infrastructure as code **might** be literally impossible

joe damato
packagecloud.io

hi, i’m joe

i like computers

i once had a blog
called timetobleed.com

@joedamato

packagecloud

packagecloud.io

@packagecloudio

follow along

blog.packagecloud.io

infrastructure as code **might** be impossible because nothing works.

code

- what is code?

code

makes Computer do complicated stuff in small steps

code

each small step is made up of a keyword (and other stuff)

code

and so the keywords
let you use Computer

code

different languages have different tradeoffs

code

some languages are difficult

code

assembly

C

C++

…

so, you need to use them defensively

Story Time

opteron revision E

+

mysql

code

some languages are perceived as easy, but are **terribly difficult**

code

Ruby

Perl

Bash

…

An Aside

You must be an expert in C to write good, fast Ruby/etc

But

that’s part of a different talk called:
“high level languages don’t exist”

hard or impossible to use these languages defensively enough

Story Time

MRI segfaults

MRI threading

INSIDIOUS

Thus

Your code does things outside of your reference frame

Unless

You’ve read every line all the way down (you haven’t).

OK.

infra code

makes Computer do complicated stuff in small steps

infra code

has really high level ‘keywords’

```
package “blah-pkg” do
  version ”1:1.2.8-1”
  action :install
end
```

```
package { “blah-pkg“:
  ensure => present,
  source => “https://packagecloud.io/...”,
  provider => rpm,
}
```

what if i told you

infra code operates outside of your reference frame, too

meaning

unless you’ve read every line all the way down…

you haven’t

OK.

some things you (probably) didn’t know

what if i told you…

an MRI bug once made puppet peg CPU usage

sigprocmask

a syscall used

via

[sg]etcontext

[sg]etcontext

used for threading and exception handling

“The “puppet” process spends 40-60% time in “system time”, which lengthens the time a single puppet run takes from a few minutes to > 20 minutes.”

I wrote a fix for this bug that was never accepted upstream

(hi)

http://timetobleed.com/fix-a-bug-in-rubys-configurein-and-get-a-30-performance-boost/

a friend working at a huge company told me that without that patch, **they couldn’t run puppet**.

(hi)

coincidence?

"We're working on rebuilding our entire client-side technology stack, so it takes fewer resources, runs faster, and is more maintainable."

- puppet blog

supposively

they are supposively rebuilding (some/all?) client side stuff in C++

similarly

OHAI-330
Ohai crashes on Solaris 11, Ubuntu 12.04 in mixins/command.rb: popen4

workaround

GC.disable / GC.enable

workaround

(The work around is to disable a major feature of the language.)

what if i told you…

it’s impossible to install a program securely on most linuxes

But, package mangers have GPG!!!11!!

No

YUM + GPG

tl;dr: doesn’t work most of the time and is nearly impossible to get it working

Story Time

pygpgme

repo_gpgcheck

gpg v3 signatures

gpg v3 signatures

```
%__gpg_sign_cmd %{__gpg} \
    gpg --force-v3-sigs --digest-algo=sha1 --batch --no-verbose --no-armor --passphrase-fd 3 --no-secmem-warning -u "%{_gpg_name}" -sbo %{__signature_filename} %{__plaintext_filename}
```

(hi)

Story Time

sslverify

APT + GPG

tl;dr: doesn’t work most of the time and is nearly impossible to get it working

Story Time

debsigs vs dpkg-sig

gpg signing deb packages is pointless

XML policy documents

/etc/debsig/policies/DDDF2F4CE732A79A/hi.pol

```
<?xml version="1.0"?>
<!DOCTYPE Policy SYSTEM "http://www.debian.org/debsig/1.0/policy.dtd">
<Policy xmlns="http://www.debian.org/debsig/1.0/">

  <Origin Name="test" id="DDDF2F4CE732A79A" Description="Test package"/>

  <Selection>
    <Required Type="origin" File="debsig.gpg" id="DDDF2F4CE732A79A"/>
  </Selection>

   <Verification MinOptional="0">
    <Required Type="origin" File="debsig.gpg" id="DDDF2F4CE732A79A"/>
   </Verification>
</Policy>
```

oh, and, um…

Both are vulnerable to replay attacks

Neither deal with key revocation

Both are vulnerable to several GPG related attacks

(these are some of the ∞ reasons why you should use packagecloud.io)

what if i told you…

the CA certificate bundle you use revoked AWS’s SSL CA ?

but before i explain that, periodic reminder that trusted CA certs come from this URL

periodic reminder

curl.haxx.se

OK, anw…

bento, vagrant, kitchen

opscode-centos-5.11
on or around 2015-02-23
updated the CA cert
bundle

bento, vagrant, kitchen

resulting in a bundle with AWS’s CA being revoked

curl.haxx.se

“We in the curl project didn't anticipate anything of this. We get the data from the Mozilla project and they changed the properties. We've run the same script daily since a long time. One day the output changed to this.” - http://curl.haxx.se/mail/archive-2014-10/0068.html

bento, vagrant, kitchen

and then accessing S3 from vagrant boxes produced by bento stopped working read more on chef’s blog: “Bento Box Update for CentOS and Fedora”

https://www.chef.io/blog/2015/02/26/bento-box-update-for-centos-and-fedora/

PS

debugging SSL is really difficult

BTW QUICK THING

cognitive load

“cognitive load refers to the total amount of mental effort being used in the working memory"

cognitive load

at some point you have to wonder: when does it become too much?

cognitive load

“just read the code” is impossible because you need to read millions of lines of code

cognitive load

“People changing our Chef recipes to make something work for them, but then breaking everyone else’s [stuff] is practically constant at [company] right now.”

- my friend who works at [company]

But

this is all part of a different talk called:
“the effect of capitalism on computing”

anw

what if i told you…

you can DoS a machine with yum/apt metadata?

when apt/yum
request metadata,
just reply with a never
ending file.

ya but i’m not an official mirror lol ??

(ya tu sabes)

“Debian and CentOS listed the mirror within a few hours, and Fedora listed the mirror in minutes.”

- academic paper

what if i told you…

createrepo
generates
incorrect metadata
sometimes?

rpmUtils bug

rpmUtils uses python’s find method instead of rfind when splitting version strings

rpmUtils bug

version strings with two ‘-‘ in them are split on the first, not the last (incorrect)

rpmUtils bug

resulting in incorrect package metadata

rpmUtils bug

this is live on the official mirrors right now

rpmUtils bug

i filed a bug about it, but from the looks of it, it won’t be fixed.

OK, these are all cool stories, but…

what gives?

people are using infrastructure as code today though?

what gives?

indeed they are, with varying levels of success and in many cases great pain

IN MY OPINION

opinion 1

we won’t be able to have truly reproducible infrastructure until we figure out better ways of building computer systems.

opinion 2

each time you move to a higher level of abstraction, you need to know more stuff.

maybe cutting out some layers in between can make this more easily solvable?

opinion 3

we need to be more honest and responsible about our choices and analysis of technology.

opinion 4

huge companies making billions of dollars on top of these software systems should take the initiative to invest in making them better.

opinion 5

we haven’t found the “answer” yet. what we have is better than what we had, but we need to think bigger.

?

packagecloud.io

@packagecloudio