Extending the Secure Boot Certificate and Signature Chain of Trust to the OS

Fionnuala Gunter, fin.gunter@hypori.com
Mimi Zohar, zohar@linux.vnet.ibm.com
Secure Boot Chains of Trust

- Secure Boot places the root of trust in hardware write protected firmware and public keys.
- Public key certificates establish a chain of trust based on validating signatures.
- Firmware uses public key(s) to validate the signed bootloader.
- The signed bootloader can then validate the signed kernel, and so on.

Diagram:
- KEK
- db
- Shim
- GRUB2
- Kernel
- MoK List
Secure Boot Chains of Trust

- **PK** - Platform Key (OEM key)
- **KEK** - Key Exchange Keys Database (OS vendor keys)
- **db** - Signature Database
- **MoK** - Machine Owner Key (the machine owner can replace boot components using mokutils tool)
Extending the Secure Boot Certificate and Signature Chain of Trust to the OS

- Load public key
- Load signed certificates
- Sign certificates
- Kernel validates certificates
- Signed/self-signed 3rd party certificates
- IMA local CA

Flow Diagram:
- Load public key
  - → system keyring
  - → Kernel validates certificates
  - → IMA keyring
- Load signed certificates
  - → Sign certificates
- IMA local CA
  - → Load signed certificates
  - → signed/self-signed 3rd party certificates
Methods for Loading IMA Local-CA Public Key on the System Keyring

1. Compile key into Linux kernel

2. Load the UEFI/MoK database keys (RedHat’s patches)

3. Pre-allocate space in the kernel image for IMA local-CA public key. Post build, install key and resign kernel image.
Sign Certificates with IMA Local-CA Private Key

• Which certificates and why?

• Signing distro/3rd party certificates without a certificate signing request (CSR)

openssl ca -ss_cert cert.pem
Load Signed Certificates onto IMA Keyring

• Certificates containing a key used to verify file signatures need to be signed by a system trusted key

• This extends the signature chain of trust to the OS

• The dracut integrity module loads signed certificate keys onto the trusted .ima keyring

Kernel validates certificates

IMA keyring

Load Signed Certificates

signed/self-signed 3rd party certificates
Labeling Filesystems with Signatures

- The Linux kernel’s integrity subsystem verifies and appraises file integrity based on file signatures

- Files are currently signed, post install, by walking the filesystem

- A better, more complete solution is to include file signatures in software packages

- This enables files to be automatically labeled with signatures during installation
RPM File Signatures

- Extended the existing rpm signing tool to include file signatures in packages
- RPM plugin installs file signatures using post transaction element hook (psm_post)
- Expected in rpm-4.13.0
RPM Including File Signatures

- New Command
  `rpmsign --addsign --signfiles PACKAGE_FILE`

- Sign Files Options
  `--fskpath` and `--fskpass`
RPM Including File Signatures

- The new option signs all the file digests included in the package with libimaevm v1.0
- File signatures are stored in the package header under the tag RPMTAG_FILESIGNATURES
- After including file signatures, the packages are signed normally
RPM Installing File Signatures

- When a package is installed, rpmfilesPopulate extracts file signatures from the package header and stores them in rpmfiles struct

- The RPM plugin instantiates the post transaction element hook (psm_post) and writes the file signatures to security.ima xattr
deb Including File Signatures

- Control.tar.gz in the .deb packages contains a md5sums file

- Include digest sums file in package (eg. sha256sums)

- Append file signatures
  ```bash
cat sha256sums | evmctl sign_hash -a sha256 -key "${PRIVKEY}" > sha256sums
  ```
deb Installing File Signatures

- debhelper script and autoscript install ELF file and script signatures stored in the sha256sums file
- debhelper script: dh_installfile-sigs
- autoscript: postinst-file-sigs
Next Steps

• Upstream deb file signature extensions - feature request #766267

• Linux software distributors ship packages with file signatures
References

• https://wiki.ubuntu.com/SecurityTeam/SecureBoot


• http://blog.hansenpartnership.com/the-meaning-of-all-the-uefi-keys/