Integrity Protection Solutions for Embedded Systems

Dmitry Kasatkin

Samsung Open Source Group
Samsung Research UK, Finland branch

FOSDEM 2014
Brussels, Belgium, February 1 – 2, 2014

Agenda

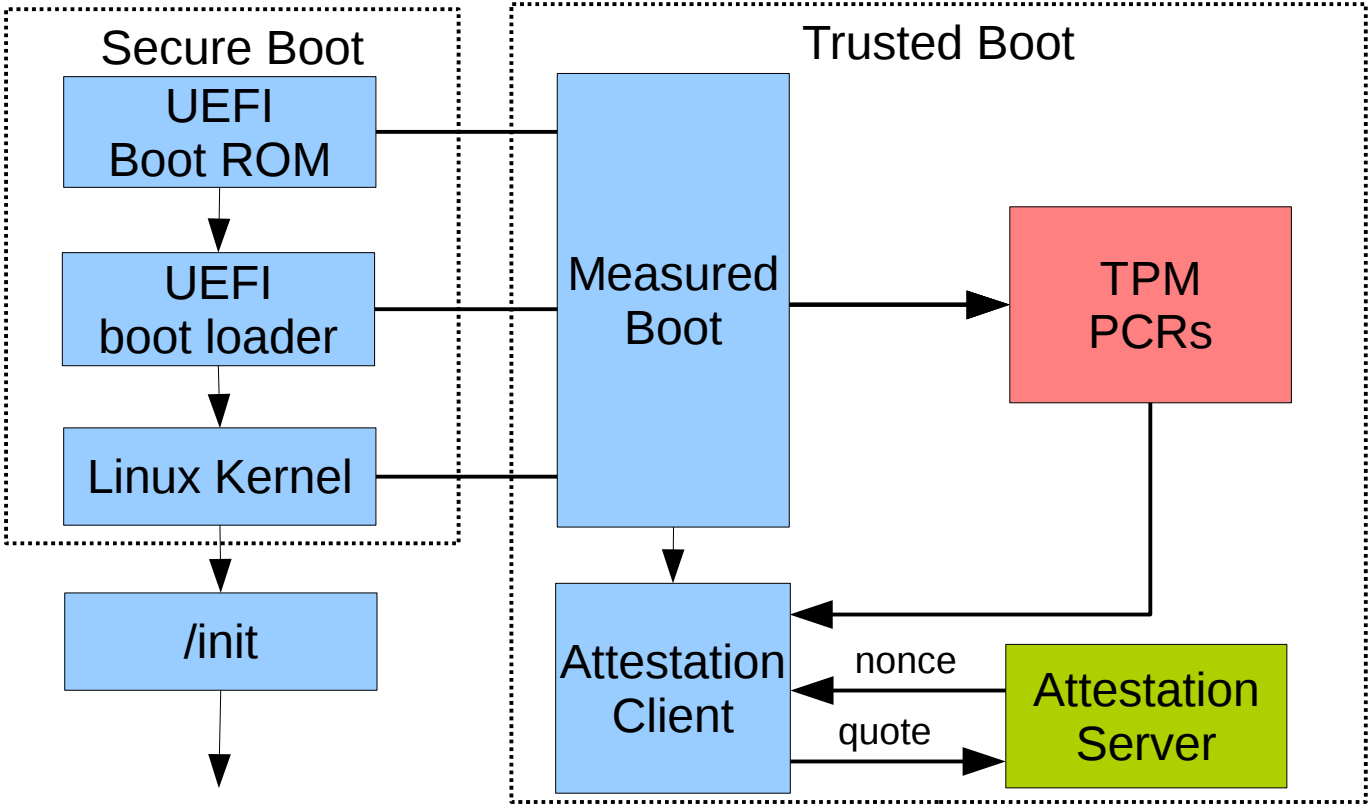
- What is integrity protection?
- Pre-OS integrity protection
- OS integrity protection
  - Block-level integrity protection
  - Linux Integrity subsystem
- Summary
- Q&A

What is integrity protection?

- Runtime system integrity is protected by Access Control mechanism, such as DAC and MACs.
- Assumes trustworthiness of the access control/security related metadata
- Integrity protection ensures that offline modification of the data will not remain undetected and access to such data will be forbidden
- Was achieved by file system encryption

Pre-OS Integrity Protection

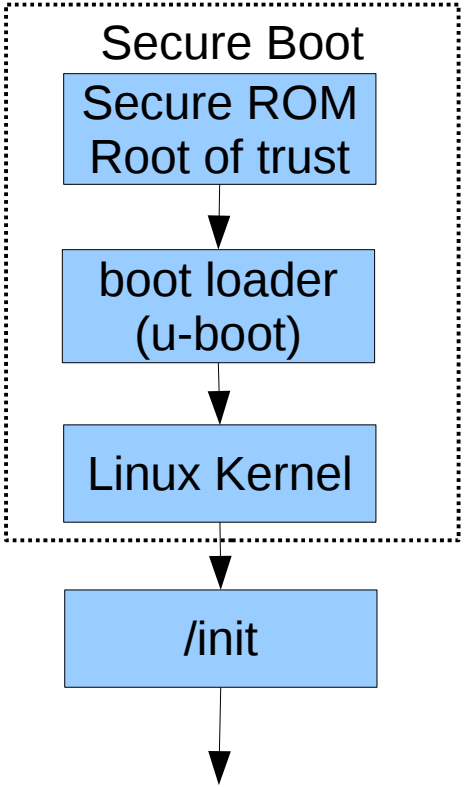
UEFI x86 boot process



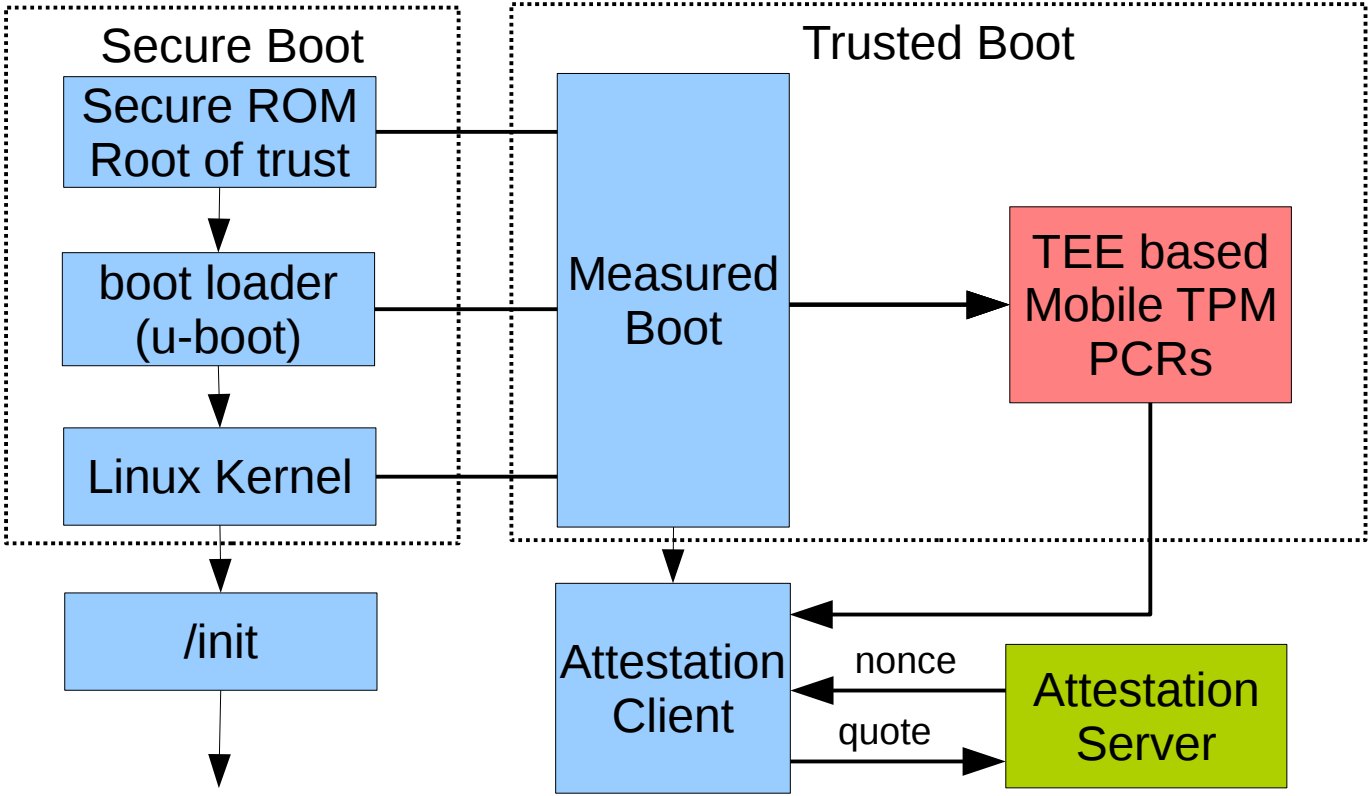
UEFI x86 boot process

- Trusted Boot (TCG, Pre-UEFI)
  - TPM based measurements
  - Does not prevent booting if measurement is wrong
- UEFI Secure Boot – PreOS boot
  - Root of trust in the firmware (UEFI)
  - Prevents modified firmware and boot loader from running
  - Boot loader verifies Linux kernel
- OS Trusted/Secure Boot?
  - Linux kernel verifies kernel modules
  - **What about user space??**

Embedded System Boot – not-connected



Embedded System Boot – for connected



Secure/Trusted u-boot (~2013.07)

- Secure boot extensions since 2013.07
  - Create FIT kernel image (Flattened uImage Tree)
    - mkimage -f kernel.its kernel.itb
  - Create signed FIT kernel image
    - mkimage -f kernel.its -k /path/to/keys -K u-boot.dtb kernel.itb
  - Resign
    - mkimage -F -k /path/to/keys -K u-boot.dtb kernel.itb
- TPM support
  - TPM library
  - Drivers for common TPMs

Signature in FIT

```
{
     description = "Simple kernel / FDT configuration";
     images {
          kernel@1 {
               data = /incbin/("../vmlinuz-3.13.0");
               kernel-version = <1>;
               signature@1 {
                    algo = "sha1,rsa2048";
                    key-name-hint = "dev";
               };
          };
          fdt@1 {
               description = "snow";
               data = /incbin/("exynos5250-snow.dtb");
               type = "flat_dt";
               arch = "arm";
          };
     };
};
```

Block level Integrity protection Approaches

dm-verity

- Transparent block-level integrity protection solution for **read-only** partitions
- dm-verity is a device mapper target
- Uses hash-tree
  - Calculates a hash of every block
  - Stores hashes in the additional block and calculates hash of that block
  - Final hash – root hash – hash of the top level hash-block
  - Root hash is passed as a target parameter
- Used in ChromeOS to protect read-only partition
- Update can be done only by overwriting entire partition

dm-integrity

- Transparent block-level integrity protection solution for **RW** partitions
- dm-integrity is a device mapper target
  - virtual block device on the top of real
- Maintains HMAC for every block in special integrity store
  - may be the same or different block device
  - additional space
- Verify HMAC on every read-request and update HMAC on every write-request (BIO request)
- Keeps a cache of LRU integrity metadata for performance purpose

Linux Integrity Subsystem

Linux Integrity Subsystem

- **Allows to extend Trusted and Secure Boot to user space**
- Linux integrity subsystem is the **Policy Based** VFS level integrity protection
- Located under <linux>/security/integrity
- Provides several integrity functions such as
  - Collect, store, attest, apprise, protect, audit
- Consists of following components:
  - IMA – Integrity Measurement Architecture module
    - IMA-measurement
    - IMA-appraisal
    - Directory integrity verification extension
  - EVM – Extended Verification Module
  - Digital signature verification support

IMA/EVM hooks

- IMA hooks
  - int ima_bprm_check();
  - int ima_file_check();
  - int ima_file_free();
  - int ima_file_mmap();
  - int ima_module_check();
- EVM hooks
  - int evm_inode_setattr();
  - void evm_inode_post_setattr();
  - int evm_inode_setxattr();
  - void evm_inode_post_setxattr();
  - int evm_inode_removexattr();
  - void evm_inode_post_removexattr();

IMA-measurement

- Since 2.6.30 (CONFIG_IMA)
- Can be used to remotely attest system's runtime integrity
- **Collect**: measure a file's content before it is accessed using cryptographic **hash**
- **Store**: add the measurement to the **runtime measurement list**, and if TPM is present, extend the IMA PCR-10
  - Incorrect value may “lock” TPM secrets such as keys
- **Attest**: if TPM is present, sign IMA PCR value to allow remote validation of the measurement list
  - Requires attestation server (challenger) to maintain hash database
- Also calculates and stores boot aggregate value over the TPM PCRs 0 - 7

IMA measurement list – original format

- /sys/kernel/security/integrity/ima/ascii_runtime_measurements
- Format: PCR template-hash template filedata-hash pathname
- Original SHA1 format:

  10 992676726c30b83e352f7bdb75e1c4dc9bab2067 ima 1f50f71b43752cd541a851a585cba3580902e7a9 /sbin/init

IMA measurement list – new format

- PCR template-hash template **algo**:filedata-hash pathname **file-signature**
- Larger hash algo:

  10 992676726c30b83e352f7bdb75e1c4dc9bab2067 **ima-ng** **sha256:**c023470c0fc8aa1dbb95504d5af5d46cad94e8bf5eea8e0ab0eeff7a7fe1697a /sbin/init
- Signatures:

  10 992676726c30b83e352f7bdb75e1c4dc9bab2067 **ima-sig** 1f50f71b43752cd541a851a585cba3580902e7a9 /sbin/init 030202db1ff72a008016c593387220a2adda990969d87a56a8a24eece51e3689fd229c4c56e7fddd4eb99f360c2ee3ff0f6344de24ecd3263f4c7a74ac6498403d7ce9e9865e4d2f32522de79e96d0cb265d5b2ab8fe54953ce53d5e59a51460f67d18e2cbacb4765ea97f2d9cdd2065816d50fb74e631efd4c2e07c72c01fd9b0f9e3efc6d91a789d

EVM – extended verification module

- Since 3.2 (CONFIG_EVM)
- **Protect**: protects integrity of file (extended) attributes against offline modification
  - attributes: ino, uid, gid, mode
  - extended attributes: security.{ima,SMACK64,selinux}
- Measures integrity using (keyed) cryptographic hash (hash/HMAC)
- Performs local integrity validation and enforcement against a “good” reference HMAC value
  - 'security.evm' extended attribute
  - May contain HMAC or signature

Digital signature extension

- Since 3.3 (CONFIG_INTEGRITY_SIGNATURE)
- Protects file attributes using digital signatures
  - security.evm may hold signature instead of hmac
  - signature is replaced with hmac on successful verification
- EVM signatures may be used
  - When there is no possibility to use device-specific HMAC key during flashing/copying
    - No special flashing/update mode (fastboot on Android)
  - When raw FS image needs to be created for use on multiple devices
    - HMAC key is device-specific and cannot be used during image creation

IMA-appraisal

- Since 3.7 (CONFIG_IMA_APPRAISE)
- **Appraise**: enforce local integrity validation of a file measurement against a “good” reference value
  - 'security.ima' extended attribute may hold hash or signature
  - signature is never replaced with hash – file is immutable
  - Protected by EVM
- In other words, allows to protect file data from offline modification
- IMA signatures may be used
  - To protect immutable files from runtime modification
  - To perform remote attestation without maintaining hash-database

IMA policy

```
# see <linux>/Documentation/ABI/testing/ima_policy
# SYSFS_MAGIC
dont_measure fsmagic=0x62656572
dont_appraise fsmagic=0x62656572
# DEBUGFS_MAGIC
dont_measure fsmagic=0x64626720
dont_appraise fsmagic=0x64626720
……
measure func=BPRM_CHECK
measure func=FILE_MMAP mask=MAY_EXEC
measure func=FILE_CHECK mask=MAY_READ uid=0
appraise obj_user=sig_t func=FILE_CHECK appraise_type=imasig
appraise fowner=1001 appraise_type=imasig
appraise fowner=0
```

Directory & special files integrity protection

- Work in progress (CONFIG_IMA_DIRECTORIES)
- New hooks for directory/special files integrity verification in fs/namei.c
  - ima_dir_check(), ima_dir_update(), ima_link_check()
- Directory measurement is a hash over directory content
  - List of (inode number, file name) tuples
- Symlink measurement
  - Hash of the target path
- Device node measurement
  - Hash over MAJOR:MINOR
- Hash is also stored in 'security.ima'
- No EVM changes are required for this

ima-evm-utils

- Use of digital signatures requires user-space tools
- ima-evm-utils (evmctl)
  - Sign file metadata and content
    - evmctl sign –imahash foo
    - evmctl sign –imasig foo
  - Verify (for testing purpose)
  - Import public keys into the kernel keyring
    - evmctl import /path/to/key
  - Supports password protected private keys

Example: initramfs script

```
grep -v "^#"  /etc/ima_policy >/sys/kernel/security/ima/policy      # load IMA policy

keyctl add user kmk "testing123" @u
keyctl add encrypted evm-key "load `cat /etc/keys/evm-key`" @u      # import EVM HMAC key

ima_id=`keyctl newring _ima @u`
evmctl import /etc/keys/pubkey_evm.pem $ima_id                      # import IMA public key

evm_id=`keyctl newring _evm @u`
evmctl import /etc/keys/pubkey_evm.pem $evm_id                      # import EVM public key

echo "1" > /sys/kernel/security/evm                                 # enable EVM
```

Example: labeling with signatures

Label one file:

$ echo Hello >foo

$ sudo evmctl sign --imahash foo

$ getfattr -e hex -m security -d foo

file: foo

security.evm=0x030155475e4e0000bc16a96303fd3e7901040060bab44648764dca46ad71827a48c3e171b7e9444b47b79b7bd7c7f1783852be9b4f038f2c1dd57320b257619b9fa3a9cadea2c679faf83a9755f2a015995ec43332fdedcc2c72cb87f2eb25a8ef524c3ec78134aaa5b6dd18c8c1bf5e16d886a03dd36587aa927e07154c0009cdaf71c1fcbc37fa15a8bd153ba360bf73bafb

security.ima=0x011d229271928d3f9e2bb0375bd6ce5db6c6d348d9

Label whole file system:

$ evmctl -r sign --imahash /

Summary

- Integrity protection can be implemented in most embedded systems
  - Main requirement is HW root of trust: Secure ROM
- U-boot provides support for secure/trusted boot
- Linux kernel provides support for protecting integrity of the storage
  - Linux Integrity Subsystem
    - policy, local appraisal, remote attestation
  - Block-level integrity protection
    - dm-verity
    - dm-integrity

Links

U-Boot

- http://www.denx.de/wiki/U-Boot

IMA/EVM

- Integrity tree: http://git.kernel.org/?p=linux/kernel/git/zohar/linux-integrity.git
- Dir tree: http://git.kernel.org/?p=linux/kernel/git/kasatkin/linux-digsig.git
- Linux IMA project page: http://sourceforge.net/projects/linux-ima
- Utils: http://sourceforge.net/p/linux-ima/ima-evm-utils/ci/master/tree

dm-integrity

- git://git.kernel.org/pub/scm/linux/kernel/git/kasatkin/linux-digsig.git#dm-integrity

Questions?