Secure DevOps
Luke Hinds

- Red Hat CTO Office.
- SEC Manager @ ODL
- OpenStack SEC Project PTL
- OPNFV Infra / Releng

Twitter @lukeahinds
#lhinds @ freenode
QUICK ROUND UP..

😊 What’s all the Fuss about? (Meet Alexsey)

😍 Good / Bad Practises (Quiz!)

😊 What we are doing in LFN.

😊 Where from Here?
An example of a CI Attack

An example of how devops environments are compromised
Alexsey Belan

- Latvian Citizen, 30 yr old.
- Yahoo Hack (500ml accts)
- Believed to work for Russian intelligence
- USD$100,000 reward
Alexsey’s Hacking Approach.

- Scouts social media (LinkedIn) to target engineers / employees

- Attacks often start from a compromised employee account

- Very Persistent, keeps going until he finds a way in
Autopsy of a recent Alexey Belan Attack
Single Sign On

Compromised Employee Gmail Acct

Internet-based JIRA access (SSO)

Found Jira issue revealing a legacy based SVN server
Eng Credentials Hacked

Discovered details from Jira were used to access the SVN server (existing CVE)

Privileges elevated and eng password hashes were taken from /etc/shadow and other files within SVN
GIT Access

Internet-exposed Git instance accessed via cracked engineer credentials

Exploit (JSP Shell) committed to Git, **self-approved**, and **merged**
Code > Production

Release tagged & went from test to production

JSP Shell allowed reading of database environment variables

‘mysqldump’ was run and user data was stolen
So this got me thinking!

DevOps (automation) allows us to shoot ourselves in the foot.
Attacks targeted at Engineers, not just non-technical uses!

Someone is putting lots of work into hacking Github developers

Dimnie recon trojan has flown under the radar for three years... until now.

DAN GOODIN - 3/16/2017, 1:24 AM
Is it really “the community” who are making changes?

We approve each others changes, but we can’t be sure it’s even really “us”.

The changes gets deployed into our Labs, Test Envs and...production!

Some of the change are complex, well...
Some of it, only makes sense to a small few..

I feel Dirty (self.devops)
submitted 5 months ago * by StephanXX

I have to confess to how dirty I feel.

I now have Jenkins (which runs on Java) that calls a Jenkinsfile (which is Groovy) which calls a python script (which make my skin crawl) that ingests YAML, then using Jinja2 string substitution from the YAML values, emits a final Dockerfile, a bash test script that calls Gradle, then a bash build script that does a docker build and then a docker push.

I wrote all of it. I don't think anyone should ever let me near a computer again.
So what can be done and what should be avoided?
What you should do?

2FA on everything your developers touch! Gmail, VPN’s etc.

Patch your systems like your business depends on it (it does).

Get developers trained up around security

Audit code, wikis, intranets for information leaks on a regular basis!
What should you not do?

Pull in objects on blind trust with no integrity checks

Hard code anything with a security context (keys, passwords, credentials)

Blindly pull in upstream dependencies.

It’s open source man and we all love each other!
Some tooling we have developed / integrated as gate checks...
Anteater

gate scan of code / objects for:

- Secrets Leakage (keys, creds etc)
- Credential Files sought out
- Vulnerable Strings & Exploits
- IPs and URLs scanned against blacklists
- Binaries are virus scanned
- Open Framework allows easy customization....

Demo:
https://asciinema.org/a/eZViK9M7yd0gD3asEVQIDi8DV

Used by OPNFV gate, security researchers, Red Hat and ZTE Eng:
private_key:
regex: -----BEGIN\sRSA\sPRIVATE\sKEY----
desc: "Possible leak of a private key"

all_interface:
regex: 0\.0\.0\.0
desc: "Interface listening on all interfaces - may break security zones"

angular_version:
regex: <script.src.*ajax\.googleapis\.com\ajax\VangularjsV1.*</script>
desc: “Old version of angularjs found, please use 2.x”
Anteater - File name scan

- (irb|plsql|mysql|bash|zsh)_history
- \.swp # vim swap file
- id_rsa
- aws_secret_access_key
- secret_token.rb
- jenkins\plugins\publish_over_ssh\BapSshPublisherPlugin\xml
- \.gem\credentials
- \.dockercfg
Anteater - 1.0 Release

- Github anteater/anteater
- Less False Positives / Noise
- Rate Limiting (Token Bucket - Redis)
- Virus Total API

INFO - Found what I believe is a URL: <script src = "https://coin-hive.com/lib/coinhive.min.js"></script></script>
INFO - Rate limit clear
INFO - Report found, job complete for https://coin-hive.com/lib/coinhive.min.js
ERROR - https://coin-hive.com/lib/coinhive.min.js is recorded as a malicious site by ADMINUSLabs
ERROR - https://coin-hive.com/lib/coinhive.min.js recorded as a malware site by Yandex Safebrowsing
ERROR - Full report available here: https://www.virustotal.com/#/url/2a6ae27b9692354ebbb693594f3aca910dc3b0418063c69859bd80bca8a6a8c6/detection
Anteater - supports multiple CI build systems
Anteater - active in OPNFV gate

<table>
<thead>
<tr>
<th>File Path</th>
<th>Comments Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>yardstick/cmd/commands/task.py</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>History</th>
<th>Mar 26 7:55 AM</th>
<th>Mar 26 7:55 AM</th>
<th>Mar 26 7:56 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ross Brattain</td>
<td>Uploaded patch set 1.</td>
<td>Jenkins Ericsson</td>
<td>Patch Set 1:</td>
</tr>
</tbody>
</table>

File contains violation: /home/opnfv/anteater/yardstick/yardstick/cmd/commands/task.py
Flagged Content: output_file_default = /tmp/yardstick.out
Matched Regular Exp: /tmp/
Rationale: tmp directories are risky. They are world writable and easily guessed
Please visit: https://wiki.opnfv.org/x/5oeY
Other projects..
Bandit

Python Security Lint scanner:

- Command Injection
- Unsafe function / libraries
- XSS exploits
- Insecure Permissions
- Many more...
findsecbugs

- Part of ODL CI Build
- Covers all projects (odlparent)
- Detects Java Exploits
- OWASP TOP 10 and CWE coverage
Quiz Time!

Some good and bad examples!
curl http://example.com/install.sh \| sudo bash

Good or Bad?
ssh-keygen -b 2048 -t rsa -f /tmp/sshkey -q -N ""

Good or Bad?
TMPDIR="$(mktemp)"

Good or Bad?
curl -SLO "https://nodejs.org/dist/latest/node-v8.1.4-linux-x64.tar.gz"

curl -SLO "https://nodejs.org/dist/latest/SHASUMS256.txt.asc"

gpg --verify SHASUMS256.txt.asc

if grep node-v8.1.4-linux-x64.tar.gz SHASUMS256.txt.asc | shasum -a 256 -c ; then
echo -e "Integrity Checks passed!\n"
do_stuff
else
  echo -e "Bad signature!\n"
  exit
fi

Good or Bad?
Where from Here?
Where from Here?

- Encourage wider use / feedback / contributions in Anteater
- Explore more sec checks at gate
- Developer Enablement - git book (openci.io)
- Key / Secrets Management in CI
- Security Dependency Check
- Database
THANKS!

Any questions / ideas?
You can find me at
- @lukeahinds
- lhinds@redhat.com
- #lhinds @ freenode