EXPLOITING WINDOWS HIBERNATION.

THE CHUCK-NORRIS NINJUSTSU.

Matthieu Suiche

matt#msuiche#net

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

Den Haag, NL. - 2 –4 December 2008
Europol High Tech Crime Expert Meeting 2-4 Dec 2008

Who am I?

- Who am I?
  - Freelance OS/Security Researcher
- Some projects that might interest you.
  - SandMan Framework
    - C Library to manipulate Windows Hibernation file. (Python support aborted)
  - Win32dd
    - Kernel-land physical memory acquisition.
- Volatility contributor

Greets , Shouts and Respect

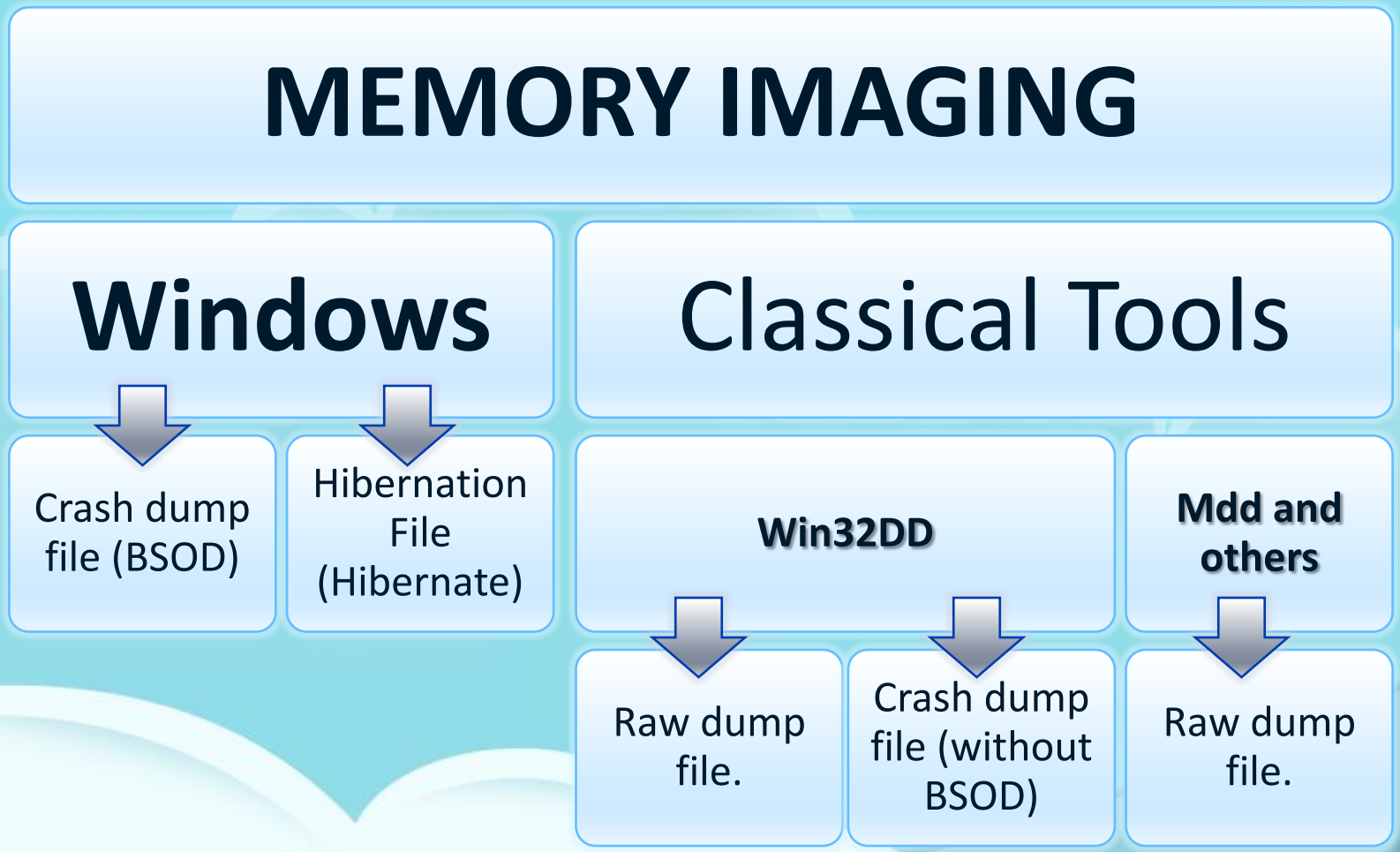
- People from #volatility @ irc.freenode.net
  - Aaron Walters, Andreas Schuster, Jon Evans and everyone I forgot ☺
  - For their contributions to the open-source forensics community and awesomeness.
- Mikael Lindström and others organizers, speakers, and attendees.
- Special greets to Alex Ionescu, Nicolas Ruff and Cedric Blancher.
- Everyone who contributes to RCE.

Outline

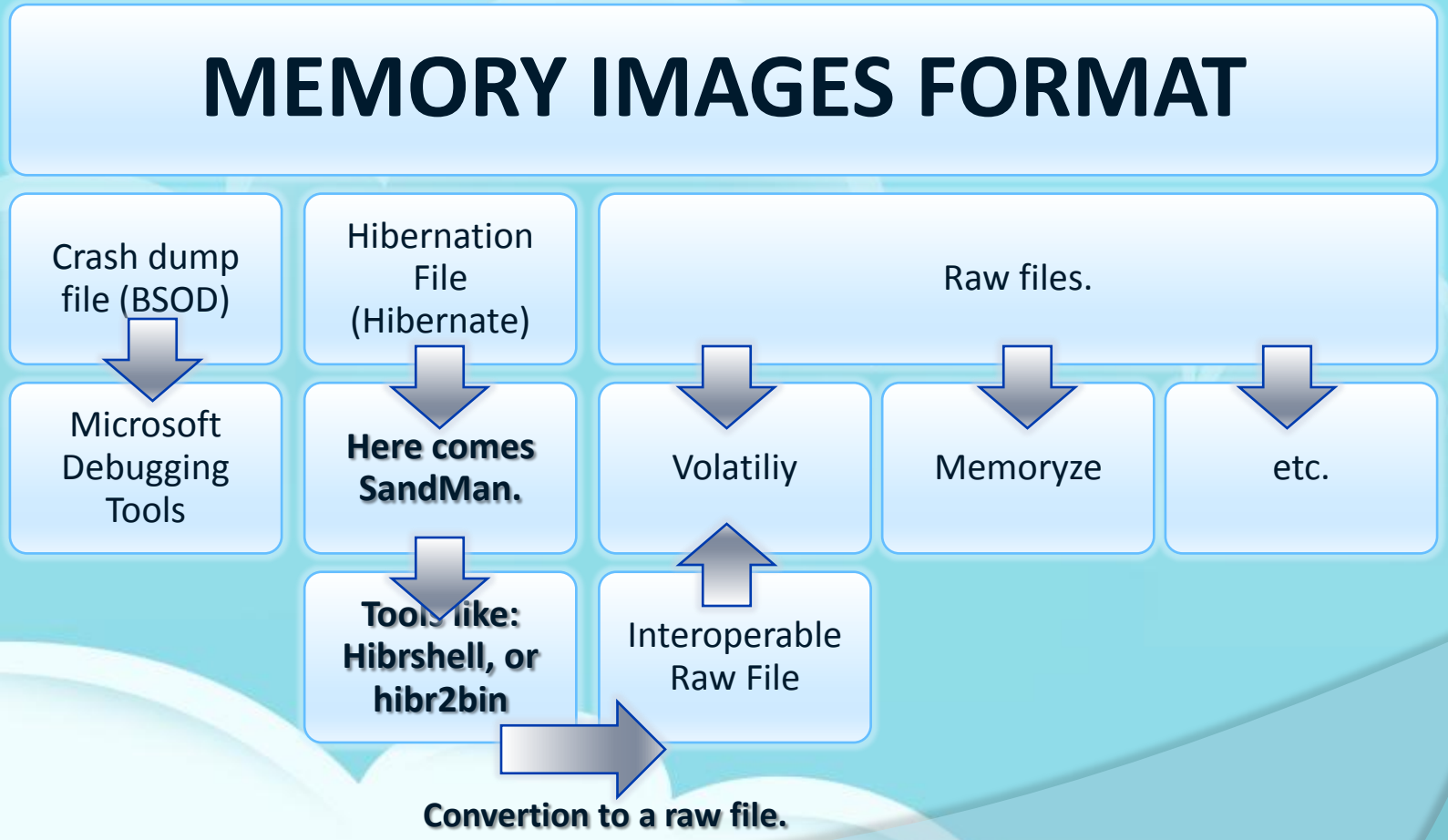
- **Objectives**
  - Introducing a new method of memory dumping
    - Advantages
    - Windows Hibernation File Internals
  - Exploiting hibernation for:
    - Defensive (forensics) use
    - Offensive (offensics) use
  - Demos

Some notes about memory imaging.

MEMORY IMAGING OVERVIEW



MEMORY EXPLOITATION OVERVIEW



HIBERNATION!

Introduction:
Just what is *hibernation*?

- Save the full state of a computer and make it ready to be restored.

- Microsoft name for *« suspend to disk »* feature
  - Available since Windows 2000.
  - This feature is also implemented in non-MSFT O.S.
    - MacOSX « safe sleep »
    - Linux « tux on ice »

- `\hiberfil.sys`
  - Yes, this is this file!
  - It contains a full dump of the memory BUT NOT A RAW COPY!

How do I hibernate ?

- **Quick and easy.**
  - Start > Hibernate
  - Command line:
    - `Powercfg /hibernate` (to activate)
    - `Shutdown /h` (to hibernate)

Advantages.

Advantages (1/4)

- **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 (2/4)

- **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 (hibr2bin)
    - Dump is interoperable with others tools like:
      - `Volatility Framework` [4]
      - `PTFinder` (Andreas Schuster) [5]
      - …
  - ZW* Kernel API are not called, because the system use alternative way through hiber_* prefix drivers.

Advantages (3/4)

- **Writable too?**
  - **Yes! MS Bootloader loads it when the machine is waking, only some checksum test are done.**
  - **Modified code can be executed!!**

Advantages (3/4)

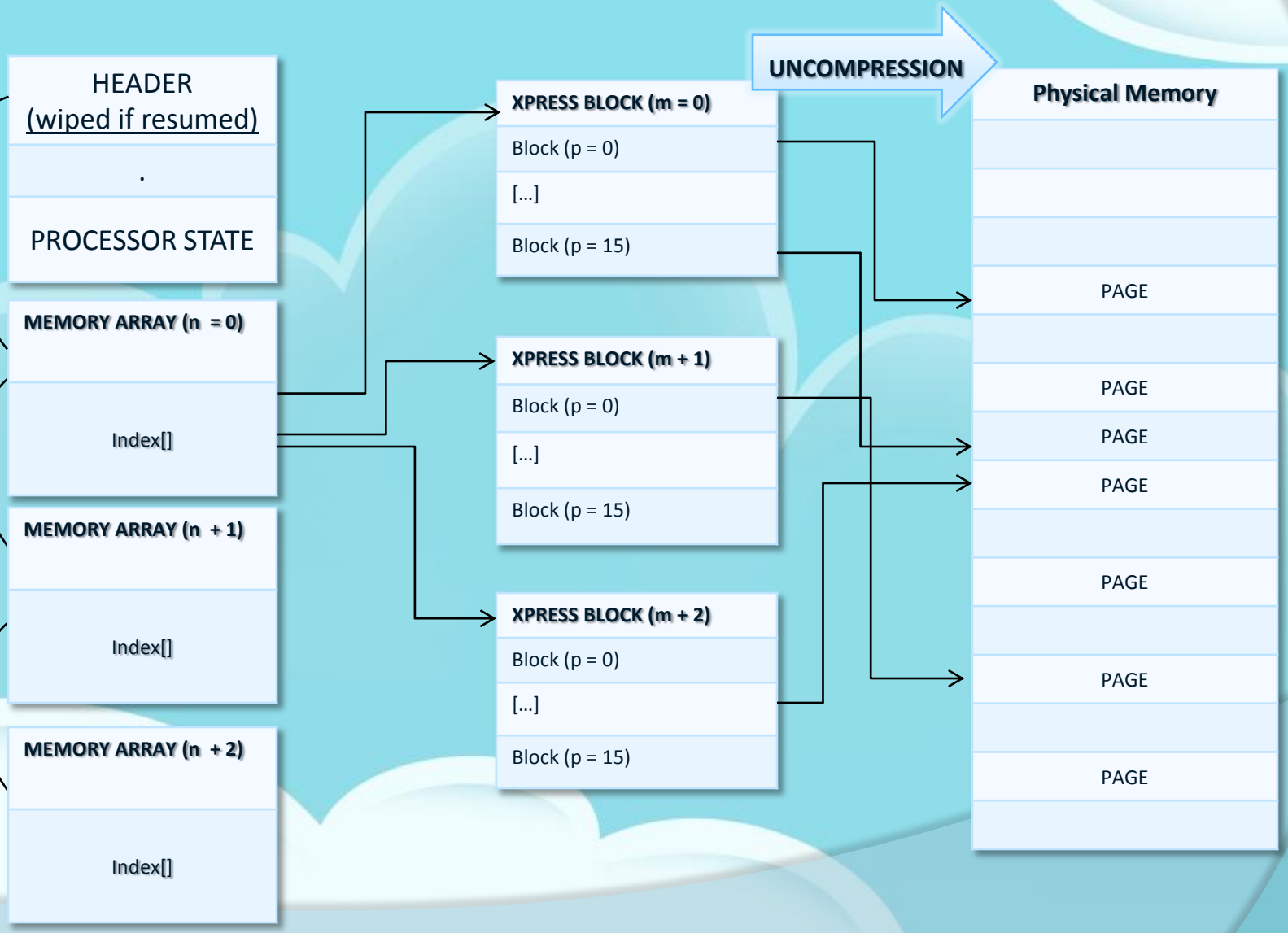
- ***My header is rich!*** **It also contains:**
  - Processor state is saved thus :
  - We can retrieve Control Registers
    - Very useful for memory management functions
      - Like virtual address translation
  - And other interesting things like previous EIP.
  - Interrupt Descriptor Table (IDT)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

Windows Hibernation file internals.

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	…

Mapping Overview



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 2000 O.S. (including Win7) uses internal functions called XpressEncode()

Windows hibernation file internals (5/5)

- **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

Some notes.

Some internal notes (1/2)

- 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()`

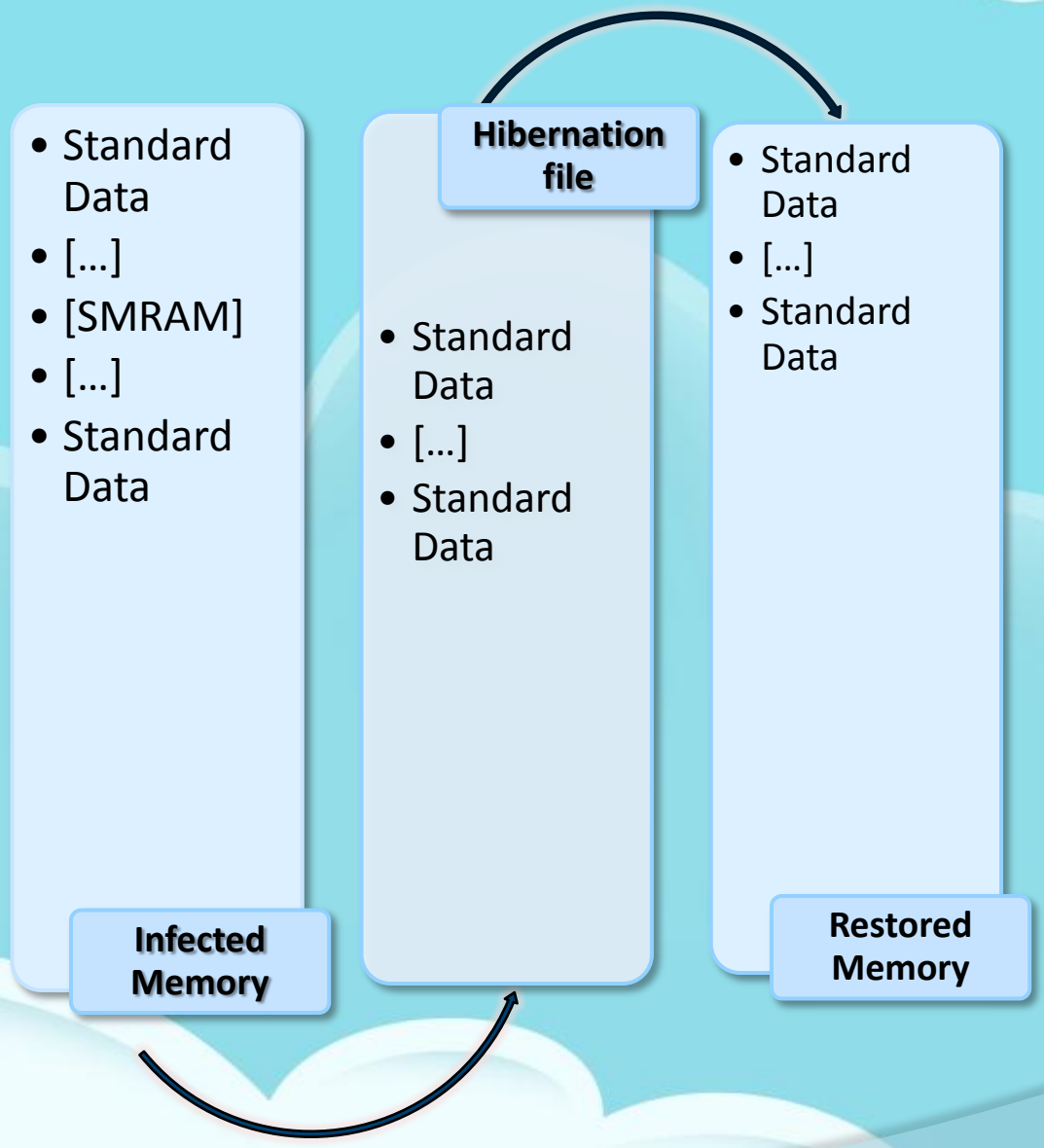
Some internal notes (1/2)

- **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!**

Defensive uses.

Defensive uses of hibernation (1/4)

- **Kernel-land malwares detection**
  - If the code isn't in the hibernation file
    - It won't resume execution
  - E.g. `SMM Rootkit`
    - People says SMM Rootkits 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.



Defeating SMM Rootkit

As you can see the SMRAM is not copied into the hibernation file because this memory area cannot be read.

-- SMRAM area is cleared because of the physical shutdown.

Then, the resumed target is no more infected by the SMM Rootkit.

Defensive uses of hibernation (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 (`GDT`)
      - …

Defensive uses of hibernation (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 of hibernation (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, existing project like Volatility Framework can use CR3 as well.

- Unlike these projects, it's not mandatory to proceed to a « blind » analysis. Hibernation approach is more like how WinDbg deal with Microsoft Crash Dump files.

Defensive uses – Exploiting information

- **Case #1.**
  - **NT OS Kernel and modules Analysis.**

Defensive uses

Hibernation file header

Scanning for Ntoskrnl

Defensive uses:
NT OS Kernel and Modules Analysis [12]

```
  hibrshell - version 0.0.0.0.0.1.
  Copyright (c) 2007 - 2008, Matthieu Suiche <http://www.msuiche.net>
  Copyright (c) 2008, MoonSols <http://www.moonsols.com>

Loading hiberfil - Copy.sys...
Retrieving Kernel Image base...
Retrieving Kernel Image size...
Kernel Image: 0x804D7000-0x806CF580

_NT_SYMBOL_PATH: SRV*C:\symbols*http://msdl.microsoft.com/download/symbols

Symbols loading...
Kernel raw image... DBG: Creating ntoskrnl.exe.. [ImageBase: 0x804D7000, Cr3: 0x068A8060]
Done.
Symbols downloading/loading..Done.
    Loaded symbols: PDB
    Module Name : ntoskrnl.exe
    PDB Name: C:\symbols\ntkrnlpa.pdb\30B5FB31AE7E4ACAABA750AA241FF3311\ntkrnlpa.pdb
PsLoadedModuleList = 0x80553FC0
sandman> help

Commands:
   ssdt      Print all entries from the system service dispatch table (SSDT).
   idt       Print all entries from the interrupt descriptor table (IDT).
   gdt       Print all entries from the global descriptor table (GDT).
   eat       Print all exported functions by Windows Kernel.
   iat       Print all imported functions by Windows Kernel.
   guid      Print PDB GUID of Windows kernel.
   haldt     Print all entries from HAL Dispatch Table.
   halpdt    Print all entries from HAL Private Dispatch Table
   ps        Print process list
   drivers   Print loaded drivers list
   ntbuild   Print NtBuildNumber
   info      Print PAE, etc.
   db        Dump some bytes
   reg       Show registers
   dump      Dump kernel into an output file.
   vtop      Convert virtual address to physical
   create    Create a classical dump

sandman>
```

Defensive uses – Exploiting information

- **Case #2.**
  - **Retrieving the target version.**
    (Technical explanation)

Kernel32.GetVersion() implementation

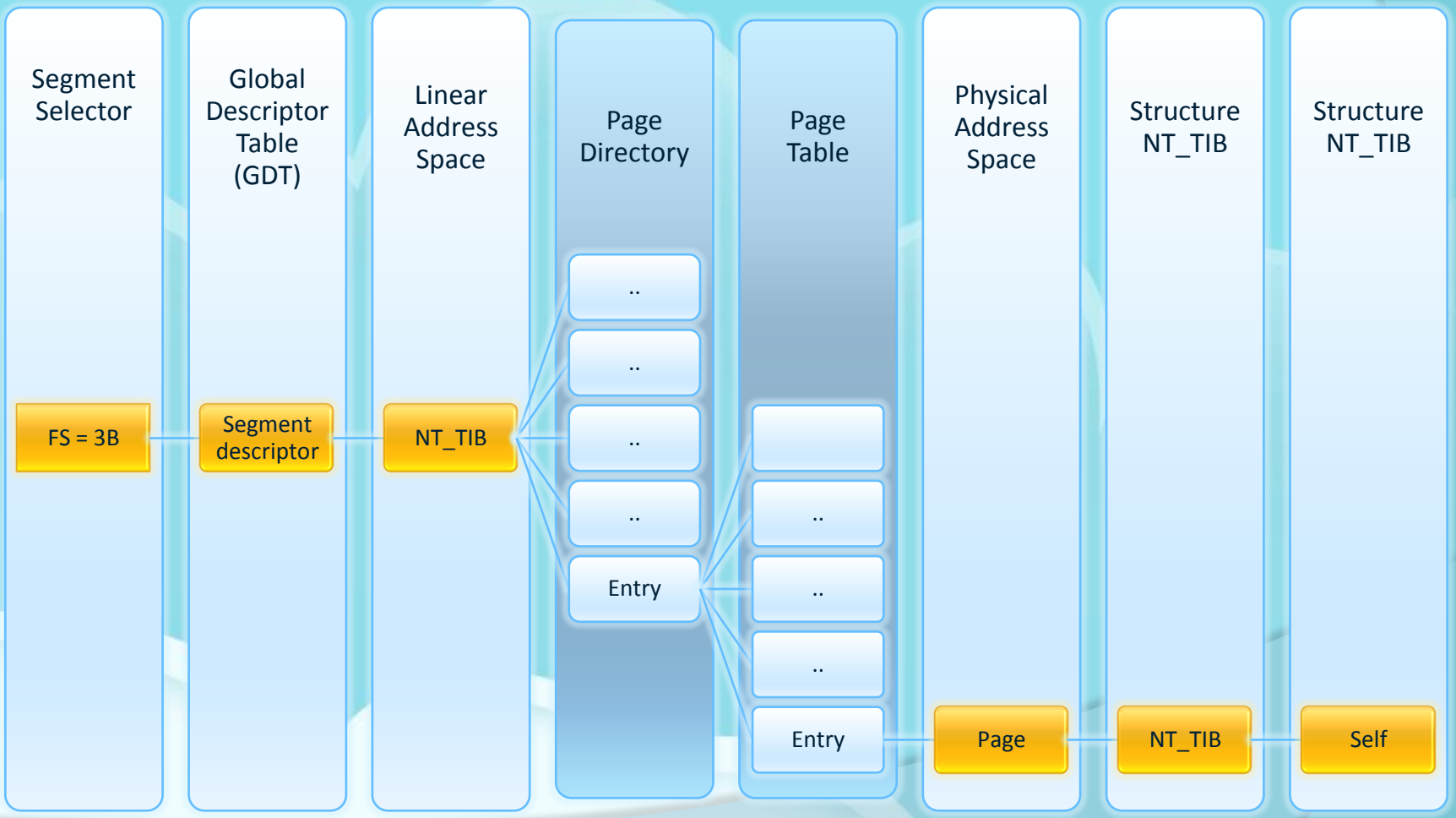
- Classical function.

```
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
```

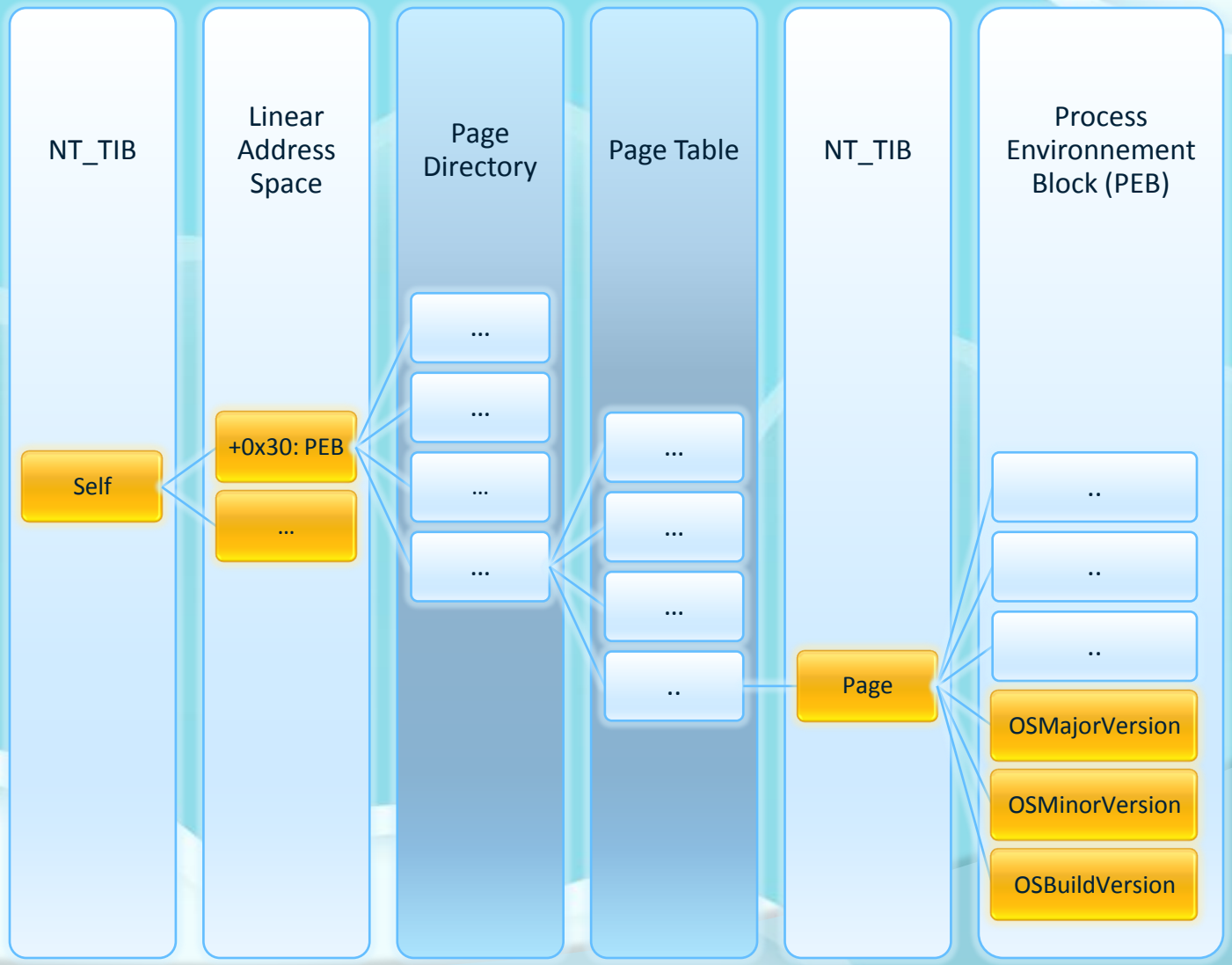
Get PEB Address from TEB.

Apply a local OR operation on OS{Minor,Major}Version, OSPlatformId, and OSBuildNumber

HiberGetVersion()



HiberGetVersion()



Defensive uses

⊙DEMO !

Offensics! (= Offensive + Forensics)

Offensive uses.

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 !

Just to summarize.

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 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.

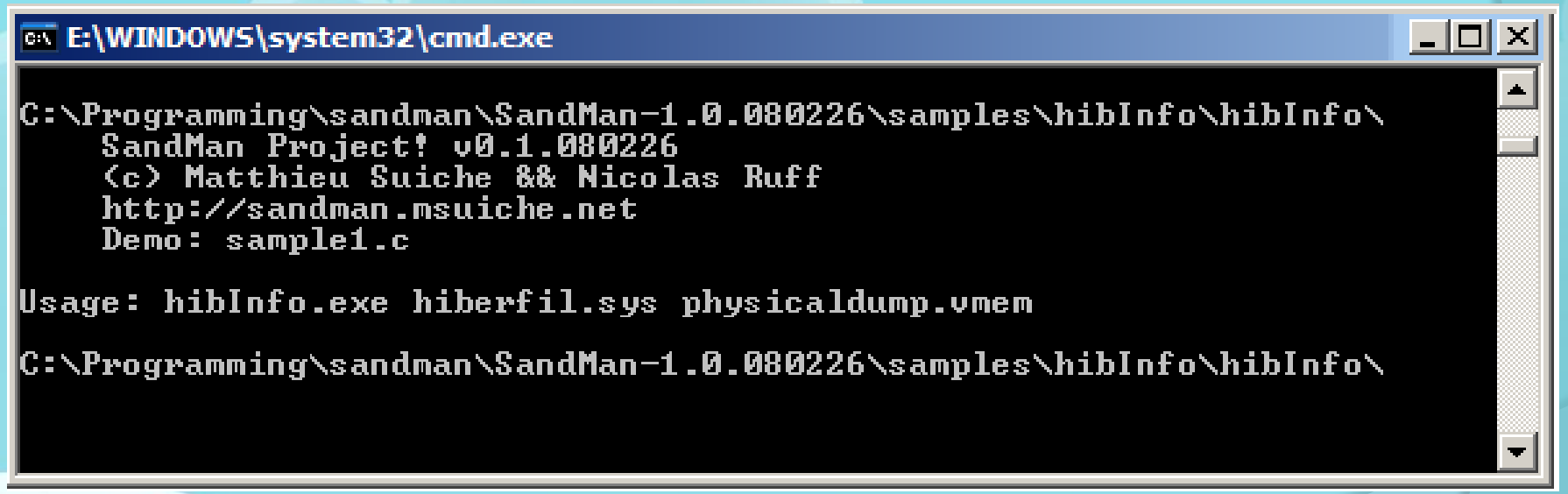
- `HiberPatch()`
- This function has been implemented to patch a sequence of bytes inside a page.

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

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

SandMan Command-Line

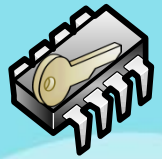
- hibr2bin



Prevention.

How to prevent from hibernation file attack?

- **Full disk encryption**



  - **Bitlocker encrypts the full hard drive including hibernation file. [11] as reminded at WinHEC 2008.**
    - **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.**

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. …
—Truecrypt Team, *Update 2008-04-02:*

How to prevent from hibernation file attack?

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?

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 (Firewire [7], …)
  - You don't have to use liquid nitrogen within 15 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

Questions ?

SandMan Framework

http://sandman.msuiche.net

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
- [11] Bitlocker: Protecting Data in Windows 7 and Windows Server 2008 R2
  - http://download.microsoft.com/download/5/E/6/5E66B27B-988B-4F50-AF3A-C2FF1E62180F/ENT-T561_WH08.pptx
- [12] HibrShell – Your hibernation file in a nutshell. (alpha version / developpement test)
  - Available on request.