BOOTJACK

A DARPA CFT Project

Nov 2012 – Aug 2013

By

Vikas N Kumar

SELECTIVE INTELLECT

Overview

- Attack Vector
- Use Cases
- Technique
- Details
- Demo

selective intellect™

Small Scale Attacker's Thoughts

- Server/Desktop/Laptop mobos have a BIOS
- If there is software in the BIOS, then there can be malware in the BIOS
- Let's write a rootkit for the BIOS
- Conventional methods of detection fail successfully
- Too difficult to support various mobos !
- Plus there is that TPM chip thing

Nation State Attacker's Thoughts

- Contact Manufacturer of Motherboards
- Take BIOS source code by hook or by crook
- *Customize* BIOS with <bad>ware
- TPM ? Haha ! Same problem as Certificate Authority in SSL
- Verify strategy with NIST Special Report 800-147
- Easy support for variety of systems

Use Case

- Govt ~~data centers~~ *cloud infrastructure* have servers and its users have desktops/laptops
- BIOS <bad>ware allows for permanent *sleeper cell* botnet
- Hostile Govt leverages <bad>ware to cause
  - DDoS
  - Malfunction
  - Tactical attacks

Another Use Case

- HFT firm's servers run next to other HFT firm's servers
- One firm's server attacks another by leveraging a targeted malicious BIOS
- Manufacturers market same servers between different firms
- One firm can help another cause a flash crash to pocket a cool profit
- A Govt can do it to other economies

Current State of Things

- Govt/Corp purchases `100`s of machines at a time
- No standard/easy way to verify BIOS after purchase or after months of use
- No single solution can solve this problem
- Hardware based BIOS extraction tools very expensive
- Need computer chassis to be opened

Technique

- Plug ***Bootjack*** device into USB port of Target
- Reboot Target
- Appear as a USB drive to the BIOS
- Inject “OS” into Target
- Extract BIOS blobs from RAM and BIOS chip onto self
- Compare with “certified” BIOS blobs
- Perform forensic analysis on blob at leisure

Why use an external device ?

- Software detection works but…
  - Needs installation on every system
  - High costs – money and employee time
  - If OS is compromised in any way, might not work as expected
- One device for N machines, N >= 1
- No special installation of software needed
- Compromised systems cannot interfere

Why use an external device ?...

- All verification and analysis done on Bootjack device
- All extraction done by custom “OS” injected into target
- Harder to tamper with
- Inexpensive solutions benefit everyone

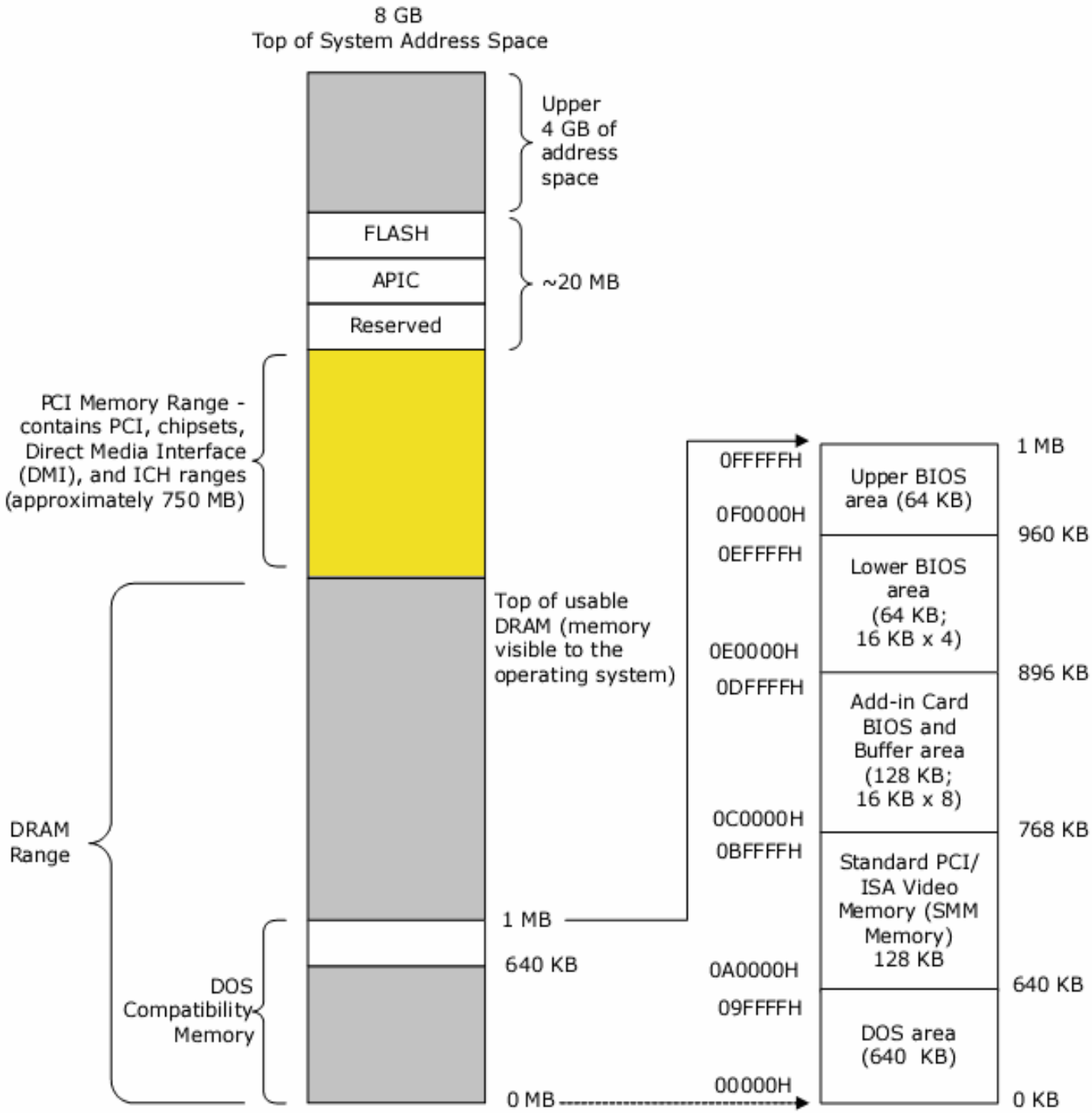
selective intellect™

Implementation Steps

- Create a custom OS to load into target
- Verify that it works by using as Live CD
- Verify it works for Hypervisor BIOSes such as QEMU/VirtualBox/VMWare
- Implement the above in hardware to create a device
- Support certain vendor systems to demonstrate Proof of Concept

Bootjack OS

- Needs to be injected into target system by device or Live CD
- Combination of a custom boot loader & customized Linux
- Boot loader extracts BIOS blobs from various areas of RAM and writes to *virtual* disk
- Linux is then loaded
- Performs BIOS extraction from chip using applications



Extracted Output

- Extract chip contents (1MB or more) using flashrom (limited)
- 64-128 KB of BIOS loaded in RAM at E000:0000h – F000:FFFFh
- 128 KB Option ROMs loaded in RAM at C000:0000 – D000:FFFFh
- Interrupt Vector Tables – most common place for MBR based rootkits to modify int 13h
- NVRAM/CMOS information
- MBR of every disk and every partition

selective intellect™

Device Implementation

- Started out with a Xilinx FPGA board costing about ~ $500
- After 6 weeks of work, realized an ARM board with Client USB capabilities can solve the problem too
  - Overo by Gumstix
  - Beaglebone Black
  - Raspberry Pi
- Picked Beaglebone Black to demonstrate
- Brought down hardware cost to < $100

Bootjack-Ångstrom OS

- Client USB capability provided by Linux kernel module for USB gadgets `g_multi` along with some patches
- Store Bootjack OS as a file along with a file formatted as `FAT32` drive for writing
- USB gadget "serves" these files as virtual drives to target system
- All reads/writes done to the `FAT32` virtual drive
- Any infection by target into this file system can be reverted easily

How to Use the Device

- User turns on Bootjack device
- Plugs device into target system
- Reboots/Starts the target system
- System loads device as USB drive
- System sees `2` "drives": `1` bootable (Bootjack OS) and the other a non-bootable `FAT32` disk
- System boots from this "drive"
- Bootjack OS runs inside the target system

How to Use the Device…

- Writes extracted data back to the *virtual* `FAT32` drive
- Waits for verification from Bootjack device
- Bootjack OS reboots the target system
- Writes extracted data again
- Waits for verification from Bootjack device
- Bootjack OS shuts the target system down
- Can now plugin the Bootjack device into another system and repeat…

selective intellect™

Verification…

- All extracted files stored on the virtual `FAT32` disk on the device
- Separate program runs in the background that mounts this disk, reads files and stores `SHA-256` signatures and motherboard details in a local SQLite database
- If target system allows, display signatures on screen of the target system for user to view any changes

Offline Forensic Analysis

```
   binwalk -W -K 8 -i 153413/nvram.dump 153915/nvram.dump

OFFSET    nvram.dump                                    nvram.dump
------------------------------------------------------------------------------------------
-----
*                                                  /
00000050  41 00 B9 61 00 B4 D9 EB  |A..a....|      \   41 00 B9 61 01 B4 D9 EB  |A..a....|
00000058  00 00 33 01 03 00 09 8E  |..3.....|      /   00 00 34 01 03 00 09 8E  |..4.....|
*                                                  \
```

- Perform manual analysis on extracted files if `SHA-256` is different
- Most changes are trivial
  - Date/time of booting
  - Soft/Hard reset flag as shown above
  - Checksum variations based on changes above
  - Change in peripherals attached
  - Changes made by user in BIOS

selective intellect

POC Target System

- BIOS is very specific to a particular mobo
- `flashrom` supports limited systems
- For POC we chose Dell Optiplex `980`, Acer Aspire One `250` Netbook
- Patched `flashrom` to support the Dell explicitly
- `flashrom` doesn't support laptops
- Custom Bootloader works on all standard BIOSes

Limitations

- Not all motherboards supported
- Manual forensic analysis on extracted blobs
- Certification of ***valid*** BIOS based on vendor BIOS available on their website
- Needs to reboot system to do task
- Boot from USB should be the top preference in the BIOS boot menu

Future Enhancements

- Automated Forensic Analysis on the extracted blobs
- Self Integrity check of the hardware itself
- Support more BIOS chips for direct extraction
- Explicit extraction of Option ROMs from peripheral devices instead of RAM only

Summary

- Bootjack device performs BIOS blob extraction
- Software (Bootjack OS) is injected into target to perform extraction
- Any verification or analysis happens on the device
- Target system is passive and does no work except run the extraction code and write data
- Can reuse the device on various machines

DEMO

selective intellect™