Thierry Laurion, Founder & CEO

INSURGO

TECHNOLOGIES LIBRES / OPEN TECHNOLOGIES

Today's presentation

Heads OEM device ownership/re-ownership : A tamper evident approach to remote integrity attestation

Current status and future plan : A call for collaboration

What is ***Insurgo PrivacyBeast X230***?

- **QubesOS Preinstalled**: a reasonably secure OS
- **Heads**: Slightly more secure, open source, reproducible, firmware
  - X230 i7 2.9Ghz Gen3 cpu (Ivy bridge) 16GB ram, 1366x768 IPS screen, 256GB SSD
  - **Neutralized+Deactivated Intel ME of 98kb**
  - Binary blob free coreboot hardware initialization
  - **Heads** as coreboots’ linux payload
  - Re-ownership needed tools
  - User friendly re-ownership Wizard

What is Heads?

Heads goal is to produce **reproducible**, **measurable** ROMs : **https://github.com/osresearch/heads**

“Since the **x86 firmware contains the initial instructions the CPU executes** when it comes out of reset, it is important that it be protected against malicious modification.”

The concern is that i**f the boot firmware is compromised or buggy, then the rest of the system security is built on an unstable foundation.**” (33c3 Heads’ presentation)

What is Heads?

“Heads is a configuration for laptops and servers that tries to bring more security to commodity hardware.”

- Use free software on the boot path (linux payload and associated initrd packed with required tools)

- Move the root of trust to coreboot’s ROM bootblock (measured boot from there)

- Remote attestation of firmware’s state (TPM/HOTP)

- Verified boot integrity: verifies auto-generated digest of /boot files against user’s signed digest and enforces change notification/approval (sign) prior of booting (GPG2)

What Heads does?

Heads accomplishes measured boot of components into the following TPM's PCRs prior to their usage:

0: Boot block

1: ROM stage

2: RAM stage and MRC

3: Heads Linux kernel and initrd

4: Boot mode (0 during /init, then recovery or normal-boot)

5: Heads Linux kernel modules

6: Drive LUKS headers

7: Heads user-specific config files

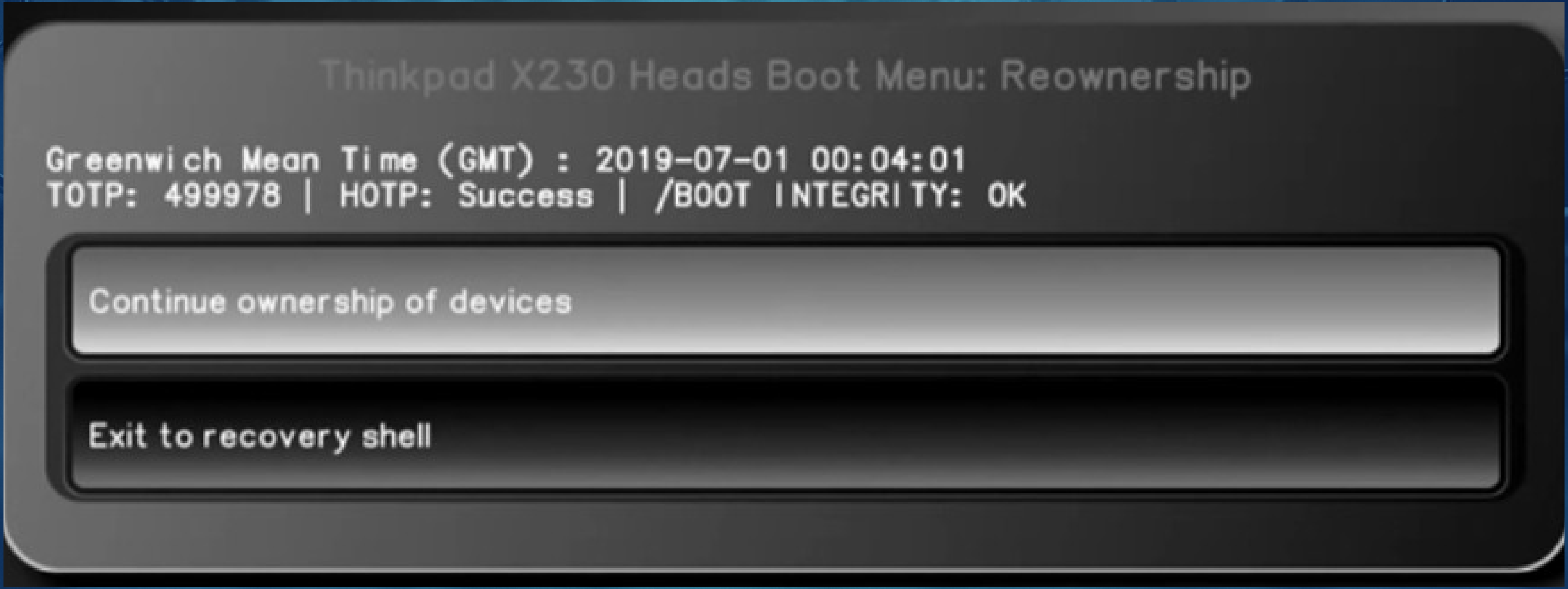
Why Ownership/Re-ownership?

- **Firmware/boot integrity need to be attested/ verified: prior to shipment, at user’s reception and then at each boot by the user.**
- **USB Security dongle used to seal measurements should be owned by OEM with randomized PINs so that interception doesn’t permit resealing of tampered firmware integrity in transit.**
- **USB Security dongle should be provisioned/usable by user once firmware integrity is verified.**

Why Ownership/Re-ownership?

- ~~Firmware/boot integrity need to be attested/verified: prior to shipment, at user's reception and then at each boot by the user.~~
- ~~USB Security dongle used to seal measurements should be owned by OEM with randomized PINs so that interception doesn't permit resealing of tampered firmware integrity in transit.~~
- ~~USB Security dongle should be provisioned/usable by user once firmware integrity is verified.~~

Why Ownership/Re-ownership?



Thinkpad X230 Heads Boot Menu: Reownership

```
Greenwich Mean Time (GMT) : 2019-07-01 00:04:01
TOTP: 499978 | HOTP: Success | /BOOT INTEGRITY: OK
```

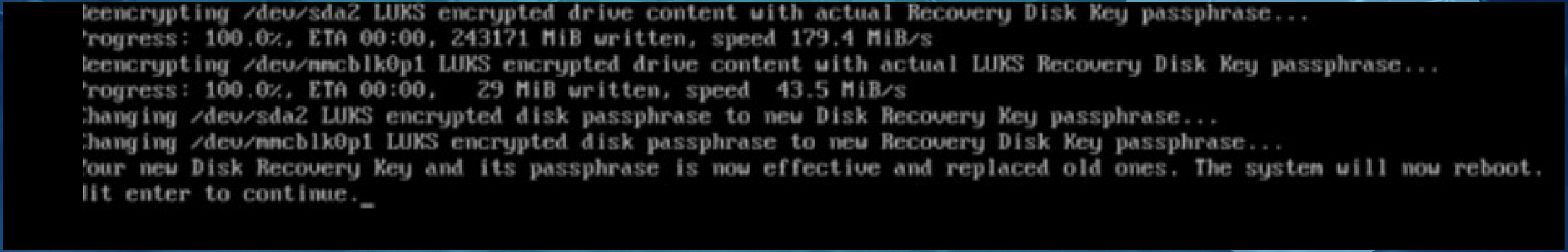
Why Ownership/Re-ownership?

- ***Preinstalled OS needs to be protected while shipped to the user:***
  - ***Initial LUKS encryption key should be unique***
  - ***Initial LUKS encryption passphrase should not be communicated to user prior to reception (interception protection).***
  - ***Final LUKS encryption key and passphrases need to be unknown from the OEM/Organization.***
  - ***Integrity of OS installation needs to be verified.***

Why Ownership/Re-ownership?

- ***Preinstalled OS needs to be protected while shipped to the user:***
  - ***~~Initial LUKS encryption key should be unique~~***
  - ***~~Initial LUKS encryption passphrase should not be communicated to user prior to reception (interception protection).~~***
  - ***~~Final LUKS encryption key and passphrases need to be unknown from the OEM/Organization.~~***
  - ***Integrity of OS installation needs to be verified.***

Why Ownership/Re-ownership?



```
Reencrypting /dev/sda2 LUKS encrypted drive content with actual Recovery Disk Key passphrase...
Progress: 100.0%, ETA 00:00, 243171 MiB written, speed 179.4 MiB/s
Reencrypting /dev/mmcblk0p1 LUKS encrypted drive content with actual LUKS Recovery Disk Key passphrase...
Progress: 100.0%, ETA 00:00,   29 MiB written, speed  43.5 MiB/s
Changing /dev/sda2 LUKS encrypted disk passphrase to new Disk Recovery Key passphrase...
Changing /dev/mmcblk0p1 LUKS encrypted disk passphrase to new Recovery Disk Key passphrase...
Your new Disk Recovery Key and its passphrase is now effective and replaced old ones. The system will now reboot.
Hit enter to continue._
```

Reownership Wizard

Activate OEM to User Re-Ownership (OEM)

User Re-Ownership Wizard (User reception)

What is Qubes OS ?

- **Security by compartmentalization** in virtualized security domains (qubes)
- Network attack surface reduction through default ingress traffic blocking (internal routing only vulnerabilities)
- **Network leak prevention** (tor) through Whonix-Gateway
- Required explicit device assignment; else confined
- E-mail attachments opened in disposable qubes
- Read only OS templates instantiated by application qubes

Under development:

- Other reasonably-secure models are currently getting VBOOT+measured boot support under coreboot 4.11+/Heads*
  - T530
  - T430
  - X230
  - T420
  - X220

  Thanks to the NlNet grant and 9Elements.

*Blob free native init. Including Neutered+Deactivated ME, expended IFD BIOS region to host more useful tools.

Soon to be developed:

Under obtained NlNet grant work:

- **3mdeb: Fwupd support under QubesOS**
- **QubesOS/Whonix: Secured, on-demand QubesOS remote administration**
- **QubesOS: safer anonymizing/forensic resistant defaults**
- **Insurgo: International keyboard keymaps support**

Needed:

- Better Heads reproducibility safeguards…
- Wider Heads(!) collaboration and involvement…
- User’s freedom respecting platform (not x86!)
- QubesOS support of alternative x86 platform…

- …More developers!
- International distributed reprogrammers
- International partners!

The future could be brighter

- **QubesOS would benefit from community involvement in supporting PPC64 to have reasonably secure OS over truly Open Source Firmware TCB sitting on top of Open Source Hardware.**
- **PowerPC coreboot support**
- **In-Heads GPG keypair generation exported on sdcard's encrypted LUKS partition, with subkeys moved into USB Security dongle at reownership (no more "I lost my USB security dongle, now what?" problem)**

Insurgo Inc!

- International reprogrammers needed!
  - Fair reprogramming fee per unit!
  - Direct hardware sourcing to partners!
  - Provided training!

!!!Poke me at FOSDEM!!!

Contact us!

Insurgo Initiative

- OpenCollective’s fund will directly be fed by ***Insurgo & partners*** sales profits (**25% of net profit donated to open source needed R&D!**)
- Open source project issues/features requiring R&D will get **direct funding upon approval and validated proof of work!**

Contact us!

Thank you!

Questions???

What is Heads?

“Since we're able to bring up the TPM and **establish our static hardware root of trust first thing in the romstage prior to initializing the memory controllers and while running out of cache, this reduces our exposure to certain types of external device attacks**. **There are still concerns about the EC and ME, which we'll address a bit later.**” (33c3 Heads’ presentation)

What Heads does?

- Heads permits to boot multiboot systems

- Heads permits to boot signed ISOs from external media (Fedora, Tails and QubesOS ISOs when accompanied with distribution signature files), validated by public distro signing keys present in the ROM. (Tails, Fedora, QubesOS)

What Heads does?

The user seals those measurements into the TPM, on which a QR code to be scanned into an OTP application is displayed on screen. The user can then:

- Validate manually on his phone at each boot that the TOTP (2FA) numbers shown matches his smartphone’s

- Use a Librem Key/Nitrokey Pro/ Nitrokey Storage to seal that original secret through HOTP. This way, the HOTP challenge result is both shown visually on screen (OK/INVALID) and through the led on the key flashing green (or red otherwise).

What Heads does?

Additionally:

- The boot files are hashed on the fly at each boot and verified against user signed digest kept in /boot/kexec.sig. If they changed, user is asked if he is the origin of the changes (dom0 updates applied on last boot?) and shows found mismatches.

- Heads also permits take advantage of the TPM to enforce a Disk Unlock Key released if provided by the right passphrase when PCRs measurements matches). **This prevents eavesdropped typed passphrase to decrypt cloned disk content.**

Heads Ownership/Reownership wizard?

- Integration into Heads
- GPG2 (4096 bits keypair generation) on smartcards
- cryptsetup-reencrypt to reencrypt cloned QubesOS installation image and sdcard partition used to store provisioning secrets. Passphrase is shared with customers prior/upon hardware reception.
- Librem Key/NitroKey Pro v2/NitroKey Storage v2 support for OEMs to provide visual tamper evident integrity attestation (HOTP: Purism/Nitrokey partnership with Heads)

How?

- Whiptail (bash based GUI) for accessibility (Purism)
- Diceware integration for passphrase generation of ownership secrets on Reownership Wizard (Used by the OEM and the user)
- flashrom to backup/flash internally from measured, trusted firmware state
- cbfs used under Heads to insert public key and configs (flammit)

What is Qubes OS ?

“**Qubes OS is a security-oriented operating system (OS)**. The OS is the software that runs all the other programs on a computer. Some examples of popular OSes are Microsoft Windows, Mac OS X, Android, and iOS. **Qubes is free and open-source software (FOSS)**. This means that everyone is free to use, copy, and change the software in any way. It also means that the source code is openly available so others can contribute to and audit it.” (Qubes OS)

What is Qubes OS ?

“Most people use an operating system like **Windows or OS X** on their desktop and laptop computers. These OSes are **popular because they tend to be easy to use and usually come pre-installed** on the computers people buy.” (Qubes OS)

What is Qubes OS ?

“However, **they present problems when it comes to security**. For example, you might **open an innocent-looking email attachment or website, not realizing that you’re actually allowing malware (malicious software) to run on your computer**. Depending on what kind of malware it is, it **might do anything from showing you unwanted advertisements to logging your keystrokes to taking over your entire computer.**” (Qubes OS)

What is Qubes OS ?

“This could **jeopardize all the information stored on or accessed by this computer, such as health records, confidential communications, or thoughts written in a private journal.** Malware can also interfere with the activities you perform with your computer. For example, **if you use your computer to conduct financial transactions, the malware might allow its creator to make fraudulent transactions in your name.”** (Qubes OS)

What is Qubes OS ?

Aren’t antivirus programs and firewalls enough?

“…Antivirus and traffic inspection technologies are… **limited to a detection-based approach.** New **zero-day vulnerabilities are constantly being discovered in the common software we all use, such as our web browsers, and no antivirus program or firewall can prevent all of these vulnerabilities from being exploited.**” (Qubes OS)

What is Qubes OS ?

“This approach allows you to **keep the different things you do on your computer securely separated from each other** in isolated qubes so that one qube getting compromised won’t affect the others. For example, **you might have one qube for visiting untrusted websites and a different qube for doing online banking**. This way, if your **untrusted browsing qube gets compromised by a malware-laden website, your online banking activities won’t be at risk**.” (Qubes OS)

What is Qubes OS ?

Similarly, if you’re concerned about malicious email attachments, Qubes can make it so that every attachment gets opened in its own single-use disposable qube. In this way, **Qubes allows you to do everything on the same physical computer without having to worry about a single successful cyberattack taking down your entire digital life in one fell swoop.”** (QubesOS)

What is Qubes OS ?

“Moreover, **all of these isolated qubes are integrated into a single, usable system**.

Programs are isolated in their own separate qubes, but all windows are displayed in a single, unified desktop environment with unforgeable colored window borders so that you can easily identify windows from different security levels.” (Qubes OS)

What is Qubes OS ?

”Common attack vectors like network cards and USB controllers are isolated in their own hardware qubes while their functionality is preserved through secure networking, firewalls, and USB device management.

Integrated file and clipboard copy and paste operations make it easy to work across various qubes without compromising security.” (Qubes OS)

What is Qubes OS ?

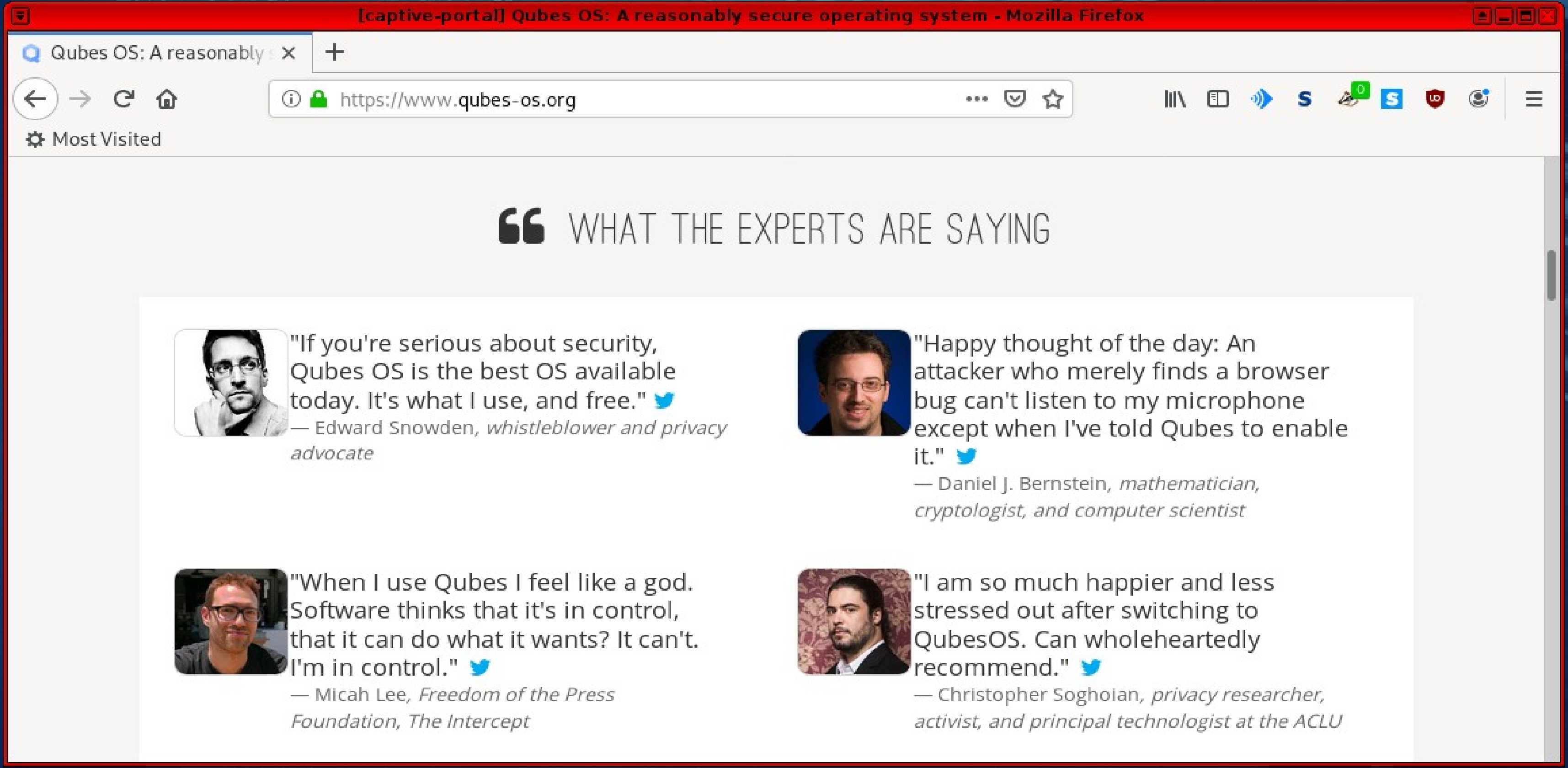
“The innovative **Template system separates software installation from software use, allowing qubes to share a root filesystem without sacrificing security (and saving disk space, to boot)**.” (QubesOS)

What is Qubes OS ?

“Qubes even allows you to sanitize PDFs and images in a few clicks.

Users concerned about privacy will appreciate the integration of Whonix with Qubes, which makes it easy to use Tor securely, **while those concerned about physical hardware attacks will benefit from Anti Evil Maid**.” (Qubes OS)

What is Qubes OS ?



[captive-portal] Qubes OS: A reasonably secure operating system - Mozilla Firefox
Qubes OS: A reasonably
https://www.qubes-os.org
Most Visited

WHAT THE EXPERTS ARE SAYING

"If you're serious about security, Qubes OS is the best OS available today. It's what I use, and free."
— Edward Snowden, *whistleblower and privacy advocate*

"Happy thought of the day: An attacker who merely finds a browser bug can't listen to my microphone except when I've told Qubes to enable it."
— Daniel J. Bernstein, *mathematician, cryptologist, and computer scientist*

"When I use Qubes I feel like a god. Software thinks that it's in control, that it can do what it wants? It can't. I'm in control."
— Micah Lee, *Freedom of the Press Foundation, The Intercept*

"I am so much happier and less stressed out after switching to QubesOS. Can wholeheartedly recommend."
— Christopher Soghoian, *privacy researcher, activist, and principal technologist at the ACLU*

Qubes OS Requirements

Minimum

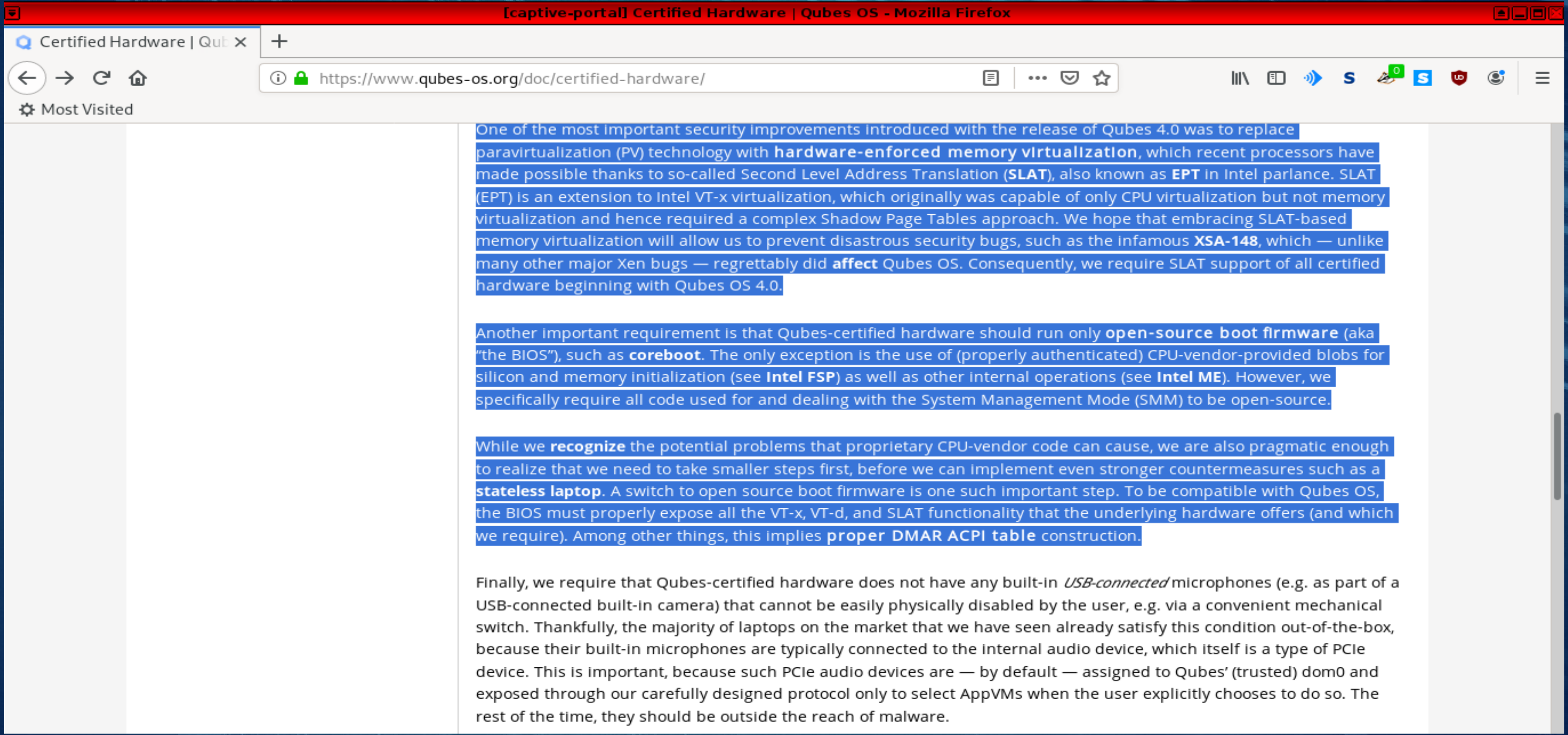
- **64-bit Intel or AMD processor (x86_64 aka x64 aka AMD64)**
- **Intel VT-x with EPT or AMD-V with RVI**
- **Intel VT-d or AMD-Vi (aka AMD IOMMU)**
- **4 GB RAM**
- **32 GB disk space**

Qubes OS Requirements

Recommended:

- **Fast SSD** (strongly recommended)
- **Intel IGP** (strongly preferred)
- Nvidia GPUs may require significant troubleshooting.
- AMD GPUs have not been formally tested, but Radeons (RX580 and earlier) generally work well
- **TPM with proper BIOS support** (required for Anti Evil Maid)
- A non-USB keyboard or multiple USB controllers

QubesOS Hardware Certification Requirements


[captive-portal] Certified Hardware | Qubes OS - Mozilla Firefox
Certified Hardware | Qub
https://www.qubes-os.org/doc/certified-hardware/
Most Visited

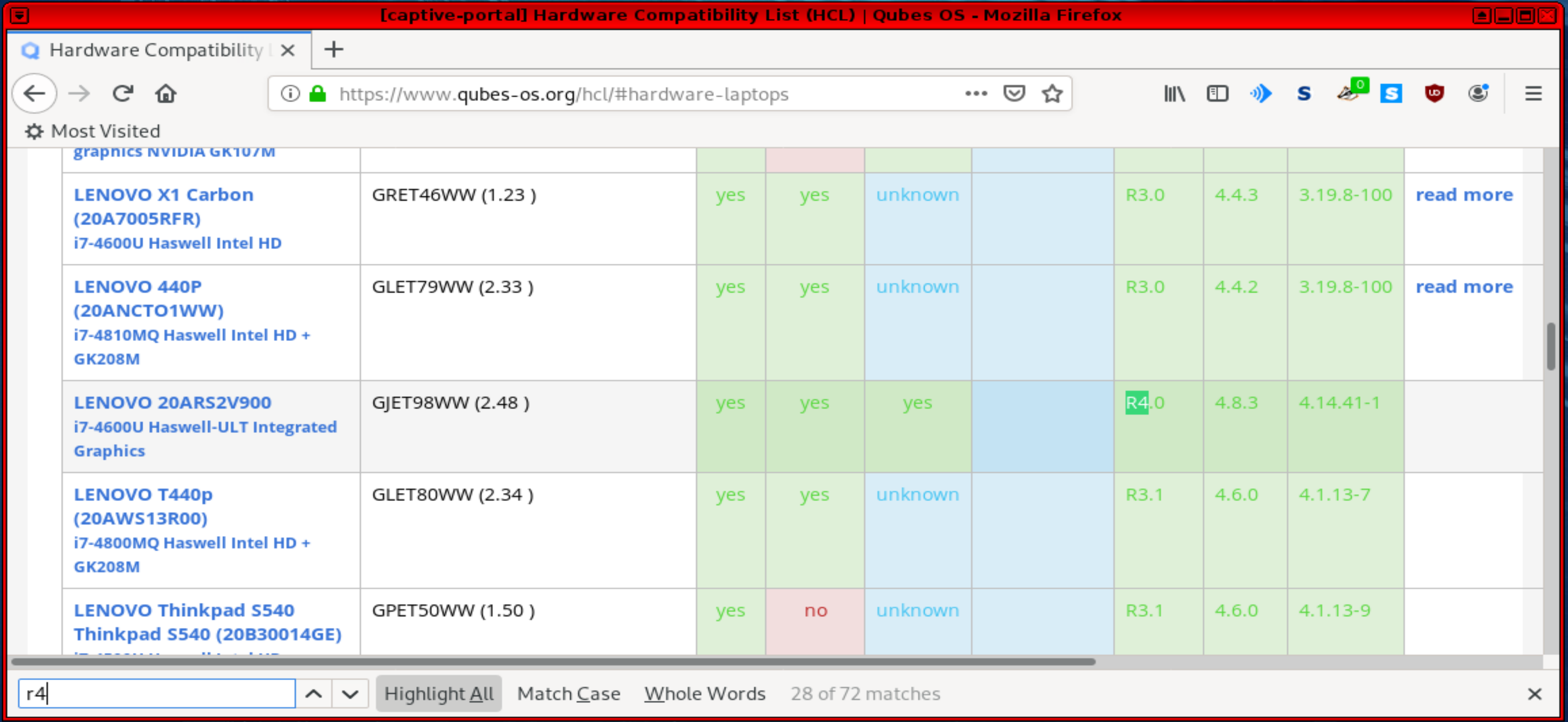
One of the most important security improvements introduced with the release of Qubes 4.0 was to replace paravirtualization (PV) technology with **hardware-enforced memory virtualization**, which recent processors have made possible thanks to so-called Second Level Address Translation (**SLAT**), also known as **EPT** in Intel parlance. SLAT (EPT) is an extension to Intel VT-x virtualization, which originally was capable of only CPU virtualization but not memory virtualization and hence required a complex Shadow Page Tables approach. We hope that embracing SLAT-based memory virtualization will allow us to prevent disastrous security bugs, such as the infamous **XSA-148**, which — unlike many other major Xen bugs — regrettably did **affect** Qubes OS. Consequently, we require SLAT support of all certified hardware beginning with Qubes OS 4.0.

Another important requirement is that Qubes-certified hardware should run only **open-source boot firmware** (aka "the BIOS"), such as **coreboot**. The only exception is the use of (properly authenticated) CPU-vendor-provided blobs for silicon and memory initialization (see **Intel FSP**) as well as other internal operations (see **Intel ME**). However, we specifically require all code used for and dealing with the System Management Mode (SMM) to be open-source.

While we **recognize** the potential problems that proprietary CPU-vendor code can cause, we are also pragmatic enough to realize that we need to take smaller steps first, before we can implement even stronger countermeasures such as a **stateless laptop**. A switch to open source boot firmware is one such important step. To be compatible with Qubes OS, the BIOS must properly expose all the VT-x, VT-d, and SLAT functionality that the underlying hardware offers (and which we require). Among other things, this implies **proper DMAR ACPI table** construction.

Finally, we require that Qubes-certified hardware does not have any built-in *USB-connected* microphones (e.g. as part of a USB-connected built-in camera) that cannot be easily physically disabled by the user, e.g. via a convenient mechanical switch. Thankfully, the majority of laptops on the market that we have seen already satisfy this condition out-of-the-box, because their built-in microphones are typically connected to the internal audio device, which itself is a type of PCIe device. This is important, because such PCIe audio devices are — by default — assigned to Qubes' (trusted) dom0 and exposed through our carefully designed protocol only to select AppVMs when the user explicitly chooses to do so. The rest of the time, they should be outside the reach of malware.

QubesOS Hardware Compatibility List


[captive-portal] Hardware Compatibility List (HCL) | Qubes OS - Mozilla Firefox
Hardware Compatibility L
https://www.qubes-os.org/hcl/#hardware-laptops
Most Visited
r4
Highlight All
Match Case
Whole Words
28 of 72 matches

graphics NVIDIA GK107M									
LENOVO X1 Carbon (20A7005RFR) i7-4600U Haswell Intel HD	GRET46WW (1.23)	yes	yes	unknown		R3.0	4.4.3	3.19.8-100	read more
LENOVO 440P (20ANCTO1WW) i7-4810MQ Haswell Intel HD + GK208M	GLET79WW (2.33)	yes	yes	unknown		R3.0	4.4.2	3.19.8-100	read more
LENOVO 20ARS2V900 i7-4600U Haswell-ULT Integrated Graphics	GJET98WW (2.48)	yes	yes	yes		R4.0	4.8.3	4.14.41-1	
LENOVO T440p (20AWS13R00) i7-4800MQ Haswell Intel HD + GK208M	GLET80WW (2.34)	yes	yes	unknown		R3.1	4.6.0	4.1.13-7	
LENOVO Thinkpad S540 Thinkpad S540 (20B30014GE)	GPET50WW (1.50)	yes	no	unknown		R3.1	4.6.0	4.1.13-9	

QubesOS Hardware Compatibility List

So, the HCL:

- promotes mostly 3.2 compatible hardware
- doesn’t expose firmware open source level (Auditable trustworthiness, presence of binary blobs)
- Doesn’t specify Intel ME maximal disablement level

User: “I’m not technical enough **to even select** the right hardware myself!!!”

Deploying Qubes OS on slightly more secure hardware

"Most people use an operating system like Windows or OS X on their desktop and laptop computers. **These OSes are popular because they tend to be easy to use and usually come pre-installed on the computers people buy.**"

So… ***What if QubesOS could be securely preinstalled on slightly more secured hardware?*** (Accomplishing 2015 promises!!!)

Intel Firmware Support Package (FSP)

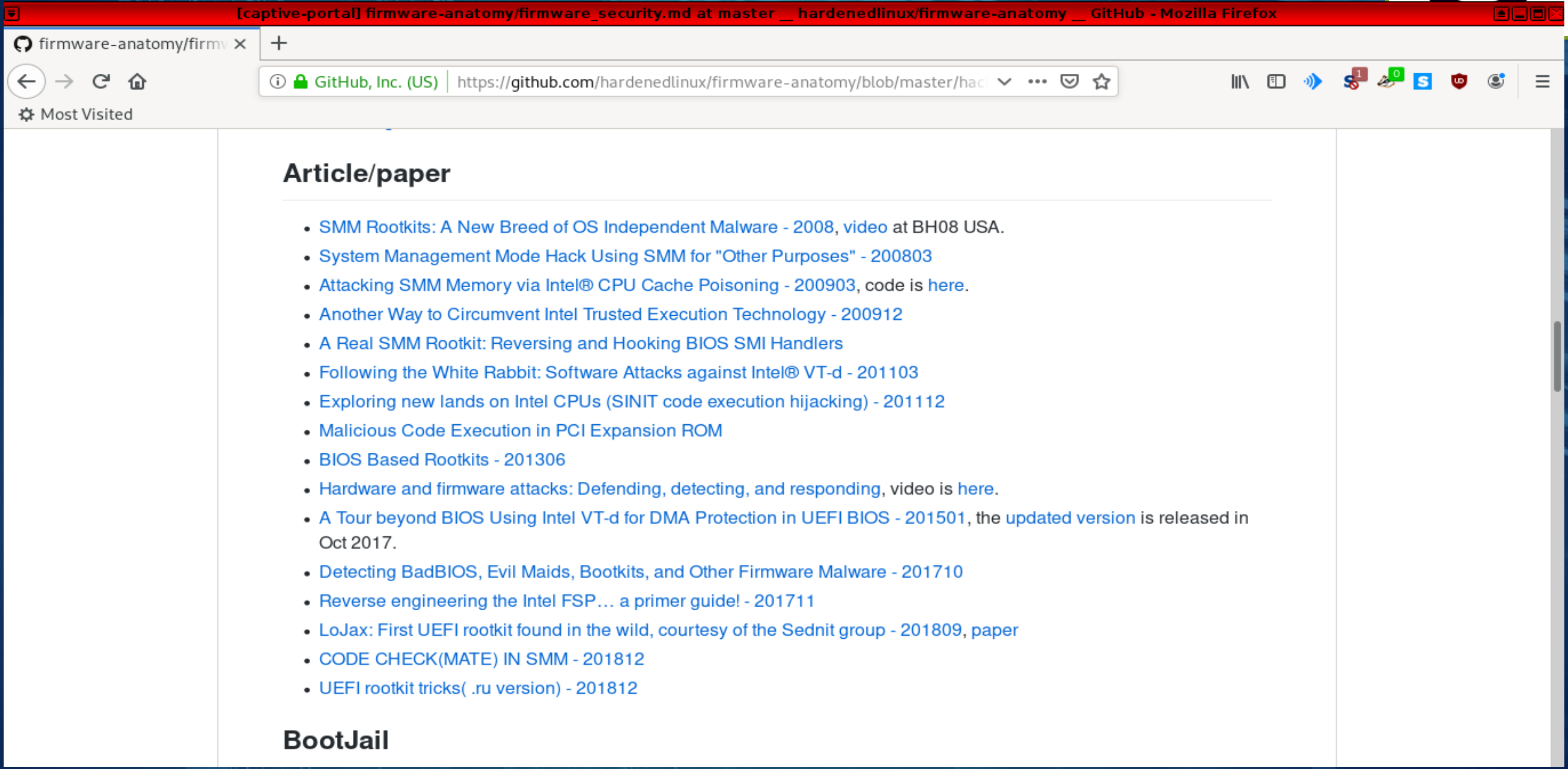
- "On all recent Intel systems, coreboot support has revolved around integrating a blob (for each system) called **the FSP** (firmware support package), which **handles all of the hardware initialization, including memory and CPU initialization**. Reverse engineering and replacing this blob is almost impossible, due to how complex it is. Even for the most skilled developer, it would take years to replace. **Intel distributes this blob to firmware developers, without source."** (libreboot)

Intel Firmware Support Package (FSP)

- “Since the FSP is responsible for the early hardware initialization, that means **it also handles SMM (System Management Mode)**. **This is a special mode that operates below the operating system level.** It’s possible that rootkits could be implemented there, which could perform a number of attacks on the user (the list is endless).”

- “In fact, **several SMM rootkits have been demonstrated in the wild**.” (**libreboot**)

Firmware vulnerabilities?



[captive-portal] firmware-anatomy/firmware_security.md at master _ hardenedlinux/firmware-anatomy _ GitHub - Mozilla Firefox
firmware-anatomy/firmw
GitHub, Inc. (US)
https://github.com/hardenedlinux/firmware-anatomy/blob/master/hac
Most Visited

Article/paper

- SMM Rootkits: A New Breed of OS Independent Malware - 2008, video at BH08 USA.
- System Management Mode Hack Using SMM for "Other Purposes" - 200803
- Attacking SMM Memory via Intel® CPU Cache Poisoning - 200903, code is here.
- Another Way to Circumvent Intel Trusted Execution Technology - 200912
- A Real SMM Rootkit: Reversing and Hooking BIOS SMI Handlers
- Following the White Rabbit: Software Attacks against Intel® VT-d - 201103
- Exploring new lands on Intel CPUs (SINIT code execution hijacking) - 201112
- Malicious Code Execution in PCI Expansion ROM
- BIOS Based Rootkits - 201306
- Hardware and firmware attacks: Defending, detecting, and responding, video is here.
- A Tour beyond BIOS Using Intel VT-d for DMA Protection in UEFI BIOS - 201501, the updated version is released in Oct 2017.
- Detecting BadBIOS, Evil Maids, Bootkits, and Other Firmware Malware - 201710
- Reverse engineering the Intel FSP… a primer guide! - 201711
- LoJax: First UEFI rootkit found in the wild, courtesy of the Sednit group - 201809, paper
- CODE CHECK(MATE) IN SMM - 201812
- UEFI rootkit tricks(.ru version) - 201812

BootJail

(**Hardened-Linux**)

Intel Management Engine (ME)

“In short, ME is a **separate processor embedded in the chipset** of any modern computer with an Intel CPU. **ME runs even when the computer is sleeping or powered off (as long as it is plugged in to a power outlet).** ME **can access any part of RAM, but the RAM region used by ME is not accessible from the OS. What’s more, ME is capable of out-of-band access to the network adapter.**” (Positive Technologies)

Intel Management Engine (ME)

“The ME firmware is compressed and consists of modules that are listed in the manifest along with secure cryptographic hashes of their contents. **One module is the operating system kernel**, which is based on a proprietary real-time operating system (RTOS) kernel called “ThreadX”. “ (libreboot)

Intel Management Engine (ME)

“Another module is the Dynamic Application Loader (DAL), which consists of a Java virtual machine and set of preinstalled Java classes for cryptography, secure storage, etc. The DAL module can load and execute additional ME modules from the PC’s HDD or SSD.“ (libreboot)

Intel Management Engine (ME)

“The ME firmware also includes a number of native application modules within its flash memory space, including Intel Active Management Technology (AMT), an implementation of a Trusted Platform Module (TPM), Intel Boot Guard, and audio and video DRM systems.“ (libreboot)

AMD Platform Security Processor (PSP)

- “This is basically AMD’s own version of the Intel Management Engine. It has all of the same basic security and freedom issues, although the implementation is wildly different.”

- “The PSP is an ARM core with TrustZone technology, built onto the main CPU die. As such, it has the ability to hide its own program code, scratch RAM, and any data it may have taken and stored from the lesser-privileged x86 system RAM (kernel encryption keys, login data, browsing history, keystrokes, who knows!).”

AMD Platform Security Processor (PSP)

- **“To make matters worse, the PSP theoretically has access to the entire system memory space, which means that it has at minimum MMIO-based access to the network controllers and any other PCI/PCIe peripherals installed on the system.” (libreboot)**

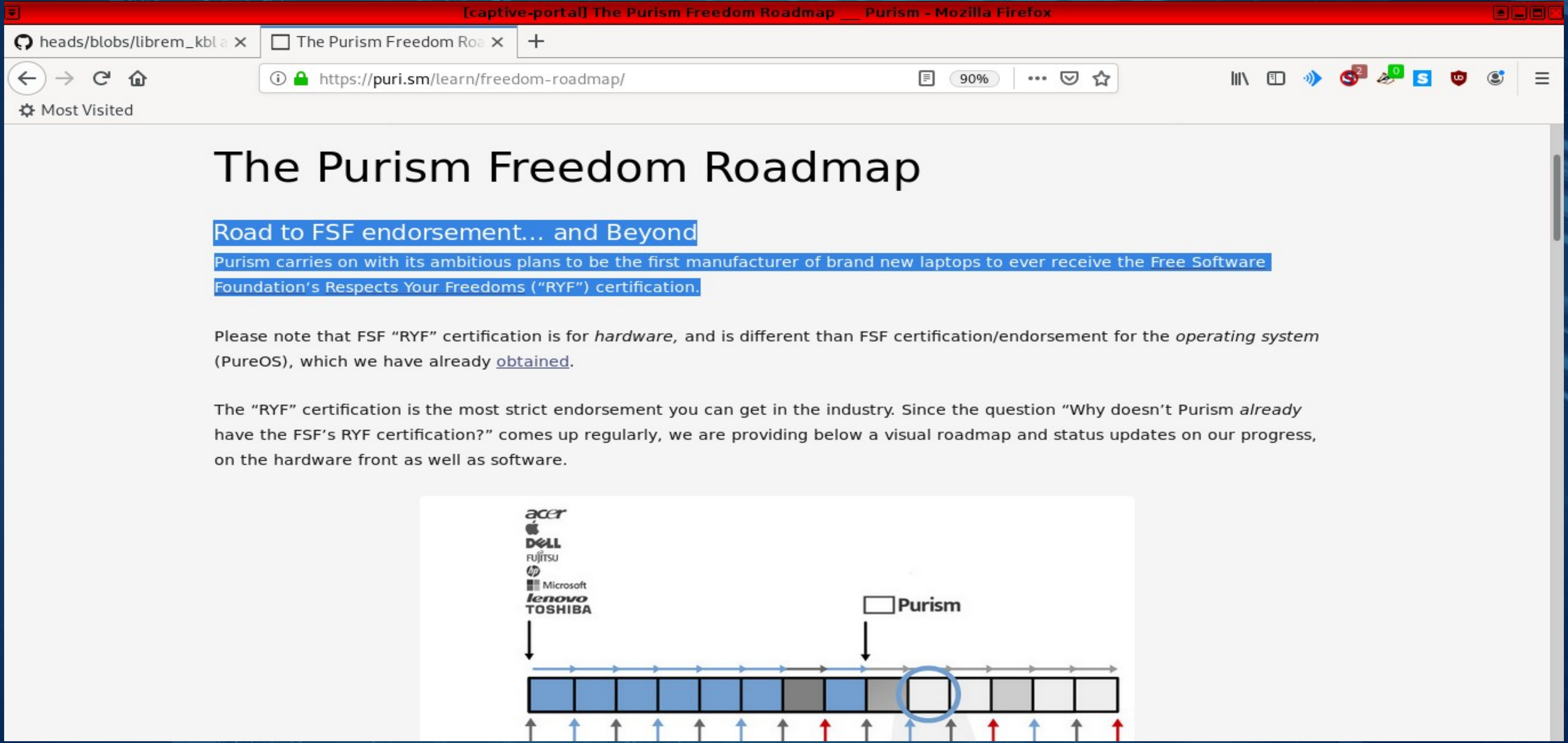
Current limited openness from the Open Source Firmware world

!!! DISCLAIMER !!!

This is not a Purism/System76 specific issue!

All recent x86 platforms require binary blobs to boot (Intel ME/FSP, PSP and others) !!!

OSF: FSP free and Intel ME neutralized?



[captive-portal] The Purism Freedom Roadmap __ Purism - Mozilla Firefox
heads/blobs/librem_kbl a
The Purism Freedom Roa
https://puri.sm/learn/freedom-roadmap/
90%
Most Visited
acer
DELL
FUJITSU
hp
Microsoft
lenovo
TOSHIBA
Purism

The Purism Freedom Roadmap

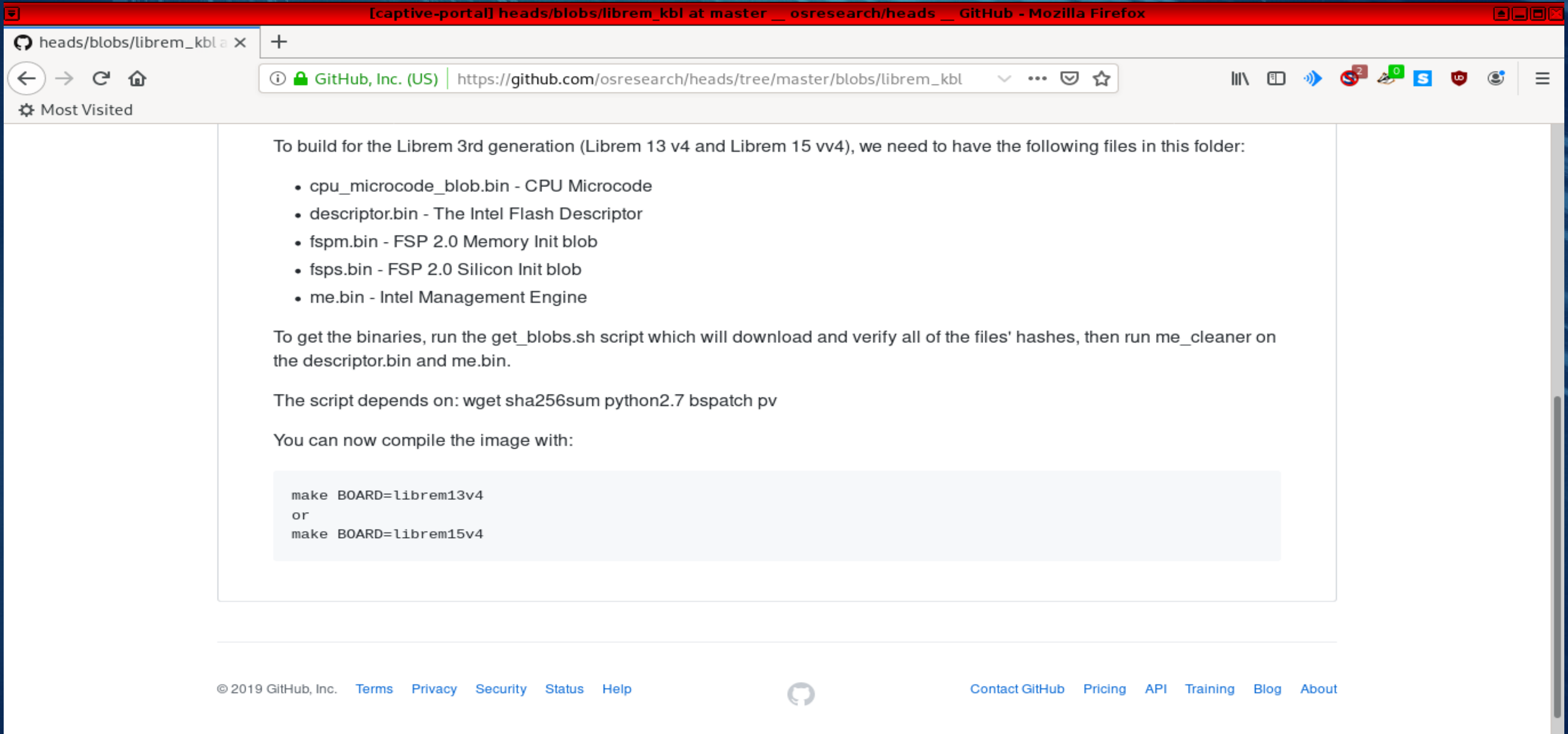
Road to FSF endorsement… and Beyond

Purism carries on with its ambitious plans to be the first manufacturer of brand new laptops to ever receive the Free Software Foundation's Respects Your Freedoms ("RYF") certification.

Please note that FSF "RYF" certification is for *hardware*, and is different than FSF certification/endorsement for the *operating system* (PureOS), which we have already obtained.

The "RYF" certification is the most strict endorsement you can get in the industry. Since the question "Why doesn't Purism *already* have the FSF's RYF certification?" comes up regularly, we are providing below a visual roadmap and status updates on our progress, on the hardware front as well as software.

Heads/Coreboot on the Librem 14 v4


[captive-portal] heads/blobs/librem_kbl at master _ osresearch/heads _ GitHub - Mozilla Firefox
heads/blobs/librem_kbl a
GitHub, Inc. (US)
https://github.com/osresearch/heads/tree/master/blobs/librem_kbl
Most Visited

To build for the Librem 3rd generation (Librem 13 v4 and Librem 15 vv4), we need to have the following files in this folder:

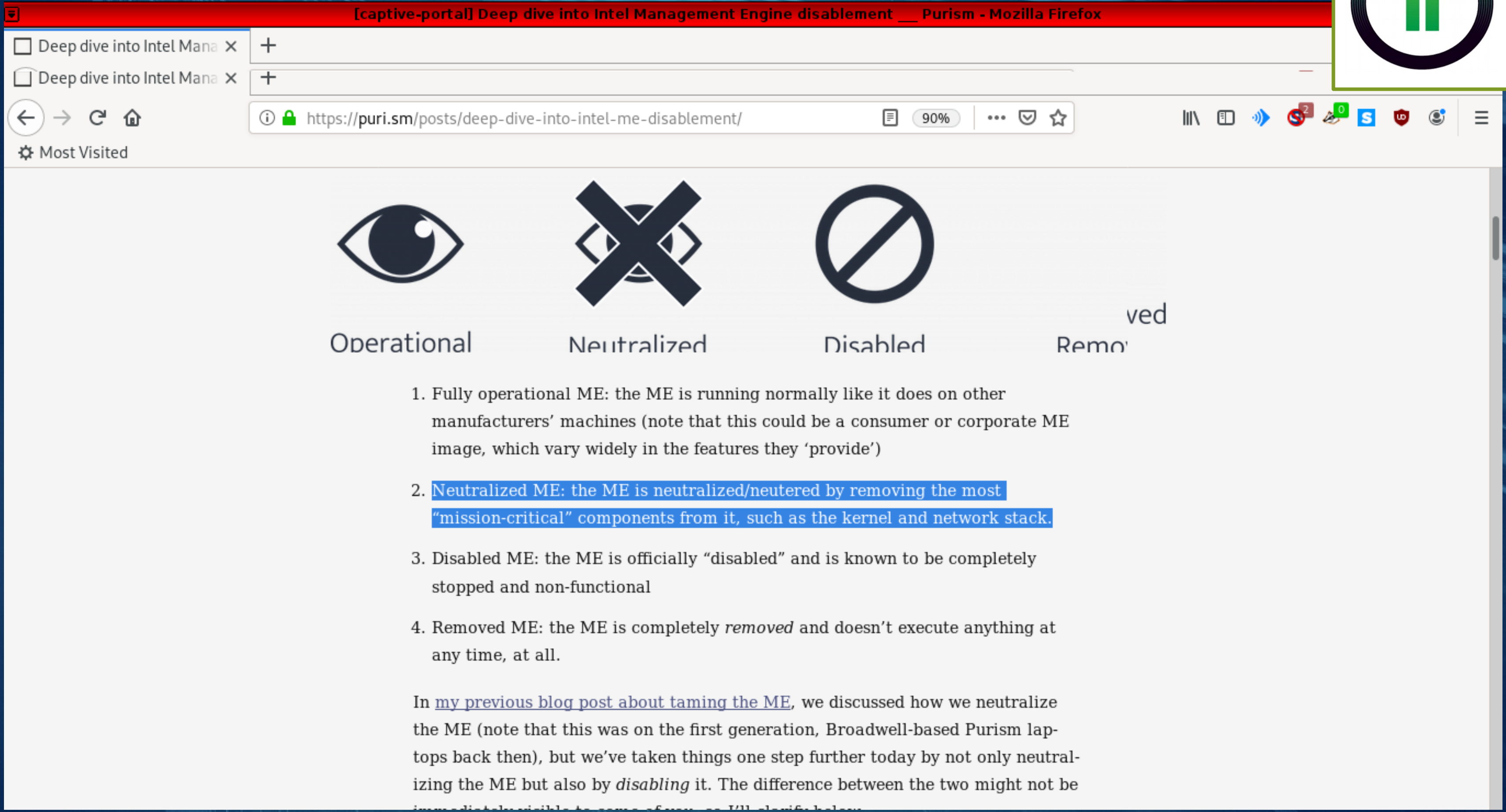
- cpu_microcode_blob.bin - CPU Microcode
- descriptor.bin - The Intel Flash Descriptor
- fspm.bin - FSP 2.0 Memory Init blob
- fsps.bin - FSP 2.0 Silicon Init blob
- me.bin - Intel Management Engine

To get the binaries, run the get_blobs.sh script which will download and verify all of the files' hashes, then run me_cleaner on the descriptor.bin and me.bin.

The script depends on: wget sha256sum python2.7 bspatch pv

You can now compile the image with:

```
make BOARD=librem13v4
or
make BOARD=librem15v4
```


[captive-portal] Deep dive into Intel Management Engine disablement __ Purism - Mozilla Firefox
Deep dive into Intel Mana
Deep dive into Intel Mana
https://puri.sm/posts/deep-dive-into-intel-me-disablement/
90%
Most Visited
Operational
Neutralized
Disabled
Removed

1. Fully operational ME: the ME is running normally like it does on other manufacturers' machines (note that this could be a consumer or corporate ME image, which vary widely in the features they 'provide')
2. Neutralized ME: the ME is neutralized/neutered by removing the most "mission-critical" components from it, such as the kernel and network stack.
3. Disabled ME: the ME is officially "disabled" and is known to be completely stopped and non-functional
4. Removed ME: the ME is completely *removed* and doesn't execute anything at any time, at all.

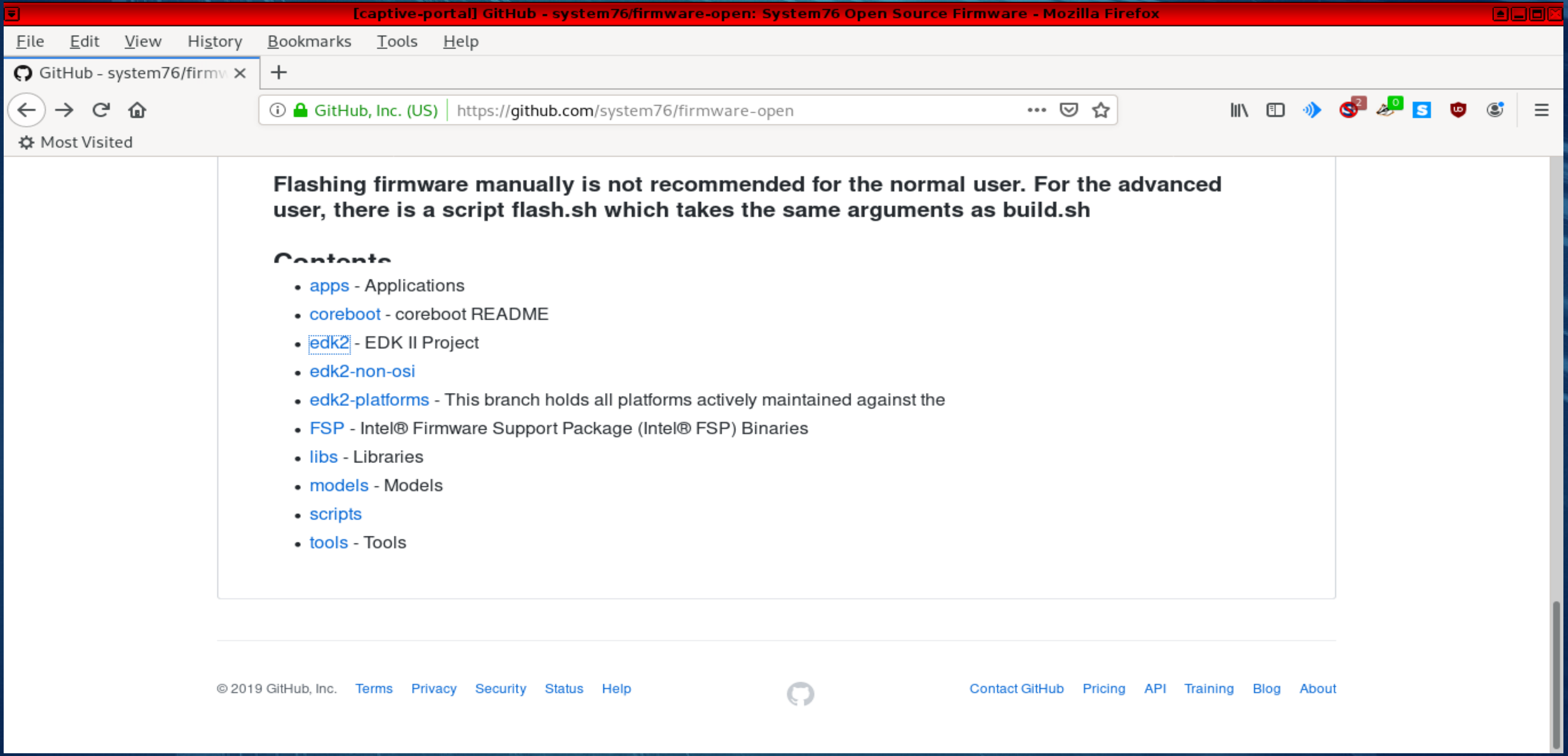
In my previous blog post about taming the ME, we discussed how we neutralize the ME (note that this was on the first generation, Broadwell-based Purism laptops back then), but we've taken things one step further today by not only neutralizing the ME but also by *disabling* it. The difference between the two might not be immediately visible to some of you, so I'll clarify below.

Intel ME on the Librem 14 v4

Me_cleaner application (excerpt)

```
ME/TXE firmware version 11.0.18.1002 (generation 3)
...
rbe        (Huffman    , 0x004a40 - 0x0070c0): NOT removed, essential
kernel     (Huffman    , 0x0070c0 - 0x015dc0): NOT removed, essential
syslib     (Huffman    , 0x015dc0 - 0x028a00): NOT removed, essential
bup        (Huffman    , 0x028a00 - 0x051600): NOT removed, essential
...
The ME minimum size should be 352256 bytes (0x56000 bytes)
Setting the HAP bit in PCHSTRP0 to disable Intel ME...
```

System76 Open Source Firmware



[captive-portal] GitHub - system76/firmware-open: System76 Open Source Firmware - Mozilla Firefox
File Edit View History Bookmarks Tools Help
GitHub - system76/firmw
GitHub, Inc. (US) https://github.com/system76/firmware-open
Most Visited

Flashing firmware manually is not recommended for the normal user. For the advanced user, there is a script flash.sh which takes the same arguments as build.sh

Contents

- apps - Applications
- coreboot - coreboot README
- edk2 - EDK II Project
- edk2-non-osi
- edk2-platforms - This branch holds all platforms actively maintained against the
- FSP - Intel® Firmware Support Package (Intel® FSP) Binaries
- libs - Libraries
- models - Models
- scripts
- tools - Tools

Terms Privacy Security Status Help
Contact GitHub Pricing API Training Blog About

History: Intel ME < 6

X200 : **GM45 bridge (Intel ME : deleted) Libreboot.**

- 0 bytes of binary blob firmware
- No FSP
- Native hardware initialization
- Respect Your Freedom (RYF) certified

But:

- No virtualization extension
- No vt-d (No interrupt remapping)

Intel ME 6 (Nehalem) <= 10 (Broadwell)

“The LZMA modules are placed after the Huffman data (after the LLUT) and their positions are clearly saved inside the manifests, so they can easily be removed.” (me_cleaner)

Ivy bridge (**Neutered**: ***no kernel nor library modules***) + AltMeDisable bit (**Disabled**)

- 98304 bytes = 98.304 KB
- Modules: BUP and ROMP (CPU BringUP)

Intel ME >= v11 (Skylake)

Kaby lake bridge (Neutralized + Deactivated.)

Should be said partly Neutralized. Kernel is still there!) + HAP bit

352256 bytes = 352.256 KB

Modules: Kernel, syslib, BUP and rbe(startup)

“...hashes of the **modules rbe, bup, kernel and syslib are checked together, increasing the number of the fundamental modules to four**.” (me_cleaner)

“...has a "HAP" bit which acts like a kill-switch, **telling Intel ME to hang after the initialization**.” (me_cleaner)

“~~***Neutralized ME: the ME is neutralized/neutered by removing the most “mission-critical” components from it, such as the kernel and network stack.***~~ (Purism)

What about the Talos II/BlackBird Power9 Systems?

- QubesOS doesn’t support PPC64le (**yet**!)
- QubesOS is more directed at laptops.
  - Spoiler alert! A laptop is on Raptor Engineering’s roadmap!
    - **Contact RaptorEngineering** to support **their Open Hardware R&D Laptop on Power10**!
    - Jump aboard on **PPC64 support of QubesOS**!

...Meanwhile...