Python in a sandbox

Holger Krekel Maciej Fijalkowski
Merlinux GmbH

PyCon 2009 - Rosemont

March 28 2009

This talk contents

- Problems with current sandboxing approaches
- PyPy sandboxing and virtualization
- A couple of demos
- Status, future, Q&A

What is a sandboxed python?

- A way to execute untrusted code separated from application
- Separation from each other
- Eventual integration with existing APIs
- Example: Google App Engine

Other solutions

- There is a lot of implementation's out there in the wild
- They do subset of:
- bytecode/source verification
- CPython source modification/monkey patching
- platform-level security (GAE)
- restrict python language to something harmless (zope's restricted python)

Problems

- Patchy approach - “we fix all places that might be potentially dangerous”
- Tradeoffs - either usability suffer or security is hard to control
- “Noone cracked it so far” approach is not “security by design”

“fixing all places manually”

sidenote: enter the “browser hack” challenge a week ago:

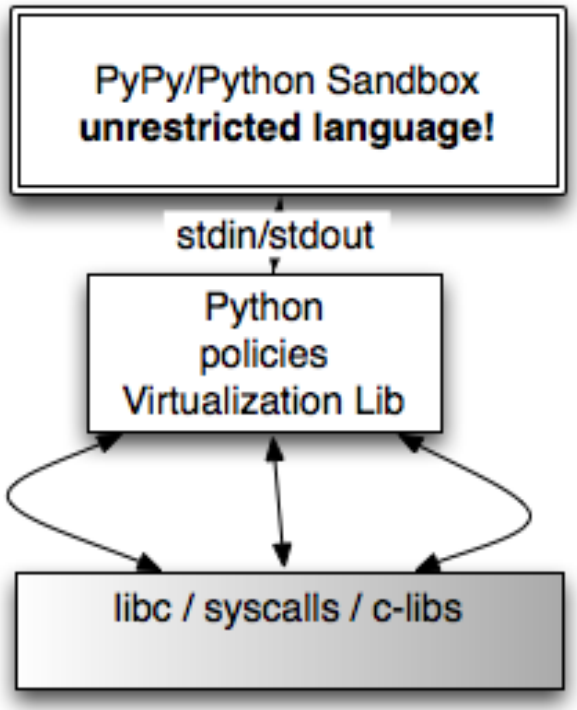
- **fully security-updated** IE8, Firefox, Safari access an URL
- hacker answers the URL access
- within a day above browsers were hacked

something is wrong with an approach that despite so much attention and effort is so easy to break

PyPy's sandboxing

automatically transform all C-lib/os calls in our low-level graph representation of the interpreter.

PyPy virtualized Interpreter



Places to security-review

- algos for transforming the graphs
- interaction code controler<->sandbox
- controler process code

Robustness and freedom!

Changes to the interpreter don't break sandbox!

Additional goodies

- Memory limit (GC support)
- CPU time limit

Drawbacks

- Each sandbox is in a separate process
- Sandbox doesn't have direct access to any APIs

How to use it today?

- translate pypy with --sandbox (takes a while)
- run using pypy_interact.py
- demo

Embedding in your web app

http://codespeak.net/svn/user/getxsick/django-sandbox/

Custom file access policies

code your own policy in plain python

What next?

- come up with nice methods of integrating with App code, try PyPy's transparent proxies?
- Improve docs, spawn separate project
- get funding for teaching and helping companies to make full use of it.

Q&A

Maciej Fijalkowski, Holger Krekel at http://merlinux.eu
Project webpage: http://pypy.org