The Apple of Your EFI

An Updated Study of EFI Security

Pepijn Bruiennes (@bruienne)
Rich Smith (@iodboi)
Duo Security
Black Hat Europe 2017
6/12/2017
About Us

- **Pepijn Bruienne (@bruienne)**
  - R&D Engineer in Duo Labs
  - Focus on R&D, RE and further breaker of all things Apple
  - Recovering Mac Admin
  - OSS maintainer of some popular Mac Admin tools

- **Rich Smith (@iodboi)**
  - Director of R&D, Duo Labs
  - Enjoy researching at scale, post-exploitation, firmware & Python
  - Worked in security far too long now I think!
About Duo

- We’re with Duo Labs, Duo Security’s research group
- We break things (or attempt to) and then:
  - Write code to un-break it
  - Talk about it
  - Write papers & blogs about it
- We build things:
  - Prototype new security products & approaches for Duo
  - Think about what customers future security needs will be
  - Release open source code to share things we experiment with
    - Check out the recent releases of IsThisLegit? and Phinn
- We’re hiring!
Shine some light onto firmware security as compared to software security.

We analysed all OS, Security and EFI firmware release by Apple for 10.10/.11/.12/13
- This is all about what Apple is releasing EFI update wise
- About 3 years worth of update data, discovered many anomalies

We got data from >73K Mac systems to see the real-world state of EFI installs
- All about how well the EFI updates Apple released are being installed

We compared both datasets to see how well the real world matched the expected state of EFI versions running.
Things We Will Cover Today

Context

- History of existing Apple EFI security research
- How Apple Macs update their firmware

The New Work

- What we did
- What we found
- What you can do
- What we’re releasing
Lots more information in our technical paper

http://duo.sc/2x1AA9R
EFI is Everywhere

Intel EFI - mid-1990s

UEFI standard - 2005

Apple EFI - 2006
EFI Killed the Open Firmware Star

- **Apple EFI**
  - Shipped in first-generation Intel Macs in early 2006
  - Intel switch = **no more Open Firmware** (PPC legacy)
  - First models shipped with EFI were iMac and MacBook Pro
  - Must support new Mac hardware and features
  - Supports platform-specific things like:
    - NetBoot
    - Internet Restore
  - Entirely invisible to end users
What Makes Attacking EFI Attractive?

- **Stealth**
  - It’s very hard to detect if EFI/firmware is compromised

- **Persistence**
  - It’s hard to remove implants from EFI
  - Reinstalling the OS or replacing HDD is not sufficient

- **Access to everything***
  - Running at Ring -2 means that security controls at higher layers can be circumvented
  - Pretty much arbitrary read/write to disk and memory

---

**Protection Rings**

- Ring 3: Applications
- Ring 0: OS / Kernel
- Ring -1: Hypervisor
- Ring -2: EFI
Who Wants to Attack EFI?

- Well-funded adversaries:
  - ‘Nation-states’
  - Industrial espionage
- This is **not** your script kiddie tool
A Brief History Of Apple EFI Security
EFI Security Research in Brief

- ‘DE MYSTERIIS DOM JOBSIVS’ - 2012 - snare - Black Hat 12
- Sonic Screwdriver - 2012+ - Wikileaks ‘Vault7’ leaks - Sea Eye Aye?
- ThunderStrike 1 - 2014 - Trammell Hudson @ 31c3
- ThunderStrike 2 - 2015 - Trammell Hudson, Xeno Kovah, Corey Kallenberg @ Defcon 23
- PCI DMA attack - 2016 - Ulf Frisk @Defcon24
- Lots of other cool EFI research
  - Pedro Vilaça - Is There an EFI Monster Inside Your EFI? @ 44Con/SyScan Beijing ...
  - https://reverse.put.as
Mac EFI Firmware Updates
How Does EFI Get Updated?

Prior to 2015 - manual

After ThunderStrike
The Apple EFI Update Process

- **As of late 2015 - ThunderStrike response**
  - Update is shipped as a standard Apple PKG installer
  - Payload-free PKG runs `efiupdate` to bootstrap EFI updater
  - Ships with complete set of current EFI `.scap` or `.fd` bundles
  - `$MODEL_$MAJOR_$MINOR_LOCKED.$EXT`
  - The postinstall script invokes `efiupdate`
  - The tool copies the correct firmware to the ESP (EFI System Partition)
  - The file is then blessed using the `bless` command:
    - `bless -mount / -firmware MBP111_0138_B21_LOCKED.scap --verbose --recovery`
  - EFI update gets one shot to get it right, will not run again until next OS update
Mac EFI and Mac Board IDs
Figuring Out Compatibility

Apple has to ship a lot of EFI payloads

Does every model have its own EFI payload?

Does the EFI payload contain the info?

How can the updater know?
Matching Specific Models With EFI

- Apple uses various identifiers for Mac models
  - Model ID: `<Model><major,minor>`
  - Example: iMac17,1
    - Minor version denotes different configs (21”, 27”)
    - Apple reuses these when spec bumps happen
    - In the past Apple shoved multiple configs into one single major,minor model ID
  - Board ID: `Mac-<8 or 16 character hex string>`
  - Unique for specific model and rev
  - Stored on logic board, get via `ioreg -l | grep -i board-id`
  - Example iMac17,1 ==
    - Mac-B809C3757DA9BB8D
    - Mac-DB15BD556843C820
    - Mac-65CE76090165799A
Matching Specific Models With EFI

- How does the firmware updater match EFI and Mac model?
  - The EFI payload contains board IDs with which it is compatible
  - Stored in GUID 781F254A-C457-5D13-9275-1BF5D56E0724
  - Firmware updater looks for 4-byte header 0x7C000019
  - 8 byte chunks are used for storing compatible board IDs, up to 120 bytes
  - Represents the hex string of the board ID Mac-B809C3757DA9BB8D
  - Grabs board ID of Mac via IOREgistry API
  - If match is found == use this EFI firmware bundle
Does Every Model Get Its Own EFI Bundle?

- Some models are rolled into a single EFI bundle
  - Example: MacBookAir7,1 and MacBookAir7,2
  - MBA 7,1 = 11” model / MBA 7,2 = 13” model
  - MacBookAir7,2 EFI version string: MBA71.0166.B26
  - What does this mean?
  - Some EFI payloads contain multiple board ID entries
  - GUID 781F254A-C457-5D13-9275-1BF5D56E0724 holds up to 15
  - Apple uses this GUID to group compatible models, fewer files to maintain
    - ...thus fewer files to potentially mess up (more later)
EFI Update Flowchart

1. **Entrypoint**
   - No: **Path flag**
     - Yes: File path containing EFI firmware updates
     - No: Obtain board-id with `IORegistryEntrySearchCFProperty`

2. Iterate EFI files

3. Go to known offset for board ID mapping GUID

4. GUID found
   - Yes: Have board-id match?
     - Yes: Assign EFI firmware
     - No: END
   - No: END

5. GUID not found
   - END
Our Research Questions
What Did We Want To Find Out?

How well does EFI firmware security support compare to software security support?

Are all Mac systems treated equally in terms of EFI patches?

Are all OS versions treated equally in terms of EFI patches?

How well does the real world compare to what Apple released?

What is the visibility to EFI security support for admins & end users?
Common EFI Update Issues

- **Do EFI updates “just work”?**
  - EFI updates are hidden, slip-streamed with OS updates
  - This must mean they *always* work and have a robust fail mode, right?

- **Do same EFI updates ship for all “supported” OS versions?**
  - If the EFI support model matches the OS one they should be identical

- **Does the real world match the ideal world?**
  - A check in the field should come back with 100% match

- **If it does not, can Apple users find out easily?**
  - Spoiler alert: not quite that simple
The Analysis
**Phase 1**
Extract the EFI updates from OS/Security updates released by Apple

**Phase 2**
Gather real world data from production Macs

**Build Dataset**
- OS Build number
- Board-ID
- EFI Version

**Analysis 1**
Look for anomalies in the EFI firmware Apple has released

**Analysis 2**
Look for discrepancies between EFI in production and EFI we expect to see

**Profit**
Conclusions!
Phase 1 - Build a Picture of **What** Apple Released

- First we gathered all OS & Security updates released by Apple for 10.10, 10.11, 10.12 & 10.13
- Extracted all the EFI updates from them
- Built a dataset of triplicates, this formed an idealised **baseline** dataset

<table>
<thead>
<tr>
<th>OS Build number</th>
<th>Mac Model/Board ID</th>
<th>EFI Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>16G29</td>
<td>IM151/ Mac-42FD25EABCABB274</td>
<td>IMI151-0207-B29</td>
</tr>
<tr>
<td>10.12.6</td>
<td>iMac 27” 5K Late 2014</td>
<td>EFI Ver. 0207 Build 29</td>
</tr>
</tbody>
</table>
• Looked for anomalies & discrepancies in the EFI updates Apple released
  ○ Which Mac models saw EFI updates and when they saw them
  ○ Missing EFI updates
  ○ Differences between the updates released for 10.10, 10.11, 10.12, 10.13
  ○ EFI version anomalies
Phase 2 - Build a Picture of **How** EFI Looked in the Real Word

- Collected data from **73,383** Macs deployed in production
  - Same data triplicates of OS Build, Mac Model/Board-ID, and EFI version

- Of those we extracted **54,744** that were running 10.10, 10.11, 10.12
  - Older OS versions were no longer under security support by Apple
  - Old EFI is the least of their problems!!

- For any Mac model, on a specific OS version we could predict the EFI it *should* be running

- We then compared the datasets to see how well the **real world** matched the **perfect world** model built from the updates themselves
<table>
<thead>
<tr>
<th>Model ID</th>
<th>Thunderstrike 1</th>
<th>Thunderstrike 2</th>
<th>CVE-2015-4800</th>
<th>DMA Attack (CVE-2016-7385)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SecUpdate2015-001 (10.9)</td>
<td>SecUpdate2015-001 (10.8)</td>
<td>SecUpdate2015-004 (10.10)</td>
<td>SecUpdate2017-001 (10.16)</td>
</tr>
<tr>
<td>Mac7.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac8.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac9.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac9.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac10.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac10.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac10.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac10.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac11.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac11.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac11.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac12.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac12.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac13.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac13.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac13.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac14.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac14.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac14.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac14.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac15.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac15.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac15.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac15.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac16.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac16.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac17.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Air 1,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Air 2,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 13,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 13,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 15,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 15,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 15,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 15,4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 16,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 16,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 17,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 17,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 17,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 17,4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 17,5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 17,6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 17,7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 18,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 18,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 18,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 18,4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 18,5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 18,6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MacBook Pro 18,7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Red indicates affected models.
- Green indicates non-affected models.
- Yellow indicates partially affected models.
Research Findings
What Issues Exist?

Real world systems are out of date

Security Updates gradually drop EFI model support

Quiet failures and lack of visibility

Unexplained EFI update regressions
Successful OS Update Does Not Mean Successful Firmware Update
Real world != perfect world

- Data analysis reveals not all endpoints get EFI updates
  - Gathered data from various Edu and Enterprise orgs
  - Anonymous data showing:
    - Model ID
    - OS version string
    - OS build string
    - EFI version string
  - Example: “MacBookPro10,1”, “10.12.4”, “16D25”, “MBP101.00EE.B12”
  - Compare against known-good lookup table
  - Measure non-compliant records
Which model X, with OS version Y, has EFI version older than it should be?
4.2% Running incorrect EFI version Average Across All Data
42.9% Most out of date model
iMac16,2 / iMac 21” Late 2015

10% Highest overall OS deviancy
macOS 10.12 Sierra
3.4% OS X 10.10 Yosemite
Overall OS deviancy

2.1% OS X 10.11 El Capitan
Overall OS deviancy
<table>
<thead>
<tr>
<th>Mac Model</th>
<th>% Running Older-Than-Expected EFI Version</th>
<th>Raw Count of Systems Running Older EFI</th>
<th>Total Count of Systems Running Older EFI</th>
</tr>
</thead>
<tbody>
<tr>
<td>iMac16,2</td>
<td>43.0%</td>
<td>941</td>
<td>2190</td>
</tr>
<tr>
<td>MacBookPro13,2</td>
<td>34.8%</td>
<td>114</td>
<td>328</td>
</tr>
<tr>
<td>MacBookPro13,1</td>
<td>28.5%</td>
<td>39</td>
<td>137</td>
</tr>
<tr>
<td>MacBookPro13,3</td>
<td>24.8%</td>
<td>78</td>
<td>314</td>
</tr>
<tr>
<td>MacBookPro8,2</td>
<td>14.9%</td>
<td>89</td>
<td>598</td>
</tr>
<tr>
<td>MacBookPro8,1</td>
<td>11.9%</td>
<td>59</td>
<td>498</td>
</tr>
<tr>
<td>Macmini3,1</td>
<td>11.5%</td>
<td>6</td>
<td>52</td>
</tr>
<tr>
<td>Macmini6,1</td>
<td>6.7%</td>
<td>13</td>
<td>194</td>
</tr>
<tr>
<td>iMac16,1</td>
<td>5.2%</td>
<td>15</td>
<td>287</td>
</tr>
<tr>
<td>MacBookAir6,1</td>
<td>5.0%</td>
<td>29</td>
<td>586</td>
</tr>
<tr>
<td>MacBook9,1</td>
<td>4.9%</td>
<td>10</td>
<td>206</td>
</tr>
<tr>
<td>Macmini7,1</td>
<td>4.8%</td>
<td>50</td>
<td>1035</td>
</tr>
<tr>
<td>MacBookAir4,1</td>
<td>4.4%</td>
<td>6</td>
<td>138</td>
</tr>
<tr>
<td>MacBookPro8,3</td>
<td>4.4%</td>
<td>3</td>
<td>69</td>
</tr>
<tr>
<td>iMac13,1</td>
<td>4.1%</td>
<td>86</td>
<td>2119</td>
</tr>
<tr>
<td>MacBookAir6,2</td>
<td>3.6%</td>
<td>81</td>
<td>2344</td>
</tr>
</tbody>
</table>
Software Secure
But Firmware Vulnerable
Does “Supported OS” Mean “Supported EFI”?

- **Apple has a “soft” support deprecation schedule**
  - Only Apple truly knows what this schedule is exactly
  - Roughly “N-2” model
  - Security Updates ($\text{SecUpd}^{<20XX>_-<YYY>}$) are subsets of current OS update
    - As name implies security patches only + EFI updates
  - Some of this is because issues don’t affect older OS versions
  - But also because Apple just chose not to backport & QA (ntp, Broadpwn)

- **Security updates also quietly drop certain EFI updates in current OS updates**
  - Not current == Security updates == EFI quietly stops getting updates
  - Only way to be sure of broadest coverage: run current OS and update immediately
EFI coverage by update:

macOS Sierra 10.12.6 Update
El Capitan Security Update 2017-003
Yosemite Security Update 2017-003

43 EFI bundles
31 EFI bundles
1 EFI bundle

Sierra 10.12
El Capitan 10.11
Yosemite 10.10
A patch for one does not mean a patch for all
One patch to rule them all?

- Apple states in their update notes when EFI vulnerabilities are fixed, but it does not give details for exactly which Mac models they patch.

- We looked at which models of Mac Apple released EFI updates for that addressed 4 high impact public EFI vulnerabilities:
  - CVE-2014-4498 - (Thunderstrike 1)
  - CVE-2015-3692 - (Thunderstrike 2)
  - CVE-2015-7035 - (“An attacker can exercise unused EFI functions”)
  - CVE-2016-7585 - (Ulf Frisk’s DMA attack)
Mac models that did not have EFI updates released for each vulnerability

47  31  25  22

Thunderstrike 1  Thunderstrike 2  CVE-2015-7035  CVE-2016-7585
Some Models Are EFI Orphans
Do All OS Supported Models Get EFI Updates?

- Apple has spotty coverage for certain “supported” models
  - A number of models are supported by current OS and updates
  - The OS updates have not bundled EFI updates for these models
  - Other models were seen with only “factory” EFI, no further updates
    - Factory EFI: low-numbered <model>.<major>.BXX version
    - MBPXXY.NNNN.B00
  - This adds to the incorrect assumption that they are fully secure
Models Lacking EFI Updates

16 Models not receiving any EFI updates
18 Models with only factory EFI versions
<table>
<thead>
<tr>
<th>Mac Model</th>
<th>EFI Versions Observed in the Real World Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM101</td>
<td>IM101.00CC.B00</td>
</tr>
<tr>
<td>IM71</td>
<td>IM71.007A.B00 IM71.007A.B01 IM71.007A.B03</td>
</tr>
<tr>
<td>IM81</td>
<td>IM81.00C1.B00</td>
</tr>
<tr>
<td>IM91</td>
<td>IM91.008D.B00 IM91.008D.B04 IM91.008D.B08</td>
</tr>
<tr>
<td>MB51</td>
<td>MB51.007D.B03 MB51.007D.B06</td>
</tr>
<tr>
<td>MB52</td>
<td>MB52.0088.B06</td>
</tr>
<tr>
<td>MB61</td>
<td>MB61.00C8.B00</td>
</tr>
<tr>
<td>MBP31</td>
<td>MBP31.0070.B07</td>
</tr>
<tr>
<td>MBP41</td>
<td>MBP41.00C1.B03</td>
</tr>
<tr>
<td>MBP51</td>
<td>MBP51.007E.B05 MBP51.007E.B06</td>
</tr>
<tr>
<td>MBP52</td>
<td>MBP52.008E.B05</td>
</tr>
<tr>
<td>MBP53</td>
<td>MBP53.00AC.B03</td>
</tr>
<tr>
<td>MBP55</td>
<td>MBP55.00AC.B03</td>
</tr>
<tr>
<td>MM31</td>
<td>MM31.00AD.B00 MM31.0081.B06</td>
</tr>
<tr>
<td>MP31</td>
<td>MP31.006C.B02 MP31.006C.B05</td>
</tr>
<tr>
<td>MP41</td>
<td>MP41.0081.B04 MP41.0081.B07 MP41.0081.B08</td>
</tr>
<tr>
<td>MP51</td>
<td>MP51.007F.B00 MP51.007F.B01 MP51.007F.B03</td>
</tr>
</tbody>
</table>

This table lists Mac models from the real world dataset that have only been observed with one, two or three updates with low build numbers. This suggests they haven’t been updated from the versions of firmware they were originally shipped with from the factory - making them likely to contain unpatched vulnerabilities.
Silent Updates Will Also Fail Silently
Do Failed Updates Generate Alerts?

- Apple designed the process to be silent to end users
  - This means it gets one shot to succeed
  - No retries outside of OS/Security updates until next version
  - No error logging or notification happens
  - User will not find out unless they know where to look for EFI version
  - Since EFI is an unknown system to users and many Mac admins it goes ignored
  - Result: EFI is often out of date
QA Failure?
Incorrect EFI Firmwares Released
Are You Getting the EFI Updates You Should?

- Apple sometimes has QA failures
  - We identified EFI version regression issues with OS X 10.10 & 10.11
  - As part of the Security Update 2017-001 (Mar 27, 2017)
  - For unexplained reasons included EFI bundles were:
    - Older than preceding SecUpd 2016-003
    - Same as prior SecUpd 2016-002
  - Example:
    - MBP112
      - SecUpd 2016-002 = MBP112.0138.B17
      - SecUpd 2016-003 = MBP112.0138.B18
      - SecUpd 2017-001 = MBP112.0138.B17
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IM121</td>
<td>0047 23B</td>
<td>0047 25B</td>
<td>0047 23B</td>
</tr>
<tr>
<td>IM31</td>
<td>010A B09</td>
<td>010A B0A</td>
<td>010A B09</td>
</tr>
<tr>
<td>IM41</td>
<td>011B B13</td>
<td>011B B14</td>
<td>011B B13</td>
</tr>
<tr>
<td>IM42</td>
<td>011B B13</td>
<td>011B B14</td>
<td>011B B13</td>
</tr>
<tr>
<td>IM43</td>
<td>011B B13</td>
<td>011B B14</td>
<td>011B B13</td>
</tr>
<tr>
<td>IM44</td>
<td>017B B13</td>
<td>017B B14</td>
<td>017B B13</td>
</tr>
<tr>
<td>IM51</td>
<td>0207 B06</td>
<td>0207 B08</td>
<td>0207 B06</td>
</tr>
<tr>
<td>IM61</td>
<td>0207 B03</td>
<td>0207 B04</td>
<td>0207 B03</td>
</tr>
<tr>
<td>MBA1</td>
<td>0164 B14</td>
<td>0164 B19</td>
<td>0164 B14</td>
</tr>
<tr>
<td>MBA2</td>
<td>0154 B05</td>
<td>0154 B09</td>
<td>0154 B05</td>
</tr>
<tr>
<td>MBA41</td>
<td>077 B14</td>
<td>077 B15</td>
<td>077 B14</td>
</tr>
<tr>
<td>MBA51</td>
<td>0099 B22</td>
<td>0099 B23</td>
<td>0099 B22</td>
</tr>
<tr>
<td>MBA61</td>
<td>0166 B13</td>
<td>0166 B12</td>
<td>0166 B13</td>
</tr>
<tr>
<td>MBA71</td>
<td>0047 2CB</td>
<td>0047 2DB</td>
<td>0047 2CB</td>
</tr>
<tr>
<td>MBA91</td>
<td>00D3 B0D</td>
<td>00D3 B0E</td>
<td>00D3 B0D</td>
</tr>
<tr>
<td>MBP101</td>
<td>00E3 B0A</td>
<td>00E3 B0E</td>
<td>00E3 B0A</td>
</tr>
<tr>
<td>MBP102</td>
<td>0106 B0A</td>
<td>0106 B08</td>
<td>0106 B0A</td>
</tr>
<tr>
<td>MBP112</td>
<td>0138 B17</td>
<td>0138 B18</td>
<td>0138 B17</td>
</tr>
<tr>
<td>MM51</td>
<td>0077 B14</td>
<td>0077 B15</td>
<td>0077 B14</td>
</tr>
<tr>
<td>MM61</td>
<td>0166 B0A</td>
<td>0106 B08</td>
<td>0106 B0A</td>
</tr>
<tr>
<td>MM71</td>
<td>0220 B07</td>
<td>0220 B08</td>
<td>0220 B07</td>
</tr>
<tr>
<td>MP61</td>
<td>0116 B17</td>
<td>0116 B21</td>
<td>0116 B17</td>
</tr>
</tbody>
</table>
Can You Downgrade EFI?

- Firmware updates can not be downgraded
  - In our testing we were unable to force older versions
  - The `efiupdater` binary only allows higher versions
    - There is a force flag but this just forces `setup`, not `flash`
  - If the version of the target EFI bundle is lower it exits
  - Unable to spoof by altering EFI bundle version due to signing
  - Unable to spoof by altering `efiupdater` due to signing
  - This made the EFI updates in SecUpd 2017-001 for 10.10/.11 just stale code
Mitigations
What Can We Do About It?

- Upgrade to macOS 10.13
- Logging and reporting
- Run updates out of band
- Apple is addressing some things
Update to macOS 10.13 High Sierra

- macOS 10.13 includes EFI updates for supported models
  - APFS requires EFI support - thus EFI updates
  - Includes Mac Pro tower models (MacPro5,1)
    - This model requires a manual update, special case in Install Assistant
  - Some older models saw their first-ever update since 2015
    - iMac (Late 2009) aka iMac10,1
    - MacBook (Late 2009) aka MacBook6,1
    - Mac Pro (Mid-2010) aka MacPro5,1
  - *Apple actually started shipping these EFI updates in 10.12.5*
What’s new with macOS 10.13 High Sierra?

- macOS 10.13 High Sierra contains new tools
  - Apple is focusing on EFI **integrity** with eficheck
  - No signs of actively updating out of date EFI versions (yet)
  - Combined standalone tool and daemon (runs once a week)
    - `/usr/libexec/firmwarecheckers/eficheck/eficheck`
  - Gets checksum of current EFI firmware
  - Compares to Apple-shipped and code signed whitelist
  - Apple ships partial whitelist with 10.13
  - Downloads full whitelist if needed
  - If checksum not found in whitelist: alert user
  - ...the UX is not great yet □

---

Xeno Kovah @XenoKovah
So I hear macOS 10.13 comes out soon. Let’s talk about what’s up if you ever see this prompt [thread] https://pbs.twimg.com/media/1KcdB_AUMAAARrkB.jpg

Twitter: Yesterday at 1:49 PM (28kB)
Logging and Reporting

- Since the OS does not log we must do it ourselves
  - Use endpoint reporting tools
    - **osquery**: `/usr/local/bin/osqueryi “select version from platform_info”`
    - **Puppet**: `/usr/local/bin/facter system_profiler.boot_rom_version`
    - **Chef**: `/opt/chef/bin/ohai hardware/boot_rom_version`
    - **Shell script**:
      
      ```
      /usr/sbin/system_profiler SPHardwareDataType | awk '/ROM/{print $4}'
      ```
  - Once we have data we can move on to fixing divergent endpoints:
    - Re-install current OS or Security update to re-apply EFI
    - Use **efiupdater** as shown in paper to kick off EFI update
  - [https://github.com/trailofbits/osquery-pr/tree/alessandro/feature/macos.efigy_support](https://github.com/trailofbits/osquery-pr/tree/alessandro/feature/macos.efigy_support)
Apply Updates Out of Band

- Don’t wait for next OS/Security Update
  - Re-apply the update
  - Requires a restart in any case
  - If re-applying OS update: use Combo updater
    - Larger but considered best practice
    - Less chance of failures
  - Security Update is always Delta
  - Other option: create standalone installer for firmware updates only
    - Requires some custom work
    - Will always require a reboot, properly set user expectations
  - [https://github.com/grahamgilbert/imagr/wiki/High-Sierra-Notes#firmware](https://github.com/grahamgilbert/imagr/wiki/High-Sierra-Notes#firmware)
Tool and API Releases
EFIgy - API and Tools to Help Visibility

- As we discussed visibility to the state of your EFI and its security is hard
- EFIgy is free RESTful API and open source client that gives you access to the data that we built up during our research
  - Identifies if you are running the most up-to-date EFI version for your Board-ID and OS Build combination
  - Highlight areas of security concern we have spoken about today such as Mac models that are not receiving EFI updates
  - CVE’s that a particular EFI version may be vulnerable to
- Supports OS versions 10.10 through 10.13
EFIgy API Information:
API Version: 0.2
Updated On: Oct 13 2017, 17:42

-------------------------------------------------------

Endpoint: 127.0.0.1
# Enumerated system information (This data will be sent to the API in order to determine

Hashed SysUUID: 44c3cfc6f15da575636ebb88a7878d7c88c54dabdb60ffadcb8d7c02845955710
Hardware Version: MacBookPro13,2
EFI Version: MBP132.0226.B25
SMC Version: 2.37f20
Board-ID: Mac-66E35819EE2D0D05
OS Version: 10.12.6
Build Number: 16G29

[?] Do you want to continue and submit this request? [Y/N] y

# Results:

EFI firmware version check:
[+] SUCCESS - The EFI Firmware you are running (None) is the expected version

Highest build number check:
[+] SUCCESS - You are running the latest build number (16G29) of the OS version

Up-to-date OS check:
[+] SUCCESS - You are running the latest major/minor/micro version of the OS

-------------------------------------------------------
EFIgy

GUI

App

github.com/duo-labs/EFIgy-GUI
EFIgy

Check your EFI version

EFI Version Number

Mac Model ID

Build Number

Go

© Duo Labs 2017
Made with ❤ by Duo Labs

NOTE: macOS 10.13 is NOT currently supported. Our dataset currently only covers 10.10, 10.11, and 10.12.
EFIgy API data

Up to date API data and graphs here
Demo
Time
Conclusions
Conclusions

- Transparency is key - vendors should be clear with what they patch
- Systems can be ‘software secure but firmware vulnerable’
- Not all hardware may be treated equally
- QA is hard! Looking into the firmware being received is a good idea
- EFI is like a full OS in many ways:
  - It affects everything running above it (ring -2 remember!)
  - You should keep it up to date to not undermine the rest of your security
  - It needs to have notifications and alerts for updates like software does
- Apple is taking EFI security seriously and is continuing to lead the way
Lots more information in our technical paper

http://duo.sc/2x1AA9R

And Blogpost

http://duo.sc/2ychJhh

EFigy Tools
https://efigy.io
Thanks!

If you’ve got Q’s, we’ve got A’s…*

Pepijn Bruienne (@bruienne)
Rich Smith (@iodboi)

* We hope!