» The Linux Foundation

Making UEFI Secure Boot Work With Open Platforms

James Bottomley, CTO, Server Virtualization at Parallels
& Linux Foundation Technical Advisory Board Chair

Jonathan Corbet, Editor at LWN.net
& Linux Foundation Technical Advisory Board Member

October 2011

The Linux Foundation
http://www.linuxfoundation.org

Making UEFI Secure Boot work with Open Platforms

"Secure boot" is a technology described by recent revisions of the UEFI specification; it offers the prospect of a hardware-verified, malware-free operating system bootstrap process that can improve the security of many system deployments. Linux and other open operating systems will be able to take advantage of secure boot if it is implemented properly in the hardware. This document is intended to describe how the UEFI secure boot specification can be implemented to interoperate well with open systems and to avoid adversely affecting the rights of the owners of those systems while providing compliance with proprietary software vendors' requirements.

The recommendations can be summarized as follows:

- All platforms that enable UEFI secure boot should ship in setup mode where the owner has control over which platform key (PK) is installed. It should also be possible for the owner to return a system to setup mode in the future if needed.
- The initial bootstrap of an operating system should detect a platform in the setup mode, install its own key-exchange key (KEK), and install a platform key to enable secure boot.
- A firmware-based mechanism should be established to allow a platform owner to add new key-exchange keys to a system running in secure mode so that dual-boot systems can be set up.
- A firmware-based mechanism for easy booting of removable media.
- At some future time, an operating-system- and vendor-neutral certificate authority should be established to issue KEKs for third-party hardware and software vendors.

The reasoning behind these recommendations is summarized below.

How UEFI Secure Boot is Designed to Work

The UEFI specification (version 2.3.1) is a bit dense, weighing in at 2,139 pages. However, the sections dealing with how secure boot should work are nicely comprehensive and very amenable to use by open platforms. The design the UEFI committee came up with is elegantly outlined in section 27.5 of the specification, but can be summarized in terms of a two-key system.

UEFI specifies (section 27.5) a Platform Key (PK), which is designed to be controlled by the Platform Owner (whoever owns the hardware) and a set of Key-Exchange Keys (KEKs), which are designed to be controlled by the OEM and OS vendors. "Controlled" in this sense means that these keys are public/private key pairs; whoever knows the private key is the key controller, but to install the key, you only need the public piece, which means KEKs may be installed by anybody without controlling them.

This separation is vital because it allows the platform owner to decide which keys they trust without compromising the ability of the KEK controllers to assure themselves that the OS booted securely.

Interoperation with Open Systems

To enable proper operation with open systems, all UEFI secure boot platforms should ship in setup mode, with no Platform Key installed. This enables the Platform Owner to take control of the platform securely by installing their own platform key or allowing the Operating System install process to do so. There should also be a mechanism (very similar to the current Trusted Platform Module reset operation) to allow a platform to be reset back to setup mode for the case where

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org
info@linuxfoundation.org

the owner wishes to resell itor has lost control of the PK.

The platform should ship with all the KEKs required to allow validation of the firmware and drivers installed in the signature database (section 27.6.1). The signature database will be inactive while the platform is in setup mode, but, once secure boot is activated, the firmware and all of the add in driver components must validate correctly for the platform to progress to boot the Operating System (since elements of the Firmware come from add in cards, getting the complete set of KEKs is problematic, but we'll address that later).

The initial ignition of an Operating System that makes use of secure boot (whether that system is pre-installed or installed from external media) should detect that the platform is in setup mode; the operating system can then switch the platform to the secure mode by installing a platform key after it has installed the KEK corresponding to the its own code. An Open System will likely generate a new PK at first boot, install the public component and save the private component to external media for the user.

Now that a secure Operating System boot loader is installed with a signature which is validated it will be the job of the boot loader to validate and run the operating system, which it will presumably do by a similar signature means.

In the scenario above, the KEK used to boot the operating system is still controlled by the producer of the operating system, as envisaged by the UEFI specification. However, an open platform owner still has the right to modify the platform and expect it to boot. This right is enabled because control of the PK permits the platform owner to generate a new KEK for their modified operating system and install it in the signature database. The platform owner, being controller of the PK, would even be able to remove the original operating system KEK, thus ensuring that the platform would only boot the modified operating system. This scheme of allowing the platform owner to boot arbitrarily-modified operating systems via KEKs that they construct is fully compliant with all OSI-approved open-source licences.

Dual Boot

The scheme above covers first boot for the case where the platform ships in setup mode. Once a platform key is installed, the platform operates in user mode, where it will only boot secure operating systems whose signatures are in the signatures database (or are signed by a KEK in the signature database). When a new operating system is presented, it typically won't have this property, so there has to be a mechanism for the platform owner to validate and install a KEK even before the system is booted.

The most secure way to enable this operation is to provide a set of EFI tools for managing the signature database that would allow the user to validate the install media, extract the OS Vendor KEK from it and install it into the database. Ideally, this tool would be activated by the UEFI system as soon as it saw unauthorised media inserted, so the platform owner could decide whether they wished to accept the key for OS install and boot.

Such chicken-and-egg first-boot problems can be avoided if all OS vendors participate in a chain of trust which allows validated root signatures to be present in the UEFI signature database ab initio (see later).

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org
info@linuxfoundation.org

Booting Closed Operating Systems

Obviously, a closed operating system could be booted identically to an open one above and still retain all its secure features, since security is guaranteed by control of the KEK which would remain in the hands of the operating system vendor. However, Steven Sinofsky has suggested in his blog posting "Protecting the pre-OS environment with UEFI":

http://blogs.msdn.com/b/b8/archive/2011/09/22/protecting-the-pre-os-environment-with-uefi.aspx

that the average platform owner might wish to give up control of the PK (and with it control of the signature database) to Microsoft and the OEM suppliers of the platform.

This mode of operation runs counter to the UEFI recommendation that the platform owner be the PK controller and would ensure that the Windows operating system would then become the only bootable operating system on the platform, but we must agree that it is a legitimate choice for an informed user to make voluntarily. It is enabled in our blueprint above by allowing the Microsoft OEM ignition system to install the OEM PK instead of generating a new PK specific to the installation. This can be achieved simply and securely because only the public half of the PK needs to be carried by the ignition system to effect this lockdown of the platform. Such a scheme is fully consonant with the current draft version of the Windows 8 UEFI logo requirements.

The ability of the platform owner to regain control should they desire it is guaranteed by the ability to securely reset the platform back to setup mode.

Establishing a Trust Model

One of the few shortcomings in the UEFI model (and it is a deliberate omission because of the complexity of running a certification system) is that there's no designated root of trust in the current proposals. This leads to the obvious shortcoming that, for a desktop system loaded with multiple third-party cards, the EFI drivers for each of the cards must have its own KEK in the signature database. Even in the open model, where the platform owner controls the PK, it becomes burdensome to verify and allow a new KEK every time a single PCI card is added or changed. In the closed model, where the OEM controls the PK, it becomes impossible without an update supplied by the OEM.

To solve this problem, UEFI allows (in section 27.6.1) X509 certificates to be present in the signature database. The X509 trust model, which is the one upon which web server and browser security certificates are based, allows signatures and signing keys to be traced back to a single root of trust. This would allow for one (or more) Certificate Authority keys to be placed in the UEFI signature database and would then allow the designated Certificate Authorities to issue both KEKs (and even signing keys that allow the production of KEKs) to third parties that would still validate back to the original CA root of trust. The UEFI specification (section 27.7.1) even allows for the revocation of KEKs should the original authorised user have lost control of them, which is all the necessary machinery you need to operate a fully functional CA.

We therefore propose that all the interested parties establish a Certificate Authority whose key should be placed in the UEFI firmware table by default; this authority would become responsible for handing out signed KEKs to UEFI device vendors (for their UEFI drivers), UEFI OEM platform vendors (for their firmware images) and OS vendors (for securely booting their OSs). The operation

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org
info@linuxfoundation.org

of such a CA would have to be platform- and OS-neutral and would have to adhere to the usual standards of trust and security (presumably by having a controlling board made up of representatives from the various parties), but it would solve a greater part of the driver and OS verification problem because anything signed with an un-revoked KEK traceable back to the CA root key would be automatically trusted by the UEFI firmware for secure boot.

Booting from External Removable Media

All non-preinstalled Operating Systems (which is in effect every Open Operating System) must be installed (or tried out as a Live CD) from some external media such as USB key, DVD or CD. The establishment of a Trust Model as described above would solve the problem of secure boot from these media (since Open OS vendors would simply sign the media with a key traceable to the root of trust). However, in the absence of a trust model, there has to be a way of allowing easy external media boot when the KEK or signature of the external media image isn't necessarily present in the signature database. In the absence of the establishment of a trust model, we therefore recommend that non-verifying external media be booted with a simple firmware based present user permission check when the system is in user mode with secure boot enabled.

Conclusion

The UEFI secure boot facility is designed to be readily usable by both proprietary and open operating systems to improve the security of the bootstrap process. Some observers have expressed concerns that secure boot could be used to exclude open systems from the market, but, as we have shown above, there is no need for things to be that way. If vendors ship their systems in the setup mode and provide a means to add new KEKs to the firmware, those systems will fully support open operating systems while maintaining compliance with the Windows 8 logo requirements. The establishment of an independent certificate authority for the creation of KEKs would make interoperation easier, but is not necessary for these platforms to support open systems.

1796 18th Street, Suite C
San Francisco, CA 94107
+1 415 723 9709
http://www.linuxfoundation.org
info@linuxfoundation.org

The Linux Foundation promotes, protects and advances Linux by providing unified resources and services needed for open source to successfully compete with closed platforms.

To learn more about The Linux Foundation, and our other initiatives please visit us at http://www.linuxfoundation.org/.