
# Windows Operating System Agnostic Memory Analysis

*By*

**James Okolica and Gilbert Peterson**

*Presented At*

The Digital Forensic Research Conference

**DFRWS 2010 USA** Portland, OR (Aug 2nd - 4th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment. As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

**http:/dfrws.org**

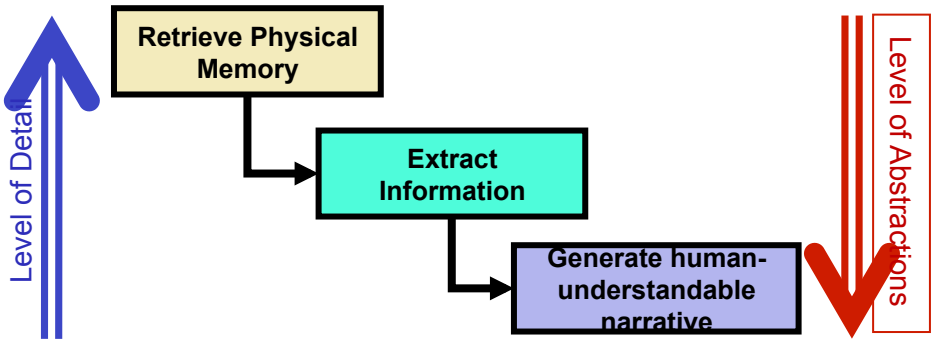
# Windows Operating Systems Agnostic Memory Analysis

James Okolica
Gilbert Peterson

# Overview



- A Process for Forensic Analysis of Memory

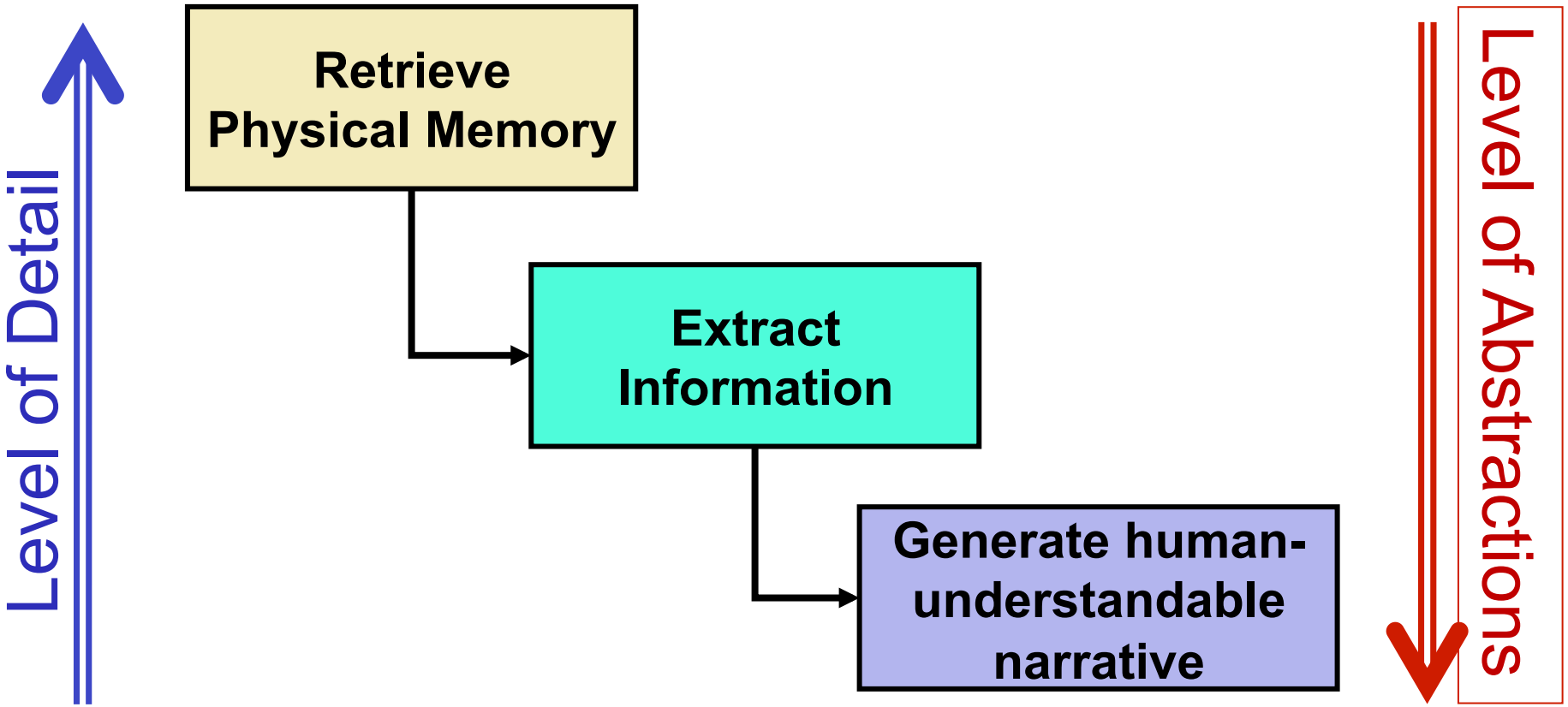



- Windows Operating System Agnostic Memory Analysis
  - Functionality
  - Experimental Set up and Results

- Next Steps

# Forensic Analysis of Memory





# Forensic Analysis of Memory

Level of Detail

**Retrieve Physical Memory**

- Hardware based acquisition
- Virtual Machines
- Hibernation File
- Operating System Patches
- Software (Memoryze, MDD, Win32dd/ Win64dd)

Level of Abstractions

# Forensic Analysis of Memory




**Extract Information**

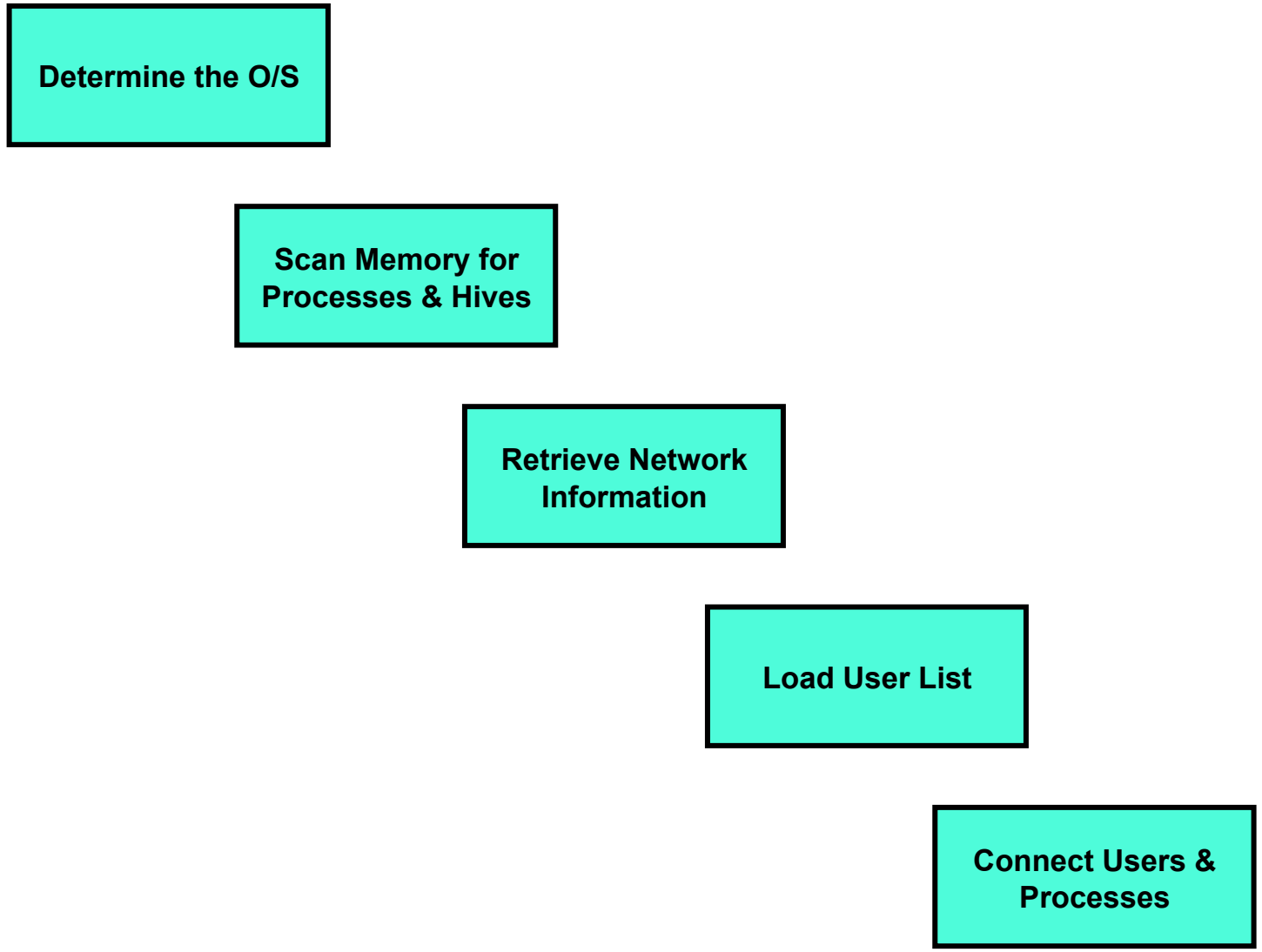
- Memoryze (freeware)
- Volatility (open source in Python)
- CMAT (open source in C++)

# CMAT Capabilities

- Programmatically determine O/S version
  - Physical Address Extensions enabled/disabled, 32-bit/64-bit
- Load O/S specific data structures
  - O/S-agnostic signatures for processes
- Locate data structures within PEs
  - O/S-agnostic retrieval of network activity
- Process Information
  - Network activity, objects accessed, modules loaded
- Configuration information
  - The users with accounts, application configurations
- Retrieve memory swapped to disk, memory-mapped files

# CMAT – The Process of Extraction





# CMAT – Determine the O/S




- Scan memory for _DBGKD_DEBUG_DATA_HEADER64
  - OwnerTag = KDBG, Size less than 4K

| _DBGKD_DEBUG_DATA_HEADER64 | | |
|---|---|---|
| 0x00 | List_Entry64 | List |
| 0x10 | ULong | OwnerTag |
| 0x14 | ULong | Size |

| _KDDEBUGGER_DATA64 | | |
|---|---|---|
| 0x00 | _DBGKD_DEBUG_DATA_HEADER64 | Header |
| 0x18 | ULong64 | KernBase |
| 0x20 | ULong64 | BreakPointwithStatus |
| 0x28 | ULong64 | SavedContext |
| 0x30 | UShort | ThCallBackStack |
| 0x32 | UShort | NextCallBack |
| 0x34 | UShort | FramePointer |
| 0x36 | UShort | PAEEnabled:1 |
| … | | |
| 0x48 | ULong64 | PSLoadedModuleList |
| … | | |
| 0xA0 | ULong64 | OBTypeObjectType |

| _DBGKD_GET_VERSION64 | | |
|---|---|---|
| 0x00 | UShort | MajorVersion |
| 0x02 | UShort | MinorVersion |
| 0x04 | UChar | ProtocolVersion |
| 0x05 | UChar | KdSecondaryVersion |
| 0x06 | UShort | Flags |
| 0x08 | UShort | MachineType |
| 0x0A | UChar | MaxPacketType |
| 0x0B | UChar | MaxStateChange |
| 0x0C | UChar | MaxManipulate |
| 0x0D | UChar | Simulation |
| 0x0E | UShort[] | Unused |
| 0x10 | UQuad | KernBase |
| 0x18 | UQuad | PsLoadedModuleList |
| 0x20 | UQuad | DebuggerDataList |

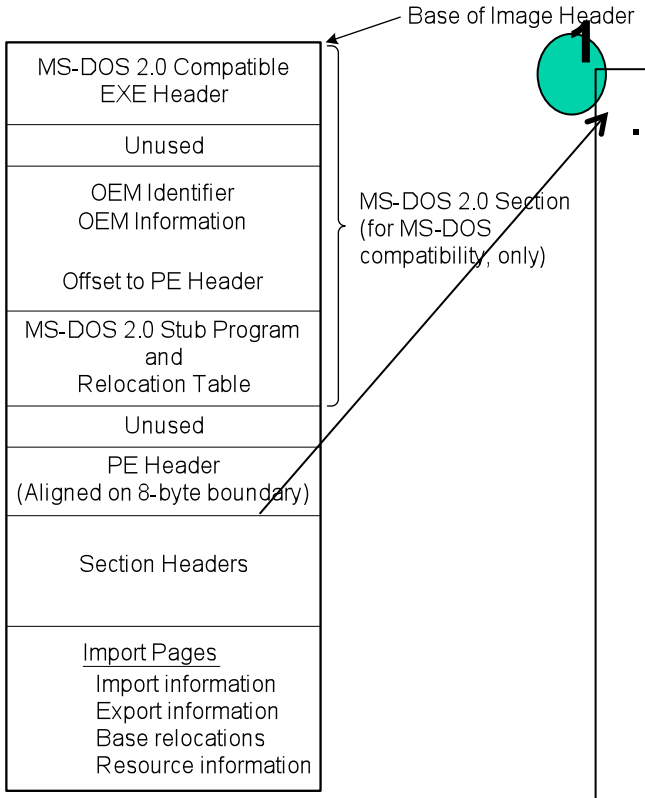
Debugging Tools for Windows\sdk\inc\wbdgexts.h
Ionescu007 Getting Kernel Variables from KdVersionBlock Part 2 (http://www.rootkit.com/newsread.php?newsid=153)

# CMAT – Determine the O/S



**Determine the O/S** → **Scan Memory for Processes & Hives** → **Retrieve Network Information** → **Load User List** → **Connect Users & Processes**

- If _KDDEBUGGER_DATA64 isn't present, go look for the kernel Portable Executable (due to Address Space Layout Randomization (ASLR)




.debug Section

| Offset | Size | Field |
|---|---|---|
| 0 | 4 | Characteristics |
| 4 | 4 | TimeDataStamp |
| 8 | 2 | MajorVersion |
| 10 | 2 | MinorVersion |
| 12 | 4 | Debugger Information Type |
| 16 | 4 | Size of Data |
| 20 | 4 | AddressOfRawData |

.debug data

| Offset | Size | |
|---|---|---|
| 0 | | |
| 24 | | Portable Database Filename |
| … | | |

"ntoskrnl.pdb"
"ntkrnlpa.pdb"
"ntkrpamp.pdb"
"ntkrnlmp.pdb"

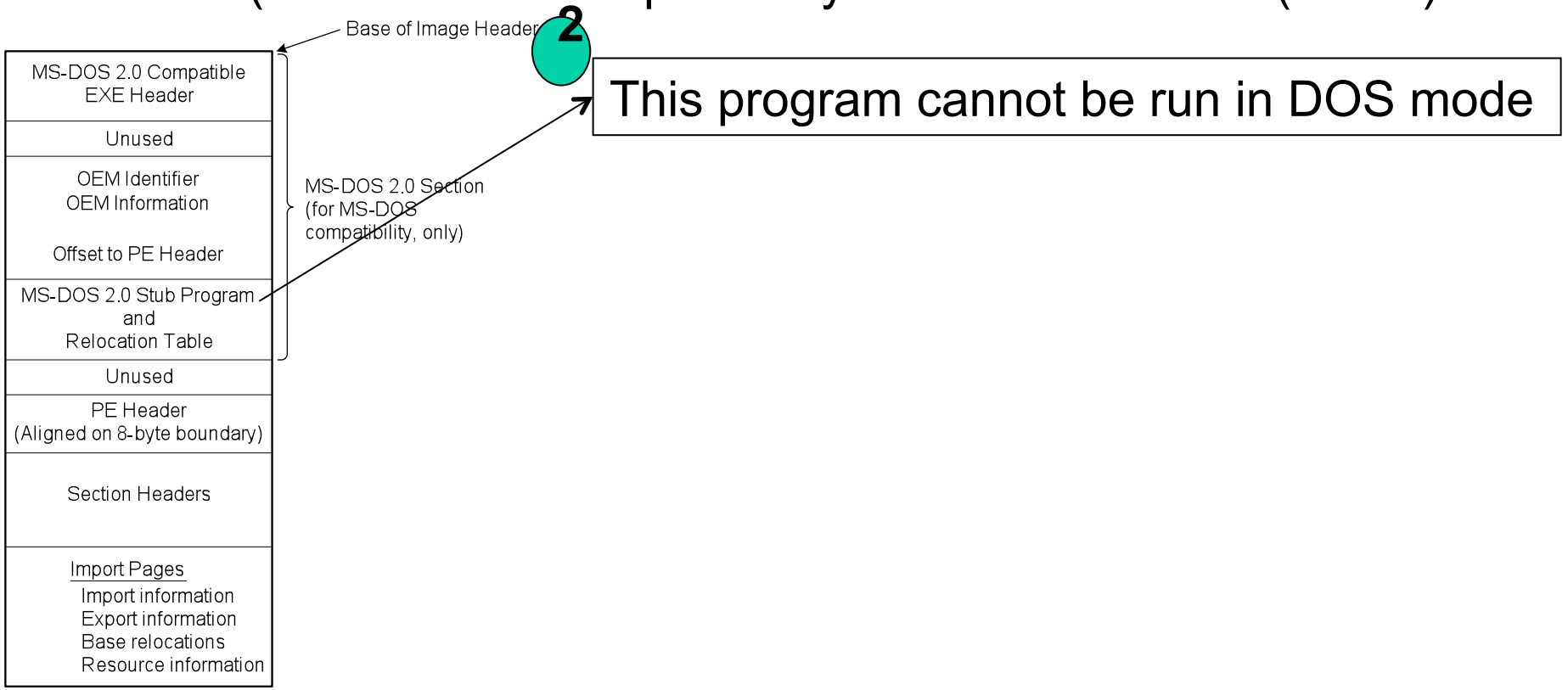
Debugging Tools for Windows\sdk\inc\wbdgexts.h
Ionescu007 Getting Kernel Variables from KdVersionBlock Part 2 (http://www.rootkit.com/newsread.php?newsid=153)

# CMAT – Determine the O/S



**Determine the O/S** → **Scan Memory for Processes & Hives** → **Retrieve Network Information** → **Load User List** → **Connect Users & Processes**

- If _KDDEBUGGER_DATA64 isn't present, go look for the kernel Portable Executable (due to Address Space Layout Randomization (ASLR)



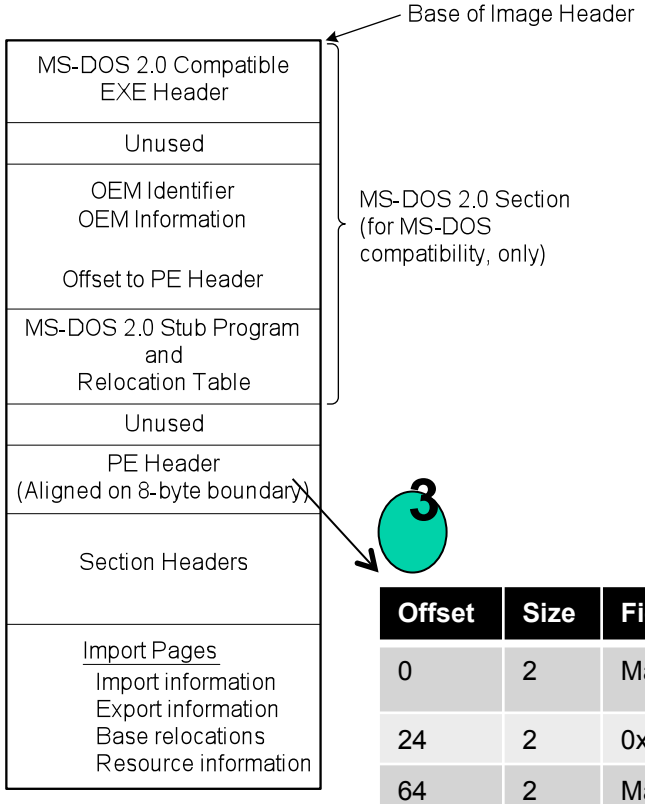

Debugging Tools for Windows\sdk\inc\wbdgexts.h
Ionescu007 Getting Kernel Variables from KdVersionBlock Part 2 (http://www.rootkit.com/newsread.php?newsid=153)

# CMAT – Determine the O/S



**Determine the O/S** → **Scan Memory for Processes & Hives** → **Retrieve Network Information** → **Load User List** → **Connect Users & Processes**

- If _KDDEBUGGER_DATA64 isn't present, go look for the kernel Portable Executable (due to Address Space Layout Randomization (ASLR)




| Offset | Size | Field |
|---|---|---|
| 0 | 2 | MachineType |
| 24 | 2 | 0x10b = 32 bit, 0x20b = 64 bit |
| 64 | 2 | MajorVersion |
| 66 | 2 | MinorVersion |

Debugging Tools for Windows\sdk\inc\wbdgexts.h
Ionescu007 Getting Kernel Variables from KdVersionBlock Part 2 (http://www.rootkit.com/newsread.php?newsid=153)

# CMAT – Determine the O/S



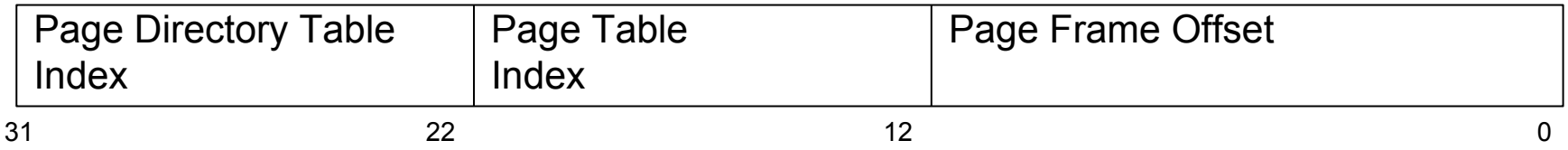

- At this point, we know:
  - 32-bit or 64-bit
  - PAE enabled or disabled
  - Machine Type
  - Operating System version

- We now need to find the tables that convert linear addresses to physical addresses

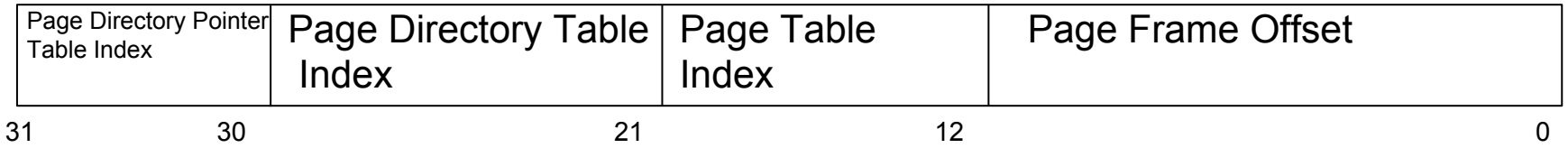
# CMAT – Determine the O/S




Virtual Address – 32-bit with PAE Disabled




Virtual Address – 32-bit with Physical Address Extensions




Virtual Address – 64-bit

# CMAT – Determine the O/S



**Determine the O/S** → **Scan Memory for Processes & Hives** → **Retrieve Network Information** → **Load User List** → **Connect Users & Processes**

- Find self-referencing pages – potential page directory table base addresses (assume the first one is good)

32-bit

| | |
|---|---|
| 0x55d000 | xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx |
| … | |
| 0x55dc00 | 0x55d000 xxxxxxxx xxxxxxxx xxxxxxxx |

32-bit with Physical Address Extensions

| | |
|---|---|
| 0x55d000 | xxxxxxxx 00000000 xxxxxxxx 00000000 |
| 0x55d010 | xxxxxxxx 00000000 0x55d000 00000000 |

64-bit

| | |
|---|---|
| 0x55d000 | xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx |
| … | |
| 0x55df60 | xxxxxxxx xxxxxxxx 0x55d000 00000000 |

# CMAT – Determine the O/S




- At this point, we know:
  - 32-bit or 64-bit
  - PAE enabled or disabled
  - Machine Type
  - Operating System version
  - The Page Directory Table Base for the Kernel

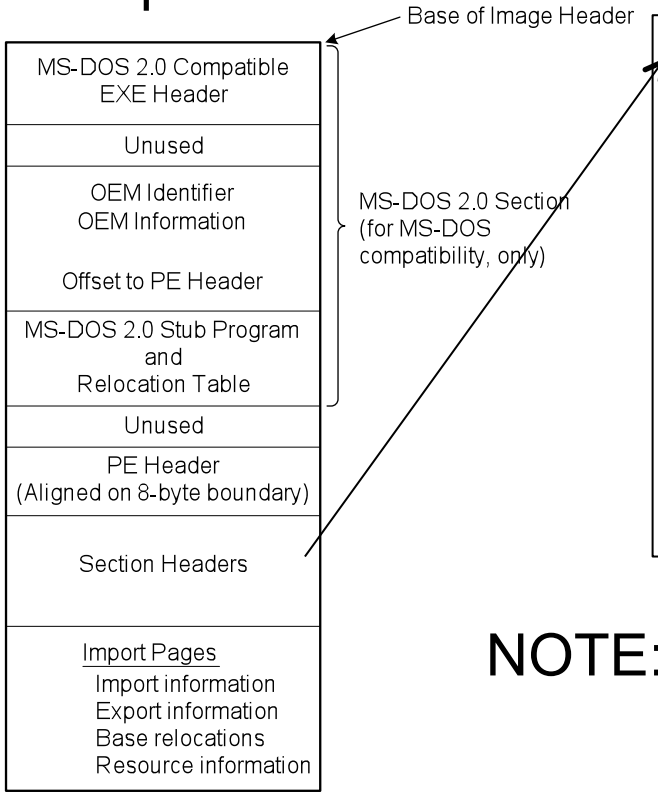
- We now need to retrieve the data structures for this version of the operating system., specifically this version of the kernel executable

# CMAT – Determine the O/S



**Determine the O/S** → **Scan Memory for Processes & Hives** → **Retrieve Network Information** → **Load User List** → **Connect Users & Processes**

- In addition to the name of the pdb file, .debug data also contains the globally unique identifier and age, the two values used to uniquely identify the pdb file on Microsoft's Symbol Server




.debug Section

| Offset | Size | Field |
|---|---|---|
| 0 | 4 | Characteristics |
| 4 | 4 | TimeDataStamp |
| 8 | 2 | MajorVersion |
| 10 | 2 | MinorVersion |
| 12 | 4 | Debugger Information Type |
| 16 | 4 | Size of Data |
| 20 | 4 | AddressOfRawData |

.debug data

| Offset | Size | |
|---|---|---|
| 0 | | |
| 04 | | GUID |
| 20 | | Age |
| 24 | | PDB Filename |
| … | | |

NOTE: This varies by Visual Studio Compiler; however it is stable through XP, Vista, and 7

# CMAT – Determine the O/S




- With the PDB Name, GUID, and Age, a URL is constructed to query the Microsoft Symbol Server and download the PDB file
- Once retrieved in compressed form, the PDB file is uncompressed
- PDB files are structured like file systems

## PDB Header Structures

```
#define PDB_SIGNATURE_200 \
        "Microsoft C/C++ program database 2.00\r\n\x1AJG\0“
 #define PDB_SIGNATURE_TEXT 40

typedef struct _PDB_SIGNATURE {
        BYTE abSignature [PDB_SIGNATURE_TEXT+4];
} PDB_SIGNATURE;

typedef struct _PDB_STREAM {
        DWORD dStreamSize;          // in bytes, -1 = free stream
        PWORD pwStreamPages;        // array of page numbers
} PDB_STREAM,;

typedef struct _PDB_HEADER {
        PDB_SIGNATURE Signature; // PDB_SIGNATURE_200
        DWORD dPageSize;            // 0x0400, 0x0800, 0x1000
        WORD wStartPage;            // 0x0009, 0x0005, 0x0002
        WORD wFilePages;            // file size / dPageSize
        PDB_STREAM RootStream;   // stream directory
        WORD awRootPages [];     // pages containing PDB_ROOT
} PDB_HEADER;

typedef struct _PDB_ROOT {
        WORD wCount;                    // # of Streams
        WORD wReserved; // 0
        PDB_STREAM aStreams [];
} PDB_ROOT;
```

Schreiber, Undocumented Windows 2000 Secrets: A Programmer's Cookbook, 2001. http://www.informit.com/articles/article.aspx?p=22685

# CMAT – Determine the O/S




- Some of the streams include:
  - Section data (e.g., where the .data section begins)
  - Structures (e.g., the _EPROCESS data structure
  - Symbols (location of non-exported symbols, like _AddrObjTable)
  - Symbol Locations (adjustment to ocation of symbols within the .data)

- The location of these streams in the file system varies and they are not labeled. Heuristics are required:
  - The section stream seems to always begin with .data or .text
  - The structures stream seems to always begin at 0x38
  - The symbols stream seems to always begin with a 2 byte record size followed by the literal 0x110E
  - The symbol locations stream seems to always follow the symbols stream

# CMAT – Determine the O/S




## Symbol Stream

| | | |
|---|---|---|
| 0x00 | UShort | Record Size |
| 0x02 | UShort | Unknown1 |
| 0x04 | ULong | Unknown2 |
| 0x08 | ULong | Offset |
| 0x0C | UShort | Type |
| 0x0E | *Char | Symbol Name |

## Section Stream

| | | |
|---|---|---|
| 0x00 | *Char | Name (not null terminated) |
| 0x08 | ULong | Virtual Size |
| 0x0C | ULong | Virtual Address |
| 0x10 | ULong | Raw Size |
| 0x14 | ULong | Raw Pointer |
| 0x18 | ULong | Relocation Pointer |
| 0x1C | ULong | Line Pointer |
| 0x20 | UShort | Relocation Count |
| 0x22 | UShort | Line Count |
| 0x24 | ULong | Characteristics |

## Symbol Relocation Stream

| | | |
|---|---|---|
| 0x00 | ULong | Relocation Address |
| 0x04 | Ulong | Data Address |

## Structure Stream

| | | |
|---|---|---|
| 0x00 | UShort | Record Size |
| 0x02 | STRUCTURE_RECORD | |

# CMAT – Determine the O/S



**Determine the O/S** → **Scan Memory for Processes & Hives** → **Retrieve Network Information** → **Load User List** → **Connect Users & Processes**

```
LF_FIELDLIST
    list of fields
LF_STRUCTURE
    UShort      Element Count
    UShort      Properties
    ULong       Field Index
    ULong       Derived
    ULong       Vshape
    UShort      Size
    Char*       Name
LF_POINTER
    ULong       Underlying Type
    ULong       Pointer Array
LF_MEMBER
    UShort      Properties
    ULong       Underlying Type
    UShort      Offset
     Char*      Name
LF_UNION
    UShort      Element Count
    UShort      Properties
    ULong       Field Index
    UShort      Size
    Char*       Name
LF_ARGLIST
    ULong       Element Count
    ULong[]     Arguments
```

```
LF_PROCEDURE
    ULong       Return Value Type
    UChar       Call Type
    UChar       Unknown
    UShort      Element Count
    ULong       Field Index
LF_ENUM
    UShort      Element Count
    UShort      Properties
    ULong       Underlying Type
    ULong       Field Index
    Char*       Name
LF_ENUMERATE
    UShort      Properties
    UShort      Value
    Char*       Name
LF_ARRAY
    ULong       Underlying Type
    ULong       Index Type
    UShort      Size
    UShort      Unknown
LF_BITFIELD
    ULong       Underlying Type
    UChar       Size
    UChar       Offset
    UShort      Unknown
```

# CMAT – Determine the O/S




- At this point, we know:
  - 32-bit or 64-bit
  - PAE enabled or disabled
  - Machine Type
  - Operating System version
  - The Page Directory Table Base for the Kernel
  - All of the data structures that the kernel uses

- Now it's time to start parsing the memory dump!

# CMAT – Extract Process Information




- Scan memory for
  - Potential processes (_DISPATCHER_HEADER: Type = 03, Absolute = 0, Size = O/S dependent)

| _EPROCESS | | |
|---|---|---|
| 0 | _KPROCESS | Pcb |
| 108 | _EX_PUSH_LOCK | ProcessLock |
| 112 | _LARGE_INTEGER | CreateTime |
| 120 | _LARGE_INTEGER | ExitTime |
| 128 | _EX_RUNDOWN_REF | RundownProtect |
| 132 | 32PVoid | UniqueProcessId |
| 136 | _LIST_ENTRY | ActiveProcessLinks |
| 144 | ULong[] | QuotaUsage |
| 156 | ULong[] | QuotaPeak |
| 168 | ULong | CommitCharge |
| 172 | ULong | PeakVirtualSize |
| 176 | ULong | VirtualSize |
| 180 | _LIST_ENTRY | SessionProcessLinks |
| 188 | 32PVoid | DebugPort |
| 192 | 32PVoid | ExceptionPort |
| 196 | *_HANDLE_TABLE | ObjectTable |
| 200 | _EX_FAST_REF | Token |
| … | | |

| _KPROCESS | | |
|---|---|---|
| 0 | _DISPATCHER_HEADER | Header |
| 16 | LIST_ENTRY | ProfileListHead |
| 24 | ULong[] | DirectoryTableBase |
| 32 | _KGDTENTRY | LdtDescriptor |
| 40 | _KIDTENTRY | Int21Descriptor |
| 48 | UShort | IopmOffset |
| 50 | UChar | Iopl |
| 51 | UChar | Unused |
| 52 | ULong | ActiveProcessors |
| 56 | ULong | KernelTime |
| 60 | ULong | UserTime |
| 64 | LIST_ENTRY | ReadyListHead |
| 72 | LIST_ENTRY | SwapListEntry |
| 76 | 32PVoid | VdmTrapcHandler |
| 80 | LIST_ENTRY | ThreadListHead |
| 88 | ULong | ProcessLock |
| 92 | ULong | Affinity |
| 96 | UShort | StackCount |
| 98 | RChar | BasePriority |
| 107 | UChar | ExecuteOptions |

| _DISPATCHER_HEADER | | |
|---|---|---|
| 0 | UChar | Type |
| 1 | UChar | Absolute |
| 2 | UChar | Size |
| 3 | UChar | Inserted |
| 4 | Long | SignalState |
| 8 | _LIST_ENTRY | WaitListHead |

Betz, Chris. Memparser Version 1.0
http://www.microsoft.com/resources/sharedsource/windowsacademic/researchkernelkit.mspx

# CMAT – Extract Registry Information




- Scan memory for
  - Potential hives (_HIVE: signature = bee0bee0)

```
_CMHIVE
0     _HHIVE                                  Hive
528  32PVoid[]                                FileHandles
540   LIST_ENTRY                              NotifyList
548   LIST_ENTRY                HiveList
556   *_FAST_MUTEX                            HiveLock
560   *_FAST_MUTEX                            ViewLock
564   LIST_ENTRY                LRUViewListHead
572   LIST_ENTRY                PinViewListHead
580   *_FILE_OBJECT                           FileObject
584  _UNICODE_STRING                          FileFullPath
592  _UNICODE_STRING                          FileUserName
600   Ushort                                  MappedViews
602   Ushort                                  PinnedViews
604   Ulong                                   UseCount
608   Ulong                                   SecurityCount
612   Ulong                                   SecurityCacheSize
616   Long                                    SecurityHitHint
620   *_CM_KEY_SECURITY_CACHE_ENTRY           SecurityCache
624   LIST_ENTRY[]                            SecurityHash
1136  *_KEVENT                                UnloadEvent
1140 *_CM_KEY_CONTROL_BLOCK                   RootKcb.
…
```

```
__HIVE
0                    ULong          Signature
4                    *13ef          GetCellRoutine
8                    *13f1          ReleaseCellRoutine
12                   *13f4          Allocate
16                   *13f7          Free
20                   *13fa          FileSetSize
24                   *13ff          FileWrite
28                   *1402          FileRead
32                   *1406          FileFlush
36                *_HBASE_BLOCK     BaseBlock
40                 _RTL_BITMAP      DirtyVector
48                   ULong          DirtyCount
52                   ULong          DirtyAlloc
56                   UChar          RealWrites
60                   ULong          Cluster
64                   UChar          Flat
65                   UChar          ReadOnly
66                   UChar          Log
68                   ULong          HiveFlags
72                   ULong          LogSize
76                   ULong          RefreshCount
80                   ULong          StorageTypeCount
84                   ULong          Version
88                 _DUAL[]          Storage
```

Dolan-Gavitt, Brendan. Push the Red Button. http://moyix.blogspot.com/2008/02/enumerating-registry-hives.html

# CMAT – Extract Network Information




- Scan list of loaded modules to find address of tcpip.sys Portable Executable
- Locate the data structures in the data section of tcpip.sys
  - Windows XP: TCP: _AddrObjTable, _AddrObjTableSize,
    UDP: _TCBTable, _MaxHashTableSize
- Traverse the data structures for open port/socket information and associate them with relevant processes

Volatile Systems, The Volatility Framework: Volatile Memory Artifact Extraction Utility Framework. https://www.volatilesystems.com/default/volatility

# CMAT – Load User List




- Traverse the REGISTRY/MACHINE/SOFTWARE hive for the key list:
  Microsoft/Windows NT/Current Version/ProfileList
- Create a user list structure using this list and associate each token id with the home directory and name

# CMAT – Connect Users/Processes




- Connect the token id in the process with the token id in the user list and then store the user's name in the process information

# Summary

- At this point, we have
  - Meta-Information
    - 32-bit or 64-bit, PAE enabled or disabled, Machine Type, O/S Version
    - All of the data structures that the kernel uses
  - All the processes that were running on the machine
    - The IP addresses and ports they were using
    - The objects (files, registry keys, etc.) they accessed
    - Modules loaded
  - The current configuration information for the machine
    - The users with accounts
    - The last run configurations for the different applications
  - The portable executables for all active programs
    - The actual code that was being executed
    - The current values of all of the variables

# Experiments

- Test Environment
  - Operating Systems: Windows XP 32-bit w/ PAE, Windows XP 32-bit w/o PAE, Windows Vista 32-bit w/ PAE, Windows 7 64-bit
  - Applications: Internet Explorer, Word, PowerPoint, Visual Studio, Calculator, Kernel Debugger, two command line shells (one hidden by FUTo)
  - Memory Dump generated by Matthew Suiche's Win32DD and Win64DD
  - Microsoft Utilities used: netstat
  - SysInternals utilities used: psinfo, pslist, logonsessions, handles, listdlls

- Results
  - CMAT provides equivalent information to SysInternals/ Microsoft tools
  - CMAT provides equivalent information to  Volatility for Windows XP

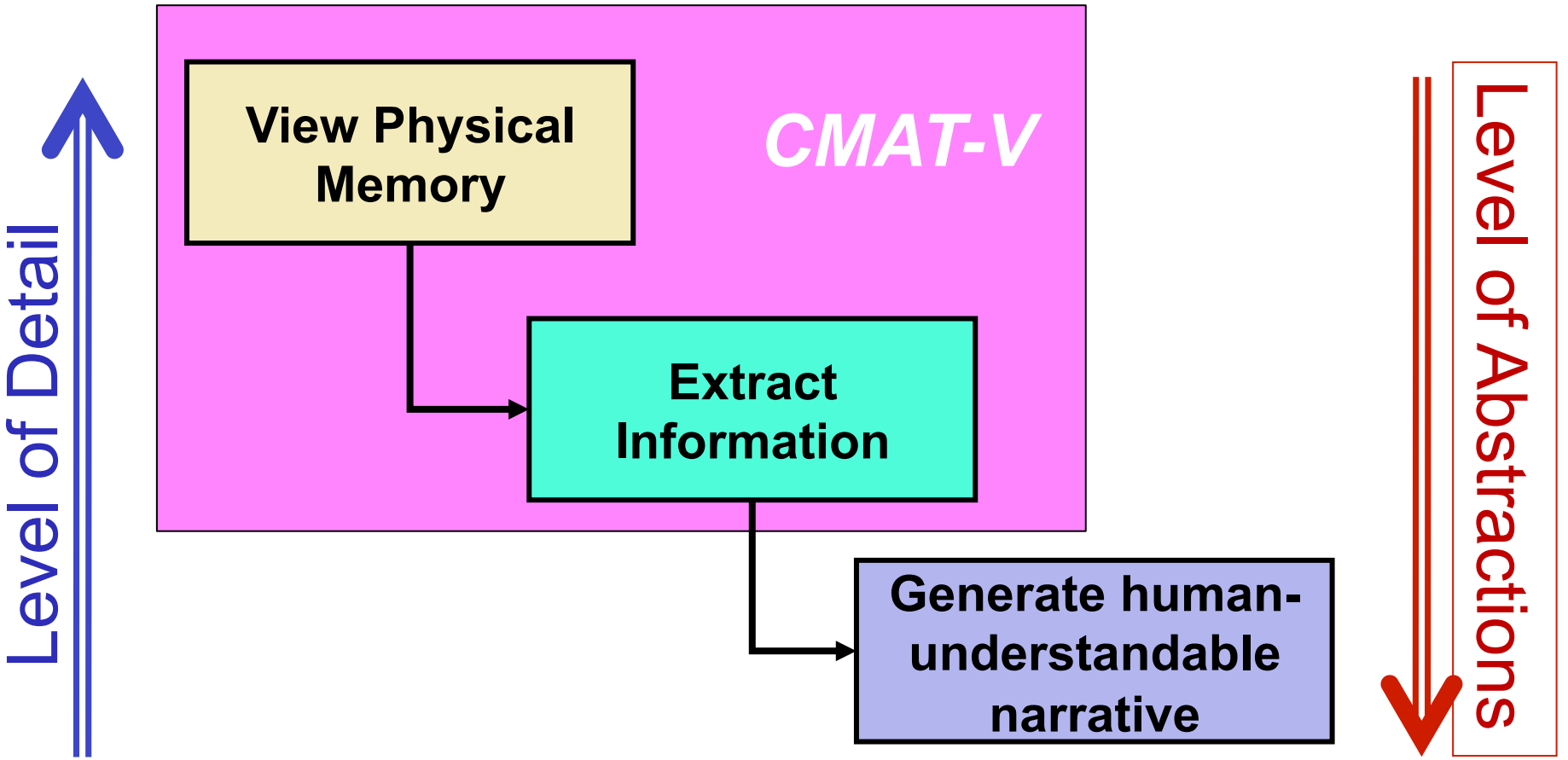
# Bottom Line

- CMAT provides equivalent results to other memory analysis tools without regard to the specific Windows O/S

- Concerns
  - CMAT still relies on some intrinsic knowledge of the O/S (.e.g., there is a structure named _EPROCESS that stores process information)
    - In Windows 7, Microsoft changed how the _OBJECT_TYPE was stored within _OBJECT_HEADER
    - In Windows Vista, Microsoft implemented to Address Space Layout Randomization (ASLR)
  - Extracting symbols from proprietary executables (e.g., tcpip.sys) remains problematic because the symbol names are not published requiring reverse engineering.

# Next Steps

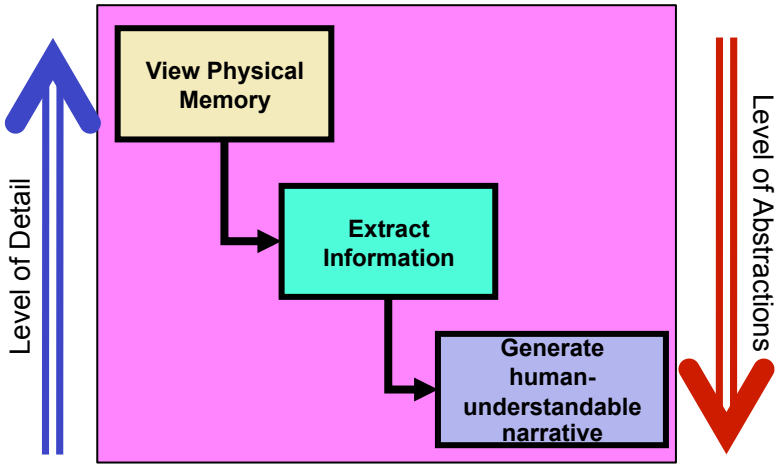





CMAT-V extends CMAT into Virtual Machines allowing Near real-time analysis of memory for Cyber SA

# Next Steps

- Is there a way to take this still overwhelming amount of information and abstract it to a level that is more useful to investigators?
  - Develop use cases for specific types of incidents
  - Determine what information investigators are most likely to need
  - Develop scripts to provide use case information in summary form while still allowing investigators to search for other information if necessary.



# Questions



?

Contact information:

| | | |
|---|---|---|
| James (Jimmy) Okolica | jokolica@afit.edu | 937-255-3636 x7255 |
| Gilbert Peterson | gpeterson@afit.edu | 937-255-3636 x4281 |

# Backup Slides

# Live Cyber Forensics



- Business Productivity
  - Lost Revenue
  - Concern of the system coming back up
- Acquisition of volatile-only information
  - Network Traffic
  - Active process and user information
- Encrypted Hard Drives
- Memory Resident Malware
- Too much data

# CMAT – Extract Registry Information




- The Registry is basically a collection of file systems
  - To find information in them
    - Start at the Directory Base and add the offset to find the Table
    - Start at the Table Base and add the offset to find the location in the Hive Table
    - Move down through the “sub-directories” until arriving at the Value List

- 2 system hives – Machine and User
- 5 machine hives – Security Account Manager, Security, Software, Hardware, System
- 6 user hives – 2 each for
  - currently logged on user
  - local service user
  - the networks service user

# CMAT – Display Information



**User Interface – Enumerate/ Dump System and Process Information**

- Enumerate system information
  - Operating system version
  - Number of processors
  - Physical address extensions enabled
  - Physical address of KDBG and (one of) the Kernel Page Directories
  - Virtual address of the Kernel Portable Executable
  - List of users with accounts
  - Virtual address of each of the registry hives

```
Showing System Information

OS:     WinXP   Version 5.1 SP 3        build:  2600
Number of processors: 2
Machine Type:   I386
KDBG (phys) = 444b50     Kernel PageDirectoryBase (phys) = 55d100       Kernel PE Image (virt) = 804d7000

Compile Memory Analysis Tool Jimmy Okolica, 2009


User List
       User            Token                                                   Path
      Twiddler         S-1-5-21-3477281491-1022105880-1400426842-500           C:\Documents and Settings\Twiddler
     jokolica4         S-1-5-21-3477281491-1022105880-1400426842-1087          C:\Documents and Settings\jokolica4
       tester          S-1-5-21-3477281491-1022105880-1400426842-1082          C:\Documents and Settings\tester
      ENHELP           S-1-5-21-3477281491-1022105880-1400426842-1020          C:\Documents and Settings\ENHELP
      bcarter          S-1-5-21-1660827705-1073358324-288910612-9817           C:\Documents and Settings\bcarter
    RunMD5Scan         S-1-5-21-1660827705-1073358324-288910612-38801          C:\Documents and Settings\RunMD5Scan
                       S-1-5-21-1660827705-1073358324-288910612-37065
      jokolica         S-1-5-21-1660827705-1073358324-288910612-35580          C:\Documents and Settings\jokolica
        runssv         S-1-5-21-1660827705-1073358324-288910612-34439          C:\Documents and Settings\runssv
     rheisman1         S-1-5-21-1660827705-1073358324-288910612-32779          C:\Documents and Settings\rheisman1
      tlacey3          S-1-5-21-1660827705-1073358324-288910612-30308          C:\Documents and Settings\tlacey3
     bcarter3          S-1-5-21-1660827705-1073358324-288910612-30300          C:\Documents and Settings\bcarter3
     RunMatLab         S-1-5-21-1660827705-1073358324-288910612-28391          C:\Documents and Settings\RunMatLab
      rheisman         S-1-5-21-1660827705-1073358324-288910612-21610          C:\Documents and Settings\rheisman
        mpeck          S-1-5-21-1660827705-1073358324-288910612-1175           C:\Documents and Settings\mpeck
NetworkService.NT AUTHORITY            S-1-5-20              C:\Documents and Settings\NetworkService.NT AUTHORITY
LocalService.NT AUTHORITY              S-1-5-19              C:\Documents and Settings\LocalService.NT AUTHORITY
  systemprofile        S-1-5-18              %systemroot%\system32\config\systemprofile


Hive List:
 Hive#          Name
e8b39b60        REGISTRY/USER/S-1-5-21-3477281491-1022105880-1400426842-1087_Classes
e3d3db60        REGISTRY/USER/S-1-5-21-1660827705-1073358324-288910612-35580
e5fe1578        REGISTRY/USER/S-1-5-21-1660827705-1073358324-288910612-35580_Classes
e8ac2008        REGISTRY/USER/S-1-5-21-1660827705-1073358324-288910612-21610
e4b939f0        REGISTRY/USER/S-1-5-19
e3bdbb60        REGISTRY/MACHINE/SOFTWARE
e4bb8b60        REGISTRY/USER/S-1-5-20_Classes
e4d176a8        REGISTRY/USER/S-1-5-20
e83e0008        REGISTRY/USER/S-1-5-21-1660827705-1073358324-288910612-32779
e39f46b8        REGISTRY/MACHINE/SAM
e4fac6a8        REGISTRY/USER/S-1-5-19_Classes
e18bd008        REGISTRY/MACHINE/HARDWARE
e190e3e0        REGISTRY/USER/.DEFAULT
e787eb60        REGISTRY/USER/S-1-5-21-3477281491-1022105880-1400426842-1087
e39f4b60        REGISTRY/MACHINE/SECURITY
e1035758        REGISTRY/MACHINE/SYSTEM
e1023758        REGISTRY
8067d18c
```

# CMAT – Display Information



**User Interface – Enumerate/ Dump System and Process Information**

- Enumerate process handles (e.g., files opened, registry keys accessed)

```
376: cmd.exe selected:
1       Display Process Environment Information
2       Display all DLLs loaded by process
3       Display all Files accessed by process
4       Display all Registry Keys accessed by process
5       Display all Sockets opened by process
<enter>:        quit
3



85205198                (RW-)  (Section) \Documents and Settings\jokolica\My Documents\CMAT
84e55620                (---)  (Section)
859b00c8                (RW-)  (Section) \WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512
w_35d4ce83
e47abdb8                (---)  (Section)
e4bee588                (---)  (Section)
e7188760                (---)  (Section)


376: cmd.exe selected:
1       Display Process Environment Information
2       Display all DLLs loaded by process
3       Display all Files accessed by process
4       Display all Registry Keys accessed by process
5       Display all Sockets opened by process
<enter>:        quit
4



Key e17f47d0            REGISTRY/MACHINE/
Key e39f6ec0            REGISTRY/MACHINE/SOFTWARE/Microsoft/Windows NT/CurrentVersion/Drivers32/
Key e1adf0b0            REGISTRY/MACHINE/SOFTWARE/Microsoft/Windows NT/CurrentVersion/Drivers32/
Key e5ffdd18            REGISTRY/USER/S-1-5-21-3477281491-1022105880-1400426842-1087/
Key e1591020            REGISTRY/MACHINE/SYSTEM/ControlSet001/Control/Nls/Locale/
Key e4514c08            REGISTRY/MACHINE/SYSTEM/ControlSet001/Control/Nls/Locale/Alternate Sorts/
Key e490faf8            REGISTRY/MACHINE/SYSTEM/ControlSet001/Control/Nls/Language Groups/
```

# CMAT – Display Information



**User Interface – Enumerate/ Dump System and Process Information**

- Enumerate modules loaded by process

```
376: cmd.exe selected:
1       Display Process Environment Information
2       Display all DLLs loaded by process
3       Display all Files accessed by process
4       Display all Registry Keys accessed by process
5       Display all Sockets opened by process
<enter>:        quit
2



Base Dll Name: cmd.exe          Full Name: C:\WINDOWS\system32\cmd.exe
Base Dll Name: ntdll.dll        Full Name: C:\WINDOWS\system32\ntdll.dll
Base Dll Name: kernel32.dll     Full Name: C:\WINDOWS\system32\kernel32.dll
Base Dll Name: msvcrt.dll       Full Name: C:\WINDOWS\system32\msvcrt.dll
Base Dll Name: USER32.dll       Full Name: C:\WINDOWS\system32\USER32.dll
Base Dll Name: GDI32.dll        Full Name: C:\WINDOWS\system32\GDI32.dll
Base Dll Name: ShimEng.dll      Full Name: C:\WINDOWS\system32\ShimEng.dll
Base Dll Name: AcGenral.DLL     Full Name: C:\WINDOWS\AppPatch\AcGenral.DLL
Base Dll Name: ADVAPI32.dll     Full Name: C:\WINDOWS\system32\ADVAPI32.dll
Base Dll Name: RPCRT4.dll       Full Name: C:\WINDOWS\system32\RPCRT4.dll
Base Dll Name: Secur32.dll      Full Name: C:\WINDOWS\system32\Secur32.dll
Base Dll Name: WINMM.dll        Full Name: C:\WINDOWS\system32\WINMM.dll
Base Dll Name: ole32.dll        Full Name: C:\WINDOWS\system32\ole32.dll
Base Dll Name: OLEAUT32.dll     Full Name: C:\WINDOWS\system32\OLEAUT32.dll
Base Dll Name: MSACM32.dll      Full Name: C:\WINDOWS\system32\MSACM32.dll
Base Dll Name: VERSION.dll      Full Name: C:\WINDOWS\system32\VERSION.dll
Base Dll Name: SHELL32.dll      Full Name: C:\WINDOWS\system32\SHELL32.dll
Base Dll Name: SHLWAPI.dll      Full Name: C:\WINDOWS\system32\SHLWAPI.dll
Base Dll Name: USERENV.dll      Full Name: C:\WINDOWS\system32\USERENV.dll
Base Dll Name: UxTheme.dll      Full Name: C:\WINDOWS\system32\UxTheme.dll
Base Dll Name: IMM32.DLL        Full Name: C:\WINDOWS\system32\IMM32.DLL
Base Dll Name: comctl32.dll     Full Name: C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5
x-ww_35d4ce83\comctl32.dll
Base Dll Name: comctl32.dll     Full Name: C:\WINDOWS\system32\comctl32.dll
Base Dll Name: Apphelp.dll      Full Name: C:\WINDOWS\system32\Apphelp.dll
Base Dll Name: MSCTF.dll        Full Name: C:\WINDOWS\system32\MSCTF.dll
Base Dll Name: msctfime.ime     Full Name: C:\WINDOWS\system32\msctfime.ime
Base Dll Name: mslbui.dll       Full Name: C:\WINDOWS\system32\mslbui.dll
```

# CMAT – Display Information

*Develop America's Airmen Today ... for Tomorrow*

**User Interface – Enumerate/ Dump System and Process Information**

- Enumerate process environment (e.g., fully qualified executable, command line)

```
Process Environment Information:
        Executable File:        C:\WINDOWS\system32\cmd.exe
        Command Line:   "C:\WINDOWS\system32\cmd.exe"
        Window Title:   C:\Documents and Settings\All Users\Desktop\Command Prompt.lnk
        Desktop Info:   WinSta0\Default
        Shell Info:
        Runtime Data:
        Dll Path:       C:\WINDOWS\system32;C:\WINDOWS\system32;C:\WINDOWS\system;C:\WINDOWS;.;C:\WINDOWS\system32;C:\WINDOWS;
INDOWS\system32\wbem;c:\progra~1\gradke~1\dbsign~1\lib;c:\texmf\miktex\bin;c:\orant\bin;c:\program files\activcard\activcard g
resources;c:\windows\system32;c:\windows;c:\windows\system32\wbem;c:\program files\common files\adaptec shared\system;c:\progr
iles\ati technologies\ati control panel;c:\program files\common files\roxio shared\dllshared;C:\Program Files\MatLab\R2007a\bi
\Program Files\MatLab\R2007a\bin\win32;C:\Program Files\ActivIdentity\ActivClient\;C:\Program Files\Common Files\Telelogic\SA;
rogram Files\Tumbleweed\Desktop Validator\;C:\strawberry\c\bin;C:\strawberry\perl\bin;C:\Program Files\ATI Technologies\ATI.AC
re-Static
```

- Enumerate process network activity

```
4: System selected:
1       Display Process Environment Information
2       Display all DLLs loaded by process
3       Display all Files accessed by process
4       Display all Registry Keys accessed by process
5       Display all Sockets opened by process
<enter>:        quit
5


Local Address          Remote Address         Protocol
        0.0.0.0:445            *.*            UDP
  192.168.145.1:137            *.*            UDP
   192.168.36.1:137            *.*            UDP
  192.168.145.1:139      0.0.0.0:0            TCP
   192.168.36.1:139      0.0.0.0:0            TCP
   129.92.36.44:138            *.*            UDP
   129.92.36.44:137            *.*            UDP
        0.0.0.0:445      0.0.0.0:0            TCP
   129.92.36.44:139      0.0.0.0:0            TCP
  192.168.145.1:138            *.*            UDP
   192.168.36.1:138            *.*            UDP
        0.0.0.0:1092     0.0.0.0:0            TCP
```