DEVCONF.cz

Securing Python projects Supply Chain

DevConf.CZ 2023

Maya Costantini
Software Engineer, Red Hat

Fridolín Pokorný
Entrepreneur

Who we are

Maya Costantini

Software Engineer,
Red Hat Emerging Technologies
Security team

@MayaCostantini

hachyderm.io/@mayacostantini

@mayaCostantini

Fridolín Pokorný

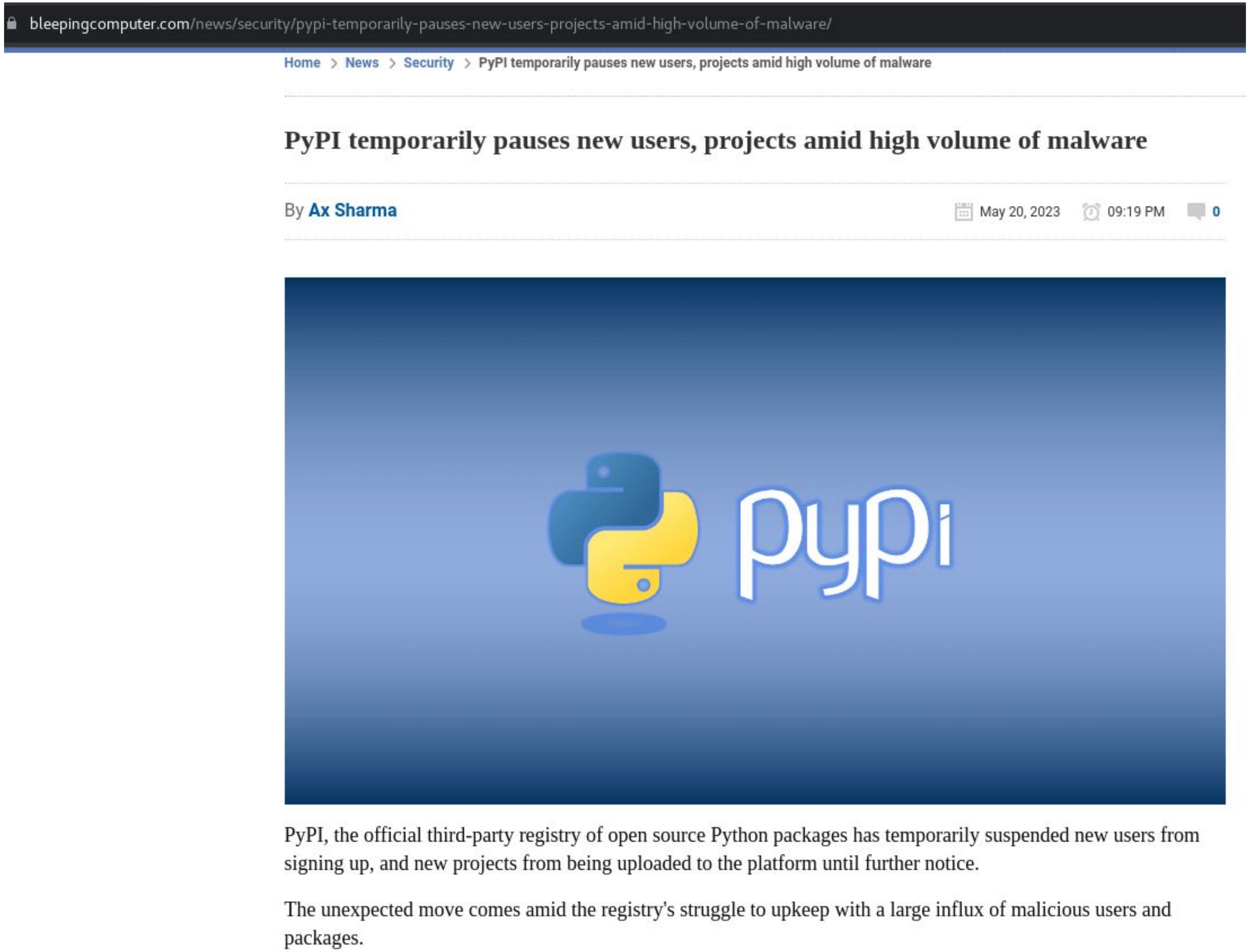
Entrepreneur
ex-DataDog
ex-Red Hatter

@fridex

fosstodon.org/@fridex

@fridex

Why protecting your supply chain matters



bleepingcomputer.com/news/security/pypi-temporarily-pauses-new-users-projects-amid-high-volume-of-malware/

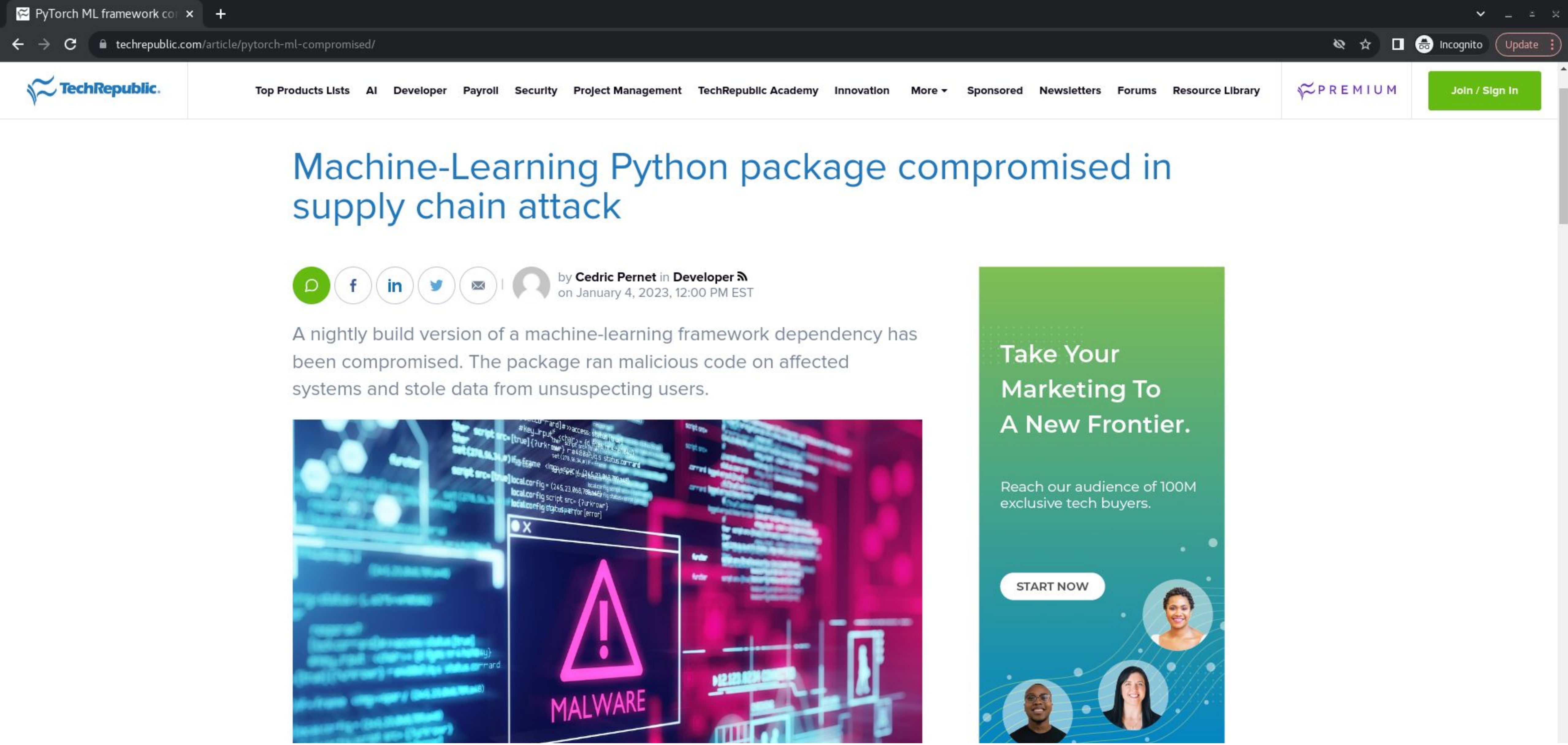
Home > News > Security > PyPI temporarily pauses new users, projects amid high volume of malware

PyPI temporarily pauses new users, projects amid high volume of malware

By Ax Sharma May 20, 2023 09:19 PM 0

PyPI, the official third-party registry of open source Python packages has temporarily suspended new users from signing up, and new projects from being uploaded to the platform until further notice.

The unexpected move comes amid the registry's struggle to upkeep with a large influx of malicious users and packages.



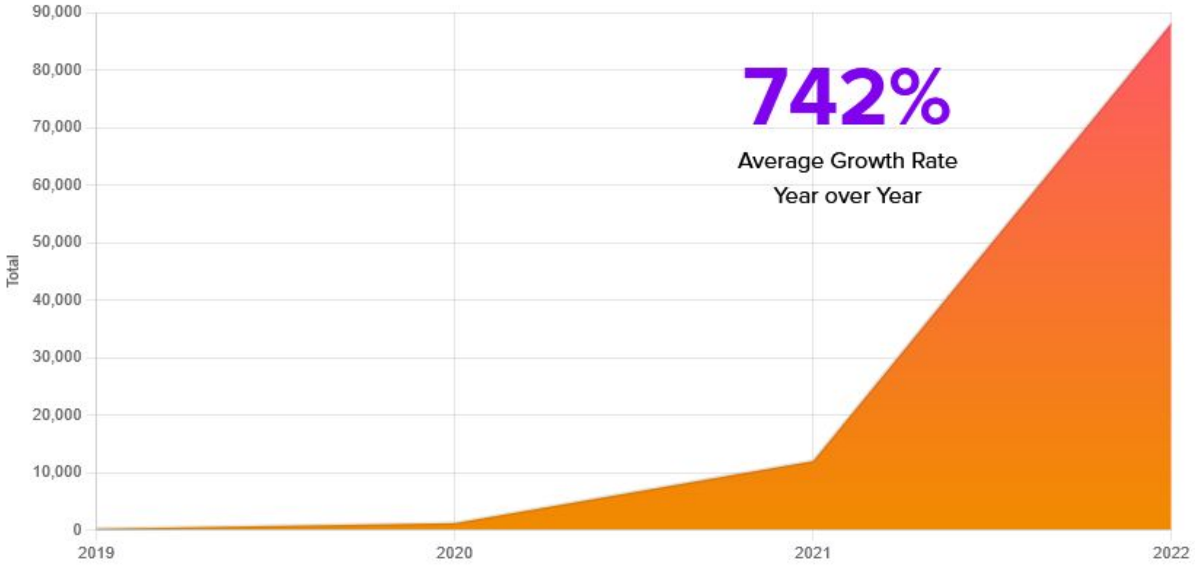
Machine-Learning Python package compromised in supply chain attack

by Cedric Pernet in Developer
on January 4, 2023, 12:00 PM EST

A nightly build version of a machine-learning framework dependency has been compromised. The package ran malicious code on affected systems and stole data from unsuspecting users.

The real cost of a vulnerable supply chain

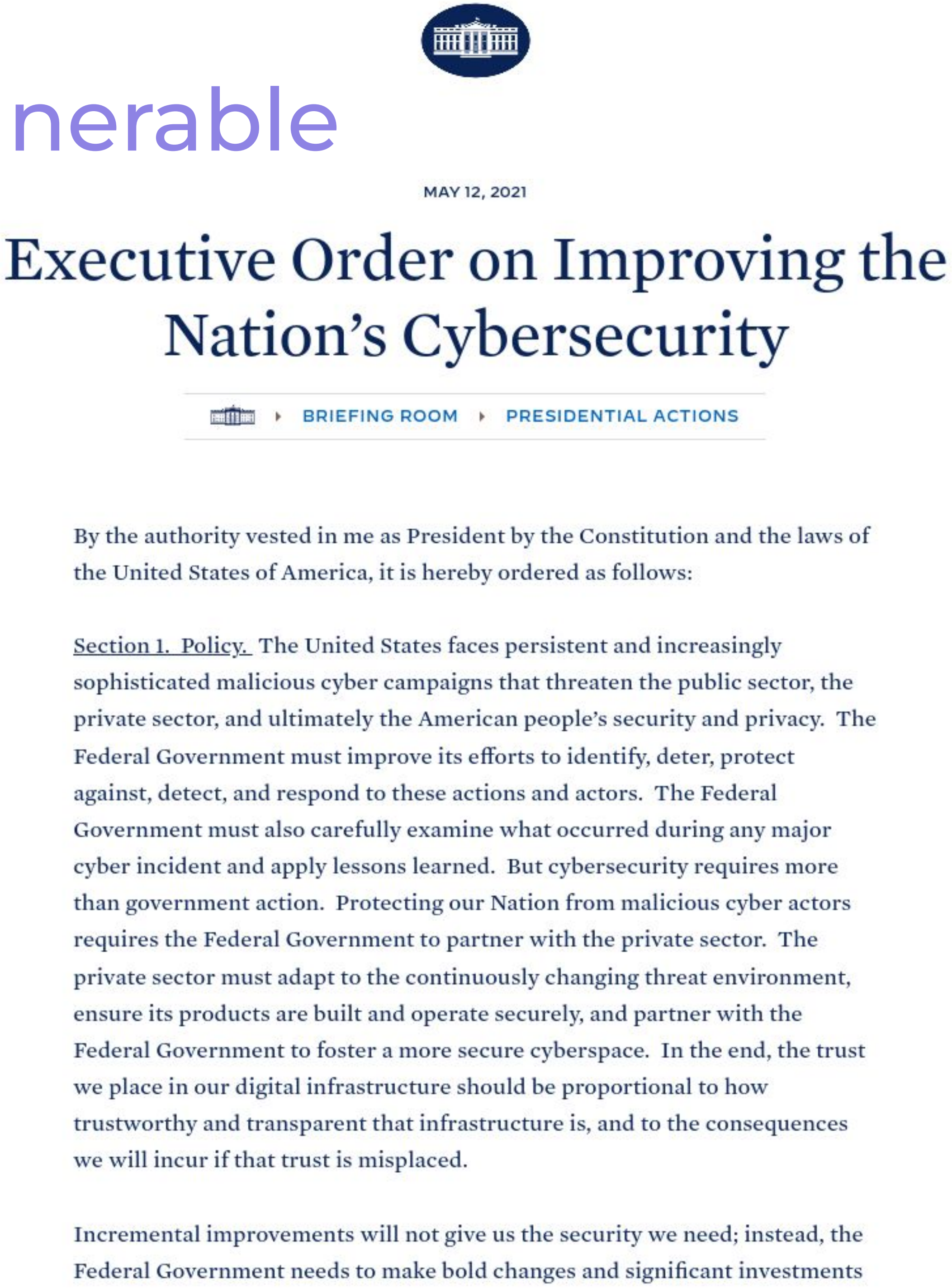
- The average annual increase in Software Supply Chain attacks over the past 3 years is of **742%**
- Financial and reputational damage
- Legal and compliance issues



Source: securityboulevard.com

The real cost of a vulnerable supply chain

- **Executive Order 14028** On Improving the Nation's Cybersecurity
- Issued in reaction to the **2020 United States federal government data breach** via exploits in enterprise software (SolarWinds)



MAY 12, 2021

Executive Order on Improving the Nation's Cybersecurity

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

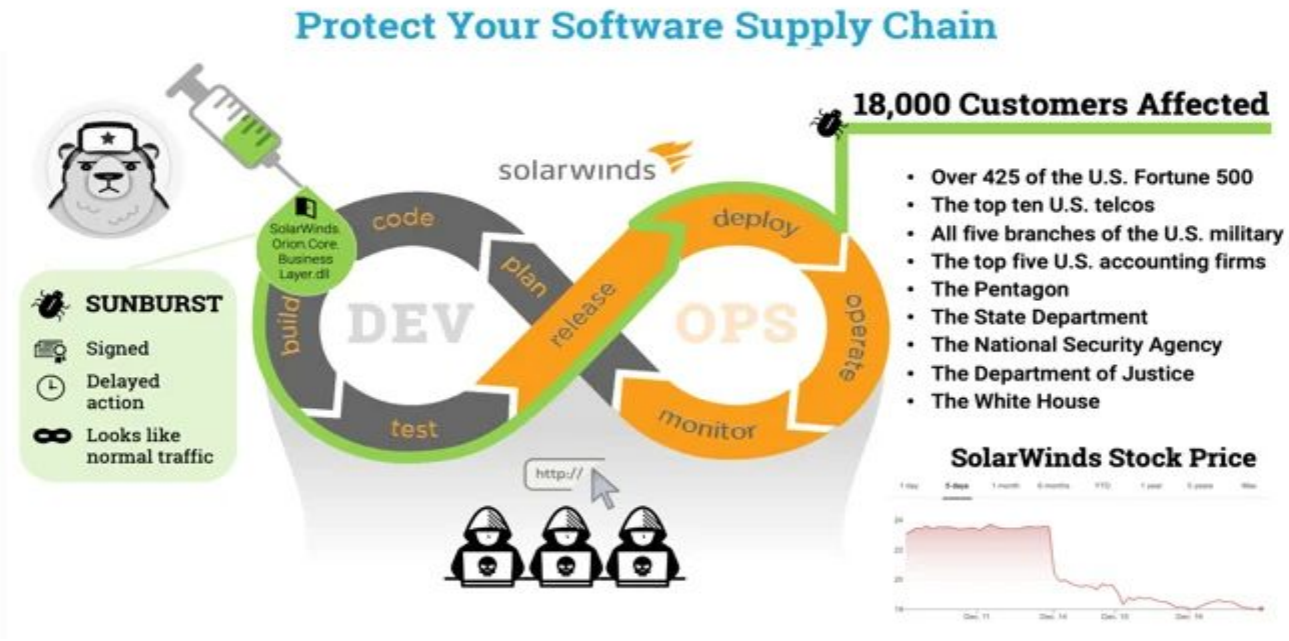
Section 1. Policy. The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people's security and privacy. The Federal Government must improve its efforts to identify, deter, protect against, detect, and respond to these actions and actors. The Federal Government must also carefully examine what occurred during any major cyber incident and apply lessons learned. But cybersecurity requires more than government action. Protecting our Nation from malicious cyber actors requires the Federal Government to partner with the private sector. The private sector must adapt to the continuously changing threat environment, ensure its products are built and operate securely, and partner with the Federal Government to foster a more secure cyberspace. In the end, the trust we place in our digital infrastructure should be proportional to how trustworthy and transparent that infrastructure is, and to the consequences we will incur if that trust is misplaced.

Incremental improvements will not give us the security we need; instead, the Federal Government needs to make bold changes and significant investments

Supply Chain Threats and Vulnerabilities

SolarWinds attack

- SolarWinds Orion
  - Network performance monitoring platform
- Injected malicious code in software updates
- Signed updates were downloaded by customers



Protect Your Software Supply Chain

18,000 Customers Affected

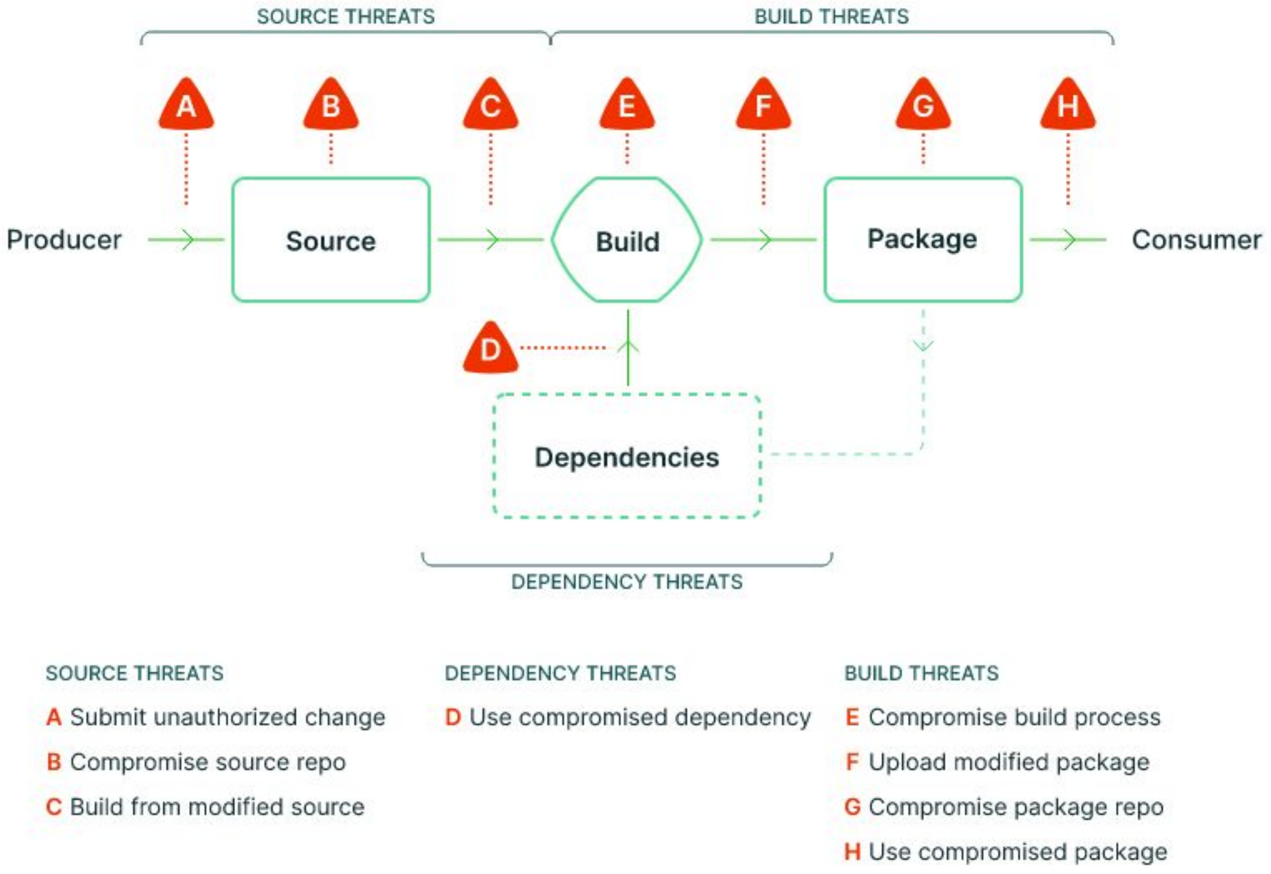
- Over 425 of the U.S. Fortune 500
- The top ten U.S. telcos
- All five branches of the U.S. military
- The top five U.S. accounting firms
- The Pentagon
- The State Department
- The National Security Agency
- The Department of Justice
- The White House

SolarWinds Stock Price

Source: Three Things the SolarWinds Supply Chain Attack Can Teach Us

SLSA

- **Supply-chain Levels for Software Artifacts**
- Levels of assurance
  - **L0:** No requirements
  - **L1:** Provenance showing how the package was built
  - **L2:** Signed provenance by a hosted build platform
  - **L3:** Hardened build platform



SOURCE THREATS

A Submit unauthorized change

B Compromise source repo

C Build from modified source

DEPENDENCY THREATS

D Use compromised dependency

BUILD THREATS

E Compromise build process

F Upload modified package

G Compromise package repo

H Use compromised package

Building a security toolbox to protect your Python project

Secure supply chain frameworks

TUF (The Update Framework)

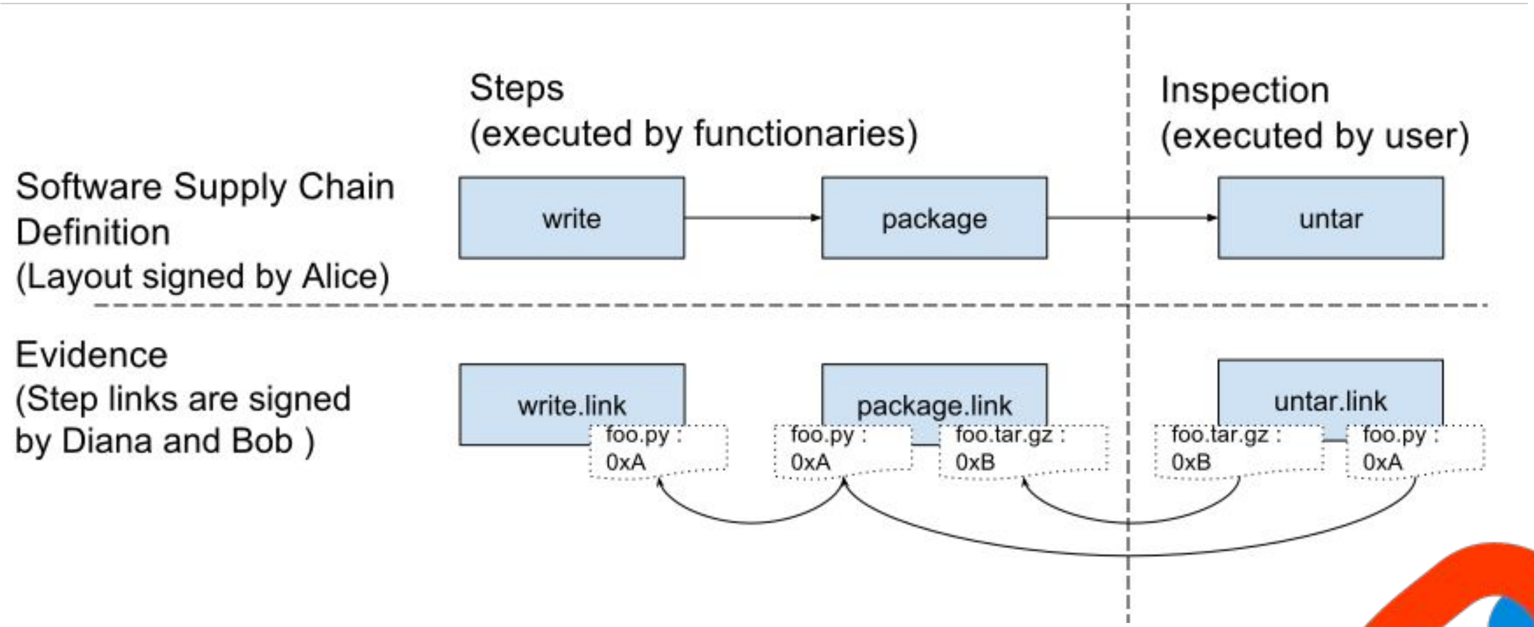
TUF

- Secure software updates and prevent tampering attacks, rollback attacks, and key compromise attacks
- The reference implementation is based on Thandy, updater for Tor
- Uptane: a secure update system for car software
- Client implementation available in Python: python-tuf
- Secure Publication of Datadog Agent Integrations with TUF and in-toto
  - PEP-458, PEP-480
- Used to securely download public keys for instances of Sigstore

in-toto

in-toto: Secure the integrity of software supply chains

- Attesting each step of a supply chain, from initiation to end-user installation



Steps
(executed by functionaries)

Inspection
(executed by user)

Software Supply Chain
Definition
(Layout signed by Alice)

write

package

untar

Evidence
(Step links are signed
by Diana and Bob)

write.link

package.link

untar.link

foo.py : 0xA

foo.py : 0xA

foo.tar.gz : 0xB

foo.tar.gz : 0xB

foo.py : 0xA

Software signing

Digitally sign artifacts with **project Sigstore**

- A secure and simple interface, no need for specific cryptography knowledge
- Sign using an **OpenID Connect identity** instead of a private key
- Available as a Python client: sigstore-python
- Easily scalable and adapted to automated supply chain workflows (CI/CD, build, releases…)

```
$ sigstore sign mypackage.whl

$ sigstore verify identity \
  --cert-identity package@maintainer.com \
  --cert-oidc-issuer https://github.com/login/oauth \
  mypackage.whl.sigstore
```

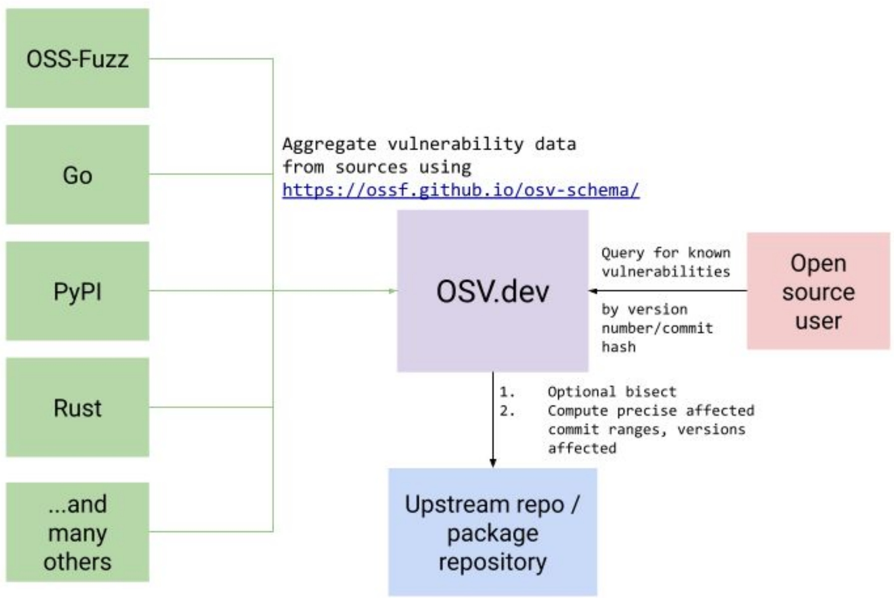
Malicious or Vulnerable

- Vulnerability in software
  - A flaw in a computer system that weakens the overall security of the system (Wikipedia)
  - Vulnerabilities can be exploited but not all vulnerabilities are exploitable
- Malicious software (Malware)
  - Any software intentionally designed to cause disruption (Wikipedia)

Vulnerability databases

OSV: A distributed vulnerability database for Open Source

- Aggregates vulnerability databases that use the [OpenSSF Vulnerability format]

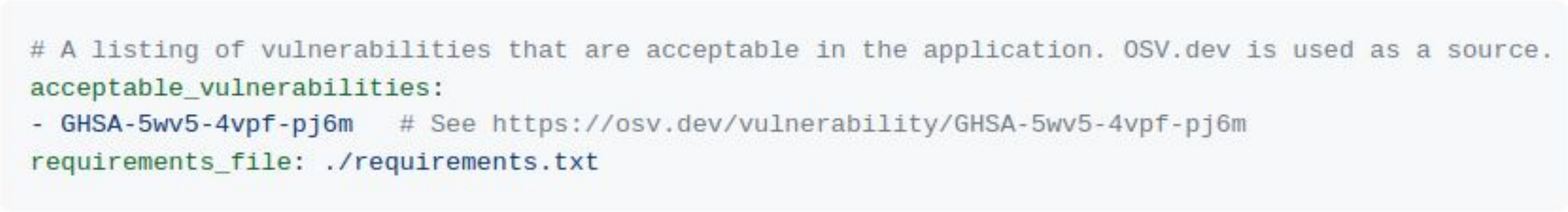

OSS-Fuzz
Go
PyPI
Rust
...and many others
Aggregate vulnerability data from sources using https://ossf.github.io/osv-schema/
OSV.dev
Query for known vulnerabilities
by version number/commit hash
Open source user
1. Optional bisect
2. Compute precise affected commit ranges, versions affected
Upstream repo / package repository

GUAC: Graph for Understanding Artifact Composition

- Aggregates software security metadata into a comprehensive **graph database**: artifacts, attestations, identities, relationships
- Prevent supply chain compromises and react to vulnerabilities by understanding how artifacts fit together

Vulnerabilities and PyPI

- No direct support in pip
  - pip-audit
  - an experiment: [pipctl]() - “*pip cuddle*”. 🤗🐍



```
# A listing of vulnerabilities that are acceptable in the application. OSV.dev is used as a source.
acceptable_vulnerabilities:
- GHSA-5wv5-4vpf-pj6m   # See https://osv.dev/vulnerability/GHSA-5wv5-4vpf-pj6m
requirements_file: ./requirements.txt
```

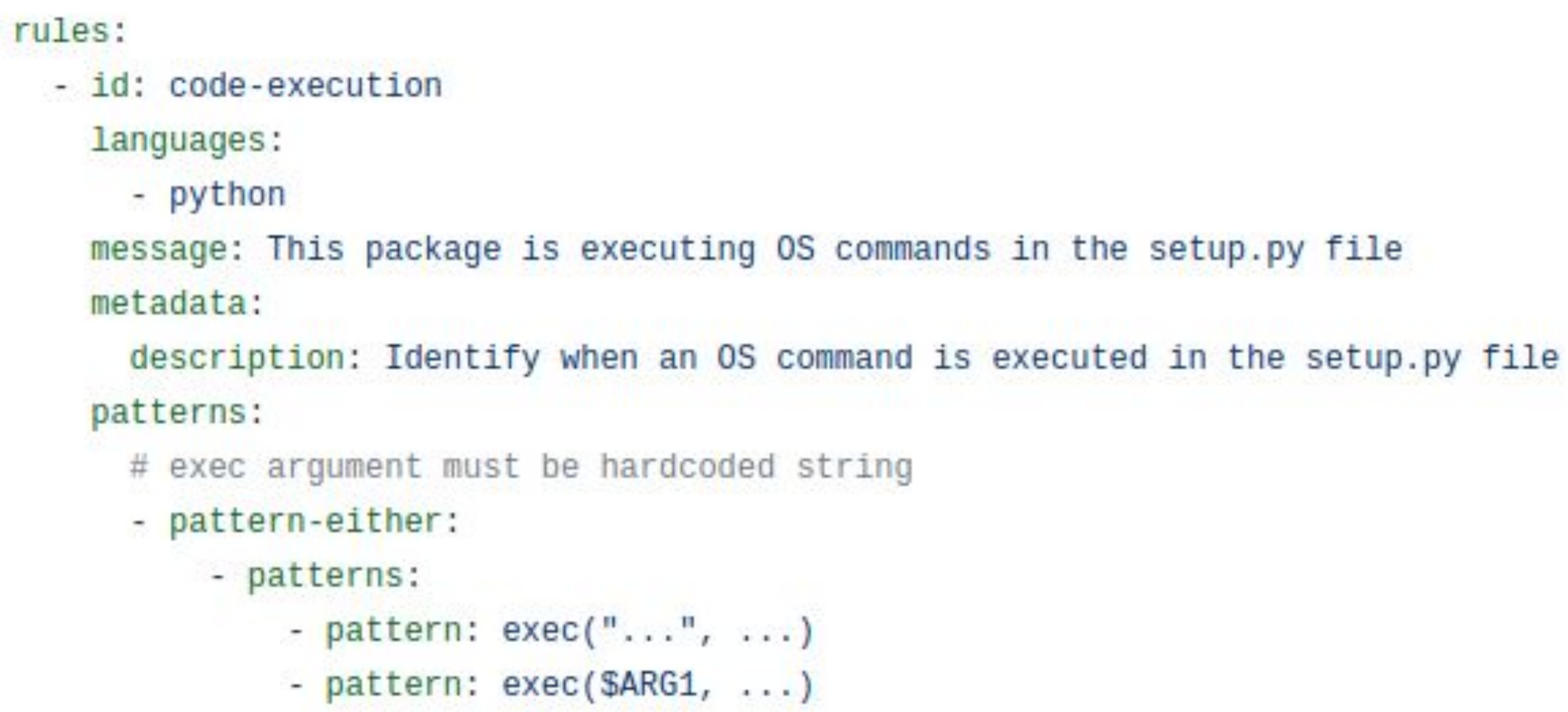
- security-constraints
  - [mam-dev/security-constraints]()

PyPI and malicious packages

- ±40 malware packages taken down each day
    - Manual work
    - DataDog/malicious-software-packages-dataset

- Finding malicious PyPI packages through static code analysis: Meet GuardDog
    - semgrep rules are used for static source-code analysis
    - not used on PyPI directly
    - DataDog/guarddog

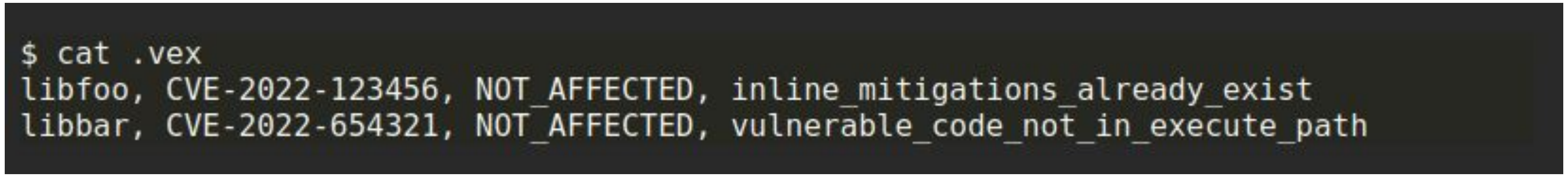
PyPI and malicious packages



```
rules:
  - id: code-execution
    languages:
      - python
    message: This package is executing OS commands in the setup.py file
    metadata:
      description: Identify when an OS command is executed in the setup.py file
    patterns:
      # exec argument must be hardcoded string
      - pattern-either:
          - patterns:
              - pattern: exec("...", ...)
              - pattern: exec($ARG1, ...)
```

SBOMs and VEX

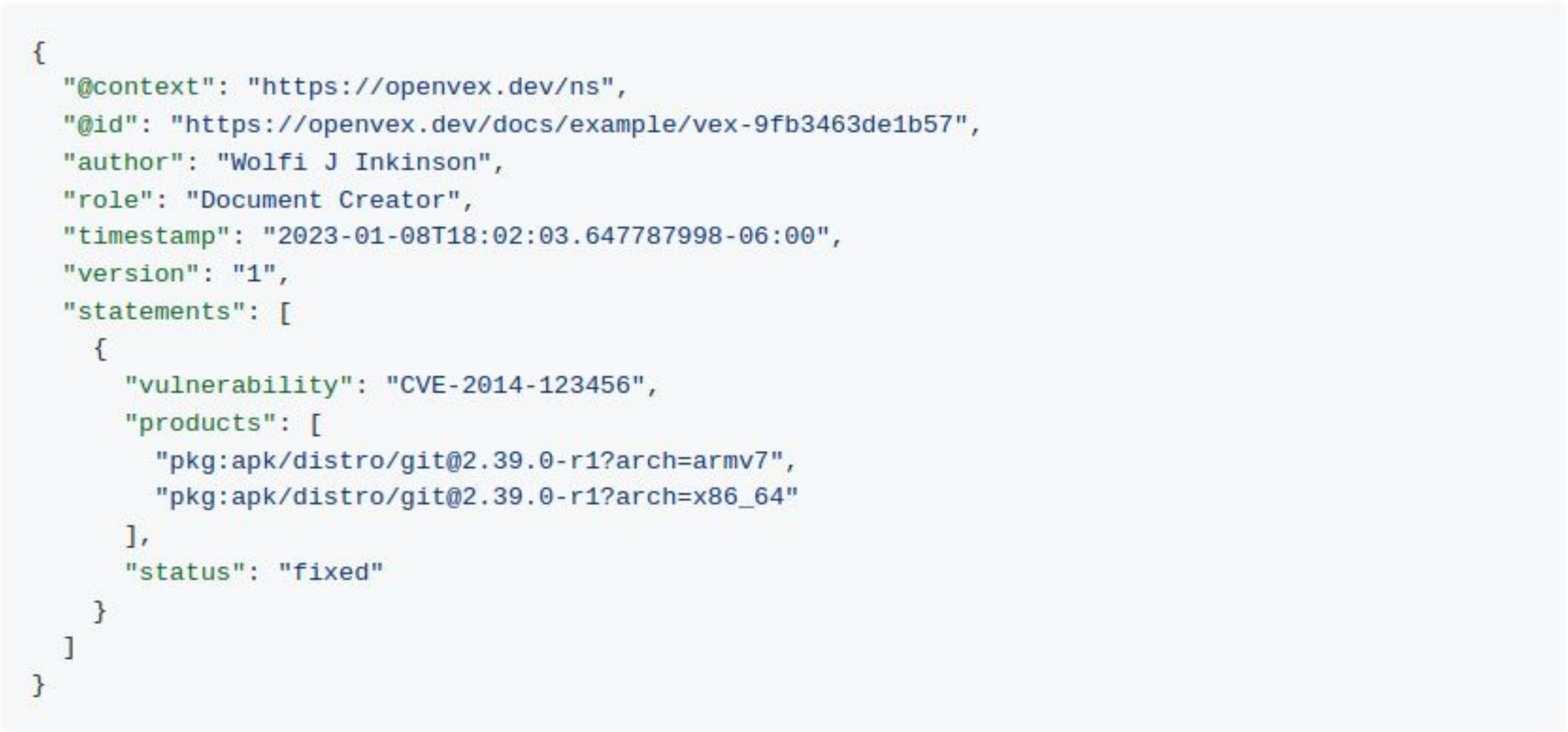
- Software Bill of Materials
  - components used to build software
  - CycloneDX, SPDX
- VEX
  - Vulnerability Exploitability eXchange
  - stating whether software is affected by a vulnerability
- osv.dev VEX generation



```
$ cat .vex
libfoo, CVE-2022-123456, NOT_AFFECTED, inline_mitigations_already_exist
libbar, CVE-2022-654321, NOT_AFFECTED, vulnerable_code_not_in_execute_path
```

- OpenVEX
  - vexctl

OpenVEX

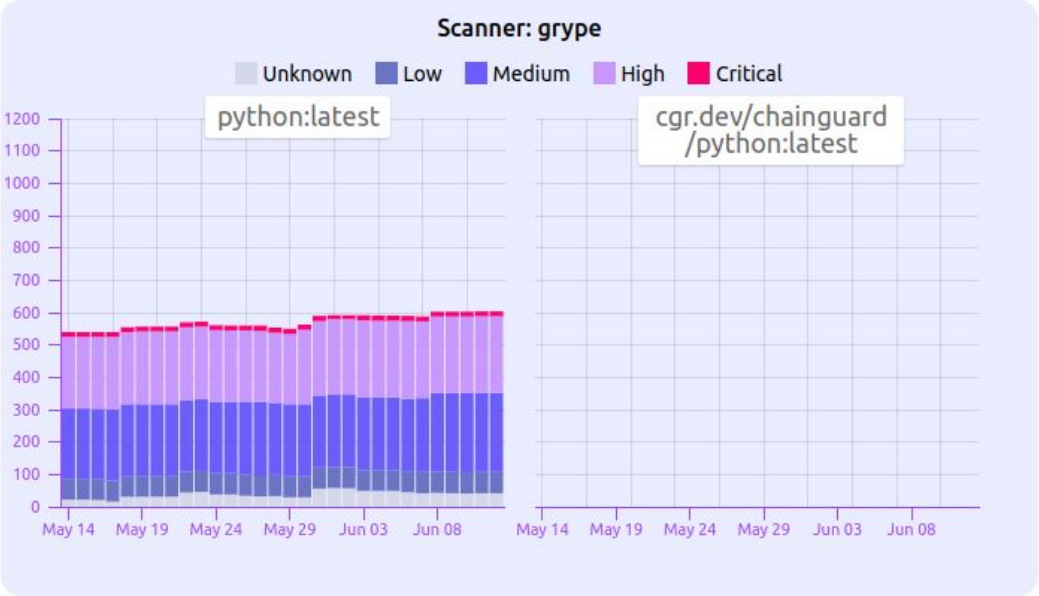


```
{
  "@context": "https://openvex.dev/ns",
  "@id": "https://openvex.dev/docs/example/vex-9fb3463de1b57",
  "author": "Wolfi J Inkinson",
  "role": "Document Creator",
  "timestamp": "2023-01-08T18:02:03.647787998-06:00",
  "version": "1",
  "statements": [
    {
      "vulnerability": "CVE-2014-123456",
      "products": [
        "pkg:apk/distro/git@2.39.0-r1?arch=armv7",
        "pkg:apk/distro/git@2.39.0-r1?arch=x86_64"
      ],
      "status": "fixed"
    }
  ]
}
```

Python container images

- Red Hat, UBI, and Fedora Python Source-to-Image (S2I)
  - Large RPM ecosystem with vetted and maintained packages
  - use micropipenv 😉

- Chainguard's Python image
  - Based on Wolfi
  - uses multi-stage builds
    - python-dev
    - python



Scanning for vulnerabilities in source code

bandit: Find common security issues with static code analysis

- A configurable tool that generates ASTs from Python files and analyses potential risks using **plugins**: hardcoded passwords, shell injection, cryptomining…

Python community initiatives

The **Python community** has deployed important changes to improve the Python packaging supply chain security:

- Mandatory **2FA** on PyPI for maintainers of critical projects (2022) with Security Key giveaway, mandatory for *all* accounts by end of 2023
  - Enforcement of 2FA begins 2023-06-01
- Publishing packages with **Trusted Publishers** using **OpenID Connect**, replacing permanent PyPI API tokens
- Removing PGP from PyPI
- … and more initiatives to come

A glimpse into the future: anticipating Python's Supply Chain Security landscape

PEP 458 & PEP 480

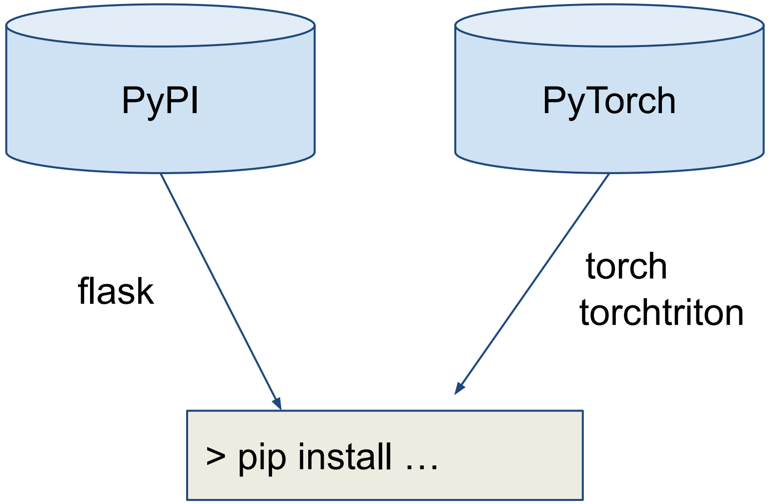
- **PEP 458 – Secure PyPI downloads with signed repository metadata**
  - Accepted
  - Use TUF to secure consumption of Python distributions
  - pypi/warehouse#10672 🚧
- **PEP 480 – Surviving a Compromise of PyPI: End-to-end signing of packages**
  - Draft
  - Built on top of PEP 458, but adding developer keys

- … there might be a new PEP in few days 😉

Dependency Confusion Attack

pip install flask torch --index-url https://pypi.org/simple --extra-index-url https://pytorch...

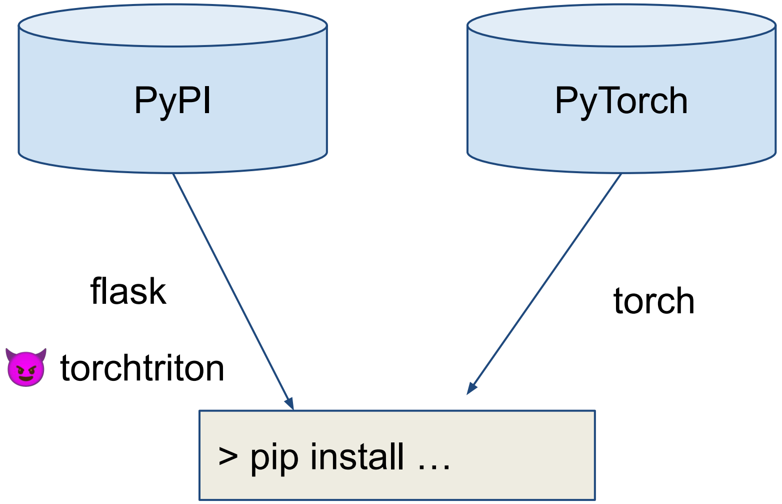
- Indexes are treated as mirrors
- Which index is used to download the requested Python distribution when combining indexes with different artifacts?



Dependency Confusion Attack

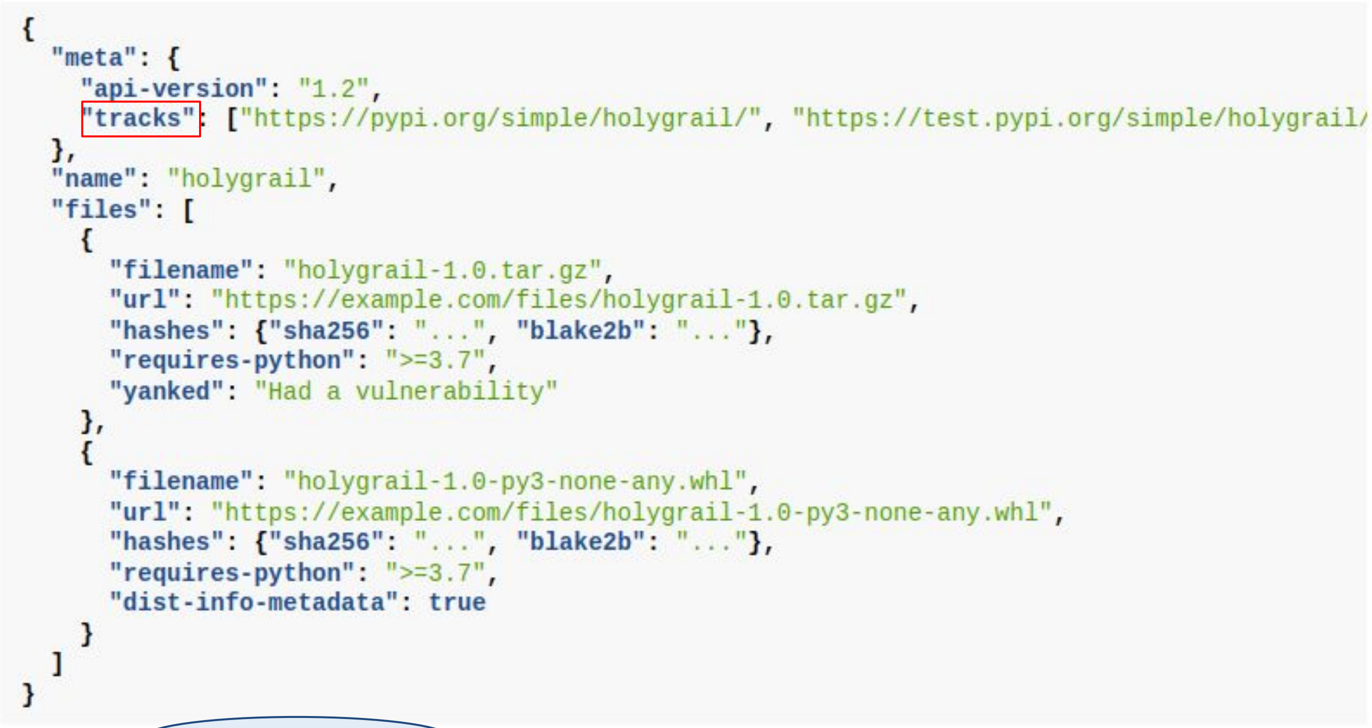
```
pip install flask torch --index-url https://pypi.org/simple --extra-index-url https://pytorch...
```

- Indexes are treated as mirrors
- Which index is used to download the requested Python distribution when combining indexes with different artifacts?
- Detecting possible dependency confusion
    - Yorkshire

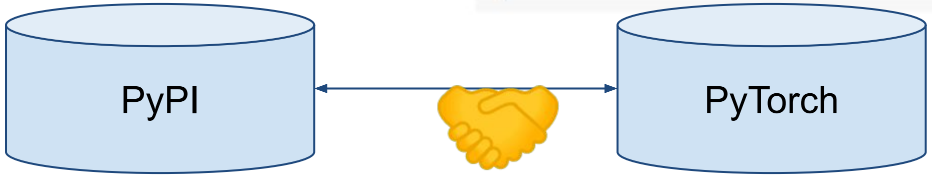


PEP 708: Extending the Repository API to Mitigate Dependency Confusion Attacks

- The PEP is still in draft state
- Creating a link between repositories
- Installers can verify the linkage



```
{
  "meta": {
    "api-version": "1.2",
    "tracks": ["https://pypi.org/simple/holygrail/", "https://test.pypi.org/simple/holygrail/
  },
  "name": "holygrail",
  "files": [
    {
      "filename": "holygrail-1.0.tar.gz",
      "url": "https://example.com/files/holygrail-1.0.tar.gz",
      "hashes": {"sha256": "...", "blake2b": "..."},
      "requires-python": ">=3.7",
      "yanked": "Had a vulnerability"
    },
    {
      "filename": "holygrail-1.0-py3-none-any.whl",
      "url": "https://example.com/files/holygrail-1.0-py3-none-any.whl",
      "hashes": {"sha256": "...", "blake2b": "..."},
      "requires-python": ">=3.7",
      "dist-info-metadata": true
    }
  ]
}
```

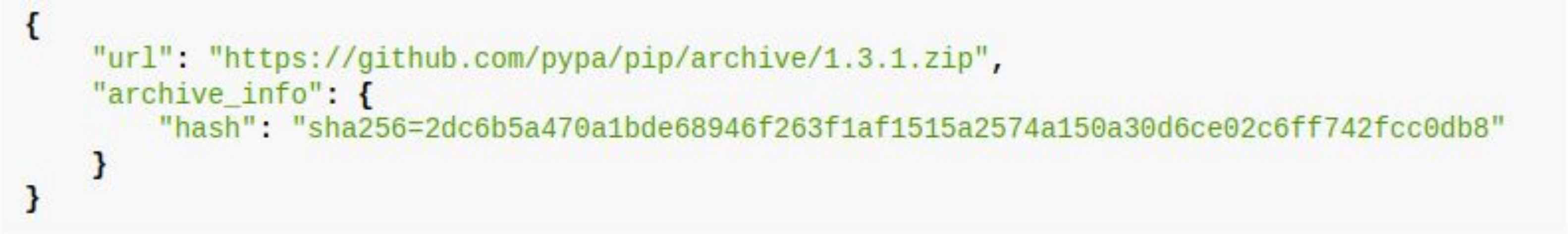


Project torchtriton on PyPI tracks torchtriton on the PyTorch index

Project torchtriton on PyPI tracks torchtriton on the PyTorch index

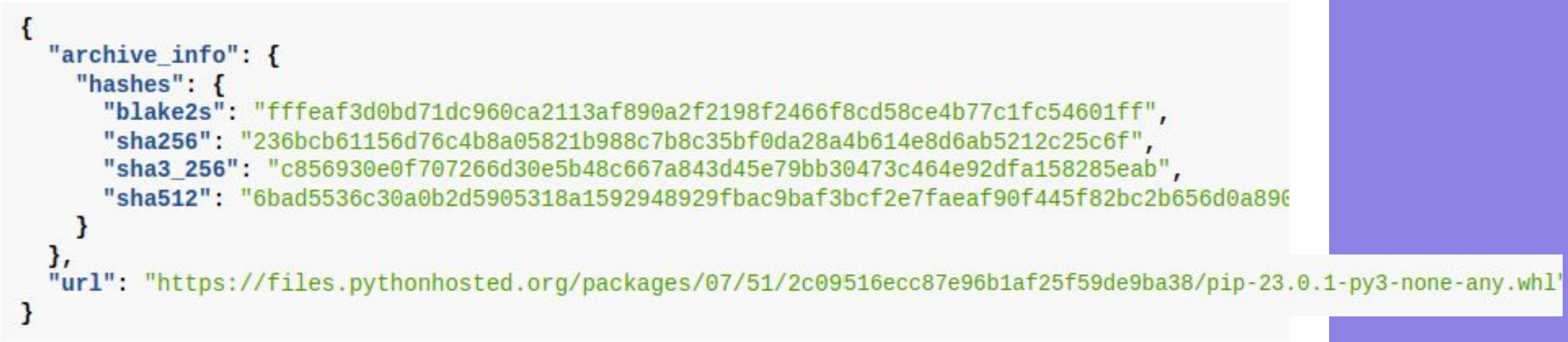
PEP 710: Recording the provenance of installed packages

- **PEP-610: Recording the Direct URL Origin of installed distributions - direct_url.json**



```
{
    "url": "https://github.com/pypa/pip/archive/1.3.1.zip",
    "archive_info": {
        "hash": "sha256=2dc6b5a470a1bde68946f263f1af1515a2574a150a30d6ce02c6ff742fcc0db8"
    }
}
```

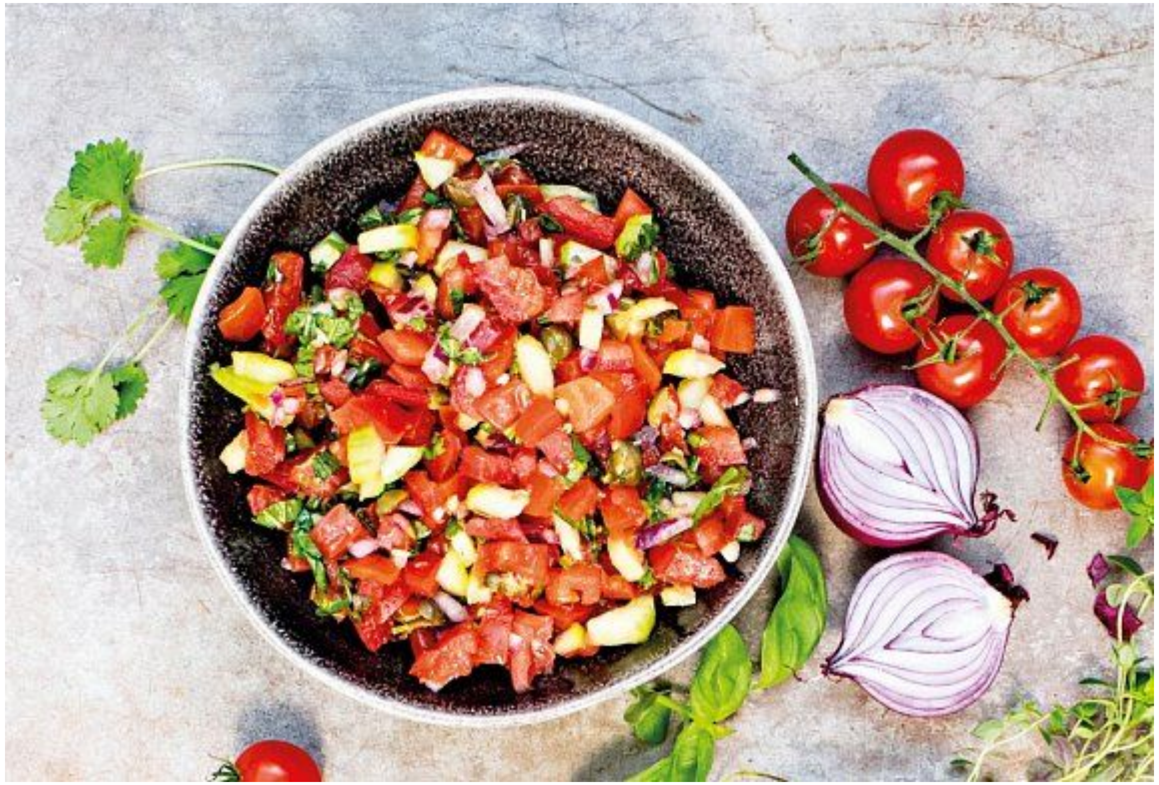
- **PEP-710: provenance_url.json**



```
{
  "archive_info": {
    "hashes": {
      "blake2s": "fffeaf3d0bd71dc960ca2113af890a2f2198f2466f8cd58ce4b77c1fc54601ff",
      "sha256": "236bcb61156d76c4b8a05821b988c7b8c35bf0da28a4b614e8d6ab5212c25c6f",
      "sha3_256": "c856930e0f707266d30e5b48c667a843d45e79bb30473c464e92dfa158285eab",
      "sha512": "6bad5536c30a0b2d5905318a1592948929fbac9baf3bcf2e7faeaf90f445f82bc2b656d0a89
    }
  },
  "url": "https://files.pythonhosted.org/packages/07/51/2c09516ecc87e96b1af25f59de9ba38/pip-23.0.1-py3-none-any.whl"
}
```

An opportunity to win...

Q0: To which project mentioned in this presentation does this photo relate to?



Source

SLSA
Supply-chain Levels for Software Artifacts

Source

Q1: To which project mentioned in this presentation does this photo relate to?

Source

GUAC
Graph for Understanding Artifact Composition

Source

Q2: To which project mentioned in this presentation does this photo relate to?

Source

SLSA
Supply-chain Levels for Software Artifacts

Source

SALSA
party
Clubname
05.23 / 22:00
dreamstime.com
ID 229696478 © Pavel Kudriavtsev

Source

Q3: To which project mentioned in this presentation does this photo relate to?

Source

GuardDog or Yorkshire?

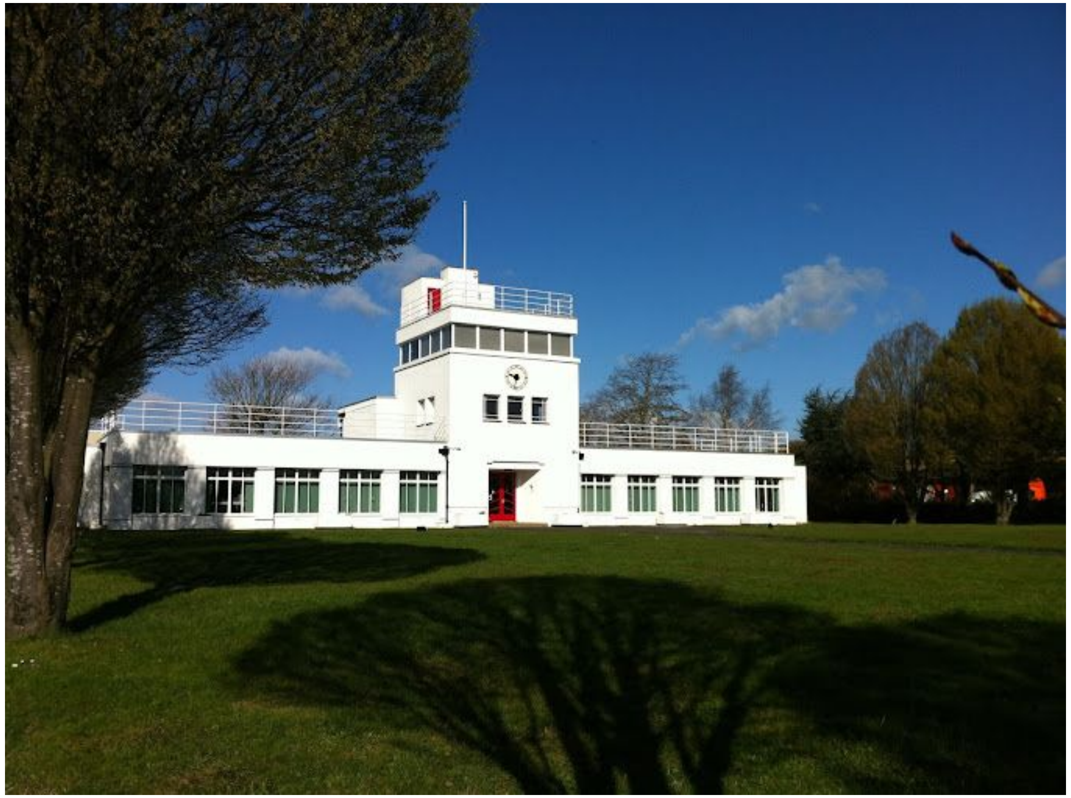
GUARDDOG
by DATADOG

Source

GuardDog or Yorkshire?

Source

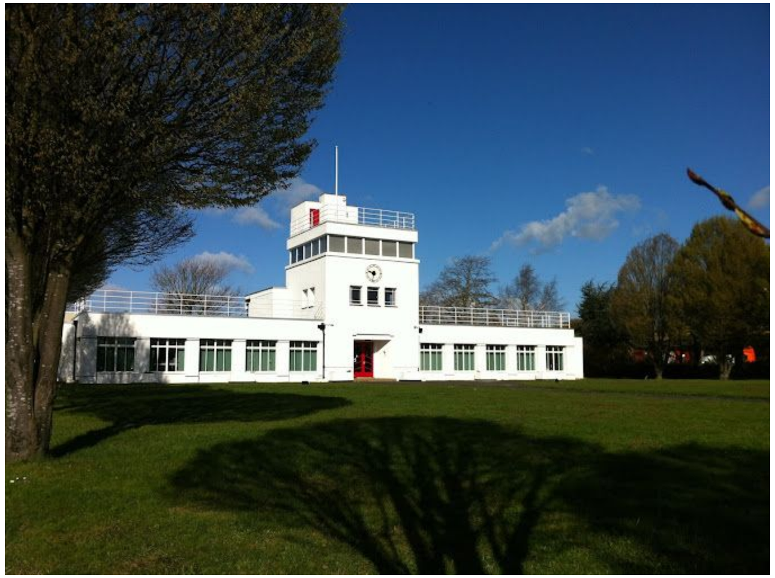
Q4: To which project mentioned in this presentation does this photo relate to?



Google reviews

Q4: To which project mentioned in this presentation does this photo relate to?

Hint: “The Signature Store is an established market leader selling genuine signed automobilia, all our signed products are guaranteed authentic, and come with proof including date, location and photographic evidence - we don't deal in 'through the fence' signatures, but strike exclusive, commercial signings with drivers which take place in private; we get the best signatures, because we pay for the best; authenticity is guaranteed.”



The Signature Store in the United Kingdom

Google reviews and https://www.thesignaturestore.co.uk/

Q4: To which project mentioned in this presentation does this photo relate to?

sigstore

Google reviews and https://www.thesignaturestore.co.uk/

Thank you!
Q&A