

A wooden sandbox is the central focus, built from dark wood and filled with light-colored sand. Inside the sandbox, there are several toys: a blue shovel, a red bucket, a yellow bucket, and a green bowl. The sandbox is situated in a lush green yard with various trees and bushes in the background. A large, dark, mossy rock is visible to the left of the sandbox. The scene is brightly lit, suggesting a sunny day.

Playing in the sandbox

Guillaume Emont - Igalia
guijemont@igalia.com

I. What do we want to solve?

<http://guij.emont.org/blog/2011/11/10/gstreamer-and-opencv-for-image-stabilisation/>

but rather in the order I should have researched them: starting with a simpler problem, then getting into the complications of my balloon problem.

The simpler problem at hand is presented to you by Joe the Hippo:



Internets = cute videos of [insert favourite animal]

the cameraman is, and the video was taken with a lot of zoom. The movement in that video stream has a particularity that can



Trustworthy?

00:10 / 00:14





Complex software written by human beings can have bugs.

`rm -rf /`

00:10 / 00:14







Ophiocordyceps unilateralis



The piece of software that handles untrusted video:

- is not evil in itself
- should be considered evil when it starts to handle untrusted data

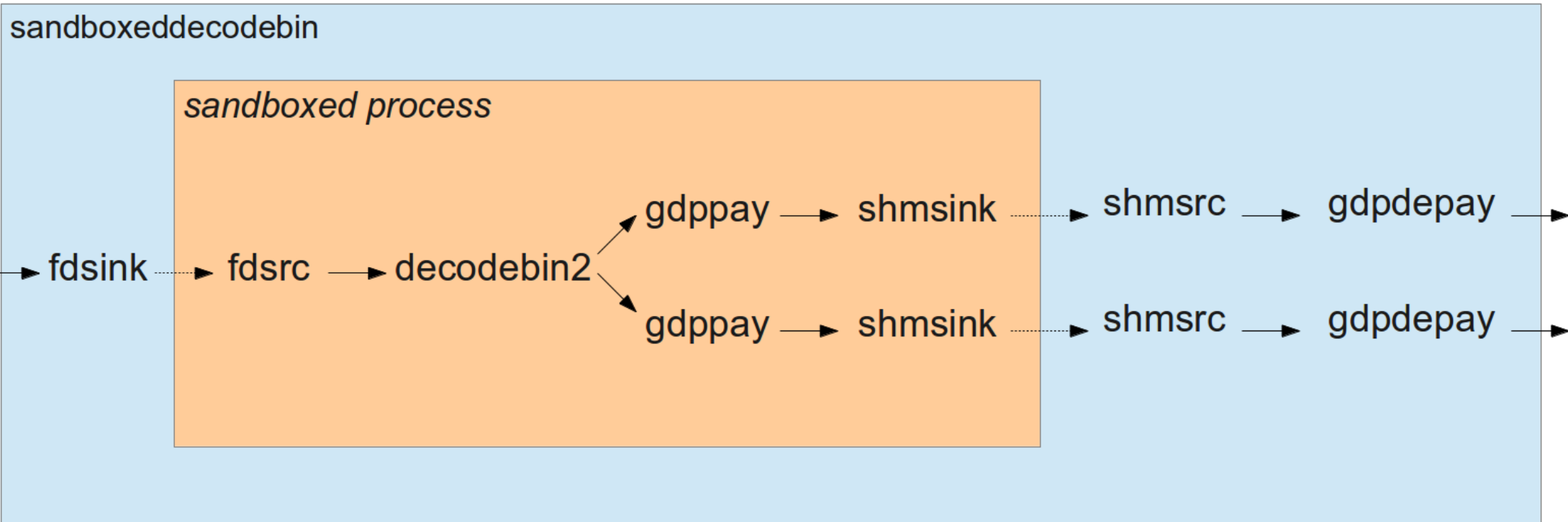
- 
1. Take resources
 2. Drop privileges
 3. Handle dodgy data


II. Experimentations



setuid-sandbox

sandboxeddecodebin



- 
1. Take resources ==> ->READY
 2. Drop privileges
 3. Handle dodgy data ==> ->PAUSED->PLAYING

Issues:

- resources acquired when going to PAUSED
- clean up

Potential solutions:

- finer grained sandbox
- modify elements to acquire everything at ->READY?
- add an all-resources-acquired signal?
- have a controlling process do the clean up

Other big limitations:

- no upstream communication => no seeking
- overhead: 720p ogg/theora on my i5: 20-30% -> 30-40% cpu

III. Usable sandbox implementations on Linux

Setuid-sandbox (CLONE_NEWPID + separate uid/gid + chroot)

- prevents access to data outside of chroot
- in kernel since 2.6.24
- ideally needs a pool of UIDs/GIDs
- not very granular

Seccomp

- only read(), write(), exit(), sigreturn()
- applied to a thread
- in kernel since 2.6.10

Seccomp + BPF

- filters on syscalls and their arguments
- libseccomp to make it easier
- in kernel since 3.5

SELinux

- by process
- quite fine grained
- rules set by administrator/distribution, developers
- not standard

IV. Going Forward

Better IPC: GstPadProxy, GstElementProxy & friends
=> control over a remote pipeline

Put the whole pipeline process in the sandbox
=> shouldn't be complicated with a granular sandbox

Architecture that is agnostic
to the sandboxing system used

Performances: profile, optimise

A close-up photograph of a glass filled with beer, topped with a thick, white head of foam. The glass is partially obscured by a dark grey semi-transparent text box in the lower-left corner. The background is dark and out of focus.

Thank You

guijmont@igalia.com
<http://guij.emont.org/blog/>
<https://github.com/guijmont/Sandboxed-Player>

Image Credits:

Sandbox: Public Domain by Hyena <http://en.wikipedia.org/wiki/File:Sandpit.jpg>
beer: CC BY-NC-SA 2.0 Martin Ibert http://www.flickr.com/photos/mar_ibert/
zombie ant: CC BY 2.5 http://en.wikipedia.org/wiki/File:Ophiocordyceps_unilateralis.png
bug: CC BY 2.0 by ThreeHeadedMonkey <http://www.flickr.com/photos/threeheadedmonkey/3856171871/>