When IMA/EVM collides with systems administration

Rossano P. Pinto[1], Rodrigo A. A. Pierini[1], Caio Teixeira[1], Felipe J. A. Rampazzo[1], Diego S. Pereira[3], Roger Immich[2], Christian E. Rothenberg[1], Marco A. A. Henriques[1]

[1]DCA - Universidade Estadual de Campinas (UNICAMP)

[2]IMD - Universidade Federal do Rio Grande do Norte (UFRN)

[3]NOCSRL - Instituto Federal do Rio Grande do Norte (IFRN)

{rossano,chesteve,marco}@dca.fee.unicamp.br

1. INTRODUCTION

System integrity aims to detect/block unauthorized component modifications. In Linux, Integrity Measurement Architecture and Extended Verification Module (IMA/EVM) enforces this through policies for measuring and appraising files. However, default policies [Linux M. 2008] can conflict with routine administration tasks when tools are unaware of IMA/EVM. We illustrate this with a Debian `apt` case study, analyzing root causes and proposing practical workarounds. To the best of our knowledge, no publicly available publication has reported the specific findings discussed in this paper.

2. BACKGROUND AND RELATED WORK

[Ozga et al. 2020] identifies software updates as a major challenge to maintaining integrity. [Cohen and Acharya 2012] notes the difficulty of deploying IMA/EVM. Over a decade later, similar issues remain: [Yocto 2021] reports that package managers in IMA appraisal enabled systems may install files without the required attributes, making them unusable. This reflects a persistent gap between software maintenance workflows and integrity policies. With IMA enabled, integrity checks ensure only trusted, verified files are accessed or executed. IMA policies define which files are measured and the response when accessing it; IMA maintains integrity by intercepting key file-related syscalls, hashing file contents, and storing the hash value in `security.ima` for future verification. EVM complements IMA by protecting i-node metadata against offline attacks, storing a hash in `security.evm` and validating metadata on access.

3. SCENARIO AND ROOT-CAUSE ANALYSIS

When Debian 12 is using IMA's `appraise_tcb` policy, package installation with `apt` fails. Neither `apt`'s nor Debian's documentation mentions IMA/EVM. For example, running `apt install vim` as root fails with "Permission denied" errors and `audit` logs showing missing hashes in downloaded files as the cause. Understanding `apt` failures under the `appraise_tcb` policy is critical, as misconfigured IMA/EVM can introduce vulnerabilities or disrupt system operation. Our investigation revealed a limitation in IMA/EVM's interaction with package managers, raising the question of whether this is a feature, bug, or vulnerability. By design, the `appraise_tcb` policy appraises only root-owned files [IBM 2023]; if a file is root-owned, valid `security.ima` and `security.evm` attributes must be present, otherwise access is denied. During an `strace apt install`, it was observed the failure occurs after a sequence of `lchown` and `openat` calls. **After further investigation the following sequence of**

events were identified: 1. A process owned by `_apt` (UID 42) downloads the `.deb` files; 2. No integrity attributes are created, as the policy does not require them for non-root processes; 3. A root-owned process moves the files to their final location via `rename`; 4. File ownership is changed to root; 5. `dpkg` attempts installation, but access is denied - attributes are absent; This shows that changing a directory entry or ownership does *not* trigger (re)creation of `security.ima` and `security.evm` attributes. **We reproduced the issue manually:** 1. A non-root user created files without integrity attributes; 2. As root, the file was moved to a new directory and ownership changed to root; attributes were still missing, causing access denial.

4. SOLUTIONS

To enable package installation, the following solutions were identified: **Solution 1-FIX_MODE:** Set the system to `appraise_tcb=fix evm=fix` mode and then install the package; **"Solution" 2-ROOT_USER:** A less elegant solution, this method changes the UID of the user `_apt` to 0, effectively making it the root user; **Solution 3-IMA_APT:** Modify `apt` to detect an IMA/EVM enabled system and act accordingly; **Solution 4-APPRAISE_TCB_APT:** Modify the policy `appraise_tcb` to include the user `_apt` in apprasings list by inserting `appraise fowner=42` in the policy. Resulting in appraising files owned by two different users. ► Solutions 1, 2, and 4 were successfully tested. **Solution 3 was simulated and can be directly adopted by `apt`:** Copying (instead of moving) a non-root-owned file as root recalculated attributes, changed ownership to root, and granted access.

5. CONCLUSIONS

The Linux IMA/EVM provide runtime verification and enforcement of file integrity. We observed a software installation failure under the `appraise_tcb` policy, caused by an unprivileged user (`_apt`) downloading files, changing ownership to `root`, and using move instead of copy operations. The solutions in Section 4 can address such failures, though the best approach depends on the scenario. Effective IMA/EVM deployment requires carefully designed policies. Standard utilities often lack support for integrity enforcement, limiting adoption. IMA/EVM use typically demands advanced systems administration skills making integrity-enforced systems difficult for average users. Maintainers of tools like `apt` should improve them to deal appropriately with IMA.

References

Cohen, J. C. and Acharya, S. (2012). Incorporating hardware trust mechanisms in apache hadoop... In *2012 IEEE Globecom Workshops*, pages 769–774.

IBM (2023). IMA Policy. `https://ima-doc.readthedocs.io/en/latest/ima-policy.html`. Accessed: Jun 25, 2025.

Linux M. (2008). IMA Policy. `https://www.kernel.org/doc/Documentation/ABI/testing/ima_policy`. Accessed: April 14, 2025.

Ozga, W., Quoc, D. L., and Fetzer, C. (2020). A practical approach for updating an integrity-enforced operating system. In *Proceedings of the 21st International Middleware Conference*, Middleware '20, page 311–325, New York, NY, USA. ACM.

Yocto (2021). Ima/evm integration. `https://git.yoctoproject.org/cgit/cgit.cgi/meta-security/tree/meta-integrity/README.md`. Accessed June 2025.