Verified Debian Packaging at Scale

Frederick Lawler

Open Source Summit

Minneapolis, MN, US 2026

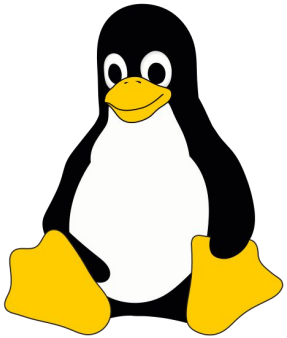
Code Red

Photo Credit: https://commons.wikimedia.org/wiki/File:Mountain_Dew_CODE_RED.jpg

Thanksgiving 2023 Security Incident

https://blog.cloudflare.com/thanksgiving-2023-security-incident/

Our Software Stack



- Debian Linux across our fleet
- First-Party Debian Packages
- Upstream Debian Packages
- All servers run the same software and services*
- Upgrades or changes can happen at any time*

*certain terms and restrictions may apply

Requirements

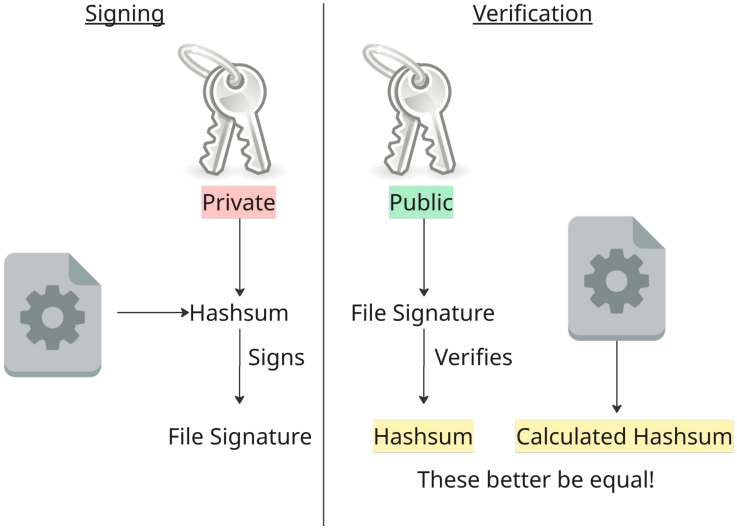
- Ensure attackers can't run or change **arbitrary scripts or programs**.
- Software can be installed, upgraded, removed at any time on machines.
- Don't break other teams workflows, and don't make them do extra work.
- Verification needs to be done *before* any code is executed.
- Signing happens in a trusted location.
- Private keys cannot be exposed.
- We need to sign all executables, dynamic objects, and scripts.

Arbitrary Scripts or Programs

- Binaries
  - Linux Executable
  - Shared Libraries
  - mmap()'d arbitrary blobs
- Scripts
  - Execve executes (#!)

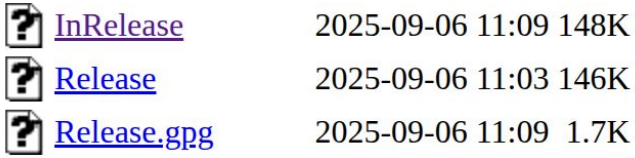
```
ffffffff812011c0 <run_init_process>:
ffffffff812011c0:       55                      push   %rbp
ffffffff812011c1:       48 89 fe                mov    %rdi,%rsi
ffffffff812011c4:       48 89 fd                mov    %rdi,%rbp
ffffffff812011c7:       53                      push   %rbx
ffffffff812011c8:       48 89 3d 71 cf a0 01    mov    %rdi,0x1a0cf71(%rip)        # ffffffff82c0e140 <s
rgv_init>
ffffffff812011cf:       48 c7 c7 75 75 9f 82    mov    $0xffffffff829f7575,%rdi
ffffffff812011d6:       e8 e5 f3 10 00          call   ffffffff813105c0 <_printk>
ffffffff812011db:       48 c7 c7 8f 75 9f 82    mov    $0xffffffff829f758f,%rdi
ffffffff812011e2:       e8 d9 f3 10 00          call   ffffffff813105c0 <_printk>
ffffffff812011e7:       48 8b 35 52 cf a0 01    mov    0x1a0cf52(%rip),%rsi        # ffffffff82c0e140 <s
rgv_init>
ffffffff812011ee:       48 85 f6                test   %rsi,%rsi
ffffffff812011f1:       74 1f                   je     ffffffff81201212 <run_init_process+0x52>
ffffffff812011f3:       48 c7 c3 40 e1 c0 82    mov    $0xffffffff82c0e140,%rbx
ffffffff812011fa:       48 c7 c7 a4 75 9f 82    mov    $0xffffffff829f75a4,%rdi
ffffffff81201201:       48 83 c3 08             add    $0x8,%rbx
ffffffff81201205:       e8 b6 f3 10 00          call   ffffffff813105c0 <_printk>
ffffffff8120120a:       48 8b 33                mov    (%rbx),%rsi
ffffffff8120120d:       48 85 f6                test   %rsi,%rsi
ffffffff81201210:       75 e8                   jne    ffffffff812011fa <run_init_process+0x3a>
ffffffff81201212:       48 c7 c7 ae 75 9f 82    mov    $0xffffffff829f75ae,%rdi
ffffffff81201219:       e8 a2 f3 10 00          call   ffffffff813105c0 <_printk>
ffffffff8120121e:       48 8b 35 fb cd a0 01    mov    0x1a0cdfb(%rip),%rsi        # ffffffff82c0e020 <s
```

Signature Verification



- Signing
  - Generate a Public and Private Key Pair
  - Generate a Hash of the File
  - Use the Private Key to Sign the Hash
- Verify
  - Use the Public key to Verify the Signature
  - Calculate the Hash of the File
  - Ensure they are the Same
- Provides
  - Authenticity
  - Integrity

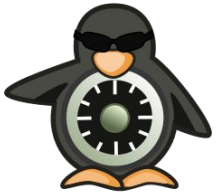
Existing Verification



- Debian Archives are GPG signed
  - This verifies the *package* came from the archive, not the individual binaries.
  - dpkg -i doesn't care about this.
- What about that _gpgorigin?
  - Embeds signature into package.
  - Solves dpkg -i problems
- An attacker can still overwrite executables. Checkmate.
- We need help from the Linux kernel!

Linux Security Modules (LSMs)

- Apparmor
- SELinux
- BPF LSM
- Lockdown
- ...and more!



Mandatory Access Control(s)

Integrity Measurement Architecture (IMA)

- Linux Security Module (LSM), enable with CONFIG_IMA.
- IMA Measurement → IMA calculates hashes
- IMA Appraisal → Verify Hashes or Signatures of a file
- Extended Verification Module (EVM) → Verifies metadata of files

IMA Appraisal Modes

Fix

- Hashes files on open based on policy
- Stores hashes raw as extended attributes (not signed)

Log

- Only logs infractions

Enforce

- Denies execution on infraction

IMA Appraisal Hashes

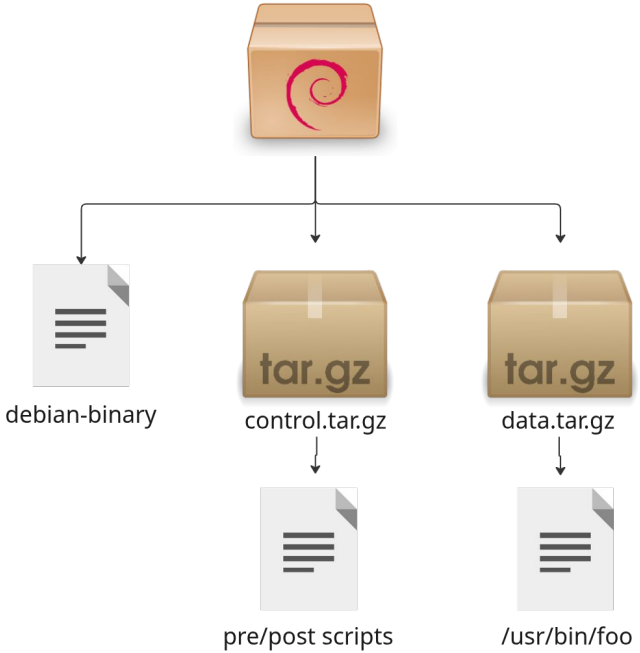
- Hashes/signatures are stored as extended attributes (**xattrs**)
  - security.ima
- Signatures = Header + File Hash + Signed with Private Key
- Notable supported file systems
  - btrfs
  - ext2/4
  - overlayfs
  - xfs
  - etc...

```
struct inode {
    ...
    struct super_block *i_sb = {
        ...
        const struct xattr_handler * const *s_xattr ...
    }
}

struct xattr_handler {
    const char *name;
    const char *prefix;
    int flags;
    bool (*list)(...);
    int (*get)(...);
    int (*set)(...);
}
```

Shipping Signatures with Packages

- Debug-symbol packages + Debuginfod
- Global manifest file
- Manifest file per package
- 2016 attempt at adding tar PAX to dpkg for xattrs

Debian Packaging



- Debian packages are AR archives.
- Contain metadata, data, control tar archives
- Control.tar contains scripts which execute during installation
- Data.tar contains binaries, files, etc.. used on system
- Debian packages *do not* preserve xattrs!

Xattrs File Format

- Established format for extended attributes
- Can be applied with tools compiled with libattr, such as setfattr
- Shipped with attr package

```
# file: /usr/bin/[
security.ima=0sAwIEIjdd5AEAUTGgtSMfrEvTkIrSPAIfM5rSdk52bv3ut4hmS446xZSjrF
6oGKKgQZb0dAIBUy5n8xZ3y3KLaTGAE0TyAUefiP3mLyyOiLfbGu679BHTeGn+llFaY4
KP2AsESGq1LXbB6MIQ6tyng+O9pxnCM7csYxBT7HIVoDSFgj6X4siEfyXQ6dL8S4SVjv/
wOvUG4r1EbA9o9o0FiLORPgia+SrQ434koPcONQHmWnY5jplvK4vUu8ooEVEi/tSNi5d
Wl+vCtm7dtAYDW48+nFD/+974EwiRAtbgIk89HezZ0BkIKsZbWYE8E/pwvI0As4XJ1eQ
fD2E2yEAwwRp7yN+8BCThpA==

# file: /usr/bin/arch
security.ima=0sAwIEIjdd5AEARS/36pBOSmjzi+8YKTZiAAvhDSv7JABvN/GiEr6GYdu0I
b4IeTgV9+KOc9OQ6geU7F4RYKFFh1ExiWR717No8TJCa4+NUhJqyVHht5JMrJEKpk9
niMLvsh2ZMnEXoJ0K1EdyLqVrJ3kDJF5WNbyOWyXFFPSmGLkGIMa6fmS0OQ1DpeqH
vVmxOPs8mrfjUBgnnUn1eE+yh0g7++tut4iG2A1cOp4IK47sniNfu2Pbktb81zMibSOukTJ
udx1UdiauszMPYH6qRd+TKy0vY6XkGU54j7Ym4NoBCF8v3bu5Zg9NPxrhyGTVe+ud8
7VNGHQrPWPIIiMTY3V6o27LG2tx/A==

# file: /usr/bin/b2sum
security.ima=0sAwIEIjdd5AEAJYsGFHy8zpT2QTjWpBp/rT2rftXQovftV/qbS1dn6PHbiE
oHkupAUOjvK5Q3Vv497VFocgK/ptLWR+vZH7bOjfa1KS+7eFjOuealJQ5qXCsNBI43Ed
AH0qlmi1Wqz7yPrlNZwfLu9iNki7M9ivOFJoVqFoB79q1×4JLQMIIAEX9+76Zjzj8e/GPgEi
muCFO8iQsweWggCoxyomp+fPpya9GowhUDW8RFpAB1lcoXhB1yh/BDBIBi5NeXkmd
HtJGQPS87cgioimtgov7IEVC18sBSlDeE4Qzdjd0572qLIN3OyPx9yJzX8ucheZKSRj/6EG
VNm82lpHkg3dJr4DYEEw==

...
```

Applying Xattr File

Install triggers

- dpkg -i ignores these unless configured not to
- apt calls triggers automatically
- Can’t handle pre/post scripts

Hijack Tar

- Natively supports extended attributes
- dpkg exec’s tar for the control.tar, and not data.tar
- data.tar installation is finely controlled via their own tar unpacker

Patch dpkg

- Solves all the problems
- Likely gets denied by upstream

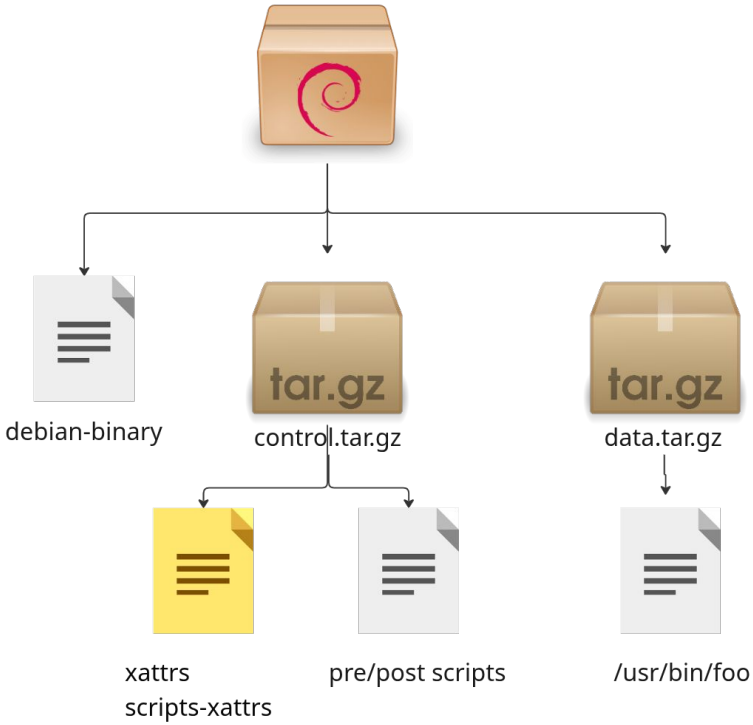
Patching dpkg

- New post install hook: --post-unpack-invoke
- Calls hook at exactly unpack & before execution of pre/post scripts
- Calls hook at exactly unpack of data.tar
- --post-install hook doesn't invoke at very end of dpkg install phase

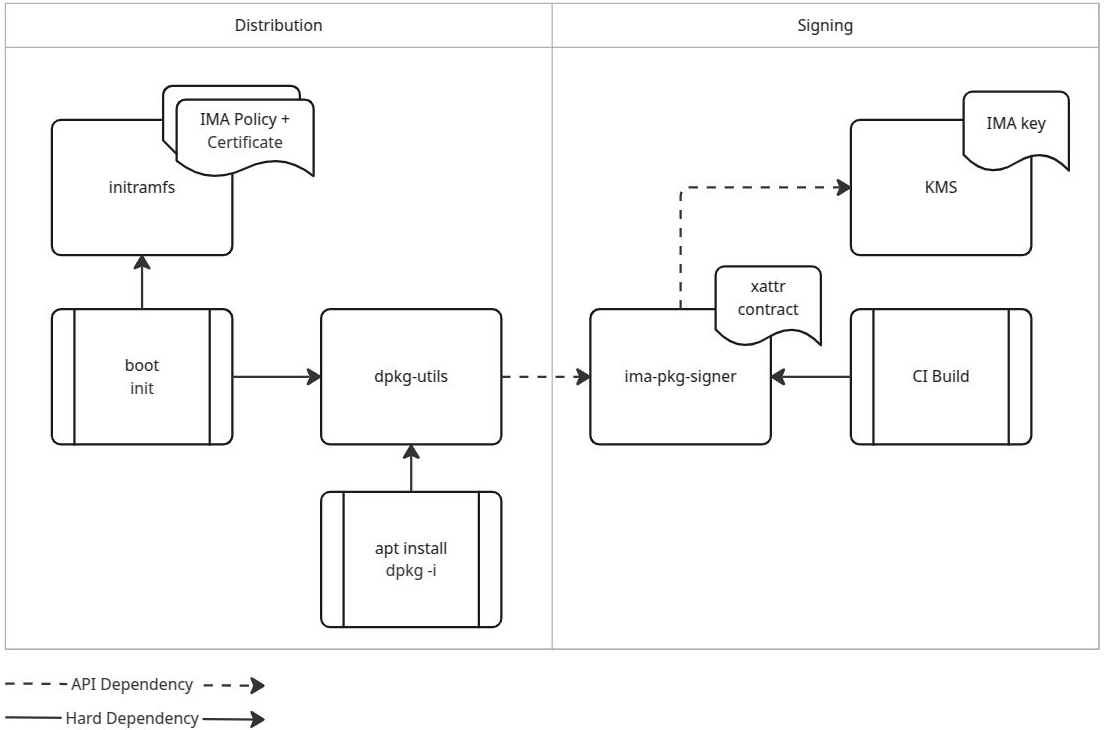
```
+static void
+invoke_unpack_hook(const char *archive, struct pkginfo *pkg, char *dir);
+
 static const char *
 summarize_filename(const char *filename)
 {
@@ -1353,6 +1356,8 @@ void process_archive(const char *filename) {
    /*
     * OK, we're going ahead.
     */
+  strcpy(cidirrest, "");
+  invoke_unpack_hook("control", pkg, cidir);

    trig_activate_packageprocessing(pkg);
    strcpy(cidirrest, TRIGGERSCIFILE);
@@ -1711,6 +1716,9 @@ void process_archive(const char *filename) {
    modstatdb_note(pkg);
    push_checkpoint(~ehflag_bombout, ehflag_normaltidy);

+  strcpy(cidirrest, "");
+  invoke_unpack_hook("data", pkg, cidir);
+
    /* Only the removal of the conflictor left to do.
     * The files list for the conflictor is still a little inconsistent in-core,
     * as we have not yet updated the filename->packages mappings; however,
@@ -1732,3 +1740,20 @@ void process_archive(const char *filename) {
    if (cipaction->arg_int == act_install)
      enqueue_package_mark_seen(pkg);
 }
```

Custom Tools

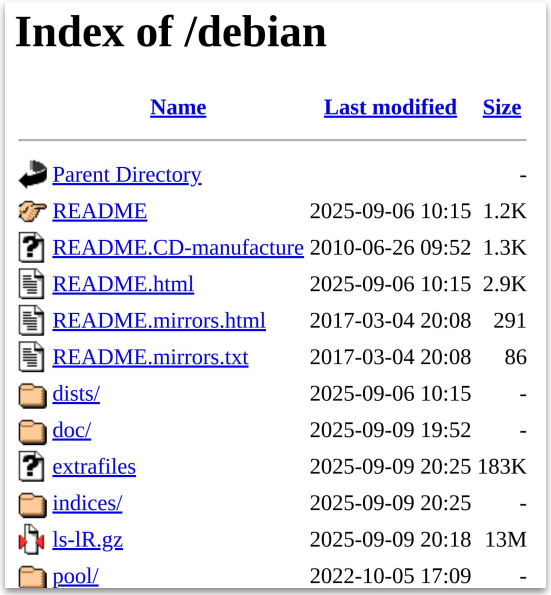
- **Dpkg-utils**
  - Something for dpkg to call into via a script
  - Supports initramfs/rootfs
- **Ima-pkg-signer**
  - Debian pkg specific support
  - Drop-in replacement for dpkg-deb -b



First Party Signing and Distribution



Signing An Entire Repository

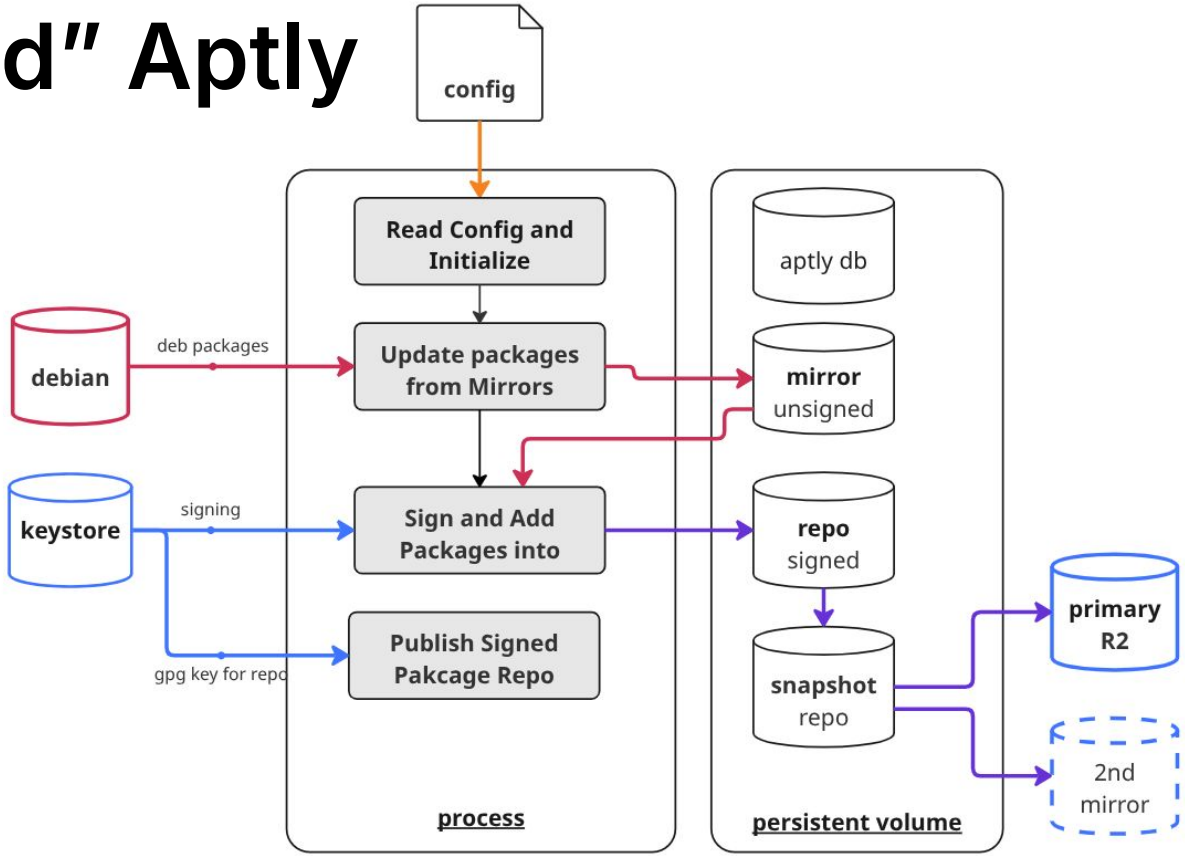


Index of /debian

Name	Last modified	Size
Parent Directory		-
README	2025-09-06 10:15	1.2K
README.CD-manufacture	2010-06-26 09:52	1.3K
README.html	2025-09-06 10:15	2.9K
README.mirrors.html	2017-03-04 20:08	291
README.mirrors.txt	2017-03-04 20:08	86
dists/	2025-09-06 10:15	-
doc/	2025-09-09 19:52	-
extrafiles	2025-09-09 20:25	183K
indices/	2025-09-09 20:25	-
ls-lR.gz	2025-09-09 20:18	13M
pool/	2022-10-05 17:09	-

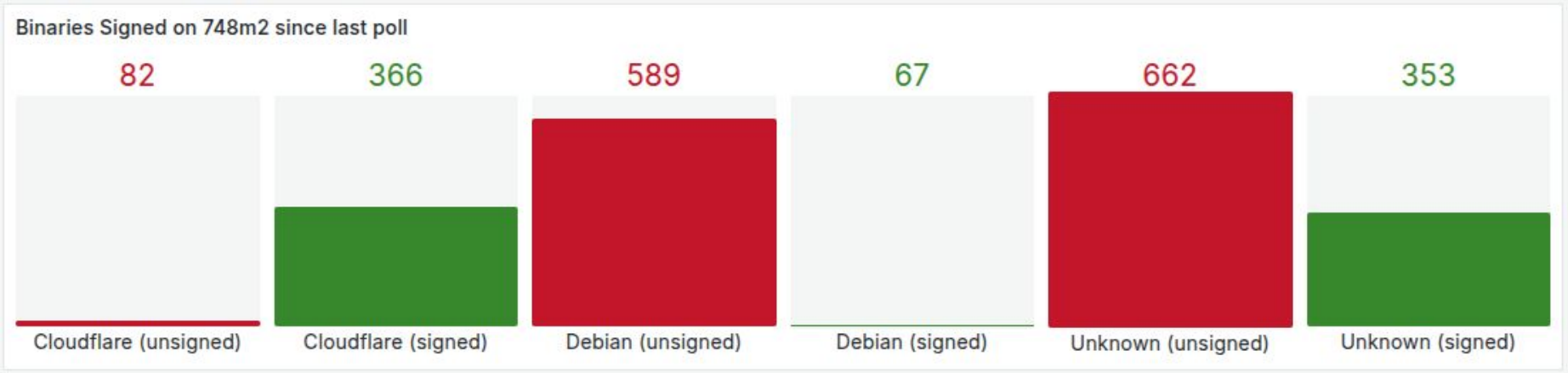
- Need a tool that can Mirror
- Modify newly mirrored packages
- Push modified packages into a Repository
  - With all the good stuff: gpg signatures, proper structures, hash-sums, etc
- Approaches we took
  - ftpsync, Reprepro
  - Writing a tool from scratch
  - Finally: Vanilla Aptly with Extra steps

“Trusted” Aptly



Are we signed yet?

```
# cat /sys/kernel/security/integrity/ima/ascii_runtime_measurements
...
10 a4b6af2f7897a10c783cbbcc89b17e50b1db5ae9 ima-sig sha256:617a2bd7e2c8410d044e85741bb39007c2d484675d6e3c8e557b093dadc7f53d filename
10 aa76cbbd3ed623d382f403ea02eac97bf8cf35ec ima-sig sha256:c56171d28f30f89ed1de5170ddd3ca87278cd72ebce0af3df15d63c0788c51ab /usr/bin/foo
10 1cc4929ef8065883e8d6a4cb9154ce64128ab905 ima-sig sha256:8365a65f36ff69f2d506900f7193cb745e355be5a0b9b169aaccaf54c4050f88 /usr/lib/x86_64-linux-gnu/libfoo-x86-64.so.2
10 3064f1207adf7243f99d4fbf57ab28087b7e942a ima-sig sha256:98da253897d6381415d88d465dddc581ac9c336db719323ce8ca4b99f52d0f93 /usr/lib/x86_64-linux-gnu/libfoo.so.1.1.0
```



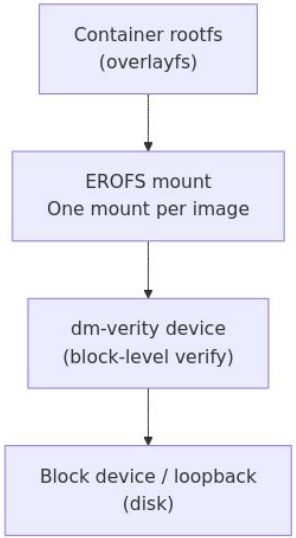
IMA Appraise Selective Enforcement

- ima_appraise=selective-enforcement
- In between state of enforce and log
- Missing signature + infraction does not result in EPERM
- Binary mutation with corresponding signature does result in EPERM
- https://github.com/cloudflare/linux/tree/master/patches

```
diff --git a/Documentation/admin-guide/kernel-parameters.txt b/Documentation/admin-guide/kerne
index 1518343bbe22..09a10ba686f6 100644
--- a/Documentation/admin-guide/kernel-parameters.txt
+++ b/Documentation/admin-guide/kernel-parameters.txt
@@ -2072,7 +2072,7 @@
                        Set number of hash buckets for inode cache.

        ima_appraise=   [IMA] appraise integrity measurements
-                       Format: { "off" | "enforce" | "fix" | "log" }
+                       Format: { "off" | "enforce" | "selective-enforce" | "fix" | "log" }
                        default: "enforce"

        ima_appraise_tcb [IMA] Deprecated.  Use ima_policy= instead.
diff --git a/security/integrity/ima/ima.h b/security/integrity/ima/ima.h
index 3c323ca213d4..3d5efaf9cb62 100644
--- a/security/integrity/ima/ima.h
+++ b/security/integrity/ima/ima.h
@@ -423,6 +423,7 @@ int ima_policy_show(struct seq_file *m, void *v);
 #define IMA_APPRAISE_FIRMWARE  0x10
 #define IMA_APPRAISE_POLICY    0x20
 #define IMA_APPRAISE_KEXEC     0x40
+#define IMA_APPRAISE_SELECTIVE_ENFORCE 0x80
```

Container Images

EROFS + dm-verity



The good

- Block level verification
- IMA can be turned off for that device
- EROFS is read-only by default, good efficiency file system

The bad

- OCI spec support??
- N... image builders
- Runtime extraction

Container Images

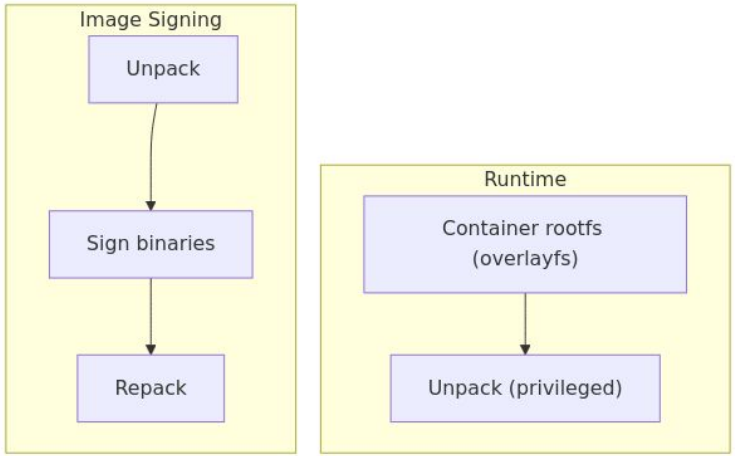
The good

- Signed prior to upload to registry
- Eventually consistent
- Native tar extended attribute support (unlike Debian)

The bad

- Repetitive signing
- Eventually consistent
- Privileged process required

Ship extended attributes with image



CLOUDFLARE

Thank you!
Questions?

fred@cloudflare.com

Links

- https://github.com/cloudflare/linux/tree/master/patches
- https://blog.cloudflare.com/thanksgiving-2023-security-incident/
- https://ima-doc.readthedocs.io/en/latest/index.html
- https://blog.linuxplumbersconf.org/2016/ocw/system/presentations/3933/original/FileSignaturesNeeded.pdf
- https://wiki.debian.org/Teams/Dpkg/Spec/MetadataTracking
- https://developers.cloudflare.com/r2/buckets/public-buckets/#custom-domains
- https://blog.cloudflare.com/using-cloudflare-r2-as-an-apt-yum-repository/