Digital Forensics
Lecture Week 6a

Memory

Processes

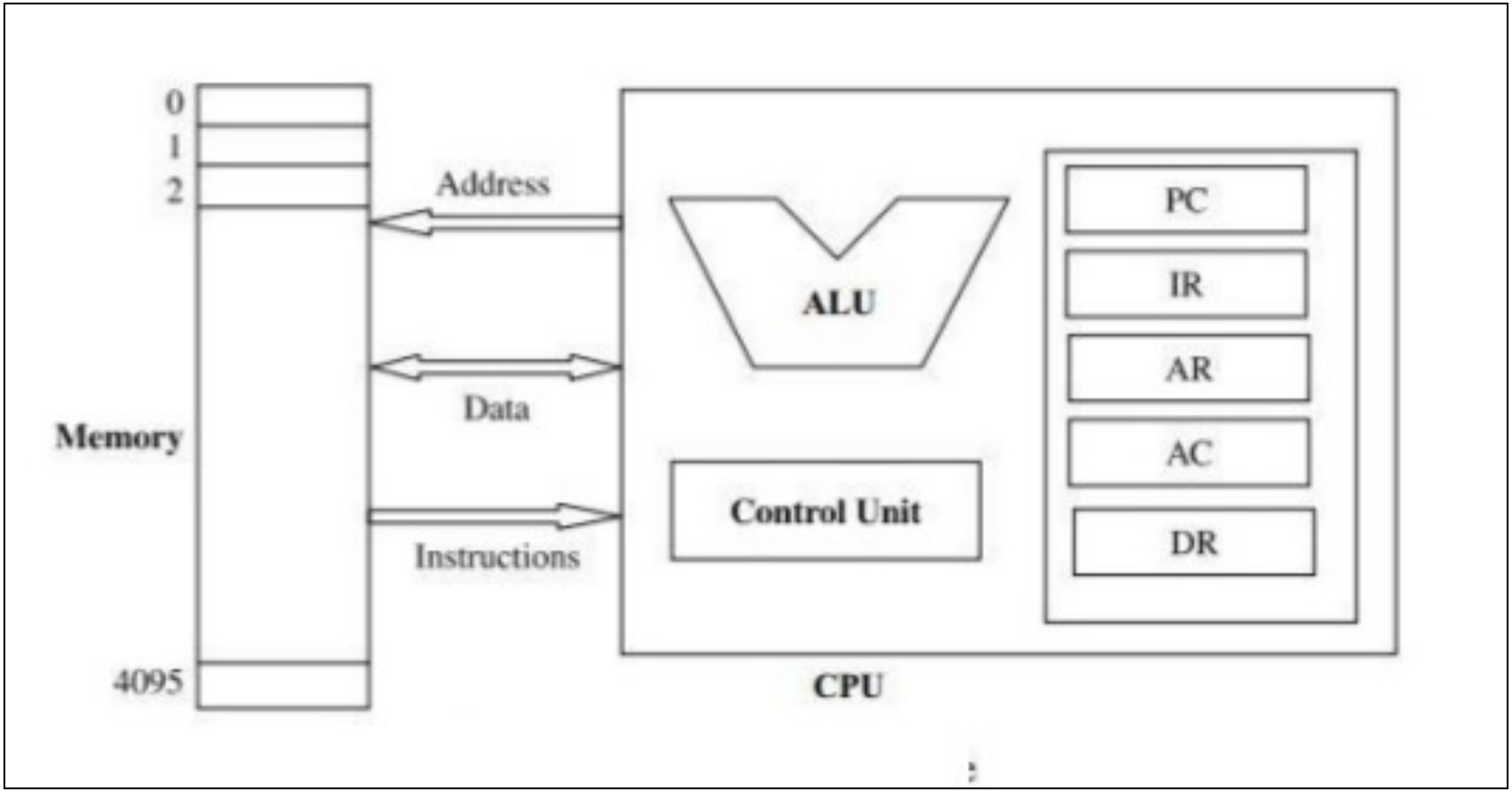
Memory Dumps

Readings

Objectives

- To revise our understanding of CPU Memory access
- To revise our understanding of Process structure
- To see how processes lead to Forensic Evidence

CPU and Memory

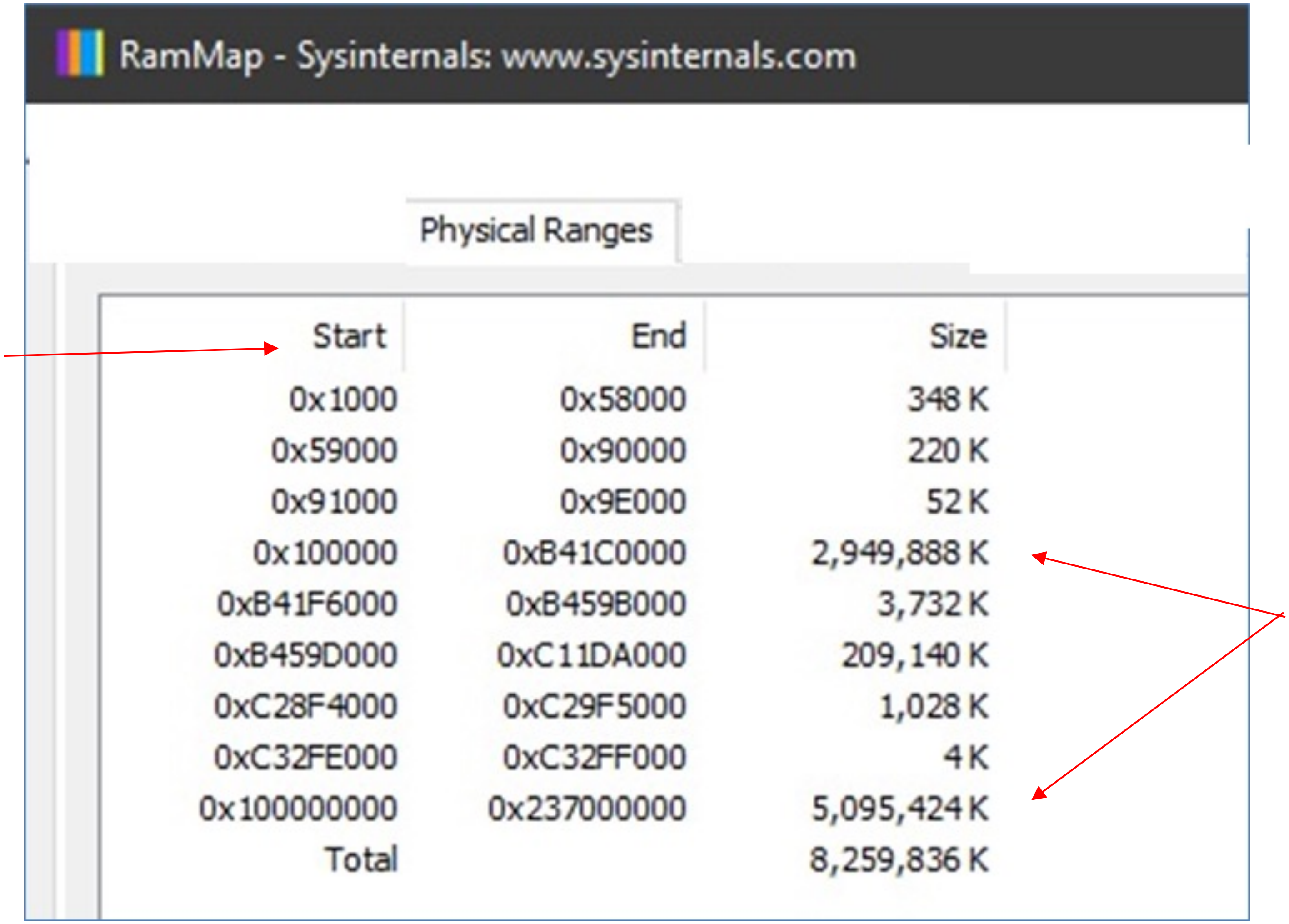


The CPU

- The Central Processing Unit executes instructions to perform actions on data
- These instructions are kept in memory as program segments
- The data is also kept in memory (data segments)
- Memory in RAM is volatile unlike disk storage
- http://en.wikipedia.org/wiki/Central_processing_unit (see operation)

Physical Memory

Physical Memory Range



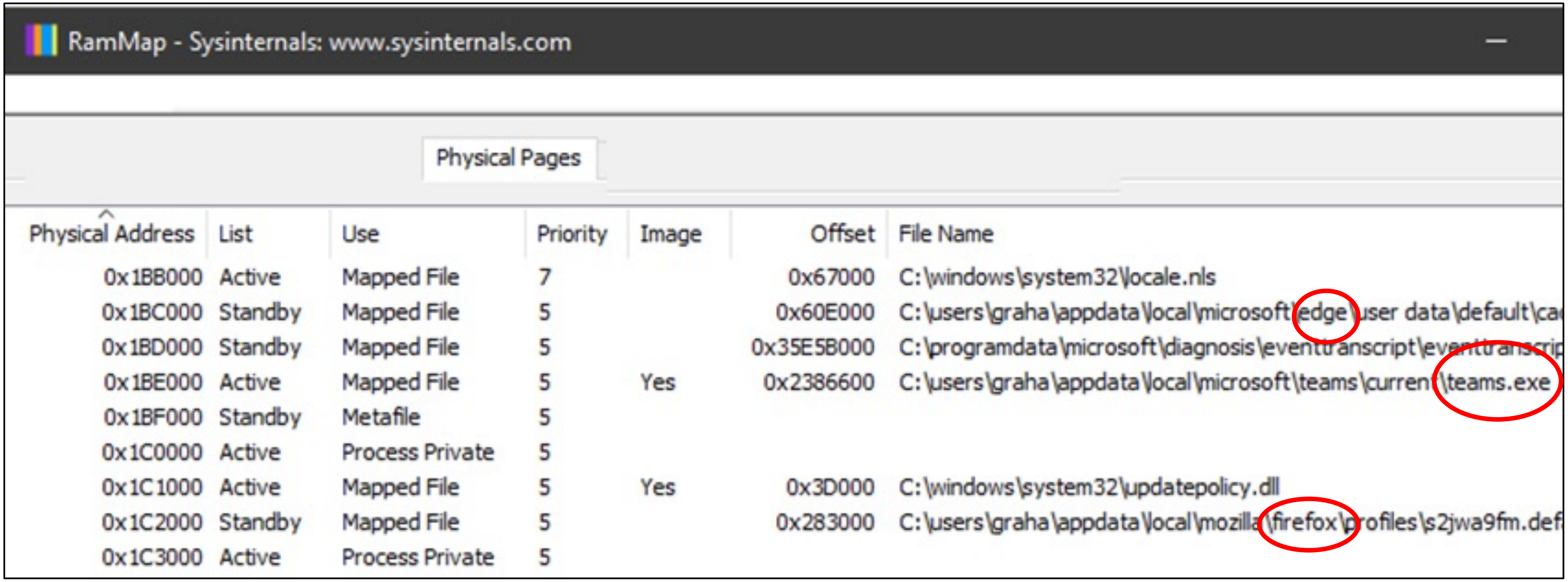
Start	End	Size
0x1000	0x58000	348 K
0x59000	0x90000	220 K
0x91000	0x9E000	52 K
0x100000	0xB41C0000	2,949,888 K
0xB41F6000	0xB459B000	3,732 K
0xB459D000	0xC11DA000	209,140 K
0xC28F4000	0xC29F5000	1,028 K
0xC32FE000	0xC32FF000	4 K
0x100000000	0x237000000	5,095,424 K
Total		8,259,836 K

Address

Busy

Physical Pages

- A file on disk can be mapped into a memory page

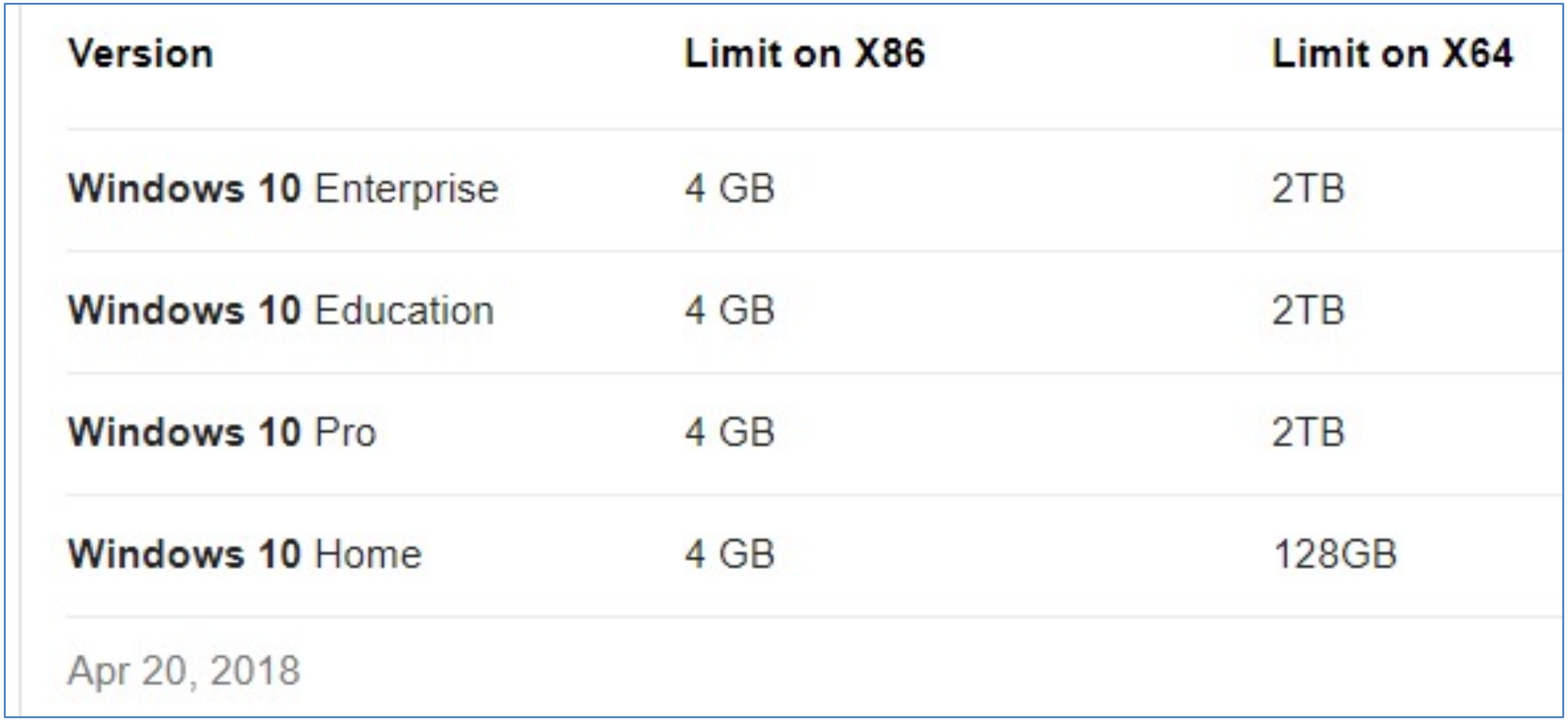


Physical Address	List	Use	Priority	Image	Offset	File Name
0x1BB000	Active	Mapped File	7		0x67000	C:\windows\system32\locale.nls
0x1BC000	Standby	Mapped File	5		0x60E000	C:\users\graha\appdata\local\microsoft\edge\user data\default\ca
0x1BD000	Standby	Mapped File	5		0x35E5B000	C:\programdata\microsoft\diagnosis\eventtranscript\eventtranscrip
0x1BE000	Active	Mapped File	5	Yes	0x2386600	C:\users\graha\appdata\local\microsoft\teams\current\teams.exe
0x1BF000	Standby	Metafile	5			
0x1C0000	Active	Process Private	5			
0x1C1000	Active	Mapped File	5	Yes	0x3D000	C:\windows\system32\updatepolicy.dll
0x1C2000	Standby	Mapped File	5		0x283000	C:\users\graha\appdata\local\mozilla\firefox\profiles\s2jwa9fm.def
0x1C3000	Active	Process Private	5			

How big is Memory?

- IA-32 Intel CPUs can access 4 GB of Memory.
- However there is a technique called Physical Address Extension (PAE) that allows access to more RAM.
- http://en.wikipedia.org/wiki/Physical_Address_Extension
- Now the OS may limit RAM as a sales incentive.
- See next slide

Windows 10 memory limits



Version	Limit on X86	Limit on X64
Windows 10 Enterprise	4 GB	2TB
Windows 10 Education	4 GB	2TB
Windows 10 Pro	4 GB	2TB
Windows 10 Home	4 GB	128GB

Apr 20, 2018

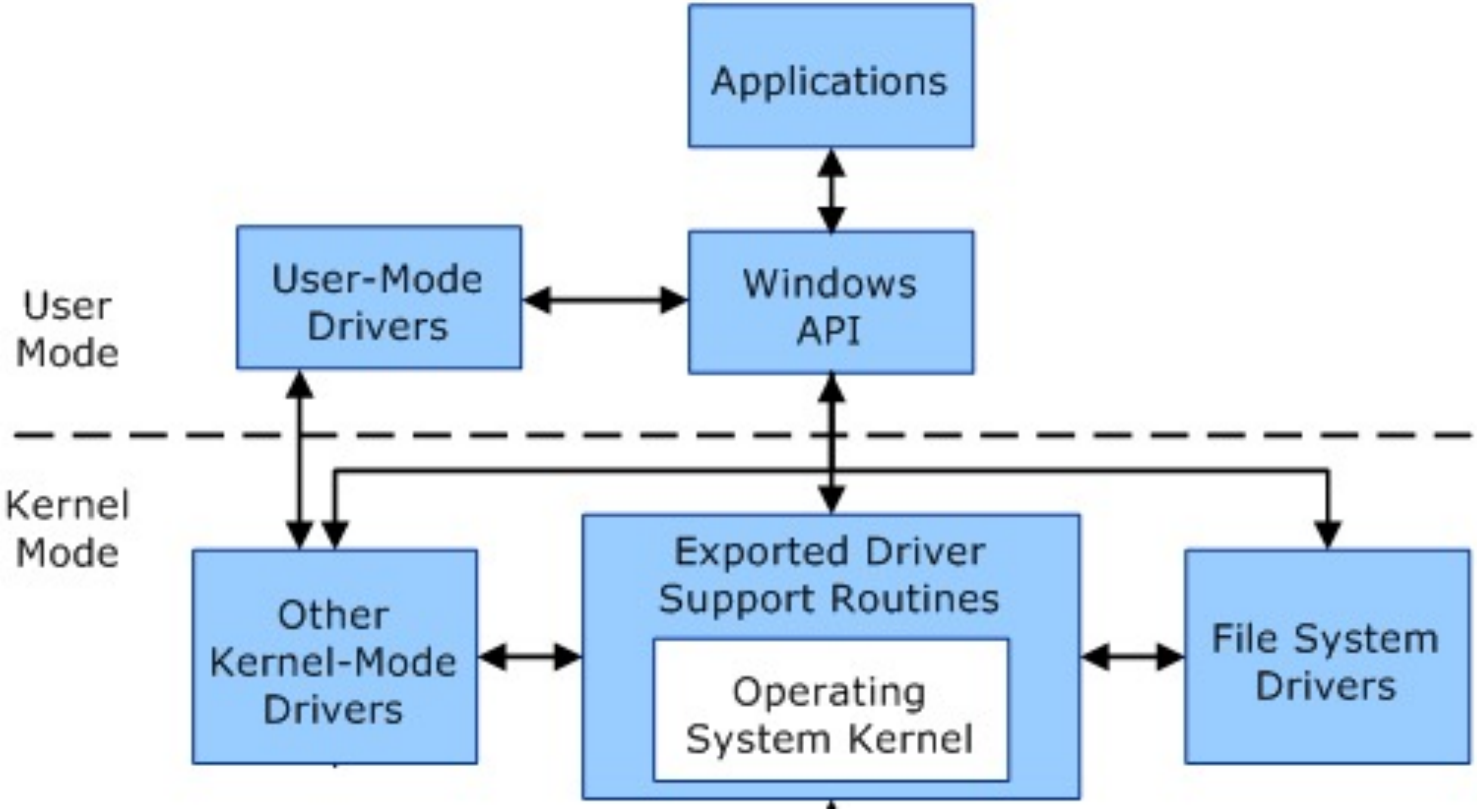
What writes to Memory?

- Hardware dependant.
- The Memory Management Unit (MMU) handles memory requests.
- http://en.wikipedia.org/wiki/Memory_management_unit
- There is also a Translation Look aside Buffer (TLB) that may hold memory data.
- http://en.wikipedia.org/wiki/Translation_lookaside_buffer
- In addition some devices like graphic cards have Direct Memory Access (DMA).
- http://en.wikipedia.org/wiki/Direct_memory_access

Operating System modes

- The core OS runs in kernel mode
  - This can access most of the RAM
  - This includes many drivers
  - All kernel mode processes can see each other's RAM
- User apps run in user mode
  - RAM access is restricted
  - Each user mode process runs in its own sand box
  - A user mode process cannot access kernel mode RAM
- http://msdn.microsoft.com/en-us/library/windows/hardware/ff554836(v=vs.85).aspx

Operating System modes #2

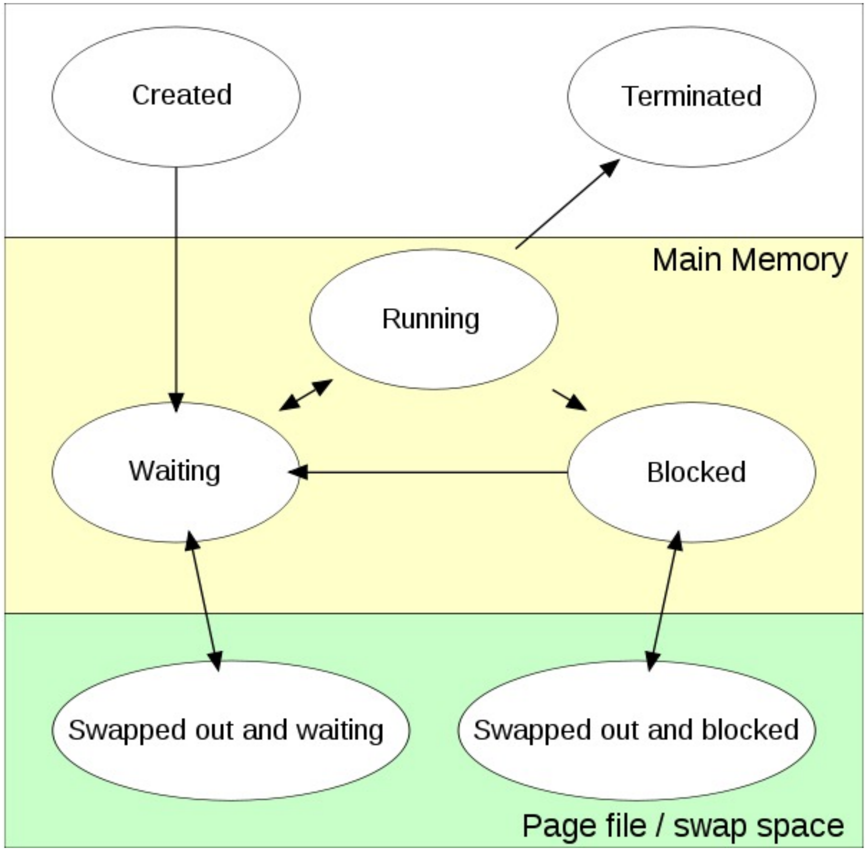


Objectives

- To revise our understanding of Memory access
- To revise our understanding of Process structure
- To see how processes lead to Forensic Evidence

Processes

- A process is a running program
- Launched from an exe
- Every task in a PC runs as a process
- Forensics examines processes to locate evidence



http://en.wikipedia.org/wiki/Process_(computing)

Data Structures in Memory

- To recover information from memory we need to know how it is stored
- A memory tool understands the many methods used.
- Arrays – usually of a fixed size
- Bit Maps – sparse arrays (example tcp ports in use)
- Records – name:value pairs
- Strings – often 00 terminated
- Linked lists
- Hash Tables
- Hierarchical Trees

Anti Forensics

- Address space layout randomization (ASLR)
- a technique to reduce memory hacking.
- Prevent an attacker from reliably jumping to an address in memory
- ASLR randomly arranges the address space positions of key data areas of a process

Process memory footprint

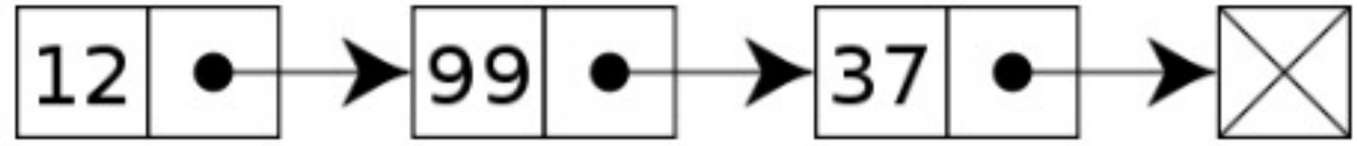
- Each process has artifacts that identify it in RAM
  - open file handles
  - recent dlls used
  - memory mappings
  - network connections (sockets)
  - privileges

Process Source

- Task Manager reveals many processes
  - how did they start?
  - who published them?
  - when were they written?
- We can see running processes with:
- Tasklist (Built-in Windows)
- PsList (SysInternals)

The Linked List

- Task Manager keeps track of processes (tasks)
- To do so, it uses a linked list of nodes

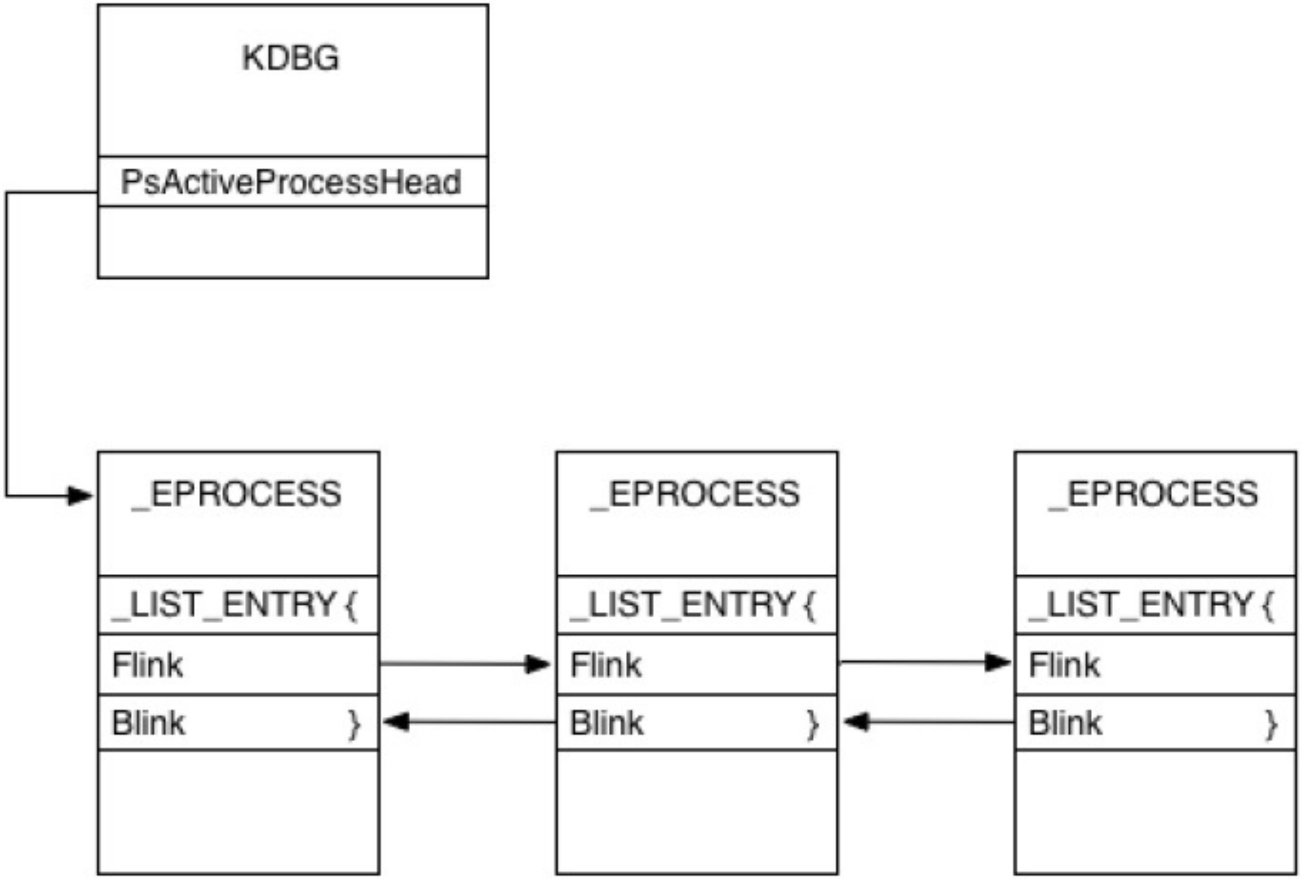


- Each node in the list has a value and a pointer to the next node
- The last node is linked to a terminator
- http://en.wikipedia.org/wiki/Linked_list

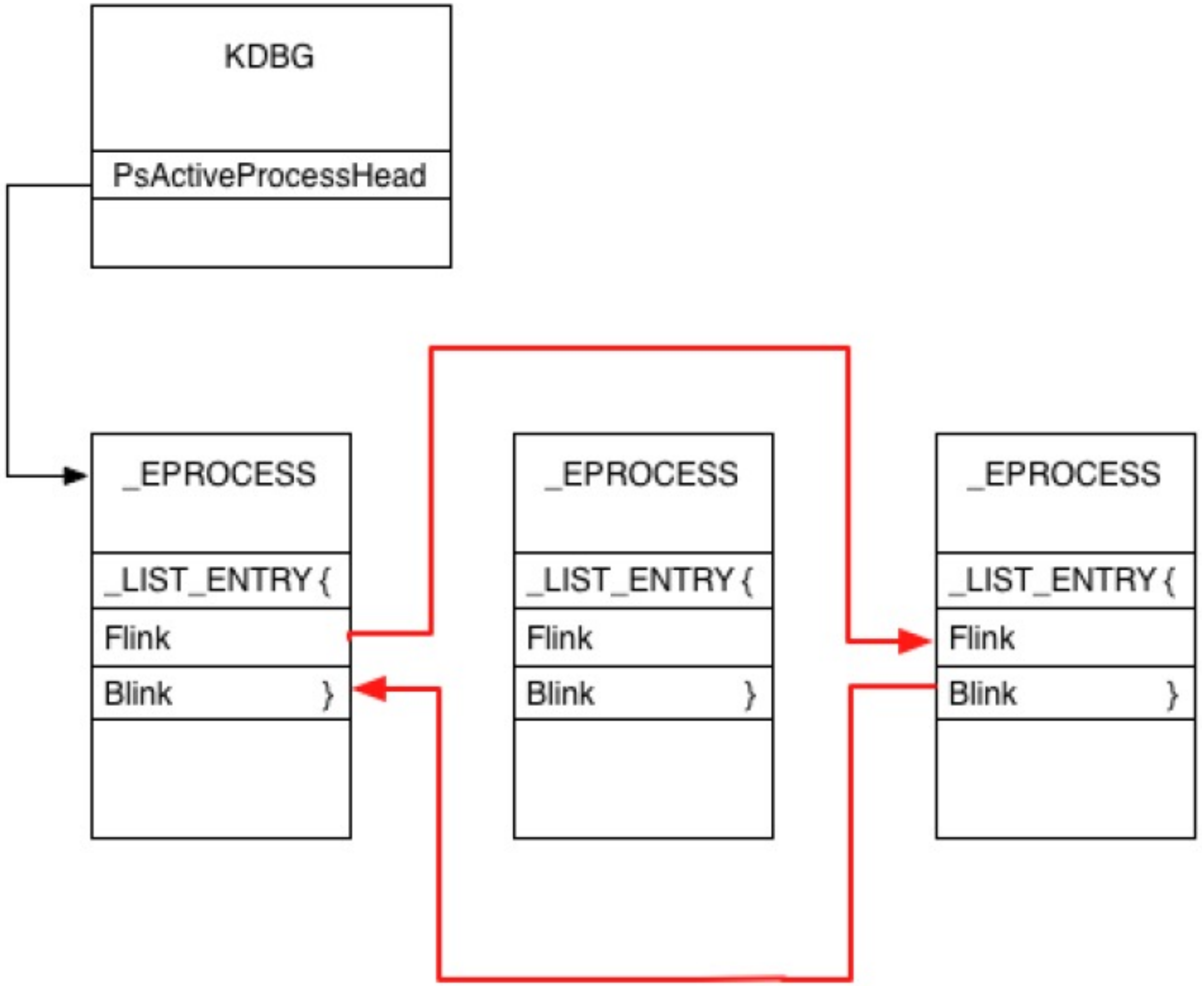
Listing processes

- Task Manager displays a list of processes
  - Starting at PsActiveProcessHead
  - Then links to each _EPROCESS structure
  - Active processes are displayed
- The Executive Process list has more processes
  - Active, Hidden, Deleted
- Some tools can dump all these
  - A virus can hide an evil process by manipulating the list

Walking the list
using forward and backward links



A rooted list – unlisted process



Executable file process

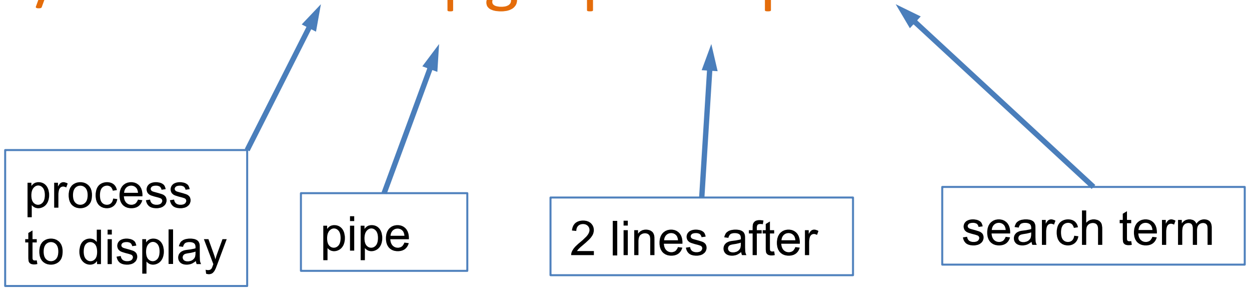
- The linked program is compiled into an exe
- When the exe is clicked, a dynamic linker reads the exe file and loads its pieces into memory
- The linker links the file dll calls into the running dlls loaded in memory
- Code is loaded into a read only, executable region
- Data is loaded into a data region

Windows dlls

- A Dynamic Link Library (dll) is a piece of code that can be shared by one or more processes
- Windows has thousands of dlls stored on disk
- Difficult to spot a dll introduced by malware
- Worse still, malware can alter an existing dll
  - Can be detected by examining the dll hash
- We can view running dlls with
  - Listdlls (SysInternals)
  - Tasklist (Windows)

Viewing a process source file

- Listdlls shows how a process was launched
- We use grep to filter out the lines of interest
- ./Listdlls cmd | grep -A2 pid

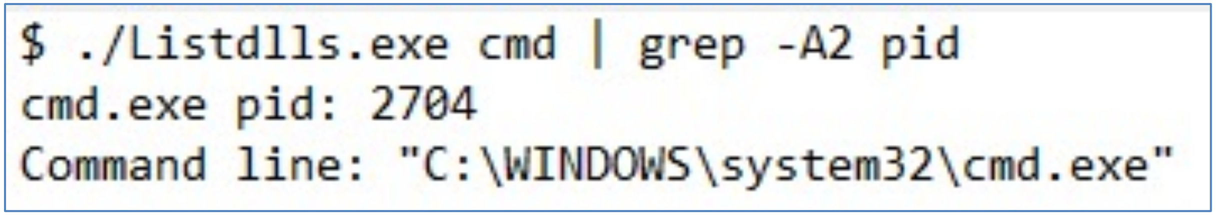


process to display

pipe

2 lines after

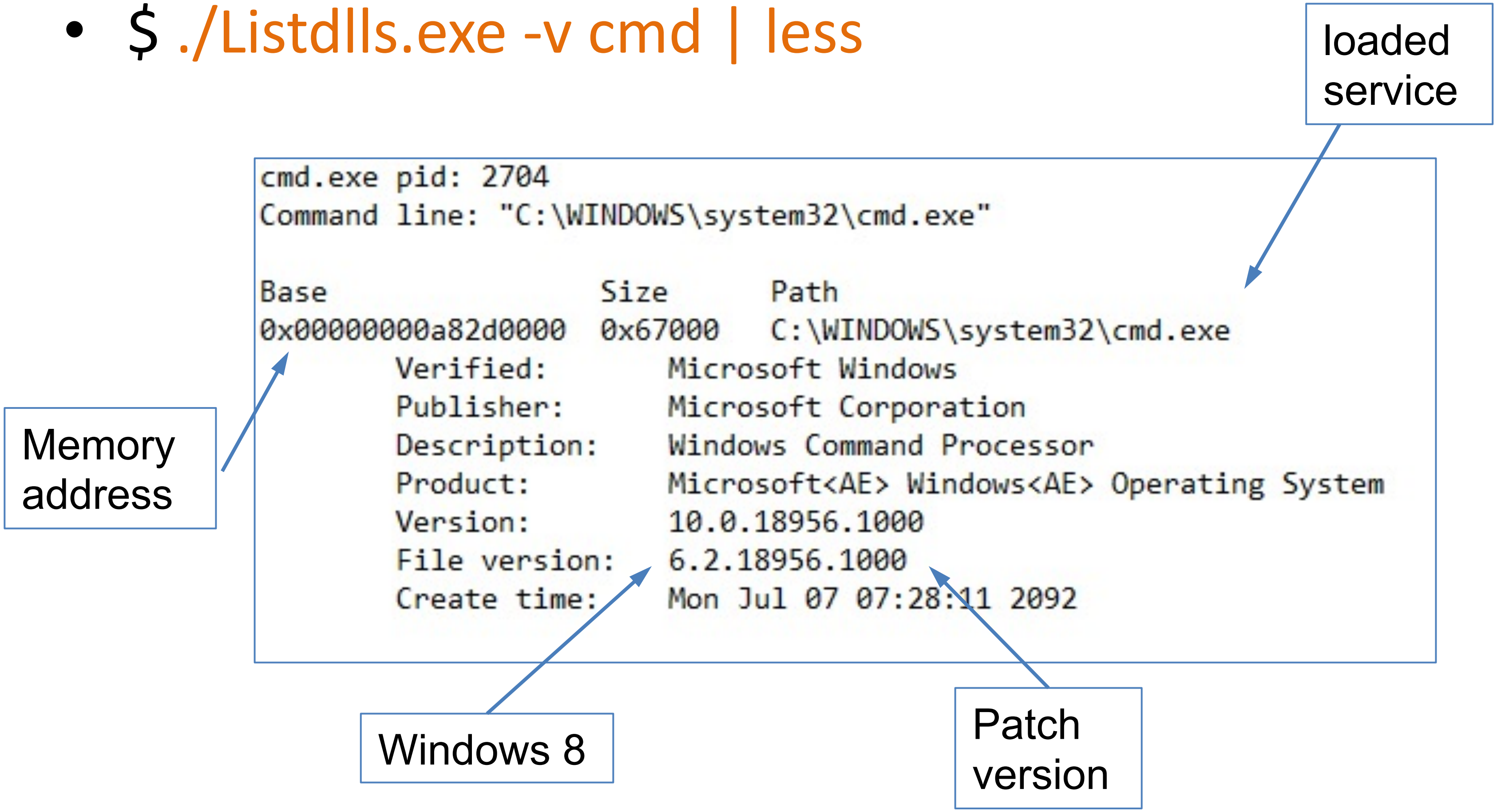
search term



```
$ ./Listdlls.exe cmd | grep -A2 pid
cmd.exe pid: 2704
Command line: "C:\WINDOWS\system32\cmd.exe"
```

Viewing dll version detail

- $ ./Listdlls.exe -v cmd | less



```
cmd.exe pid: 2704
Command line: "C:\WINDOWS\system32\cmd.exe"

Base                Size      Path
0x00000000a82d0000  0x67000   C:\WINDOWS\system32\cmd.exe
        Verified:       Microsoft Windows
        Publisher:      Microsoft Corporation
        Description:    Windows Command Processor
        Product:        Microsoft<AE> Windows<AE> Operating System
        Version:        10.0.18956.1000
        File version:   6.2.18956.1000
        Create time:    Mon Jul 07 07:28:11 2092
```

loaded service

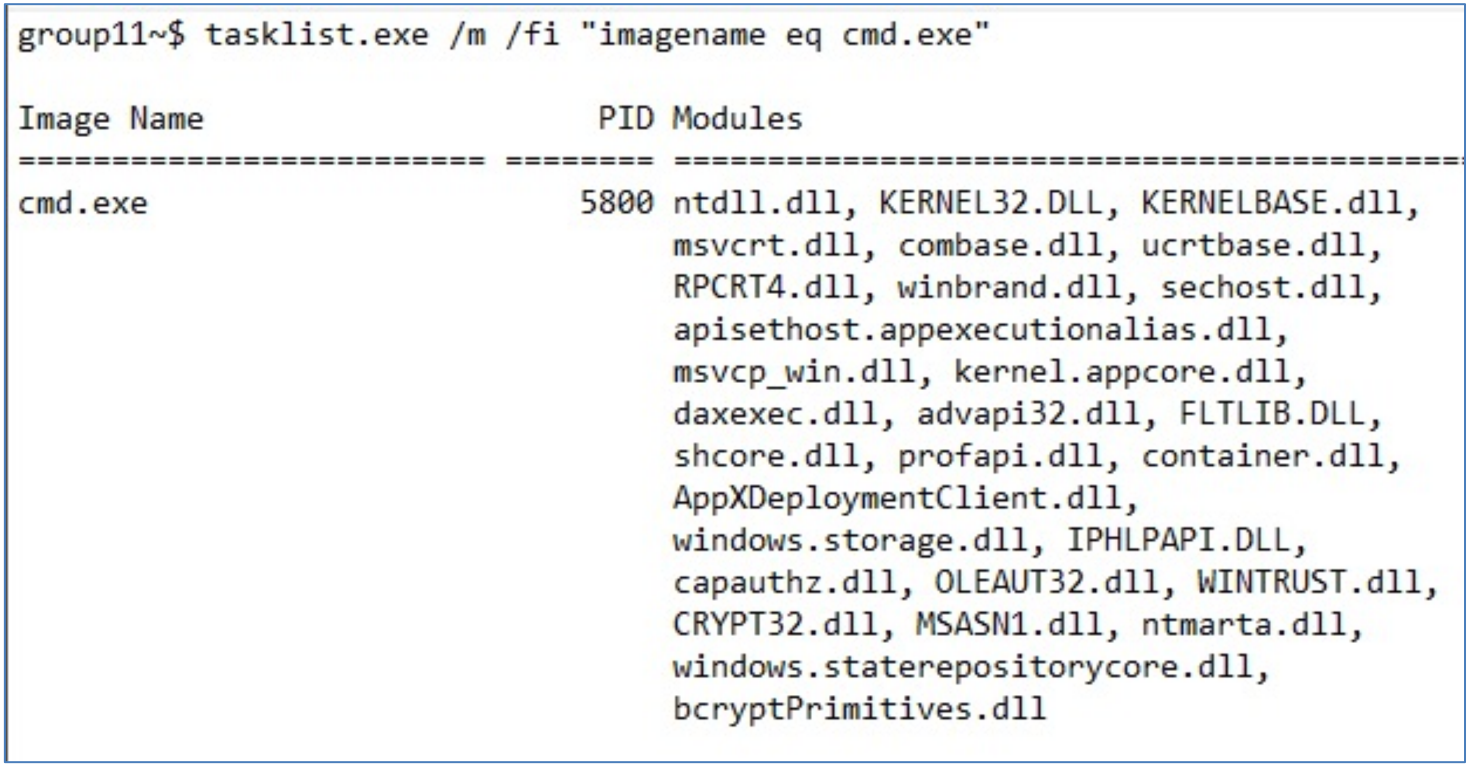
Memory address

Windows 8

Patch version

Viewing Windows dlls #2

- Viewing dlls with TaskList
- tasklist /m /fi "imagename eq cmd.exe"
  - the /m option lists modules (dlls)
  - the /fi option filters by name or PID



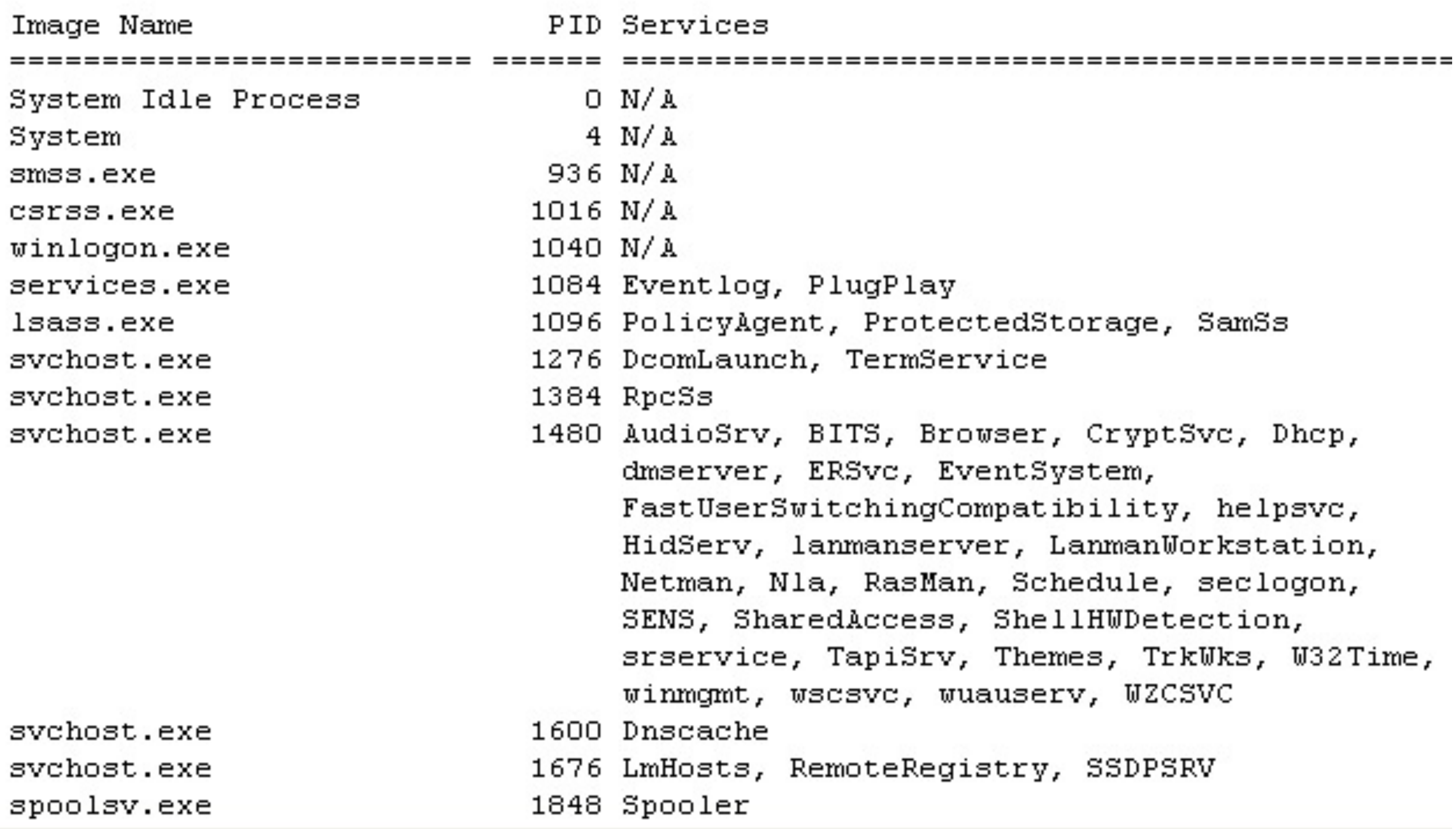
```
group11~$ tasklist.exe /m /fi "imagename eq cmd.exe"

Image Name                     PID Modules
========================= ======== ============================================
cmd.exe                       5800 ntdll.dll, KERNEL32.DLL, KERNELBASE.dll,
                                   msvcrt.dll, combase.dll, ucrtbase.dll,
                                   RPCRT4.dll, winbrand.dll, sechost.dll,
                                   apisethost.appexecutionalias.dll,
                                   msvcp_win.dll, kernel.appcore.dll,
                                   daxexec.dll, advapi32.dll, FLTLIB.DLL,
                                   shcore.dll, profapi.dll, container.dll,
                                   AppXDeploymentClient.dll,
                                   windows.storage.dll, IPHLPAPI.DLL,
                                   capauthz.dll, OLEAUT32.dll, WINTRUST.dll,
                                   CRYPT32.dll, MSASN1.dll, ntmarta.dll,
                                   windows.staterepositorycore.dll,
                                   bcryptPrimitives.dll
```

Services

- Services are long running processes
- They have no user interface
- Many services start automatically at boot
- Similar to daemons in Linux
- Some services are used for networking
  - Webclient
  - Remote Procedure Calls (rpc)
- Services can be run by Service Host Processes
  - svchost.exe
- http://en.wikipedia.org/wiki/Windows_services

Services

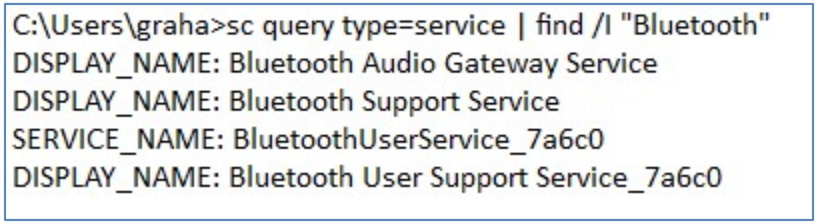
- To see processes running Services, use TaskList.exe /svc



```
Image Name                   PID Services
========================= ====== =============================================
System Idle Process            0 N/A
System                         4 N/A
smss.exe                     936 N/A
csrss.exe                   1016 N/A
winlogon.exe                1040 N/A
services.exe                1084 Eventlog, PlugPlay
lsass.exe                   1096 PolicyAgent, ProtectedStorage, SamSs
svchost.exe                 1276 DcomLaunch, TermService
svchost.exe                 1384 RpcSs
svchost.exe                 1480 AudioSrv, BITS, Browser, CryptSvc, Dhcp,
                                 dmserver, ERSvc, EventSystem,
                                 FastUserSwitchingCompatibility, helpsvc,
                                 HidServ, lanmanserver, LanmanWorkstation,
                                 Netman, Nla, RasMan, Schedule, seclogon,
                                 SENS, SharedAccess, ShellHWDetection,
                                 srservice, TapiSrv, Themes, TrkWks, W32Time,
                                 winmgmt, wscsvc, wuauserv, WZCSVC
svchost.exe                 1600 Dnscache
svchost.exe                 1676 LmHosts, RemoteRegistry, SSDPSRV
spoolsv.exe                 1848 Spooler
```

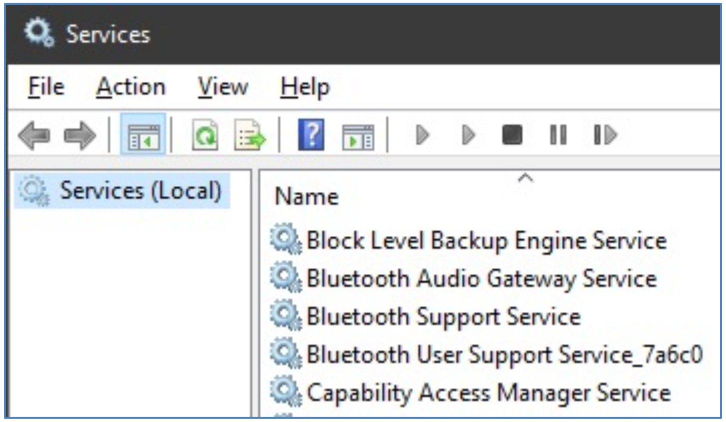
More ways to see services

- Use the Service Controller SC
- SC query type=service



```
C:\Users\graha>sc query type=service | find /I "Bluetooth"
DISPLAY_NAME: Bluetooth Audio Gateway Service
DISPLAY_NAME: Bluetooth Support Service
SERVICE_NAME: BluetoothUserService_7a6c0
DISPLAY_NAME: Bluetooth User Support Service_7a6c0
```

- Use the services snap in
- Services.msc



Block Level Backup Engine Service
Bluetooth Audio Gateway Service
Bluetooth Support Service
Bluetooth User Support Service_7a6c0
Capability Access Manager Service

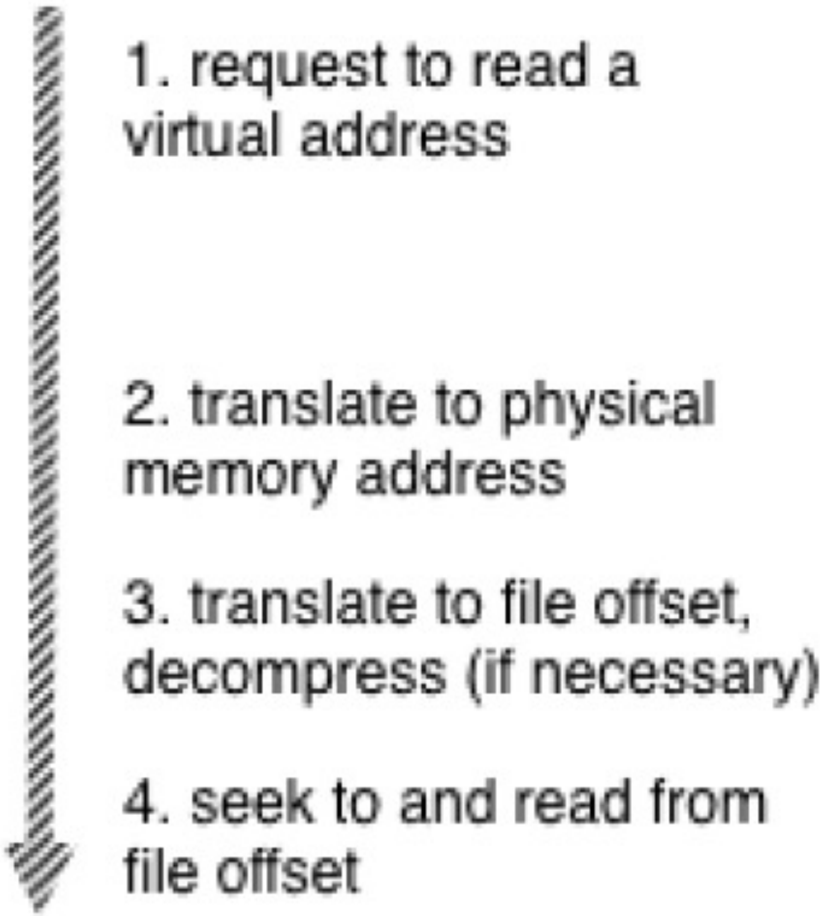
Objectives

- To revise our understanding of Memory access
- To revise our understanding of Process structure
- To see how processes lead to Forensic Evidence

Windows Memory

- Memory accesses are far faster than disk accesses
- A process opens the files it requires and places the contents in memory
- It decodes encryption (ssl and vpn) in memory
- Passwords are also placed in memory
- Memory dumping is an important forensic activity
- However memory addressing is complicated and requires specialised tools

Memory addressing



1. request to read a virtual address

2. translate to physical memory address

3. translate to file offset, decompress (if necessary)

4. seek to and read from file offset

Windows Memory #2

- Memory data may be:
  - Incomplete
  - Randomly organised
  - Partly overwritten
  - repeated in different locations
  - changed by memory managers at any instant

Dumping all Memory

- Memory can be quite large, 8 – 24 GB
- So we need 8 – 24 GB disk space for the dump
- Do not dump onto the system disk as this may upset paging and swap files (see later)
- The act of dumping may interfere with Memory Managers
- To dump all of memory, we visit System, About, Advanced, Startup and Recovery

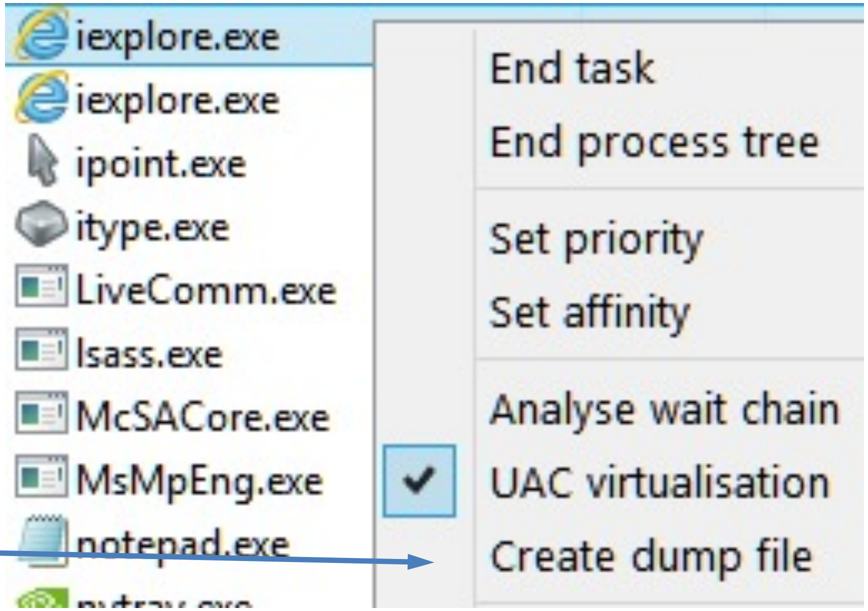
Searching all memory

- We use the Volatility tool add-on for Python
- Volatility can analyse memory dumps from Windows, Linux, MAC OSX and Android ARM
- Volatility can recover process lists, network connections, passwords and web sessions

- https://github.com/volatilityfoundation/volatility3
- Their reference text is excellent
- http://www.amazon.com/gp/product/1118825098/

Dumping Process Memory

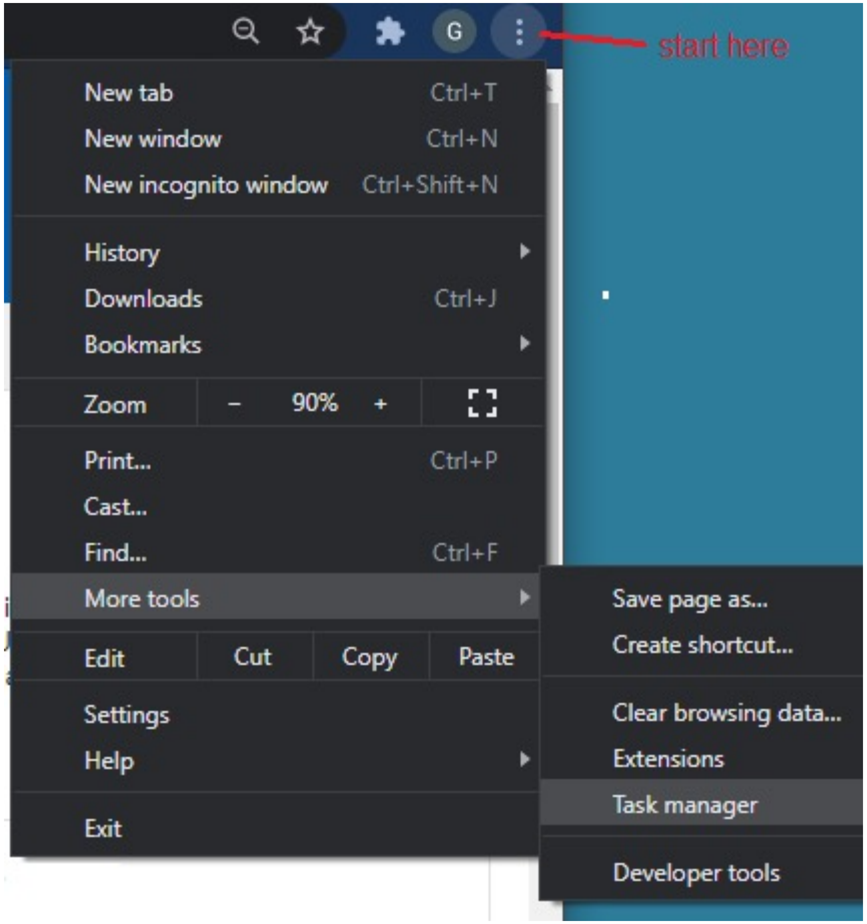
- We can dump a process in Windows 10 using task manager



Dumping a chrome process #1

- Open chrome.
- Select the target website

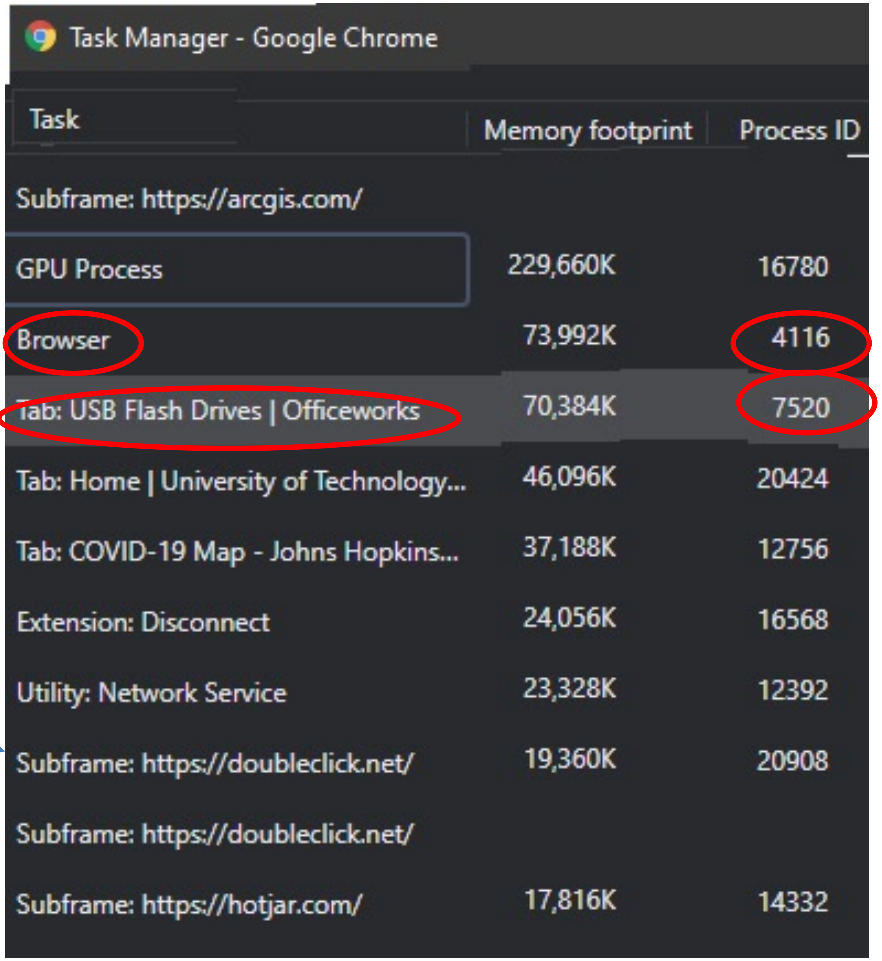
- Open chrome task manager



Dumping a chrome process #2

- Note the PIDs you want
  - Browser
  - Your tab

- Note the tracker processes

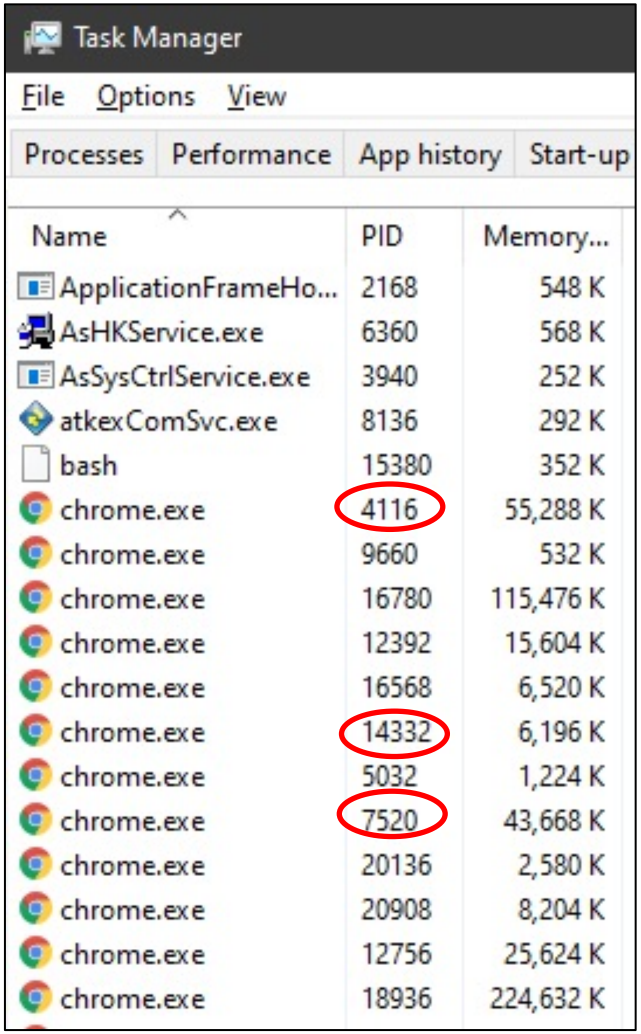


Task Manager - Google Chrome

Task	Memory footprint	Process ID
Subframe: https://arcgis.com/		
GPU Process	229,660K	16780
Browser	73,992K	4116
Tab: USB Flash Drives | Officeworks	70,384K	7520
Tab: Home | University of Technology...	46,096K	20424
Tab: COVID-19 Map - Johns Hopkins...	37,188K	12756
Extension: Disconnect	24,056K	16568
Utility: Network Service	23,328K	12392
Subframe: https://doubleclick.net/	19,360K	20908
Subframe: https://doubleclick.net/		
Subframe: https://hotjar.com/	17,816K	14332

Dumping a chrome process #3

- Open Windows Task Manager
- Dump the processes
  - Matching the PID you want.



Name	PID	Memory...
ApplicationFrameHo...	2168	548 K
AsHKService.exe	6360	568 K
AsSysCtrlService.exe	3940	252 K
atkexComSvc.exe	8136	292 K
bash	15380	352 K
chrome.exe	4116	55,288 K
chrome.exe	9660	532 K
chrome.exe	16780	115,476 K
chrome.exe	12392	15,604 K
chrome.exe	16568	6,520 K
chrome.exe	14332	6,196 K
chrome.exe	5032	1,224 K
chrome.exe	7520	43,668 K
chrome.exe	20136	2,580 K
chrome.exe	20908	8,204 K
chrome.exe	12756	25,624 K
chrome.exe	18936	224,632 K

Searching Process Memory

- We use strings to extract text in the binary dump
- strings chrome.dmp > chrome.txt
- We search the text file using grep
- grep passwd chrome.txt
  - looks for login passwords
- grep Set-Cookie chrome.txt
  - looks for cookies from websites

Memory search filters

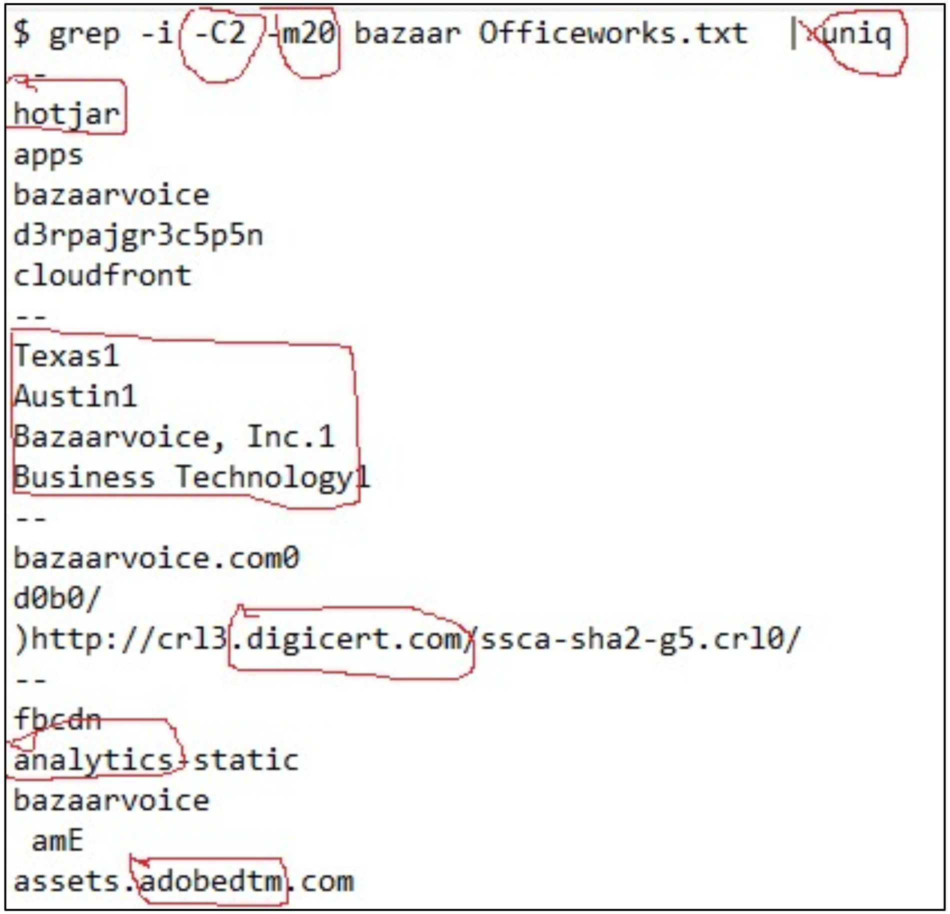
- Memory is disorganised
- You will see lots of unwanted hits (noise).
- Use filters to combat noise

-
  ```
  $ grep -i bazaar chrome.txt  | cut -c 1-120 | grep -i officew –m20 | uniq
  ```

Comparing packet capture and memory

- Wireshark text dump

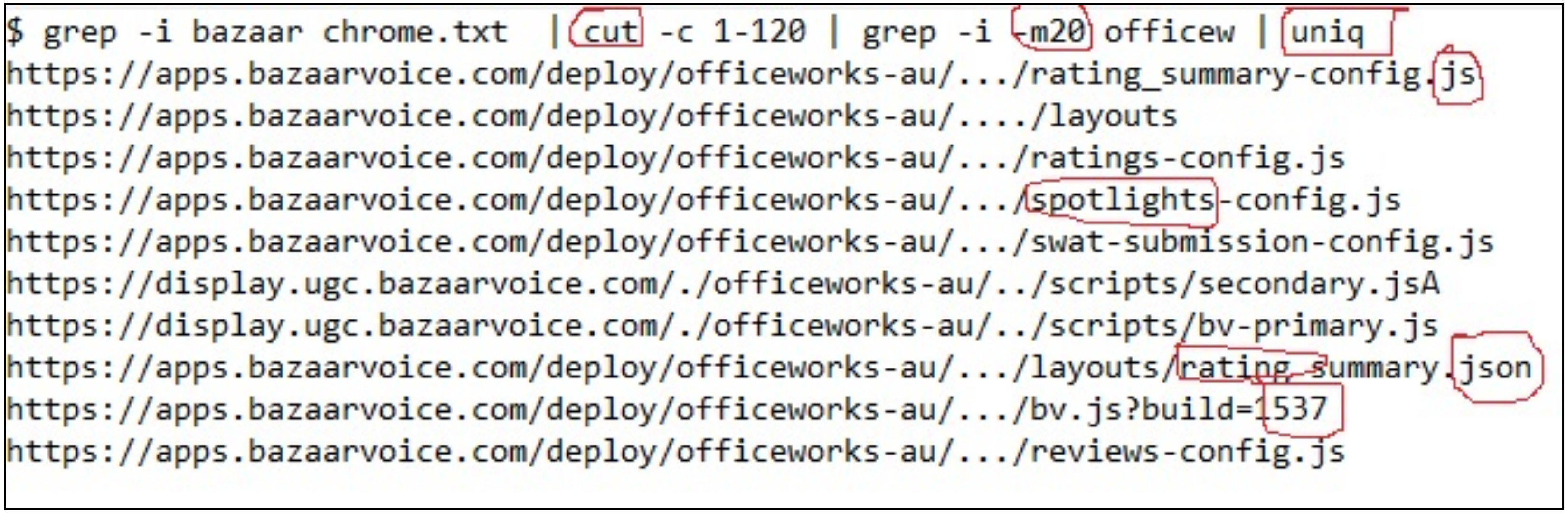
Shows third party website pages



```
$ grep -i -C2 -m20 bazaar Officeworks.txt  |uniq
--
hotjar
apps
bazaarvoice
d3rpajgr3c5p5n
cloudfront
--
Texas1
Austin1
Bazaarvoice, Inc.1
Business Technology1
--
bazaarvoice.com0
d0b0/
)http://crl3.digicert.com/ssca-sha2-g5.crl0/
--
fbcdn
analytics-static
bazaarvoice
 amE
assets.adobedtm.com
```

Comparing packet capture and memory

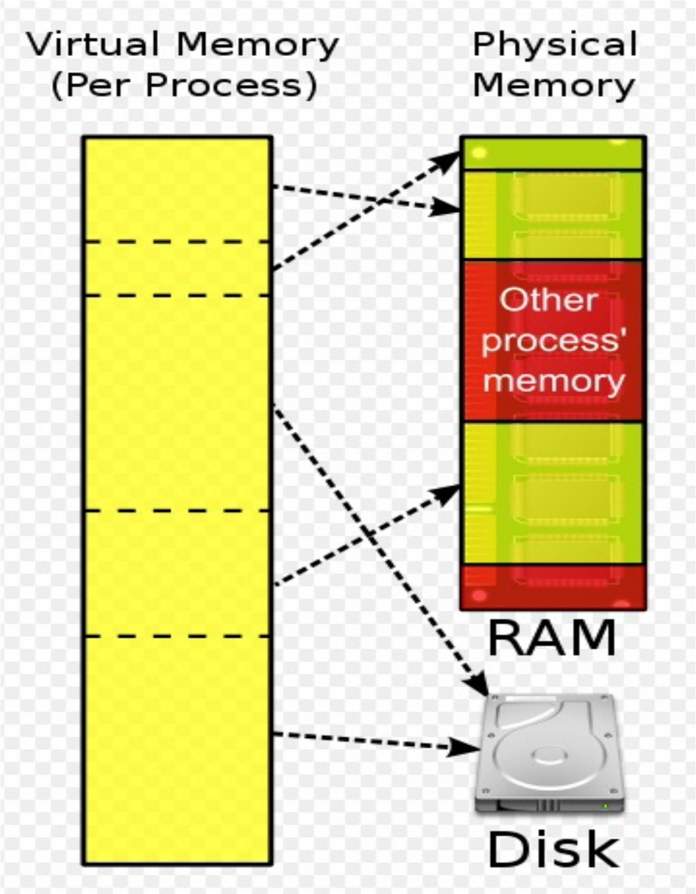
- Memory text dump
  - See running JavaScript on the client



```
$ grep -i bazaar chrome.txt  | cut -c 1-120 | grep -i -m20 officew | uniq
https://apps.bazaarvoice.com/deploy/officeworks-au/.../rating_summary-config.js
https://apps.bazaarvoice.com/deploy/officeworks-au/..../layouts
https://apps.bazaarvoice.com/deploy/officeworks-au/.../ratings-config.js
https://apps.bazaarvoice.com/deploy/officeworks-au/.../spotlights-config.js
https://apps.bazaarvoice.com/deploy/officeworks-au/.../swat-submission-config.js
https://display.ugc.bazaarvoice.com/./officeworks-au/../scripts/secondary.jsA
https://display.ugc.bazaarvoice.com/./officeworks-au/../scripts/bv-primary.js
https://apps.bazaarvoice.com/deploy/officeworks-au/.../layouts/rating_summary.json
https://apps.bazaarvoice.com/deploy/officeworks-au/.../bv.js?build=1537
https://apps.bazaarvoice.com/deploy/officeworks-au/.../reviews-config.js
```

Virtual Memory

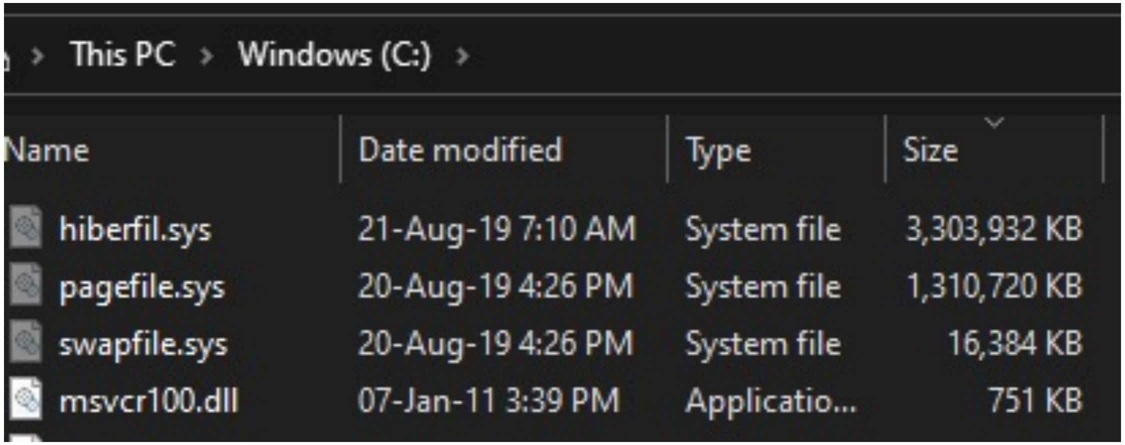
- The CPU would like to access all its programs in RAM
- However there is not enough RAM and it is volatile
- So unused RAM is swapped to disk files
-



http://en.wikipedia.org/wiki/Virtual_memory

Memory on Disk

- We can recover memory from disk!
  - virtual memory page files (25% of RAM)
  - hibernation files (75% of RAM)
  - windows 10 swap file
  - crash files

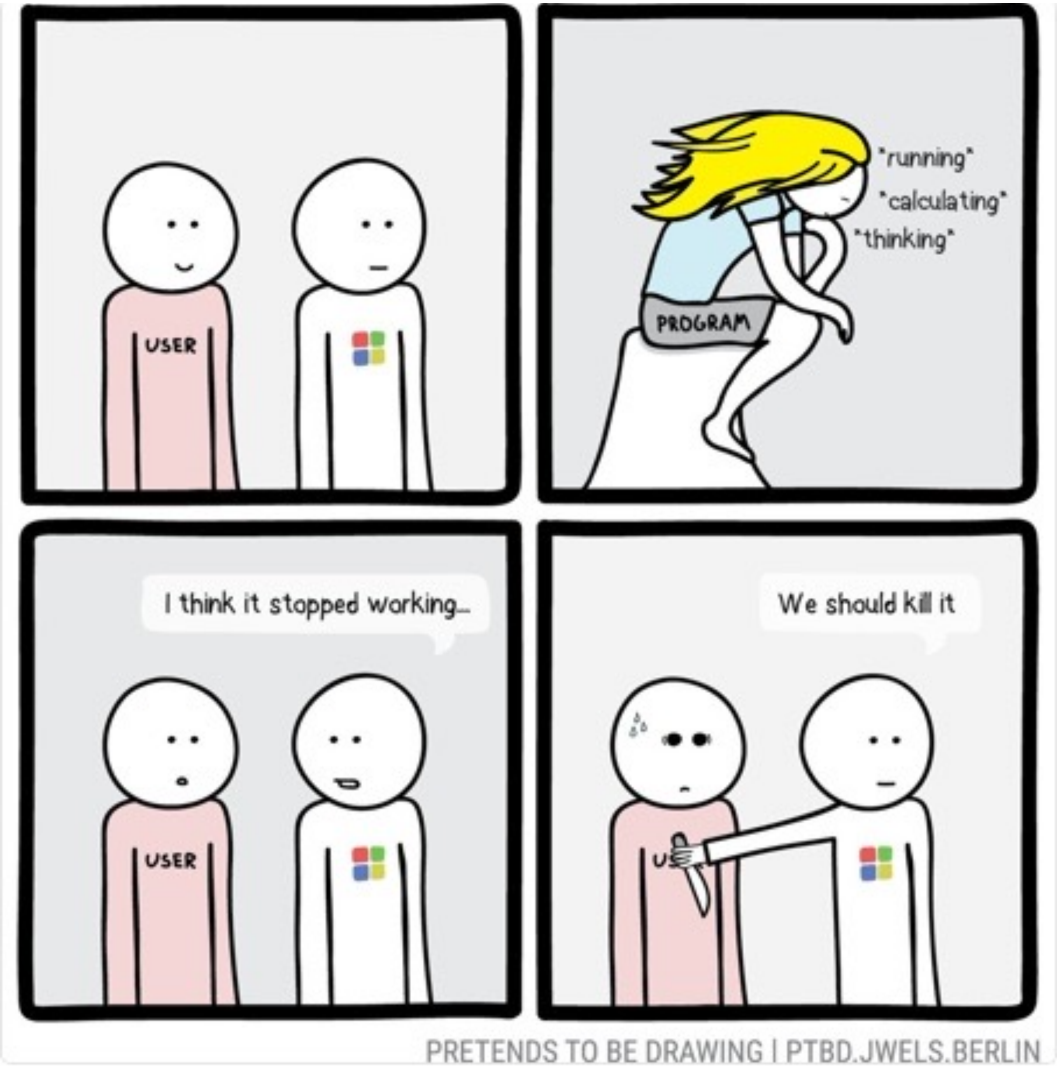


Name	Date modified	Type	Size
hiberfil.sys	21-Aug-19 7:10 AM	System file	3,303,932 KB
pagefile.sys	20-Aug-19 4:26 PM	System file	1,310,720 KB
swapfile.sys	20-Aug-19 4:26 PM	System file	16,384 KB
msvcr100.dll	07-Jan-11 3:39 PM	Applicatio...	751 KB

Memory – more things to look for

- Memory may also contain:
  - Parts of the Windows Registry
  - Parts of the Disk File Table
  - Terminated processes
  - Malware

FIN


USER
PROGRAM
running
calculating
thinking
USER
I think it stopped working...
US
We should kill it
PRETENDS TO BE DRAWING | PTBD.JWELS.BERLIN