

**ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA**
Universidad de Córdoba



DOCUMENTACIÓN TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Manual de Usuario

Autor

Emilio García Gutiérrez

Director

Juan Antonio Romero del Castillo

Junio, 2023



UNIVERSIDAD DE CÓRDOBA

Autor del Documento

Emilio García Gutiérrez

Contacto

Correo Electrónico: i92gague@uco.es

Versión del Documento 1.0

Fecha: 9-06-2023

Licencia del Documento

CopyRight © 2023, Emilio García Gutiérrez

Este documento está licenciado bajo la Licencia de Documentación Libre de GNU (GFDL) versión 1.3, 3 Noviembre 2008

Usted es libre de:

- Copiar, distribuir y comunicar públicamente la obra.
- Hacer obras derivadas.

Bajo las condiciones siguientes:

- Reconocer los créditos de la obra de la manera especificada por el autor o el licenciador.
- Compartir bajo la misma licencia.
- Indicar claramente los cambios realizados y proporcionar una copia actualizada del documento modificado bajo la misma licencia.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.

Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor.

Nada en esta licencia menoscaba o restringe los derechos morales del autor. Para ver la licencia completa, visite: <https://www.gnu.org/licenses/fdl-1.3.html>

Aviso legal

Las Marcas, logotipos y nombres comerciales aparecidos en este documento son propiedad de sus respectivos dueños.

Índice

1	Introducción	1
1.1	Propósito del Programa	1
1.2	Audiencia Objetivo	1
1.3	Requisitos Previos	1
2	Interfaz	3
2.1	Inicio	3
3	Ejemplos de Uso	5
3.1	Añadir Proyectos	5
3.2	Eliminar Proyectos	6
3.3	Realizar Análisis	7
3.3.1	Análisis Recursivo	7
3.3.2	Análisis No Recursivo	8
3.4	Comparar Análisis	9
3.5	Ver Vulnerabilidades	10
4	Desinstalación	12

Índice de figuras

1	Ventana Inicial de Dependency Evaluator	3
2	Iconos de Acceso Rápido	4
3	Formulario para Añadir un Proyecto	5
4	Listado de Proyectos	5
5	Eliminación de un Proyecto	6
6	Proyecto Eliminado	6
7	Seleccionar Proyecto	7
8	Realizar Análisis Recursivo	7
9	Resultado Análisis Recursivo	8
10	Seleccionar Análisis No Recursivo	8
11	Seleccionar Fichero	8
12	Realizar Análisis No Recursivo	9
13	Resultado Análisis No Recursivo	9
14	Comparar análisis	9
15	Resultado Análisis No Recursivo	10
16	Abrir Vulnerabilidad	10
17	Detalles de una Vulnerabilidad en OSV	11

1 Introducción

Bienvenido al manual de usuario de la aplicación Dependency Evaluator. Este manual tiene como objetivo proporcionar una guía completa sobre cómo utilizar el programa de manera efectiva y aprovechar al máximo sus funcionalidades.

1.1 Propósito del Programa

El programa Dependency Evaluator trata de una aplicación desarrollada en Python que tiene como objetivo informar al usuario de las vulnerabilidades halladas en los proyectos que especifique, poniendo en riesgo su seguridad. A continuación, se destacan algunas de las características principales de Dependency Evaluator:

- **Análisis de Proyectos:** Análisis completo de un proyecto de programación, con el fin de hallar las vulnerabilidades que pongan en riesgo la seguridad del mismo.
- **Análisis de Ecosistemas:** En proyectos en los que se utilicen más de un lenguaje de programación puede parecer más óptimo realizar análisis por separado, filtrando las vulnerabilidades por ecosistemas del proyecto.
- **Comparación de Análisis:** Los análisis tienen como objetivo la corrección de dependencias para desarrollar softwares seguros, comparando los análisis, veremos la evolución en cuanto a seguridad dentro del proyecto.

1.2 Audiencia Objetivo

Esta aplicación está diseñada para usuarios que realicen proyectos de programación y quieran garantizar la seguridad del software principalmente, sin embargo puede ser utilizada por usuarios con cualquier nivel de experiencia ya que no será de vital importancia. El sistema se ha desarrollado bajo una interfaz intuitiva y amigable para facilitar su uso.

1.3 Requisitos Previos

Antes de comenzar a utilizar Dependency Evaluator ([descárguelo aquí](#)), asegúrate de cumplir con los siguientes requisitos:

- **Intérprete de Python:** Dado que la aplicación está desarrollada en este lenguaje de programación en sí, es necesario haber realizado una instalación previa de su intérprete. Se recomienda como mínimo tener la versión 3.8.0.
- **Dependencias:** El instalador de paquetes de Python debe tener instalado la dependencia Pillow (9.5.0). Para comprobarlo puede ejecutar:

```
pip3 freeze
```

La salida sería la siguiente:

```
Pillow==9.5.0
```

Si no, realice su instalación con:

```
pip3 install pillow
```

- **OSV-Scanner**

Descarga y poseer en la misma ruta el ejecutable de OSV-Scanner, nombrado como osv-scanner. Además, otorgar permisos de ejecución junto con el ejecutable de la aplicación, en sistemas Unix puede realizarlo de la siguiente manera:

```
chmod 755 osv-scanner
```

Si todos los pasos de verificación se completan sin problemas, puedes estar seguro de que Dependency Evaluator puede ser ejecutado correctamente en su sistema, para ello, en la terminal:

```
./dependency-evaluator
```

¡Felicitaciones! Ahora que has instalado Dependency Evaluator con éxito, puedes pasar a la siguiente sección para aprender cómo utilizarlo y aprovechar todas sus funcionalidades.

2 Interfaz

2.1 Inicio

La aplicación nada más iniciarla mostrará la siguiente ventana:



Figura 1: Ventana Inicial de Dependency Evaluator

Como podremos observar, tenemos en la parte superior un menú que proporciona organización de algunas características y acceso rápido ciertas funcionalidades:

- **Archivo:** Aquí encontramos las funciones que tienen relación con los proyectos (Añadir proyectos, abrir un proyecto ya registrado y cerrar un proyecto).

- *Nuevo Proyecto:* Nos abrirá una ventana emergente para realizar un formulario acerca del proyecto que queremos añadir.
- *Abrir Proyecto:* Mostrará un explorador de archivos para seleccionar el proyecto que queremos abrir (no funcionará si no está registrado).
- *Cerrar Proyecto:* Si hemos abierto un proyecto, ya sea viendo sus análisis y respectivas vulnerabilidades, saldrá del mismo y mostrará el registro total de proyectos.

A su vez, también dispondremos de atajos para cada opción del submenú:

- * *Control+N:* Añadir un proyecto.
 - * *Control+O:* Abrir un proyecto existente.
 - * *Control+Q:* Cierre del proyecto.
- **Ver:** Opciones para cambiar entre diferentes vistas.
 - *Inicio:* Volver al Inicio

- *Proyectos*: Mostrar el registro de proyectos.
- **Ayuda**: Si el usuario tiene algún problema o alguna duda.
 - *Enlaces*: Redireccionar a las páginas web relacionadas con OSV, su base de datos y su software.
 - * Proyecto OSV
 - * Base de Datos Vulnerabilidades
 - * Github OSV
 - *Acerca de*: Información sobre el software, autor y enlace al manual de usuario.
 - * Propiedad Intelectual
 - * Manual de Usuario
- **Iconos de Funciones Más Utilizadas**:

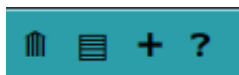


Figura 2: Iconos de Acceso Rápido

En orden, izquierda a derecha corresponde a:

- Inicio
- Registro de Proyectos
- Añadir un Proyecto
- Propiedad Intelectual

3 Ejemplos de Uso

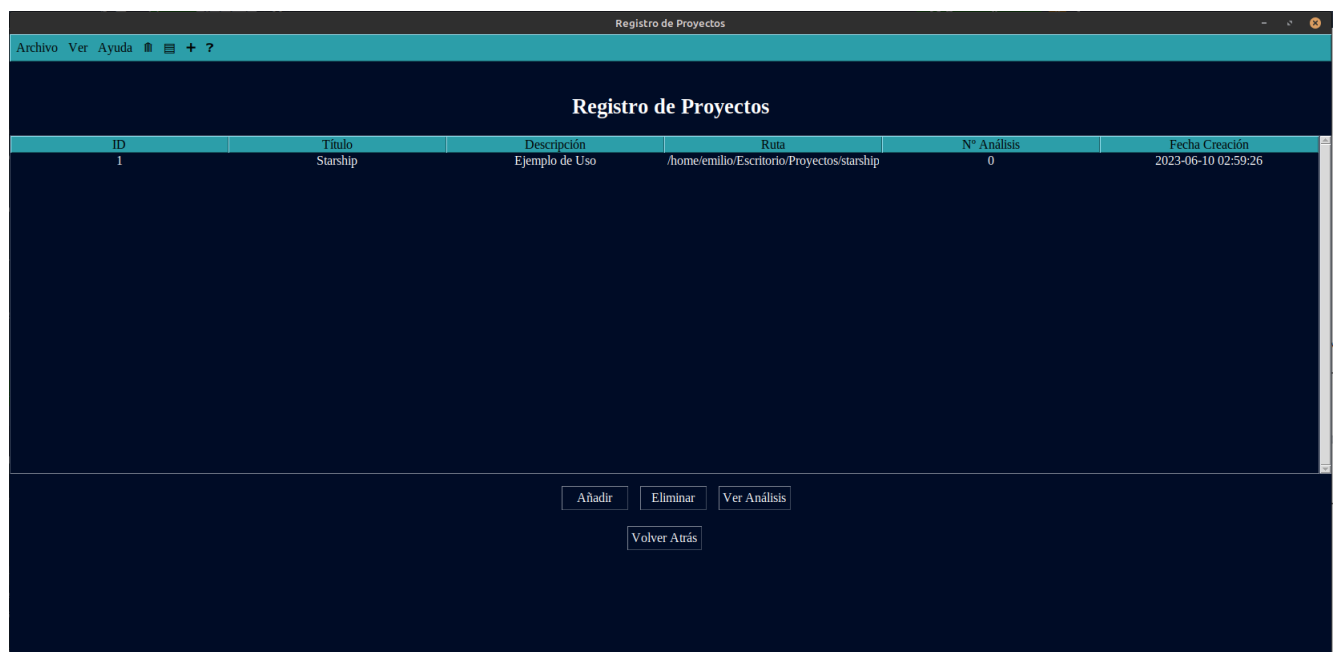
3.1 Añadir Proyectos

En el menú, seleccionaremos del submenú de Archivo la primera opción (Nuevo Proyecto), esta acción se puede realizar con su icono de acceso rápido o con Control+N. Al momento, nos saldrá una ventana emergente para realizar el formulario de registro del proyecto que queremos añadir, lo rellenamos y hacemos click en 'Añadir Proyecto':



Figura 3: Formulario para Añadir un Proyecto

Una vez realizado, nos que mostrará el nuevo proyecto se ha añadido:



Registro de Proyectos					
ID	Título	Descripción	Ruta	Nº Análisis	Fecha Creación
1	Starship	Ejemplo de Uso	/home/emilio/Escritorio/Proyectos/starship	0	2023-06-10 02:59:26

Figura 4: Listado de Proyectos

3.2 Eliminar Proyectos

En este caso, necesitaremos tener un proyecto añadido al menos. Siguiendo con el ejemplo anterior, seleccionamos el proyecto y hacemos click en el botón 'Eliminar' (también es posible haciendo click izquierdo encima del proyecto tras su selección):

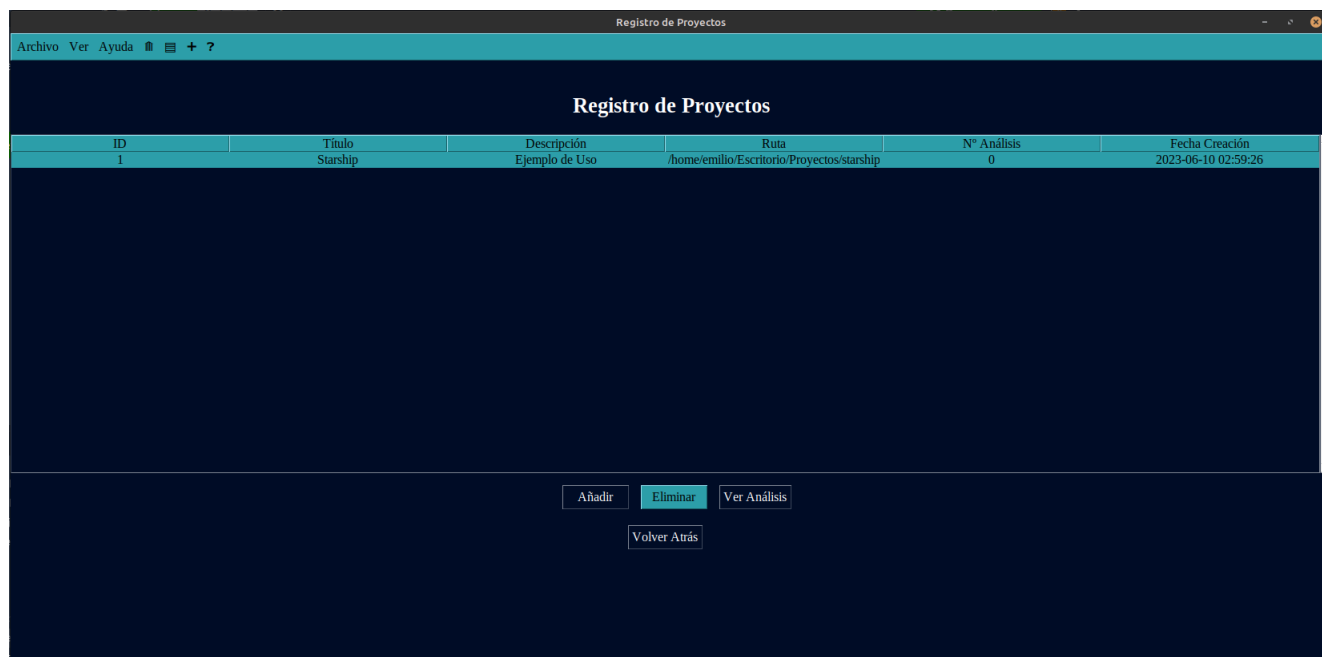


Figura 5: Eliminación de un Proyecto

Tras esto, el proyecto se habrá eliminado:

ID	Título	Descripción	Ruta	N° Análisis	Fecha Creación

Figura 6: Proyecto Eliminado

3.3 Realizar Análisis

Una vez que tenemos un proyecto registrado, vamos a realizar los distintos análisis que se pueden hacer en él. Primeramente, debemos abrir el proyecto a analizar, seleccionando el proyecto y seguidamente 'Ver Análisis':

ID	Título	Descripción	Ruta	N° Análisis	Fecha Creación
2	Starship	Ejemplo de Uso	/home/emilio/Escritorio/Proyectos/starship	0	2023-06-10 12:35:38

[Añadir](#) [Eliminar](#) [Ver Análisis](#)
[Volver Atrás](#)

Figura 7: Seleccionar Proyecto

3.3.1 Análisis Recursivo

Análisis completo del proyecto el cual devolverá todas las vulnerabilidades que se han encontrado en el mismo. Esta opción vendrá por defecto, por lo que haremos click en 'Realizar Análisis Recursivo':

Análisis del Proyecto					
ID del Proyecto: 2		Título del Proyecto: Starship		Descripción del Proyecto: Ejemplo de Uso	
Ruta del Proyecto: /home/emilio/Escritorio/Proyectos/starship		N° Análisis Realizados: 0		Fecha Creación Proyecto: 2023-06-10 12:35:38	
ID	Archivo	N° de Vulnerabilidades	Fecha	Peligrosidad	

☐ Análisis Recursivo

☒ Realizar Análisis Recursivo

[Ver Vulnerabilidades](#) [Eliminar Análisis](#) [Descargar Análisis.json](#) [Comparar Análisis
\(Seleccionar 2 con Ctrl+Click\)](#)
[Volver Atrás](#)

Figura 8: Realizar Análisis Recursivo

Veremos que al terminar el proceso se añadirá un análisis a la tabla:

ID	Archivo	Nº de Vulnerabilidades	Fecha	Peligrosidad
1	starship.json	33	2023-06-10 12:49:27	Alta

Figura 9: Resultado Análisis Recursivo

3.3.2 Análisis No Recursivo

Análisis donde elegiremos el ecosistema que queremos analizar, útil en proyectos donde trabajamos con diferentes lenguajes de programación y no queremos realizar el análisis completo. Cambiaremos la lista de opciones a 'Análisis No Recursivo':



Figura 10: Seleccionar Análisis No Recursivo

Ahora, necesitaremos seleccionar el archivo del ecosistema que queremos analizar, en nuestro caso, elegiremos el archivo Cargo.lock de Rust.:

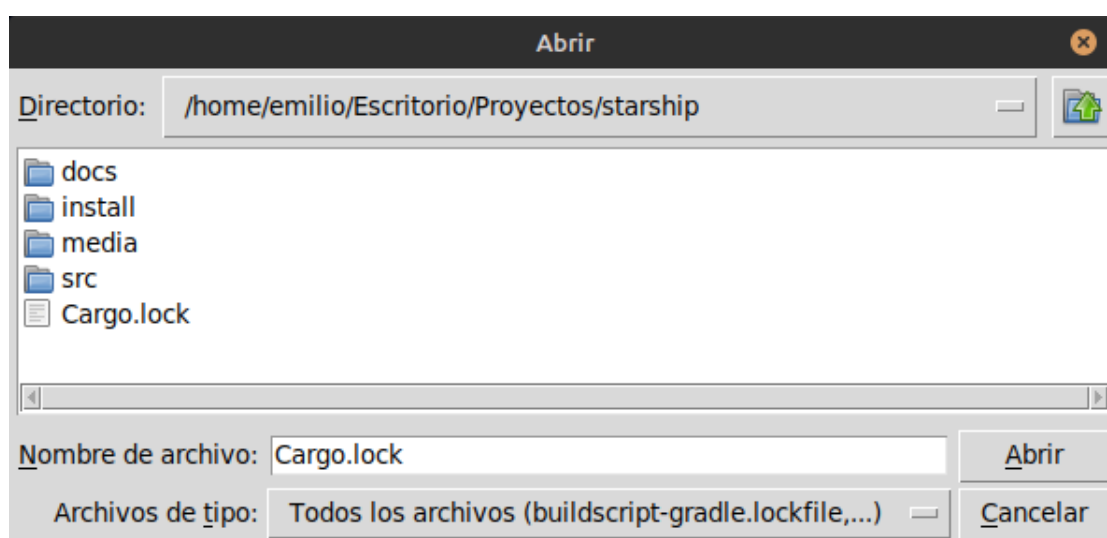


Figura 11: Seleccionar Fichero

*Figura 12: Realizar Análisis No Recursivo*

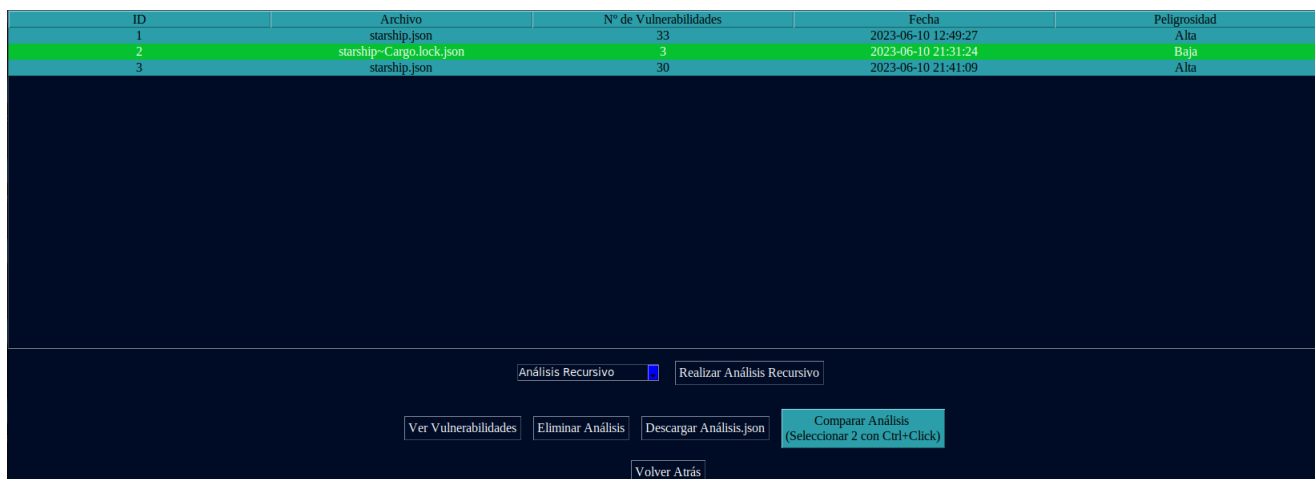
Como resultado se añadirá otro análisis, se puede observar que en este ecosistema encontramos menos vulnerabilidades, por lo que debe ser otro el que presente más dependencias con fallas:

ID	Archivo	Nº de Vulnerabilidades	Fecha	Peligrosidad
1	starship.json	33	2023-06-10 12:49:27	Alta
2	starship~Cargo.lock.json	3	2023-06-10 21:31:24	Baja

Figura 13: Resultado Análisis No Recursivo

3.4 Comparar Análisis

Tras resolver los paquetes con problemas de seguridad, realizamos otro análisis y procedemos a hacer una comparativa. Para ello, seleccionamos dos análisis con la tecla control mantenida y presionamos 'Comparar Análisis':



ID	Archivo	Nº de Vulnerabilidades	Fecha	Peligrosidad
1	starship.json	33	2023-06-10 12:49:27	Alta
2	starship~Cargo.lock.json	3	2023-06-10 21:31:24	Baja
3	starship.json	30	2023-06-10 21:41:09	Alta

Figura 14: Comparar análisis

Nos mostrará ambos análisis con sus vulnerabilidades y la evolución de las mismas (Nuevas, corregidas y que se han mantenido):

Comparación de Análisis							
Análisis con ID: 1							
ID	Paquete	Ecosistema	Resumen	Detalles	Rango afectado	Versión introducida	Versión corregida
GHSA-qvc4-78gw-pv8p	enumflags2	crates.io	Adversarial use of 'make_bitfla	The macro relied on an express	0.7.0-Before Actual Version	0.7.0	0.7.7
RUSTSEC-2023-0035	enumflags2	crates.io	Adversarial use of 'make_bitfla	The macro relied on an express	0.7.0-Before Actual Version	0.7.0	0.7.7
RUSTSEC-2020-0168	mach	crates.io	mach is unmaintained	Last release was almost 4 years	0.0.0	0.0.0	N/A
GHSA-93q8-gg69-wqmw	ansi-regex	npm	Inefficient Regular Expression	ansi-regex is vulnerable to Inef	6.0.0-Before Actual Version	6.0.0	6.0.1
GHSA-93q8-gg69-wqmw	ansi-regex	npm	Inefficient Regular Expression	ansi-regex is vulnerable to Inef	6.0.0-Before Actual Version	6.0.0	6.0.1
GHSA-lwr7-v2mv-hh25	async	npm	Prototype Pollution in async	A vulnerability exists in Async	3.0.0-Before Actual Version	3.0.0	3.2.2
GHSA-w573-4hg7-7wgg	decode-uri-component	npm	decode-uri-component vulneral	decode-uri-component 0.2.0 is	0-Before Actual Version	0	0.2.1
Resultados							
Se han hallado 0 nuevas vulnerabilidades:							
Se han corregido 3 vulnerabilidades: GHSA-qvc4-78gw-pv8p, RUSTSEC-2023-0035, RUSTSEC-2020-0168							
Se han mantenido 25 o más vulnerabilidades: GHSA-tp65-9cf3-cjxr, GHSA-w573-4hg7-7wgg, GHSA-cfm4-qh2-4765, GHSA-4wf5-vphf-c2xc, GHSA-hrpp-h998-j3pp, GHSA-8fr3-hfg3-ggpp, GHSA-p8p7-x288-28g6, GHSA-3rfm-jhwj-7488, GHSA-76							
Análisis con ID: 3							
ID	Paquete	Ecosistema	Resumen	Detalles	Rango afectado	Versión introducida	Versión corregida
GHSA-93q8-gg69-wqmw	ansi-regex	npm	Inefficient Regular Expression	ansi-regex is vulnerable to Inef	6.0.0-Before Actual Version	6.0.0	6.0.1
GHSA-93q8-gg69-wqmw	ansi-regex	npm	Inefficient Regular Expression	ansi-regex is vulnerable to Inef	6.0.0-Before Actual Version	6.0.0	6.0.1
GHSA-fwr7-v2mv-hh25	async	npm	Prototype Pollution in async	A vulnerability exists in Async	3.0.0-Before Actual Version	3.0.0	3.2.2
GHSA-w573-4hg7-7wgg	decode-uri-component	npm	decode-uri-component vulneral	decode-uri-component 0.2.0 is	0-Before Actual Version	0	0.2.1
GHSA-6h5x-7c5m-7cr7	eventsources	npm	Exposure of Sensitive Informat	When fetching an url with a lin	0-Before Actual Version	0	1.1.1
GHSA-ww39-953v-wcq6	glob-parent	npm	glob-parent before 5.1.2 vulner	This affects the package glob-p	0-Before Actual Version	0	5.1.2
GHSA-pfrx-2q88-qc97	got	npm	Got allows a redirect to a UNID	The got package before 11.8.5	12.0.0-Before Actual Version	12.0.0	12.1.0
Ver en OSV Vulnerability Database							
Volver Atrás							

Figura 15: Resultado Análisis No Recursivo

3.5 Ver Vulnerabilidades

Si queremos ver las vulnerabilidades de un análisis, lo abriremos y seguidamente nos listará las vulnerabilidades halladas. Seleccionamos una vulnerabilidad y seleccionamos 'Ver en OSV Vulnerability Database' paar ver la información detallada:

Vulnerabilidades Halladas							
ID	Paquete	Ecosistema	Resumen	Detalles	Rango afectado	Versión introducida	Versión corregida
GHSA-qvc4-78gw-pv8p	enumflags2	crates.io	Adversarial use of 'make_bitfla	The macro relied on an express	0.7.0-Before Actual Version	0.7.0	0.7.7
RUSTSEC-2023-0035	enumflags2	crates.io	Adversarial use of 'make_bitfla	The macro relied on an express	0.7.0-Before Actual Version	0.7.0	0.7.7
RUSTSEC-2020-0168	mach	crates.io	mach is unmaintained	Last release was almost 4 years	0.0.0	0.0.0	N/A
Ver en OSV Vulnerability Database							
Volver Atrás							

Figura 16: Abrir Vulnerabilidad

GHSA-qvc4-78gw-pv8p

Source <https://github.com/github/advisory-database/blob/main/advisories/github-reviewed/2023/04/GHSA-qvc4-78gw-pv8p/GHSA-qvc4-78gw-pv8p.json>

Published 2023-04-24T16:47:24Z

Modified 2023-04-24T16:47:24Z

Details The macro relied on an expression of the form `Enum::Variant` always being a variant of the enum. However, it may also be an associated integer constant, in which case there's no guarantee that the value of said constant consists only of bits valid for this bitflag type.

Thus, code like this could create an invalid `BitFlags<Test>`, which would cause iterating over it to trigger undefined behavior. As the debug formatter internally iterates over the value, it is also affected.

```
use enumflags2::{bitflags, make_bitflags};

#[bitflags]
#[repr(u8)]
#[derive(Copy, Clone, Debug)]
enum Test {
    A = 1,
    B = 2,
}

impl Test {
    const C: u8 = 69;
}

fn main() {
    let x = make_bitflags!(Test::C);
    // printing or iterating over x is UB
}
```

References <https://github.com/meithecatte/enumflags2>

<https://github.com/meithecatte/enumflags2/releases/tag/v0.7.7>

<https://rustsec.org/advisories/RUSTSEC-2023-0035.html>

Figura 17: Detalles de una Vulnerabilidad en OSV

4 Desinstalación

La desinstalación completa de este software y sus archivos consistirá en la eliminación de:

- Ejecutable de *Dependency Evaluator*
- Ejecutable de *OSV-Scanner*
- Base de Datos *DependencyEval.db*
- Archivos JSON creados por la aplicación

Y hasta aquí sería el manual de usuario para utilizar Dependency Evaluator, espero que te haya sido útil para manejar esta herramienta.