UEFI LOCKDOWN QUICK GUIDANCE

HARNESSING UEFI'S ACCOUNTS, SETTINGS, AND KEYS TO ENHANCE SECURITY

UEFI CONFIGURATION PASSWORDS

Unified Extensible Firmware Interface (UEFI) provides multiple levels of password-based boot control. Three password levels are used to interact with machine firmware prior to the operating system boot. Failure to secure these accounts can open machines up to unauthorized, undesired, and repudiated boot device changes, device/component firmware configuration changes, and unauthorized connectivity to peripheral devices.

Administrator Password

Set a unique UEFI administrator password for each device. Administrator passwords limit access to all UEFI configuration options. Only authorized enterprise support or administrative personnel should have access to the device password.

User Password

Set a UEFI user password or disable the user account. Use a device-specific, office-specific, or enterprise-wide password value based on user, support, and mission needs. Disable the user account to restrict UEFI changes to administrators only.

System and Storage Passwords

Avoid using UEFI system and storage passwords – both may disrupt the operating system update process by requiring user intervention during boot/reboot. Evaluate update-friendly, disk encryption alternatives provided by the Operating System (OS) or software vendors.

RECOMMENDED UEFI CONFIGURATION SETTINGS

Develop a specific UEFI configuration for each make and model device. Consider the following guidelines:

- Configure boot order to prioritize the boot drive. Disable unused boot options, media, and shells.
- Disable unused drive, expansion, and peripheral device ports (USB, SATA, eSATA, PCIe, etc.).
- Disable Management Engine BIOS Extension (MEBx) keyboard access if present.
- Enable UEFI Secure Boot. Normal/standard mode supports most operating systems and hypervisors. Consider custom/user mode for more control.
- Enable and activate the Trusted Platform Module (TPM).
- Disable non-admin UEFI configuration changes. Verify that boot order can't be changed by users.
- Disable Option ROM (OROM) keyboard access. Do not allow non-administrative changes to RAID, graphics, network, or other OROM device configuration.

UEFI SECURE BOOT CUSTOM KEY DISTRIBUTION

Optionally, Secure Boot can be customized for more in-house control. Devices normally ship with a vendor PK, Microsoft KEK, and Microsoft DB keys – keys that may permit more boot flexibility than desired. Secure Boot certificates and

hashes are recorded in Trusted Platform Module (TPM) Platform Configuration Register (PCR) 7 for device integrity auditing purposes. Customize UEFI Secure Boot keys in the following way:

Platform Key (PK)

Create a unique PK (RSA 2048-bit key pair) for each device. Each PK should be signed by an enterprise Certificate Authority (CA) if available (a commercial or in-house CA is also acceptable). Replace the vendor PK with the signed PK public key certificate. The PK functions as a root key for UEFI Secure Boot key storage. PKs can authorize runtime changes to Secure Boot keys and variables.

Key Exchange Key (KEK)

Create an enterprise KEK (signed by the enterprise/commercial/in-house CA if available) and load it on all devices booting a non-Microsoft OS. Use the Microsoft KEK when booting a Microsoft OS. Use both the enterprise KEK and Microsoft KEK together if the device allows multiple KEKs. The enterprise KEK should sign DB and DBX keys used by the enterprise, organizations, or individual users as appropriate. The KEK can authorize runtime changes to some Secure Boot variables, DB keys, and DBX keys. Note that the KEK does not need to be signed by a PK.

Whitelist Database (DB)

Create one or more DB keys to be deployed by the enterprise, organizations, and/or users according to device mission. Each DB key should be signed by the enterprise KEK. DB keys are used to sign trustworthy bootable drivers, binaries, and modules. SHA-256 hashes of trusted boot content may also be entered into the DB.

Blacklist Database (DBX)

Place revoked, retired, or otherwise untrusted signing keys, KEKs, and SHA-256 hashes of bootable content in the DBX.

REFERENCES

Developer Zone (21 March 2011). About UEFI. Intel. Retrieved from http://software.intel.com/en-us/articles/about-uefi

Hardware Dev Center (2 May 2017). UEFI Firmware. Microsoft. Retrieved from https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/uefi-firmware

DISCLAIMER OF WARRANTIES AND ENDORSEMENT

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government, and this guidance shall not be used for advertising or product endorsement purposes.

CONTACT INFORMATION

Client Requirements and General Information Assurance/Cyber Security Inquiries
Cybersecurity Requirements Center (CRC)
410-854-4200
Email: Cybersecurity_Requests@nsa.gov