

Don't Fear the Secure Boot!



Steve McIntyre <93sam@debian.org>

13th October 2024



Agenda

- What is UEFI Secure Boot?
- Process
- Is SB worthwhile?
- Misunderstandings and misinformation
- Challenges
- Future



What is UEFI?

- Unified Extensible Firmware Interface
 - Rich(er) set of APIs than BIOS
 - UEFI Boot Services
 - UEFI Runtime Services
 - ExitBootServices() changes the world
- EDK2 reference implementation
 - Started by Intel, now maintained in the Tianocore project





What is Secure Boot?

- A way to protect against persistent boot-time malware
- Software is verified by signature
- (Typically) each step in the boot chain verifies the next
- Just(!) a problem of key management...
- Can act as a base for further security solutions
 - Measured Boot
 - Locked-down kiosk systems
 - Etc.



Key Management

- Microsoft keys included with most (x86) PCs
- Logistics of being a CA
- Arm-based machines too
 - And likely further architectures
- Multiple keyrings defined
 - PK, KEK, DB, DBX
- Two root CA keys currently in common use:
 - Microsoft Root CA (2010)
 - Microsoft Third Party UEFI CA (2011)
- On most machines you can modify the list of trusted keys

What is Secure Boot **not**?



- A way to lock people out of their own machines
 - Enrol your own keys
 - Or turn off SB
- A way to stop people using Linux and other Free Software
 - Microsoft and Linux folks talk regularly
 - The point is to add security for users **on both sides**



The Linux story

- Firmware boots a signed shim binary
- Shim includes key(s)
 - Adds an extra root of trust
 - Also adds Machine Owner Keyring (MOK)
- Further programs signed using that key chain
 - GRUB, fwupd, kernel image, UKI
- Shim: small bootloader with minimal dependencies
 - Small enough to be audited
 - BSD-licenced



The process

- Using your own keys is possible and (sometimes!) easy
 - Enrol your own keys in the firmware
 - Or: use an existing shim and add keys to MOK
 - Per machine...
- Build a shim including your keys, get that signed by Microsoft
 - Reproducible binary build
 - Submit for review, paperwork
 - Reviewed by the shim-review team
 - If all goes well, you get a signed binary back
 - But...



Revocations

- Staying secure means keeping up with fixes
 - Replace older software with known security holes
- Revocations are **hard**
 - DBX doesn't work as designed
 - SBAT to the rescue!
- New revocations are pushed out from time to time
 - Might be from firmware updates
 - Might be from new versions of shim **or other software**
- Shim (and GRUB, etc.) will not be trusted forever
 - Keep up to date
 - LTS?!?



Is SB Worthwhile?

- **Probably**, for **most** people
 - Persistent boot-time malware **is** a real problem
- It does make some things harder
 - Hibernation
 - Loading third-party kernel modules
 - Kexec
 - Direct access to memory and I/O ports

Misunderstandings

(and misinformation?)



- “Secure Boot is just designed to lock you into Windows”
- “Secure Boot adds more vulnerabilities”
- “Secure Boot doesn’t work if you’re using Testing”
- “Secure Boot doesn’t protect the average user”
- ...



Challenges

- Firmware vendors Doing It Wrong™
 - Broken UEFI, plus new ways!
 - Using broken keys
 - Leaking keys
 - Mis-handling revocations
- CA rollover
 - Microsoft 3rd-party UEFI CA
Not After : Jun 27 21:32:45 2026 GMT
 - Microsoft Root CA
Not After : Oct 19 18:51:42 2026 GMT
- Revocations



Future

- More security in UEFI binaries
 - NX
 - ASLR
- Unified Kernel Images
 - Kernel, initramfs, command line all baked in
- More use of TPM
- Architecture updates
- Better handling of revocations
 - Maybe via fwupd?



Questions?

- More background:
 - <https://wiki.debian.org/UEFI>
 - <https://wiki.debian.org/SecureBoot>



- Slides at <https://www.einval.com/~steve/talks/Mini-DebConf2024-SecureBoot>