DevSecCon 2020

Hardening your soft software supply chain

June 15, 2020

Presented by Maya Kaczorowski, Product Manager, GitHub

 @mayakacz  @MayaKaczorowski

@MayaKaczorowski

Maya Kaczorowski

Product Manager,
Software Supply Chain Security

GitHub

Agenda

What's a software supply chain

Supply chain compromises

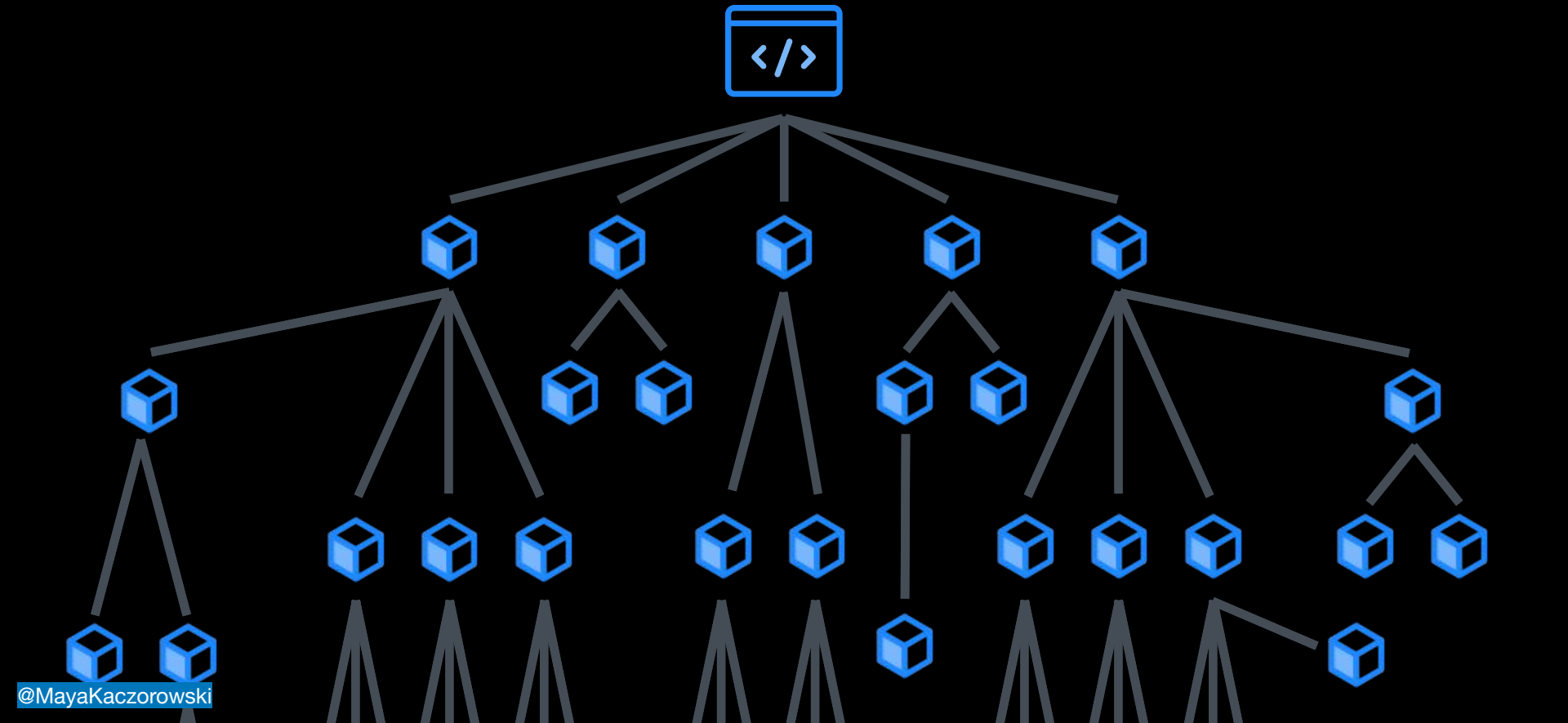
- Kinds of attacks
- Real world examples 😰

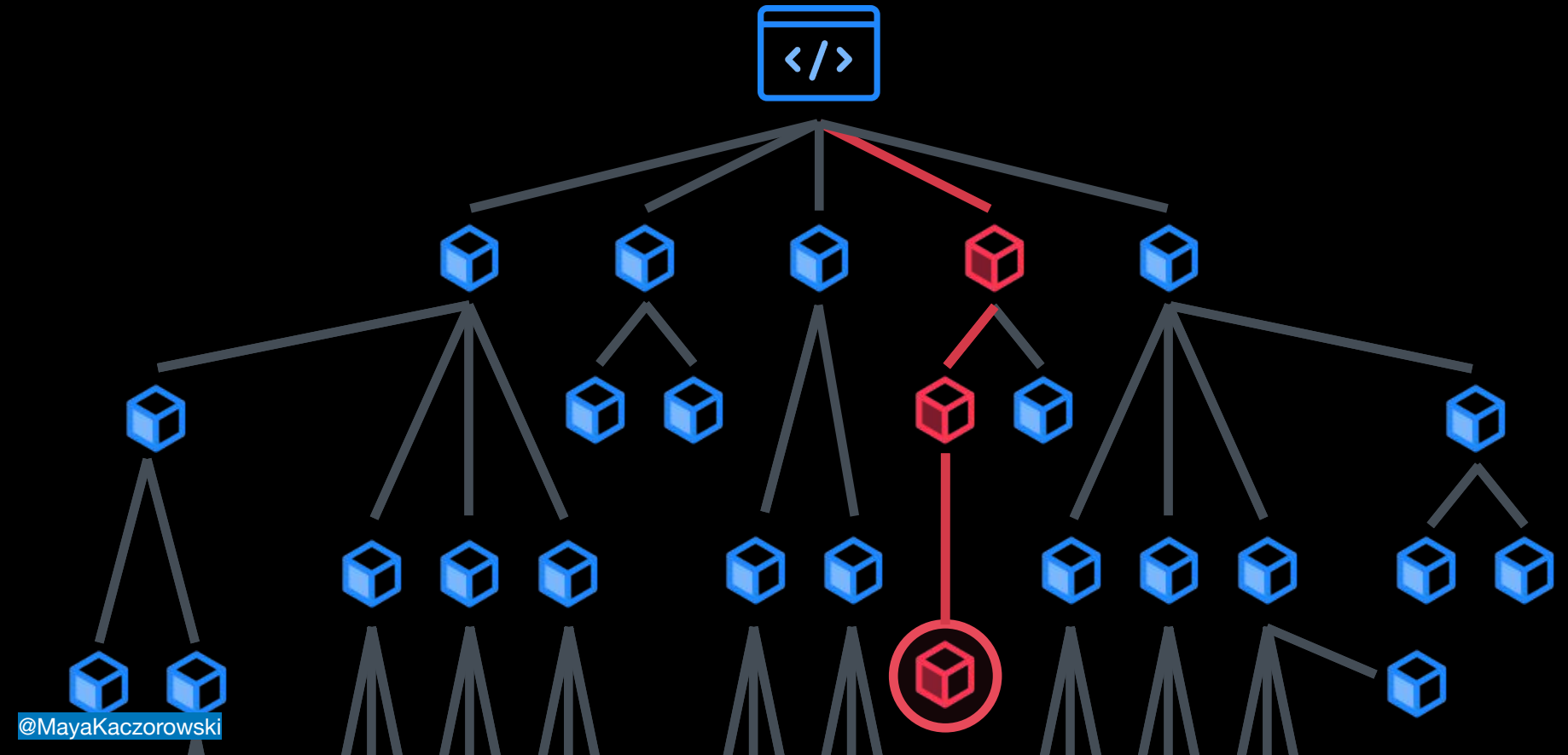
What can you do?

- Developers
- Maintainers
- Security researchers

1 What's a software supply chain?

A software supply chain is anything that touches your code - from development, through your CI/CD pipeline, until it gets deployed into production.


@MayaKaczorowski


@MayaKaczorowski

Every time you `pip install`, `go get`, or `mvn fetch` something, you're doing the equivalent of plugging a thumb drive you found on the sidewalk into your production server.

- Dan Lorenc

https://medium.com/better-programming/getting-serious-about-open-source-security-1d15609478fa

The issue is

unpatched software

To better patch software...

Know what dependencies you use

Know about vulnerabilities in those dependencies

Patch them

… Get back to work!

52% of developers find it “painful” to update vulnerable component releases

2019 State of the Software Supply Chain Report, Sonatype

2 Supply chain compromises

Supply chain attacks

Method

- Malicious code, e.g., backdoor, malware, known vulnerability
- Compromised build tool
- Compromised signing keys
- Compromised package manager
- Compromised vulnerability reporting
- Account takeover
- Project takeover
- Accidental, e.g., typosquatting
- Deletion

Goal

- Backdoor, e.g., targeted, watering hole
- Malware, e.g., cryptocurrency mining
- Service disruption, e.g., deletion

event-stream

Widely used npm library
CVE-2018-1000851

Method: Project takeover
“he emailed me and said he wanted to maintain the module, so I gave it to him. I don't get any thing from maintaining this module, and I don't even use it anymore, and havn't for years.”

Goal: Backdoor
Highly targeted to Copay developers, to distribute malware to capture credentials for Dash Copay bitcoin wallets

- `right9ctrl` volunteered to take over the package on GitHub, then added `flatmap-stream` as a dependency
- Another user `hugeglass` added malware to steal credentials to bitcoin wallets to `flatmap-stream`
- Profit

https://blog.npmjs.org/post/180565383195/details-about-the-event-stream-incident

eslint

npm package for Javascript static code analyzer

Method: Account takeover
Credential stuffing

Goal: Backdoor
To get `.npmrc` npm publishing creds!

- Attacker generated new auth tokens and published two malicious packages
- Second package discovered within an hour; altogether published for <4 hrs
- npm revoked ALL access tokens before the incident
- “A very small number” of packages and users affected

https://eslint.org/blog/2018/07/postmortem-for-malicious-package-publishes
https://gist.github.com/hzoo/51cb84afdc50b14bffa6c6dc49826b3e
https://status.npmjs.org/incidents/dn7c1fgrr7ng

Webmin

Web app with 1M+ installs
CVE-2019-15107

Method: Compromised build tool
“malicious code inserted into Webmin and Usermin at some point on our build infrastructure”

Goal: Backdoor
Unauthenticated RCE

- Backdoor for unauthenticated RCE disclosed as a 0day at Defcon 27
- Unauthenticated requests, or where password expiry policy allowed users with expired passwords to reset them
- Backdoored packages on SourceForge only
- Distributed for more than a year

https://www.virtualmin.com/node/66890
https://www.pentest.com.tr/exploits/DEFCON-Webmin-1920-Unauthenticated-Remote-Command-Execution.html
https://www.zdnet.com/article/backdoor-found-in-webmin-a-popular-web-based-utility-for-managing-unix-servers/

docker123321

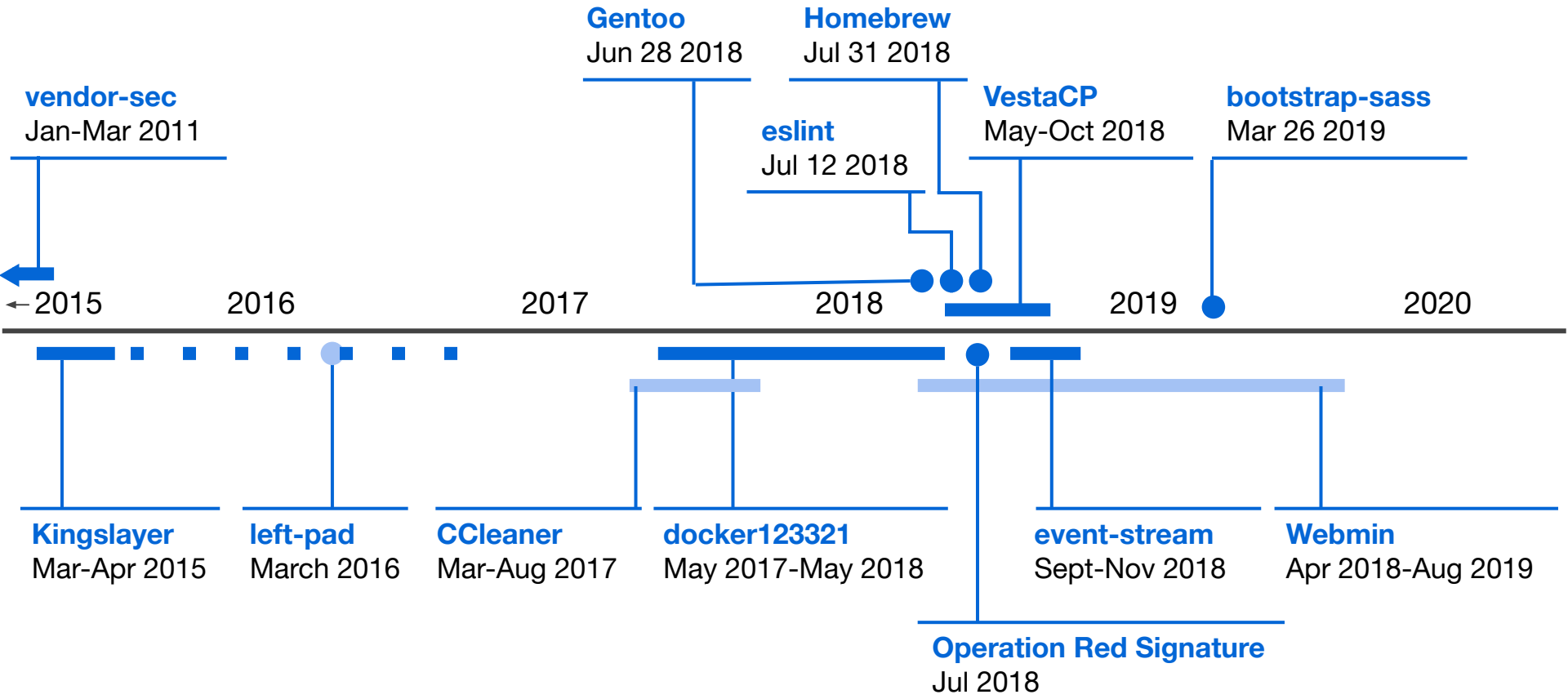
17 Docker Hub images

Method: Accidental
Easy to type registry name

Goal: Malware
Mining ~$90k of Monero

- 17 images in `docker123321` registry
- Contained malware since at least July 2017
- Suspected malware, positively identified as part of a cryptomining botnet
- Removed by Docker Hub in May 2018

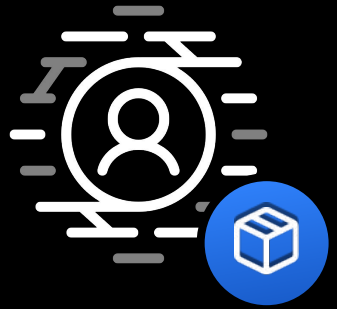
https://kromtech.com/blog/security-center/cryptojacking-invades-cloud-how-modern-containerization-trend-is-exploited-by-attackers



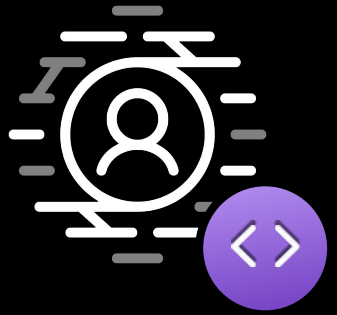
Timeline of select (known) Software Supply Chain attacks

@MayaKaczorowski

3 What can you do?

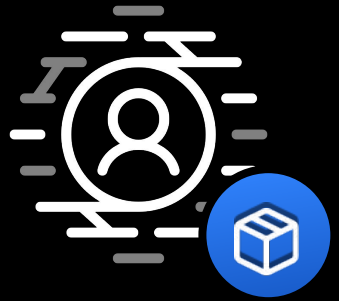


Developers



Maintainers

Security Researchers

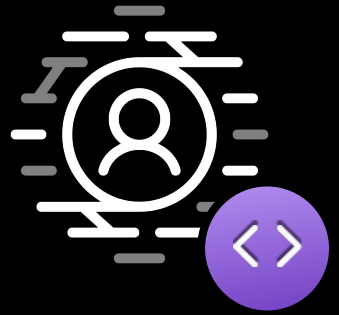


Developers

Know what's in your environment

Manage your dependencies

Monitor your supply chain



Maintainers

Fix and publish vulnerability information

Security Researchers

Discover and report vulnerabilities

How to patch vulnerable dependencies

Know
what's in your environment

- **Discover your dependencies**, including transitive and checked in dependencies
- **Understand your risks**, such as vulnerabilities and licensing restrictions

How to patch vulnerable dependencies

Know
what's in your environment

- **Discover your dependencies**, including transitive and checked in dependencies
- **Understand your risks**, such as vulnerabilities and licensing restrictions

Manage
your dependencies

- **Determine if you are impacted** by a new security issue
- **Update** for the latest functionality and security patches
- **Review changes** to understand and approve new dependencies you're introducing
- **Remove unnecessary dependencies**, to reduce surface of attack

40 days

Mean time to remediate (MTTR)
for repos with Dependabot security updates

180+ days

Mean time to remediate (MTTR)
Industry norm

How to patch vulnerable dependencies

Know
what's in your environment

- **Discover your dependencies**, including transitive and checked in dependencies
- **Understand your risks**, such as vulnerabilities and licensing restrictions

Manage
your dependencies

- **Determine if you are impacted** by a new security issue
- **Update** for the latest functionality and security patches
- **Review changes** to understand and approve new dependencies you're introducing
- **Remove unnecessary dependencies**, to reduce surface of attack

How to patch vulnerable dependencies

Know
what's in your environment

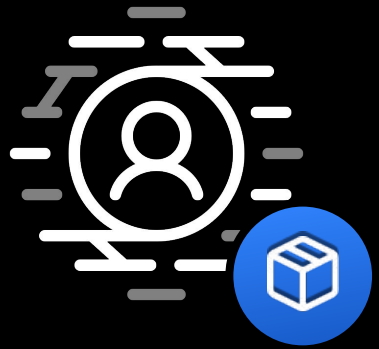
- **Discover your dependencies**, including transitive and checked in dependencies
- **Understand your risks**, such as vulnerabilities and licensing restrictions

Manage
your dependencies

- **Determine if you are impacted** by a new security issue
- **Update** for the latest functionality and security patches
- **Review changes** to understand and approve new dependencies you're introducing
- **Remove unnecessary dependencies**, to reduce surface of attack

Monitor
your supply chain

- **Audit** your current environment for potential risks
- **Enforce policies** to prevent new issues from being introduced



Developers

Know your environment

Dependency Graph

Understand your project's dependencies

Manage your dependencies

Dependabot alerts

Be notified of a vulnerability in a dependency

Dependabot security updates

Review a PR to update to the minimum fixed version

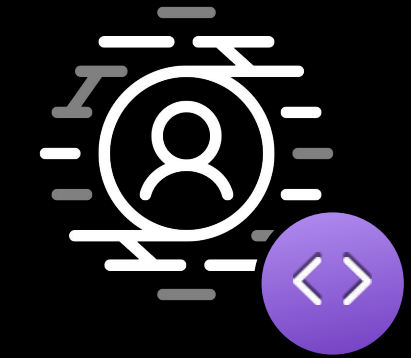
Dependabot version updates

Review a PR to update to the latest stable dependency version

How to address a vulnerability

Fix and publish vulnerability information

- **Respond** to a report of a security issue in your project
- **Develop the fix**, addressing the vulnerability, ideally before it's public
- **Release the fix** and backport to any supported versions
- **Notify your users** that a patch addresses a security vulnerability



Fix and publish vulnerability information

Security Advisories

Fix and publish a notice about a vulnerability

GitHub Advisory Database

Refer to a curated, open-source database of vulnerabilities

SECURITY.md

Share your reporting and disclosure policy

How to report a vulnerability

Discover and report vulnerabilities

- **Report it** to the maintainer 😊

Security Researchers

Discover and report vulnerabilities

CodeQL

Query code as data to find multiple instance of a vulnerability

SECURITY.md

Share your reporting and disclosure policy

Hardening your software supply chain

Developers

Know what's in your environment
- Discover your dependencies
- Understand your risks

Manage your dependencies
- Determine if you are impacted
- Update
- Review changes
- Remove unnecessary dependencies

Monitor your supply chain
- Audit
- Enforce policies

Maintainers

Fix and publish vulnerability information
- Respond
- Develop the fix
- Release the fix
- Notify your users

Security Researchers

Discover and report vulnerabilities
- Please report 😊

Learn more

Supply chain compromises:
https://github.com/cncf/sig-security/tree/master/supply-chain-security/compromises

Getting serious about open source security:
https://medium.com/better-programming/getting-serious-about-open-source-security-1d15609478fa

Enable GitHub security features:

https://github.co/dependency-graph

https://github.co/security-alerts

https://github.co/security-updates