title.py

README.md

```
1
2
3
4   match pycon_talk:
5       case ”Python Security”:
6           print(
7               “Because “
8               “\“Nobody Would Hack my Project\” ”
9               “is not a Security Strategy“
10          )
11
12
13  # Benjamin Müllner + Pamphile T. Roy
14
```

Pamphile Roy {

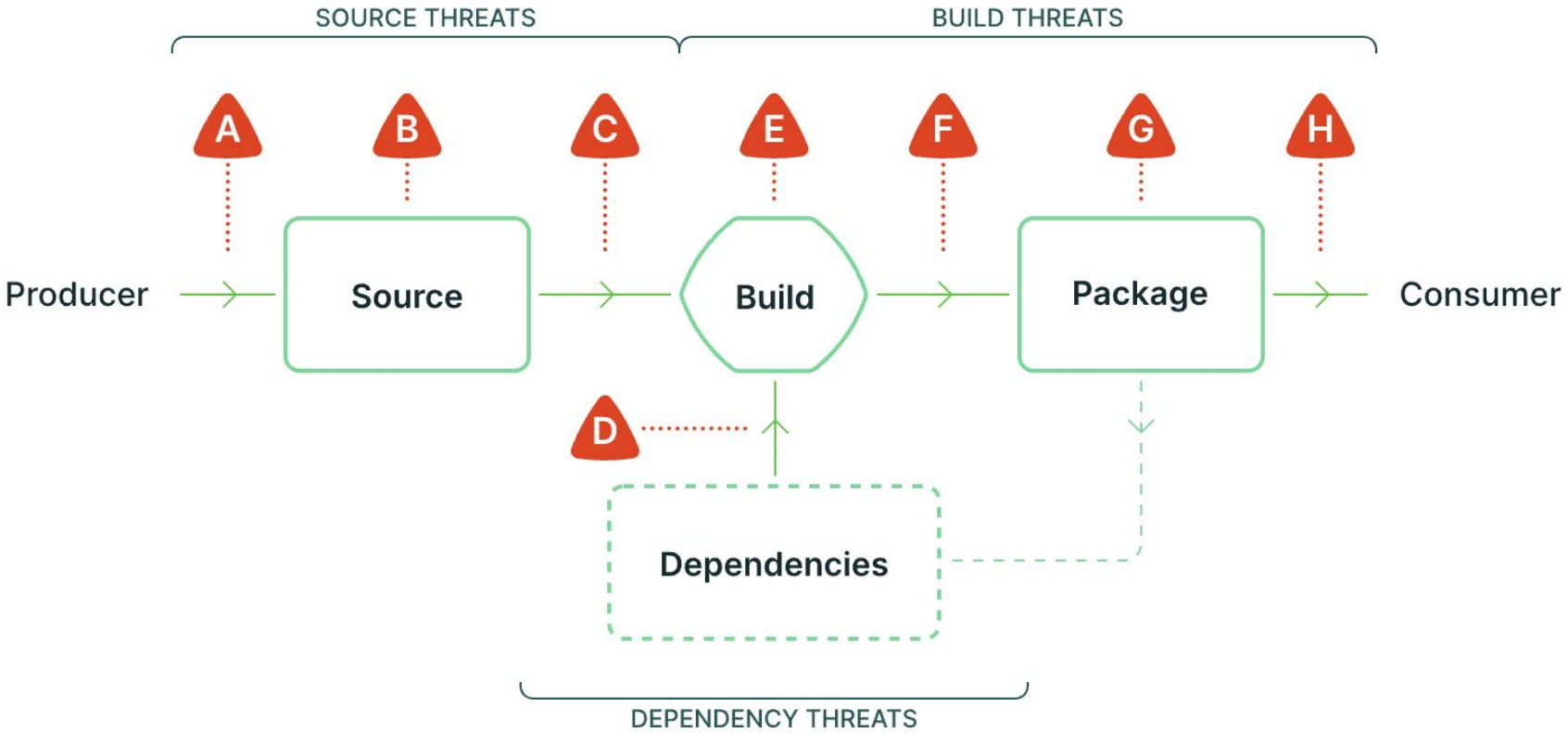
- Senior Engineer @ Bitpanda
- SciPy and SALib maintainer
- Scientific Python SPEC Steering Committee
- Stellar Community Fund Pilot

}

Benjamin Müllner {

- Staff IT Security Engineer @ Bitpanda
- (Former) Golang Engineer
- Building Dev(Sec)Ops Tooling

}


SOURCE THREATS
BUILD THREATS
A
B
C
E
F
G
H
Producer
Source
Build
Package
Consumer
D
Dependencies
DEPENDENCY THREATS

Solarwinds ‘the Hack’ {

- 09.2019 “Nobelium Group” get access
- 10.2019 Code Injection Test into Orion
- 02.2020 “Sunburst” Injected into Orion
- 03.2020 Solarwinds Ships out Malicious Software

}

```
PyTorch ‘dependency injection’ {
    12.2022  torchtriton uploaded to PyPi

        5 days   SSH, /etc, configs…

            Happy New Year   PyPi notified and PyTorch team
                             removed the internal dependency
}
```

name-confusion.py

Name Confusion Attacks

‘OSS-RISK-3’ {

- Synonym — Typosquatting
- Example — Colo(u)rama (PyPI, 2018)
- AI Helpers — Make it Easier (for the Attacker)

}

workflows/my-action.yml

dependabot.yml

tj-actions/changed-files {

- **Attack Vector** — Compromised PAT of Renovate Bot (@tj-actions-bot)
- **Issue** — Overprovisioned access for Bot
- **Attack** — Memory Dump via Python Script
- **Avoid** — Check Actions and Dependabot Config

}

Verify Your Dependencies

PyCon Austria

Python Packaging Authority ‘PyPA’ {

- Transparency — Disclosure and impact assessment
- Analysis — Security incident analysis
- Features — 2FA, OIDC, attestations, quarantine, …
- Collaboration — PEP, security support, SPEC

}

release-1.0-py3-none-any.whl

action.yaml

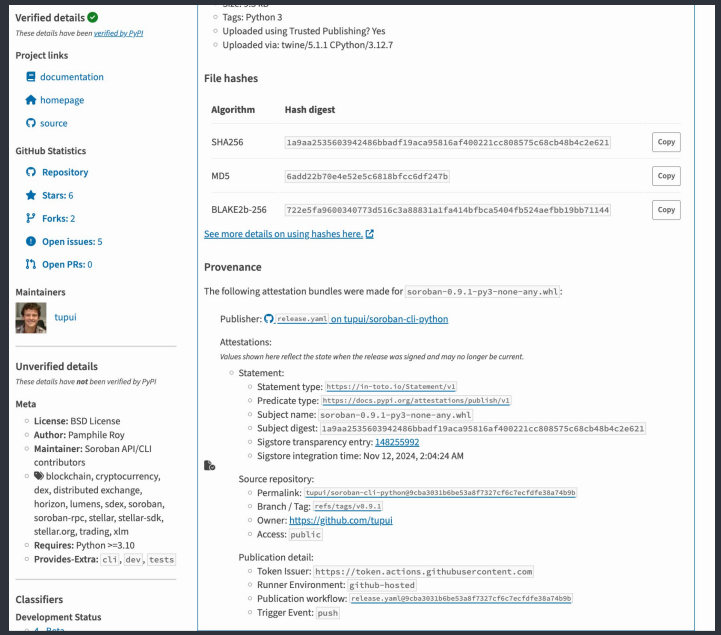
Trusted Publisher ‘attestations’ {

Works on my machine, ship it!

Making a release:

* CI
* Build sdist and wheels
* Sign
* Push it! The CD part

}



```
PyPI ‘organization’ {

    # We can now create organizations! But why?


        >>>     * Branding
                * Logical grouping for your community
                * Permissions management


        >>>     * Support PyPI
                * Organization plans with maybe extras ;)


}
```

pylock.toml

cheese_shop.py

PEP 751 ‘The Standard’ {

- File Name — `r"^pylock\.([^.]+)\.toml$"`
- Global — Not depending on pip, uv, poetry, etc.
- Standard — Makes life easier for Security Tools.

}

Scientific Python ‘SPEC’ {

SLSA

Validation of software artifacts

OIDC

Use mechanism such as TrustedPublisher from PyPi

Permissions

Hardening workflow environment; especially for releases

Dependencies

Pin dependencies with full hashes

https://scientific-python.org/specs/spec-0008

https://scientific-python.org/specs/spec-0006

}

Open Source ‘Maintainer’ {

Your pipeline has access to secrets, artifacts, and deploy targets.

>>> Know your Maintainers

>>> Treat workflow files like application code – review, audit, and lock them down.

>>> Attest your Builds

}

Open Source ‘User’ {

- **Trust** — Use Trusted Sources (Github, PyPI, …)
- **Size** — Prefer Well Maintained Projects
- **Review** — Spend some time on checking what you are using and installing
- **Stay Alert** — Lock and continuously monitor dependencies

}

Thanks ‘Now go check your dependencies’ {

https://github.com/tupui

https://github.com/prempador

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

}