Software supply chain risks of using 3rd party repositories

ALEX DEVRIES

MARCH 21, 2019

About me

- Embedded operating systems
- Security incident response
- Security development lifecycle
- Penetration testing
- Paid to worry
- Not a Python developer

What I’m going to show you

- Security risks of the supply chain
- Using Python’s package repository as an example
- The problem with pycrypto
- Comparison to other repositories
- What you should look for

The Dev Sec Ops problem

- Everyone is responsible for security
- More people Getting more people to develop with security can increase exposure unless there's a process to follow
- Most developers don't understand attack surfaces

The supply chain is an attack surface.

Alex deVries

The security nihilist's view

Stereotypical hacker

Open source project

Contributes malware

Repository

Production applications use malware

App 1

App 2

App 3

App n

Production applications use malware

Hacker profits!

- Without proper controls, a hacker could install code that would be distributed to many production applications

Some of it might be true!

- Docker Hub
  - hosts 17 container images with backdoors (May 2017 – May 2018)
- NPM
  - Through typo-squatting, 37 libraries would exfiltrate environment (Aug 2017)
  - getcookies package found to have a backdoor (May 2018)
- Python's pip/pypi
  - By creating packages with similar names, hackers upload malware that is used (Sept 2017)
  - hacker tampers existing ssh-decorate package to insert malware (May 2018)
  - And more!

Checking for known vulnerabilities

- Static analysis tools are typically intended for writing new code
- Dependency checkers will look for everything that's being used
- Rulesets constantly evolve as new vulnerabilities are found
- Examples:

About Pypi

- Backend to pip, which will resolve dependencies and install required packages
- Online repository of thousands of 3rd party libraries
- Makes using complex libraries very easy

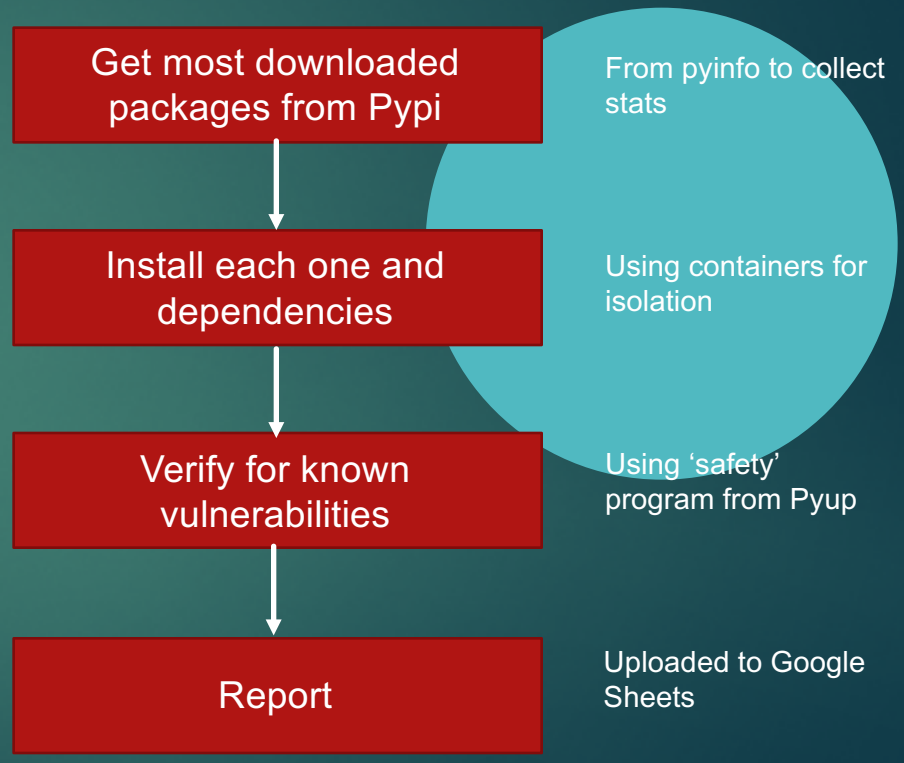
```
adevries@ubuntu:~$ pip install --user opencv
```

- What could possibly go wrong?

Pypi vulnerability scanner

- pypi-vuln analyzes the frequency of known vulnerabilities in pypi

github.com/alexdevsec/pypi-vuln



Alex deVries

Pyup's safety vulnerability scanner

- Checks what packages were actually installed
- Verifies them against a list of known vulnerabilities
- Priced attractively!

```
developer@48cf6caf8fd3:~$ ~/.local/bin/safety   check --full-report
```

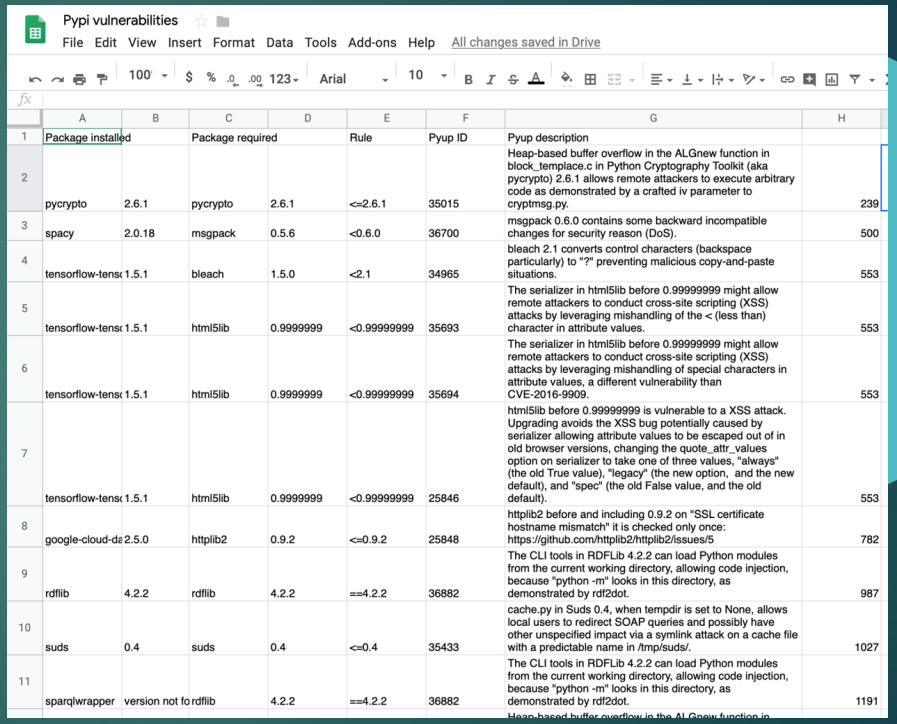
```
                         /$$$$$$            /$$
                        /$$__  $$          | $$
          /$$$$$$$  /$$$$$$ | $$  \__//$$$$$$  /$$$$$$   /$$   /$$
         /$$_____/ |____  $$| $$$$   /$$__  $$|_  $$_/   | $$  | $$
        |  $$$$$$   /$$$$$$$| $$_/   | $$$$$$$$  | $$     | $$  | $$
         \____  $$ /$$__  $$| $$     | $$_____/  | $$ /$$| $$  | $$
         /$$$$$$$/|  $$$$$$$| $$     |  $$$$$$$  |  $$$$/|  $$$$$$$
        |_______/  \_______/|__/      \_______/   \___/   \____  $$
                                                             /$$  | $$
                                                            |  $$$$$$/
 by pyup.io                                                  \______/
```

```
REPORT
checked 25 packages, using default DB
```

package	installed	affected	ID
pycrypto	2.6.1	<=2.6.1	35015
Heap-based buffer overflow in the ALGnew function in block_templace.c in Python Cryptography Toolkit (aka pycrypto) 2.6.1 allows remote attackers to execute arbitrary code as demonstrated by a crafted iv parameter to cryptmsg.py.			
cryptography	2.1.4	>=1.9.0,<2.3	36351
python-cryptography versions >=1.9.0 and <2.3 did not enforce a minimum tag length for finalize_with_tag API. If a user did not validate the input length prior to passing it to finalize_with_tag an attacker could craft an invalid payload with a shortened tag (e.g. 1 byte) such that they would have a 1 in 256 chance of passing the MAC check. GCM tag forgeries can cause key leakage.			

Pypi-vuln output

- Tool generates a spreadsheet showing vulnerable package and everything it affects



Pypi vulnerabilities

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

	A	B	C	D	E	F	G	H
1	Package installed		Package required		Rule	Pyup ID	Pyup description	
2	pycrypto	2.6.1	pycrypto	2.6.1	<=2.6.1	35015	Heap-based buffer overflow in the ALGnew function in block_templace.c in Python Cryptography Toolkit (aka pycrypto) 2.6.1 allows remote attackers to execute arbitrary code as demonstrated by a crafted iv parameter to cryptmsg.py.	239
3	spacy	2.0.18	msgpack	0.5.6	<0.6.0	36700	msgpack 0.6.0 contains some backward incompatible changes for security reason (DoS).	500
4	tensorflow-tens	1.5.1	bleach	1.5.0	<2.1	34965	bleach 2.1 converts control characters (backspace particularly) to "?" preventing malicious copy-and-paste situations.	553
5	tensorflow-tens	1.5.1	html5lib	0.9999999	<0.99999999	35693	The serializer in html5lib before 0.99999999 might allow remote attackers to conduct cross-site scripting (XSS) attacks by leveraging mishandling of the < (less than) character in attribute values.	553
6	tensorflow-tens	1.5.1	html5lib	0.9999999	<0.99999999	35694	The serializer in html5lib before 0.99999999 might allow remote attackers to conduct cross-site scripting (XSS) attacks by leveraging mishandling of special characters in attribute values, a different vulnerability than CVE-2016-9909.	553
7	tensorflow-tens	1.5.1	html5lib	0.9999999	<0.99999999	25846	html5lib before 0.99999999 is vulnerable to a XSS attack. Upgrading avoids the XSS bug potentially caused by serializer allowing attribute values to be escaped out of in old browser versions, changing the quote_attr_values option on serializer to take one of three values, "always" (the old True value), "legacy" (the new option, and the new default), and "spec" (the old False value, and the old default).	553
8	google-cloud-da	2.5.0	httplib2	0.9.2	<=0.9.2	25848	httplib2 before and including 0.9.2 on "SSL certificate hostname mismatch" it is checked only once: https://github.com/httplib2/httplib2/issues/5	782
9	rdflib	4.2.2	rdflib	4.2.2	==4.2.2	36882	The CLI tools in RDFLib 4.2.2 can load Python modules from the current working directory, allowing code injection, because "python -m" looks in this directory, as demonstrated by rdf2dot.	987
10	suds	0.4	suds	0.4	<=0.4	35433	cache.py in Suds 0.4, when tempdir is set to None, allows local users to redirect SOAP queries and possibly have other unspecified impact via a symlink attack on a cache file with a predictable name in /tmp/suds/.	1027
11	sparqlwrapper	version not fo	rdflib	4.2.2	==4.2.2	36882	The CLI tools in RDFLib 4.2.2 can load Python modules from the current working directory, allowing code injection, because "python -m" looks in this directory, as demonstrated by rdf2dot.	1191

Alex deVries

Findings

- Directly vulnerable:

  astropy, bleach, bottle, cryptography, django, docker, html5lib, httplib2, luigi, msgpack, numpy, pycrypto, pycryptodome, pyjwt, pyopenssl, pyowm, rdflib, requests, restkit, suds, urllib3

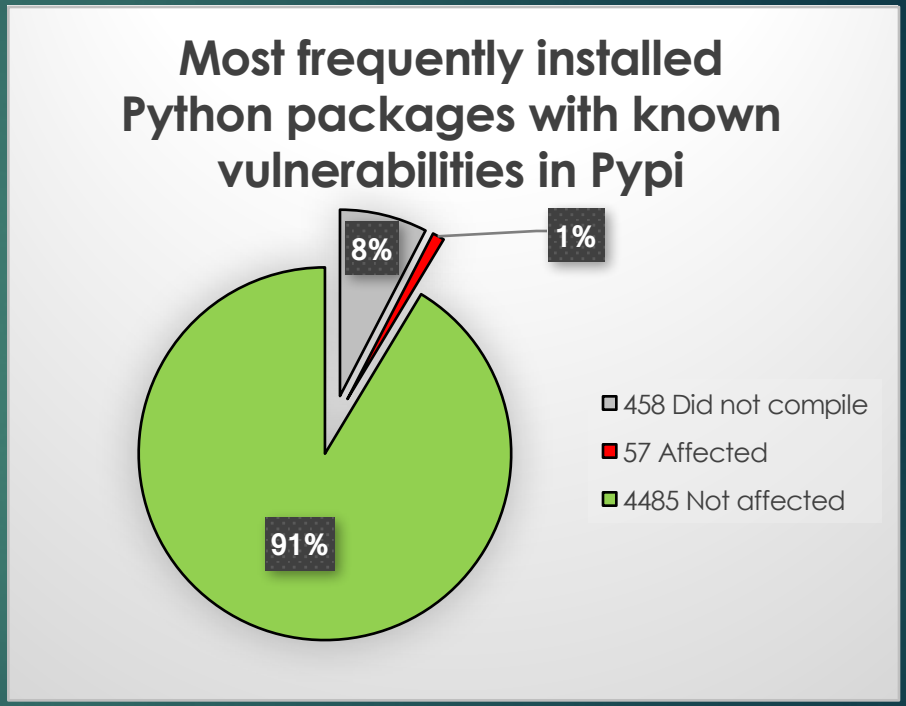
- Which then indirectly affect:

  spacy, tensorflow-tensorboard, google-cloud-dataflow, rdflib, suds, sparqlwrapper, google-endpoints-api-management, google-endpoints, mxnet, astropy, lftools, pyowm, python-jose-cryptodome, python-keyczar, warrant, salt, mxnet-mkl, rdflib-jsonld, flask-jwt, textacy, ssh, pantsbuild-pants, vcd-cli, restkit, luigi-td, prov, simple-crypt, wmic, couchdbkit, servicemanager, django-encrypted-fields, sdklib, pyrebase, sslyze, shudder, schema-salad, oraclebmc, cwltool, deployv, extruct, django-currentuser

Findings

- 57 of the top 5000 packages have at least one known vulnerabilities
- This is concerning

- Fine print:
  - Existence does not necessarily mean exploitable
  - Results vary between Python 2.7 and 3.6
  - As of mid-January 2019
  - Not all versions of a package are necessarily affected



The highlight: Pycrypto 2.6.1

Popular

- 240th most downloaded package on Pypi
- 1.6M downloads in February
- Required by 9 packages in the top 5000

Dangerous

- CVE-2013-2445 PRNG access
- CVE-2013-7459 Remote code execution
- CVE-2018-6594 Weak encryption

Abandoned

- Last updated October 2013
- Maintainer appears VSA (Darsey Litzenberger)

I can’t watch this train wreck and do nothing

My choices:

Fix and maintain Pycrypto

I am not a cryptographer.

Fix all packages that use Pycrypto

Infinite amount of work.

Have Pycrypto removed from Pypi

Seems easy…

Having Pycrypto removed

Q: How does one get a vulnerable package removed from Pypi?

A: *“It is up to the package maintainers to remove vulnerable packages or for users to do their best to not use vulnerable packages”*

- Possible next steps
  - Follow official mechanisms to inherit Pycrypto
  - Uprev Pypi entry with official notice that it shouldn’t be used
  - At some point, remove it (which would break 9+ packages)

The real problem

- Secure

- Easy to use

The upstream repo checklist

- Clear security governance statement
- Defined and practiced SIRT process
- Submission control
  - identity verified
  - Packages signed
- Packages manually reviewed before acceptance
- Integrity checking
  - Package signatures mandatory
  - Tools to verify integrity of installed packages
- Monitoring
  - Free or commercial

Alex deVries

Comparison of repos

Measure	Pypi	NPM	PHP Packagist	Docker hub (unpaid)	Red Hat / Microsoft / Apple
Security governance statement	✗	✓	✗		✓
Incident response process	✗	✓	✗	✗	✓
Submission control	✗	✓	✗	✗	Vendor selects content
Packages vetted	✗	✗	✗	✗	✓
Integrity checking	✗	✓	✗	✗	✓
Free monitoring	✓	✓	✓	✗	✓
Commercial monitoring	✓	✓	✓	✓	✓

Alex deVries

Recommendations

- Recognize the supply chain as an attack surface
- Evaluate upstream sources carefully
- Use a free or commercial scanning tool as part of CI/CD
- Respond to vulnerabilities

Questions?

alex@alexdev.ca

@alexdevsec

This QR code points to
github.com/alexdevsec/pypi-vuln

Alex deVries

OWASP
Open Web Application Security Project