Advanced x86:
BIOS and System Management Mode Internals
Conclusion

Xeno Kovah && Corey Kallenberg

LegbaCore, LLC

All materials are licensed under a Creative Commons “Share Alike” license.

http://creativecommons.org/licenses/by-sa/3.0/

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

Attribution condition: You must indicate that derivative work
"Is derived from John Butterworth & Xeno Kovah’s ’Advanced Intel x86: BIOS and SMM’ class posted at http://opensecuritytraining.info/IntroBIOS.html”

What did we learn?

- Chipsec architecture (and it's evolution)
- PCI & the importance of port & memory-mapped IO to system configuration
- SMM & it's protections/vulnerabilities
- SPI flash & its protections/vulnerabilities
- UEFI, SecureBoot, firmware integrity checking
- "Any sufficiently advanced {attack} is indistinguishable from {black} magic"

- Doesn't seem like a lot does it? :)
- Gotta go *deep*, not broad, to find the land of *the deep dark*

What do I want you to know?
Everyone

- *Any child can break things*
- *The true measure of skill* is being able to *make* things that neither child nor adult can break
- Those who value civilization only break things when it can make defenses stronger

What do I want you to know? Consultants/Everyone

- Make sure your customers know that they almost certainly have a ton of vulnerable BIOSes.
- Make sure they know that they can sometimes fix vulnerabilities through BIOS updates
  - More true for the top 3, Lenovo, HP, Dell than it is for all the other asian PC-makers :-/
- And if there aren't BIOS updates available for their machines that make them protected, if they're a large purchaser from a particular vendor they can go push that vendor to secure their systems
  - And we can help, because we have a lot of experience talking to OEMs, explaining this stuff to them so they understand how to properly lock a BIOS

- And just in general I want you to know this sort of stuff so you can show off and make it clear you know stuff other people don't know when you start your first job :)

What do I want you to know? Forensics

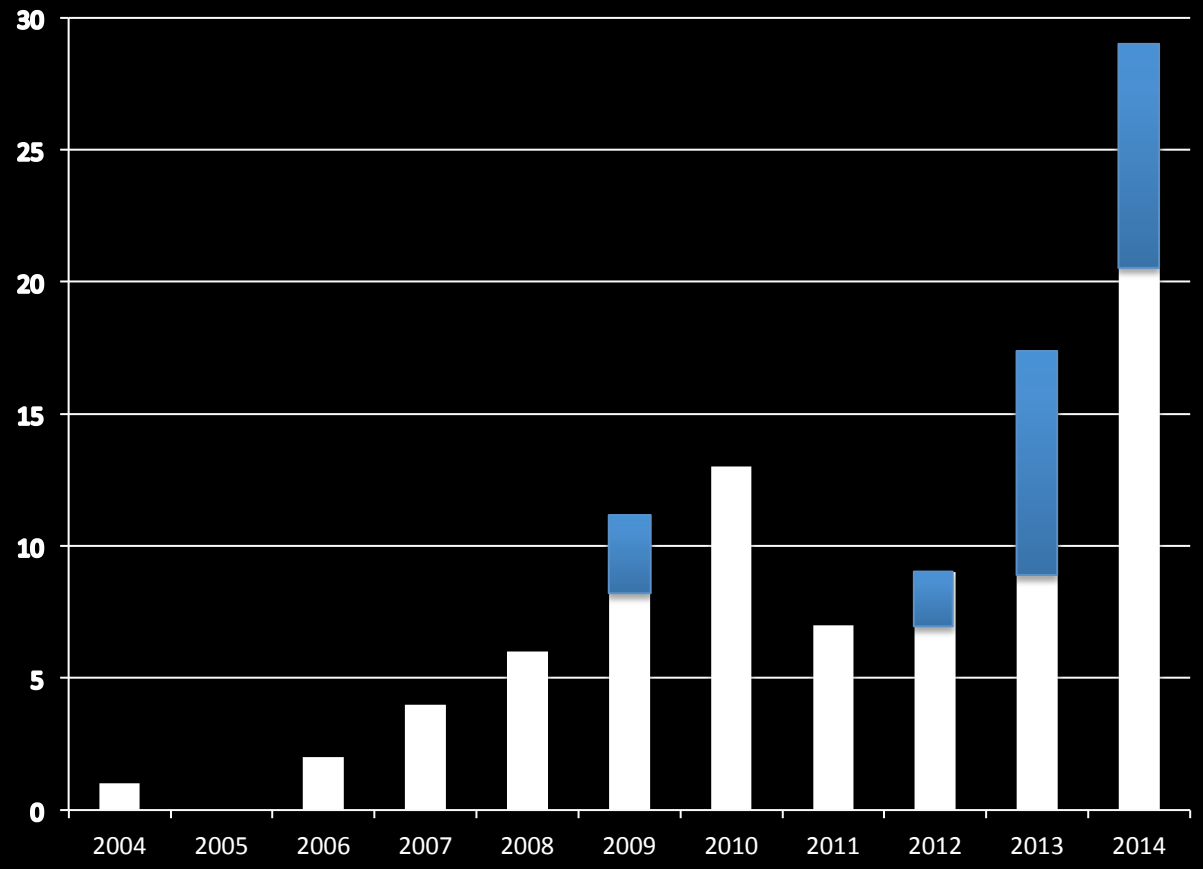
- How to integrity check a BIOS, so you can determine not just if an attacker *could* infect it, but if an attacker *did* infect it
- Know how to reverse engineer a BIOS, so that if you ever find a BIOS with a suspicious change
- Limits of the tools' trustworthiness. There's no real "forensically sound" way to capture BIOS contents other than physical SPI readers, and there's no way to capture SMM contents other than to have an exploit to break in and see what's there at runtime :-/
- If you only ever do forensics on disk/memory, you will never find sophisticated adversaries who hide in SMM/BIOS.
- "Firmware forensics" is going to be the next evolution in forensics. (it went "disk" -> "memory" and should now go -> firmware.) But very few people know about firmware, and firmware is especially variable on embedded systems. So you're positioned ahead of the curve, which means you could to research and make tools for others and become well-known within the forensics community.

What do I want you to know?
Product Security/Penetration Testing

- Know how to evaluate BIOS security directly, and know what all the security bits mean, and how they fail
- If you're pentesting a company, run Copernicus on every box you get on. Make it clear to the customer that a real attacker could have *bricked* all the unlocked computers (maybe demo it on one box with their permission using the manual flash writing we learned about on the start of day 4)
- And again, pentester reports need to include mitigation, which should be BIOS updates. So you need to let them know that they (and we) can push their vendor to improve their security posture for their BIOS/SMM

I've only talked about a small sampling of attacks (mostly ones that we found)

But I've given you the knowledge to go out and understand the rest



Mad skillz: go get some!

- There's a whole lot more material at OpenSecurityTraining.info for you to learn (because clearly some of you didn't know it yet :P)
- If you already know a topic that's there (like x86 assembly) try to go out and become a teacher for it.
  - There's no better way to refine your skills than teaching them to other people (because you need to know how to answer, or lookup, any random question that a student thinks up :))
- If you know a topic that's not already there, contribute it!
  - Needs to be a full day's worth of class material, which I judge to be >= 6 hours
  - Doesn't need to be in English

Reminder:
Optional overall homework for the class

- Find an integer overflow in the EDK2 code: https://tianocore.github.io/edk2/
- We will have Corey evaluate it, and if he thinks it's exploitable we'll handle the disclosure and give you co-authorship for the disclosure, and give you the option to present with us if we are able to successfully turn it into a conference talk
- To pick your starting point for analysis, you'll want to think about what kind of information a BIOS might be processing that an attacker could potentially control
  - E.g. images, USB messages, firmware updates, PE executables, UEFI firmware filesystem, FAT filesystem, network packets, etc

I hope you have fun exploring and doing voodoo in the Deep Dark!

xeno@legbacore.com