Hibernation File Attack – Reino de España

Summer 2010

Peter Kleissner

About me

Peter Kleissner, Aloha from Vienna (Austria)

20 years old

- 2008-2009 Worked as developer at an anti-virus company
- 2010 Own startup company

- Black Hat USA 2009: Stoned Bootkit
- Hacking at Random & DeepSec 2009

Website with publications: http://web17.webbpro.de/

Hibernation File Attack – The Training

29 June 2010		
	9:00 – 14:00	Training: Hibernation File, Page File
	14:00 – 15:00	Lunch
	15:00 – 18:00	Training: Attacking the Hibernation File
30 June 2010		
	9:00 – 14:00	Training: Bootkit
	14:00 – 15:00	Lunch
	15:00 – 18:00	Training: Real life attacks via bootkit

Topics: Hibernation File, Page File, gathering information from them and attacking them, including bootkit possibilities

Hibernation

ACPI global power states:

G0	Working (S0)
G1	Sleeping (S1..S4)
G2	Software off (S5)
G3	Mechanical off

S3	Standby (XP), Sleep (Vista/7), (also known as suspend-to-ram)
S4	Hibernation (also known as suspend-to-disk)

S3 and S4 run under G1. S5 is used to differ from S4.

The S4 sleeping state is the lowest power, longest wake latency sleeping state supported by ACPI. In order to reduce power to a minimum, it is assumed that the hardware platform has powered off all devices. Platform context is maintained.

When hibernating, the system stores everything to the hibernation file and enters S4.

Hibernation File

C:\hiberfil.sys (note the missing “e”, to conform 8.3)

Stores entire physical RAM based in pages, is at least 50% size of memory pre-allocated.

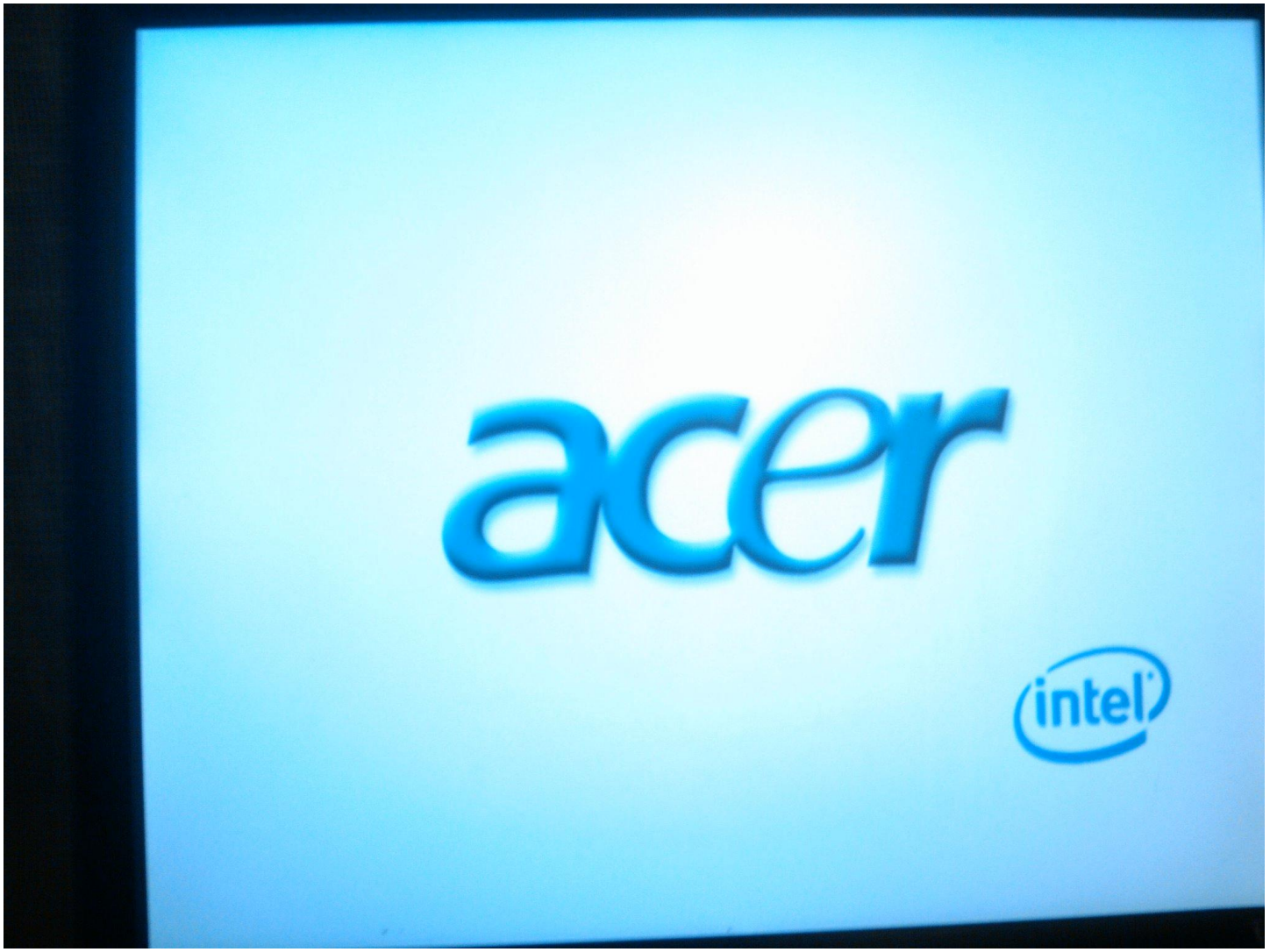
`powercfg -h on|off` to activate/deactivate

`Shutdown /h` to hibernate (or Start -> Hibernate)

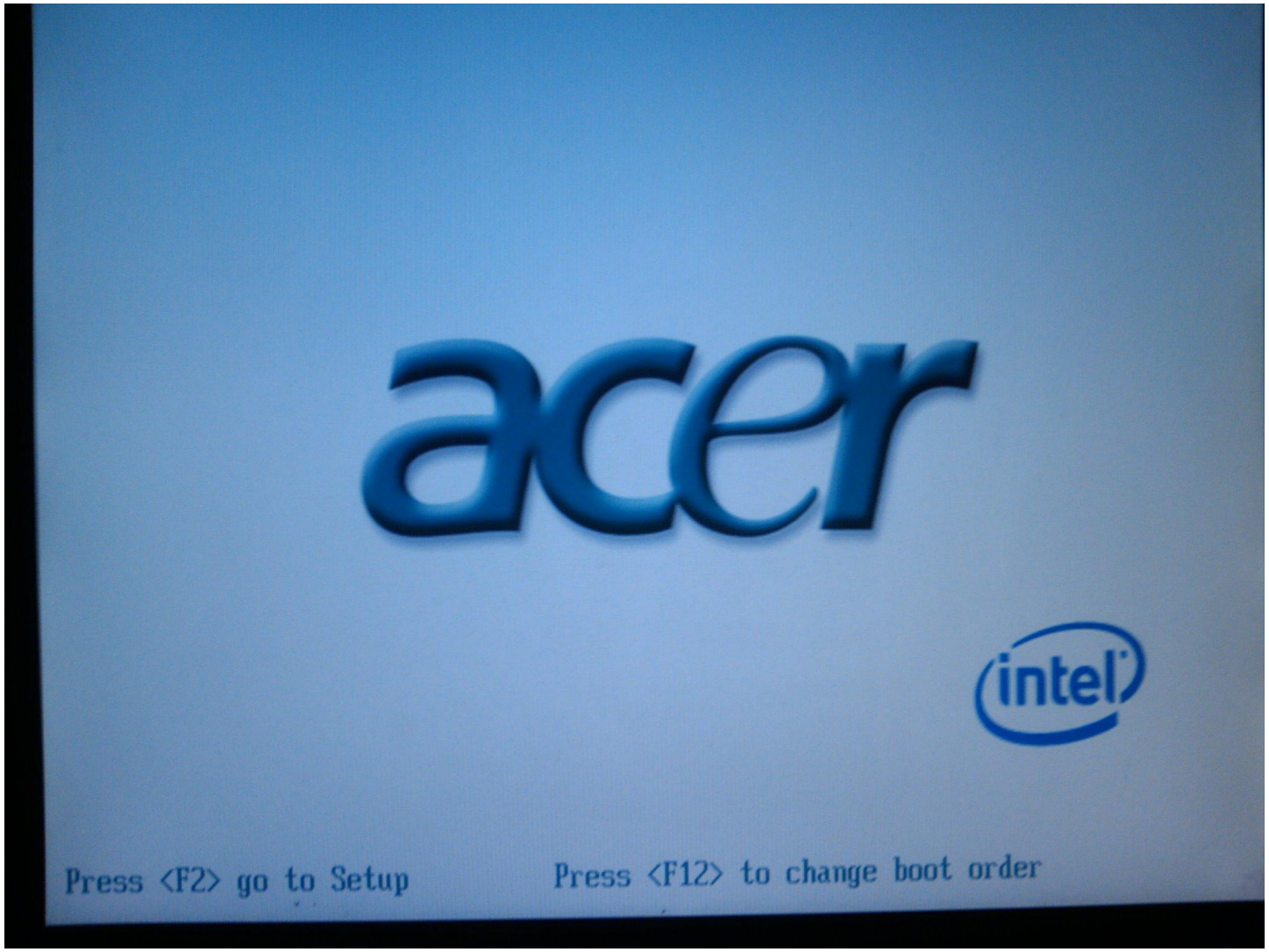
Important! Hibernation is an ACPI state! Thus our BIOS *wants* to boot original OS and does not provide the BIOS setup or menu.

Windows, however, checks only signature in hibernation file. The contents is by default not cleared.

Booting from S4 (hibernation)



Booting from G2 (soft off) or G3 (mechanical off)



Escaping from S4

Why?

- To analyze an original hibernated hiberfil.sys.

How?

1. Pressing the power button for more than 3 seconds -> G3
2. Restarting system in BIOS through Ctrl + Alt + Entf -> G2

Windows does not care if started in S4, G2 or G3, it always checks the hibernation file signature.

The BIOS cares, it won't let start us from another device.

Windows Vista/7: New "Hybrid Sleep" state

Hybrid sleep saves the OS state into RAM, but it also writes it all to the hard drive as well (sort of like hibernate does). This ensures that even if power is lost, the data will remain. [2]

The computer uses the Hiberfil.sys file to store a copy of the system memory on the hard disk when the hybrid sleep setting is turned on.

$\Rightarrow$ We can analyze RAM at time of Hybrid Sleep

Encryption (1)

Bitlocker always supported it (Microsoft)

TrueCrypt [3], back in 2008:

* *Disclaimer: As Microsoft does not provide any API for handling hibernation, 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. 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.*

Encryption (2)

[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. We responded on March 31 providing details of the issues and suggested solutions.]

[Update 2009-05-10: Since April 2008, we have been working with Microsoft to explore possible ways to solve this issue. We have private access to a draft version of a document specifying the future API, which should allow us to solve the issue on Windows Vista and later versions of Windows. Note: We have been asked not to disclose the content of the document to any third parties, so please do not ask us to send you a copy of the document.]

Current status: TrueCrypt works with hibernation under Vista and 7.

Pagefile

When the system is hibernated, the pagefile is still valid!

⇒ We also have to care about the page file

C:\pagefile.sys

Contains raw pages (4 KB and 4 MB parts)

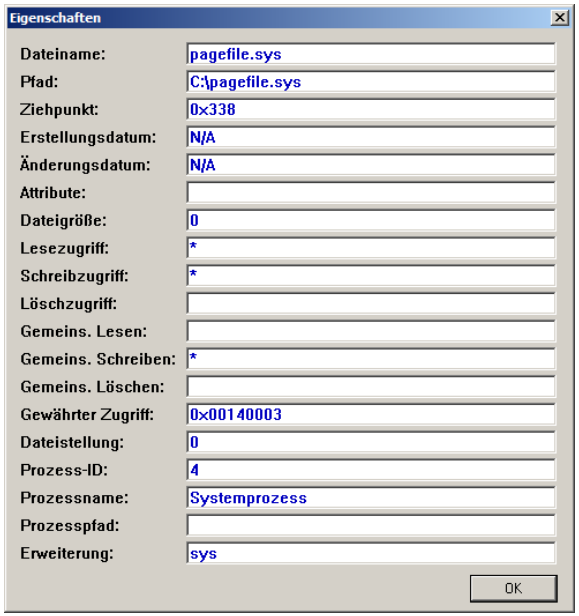
Security of pagefile and hibernation file

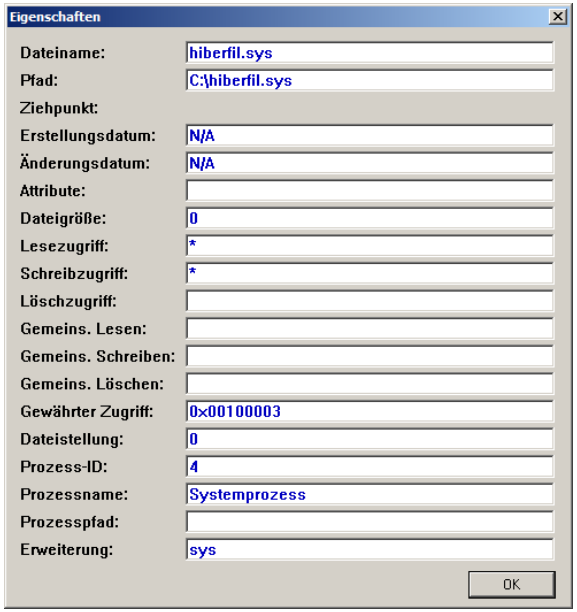
On running system protected with a kernel handle from System (PID 4), that prevents reading the page and hibernation file. However, you can read (XP, Vista, 7) it directly from partition with raw sector access, and write (XP) it.

There is a policy to clear both files on shut down:

`Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options`

You can simply read or modify these files when the OS is not active.



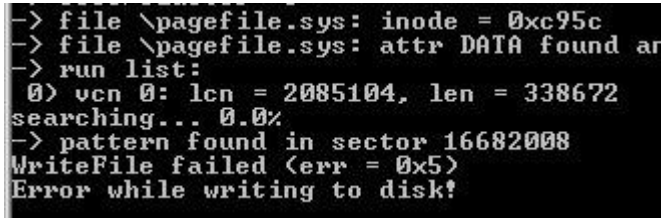


Live demo!

Pagefile Attack

Black Hat USA 2006 “Subverting Vista Kernel”
By Joanna Rutkowska and Alexander Tereshkin

I met Alex in a taxi cab in Las Vegas!



In Vista RC1 they overwrote pages in the pagefile through raw sector access.
CreateFile(\\.\C:) - handle to partition
CreateFile(\\.\PHYSICALDRIVE0)) - handle to entire disk

With Vista RC2 Microsoft prevents all write access to mounted partitions.

Open Source [5]

Live demo!

API Support (1)

Kernel32.dll

```
BOOL WINAPI SetSystemPowerState(
  __in BOOL fSuspend,
  __in BOOL fForce
);
```

If fSuspend = false, then the system hibernates. Requires SE_SHUTDOWN_NAME privileges. Marked as deprecated, SetSuspendState should be used instead.

PowrProf.dll

```
BOOLEAN WINAPI SetSuspendState(
  __in  BOOLEAN Hibernate,
  __in  BOOLEAN ForceCritical,
  __in  BOOLEAN DisableWakeEvent
);

BOOLEAN WINAPI IsPwrHibernateAllowed(void);
```

API Support (2)

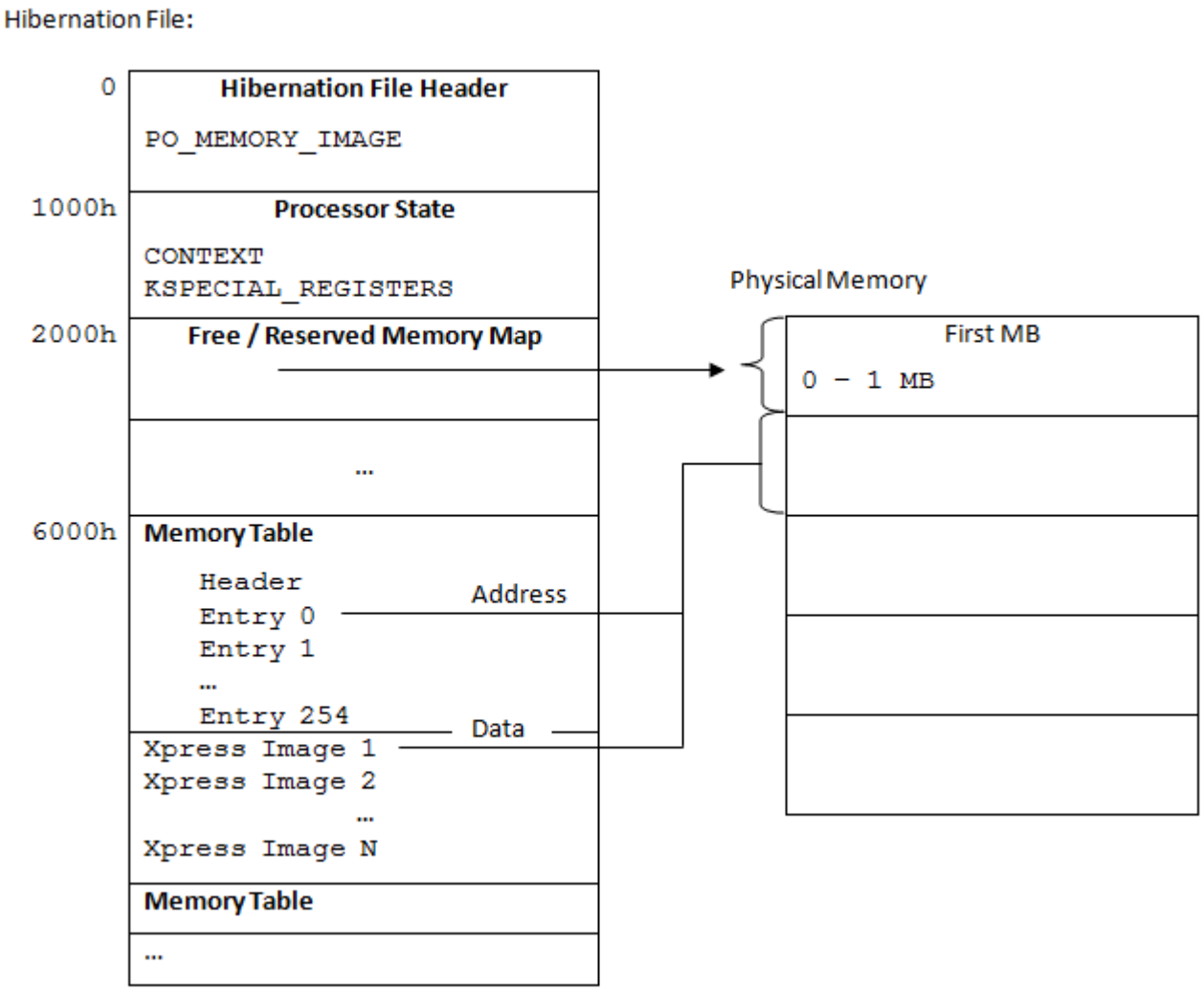
```
BOOLEAN WINAPI GetPwrCapabilities(
  __out  PSYSTEM_POWER_CAPABILITIES lpSystemPowerCapabilities
);
```

Should be used instead of IsPwrHibernateAllowed.

```
NTSTATUS WINAPI CallNtPowerInformation(
  __in   POWER_INFORMATION_LEVEL InformationLevel,
  __in   PVOID lpInputBuffer,
  __in   ULONG nInputBufferSize,
  __out  PVOID lpOutputBuffer,
  __in   ULONG nOutputBufferSize
);
```

With InformationLevel = SystemReserveHiberFile the hibernation file size can be modified.

Hibernation File Format (1)



Hibernation File Format (2)

Format for Windows Vista:

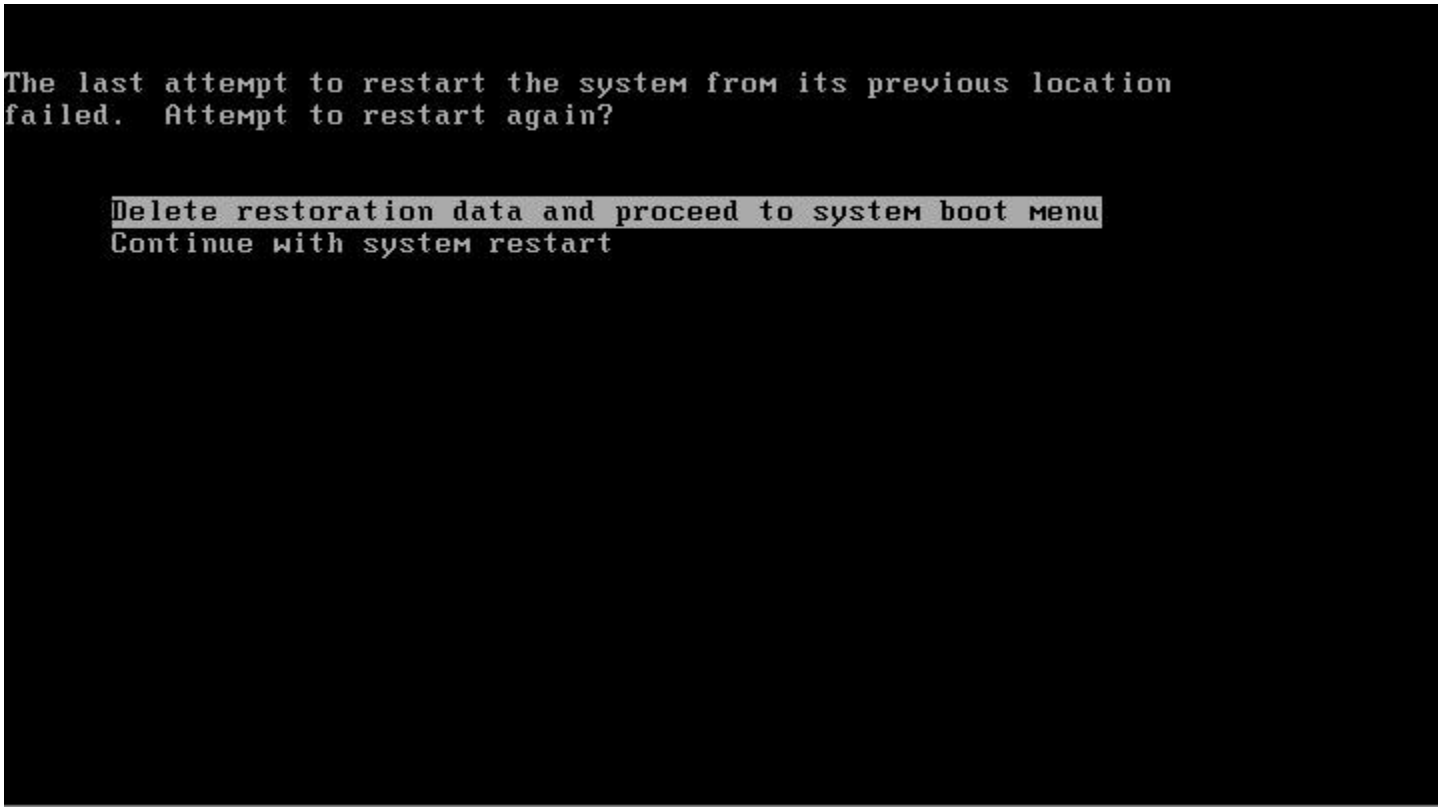
0000h	Hibernation File Header
1000h	Processor State (in Windows XP reversed)
2000h	Reserved Memory Map (in Windows XP reversed)
6000h	Memory Table 1
	Xpress Image 1
	Xpress Image 2
	…
	Memory Table 2
	Xpress Image 1
	Xpress Image 2
	…
	Memory Table 3
	…

The file is organized as pages (4 KB). The header contains in the first 4 bytes a signature:

HIBR	Active hibernation file, system shall process resume from hibernation
WAKE	Inactive hibernation file, system shall ask user
0000h	Successful restoration (entire first page is cleared)

Signatures are uppercase in Vista/7 and lowercase in XP.

Wake signature



Live demo!

Hibernation File Header (Vista)

```
typedef struct
{
      ULONG Signature;
      ULONG ImageType;
      ULONG CheckSum;
      ULONG LengthSelf;
      ULONG PageSelf;
      ULONG PageSize;
      LARGE_INTEGER SystemTime;
      UINT64 InterruptTime;
      ULONG FeatureFlags;
      UCHAR HiberFlags;
      UCHAR spare[3];
      ULONG NoHiberPtes;
      ULONG HiberVa;
      LARGE_INTEGER HiberPte;
      ULONG NoFreePages;
      ULONG FreeMapCheck;
      ULONG WakeCheck;
      ULONG TotalPages;
      ULONG FirstTablePage;
      ULONG LastFilePage;
      PO_HIBER_PERF PerfInfo;              // with Windows XP
      ULONG NoBootLoaderLogPages;          // with Windows Vista
      ULONG BootLoaderLogPages[8];
      ULONG TotalPhysicalMemoryCount;
} PO_MEMORY_IMAGE, *PPO_MEMORY_IMAGE;
```

Memory Tables (XP, Vista)

```
struct MEMORY_TABLE
{
      DWORD PointerSystemTable;
      UINT32 NextTablePage;
      DWORD CheckSum;
      UINT32 EntryCount;
      MEMORY_TABLE_ENTRY MemoryTableEntries[EntryCount];
};
```

Each table (except the last) contains 255 entries. The checksum is unused. Each entry describes a physical range of pages:

```
struct MEMORY_TABLE_ENTRY
{
      UINT32 PageCompressedData;
      UINT32 PhysicalStartPage;
      UINT32 PhysicalEndPage;
      DWORD CheckSum;
};
```

These two structures are not exported as symbols and are different in 7.

Xpress Images (XP, Vista)

Physical data is stored compressed in Xpress Images. The LZ77 + DIRECT2 algorithm is used (described in MS-OXCRPC).

Xpress Images are following each other directly (they are not page aligned). Physical pages are stored continuously in xpress images.

```
struct IMAGE_XPRESS_HEADER
{
     CHAR Signature[8] = 81h, 81h, "xpress";
     BYTE UncompressedPages = 15;
     UINT32 CompressedSize;
     BYTE Reserved[19] = 0;
};
```

Size of compressed data = CompressedSize / 4 + 1, rounded up to 8

Size of uncompressed data = (UncompressedPage + 1) * 1000h

Live demo!

Hibernate Once/Resume Many

Fun with the hibernation file! In Windows XP Embedded a feature.

We can hibernate once – and if we restore the header, we can resume as many times as we want!

Live demo!

Our target: Injecting unsigned code into kernel

We want to parse and replace code into the page and hibernation file. We achieve this by patching the Null Device Driver null.sys.

The Null Device Driver component provides the functional equivalent of \dev\null in the Unix environment by accepting I/O request packets and returning them to the caller. [6]

Patching the IRP dispatcher and appending our code

Live demo!

Hibernation File Header Differences (1)

	XP	Vista	7
00h	Signature	Signature	Signature
04h		ImageType	ImageType
04h	Version		
08h	CheckSum	CheckSum	Checksum
0Ch	LengthSelf = 168	LengthSelf = 240	LengthSelf =224
10h	PageSelf	PageSelf	PageSelf
14h	PageSize	PageSize	PageSize
	ImageType		
	SystemTime	SystemTime	SystemTime
	InterruptTime	InterruptTime	InterruptTime
	FeatureFlags	FeatureFlags	FeatureFlags
	HiberFlags	HiberFlags	HiberFlags
	Spare	Spare	Spare
	NoHiberPtes	NoHiberPtes	NoHiberPtes

Hibernation File Header Differences (2)

	XP	Vista	7
	HiberVa	HiberVa	HiberVa
	HiberPte	HiberPte	HiberPte
	NoFreePages	NoFreePages	NoFreePages
	FreeMapCheck	FreeMapCheck	FreeMapCheck
	WakeCheck	WakeCheck	WakeCheck
	TotalPages	TotalPages	
	FirstTablePage	FirstTablePage	FirstTablePage
	LastFilePage	LastFilePage	
	PerfInfo	PerfInfo	PerfInfo
			FirmwareRuntimeInformationPages
			FirmwareRuntimeInformation

Hibernation File Header Differences (3)

	XP	Vista	7
		NoBootloaderLogPages	
		BootLoaderLogPages	
		TotalPhysicalMemoryCount	
			Not Used
			ResumeContextCheck
			ResumeContextPages

Other OS differences

Windows 2000

Different compression algorithm: LZNT1, internal function XpressEncode() according to Matthieu Suiche [7]

Uses checksums? With XP the checksums (except in the header) are all set to zero.

Windows 7

Different structure for Memory Tables.

Use the LengthSelf field in the header to determine the operating system: 168 = XP, 240 = Vista, 224 = 7

Modifying the header

The checksum algorithm is according to Matthieu [7] calculated by the tcpxsum() function (original within tcpip.sys)

More information TBA; creating a new header to boot into previous hibernation state

References

[1] Frontpage picture by Ela2007
http://www.flickr.com/photos/64479867@N00/523730762

[2] Hybrid Sleep State
http://www.howtogeek.com/howto/windows-vista/disable-hybrid-sleep-mode/

[3] TrueCrypt about the Hibernation File
http://www.truecrypt.org/docs/?s=hibernation-file

[4] Clear virtual memory pagefile when system shuts down
http://msdn.microsoft.com/en-us/library/ms814147.aspx

[5] Pagefile Attack (french)
http://www.ivanlef0u.tuxfamily.org/?p=77

[6] Null Device Driver
http://msdn.microsoft.com/en-us/library/aa939249(WinEmbedded.5).aspx

[7] Matthieu Suiche, Windows hibernation file for fun ‘n’ profit
http://msuiche.net/con/bhusa2008/Windows_hibernation_file_for_fun_'n'_profit-0.6.pdf