UEFI Secure Boot

Where we stand

Parallels™

Profit from the Cloud

James Bottomley

CTO, Server Virtualization; SCSI Subsystem, Parisc Kernel Maintainer

25 October 2012

Introduction

- UEFI Secure boot is a static way of assigning trust to the boot system
- It is mandated by Microsoft to be enabled in all shipping Windows 8 systems
- The Microsoft Mandate requires all keys to be owned either by the OEM or by Microsoft
- Secure Boot must be capable of being Disabled and the keys replaced
- But no standard mechanism for doing this exists

The Secure Boot Keys

- There are three sets of keys
  - The Platform Key (PK) , designed to be owned by the owner of the hardware
    - Microsoft mandates that this belong to the OEM
  - The Key Exchange Keys (KEK) designed to be owned by trusted entities for boot
    - Microsoft mandates they own at least one of these
  - The Signature Database (db) designed to verify trusted binaries
    - Microsoft mandates they have a key here too.
    - db signatures are required to boot in a trusted environment

How it Works

- PK may only be used to update KEK
  - So the PK owner decides what keys to trust in the KEK list
- KEK may only be used to update db
  - So all owners of KEKs can update or revoke db keys
- db keys must be used to sign binaries which are trusted by the system.

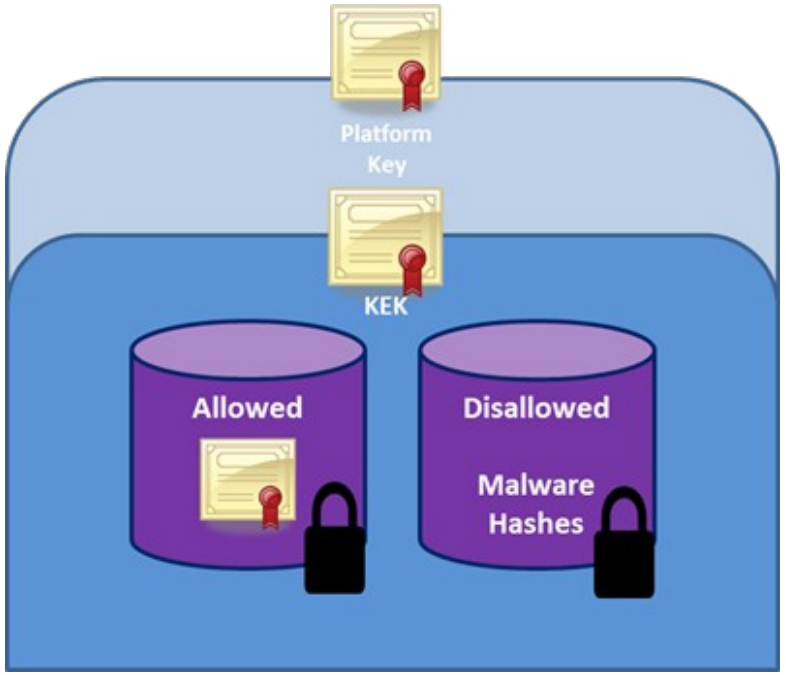

Diagram from Microsoft

How Microsoft Mandates that it Work

- The Windows 8 Logo Requirements are
  - OEM controls Owner Key
  - Microsoft owns keys in KEK and db
    - Several keys, in fact: it looks like Windows boot will be signed by a separate root of trust from the third party signing system
  - On non-ARM systems, secure boot must be disabled via a UEFI menu
    - No mandate for where this is or how easy it is to do.
  - On non-ARM systems, the user must be able to replace all the keys
    - Again, no requirement for key administration
    - OEM can comply by simply having the system remove all the keys

GPLv3 and Secure Boot

- People think GPLv3 requires disclosure of signing keys in a lock down environment
- The Linux Foundation saw this problem in the early drafts of the Microsoft Windows 8 Logo docs and sought to fix it
- However requirement is only that the user be able to boot their own system
- Ejecting the preset keys and installing your own, with which you can then sign your system is sufficient
- Implies reset to setup mode in UEFI interface, as Mandated by Microsoft, satisfies GPLv3 obligation
- FSF Supports this interpretation

The Threat

- Since Microsoft owns all the Signing keys, no Linux boot system will work out of the box without their approval
- Approval requires not booting malware
  - Implies simply getting Microsoft to sign a Linux bootloader isn't an option
- Linux won't boot on Windows 8 systems without a Microsoft approved method of booting
  - Trying to explain to users how to disable secure boot isn't an option
  - Because of the non-standard mechanisms for doing so.

The Opportunity

- Secure boot gives users a way of protecting their systems from external intrusion
- Supporting it end to end would facilitate Linux playing in secure environments
- To be effective, must carry the root of trust through the secure boot to the Operating System environment
  - May require other trust implementations like signed modules
  - Or disallowing root access to PCI configuration space

The Linux Response

- Two Challenges
  1. Keep the Ecosystem booting easily in the face of secure boot
  2. Enhance Security policy for distributions by taking advantage of secure boot.
- The Linux Foundation response has concentrated exclusively on 1.
- The Linux Distributions are Investigating and preparing for 2.
- Both may be required to be ready for windows 8 release day
  - 26 October

What the LF is doing

- Develop a set of tools to enable owner to easily take control of the system and manage the keys
  - Allows ejecting of OEM and Microsoft keys and installing your own
- Tools also permit the creation of signed binaries to reset the platform to setup mode
  - Just in case something goes wrong with UEFI interface
- A signed pre-bootloader that will boot any unsigned bootloader with a present user test
  - And will install bootloader signature in setup mode to avoid the present user test

What the Distributions are Doing

- Red Hat (Matthew Garrett) interacted with UEFI forum and OEMs to create shim bootloader
  - Boots a signed second stage loader, which boots a signed kernel
  - Kernel is locked down by module signing and other measures
- SUSE has Machine Owner Key (MOK) approach
  - Shim modified to accept key updates from present user
  - Means user can resign the boot loader and install their own key
- Both approaches require signing shim with the microsoft key

Secure Boot Solutions Converging

- Red Hat and SUSE moving towards shim + MOK solution
- Matthew Garrett adding ability to store hashes in MOK database
  - Means shim + MOK can now chain unsigned bootloaders
  - Provides essentially all the functionality of the LF pre-bootloader.

The Convergence

- Secure Boot D-Day is Windows 8 Release day
  - 26 October (10 days ago)
- Linux is ready to continue booting
  - LF pre-bootloader as stop gap measure
- As well as take advantage of secure boot
  - Red Hat and SUSE shim + MOK approach
- Plus users will be able to replace all the keys (PK, KEK, db) if they wish
- Secure Boot should therefore be business as usual for Linux

Demo

- Resources:
  - https://build.opensuse.org/project/show?project=home%3Ajejb1%3A
  - http://git.kernel.org/?p=linux/kernel/git/jejb/efitools.git;a=summary
- Includes tianocore qemu image for UEFI plus tools for taking control of system and building keys and signature lists.

Questions?

jbottomley@parallels.com
James.Bottomley@HansenPartnership.com