Estudio
SBOM en Entornos Industriales

Financiado por la Unión Europea
NextGenerationEU

Plan de Recuperación, Transformación y Resiliencia

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

TLP:CLEAR

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

Abril 2026

INCIBE-CERT_ESTUDIO_SBOM EN ENTORNOS INDUSTRIALES_v1.0

Índice

ÍNDICE DE FIGURAS

1. Sobre este estudio

Este estudio tiene el objetivo de ofrecer al lector los conocimientos necesarios para poder realizar un **SBOM** (Software Bill of Materials o Lista de Materiales de Software) completo sobre una aplicación o un dispositivo en concreto, pudiendo realizar así el análisis de este y conocer en detalle los componentes y dependencias del software. Este **análisis facilita la gestión del software de manera segura**, identificando las vulnerabilidades presentes y promoviendo la transparencia en la cadena de suministro.

Inicialmente se realizará una introducción sobre los SBOM, explicando qué son, su enfoque en entornos industriales, cuáles son sus componentes, etc. De esta forma, sin conocimientos previos, el lector podrá seguir una lectura guiada y aprender sin mayores problemas todos los puntos que se van a tratar más adelante sobre los SBOM.

En la segunda fase del estudio, se abordarán en detalle varias herramientas con capacidad para crear y leer un SBOM, destacando la manera en la que facilitan la gestión de componentes.

Finalmente, se realizará una demostración práctica para mostrar el funcionamiento de algunas de las herramientas más utilizadas.

Ilustración 1: Imagen de recurso. Código.

2. Introducción

Desde hace años, los ataques cibernéticos han aumentado de forma exponencial, tanto en número como en sofisticación, por eso, **las empresas cada vez dan más importancia a la ciberseguridad**, llegando a dedicar muchos recursos para intentar mitigar el impacto de un posible ciberataque.

El sector de los proveedores de *hardware* ha experimentado un aumento crítico y sin precedentes en el nivel de ciberataques durante los últimos años (2024-2026), consolidándose como uno de los principales objetivos debido a su posición estratégica en la cadena de suministro global.

Los ataques contra proveedores de componentes y fabricantes de *hardware* se duplicaron en 2025 en comparación con el año anterior, alcanzando un coste medio de 4,33 millones de euros por brecha de seguridad.

Los ciberdelincuentes priorizan la cadena de suministro (suministro de *hardware* y *software*) para acceder a clientes finales más grandes y mejor protegidos. El nivel de amenaza es muy alto y sigue una tendencia ascendente, con un aumento del 87% en ataques al sector industrial en 2025, impulsado por la digitalización de los procesos de fabricación y la sofisticación de los atacantes, a menudo utilizando inteligencia artificial. El último balance de ciberseguridad de INCIBE publicado en febrero de 2026, indica un crecimiento sostenido de ciberataques en España, con 122.223 incidentes gestionados en 2025, un 26% más que el año anterior. 55.411 de estos incidentes, estaban relacionados con *malware*, incluyendo virus y otros *softwares* maliciosos que afectan dispositivos, lo que supone más de un 45% del total de los incidentes.

De los sistemas infectados controlados por un ciberdelincuente de forma remota (botnet) identificados por INCIBE-CERT, el 85% estaban relacionados con dispositivos inteligentes (IoT), como televisores, decodificadores, reproductores multimedia, etc. Además, hay que tener en cuenta que el sector TIC y los proveedores de *hardware* son objetivos estratégicos, y representaron el 24,6% de los incidentes críticos en 2024.

Por esta razón, es de vital importancia disponer de un inventario de activos en el que se detallen no solo las características de estos, sino también saber la relación entre sí, para conocer sus dependencias y el alcance real de una vulnerabilidad. Al conocer cada dependencia y procedencia, las organizaciones son capaces de responder de forma más ágil y eficiente. Incluir un **SBOM** es crucial, en especial en sistemas de control industrial (SCI) debido a que **otorga una visión integral y precisa de toda la infraestructura**.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

3. Organización del documento

El tema principal de este estudio son los **SBOM** y su implementación en entornos industriales, destacando algunas herramientas para automatizar las tareas de creación y lectura. Para ello, el primer punto **Explicación SBOM**, introduce al lector en los SBOM explicando qué son, cuáles son sus estándares y la importancia del VEX. Una vez explicados los conocimientos básicos, en los apartados **SBOM en Entornos Industriales** y **Problemas de SBOM en el mundo industrial**, se explican las ventajas y problemáticas de utilizar SBOM en entonos industriales.

El apartado **Buenas prácticas de un SBOM**, explica algunas de las acciones más importantes a tener en cuenta al realizar un documento SBOM. En el apartado **Herramientas** se explican algunas herramientas desarrolladas para la creación y lectura de documentos SBOM de forma sencilla y automatizada.

Finalmente, el apartado **Conclusiones** expone un breve resumen donde se juntarán todas las ideas principales de este estudio, de forma clara y concisa sobre los SBOM.

4. Explicación SBOM

Los **SBOM** son un **inventario detallado de los componentes de origen y las dependencias de una aplicación o de un servicio**. Este inventario se compone de objetos compartidos, bibliotecas y código fuente tanto abierto como privado. Es similar a un inventario tradicional, pero con la diferencia de que un SBOM también detalla las dependencias de la aplicación o dispositivo, generando un árbol de dependencias que proporciona una completa visibilidad de la infraestructura de la empresa.

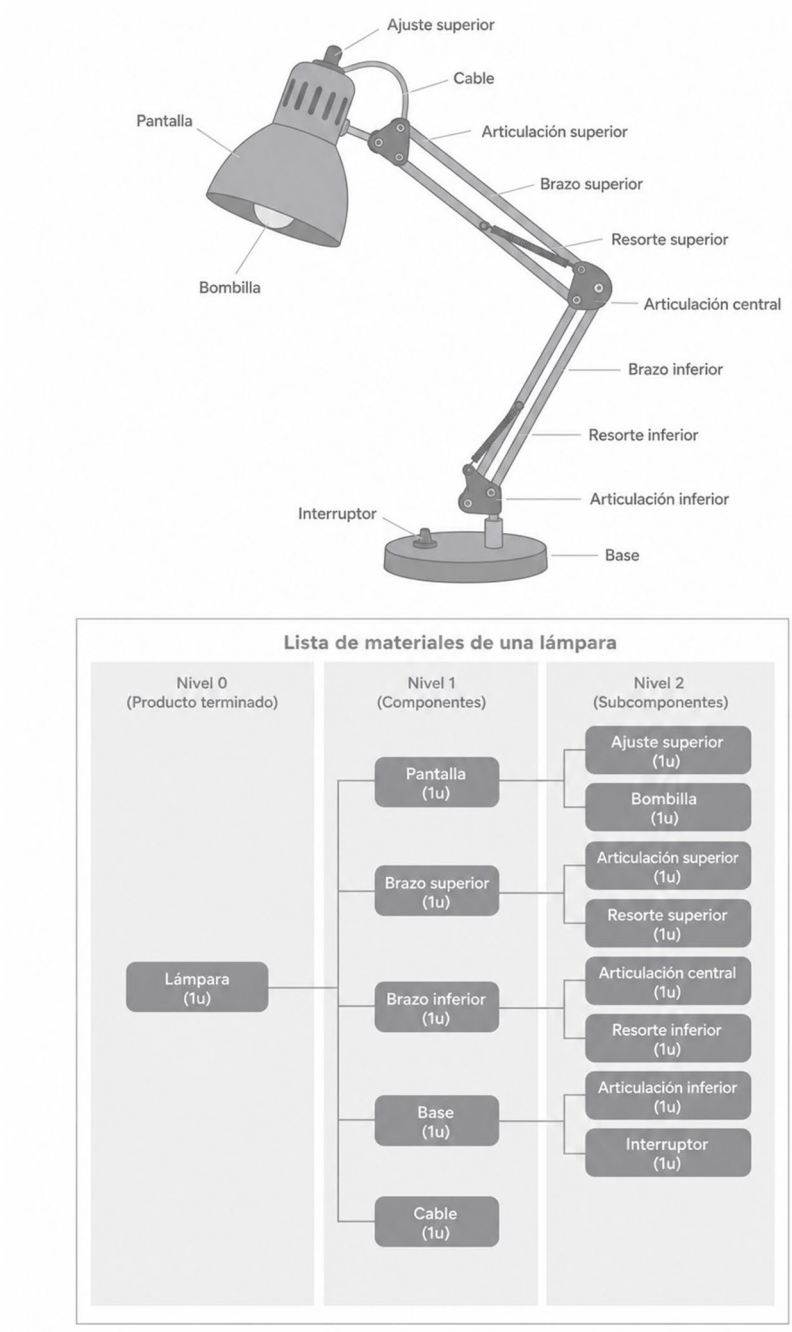


Ilustración 1: Ejemplo árbol de dependencias.

La imagen anterior muestra un **BOM** (Bill of Materials) de una lámpara, en el que se detallan sus componentes y subcomponentes. Esta imagen es perfecta para explicar los componentes y dependencias de un SBOM. Un **componente** es **cualquier pieza** de **software que forma parte del sistema o aplicación**. Pueden ser bibliotecas, módulos, dependencias externas, etc. En este ejemplo, los componentes son los presentes en el nivel 1 (pantalla, brazo superior, brazo inferior, base, cable), que a su vez dependen de otros subcomponentes presentes en el nivel 2 (ajuste superior, bombilla, etc). Estos subcomponentes reflejan las dependencias, es decir, las dependencias son otros paquetes de software, módulos o componentes adicionales de los cuales cada componente dependen para funcionar correctamente.

A mediados del 2021, la National Telecommunications and Information Administration (NTIA) emitió una guía que detalla los elementos mínimos que debe contener un SBOM, en las que se incluyen las siguientes **tres áreas fundamentales**:

- **Campos de datos**: Es el componente de inventario de activos que describe el árbol de dependencias de una aplicación. Los campos deben incluir el nombre, la versión, la información de la licencia y el nombre del proveedor entre otras características.
- **Automatización**: Debido a la complejidad de los procesos de desarrollo actuales, la NTIA recomienda un mínimo de automatización en la generación de los SBOM y la transferencia de datos, de modo que otros sistemas sean capaces de comprender la información. Hay una gran variedad de herramientas para la creación de un SBOM y algunas de ellas se explicarán más adelante.
- **Procesos**: Implementación de procesos adecuados para facilitar la recopilación continua de datos.

Ilustración 2: Contenido de los SBOM y contenidos mínimos - Estándares SBOM

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

Para facilitar esta adopción, se han desarrollado varios **estándares SBOM** para proporcionar una estructura unificada tanto para generar SBOM como para compartirlos. Estos estándares proporcionan un formato común para describir la composición del software de tal forma que sea utilizable por otras herramientas, como los escáneres de vulnerabilidades. Los estándares más utilizados son CycloneDX y Software Package Data Exchange (SPDX).

El estándar **CycloneDX** es un estándar SBOM ligero útil para contextos de seguridad de aplicaciones y análisis de componentes de la cadena de suministro. Es un proyecto de código abierto creado por la comunidad OWASP y es la encargada del mantenimiento y la actualización del estándar. La especificación detallada se puede encontrar en su github o en su página web. El equipo detrás de CycloneDX es también responsable de Dependency Track y las numerosas herramientas comerciales de código abierto que implementan la especificación CycloneDX. Este estándar evoluciona muy rápido y es capaz de admitir VEX (Vulnerability Exploitability eXchange) y HBOM para componentes hardware entre muchas otras características.

Por otro lado, el segundo estándar más utilizado es **Software Package Data Exchange** o **SPDX**. SPDX es un formato de estándar abierto internacional para comunicar los componentes, las licencias y los derechos de autor asociados a un paquete de software. Es la única especificación que se reconoce como estándar ISO en forma de ISO/EIC 5962:2021, la cual describe el estado de SPDX en la versión 2.2.21. Este estándar es un proyecto de código abierto organizado por la Linux Foundation, la cual reúne a representantes de proveedores, fundaciones e integradores de sistemas. Este proyecto mantiene informados a los usuarios mediante informes mensuales en los que detallan el estado del proyecto. La especificación detallada está disponible en el github de SPDX o en su página web.

Ambos estándares se pueden utilizar para generar, compartir y administrar datos SBOM, son muy utilizados y aceptados en la mayoría de las plataformas. **SPDX** se desarrolló principalmente para casos de uso de propiedad intelectual y cumplimiento de licencias de código abierto, aunque ha evolucionado significativamente, mientras que **CycloneDX**, al ser más reciente, se creó desde un principio teniendo en cuenta la seguridad. Tiene un soporte más amplio para la gestión de vulnerabilidades, incluidos los datos VEX, los cuales se utilizan para comunicar la explotabilidad de los componentes de software en el contexto en el que se utilizan. Tanto SPDX como CycloneDX están disponibles en JSON y XML.

4.1. Vulnerability Exploitability eXchange (VEX)

Como se ha mencionado anteriormente, los SBOM ofrece una mayor visibilidad de los activos de una empresa, mejorando en gran medida la seguridad. A pesar de esto, puede ir acompañado de muchos falsos positivos o vulnerabilidades asociadas al código fuente de la organización que no pueden ser explotadas por diversas circunstancias específicas o que simplemente no son un problema a tener en cuenta debido a su difícil explotación por parte de un ciberdelincuente. Este aumento de falsos positivos puede obstaculizar la seguridad y hacer que utilizar un SBOM no sea tan eficaz.

Para abordar este problema, idearon el **Vulnerability Exploitability eXchange** (VEX), un sistema diseñado para ayudar a los usuarios a eliminar aquellas vulnerabilidades poco peligrosas por ser muy difícil de explotar y centrarse en las vulnerabilidades que presentan un riesgo real.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

Al publicar un documento VEX para una vulnerabilidad, se asigna al producto uno de los siguientes estados:

- **No afectado**: No se requiere ninguna corrección para esta vulnerabilidad.
- **Afectado**: Se recomiendan acciones para remediar la vulnerabilidad.
- **Corregido**: Las versiones del producto disponen de soluciones o parches para la vulnerabilidad.
- **En investigación**: Aún no se sabe si dichas versiones están afectadas.

Estos **documentos VEX** se utilizan para determinar si los activos están **afectados por alguna vulnerabilidad y qué medidas hay que tomar para solucionarlo**. Implementar los VEX al SBOM aumenta la eficacia y permite realizar un escaneo exhaustivo del código fuente en busca de vulnerabilidades.

5. SBOM en Entornos Industriales

La **implementación de los SBOM en los entornos industriales** proporcionan una gran cantidad de ventajas entre las cuales, destacan la mejora de la ciberseguridad y la gestión de activos. Proporcionan un inventario muy detallado sobre los dispositivos y software presentes en el entorno industrial, lo que permite tener una mayor visibilidad y transparencia de qué se está ejecutando. Estas son algunas de las ventajas que ofrece SBOM:

- **Visibilidad de componentes**: Proporciona una visión clara y completa de los activos presentes. Esta transparencia es crucial para comprender la estructura del software, las dependencias y sus posibles vulnerabilidades.
- **Gestión de vulnerabilidades**: Facilita la gestión de vulnerabilidades, desde su descubrimiento hasta su solución. Si se descubre una nueva vulnerabilidad, un SBOM permite conocer al instante si ese componente se encuentra desplegado en los sistemas OT, facilitando una respuesta rápida y eficiente.
- **Agilización de las auditorías**: Los entornos OT están regulados por estrictas normativas en cuanto a la ciberseguridad y los auditores requieren de información detallada y actualizada sobre los componentes que se utilizan en el software. Un SBOM proporciona dicha información de forma clara y concisa, ahorrando tiempo.
- **Mejorar la gestión de parches**: Facilitan el seguimiento de las versiones de los activos garantizando que todos estén actualizados a la última versión o como mínimo, a la última versión posible.
- **Mitigación de riesgos en la cadena de suministro**: En los entornos OT es común utilizar hardware y software de varios proveedores. Un SBOM ayuda a identificar el software de terceros que puede estar presente y a detectar posibles problemas de seguridad provenientes de la cadena de suministro.
- **Facilita la adquisición de activos:** Los SBOM ayudan en la toma de decisiones sobre que software o hardware escoger, debido a que ofrecen información muy valiosa sobre los componentes y su estado de seguridad.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

6. Problemas de SBOM en el mundo industrial

Los **SBOM** ofrecen muchas ventajas como se ha visto en el punto anterior, pero a veces su implementación puede resultar difícil. Los "*SBOM*" se encuentran en una etapa temprana de su desarrollo, por ello es muy difícil encontrarlos en el entorno industrial, pero seguramente con las continuas mejoras y el aumento de su uso por parte de los responsables de ciberseguridad en el entorno IT va a hacer que se empiece a implementar en el mundo OT. En los siguientes puntos se explicarán las **principales problemáticas que puede llevar a que el método *"SBOM"*** sea complicado implementarlo en el entorno industrial.

- **Ausencia de estándares**: A diferencia de los entornos IT, donde se están comenzando a proponer ciertos modelos, los entornos OT no disponen de estándares unificados.
- **Propiedad intelectual**: Muchos proveedores consideran que la información de sus productos son parte de su propiedad intelectual, además de ser información altamente sensible. Esto requiere ciertas formas para poder compartir dicha información y por lo tanto implica una dificultad extra para generar "*SBOM*" completos y que puedan ser compartidos con facilidad.
- **Falta de tecnologías:** Por el momento, la mayoría de las tecnologías y los avances en los "*SBOM*" están enfocados a los entornos IT.
- **Dificultad en la automatización:** En gran parte de las ocasiones, no es posible automatizar los procesos de creación y actualización de los "*SBOM*" debido a limitaciones en los entornos industriales como la falta de conexión o la localización de algunos dispositivos. Por lo tanto, las tareas de gestión y mantenimiento de los "*SBOM*" requiere de una gran cantidad trabajo manual.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

7. Buenas prácticas de un SBOM

A pesar de las ventajas que ofrece SBOM, puede ser complicada su implementación dando lugar a problemáticas que puedan derivar en una mala implementación del SBOM o incluso a no llegar a desarrollarlo. A continuación, se detallan **algunas buenas prácticas a tener en cuenta cuando se desarrolle un SBOM**:

- **Establecer políticas y procedimientos claros**: Crear un marco estructurado para gestionar los componentes del software. Estas políticas ayudan a la empresa a definir las reglas para crear, mantener y actualizar los SBOM.
- **Utilizar formatos de datos estándar**: Los formatos estándar como CycloneDX y SPDX garantizan la uniformidad, interoperabilidad y facilidad de comprensión en diferentes proyectos y organizaciones.
- **Automatizar la generación de SBOM**: Los documentos SBOM son muy complejos, rastrean cada componente de software, sus dependencias y recopilan metadatos. Este proceso se vuelve más complicado en proyectos a gran escala, por lo que una buena práctica es automatizar la creación de estos documentos.
- **Actualizar periódicamente el SBOM**: La actualización periódica permite mantener la relevancia y la precisión de la información. La frecuencia de las actualizaciones puede ser muy variada dependiendo del proyecto, pero es recomendable alinear las actualizaciones con los logros de desarrollo del software o los ciclos de mantenimiento.
- **Permitir la resolución de errores**: A pesar de que la información debe analizarse de forma minuciosa, pueden existir errores en la creación de los SBOM y las organizaciones deben incluir exenciones de responsabilidad en estos documentos.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

8. Herramientas

Como se ha visto anteriormente en este estudio, la mejor forma de realizar e interpretar un SBOM es mediante la utilización de herramientas que automaticen estas tareas para crear documentos lo más completos posibles. En este apartado, se verán algunas de las **herramientas más utilizadas actualmente para la creación y gestión de los SBOM**.

8.1. Herramienta cdxgen

La **primera herramienta** que se va a mostrar en este artículo es **cdxgen**, creado por OWASP. Es una herramienta capaz de crear un SBOM a partir de un repositorio git, un contenedor o un sistema operativo. El estándar en el que se basa es CycloneDX y soporta varios lenguajes y formatos.

8.1.1. Instalación

Su instalación es muy sencilla y rápida. Hay diferentes formas de instalarlo como se puede ver a continuación:

8.1.1.1. Instalación con npm

Esta vía es la más cómoda de todas, pero requiere tener instalado node.js. Una vez instalado, la **cdxgen** se instala con un único comando:

```
npm install -g @cyclonedx/cdxgen
```

Si surge algún problema con este método, puede ser porque node.js requiere de los siguientes permisos:

- **–allow-fs-read**: Permisos de lectura para la aplicación o el directorio actual.
- **–allow-fs-write**: Permisos de escritura en el Directorio temp.
- **–allow-child-process**: Para algunos lenguajes, se requieren permisos “ChildProcess” para generar comandos.

Estos permisos se pueden otorgar mediante el siguiente comando:

```
node --experimental-permission --allow-fs-read="/home/almalinux/work*" --allow-fs-write=/tmp --allow-child-process bin/cdxgen.js -o /tmp/bom.json ~/work/sandbox/vuln-spring -t java
```

8.1.1.2. Instalación con Homebrew

Otra forma sencilla de instalar **cdxgen** es mediante la herramienta **homebrew**. Se requiere tener instalado homebrew pero al igual que el método anterior, es muy sencilla su instalación. Una vez instalado, la herramienta cdxgen se instala mediante un comando:

```
brew install cdxgen
```

8.1.1.3. Instalación con Docker

Como última opción, esta herramienta se puede instalar mediante la utilización de un contenedor con **docker**. Hay tres versiones diferentes:

La versión por defecto que utiliza **Node.js 22**:

```
docker run --rm -e CDXGEN_DEBUG_MODE=debug -v /tmp:/tmp -v $(pwd):/app:rw -t ghcr.io/cyclonedx/cdxgen:master -r /app -o /app/bom.json
```

La versión con **deno**:

```
docker run --rm -e CDXGEN_DEBUG_MODE=debug -v /tmp:/tmp -v $(pwd):/app:rw -t ghcr.io/cyclonedx/cdxgen-deno:master -r /app -o /app/bom.json
```

La versión con **bun**:

```
docker run --rm -e CDXGEN_DEBUG_MODE=debug -v /tmp:/tmp -v $(pwd):/app:rw -t ghcr.io/cyclonedx/cdxgen-bun:master -r /app -o /app/bom.json
```

8.1.2. Uso cdxgen

La herramienta **cdxgen** se puede ejecutar en dos modos diferentes, “cli mode” y “server mode”.

8.1.2.1. Cli mode

El **modo cliente**, se puede ejecutar usando como datos de entrada código fuente, una imagen de contenedor o un artefacto binario para crear el documento SBOM.

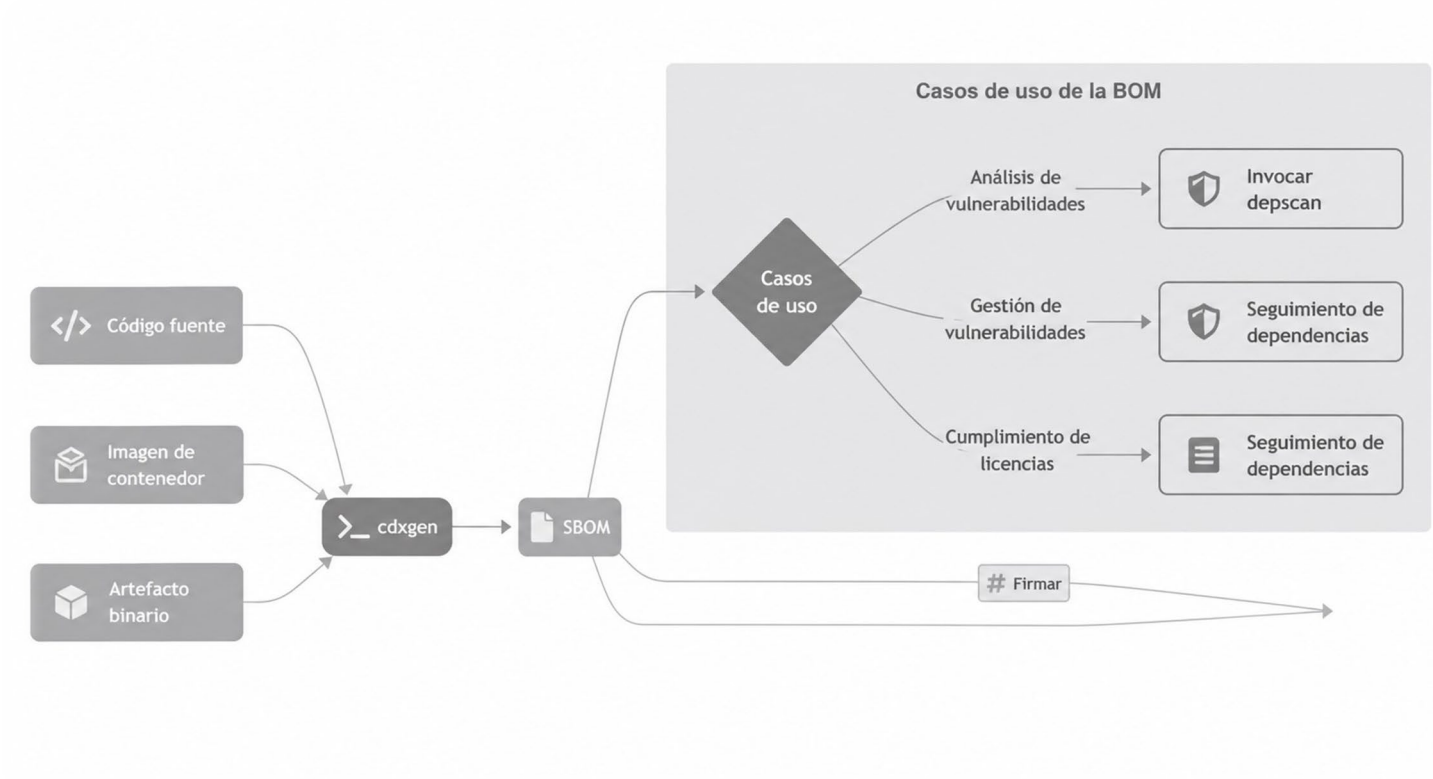


Ilustración 3: Modelo funcionamiento herramienta cdxgen en client mode

Algunos ejemplos de cómo funciona la herramienta son los siguientes:

```
cdxgen -r -t java -o bom.json
```

Este comando se ejecuta en el mismo directorio en dónde se encuentra el proyecto a analizar. El parámetro "-r" indica que se realice recursividad para analizar todos los archivos, el parámetro "-t" indica el tipo del proyecto, en este caso en un software desarrollado en java y el parámetro "-o" indica el archivo en donde se va a guardar el output del comando.

Cabe destacar que el parámetro "-t" puede ser "universal", esto hace que **cdxgen** recolecte toda la información posible escaneando todos los archivos, independientemente del formato o lenguaje en el que estén desarrollados.

Otra utilidad importante de esta herramienta es la posibilidad de realizar un **OBOM** (Operations Bill of Materials) de un sistema operativo activo o una máquina virtual, pudiendo gestionar las vulnerabilidades.

```
cdxgen -t os
```

8.1.2.2. Server mode

Este modo utiliza la herramienta en **modo servidor**, es decir, poder hacer consultas y realizar documentos SBOM de proyectos sin necesidad de descargarlos o instalarlos, tal y como se muestra en la siguiente imagen:

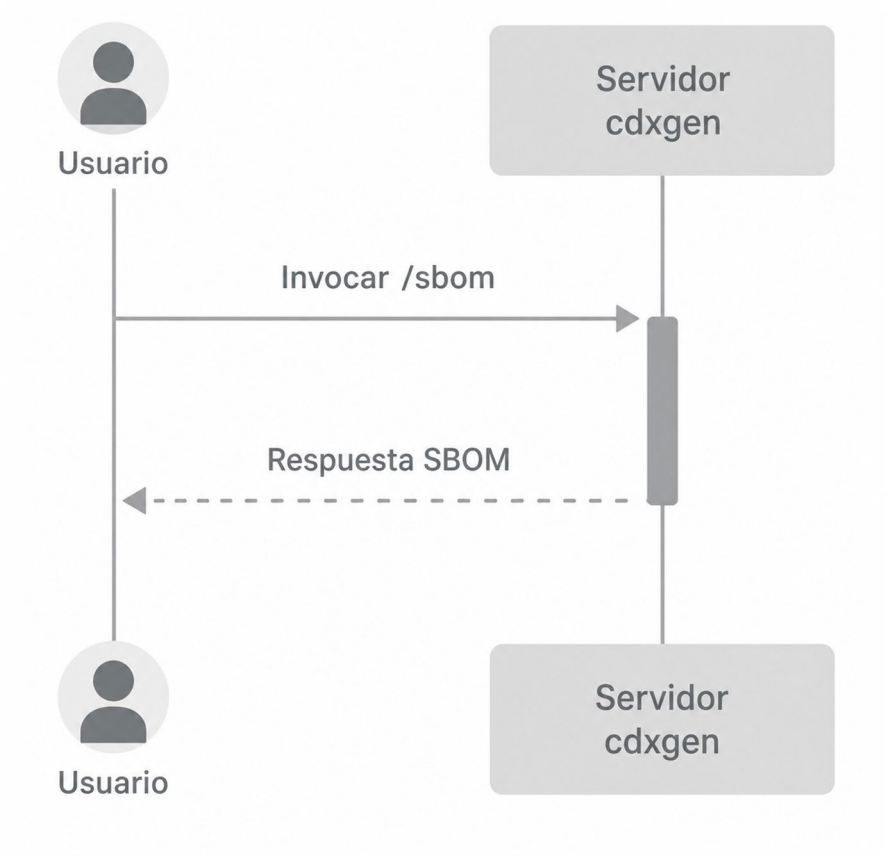


Ilustración 4: Modelo funcionamiento herramienta cdxgen en server mode

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

Ejecutar **cdxgen en modo servidor** es realmente sencillo, basta con ejecutar el siguiente comando:

```
cdxgen --server
```

O utilizar una imagen contenedor y ejecutar el siguiente comando, aunque este método es menos recomendable:

```
docker run --rm -v /tmp:/tmp -p 9090:9090 -v $(pwd):/app:rw -t ghcr.io/cyclonedx/cdxgen
-r /app --server --server-host 0.0.0.0
```

Una vez levantado el servidor, hay que verificar que funciona correctamente. Para ello, hay que ejecutar el siguiente comando:

```
curl http://127.0.0.1:9090/health
```

Si se recibe un mensaje con un "Ok" como se muestra a continuación, significa que el servidor funciona correctamente:

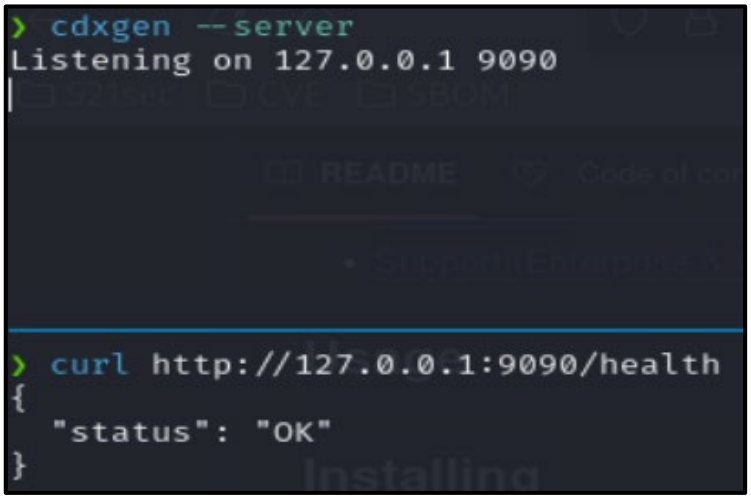


Ilustración 5: cdxgen modo servidor. Fuente: Propia.

Por defecto, el modo servidor se levanta en la IP 127.0.0.1 por el puerto 9090, pero estos parámetros se pueden modificar. En modo servidor, cdxgen es capaz de escanear directorios locales y repositorios git, tanto públicos como privados en los que se requieran credenciales, como se puede ver en los siguientes ejemplos:

```
curl
http://127.0.0.1:9090/sbom?url=https://github.com/some/repo.git&multiProject=true&type
=js

curl
"http://127.0.0.1:9090/sbom?url=https://<access_token>@github.com/some/repo.git&multiP
roject=true&type=js"

curl
"http://127.0.0.1:9090/sbom?url=https://<username>:<password>@bitbucket.org/some/repo.
git&multiProject=true&type=js"
```

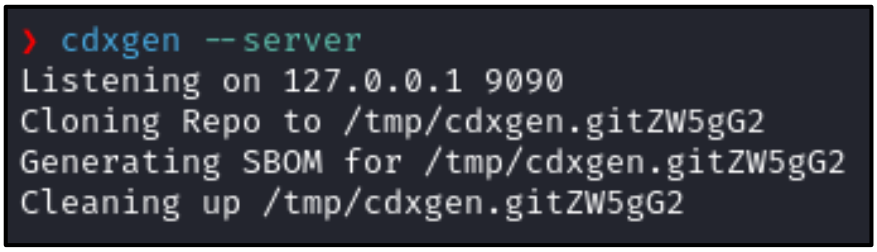


Ilustración 6: Output cdxgen en modo servidor.

Como se puede ver en la imagen anterior, al hacer una petición al servidor con un repositorio, lo clona en la máquina local para realizar el escaneo y generar el documento SBOM y finalmente lo elimina.

8.2. Herramienta syft

La herramienta **syft** es una herramienta "open source" desarrollada por Anchore para la creación de documentos SBOM. Es capaz de analizar contenedores y sistemas de archivos ofreciendo información detallada de los paquetes y de las dependencias del software.

8.2.1. Instalación

La instalación de esta herramienta es muy rápida y dispone de varios métodos por si alguno fallase.

8.2.1.1. Instalación recomendada

Este método es el recomendado para realizar la instalación.

```
curl -sSfL https://raw.githubusercontent.com/anchore/syft/main/install.sh | sh -s -- -b /usr/local/bin
```

8.2.1.2. Instalación con Homebrew

Al igual que la herramienta cdxgen, esta también permite su instalación por medio de homebrew:

```
brew install syft
```

Existen otros métodos de instalación disponibles en el github de la herramienta pero que la ser poco comunes, no se detallarán en este estudio.

8.2.2. Uso syft

Una vez instalada la **herramienta syft**, se puede utilizar para analizar sistemas de archivos o contenedores sin ningún paso previo. Para ello, hay que ejecutar el siguiente comando:

```
syft <image>
```

Este es el comando más simple que se puede ejecutar y sólo incluye el software visible en el contenedor. Si se quiere hacer un escaneo más exhaustivo e incluir todas las capas de la imagen en el SBOM, hay que añadir el siguiente parámetro:

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

```
syft <image> --scope all-layers
```

También cabe destacar que se puede configurar el "output" que ofrece syft mediante el parámetro "-o" o "- -output". Se puede elegir entre varias opciones en las que destaca poder exportar la información en formato cycloneDX, SPDX y json, siendo este último el más interesante porque obtiene toda la información posible.

Ilustración 7: Resultado ejecución comando syft.

Ilustración 8: Resultado ejecución comando syft.

8.3. Herramienta Dependency Track

La herramienta Dependency Track, desarrollada por OWASP, es una plataforma de análisis de componentes que permite identificar y reducir riesgos en la cadena de suministro. Esto lo consigue aprovechando las características de los SBOM para identificar los riesgos. La plataforma tiene un diseño API-first y su uso es muy útil en entornos CI/CD. CI/CD son las siglas para la integración y distribución continua, cuyo objetivo es mejorar y agilizar el ciclo de vida del desarrollo del software.

8.3.1. Instalación

Esta herramienta se puede instalar con docker, kubernetes y un archivo ejecutable WAR. En este estudio se va a explicar cómo instalarlo con Docker por ser el método más sencillo y rápido.

Pero antes de comenzar a explicar la instalación, hay que resaltar que esta herramienta consta de dos componentes principales:

- **API Server**: Un servidor con diseño API-first. Sus requisitos son los siguientes:
  - **Requisitos mínimos**: 4.5GB de RAM y 2 procesadores.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

- **Requisitos recomendados**: 16GB de RAM y 4 procesadores.

- **Front End**: La interfaz gráfica. Sus requisitos son los siguientes:
  - **Requisitos mínimos**: 512MB de RAM y 1 procesador.
  - **Requisitos recomendados**: 1GB de RAM y 2 procesadores.

8.3.1.1. Quickstart (Docker Compose)

Este método es el más sencillo y recomendado por OWASP para instalarlo. Para ello, se recomienda instalar Docker Desktop, el cual incluye la instalación de Docker Compose. Una vez instalado Docker Desktop, tan solo hay que ejecutar los siguientes comandos:

```
curl -LO https://dependencytrack.org/docker-compose.yml

docker compose up -d
```

Primero se descarga el archivo docker-compose.yml necesario para crear la plataforma y con el segundo comando se instala de forma autómatica.

8.3.1.2. Quickstart (Docker Swarm)

Este método es rápido, pero es un poco más complicado, ya que se va a utilizar Docker Swarm. Al igual que en el método anterior, es necesario descargar el archivo docker-compose.yml. A continuación, se inicia Docker Swarm sino se ha iniciado automáticamente y finalmente se inicia el proceso de instalación.

```
curl -LO https://dependencytrack.org/docker-compose.yml

docker swarm init

docker stack deploy -c docker-compose.yml dtrack
```

8.3.1.3. Quickstart (Manual Execution)

Este método es el más complejo y se requiere de conocimientos técnicos sobre docker para comprender cómo se lleva a cabo la instalación.

En primer lugar, hay que obtener la imagen del repositorio de OWASP:

```
docker pull dependencytrack/bundled
```

Una vez obtenido la imagen, es necesario crear un volumen en el que se puedan almacenar los datos fuera del contenedor:

```
docker volume create --name dependency-track
```

Finalmente, hay que ejecutar el contenedor. Es recomendable desplegarlo con 8 GB de RAM:

```
docker run -d -m 8192m -p 8080:8080 --name dependency-track -v dependency-track:/data
dependencytrack/bundled
```

Independientemente del método de instalación usado para desplegar Dependency Track, hay que tener en cuenta que la primera vez que se ejecuta tarda en configurarse por completo entre 10 y 30 minutos.

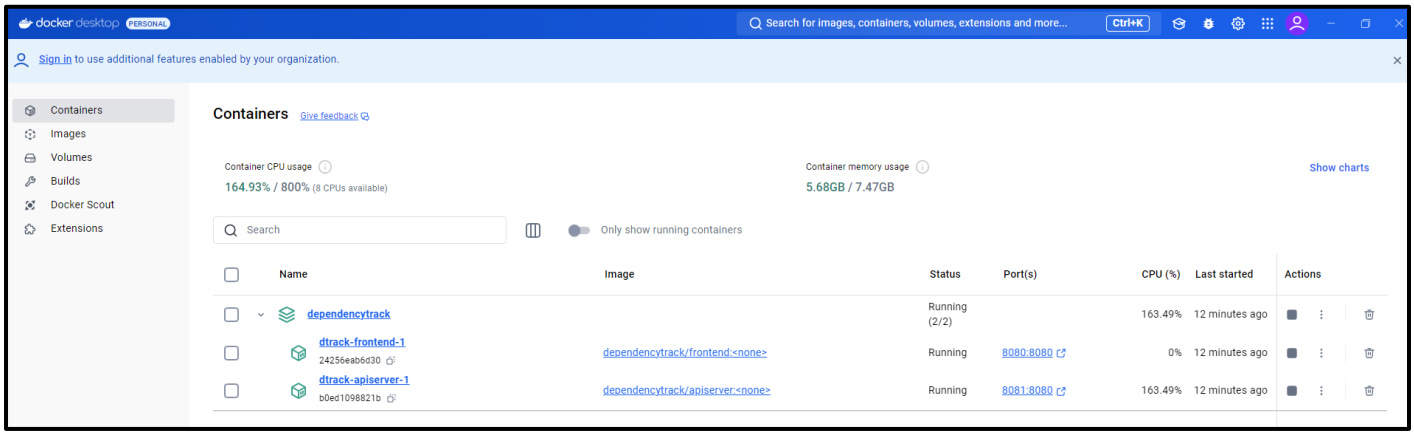


Ilustración 9: Contenedores Dependency Track.

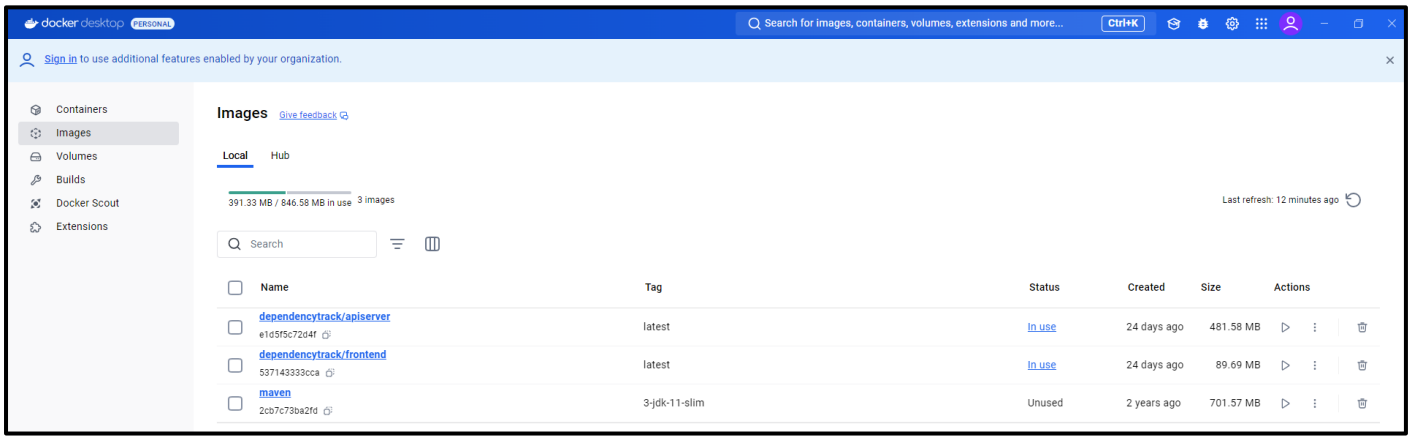


Ilustración 10: Imágenes Dependency Track.

8.3.2. **Acceso a Dependency Track**

Una vez instalada **Dependency Track** y se haya levantado la plataforma correctamente, el siguiente paso es acceder. Por defecto, se accede al panel de inicio de sesión mediante la url http://127.0.0.1:8080. Si todos los pasos anteriores se han realizado correctamente debería aparecer la siguiente página:

Ilustración 11: Panel de inicio de sesión Dependency Track.

Las credenciales por defecto son “admin:admin” y es necesario cambiar la contraseña después del primer inicio de sesión. Después de iniciar sesión correctamente, se podrá acceder a la siguiente interfaz:

Ilustración 12: Dashboard Dependency Track.

En ambas imágenes se puede ver un menú y diferentes paneles con características importantes sobre el software analizado.

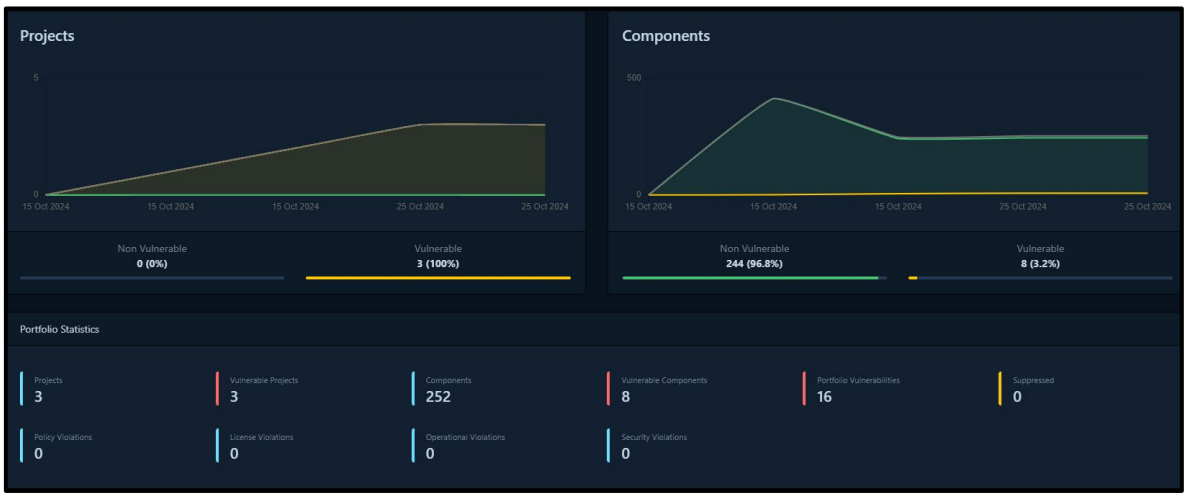


Ilustración 13: Dashboard Dependency Track.

8.3.3. **Distribución de Dependency Track**

Como se puede ver en la siguiente imagen, la plataforma esta agrupada en tres grandes grupos que son “*Portfolio”, “Global Audit”* y “*Administration”,* que a su vez están formados por varios subgrupos.

Ilustración 14: Menú principal.

En este caso, los **subgrupos más relevantes** son los pertenecientes al apartado "***Portfolio***".

8.3.3.1. Projects

En este primer apartado, "***Projects***", es donde se crean los proyectos y en donde se tiene un primer vistazo de las características principales de los proyectos ya creados.

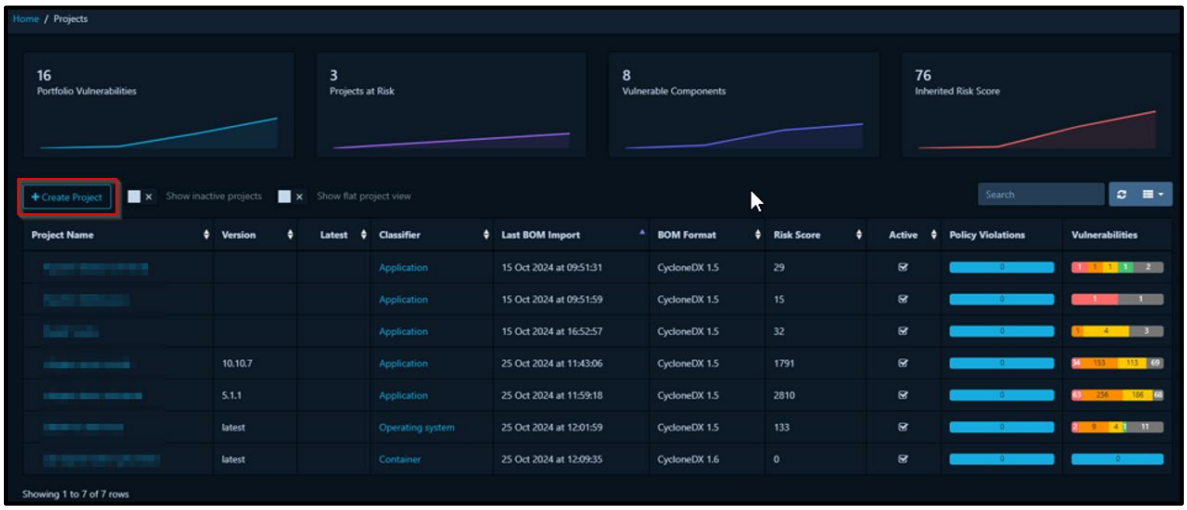


Ilustración 15: Panel principal apartado proyectos.

En la ilustración previa, se puede observar la información destacable de cada proyecto, el nombre del proyecto, la versión, el formato del BOM, la valoración de riesgo, las vulnerabilidades, etc.

Desde esta vista, también es posible crear un proyecto haciendo click en el botón "*Create Project*" remarcado en rojo en la ilustración previa.

La creación del proyecto se puede apreciar en la siguiente imagen:

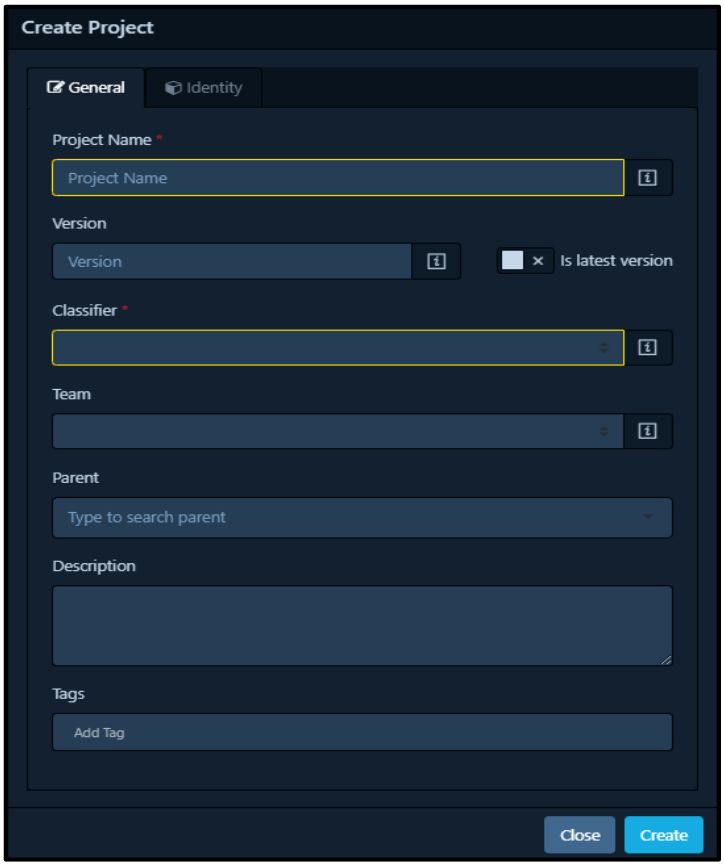


Ilustración 16: Crear proyecto en Dependency Track.

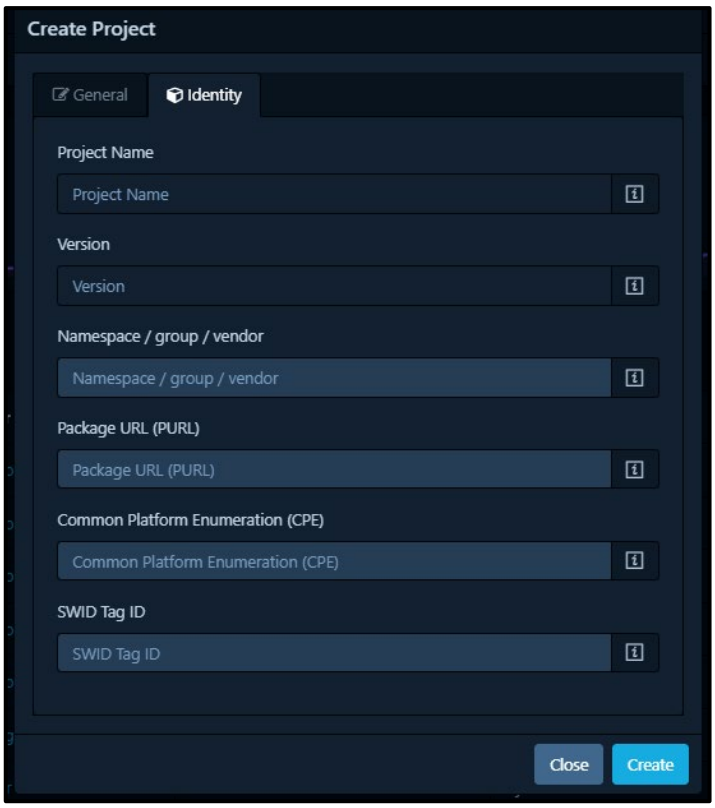


Ilustración 17: Crear proyecto en Dependency Track.

En la Ilustración 17 y la Ilustración 18 se muestran los campos a rellenar para crear un proyecto. Solo **son obligatorios los apartados "*Project Name*" y "*Classifier*",** el resto son opcionales. Una vez creado el proyecto, estos campos se pueden editar con total libertad.

A continuación, es necesario **subir un archivo BOM** para que Dependency Track pueda escanearlo y muestre toda la información relevante. Para ello, hay que hacer click en el proyecto creado e ir al apartado "*Components*".

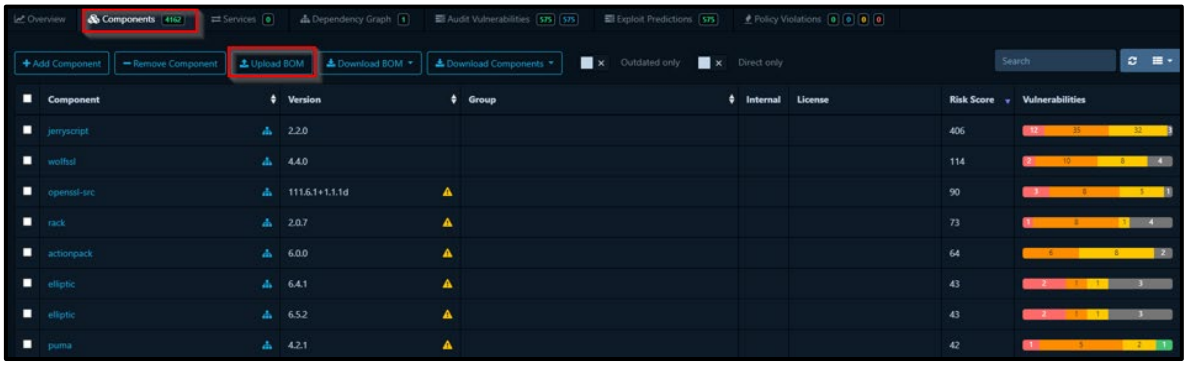


Ilustración 18: Panel Components.

A continuación, se ejecuta el botón "*Upload BOM*" y se elige el archivo creado con herramientas dedicadas especialmente para la creación de estos documentos, como las enseñadas en este estudio. Después de subir con éxito el archivo, la herramienta tarda un par de minutos en analizar toda la información y en mostrarla.

Una vez se ha analizado el archivo, en esta pestaña se muestran todos los componentes implicados en el software, así como su versión y la cantidad de vulnerabilidades que tienen. Si algún componente no se ha detectado, se puede añadir manualmente.

En la pestaña "***Overview***", se muestra un pequeño resumen de todo lo que se ha analizado.

Ilustración 19: Overview del proyecto.

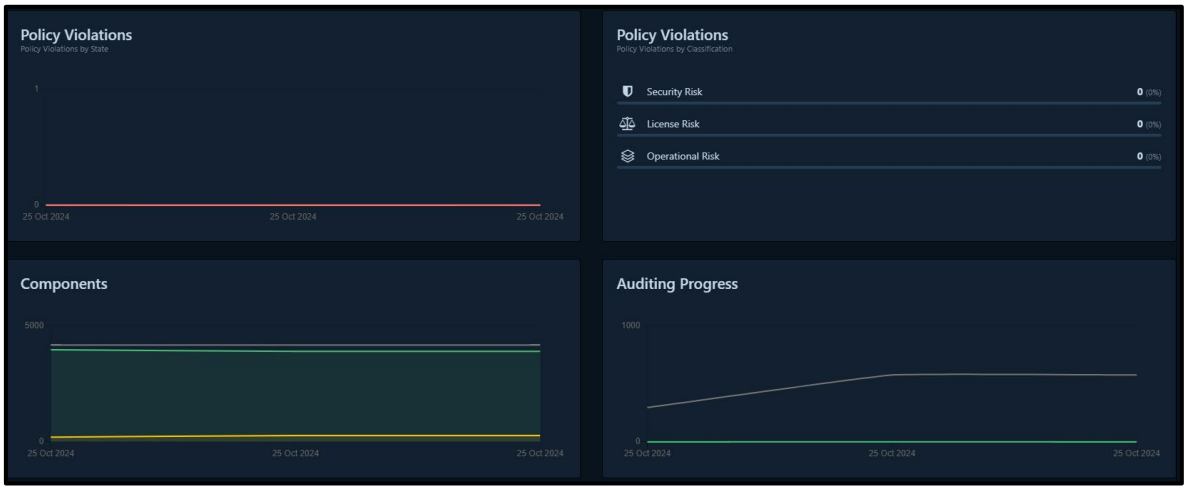


Ilustración 20: Overview del proyecto.

En esta pestaña se pueden ver la cantidad total de vulnerabilidades que se han encontrado categorizadas por criticidad, cantidad de componentes, incumplimientos de políticas y la puntuación de riesgo.

El apartado "***Services***" está diseñado para mostrar información detallada sobre los servicios utilizados en el proyecto, como el nombre, la versión, la puntuación de riesgo, etc..

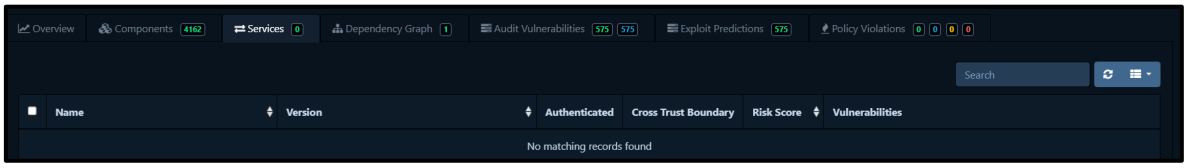


Ilustración 21: Services.

En "***Dependency Graph***", se muestra un árbol de todas las dependencias de los componentes permitiendo de forma muy visual y fácil, consultar las dependencias susceptibles de ser vulnerables.

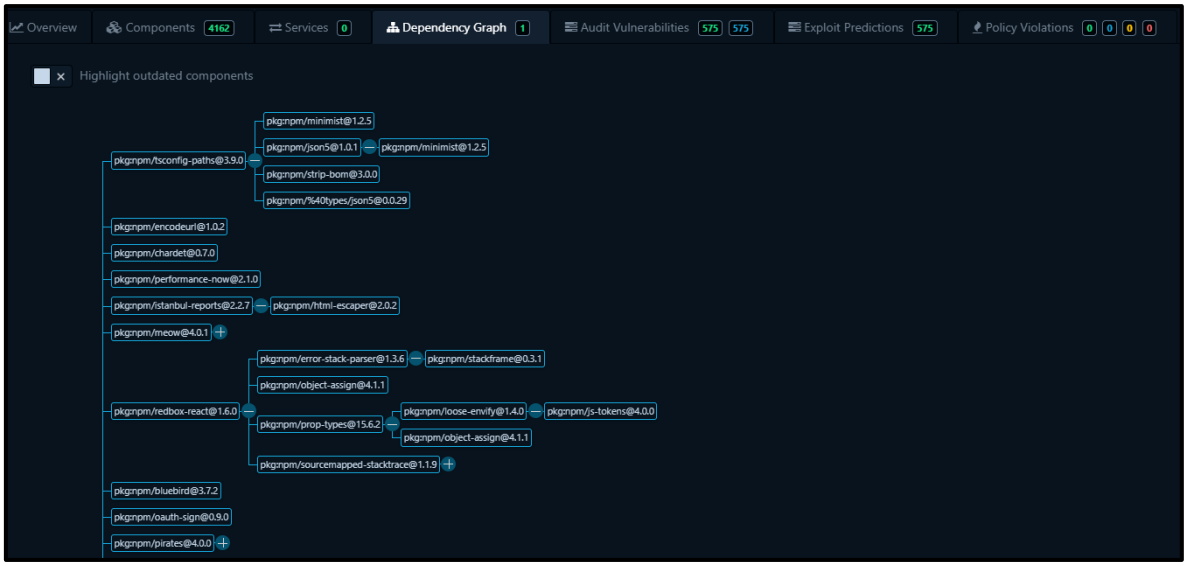


Ilustración 22: Dependency Graph.

El apartado "***Audit Vulnerabilities***" muestra las vulnerabilidades detectadas mostrando en una tabla información relevante como la versión, el CVE o la criticidad.

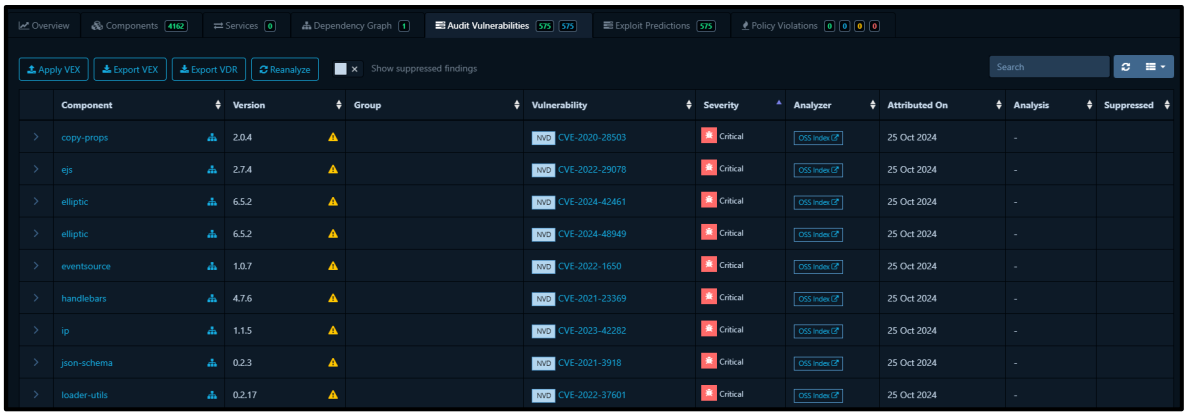


Ilustración 23: Audit Vulnerabilities.

También se puede consultar la información de cada componente y la de los CVE específicamente:

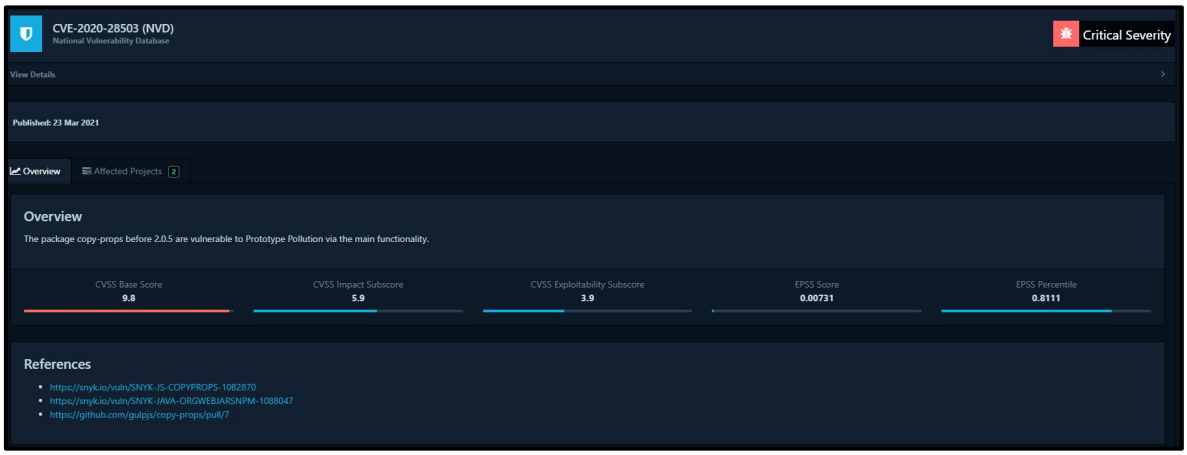


Ilustración 24: Información CVE.

En el apartado "***Exploit Predictions***" se muestra una tabla en la que se indican las vulnerabilidades detectadas según su criticidad (eje de abscisas o eje X) y la probabilidad real de explotar cada una de ellas (eje de ordenadas o eje Y).

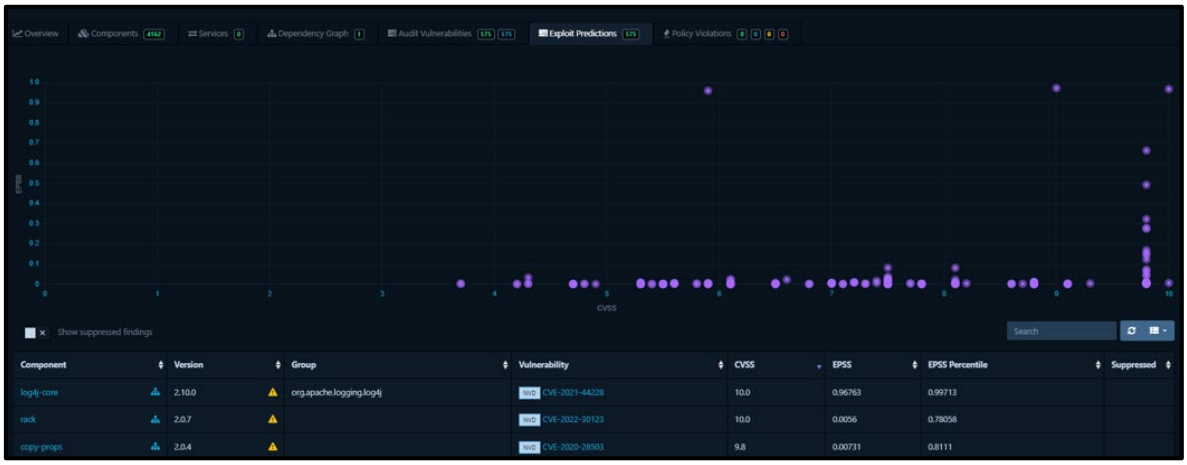


Ilustración 25: Exploit Predictions.

Por último, en la pestaña "***Policy violations***" se muestran los incumplimientos de las políticas de seguridad.

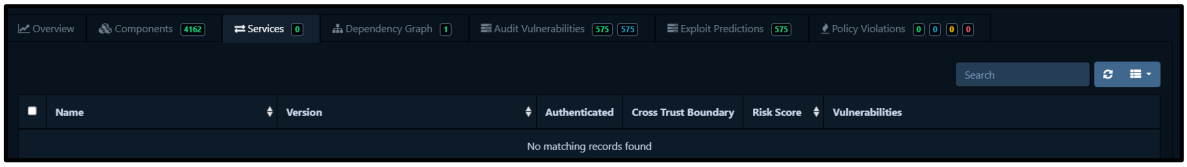


Ilustración 26: Pestaña Policy violations.

8.3.3.2. Components

En esta sección aparecen todos los componentes analizados por Dependency Track en los proyectos activos.

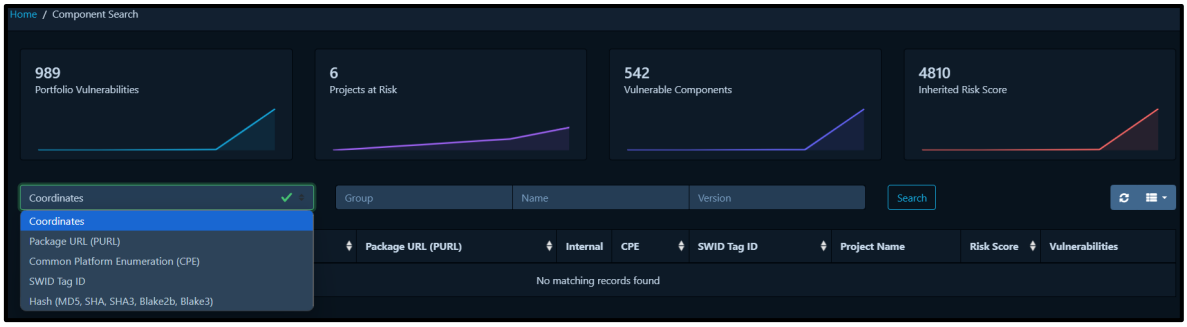


Ilustración 27: Components.

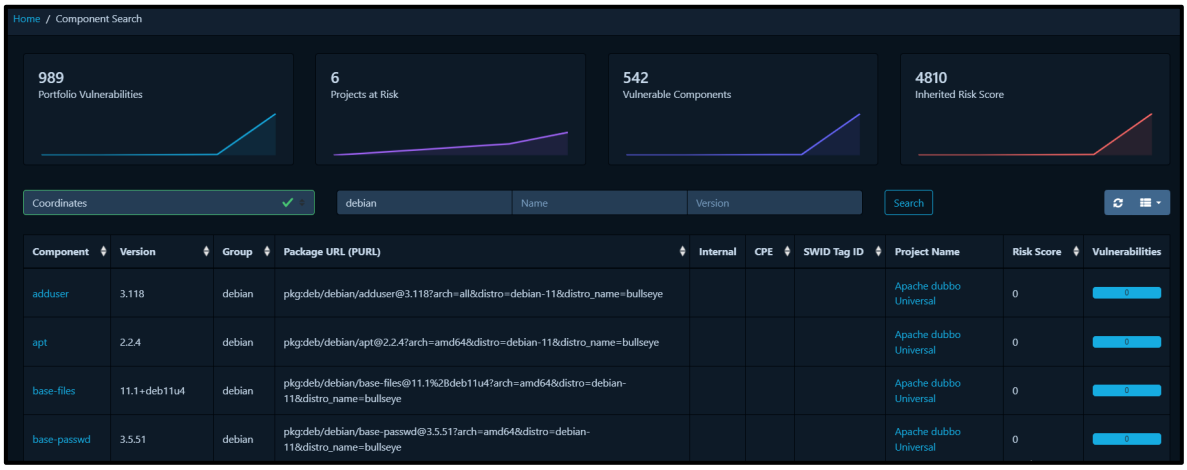


Ilustración 28: Components.

Dispone de un buscador para poder filtrar por "*Coordinates*", "*Package URL (PURL)*", "*Common Platform Enumerarion (CPE)", "SWID Tag ID"* y "Hash", además del grupo, nombre y versión. En el ejemplo anterior, se filtra por "*Coordinates"* y el grupo "*debian*" mostrando información relevante de la búsqueda realizada. Se puede obtener información más detallada de cada uno de los componentes pinchando en él.

8.3.3.3. Vulnerabilities

Este apartado **muestra todas las vulnerabilidades** analizadas en los proyectos.

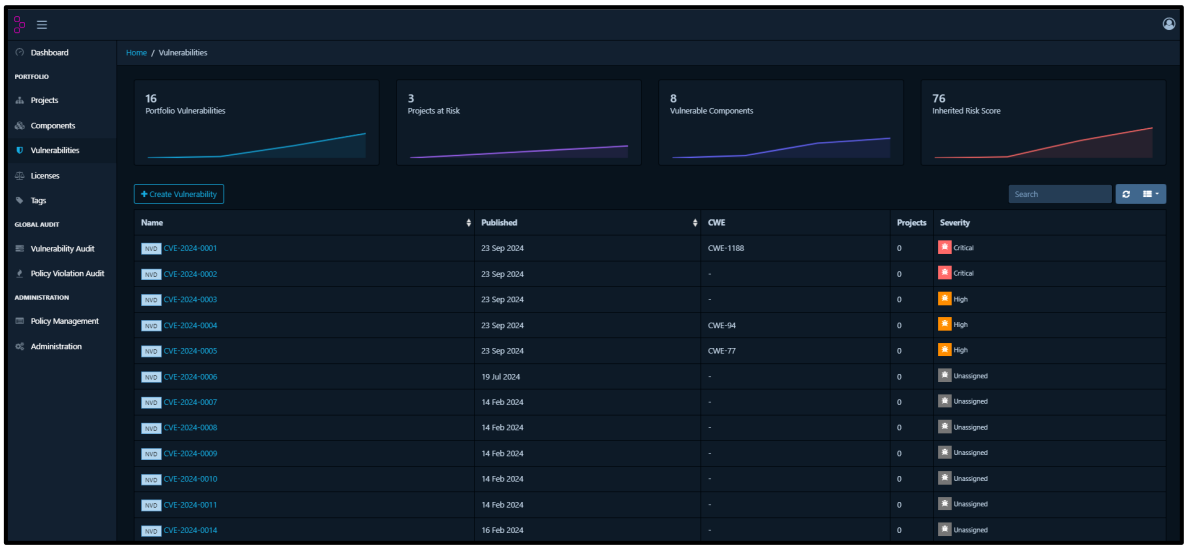


Ilustración 29: Vulnerabilities.

Como se puede ver en la Ilustración 30, aparecen todas las vulnerabilidades guardadas en el servidor de la aplicación. Muestra información relevante como la criticidad y el número de proyectos en la que está presente la vulnerabilidad. También dispone de un buscador sencillo para buscar vulnerabilidades por el nombre.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

8.3.3.4. Licenses y tags

Estos dos apartados son muy parecidos, en uno se muestran las licencias detectadas y en el otro los tags creados para identificar los proyectos. En ambos casos se muestran los resultados en una tabla.

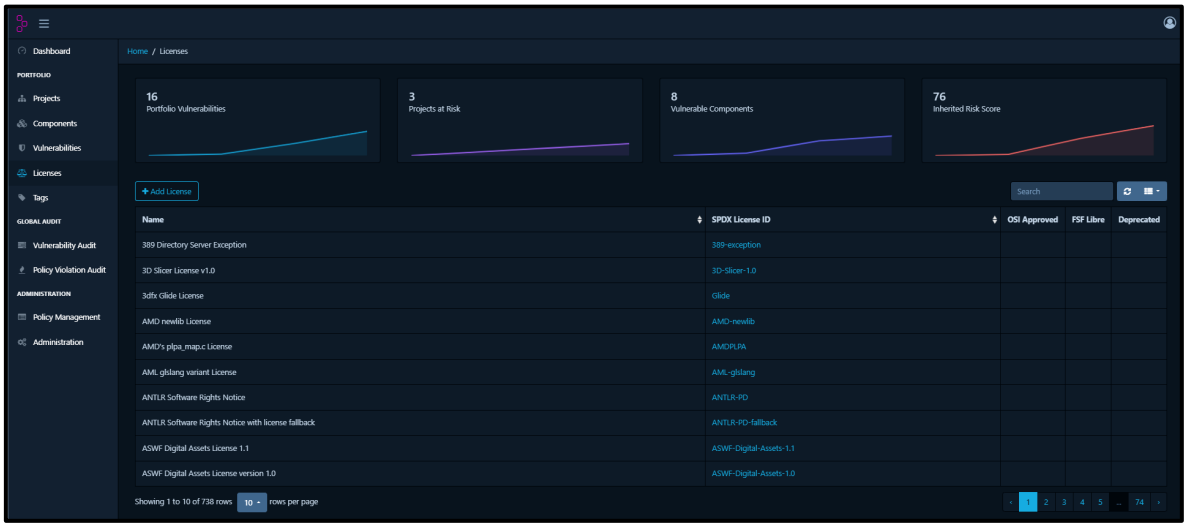


Ilustración 30: Licenses.

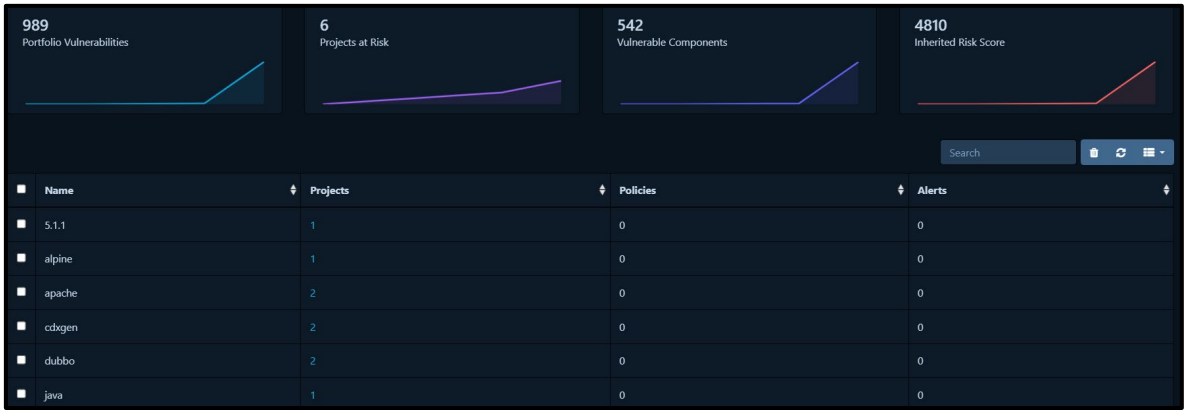


Ilustración 31: Tags.

8.3.3.5. Vulnerability Audit

En esta sección se puede hacer una **búsqueda exhaustiva de las vulnerabilidades más importantes**. Es un listado de todas las vulnerabilidades encontradas y se pueden configurar varios filtros como se ve en la siguiente imagen.

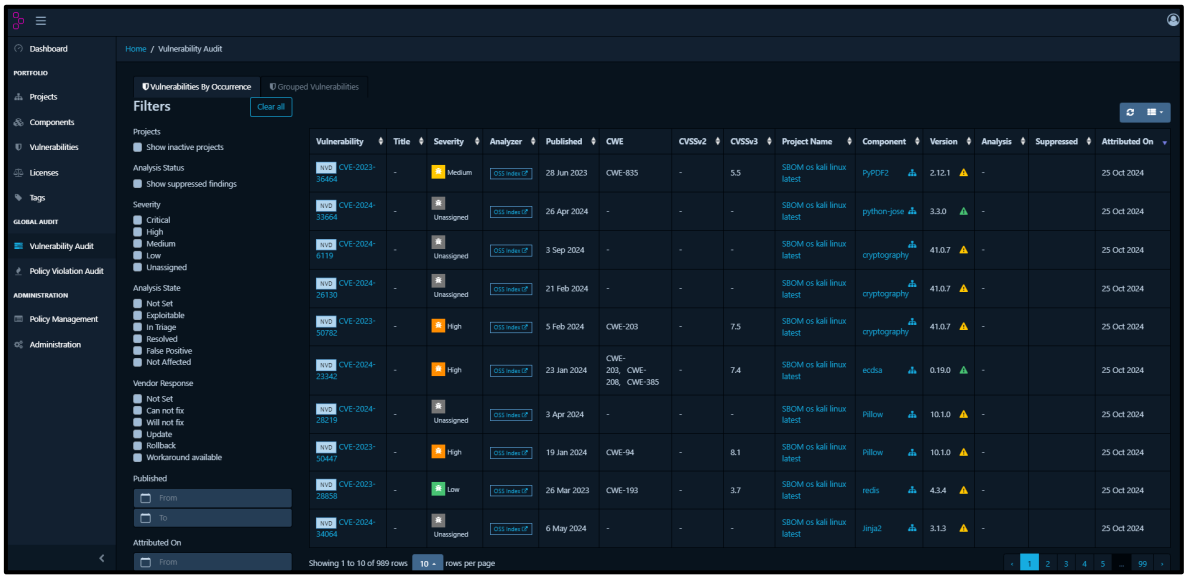


Ilustración 32: Vulnerability Audit.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

9. Conclusiones

A lo largo de este estudio, se ha podido observar la importancia que puede llegar a tener en la ciberseguridad industrial implementar exitosamente los SBOM. Su incorporación a los inventarios de activos permite obtener una visión clara de los componentes de la infraestructura y, por tanto, mejorar la eficacia en la identificación de sistemas vulnerables.

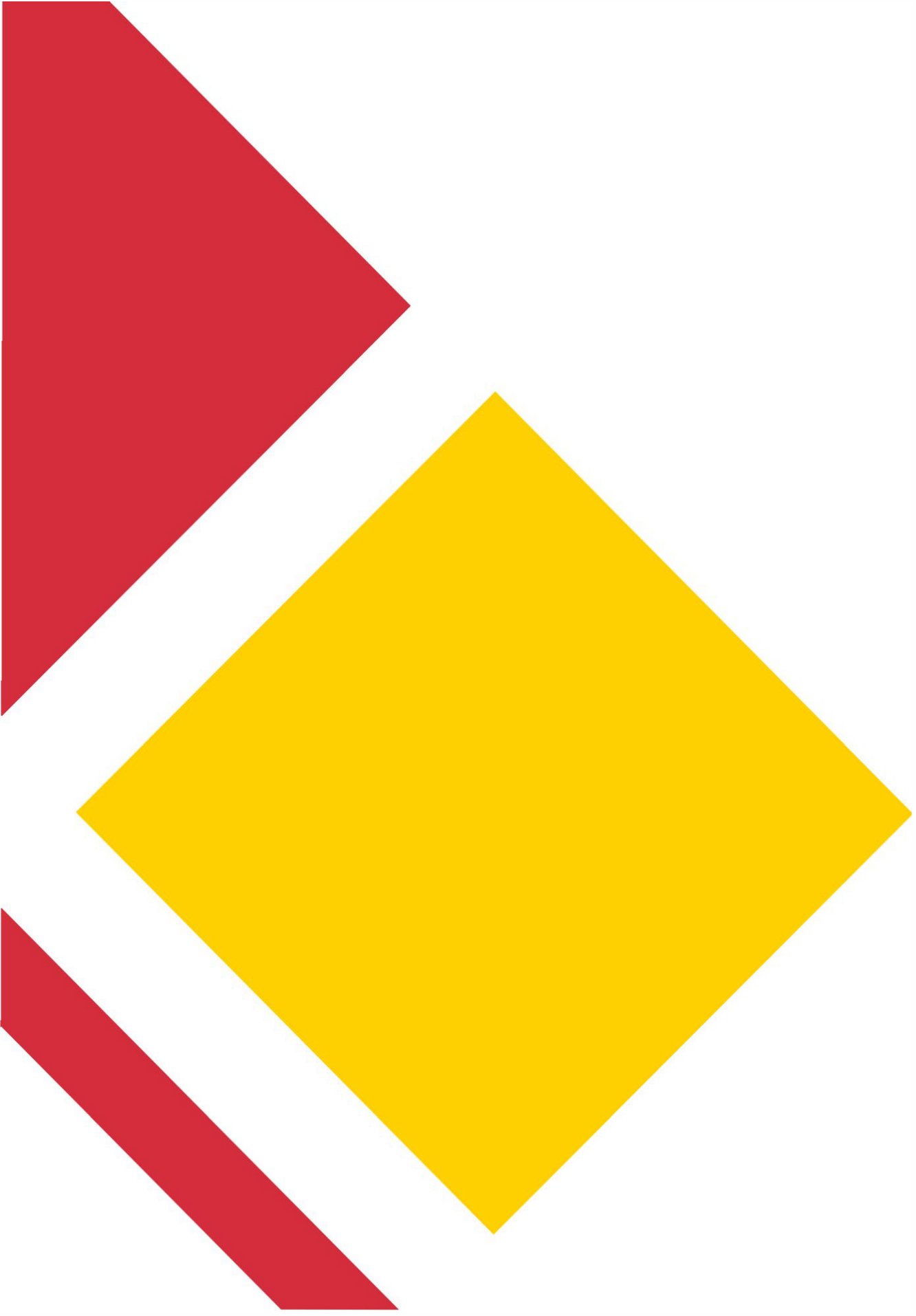
Se ha introducido el concepto de los SBOM, dando a conocer las ventajas y las problemáticas que pueden surgir de su implementación en entornos industriales.

Además, se han explicado varias herramientas para la creación y lectura de documentos SBOM. Estas herramientas permiten automatizar de forma sencilla las tareas de recopilación y análisis con gran precisión, haciendo más fácil y eficaz la implementación de los SBOM en entornos industriales.

Por ello, este estudio pretende servir de ayuda, para que el lector disponga de los conocimientos necesarios sobre los SBOM y las herramientas expuestas, facilitando y animando su uso para su beneficio.

10. Referencias

Referencia	Título, autor, fecha y enlace web
[Ref.- 1]	Documentación sobre SBOM por CISA URL: https://www.cisa.gov/sbom
[Ref.- 2]	SBOM en IoT y dispositivos OT URL: https://www.iotsecurityfoundation.org/wp-content/uploads/2023/02/RELEASE-2022-02-19-IoTSF-SBOM-whitepaper-v1-1-0.pdf
[Ref.- 3]	Buenas prácticas URL: https://www.kiuwan.com/blog/a-guide-to-sbom-best-practices-and-fundamentals/
[Ref.- 4]	Common security Advisory Framework (CSAF) URL: https://oasis-open.github.io/csaf-documentation/
[Ref.- 5]	OpenVex URL: https://edu.chainguard.dev/open-source/sbom/what-is-openvex/
[Ref.- 6]	Documentación Dependency Track URL: https://docs.dependencytrack.org/
[Ref.- 7]	Documentación cdxgen URL: https://github.com/CycloneDX/cdxgen
[Ref.- 8]	Documentación Syft URL: https://github.com/anchore/syft
[Ref.- 9]	Centro herramientas CycloneDX URL: https://cyclonedx.org/tool-center/
[Ref.- 10]	Estándar CycloneDX URL: https://cyclonedx.org/
[Ref.- 11]	Estándar SPDX URL: https://spdx.github.io/spdx-spec/v3.0.1/
[Ref.- 12]	Comunidad OWASP URL: https://owasp.org/



Financiado por la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES

Plan de Recuperación, Transformación y Resiliencia

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

TLP:CLEAR