





# Security Training for ●●●●●●

## Developers

Part 2: Section 5: Software Bill of Materials (SBOM)



# Software Bill of Materials

- **What is an SBOM?**
  - **Why we need them?**
    - **How to generate an SBOM for your project?**
  - **SCA Platforms**
- **Early Adopters at EF**



# What is an SBOM?

# SBOM



Inventory of

1. **software components** and **dependencies**
2. component **metadata**
3. and their **hierarchical relationships**

of a **software product**.



## Chocolate Chip Cookies

### Ingredients:

- Flour (all-purpose/ 2 cups)
- Sugar (granulated/ 1 cup)
- Brown sugar (regular/ ½ cup)
- Butter (unsalted/ 1 stick)
- Eggs (large/ 2)
- Vanilla extract (pure/ 1tsp)
- Baking soda (regular/ 1tsp)
- Salt (regular/ ½ tsp)
- Chocolate Chips (loads/ 1 cup)



meal = software product

## Chocolate Chip Cookies Task Management App

### Ingredients:

- Flour (all-purpose/ 2 cups)
- Sugar (granulated/ 1 cup)
- Brown sugar (regular/ ½ cup)
- Butter (unsalted/ 1 stick)
- Eggs (large/ 2)
- Vanilla extract (pure/ 1tsp)
- Baking soda (regular/ 1tsp)
- Salt (regular/ ½ tsp)
- Chocolate Chips (loads/ 1 cup)



## Chocolate Chip Cookies

### Ingredients



- Flour (all-purpose/ 2 cups)
- Sugar (granulated/ 1 cup)
- Brown sugar (regular/ ½ cup)
- Butter (unsalted/ 1 stick)
- Eggs (large/ 2)
- Vanilla extract (pure/ 1tsp)
- Baking soda (regular/ 1tsp)
- Salt (regular/ ½ tsp)
- Chocolate Chips (loads/ 1 cup)

## Task Management App

### Software Components

- Front-end framework
- UI library
- Utility library
- State management
- Authentication
- Logging
- Bundler
- Security middleware
- Third-party plugins



## Chocolate Chip Cookies

## Task Management App

quantities, types = component metadata

### Ingredients

- Flour (**all-purpose/ 2 cups**)
- Sugar (**granulated/ 1 cup**)
- Brown sugar (**regular/ ½ cup**)
- Butter (**unsalted/ 1 stick**)
- Eggs (**large/ 2**)
- Vanilla extract (**pure/ 1tsp**)
- Baking soda (**regular/ 1tsp**)
- Salt (**regular/ ½ tsp**)
- Chocolate Chips (**loads/ 1 cup**)

### Components

- Front-end framework: **React (18.2.0/MIT)**
- UI library: **Material-UI (5.4.4/MIT)**
- Utility library: **Lodash (4.17.21/MIT)**
- State management: **Redux (4.2.0/MIT)**
- Authentication: **OAuth2 (3.1.0/Apache-2.0)**
- Logging: **Winston (3.8.2/MIT)**
- Bundler: **Webpack (3.8.2/MIT)**
- Security middleware: **Helmet.js (6.0.0/MIT)**
- Third-party plugins: **calendar (various)**





## Chocolate Chip Cookies

### Steps

1. For **dough** mix **flour**, **eggs** and **butter** together
2. For **filling** mix **chocolate chips**, **sugar** and **vanilla** extract



## Chocolate Chip Cookies

### Steps

1. For **dough** mix **flour**, **eggs** and **butter** together
2. For **filling** mix **chocolate chips**, **sugar** and **vanilla** extract

## Task Management App

### Transitive dependencies

- The **front-end framework** depends on **react**, **webpack**, **redux**
- The **database** relies on **openssl**, **zlib**, **libcurl**, **protobuf**



## Chocolate Chip Cookies

### Steps

1. For **dough** mix **flour**, **eggs** and **butter** together
2. For **filling** mix **chocolate chips**, **sugar** and **vanilla** extract



steps to mix ingredient

## Task Management App

### Transitive dependencies

- The **front-end framework** depends on **react**, **webpack**, **redux**
- The **database** relies on **openssl**, **zlib**, **libcurl**, **protobuf**



hierarchical relationships

# RECIPE

## Chocolate Chip Cookies

### Ingredients

- Flour (all-purpose/ 2 cups)
- Sugar (granulated/ 1 cup)
- Brown sugar (regular/ ½ cup)
- Butter (unsalted/ 1 stick)
- Eggs (large/ 2)
- Vanilla extract (pure/ 1tsp)
- Baking soda (regular/ 1tsp)
- Salt (regular/ ½ tsp)
- Chocolate Chips (loads/ 1 cup)

### Steps

1. The **dough** is made by mixing flour, eggs and butter together
2. The **filling** is made by mixing chocolate chips, sugar and vanilla extract

## Task Management App

### Components

- Front-end framework: React (18.2.0/MIT)
- UI library: Material-UI (5.4.4/MIT)
- Utility library: Lodash (4.17.21/MIT)
- State management: Redux (4.2.0/MIT)
- Authentication: OAuth2 (3.1.0/Apache-2.0)
- Logging: Winston (3.8.2/MIT)
- Bundler: Webpack (3.8.2/MIT)
- Security middleware: Helmet.js (6.0.0/MIT)
- Third-party plugins: calendar (various)

### Transitive dependencies

- The **front-end framework** depends on react, webpack, redux
- The **database** relies on openssl, zlib, libcurl, protobuf

## RECIPE

### Chocolate Chip Cookies

#### Ingredients

- Flour (all-purpose/ 2 cups)
- Sugar (granulated/ 1 cup)
- Brown sugar (regular/ ½ cup)
- Butter (unsalted/ 1 stick)
- Eggs (large/ 2)
- Vanilla extract (pure/ 1tsp)
- Baking soda (regular/ 1tsp)
- Salt (regular/ ½ tsp)
- Chocolate Chips (loads/ 1 cup)

#### Steps

1. The **dough** is made by mixing flour, eggs and butter together
2. The **filling** is made by mixing chocolate chips, sugar and vanilla extract

## SBOM

### Task Management App

#### Components

- Front-end framework: React (18.2.0/MIT)
- UI library: Material-UI (5.4.4/MIT)
- Utility library: Lodash (4.17.21/MIT)
- State management: Redux (4.2.0/MIT)
- Authentication: OAuth2 (3.1.0/Apache-2.0)
- Logging: Winston (3.8.2/MIT)
- Bundler: Webpack (3.8.2/MIT)
- Security middleware: Helmet.js (6.0.0/MIT)
- Third-party plugins: calendar (various)

#### Transitive dependencies

- The **front-end framework** depends on react, webpack, redux
- The **database** relies on openssl, zlib, libcurl, protobuf

# SBOM



Inventory of **software components** and **dependencies** component **metadata** and their **hierarchical relationships** of a **software product**.

- Unique identifiers for dependencies
- Formats: **SPDX, CycloneDX**
- File formats: **JSON, XML, YAML**
- Generation depends on **ecosystem**, lots of tools available
- Optional phase in SDLC - **build** time
  - Accuracy: Captures exact dependencies used during the build
  - Integrates seamlessly in CI/CD pipelines



# Why we need them?

# Software Supply Chain Security



- Software supply chains are major attack surface
- Modern apps use hundreds of third-party components
- Vulnerabilities often hide in dependencies
- The compromise of a dependency can lead to the compromise of the software that is using it



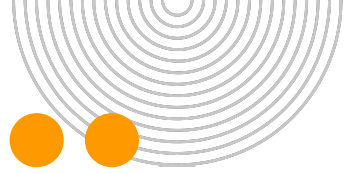
# Compliance



- Tool for user self-evaluation
  - Know what's inside the software that you are using
- U.S. Executive Order 14028
  - Requires SBOMs for software sold to US federal agencies
  - Enforced via NIST and other federal procurement rules
- EU Cyber Resilience Act (CRA)
  - Applies to software/hardware marketed in the EU
  - Vendors must provide SBOMs and vulnerability handling processes



# How to generate an SBOM for your project?



Ask yourself these questions:



Ask yourself these questions:

1. **How many products are in my project?**  
Example: frontend, backend, multiple packages...
2. **How can I integrate it into existing CI/CD pipelines?**  
Example: Github Actions workflows, ...
3. **What is the release process?**  
Example: Github Releases, git tags, ...
4. **Which tool to use?**  
Depends on ecosystem: maven, gradle, python, npm, yarn...  
Tooling options at: <https://eclipse-csi.github.io/security-handbook/sbom/tooling.html>
5. **Where do we want to publish it?**  
Example: artifacts, DependencyTrack instance



Ask yourself these questions:

1. **How many products are in my project?**  
Example: frontend, backend, multiple packages...
2. **How can I integrate it into existing CI/CD pipelines?**  
Example: Github Actions workflows, ...
3. **What is the release process?**  
Example: Github Releases, git tags, ...
4. **Which tool to use?**  
Depends on ecosystem: maven, gradle, python, npm, yarn...  
Tooling options at: <https://eclipse-csi.github.io/security-handbook/sbom/tooling.html>
5. **Where do we want to publish it?**  
Example: artifacts, **DependencyTrack instance** -> **What is that?**



# Software Composition Analysis (SCA) Tools

# Why we need them?



- SBOMs are ... really lengthy files ...

```
1  {
2    "bomFormat": "CycloneDX",
3    "specVersion": "1.6",
4    "serialNumber": "urn:uuid:2af282f8-91f6-4ee
5    "version": 1,
6  >  "metadata": { ...
68    },
69  >  "components": [ ...
270011    ],
270012    "services": [],
270013  >  "dependencies": [ ...
283978    ],
283979  >  "annotations": [ ...
284004    ]
284005 }
```

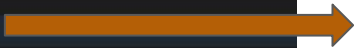


difficult to analyse

# Why we need them?

- Components overview

```
1  {
2    "bomFormat": "CycloneDX",
3    "specVersion": "1.6",
4    "serialNumber": "urn:uuid:2af282f8-91f6-4ee
5    "version": 1,
6  > "metadata": { ...
68  },
69  > "components": [ ...
270011 ],
270012 "services": [],
270013 > "dependencies": [ ...
283978 ],
283979 > "annotations": [ ...
284004 ]
284005 }
```



<input type="checkbox"/>	Component		Version		Group	Internal	License	Risk Score
<input type="checkbox"/>	flask		3.1.0					0
<input type="checkbox"/>	graphql-core		3.2.6					0
<input type="checkbox"/>	hyperframe		6.1.0					0
<input type="checkbox"/>	idna		3.10					0
<input type="checkbox"/>	referencing		0.36.2					0
<input type="checkbox"/>	jsonbender		0.9.3					0
<input type="checkbox"/>	gitpython		3.1.44					0
<input type="checkbox"/>	rich		13.9.4					0
<input type="checkbox"/>	werkzeug		3.1.3					0
<input type="checkbox"/>	certifi		2025.1.31					0

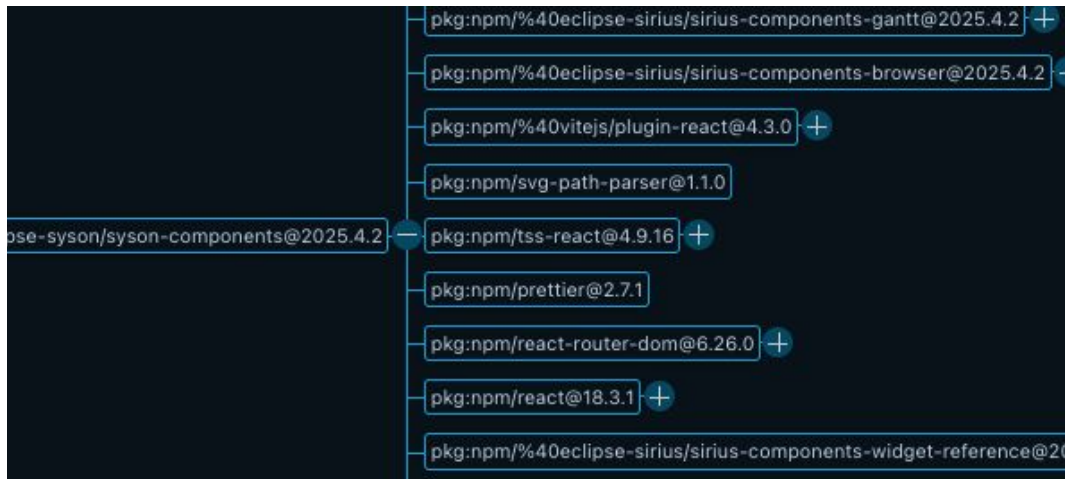
Showing 1 to 10 of 87 rows   10 rows per page   < 1 2



# Why we need them?

- Dependency tree visualization

```
1  {
2    "bomFormat": "CycloneDX",
3    "specVersion": "1.6",
4    "serialNumber": "urn:uuid:2af282f8-91f6-4ee
5    "version": 1,
6  > "metadata": { ...
68  },
69  > "components": [ ...
270011 ],
270012 "services": [],
270013 > "dependencies": [ ...
283978 ],
283979 > "annotations": [ ...
284004 ]
284005 }
```



# Why we need them?



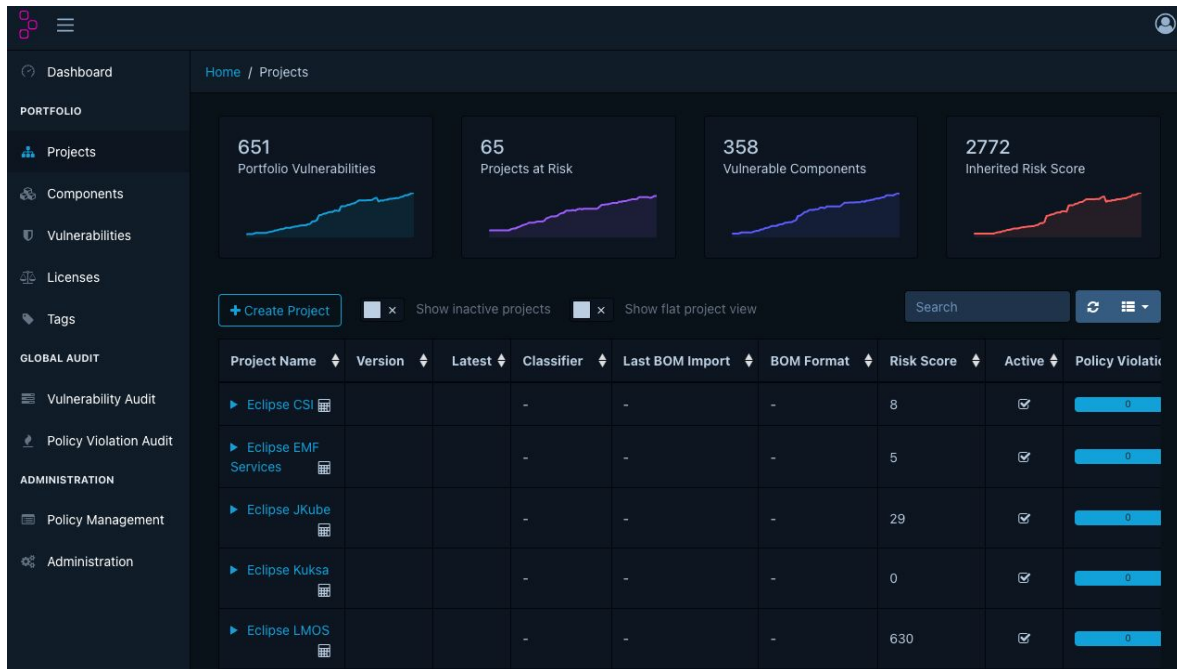
- Vulnerability data enrichment
- Alerts on vulnerable dependencies

	Component ⚡	Version ⚡	Group ⚡	Vulnerability ⚡	Severity ⚡	Analyzer ⚡	Attributed On ⚡
>	esbuild 	0.20.2 		 GHSA-67mh-4wv8-2f99	 Medium	<a href="#">GITHUB ↗</a>	13 May 2025
>	vitest 	1.6.0 		 GHSA-9crc-q9x8-hgqq	 Critical	<a href="#">GITHUB ↗</a>	13 May 2025
>	vite 	5.2.11 		 GHSA-64vr-g452-qvp3	 Medium	<a href="#">GITHUB ↗</a>	13 May 2025

# EF DependencyTrack instance



- Link: <https://sbom.eclipse.org>
- Login with EF account credentials



# Upload integration



- Append this job at the end of your SBOM generation workflow:

```
store-sbom-data:
  needs: ['generate-sbom']
  uses: eclipse-csi/workflows/.github/workflows/store-sbom-data.yml@ma
  with:
    projectName: '<product_name>' # display name
    projectVersion: '${{ needs.generate-sbom.outputs.project-version }}'
    bomArtifact: '<artifact_name>' # name from upload in generate-sbom
    bomFilename: 'bom.json'
    parentProject: '<parentProject_ID>' # provisioned by us
```

- More about it at:  
<https://eclipse-csi.github.io/security-handbook/sbom/howto.html#how-to-upload-an-sbom-to-dependencytrack>



# **SBOMs Early Adopters at Eclipse Foundation**

# SBOM Early Adopters



- **Early Adopters:** *Implementing Software Bill of Materials (SBOM) for projects*
  - Support in equipping projects with:
    - Github Actions workflows to generate SBOMs for new releases
    - Upload them to our DependencyTrack instance
- **How we do that?**
  - Fork one of your repositories
  - Design and implement an SBOM workflow
  - Integrate it into existing release process
  - PR for your review

# Early Adopters Projects at EF



PROJECT	ECOSYSTEM	WORKFLOW
Eclipse CSI	python	<a href="https://github.com/eclipse-csi/otterdog/blob/main/.github/workflows/generate-sbom.yml">https://github.com/eclipse-csi/otterdog/blob/main/.github/workflows/generate-sbom.yml</a>
Eclipse SysON	nodejs, maven	<a href="https://github.com/eclipse-syson/syson/blob/main/.github/workflows/generate-npm-sbom.yml">https://github.com/eclipse-syson/syson/blob/main/.github/workflows/generate-npm-sbom.yml</a> <a href="https://github.com/eclipse-syson/syson/blob/main/.github/workflows/generate-maven-sbom.yml">https://github.com/eclipse-syson/syson/blob/main/.github/workflows/generate-maven-sbom.yml</a>
Eclipse Kuksa	python	<a href="https://github.com/eclipse-kuksa/kuksa-python-sdk/blob/main/.github/workflows/generate-python-sbom.yml">https://github.com/eclipse-kuksa/kuksa-python-sdk/blob/main/.github/workflows/generate-python-sbom.yml</a>
Eclipse LMOS	gradle	<a href="https://github.com/eclipse-lmos/arc/blob/main/.github/workflows/generate-gradle-sbom.yml">https://github.com/eclipse-lmos/arc/blob/main/.github/workflows/generate-gradle-sbom.yml</a>
Eclipse JKube	maven	<a href="https://github.com/eclipse-jkube/jkube/blob/master/.github/workflows/generate-maven-sbom.yml">https://github.com/eclipse-jkube/jkube/blob/master/.github/workflows/generate-maven-sbom.yml</a>

# How to become an early adopter



- We currently have a queue of projects that we help with implementing **Github Actions workflows**
- Email [security@eclipse-foundation.org](mailto:security@eclipse-foundation.org) if you would like to be added
  - Subject: "SBOM Early Adopters"
  - Provide project context
    - Number of products in repository
    - Ecosystems
    - Versioning
    - Release process
    - CI/CD pipelines



# Future Goals



- SBOM Registry for all EF projects
- Reusable workflow templates collection
- Enriched resources library



# Resources

# Readings



- Introduction to SBOMs:  
<https://eclipse-csi.github.io/security-handbook/sbom/introduction.html>
- How to generate and upload SBOMs:  
<https://eclipse-csi.github.io/security-handbook/sbom/howto.html>
- Tooling ecosystem for CycloneDX:  
<https://eclipse-csi.github.io/security-handbook/sbom/tooling.html>



**Thank you!**