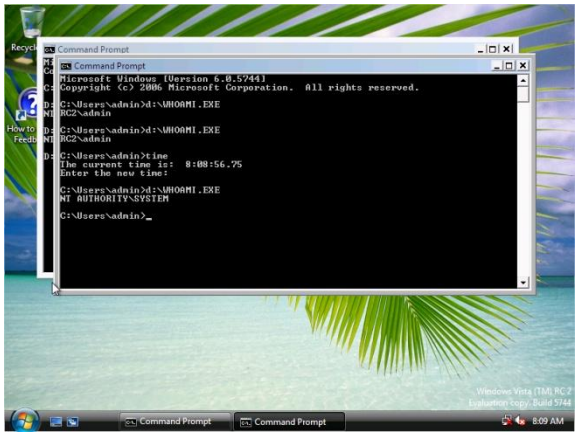
The Rise of MBR Rootkits And Bootkits in the Wild



Vbootkit

Mebroot

Stoned Bootkit

Black Hat déjà vu - Stoned again

Peter Kleissner

Agenda

- History
- Windows Product Activation
- Development, Installation & Usage
- Stoned Bootkit
- Future

Who the hack am I?

- Independent Operating System Developer
- Hometown Vienna (Austria)
- Startup “Insecurity Systems” (InSec)

About Bootkits

A Bootkit is a Rootkit in the Master Boot Record

Introduced by Vipin and Nitin Kumar

"A bootkit is a rootkit that is able to load from a master boot record and persist in memory all the way through the transition to protected mode and the startup of the OS. It's a very interesting type of rootkit."

Robert Hensing about bootkits

Firmware	Malware	Operating System

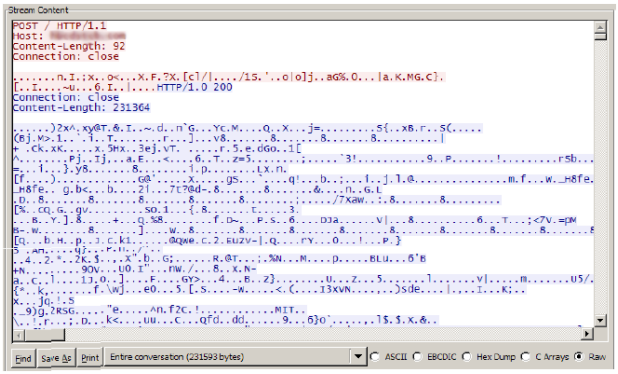
Timeline

	…	2006	2008	2010
		Mebroot	Vista Loader	Stoned Bootkit
	…	BOOT KIT		Tophet
		TPMkit		Kon-Boot
Stoned	BootRoot	Vbootkit		Vbootkit 2.0
1987	**2005**	**2007**		**2009**

BootRoot	Windows XP	Black Hat USA 2005
Vbootkit	Windows Vista	Black Hat Europe 2007
Tophet		XCon 2008
Vbootkit 2.0	Windows 7 (x64)	Hack In The Box Dubai 2009
Stoned Bootkit	All Windows Systems	Black Hat USA 2009

Typical Usage

Stoned	Keeping the user happy with text and sound messages :)



Mebroot	Stealing your banking data
Vista Loader	Spoofing OEM BIOS for Windows Product Activation
Kon-Boot	Bypassing Windows Logon
Stoned Bootkit	For forensics and law enforcement agencies
Vbootkit 1+2	Proof of concept

Windows Product Activation

OEM BIOS + **OEM Certificate** + **OEM Key** = **No Internet Activation!**

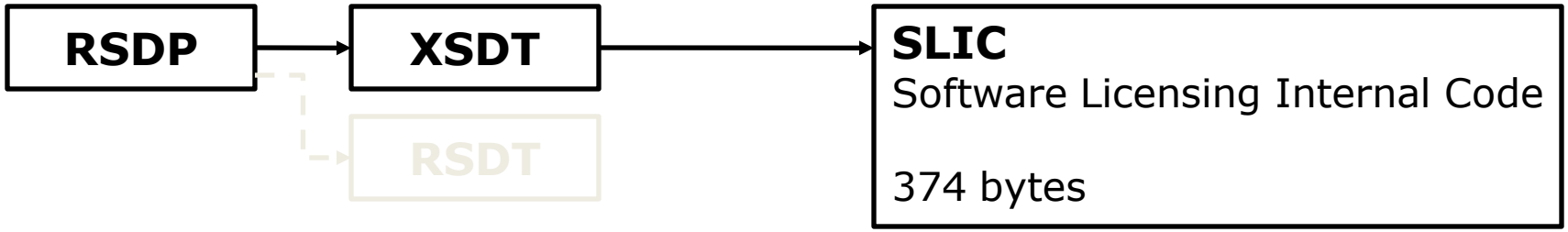
For Windows Vista and 7

Microsoft has a secret arrangement with OEM hardware manufacturers to include a secret additional ACPI table to identify the system as OEM

Acer, ASUS, Dell, Fujitsu Siemens, Gateway, HP, Lenovo, Medion, NEC, Sony, Sotec, Toshiba, MSI, Intel, and others

OEM BIOS – SLIC Table

The SLIC (Software Licensing Internal Code) table identifies the system as OEM.



These are some simple ACPI structures:

RSDP	Root System Description Pointer	**40h:0Eh**
XSDT	Extended System Description Table	**RSDP + 24**
SLIC	Software Licensing Internal Code	**found ☺**

The BIOS (= firmware) sets up these tables. So your bootkit can too!

SLIC Table

```
00000000  53 4C 49 43 76 01 00 00 01 47 44 45 4C 4C 20 20  SLICv....GDELL
00000010  4D 30 37 20 20 20 20 00 12 0C D6 27 41 53 4C 20  M07     ...Ö'ASL
00000020  61 00 00 00 00 00 00 00 9C 00 00 00 06 02 00 00  a.......œ.......
00000030  00 24 00 00 52 53 41 31 00 04 00 00 01 00 01 00  .$..RSA1........
00000040  7F F6 C1 05 BE 5C 57 63 A5 8A 68 F3 6E 8F 06 FA  .öÁ.¾\Wc¥Šhón..ú
00000050  AF B4 9F 68 82 23 EC 50 40 5A 73 7F EC E4 07 CB  ¯´Ÿh‚#ìP@Zs.ìä.Ë
00000060  DC 25 1A 9C E3 E3 66 11 E0 A5 98 06 C5 80 0A FA  Ü%.œããf.à¥˜.Å€.ú
00000070  42 93 86 98 E7 D5 1B D4 D7 3A A4 0B EE E2 7D BE  B"†˜çÕ.Ô×:¤.îâ}¾
00000080  5F 5B 15 0C AB D0 21 DE BF E9 B5 6E A4 57 B9 8C  _[..«Ð!Þ¿éµn¤W¹Œ
00000090  0C D2 BA 3A 69 30 76 94 71 A2 64 D7 4C D8 85 BF  .Ò°:i0v"q¢d×LØ…¿
000000A0  DF A5 6A C8 DC 45 D5 4D 8C B8 8C 05 2F FC 2E 23  ß¥jÈÜEÕMŒ¸Œ./ü.#
000000B0  C4 29 C5 6F 3F 29 6C 6D 57 79 0E B6 75 ED 21 95  Ä)Åo?)lmWy.¶uí!•
000000C0  01 00 00 00 B6 00 00 00 00 00 02 00 44 45 4C 4C  ....¶.......DELL
000000D0  20 20 4D 30 37 20 20 20 20 00 57 49 4E 44 4F 57    M07     .WINDOW
000000E0  53 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00  S ..............
000000F0  00 00 00 00 00 00 51 E9 A5 CD 35 30 91 B0 9B C0  ......Qé¥Í50'°›À
00000100  CE 05 FA 26 B5 43 29 40 1C 13 16 EF E3 BF 17 2F  Î.ú&µC)@...ïã¿./
00000110  BD 3B 99 B5 6E 23 49 F7 97 BC ED FF C9 4A 95 F4  ½;™µn#I÷—¼íÿÉJ•ô
00000120  A5 CD 33 0B 40 2E C8 E1 8B E6 8F B6 74 8E 94 43  ¥Í3.@.Èá‹æ.¶tŽ"C
00000130  E0 2F B6 CE 53 F0 09 3D B4 18 0F 44 23 10 64 F3  à/¶ÎSð.=´..D#.dó
00000140  74 06 2E 1D 00 71 13 6A C7 C9 9E 82 CB 71 09 B1  t....q.jÇÉž‚Ëq.±
00000150  9E 42 5A 7D F3 F8 CC D1 FD 22 90 BF 37 3E 2C 68  žBZ}óøÌÑý".¿7>,h
00000160  BB 30 FF 84 0F B5 2B B3 C0 7A 71 44 C5 EB 13 15  »0ÿ„.µ+³ÀzqDÅë..
00000170  C3 CA 66 1B 80 2E                               ÃÊf.€.
```

RSA Key 1024 Bit

OEM identifier

Certificate


```
<?xml version="1.0" encoding="utf-8"?><r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"
    licenseId="{e56c50ff-e9fe-461b-a5f2-1573cf933dbf}" xmlns:sx="urn:mpeg:mpeg21:2003:01-REL-
    SX-NS" xmlns:mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
    xmlns:sl="http://www.microsoft.com/DRM/XrML2/SL/v2"
    xmlns:tm="http://www.microsoft.com/DRM/XrML2/TM/v2"><r:title>OEM
    Certificate</r:title><r:grant><sl:binding><sl:data
    Algorithm="msft:rm/algorithm/bios/4.0">kgAAAAAAgBERUxMICABAAEAf/bBBb5cV2Olimjzbo8G+q+0n2iC
    I+xQQFpzf+zkB8vcJRqc4+NmEeClmAbFgAr6QpOGmOfVG9TXOqQL7uJ9vl9bFQyr0CHev+m1bqRXuYwM0ro6aTB2lHG
    iZNdM2IW/36VqyNxF1U2MuIwFL/wuI8QpxW8/KWxtV3kOtnXtIZU=</sl:data></sl:binding><r:possessPrope
    rty/><sx:propertyUri definition="trustedOem"/></r:grant><r:issuer><Signature
    xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMethod
    Algorithm="http://www.microsoft.com/xrml/lwc14n"/><SignatureMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/><Reference><Transforms><Transform
    Algorithm="urn:mpeg:mpeg21:2003:01-REL-R-NS:licenseTransform"/><Transform
    Algorithm="http://www.microsoft.com/xrml/lwc14n"/></Transforms><DigestMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><DigestValue>mylUeSOamDoBwptofZ7FKoCePH
    k=</DigestValue></Reference></SignedInfo><SignatureValue>OQojHOugcB3VvUc7xRonmHv/DP136N/mKu
    l3wR7gXg9OgmlSlm2Gjm59QO9xt7LvWDjdNWUNwNudww9+Ay1wjly0fGXRcMBO1rObJgAbGMC7ejtxMETpNZ8Ukzn9n
    hsnBJAUtzvynXSFqJQvboe45dNN6FBh9uaEj4zPiUKlk2c3B9GwFZi0554cC/tgF7mA8Bb+Hsa7e2jMrRN5KIjxD5di
    RNZr7XRzH0RLm/S9+sKtl9SkVQ5b3bIZhfAqVJ4hsCFpvyaVKW/XYbc4wOxf6r377ONOQD3NJX4nqELg3S4GCUG7xyK
    HFL2/QVqygiGr+CRCxJfZxf2feucbSWOgMQ==</SignatureValue><KeyInfo><KeyValue><RSAKeyValue><Modu
    lus>sotZn+w9juKPf7bMO9rNFriB+10v/t9bo/XWG+rzoDbw/uF4INZ5rGRIitiITY/bI4rANkv4Z5hG/8VxGMbqvqc
    aXJqnRFda7XAjgm1z9wkgX1R/d2tXLUUUQP0J1XuSbgzR89T/lpnc5q2Cdvy7Gv2pZvAzSeLOponXc8J3zOFr0IUXBG
    prXKnemVk1iJBFnyQGlWG3UoSpdlF0ichBQwPx/PgoTbcZsA7Gg62BGwPx/uDA3ZgwowrPlRwfLVAO6qE9xPJqRZdRF
    fPHbdQjp1YAq27wc6cTz5sPSTB1pJ4L9MD+NpvHj2OMZV5+LJ+bxZbTqhPcrzCp7ckkyD7Hzw==</Modulus><Expon
    ent>AQAB</Exponent></RSAKeyValue></KeyValue></KeyInfo></Signature><r:details><r:timeOfIssue
    >2006-03-16T20:17:30Z</r:timeOfIssue></r:details></r:issuer><r:otherInfo
    xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS"><tm:infoTables
    xmlns:tm="http://www.microsoft.com/DRM/XrML2/TM/v2"><tm:infoList tag="#global"><tm:infoStr
    name="applicationId">{55c92734-d682-4d71-983e-d6ec3f16059f}</tm:infoStr><tm:infoStr
    name="licenseCategory">msft:sl/PPD</tm:infoStr><tm:infoStr
    name="licenseType">msft:sl/OEMCERT</tm:infoStr><tm:infoStr
    name="licenseVersion">2.0</tm:infoStr><tm:infoStr
    name="licensorUrl">http://licensing.microsoft.com</tm:infoStr></tm:infoList></tm:infoTables
    ></r:otherInfo></r:license>
```

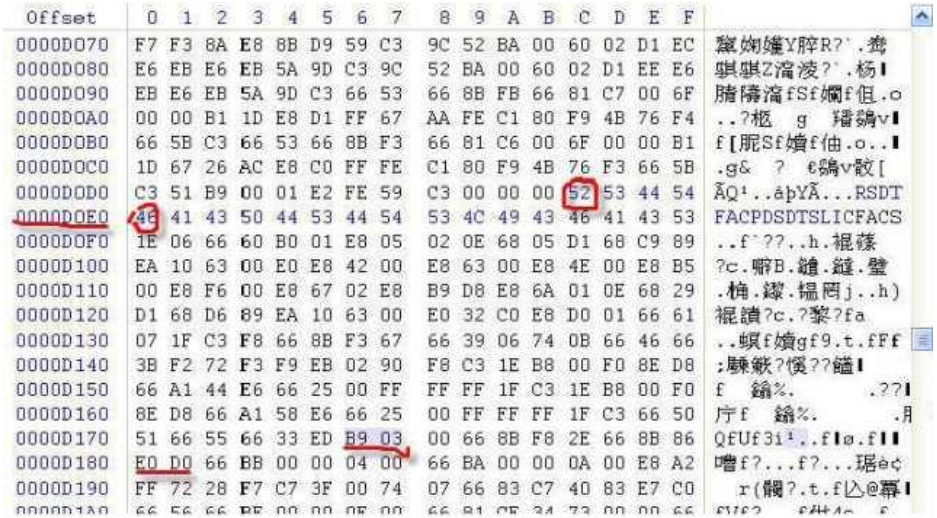

Install it: cscript %windir%\system32\slmgr.vbs -ilc Dell.xrm-ms

SLP OEM Key

Install System-Locked Preinstallation master product key:

```
slmgr -ipk 223PV-8KCX6-F9KJX-3W2R7-BB2FH
```

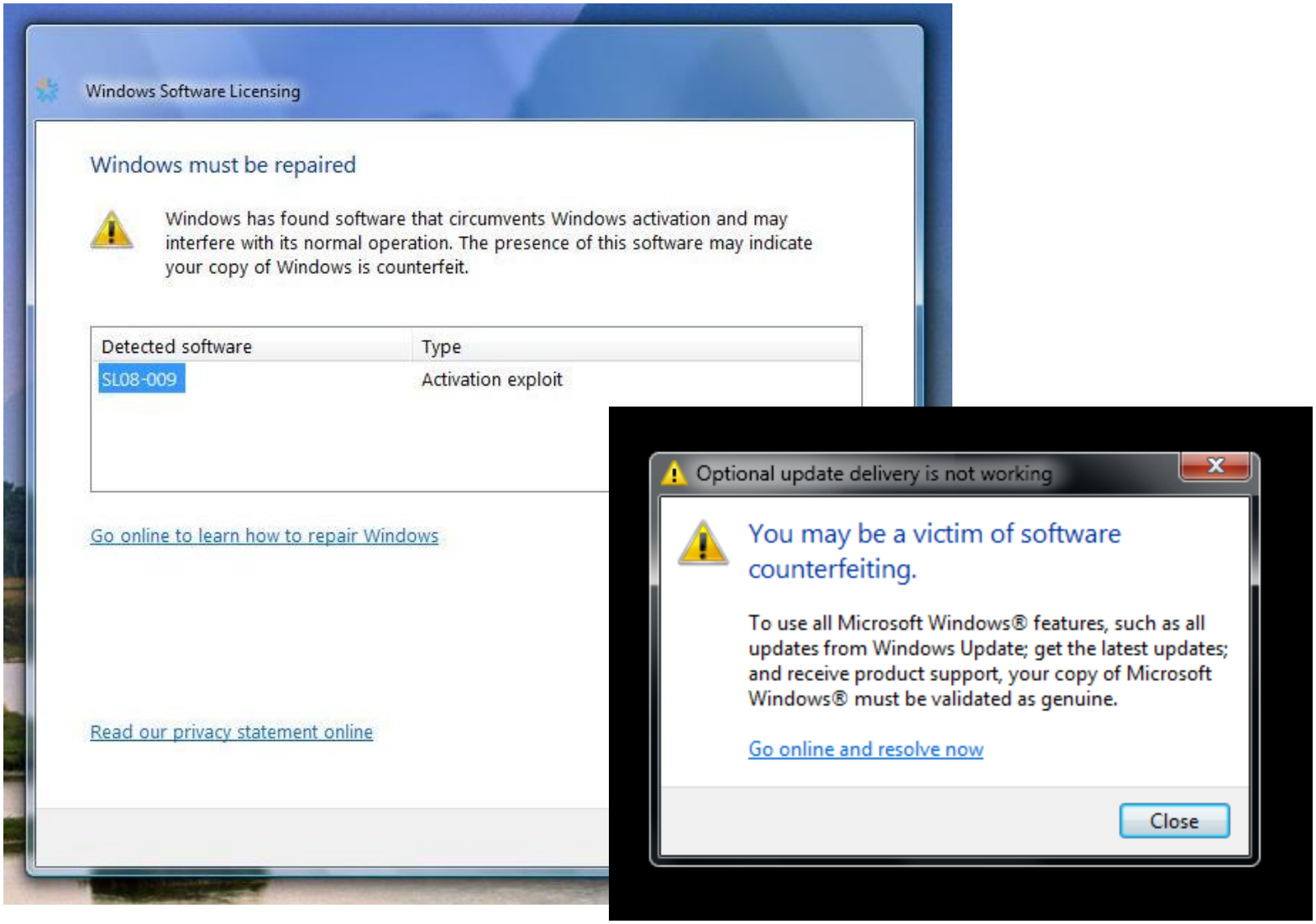
The dynamic injection vs. the persistent way



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000D070	F7	F3	8A	E8	8B	D9	59	C3	9C	52	BA	00	60	02	D1	EC
0000D080	E6	EB	E6	EB	5A	9D	C3	9C	52	BA	00	60	02	D1	EE	E6
0000D090	EB	E6	EB	5A	9D	C3	66	53	66	8B	FB	66	81	C7	00	6F
0000D0A0	00	00	B1	1D	E8	D1	FF	67	AA	FE	C1	80	F9	4B	76	F4
0000D0B0	66	5B	C3	66	53	66	8B	F3	66	81	C6	00	6F	00	00	B1
0000D0C0	1D	67	26	AC	E8	C0	FF	FE	C1	80	F9	4B	76	F3	66	5B
0000D0D0	C3	51	B9	00	01	E2	FE	59	C3	00	00	00	52	53	44	54
0000D0E0	46	41	43	50	44	53	44	54	53	4C	49	43	46	41	43	53
0000D0F0	1E	06	66	60	B0	01	E8	05	02	0E	68	05	D1	68	C9	89
0000D100	EA	10	63	00	E0	E8	42	00	E8	63	00	E8	4E	00	E8	B5
0000D110	00	E8	F6	00	E8	67	02	E8	B9	D8	E8	6A	01	0E	68	29
0000D120	D1	68	D6	89	EA	10	63	00	E0	32	C0	E8	D0	01	66	61
0000D130	07	1F	C3	F8	66	8B	F3	67	66	39	06	74	0B	66	46	66
0000D140	3B	F2	72	F3	F9	EB	02	90	F8	C3	1E	B8	00	F0	8E	D8
0000D150	66	A1	44	E6	66	25	00	FF	FF	FF	1F	C3	1E	B8	00	F0
0000D160	8E	D8	66	A1	58	E6	66	25	00	FF	FF	FF	1F	C3	66	50
0000D170	51	66	55	66	33	ED	B9	03	00	66	8B	F8	2E	66	8B	86
0000D180	E0	D0	66	BB	00	00	04	00	66	BA	00	00	0A	00	E8	A2
0000D190	FF	72	28	F7	C7	3F	00	74	07	66	83	C7	40	83	E7	C0

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	52	53	44	54	2C	00	00	00	01	00	56	49	41	36	39	34
00000010	41	57	52	44	41	43	50	49	31	2E	30	42	41	57	52	44
00000020	00	Add 4		00	00	00	00	00	00	00	00	00	46	41	43	50
00000030	74	Bytes		01	00	56	49	41	36	39	34	41	57	52	44	
00000040	41	43	50	49	31	2E	30	42	41	57	52	44	00	00	00	00
00000050	00	00	00	00	00	0(	Position SLIC table						F	40	00	00
00000060	A1	A0	A4	00	00	4(	here by Inserting 4						0	40	00	00
00000070	00	00	00	00	00	0(	new bytes. Remember						0	40	00	00
00000080	00	00	00	00	04	0(	the location for later use.						A	00	84	03

Microsoft against activation exploits



Installation

1. **Physical Access**

   Live CD, writing it raw to the hard disk, …

2. **Administrator Rights (Infector in Windows)**

   Elevate the rights at runtime using `ShellExecute()` or via a manifest

   Use some exploit

Elevated Administrator Rights

Application Manifest (embedded into executable)

```
<requestedPrivileges>
    <requestedExecutionLevel level="asInvoker" uiAccess="true"/>
</requestedPrivileges>
```

ShellExecute() at runtime

```
HINSTANCE ShellExecute(
    HWND hwnd,
    LPCTSTR lpOperation = "runas",
    (…)
);
```

Create a small loader that tries ShellExecute() until the user clicks "Yes" on Consent UI

Environment

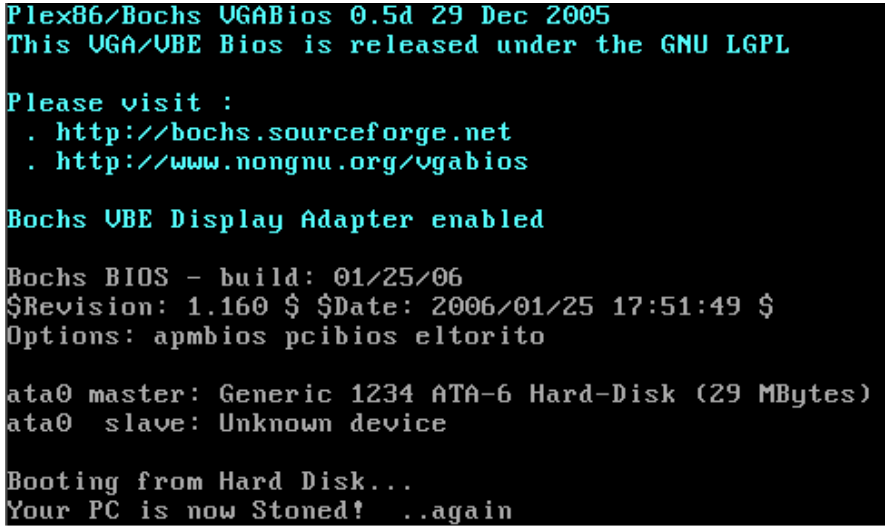
Real Mode (old school)

cs:ip = 0000h:7C00h

16 bit!

Directly loaded by the BIOS

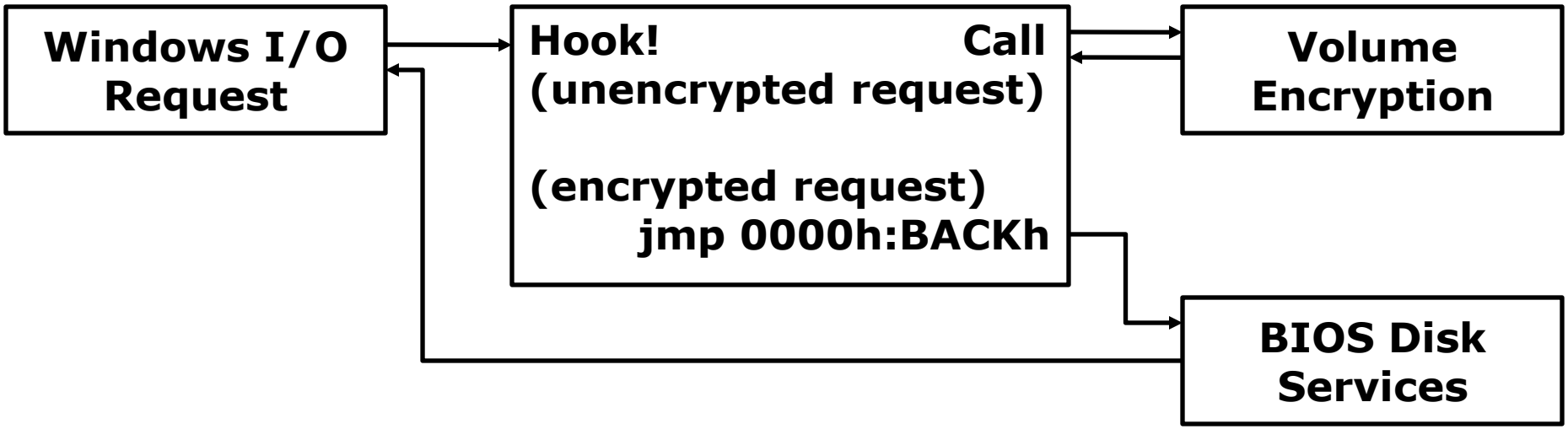
Must be programmed in assembly language low-level



The bootkit must be able to be memory persistent.

It is OS independent but attacks specific operating systems.

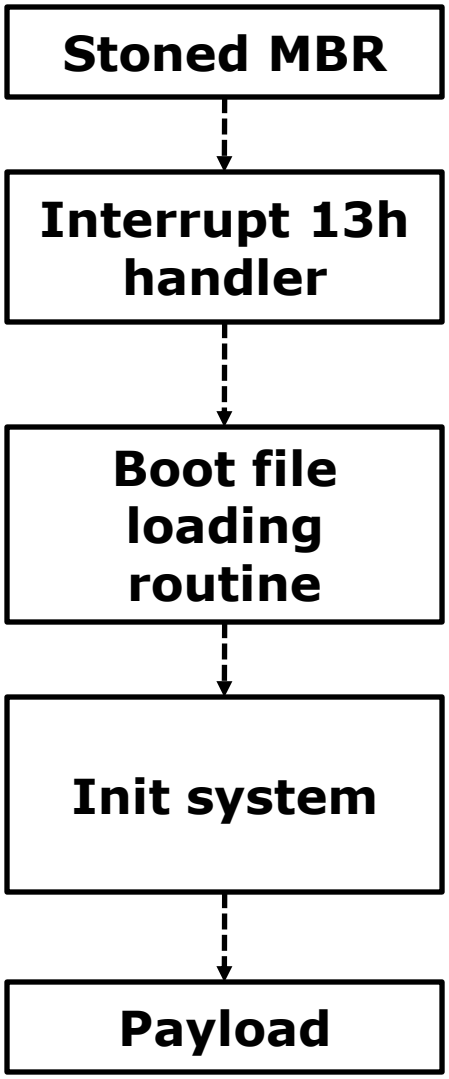
Bypassing Full Volume Encryption



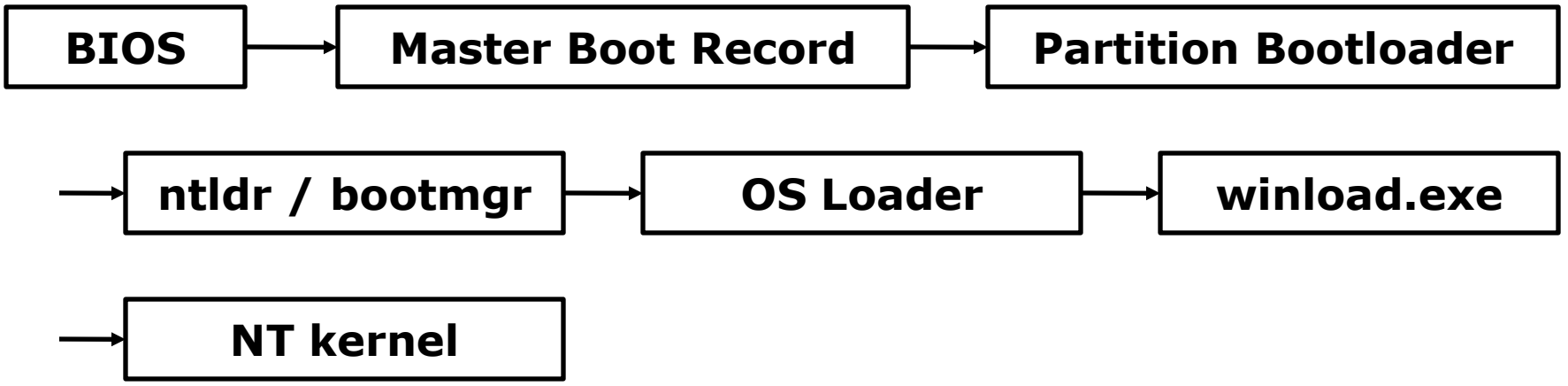
A double forward for intercepting the encrypted and decrypted disk I/O.
Does not modify the decryption software (it is independent)!

Owning Operating Systems from the boot

Bootkit Real Mode	Relocates the code to the end of memory (4 KB), hooks interrupt 13h and patches code integrity verification
Bootkit Protected Mode	Patches image verification and hooks NT kernel
Kernel Code	NT kernel base address and PsLoadedModuleList are used for resolving own imports
Driver Code	Loads, relocates, resolves, executes all drivers in the list
PE Loader	PE-image relocation & resolving
Subsystem	Core functions for the Stoned Subsystem installed in Windows
Payload	Kernel drivers Applications using the subsystem



Windows Boot Process

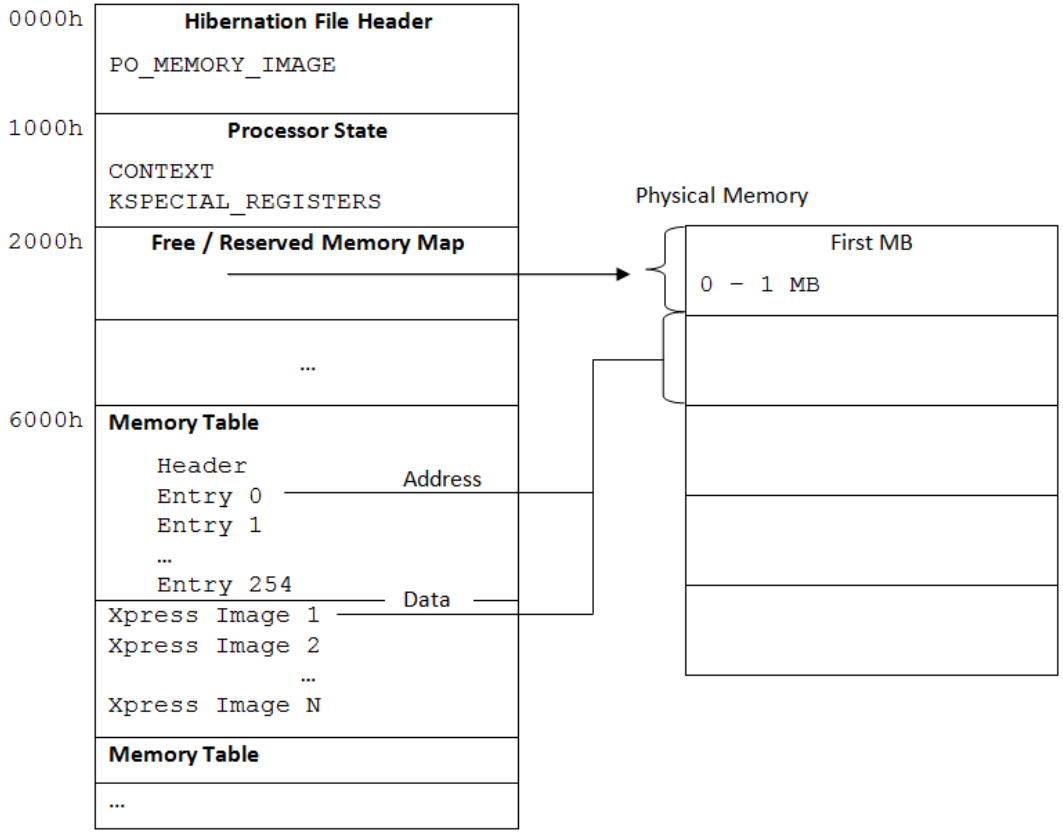


`Ntldr` = 16-bit stub + **`OS Loader`** (just binary appended)
Windows Vista splits up **`ntldr`** into **`bootmgr`**, **`winload.exe`** and **`winresume.exe`**

Windows XP	Windows Vista	Processor Environment
ntldr	bootmgr	Real Mode
OS Loader	OS Loader	Protected Mode
-	winload.exe	Protected Mode
NT kernel	NT kernel	Protected Mode + Paging

Not only “on-the-fly” attacks

For example Hibernation File Attack



Owning the system **before** it has started

Signatures – The magic behind

Signatures against operating system files ensure that

1. The bootkit stays undetected
2. The bootkit gets executed

They are all assembly code instructions.

Bypass NT Loader code integrity verification

```
+ 83 C4 02 E9 00 00 E9 FD FF
  Windows XP in NTLDR at +1C81h

00021c6e: call .+0x0c1e/+0x0c39     ; e8390c      ->    nop, nop, nop
00021c71: add sp, 0x0002            ; 83c402      ->    add sp, 0x0002
00021c74: jmp .+0x0000              ; e90000      ->    jmp .+0x0000
00021c77: jmp .+0xfffd              ; e9fdff      ->    jmp .+0x0000
```

Solutions to close out bootkits

Use the Trusted Platform Module in connection with full volume encryption

Full volume encryption software should:

1. Secure its own software
2. Disable MBR overwrite in Windows
3. Make MBR genuine verifications

Consider the attacking vector, do not excuse with policies (“physical security”)

Bootkits for law enforcement agencies

Might become interesting for LEAs:

- Install a trojan even if the hard disk is fully encrypted
- “Undetectable”, bootkit starts first and can hide itself
- Owns the whole system (full access)

- Physical access required

Stoned.. Again!

Attacks:

Windows 2000

Windows XP

Windows Server 2003

Windows Vista

Windows Server 2008

Windows 7

TrueCrypt

DiskCryptor

Main targets:

- Pwning all Windows systems from the boot
- Being able to bypass code integrity verifications & signed code checks
- Creating the most sophisticated bootkit

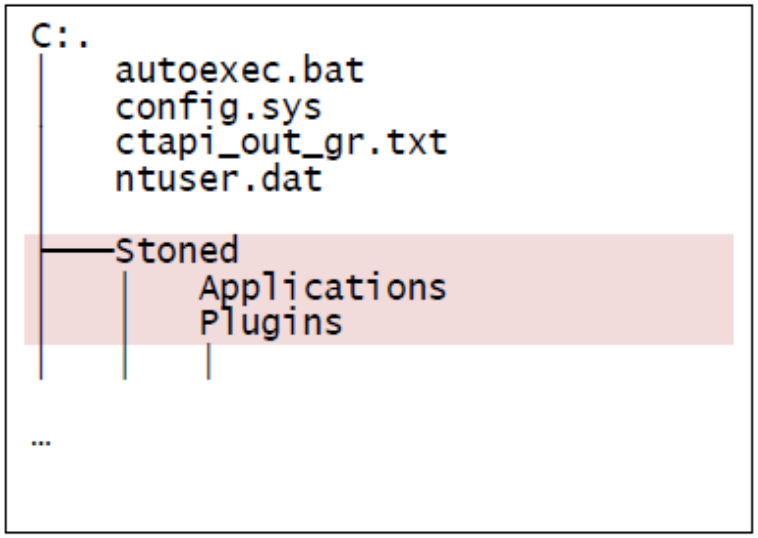
Much more features in the future!

```
Your PC is now Stoned!                    (1987)
Your PC is now Stoned!  ..again           (2010)
```

Architecture of Stoned

Address	Size	Description
0000	440	Code Area
01B8	6	Microsoft Disk Signature
01BE	4*16	IBM Partition Table
01FE	2	Signature, 0AA55h
0200	-	Stoned Kernel Modules
-	-	Stoned Plugins
7A00	512	Backup of Original Bootloader
7C00	512	Configuration Area

Master Boot Record

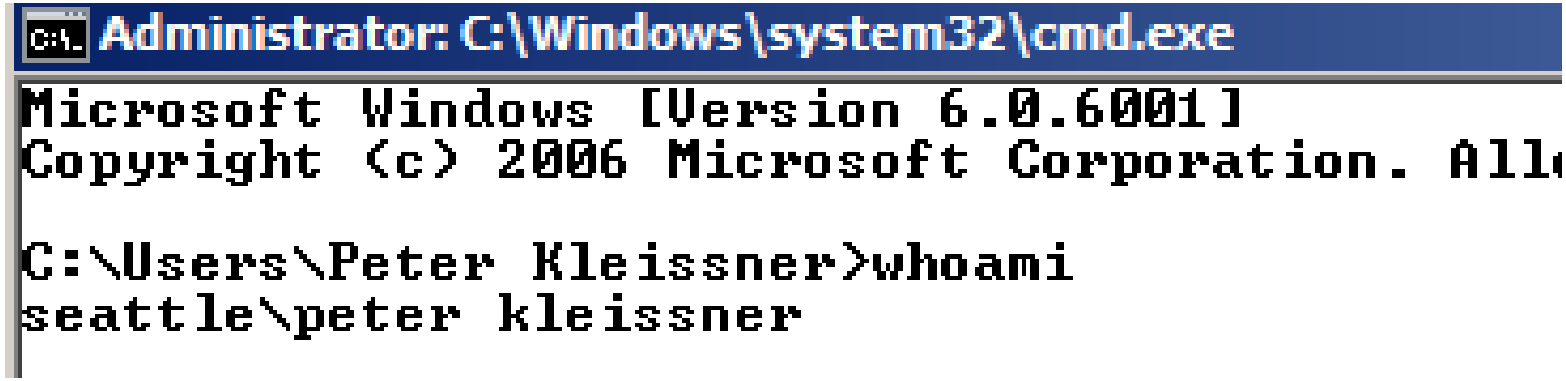


File System

- **Modularized Master Boot Record**
- **Boot Applications**
- **Plugins**
- **Proof of concept payload (cmd.exe privilege escalation)**

Time for a live demonstration!

With Stoned v2 Infector (Live CD)



Based on Windows PE

Infects any drive

Example Plugin: CO_2 Plugin

Save The Environment!

- Throttling CPU speed down to 80%
- Normal user should not take any notice but our earth does :)
- Using the Advanced Configuration Programming Interface

Using open source “Throttle source code”

```
mov     bx, ax                          ; save input
mov     dx, [ioBase]
call    getACPITHTReg                   ; get register and data
push    ax
xor     ah, ah                          ; for throttle
add     dx, ax                          ; throw away data
pop     ax

in      al, dx                          ; read current value
and     al, NOT THROTTLE_MASK           ; clear 3:0
not     ah                              ;
and     al, ah                          ;
not     ah                              ;
out     dx, al                          ; disable all throttle
cmp     bl, 0                           ; user wants none?  quit
jz      exit

or      al, bl                          ; new throttle
or      al, ah                          ; new enable
out     dx, al
```

Not only malicious purposes

Using Stoned Bootkit to execute Sinowal and extract the unpacked kernel driver

1. Tracing the memory by hooking the exports for **`ExAllocatePool()`** and **`ExFreePool()`** using the installed Stoned Subsystem
2. Writing it out to disk for further analysis

```
0007f720h: 50 4C 55 47 00 00 00 00 49 4E 46 4F 00 00 00 00 ; PLUG....INFO....
0007f730h: 42 49 50 00 2F 00 00 00 4E 4F 4F 50 00 00 00 00 ; BIP./...NOOP....
0007f740h: 55 4E 53 54 00 00 00 00 49 4E 53 54 00 00 00 00 ; UNST....INST....
0007f750h: 44 65 63 00 4E 6F 76 00 4F 63 74 00 53 65 70 00 ; Dec.Nov.Oct.Sep.
0007f760h: 41 75 67 00 4A 75 6C 00 4A 75 6E 00 4D 61 79 00 ; Aug.Jul.Jun.May.
0007f770h: 41 70 72 00 4D 61 72 00 46 65 62 00 4A 61 6E 00 ; Apr.Mar.Feb.Jan.
0007f780h: 53 61 74 00 46 72 69 00 54 68 75 00 57 65 64 00 ; Sat.Fri.Thu.Wed.
0007f790h: 54 75 65 00 4D 6F 6E 00 53 75 6E 00 0D 0A 00 00 ; Tue.Mon.Sun.....
0007f7a0h: 0D 0A 25 73 3A 20 00 00 25 78 00 00 63 68 75 6E ; ..%s: ..%x..chun
```

(Unpacked Sinowal kernel driver, here you see commands & domain name generation strings)

Future Outlook

Totally operating system independency

- Linux support
- Support for 64-bit Windows systems

Defeating Trusted Platform Module (for my next presentation)

References

[1] **Your Computer is Now Stoned (...Again!): The Rise of MBR Rootkits**
Elia Florio (Symantec) and Kimmo Kasslin (F-Secure)
http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/your_computer_is_now_stoned.pdf

[2] **VBootkit vs. Bitlocker in TPM mode**
Robert Hensing's Blog
http://blogs.technet.com/robert_hensing/archive/2007/04/05/vbootkit-vs-bitlocker-in-tpm-mode.aspx

[3] **An Analysis of the Windows PE Checksum Algorithm**
Jeffrey Walton
http://www.codeproject.com/KB/cpp/PEChecksum.aspx

[4] **Analysis of Sinowal**
Paul Kleissner
http://web17.webbpro.de/index.php?page=analysis-of-sinowal

[5] **Mebroot Source Code**
http://web17.webbpro.de/downloads/Sinowal%20Article/Sinowal%20Source%20Code.zip

[6] **Anti-Sinowal strategies and Sinowal Bootkit Extractor**
www.bootkitanalytics.com

[7] **Stoned Bootkit Project Site**
www.stoned-vienna.com

[8] **Improved Way to Add SLIC (SLP 2.0) Table into BIOS ACPI to Activate Windows Vista OEM**
http://www.betalog.com/read.php/152.htm

Thanks for your attention!

The Rise of MBR Rootkits & Bootkits in the Wild

Presentation materials: www.stoned-vienna.com

Contact: Peter@Kleissner.at

Questions?

Comments?

And have a good night =),

Peter Kleissner