FAQ

7-8 minutes

FAQ

Heads is an open source firmware, OS configuration and hardware hardening guide for building slightly more secure systems. Installing Heads requires opening the machine and extensive warranty voiding, so this document tries to answer some of the questions about it.

▾ Table of contents

Why replace UEFI with coreboot

While Intel's edk2 tree that is the base of UEFI firmware is open source, the firmware that vendors install on their machines is proprietary and closed source. Updates for bugs fixes or security vulnerabilities are at the vendor's convenience; user specific enhancements are likely not possible; and the code is not auditable.

UEFI is much more complex than the BIOS that it replaced. It consists of millions of lines of code and is an entire operating system, with network device drivers, graphics, USB, TCP, https, etc, etc, etc. All of these features represents increased "surface area" for attacks, as well as unnecessary complexity in the boot process.

coreboot is open source and focuses on just the code necessary to bring the system up from reset. This minimal code base has a much smaller surface area and is possible to audit. Additionally, self-help is possible if custom features are required or if a security vulnerability needs to be patched.

What's wrong with UEFI Secure Boot

Can't audit it, signing keys are controlled by vendors, doesn't handle hand off in all cases, depends on possible leaked keys.

Why use Linux instead of vboot2

vboot2 is part of the coreboot tree and is used by Google in the Chromebook system to provide boot time security by verifying the hashes on the coreboot payload. This works well for the specialized Chrome OS on the Chromebook, but is not as flexible as a measured boot solution. By moving the verification into the boot scripts we're able to have a much flexible verification system and use more common tools like PGP to sign firmware stages.

What about Trusted GRUB

The mainline grub doesn't have support for TPM and signed kernels, but there is a Trusted grub fork that does. Due to philosophical differences the code might not be merged into the mainline. And due to problems with secure boot (which Trusted Grub builds on), many distributions have signed insecure kernels that bypass all of the protections secure boot promised. Additionally, grub is closer to UEFI in that it must have device drivers for all the different boot devices, as well as filesystems. This duplicates the code that exists in the Linux kernel and has its own attack surface.

Using coreboot and Linux as a boot loader allows us to restrict the signature validation to keys that we control. We also have one code base for the device drivers in the Linux-as-a-boot-loader as well as Linux in the operating system.

What is the concern with the Intel Management Engine

- "Rootkit in your laptop" (PDF) by Igor Skochinsky of Hex-Rays, Breakpoint 2012 Melbourne
- "Intel ME Secrets" (PDF) by Igor Skochinsky of Hex-Rays, RECON 2014 Montreal
- "x86 considered harmful" (PDF) by Joanna Rutkowska, October 2015

How about the other embedded devices in the system

- "Hardening hardware and choosing a #goodBIOS" by Peter Stuge, 2013
- Funtenna uses software to make embedded devices broadcast data on radio frequencies by Ang Cui 2015

Should we be concerned about the binary blobs

Maybe. x230 has very few (MRC) since it has native vga init.

Why use ancient Thinkpads instead of modern Macbooks

The x230 Thinkpad has coreboot support, TPM, nice keyboards and are very cheap to experiment on. Newer Thinkpads contain Bootguard, a closed source security function implemented by Intel to prevent unsigned custom firmware, such as coreboot and heads, from being installed.

How likely are physical presence attacks vs remote software

attacks

Who knows.

Defense in depth vs single layers

Yes.

is it worth doing the hardware modifications

Depends on your threat model.

Should I validate the TPMTOTP on every boot

Probably. I want to make it also do it at S3. See Heads issue #69

suspend vs shutdown

S3 is subject to cold boot attacks, although they are harder to pull off on a Heads system since the boot devices are constrained.

However, without tpmtotp in s3 it is hard to know if the system is in a safe state when the xscreensaver lock screen comes up. Is it a fake to deceive you and steal your login password? Maybe! It wouldn't get your disk password, which is perhaps an improvement.

Disk key in TPM or user passphrase

Depends on your threat model. With the disk key in the TPM an attacker would need to have the entire machine (or a backdoor in the TPM) to get the key and their attempts to unlock it can be rate limited by the TPM hardware. However, this ties the disk to that one machine (without having to recover and type in the master key), which might be an unacceptable risk for some users.

Why is it called Heads

The flip side of Tails.

Unlike Tails, which aims to be a stateless OS that leaves no trace on the computer of its presence, Heads is intended for the case where you need to store data and state on the computer.

HOTP vs TOTP

HOTP (HMAC-based One-time Password algorithm) generates a password using hash-based message authentication codes (HMAC) that can be used only for the one authentication attempt. Uniqueness is based on a counter which is incremented each authentication attempt.

TOTP (Time-based One-time Password algorithm) is an extension of HOTP but replaces the counter with time. Because of latency, both network and human, and unsynchronised clocks, the one-time password must validate over a range of times between the authenticator and the user. Here, time is downsampled into larger durations (e.g., 30 seconds) to allow for validity between the parties.

Secuirty wise, HOTP is more susceptible to brute force attacks without throttling or limiting the number of failed attempted while TOTP is susceptible to phishing attacks and requires a user to enter the code within a given time period.

coreboot vs Linuxboot

TO BE WRITTEN

What happens if I lose/break my security key

TO BE WRITTEN