- Persistent document file
  - All results from analysis are stored
    - You do not need the original
  - Can include arbitrary files inside th

dis *offset* = nt!MmGet

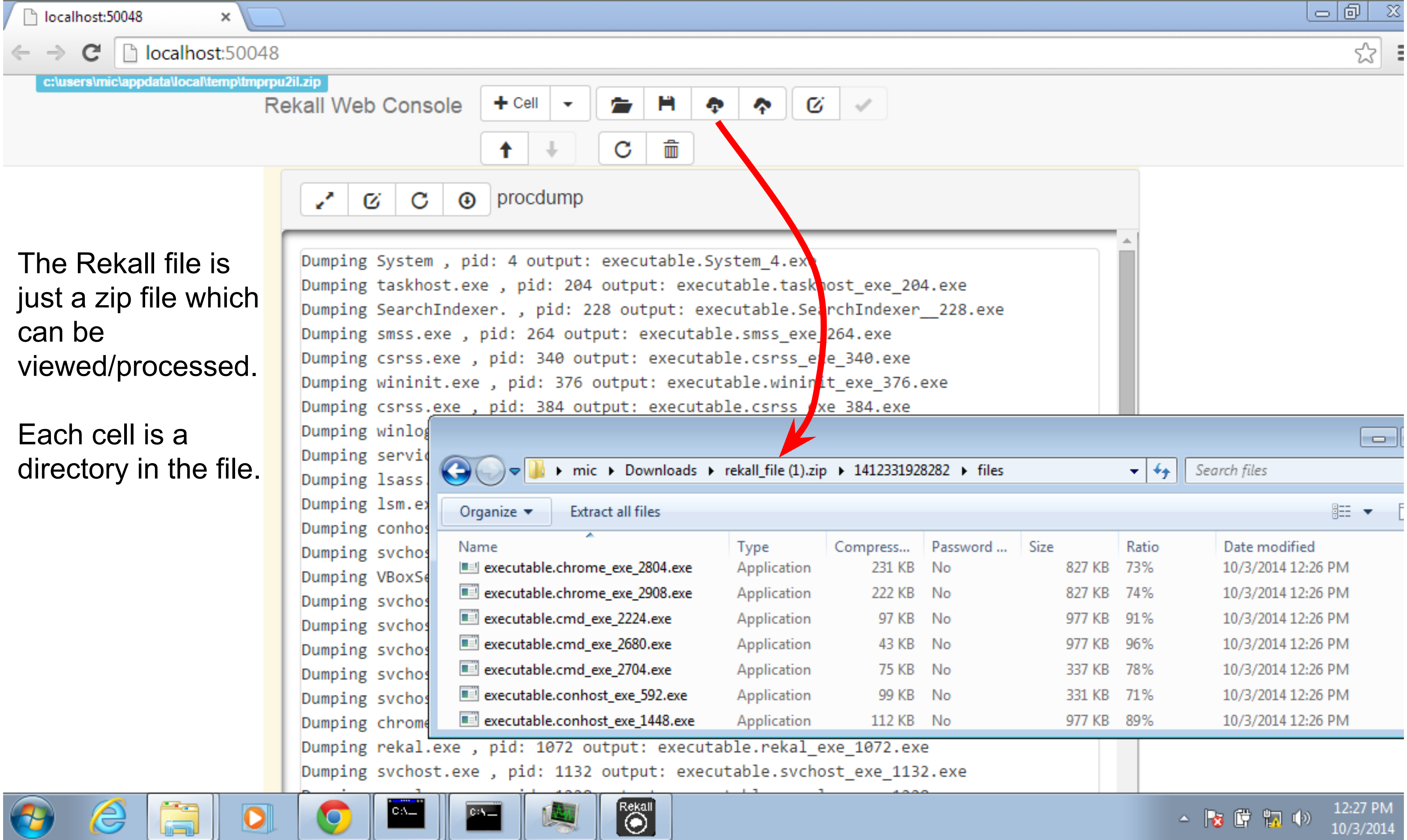
```
Address
------ nt!MmGetKerne
0xf80002747530
0xf80002747535
```

pstree

```
Name                PPid  Thds  Hnds  Time
 csrss.exe (348)    340   9     436   Oct 1, 2012 11:39:57 PM
 wininit.exe (384)  340   3     75    Oct 1, 2012 11:39:57 PM
 System (4)         0     84    511   Oct 1, 2012 11:39:51 PM
```

The Rekall file is just a zip file which can be viewed/processed.

Each cell is a directory in the file.



```
Dumping System , pid: 4 output: executable.System_4.exe
Dumping taskhost.exe , pid: 204 output: executable.taskhost_exe_204.exe
Dumping SearchIndexer. , pid: 228 output: executable.SearchIndexer__228.exe
Dumping smss.exe , pid: 264 output: executable.smss_exe_264.exe
Dumping csrss.exe , pid: 340 output: executable.csrss_exe_340.exe
Dumping wininit.exe , pid: 376 output: executable.wininit_exe_376.exe
Dumping csrss.exe , pid: 384 output: executable.csrss_exe_384.exe
Dumping winlog
Dumping servi
Dumping lsass.
Dumping lsm.ex
Dumping conhos
Dumping svchos
Dumping VBoxSe
Dumping svchos
Dumping svchos
Dumping svchos
Dumping svchos
Dumping svchos
Dumping chrome
Dumping rekal.exe , pid: 1072 output: executable.rekal_exe_1072.exe
Dumping svchost.exe , pid: 1132 output: executable.svchost_exe_1132.exe
```

mic ▸ Downloads ▸ rekall_file (1).zip ▸ 1412331928282 ▸ files

Name	Type	Compress...	Password ...	Size	Ratio	Date modified
executable.chrome_exe_2804.exe	Application	231 KB	No	827 KB	73%	10/3/2014 12:26 PM
executable.chrome_exe_2908.exe	Application	222 KB	No	827 KB	74%	10/3/2014 12:26 PM
executable.cmd_exe_2224.exe	Application	97 KB	No	977 KB	91%	10/3/2014 12:26 PM
executable.cmd_exe_2680.exe	Application	43 KB	No	977 KB	96%	10/3/2014 12:26 PM
executable.cmd_exe_2704.exe	Application	75 KB	No	337 KB	78%	10/3/2014 12:26 PM
executable.conhost_exe_592.exe	Application	99 KB	No	331 KB	71%	10/3/2014 12:26 PM
executable.conhost_exe_1448.exe	Application	112 KB	No	977 KB	89%	10/3/2014 12:26 PM

What makes it work?

- The UI uses Rekall's data export facility.
    - Rekall exports **structured, semantically aware** data:

Rekall uses Cybox "like" format to describe higher level objects in JSON.

```
$ rekal -v -f ~/images/win7.elf pslist -r data | json_pp
         "_EPROCESS" : {
...
            "Cybox" : {
               "Image_Info" : {
                  "Path" : "C:\\Windows\\system32\\csrss.exe",
                  "File_Name" : "\\Device\\HarddiskVolume2\\Windows\\System32\\csrss.exe",
                  "type" : "ProcessObj:ImageInfoType",
                  "Command_Line" : "%SystemRoot%\\system32\\csrss.exe ObjectDirectory=\\Windows
SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:
UserServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2 ServerDll=sxssrv,4
ProfileControl=Off MaxRequestThreads=16"
               }
```

Rekall Export System

- Rekall has a rich and highly customizable export system
    - Output format chosen by "Renderer"
        - Text renderer is default.
        - Data Export renderer produces rich JSON (used by the UI).
        - XLS renderer produces Excel sheets.
    - If we can make the GUI work with the exported data, any application can work with it!
        - This means you do not have to use Rekall as a library. Can be part of arbitrary pipeline.

Future - where are we heading?

- The Memory analysis field is exciting!
    - Lots of cool contributions
    - We want to integrate them all into the one tool.
- Linux analysis without custom profile
    - This is a big problem for Linux analysis - you must generate exact profile for each kernel you look at.
    - We have a way for collecting memory but we can not analyze it.
- More accessible and powerful user interface - we need feedback!

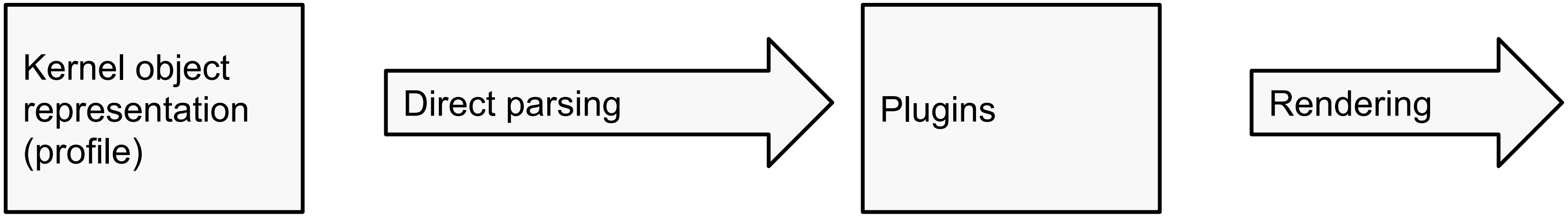
Focus point: High level abstraction

- Currently Memory analysis requires a lot of experience and knowledge.
  - Learning curve is steep and analysis is less accessible.
- We want to automate a lot of this.
  - Most of the time you just want to know:
    - Is this system owned?
  - There are common patterns to analysis
    - At least fire a red flag when something is weird.

Entity System - by Adam Sindelar

- Semantic layer to Rekall
    - Lives between overlay (data) and plugins (analysis)
    - Separates presentation (plugins) and collection (entity collectors)
- Features & Goals:
    - Speed (caching, de-duplication of effort)
    - Robustness (multiple entities for same object merge into one, preserving origin)
    - Treats analysis as a search problem

Old Way



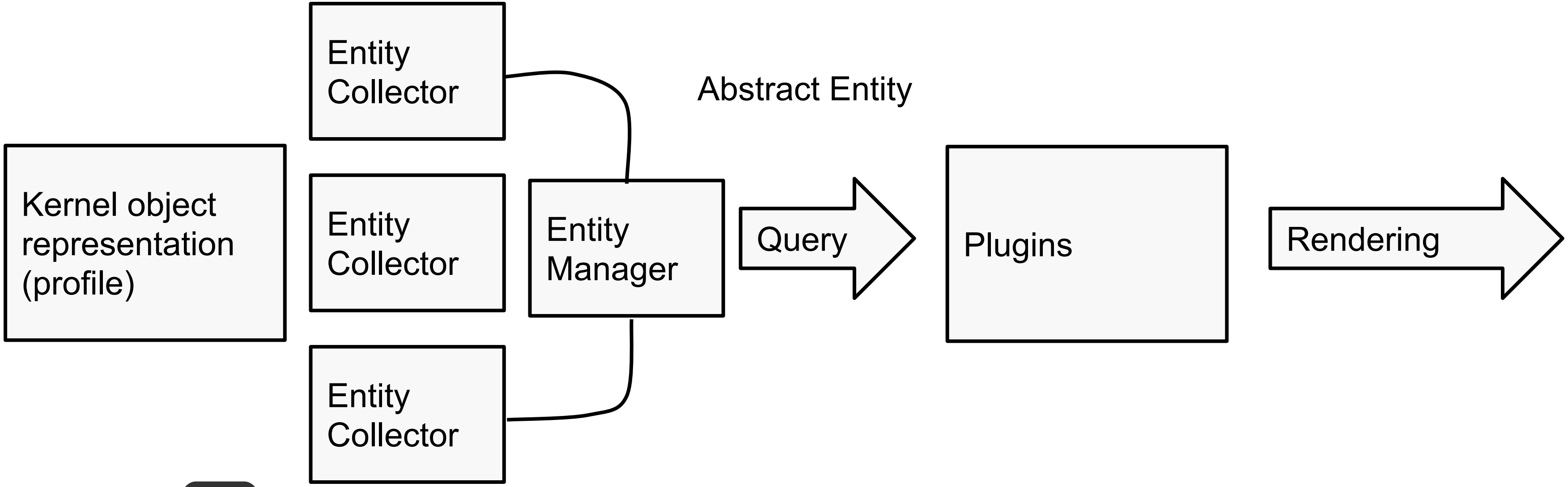
Kernel object representation (profile)

Direct parsing

Plugins

Rendering

Entity Way



Kernel object representation (profile)

Entity Collector

Entity Collector

Entity Collector

Entity Manager

Abstract Entity

Query

Plugins

Rendering

Entity System - Search

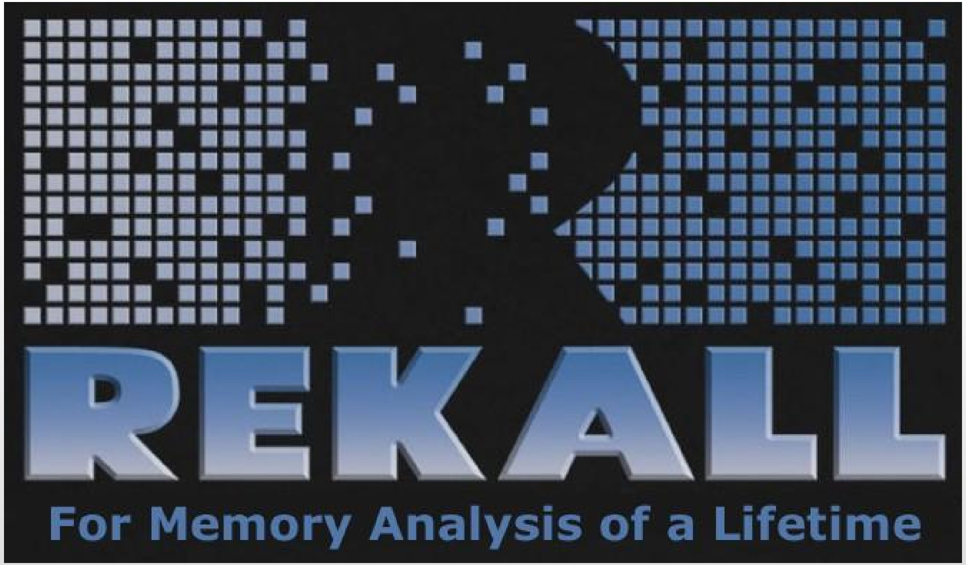
- Analysis as a search:
    - pslist/netstat and many others are basically search queries:
        - find all processes
        - find sockets where type=inet
    - IOCs and Artifacts are essentially a query language:
        - find process with comm=svchost.exe and parent.comm != services.exe

Entity System - now & future

- Current Status:
  - OS X partially migrated to entities
  - Windows has initial support
- Future work:
  - Artifacts implementation
  - Query languages - objectfilter, IOC, etc.
  - Data exchange (cybox/semantic protos) with plaso, timesketch and others.

http://www.rekall-forensic.com/

Sorry, Quaid. Your whole life is just a dream.



See you at the party, Richter!