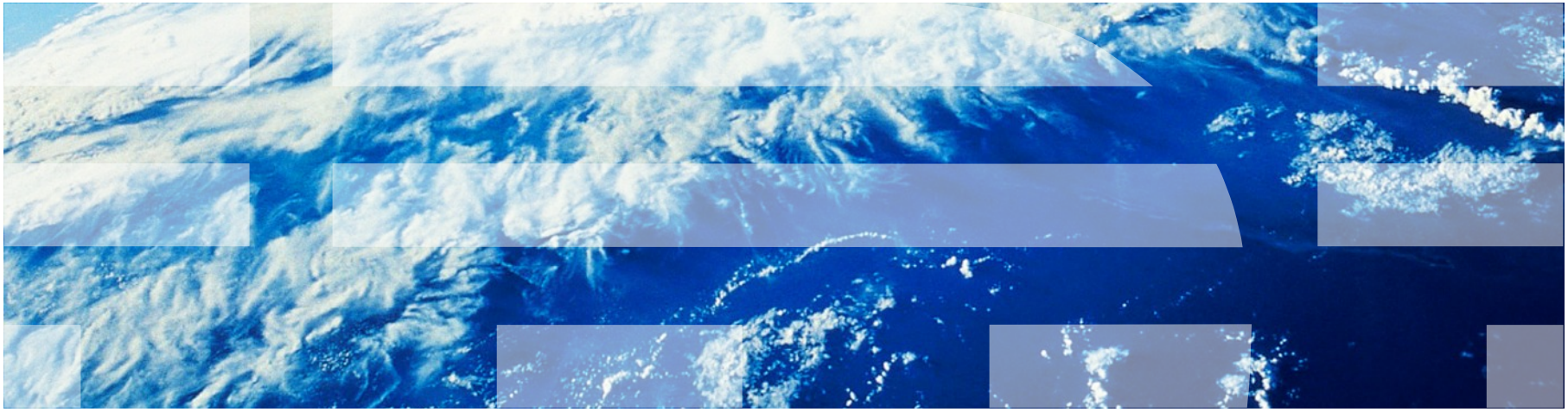
Overview of the Linux Integrity Architecture

David Safford, Mimi Zohar



Overview of the Linux Integrity Architecture

http://linux-ima.sourceforge.net

“An Overview of the Linux Integrity Architecture” (David Safford)

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

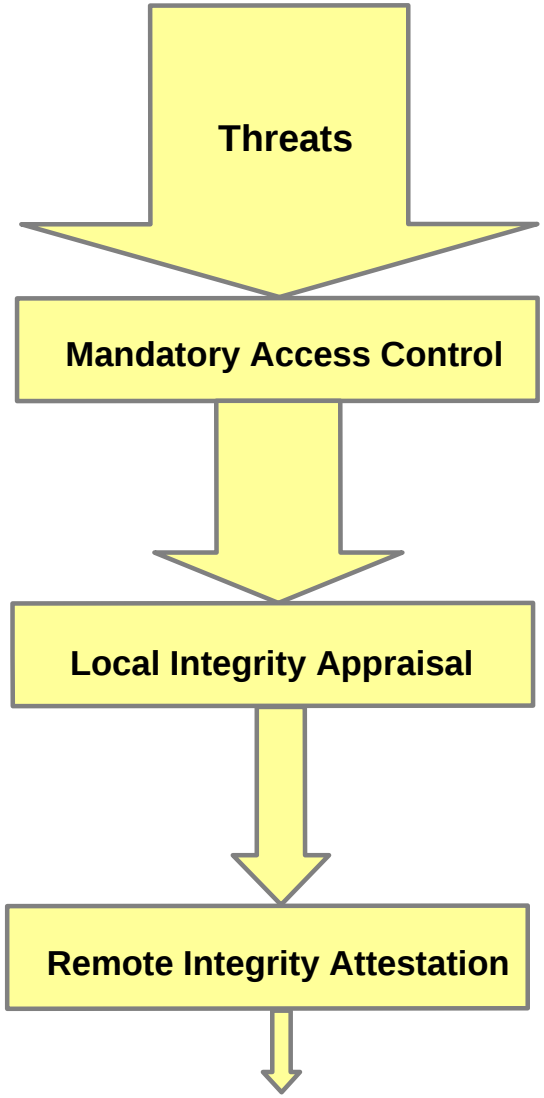
Today's talk will:

- Briefly overview all of the components
- Relate to today's other talks
- Describe some use cases of interest
- Demonstrate trusted cloud use case

Related Talks Today

- David Safford (IBM)
  - Trusted cloud
    - vTPM extensions to QEMU-KVM
    - IMA measurement
    - OpenPTS attestation
- Casey Schaufler (Intel)
  - Meego
    - IMA-Appraisal-Hash + EVM-digital-signatures
- Peter Kruus (JHAPL)
  - IMA-Appraisal-Hash + EVM-HMAC
  - Running on vmware (no vTPM)

Integrity Overview: Defense in Depth



Selinux
Smack

IMA-Appraisal	(data integrity)
EVM	(metadata integrity)
IMA-Appraisal-Digital-Signature	(data authenticity)
EVM-Digital Signature	(metadata authenticity)
IMA-Appraisal-Directories	(path integrity)

IMA-Attestation
OpenPTS

Integrity Component Overview

Area	Component	Author	Status
Instantiation	SRTM (trusted grub)	Seiji Munetoh	Available
	DRTM	Intel	Available (F15)
	Trusted keys	David Safford	2.6.38 (F15)
Appraisal	IMA-Appraisal-Hash	Mimi Zohar	Posted
	EVM-HMAC	Mimi Zohar	linux-next
	IMA-Appraisal-Digital-Signature	Dmitry Kasatkin	Posted
	EVM-Digital-Signature	Dmitry Kasatkin	Posted
	IMA-Appraisal-Directories	?	future
Attestation	IMA	Mimi Zohar	2.6.30 (F15)
	OpenPTS	Seiji Munetoh	Available (F15)
Virtualization	vTPM	Ken Goldman	Available
	QEMU-KVM patches	Stefan Berger	Posted
	Qemu-launcher patches	David Safford	To be posted

Integrity Instantiation

- **TPM based Trusted Boot**
  - SRTM (Trusted Grub)
  - DRTM (tboot – Intel, dboot IBM)
  - Trusted/encrypted keys (TPM sealed or encrypted symmetric keys in kernel
    - Extension to existing kernel key ring
    - Keys created/encrypted/decrypted in Kernel
    - User space sees/stores only encrypted blobs
    - trusted key type:
      - TPM generated random number, RSA sealed by TPM
      - unsealed by TPM, only if boot PCRs & other criteria match
      - Slow
    - encrypted key type:
      - kernel random number encrypted/decrypted using trusted key
      - fast
- **Non-TPM based boot integrity**
  - USB Token
  - Boot Password

IMA Attestation

```
$ cat /sys/class/misc/tpm0/device/pcrs

PCR-00: 04 E2 36 3F 12 BA AA BF 63 9F 2A 9C 6B 2A 02 C5 65 FC 7D F8
PCR-01: 77 EF 49 CA D5 50 54 7E 82 19 83 35 26 3A 9D D3 75 CF 4C 2F
PCR-02: 53 DE 58 4D CE F0 3F 6A 7D AC 1A 24 0A 83 58 93 89 6F 21 8D
PCR-03: 3A 3F 78 0F 11 A4 B4 99 69 FC AA 80 CD 6E 39 57 C3 3B 22 75
PCR-04: FA 98 55 9A C1 14 9C 6C D1 42 CE 76 F9 06 15 98 14 8B CB 8D
PCR-05: E1 32 A2 95 8A 85 0A 2C 29 93 86 89 40 BB 74 25 65 F0 C9 2C
PCR-06: 58 5E 57 9E 48 99 7F EE 8E FD 20 83 0C 6A 84 1E B3 53 C6 28
PCR-07: 3A 3F 78 0F 11 A4 B4 99 69 FC AA 80 CD 6E 39 57 C3 3B 22 75
PCR-08: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR-09: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR-10: 9F FF 76 4D 71 31 30 C7 E3 46 99 B3 F1 FE 69 45 09 0F 0F 23 <== IMA anchors the list here
```

Attestation proves to third party that the IMA measurement list is valid, because it matches the hash in the IMA PCR, which is signed by the TPM.

(IMA LTP test: ima_tpm.sh)

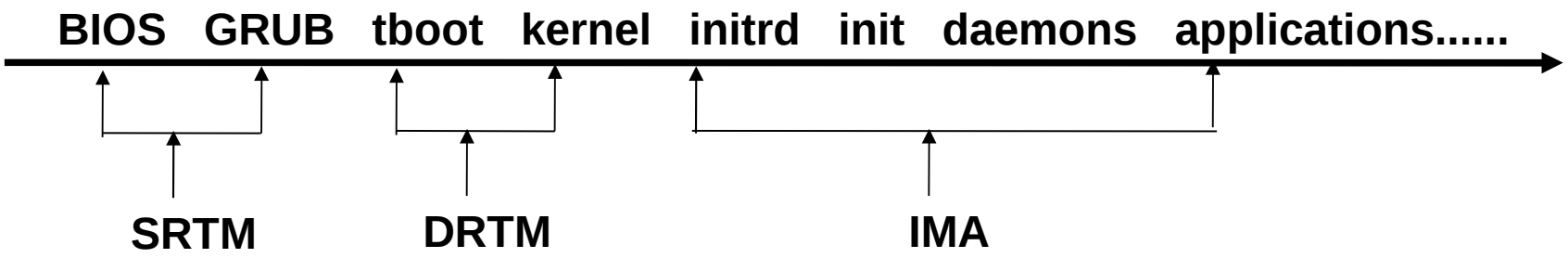
Even evil software cannot remove its measurement from the list without invalidating the signature.

Comprehensive Integrity Measurement

If you run someone else's program on your computer, it's no longer your computer.

If you don't know what's running on your computer, you cannot know if it is still your computer.

This includes ALL TCB files, at ALL levels



IMA-Appraisal-Hash – Cryptographic binding of file data

- IMA saves the file measurement hash for each file as '**security.ima**'
- At access time (exec/open), IMA verifies file measurement hash and denies access if different
- 'security.ima' updated on file close (based on policy)
- Requires initial labeling of system

example: running “/tmp/more” without hash (from dmesg/auditd logs):

type=1800 audit(1273181698.565:1492): pid=19501 uid=0 auid=500 ses=1 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 op="appraise_data" cause="missing-hash" comm="bash" name="more" dev=sda6 ino=37817 res=1

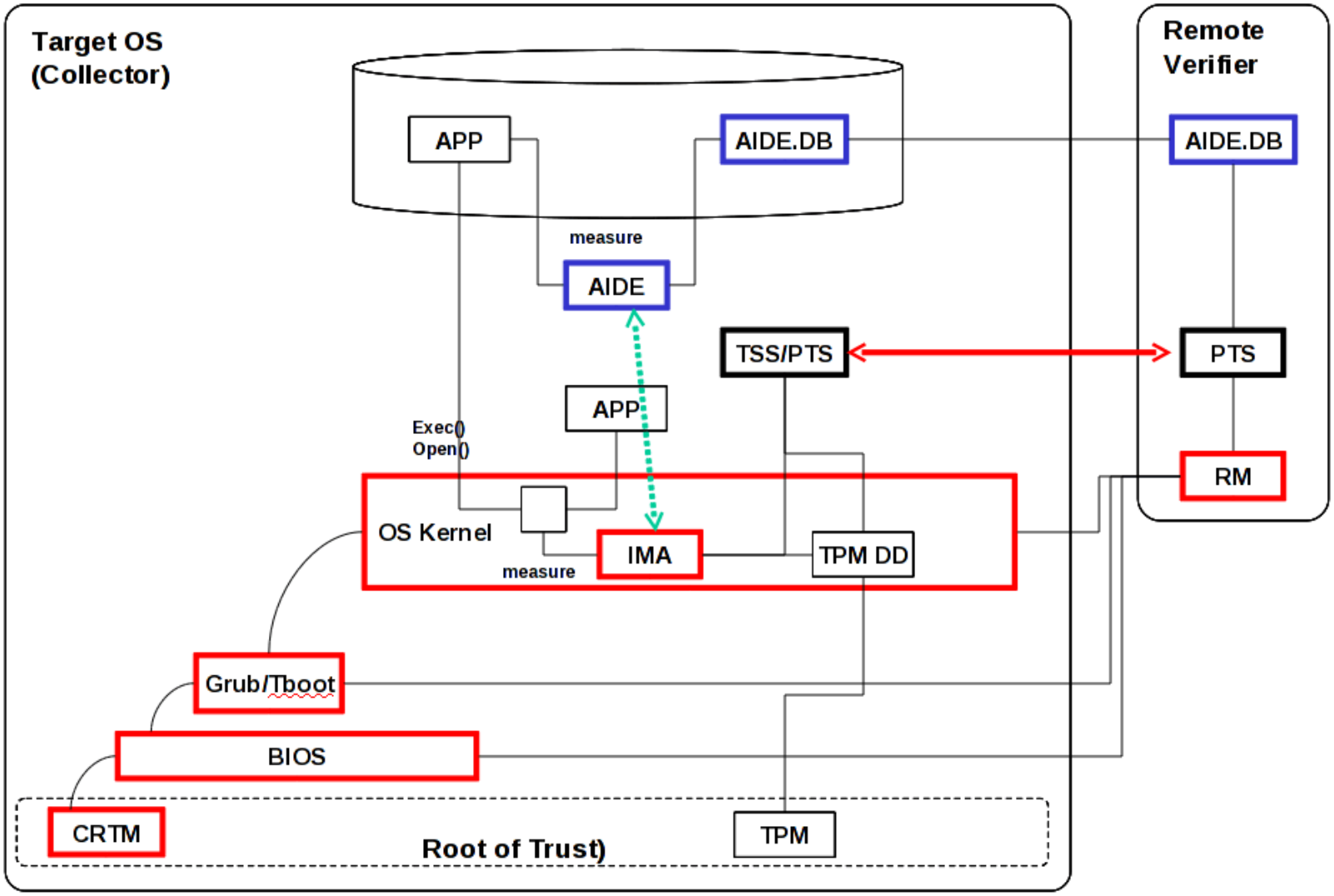
example: running “/tmp/more” with invalid hash with selinux disabled:

type=1800 audit(1273187087.958:95): pid=2655 uid=0 auid=500 ses=1 op="appraise_data" cause="invalid-hash" comm="bash" name="more" dev=sda6 ino=37817 res=1

EVM-HMAC: Cryptographic binding of file metadata

- Goal – cryptographic protection of security extended attributes
  - security.selinux
  - security.SMACK64
  - security.capability
  - security.ima
- Symmetric key HMAC is used to bind attributes and other file metadata
  - Inode number
  - Owner, Group
  - Mode
- Protects all of this metadata against off-line attack
  - The signing key is released by the TPM only if trusted kernel booted
  - Even someone with physical access cannot get key and forge HMAC
  - Thus IMA hashes are trusted, and subject only to kernel level vulnerabilities
  - Protected xattrs configured at compile time

PTS – integration with AIDE (an opensource measurement database)

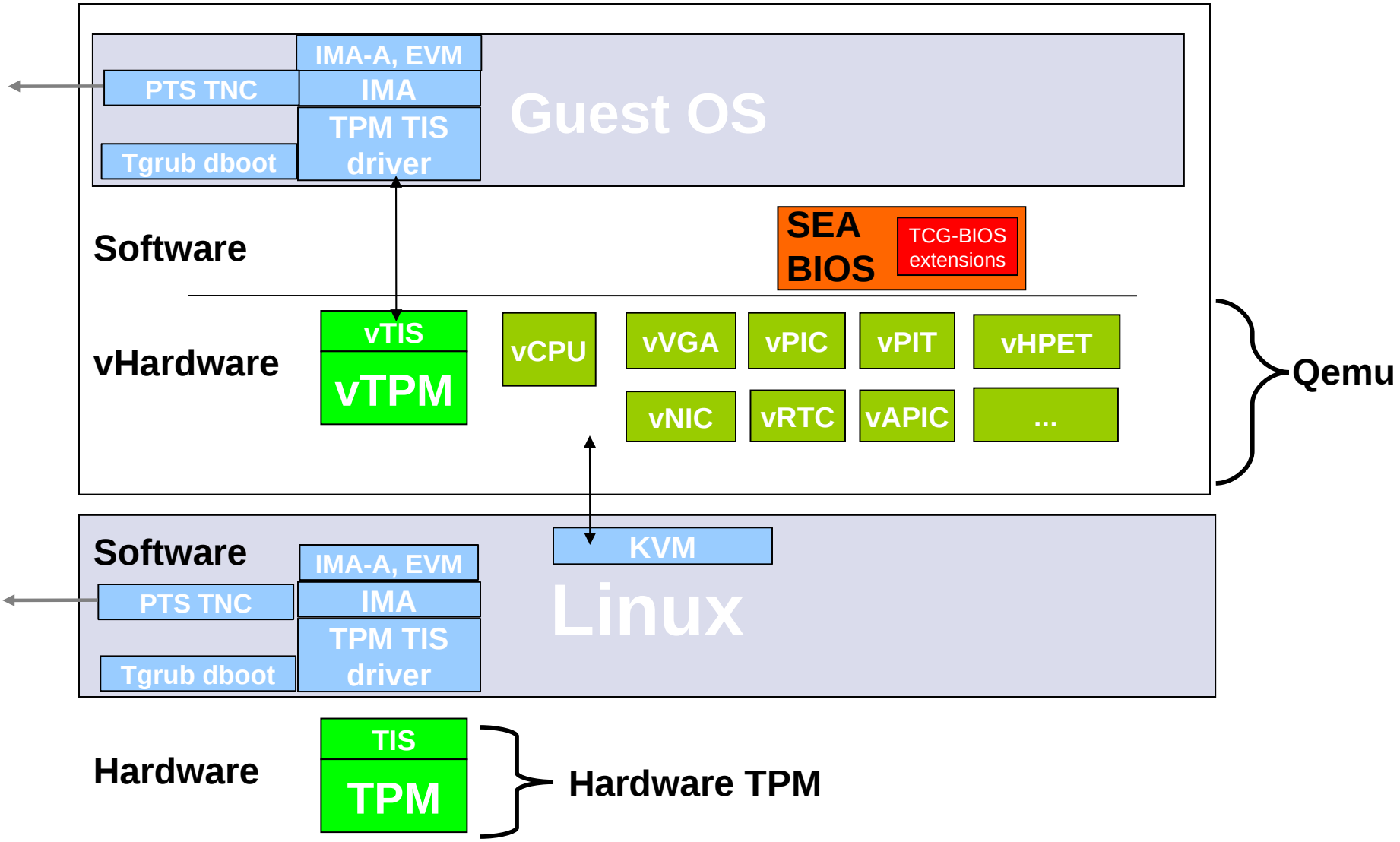


Trusted Cloud Use Case

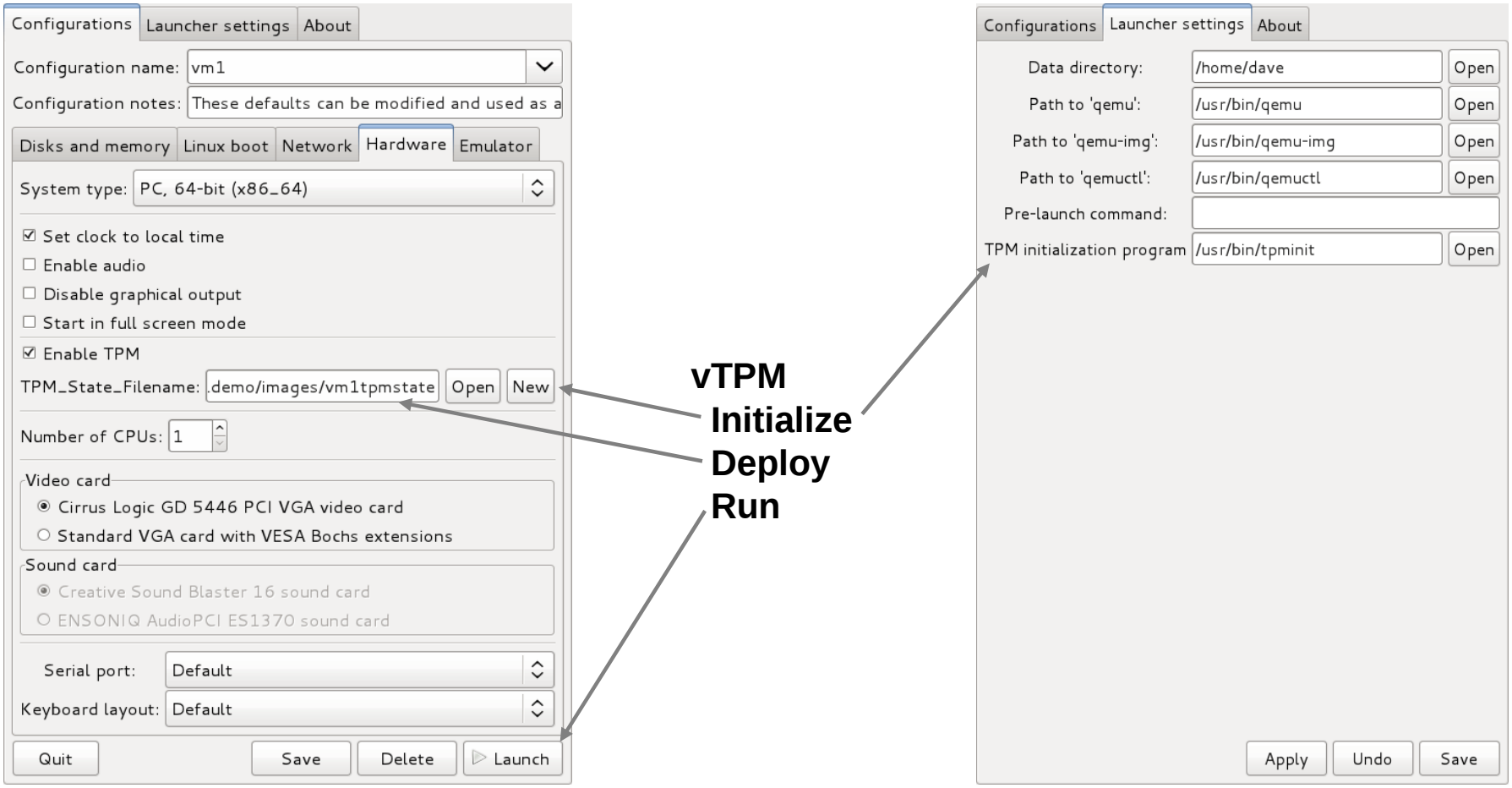
- Cloud Security Issues
  - Has the provider's native host been compromised?
  - Has my VM image been compromised?
  - Have I booted my image?
  - Am I protected from other VM's?

- Approach
  - Attest native platform (and vTPM) with PTS
  - Attest VM with PTS

X86 Qemu/KVM Architecture



Qemu-launcher



Demo

- Qemu-launcher with vTPM mods
- Qemu-KVM with vTPM mods
- Unmodified Fedora 15 Guest
  - IMA measurement and attestation
  - OpenPTS