20171019_Secure Boot feature Study and practice

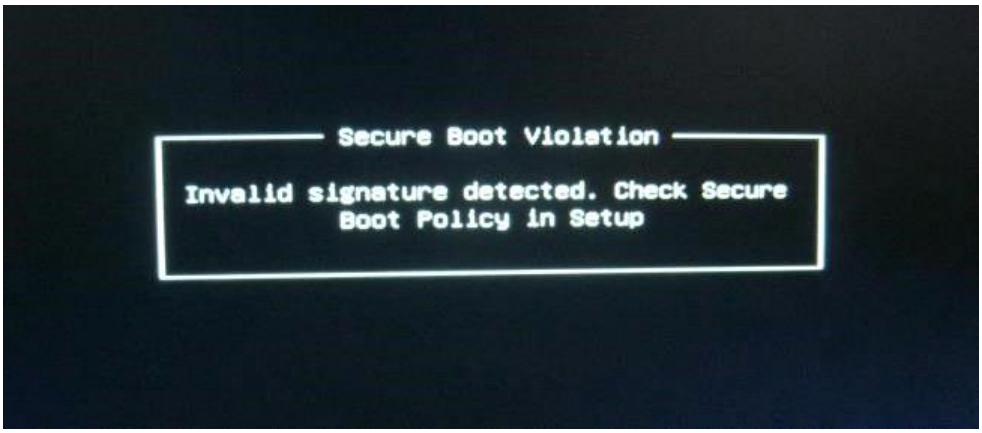
2017年10月19日　　10:52

What's UEFI secure boot?

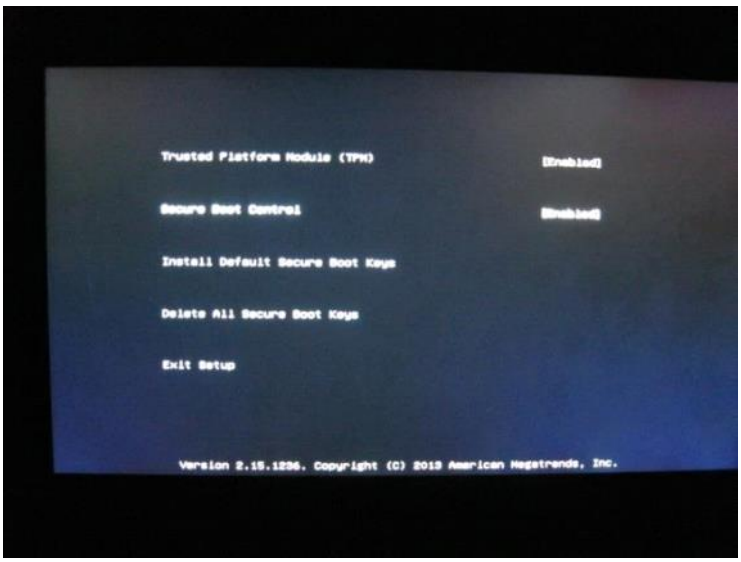
A UEFI firmware feature, which can secure the boot process by preventing the loading of drivers or OS loaders that are not signed with an acceptable digital signature.

Secure boot is supported by Windows 8 and 8.1, Windows Server 2012, and 2012 R2, and Windows 10, VMware vSphere 6.5 and a number of Linux distributions including Fedora (since version 18), openSUSE (since version 12.3), RHEL (since RHEL 7), CentOS (since CentOS 7) and Ubuntu (since version 12.04.2).As of January 2017, FreeBSD support is in a planning stage.

What secure boot looks like if enabled?

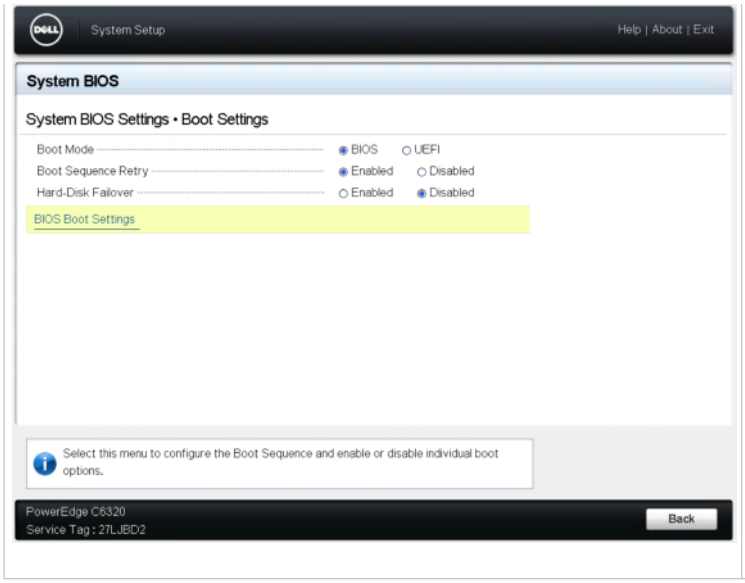


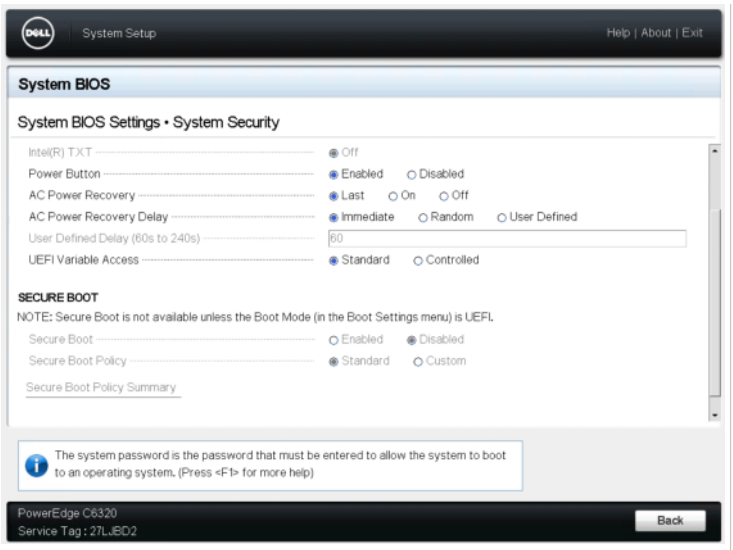
How to disable secure boot in BIOS?



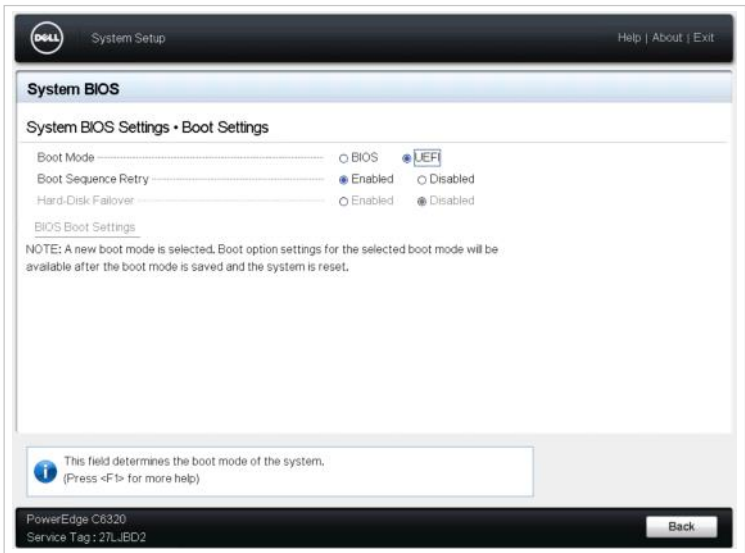
UEFI Secure Boot on Dell C6320 (in BIOS setup)

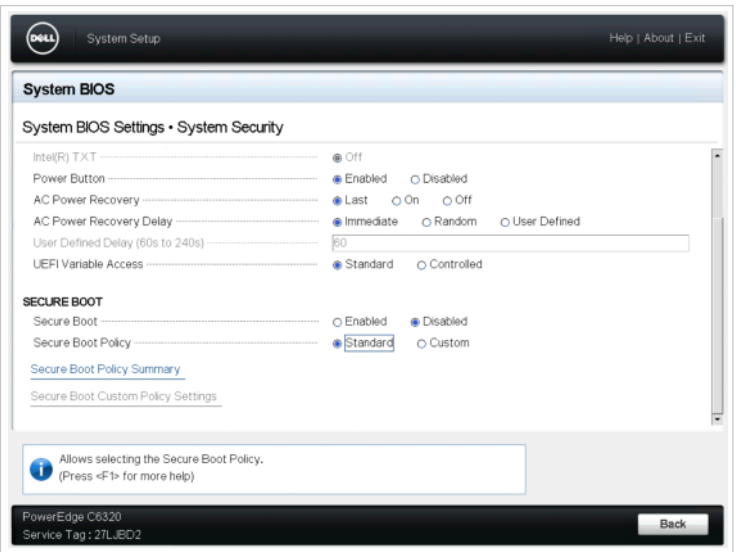
- if Boot Mode = BIOS, then Secure Boot feature is not available (become grey)

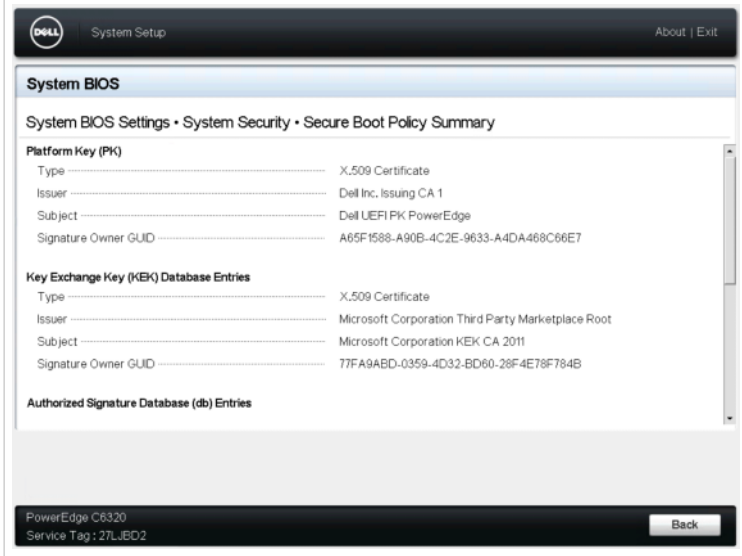




- if Boot Mode = UEFI, then Secure Boot feature is able to be configured.







UEFI Secure Boot In-Depth

UEFI Secure Boot

Secure boot is designed to protect a system against malicious code being loaded and executed early in the boot process, before the operating system has been loaded. This is to prevent malicious software from installing a "bootkit" and maintaining control over a computer to mask its presence. If an invalid binary is loaded while secure boot is enabled, the user is alerted, and the system will refuse to boot the tampered binary.
On each boot-up, the UEFI firmware inspects each EFI binary that is loaded and ensures that it has either a valid signature (backed by a locally trusted certificate) or that the binary's checksum is present on an allowed list. It also verifies that the signature or checksum does not appear in the deny list. Lists of trusted certificates or checksums are stored as EFI variables within the non-volatile memory used by the UEFI firmware environment to store settings and configuration data.

UEFI Key Overview

The four main EFI variables used for secure boot are shown in Figure a. The Platform Key (often abbreviated to PK) offers full control of the secure boot key hierarchy. The holder of the PK can install a new PK and update the KEK (Key Exchange Key). This is a second key, which either can sign executable EFI binaries directly or be used to sign the db and dbx databases. The db (signature database) variable contains a list of allowed signing certificates or the cryptographic hashes of allowed binaries. The dbx is the inverse of db, and it is used as a blacklist of specific certificates or hashes, which otherwise would have been accepted, but which should not be able to run. Only the KEK and db (shown in green) keys can sign binaries that may boot the system.

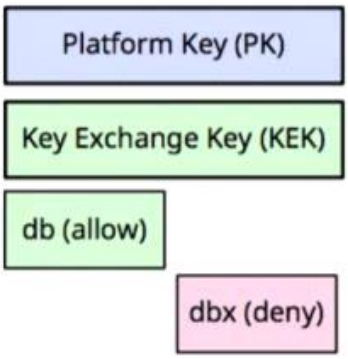


Figure a. Secure Boot Keys

From <http://www.linuxjournal.com/content/take-control-your-pc-uefi-secure-boot?page=0,0>

Reference

- Take Control of Your PC with UEFI Secure Boot
- How to Boot and Install Linux on a UEFI PC With Secure Boot
- RAC-4902 UEFI Secure Boot Redfish API
  As a RackHD user, I want to be able to manage UEFI Secure Boot functionality on a dell node using the Redfish API.

  :
  Develop Redfish APIs for:
  Add secure boot resource to a node.
  Enable or disable secure boot on a node.
  Get current secure boot state.

  https://github.com/RackHD/on-http/pull/648