

2026-10-07

Hash-based module integrity checking

Presented to you by:

Thomas Weißschuh (Linutronix)

About me

- ▶ Kernel hobbyist since 2018
- ▶ Maintainer of
 - nolibc
 - multiple ChromeOS Embedded Controller drivers
 - multiple hwmon drivers
- ▶ Various Bug fixes
- ▶ Working for Linutronix on the kernel since 2024
 - vDSO
 - timekeeping/timers

Background: Reproducibility

A build is reproducible if given the same source code, build environment and build instructions, any party can recreate bit-by-bit identical copies of all specified artifacts.

<https://reproducible-builds.org/docs/definition/>

Background: Reproducibility (2)

- ▶ Two other talks
 - OpenWrt is reproducible! How we got there and what we learned
 - Bitwise-Reproducible Kernel Builds for the WhatsApp TEE — Proof of Concept
- ▶ Community distributions want independent rebuilders.
 - These do exist today.
- ▶ Distributions reach $\geq 95\%$ package reproducibility today.
- ▶ Only bit-for-bit reproducibility is generally useful.

“Just ignore the changing parts during validation”

- ▶ Works fine(ish) for the appended module signatures.
- ▶ The root of trust is embedded in `vmlinux`,
 - ... which might be compressed,
 - ... and wrapped in a loader stub,
 - ... and compressed again.
- ▶ Specifics depend on the architecture and configuration options.
- ▶ Everybody will require the special tooling to know what to exclude.

Background: Module signatures

- ▶ Loaded modules have full access to kernel internals.
- ▶ Lockdown restricts module loading to signed ones.
- ▶ X.509 signatures, based on generic signature infrastructure.
 - Signatures are appended to the `.ko` module files.
 - Root certificate is embedded in `vmlinux`.
- ▶ Most common setup uses an ephemeral key during the build.
 - No infrastructure requirements.
 - Key is discarded after the build.

Module signatures and reproducibility

- ▶ Ephemeral keys break reproducibility.
 - Each build produces different modules and vmlinux.
- ▶ Fixed keys break reproducibility by the public.
 - Reproducers require access to the secret signing key.
 - A public signing key makes the signing pointless.
 - Also they are much harder to implement.

Distributions tradeoffs

- ▶ ArchLinux: Ephemeral module signatures.
 - Security over security.
- ▶ NixOS: No modules signatures.
 - Reproducibility over security.
- ▶ Debian: Non-public reproducibility
 - Achieves security and reproducibility.
 - Complex implementation.

First solution: Linear list of all module hashes

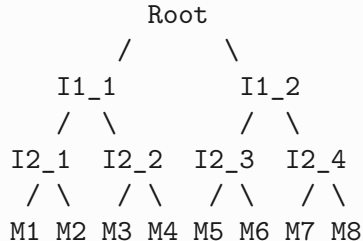
- ▶ Embed a list of modules hashes in `vmlinux`.
- ▶ Uses a lot of memory on generic distributions
 - 5000 modules * 32 bytes for SHA256 -> 150KiB
- ▶ Independently “discovered” by:
 - kpyrcd from reproducible-builds.org in 2024.
 - one commercial Linux vendor 10 years ago.
 - me in 2024.

Solution: Merkle tree signatures

- ▶ Deterministic, reproducible.
- ▶ No public-private key cryptography involved.
- ▶ The only necessary primitive is a hash algorithm
 - ... which is a dependency anyways.
 - ... which we can freely choose as required.
- ▶ Root of trust: the root hash and the depth of the tree.

Merkle tree signatures

- ▶ Simple Binary tree.
- ▶ Modules are leafs.
- ▶ Each node is hashed with its neighbor to form the parent.
- ▶ All the way up to the root.
- ▶ Signature data: Position in the list and hash complements.
- ▶ $\text{Sig}(M3) = 3, H(M4), I2_1, I_1_2$



Side-effects

- ▶ No more complex X.509 handling at runtime.
- ▶ Potentially avoid OpenSSL dependency at buildtime.
- ▶ Only one artifact per build needs to be signed externally.
 - Reduces load on expensive hardware security modules.
 - Speeds up the build.

Future changes

- ▶ Drop modversions.
- ▶ Expose symbols only to builtin modules.
- ▶ Trusted `MODULE_INFO(intree, "Y");`

kbuild complexity

- ▶ Normally `vmlinux` is built *before* the modules, now it is *after*.
 - Requires some ifdeffery in the Makefiles.
- ▶ Normally modules are stripped during installation, now they are during the build.
 - More ifdeffery.
- ▶ `make vmlinux modules` is racy:
 - These are special targets, not actually orderable.
 - Fixing this might needs some kbuild refactoring.

Open questions

- ▶ Reduce or clean up the kbuild ifdeffery.
- ▶ How to interact with IMA?
 - IMA can validate files based on module signatures.
 - Does IMA need to recognize this new scheme?
- ▶ Share merkle tree implementations?

Current state

- ▶ Expressed interest by:
 - The reproducible-builds.org folks.
 - Various community distributions.
 - One enterprise distribution.
 - Kernel crypto folks.
- ▶ Reviews wanted: Crypto bits, kbuild bits, everything else.
- ▶ Please test!
- ▶ 20260505-module-hashes-v5-0-e174a5a49fce@weissschuh.net
- ▶ git.kernel.org/.../thomas.weissschuh/linux.git b4/module-hashes
- ▶ <https://lwn.net/Articles/1012946/> (old revision)