Python application security auditing with Bandit

Michael Scherer

mscherer@redhat.com

Who am I ?

Sysadmin @

Security ?

C I A

Confidentiality

Integrity

Availability

Different levels

Infrastructure

Firewall, WAF, etc

Configuration

Redundancy

And everything else

Code

Cost of a update

Detect as soon as possible

Static analysis

Clang, Coverty, etc

Focused on C

Few tools for python

Bandit

Openstack

Walk the AST

Use a blacklist

Plugins for fine grained check

Numerous false positives

Various alerts severity

A few examples

YAML

```
import yaml
yaml.load(astring)
```

---

upgrade_helper: !!python/object/apply:eval
["eval(….

SQL Injection

```
query='SELECT u.admin
FROM  users AS u

WHERE u.name
= {}'.format(username)
```

username = 'foo OR u.admin
= 1 LIMIT 1'

Use of a ORM

XSS

Javascript returned to the user

Jinja

No default protection

/tmp

Predictable name

Attack with symlink, etc

SSL v2, certificat not verified, marshal, pickle, use of exec, etc, etc

Data with multiple semantic

What to do if you find something ?

Estimate the problem

Read the source code

Estimate severity

Contact a specialist

Contact the project...

...in a private way

Get a CVE identifier

Request to Linux distributions

List oss-sec

Bandit do not
do everything

2 examples

softwarecollection.org

github.com/misli/softwarecollections/commit 303de6df36727

Injection in a rpm build

Ipsilon

Provider
SAML/Openid/persona

CVE-2015-5301

No verification of permission

Where to find more ?

owasp.org

Review security patchs

Make sure to add CVE ID in changelog (and commit)

Updates list on LWN

Start your own audit

Look at critical code

?

Contact:

Twitter : no account

Linkedin : no account

Mail : misc@{redhat.com,zarb.org}

Irc : misc on freenode (and others)