AboutCode

# The future of cybersecurity, today: Free and open source tools for compliance

Philippe Ombredanne,
Lead maintainer of AboutCode

# Agenda

1. Introductions
   - Philippe and AboutCode
   - What you need to know about:
     - CRA
     - SCA

2. Cybersecurity challenges for software supply chains
   - Open source is everywhere
   - Compliance = critical
   - Modern software requires modern cybersecurity
   - Proprietary != scalable and efficient

3. Overview of FOSS tools for cybersecurity and compliance
   - Discovering and identifying third-party code
   - Discovering, triaging, and managing vulnerabilities
   - Standards for tool interoperability
   - Open license, package, and vulnerability databases
   - Automating compliance processes

4. So, what's next?
   - Questions?

AboutCode

# About Philippe and AboutCode

- FOSS-first mission: Make it easier to reuse open source, safely and efficiently, with open source code and open data
  - Creator of Package-URL (PURL), co-founder of SPDX and ClearlyDefined, contributor to CycloneDX, and trusted SCA expert since 2007
    - pombredanne@aboutcode.org
    - https://github.com/pombredanne
    - https://www.linkedin.com/in/philippeombredanne

- Lead maintainer of AboutCode: https://aboutcode.org
  - Open source tools and open knowledge base: ScanCode, VulnerableCode
  - Simple and practical standards: PURL
  - Apps for legal, security, and business users with APIs for everything: DejaCode

# What you need to know about the Cyber Resilience Act (CRA)

# CRA = Cybersecurity for digital products

- Adopted on October 10th, 2024 and applicable once published in the EU’s official journal (12/24/36 months)
  - Declaration of conformity by adding CE marking on products
- Different requirements depending on the economic actors putting the product on the market and the category of the product
  - Economic operators: manufacturers, importers, distributors, open source stewards
    - Manufacturers: full range of obligations
    - Open source stewards: light-touch regulatory regime
  - Category of products
    - Includes open source and other third-party components

# CRA’s essential requirements

- Secure by design
- Secure default configuration
- No known vulnerabilities
- Security updates
- Access control
- Confidentiality and integrity protection
- …

- Vulnerability handling
  - SBOMs
  - Address and publish vulnerabilities
  - CVD policy
- Documentation obligations
  - Risk assessment
  - Processes
  - Intended use

# SCA = Software Composition Analysis

- SCA is essential to know what components are actually in the software
  - Includes processes to identify components, their licensing, and known vulnerabilities (like the AboutCode stack), and evaluate the quality of a software unit (like the CHAOSS project)
    - Read "SCA the FOSS Way": https://www.nexb.com/software-composition-analysis/
  - Critical to comply with mandated Software Bill of Materials (SBOMs) and other regulations
- SCA needs to be a core competency for any software development organization
  - Embed in the software development workflow from design through release -
    - Similar to manufacturing
  - The choice of SCA tools will depend on your platform, stack and product

# The letter "F" in "Compliance" is for Fun

# "The 'SB' in SBOM does not stand for Silver Bullet"

**– Allan Friedman, US Cybersecurity and Infrastructure Security Agency (CISA)**

AboutCode

# Cybersecurity challenges for software supply chains

# Open source is everywhere

- Defined by open source licenses
  - Identifying licenses and license compliance still a problem at scale
- Modern software is composed of mostly open source
  - Common to see a software product or system include 99% open source components
  - Driven by modern software development, easy to have an app that depends on 10,000+ packages
- FOSS compliance is licensing AND security
  - Requirement for everyone organization with regulations and SBOM mandates
  - Very difficult to track all open source and third-party components - including dependencies, licensing, and compliance obligations - with the high volume and rate of change

# Compliance = critical

- Always important, now urgent with CRA and other regulations and more cybersecurity attacks
    - Disproportionate effect on SMEs, nonprofits and other organizations with same compliance needs as big companies and governments but without the resources
        - No dedicated security teams (usually) or budgets for expensive tooling and processes
- Must automate compliance processes (when possible) for efficiency
    - Imperative to balance compliance efforts and shipping products
    - Critical to ensure software supply chain security and integrity

# Modern software requires modern cybersecurity

- Explosion in volume of vulnerabilities and vulnerability data sources
    - Each project provides reference vulnerability data (good), but requires multiple sources to access all the data (bad)
- Biggest threat = false positives and vulnerability fatigue
    - Also challenging to triage and mitigate vulnerabilities at scale
- Fundamental mismatch between legacy DBs and FOSS-driven modern software development
    - Centralized vulnerability databases, keyed by assigned CVEs + CPE, failing
        - US government-funded NVD is not reliable with CPEs and CVSS no longer assigned

# Proprietary != scalable and effective

- Commercial tools for security are cost-prohibitive and not efficient
    - Increasing expensive with surge of interest in SBOMs and developer-based pricing
        - Gold rush from commercial vendors to sell anything related to CRA, SBOM, compliance, vulnerability, cybersecurity
    - Not efficient for compliance tooling and processes
        - Cost of scan curation is prohibitive with high false positive rates and poor origin and license detection accuracy
- Proprietary data for FOSS is wrong
    - Most current data about FOSS packages and vulnerabilities is proprietary
        - Vendors may offer some free or open source tools but must pay for access to their data
    - Vulnerability and security data about open source must be free and open
        - Security is a fundamental right
        - Safe open source software is a public good

# Overview of FOSS tools for cybersecurity and compliance

# Modern software requires FOSS for FOSS tools and open data

LEGACY

- ❌ Vulnerability-centric
- ❌ Proprietary data
- ❌ Siloed
- ❌ Vendor-driven
- ❌ Centralized
- ❌ Security team
- ❌ Reactive

FUTURE = Open source

- ✅ Package-centric
- ✅ Open data
- ✅ Interoperable
- ✅ Community-driven
- ✅ Decentralized, federated
- ✅ Security team + developers
- ✅ Proactive

# Identify third-party code

1. Scan code
   - Based on package manifests, and other clues present locally in the code
2. Match code
   - Based on content and fuzzy fingerprints matched to an external open knowledge base
   - PURL-based
3. Identify license, copyright, other origin clues
   - Including binary analysis and build tracing

Many tools, but still "unsolved"

- Recent study to compare commercial and FOSS SCA tools for containers was ... sad 😿
  - More on this later
- Email pombredanne@aboutcode.org for the sanitized report

# FOSS tools to identify third-party code

AboutCode

| FOSS Tool | Scanning | Matching | Other origin clues |
|---|---|---|---|
| Google OSV | ✅ | ❌ | ❌ |
| SCANOSS | ❌ | ✅ (source only) | ❌ |
| ORT | ✅ | ✅ | ✅ |
| Syft | ✅ (mostly containers) | ❌ | ❌ |
| Trivy | ✅ (mostly containers) | ❌ | ❌ |
| BANG | ❌ | ❌ | ✅ (including binary) |
| ScanCode | ✅ | ❌ | ✅ (including binary) |
| MatchCode | ❌ | ✅ (including binary) | ❌ |
| Many other tools | ✅ | ❌ | ❌ |

# Triage vulnerabilities

1. Lookup (open) vulnerability databases
2. Rank severity and exploitability
3. PURL-based
4. VEX export

# Package-URL (PURL) enables tool interoperability

- Critical for managing software supply chain security and integrity
- URL string to identify and locate software packages across various ecosystems and repositories, adopted by:
    - All SBOM and VEX standards including CycloneDX, SPDX, CSAF, and OpenVEX
    - All open source SCA and SBOM tools and most proprietary SCA, SBOM, and code host tools
    - Most open vulnerability databases (part of CVE specification v5.1)
    - Recommended by US CISA, German BSi and the CERT-India
- In the process of Ecma standardization: https://tc54.org/purl/
- Read more: https://nexb.com/purl-universal-software-package-identification/

# FOSS tools to triage vulnerabilities

| FOSS Tool | Lookup vulnerability databases | Rank severity and exploitability | PURL-based | VEX export |
|---|---|---|---|---|
| DependencyTrack | ✅ | ✅ | ✅ | ✅ |
| DefectDojo | ✅ | ✅ | ❌ | ❌ |
| DejaCode CRAVEX | ✅ | ✅ | ✅ | ✅ |

**Need more (and better) tools with more capabilities, especially for mitigating and managing vulnerabilities**

# And we don't need more vulnerability databases

- We need just one good open package-based vulnerability database
  - Federated with projects submitting vulnerabilities
  - Keyed by PURL to ensure tool interoperability

**Peter J. Yost (© 2021) "The One Ring made from scratch in Blender 3D software." (CC BY-SA 4.0). https://upload.wikimedia.org/wikipedia/commons/d/d4/One_Ring_Blender_Render.png**

# Open vulnerability databases

| Open vulnerability database | Open source code | Open infrastructure | PURL-based | Updated data | Scope |
|---|---|---|---|---|---|
| US NVD | ❌ | ❌ | ❌ | ❌ (delayed) | System + app package + prop |
| Google OSV | ✅ | ❌ | ✅ (mostly) | ✅ | System + app package |
| GitHub Advisories | ❌ | ❌ | ✅ (compatible) | ✅ | App package |
| GitLab Advisories | ✅ | ❌ | ✅ (mostly) | ❌ (1 month delay) | App package |
| VulnerableCode | ✅ | ✅ | ✅ | ✅ | System + app package |
| Linux distro advisories | ❌ | ❌ | ✅ (compatible) | ❌ | System |
| Ecosystem advisories | ❌ | ❌ | ✅ (compatible) | ❌ | App package |

# Manage compliance

1. Aggregate SBOMs
2. Export VEX and SBOMs
3. PURL-based
4. Dependency updates and remediation

# FOSS tools to manage compliance

| FOSS Tool | Aggregate SBOMs | Export VEX, SBOMs | PURL-based | Dependency updates and remediation |
|---|---|---|---|---|
| AboutCode stack (WIP) | ✅ | ✅ | ✅ | ❌ |
| OCCTET (WIP) | ✅ | ✅ | ✅ | ❌ |
| DependencyTrack | ✅ | ✅ | ✅ | ❌ |
| RenovateBot | ❌ | ❌ | ❌ | ✅ |
| DependendaBot | ❌ | ❌ | ❌ | ✅ |

**Need more (and better) tools with more capabilities, especially for compliance automation**

# FOSS tools still have work to do

- The state of SCA tooling accuracy is not great
  - Recent large scale comparison of both FOSS and commercial container scanners using SBOMs to compare scans of the same container images
    - Commercial tools made up packages and PURLs
    - Several tools created invalid SBOMs
    - Most only looking at package manifests and DB
    - Beyond package origin, quality of report licenses is bad and misleading
    - In most cases, this is a grep on the declared license of package manifests
- We can do better!
  - FOSS tools performed better than commercial
  - Still many functionality missing to complete end-to-end automation of compliance processes

# So, what's next?

# We need your help.

AboutCode

# We are still missing critical parts.

# We need more open tools, with more capabilities.

# We need process guides for CRA compliance.

# We need more open reference data for FOSS.

AboutCode

# We need this to solve license AND security!

# Solve the problem(s) with open source tools and open data

- More work to build a complete end-to-end compliance solution:
  - Compliance of open source projects against the CRA compliance
  - Security by design and by default
- Start small and avoid complexity
  - Waste of resources
- Contribute to open source projects
  - https://github.com/aboutcode-org
  - https://www.osadl.org/Projects.osadl-projects.0.html

- Engage with the community
  - Attend the FOSS compliance tools workshop before FOSDEM 2025: https://workshop.aboutcode.org
  - Join the Open Regulatory Compliance Working Group: https://orcwg.org/
  - AboutCode Slack: https://join.slack.com/t/aboutcode-org/shared_invite/zt-2hjzc448i-SZULSuI0~h6YNSUnBWlAqA

# Part two will cover how to use FOSS tools to automate compliance

# Questions?

**Connect on LinkedIn!**

**Philippe Ombredanne**

**Lead Maintainer, AboutCode**

Project Nayuki (© 2024) QR-Code-generator [Source code] (MIT). https://github.com/nayuki/QR-Code-generator