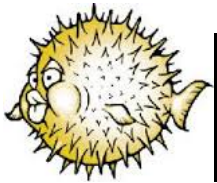
Breaking bad: there and back again

Marc Espie <espie@openbsd.org>, <espie@lse.epita.fr>



September 22, 2017

The case of the missing signatures

I broke pkg_add

A few months ago, suddenly, `pkg_add` didn't work with old packages.

```
pkg_add zarafa-7.2.4p1.tgz
file:./zarafa-7.2.4p1.tgz: unsigned package
```

Actually not missing

wtf just happened

- we changed the way signature worked
- in incompatible ways
- ... so old tools couldn't cope with the new one (expected)
- ... and new tools couldn't see the old one (worse)

Why – old style

old style signatures

- every file is checksummed individually (md5, then sha256)
- checksums are stored in the packing-list
- the packing-list is signed

Benefits and drawbacks

- on-the-fly checks
- can stop extracting before the end
- need to rewrite the package for signing
- *have to ungzip package first!*

New style

Inside-out

- we store the signature outside gzip
- using standard gzip fields (comment)...
- ... so that nothing is unpacked before checking

A typical signature

```
untrusted comment: verify with openbsd-62-pkg.pub
RWRvEq+UPCq0VMAVTDQnijwATE9Tmi5SRbfjKMQ6bD/nRXwweq58X9WSGS2UreG6wNhKrJr5QGoU
date=2017-09-22T12:09:32Z
key=/etc/signify/openbsd-62-pkg.sec
algorithm=SHA512/256
blocksize=65536

b8e68f4eda801af84443b8d6e4af09492247c0d54fa8f6fa80207c1afd56ef3d
76c0a74f9493a92b90fd4d1f52e037b3c5f8182037c10612c97c7734012789f5
```

Hear no evil, see no evil

Why it breaks

it's just that `pkg_add` no longer sees signatures.
It's all handled by `signify` now.
Hence the message.

Decisions, decisions

Necessary breakage

This has security implications, so we decided to break.
The error messages could be more helpful.

Small sample

This happens all the time

We change internal details of packages and ports all the time.
The end user seldom notices.
OpenBSD releases are supported for a year.
OpenBSD package format is supported for much longer (in general)

Design

old shitz

- Don't break backward compatibility if we can
- for instance I have `OpenBSD::PackingElement::Old` for old keywords
- ... cleaned up every 5 or 6 years.

Hackathons

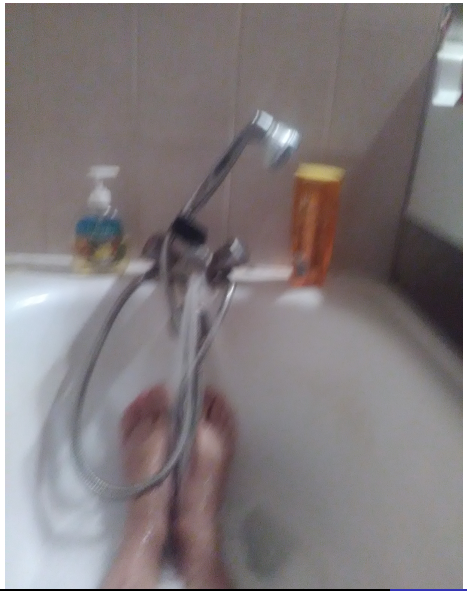
Hackathons

Rat lab

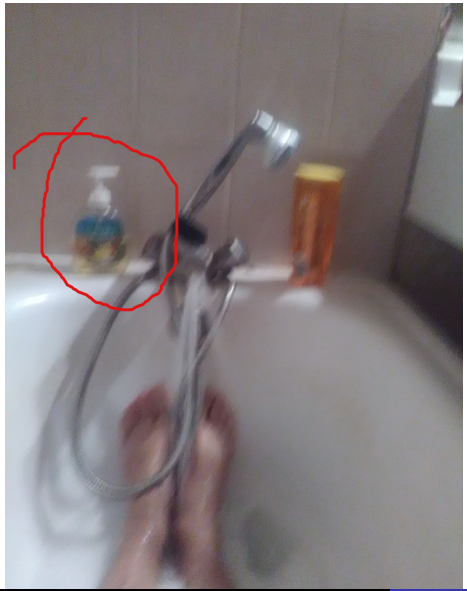
Work environment

Our fearless leader

Work environment



Work environment



Work environment

The file format

I inherited gzip'd tarballs.
It is a perfectly standard format.
Each time we improve things, we see whether we can stay with it.
Sometimes it evolves when other tools are ready.
Serendipity: chunked gzip for signature, then for speed.

The repository format

At first we didn't have any index, didn't need it.
No update: so each package is self-contained.
Then it became a game: how far can it go.
Shearing snapshots: each package is self-signed.

Limits to the model

Based on package names.
Open each package to check further.
Handling is external (ftp command)
That becomes a problem.

Documentation

At first `pkg_add` could just add packages. Then it became able to replace packages, then full upgrade.
The documentation is an history lesson.
Sooner or later, we need to remove old stuff.
A better command has less options.

Dependencies

The initial model was mostly stolen from FreeBSD.
But there were differences.
It became very complex thanks to multi-packages in part.
So I streamlined it, with my lab rats.

Case study

`@version`

Two times where everything got updated:

- `typedef unsigned long` breaks all C++.
- gcc to clang breaks all C++

Is there something wrong with C++ ?

We did bump every package the first time.

Case study

Wrong approach: dependency.
Can be self-contained.
Write version using -V n1 -V n2 -V n3... to mean $n_1 + n_2 + n_3$.
That way, you can get MI and MD parts easily.
So little code, it went in and worked.

I am a tease

Talk about the future a bit.
This is obviously never going to end.

- talking to ftp becomes awkward and slow
- we probably need a framework for testing updates. Or maybe better checks
- we need to get better diagnostics when it fails
- I still don't have on-the-fly update