



DEPENDENCY-TRACK COMMUNITY MEETING

SEPTEMBER
2024

THE DEPENDENCY-TRACK TEAM

www.owasp.org

AGENDA

Project Updates

01

Demo: v4.12.0

02

A Vulnerability DB for DT

03

Q&A

04



PROJECT UPDATES



PROJECT UPDATES

v4.11.6, v4.11.7 Released 🎉

- v4.11.6: August 10th
 - Handle breaking change in Trivy v0.54.0 server API
 - Fix validation error when XML BOM declares multiple namespaces
 - Fix VEX export returning invalid CycloneDX
 - ... and more fixes backported from v4.12.0
- v4.11.7: August 14th
 - Fix missing fields in `/api/v1/project/{uuid}` response



<https://docs.dependencytrack.org/changelog/>

PROJECT UPDATES

v4.12.0 – Recall

- Java 21 baseline
- Swagger 2 → OpenAPI v3
- Java EE → Jakarta EE 10
- Default to BOM processing V2
- Tag management

Tag Management Demo



<https://www.youtube.com/watch?v=rvigQKVvoN8&t=752s>

PROJECT UPDATES

v4.12.0 - New

- Test button for notifications **DEMO**
- Restrict notifications to projects with specific tags **DEMO**
- Use tags to exclude or include projects from BOM validation **DEMO**
- Switch Trivy integration from JSON to Protobuf
- Support component .authors field of CycloneDX v1.6
- Better performance of findings API




DEMO: v4.12.0



PROJECT UPDATES

v4.12.0 – Ok, but... when?

 DependencyTrack / dependency-track


🔍

+

🕒

🔗

📁



<> Code

🕒 **Issues** 708

🔗 Pull requests 32

💬 Discussions

🎬 Actions

📁 Projects 1

⚙️ Settings

⋮

🏷️ Labels

🎯 **Milestones**

New milestone

🎯 4 Open ✓ 40 Closed

Sort ▼

4.12

No due date 🕒 Last updated 3 minutes ago

97% complete 4 open 195 closed

[Edit](#) [Close](#) [Delete](#)

PROJECT UPDATES

hyades v0.6.0

- “Deconstruction” of permissions to be more fine-grained
 - PORTFOLIO_MANAGEMENT → PORTFOLIO_MANAGEMENT_UPDATE etc.
- Improved performance of various REST endpoints (ongoing effort)
 - More efficient SQL queries, optimized data types in PostgreSQL (i.e. UUID, JSONB)
- Option to execute database migrations and seeding in separate container
 - Good fit for k8s init containers or Jobs (Helm chart does this)
 - Prevents k8s from killing Pods for long-running migrations

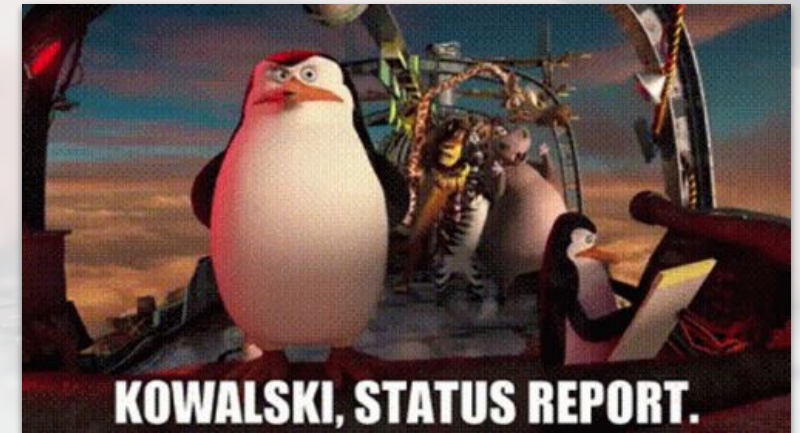


TOWARDS A CENTRALIZED VULNERABILITY DATABASE

TOWARDS A CENTRALIZED VULNERABILITY DATABASE

Status Quo

- We support only a subset of all available public databases
 - Supported: NVD, GHSA, OSV
 - Available: Amazon Linux, Oracle Linux, Red Hat, SUSE, ...
- Every DT instance downloads and processes multiple databases
 - Only NVD enabled per default, GHSA requires authentication
 - Causes unnecessary load on public infrastructure
- NVD still doesn't support Package URL
 - Out-of-the-box experience of DT mostly relies on Sonatype OSS Index
- Resolving vulnerability aliases is a challenge
- Making data corrections is a **big** challenge



TOWARDS A CENTRALIZED VULNERABILITY DATABASE

Goals



- Build a pre-compiled database specifically for DT's needs
 - Built centrally, consumed by all DT instances
 - Sourced from many public databases
- Package URL and vers as first-class citizens
 - `pkg:maven/org.dependencytrack/dependencytrack@4.12.0`
 - `vers:maven/>3|<4.13.0`
- Support software **and hardware** components
 - Must still support CPE (P^□°)^ ^ — —
 - Potentially *GS1 Global Trade Item Number, Global Model Number*

TOWARDS A CENTRALIZED VULNERABILITY DATABASE

Goals (cont.)

- Updated *at least* daily
- Potentially support auxiliary matching information
 - Imports, symbols, etc. for reachability analysis
 - Go's vulnerability database provides this data
 - Tools like cdxgen can produce reachability information
- Support data correction through community contributions
- Cheap or free to operate, preferably file-based
- Optimized for continuous consumption
 - Avoid having to download 10s of GB of unchanged data



TOWARDS A CENTRALIZED VULNERABILITY DATABASE

Non-Goals

- We're not competing with any existing vulnerability database
- We're not trying to cater to tools other than DT
- We're not trying to get rid of OSS Index, Snyk, or VulnDB integrations
 - Commercial sources still provide value



TOWARDS A CENTRALIZED VULNERABILITY DATABASE

Prior Art

- Anchore gype-db: <https://github.com/anchore/gype-db>
- AquaSecurity trivy-db: <https://github.com/aquasecurity/trivy-db>
- AppThreat vulnerability-db: <https://github.com/AppThreat/vulnerability-db>
- All of the above are permissively licensed (Apache-2.0 or MIT)
- We don't necessarily have to start from scratch

TOWARDS A CENTRALIZED VULNERABILITY DATABASE

We Need You

- Doesn't require in-depth knowledge of how DT works
- Free choice of technology, not bound to existing stack
- High visibility: makes a positive impact for **all** users of DT



<https://github.com/DependencyTrack/dependency-track/issues/4122>





QUESTIONS & ANSWERS

