Reversing firmware using radare2 [H2HC]

A. Kochkov
October, 2014
Motives

- Implement FOSS alternative (coreboot, OpenEC)
- Figure out possible attack vectors via firmware trojans

We will take only case of modern PC/Laptop/Server firmware(s).

Why?

Because it is a HUGE work!
Simple old computer diagram
Nothing really new. Usual x86/x86_64. But we just talking about early boot stages. As in the very ancient times CPU still starts in the 16bit emulation mode. Instead of old good 386 modern processors (like Intel ones) can check signature of the reset vector. But all after that can be tampered very easy.

```
r2 -a x86 -b 16 some-bios.bin
[f000:fff0]> pd 1
            ;   --  entry0;
                f000:fff0 e9f591 jmp 0xf91e8
[f000:fff0]>```
Intel Laptop booting process (simplified)
Main purpose - to fix hardware bugs. Sometimes you can see interesting erratas. Modern CPU microcode firmware is signed for both AMD and Intel (2048 bits) (PKCS#1 v1.5 padding, SHA-1, later SHA-2)\(^1\)

There is no public reversing of the microcode, while it possibly (speculation?) allow to patch MOV instruction behaviour to copy important data somewhere

\(^1\)Ben Hawker (2012-2013). *Notes on Intel Microcode Updates.*
A lot of research, ARC4 was ‘broken’ and can be exploited. ARC5 research is in progress, problems with unpacking (huffman tables, etc).

Table: ARC hardware

<table>
<thead>
<tr>
<th>Name</th>
<th>Generation 1</th>
<th>Generation 2</th>
<th>Generation 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME versions</td>
<td>1.x - 5.x</td>
<td>6.x - 10.x</td>
<td>BayTrail</td>
</tr>
<tr>
<td>Core</td>
<td>ARCTangent-A4</td>
<td>ARCTangent-A5</td>
<td>SPARC</td>
</tr>
<tr>
<td>ISA</td>
<td>ARC (32 bit)</td>
<td>ARCompact (both 32 and 16 bit)</td>
<td>SPARC v8</td>
</tr>
<tr>
<td>Manifest tag</td>
<td>$MAN</td>
<td>$MN2</td>
<td>$MN2</td>
</tr>
<tr>
<td>Module header tag</td>
<td>$MOD</td>
<td>$MME</td>
<td>$MME</td>
</tr>
<tr>
<td>Code compression</td>
<td>None, LZMA</td>
<td>None, LZMA, Huffman</td>
<td>None, LZMA</td>
</tr>
</tbody>
</table>

Table from Intel ME secrets talk

---


3 MEre project (2013-2014).


Various peripheral and power management tasks
Experimental open firmware is available from Rudolf Marek
An embedded controller of sorts in the southbridge. The controller is either enabled by hardware strap option. Or if you provide a firmware, the controller is enabled via soft strapping the chipset. It is 8051 controller.
Another embedded controller, The SMU seems to be handling PCIe power management stuff in AMD northbridges (from RS880 onwards?) the firmware is loaded during system boot. It is unknown if the firmware has to be loaded. The SMU is most likely Altera LM32 CPU.
Intel 82574L ethernet controller has had at least a few problems. Including, but not necessarily limited to, EEPROM issues, ASPM bugs, MSI-X quirks, etc. Sometimes internal CPU is so powerful that allows to run custom code on it, like e.g. SSH server.

---


7 Arrigo Triulzi (2008). “A SSH server in your NIC”. In: PacSec;
Formerly NEC V850 architecture, now Renesas Electronics V850.\textsuperscript{8} 32-bit RISC, gcc toolchain available. This firmware can modified, placed inside UEFI.

ARM and MIPS are most common controllers. Part of firmware stored in embedded flash chip and rest of it - on the hidden sectors of disk.

- Seagate HDDs firmware research\textsuperscript{9}
- Western Digital HDDs firmware research\textsuperscript{10}
- Only Toshiba HDD firmware is not reversed (yet).

\textsuperscript{9} Jonas Zaddach (2014). “Exploring the impact of a hard drive backdoor”. In: REcon.
\textsuperscript{10} Jeroen Domburg (2013). Hard disk hacking. OHM.
ATmega32u2 in Logitech G600
This is an AVR architecture ("r2 -a avr")
KBT Poker II[^12] - mechanical keyboard Nuvoton NUC122SC1AN ARM Cortex-M0 CPU

See firmware here: *Extracted Poker II binary*. gist.github.com

[^12]: *Finding the actual Thumb code in firmware*. RE stackexchange.
Keyboard controller, SPI/FWH flash access I2C bus master access

**Table:** Available vendors

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENE</td>
<td>8051 (8-bit)</td>
</tr>
<tr>
<td>Futjitsu</td>
<td>F2MC-16LX (16-bit)</td>
</tr>
<tr>
<td>ITE 8051</td>
<td>(8-bit)</td>
</tr>
<tr>
<td>Nuvoton (including former Winbond)</td>
<td>CR16 (16-bit), 8051 (8-bit)</td>
</tr>
<tr>
<td>Renesas</td>
<td>8051 (8-bit), H8S (16-bit), 740 (8-bit)</td>
</tr>
<tr>
<td>NSC</td>
<td>CR16 (16-bit), 8051 (8-bit)</td>
</tr>
<tr>
<td>SMSC</td>
<td>8051 (8-bit)</td>
</tr>
<tr>
<td>SST</td>
<td>8051 (8-bit)</td>
</tr>
</tbody>
</table>

You can get dump of your EC registers using ectool.13

---

13 **ECtool.** coreboot project.

14 **Embedded Controller.** coreboot project.
AVR or PIC architecture\textsuperscript{15,16}

\textsuperscript{15} Synaptics RMI3 Interfacing Guide (2008).
Full featured computer, including CPU and video processor. Very often 8051, H8, AVR or ARM based. For example Vimicro VC0343\textsuperscript{17} 8051 based one. Can be exploited using the Device Firmware Update (DFU) standard. It allows to start the update without administrator privileges (for Windows systems).\textsuperscript{18}

\textsuperscript{17}Vimicro VS0343 - USB 2.0 Camera Processor (2011).
\textsuperscript{18}Robert Graham (2013). How to disable webcam light on Windows.
SD/MMC cards

8051 and H8 processors\textsuperscript{19,20,21}

\textsuperscript{19} “The Exploration and Exploitation of an SD Memory Card” (2013). In: 30C3.
\textsuperscript{21} Phison microcontroller firmwares and flashers. usbdev.ru.
The baseboard management controller is the implementation of IPMI. It is a specialised microcontroller embedded on the mainboard of the server. There are a few various vendors of BMC/IPMI:

- HP iLO
- Dell iDRAC
- IBM RSA
- Intel AMT
- DTMF DASH
- and less common
BMC and IPMI

IPMI Block Diagram

- Southbridge, Super IO
- NIC
- LPC Bus
- SMBus
- SideBand
- BMC
- IPMI & OEM Signals
- I2C Bus
- IPMB, HW Monitor, Power Supply, DIMM, Chipset, PCI Slots etc.
- Serial Port
- Switching Logic
- Serial Port Connector
- Super IO
HP iLO

- iLO 2 - 66 MHz NEC v850
- table iLO 3/4 - ARM cpu + NAND flash to store firmware (up to 4Gb)
**Dell iDRAC**

Table: Available versions

<table>
<thead>
<tr>
<th>Family</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAC II - 5(^{22})</td>
<td>External PCI/PCle card</td>
</tr>
<tr>
<td>iDRAC 6 - 7</td>
<td>Integrated on motherboard</td>
</tr>
</tbody>
</table>

\(^{22}\) Ruben (2011). *Reversing Dell’s DRAC Firmware.*
8051 and H8 controllers (similar to SD/MMC cards) Alcor Micro, CION, Etron, Hisun, ITE, JMicron, KTC, Netac, OTi, Phison\textsuperscript{23}, Prolific, SanDisk, TM, Winyatek and many others.

\textsuperscript{23} Phison 2251-03 (2303) Custom Firmware and Existing Firmware Patches.
Each PCI/PCle device can provide its firmware to start on the main CPU, in the BIOS/UEFI environment, as a driver. So we can just use the same tools and techniques as for BIOS/UEFI reverse engineering. A lot of research has been done here already.²⁴²⁵²⁶

What is radare2

This is reverse engineering framework and toolset. Main tool (r2) have two modes of work: command line and visual (V* commands). Also there is a bokken GTK GUI. But we'll use r2 tool instead.
Important commands:
- pd - print disassembly
- f - set/show flag
- s - seek
- af - add function
- CC - add comment
- Cd - mark as data
- w* - write back to the file
- Vp - visual modes (note pressing ‘p’ to switch between them)
Here we do:

- Open legacy BIOS file to reverse
- Open modern system UEFI firmware
- Open PCIe device option rom
Used for iLO and iDRAC. You need to properly manage loading address.

- **S** - sections command
- **io.va** - virtual addresses evaluation

```
r2 -a arm -b 32 -e io.va=true some-arm-firmware.bin
[0000000]> S 0 0x10000 0x2000 0x2000 load rwx
```
<table>
<thead>
<tr>
<th>Demo 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEMO 2</td>
</tr>
</tbody>
</table>
Using as part of:

- EC (Embedded Controller)
- Webcam controller
- SD/MMC card controller
- USB Flash sticks controller
- AMD IMC controller
- HDD servo control
Using as part of Intel ME/AMT
We will open both ARC4 and ARC5 examples
Using as part of EC (Embedded Controller)
DEMO 5

Reversing HP Omnibook 500 EC firmware
This tool have both GUI and CLI versions

Rest of tools

- bios_extract\textsuperscript{28}
- dump from memory
- dump from device (using some equipment)
- copy file from the linux sources (for uploadable firmwares)

\textsuperscript{28} Bios\textunderscore extract.
Emulation

- SerialICE\textsuperscript{29}
- S2E/Avatar\textsuperscript{30}
- PANDA\textsuperscript{31}

We can use these tools with r2, due to support of gdb: protocol.

\textsuperscript{29} SerialICE.  Tracing PC firmware using patched QEMU.
\textsuperscript{30} Avatar - dynamic firmware analysis framework.  based on QEMU.
\textsuperscript{31} PANDA - Platform for Architecture-Neutral Dynamic Analysis.  based on QEMU.
BIOS/UEFI

- UEFI Tool
- + flashrom\textsuperscript{32}
- + external programmer (rpi/buspirate)

\textsuperscript{32} flashrom - crossplatform PC firmware flashing tool.
External Programmer

- patched flashrom or
- some custom tools
Thanks for your attention!

Questions?
Avatar - dynamic firmware analysis framework. based on QEMU.
Bios_extract.
Domburg, Jeroen (2013). Hard disk hacking. OHM.
ECtool. coreboot project.
Embedded Controller. coreboot project.
Extracted Poker II binary. gist.github.com.
Finding the actual Thumb code in firmware. RE stackexchange.
flashrom - crossplatform PC firmware flashing tool.
Intel ME 6.x Huffman algorithm (2014).

MEre project (2013-2014).

PANDA - Platform for Architecture-Neutral Dynamic Analysis. based on QEMU.


Phison 2251-03 (2303) Custom Firmware and Existing Firmware Patches.

Phison microcontroller firmwares and flashers. usbdev.ru.


Salihun, Darmawan. Building a Kernel in PCI Expansion ROM.

SerialICE. Tracing PC firmware using patched QEMU.


Vimicro VS0343 - USB 2.0 Camera Processor (2011).
