

# Exposed Detection Rules: What's Next for Defenders?



DATADOG

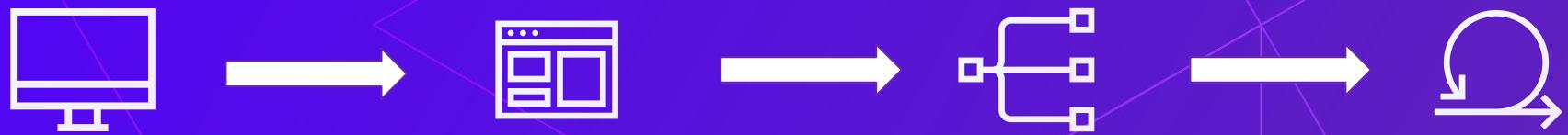


# Kennedy Toomey

Application Security Researcher & Advocate @ Datadog

# Open Source Security Tools

# Focusing on the SDLC



# How do these tools work?

# How do these tools work?

Flagging against  
detection rules

# How do these tools work?

Flagging against  
detection rules

Comparing  
against a  
database

# How do these tools work?

Flagging against  
detection rules

Comparing  
against a  
database

Running an  
attack script

# Key Questions

# Key Questions

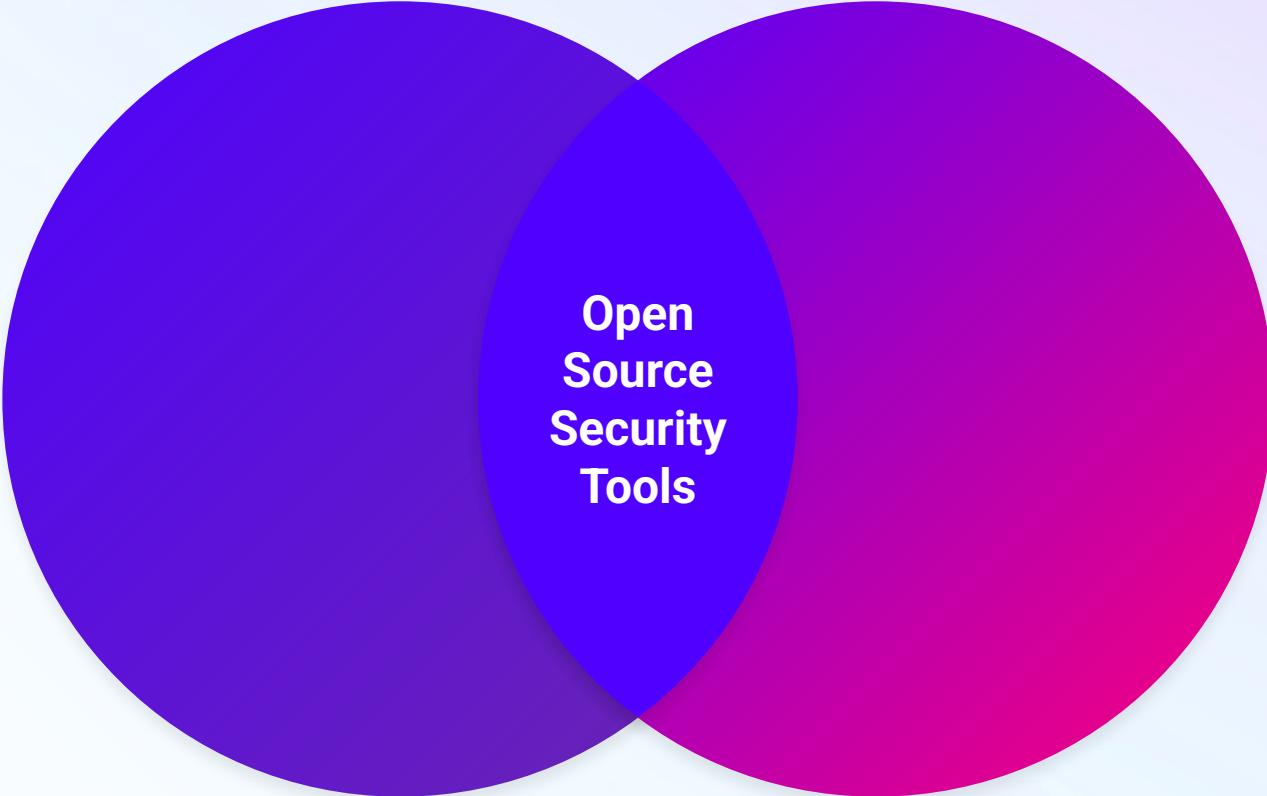
How do attackers  
leverage open source  
security tools?

# Key Questions

How do attackers  
leverage open source  
security tools?

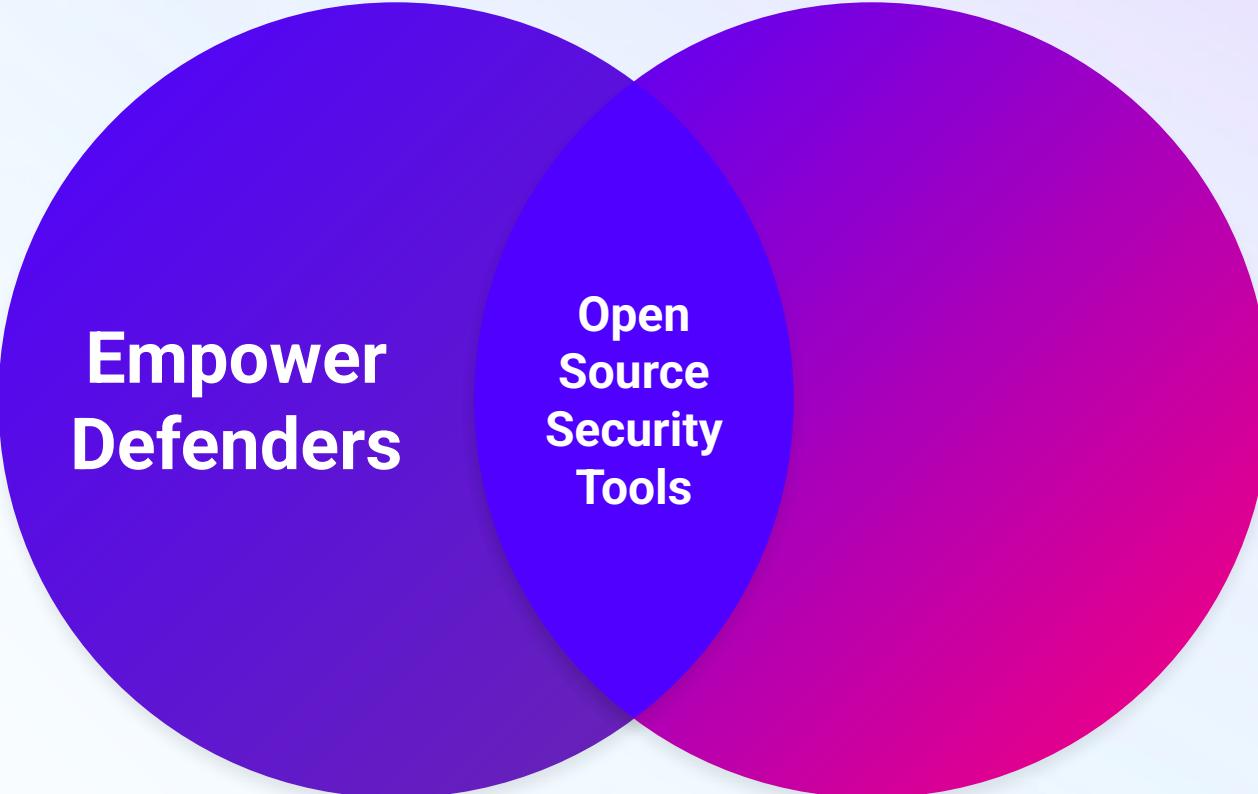
How can open source  
contributors stay ahead  
of adversaries?

# The Challenge



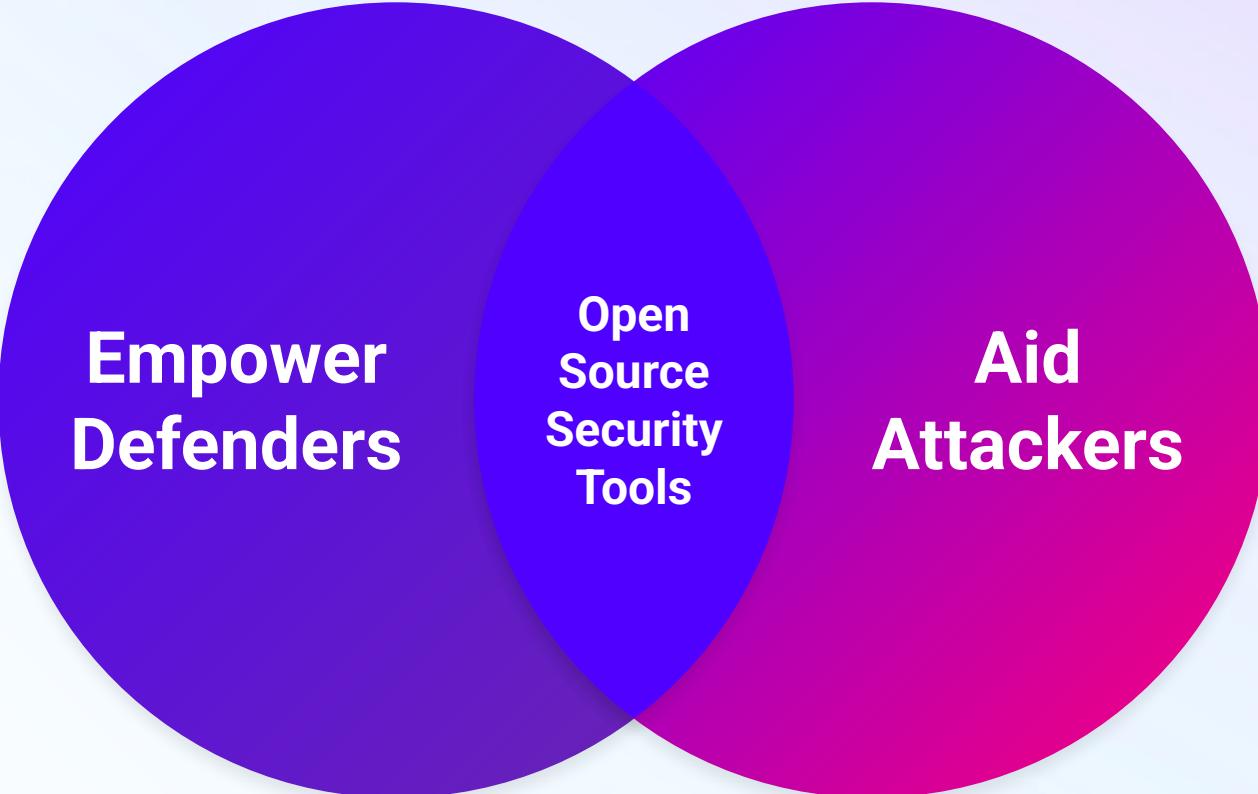
A Venn diagram consisting of two overlapping circles. The left circle is blue and the right circle is red. The overlapping area, where the two circles intersect, contains the text "Open Source Security Tools".

**Open  
Source  
Security  
Tools**



**Empower  
Defenders**

**Open  
Source  
Security  
Tools**

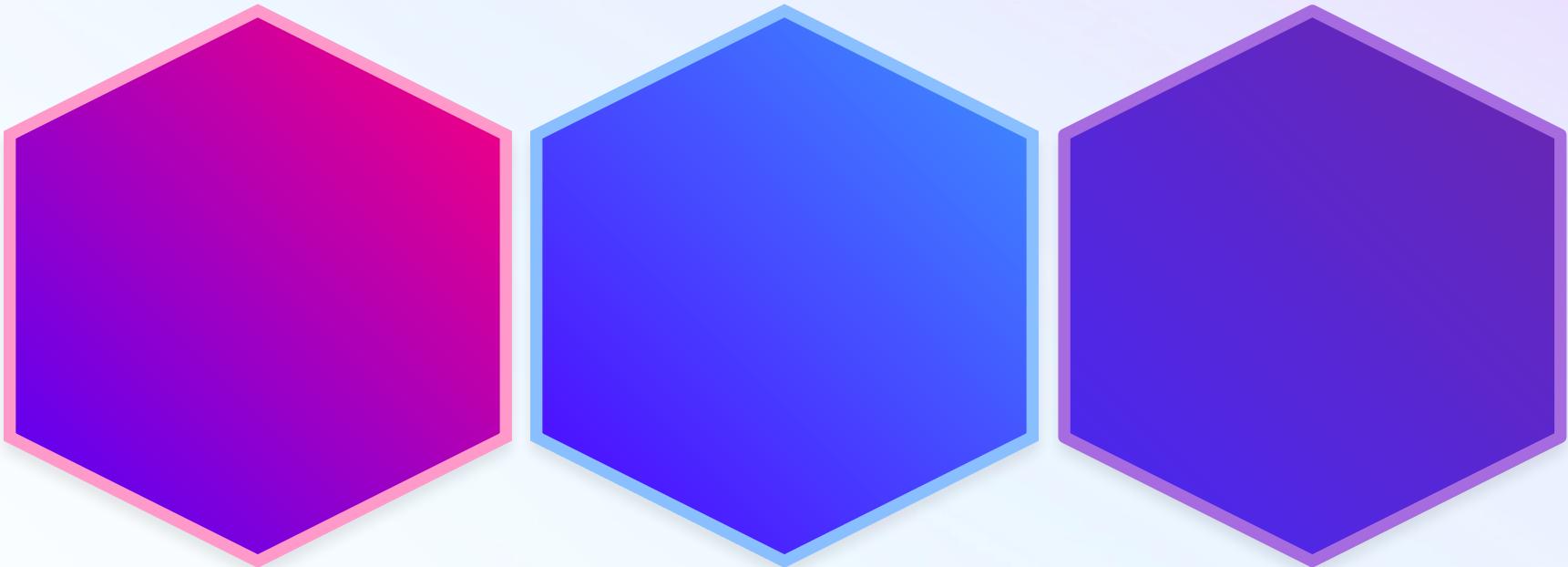


**Empower  
Defenders**

**Open  
Source  
Security  
Tools**

**Aid  
Attackers**

# Empowering Defenders



# Empowering Defenders



Accessible security  
solutions

# Empowering Defenders



Accessible security  
solutions

Community-driven  
improvements

# Empowering Defenders

Accessible security  
solutions

Community-driven  
improvements

Rapid innovation

# Aiding Attackers



# Aiding Attackers



Access to detection  
rules

# Aiding Attackers

Access to detection  
rules

Understanding  
limitations & blind  
spots

# Aiding Attackers

Access to detection  
rules

Understanding  
limitations & blind  
spots

Ability to adapt based  
on tool updates

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

# GuardDog



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

guarddog scan xolokvhcqifyf
Found 2 potentially malicious indicators in xolokvhcqifyf

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 xolokvhcqifyf-0.0.0/xolokvhcqifyf/__init__.py:1
   import base64 as b;exec(b.b64decode('dHJ50gogICAgX19QWU9fXzAyNTQgPSBsYW1iZGEgeDp4LnJlcGxhY2UoIl9fUFlPX18z0TYzIiwibSIpLnJl
cGxhY2UoIl9fUFlPX185NjUxIiwgInAiKS5yZXBsYWNlKCJfx1BZT19fMDc1NCIsICJhIikucmVwbGFjZSgiX19QWU9fXzkzNjQiLCAicyIpLnJlcGxhY2U...XB0Jyk=')).
```



DATADOG

# GuardDog



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

guarddog scan xolokvhcqifyf
Found 2 potentially malicious indicators in xolokvhcqifyf

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 xolokvhcqifyf-0.0.0/xolokvhcqifyf/__init__.py:1
  import base64 as b;exec(b.b64decode('dHJ50ogICAgX19QWU9fXzAyNTQgPSBsYW1iZGEgeDp4LnJlcGxhY2UoIl9fUFlPX18z0TYzIiwibSIpLnJl
cGxhY2UoIl9fUFlPX185NjUxIiwgInAiKS5yZXBsYWNlKCJfx1BZT19fMDc1NCIsICJhIikucmVwbGFjZSgiX19QWU9fXzkzNjQiLCi
acyIpLnJlcGxhY2U...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 

 christophedt Autogenerate 'list-rules' output and autoinject rules docs in README ... 

414db84 ·

**Code**    Blame    17 lines (17 loc) · 682 Bytes

Raw

```
1   rules:
2     - id: cmd-overwrite
3       languages:
4         - python
5       message: This package is overwriting the 'install' command in setup.py
6       metadata:
7         description: Identify when the 'install' command is overwritten in setup.py, indicating a piece of code automatically running when the package is installed
8       patterns:
9         - pattern-either:
10           - pattern: |
11             setuptools.setup(..., cmdclass = { ..., "$COMMAND": $SCRIPT, ... }, ...)
12           - pattern: |
13             setup(..., cmdclass = { ..., "$COMMAND": $SCRIPT, ... }, ...)
14         - metavariable-regex:
15           metavariable: $COMMAND
16           regex: install|develop|egg_info
17       severity: WARNING
```

# Evading GuardDog

# Evading GuardDog

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

 christophedt Autogenerate 'list-rules' output and autoinject rules docs in README ... 

414db84 ·

**Code**    Blame    17 lines (17 loc) · 682 Bytes

Raw

```
1  rules:
2    - id: cmd-overwrite
3      languages:
4        - python
5      message: This package is overwriting the 'install' command in setup.py
6      metadata:
7        description: Identify when the 'install' command is overwritten in setup.py, indicating a piece of code automatically running when the package is installed
8      patterns:
9        - pattern-either:
10          - pattern: |
11            setuptools.setup(..., cmdclass = { ..., "$COMMAND": $SCRIPT, ... }, ...)
12          - pattern: |
13            setup(..., cmdclass = { ..., "$COMMAND": $SCRIPT, ... }, ...)
14        - metavariable-regex:
15          metavariable: $COMMAND
16          regex: install|develop|egg_info
17        severity: WARNING
```

# Maintaining GuardDog

# Maintaining Guarddog



Check for new  
malicious packages  
with advisories on  
OSV.dev



# 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



Balance new rules  
while ensuring a low  
false positive rate

# 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

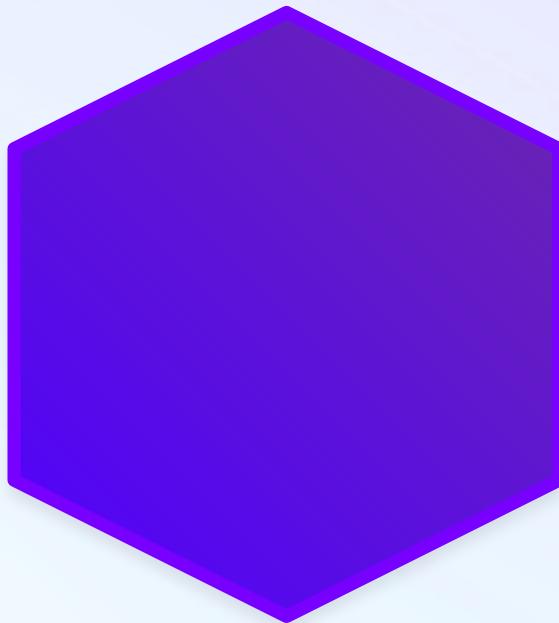


```
-bash
$ |
```

A large black rectangular area representing a terminal window, showing a command prompt starting with '\$ |'. The window has a dark grey header bar with three colored window control buttons (red, yellow, green) on the left and the text '-bash' on the right. The overall background of the slide is purple.

# Supply-Chain Firewall Databases

Datadog Security  
Research's public  
malicious packages  
dataset



# Supply-Chain Firewall Databases

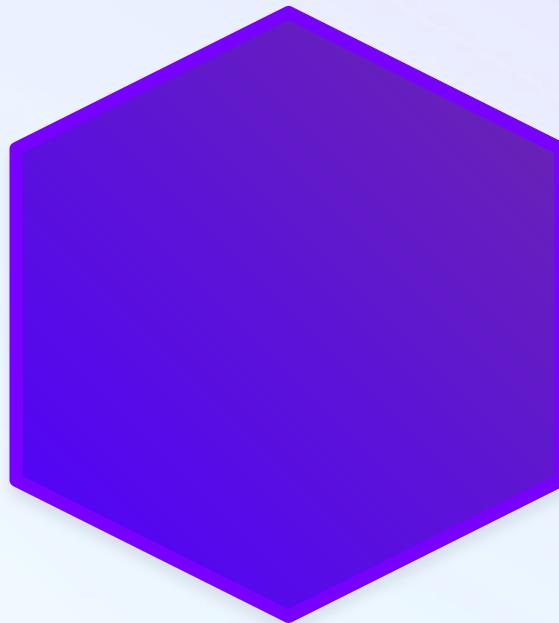
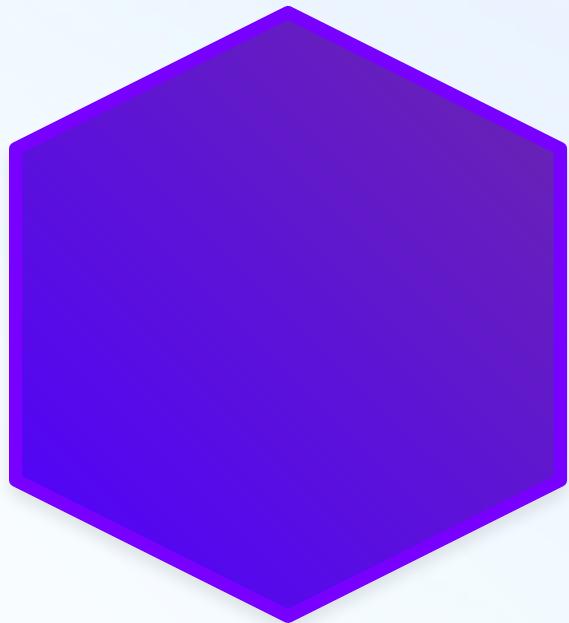
Datadog Security  
Research's public  
malicious packages  
dataset

OSV.dev

# Evading the SCFW

The malicious package has to exist in the database.

# Database Accuracy and Trust



# Database Accuracy and Trust

Datadog Security  
Research's public  
malicious packages  
dataset

# Database Accuracy and Trust

Datadog Security  
Research's public  
malicious packages  
dataset

OSV.dev

# 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!



Use open source  
security tools

# How You Can Help!

Use open source  
security tools

Create a new  
issue in the  
repository

# How You Can Help!

Use open source  
security tools

Create a new  
issue in the  
repository

Contribute to  
the code

# The Bad News

# The Good News

# Datadog Security Newsletter



Security Labs

**Securely integrating with customers' AWS accounts, supply chain security, and investigating a DoS attack**



We've recently released a guide for SaaS providers to **securely integrate with customers' AWS accounts**. It goes through not only the basics, but also hardening and architectural tips, including:

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



# Repositories Discussed

<https://github.com/DataDog/guarddog>

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

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

# Questions?



DATADOG

# Thank you!



DATADOG