New extensions to Linux Kernel Integrity Protection Subsystem

Dmitry Kasatkin
Intel Open Source Technology Center

Linux Security Summit 2012
30.08.2012

Agenda

- Why integrity protection?
- Kernel Integrity Subsystem
- What is missing?
- New extensions
- Performance comparison
- Demos
- References & discussion

Why 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

Kernel Integrity Subsystem

Located under <linux>/security/integrity

- IMA – Integrity Measurement Architecture
- EVM – Extended Verification Module
- Digital Signature extension

- IMA-Appraisal
- Directory integrity verification
- Special files integrity verification
- Module integrity verification

IMA – Integrity Measurement Architecture

- Since 2.6.30 (CONFIG_IMA)
- Measures integrity of the file content using cryptographic hash
- Maintains runtime measurement list
  - /sys/kernel/security/integrity/ima/ascii_runtime_measurements
- Calculates the boot aggregate value over TPM registers
- Extends IMA PCR
  - Incorrect value “locks” TPM secrets such as keys
- Can be used to attest system's runtime integrity

EVM – extended verification module

- Since 3.2 (CONFIG_EVM)
- Provides integrity protection of inode metadata against offline modification
  - security.{ima,SMACK64,selinux, caps}, ino, mode, owner, …
- Measures integrity of the inode metadata using hash-based message authentication code (HMAC)
- Performs local integrity validation and enforcement against “good” reference HMAC value
- Reference HMAC value is stored in 'security.evm' extended attribute

Digital Signature Extension

- Since 3.3 (CONFIG_INTEGRITY_SIGNATURE)
- Allows to protect file metadata data and data using digital signature
- security.evm and security.ima may hold signature instead of HMAC or hash
- security.evm: signature is replaced by hmac on successful verification
- security.ima: sinature is never replaced with hash – file is immutable
- If image copying/flashing method does not have access to the HMAC key, file system can be labeled with digital signatures

IMA-appraisal

- Hope for 3.7 (CONFIG_IMA_APPRAISE)
- Provides local integrity validation and enforcement against “good” reference hash value
- Reference hash is stored in 'security.ima' extended attribute
- 'security.ima' is protected by EVM

What is missing?

- EVM protects integrity of inode metadata
- Currently IMA protects integrity of the content of regular files
- Inode itself does not have a name associated with it
- Name is associated with inode via directory entry – not protected
- Offline, files can be deleted, renamed or moved from one directory to another one
- Directory content integrity verification is needed to prevent that
-
- Symlinks, device nodes are not protected
- Kernel modules are not protected

Demo – possible attacks

- Renaming
- Removing file
- Moving file (adding new file)
- Pasting from backup
  - Old file might contain exploit

Directory integrity verification

- Config option: CONFIG_IMA_DIRECTORIES
- Two new IMA hooks: ima_dir_check() and ima_dir_update()

- ima_dir_check(path) - integrity verification
  - during path lookup (may_lookup) or on chdir/fchdir

- ima_dir_update(path, dentry) – integrity measurement update
  - when dir entries are added/removed
  - mknodat, mkdirat, rmdir, linkat, unlinkat, symlinkat, renamat

Implementation details

- Functionality
  - collect → appraise
  - collect → update
- Verification starts from root dentry/inode
- Callbacks are call when new dentry has been just allocated
  - Does not break RCU path walk
- Directory measurement
  - hash over list of entries: (inode number, name, type, offset)
  - xattr: security.ima

Special files integrity integrity verification

- One new hook: ima_link_check()
- ima_link_check() - symlink integrity verification during path lookup (follow_link) or on sys_readlink
- Symlink measurement – security.ima
  - hash of the target path
  - Initial value is set on sys_symlinkat()
- Device node measurement
  - Hash over MAJOR:MINOR

User space tools - labeling

- Needed for image creation
- ima-evm-utils (evmctl)
- Added support for setting reference hash value for directories, symlinks and device nodes

Demo – how it works with new features

Performance comparison

Ubuntu 11.10 on Cedar Trail	No Integrity	IMA/EVM	IMA/EVM (with dir)	dm-crypt
Boot time	48.5 s	46 s	47 s	60 s
Boot time (readahead)	30 s	37.4 s	37.7 s	33 s
File copy	13.3 MBs	12.1 MBs	10.9 MBs	9.3 MBs

Module integrity verification

- Sent for RFC (CONFIG_INTEGRITY_MODULES)
- Does not require IMA and EVM
- One hook: integrity_module_check() - called from load_module() syscall
- Integrity measurement - digital signature of the module hash
- Signature is appended to the kernel module
  - Does not require modification of tools
- Public key can be embedded to the kernel
- Support for key creation and signing added to the kernel build scripts

References

- Integrity GIT: http://git.kernel.org/?p=linux/kernel/git/zohar/linux-integrity.git
- DIRS GIT: http://git.kernel.org/?p=linux/kernel/git/kasatkin/linux-digsig.git
- Tools GIT: http://linux-ima.git.sourceforge.net/git/gitweb-index.cgi
- Linux IMA project page: http://sourceforge.net/projects/linux-ima
- Linux IMA/EVM wiki: http://sourceforge.net/apps/mediawiki/linux-ima

- Discussion