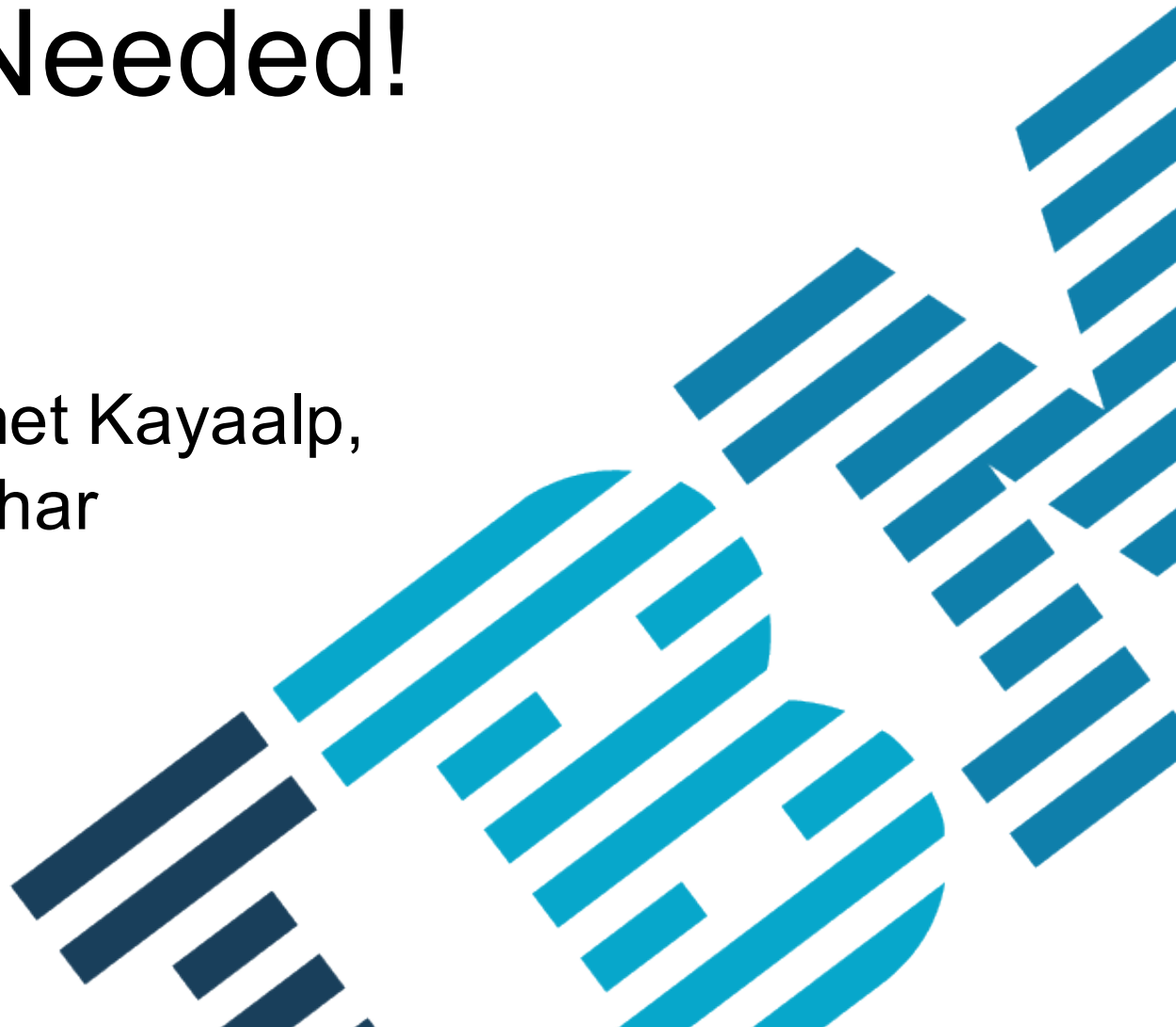
File Signatures Needed!

Authors: Stefan Berger, Mehmet Kayaalp,
Dimitrios Pendarakis, Mimi Zohar



File Signatures Needed! -- Outline

- Overview of File and Package Signature Security in Common Operating Systems
- Background on IMA and EVM
- File Signature Enablement for RPM and Ubuntu Packages
- Demo
- More Details
- Future Work

Overview of File and Package Signature Security in Common Operating System

	Base System			Additional Software			
				Runtime			
	Kernel Modules	System Files	Install	Exe	Lib	Scripts	Other
Linux (this talk)	Enforces DS	(DS)	DS[1]	(Enforces DS)	(Enforces DS)	(Enforces DS)	(Enforces DS)
Windows[2]	Enforces DS	DS	Can Enforce DS	Can Enforce DS	Can Enforce DS	Can Enforce DS	Can Enforce DS
macOS[3]	Enforces DS	DS	Can Enforce DS	DS	DS	DS	DS
iOS	Enforces DS	Enforces DS	Enforces DS	Enforces DS	Enforces DS	DS	DS
AIX[4]	Can Enforce DS	DS	DS	Can Enforce DS	Can Enforce DS	Can Enforce DS	Can Enforce DS
NetBSD[5]	-	-	DS	Can Enforce WL	Can Enforce WL	Can Enforce WL	Can Enforce WL
OpenBSD	-	-	DS	-	-	-	-
FreeBSD	-	-	DS	-	-	-	-
ChromeOS	Enforces DS	Enforces DS[6]	Enforces DS (not for trust)	-	-	-	-
Android	Enforces DS	Enforces DS[7]	Enforces DS (not for trust)	-	-	-	-

DS: Digital Signatures
WL: Whitelist of hashes
Enforces: Only allows if the verification is successful
Can Enforce: Can be configured to enforce
Not for trust: Self signed signatures accepted

(1): Distro package managers, e.g. rpm, apt-secure, signify etc.
(2): AppLocker policies can enforce Authenticode signatures
(3): Gatekeeper can enforce Apple-issued certificates
(4): Trusted Execution can enforce a path based signature database
(5): Veriexec can enforce a path based whitelist of hashes
(6): The rootfs partition is signed
(7): System app JARs are signed

Linux Kernel Features

- Integrity Measurement Architecture (IMA) (>2.6.30)
  - Detect if files have been accidentally or maliciously altered
  - Enabled from command line and configured with a policy specifying what to measure:
    - Based on the UUID or the type of the file system (e.g. exclude proc, sysfs etc.)
    - Based on the owner, user, or effective user (e.g. only if the file is owned/accessed by root)
    - Whether the file is opened/mmapped/executed, with permissions read/write/append/execute
    - Based on other LSM definitions (e.g. exclude if the SELinux label is var_log_t)
  - Measurements are logged and extended into TPM PCRs for remote attestation
- IMA Appraisal (>3.7)
  - Local integrity validation and enforcement of the measurement against an extended attribute (xattr)
  - Either a "good" hash value or a digital signature is stored as the security.ima xattr
  - With a signature, we can further establish provenance
  - During runtime, the kernel protects the security xattrs from being modified

Linux Kernel Features

- Extended Verification Module (EVM) (>3.2)
  - IMA Appraisal ensures integrity of file contents but not the security xattrs
  - EVM detects offline tampering of file metadata or the security xattrs by storing an HMAC as the "security.evm" xattr
  - An "encrypted key", configured as the EVM key, is used to update the xattr when one of the security xattrs is updated
- Trusted and Encrypted Keys (>2.6.38)
  - Symmetric keys generated in kernel
  - Exposed to the userspace only as encrypted blobs
  - Trusted keys are sealed to TPM PCRs and can be decrypted only when a trusted system is booted
  - Encrypted keys are encrypted using a "master" trusted key or user key

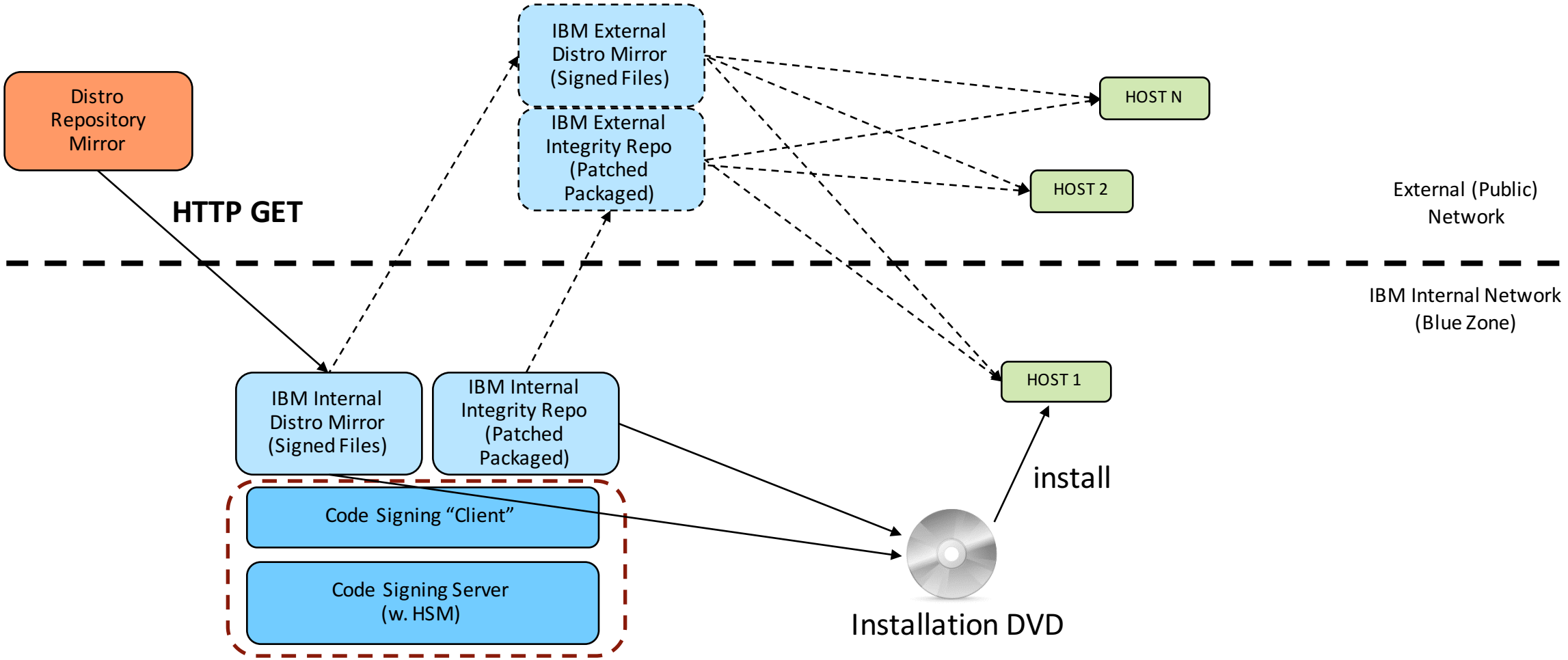
Linux Kernel Features

- Trusted keyrings: “.builtin_trusted_keys” (>3.13), “.ima” (>3.17), “.evm” (>4.5)
  - Userspace may only add a key if it can be verified by a built-in trusted key
  - Initial built-in trusted keys can be embedded in the kernel binary at compile time
  - Built-in trusted keys are also used for module signature checking
- Reserve Extra Certificate (>4.6 +patches in mailing lists)
  - Decouple the embedding of an extra built-in trusted key from the compilation
  - Reserve space in the kernel binary during compile time
  - The user can insert a new certificate to the binary and sign the resulting image for secure boot
  - The inserted key can then be used to populate IMA and EVM keyrings

IMA Appraisal

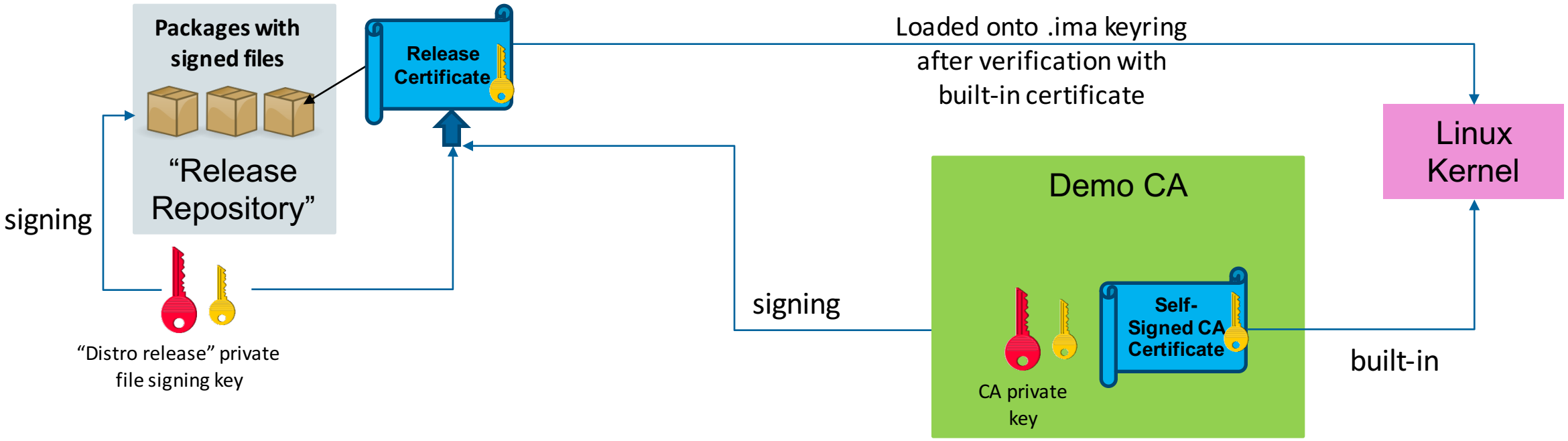
- What it does:
  - Extension of Secure Boot signature verification into the Linux OS
  - Reduces attack surface by only allowing 'sanctioned' software (= signed software from trusted repositories) to run
    - Attacker cannot execute software that's not signed, signed with uncertified key, has bad signature
  - Keeps list of executed applications and their measurements and signatures
- What it does **not** do:
  - Prevent 'sanctioned' malware from running – no guarantees on software behavior
    - Limit software behavior through other security frameworks, i.e., SELinux, AppArmor, …
  - Prevent abuse of 'sanctioned' applications
    - Examples: mmap ports on victim; use scp, curl, etc. to exfiltrate data
- Challenges:
  - Keep rogue software out of trusted repositories
    - Limit to core set of trusted packages
    - e.g. Core Infrastructure Initiative (Badge Program)
  - Identify trusted repositories

PoC: Maintaining IBM Mirrored Repositories with Signed Files



Note: Integrity Repo is temporary until all modifications are in public packages

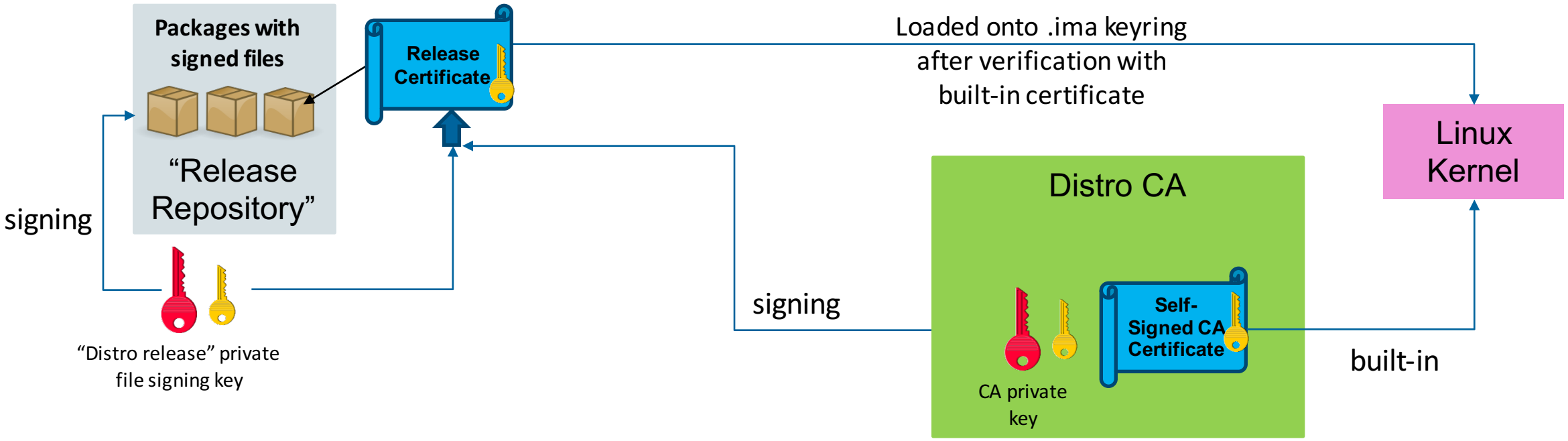
Keys and Certificates: Demo Setup



Public key

Private key

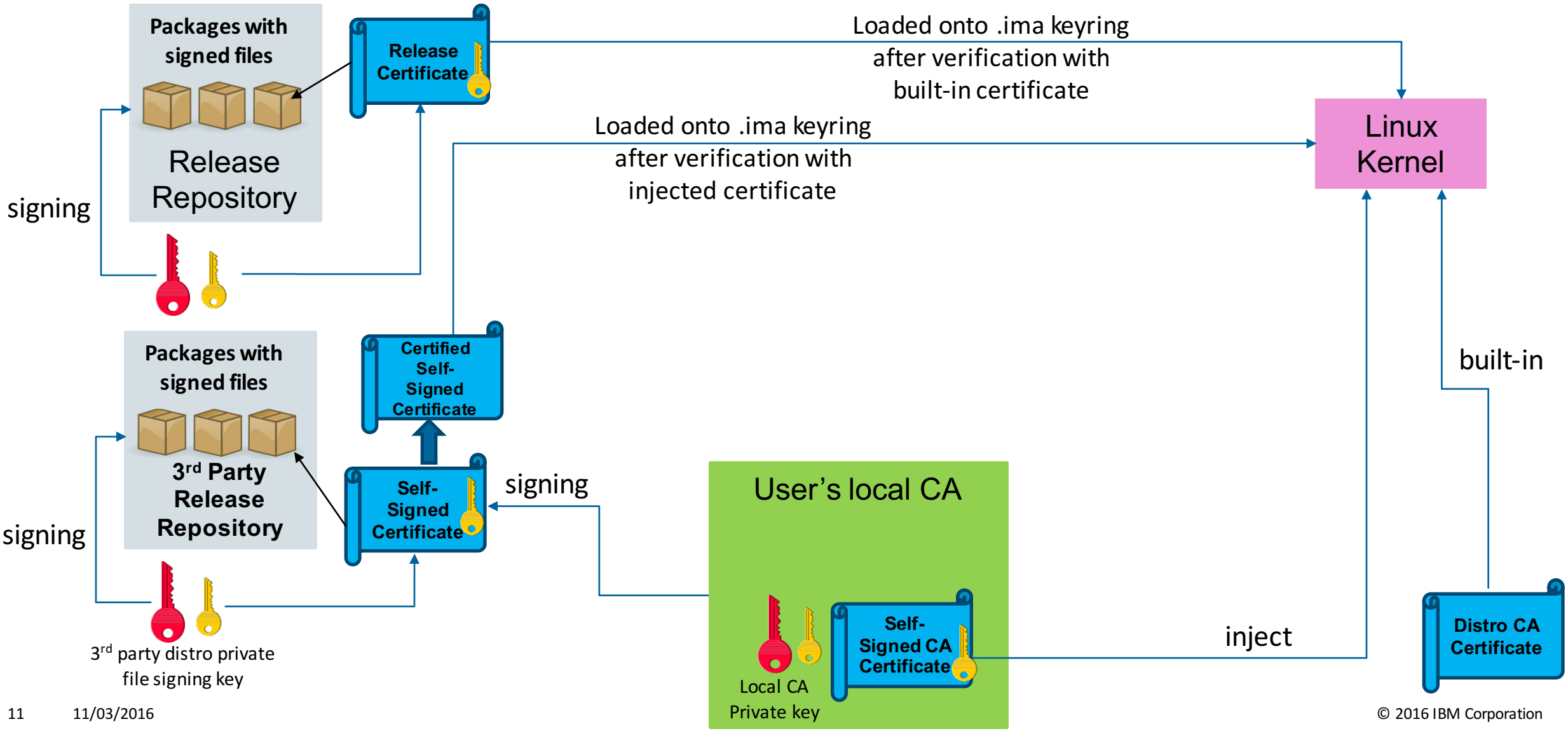
Keys and Certificates: Linux Distribution



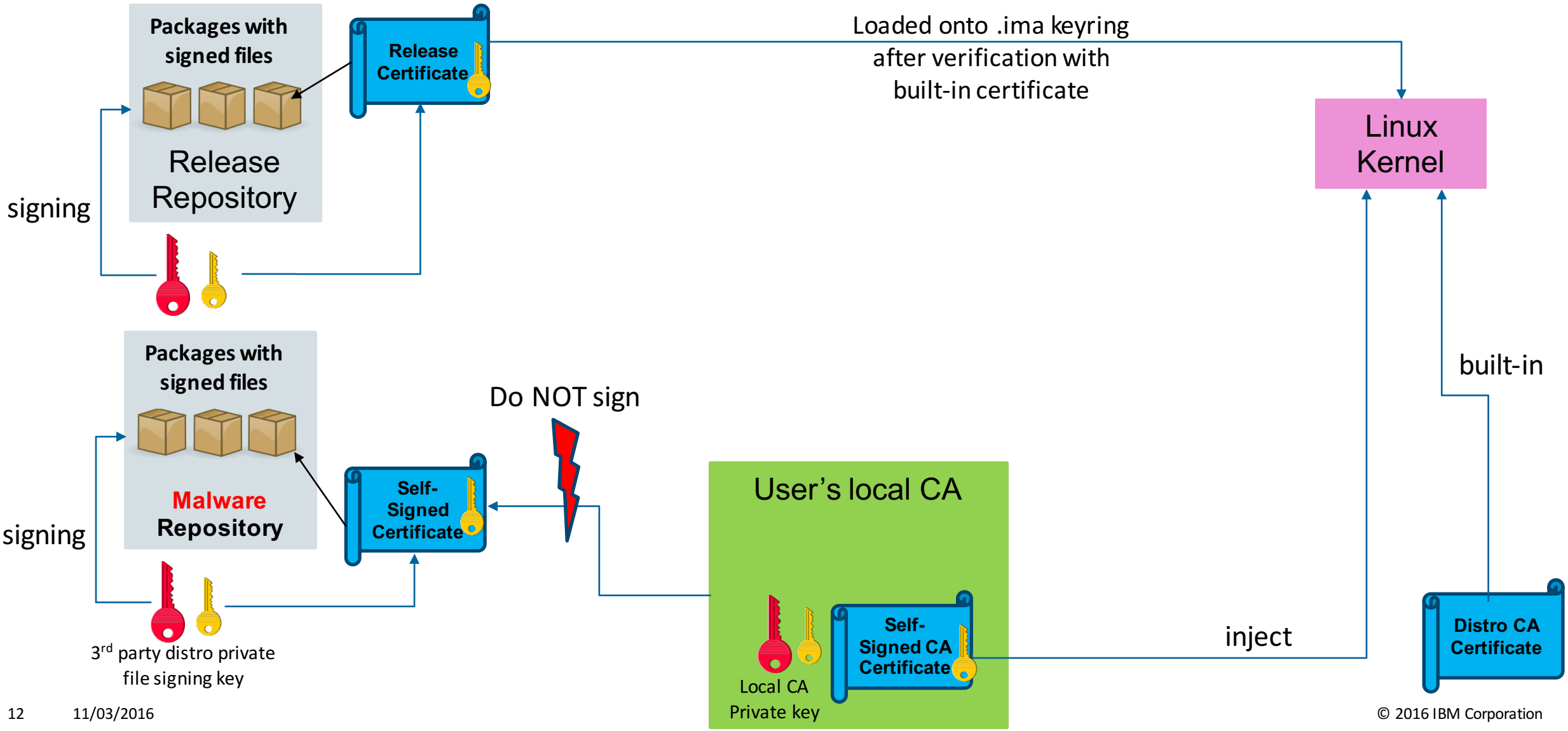
Public key

Private key

Keys and Certificates: Distro + 3rd Party Repo



Keys and Certificates: Distro + 3rd Party Repo



New Components and Packages

RPM-based (Fedora)

- Mirroring + signing: rpmmirror (package) [new]
- Uses rpmsign for signing files in packages
- Mirror is equivalent to original RPM mirror with file signatures + additional key file package:
  - fedora-24-ima-signing-key-0.1-1.fc24.noarch.rpm
    → install 'manually'
- Additional repository with extra packages
  - Linux 4.7.x with built-in certificate + IMA fix patches
  - dracut-integrity
  - ima-appraisal-setup
  - rpm (with latest modifications)
  - Rsync, systemd

Debian-based (Ubuntu)

- Mirroring + signing: secdebmirror (package) [new]
- Uses debsign [new] for signing files in packages
- Mirror is equivalent to original Debian mirror with file signatures + additional key file package:
  - xenial-ima-signing-key_0.1-1_all.deb
    → install 'manually'
- Additional repository with extra packages
  - Linux 4.8.0 with built-in certificate + IMA fix patches
  - initramfs-tools-ima
  - ima-appraisal-setup
  - apt + dpkg (with xattr support patches)

Debian <distro release>-ima-signing-key package

```
# dpkg -c xenial-integrity-ima-signing-key_0.1-1_all.deb
drwxr-xr-x  root/root    0 2016-10-14  12:53  ./etc/keys/ima/
-rw-r--r--  root/root  817 2016-10-14  12:53  ./etc/keys/ima/xenial-integrity-ss-cert-b0e9d679.der
-rw-r--r--  root/root  814 2016-10-14  12:53  ./etc/keys/ima/xenial-integrity-distro-cert-b0e9d679.der
drwxr-xr-x  root/root    0 2016-10-14  12:53  ./etc/pki/deb-gpg-debmirror/
-rw-r--r--  root/root  955 2016-10-14  12:53  ./etc/pki/deb-gpg-debmirror/DEB-GPG-KEY-xenial-integrity
```

Self-signed certificate to be certified by local CA

Key ID

Release certificate loaded onto .ima keyring

Fedora <distro release>-ima-signing-key package

```
# rpm -qlp fedora-integrity-24-ima-signing-key-0.1-1.fc24.noarch.rpm
/etc/keys/ima/fedora-integrity-24-distro-cert-6e6c1046.der
/etc/keys/ima/fedora-integrity-24-ss-cert-6e6c1046.der
/etc/pki/rpm-gpg-rpmmirror
/etc/pki/rpm-gpg-rpmmirror/RPM-GPG-KEY-fedora-integrity-24
/etc/pki/rpm-gpg-rpmmirror/RPM-GPG-KEY-fedora-integrity-24-aarch64
/etc/pki/rpm-gpg-rpmmirror/RPM-GPG-KEY-fedora-integrity-24-armhfp
/etc/pki/rpm-gpg-rpmmirror/RPM-GPG-KEY-fedora-integrity-24-i386
/etc/pki/rpm-gpg-rpmmirror/RPM-GPG-KEY-fedora-integrity-24-ppc64
/etc/pki/rpm-gpg-rpmmirror/RPM-GPG-KEY-fedora-integrity-24-ppc64le
/etc/pki/rpm-gpg-rpmmirror/RPM-GPG-KEY-fedora-integrity-24-s390
/etc/pki/rpm-gpg-rpmmirror/RPM-GPG-KEY-fedora-integrity-24-s390x
/etc/pki/rpm-gpg-rpmmirror/RPM-GPG-KEY-fedora-integrity-24-x86_64
```

Key ID

Release certificate loaded onto .ima keyring

Self-signed certificate to be certified by local CA

Demo: IMA Measurement & Appraisal Policy

```
[…]
# ISOFS_MAGIC
dont_measure  fsmagic=0x9660
dont_appraise  fsmagic=0x9660
# CGROUP_SUPER_MAGIC
dont_measure  fsmagic=0x27e0eb
dont_appraise  fsmagic=0x27e0eb
# MSDOS_SUPER_MAGIC
dont_appraise  fsmagic=0x4d44

measure  func=BPRM_CHECK
measure  func=FILE_MMAP  mask=MAY_EXEC
measure  func=FILE_CHECK  mask=MAY_READ  uid=0

appraise  func=BPRM_CHECK  fowner=0  appraise_type=imasig
appraise  func=MMAP_CHECK  fowner=0  appraise_type=imasig

hash  fowner=0
```

File systems whose files will not be appraised or measured

Measure libraries, other mmap'ed executables, and files opened for reading by root

Appraise libraries and other mmap'ed executables owned by root and require them having a signature

New: Create hashes on all files created by root

Reference: https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/ABI/testing/ima_policy

Demo: Files and Commands

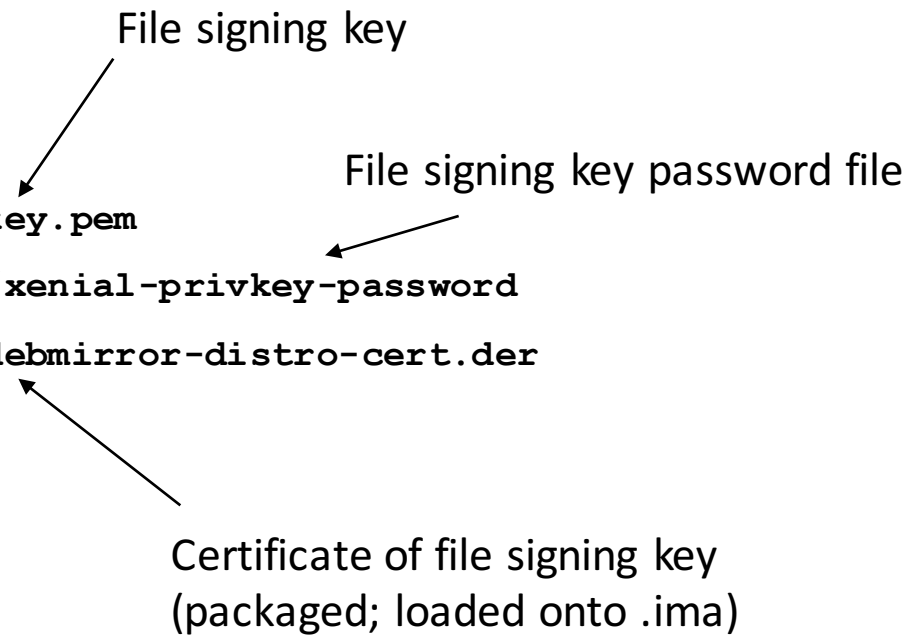
- IMA policy file loaded by initramfs/initrd
  - Fedora: `/etc/sysconfig/ima-policy`
  - Ubuntu: `/etc/default/ima-policy`
- IMA certificates: `/etc/keys/ima/*.der`
- Display signature on files: `getfattr -m ^security -e hex -dump <filename>`
- Show .ima keyring content: `sudo keyctl show %keyring:.ima`
- Load key onto .ima keyring: `sudo evmctl import <cert file> <keyring id>`

Demo

secdebmirror

- secdebmirror
  - Program for mirroring a Debian mirror
  - Can add signatures to files in all Debian packages (debsign)
  - Written in bash scripting language
  - Builds on reprepro
- Example config file:

```
Basedir: /root/pub-repo/ubuntu/xenial
Origin: http://archive.ubuntu.com/ubuntu/
Signingkey: /root/secdebmirror-configs/xenial-privkey.pem
SigningkeyPasswordFile: /root/secdebmirror-configs/xenial-privkey-password
Certificate: /root/secdebmirror-configs/xenial-secdebmirror-distro-cert.der
Threads: 8
```



rpmmirror

- rpmmirror
  - Program for mirroring an RPM mirror
  - Can add signatures to files in all RPMs in the mirror (rpmsign)
  - Written in bash scripting language
- Example config file:

```
Basedir: /home/rpmmirror/pub-repo/fedora/linux/releases/24
Origin: http://mirror.math.princeton.edu/pub/fedora/linux/releases/24/
Signingkey: /root/rpmmirror-configs/fc24-privkey.pem
Certificate: /root/rpmmirror-configs/fedora-24-rpmmirror-distro-cert.der
Distro: fedora
Threads: 4
Architectures: x86_64
Products: Server Workstation Everything
GPG-Name: Fedora-24-rpmmirror
```

File signing key

Certificate of file signing key (packaged; loaded onto .ima)

Package signing key

Building ISOs, Installing & Running a System

Fedora 24

- ISO built with modified kickstart scripts
- RPM packages fetched from internal Fedora mirror
- Background script adjusting installed system (did not modify Anaconda)
- Kernel boot parameters on installed system: ima_appraise_tcb ima_tcb
  - EVM is activated
- Updating packages using dnf with (patched) rpm possible

Ubuntu 16.04

- ISO built with collection of bash scripts
- Debian packages fetched from internal Debian mirror
- Background script adjusting installed system (did not modify Ubiquity)
- Kernel boot parameters on installed system: ima_appraise_tcb ima_tcb
  - EVM is activated
- Updating packages using apt-get with (patched) apt-get possible

Mutable and Immutables Files in Packages

- Immutable files need signatures: executables, libraries, scripts, some data files
  - Config/Datafile examples: game level files, firmware files
- Mutable files must **not** have signatures: Configuration files, cache files, some other data files
  - Config/Datafile examples: /etc/hosts, /etc/ld.so.cache, game score files
  - Exception: /etc/rc.local, IMA policies, …
- Challenges with packaging
  - Cache files need to be marked as config files (=mutables)
  - Packaging errors: applications may fail to install or run if mutable files have signatures
- Package configuration (mutable) files properly:
  - RPMs: %config in spec
  - DEBs: Debian/conffiles
  - May be necessary introduce identifier for mutable files: game score file

File Signing Strategies

RPMs

- rpmsign tool adds signatures to all files
- RPM extraction requires IMA plugin
- IMA plugin applies signatures to all files except
  - config files (%config)
  - Exception: executable config files

Debian Packages

- debsign tool adds signatures to all files with exception:
  - Files under /etc are only signed if executable
  - Files in ‘conffiles’ are not signed unless executable
- Dpkg & apt: Apply signatures on all files for which there are signatures

Caveats

- Don’ts:
  - Don’t run your system without IMA Appraisal activated – file hashes will not be created
    - Also: Currently we need a patched kernel
  - Don’t install from repositories where packages don’t have signatures
    - Package may not install; post installation scripts may not run
    - Applications will not run
  - Don’t blindly sign applications or packages from unknown origins …

Current Status (11/2/2016)

RPM Based

- Kernel: Latest patches to be posted
  - Fedora needs to enable several CONFIG options related to IMA, EVM etc.
- Dracut: latest patches on mailing list
- rpmmirror: legal review
- ima-appraisal-setup: legal review
- ima-evm-utils
  - Fedora should update to 1.0
- rpm: all patches upstreamed
- Kickstart scripts: legal review
- Public mirrors: not available
- Others:
  - attr: on mailing list
  - rsync, system: local testing

Debian Based

- Kernel: Latest patches on mailing list
- initramfs-tools: not posted yet
- secdebmirror: legal review
- ima-appraisal-setup: legal review
- ima-evm-utils
  - Ubuntu should update to 1.0
- dpkg, apt, libarchive: patches on mailing lists
- ISO build scripts: legal review
- Public mirrors: not available
- Others:
  - attr: on mailing list
  - rsync, system

Conclusion / Future Work

- Conclusion
  - We built a secure Linux system that enforces the verification of file signatures
  - Base installation from ISO images; package updates from mirrored .deb & .rpm repositories
- Our next steps
  - Make new packages publicly available:
    - rpmmirror, secdebmirror, ima-appraisal-setup, initramfs-tools & Dracut IMA extension, …
    - Build scripts for installation ISOs
    - Contents of temporary mirror…
- Future work with community:
  - More tools to simplify usage
  - Build repos with packages containing file signatures
  - Extensions needed for installers: Ubiquity, Anaconda, …
  - Support for hardware used for signing or CAs (Yubikey, TPM, etc.)
  - Support for other package formats: Snappy, APK, python pip packages, …
  - Key Granularity (→ BoF)
  - Testing, testing, testing → adjusting code
    - dealing with xattrs: dracut, libattr, rsync, …
    - System boot: systemd, …

File Signatures Needed! -- BoF

BoF – File Signatures Needed!

- Goals – let's start small:
  - Getting patches accepted (dpkg, apt, libarchive, …)
  - Getting you to try it … once components are available
  - Forming a community around Linux with file signatures
  - Getting Distros to pick it up
  - Making system easier to use: better tools - with GUIs
- Bigger
  - Infrastructure for certificate revocation (untrusted repos, package version)