WINDOWS HIBERNATION FILE FOR FUN 'N' PROFIT

Matthieu Suiche

matt[at]msuiche.net

http://www.msuiche.net
http://www.moonsols.com

Outline

- Who am I?
  - OS/Security Researcher
- Objectives
  - Introducing a new method of memory dumping
    - Advantages
    - Windows Hibernation File Internals
  - Exploiting this file for:
    - Defensive (forensics) use
    - Offensive (offensics) use
  - Showing some demos
  - Giving away SandMan's newest release :-)

Introduction: Just what is *hibernation*?

- **Microsoft name for « *suspend to disk* » feature**
  - Available since Windows 2000.
  - This feature is also implemented in non-MS O.S. (MacOSX, Linux,..)
- **`\hiberfil.sys`**
  - Yes, this is this file!
  - It contains a full dump of the memory
- **How do I hibernate?**
  - Start > Hibernate
  - Command line:
    - `Powercfg /hibernate` (to activate)
    - `Shutdown /h` (to hibernate)

Advantages (1/3)

- It is quick and easy.
  - No hardware prerequisite
  - Hibernation can be activated without reboot
  - Contains additional and useful information
- An efficient way to get a physical snapshot to avoid
  - Generating a crash dump through BSOD
  - Using standalone tools like `win32dd` [1], `mdd` [2], `dd` [3]
    - These kind of tools aren't necessary compatible with 64bits
      - E.g. Drivers signing.
    - Because *in MICROSOFT we trust !*

Advantages (2/3)

- Significant advantages for investigators
  - System activity is totaly frozen
    - No software tool is able to block the analysis
    - System is left perfectly functional after analysis
  - `SandMan` can generate a readable dump
    - Dump is interoperable with others tools like:
      - `Volatility Framework` [4]
      - `PTFinder` (Andreas Schuster) [5]
      - …
- Writable too?
  - Yes! MS Bootloader loads it when the machine is waking up.
  - Modified code can be executed!!

Advantages (3/3)

- *My header is rich!* It also contains:
  - Processor state is saved thus :
  - We can retrieve Control Registers
    - Very useful for memory management functions
      - Like virtual address translation
  - And other interesting things like previous EIP.
  - Interrupt Descriptor Table (IDT) base address
  - Global Descriptor Table (GDT) base address
    - Segmentation!

Windows hibernation file internals (1/7)

Field	Content
Header	PO_MEMORY_IMAGE structure
Page list	An array of physical page.
Processor State	CONTEXT + KSPECIAL_REGISTERS
Memory Range Array n	Header: NextTable page + Number of Entries. Entries: Destination Page + Checksum
Xpress compressed block p	Magic « \x81\x81xpress » (> Win2K) Compressed data
Xpress compressed block p+1	…
Memory Range Array n+1	…

Windows hibernation file internals (2/7)

Header

- `PO_MEMORY_IMAGE`
  is exported in debugging symbols

- Main magic bytes are:
  - `hibr`:
    hibernation file is valid, system shall be resumed on boot
    - Vista (and above) makes use of caps (HIBR)
  - `wake`:
    hibernation file is invalid, system shall be start anew.

Windows hibernation file internals (3/7)

Processor State

- `KPROCESSOR_STATE` is exported in debugging symbols
- This structure is filled by calling `KeSaveStateForHibernate()` in `ntoskrnl`.
- This structure contains very interesting values like:
  - `GS`, `FS`, `ES`, `DS` segments registers
  - `EIP` (If we apply a mask we can get Ntoskrnl image base)
  - Global Descriptor Table (GDT) Offset
  - Interrupt Descriptor Table (IDT) Offset
  - Control registers (`CR0`, `CR3`)

Windows hibernation file internals (4/7)

Memory Range Array

- `PO_MEMORY_RANGE_ARRAY` is exported in debugging symbols
  - However, this structure *does* change accros Windows versions.
  - Number of entries per array never exceed `0xFF`
  - Pages are not ordered.

Xpress blocks

- Uncompressed block size is 64kB (0x10 Pages)
- Windows 2000 uses LZNT1
- > Windows 200 O.S. uses internal functions called XpressEncode()

Windows hibernation file internals (5/7)

- **Xpress compression algorithm**
  - Xpress algorithm has been implemented by Microsoft Exchange Team
    - Used for LDAP protocol
    - In Microsoft Embedded O.S. Windows CE
    - In Windows IMaging format (WIM) implemented in Windows Vista.
  - This algorithm has been publicly documented since recent Microsoft Interoperability initiative (February 2008)
    - Even, if beta version of SandMan supported it before ☺
  - According to Microsoft Exchange documentation, XPRESS algorithm is:
    - LZ77 + DIRECT2
    - LZ77 for compression and DIRECT2 encode bytes positions in meta-data

MoonSols

Windows hibernation file internals (6/7)

- **Some notes**
  - If Checksum are set to 0, MS Boot Loader doesn't check compressed pages ☺
  - Checksum algorithm computed via `tcpxsum()`
  - Everything is page-aligned (`PAGE_SIZE = 0x1000` (4kb))
  - O.S. fingerprinting is possible using slight variations
    - Header magic bytes (hibr or HIBR?)
    - `PO_IMAGE_MEMORY` size
    - It's useful, but not very informative.
      - `-> HiberGetVersion()`

Windows hibernation file internals (7/7)

- Some notes
  - Hibernation file is NEVER wiped out,
    - only the first page (page header) is wiped after being resumed.
  - Then we can still analyze the hibernation file!

SandMan Framework

- **Here SandMan comes!**

- **Objectives:**
  - Providing a READ and WRITE access to the hibernation file.

Defensive uses (1/4)

- **Kernel-land malwares**
  - If the code isn't in the hibernation file
    - It won't resume execution
  - E.g. `SMM Rootkit`
    - People says SMM Rootkit are great because of SMRAM.
    - Why? Because nobody can access to it.
    - … even Windows… Then, after resume, hibernation process will clear the SMRAM ☺
    - Bye bye SMM Rootkit.

Defensive uses (2/4)

- **Kernel-land malwares detection**
  - We can imagine some detection cases for kernel-land malwares:
    - Like writting an SandMan extensions to check the Integrity of :
      - System Service Dispatch Table (`SSDT`)
        - `ZwCreateFile`, `NtQueryDirectoryFile`, etc..
      - Interrupt Descriptor Table (`IDT`)
        - `Int 3`, for anti-debugging tricks.
      - Global Descriptor Table (G`DT`)
      - …

Defensive uses (3/4)

Kernel-land malwares detection

- Like writting an SandMan extensions to check the Integrity of :
  - Import Address Table (`IAT`)
  - Export Address Table (`EAT`)

- What about inline patching?
  - A disassembling library could help to prevent from inline patching.

- There are a lot of possiblities, to identify rootkit it depends on their behavior.
- Every running driver is mapped in the hibernation file, else it won't be resumed. (e.g. if a driver try to hook hibernation process)

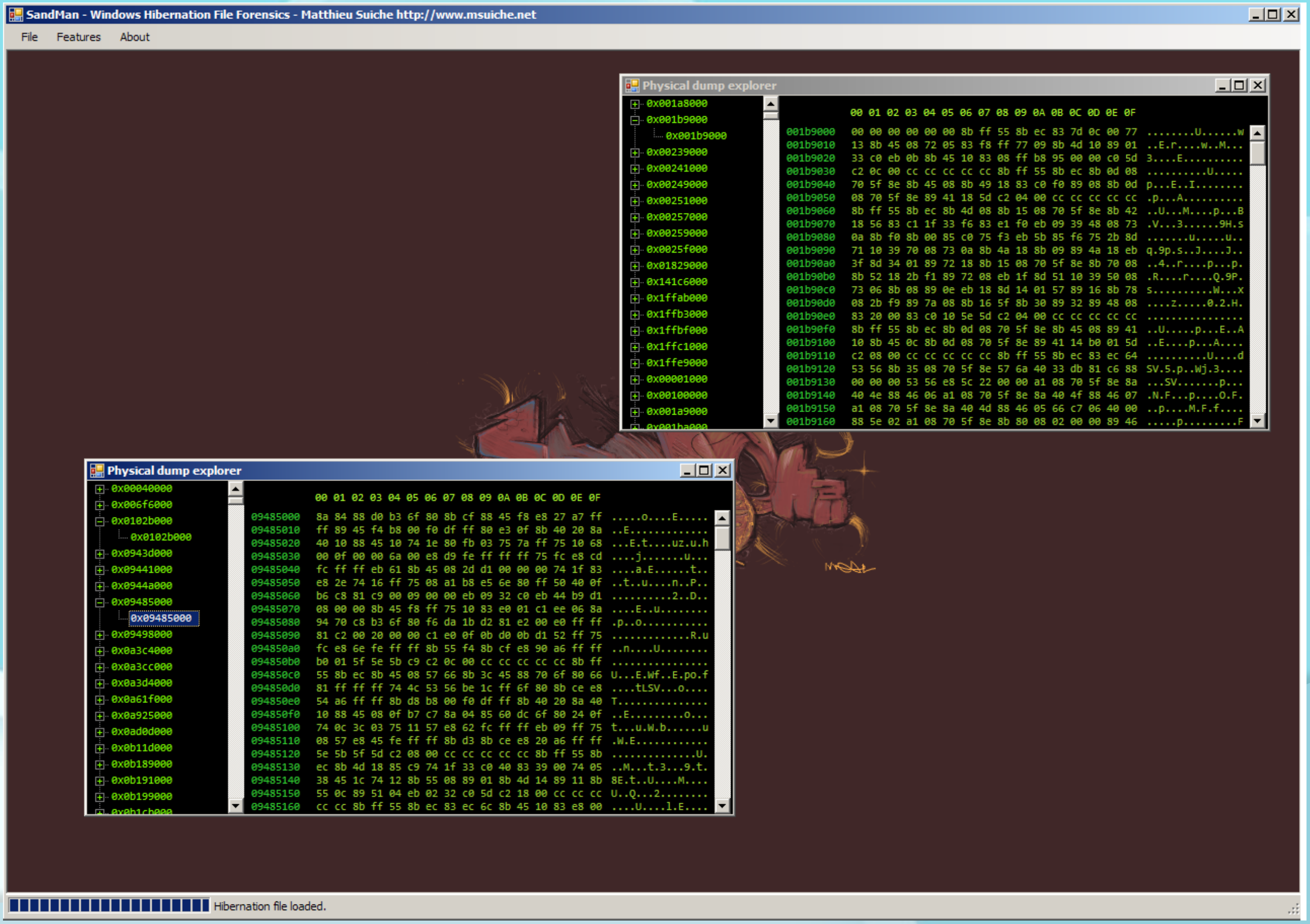
Defensive uses (4/4)

Forensics through hibernation

- Live memory analysis is growing interest since DFRWS 2005
  - PTFinder, MemParser, Windows Memory Forensics Toolkit, PMODump, FATKit, Volatility, etc..
- Hibernation file exploitation is powerful because additional information is provided. For instance, Volatility Framework can use CR3 as well.
- Unlike these projects, it's not mandatory to proceed to a « blind » analysis.

SandMan GUI

- A Stand-alone graphical hibernation page explorer.


SandMan - Windows Hibernation File Forensics - Matthieu Suiche http://www.msuiche.net
File
Features
About
Physical dump explorer
Physical dump explorer
Hibernation file loaded.

```
0x001a8000
0x001b9000
    0x001b9000
0x00239000
0x00241000
0x00249000
0x00251000
0x00257000
0x00259000
0x0025f000
0x01829000
0x141c6000
0x1ffab000
0x1ffb3000
0x1ffbf000
0x1ffc1000
0x1ffe9000
0x00001000
0x00100000
0x001a9000
```

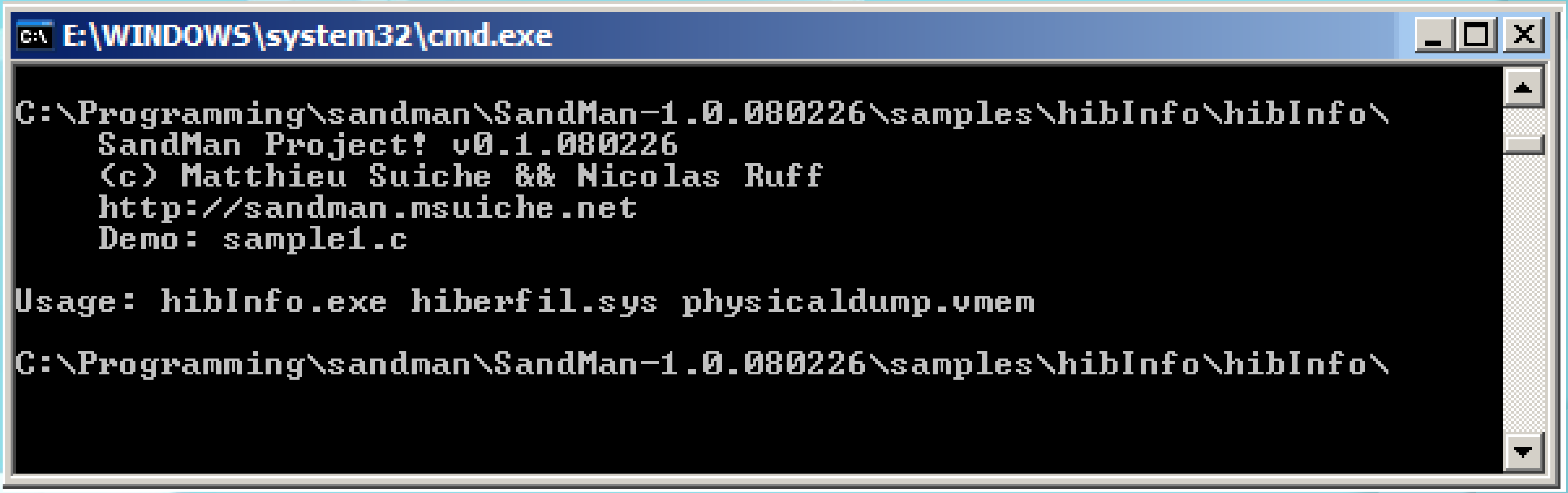
```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

001b9000  00 00 00 00 00 00 8b ff 55 8b ec 83 7d 0c 00 77  ........U......w
001b9010  13 8b 45 08 72 05 83 f8 ff 77 09 8b 4d 10 89 01  ..E.r....w..M...
001b9020  33 c0 eb 0b 8b 45 10 83 08 ff b8 95 00 00 c0 5d  3....E..........
001b9030  c2 0c 00 cc cc cc cc cc 8b ff 55 8b ec 8b 0d 08  ..........U.....
001b9040  70 5f 8e 8b 45 08 8b 49 18 83 c0 f0 89 08 8b 0d  p...E..I........
001b9050  08 70 5f 8e 89 41 18 5d c2 04 00 cc cc cc cc cc  .p...A..........
001b9060  8b ff 55 8b ec 8b 4d 08 8b 15 08 70 5f 8e 8b 42  ..U...M....p...B
001b9070  18 56 83 c1 1f 33 f6 83 e1 f0 eb 09 39 48 08 73  .V...3......9H.s
001b9080  0a 8b f0 8b 00 85 c0 75 f3 eb 5b 85 f6 75 2b 8d  .......u.....u..
001b9090  71 10 39 70 08 73 0a 8b 4a 18 8b 09 89 4a 18 eb  q.9p.s..J....J..
001b90a0  3f 8d 34 01 89 72 18 8b 15 08 70 5f 8e 8b 70 08  ..4..r....p...p.
001b90b0  8b 52 18 2b f1 89 72 08 eb 1f 8d 51 10 39 50 08  .R....r....Q.9P.
001b90c0  73 06 8b 08 89 0e eb 18 8d 14 01 57 89 16 8b 78  s..........W...x
001b90d0  08 2b f9 89 7a 08 8b 16 5f 8b 30 89 32 89 48 08  ....z.....0.2.H.
001b90e0  83 20 00 83 c0 10 5e 5d c2 04 00 cc cc cc cc cc  ................
001b90f0  8b ff 55 8b ec 8b 0d 08 70 5f 8e 8b 45 08 89 41  ..U.....p...E..A
001b9100  10 8b 45 0c 8b 0d 08 70 5f 8e 89 41 14 b0 01 5d  ..E....p...A....
001b9110  c2 08 00 cc cc cc cc cc 8b ff 55 8b ec 83 ec 64  ..........U....d
001b9120  53 56 8b 35 08 70 5f 8e 57 6a 40 33 db 81 c6 88  SV.5.p..Wj.3....
001b9130  00 00 00 53 56 e8 5c 22 00 00 a1 08 70 5f 8e 8a  ...SV.......p...
001b9140  40 4e 88 46 06 a1 08 70 5f 8e 8a 40 4f 88 46 07  .N.F...p....O.F.
001b9150  a1 08 70 5f 8e 8a 40 4d 88 46 05 66 c7 06 40 00  ..p....M.F.f....
001b9160  88 5e 02 a1 08 70 5f 8e 8b 80 08 02 00 00 89 46  .....p.........F
```

```
0x00040000
0x006f6000
0x0102b000
    0x0102b000
0x0943d000
0x09441000
0x0944a000
0x09485000
    0x09485000
0x09498000
0x0a3c4000
0x0a3cc000
0x0a3d4000
0x0a61f000
0x0a925000
0x0ad0d000
0x0b11d000
0x0b189000
0x0b191000
0x0b199000
```

```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

09485000  8a 84 88 d0 b3 6f 80 8b cf 88 45 f8 e8 27 a7 ff  .....o....E.....
09485010  ff 89 45 f4 b8 00 f0 df ff 80 e3 0f 8b 40 20 8a  ..E.............
09485020  40 10 88 45 10 74 1e 80 fb 03 75 7a ff 75 10 68  ...E.t....uz.u.h
09485030  00 0f 00 00 6a 00 e8 d9 fe ff ff ff 75 fc e8 cd  ....j.......u...
09485040  fc ff ff eb 61 8b 45 08 2d d1 00 00 00 74 1f 83  ....a.E......t..
09485050  e8 2e 74 16 ff 75 08 a1 b8 e5 6e 80 ff 50 40 0f  ..t..u....n..P..
09485060  b6 c8 81 c9 00 09 00 00 eb 09 32 c0 eb 44 b9 d1  ..........2..D..
09485070  08 00 00 8b 45 f8 ff 75 10 83 e0 01 c1 ee 06 8a  ....E..u........
09485080  94 70 c8 b3 6f 80 f6 da 1b d2 81 e2 00 e0 ff ff  .p..o...........
09485090  81 c2 00 20 00 00 c1 e0 0f 0b d0 0b d1 52 ff 75  .............R.u
094850a0  fc e8 6e fe ff ff 8b 55 f4 8b cf e8 90 a6 ff ff  ..n....U........
094850b0  b0 01 5f 5e 5b c9 c2 0c 00 cc cc cc cc cc 8b ff  ................
094850c0  55 8b ec 8b 45 08 57 66 8b 3c 45 88 70 6f 80 66  U...E.Wf..E.po.f
094850d0  81 ff ff ff 74 4c 53 56 be 1c ff 6f 80 8b ce e8  ....tLSV...o....
094850e0  54 a6 ff ff 8b d8 b8 00 f0 df ff 8b 40 20 8a 40  T...............
094850f0  10 88 45 08 0f b7 c7 8a 04 85 60 dc 6f 80 24 0f  ..E.........o...
09485100  74 0c 3c 03 75 11 57 e8 62 fc ff ff eb 09 ff 75  t...u.W.b......u
09485110  08 57 e8 45 fe ff ff 8b d3 8b ce e8 20 a6 ff ff  .W.E............
09485120  5e 5b 5f 5d c2 08 00 cc cc cc cc cc 8b ff 55 8b  ..............U.
09485130  ec 8b 4d 18 85 c9 74 1f 33 c0 40 83 39 00 74 05  ..M...t.3...9.t.
09485140  38 45 1c 74 12 8b 55 08 89 01 8b 4d 14 89 11 8b  8E.t..U....M....
09485150  55 0c 89 51 04 eb 02 32 c0 5d c2 18 00 cc cc cc  U..Q...2........
09485160  cc cc 8b ff 55 8b ec 83 ec 6c 8b 45 10 83 e8 00  ....U....l.E....
```

SandMan Command-Line

- A Win32DD-like command line to generate a dump.



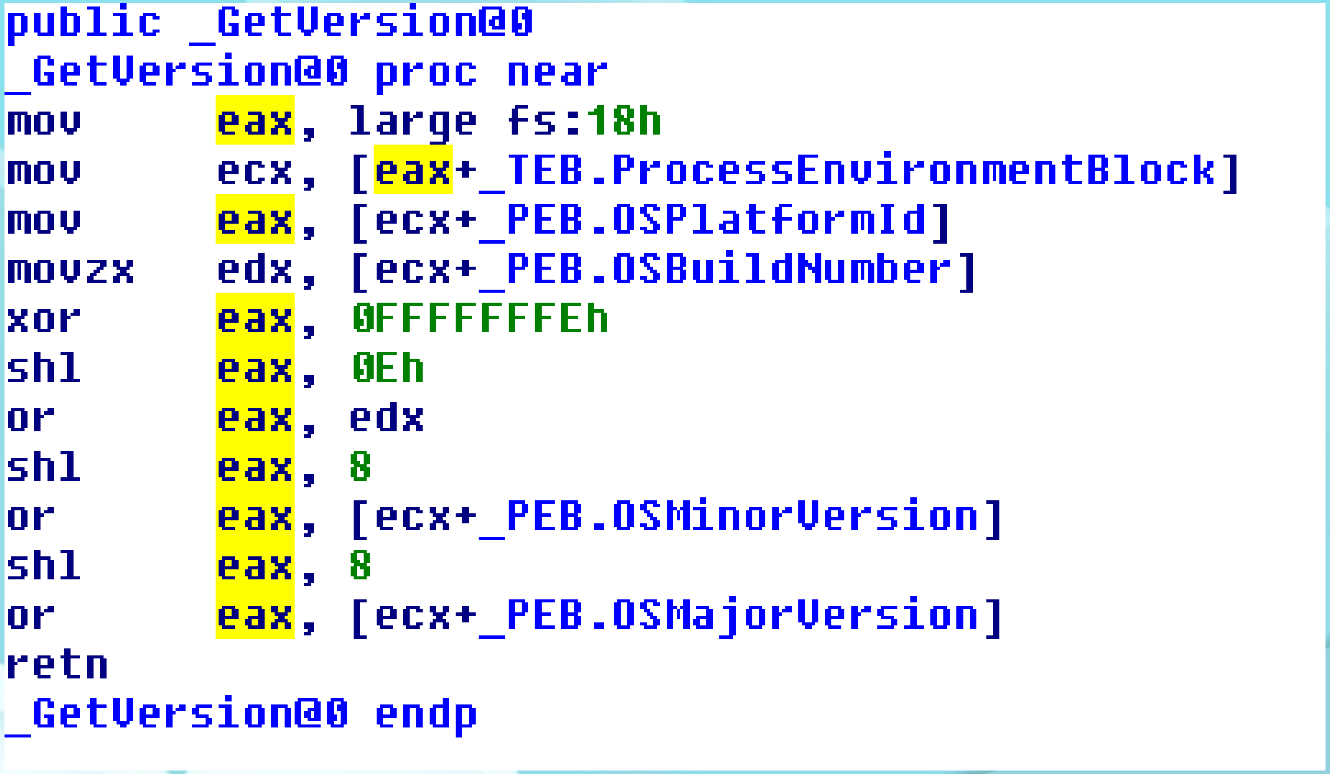
```
C:\Programming\sandman\SandMan-1.0.080226\samples\hibInfo\hibInfo\
    SandMan Project! v0.1.080226
    (c) Matthieu Suiche && Nicolas Ruff
    http://sandman.msuiche.net
    Demo: sample1.c

Usage: hibInfo.exe hiberfil.sys physicaldump.vmem

C:\Programming\sandman\SandMan-1.0.080226\samples\hibInfo\hibInfo\
```

Defensive uses – Exploiting information

- Case #1.
- How to reproduce system behavior (segmentation, paging)
  - GetVersion()

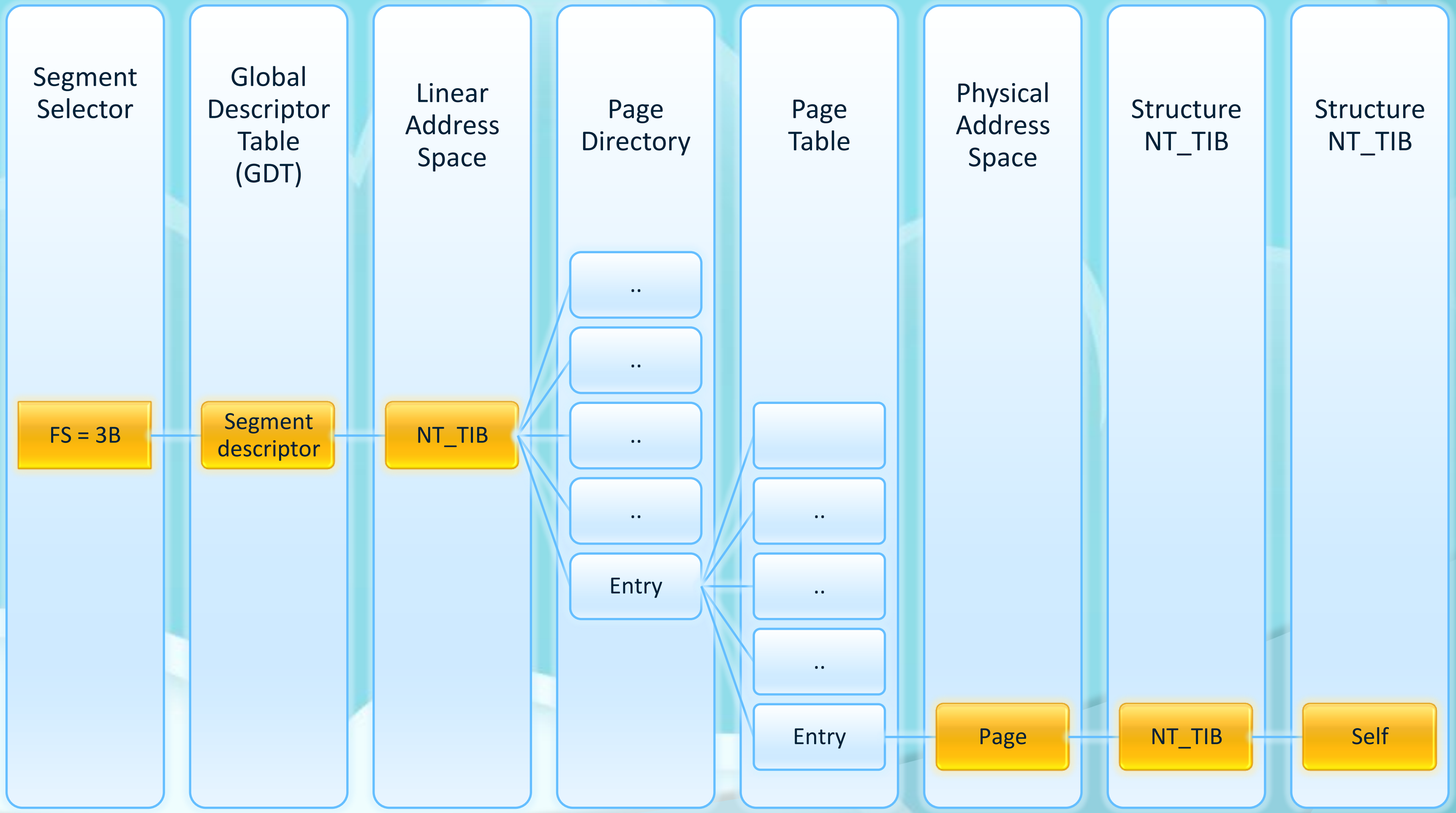
Kernel32.GetVersion() implementation

- Classical function.

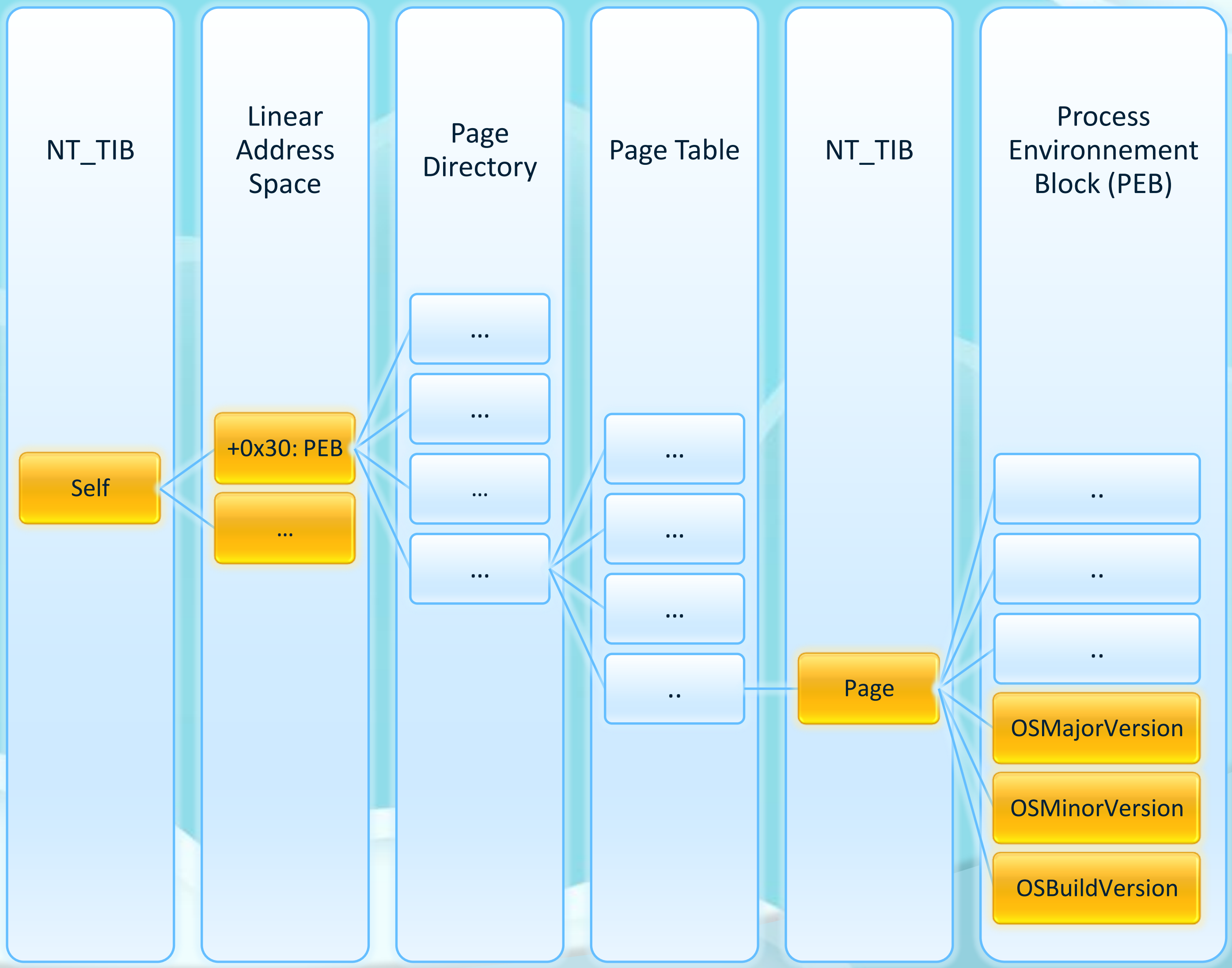


```
public _GetVersion@0
_GetVersion@0 proc near
mov     eax, large fs:18h
mov     ecx, [eax+_TEB.ProcessEnvironmentBlock]
mov     eax, [ecx+_PEB.OSPlatformId]
movzx   edx, [ecx+_PEB.OSBuildNumber]
xor     eax, 0FFFFFFFEh
shl     eax, 0Eh
or      eax, edx
shl     eax, 8
or      eax, [ecx+_PEB.OSMinorVersion]
shl     eax, 8
or      eax, [ecx+_PEB.OSMajorVersion]
retn
_GetVersion@0 endp
```

HiberGetVersion()



HiberGetVersion()



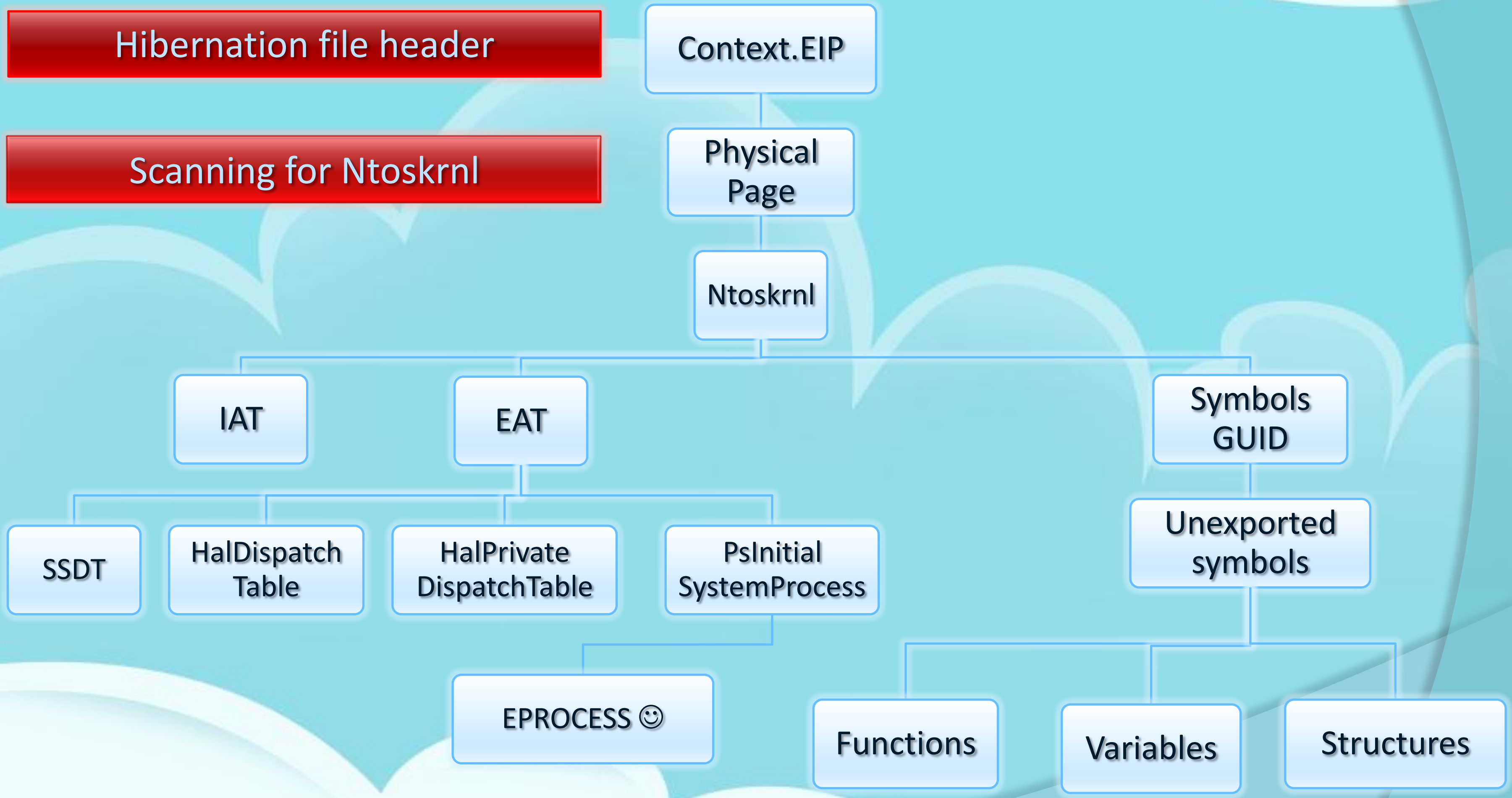
Defensive uses – Exploiting information

- **Case #1.**
- « PatchGuard » like .
  - Kernel Analyze
- Current version objectives
  - Interrupt Description Table (IDT)
  - Global Description Table (GDT)
  - System Service Descriptor Table (SSDT)
  - Kernel Import Address Table (IAT)
  - Kernel Export Address Table (EAT)
  - Hal dispatch Table (HDT)
  - Hal private dispatch table (HPDT)
  - Symbols GUID dumping (can be used for an advanced version)

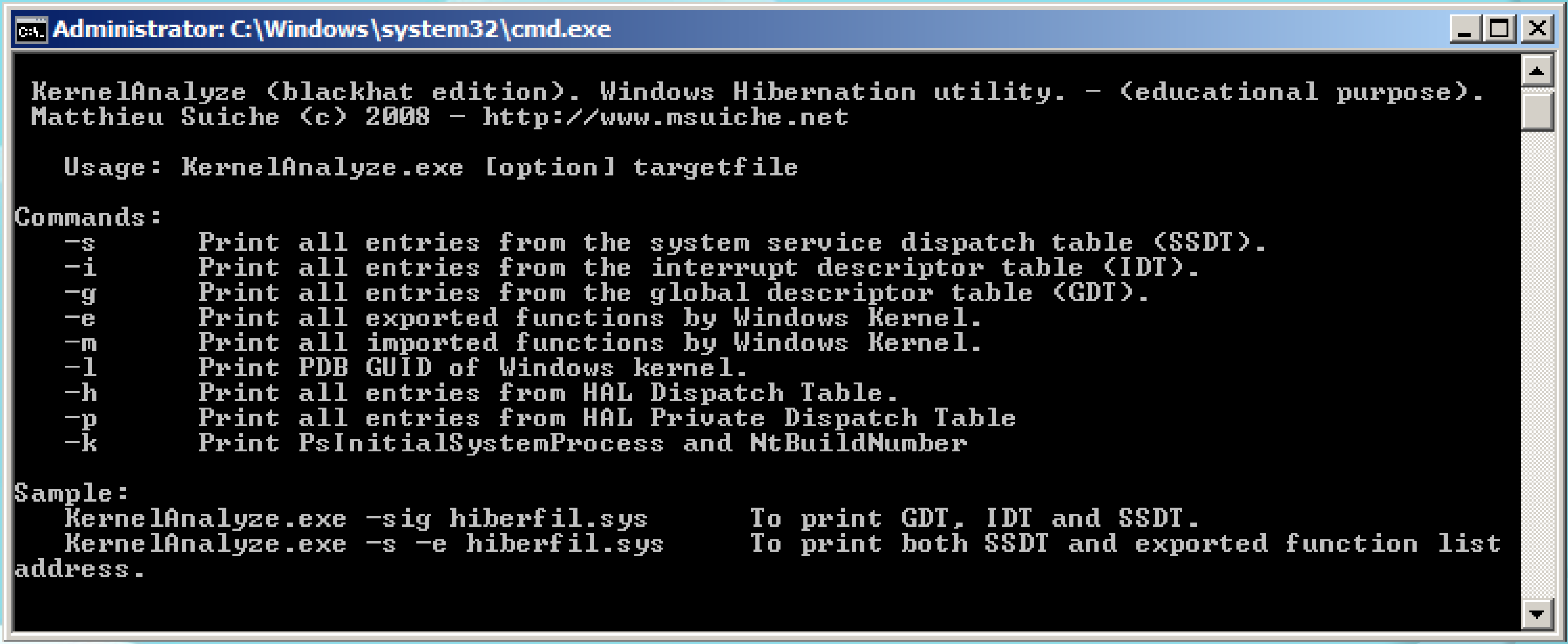
Defensive uses

Hibernation file header

Scanning for Ntoskrnl



Defensive uses - KernelAnalyze



```
 KernelAnalyze (blackhat edition). Windows Hibernation utility. - (educational purpose).
 Matthieu Suiche (c) 2008 - http://www.msuiche.net

   Usage: KernelAnalyze.exe [option] targetfile

Commands:
   -s        Print all entries from the system service dispatch table (SSDT).
   -i        Print all entries from the interrupt descriptor table (IDT).
   -g        Print all entries from the global descriptor table (GDT).
   -e        Print all exported functions by Windows Kernel.
   -m        Print all imported functions by Windows Kernel.
   -l        Print PDB GUID of Windows kernel.
   -h        Print all entries from HAL Dispatch Table.
   -p        Print all entries from HAL Private Dispatch Table
   -k        Print PsInitialSystemProcess and NtBuildNumber

Sample:
   KernelAnalyze.exe -sig hiberfil.sys      To print GDT, IDT and SSDT.
   KernelAnalyze.exe -s -e hiberfil.sys     To print both SSDT and exported function list
address.
```

Defensive uses

- DEMO !

Offensive uses

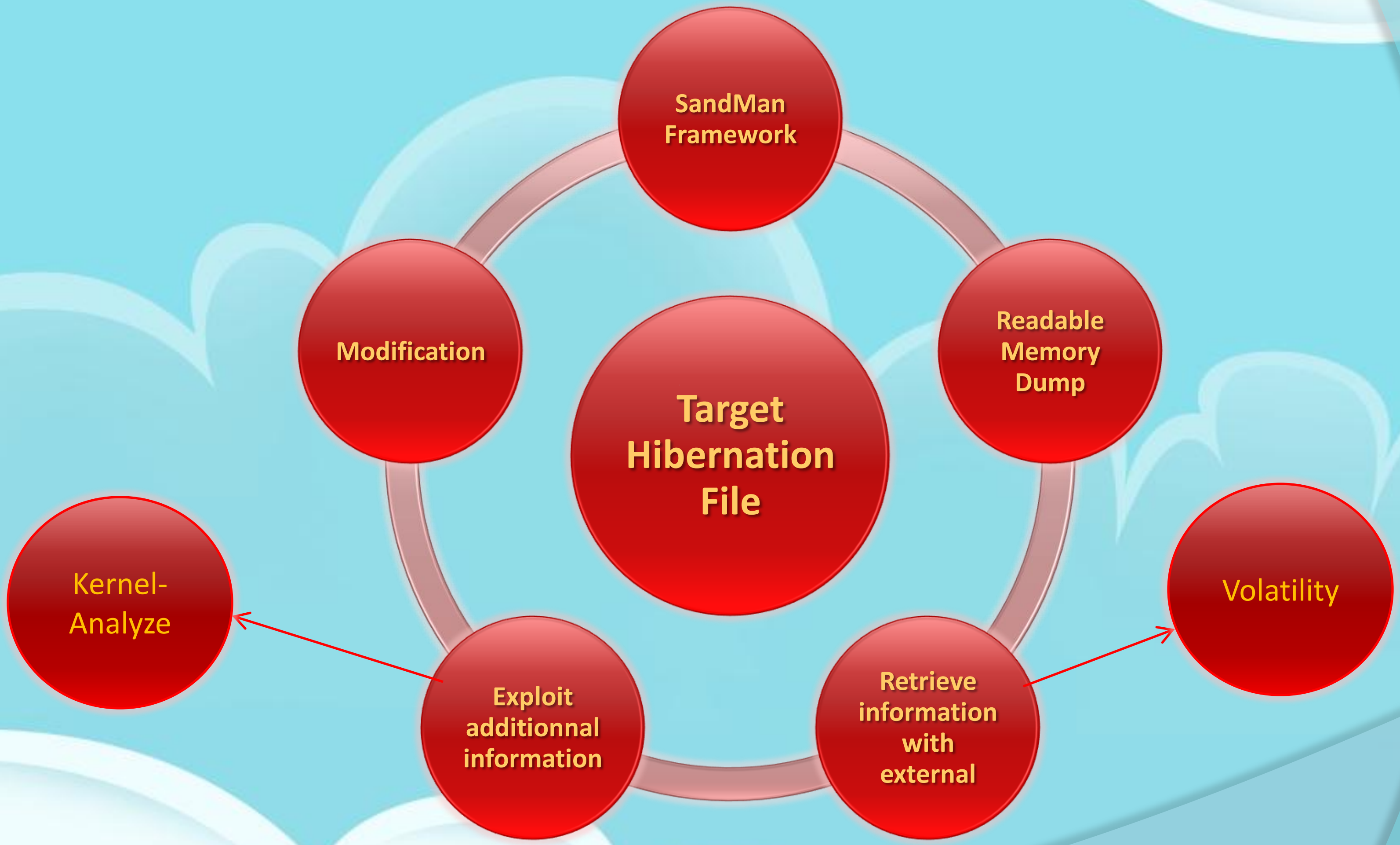
- **Read, Write and eXecution Access**
  - **Can hold sensitive data (password and keys)**
  - **Patching a sleeping machine.**
    - **Privilege escalation?**
    - **Target #1:** EPROCESS
    - **Bypass the login Prompt password?**
    - **Target #2:** msv1_0!MsvpPasswordValidate
  - **Random notes:**
  - **We can also imagine a way using Microsoft Debugging symbols to localize unexported functions instead of using a fingerprint operation.**

MoonSols

Offensics! (= Offensive + Forensics)

- DEMO !

SandMan FrameWork



SandMan APIs - General

- `HiberOpen() / HiberClose()`
- *These functions Open/Close an hibernation file and handles an object internaly called `SANDMAN_OBJECT`*

- `HiberBuildPhysicalMemoryDump()`
- *This function aims at generating a full memory dump, to provide readable snapshot.*

- `HiberGetPhysicalMemorySize()`
- *This function can be used to get the target's physical memory size.*

- `HiberReadFileHeader() /HiberWriteFileHeader()`
- *These functions stores the hibernation header into a buffer, and applies checksum header to the target file if modified.*

- `HiberReadProcState() / HiberWriteProcState()`
- *These functions store the processor state into a buffer, and apply checksum header to the target file if modified.*

SandMan APIs - Reading

- `HiberGetVersion()`
- *The home-made proof of concept GetVersion() that SandMan provides. ☺*

- `HiberCreateTree() / HiberDestroyTree() / HiberGetPageFirst() / HiberGetPageNext()`
- *These functions has been implemented to provide an efficient way to browse through hibernated pages.*

- `HiberIsPagePresent()`
- *This function aims to return if a physical page is available or not.*

- `HiberGetPageAt() / HiberGetPageAtVirtualAddress()`
- *These functions make possible to read a page with its virtual address or physical address.*

- `HiberCountMemoryRanges()`
- *This function can be used to generate internals statistics about the number of Memory Range Array structures.*

SandMan APIs – Writting

- **`HiberGetPhysicalAddress()`**
- *This function is used to translate virtual address to physical address.*

- **`HiberPatch()`**
- *This function has been implemented to patch a sequence of bytes inside a page.*

- **`HiberPageReplace()`**
- *This function replace a whole page at a specific physical address.*

- **`HiberPageRemove()`**
- *This function fill the target page with a 4Kb null buffer.*

How to prevent from hibernation file attack?

- **Full disk encryption**
  - **Bitlocker encrypts the full hard drive including hibernation file.**
    - **Disadvantages: It requires a specific hardware configuration.**
  - TrueCrypt Team **is in touch with Microsoft since April 2008, to find a solution to the hibernation file issue.**
  - **TrueCrypt Disclamer**
  - *[Update 2008-04-02: Although we have not filed any complaint with Microsoft yet, we were contacted (on March 27) by Scott Field, a lead Architect in the Windows Client Operating System Division at Microsoft, who stated that he would like to investigate our requirements and look at possible solutions. …*

How to prevent from hibernation file attack?

- **TrueCrypt Disclaimer**

  - *Disclaimer: As Microsoft does not provide any API for handling hibernation, all non-Microsoft developers of disk encryption software are forced to modify undocumented components of Windows in order to allow users to encrypt hibernation files.*
  - *Therefore, no disk encryption software (except for Microsoft's BitLocker) can currently guarantee that hibernation files will always be encrypted. At anytime, Microsoft can arbitrarily modify components of Windows (using the Auto Update feature of Windows) that are not publicly documented or accessible via a public API.*

How to prevent from hibernation file attack?

- **TrueCrypt Disclaimer**

  - *Any such change, or the use of an untypical or custom storage device driver, may cause any non-Microsoft disk encryption software to fail to encrypt the hibernation file. Note: We plan to file a complaint with Microsoft (and if rejected, with the European Commission) about this issue, also due to the fact that Microsoft's disk encryption software, BitLocker, is not disadvantaged by this.*

Conclusion

- **Hibernation file rocks!**
  - It doesn't require specific hardware like (Fireware [7], …)
  - You don't have to use liquid nitrogen within 60 seconds to get a physical dump. [8]
  - You don't have to load an untrusted driver.
- **High potential for *(ab)use***
  - « Ultimate LiveKd »
  - RootKit and Malware detection
  - Live memory forensics
- **Future?**
  - Why not a MacOS X version of SandMan? ☺

MoonSols

What are your Questions?

(no, I don’t know the time schedule for planes out of McCarran airport)

SLIDES AND DEMOS WILL BE AVAILABLE AT:
SandMan Framework
http://sandman.msuiche.net

Uncle sam doesn’t want me drinking beers ☹

References

- [1] win32dd, Matthieu Suiche
  - http://win32dd.msuiche.net
- [2] Mdd, Ben Stotts, ManTech
  - https://sourceforge.net/projects/mdd/
- [3] dd, George M. Garner Jr.
  - http://gmgsystemsinc.com/fau/
- [4] Volatility
  - https://www.volatilesystems.com/default/volatility
- [5] PTFinder, Andreas Schuster
  - http://computer.forensikblog.de/en/2007/11/ptfinder_0_3_05.html
- [6] TrueCrypt and hibernation file.
  - http://www.truecrypt.org/docs/hibernation-file.php
- [7] WinLockPwn, Adam Boileau
  - http://storm.net.nz/projects
- [8] ColdBoot Attack
  - http://citp.princeton.edu/memory/
- [9] Enter SandMan, Matthieu Suiche & Nicolas Ruff, PacSec 2007
  - http://www.msuiche.net/pres/PacSec07-slides-0.4.pdf
- [10] Physical Memory Forensics, Burdach, BH USA 2006
  - http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Burdach.pdf