OWASP and AboutCode Foundation

Collaborative Standardization: How communities built PURL and CycloneDX

with Steve Springett, OWASP
and Philippe Ombredanne, AboutCode

Agenda

What is PURL? A history of why Package-URL and how CycloneDX contributed to the specification

Community-driven standardization: Community collaboration and shared ownership with a bottom-up approach

Why Ecma? Professional and community-based, lightweight and rigorous

Latest developments: Current status and roadmap for Ecma TC54 projects

Calls to action: Encouraging community participation

FOSS-first mission: Make it easier to reuse open source, safely and efficiently, with open source code and open data

Philippe Ombredanne

pombredanne@aboutcode.org
https://github.com/pombredanne
https://www.linkedin.com/in/philippeombredanne

- Lead maintainer of AboutCode
  - Open source code, data, and standards to automate and secure software supply chains with transparency and confidence
  - https://aboutcode.org
- Creator of PURL (Package-URL) and VERS, co-founder of SPDX and ClearlyDefined, and core contributor to CycloneDX
- CTO and co-founder of nexB
  - Providing SCA services and AboutCode support since 2017
  - https://nexb.com

Make secure software a reality through open collaboration, education, and innovation

steve.springett@owasp.org
https://github.com/stevespringett
https://www.linkedin.com/in/stevespringett

Steve Springett

- Chair on the Board of Directors of the OWASP Foundation
    - Lead for OWASP Dependency-Track project and OWASP Software Component Verification Standard (SCVS)
    - https://owasp.org/
- Chair of Ecma International TC54
    - Chair of the OWASP CycloneDX Core Working Group
    - https://tc54.org/
- Creator of dependency-track and CycloneDX

What is PURL?

PURL is the standard for package identification across ecosystems

- Package-URL (PURL) is the glue between all your software supply chain tools, data, and standards
  - Adopted in all SBOM and VEX specs
  - Most SCA tools
  - Many vulnerability databases
- Across Dev to Ops to Sec (to DevSecOps)

- Simple, obvious, expressive syntax:
  - **`pkg:npm/file@1.9.1`**
  - **`pkg:deb/ubuntu/7zip@21.07+dfsg-4`**
  - **`pkg:pypi/django@1.11.1`**

- Ecma standard at TC54
  - Planned ISO JTC1 standardization
  - Already included in CycloneDX, OASIS, ISO CSAF, and CVE schema

How to identify a package across tools and ecosystems in 2017?

1. Scan a package manifest
2. Search CVE through NVD
3. Review and finnick, search more
4. Or try to map scanned package to CPE?

- Overall painful process
  - `file-utils` package exists in npm, PyPI, and Rubygems, the name is not unique!

- Solution: Craft a small identifier to tie scanned package to vulnerability database
  - Call this Package-URL, because it's a URL for a package
- Insight: Each ecosystem provides the invisible hand needed for sanity 🙌 ensuring unique names and versions
- Ecosystem == package type
  - And create a new aggregated DB keyed by this thing

PURL history 📚

Extracted the PURL spec, because it was useful

- We needed to identify the same package across ScanCode and VulnerableCode
- Everybody was facing same (or similar) problems

- Extracted a minimal spec and invited comments and participation…
    - A simple, clear spec is hard
- Extracting is more work, yet …
    - Sharing, this is the way
- Key to success is sharing control

PURL today

PURL adopted industry- and community-wide

All open source SCA and SBOM tools use PURL, including:

- Linux Foundation's OSS Review Toolkit, and Fossology
- OpenSSF OSV and GUAC
- OWASP Dependency-Track, Dependency-Check, and cdxgen
- All OWASP CycloneDX libraries
- GitHub's Dependency Graph
- Microsoft's OSS Gadget, SBOM tool
- Anchore's Syft and Grype
- Aquasec's Trivy
- LG's FOSSLight
- SCANOSS
- Snyk's Parlay
- and many more!

Most proprietary SCA, SBOM, and code host tools use PURL, including:

- GitHub
- GitLab
- Snyk
- Mend
- BlackDuck
- Sonatype

Vulnerability databases use PURL, including:

- Google's OSV
- Sonatype's OSS Index
- CVE specification v5.1
- VulnerableCode

PURL facilitates better compliance processes for end users, including:

- Most free and open source software foundations
- Five of the Big Tech companies, with three building their entire SCA compliance operations on PURL
- A leading database company
- A leading Linux company
- European and US government agencies
- All major European car manufacturers and most of their vendors
- Major US chip and microprocessor providers
- Four leading European industrial companies
- A leading European medical devices company

PURL today

Actionable data, keyed by PURL, metadata and vulnerabilities

ClearlyDefined

ClearlyDefined supports converting PURLs to " Coordinates" format in its data model. https://docs.clearlydefined.io/docs/resources/coordinates#purl-type-mapping

AboutCode PurlDB

Continuously updated reference data for open source packages' origin, information and licensing, utilizing PURL. https://github.com/aboutcode-org/purldb

deps.dev

PURLs serve as unique identifiers for allows for precise tracking and analysis of dependencies within the deps.dev system. https://docs.deps.dev/api/v3alpha/#purllookup

Sonatype OSS Index

OSS Index can consistently map components to their corresponding vulnerability data using PURL. https://ossindex.sonatype.org/doc/coordinates

ecosyste.ms

Lookup information about a PURL in both text and JSON formats. https://github.com/andrew/purl

OSV

OSV uses PURLs to identify packages within its database and API queries. https://ossf.github.io/osv-schema/#affectedpackage-field

Beside PURLs, what else?

- CPE
- Checksums, hashes
- OmniBOR: Checksum-based
- SWHID: Checksum-based
- PURL look-alike with an ecosystem, name and version (in latest CVE and also in OSV)
- Plain URLs for download

- All of these can be useful
- Checksums actually help point to precise files
  - Like a GPS vs. street address
  - Not human readable

PURL history 📚

This adoption meant PURL needed to become standardized

- Ecma to the rescue!
  - Effort supported by an investment of the German Sovereign Tech Agency
  - And a large US-based financial organization

- Standardization necessary to clean, clarify, remove ambiguities
  - Focus on spec essentials
    - Figure out other details like PURL types later

Standards need to be:

1. Professional and community-based, and
2. Lightweight and rigorous

CycloneDX history

- Once upon a time, **`dependency-track`**
  - Needed to collect inventories of hardware and software
  - Needed to import inventories and lookup vulnerabilities in one place
  - And decouple identity from the analysis of software then promoted by SCA tools

- CycloneDX developed to solve that problem
  - Originally a simple list of packages
    - 1st serious PURL adopter
  - Growing community of CycloneDX adopters and contributors
  - Needed standardization for consistent usage of the spec

Community-driven standardization

Bottom-up enables community collaboration and shared ownership

TOP-DOWN **vs**

1. Create a committee
2. Work for a few years to design standard
3. Create reference implementation
4. Promote adoption
    - By the time the standardization process is finished, industry has moved on

* This could be the better approach for hardware but not for software

BOTTOM-UP

1. Identify a problem in your tool
2. Build a solution to that specific problem
3. Draft a small spec
4. Promote and share with others to review and improve
5. Grow into standard
    - By the time the standardization process is finished, standard already adopted industry-wide

Professional and community-based, lightweight and rigorous

Why Ecma?

- 60+ years of developing and publishing 400+ standards
    - Hardware, software, programming languages, IoT, other domains
        - ECMAScript®

- Open collaboration and streamlined processes, based on open source principles
    - Flexible governance framework allows for **royalty-free patent policy**.
- Works with ISO/IEC JTC 1 and others ensure Ecma standards recognized worldwide

Professional and community-based, lightweight and rigorous

Ecma standardization with TC54 (software and system transparency)

- Covers PURL and VERS (standard for version ranges)
  - https://github.com/Ecma-TC54/tg2
  - PURL and VERS are already part of EcmaISO standards, indirectly with CSAF 2.0 and SPDX 2
- Active participation by key community contributors, industry players and open source foundations

- TC54 started for CycloneDX
  - CycloneDX is now **ECMA-424**
- PURL and VERS in CycloneDX specs
  - PURL is now **ECMA-427**
- Includes TEA (Transparency Exchange API) and CLE (Common Lifecycle Events) task groups
  - CLE is now **ECMA-428**
- Other TG for Contributing.yaml

Professional and community-based, lightweight and rigorous

TC54 working model can expand, and already has!

- Based on Ecma TC39 - Specifying JavaScript
  - Technically, EcmaScript is the standard and JavaScript the implementation
- Built around real-world use cases
  - Specification evolves to solve practical and achievable outcomes

- Community-based standardization with rigor necessary for international standards:
  1. Community
  2. Ecma
  3. ISO fast track
- Created TC54 for software system transparency
  - TGs for specific projects
  - Model can be reused by other TCs

Latest developments

What's next for Ecma TC54?

- CycloneDX 2.0
  - cyclonedx.org
- OSS Sustainability (contributing.yaml)
  - tc54.org/contributing-yaml
- Version Range Specification (VERS) 1.0
  - tc54.org/vers
- Transparency Exchange API (TEA) 1.0
  - c54.org/tea

- ISO JTC1 = fast track for IT standards
  - CycloneDX
  - PURL
- Updates to Common Lifecycle Enumeration (CLE)
  - tc54.org/cle

Latest developments

What's next for the PURL community

- More open source tools and open data to validate PURLs
- Validate package source vs. binaries?
- AboutCode + ClearlyDefined
  - Share all the scans, all the SBOMs
- AboutCode + Software Heritage
  - Scan billions of files
  - Get billions of PURLs

✅ PURL adopted in the CVE schema

✅ PURL for all Rust crates in crates.io

✅ PURL for all JARs at Maven Central

- Adoption in language ecosystems
  - Perl, PEP 725 in Python, Raku
- Commando beach operations cleaning packages for Rust, Maven, and nixpkgs

-

Latest developments

PURL (and VERS) need you!

- PURL types for your ecosystems!
- VERS standardization at Ecma, then ISO
  - Already included in CycloneDX
  - Already included in OASIS and ISO CSAF
- Help all CVEs contain (correct) PURLs
- PURL+VERS-based general purpose dependency resolution

- C/C++ PURL open registry with C++ community
  - And beyond C/C++
  - The missing registry for packages that do not have a registry
- PURL-keyed standard API for every ecosystems
  - We must stop rewriting API parsers and remapping package metadata schemas everywhere

Latest developments

Community support to help us do good!

- German Sovereign Tech Agency (STF) to invest in PURL
- Grants from NLnet Foundation with the EU NGI programs all about PURL:
  - CRAVEX, FederatedCode, T-Rust, back2source, Maven Heaven
- Grants from EU NGI Search:
  - AI-Generated Code Search, matching PURLs
- Completed GitHub Secure OSS Fund

- EU and ECCC (European Cybersecurity Competence Centre)-funded OCCTET project: occtet.eu
  - with Eclipse Foundation, DoubleOpen, Bitsea, European Digital SME Alliance, Expertware, Red Alert Labs
- Received ZEISS FOSS Award
- Grants from from two US big tech, two large US and several EU companies
- Help us do more!

Together, we can build better software compliance processes

Join the community and contribute!

AboutCode

GitHub: github.com/aboutcode-org

Slack: join.slack.com/t/aboutcode-org/shared_invite/zt-1paqwxccw-IuafuiAvYJFkTqGaZsC1og

ClearlyDefined

docs.clearlydefined.io/docs/get-involved/

TC 54 PURL + VERS

github.com/package-url/purl-spec

Join #purl channel on AboutCode slack for latest updates

Community calls:
Biweekly on Wednesdays at 16:00 UTC at
meet.google.com/ryq-aimn-ghd

More about TC54: tc54.org

CycloneDX

cyclonedx.org/participate/contribute/

dependency track

dependencytrack.org

TEA

OWASP TRANSPARENCY EXCHANGE API

tc54.org/tea

CLE

COMMON LIFECYCLE ENUMERATION

OWASP

tc54.org/cle

Come talk to us at the conference!

Any questions?

pombredanne@aboutcode.org
https://github.com/pombredanne
https://www.linkedin.com/in/philippeombredanne

steve.springett@owasp.org
https://github.com/stevespringett
https://www.linkedin.com/in/stevespringett