Fun with BIOS option ROMs

Tobias Kaiser, mail@tb-kaiser.de

January 26, 2018

AFRA Lightning Talk, afra-berlin.de

Outline

1. Boot process
2. What can be done with option ROMs
3. Making option ROMs
4. Example option ROM application: ahci_sbe

Boot process

PC boot process

1. PC gets powered on, CPU reset
2. CPU jumps to entry point, which is in BIOS ROM
3. BIOS initializes hardware and looks for OS
4. Bootloader (e. g. GRUB) is loaded and invoked by BIOS
5. Operating system is loaded and invoked by bootloader

PC boot process

1. PC gets powered on, CPU reset
2. CPU jumps to entry point, which is in BIOS ROM
3. BIOS initializes hardware and looks for OS
    - PCI devices can come with own code that is executed as part of this process → **option ROM**
4. Bootloader (e. g. GRUB) is loaded and invoked by BIOS
5. Operating system is loaded and invoked by bootloader

What can be done with option ROMs

What can be done with option ROMs

Intended purposes:

1. Run initialization code for PCI card hardware
2. Enable network boot

Unintended purposes:

1. Implement a functionality that your BIOS does not support, such as ATA security commands
2. Custom power-on authentication
3. Rootkits
4. Be creative here

Making option ROMs

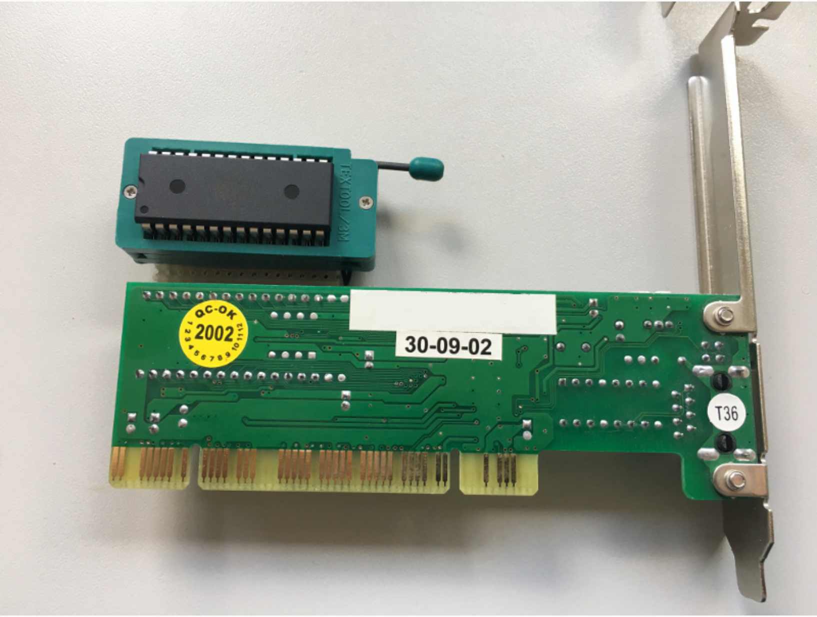
How to start hacking: software

- Look for “PCI Expansion ROM Header” in PCI Local Bus Specification
- Machine code from the option ROM will be loaded into RAM by BIOS and then executed
- Start writing x86 real mode assembly
- Call POST memory manager (PMM) if you need memory.
- Use BIOS calls for Basic I/O Stuff
- Writing C and using gcc is not straightforward, because you will need to switch to protected mode and lose BIOS call capability

How to start hacking: hardware

1. Start out with virtual hardware
   - QEMU option `-option-rom`
   - VirtualBox `vboxmanage setextradata ... VBoxInternal/Devices/pcbios/0/Config/LanBootRom MYROM`
   - Limitations: stuff that is not emulated
2. PCI card with interchangeable ROM socket
   - Can be reprogrammed
   - Unlikely that you will brick your mainboard this way
3. Embed it into your BIOS image
   - Be careful that you don't brick your mainboard!
   - I have never done this.

Option ROM deployment: best practices?



Example option ROM application: ahci_sbe

Example option ROM application: ahci_sbe

AHCI: Advanced Host Controller Interface = standard SATA controller interface

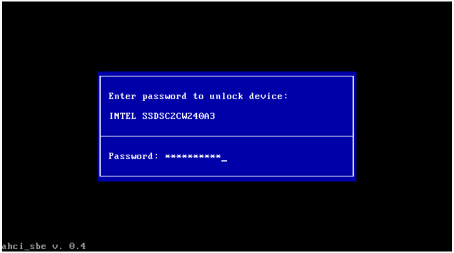
ATA security commands: Allow password protection for (self-encrypting) hard disks

ahci_sbe: AHCI secure BIOS extension

Example option ROM application: ahci_sbe

Some mainboards come with a BIOS that support ATA security commands (i. e. ask for hard disk password and send that to the self-encrypting hard disk), others don't. **If your mainboard doesn't, ahci_sbe is the hack to make it work anyway.**

Surprising that there is no easier solution **(yet)**.



```
Enter password to unlock device:

INTEL SSDSC2CW240A3

Password: **********_
```

```
ahci_sbe v. 0.4
```

- `https://github.com/TobiasKaiser/ahci_sbe`