UNSW SYDNEY | Australia's Global University

sec edu

security
engineering
capability

COMP6445 – Digital Forensics

Term 3 2019 - Week 3 part 2

1 October 2019

Topics for this lecture

Memory forensics

1. Windows memory forensics
2. Instructor-led exercise

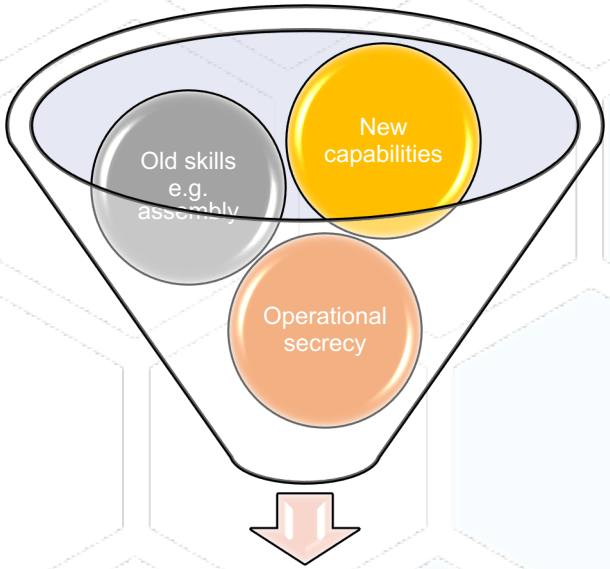
My introduction to Memory Forensics…

- Was in 1993 when I was attached to NSW Police Joint Technical Support Group (what is now knows as SEEB)
  - Decided in Burwood District Court in 1995
  - Defended by Chris Murphy and Clive Stern
  - More than 30 years has passed so no longer an official secret

- Convert arcade games such as pacman, frogger and space invaders into betting games such as blackjack and slot machines
  - A sequence of joystick movements combined with pressing buttons
  - The sequence was programmed by the operator at the beginning of the day
  - A kill switch sent it back to the arcade game and erased the sequence

Memory Forensics

- Is typically practiced as part of incident response and usually relating to malware
  - An area that is being well researched and becoming better understood
  - Partly due to commercial application
- Is also used for extracting credentials
  - By intelligence and law enforcement ⇨ meaning it is not well publicised
  - Every Australian case I have been involved in, the evidence has been suppressed, usually under the guise of operational secrecy
- It is the discipline of digital forensics surrounded by most mystique and often referred to as a “black art”

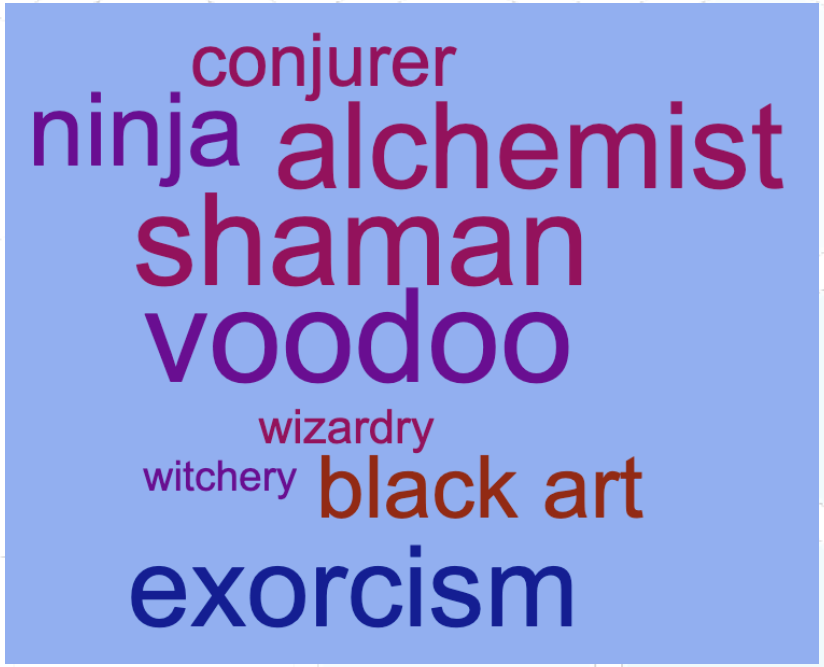


Outcome has to be understood and persuasive

Presenting memory forensic as evidence

- The biggest challenge is that it is often regarded as “black magic”. Technicians have been really bad at presenting and have too often avoided presenting under a veil of operational secrecy
  - Judges and barristers are both fascinated and sceptical regarding the ”outcome” of memory forensics
  - Juries (and clients) expect you to be able to do this in 15 minute i.e. the CSI effect
- Be persuasive regarding your credibility and the credibility of your chosen process(es)
  - How many times have you done this before – be prepared to give specific case references
  - How have you avoided bias? Why only selected memory artefacts?
    - The other expert will try to pull some exculpatory artefacts – be proactive to explain why these are irrelevant or unreliable
  - Have you tried different processes and arrived at the same result?
  - Is there an academic research base?

- Some words used in Court to describe what I have done…



Extract secrets from pagefile.sys and hibefil.sys

- Windows passwords
- Truecrypt and Veracrypt
- Password managers
  - KeePass
  - OnePassword
  - LastPass
- Grab TOR browsing i.e. Darkweb

Asar al mujedeen

- Used extensively by jihadists across in 2010s
  - Still being used
  - Rumour that NSA inserted monitor into Asar al mujdeen

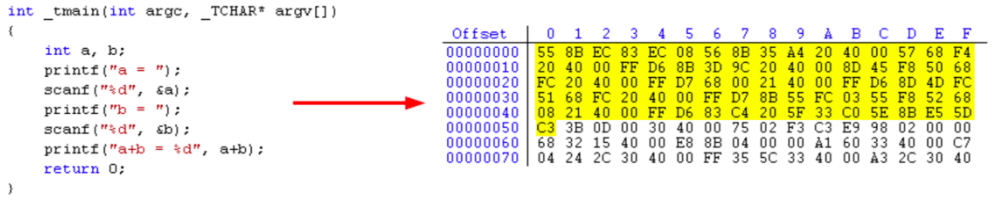
- GUI collection of tools based on shared libraries e.g. openpgp

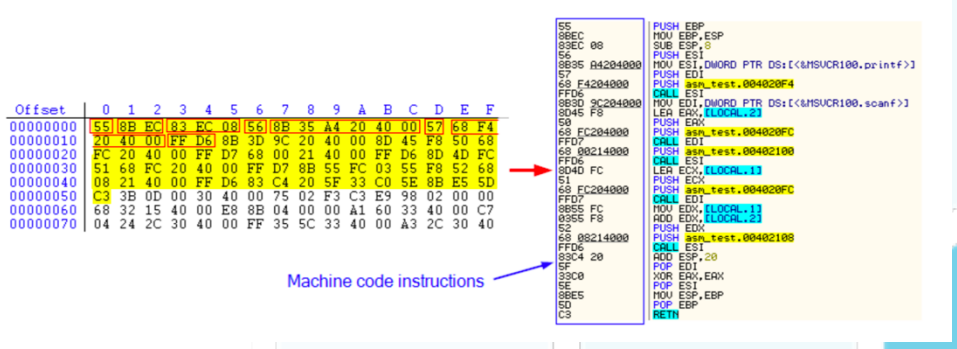
- Asar.exe and keyfile on a USB key

- Extract the key from the virtual memory of a computer used by jihadist
  - pagfile.sys
  - hiberfile.sys

Computer programs

- Programs written in languages like C, C++, etc are compiled into machine code
- Machine code can be executed directly by the CPU
- Machine code is difficult for a human to read
- Machine code cannot be reliably converted back to the source language
- Assembler is an intermediate form which is a direct translation of machine code

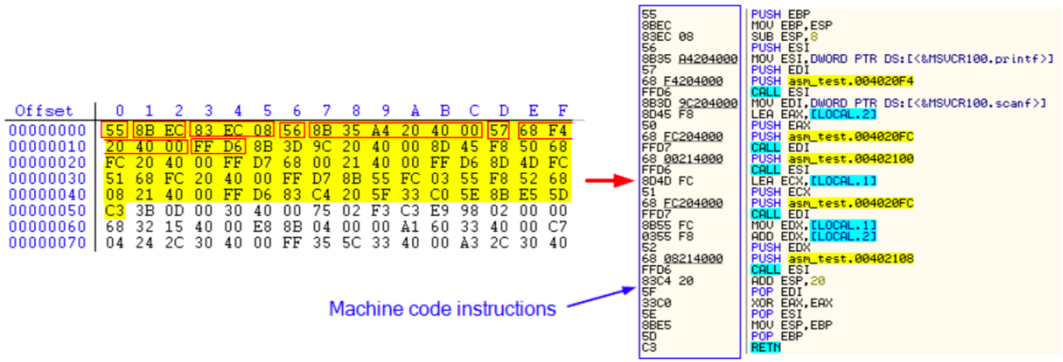




Higher language ⇨machine code

Higher language ⇎ machine code

Machine code ⇔assembler



SUB ESP, 8

83	EC	08
opcode		operand

- Opcode = operation that should be performed

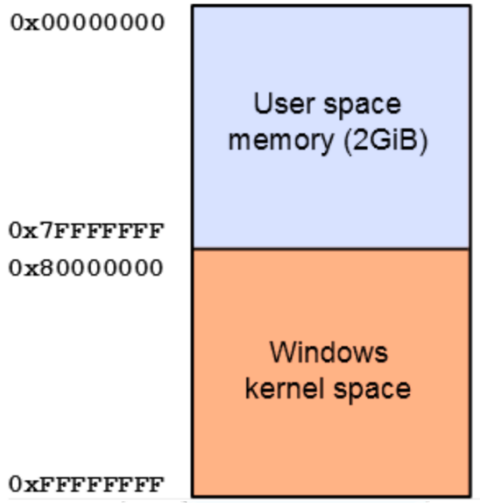
- Operand = argument to the operation
  - Immediate value – value encoded in the instruction itself e.g. sub esp, 8
  - Register – operand is one of the registers
  - Memory – operand is in the memory (specified by offset encoded in the instruction)

- 16 types of register
  - 8x general purpose registers (EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP)
  - 6x segment registers (CS, DS, SS, ES, FS, GS)
  - 1x flags register (EFLAGS)
  - 1x instruction pointer register (EIP)

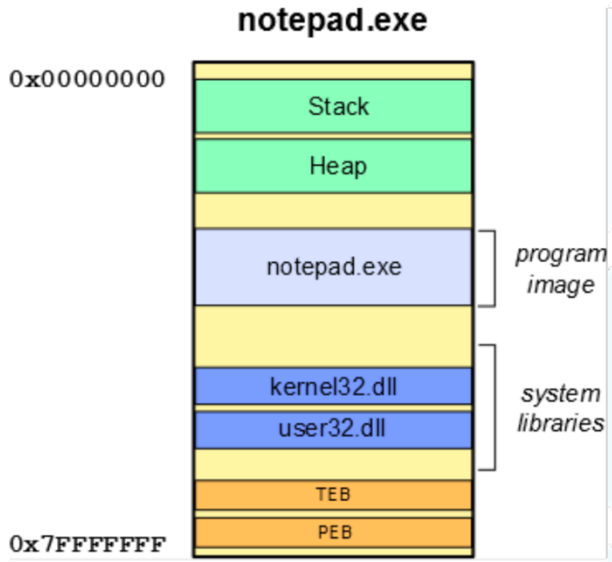
Memory

- The Microsoft Windows operating system (as well as most other contemporary operating systems) uses a flat memory model in which programs see memory as a contiguous and linear address space

- On Microsoft Windows, the system memory of 32-bit processes is addressed through 32-bit addresses starting from 0 up to 0xFFFFFFFF (4GB)
  - Not all address space is available to the user-mode processes.
  - User-mode processes can access freely only memory from 0 up to 0x7FFFFFFF (2GB).
  - The second half, that is addresses from 0x80000000 up to 0xFFFFFFFF, is reserved for the operating system

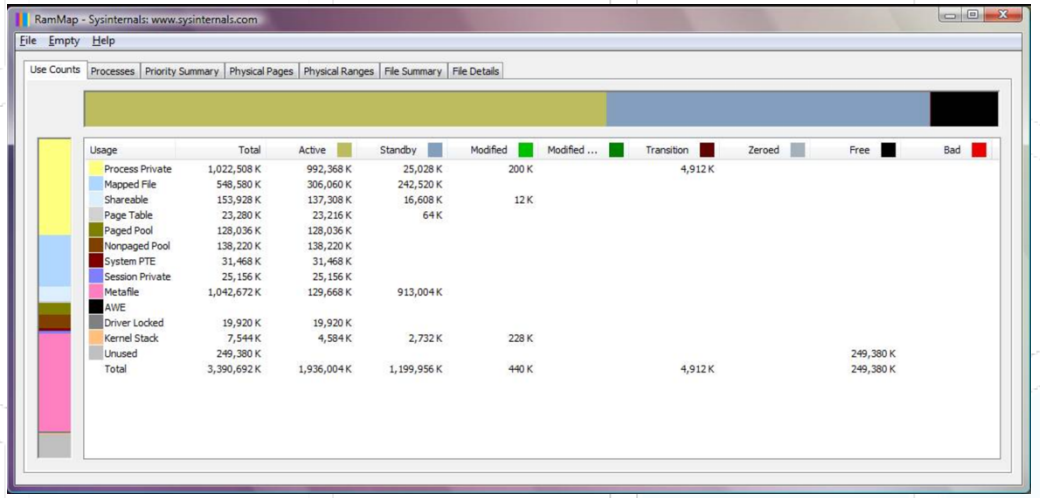


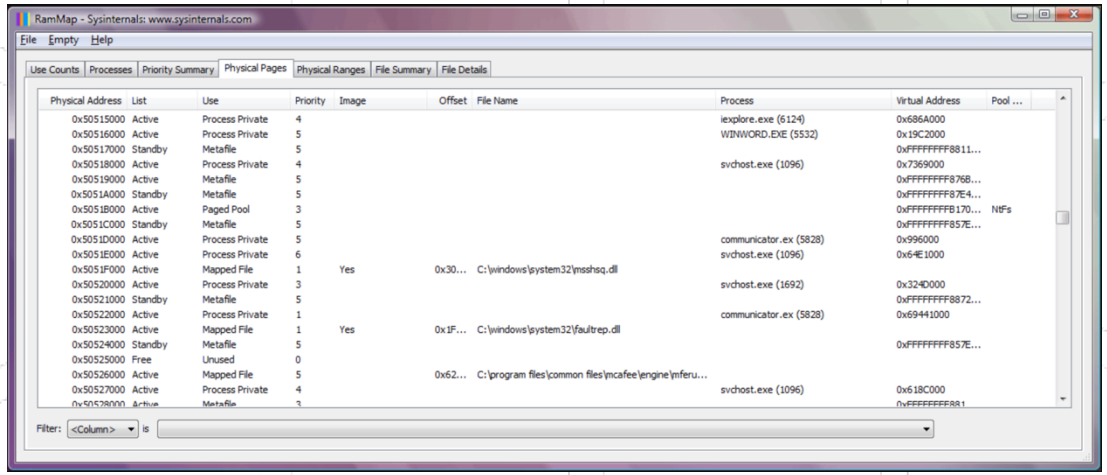
- When a new PE executable is started on a Windows system, a new process is created and the system loader maps the PE file into the process's address space as well as loads all DLL libraries needed by the program
- Process heap and stack are also created
- The Thread Environment Block (TEB) and the Process Environment Block (PEB) are system structures providing information about the current thread's context and the process itself
- Two important memory structures are stack and heap.
  - The **heap** is a memory region where dynamically allocated variables are put.
  - The **stack** is used for storing local variables and tracing function calls in the current thread
  - The stack is also used for passing function arguments and tracing function calls

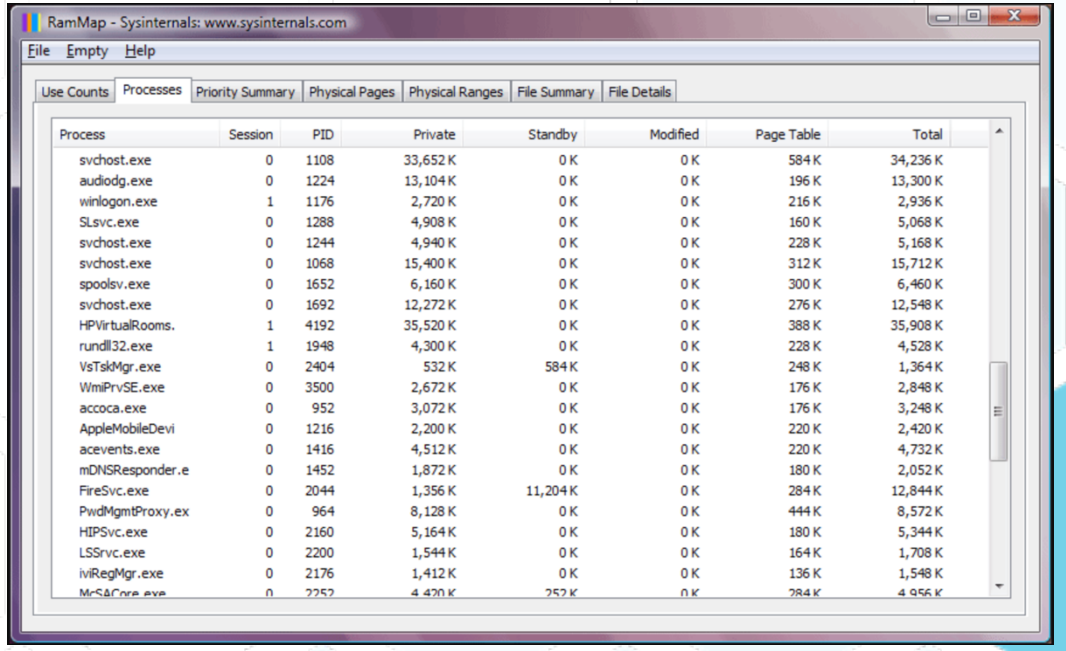


RAMMap

- RAMMap is a tool from Microsoft sysinternals that allows you to view memory

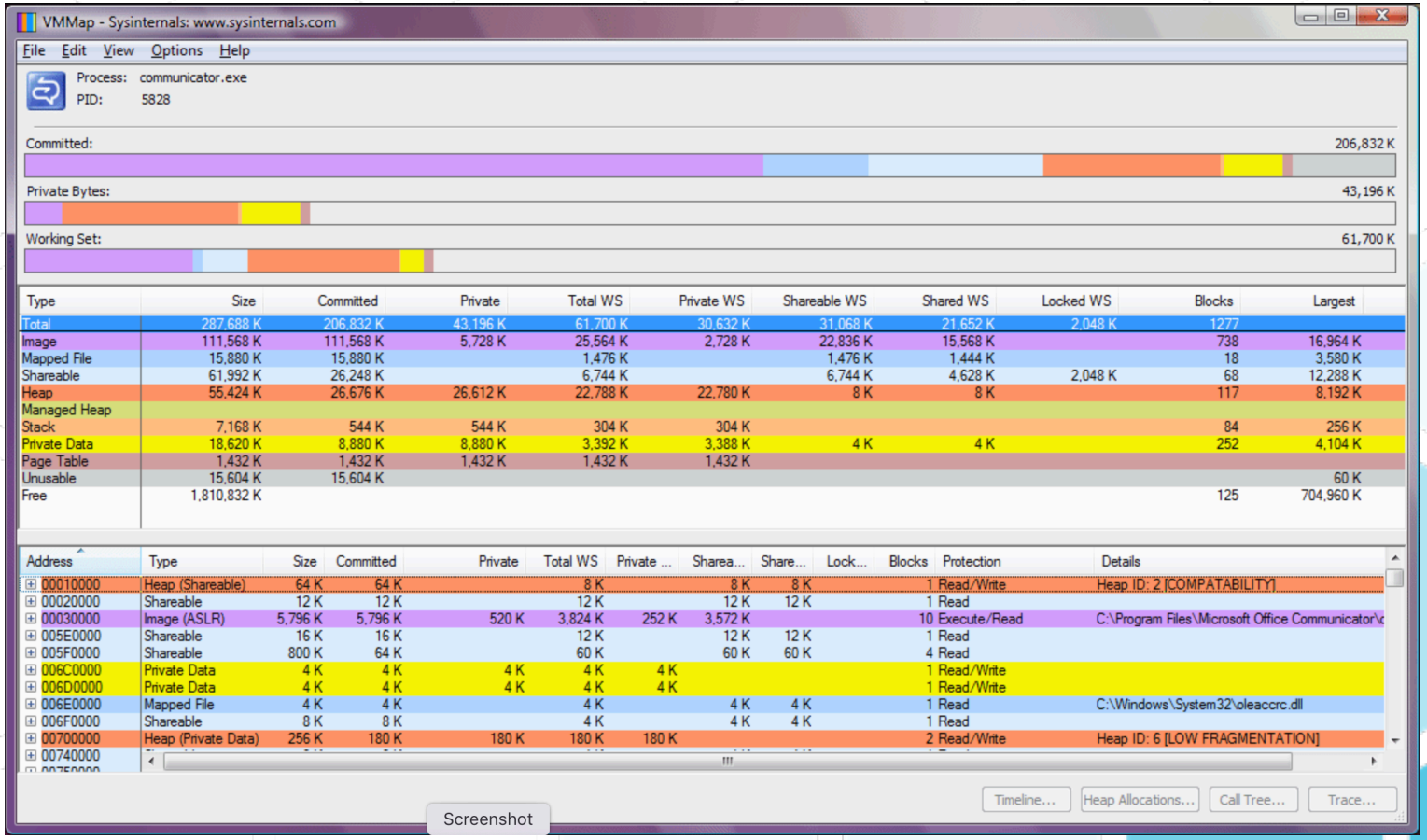






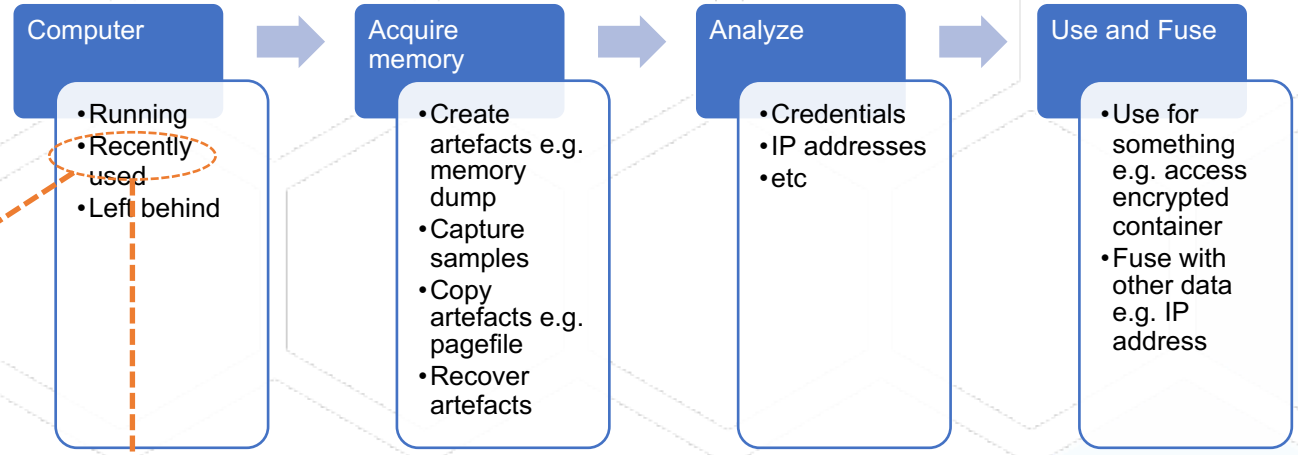
VMMap

- VMMap is another tool from Microsoft sysinternals that allows you to view memory



Screenshot

Workflow



You will be surprised how many times a laptop is closed and stored somewhere

You will be surprised how many times a laptop is seized and put into evidence with memory being analysed years later (usually as part of the defendants case)

You will be surprised how many times a person is coerced to provide a password when it is there (with a little bit of work)

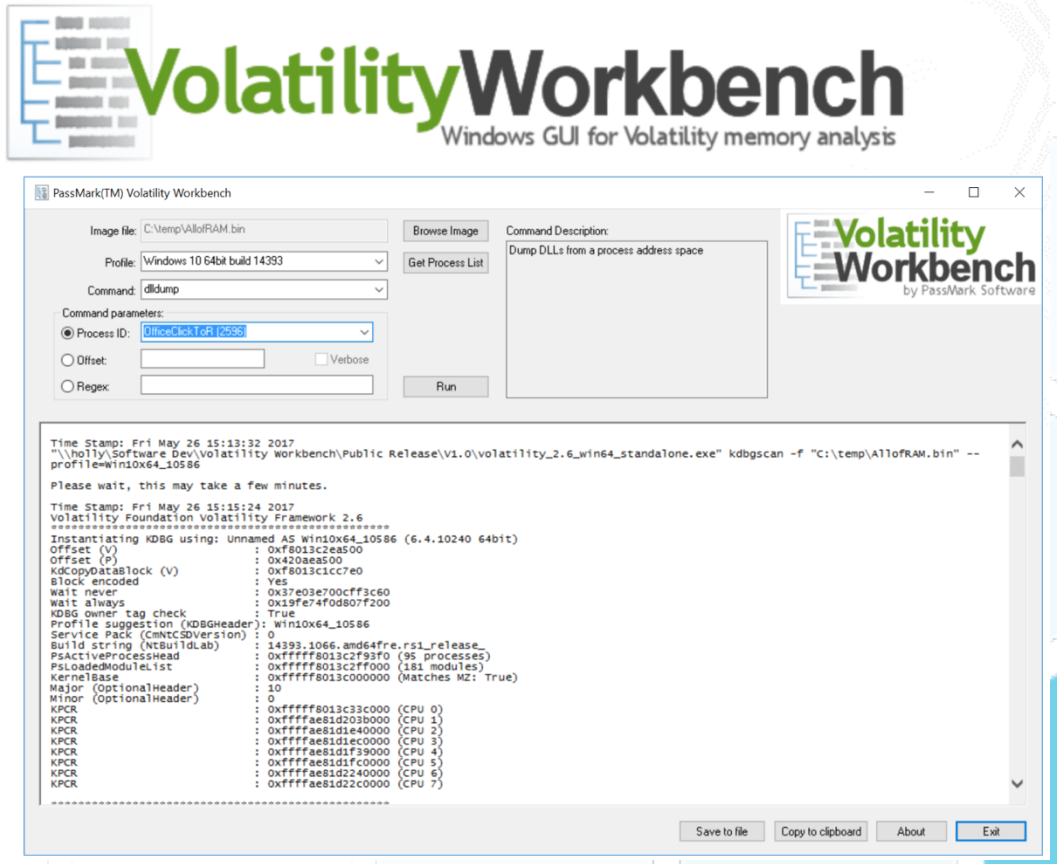
What does this mean for the Government's assertion that they need extra powers to require assistance?

Capturing memory using tools

- Windows

- Mandient Memoryze
- FTK Imager
- Encase winen.exe
- Magnet RAM Capture
- Belkasoft Live RAM Capture
- OSForensics

- DumpIt
- Memory DD
- WinPmem



https://www.osforensics.com/tools/volatility-workbench.html

Copying memory files

Windows

PAGEFILE.SYS

- Virtual memory or “swap file”
- Locked whilst computer is being used

HIBERFIL.SYS

- Supports hibernation
- In root directory i.e. c:\hiberfil.sys

Mac (and Linux)

SWAPFILE

- SWAPFILE and a number
- /private/var/vm
- Only starts when computer is low on memory

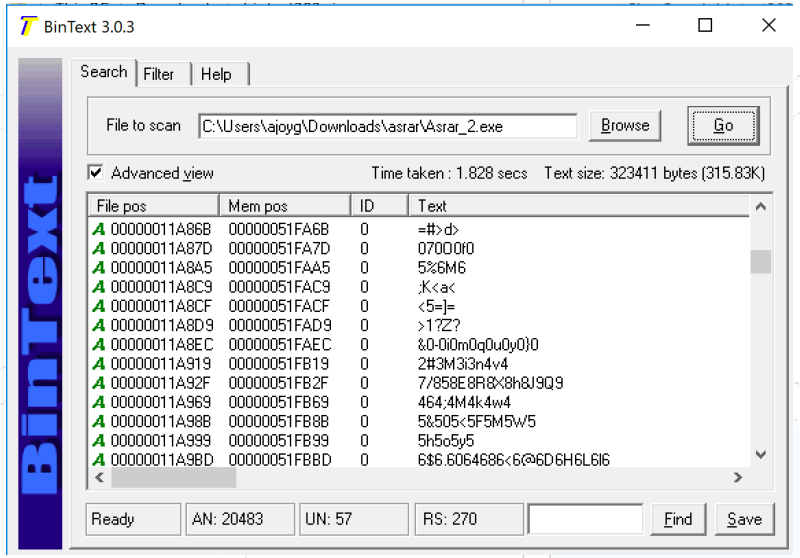
SLEEPIMAGE

- Hibernation
- /var/vm

BinText

- Is a free tool provided by McAfee

- Used to extract strings from binary files
- IP address and domain names
- Passwords
- Anti-forensics




```
A 00000001B380  00000041B380  0  %%%c%c%%%c%c
A 00000001B398  00000041B398  0  %d.%d.%d.%d
A 00000001B3A4  00000041B3A4  0  192.168.1.244
```

```
A 00000001B69C  00000041B69C  0  GetStartupInfoA
A 00000001B6B0  00000041B6B0  0  ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
A 00000001B6F8  00000041B6F8  0  1107791273.f3322.org
```

```
A 00000001BF7C  00000041BF7C  0  asdfgh
A 00000001BF84  00000041BF84  0  1314520
A 00000001BF8C  00000041BF8C  0  5201314
A 00000001BF94  00000041BF94  0  caonima
A 00000001BF9C  00000041BF9C  0  88888
A 00000001BFA4  00000041BFA4  0  bbbbbb
A 00000001BFAC  00000041BFAC  0  12345678
A 00000001BFB8  00000041BFB8  0  memory
A 00000001BFC0  00000041BFC0  0  abc123
A 00000001BFC8  00000041BFC8  0  qwerty
A 00000001BFD0  00000041BFD0  0  123456
A 00000001BFDC  00000041BFDC  0  password
A 00000001BFE8  00000041BFE8  0  enter
A 00000001BFF8  00000041BFF8  0  xpuser
A 00000001C000  00000041C000  0  money
```

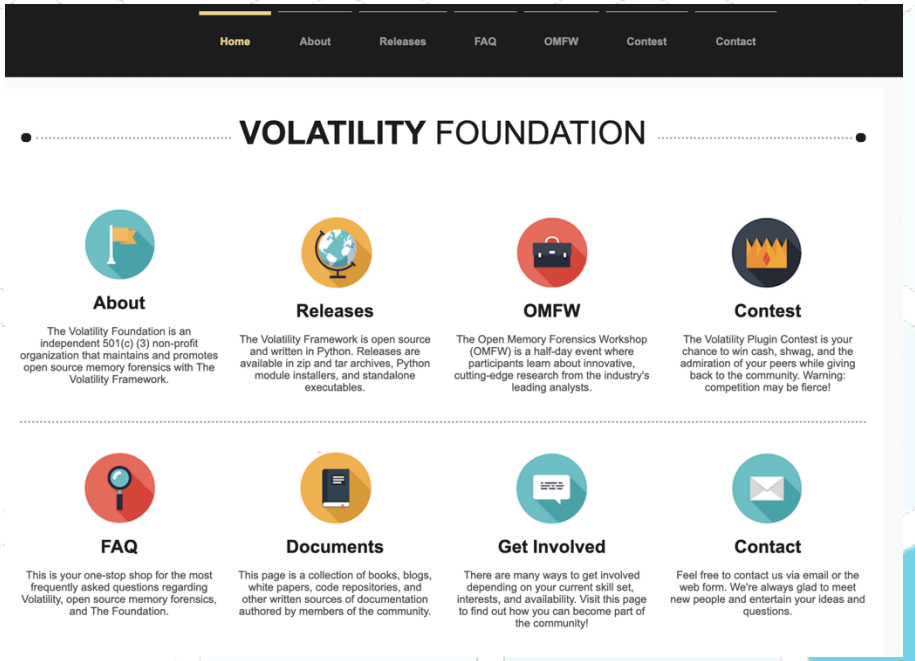
Usually there is a bit of work to decode these

```
A 00000001BA70  00000041BA70  0  F-secure
A 00000001BA7C  00000041BA7C  0  f-secure.exe
A 00000001BA94  00000041BA94  0  FortiTray.exe
A 00000001BAA8  00000041BAA8  0  avg.exe
A 00000001BAB0  00000041BAB0  0  Norman
A 00000001BAB8  00000041BAB8  0  NVCSched.exe
A 00000001BAC8  00000041BAC8  0  ClamAV
A 00000001BAD0  00000041BAD0  0  agent.exe
```

Volatility

- Is the open source tool of choice for memory forensics

Let's do a small exercise using Volatility – you will do this in the tutorial/lab

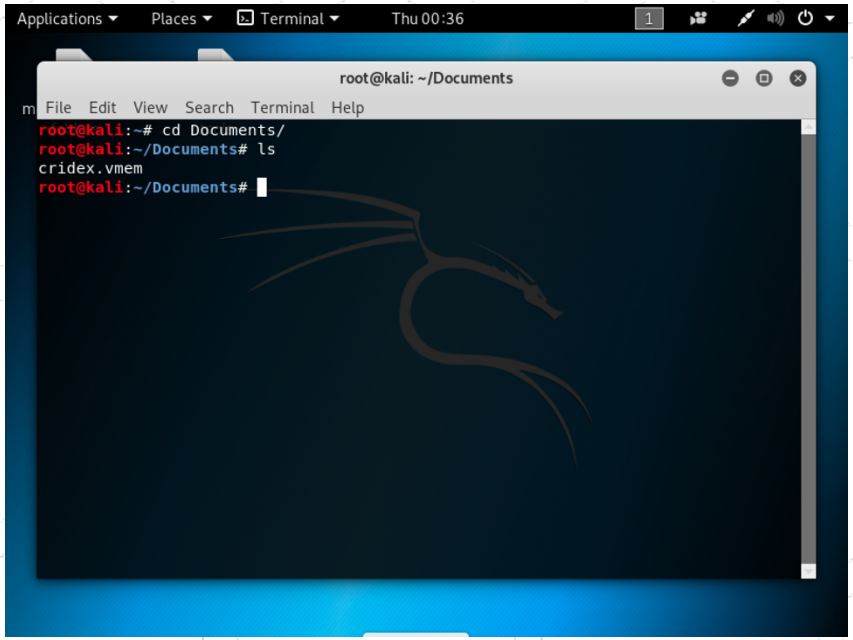


Use Kali Linux

- Volatility is pre-installed in Kali Linux
- Screenshots are for Kali Linux
  - If you prefer, you can follow on using Windows

User = root

Password = toor



- Open a terminal
- Change directory to Documents
- Check that the cridex.mem file is there
  - Memory dump used for exercise

- View image information

$ volatility -f cridex.vmem imageinfo

- Write down the suggested profile – you will need it later

WinXPSP2x86

```
root@kali:~/Documents# volatility -f cridex.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with Win
XPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/root/Documents/cridex.vmem)
                      PAE type : PAE
                           DTB : 0x2fe000L
                          KDBG : 0x80545ae0L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2012-07-22 02:45:08 UTC+0000
     Image local date and time : 2012-07-21 22:45:08 -0400
root@kali:~/Documents#
```

- View running processes

$ volatility -f cridex.vmem --profile=WinXPSP2x86 pslist

```
root@kali:~/Documents# volatility -f cridex.vmem --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x823c89c8 System                    4      0     53      240 ------      0
0x822f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31 UTC+0000
0x822a0598 csrss.exe               584    368      9      326      0      0 2012-07-22 02:42:32 UTC+0000
0x82298700 winlogon.exe            608    368     23      519      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2ab28 services.exe            652    608     16      243      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2a3b8 lsass.exe               664    608     24      330      0      0 2012-07-22 02:42:32 UTC+0000
0x82311360 svchost.exe             824    652     20      194      0      0 2012-07-22 02:42:33 UTC+0000
0x81e29ab8 svchost.exe             908    652      9      226      0      0 2012-07-22 02:42:33 UTC+0000
0x823001d0 svchost.exe            1004    652     64     1118      0      0 2012-07-22 02:42:33 UTC+0000
0x821dfda0 svchost.exe            1056    652      5       60      0      0 2012-07-22 02:42:33 UTC+0000
0x82295650 svchost.exe            1220    652     15      197      0      0 2012-07-22 02:42:35 UTC+0000
0x821dea70 explorer.exe           1484   1464     17      415      0      0 2012-07-22 02:42:36 UTC+0000
0x81eb17b8 spoolsv.exe            1512    652     14      113      0      0 2012-07-22 02:42:36 UTC+0000
0x81e7bda0 reader_sl.exe          1640   1484      5       39      0      0 2012-07-22 02:42:36 UTC+0000
0x820e8da0 alg.exe                 788    652      7      104      0      0 2012-07-22 02:43:01 UTC+0000
0x821fcda0 wuauclt.exe            1136   1004      8      173      0      0 2012-07-22 02:43:46 UTC+0000
0x8205bda0 wuauclt.exe            1588   1004      5      132      0      0 2012-07-22 02:44:01 UTC+0000
root@kali:~/Documents#
```

- View parent processes

$ volatility -f cridex.vmem --profile=WinXPSP2x86 pstree

- Notice reader_sl.exe spawned by explorer

```
root@kali:~/Documents# volatility -f cridex.vmem --profile=WinXPSP2x86 pstree
Volatility Foundation Volatility Framework 2.6
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x823c89c8:System                                      4      0     53    240 1970-01-01 00:00:00 UTC+0000
. 0x822f1020:smss.exe                                 368      4      3     19 2012-07-22 02:42:31 UTC+0000
.. 0x82298700:winlogon.exe                            608    368     23    519 2012-07-22 02:42:32 UTC+0000
... 0x81e2ab28:services.exe                           652    608     16    243 2012-07-22 02:42:32 UTC+0000
.... 0x821dfda0:svchost.exe                          1056    652      5     60 2012-07-22 02:42:33 UTC+0000
.... 0x81eb17b8:spoolsv.exe                          1512    652     14    113 2012-07-22 02:42:36 UTC+0000
.... 0x81e29ab8:svchost.exe                           908    652      9    226 2012-07-22 02:42:33 UTC+0000
.... 0x823001d0:svchost.exe                          1004    652     64   1118 2012-07-22 02:42:33 UTC+0000
..... 0x8205bda0:wuauclt.exe                         1588   1004      5    132 2012-07-22 02:44:01 UTC+0000
..... 0x821fcda0:wuauclt.exe                         1136   1004      8    173 2012-07-22 02:43:46 UTC+0000
.... 0x82311360:svchost.exe                           824    652     20    194 2012-07-22 02:42:33 UTC+0000
.... 0x820e8da0:alg.exe                               788    652      7    104 2012-07-22 02:43:01 UTC+0000
.... 0x82295650:svchost.exe                          1220    652     15    197 2012-07-22 02:42:35 UTC+0000
... 0x81e2a3b8:lsass.exe                              664    608     24    330 2012-07-22 02:42:32 UTC+0000
.. 0x822a0598:csrss.exe                               584    368      9    326 2012-07-22 02:42:32 UTC+0000
 0x821dea70:explorer.exe                             1484   1464     17    415 2012-07-22 02:42:36 UTC+0000
. 0x81e7bda0:reader_sl.exe                           1640   1484      5     39 2012-07-22 02:42:36 UTC+0000
root@kali:~/Documents#
```

- Are any processes trying to hide?

$ volatility -f cridex.vmem --profile=WinXPSP2x86 psxview

- If so, FALSE will appear in the first two columns

```
root@kali:~/Documents# volatility -f cridex.vmem --profile=WinXPSP2x86 psxview
Volatility Foundation Volatility Framework 2.6
Offset(P)  Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd ExitTime
---------- -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --------
0x02498700 winlogon.exe            608 True   True   True     True   True  True    True
0x02511360 svchost.exe             824 True   True   True     True   True  True    True
0x022e8da0 alg.exe                 788 True   True   True     True   True  True    True
0x020b17b8 spoolsv.exe            1512 True   True   True     True   True  True    True
0x0202ab28 services.exe            652 True   True   True     True   True  True    True
0x02495650 svchost.exe            1220 True   True   True     True   True  True    True
0x0207bda0 reader_sl.exe          1640 True   True   True     True   True  True    True
0x025001d0 svchost.exe            1004 True   True   True     True   True  True    True
0x02029ab8 svchost.exe             908 True   True   True     True   True  True    True
0x023fcda0 wuauclt.exe            1136 True   True   True     True   True  True    True
0x0225bda0 wuauclt.exe            1588 True   True   True     True   True  True    True
0x0202a3b8 lsass.exe               664 True   True   True     True   True  True    True
0x023dea70 explorer.exe           1484 True   True   True     True   True  True    True
0x023dfda0 svchost.exe            1056 True   True   True     True   True  True    True
0x024f1020 smss.exe                368 True   True   True     True   False False   False
0x025c89c8 System                    4 True   True   True     True   False False   False
0x024a0598 csrss.exe               584 True   True   True     True   False True    True
```

- Look at connections and sockets

$ volatility -f cridex.vmem --profile=WinXPSP2x86 connscan

$ volatility -f cridex.vmem --profile=WinXPSP2x86 sockets

- Normally use netscan but not available for WinXPSP2x86

```
root@kali:~/Documents# volatility -f cridex.vmem --profile=WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x02087620 172.16.112.128:1038      41.168.5.140:8080         1484
0x023a8008 172.16.112.128:1037      125.19.103.198:8080       1484
root@kali:~/Documents# volatility -f cridex.vmem --profile=WinXPSP2x86 sockets
Volatility Foundation Volatility Framework 2.6
Offset(V)       PID   Port  Proto Protocol        Address         Create Time
---------- -------- ------ ------ --------------- --------------- -----------
0x81ddb780      664    500     17 UDP             0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x82240d08     1484   1038      6 TCP             0.0.0.0         2012-07-22 02:44:45 UTC+0000
0x81dd7618     1220   1900     17 UDP             172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x82125610      788   1028      6 TCP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x8219cc08        4    445      6 TCP             0.0.0.0         2012-07-22 02:42:31 UTC+0000
0x81ec23b0      908    135      6 TCP             0.0.0.0         2012-07-22 02:42:33 UTC+0000
0x82276878        4    139      6 TCP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x82277460        4    137     17 UDP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x81e76620     1004    123     17 UDP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x82172808      664      0    255 Reserved        0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x81e3f460        4    138     17 UDP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x821f0630     1004    123     17 UDP             172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x822cd2b0     1220   1900     17 UDP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x82172c50      664   4500     17 UDP             0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x821f0d00        4    445     17 UDP             0.0.0.0         2012-07-22 02:42:31 UTC+0000
```

- Look at PID 1484 i.e. explorer
  - two connections, port 1038 and port 1037

- One is still open i.e. port 1038
- There is no socket entry for 1037 so that is suspicious

```
root@kali:~/Documents# volatility -f cridex.vmem --profile=WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x02087620 172.16.112.128:1038      41.168.5.140:8080         1484
0x023a8008 172.16.112.128:1037      125.19.103.198:8080       1484
root@kali:~/Documents# volatility -f cridex.vmem --profile=WinXPSP2x86 sockets
Volatility Foundation Volatility Framework 2.6
Offset(V)       PID   Port  Proto Protocol        Address         Create Time
---------- -------- ------ ------ --------------- --------------- -----------
0x81ddb780      664    500     17 UDP             0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x82240d08     1484   1038      6 TCP             0.0.0.0         2012-07-22 02:44:45 UTC+0000
0x81dd7618     1220   1900     17 UDP             172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x82125610      788   1028      6 TCP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x8219cc08        4    445      6 TCP             0.0.0.0         2012-07-22 02:42:31 UTC+0000
0x81ec23b0      908    135      6 TCP             0.0.0.0         2012-07-22 02:42:33 UTC+0000
0x82276878        4    139      6 TCP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x82277460        4    137     17 UDP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x81e76620     1004    123     17 UDP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x82172808      664      0    255 Reserved        0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x81e3f460        4    138     17 UDP             172.16.112.128  2012-07-22 02:42:38 UTC+0000
0x821f0630     1004    123     17 UDP             172.16.112.128  2012-07-22 02:43:01 UTC+0000
0x822cd2b0     1220   1900     17 UDP             127.0.0.1       2012-07-22 02:43:01 UTC+0000
0x82172c50      664   4500     17 UDP             0.0.0.0         2012-07-22 02:42:53 UTC+0000
0x821f0d00        4    445     17 UDP             0.0.0.0         2012-07-22 02:42:31 UTC+0000
```

- Look at last commands run using cmdscan, consoles and cmdline
  - cmdscan and consoles don't produce relevant results in this example

$ volatility -f cridex.vmem --profile=WinXPSP2x86 cmdline

```
Command line : C:\WINDOWS\system32\lsass.exe
************************************************************************
svchost.exe pid:    824
Command line : C:\WINDOWS\system32\svchost -k DcomLaunch
************************************************************************
svchost.exe pid:    908
Command line : C:\WINDOWS\system32\svchost -k rpcss
************************************************************************
svchost.exe pid:   1004
Command line : C:\WINDOWS\System32\svchost.exe -k netsvcs
************************************************************************
svchost.exe pid:   1056
Command line : C:\WINDOWS\system32\svchost.exe -k NetworkService
************************************************************************
svchost.exe pid:   1220
Command line : C:\WINDOWS\system32\svchost.exe -k LocalService
************************************************************************
explorer.exe pid:   1484
Command line : C:\WINDOWS\Explorer.EXE
************************************************************************
spoolsv.exe pid:   1512
Command line : C:\WINDOWS\system32\spoolsv.exe
************************************************************************
reader_sl.exe pid:   1640
Command line : "C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe"
************************************************************************
alg.exe pid:    788
Command line : C:\WINDOWS\System32\alg.exe
************************************************************************
wuauclt.exe pid:   1136
Command line : "C:\WINDOWS\system32\wuauclt.exe" /RunStoreAsComServer Local\[3ec]SUSDSb81eb56fa3105543beb3109274ef8ec1
************************************************************************
wuauclt.exe pid:   1588
Command line : "C:\WINDOWS\system32\wuauclt.exe"
```

- Look at last commands run using cmdscan, consoles and cmdline
  - cmdscan and consoles don't produce relevant results in this example

$ volatility -f cridex.vmem --profile=WinXPSP2x86 cmdline

```
Command line : C:\WINDOWS\system32\lsass.exe
**************************************************
svchost.exe pid:    824
Command line : C:\WINDOWS\system32\svchost -k DcomLaunch
**************************************************
svchost.exe pid:    908
Command line : C:\WINDOWS\system32\svchost -k rpcss
**************************************************
svchost.exe pid:   1004
Command line : C:\WINDOWS\System32\svchost.exe -k netsvcs
**************************************************
svchost.exe pid:   1056
Command line : C:\WINDOWS\system32\svchost.exe -k NetworkService
**************************************************
svchost.exe pid:   1220
Command line : C:\WINDOWS\system32\svchost.exe -k LocalService
**************************************************
explorer.exe pid:   1484
Command line : C:\WINDOWS\Explorer.EXE
**************************************************
spoolsv.exe pid:   1512
Command line : C:\WINDOWS\system32\spoolsv.exe
**************************************************
reader_sl.exe pid:   1640
Command line : "C:\Program Files\Adobe\Reader 9.0\Reader\Reader_sl.exe"
**************************************************
alg.exe pid:    788
Command line : C:\WINDOWS\System32\alg.exe
**************************************************
wuauclt.exe pid:   1136
Command line : "C:\WINDOWS\system32\wuauclt.exe" /RunStoreAsComServer Local\[3ec]SUSDSb81eb56fa3105543beb3109274ef8ec1
**************************************************
wuauclt.exe pid:   1588
Command line : "C:\WINDOWS\system32\wuauclt.exe"
```

- Look at last commands run using cmdscan, consoles and cmdline

$ volatility -f cridex.vmem --profile=WinXPSP2x86 procdump -p 1640 --dump-dir .

$ volatility -f cridex.vmem --profile=WinXPSP2x86 memdump -p 1640 --dump-dir .

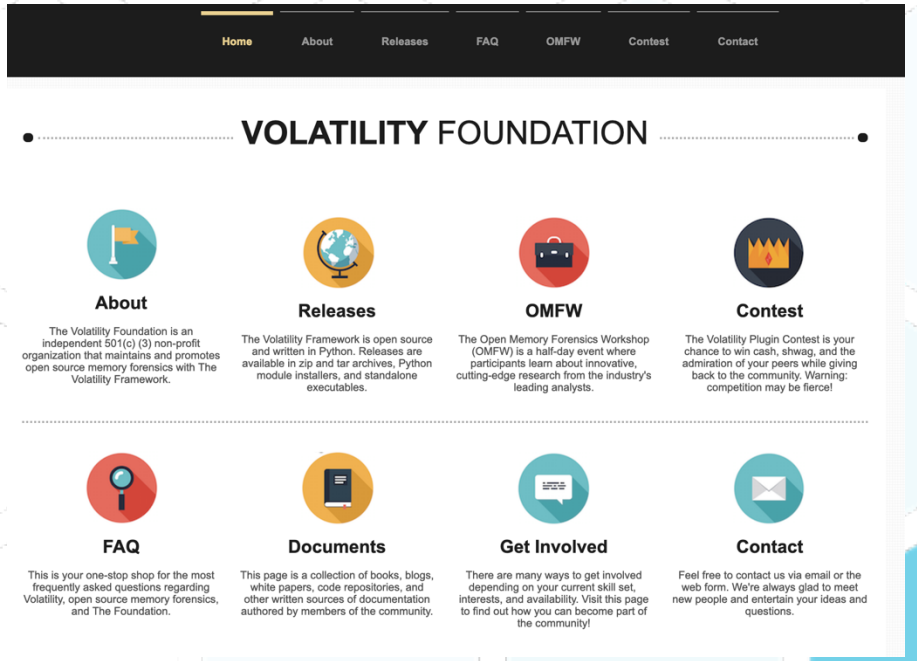
```
root@kali:~/Documents# volatility -f cridex.vmem --profile=WinXPSP2x86 procdump -p1640 --dump-dir .
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x81e7bda0 0x00400000 reader_sl.exe        OK: executable.1640.exe
root@kali:~/Documents# volatility -f cridex.vmem --profile=WinXPSP2x86 memdump -p1640 --dump-dir .
Volatility Foundation Volatility Framework 2.6
************************************************************************
Writing reader_sl.exe [  1640] to 1640.dmp
```

- Result is a file 1640.dmp
- Extract strings and save as text

$ strings 1640.dmp > 1640.txt

Other useful plug-ins

- Convert raw memory dump to dmp format
  - Raw2dump
- Extract memory cached files
  - dumpfiles
- Dump password hashes from memory
  - dummphashes
  - hashdump
- Truecrypt
  - Truecryptmaster
  - Truecryptpassphrase
  - Truecryptsummary



See https://www.volatilityfoundation.org/

Short break – 5 mins

And then Week 3 part 3:

- Explain weekly tutorial/lab