SunSecCon

Exposed Detection Rules: What's Next for Defenders?

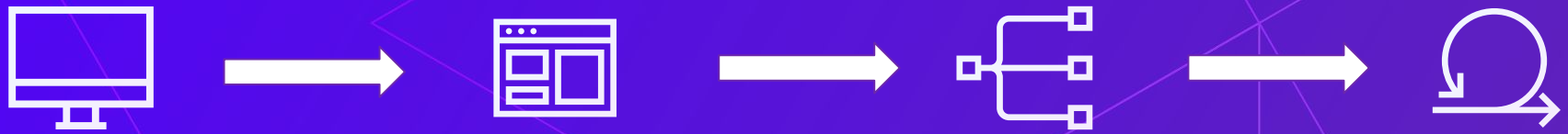
DATADOG

Kennedy Toomey

Application Security Researcher & Advocate @ Datadog

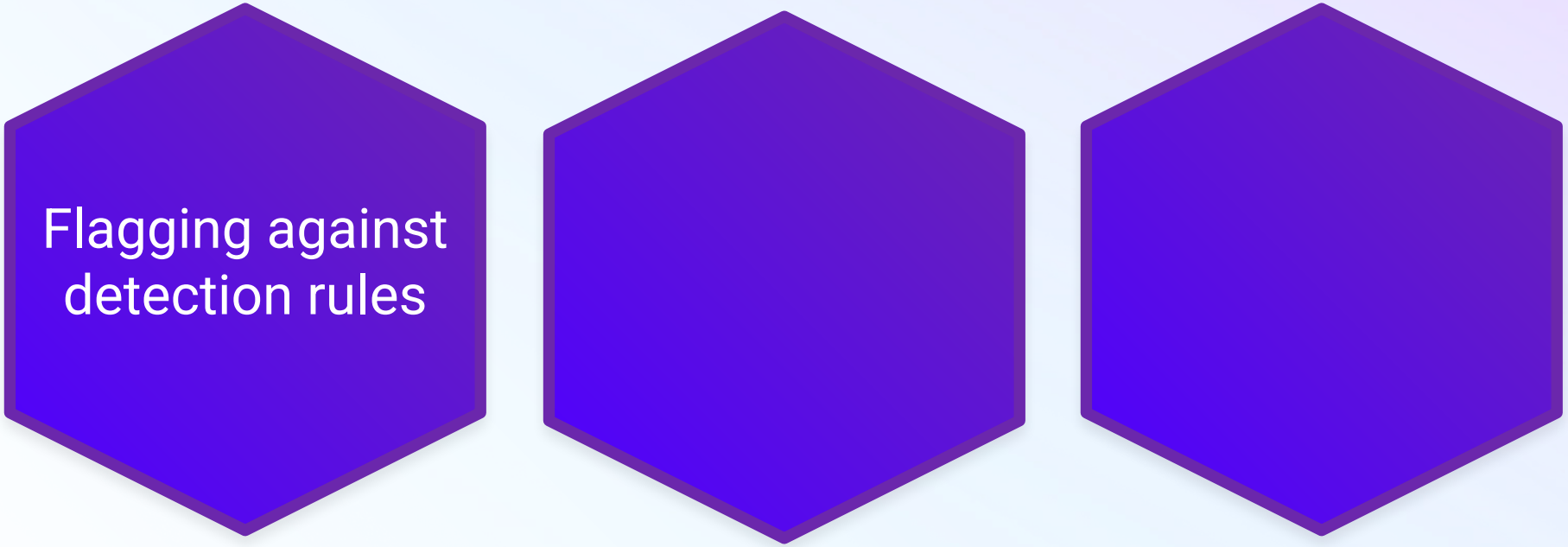
Open Source Security Tools

Focusing on the SDLC



How do these tools work?

How do these tools work?



How do these tools work?

How do these tools work?

Key Questions

Key Questions

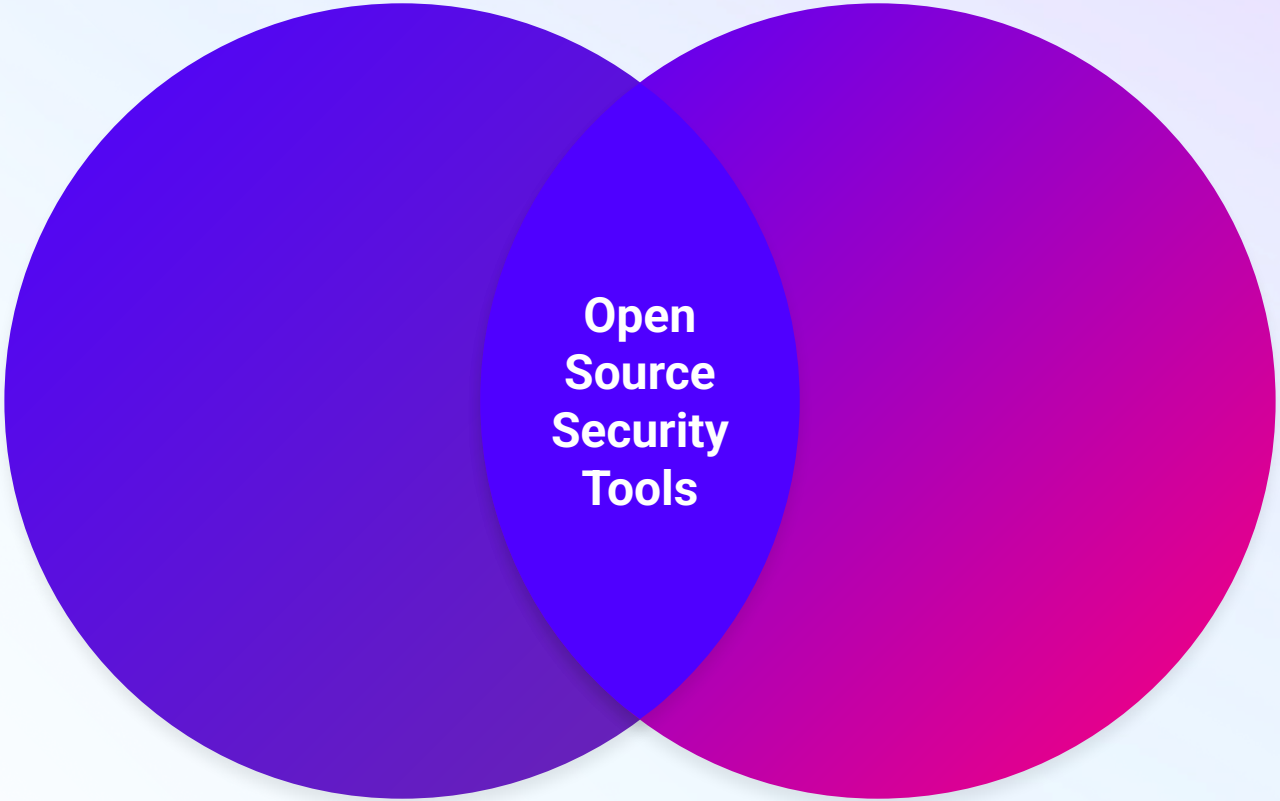
How do attackers leverage open source security tools?

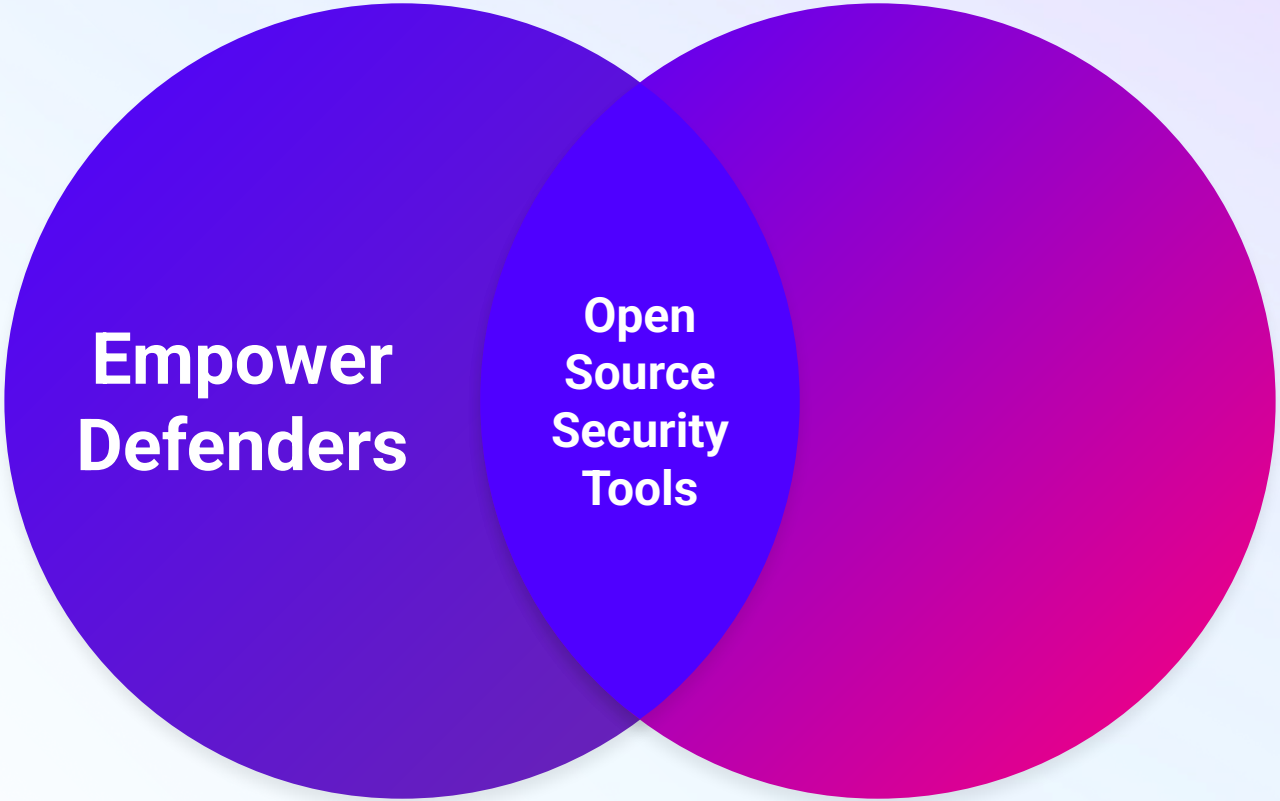
Key Questions

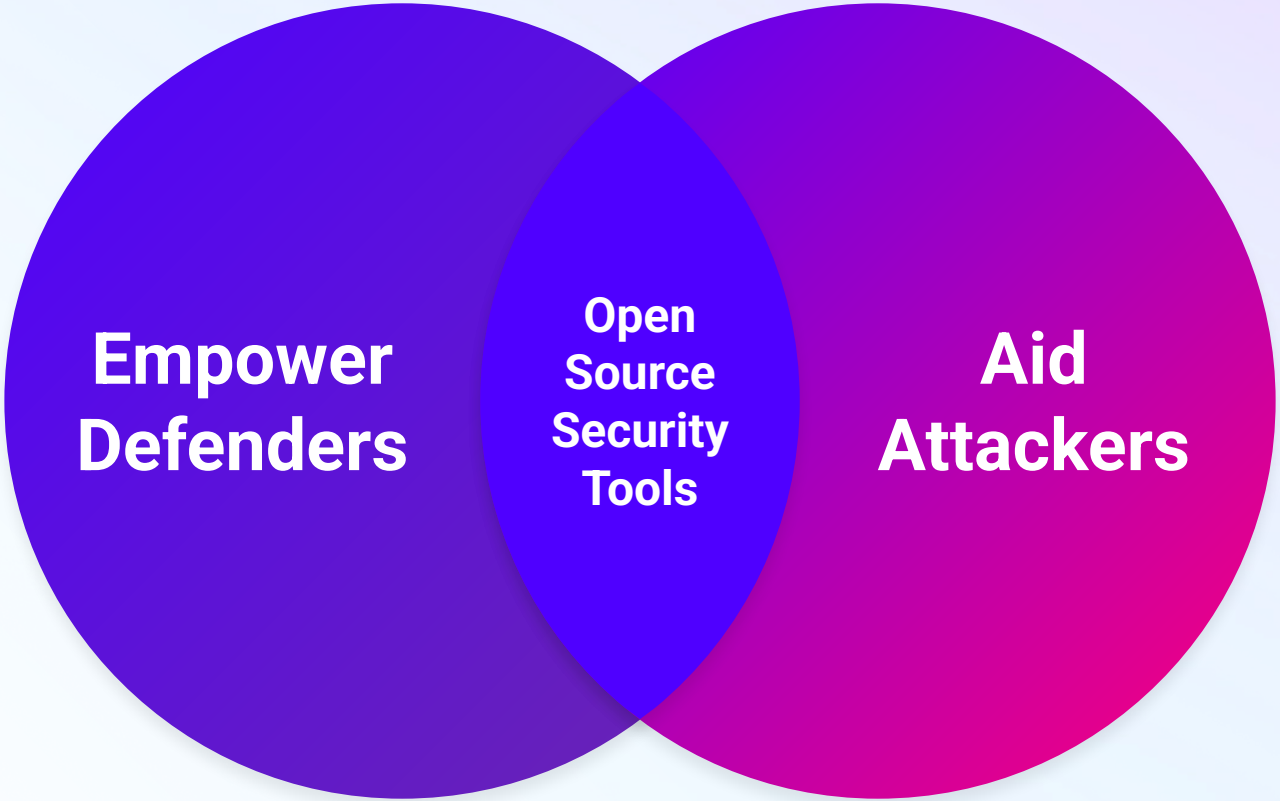
How do attackers leverage open source security tools?

How can open source contributors stay ahead of adversaries?

The Challenge


Open
Source
Security
Tools


Empower
Defenders
Open
Source
Security
Tools


Empower Defenders
Open Source Security Tools
Aid Attackers

Empowering Defenders

Empowering Defenders

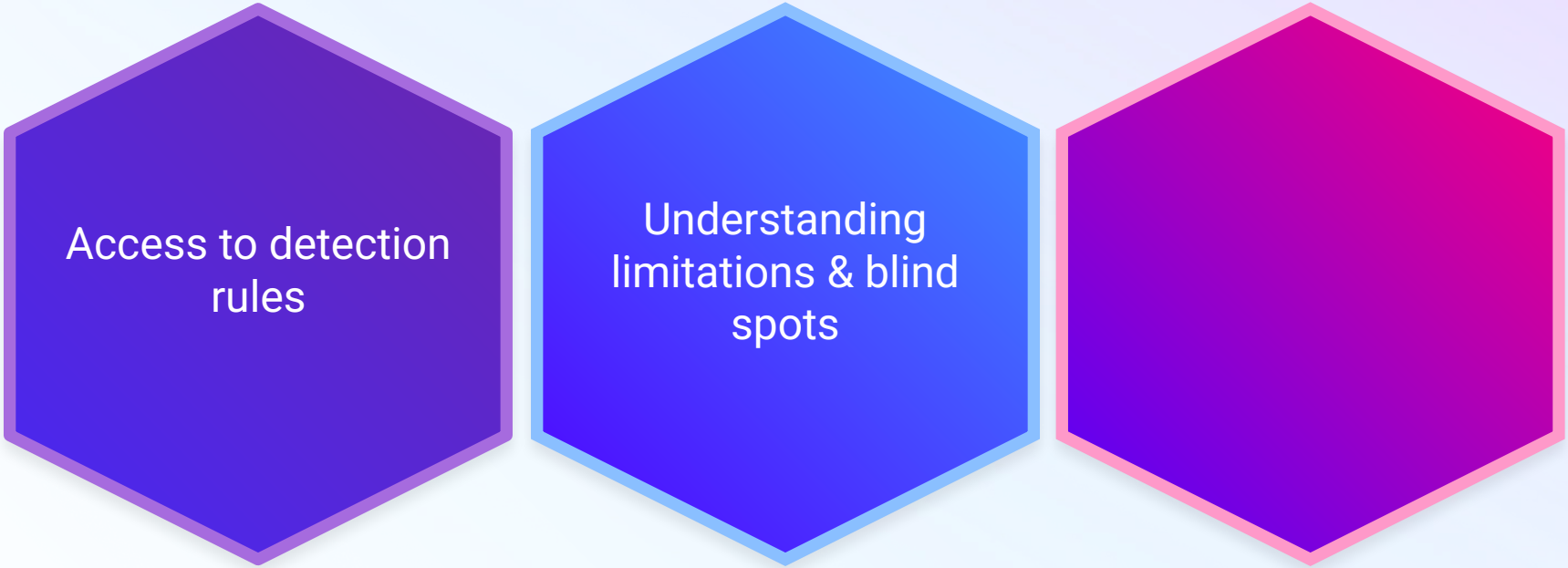
Empowering Defenders

Empowering Defenders

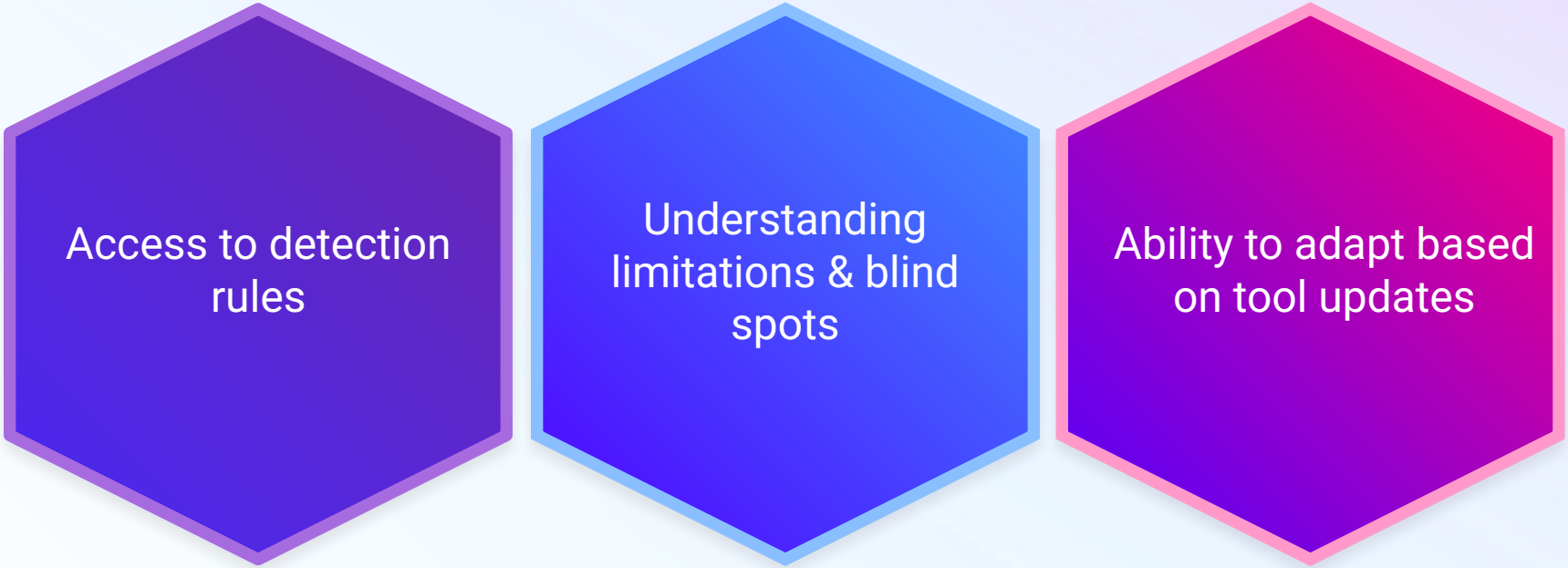
Aiding Attackers

Aiding Attackers

Aiding Attackers



Aiding Attackers



Case Studies:

GuardDog and the Supply-Chain Firewall

GuardDog

GuardDog

Open source tool to identify malicious packages

GuardDog

Open source tool to identify malicious packages

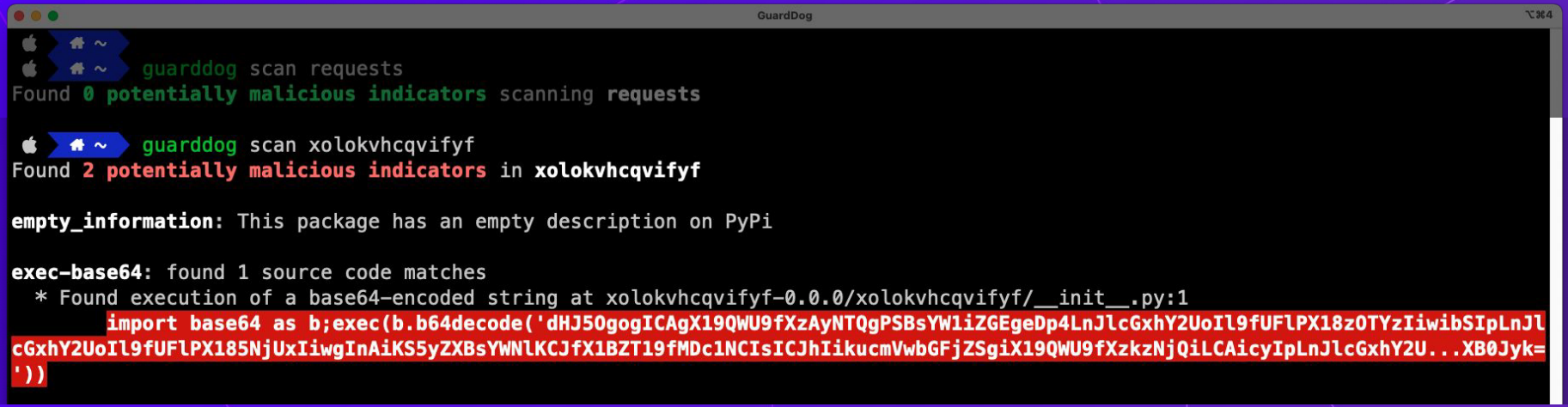
Uses metadata analysis, semgrep rules, and yara rules

GuardDog


GUARDDOG
by DATADOG


GuardDog
⌥⌘4
~
~ guarddog scan requests
Found 0 potentially malicious indicators scanning requests

GuardDog


GuardDog
~
guarddog scan requests
Found 0 potentially malicious indicators scanning requests
~
guarddog scan xolokvhcqvifyf
Found 2 potentially malicious indicators in xolokvhcqvifyf
empty_information: This package has an empty description on PyPi
exec-base64: found 1 source code matches
* Found execution of a base64-encoded string at xolokvhcqvifyf-0.0.0/xolokvhcqvifyf/__init__.py:1
import base64 as b;exec(b.b64decode('dHJ5OgogICAgX19QWU9fXzAyNTQgPSBsYW1iZGEgeDp4LnJlcGxhY2UoIl9fUFlPX18zOTYzIiwibSIpLnJlcGxhY2UoIl9fUFlPX185NjUxIiwgInAiKS5yZXBsYWNlKCJfX1BZT19fMDc1NCIsICJhIikucmVwbGFjZSgiX19QWU9fXzkzNjQiLCAicyIpLnJlcGxhY2U...XB0Jyk='))

GuardDog

GuardDog ⌥⌘4

```
~
~ guarddog scan requests
Found 0 potentially malicious indicators scanning requests

~ guarddog scan xolokvhcqvifyf
Found 2 potentially malicious indicators in xolokvhcqvifyf

empty_information: This package has an empty description on PyPi

exec-base64: found 1 source code matches
  * Found execution of a base64-encoded string at xolokvhcqvifyf-0.0.0/xolokvhcqvifyf/__init__.py:1
        import base64 as b;exec(b.b64decode('dHJ5OgogICAgX19QWU9fXzAyNTQgPSBsYW1iZGEgeDp4LnJlcGxhY2UoIl9fUFlPX18zOTYzIiwibSIpLnJlcGxhY2UoIl9fUFlPX185NjUxIiwgInAiKS5yZXBsYWNlKCJfX1BZT19fMDc1NCIsICJhIikucmVwbGFjZSgiX19QWU9fXzkzNjQiLCAicyIpLnJlcGxhY2U...XB0Jyk='))

~ guarddog scan beautifulsup4
Found 2 potentially malicious indicators in beautifulsup4

typosquatting: This package closely ressembles the following package names, and might be a typosquatting attempt: beautifulsoup4

empty_information: This package has an empty description on PyPi
```

GuardDog Detection Rules

guarddog / guarddog / analyzer / sourcecode / **cmd-overwrite.yml**

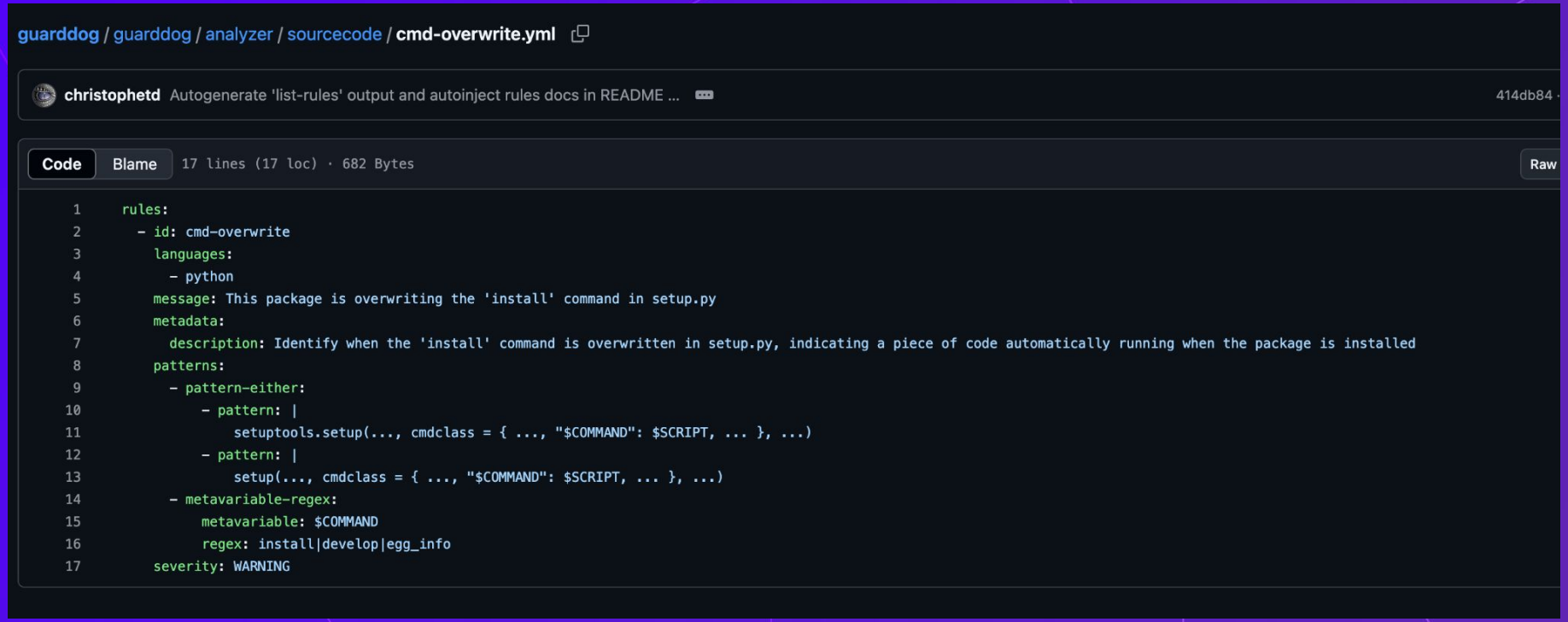
christophetd Autogenerate 'list-rules' output and autoinject rules docs in README ... 414db84

Code | Blame | 17 lines (17 loc) · 682 Bytes | Raw

```yaml
rules:
  - id: cmd-overwrite
    languages:
      - python
    message: This package is overwriting the 'install' command in setup.py
    metadata:
      description: Identify when the 'install' command is overwritten in setup.py, indicating a piece of code automatically running when the package is installed
    patterns:
      - pattern-either:
          - pattern: |
              setuptools.setup(..., cmdclass = { ..., "$COMMAND": $SCRIPT, ... }, ...)
          - pattern: |
              setup(..., cmdclass = { ..., "$COMMAND": $SCRIPT, ... }, ...)
      - metavariable-regex:
          metavariable: $COMMAND
          regex: install|develop|egg_info
    severity: WARNING
```

Evading GuardDog

Evading GuardDog



Maintaining GuardDog

Maintaining Guarddog

Maintaining Guarddog

Check for new malicious packages with advisories on OSV.dev

Stay up to date with blog posts from other researchers

Maintaining Guarddog - the Challenge

Maintaining Guarddog - the Challenge

Supply-Chain Firewall

Supply-Chain Firewall

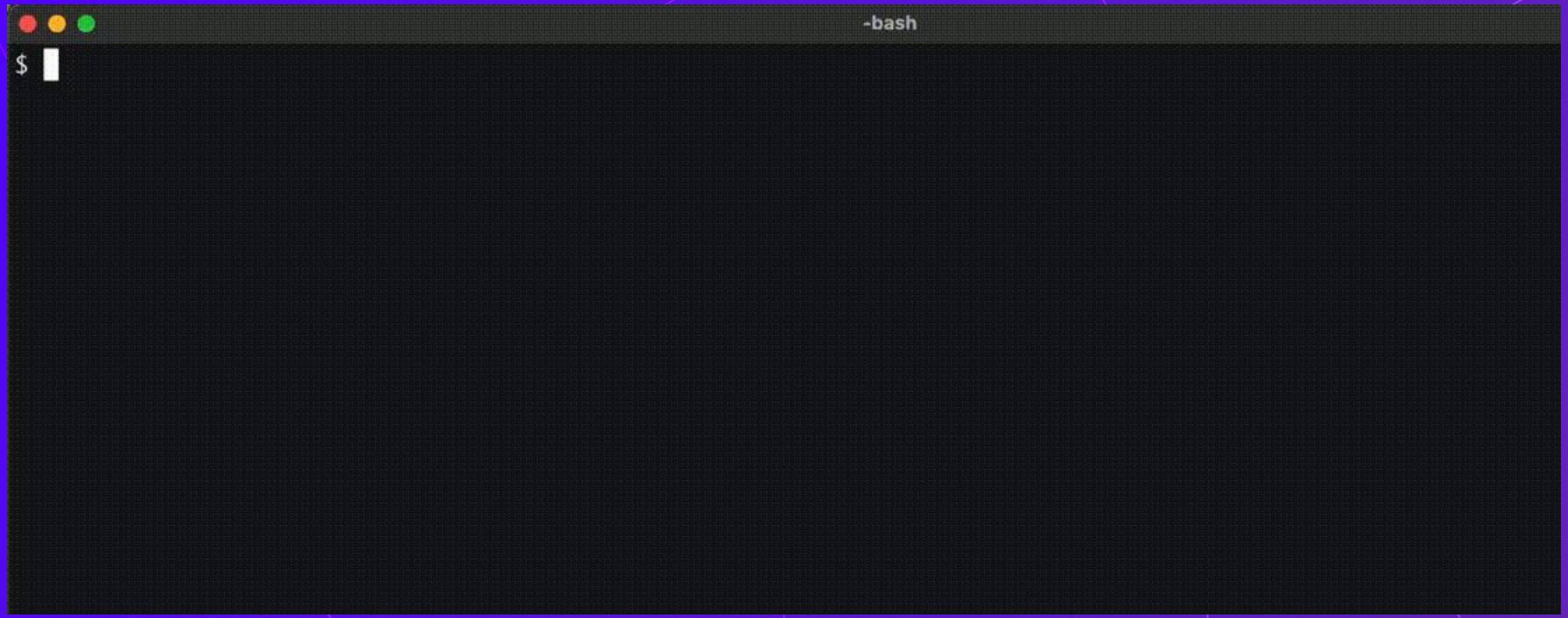
Open source tool to block malicious packages from installation

Supply-Chain Firewall

Open source tool to block malicious packages from installation

Uses a database for detection

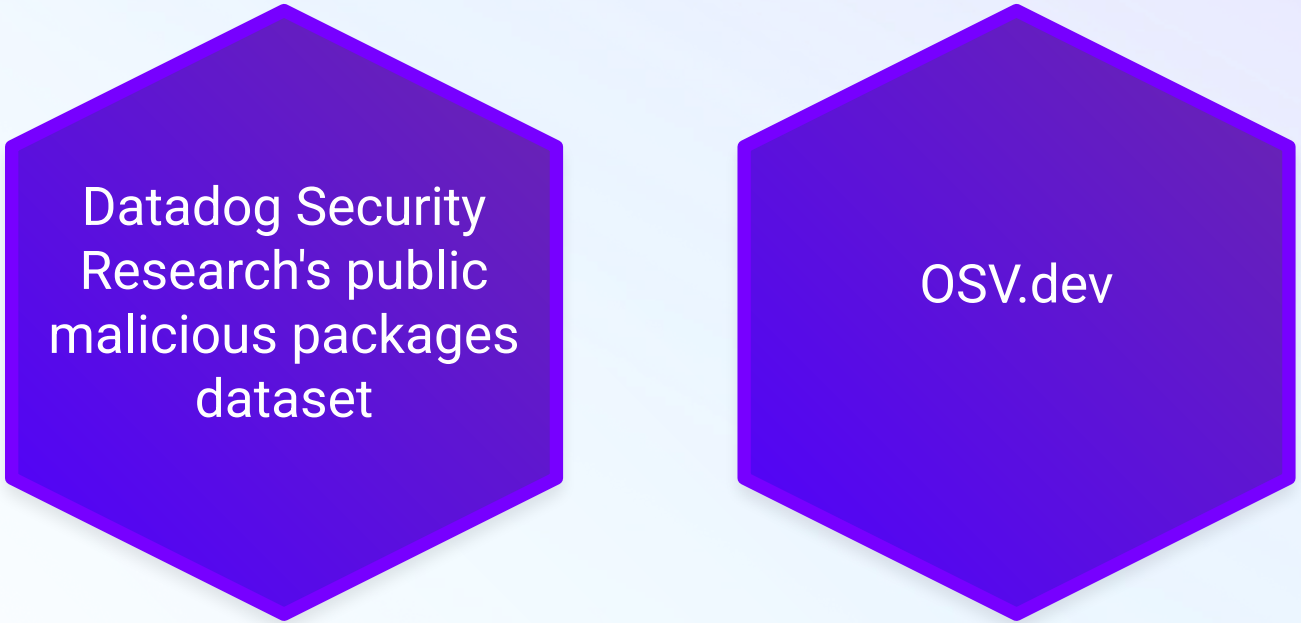
Supply-Chain Firewall



Supply-Chain Firewall Databases

Datadog Security Research's public malicious packages dataset

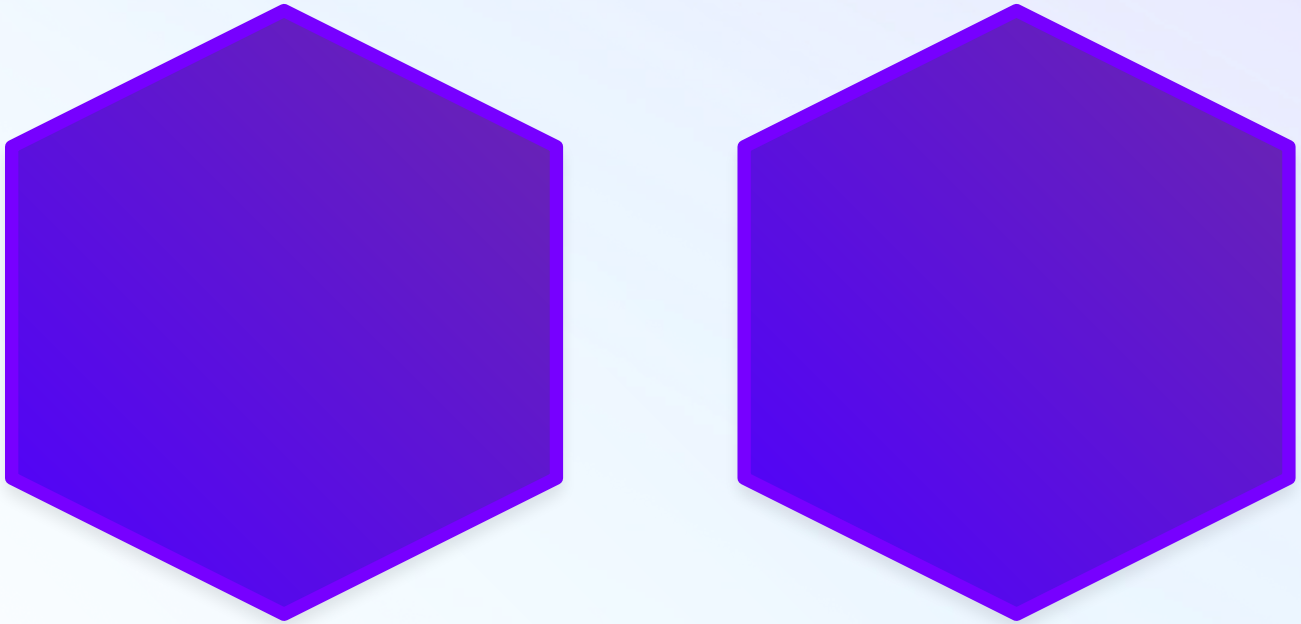
Supply-Chain Firewall Databases



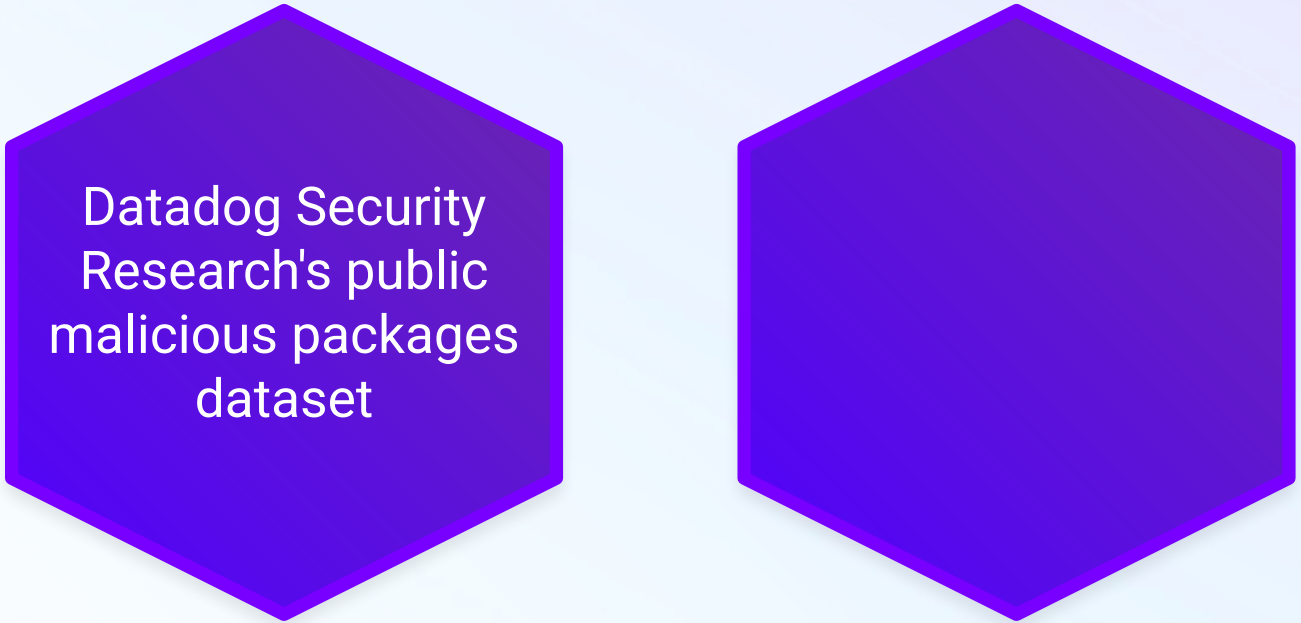
Evading the SCFW

The malicious package has to exist in the database.

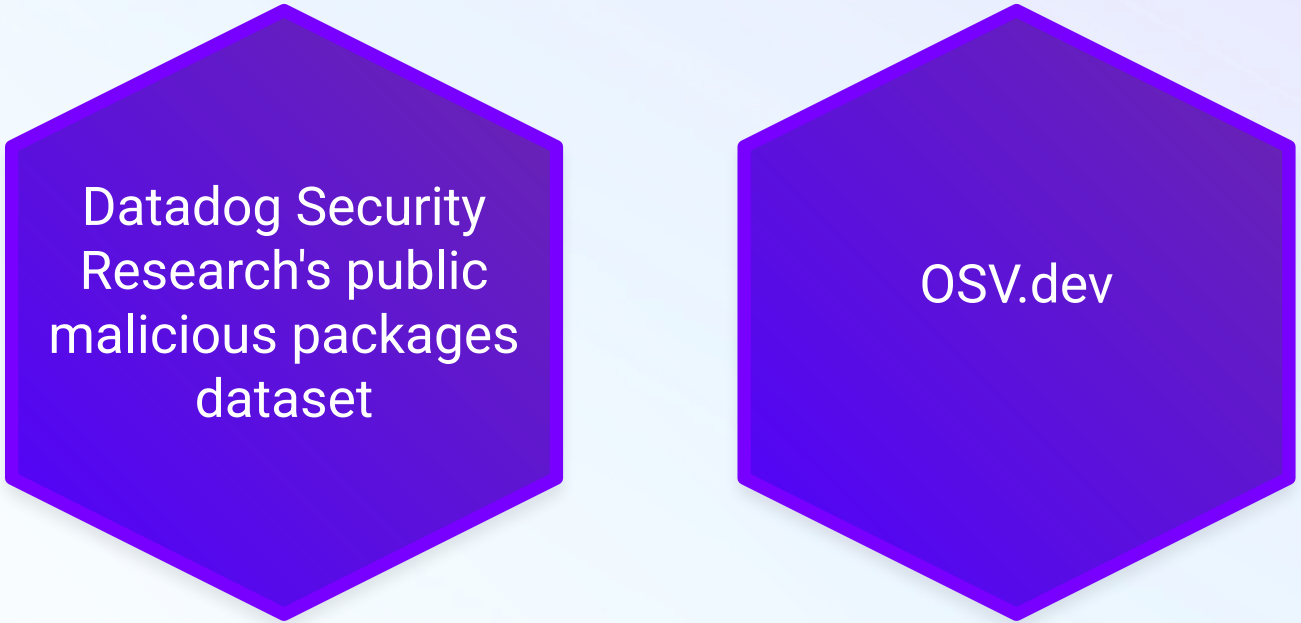
Database Accuracy and Trust



Database Accuracy and Trust



Database Accuracy and Trust



Staying Ahead

Staying Ahead

(or at least not falling behind)

Staying Ahead

Proactive Threat Modeling

Staying Ahead

Proactive Threat Modeling

Leverage the Community

Staying Ahead

Proactive Threat Modeling

Leverage the Community

Automate Adaptive Defense Mechanisms

How You Can Help!

How You Can Help!

How You Can Help!

How You Can Help!

The Bad News

The Good News

Datadog Security Newsletter

https://securitylabs.datadoghq.com/newsletters/

DATADOG

Repositories Discussed

https://github.com/DataDog/guarddog

https://github.com/DataDog/supply-chain-firewall

https://github.com/DataDog/malicious-software-packages-dataset

Questions?

Thank you!