Protecting the Filesystem Integrity of a Fedora 15 Virtual Machine from Offline Attacks using IMA/EVM

Linux Security Summit
8 September 2011

Peter Kruus
Peter.Kruus@jhuapl.edu

Agenda

- Need for integrity protection in virtual machines
- Use Case: VMware ESXi environment
- Integrity protection with IMA, IMA Appraise, and EVM
- Setup and booting in a VM
- Example attack
- Next Steps

Need for Integrity in a Virtual Environments

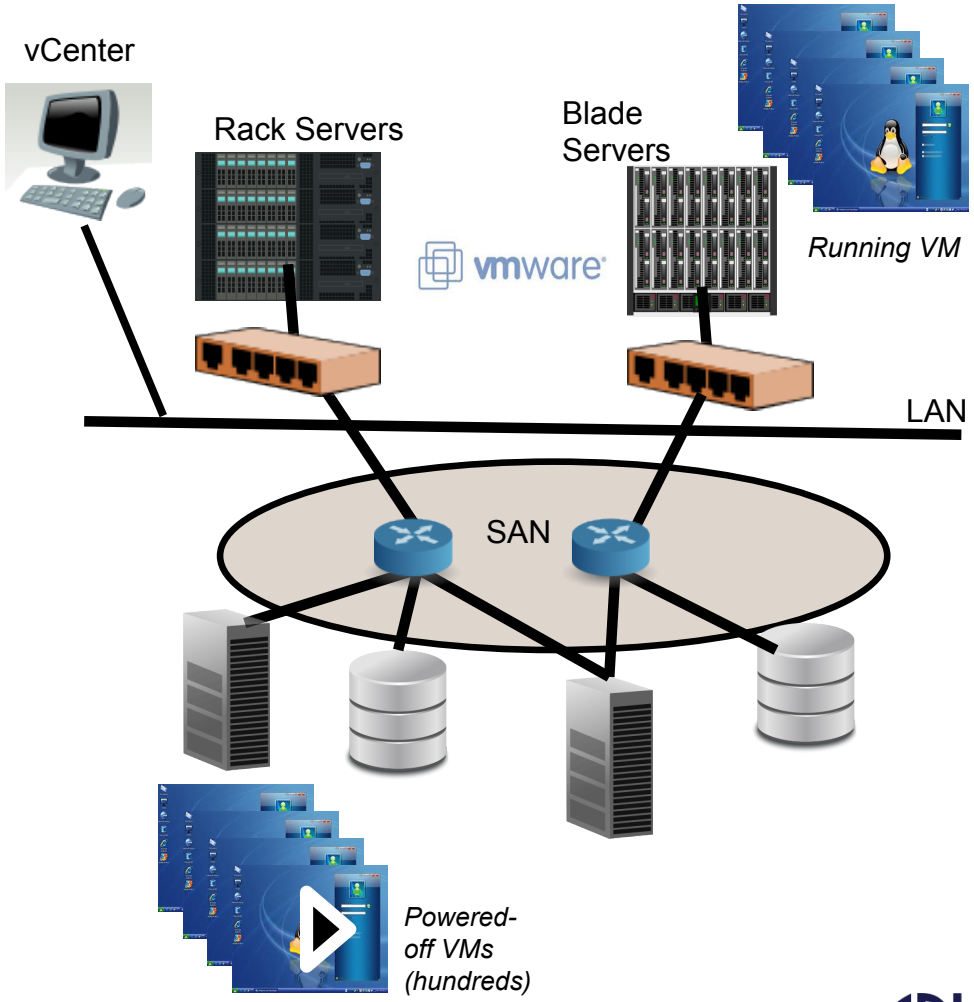
- **Problem:**
  - Virtual machines (VMs) are vulnerable to integrity attacks when **running** *and* ***while powered off***
- **Need:**
  - Disk encryption may offer protection while powered off, but may not when VM is powered on
  - Integrity protection for files even when the system is not running and when MAC controls are not effective.
- **Our Approach:**
  - Apply IMA Appraisal and Extended Verification Module (EVM) in a VMware VM to protect against offline attacks
  - Illustrate loading EVM keys via initramfs (via *dracut* patches)

Security Goals in a VM

- **Monitor file integrity**
  - Critical files - executables, config files, libraries
  - While the system is running and protect while *powered off*
- **Detect file modifications *locally* at *load-time,***
  - Before a file is executed, read, mapped to memory
  - Compare measurement against known local golden value
- **Local Enforcement**
  - Block access to files based on appraisal results
- **Hardware root of trust**
  - Build a root of trust starting in immutable firmware
  - Measure the kernel and initramfs
  - Use TPM/vTPM for immutable integrity protection of measurements

Use Case: *Virtualization with VMware ESXi*

- **VMware ESXi**
  - Baremetal
  - PXE booted onto hardware to protect from hardware failure
- **Hardware (TPM support, *but…*)**
  - Dell PowerEdge
  - HP Proliant Blade Server
  - IBM BladeCenter
- **VM images stored on Storage Area Network (SAN)**
  - 100's of VMs
- **Management**
  - vCenter
  - Migration between servers for load balancing



Use Case: *ESXi Hardware Platforms*

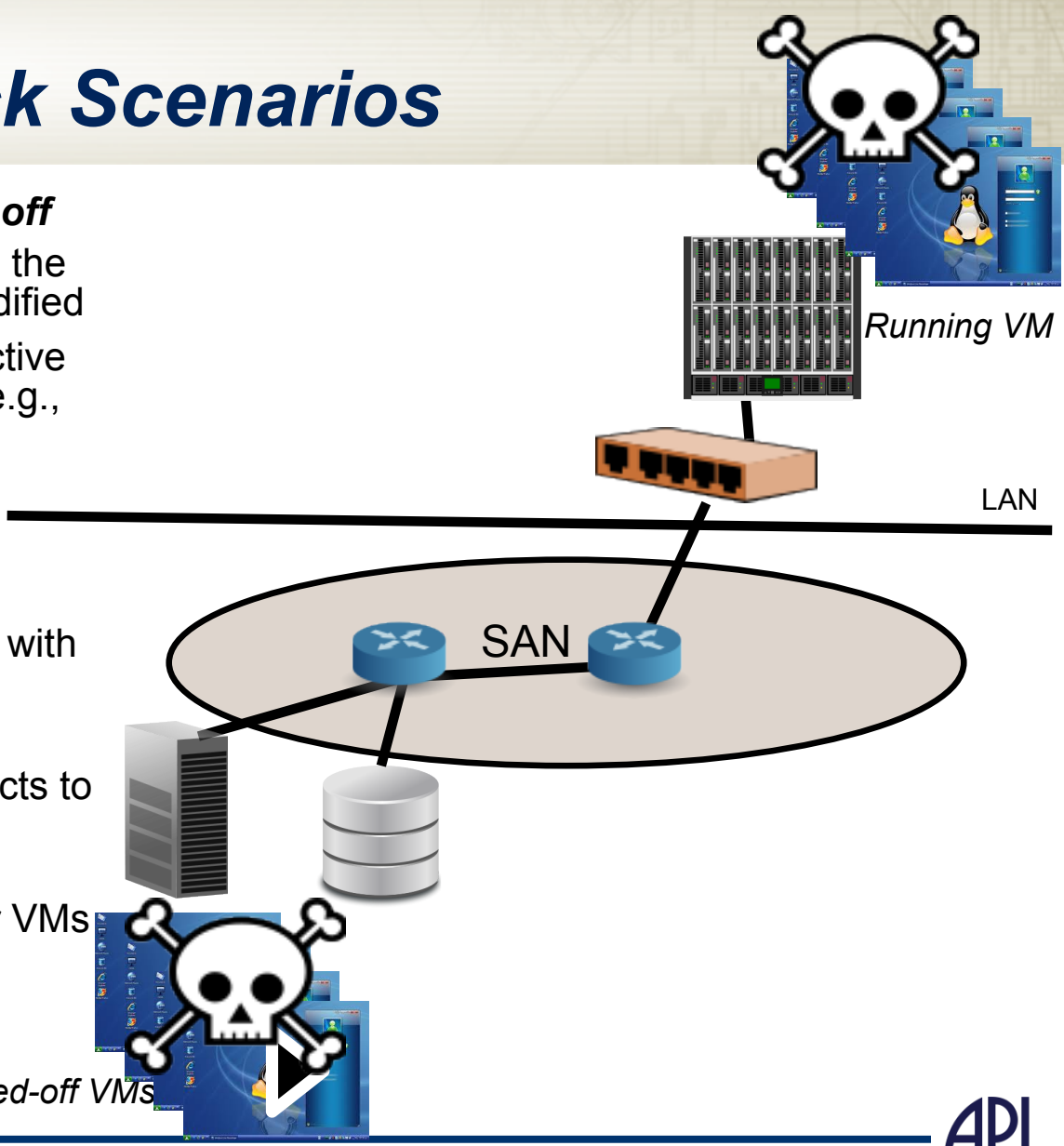
- **Hardware platforms available with TPM support**
- **TPM enables a hardware rooted chain of trust**
  1. **Measurements start in immutable BIOS**
  2. **Measure the BIOS – then give control to the BIOS**
  3. **Measure the ESXi hypervisor kernel**
  4. **….**
- **Measurements extended into the TPM**
  - Immutable
  - Preserves integrity for reporting
- **TPM measurements can be reported in vCenter**
  - TPM PCRs are populated
- **Unfortunately, TPM is inaccessible to the VMs**
  - No vTPM support yet
  - Measurements of VM kernel/initramfs need protection

TRUSTED COMPUTING GROUP™

APL

Use Case: *Attack Scenarios*

- **VMs vulnerable when *powered off***
  - Virtual disks can be mounted, the file system inspected and modified
  - MAC protections are not effective when system is not running (e.g., SELinux, SMACK)
- **Possible file system attacks**
  - Modification of critical files
  - Injection of malicious files
  - Replacing known trusted files with malicious files
- **Cloudburst (CVE-2009-1244)**
  - Attack against VMware products to break out of VM and execute arbitrary code on the host
  - Enables attacks against other VMs on the same server



Integrity Protection in a VM

- **Integrity kernel patches**
  - `git://git.kernel.org/pub/scm/linux/kernel/git/zohar/ima-2.6.git`
  - IMA for remote attestation (included since 2.6.30)
  - **IMA Appraisal for load time integrity**
  - **Extended Verification Module (EVM)**
- **New key types supported on kernel's key ring can be used by EVM**
  - ***`trusted`*** **– keys sealed to a TPM**
  - ***`encrypted`*** **– keys encrypted by another key (e.g. trusted or user)**
- **Our approach**
  - **Use IMA Appraisal with EVM**
  - **Use *encrypted* key type**
  - **Load keys during initramfs (*dracut patches)***

APL

Review of IMA Appraisal and EVM

- **Local load time integrity measurement, appraisal, and enforcement**
  - Hashes added as extended security attributes

```
# file: boot/vmlinuz-3.0.0-rc1+
security.ima=0x0123fef68e5920129b30c80fb6b1987dd58ff3e0a4
security.evm=0x022a27e8166e244ffbc1ad4bf045247a8d493dd567
security.selinux=0x73797374656d5f753a6f626a6563745f723a626f
    6f745f743a733000
```

- **`security.ima` : sha1 hash of the file**
  - Updated on file write when `ima_appraise=fix`
  - Cannot be updated when `ima_appraise=enforce`
  - Testable using *`sha1sum`*
- **`security.evm` : keyed HMAC of selected file attributes**
  - `security.ima, security.selinux, security.SMACK64, security.capability`
  - Key must be loaded into the kernel to compute HMAC
  - Can can be protected with *trusted* or *encrypted* key types

Setup in a VM: *First Use*

- **Build and install integrity patched kernel**

```
CONFIG_IMA=y
CONFIG_IMA_MEASURE_PCR_IDX=10
CONFIG_IMA_AUDIT=y
CONFIG_IMA_LMS_RULES=y
CONFIG_IMA_APPRAISE=y
CONFIG_EVM=y
```

- **Reboot into "fix" mode**

```
root (hd0,0)
  kernel /vmlinuz-3.0.0-rc5+ ro root=/dev/mapper/VolGroup-lv_root rd_LVM_LV=VolGroup/
  lv_root rd_LVM_LV=VolGroup/lv_swap ima_tcb ima_appraise=fix evm=fix
  initrd /initramfs-3.0.0-rc5+.img
```

- **Generate keys for EVM**
- **Label the filesystem**
  - LSM labels
  - Integrity labels for IMA Appraisal and EVM
  - `find / -fstype ext4 -type f -uid 0 -exec head -n 1 '{}' >/dev/null \;`
- **Rebuild the initramfs with IMA/EVM patches**
  - `dracut –f`

APL

Booting in a VM

- **Boot into "enforce" mode**

```
root (hd0,0)
kernel /vmlinuz-3.0.0-rc1+ ro root=/dev/mapper/VolGroup-lv_root
rd_LVM_LV=VolGroup/lv_root rd_LVM_LV=VolGroup/lv_swap ima_tcb
initrd /initramfs-3.0.0-rc5+.img
```

- **Load initramfs (see next slide)**
  - Prompt user for password and load EVM keys
  - Load IMA measurement and appraisal policy
    - `cat measure.selinux > /sys/kernel/security/ima/policy`
  - Enable EVM
    - `echo "1" > /sys/kernel/security/evm`
- **System boots and logs indicate success**

```
Aug  4 01:34:12 localhost kernel: [    3.551290] IMA: No TPM chip found, activating TPM-bypass!
Aug  4 01:34:12 localhost kernel: [    0.156071] EVM: security.selinux
Aug  4 01:34:12 localhost kernel: [    0.156072] EVM: security.ima
Aug  4 01:34:12 localhost kernel: [    0.156073] EVM: security.capability
Aug  4 01:34:12 localhost kernel: [  143.190579] EVM: initialized
```


APL

Details of initramfs (dracut)

1. Prompts for integrity password and load as “user key” into kernel key ring

   Dracut 97masterkey

2. Loads encrypted EVM key into kernel as “encrypted-key”

   Dracut 97masterkey

3. Kernel decrypts “encrypted-key” with “user-key” (password) to recover EVM key
4. EVM key ready for use!
5. Initializes IMA/IMA Appraise policy

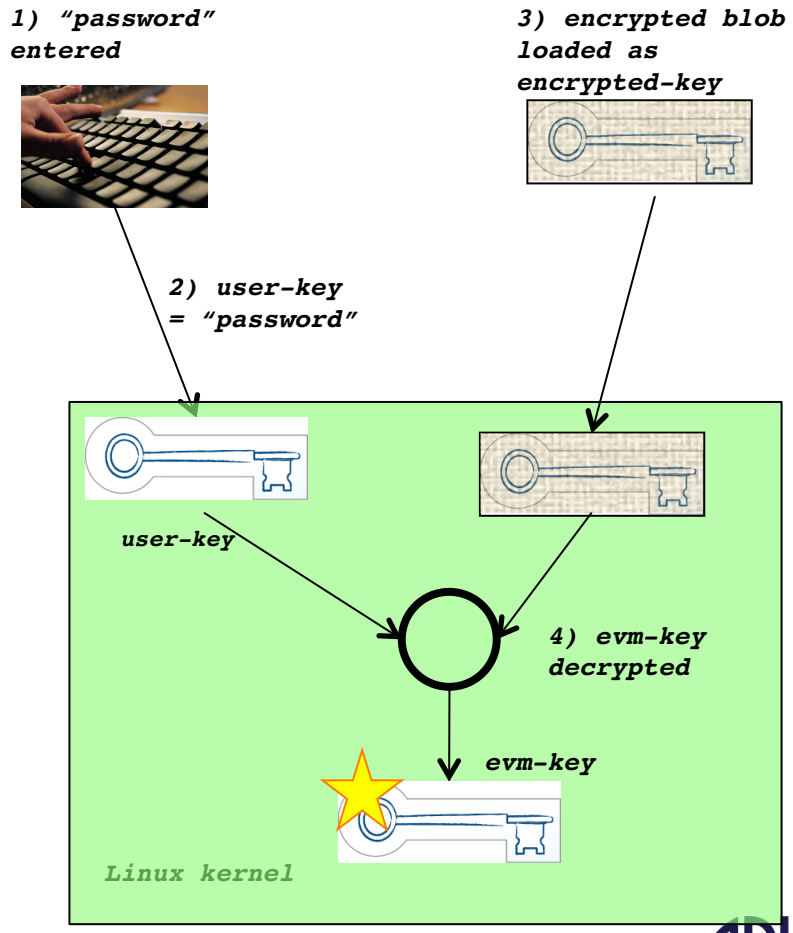
   `Dracut 98integrity`

6. Initializes EVM

   `Dracut 98integrity`

7. Ready for measurement, local appraisal, local enforcement!

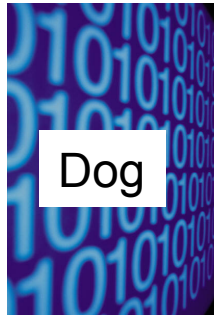
```
Session Keyring
       -3 --alswrv      0     0  keyring: _ses
590585570 --alswrv      0    -1   \_ keyring: _uid.0
924524992 --alswrv      0     0       \_ user: kmk-user
407956701 --alswrv      0     0       \_ encrypted: evm-key
```



Integrity Protection in Action!

Good file with Integrity/LSM Labels

```
# file: boot/vmlinuz-3.0.0-rc1+
security.evm=0x022a27e8166e244ffbc1ad4bf045247a8d493dd567
security.ima=0x0123fef68e5920129b30c80fb6b1987dd58ff3e0a4
security.selinux=0x73797374656d5f753a6f626a6563745f723a626
  f6f745f743a733000
```



Offline attacker modifies file (dog to cat)….security labels unchanged…..but

```
# file: boot/vmlinuz-3.0.0-rc1+
security.evm=0x022a27e8166e244ffbc1ad4bf045247a8d493dd567
security.ima=0x0123fef68e5920129b30c80fb6b1987dd58ff3e0a4
security.selinux=0x73797374656d5f753a6f626a6563745f723a626
  f6f745f743a733000
```

Measurements taken on file load/execute/mmaped in “enforce” mode detect changes….access denied!

```
security.evm=0xc1ad4bf04166e244f0fb6b4bf045247a0fb6b32267
security.ima=0x29b30c80f68e592012f5782eab99002346abc51010
```


APL

Next Steps:

- **Possible improvements**
  - Measure VM kernel and initramfs during boot
    - E.g., trusted boot
  - vTPM to protect critical measurements
  - Tie EVM keys to platform state
    - Trusted key type
    - Use vTPM
- **Interim solution**
  - Measure VM kernel and initramfs from hypervisor
  - Only boot if measurements attest successfully
- **Ultimate solution!**
  - vTPM support in linked to hardware TPM
  - vTPM supported trusted boot