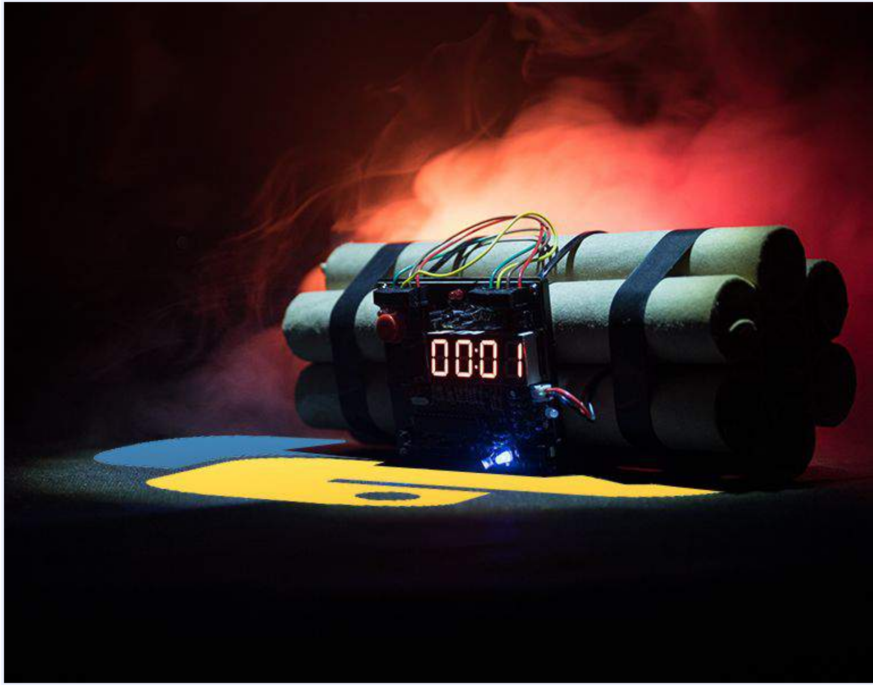
ActiveState

The Python 2 Time Bomb

Securing Python 2 Risk in Your Software Supply Chain



About ActiveState

ALTAIR GILEAD MSC Software Moody's Honeywell

DXC.technology STATE STREET Mentor A Siemens Business druva zenoss Own IT.

Used by Millions of Developers and 97% of Fortune 1000

20+ Years of Open Source Language Experience

Introductions

Jeff Rouse

Senior Product Strategist

Dana Crane

Product Marketing Manager

Housekeeping

- Ask questions in the Q&A tab
- There will be a poll midway through and a survey afterwards - your feedback is valuable
- Recording of this will be available and sent to you

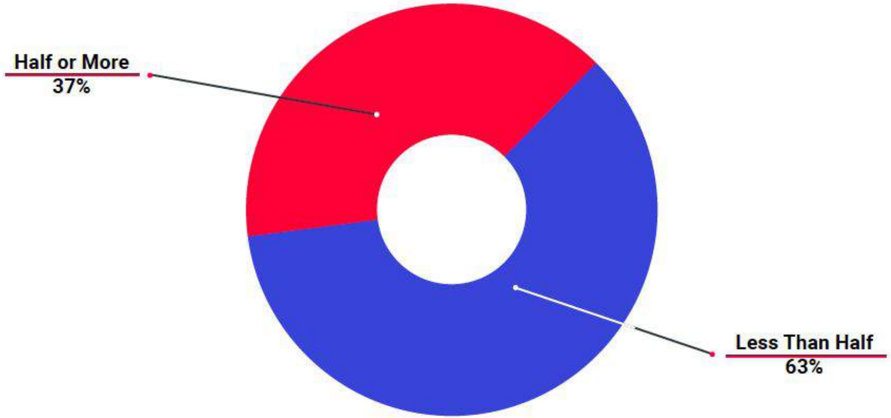
Agenda

- Python 2 EOL strategies
- Python 2 downloads over time
- Python 2 threats in the supply chain
- Demo

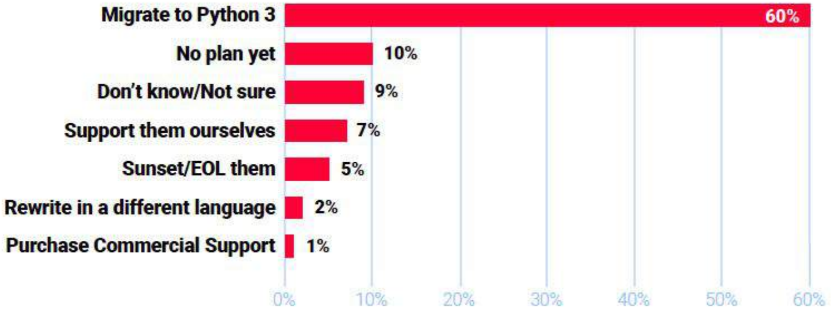
ActiveState

Python 2 EOL Strategies - Sept 2019

How many of your Python apps are Python 2?

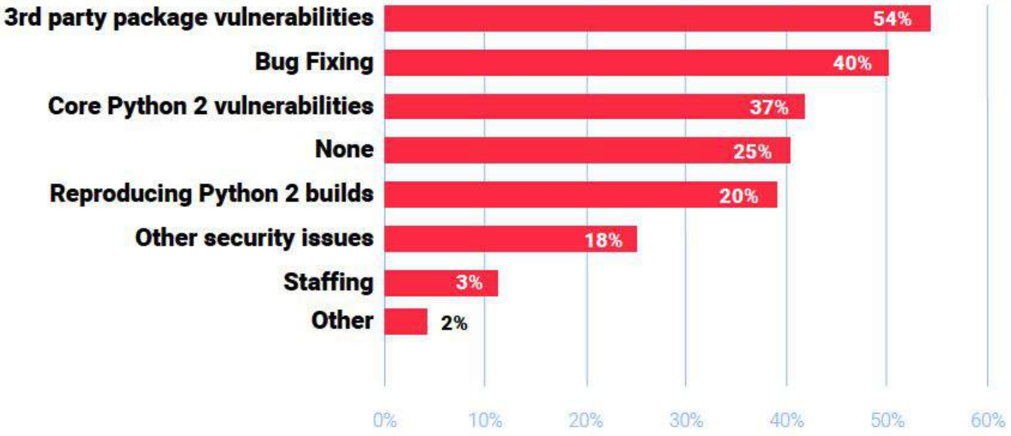


What will your organization do with your Python 2 apps?



Python 2 EOL Needs

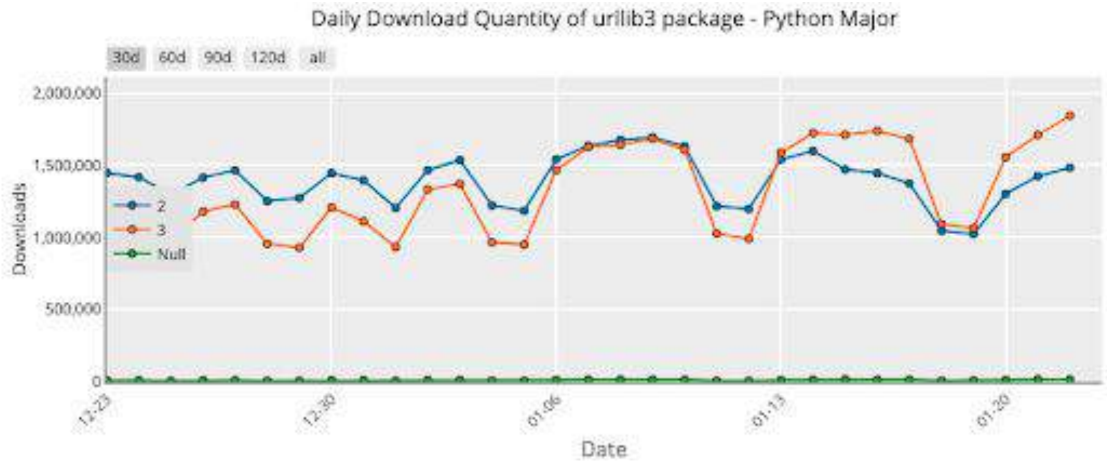
If supporting Python 2 yourself, what challenges do you expect? (check all that apply)



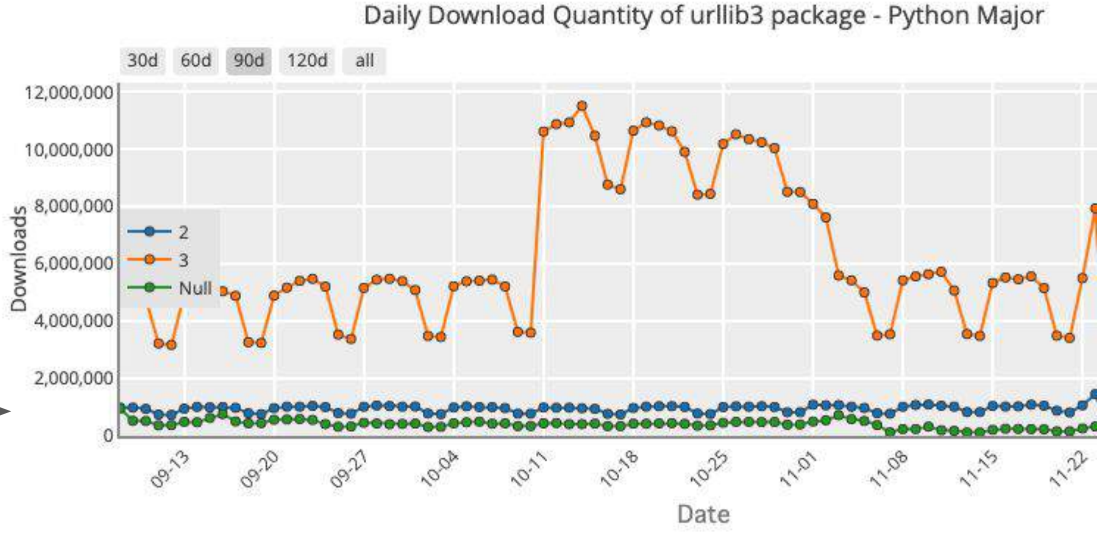
ActiveState

PyPI Stats

1.5M
downloads


Daily Download Quantity of urllib3 package - Python Major

Dec 2019
to
Jan 2020


Daily Download Quantity of urllib3 package - Python Major

Sep 2021
to
Nov 2021

The Growing Software Supply Chain Threat

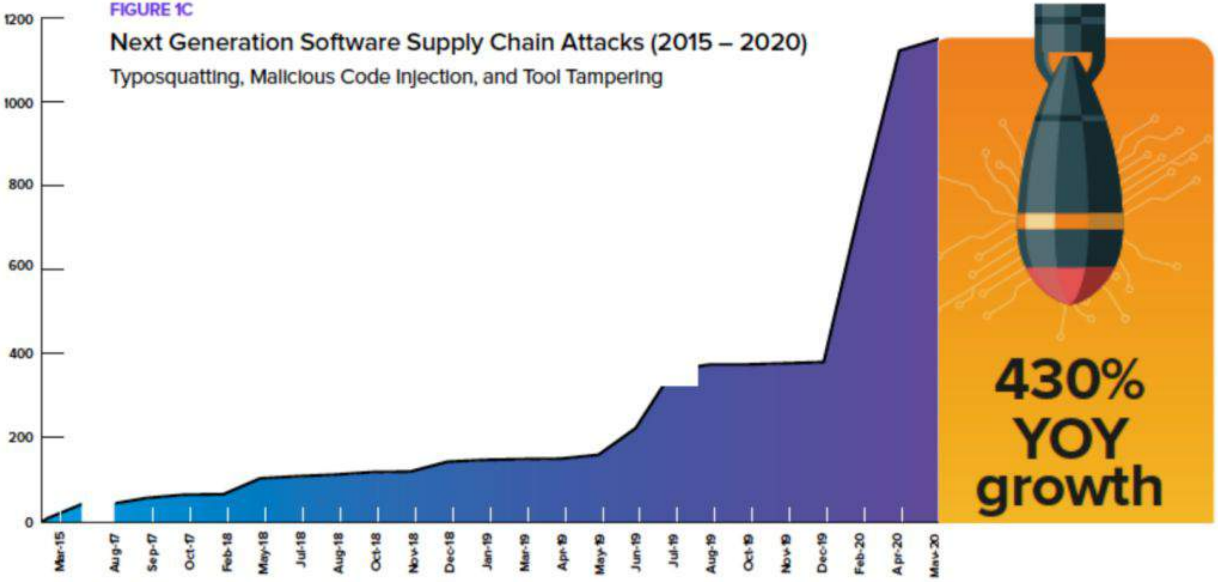


FIGURE 1C

Next Generation Software Supply Chain Attacks (2015 – 2020)

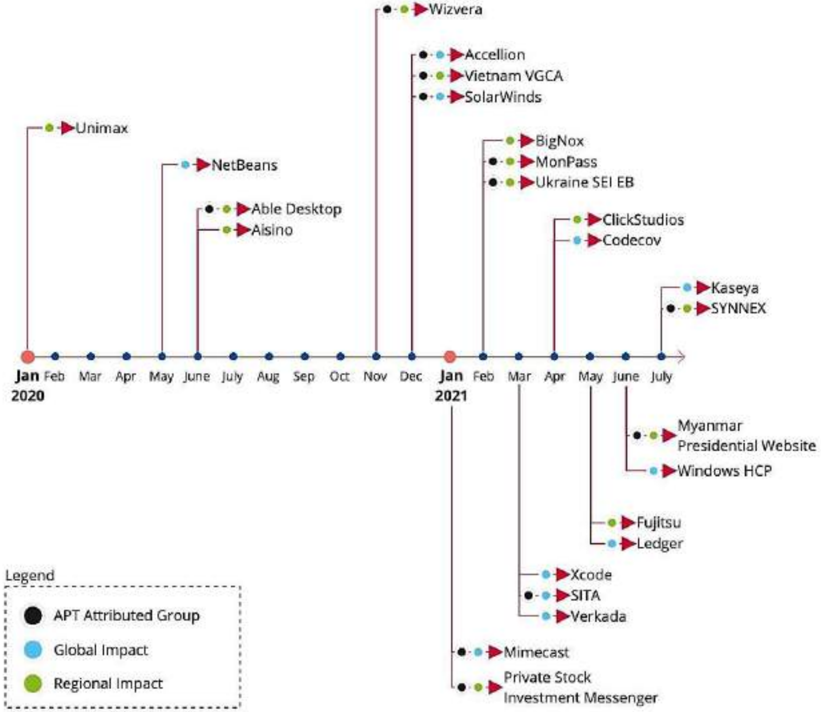
Typosquatting, Malicious Code Injection, and Tool Tampering

430% YOY growth

Source: Sonatype State of the Software Supply Chain 2020

ActiveState

Supply Chain/Dev Environment Attacks

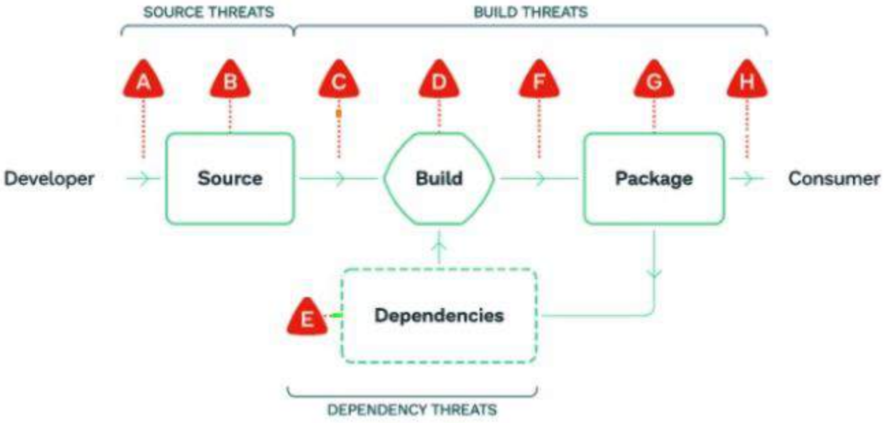


Securing the Software Supply Chain

Requires both:

- Software security
  - ie., vulnerability management
- Software development process integrity
  - ie., how you import, build and run/use software components

The Growing Integrity Threat



SOURCE THREATS

A Bypassed code review

B Compromised source control system

BUILD THREATS

C Modified code after source control

D Compromised build platform

F Bypassed CI/CD

G Compromised package repo

H Using a bad package

DEPENDENCY THREATS

E Using a bad dependency

Example: Insecure Build Service

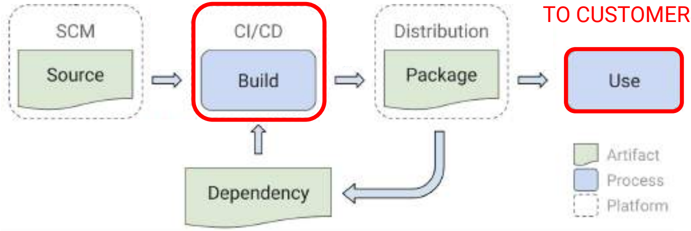
solarwinds

Business Impact

- Millions in direct losses
- Billions in cleanup costs
- SWI stock dropped 40% in a day

CI/CD PIPELINE COMPROMISED

SIGNED DLL W/ TROJAN SENT TO CUSTOMERS



SCM: Source → CI/CD: Build → Distribution: Package → Use

Dependency → Build

🟩 Artifact
🟦 Process
⬚ Platform

18,000 customer affected, including:

- 80% of the Fortune 500
- The top 10 US telecom companies
- The top 5 US accounting firms
- The CISA, FBI & NSA
- All 5 branches of the US military

The Growing Vulnerability Threat

Core CVEs	SEVERITY	STATUS	PUBLISH DATE
CVE-2021-23336	High	Fix available	2021/02/15
CVE-2021-3177	Critical	Fix available	2021/01/19
CVE-2020-27619	Critical	Fix available	2020/10/21
CVE-2020-26116	High	Fix available	2020/09/27
CVE-2019-20907	High	Fix available	2020/07/13
CVE-2020-8492	Medium	Fix available	2020/01/30

3rd Party CVEs	SEVERITY	STATUS	PUBLISH DATE
CVE-2021-43818 - lxml	High	Fix pending	2021/12/13
CVE-2021-3711 - OpenSSL	Critical	Fix available	2021/08/24
CVE-2021-25289 - Pillow	Critical	Fix available	2021/03/19
CVE-2021-3712 - OpenSSL	High	Fix available	2021/08/24
CVE-2021-33203 - Django	High	Fix available	2021/06/08
CVE-2020-36242 - Django	Critical	Fix available	2021/02/07

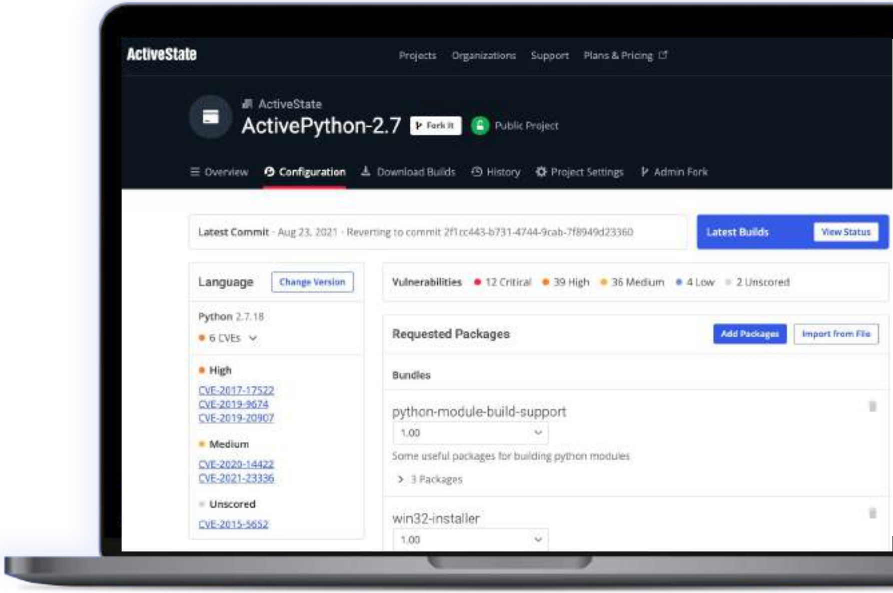
AS Platform: Securing the Python 2 Supply Chain

Automatically build, update and maintain Open Source runtime environments:

- Per software project
- Per use case
- Per customer

In order to:

- Ensure security and integrity of open source components
- Reduce the Mean Time to Remediation (MTTR) of vulnerabilities
- Secure your import, build and run processes



ActivePython-2.7

Latest Commit · Aug 23, 2021 · Reverting to commit 2f1cc443-b731-4744-9cab-7f8949d23360

Language

Python 2.7.18

● 6 CVEs

● High

CVE-2017-17522
CVE-2019-9674
CVE-2019-20907

● Medium

CVE-2020-14422
CVE-2021-23336

● Unscored

CVE-2015-5652

Vulnerabilities ● 12 Critical ● 39 High ● 36 Medium ● 4 Low ● 2 Unscored

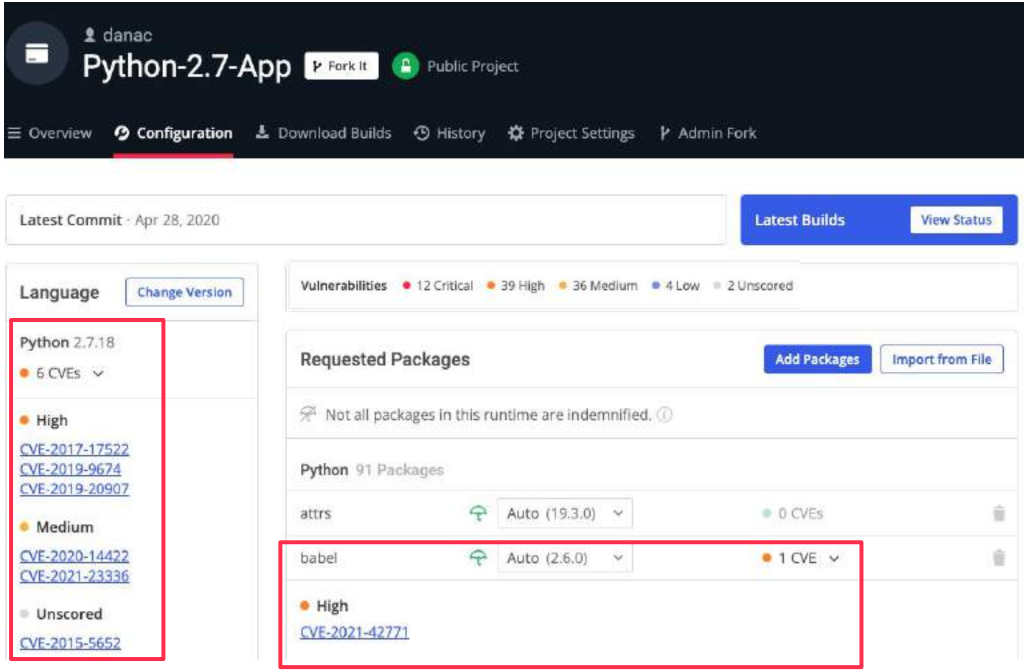
Requested Packages

Bundles

python-module-build-support
1.00
Some useful packages for building python modules
› 3 Packages

win32-installer
1.00

Identifying Vulnerabilities



Latest Commit · Apr 28, 2020

Language

Python 2.7.18

● 6 CVEs

● High

CVE-2017-17522
CVE-2019-9674
CVE-2019-20907

● Medium

CVE-2020-14422
CVE-2021-23336

● Unscored

CVE-2015-5652

Vulnerabilities ● 12 Critical ● 39 High ● 36 Medium ● 4 Low ● 2 Unscored

Requested Packages

Not all packages in this runtime are indemnified.

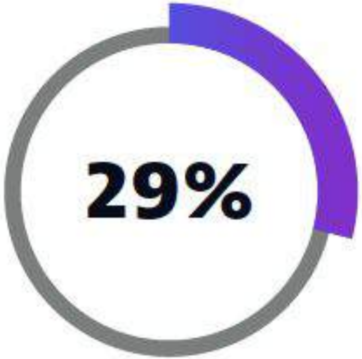
Python 91 Packages

attrs	Auto (19.3.0)	● 0 CVEs
babel	Auto (2.6.0)	● 1 CVE

● High

CVE-2021-42771

Cost of Vulnerabilities



is the average proportion of time application security teams in large enterprises (1,000+ employees) spend each week doing vulnerability management tasks that could be automated*

$1.51 million

is the average annual labor cost organizations incur for the time their application security teams spend on manual vulnerability management tasks that could be automated*

Source: Dynatrace Global CISO Report

Poll: How many tools/scripts/services/apps are you still running Python 2 in prod and/or non-prod?

- 0
- 1-2
- <10
- >10

Python 2 End of Life

- Python 2 core language - community maintenance ended January 1st, 2020
  - No updates whatsoever, not even for critical security updates
- What about the third-party Python 2 packages you rely on
  - Support for the third-party Python 2 packages, libraries and modules have continually dropped support since this date, most major projects no longer provide support

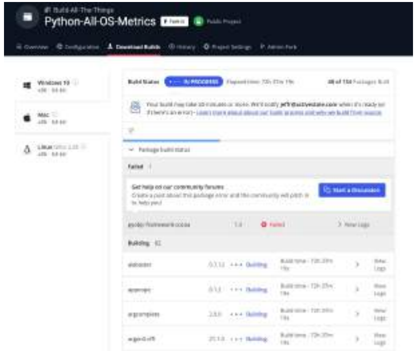
Yet, there are a huge number of python 2 applications in use and will be for the foreseeable future!

Python 2 End of Life - Hidden Dangers

- Python 2 core language relies on shared libraries, which also are subject to vulnerailities
  - C/C++ libraries that support functionality in the Python core language
    - Examples: bzip, openSSL, expat
  - Many of these libraries are “vendored in”, they aren’t easily upgradeable
    - We also update these libraries as well, as we build everything from source
  - Third-party packages you use may also have dependencies on shared libraries, these also need to be continually updated

Our Python 2 Extended Support

- Python 2 core language
  - Support for the features and functionality of the core Python 2 language and standard libraries.
- The third-party Python 2 packages you use
  - Support for the third-party Python 2 packages, libraries and modules included in your applications.
- Backported core language security fixes
  - Fixes in Python 3 core language code will be backported to Python 2 and made available as a patch.
- Backported third-party package security fixes from Python 3 to Python 2
  - Fixes implemented in Python 3 third-party packages will be backported to Python 2 and made available



Addressing Python 2 Vulnerabilities

- ActiveState forked and continues to maintain Python 2.7:
  - On the ActiveState Platform for customers
  - All fixes released back to the community
  - Currently on Version 2.7.18.4
- Address security fixes for vulnerabilities (CVEs) to the Python core, prioritized by severity
- Address 3rd-party vulnerabilities (CVEs) for the packages they use in their applications

CVE - Common Vulnerabilities and Exposures

Python 2 Supply Chain Security

Platform Demo

ActiveState

Demo: Python 2 Vulnerabilities

- See the differences between existing Python 2.7.18 final builds, and ActiveState’s updates
- See vulnerability reporting
- Building a python 3 project if you plan to work on migration from Python 2

Q&A and Next Steps

Learn more about Python 2 extended support
https://www.activestate.com/products/python/python-2-7/

See Python 2 CVE updates
https://www.activestate.com/products/python/python-2-end-of-life-security-updates/

Try the ActiveState Platform
https://platform.activestate.com/

ActiveState

Webinar Feedback

Take our quick survey!
https://www.surveymonkey.com/r/python-2

ActiveState

Demo: Python 2 Vulnerability Resolution

1. Import a Python 2 requirements.txt to the ActiveState Platform
2. Identify vulnerabilities
3. Resolve vulnerabilities
4. Deploy the secure runtime environment