Konsulko Group

The role of Linux IMA in Automotive System Security

Petko Manolov
Konsulko Group
petko.manolov@konsulko.com

Konsulko Group

Who Am I

- Software engineer and long time Linux kernel hacker since 1995
- Writes embedded software for work and fun
- 15 years of experience in design and deployment of software security systems
  - Co-Founder and CEO of Nucleus System, a Cable TV / settop box security company
    - “Griffin CAS” product, (Conditional Access System) product, 12 years of unhacked track record
- Chief Security Architect and Managing Director of Konsulko Group Bulgaria

Terminology

- IMA – Integrity Measurement Architecture
- EVM – Extended Verification Module
- TPM – Trusted Platform Module
- xattr - extended attributes, metadata not interpreted by the filesystem

IMA/EVM history

- First introduced in v2.6.30 by IBM
- EVM upstreamed in v3.2
- Support for protecting file metadata based on digital signatures since v3.3
- IMA-appraisal upstreamed in v3.7

IMA/EVM Goals

Konsulko Group

- Integrity: state of being entire, complete, unbroken, free from corrupting influence
- Authenticity: being of established authority for truth and correctness
- Confidentiality: the state of being secret
- Gaining root access is considered a severe breach. With IMA/EVM this is no longer the case.
- Fine granularity of what your files may do or be done to them: read, write, execute, append, etc.

IMA/EVM Internals

- Not all attacks may be reflected, we should at least know when one succeeded
- IMA, trusted computing component, anchored to a TPM
- IMA hashes stored in the file’s extended attributes
- IMA-appraisal, authority signature stored in the file’s extended attributes

Konsulko Group

IMA/EVM Internals, cont

- EVM - hash or signature stored in the xattrs
- When TPM is used

```
PCR    template-hash                                             filedata-hash                                                              filename-hint

10     91f34b5c671d73504b274a919661cf80dab1e127          ima-ng    sha1:1801e1be3e65ef1eaa5c16617bec8f1274eaf6b3                              boot_aggregate
10     8b1683287f61f96e5448f40bdef6df32be86486a          Ima-ng    sha256:efdd249edec97caf9328a4a01baa99b7d660d1afc2e118b69137081c9b689954    /init
10     ed893b1a0bc54ea5cd57014ca0a0f087ce71e4af          ima-ng    sha256:1fd312aa6e6417a4d8dcdb2693693c81892b3db1a6a449dec8e64e4736a6a524    /usr/lib64/ld-2.16.so
10     9051e8eb6a07a2b10298f4dc2342671854ca432b          ima-ng    sha256:3d3553312ab91bb95ae7a1620fedcc69793296bdae4e987abc5f8b121efd84b8    /etc/ld.so.cache
```

- Not all systems have TPM, the kernel support a software version

IMA/EVM Internals, cont

- EVM, protecting the values stored in the extended attributes: .selinux, .ima, etc.
- Prior to VFS accessing a file, its extended attributes are first verified
- IMA one-time policy upload: /sys/kernel/security/ima/policy

Konsulko Group

IMA/EVM Internals, cont

- Simple IMA policy file

```
dont_appraise fsmagic=SYSFS_MAGIC
appraise func=FILE_MMAP mask=MAY_EXEC
appraise func=FILE_CHECK mask=MAY_READ uid=0
```

- Upload the policy

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

IMA/EVM Internals, cont

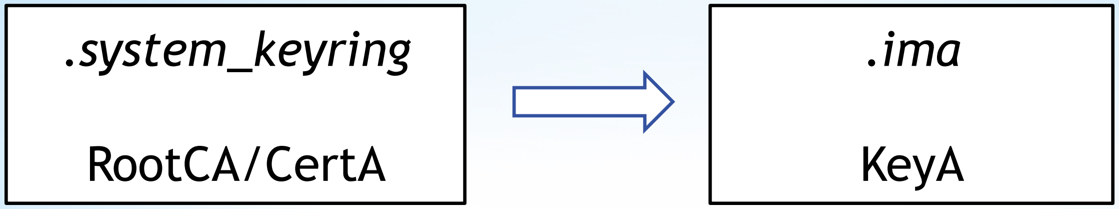
- Private key and certificate generation

1. openssl genrsa -aes256 -out ima.key
2. openssl req -sha256 -new -key ima.key -out ima.csr
3. openssl ca -policy policy_anything -keyfile RootCA.key –cert RootCA.pem -md sha256 -in ima.csr -out ima.pem
4. openssl x509 -outform der -in ima.pem -out ima.x509

Konsulko Group

IMA/EVM Internals, cont

- How do we sign a file
  - ***evmctl*** utility:
    git://git.code.sf.net/p/linux-ima/ima-evm-utils
  - sign the thing:
    evmctl ima_sign --key /path/to/ima.key -a sha256 /path/to/file
  - Import the key:
    evmctl import /path/to/ima.x509 ima_keyring_id



IMA/EVM Internals, cont

- Look at the result

```
root@bender /home/user # getfattr -d -m ima -e hex busybox
# file: busybox
security.ima=0x030204913161a101802cb553e5bbe60f837a9ae1be8905faa7dfec18ff66
46483e2a39d299210d48ab01fb6bc0748823564f293b1ff254b27004475ffb9260d8d7300
04ecc852a5b9f4074b2d5aba14d6bc33dab911a5cc07920e44ada5bf36594cbe0661df2fb
57b81fa192d6572cbf38d4a1f198e8371497bea9ad523d286a5d5bacf2359324b3624090d
564ac9da731998df3a3f6c4f224dd1d2026929193207f8eea32266f4934fefc0354790764d
228a90f8a7864b70aba44aeda04d82d999a91b17ddc15161026973f80d9328c32f80d3de
d31bf4695e10fefc8c9cbb6946cddebed98352c4d4920fb36213d4e66dddc7cdc094e8d3
235981d5d87cb4a35eb95c9c5ad3eae9a8354dac05308a4192d8e479588ef3acef6bc283
be7d24fbaa53fd3a756fdad52a304e79d55ab3e136ed28b05caf91306ce90d230624ba75
951b01273f1b74bae53a385dd996d464f17eda0c640a7349784a61d1c90da9e95196f06a
60a5d46ec1b06b751c556f580f80f8f7cf66c5faa2db62f3114110fde55d015d53fe86855
```

Konsulko Group

IMA/EVM Internals, cont

- Audit log of failed appraisal (dmesg):

```
[  135.906266] integrity: Request for unknown key 'id:913161a1' err -11

[  135.906374] audit: type=1800 audit(1433063480.798:18): pid=1361
uid=1001 auid=4294967295 ses=4294967295 op="appraise_data"
cause="invalid-signature" comm="busybox" name="/home/user/busybox"
dev="sda1" ino=4610 res=0
```

How IMA/EVM Integration, Problems and Challenges

- IMA/EVM kernel code is well documented, but a good User’s Guide does not exist
- IMA/EVM must be handled with great care - it is very easy to get it wrong
- .system_keyring stores the RootCA
- .ima keyring stores the IMA public key or CA
- no intermediate keyring for CA hierarchy - why is this a problem

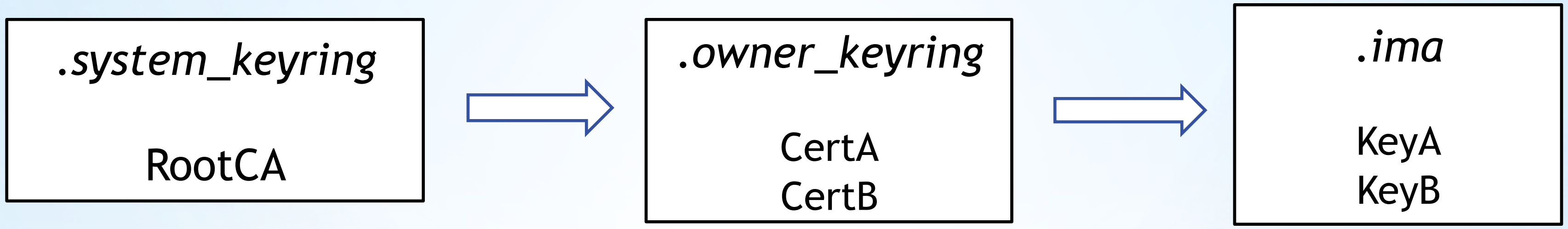
Konsulko Group

Real World Examples

- Use read only filesystem that supports extended attributes – SquashFS
  - Avoid the complexity of EVM
- TPM may not be needed
- Certificate Hierarchy (patches by Konsulko pending)

Roadmap

- Introducing intermediate keyring that may be used to build CA hierarchy in the kernel



- Dynamically loadable IMA policy: proper locking and verifying file's authenticity

- IMA policy creation tool

Conclusions

- Security done right is hard

References

http://linux-ima.sourceforge.net

https://wiki.gentoo.org/wiki/Extended_Verification_Module