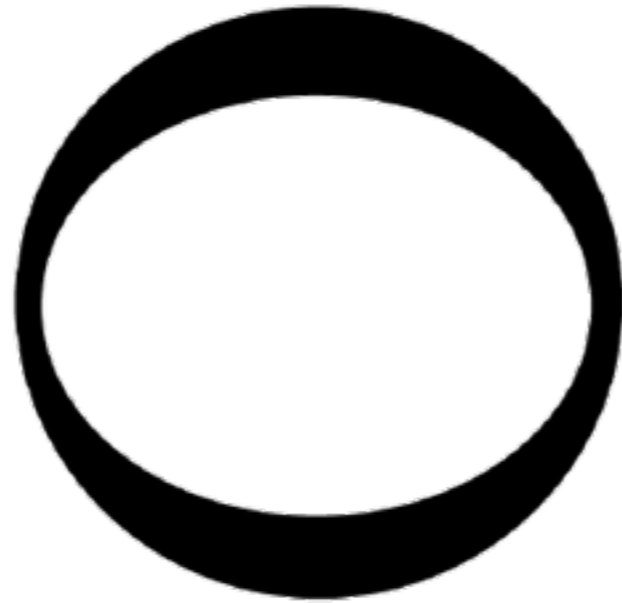


Is This Your Pipe?

Hijacking the Build Pipeline

\$ whoami
@r**g**bkrk



IP[y]:
IPython

OSS, Builds and Testing



Protecting pipelines

- ~~Need~~ Want benefits of continuous delivery
- Open source pathways to real, running infrastructure
- Community services with $> 200,000$ users



Build Pipeline Components

- **Source Control**
- **Continuous Integration**
- **Upstream Sources**

Real Sites

Need Secrets

What secrets?

- Cloud Credentials
- OAuth Secrets
- Integrate with   etc.

Managing Secrets

Not Managing
Secrets

Credentials Get Leaked

“I did not completely scrub my code before posting to GitHub. I did not have billing alerts enabled ... This was a real mistake ... I paid the price for complacency.”

-Rich Mogull



What can be done with Cloud Credentials?

- Build new infrastructure
- Delete your infrastructure
- Append SSH keys to your primary set
- Change root passwords
- “Redistribute” your DNS and Load Balancers



Secret Finding Demo

**Can't we just let people
know when they fuck up?**

gitsec/nanny

- Search repositories for security oops
- Email the original committer & owner of the project
- Let them know how to revoke keys, panic

Responses

- **“Wow, thank you. How did you find these?”**
- **“This is only a testing project”**
- **“I don’t even own this repository”**

```
config/initializers/  
secret_token.rb
```



What if you **need**
secrets for testing?

Travis CI

Continuous Integration
Build Platform



Travis CI

- Open Source, free for public repos
- git push -> web hook -> tasks
- Less control than Jenkins
- **Encrypted Secrets!**

language: python

python:

- 2.7

before_install:

- pip install invoke==0.4.0 pytest==2.3.5

install:

- pip install .

script: invoke test

Travis & Encrypted Secrets

Can we leak
decrypted secrets?

Update .travis.yml #1

Merged rgbkrk merged 1 commit into rgbkrk:master from gitsec:legit 26 minutes ago

Conversation 0

Commits 1

Files changed 1



rgbkrk commented 30 minutes ago

Owner



```
echo $SECRET_MESSAGE
```

Update .travis.yml

✓ 44cd1ae



rgbkrk merged commit a6af05a into rgbkrk:master from gitsec:legit 26 minutes ago

Revert





rgbkrk closed this 26 minutes ago

master - Update .travis.yml

#2 passed

ran for 3 sec
9 minutes ago

 Kyle Kelley authored and committed

[Commit cf76c08](#)  [#1: Update .travis.yml](#) 


```
1 Using worker: worker-linux-4-1.bb.travis-ci.org:travis-linux-17
2
3 $ git clone --depth=50 git://github.com/rgbkrk/secrets-in-public.git
11 $ cd rgbkrk/secrets-in-public
12 $ git fetch origin +refs/pull/1/merge:
13 $ git checkout -qf FETCH_HEAD
14 $ source -/virtualenv/python2.7/bin/activate
21 $ python --version
22 Python 2.7.6
23 $ pip --version
24 pip 1.5.4 from /home/travis/virtualenv/python2.7.6/lib/python2.7/site-packages (python 2.7)
25 $ echo "Life is good"
27 $ echo "$SECRET_MESSAGE"
28
29
30 The command "echo "$SECRET_MESSAGE"" exited with 0.
31
```

master - Merge pull request #1 from gitsec/legit

#3 passed

Update .travis.yml

ran for 5 sec
less than a minute ago

 Kyle Kelley authored and committed

[Commit a6af05a](#)  [Compare 4ac4b6a..a6af05a](#) 

```
1 Using worker: worker-linux-6-2.bb.travis-ci.org:travis-linux-2
2
3
4 $ git clone --depth=50 --branch=master git://github.com/rgbkrk/secrets-in-
5
6 $ cd rgbkrk/secrets-in-public
7
8 $ git checkout -qf a6af05a90a917cd803fdfa814fe58dfel2d9d269
9
10 $ export SECRET_MESSAGE=[secure]
11
12
13 $ source ~/virtualenv/python2.7/bin/activate
14
15 $ python --version
16 Python 2.7.6
17
18 $ pip --version
19 pip 1.5.4 from /home/travis/virtualenv/python2.7.6/lib/python2.7/site-packages (python 2.7)
20 $ echo "Life is good"
21
22 $ echo "$SECRET_MESSAGE"
23 Drink
24
25 The command "echo "$SECRET_MESSAGE"" exited with 0.
```

“Keys used for encryption and decryption are tied to the repository. If you fork a project and add it to travis, it will have a different pair of keys than the original.”

– *Travis CI*

Masquerade Process

1. Find repository with credentials
2. Do legitimate work on a feature or bug
3. Include your security oops ...
4. Profit

Speaker Transition!

**What's
the
Build
Pipeline?**

The Build Pipeline



Pypi

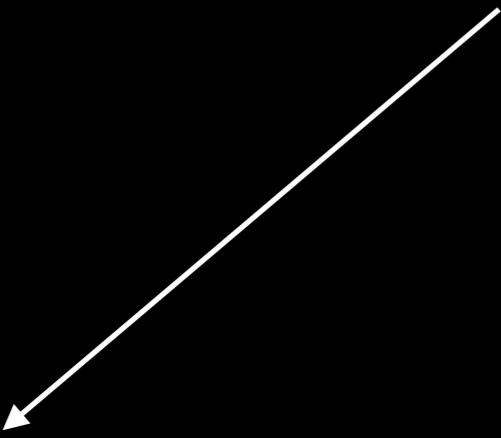


openstack™



Dev Box

OpenSSL



CI



Production



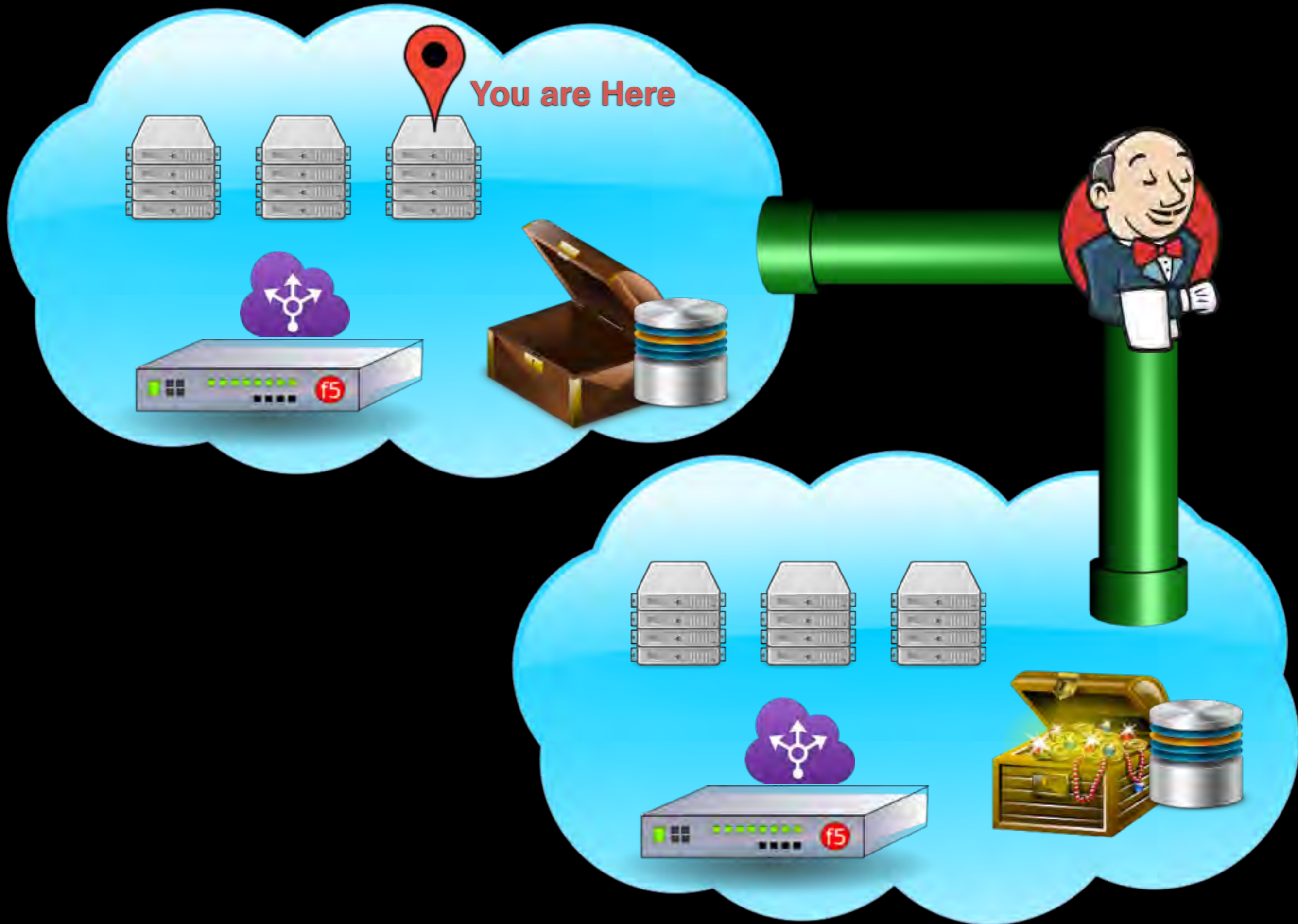
Contaminate the Pipeline

Compromise Everything.



Breaking into the Pipeline







Hijacking the Pipeline

Identifying CI in Code



```
jenkinsapi 0.2.20
```

```
from jenkinsapi import api
jenkins = api.Jenkins('http://your-jenkins-server/
api/python/')
job = jenkins.get_job('MyAwesomeJob')
build = job.get_last_build()
if build.is_good():
    print "The build passed successfully"
else:
    # Know that
    pass
```



OR



```
curl http://jenkins/job/$JOB_NAME/build -F  
file0=@PATH_TO_FILE -F json='{ "parameter":  
[{"name": "FILE_LOCATION_AS_SET_IN_JENKINS",  
"file": "file0"}]}' --user USER:PASSWORD
```

OR

```
wget -q --output-document - \  
'http://server/jenkins/crumbIssuer/api/xml?  
xpath=concat(//crumbRequestField,":",//  
crumb)'
```


🍒 The Low Hanging Fruit 🍒

```
Utilizing the Jenkins Python API...  
# create a malicious deploy & check status  
mal_job = jenkins.get_job('PWN')  
mal_build = mal_job.get_last_build()  
if mal_build.is_good():
```



The Not-So-Low Hanging Fruit. You need MOAR permissions.

```
Cloning  
into '<DIRECTORY>' ...  
Permission denied (publickey).  
fatal: The remote end hung up unexpectedly
```

```
Access Denied- USER@DOMAIN is missing the X permission
```

```
Building in workspace /var/lib/jenkins/jobs/Test  
Deployment/workspace
```

```
stderr: Host key verification failed.  
fatal: The remote end hung up unexpectedly
```

In other cases it may not...

```
Building in workspace /var/lib/  
jenkins/jobs/Test
```

```
ERROR: Could not clone repository
```

```
FATAL: Could not clone
```

WE'RE NOT THROWING IN THE TOWEL JUST YET



Challenge Accepted.

So many options....

Security Realm

- Delegate to servlet container
- LDAP
- Hudson's own user database

Authorization

- Logged-in users can do anything
- Legacy mode
- Anyone can do anything
- Matrix-based security

| User/group | Overall | | Job | | | Run | | | View | | | SCM |
|------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | Administer | Read | Create | Delete | Configure | Build | Delete | Update | Create | Delete | Configure | Tag |
| Anonymous | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| kohsuke | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

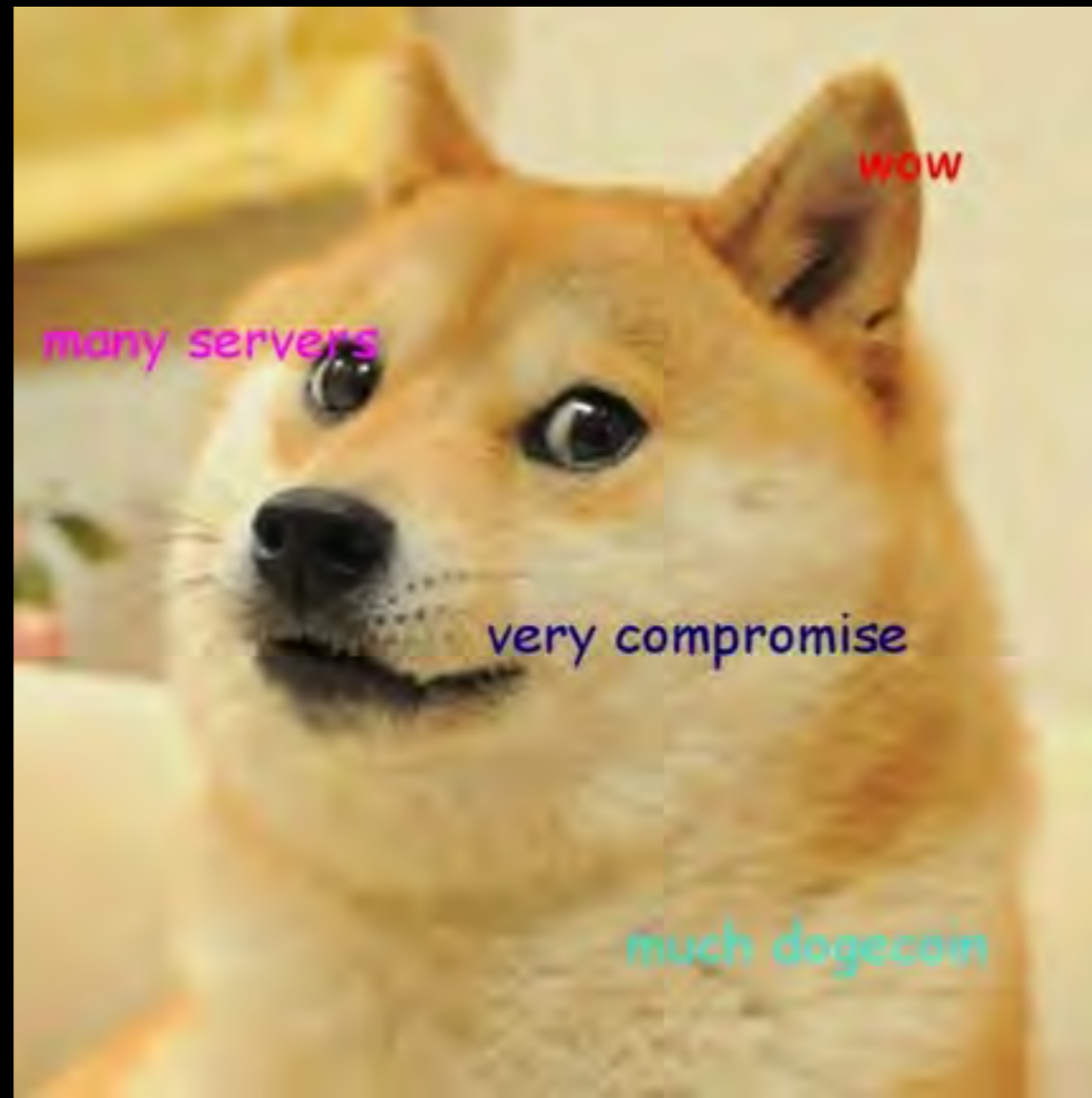
User/group to add:

The worst case scenario

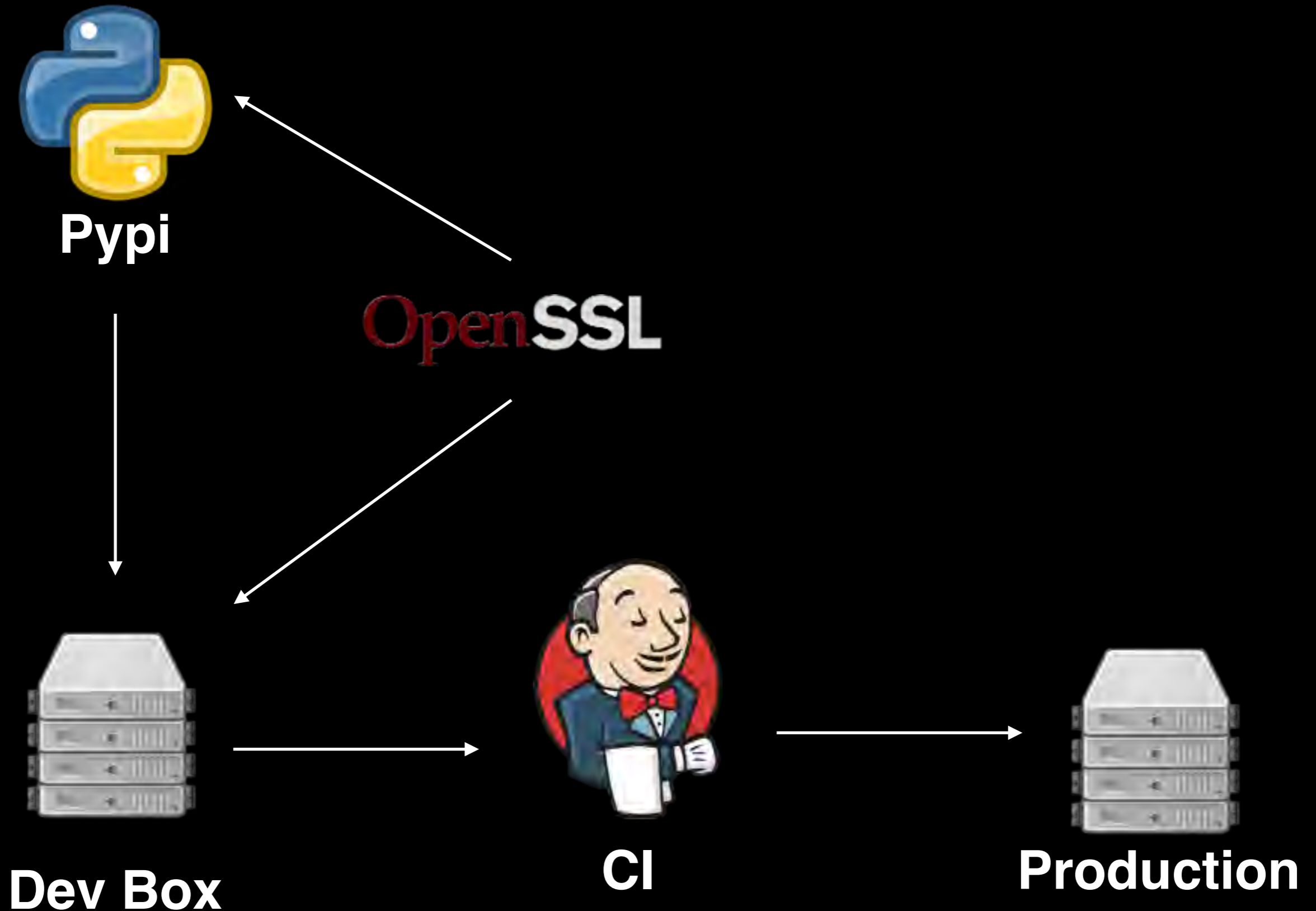


Contaminate the Pipeline

Compromise Everything.



The Build Pipeline

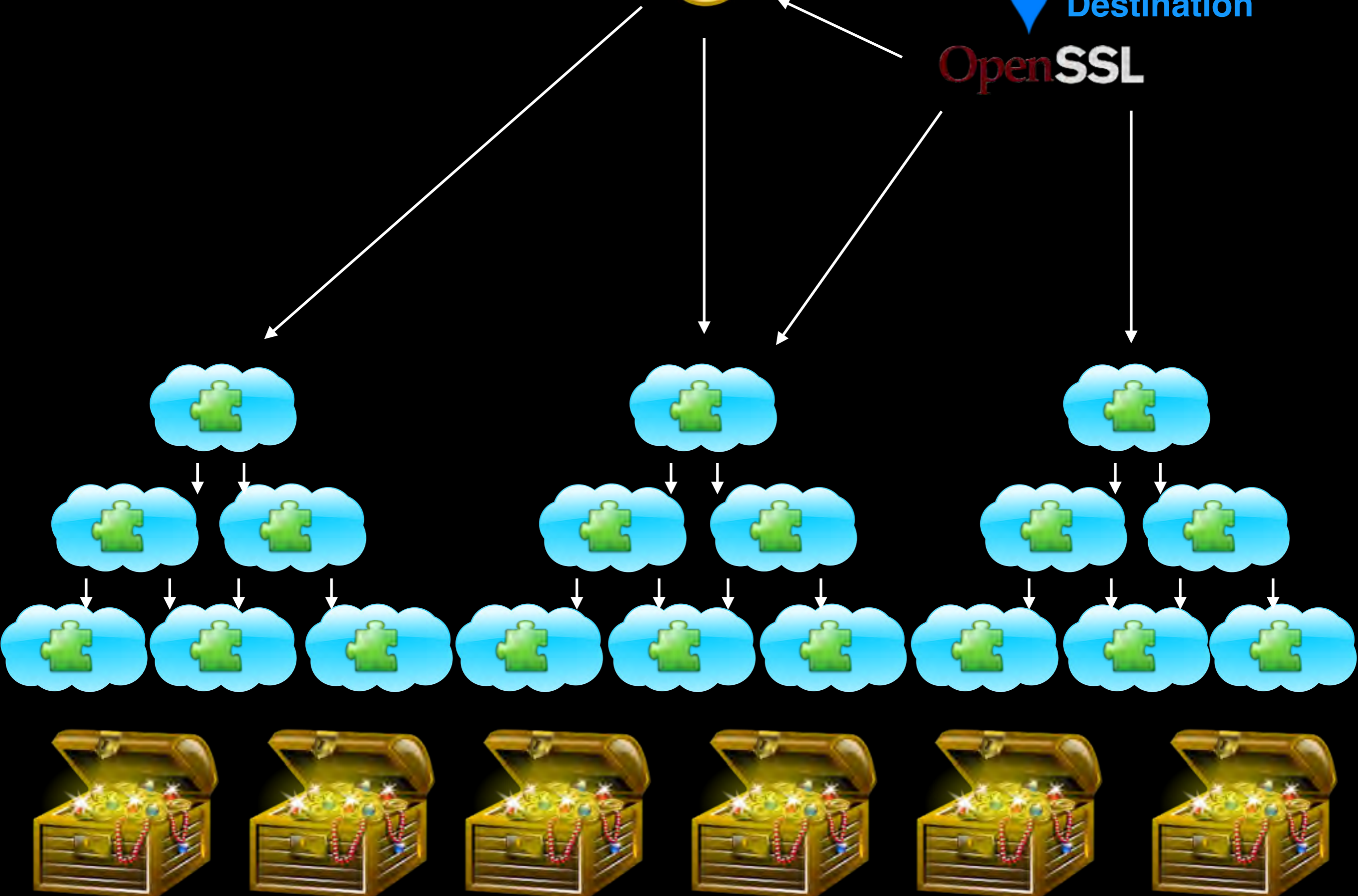


Destination



Destination

OpenSSL



Create your own Heartbleed



```
/* Enter response type, length and copy payload */  
*bp++ = TLS1_HB_RESPONSE;  
s2n(payload, bp);  
memcpy(bp, pl, payload);
```

```
r = ssl3_write_bytes(s, TLS1_RT_HEARTBEAT, buffer, 3 + payload + padding;
```

Remote Code Execution for all!

socket.recvfrom_into()

Some are easier
than others...

What defenses
do we have?

Take code review seriously.

```
///</summary>
///<param name="orderedChildIds">A collection of child ids.</param>
///<param name="movedChildId">The id of the moved child.</param>
public void ChangeChildSortOrder(int[] orderedChildIds, int movedChildId)
{
    + if (orderedChildIds == null)
    + {
    +     + throw new ArgumentNullException("orderedChildrenIds");
    + }

    + bool found = false;
    + ItemToItem moved = null;
    + ItemToItem previous = null;
    + ItemToItem next = null;
    + foreach (int orderedChildId in orderedChildIds)
    + {
    +     + ItemToItem current = ChildItems.FirstOrDefault(c => c.ChildId == orderedChildId);
    +     + if (current != null)
    +     + {
    +         + if (current.ChildItem.ItemId == movedChildId)
    +         + {
    +             + moved = current;
    +             + found = true;
    +         }
    +         + else
    +         + {
    +             + // TODO: Handle the case where the child is not found.
    +         }
    +     }
    + }
}
```



Gate your deploys.....

Or they will be my
deploys.