LSS-EU 2018: Overview and Recent Developments Linux Integrity Subsystem

Mimi Zohar

Linux Integrity Subsystem & Ecosystem

- IMA Overview
  - IMA-measurement, IMA-appraisal, IMA-audit
  - Relationship between EVM & IMA-appraisal
  - Keys & keyrings
- IMA measurement list usage
  - Attestation: compromise detection
  - Integrity analytics, forensics
- Namespacing IMA
- Recent developments, current & future work

Goals

"The goals of the **kernel integrity subsystem** are to:

- **detect if files have been accidentally or maliciously altered**, both remotely and locally,
- appraise a file's measurement against a 'good' value stored as an extended attribute, and
- enforce local file integrity." –IMA Wiki

White paper:
http://downloads.sf.net/project/linux-ima/linux-ima/Integrity_overview.pdf

IMA Features

- **IMA-measurement**: detect if files have been accidentally or maliciously altered
  - Need to know if/when a system has been compromised
  - Pointless to ask the system if it has been compromised
  - Available in Linux 2.6.30
- **IMA-appraisal**: appraise a file's measurement against a "good" value and enforce local file integrity
  - Good value: file hash or signature stored as an extended attribute
  - Signature verification uses keys in specific keyring
  - Available since Linux 3.7
- **IMA-audit**: augment the audit subsystem with IMA records
  - Used to assist with security analytics/forensics
  - Available in Linux 3.7

Example Policy

Simple ima-policy:

```
measure func=KEXEC_KERNEL_CHECK

appraise
  func=KEXEC_KERNEL_CHECK
  appraise_type=imasig

audit func=KEXEC_KERNEL_CHECK
```

- Calculate file hash once
- Add hash to measurement list
- Verify kernel image signature*
- Add audit records

* Requires loading key(s)

```
/* collect - calculate file hash */
rc = ima_collect_measurement();
if (rc != 0) goto out_locked;

if (action & IMA_MEASURE)
  ima_store_measurement();

if (rc == 0 && (action &
  IMA_APPRAISE_SUBMASK))
  rc = ima_appraise_measurement();

if (action & IMA_AUDIT)
  ima_audit_measurement();


out:
  return rc;
```

Secure vs. Trusted Boot*

Secure Boot

Each stage of booting verifies the signature of the next stage before transferring control

Signature verification failure causes the boot to fail

Hardware root of trust is stored in nonvolatile memory

Trusted Boot

Each stage of booting, measures the next stage, adds the measurement to an event log, and extends the TPM, before transferring control

TPM is hardware root of trust

* Some people use the terms “Verified and Measured Boot”

Linux Integrity Subsystem & Ecosystem: IMA-measurement & IMA-appraisal

Secure Boot

Software Distribution with File Signatures

- "Signing Linux Executables for Fun and Security" (2017 LPC Matthew Garrett)
- "File signatures needed!" (2016 LPC talk)

Key Management

- Key granularity
- Blacklisting keys
- Platform keyring

Policies

ima_policy= ["tcb | appraise_tcb | secure_boot | fail_securely"]

Relationship between EVM & IMA-appraisal

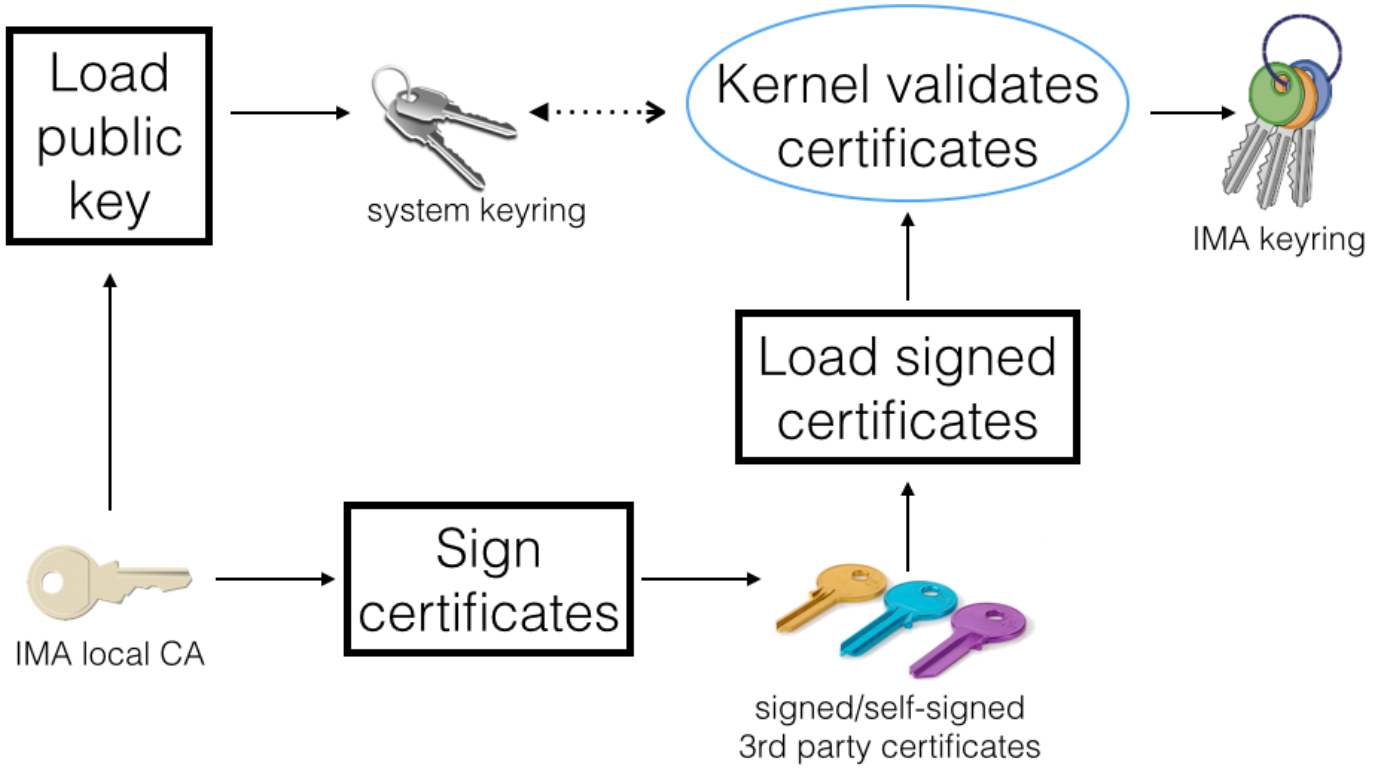
- IMA-appraisal protects file data
  - Good value: file hash or signature stored as an extended attribute
  - Signature verification based on keys in specific keyring (.ima)
- EVM protects file metadata
  - Either HMAC or signature of file metadata
    - HMAC value calculated/verified using EVM key (encrypted key)
    - Signature verification based on keys in specific keyring (.evm)
  - Trusted key: TPM symmetric key
  - Encrypted key: encrypted/decrypted w/trusted key (or user key)
- EVM binds the file metadata to the file data
  - inode: ino, generation, uid, gid, & mode
  - SELinux, smack, apparmor, IMA, & capabilities xattrs (optional)
  - **New** immutable & portable signatures (Matthew Garrett, Google)
    - No inode filesystem specific data
    - Requires security.ima xattr

File hashes, HMAC, and signature usage

- Files may change
  - Requires access permission (eg. DAC, MAC)
  - Must be hashed
  - Are HMAC protected
- "immutable" files may be signed
  - But with which keys, on which keyring?
  - Userspace keyrings: _ima, _evm
    - Keys are normally added to these keyrings in the initramfs, prior to pivoting root
  - Kernel trusted keyrings: .ima, .evm
    - Keys added to trusted keyrings must be signed by keys on the .builtin_trusted_keys or, if enabled, .secondary_trusted_keys

IMA-appraisal: local CA (the linchpin)

Extending the secure boot certificate and signature chains of trust to the OS

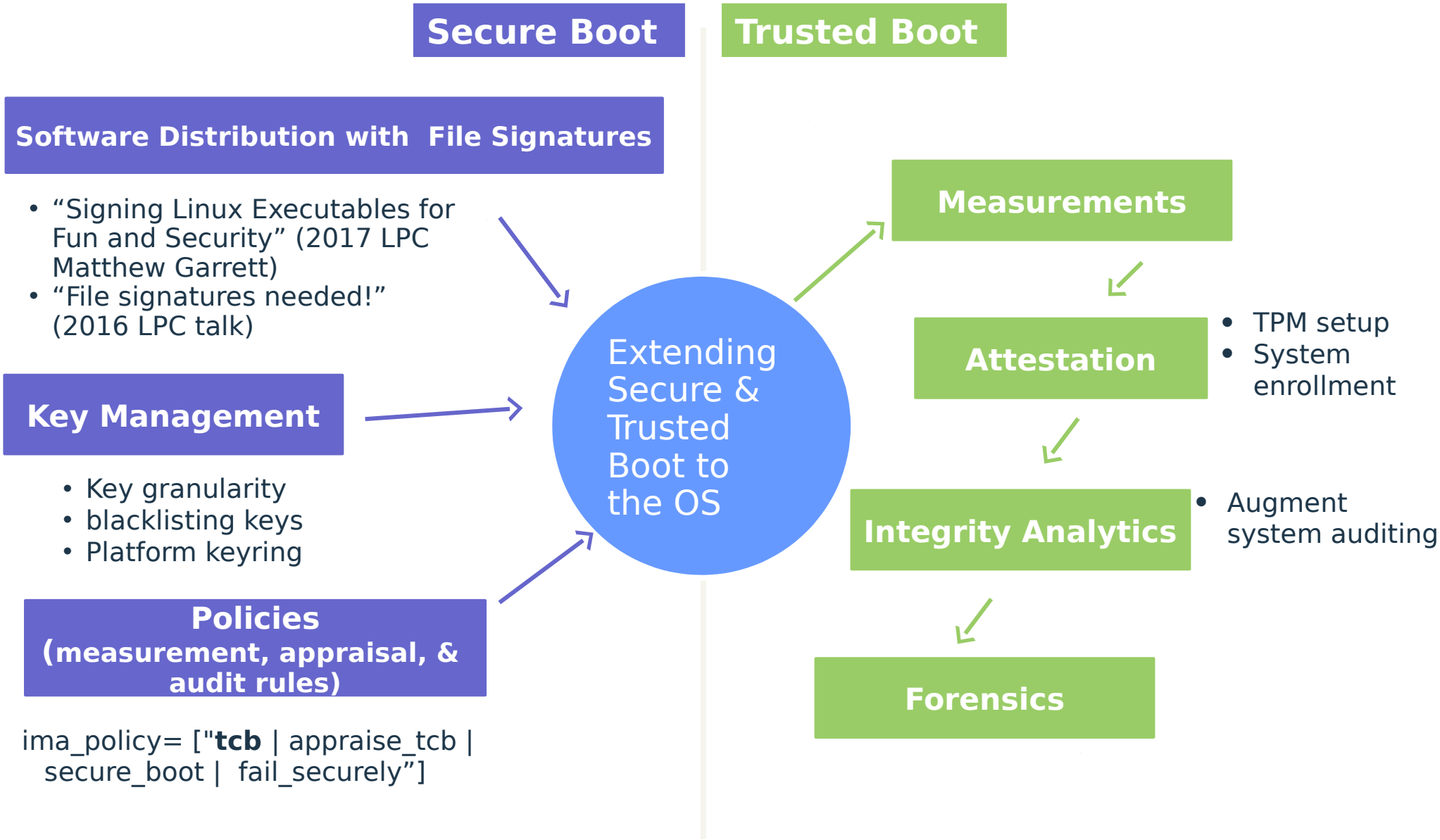


Methods for loading IMA Local-CA public key on to the .builtin_trusted_keys keyring

- Build keys into the kernel image
- Reserve space in the kernel image for IMA local-CA. Post build, install key and resign kernel image
  (Mehmet Kayaalp, partially upstreamed)

- Or, load the firmware database keys onto the .secondary_trusted_keys keyring and configure secondary trusted keyring (not mainline)

Linux Integrity Subsystem & Ecosystem: IMA-measurement & IMA-appraisal



IMA measurement list

Too few measurements, system integrity is unknown

Too many measurements, memory pressure on system and possible performance issues

- Identify what to measure
- Changes are coming, but do they all retain the existing security guaranties?
  - TPM 2.0 crypto agile measurement list support
  - Exporting the measurement list
  - Only store "unknown" measurements based on an in kernel white list
  - Request the file hash from the filesystem, instead of calculating it

Remote Attestation: Compromise Detection

- What is “**remote attestation**”?
  - Local machine reports to a remote server the exact software running on the platform
  - From boot firmware through OS load and into applications
- What is an **attestation quote**?
  - Digitally signed report
- Why can we trust it?
  - Chain of measurements stored on crypto hardware (TPM)
  - Signing key (attestation key, quote key) never leaves the TPM

IBM Attestation Client/Server & TSS

- Attestation Client (Open sourced 2016)
  - **Pure C implementation**
  - Pre-OS event log and IMA measurement list
  - Supports both TPM 2.0 and now TPM 1.2
  - TCG enrollment protocol using EK certificate as root of trust

- IBM TSS (open sourced 2015)
  - Designed for **ease of understanding, ease of use**
  - Support all TPM 2.0 requirements with single function call and now supports TPM 1.2
  - Command line utility for each TPM command
    - Sample code for application programmers
    - Scriptable for rapid prototyping and TSS regression test
  - Linux, Windows, AIX, Raspian, x86, Power, ARM
  - HW and SW TPM support built in

TPM 2.0 Integrity Measurement Architecture (IMA) Reports

Ken Goldman, IBM Research, kgoldman@us.ibm.com
Attestation Server: cainl.watson.ibm.com

Integrity Measurement Architecture (IMA) Reports for cainl.watson.ibm.com

Timestamp	Event	Event	Signed	Key	Sig	File Name
2018-06-27 15:14:42	1082	✓	✓	✓	✗	/usr/bin/trojan_badsig
2018-06-27 15:14:42	1081	✓	✓	✗		/usr/bin/trojan_badkey
2018-06-27 15:14:42	1080	✓	✗			/usr/bin/trojan_nosig
2018-06-27 15:14:42	1079	✗				
2018-06-27 15:14:42	1078	✓	✓	✓	✓	/usr/bin/kill
2018-06-27 15:14:42	1077	✓	✓	✓	✓	/usr/libexec/gvfsd-metadata
2018-06-27 15:14:42	1076	✓	✓	✓	✓	/usr/lib64/tracker-1.0/extract-modules/libextract-msoffice.so
2018-06-27 15:14:42	1075	✓	✓	✓	✓	/usr/lib64/security/pam_xauth.so
2018-06-27	1074	✓	✓	✓	✓	/usr/lib64/security/pam_fprintd.so

IMA: Running on the Host, in VMs, and in Containers

Level 2
Container Integrity

Docker | Docker | Docker

Attestation

Container Owner

Level 1
Guest VM Integrity

VM_1 | VM_2 | VM_3

KVM

Attestation

Virtual Machine Owner

Level 0
Infrastructure Integrity

OS

UEFI / BIOS, Petitboot

Attestation

Bare Metal Infrastructure Owner

Staging IMA namespacing

- **IMA-audit**: differentiating between what ran on the system vs. in a namespace, and between namespaces.
  - Define how to initialize IMA namespaces
  - Extend integrity audit messages with container id
- **IMA-measurement**: per IMA namespace measurement list
  - Define per namespace: policy, measurement list
  - Requires virtualizing securityfs, prevent securityfs information leakage, and optionally vTPM support
- **IMA-appraisal**: signature verification on a per IMA namespace basis
  - Per namespace IMA keyring
  - Permit root in the namespace to write security xattrs
  - Requires: per namespace xattrs (scaling issues)

Recent Developments

New Features

- EVM (file metadata) new features (M. Garrett, Google)
  - Portable & Immutable EVM signatures
  - Runtime support for defining and including additional EVM protected security attributes
  - Support for larger EVM digests

- IMA (file data)
  - Build time IMA policy, persists after loading a custom policy
  - Run time enabled IMA architecture specific policy (N. Jain, E. Richter IBM LTC)

Closed IMA-Measurement, IMA-appraisal, & IMA-audit gaps

- Re-measuring, re-appraising, & re-auditing files on filesystems without i_version support (Sascha Hauer, pengutronix)

- Prevent kexec'ing unsigned kernel image (kexec_load syscall)
- Prevent loading unsigned firmware (sysfs)

- FUSE file systems
  - Unprivileged mounted FUSE filesystems is always untrusted
  - Privileged mounted FUSE filesystems
    - Default: re-measure, re-appraise, re-audit on each file access
    - Optional: builtin "fail-securely" IMA policy
    - **New Discussion**: support file change detection (Matthew Garrett, Google)

Outstanding problems resolved

- Re-introduced an IMA specific lock to resolve the i_rwsem lockdep (Dmitry Kasatkin, Huawei)
- Major TPM performance improvements (N. Jain, IBM LTC)
- Disambiguated IMA audit records (S. Berger, IBM Research)

Testing!

- IMA Linux Test Program(LTP) tests updated (Petr Vorel, Suse)
- evmtest: a standalone IMA test framework (David Jacobson, IBM Research summer intern)
  - Direct testing of currently running kernel image
  - Designed for integration with LTP and xfstests in mind

Current and future work

- **Continue closing measurement/appraisal gaps**: initramfs, eBPF, mmap, interpreters
- Storing firmware keys on the **Platform keyring**
- **Regression testing**: ima-evm-utils, Linux Test Program(LTP), xfstests
- **Simplify usage**: sample policies, updating documentation
- **Key management**: black lists, revocation
- **Directory filename protection support**
- Re-work the IMA measurement **list memory allocation**
- Support **other file signature verification methods** (eg. appended signatures, fs-verity)
- **Hash agile measurement list**

How can you help?

- Review patches*
- Participate in new feature discussions*
- evmtests: extend the testing framework with additional tests
- Simplify usage: sample IMA policies, updating documentation
- Don’t introduce new IMA-measurement, IMA-appraisal, or IMA-audit gaps.

* Mailing list: linux-integrity@vger.kernel.org
Archive: https://lore.kernel.org/linux-integrity/

Thank you!

- Special thanks to my colleagues:
  - Stefan Berger, Ken Goldman, Guerney Hunt, Elaine Palmer, Mehmet Kaylaap, Hani Jamjoom, [David Safford, Dimitrios Pendarakis] (IBM Research)
  - George Wilson, Nayna Jain, Thiago Bauermann (IBM LTC)
- To all of you
  - For the new IMA & EVM features
  - For all the "minor" bug fix patches
  - For the help with updating and packaging ima-evm-utils

Questions?

Remote Attestation: Compromise Detection

- The IBM TSS handles the following, hidden from the caller:
  - HMAC, password, and policy sessions
  - Session and HMAC, key calculations, including bind and salt sessions
  - HMAC generation and verificaton (including cpHash and prHash)
  - Parameter encryption and decryption, XOR and AES
  - Nonces and nonce rolling
  - Session continue flag
  - TPM 2.0 "Name" and bind session tracking
  - Different session hash algorithms
  - Marshaling, unmarshalling
  - Communications with the TPM, Windows, and Linux, HW and SW TPM

IBM TSS: API

```
TPM_RC   TSS_Execute( TSS_CONTEXT   *tssContext,
RESPONSE_PARAMETERS *out,
COMMAND_PARAMETERS *in,
EXTRA_PARAMETERS       *extra,
 TPM_CC commandCode,  . . .)
```

out: The standard TPM2 Part 3 response parameter
in: The standard TPM2 Part 3 command parameter
extra: Some commands (only two so far) require extra parameters
commandCode: The standard TPM2 Part 2 command code.
... : A list of session 3-tuples, of the form
- TPMI_SH_AUTH_SESSION sessionHandle,
- const char *password,
- unsigned int sessionAttributes

The list is terminated with (TPM_RH_NULL, NULL, 0)