Micro-Renovator

Bringing Processor Firmware up to Code

Bio

Matt (a.k.a. Syncsrc)

- Recovering EE / CompE
- Builder and user of hardware debug features
- Uses “BIOS” and “UEFI” interchangeably
- Currently responsible for platform security of a cloud
- Religiously updates firmware
- Formerly a product security validation lead at Intel

Background

Meltdown

Meltdown breaks the most fundamental isolation between user applications and the operating system. This attack allows a program to access the memory, and thus also the secrets, of other programs and

Spectre

Spectre breaks the isolation b
different applications. It allow
attacker to trick error-free pro
which follow best practices, ir
leaking their secrets. In fact, t
checks of said best practices

the INQUIRER

tificial Intelligence | Internet of Things | Open Source | Hardware | Software | Security

tmaster hack | iOS 12 | Galaxy Note 9 | iPhone 9 | World Cup

Intel, ARM and AMD all affected by security-bypassing, kernel-bothering CPU bugs

Fixes exist but it looks like fundamental processor designs are borked

ars TECHNICA

BIZ & IT | TECH | SCIENCE | POLICY | CARS | GAMING & CULTURE

TECH —

“Meltdown” and “Spectre”: Every modern processor has unfixable security flaws

Immediate concern is for Intel chips, but everyone is at risk.

PETER BRIGHT - 1/3/2018, 4:30 PM

Background - Patches for Spectre & Meltdown

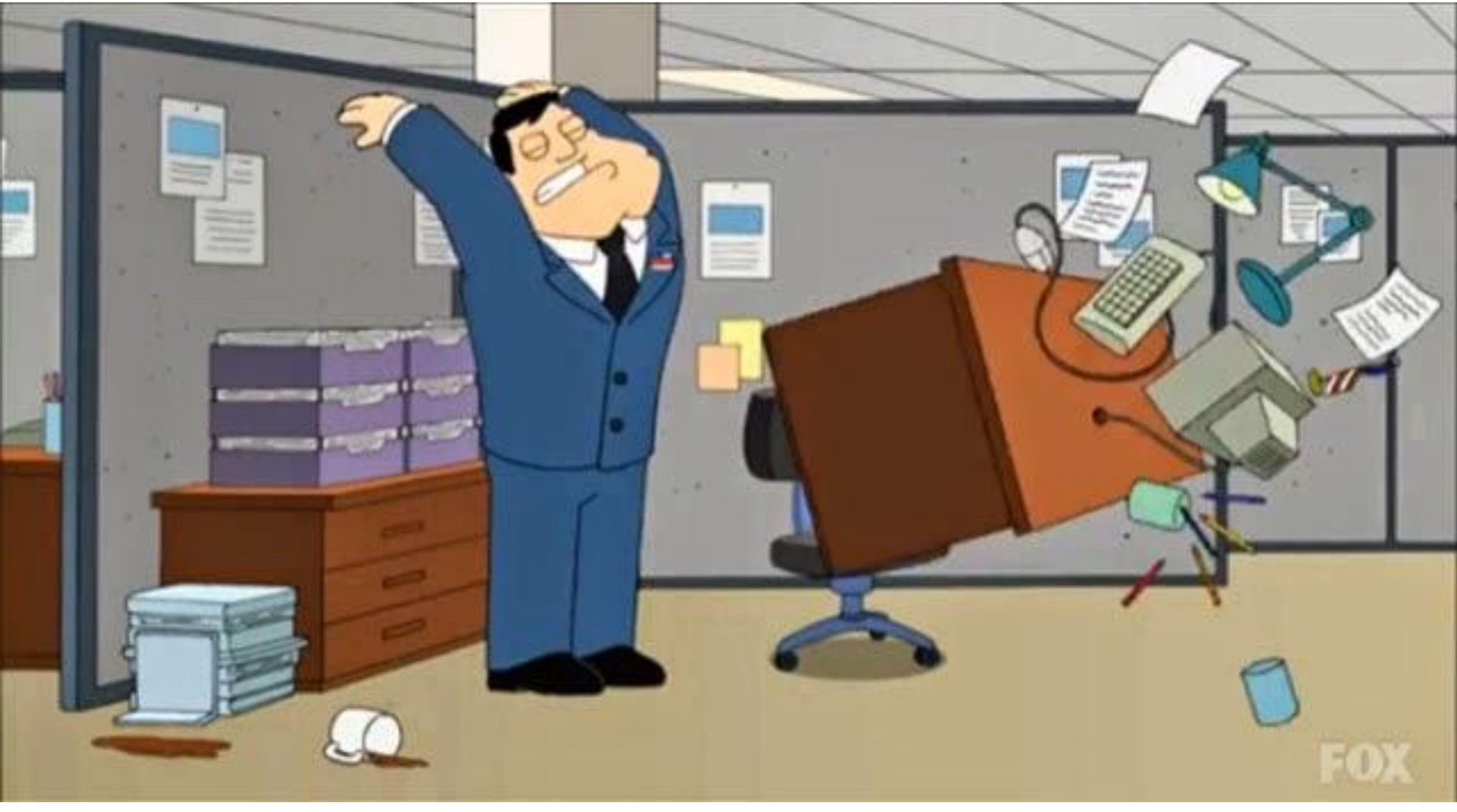
- Meltdown
  - PTI (Linux), KVA Shadow (Windows)
- Spectre v1
  - Lfence (forced serialization)
- Spectre v2
  - Retpoline (“pollutes” BTB) - *An incomplete fix, per Intel*
  - IBRS & IBPB (new MSRs to control BTB)
  - *Microcode updates are necessary to expose these new MSRs*

What is Microcode?

- Can be thought of as “Processor Firmware”
  - A Brief History of Microprogramming
- Can be patched to fix bugs & errata
  - However, processors don’t have any non-volatile storage
  - Any applied patches are lost on reset or power-down
- Stored and applied by the *BIOS* or *Operating System*
  - Reloaded on every boot, reset, S3 resume, etc...
  - Intel 64 and IA-32 Architectures SDM Volume 3A, section 9.11.6

Obtaining Microcode Patches

- From BIOS Updates
  - MacBook and Surface users are covered
- From Operating System Updates
  - Linux: redistributed by most distros (via microcode_ctl.rpm or intel-microcode.deb)
    - Users can also download microcode.dat directly from Intel
  - Microsoft didn't start distributing microcode until 2 months after it was released
    - Still only distributing for some processor and Windows versions
- *Patches existed, but were impossible to apply to many systems*


FOX

Systems Unable to Patch Spectre

- Still no way to mitigate Spectre v2 on millions of systems
  - Other than buying a new OS (or a new computer)
- Windows PCs that are 3 to 9 years old *
  - Mostly shipped with Windows 7 or 8 pre-installed
  - BIOS updates delayed or unavailable
- 3rd-party microcode update drivers are ineffective

* CPUs more than 9 years old are not receiving microcode updates from Intel

Systems Unable to Patch Spectre

CPU	BIOS Updates	Windows 7 & 8	Windows 10
8th Gen	Available	N/A	None
7th Gen	Available	N/A	None
6th Gen	Available	Negligible	None
5th Gen	Available *	49 Million	None
4th Gen	Available *	114 Million	None
3rd Gen	Xeon Only	141 Million	None
2nd Gen	Xeon Only	149 Million	None
1st Gen	Xeon Only	216 Million	216 Million

Are there any other options?

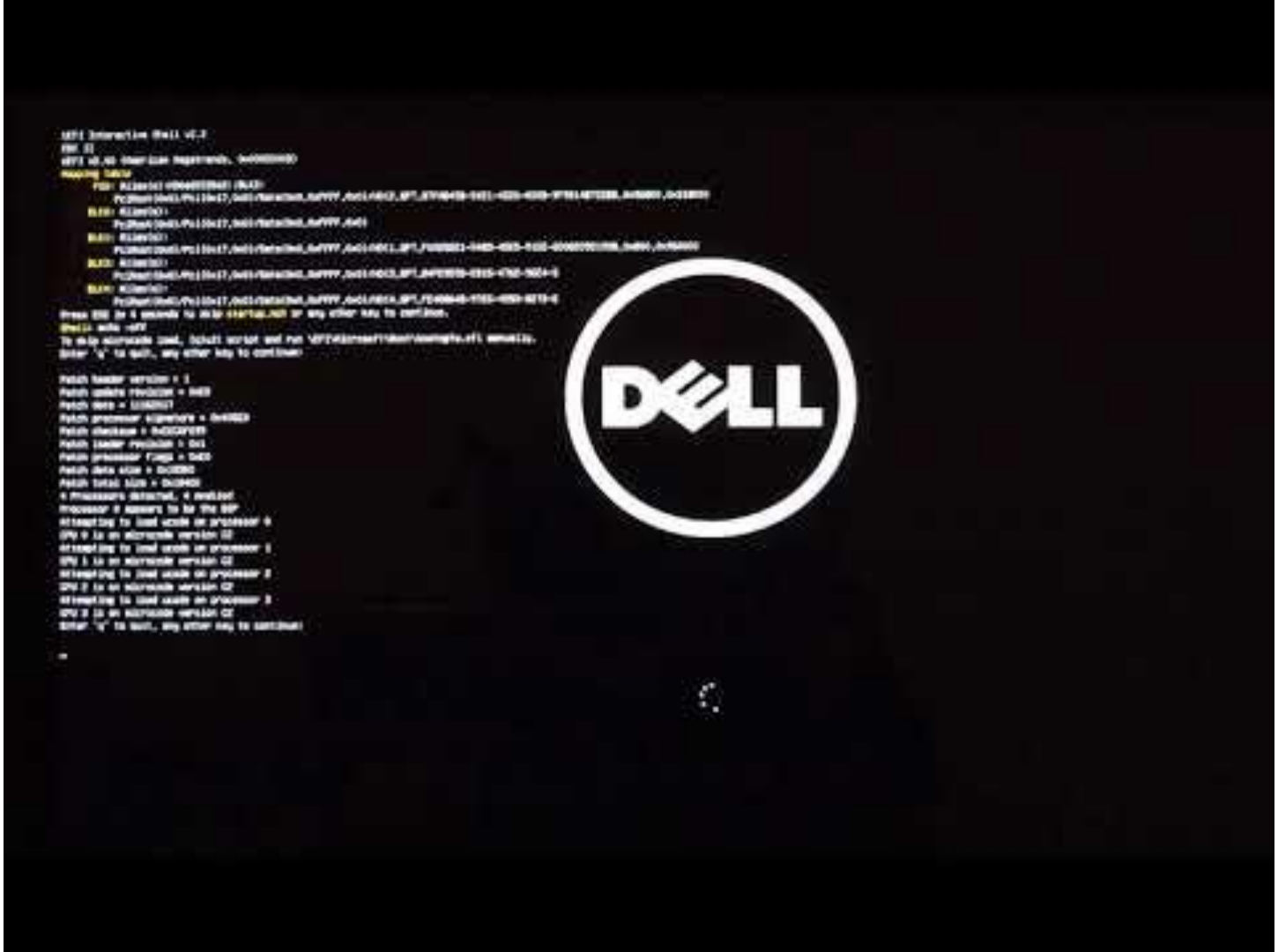
- When can microcode patches be applied?
  - BIOS - users can’t modify
  - OS - microcode drivers run too late
  - Bootloader - *maybe?*
- No existing EFI utility to load microcode
  - TianoCore is open source though
  - And already has code that applies microcode updates
  - How hard could it be?

Uload.efi

- Built using EDK2
- Mostly code appropriated from MicrocodeUpdateDxe
  - Made into an EFI shell app
- Loads microcode to all Processors/Threads

Inserting Uload into EFI Boot

- POST: Power On Stuff That-happens
- Determine bootloader from NVRAM variables
- Locate boot drive & partition
  - Involves UUIDs somehow
- Run Bootloader (eg: bootx64.efi, shim.efi)
  - Bootloader is just an EFI application, it can be replaced
- Bootloader launches kernel


DELL

Micro-Renovator

- Script to automatically update EFI boot partition
  - Runs from a Linux Live CD
- Finds EFI partition and bootloader
  - Copies microcode and Uload.efi to the boot partition
  - Installs Shell.efi and sets as the primary boot option
  - On boot, startup script runs Uload prior to the OS bootloader

https://github.com/syncsrc/MicroRenovator

Limitations

- Breaks Sleep (S3)
  - Hibernation still works
- No secure boot support (yet)
- Occasional inconsistent behavior after booting into Windows
- Microsoft appears to be actively reverting the changes made by earlier versions of MicroRenovator

Summary

- Firmware patching is an unsolved problem
  - UEFI should have made things better, but didn’t
- Component vendors needs to focus on enabling patching for end-users
  - Not system builders (they aren’t incentivized)
  - IoT and Mobile spaces have the same issue
- *It shouldn’t take bootloader hacks to apply security patches to operating systems that are still under support*

Questions?