Detecting Rootkits in Memory Dumps

Pär Österberg Medina – SITIC

2009-05-18

About the presentation

I am going to talk about different techniques to dump the memory on a system and how to analyze it, looking for the presence of a kernel level rootkit.

60 minutes

- usually takes hours to explain

High technical level

- Hopefully comprehensive

About the presenter

Pär Österberg Medina
SITIC - Swedish IT-incident Centre
http://www.sitic.se

Previous presentations:

- Sitic seminars - http://www.sitic.se/seminarium
- T2 - http://www.t2.fi
- FIRST2007 and FIRST2008 - http://www.first.org
- GOVCERT.NL Symposium, SecHeads, IP-dagarna, Susec …

SITIC - Swedish IT Incident Centre

SITIC is a section of Network Security at the Post & Telecom regulatory authority (PTS). PTS is a civilian agency under Dept. of Commerce.

Mandate is based on the instruction from the elected government to the agency, and states that SITIC shall:

- Be a national function that supports mitigation and prevention of IT incidents
- Act jointly with other agencies having special tasks within the information security area
- Give advise and support to governmental and private companies & organisations about network security
- Be the Swedish contact point for corresponding functions in other countries

Collaborations & Interactions

- Nationally
- FRA, MSB, FMV, RPS, Military
- ISPs
- Media

- Internationally
- EGC, NCF, IWWN
- FIRST, TF-CSIRT
- ENISA

Agenda

- What is a rootkit?
- Dumping the memory
- How-to analyze a memory dump?
- Different rootkit techniques and how we detect it

Objective

After this presentation, the attendee will have a good knowledge about:

- Different techniques to use when dumping and analyzing a memory dump
- Different rootkit types and technologies

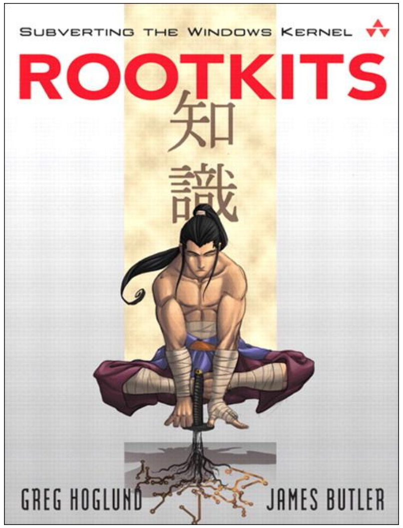
Credit and Kudos

Andreas Schuster – Deutche Telekom
http://computer.forensikblog.de/en/

George M. Garner Jr. – GMG Inc
http://www.gmgsystemsinc.com/knttools/

What is a rootkit?

“The term rootkit has been around for more than 10 years. A rootkit is a "kit" consisting of small and useful programs that allow an attacker to maintain access to "root," the most powerful user on a computer. In other words, a rootkit is a set of programs and code that allows a permanent or consistent, undetectable presence on a computer.”



Different types of rootkits

- Ring 3 (User-mode)
- Ring 0 (Kernel-mode)
- Hardware/Firmware based
- Virtualization based

Different types of rootkit techniques

Basic types of rootkit techniques used:

- Hooking
- Injecting
- Unlinking

Persistent rootkits

Persistent Rootkits wants to survive a reboot, hence the rootkit must be initiated from some ware

- Registry keys (run keys, file extensions)
- Startup files (win.ini, system.ini, config.nt, autoexec.nt)
- Using non-existing SafeDllSearchMode
- Add-on to an existing application (BHO, Firefox/Thunderbird extensions)
- Patching binaries on disk (Boot Loader, Kernel, Drivers)
- Using a custom Master Boot Record

Memory-based rootkits

Memory-based Rootkits exist only in memory and does care about surviving a reboot

- Most traces of this types of rootkits disappears when the system is rebooted
- Be sure to include memory acquisition as a part of your standard incident handling/forensic process

The rootkit paradox

All rootkits obey two basic principles

1. They want to remain hidden
2. They need to run

J. Kornblum, - http://jessekornblum.com/research
Exploiting the Rootkit Paradox with Windows Memory Analysis
International Journal of Digital Evidence, 5(1), Fall 2006.

Rootkit scanners

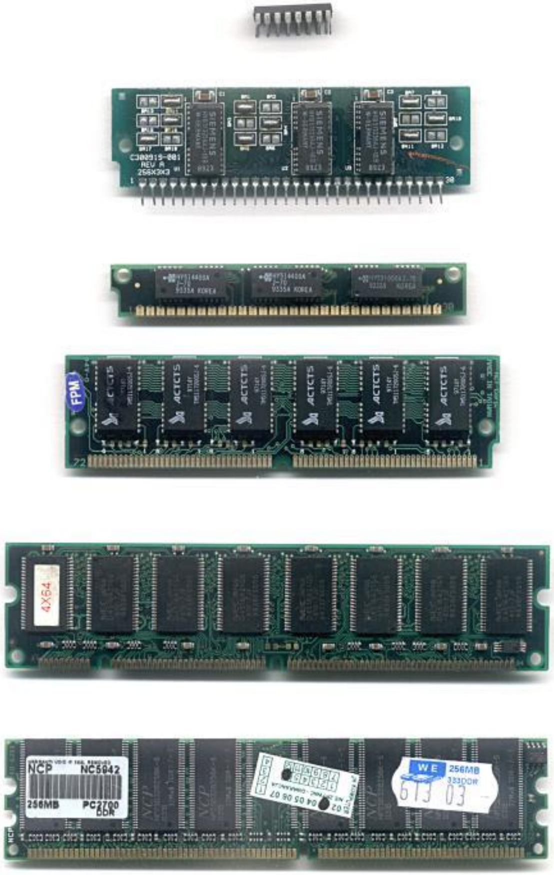
Software running on a live system with the goal to find irregularities that could indicate a rootkit infection

- Fast - almost instant result
- Runs on a live system and therefore changes the integrity of the system
- Can the result be trusted?

What is memory?

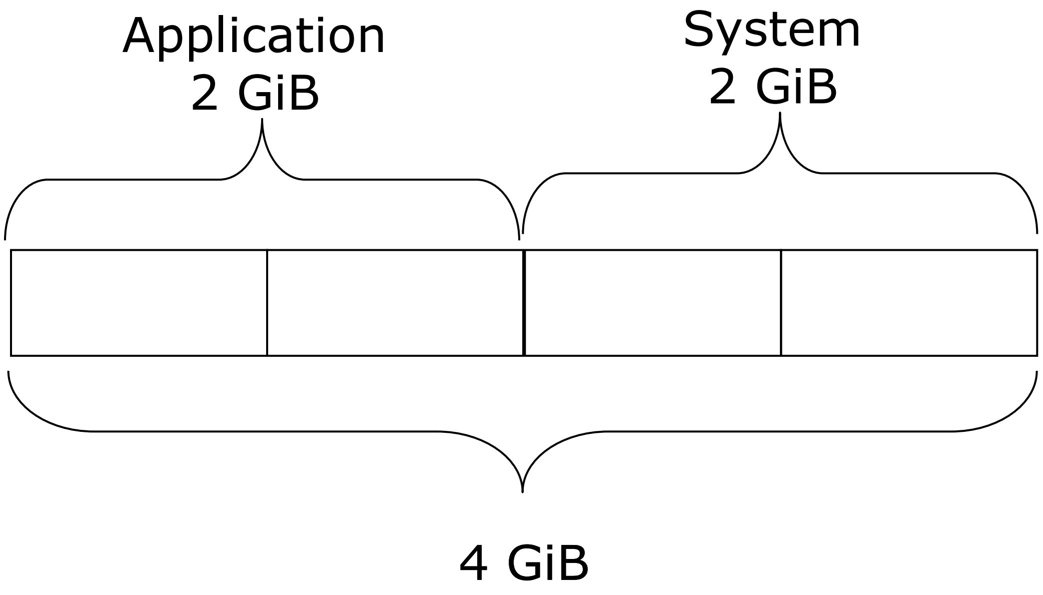
Physical memory is the short-term memory of a computer

- Rapid decay of information as soon as memory module is disconnected from power and clock sources.
  - More on the rapid decay later!



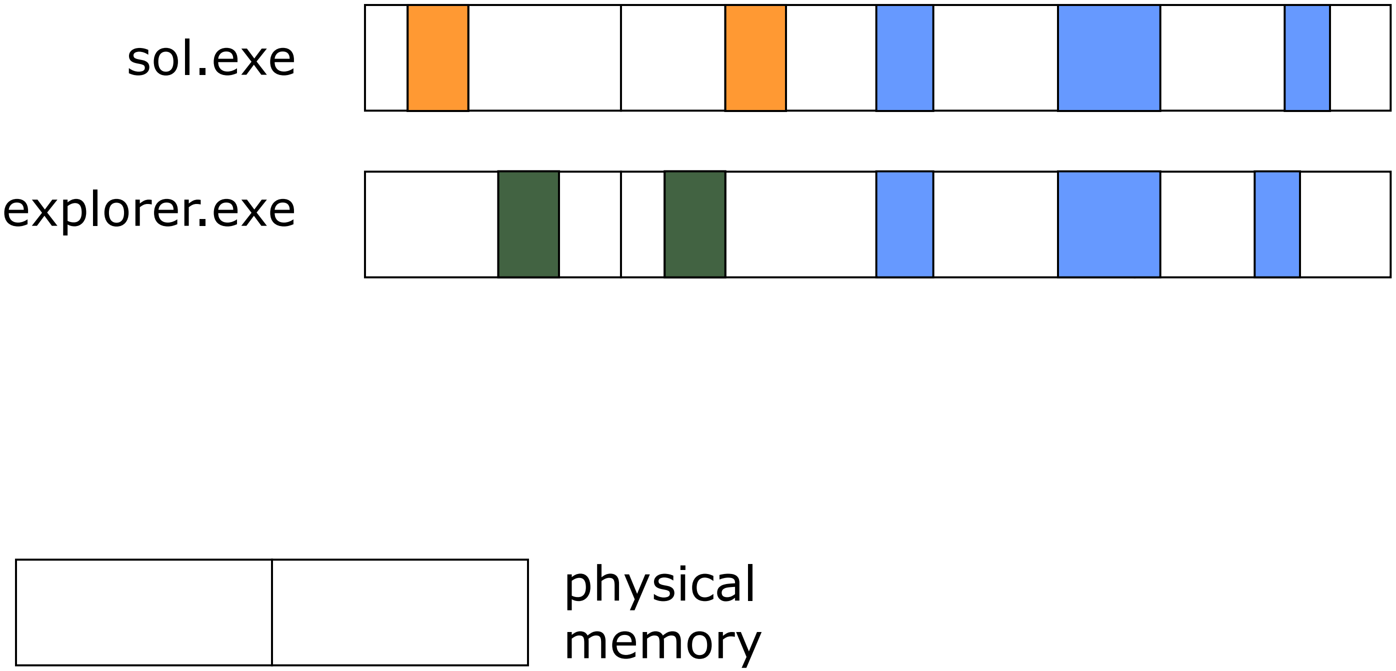
Virtual memory

4 GiB of (virtual) address space per process split into halves



Physical memory

Physical memory is divided into so called “pages” and allocated virtual memory is mapped onto physical memory page by page.



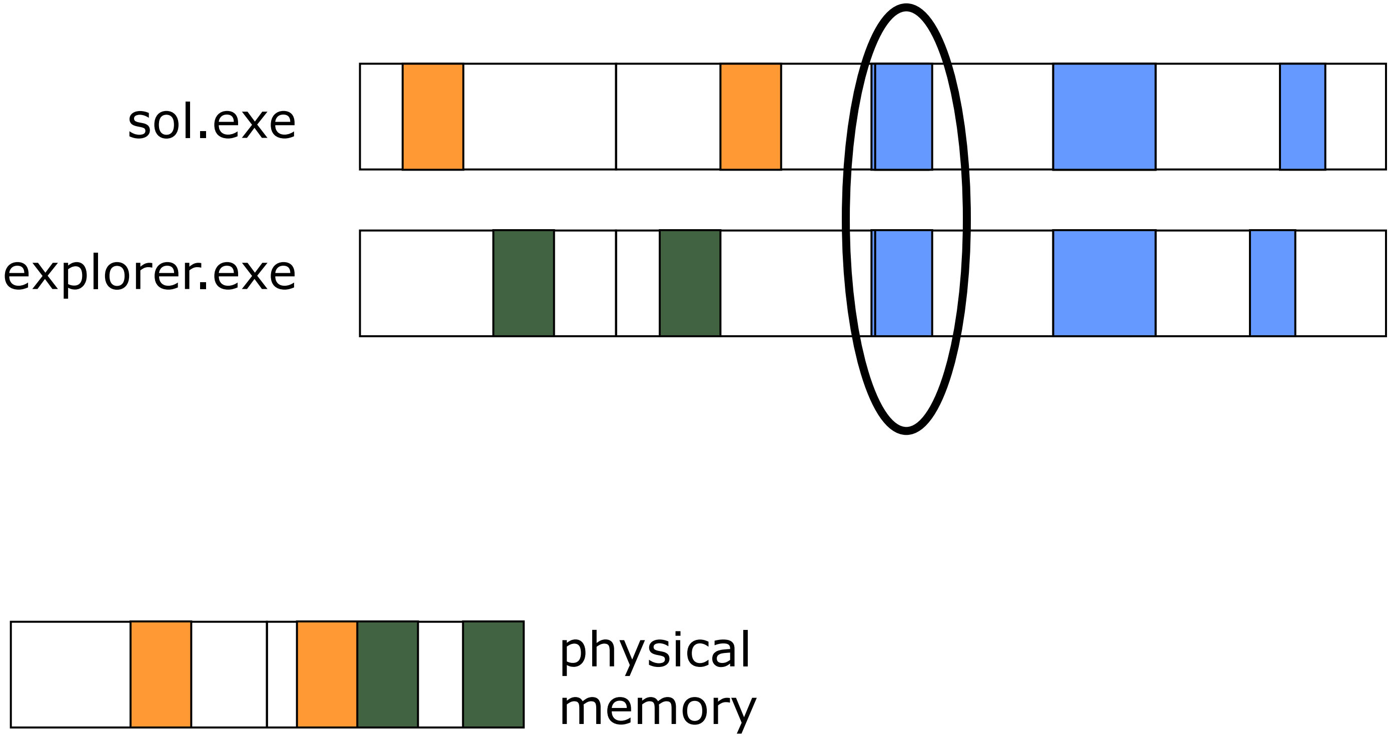
sol.exe

explorer.exe

physical memory

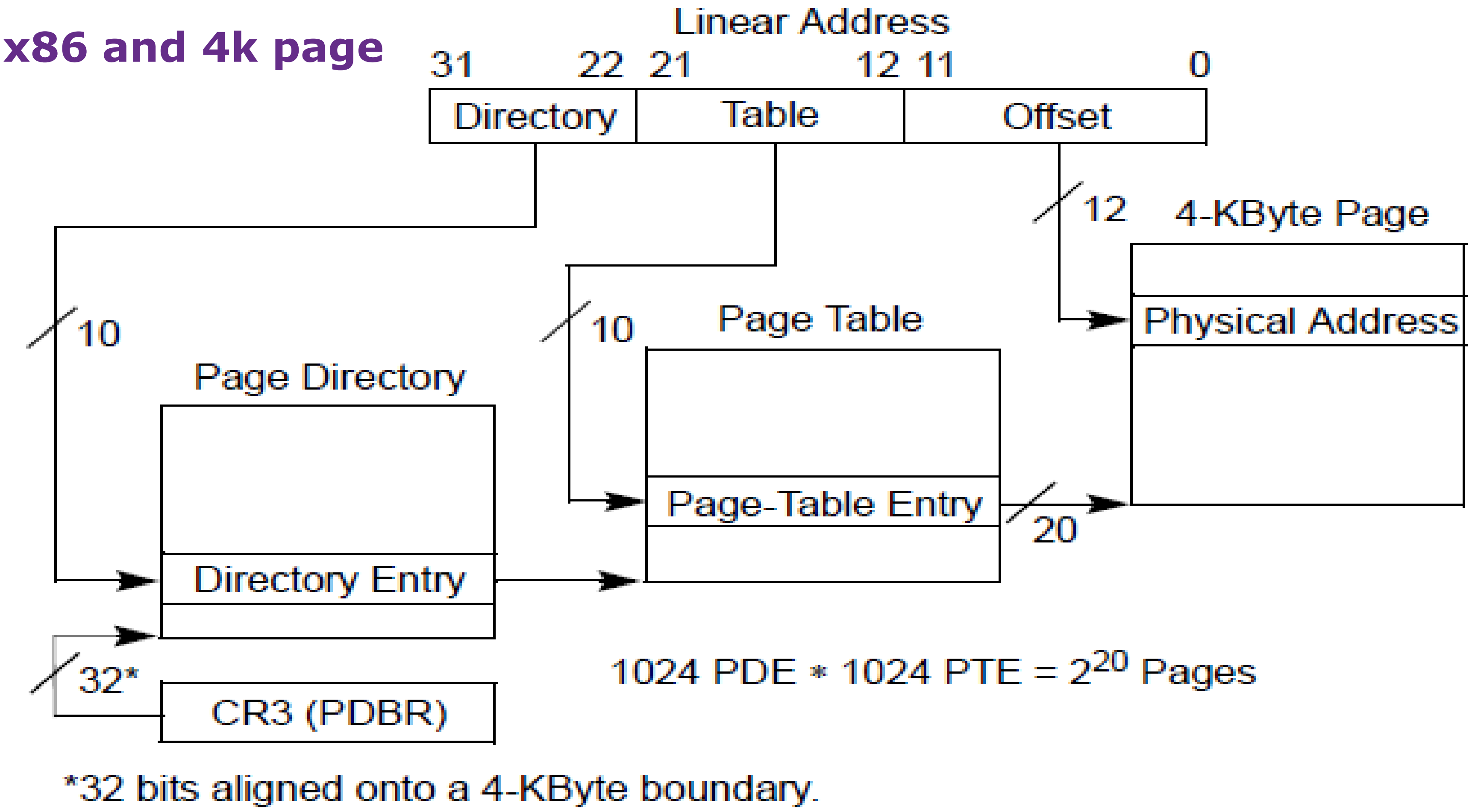
Sharing the same physical page

The same page of physical memory can appear at different locations within the same address space or in different address spaces.

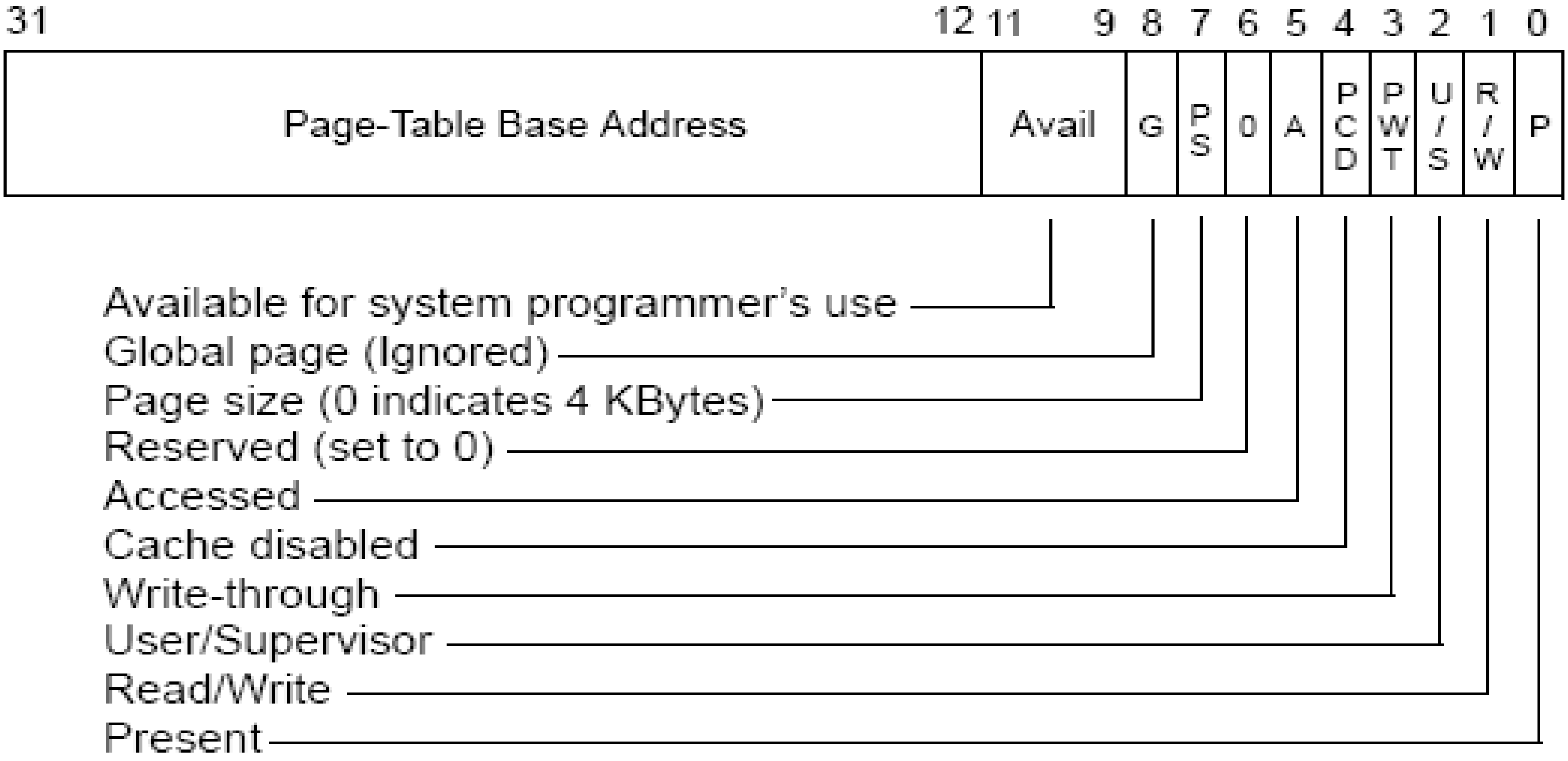


Virtual to Physical memory translation

x86 and 4k page



Important bits in the PTBD

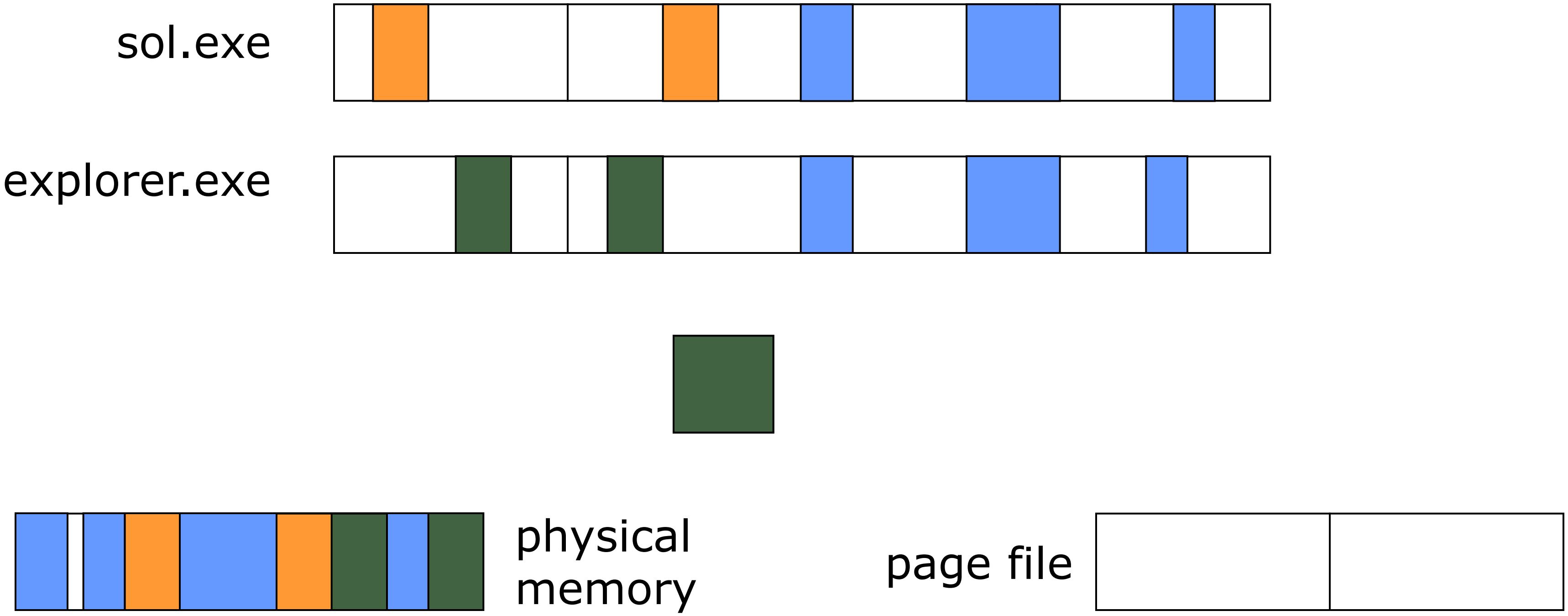


31 … 12	11 … 9	8	7	6	5	4	3	2	1	0
Page-Table Base Address	Avail	G	PS	0	A	PCD	PWT	U/S	R/W	P

- Available for system programmer's use
- Global page (Ignored)
- Page size (0 indicates 4 KBytes)
- Reserved (set to 0)
- Accessed
- Cache disabled
- Write-through
- User/Supervisor
- Read/Write
- Present

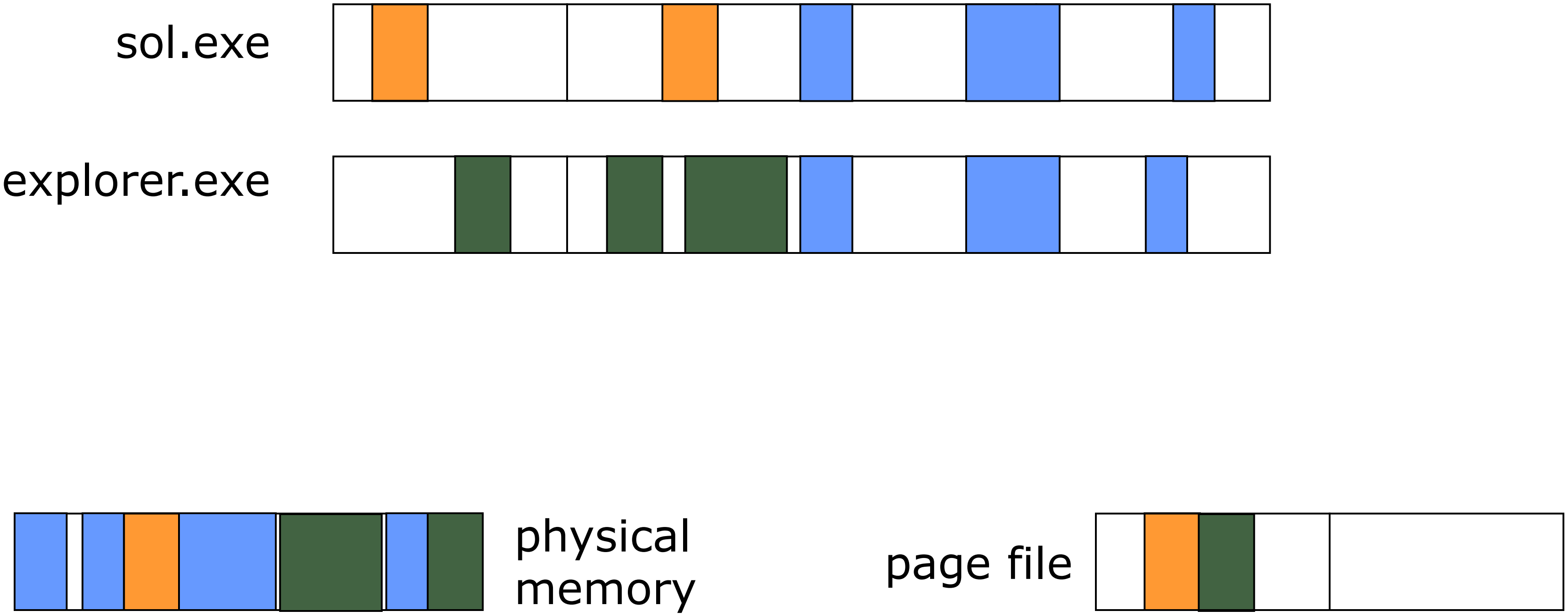
Page file

Data can be moved from physical memory into a page file to clear some space



Freed pages

Memory does not get over written when it is marked as free



sol.exe

explorer.exe

physical memory

page file

Dumping the memory

Software or Hardware

- Executable code running on the machine or dedicated hardware using DMA to capture the memory

High or low atomicity of the memory dump

Format of the memory dump

- 1:1 copy of the physical memory or a Microsoft crash dump

User rights or Administrator Privileges required

Dumping the memory using dd

Windows makes physical memory accessible through the \\.\PhysicalMemory and \\.\DebugMemory devices.

- ~~Port by George. M. Garner Jr. http://users.erols.com/gmgarner/forensics/~~
- X-Ways Capture (does a lot of other things, too)

Dumping the memory by loading a driver

mdd – ManTech Memory DD
http://sourceforge.net/projects/mdd/

- Works on all Windows versions from Windows 2000 to Windows Server 2008

win32dd - Matthieu Suiche
http://win32dd.msuiche.net/

- Mainly a kernel mode application that does everything with native functions

KnTDD

GMG Systems, Inc. (George M. Garner Jr)
http://www.gmgsystemsinc.com/knttools/

Available to law enforcement and CERT organizations

Also obtains for later analysis

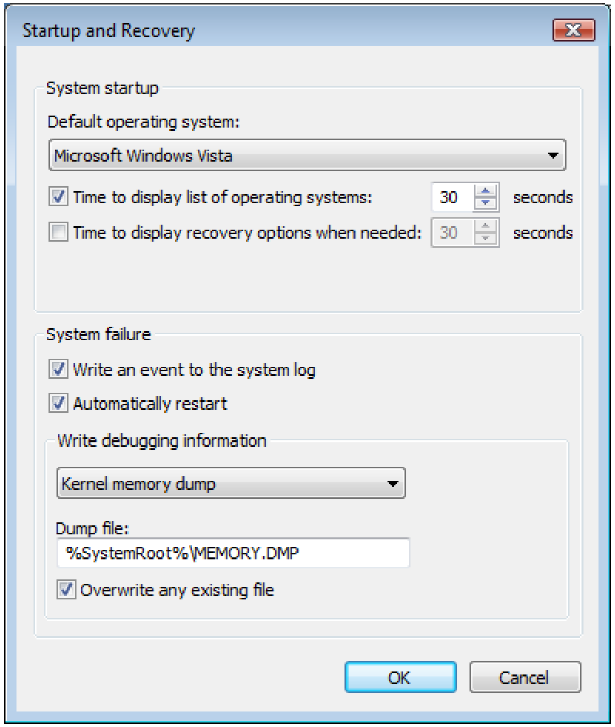
- kernel and network driver binaries
- system status as seen from userland

Enterprise edition allows for digitally signed work packages and encrypted evidence

Microsoft Crash Dump

Configure Windows to write the memory to a file incase of a Blue Screen of Death

- High atomicity of the memory image
- The machine stops temporarily to function



Startup and Recovery

System startup

Default operating system:

Microsoft Windows Vista

☑ Time to display list of operating systems: 30 seconds

□ Time to display recovery options when needed: 30 seconds

System failure

☑ Write an event to the system log

☑ Automatically restart

Write debugging information

Kernel memory dump

Dump file:

%SystemRoot%\MEMORY.DMP

☑ Overwrite any existing file

How to generate a Crash Dump

- Kill csrss.exe (Client Server Subsystem) or write your own driver that calls nt!KeBugCheck or nt!KeBugCheckEx.
- NotMyFault from Sysinternals http://download.sysinternals.com/Files/Notmyfault.zip
- SystemDump from Citrix (Dimitry Vostokov) http://support.citrix.com/article/CTX111072
- Bang from OSR http://www.osronline.com/article.cfm?article=153
- Activate crash sequence in PS/2 keyboard driver (USB supported in Windows 2003 SP 1).

LiveKD

LiveKD allows you to run the Kd and Windbg Microsoft kernel debuggers, which are part of the Debugging Tools for Windows package, locally on a live system

- The .dump command generate a crash dump on a live system
- Requires machine specific symbols in order to work

Anti-forensic techniques (1)

Shadow Walker by Sparks and Butler (2005)
http://www.blackhat.com/presentations/bh-jp-05/bh-jp-05-sparks-butler.pdf

- Controls the contents of memory viewed by another application or driver.
- Modifies page fault handler, marks page as not present, then flushes the Translation Lookaside Buffer (TLB).

Anti-forensic techniques (2)

Ddefy by Darren Bilby (2006)
http://www.blackhat.com/presentations/bh-jp-06/BH-JP-06-Bilby-up.pdf

- Hooks entry for `nt!NtMapViewofSection` in System Service Descriptor Table (SSDT).
- Monitors access to \\.\PhysicalMemory

Dumping the memory using DMA

Tribble by Brian Carrier and Joe Grand (2004)
http://www.digital-evidence.org/papers/tribble-preprint.pdf

Copilot by Komoku (2004)
http://www.usenix.org/events/sec04/tech/full_papers/petroni/petroni.pdf

- PCI add-in card (requires installation before the incident)
- Not available to the public

Using FireWire to dump the memory

OHCI controller can read and write the first 4 GiB of main memory

- frequently found on laptops
- rarely installed on desktop computers

Adam Boileau
http://www.storm.net.nz/projects/16

Anti-forensic techniques - DMA

Redirecting physical memory access by J. Rutkowska (2007)
http://invisiblethings.org/papers/cheating-hardware-memory-acquisition-updated.ppt

- Manipulates configuration of Northbridge
- At the same physical address CPU and DMA see different

Cold booting the system (1)

Based on research from Princeton University
J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten
http://citp.princeton.edu/memory

- Showed that memory could retain their contents for seconds to minutes after power is lost.
- Cut the power and boot up the system with a very low memory-impact OS that dumps the memory.

Cold booting the system (2)

Ideal solution when you:

- do not trust the operating system you are investigating
- have the possibility to shutdown the system

Freezing the memory circuits

Same researchers from Princeton showed that by freezing the memory modules and transporting them to a secure location, data will survive up to 10 minutes without power.
http://citp.princeton.edu/memory

Freezing the memory circuits (2)

Ideal solution when you:

- do not trust the hardware of the system you are investigating
- have the possibility to shutdown the system

Analyzing the raw memory dump

Different methods to enumerate information

1. Look for a printable string
2. Reconstruct internal data structures
3. Search for static signatures of kernel data structures

Using strings

Sysinternals strings - defaults to Unicode and ASCII, minimum length 3 characters http://www.microsoft.com/technet/sysinternals/utilities/strings.mspx

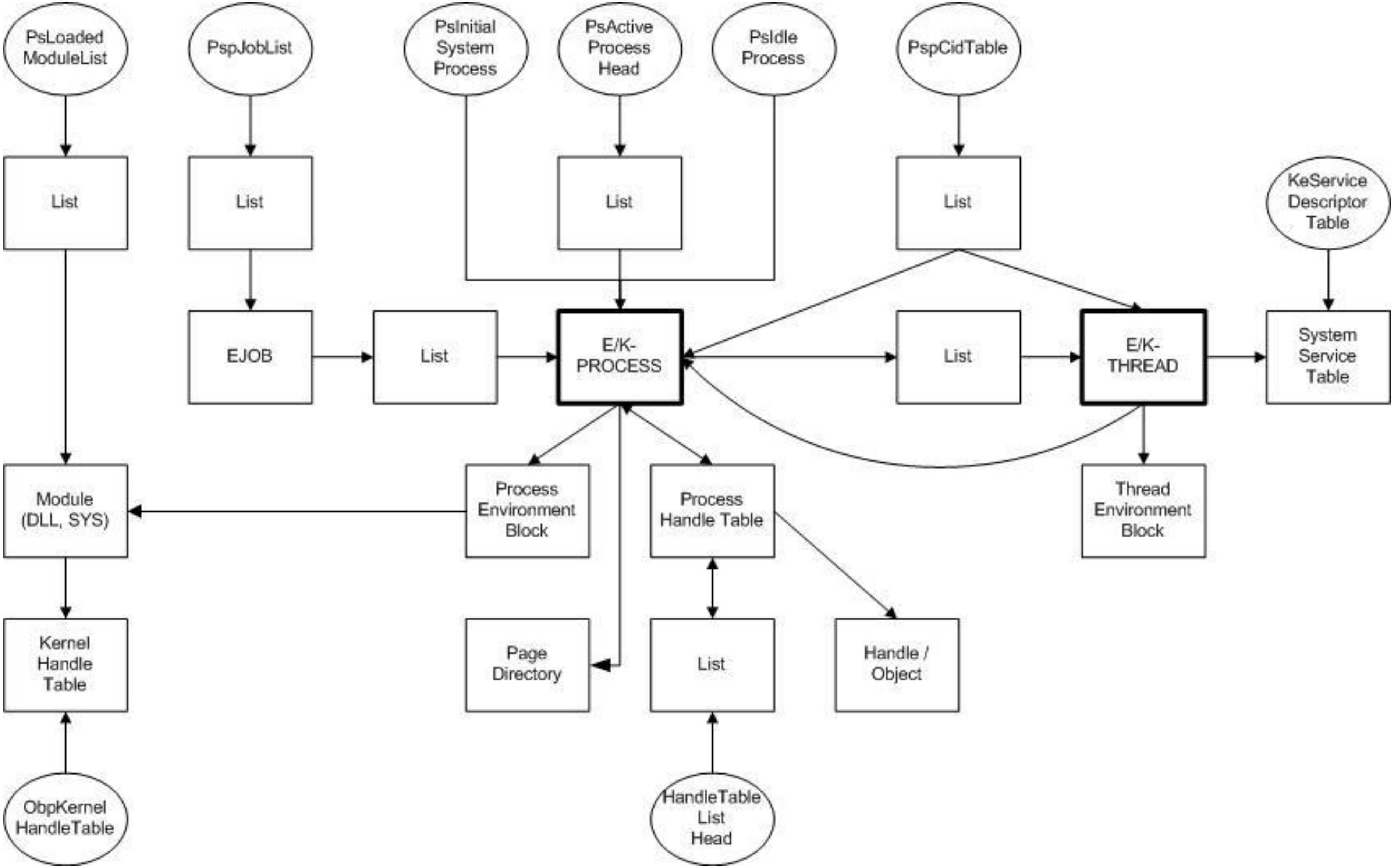
- No context, difficult to interpret
- A lot of interesting information is not in a printable format:
  - Timestamps (FILETIME, uint32)
  - IP addresses

Reconstruct internal data structures

Most data is kept in Lists and Trees.

- From a known starting point reconstruct and follow the list/tree and enumerate the objects found (aka “list-walking”).
- The most important structure is: _LIST_ENTRY, a double-linked list element.

```
kd> dt _LIST_ENTRY
   +0x000 Flink            : Ptr32 _LIST_ENTRY
   +0x004 Blink            : Ptr32 _LIST_ENTRY
```



PsLoaded ModuleList
PspJobList
PsInitial System Process
PsActive Process Head
PsIdle Process
PspCidTable
List
List
List
List
KeService Descriptor Table
EJOB
List
E/K-PROCESS
List
E/K-THREAD
System Service Table
Module (DLL, SYS)
Process Environment Block
Process Handle Table
Thread Environment Block
Kernel Handle Table
Page Directory
List
Handle / Object
ObpKernel HandleTable
HandleTable List Head

Enumerating internal data structures

Pmondump Joe Stewart
http://www.secureworks.com/research/tools/pmodump.pl.gz

lspi - LiSt Process Image by Harlan Carvey
http://windowsir.blogspot.com

Windows Memory Forensic Toolkit (WMFT) by Mariusz Burdach
http://forensic.seccure.net

Search for static signatures of kernel data structures

Simple, brute-force searching

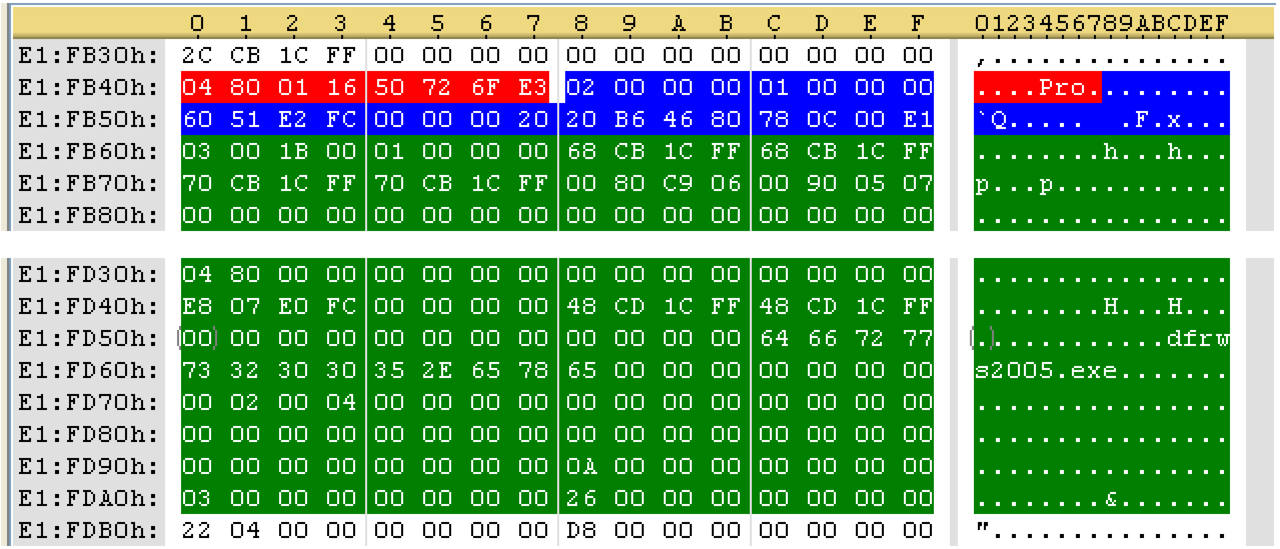
- Largely independent from the dump file format
- Fast, low memory requirements

Problems:

- Assuring a sufficient selectivity
- Signature should be based on essential data, otherwise it can be easily defeated

Memory pool allocations

- Memory management – POOL_HEADER
- Object management – OBJECT_HEADER
- Object – EPROCESS in this example



```
            0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
E1:FB30h:  2C CB 1C FF 00 00 00 00 00 00 00 00 00 00 00 00   ,...............
E1:FB40h:  04 80 01 16 50 72 6F E3 02 00 00 00 01 00 00 00   ....Pro.........
E1:FB50h:  60 51 E2 FC 00 00 00 20 20 B6 46 80 78 0C 00 E1   `Q.....  .F.x...
E1:FB60h:  03 00 1B 00 01 00 00 00 68 CB 1C FF 68 CB 1C FF   ........h...h...
E1:FB70h:  70 CB 1C FF 70 CB 1C FF 00 80 C9 06 00 90 05 07   p...p...........
E1:FB80h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................

E1:FD30h:  04 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
E1:FD40h:  E8 07 E0 FC 00 00 00 00 48 CD 1C FF 48 CD 1C FF   ........H...H...
E1:FD50h:  00 00 00 00 00 00 00 00 00 00 00 00 64 66 72 77   ............dfrw
E1:FD60h:  73 32 30 30 35 2E 65 78 65 00 00 00 00 00 00 00   s2005.exe.......
E1:FD70h:  00 02 00 04 00 00 00 00 00 00 00 00 00 00 00 00   ................
E1:FD80h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
E1:FD90h:  00 00 00 00 00 00 00 00 0A 00 00 00 00 00 00 00   ................
E1:FDA0h:  03 00 00 00 00 00 00 00 26 00 00 00 00 00 00 00   ........&.......
E1:FDB0h:  22 04 00 00 00 00 00 00 D8 00 00 00 00 00 00 00   "...............
```

Enumerating static kernel data structures

PTFinder and PoolTools by Andreas Schuster
http://computer.forensikblog.de

- Enumerates pool allocations in the memory dump
  - Even exited ones!

Volatity by Aaron Walters and Nick L. Petroni
https://www.volatilesystems.com/default/volatility

- Dumps running processes, threads, loaded modules and much, much more

Windows Debugger

Multipurpose debugger from Microsoft that can be used to debug user mode applications, drivers, and the operating system itself in kernel mode

- Operates on a live system and on crash dumps
- Public symbol server from Microsoft that has most of the public symbols for Windows 2000 and later versions
- Uses extensions to execute custom commands from within the debugger

Converting a raw memory dump to a crash dump

Volatility 1.3
https://www.volatilesystems.com/default/volatility

- Currently supports conversions between different memory formats on Windows XP SP2 and SP3

KntDD
http://www.gmgsystemsinc.com/knttools/

- Saves system state so that a memory dump later can be converted into a crash dump

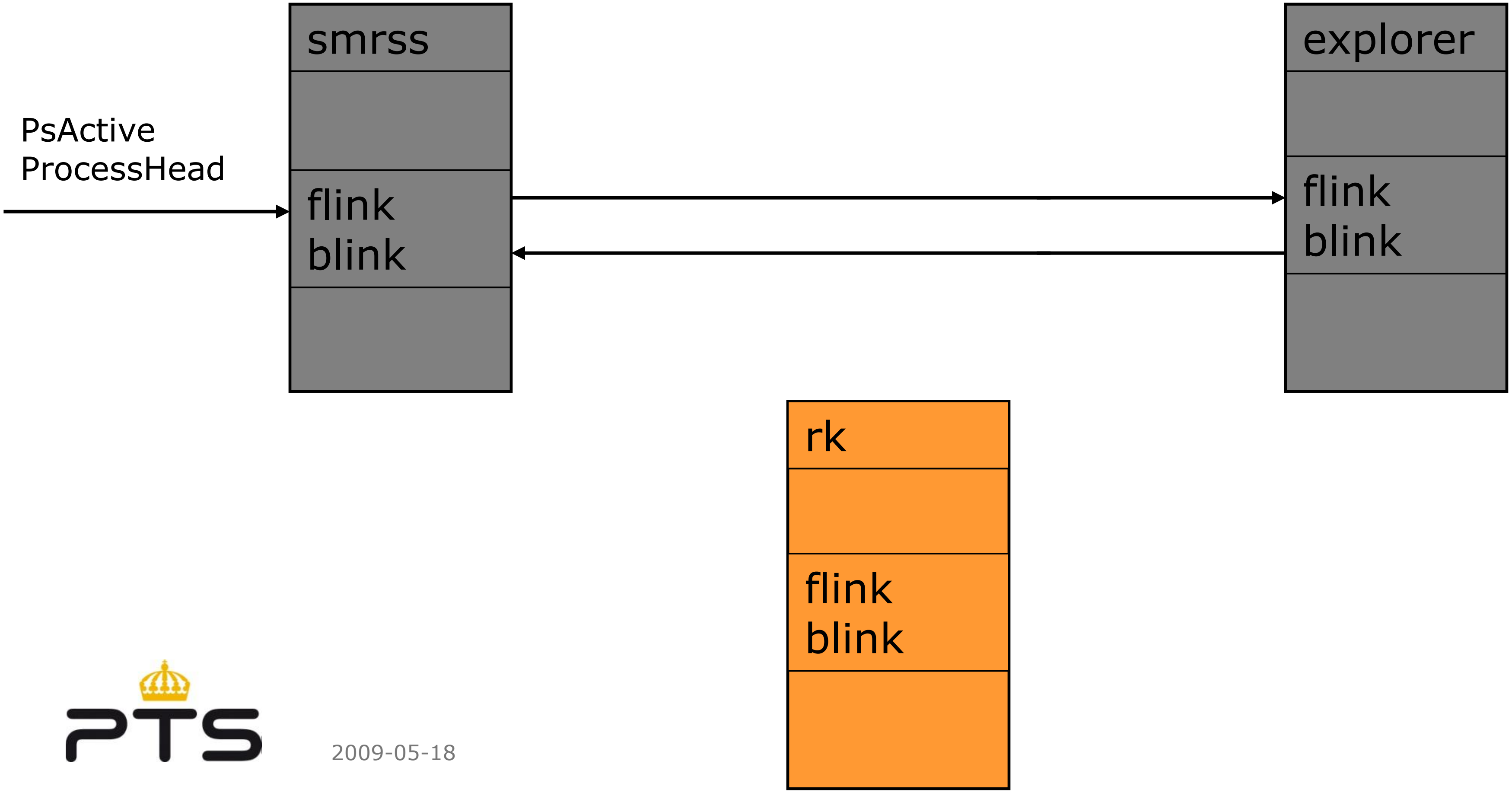
Different rootkit techniques and how we detect it

Three different types of rootkits we will discuss

- DKOM rootkits
- Injecting in a running processes
- Hooking

DKOM rootkits

Works by unlinking doubly linked lists in Windows



Detecting DKOM rootkits

List all loaded objects by enumerating memory pool allocations

- Processes
- Threads
- Drivers

Compare with list enumerated from following doubly linked lists

- Cross view detection

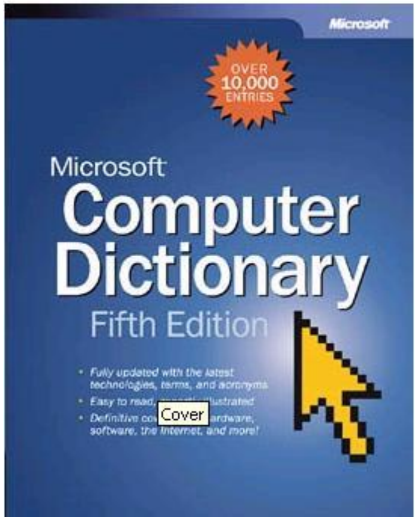
Injecting threads in a running processes

The threads in a processes are the ones that gets execution time. Not the process itself.

- leaching the process

Hooking

“**hook** n. A location in a routine or program in which the programmer can connect or insert other routines for the purpose of debugging or enhancing functionality”



- Hooking of a single program (API hooking)
- Hooking of system tables or exported functions
- Hooking unexported functions

Hooking exported functions

Some of the popular functions and tables to hook

- GDT (Global Descriptor Table)
- LDT (Local Descriptor Table)
- IDT (interrupt Descriptor Table)
- SSDT (System Service Dispatch Table)
- EAT (Export Address Table)
- IAT (Import Address Table)
- IRP (I/O Request Packet)

Detecting hooking

Highly dependent of the type of function that is being hooked

- `kd> dps win32k!W32pServiceTable`
- `kd> !drvobj Tcpip 0x3`

The `!chkimg` command compares the binary on disk with the one loaded into memory

  - Disk image of the loaded drivers must also be collected so the debugger have something to compare with

Hooking unexported functions

Works by changing code deep down in the kernel. Also referred to as “Stealth by design”.

- Deepdoor by Joanna Rutkowska
  http://www.invisiblethings.org
  - Patches deep down in the NDIS structure
  - Deepdoor idea implemented by the uay rootkit

Detecting hooking of unexported function

Detection is generally very difficult.

- Using specific debugger extension

Conclusions

Memory based forensic should be a part of your incident investigation process

We (the good guys) are always going to be one step behind the rootkit developers

- Virtualization-based rootkits
- Hardware/Firmware rootkits

Thank you for your attention!

Pär Österberg Medina
par.osterberg@sitic.se