Análisis de vulnerabilidades web con OWASP

Gabriela García

Líder OWASP Capítulo Ciudad de México

06 de octubre, 2023

Agenda

1. ¿Qué es OWASP?
2. OWASP Projects.
3. Conceptos básicos.
4. Laboratorio.

1. ¿Qué es?

OWASP

- ✓ **O**pen **W**orldwide **A**pplication **S**ecurity **P**roject.
- ✓ Fundación sin fines de lucro cuyo objetivo es mejorar la **seguridad** del **software**. Arrancó el 1ro de Diciembre, 2001.
- ✓ ¿Cómo?
  - ❑ Mediante sus **proyectos** de software de **código abierto** lidereados por su **comunidad**,
  - ❑ sus miembros,
  - ❑ sus **Capítulos** y **eventos** locales e internacionales.
- ✓ Todos los Proyectos, Documentos, Herramientas, Foros Y Capítulos son **gratuitos** y abiertos a todo interesado en fortalecer la seguridad de aplicaciones.

1. ¿Qué es?

OWASP Capítulos

- ✓ Los Capítulos locales buscan conformar comunidad de profesionales de seguridad informática en todo el mundo.
- ✓ Dirigidos por líderes locales conforme a políticas bien establecidas – Política de los capítulos.
- ✓ Desarrollan y educan a través de eventos y reuniones alrededor del mundo.
- ✓ Existe alrededor de 300 Capítulos.
- ✓ En México hay 4: Aguascalientes, Ciudad de México, Querétaro y Rivera Maya.

OWASP Projects

- Los OWASP *projects* son proyectos de **código abierto** y construidos por miembros de la **comunidad de voluntarios**.
- Existen proyectos de herramientas (*Code Projects*) y de documentación (*Documentation Projects*).
- Actualmente, ¡el inventario tiene 302 proyectos!
- Para mayor detalle de todos los OWASP *projects*, visitar https://owasp.org/projects/

OWASP Projects relacionados

http://www.quickmeme.com/

Proyectos de herramientas.

- ✓ Dependency check.
- ✓ Juice Shop.

Proyectos de documentación.

- ✓ **Web** Security Testing Guide (465pp) – Enlace.

  Guía de cómo aplicar la Metodología de Revisión de OWASP. Basado en el enfoque de caja negra. Lista de pruebas de seguridad – método de evaluación de la seguridad mediante la validación y verificación metodológica de los controles de seguridad. Incluye reporte de hallazgos.

- ✓ **Application** Security Verification Standard (74 pp). – Enlace.

  Marco de requisitos y controles de seguridad requeridos al diseñar, desarrollar y probar aplicaciones web y servicios web modernos. Define tres niveles de verificación de seguridad. Verificable y revisable.

Más OWASP Projects relacionados

- ✓ **Automated** Threats to Web Applications (80pp) – Enlace.

  Estándar de facto de la industria de detección y mitigación de amenazas automatizadas web (Escaner de vulnerabilidades).

- ✓ **OWASP Top** 10:2021 (aprox. 25pp, 2017) – Enlace.

  Lista los 10 principales riesgos de seguridad en aplicaciones web. Abarca las vulnerabilidades más comunes asociadas a los riesgos, medidas de prevención, escenarios de ataque de ejemplo, referencias. Principalmente para **concientización**.

- ✓ **OWASP API** Security Top 10:2023 (aprox. 31pp, 2019) – Enlace.

  Application Programming Interface (API). Estructura similar al OWASP Top 10. ¿Cómo saber si el API es vulnerable al riesgo?. Principalmente para **concientización**.

Más OWASP Projects relacionados

- ✓ **OWASP Top** 10 Privacy Risks:2021 – Enlace.

  Enfocado a riesgos de privacidad en aplicaciones web y sus controles. Provee información de como implementar privacidad por diseño. ¿Como revisar si la aplicación web es susceptible al riesgo?

- ✓ **OWASP Top** 10 Proactive Controls:2018 (40pp) – Enlace.

  Describe el top 10 de controles de seguridad que los desarrolladores deben incluir al desarrollar. Incluye descripción, mejores practices de implementación y las vulnerabilidades prevenidas.

- ✓ Y más.

Otros OWASP Projects

- OWASP Cloud-Native Application Security Top 10
- OWASP Desktop App Security Top 10
- OWASP Docker Top 10
- OWASP Low-Code/No-Code Top 10
- OWASP Machine Learning Security Top Ten
- OWASP Mobile Top 10
- OWASP TOP 10
- OWASP Top 10 CI/CD Security Risks
- OWASP Top 10 Client-Side Security Risks
- OWASP Top 10 Privacy Risks
- OWASP Serverless Top 10

https://cdn.acidcow.com/pics/20181231/tired_memes_27.jpg

5 minutos

Pan de cada día de OWASP

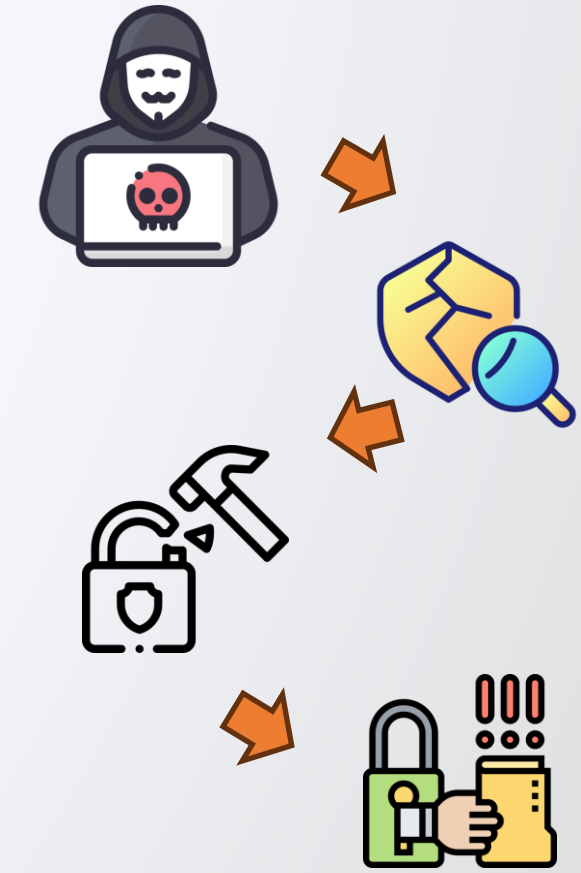
Vulnerabilidades:

- ❑ En aplicaciones web.
- ❑ En servicios web.
- ❑ En *APIs*.
- ❑ En dispositivos móviles.
- ❑ En *firmware*.
- ❑ eeeeetc.

http://www.quickmeme.com/meme/3tspp0

Vulnerabilidad

- Falla, debilidad, *flag* (bandera).
- ¿**Dónde** puede haber vulnerabilidades? En el **diseño** de un sistema, en su **implementación**, en su **operación** o en su **administración**.*
- ¿Qué se puede hacer con ella? **Explotarla** para **comprometer** los objetivos de seguridad del sistema.*



* Fuente: Web Security Testing Guide v4.2

Análisis de vulnerabilidades

- Es la identificación y validación de vulnerabilidades*.
- Se emplea para identificar y evaluar los riegos de seguridad que pudiera haber debido a las vulnerabilidades identificadas*.

Ejemplos de vulnerabilidades.

- Formulario sin captcha.
- Manejo inapropiado de errores.
- Validación inapropiada de datos de entrada en un *input*.

Lista de vulnerabilidades publicadas en OWASP:

https://owasp.org/www-community/vulnerabilities/

* Fuente: Penetration Testing Execution Standard (PTES

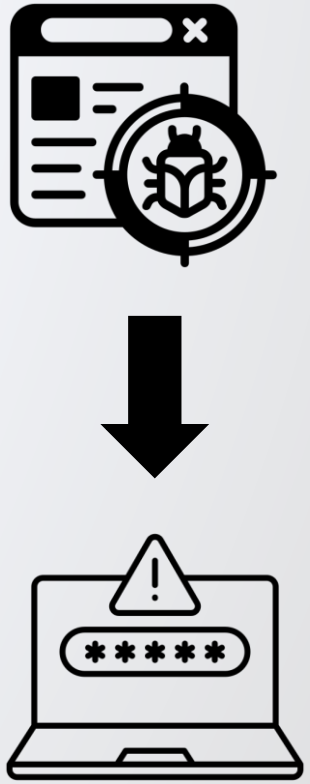
¿Cómo hacer un AVW?

Identificación.

- Herramientas automatizadas que buscan e identifican componentes, codificaciones vulnerables conocidas.
- De forma manual.
- Mediante pruebas de seguridad - Método de evaluación de la seguridad mediante la validación y verificación metodológica de los controles de seguridad*.

Validación.

- Para reducir las vulnerabilidades identificadas a solo las válidas.
- Verificando que la vulnerabilidad es explotable.



* Fuente: OWASP Web Security Testing Guide.

Análisis vs Gestión de vulnerabilidades

https://butterflylabs.com/wp-content/uploads/2019/07/report.jpg

- ✓ **Análisis** también puede encontrarse como: evaluación, valoración (*Analysis*, *asessment*).
- ✓ La **gestión** (*management*) de vulnerabilidades implica detección, reporteo, remediación de vulnerabilidades y mejora continua.*
  - OWASP Vulnerability Management Guide – Enlace.

* Fuente: OWASP Vulnerability Management Guide (OVMG) (2020)

Análisis de vuln vs *Pentesting*

- Pruebas de penetración o *Pentesting* son pruebas de seguridad donde el evaluador **imita** ataques informáticos reales sobre una aplicación, sistema o red, para **identificar** formas de atentar contra los **objetivos de seguridad** del mismo*.
- Se busca **explotar** vulnerabilidades para comprometer la aplicación, sus datos o sus recursos.
- El análisis de vulnerabilidades forma parte del *Pentesting*.
- Existen diversas metodologías de *Pentestings*.

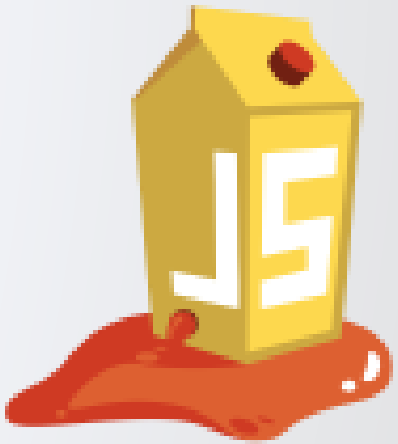
* Fuente: https://csrc.nist.gov/glossary/term/penetration_testing

4. Laboratorios

Importante: Contar con NodeJS 20.5 o superior y Docker Desktop 4.22 o superior, instalados.

4. Laboratoris

Juice Shop



- ✓ Aplicación web intencionalmente vulnerable.
- ✓ Es lo opuesto a las "mejores prácticas" de seguridad para los desarrolladores web.
- ✓ Más de **105 retos**, diferentes niveles de dificultad.
- ✓ Puede usarse como *target* de herramientas de seguridad, para *CTFs*.
- ✓ Abarca diversos riesgos o tipos de vulnerabilidades*.
- ✓ Documento detallado en formato PDF (422pp) u *online* – Enlace.

L1: Juice Shop – Instalación

Instalación con Node.js

https://owasp.org/www-project-juice-shop/

OWASP Juice Shop

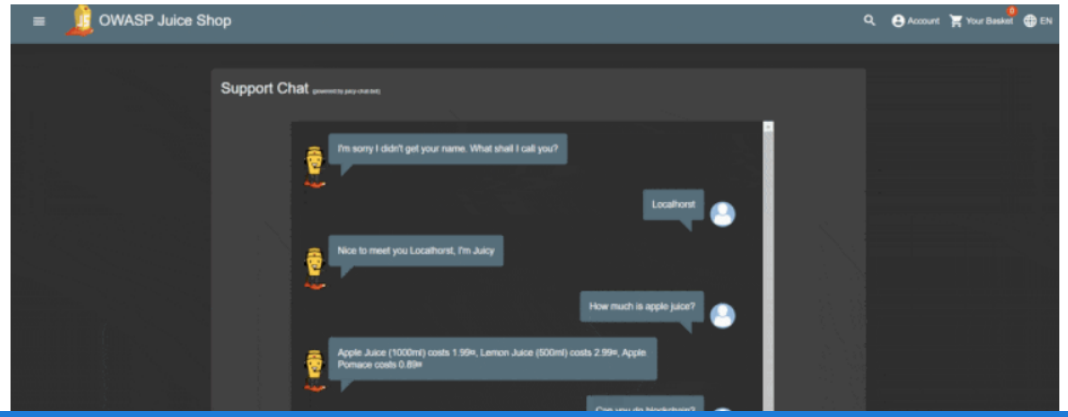
Watch 152 Star 8,734

Main | Overview | News | Challenges | Learning | CTF | Ecosystem | Supporters

owasp flagship project | release v15.2.1 | GitHub ★ 8.7k | Follow

openssf best practices gold | Contributor Covenant v2.0 adopted

OWASP Juice Shop is probably the most modern and sophisticated insecure web application! It can be used in security trainings, awareness demos, CTFs and as a guinea pig for security tools! Juice Shop encompasses vulnerabilities from the entire OWASP Top Ten along with many other security flaws found in real-world applications!



The OWASP® Foundation works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

Project Information

Flagship Project

Classification

Tool

Audience

Builder
Breaker
Defender

Installation

From Source
Packaged (GitHub/SourceForge)
Docker Image

L1: Juice Shop - Instalación

https://github.com/juice-shop/juice-shop#from-sources

README.md

Packaged Distributions

downloads 245k | sourceforge downloads 1.3k/month | sourceforge downloads 53k

1. Install a 64bit node.js on your Windows, MacOS or Linux machine
2. Download `juice-shop-<version>_<node-version>_<os>_x64.zip` (or `.tgz`) attached to latest release
3. Unpack and `cd` into the unpacked folder
4. Run `npm start`
5. Browse to http://localhost:3000

> Each packaged distribution includes some binaries for `sqlite3` and `libxmljs` bound to the OS and node.js version which `npm install` was executed on.

L1: Juice Shop - Instalación

Instalación con Node.js

From pre-packaged distribution

1. Install a 64bit Node.js on your Windows, MacOS or Linux machine.
2. Download `juice-shop-<version>_<node-version>_<os>_x64.zip` (or `.tgz`) attached to the latest release on GitHub.
3. Unpack the archive and run `npm start` in unpacked folder to launch the application
4. Browse to http://localhost:3000

https://github.com/juice-shop/juice-shop/releases/tag/v15.2.1

L1: OWASP Juice Shop – Instalación

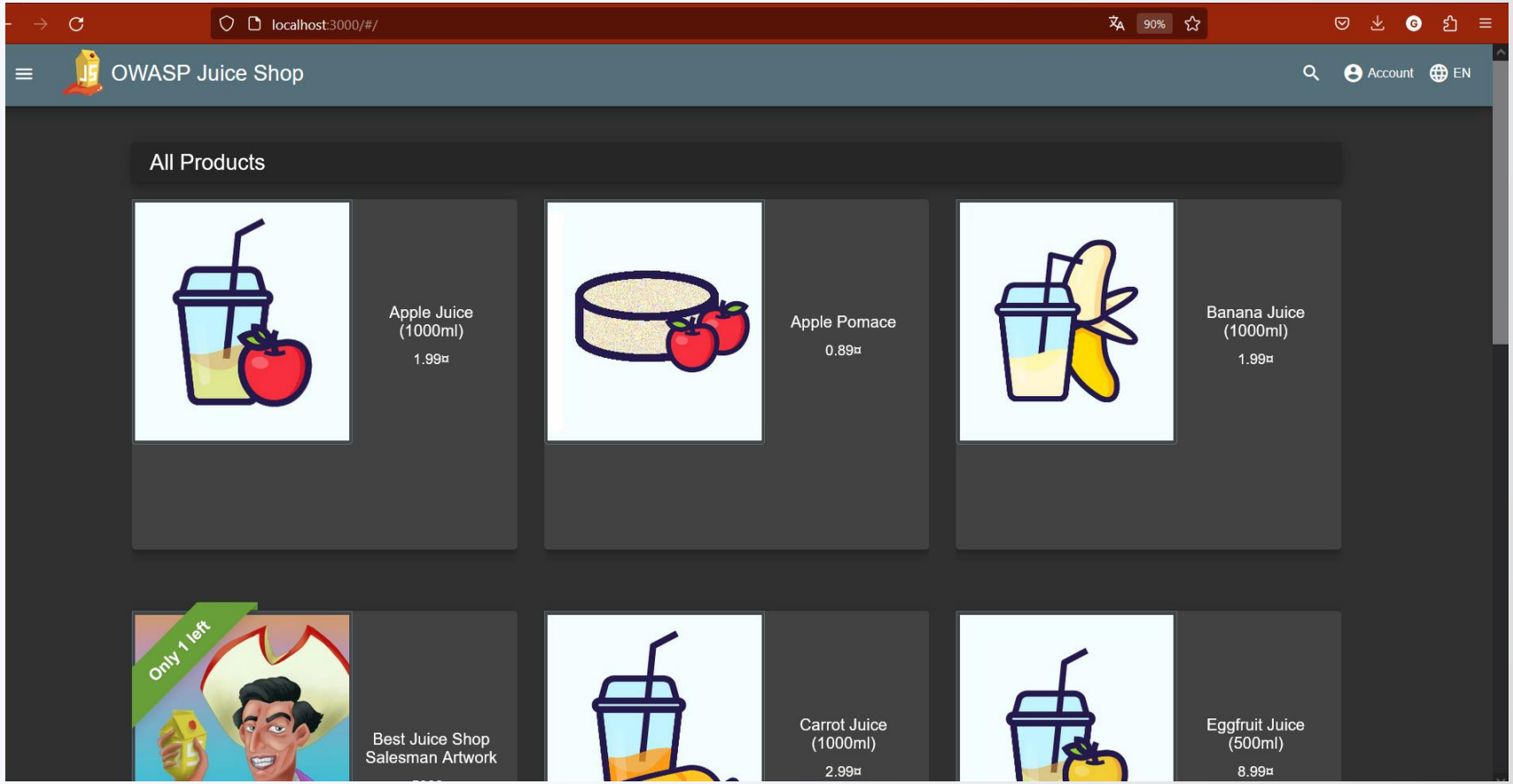
```
C:\Users\GAGAMO\Documents\14_INFOSEC\CSI23_tallerAVOWASP\juice-shop_15.2.1>npm start

> juice-shop@15.2.1 start
> node build/app

info: All dependencies in ./package.json are satisfied (OK)
info: Detected Node.js version v20.6.1 (OK)
info: Detected OS win32 (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 19 of 19 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file styles.css is present (OK)
info: Required file main.js is present (OK)
info: Required file polyfills.js is present (OK)
info: Required file runtime.js is present (OK)
info: Required file vendor.js is present (OK)
info: Port 3000 is available (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Server listening on port 3000
```

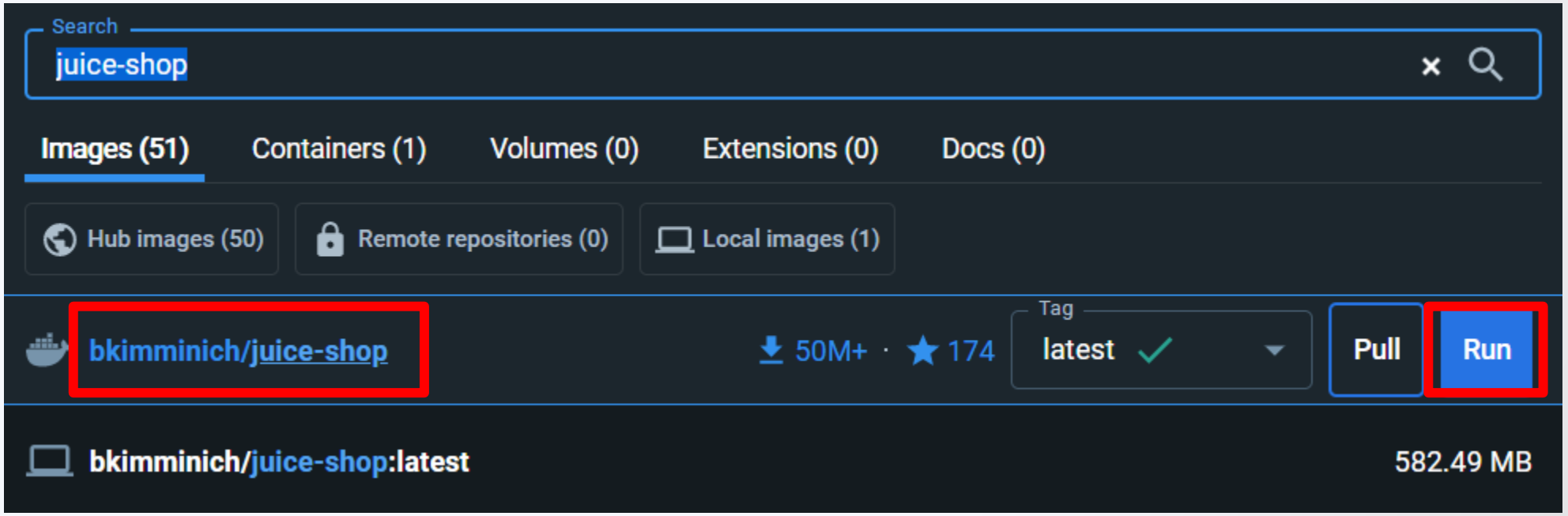
L1: OWASP Juice Shop – Instalación

Recomendación: Emplear Chrome para abrir JS.



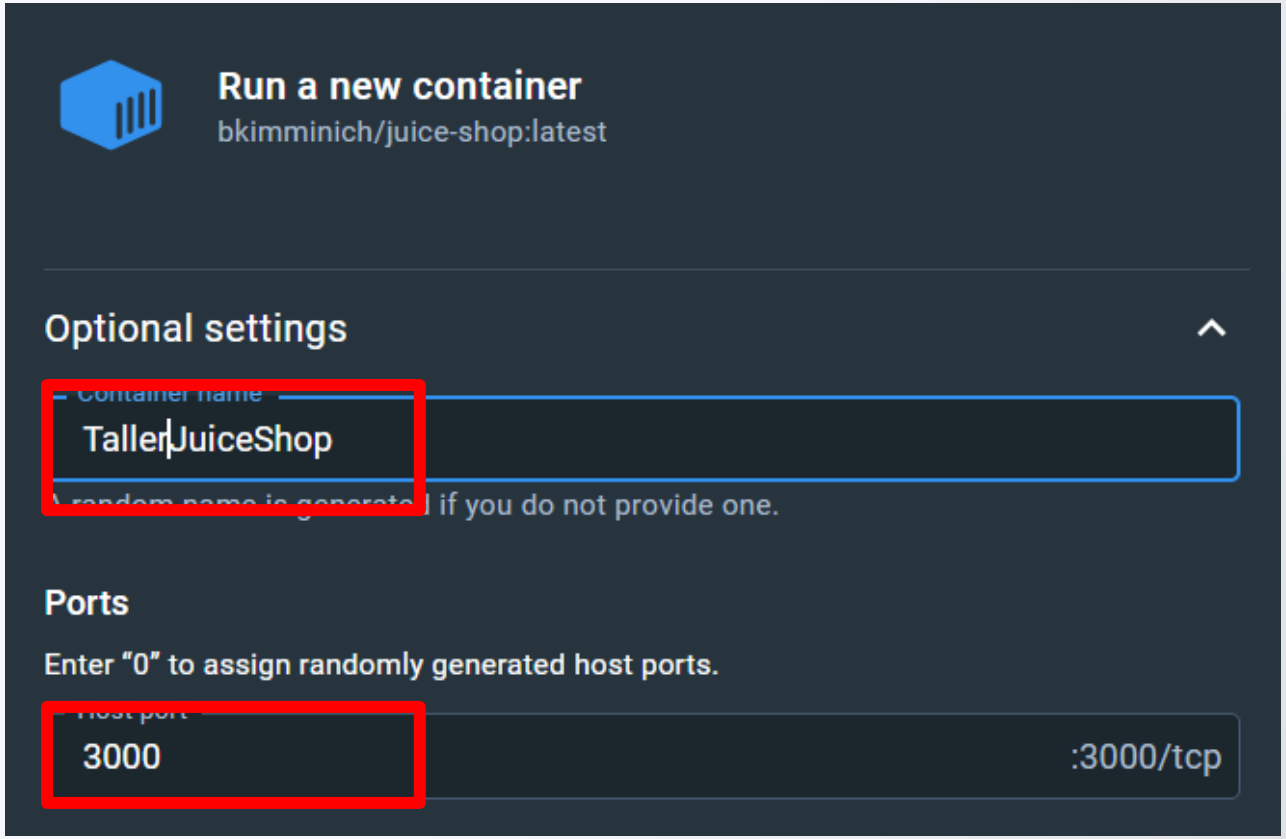
L1: OWASP Juice Shop – Instalación

Instalación con Docker Desktop



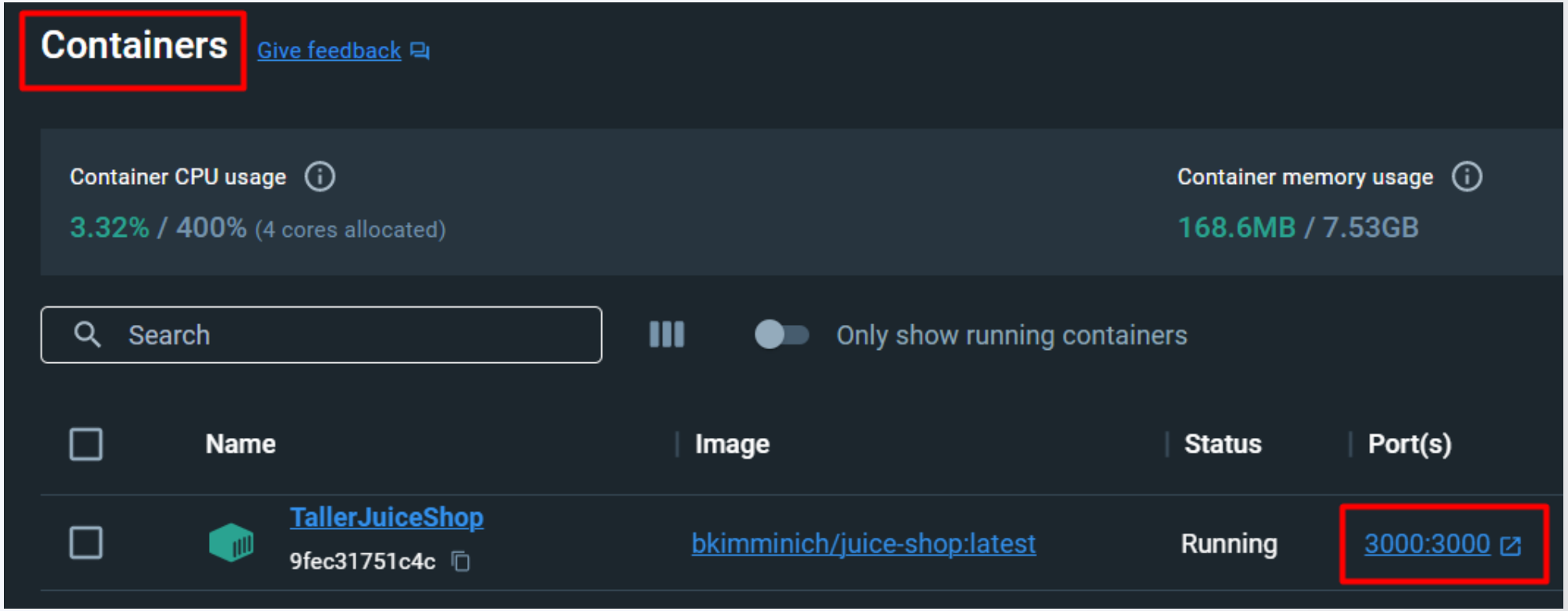
L1: OWASP Juice Shop – Instalación

Instalación con Docker Desktop



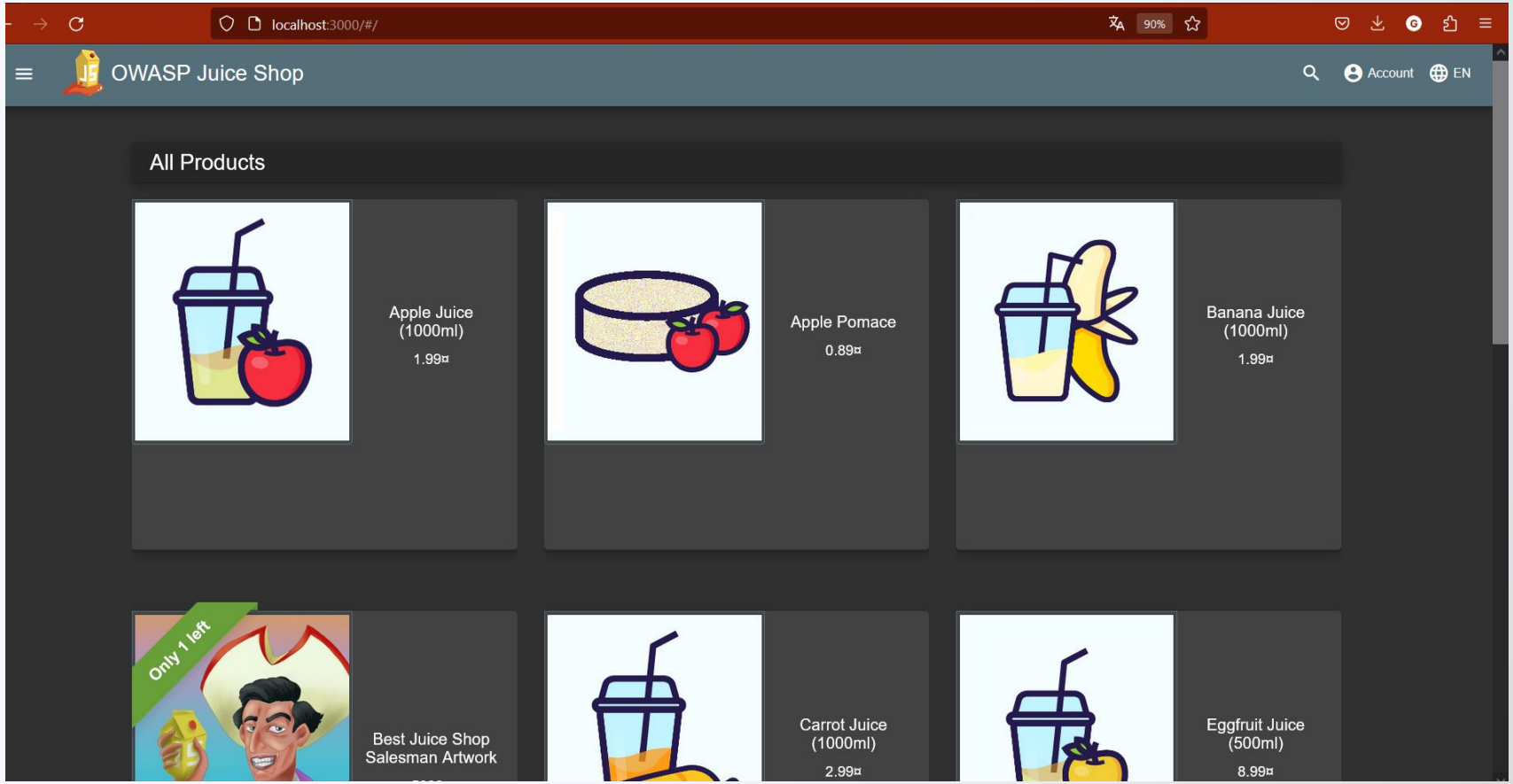
L1: OWASP Juice Shop – Instalación

Instalación con Docker Desktop



L1: OWASP Juice Shop – Instalación

Recomendación: Emplear Chrome para abrir JS.



L1: OWASP Juice Shop

Primero, a navegar para **conocer** el objetivo.

¿Qué **herramienta** de tu navegador crees que podría ayudarte a observar más detalladamente la página?

¿Identificas elementos **posiblemente vulnerables**?

Dependecy check

- Analizador de código abierto cuyo objetivo es detectar componentes o dependencias que tengan vulnerabilidades **divulgadas públicamente**.
- ¿Cómo lo hace?
  - Determina si existe un identificadores único y estándar (CPE) asociado a cada una de las dependencias encontradas.
  - Si los encuentra, lista las vulnerabilidades puntuales (CVEs) de dicho CPE.

4. Laboratoris

Dependecy check

- Algunos de los analizadores y métodos de análisis que incluye.

Analyzer	File Types Scanned	Analysis Method
Archive	Zip archive format (*.zip, *.ear, *.war, *.jar, *.sar, *.apk, *.nupkg); Tape Archive Format (*.tar); Gzip format (*.gz, *.tgz); Bzip2 format (*.bz2, *.tbz2); RPM format (*.rpm)	Extracts archive contents, then scans contents with all available analyzers.
Assembly	.NET Assemblies (*.exe, *.dll)	Uses GrokAssembly.exe ; requires the dotnet core 6.0 runtime to be installed.
Jar	Java archive files (*.jar); Web application archive (*.war)	Examines archive manifest metadata, and Maven Project Object Model files (pom.xml).

- Lista completa disponible en:

https://jeremylong.github.io/DependencyCheck/analyzers/index.html

L2: Dependency Check

DEPENDENCY-CHECK

Instalación del Comman Line Interface (CLI) en Windows

- Descargar el zip y descomprimir:

https://owasp.org/www-project-dependency-check/

- Ir al directorio por línea de comando.

Windows

```
1. dependency-check.bat --help
```

L2: Dependency Check

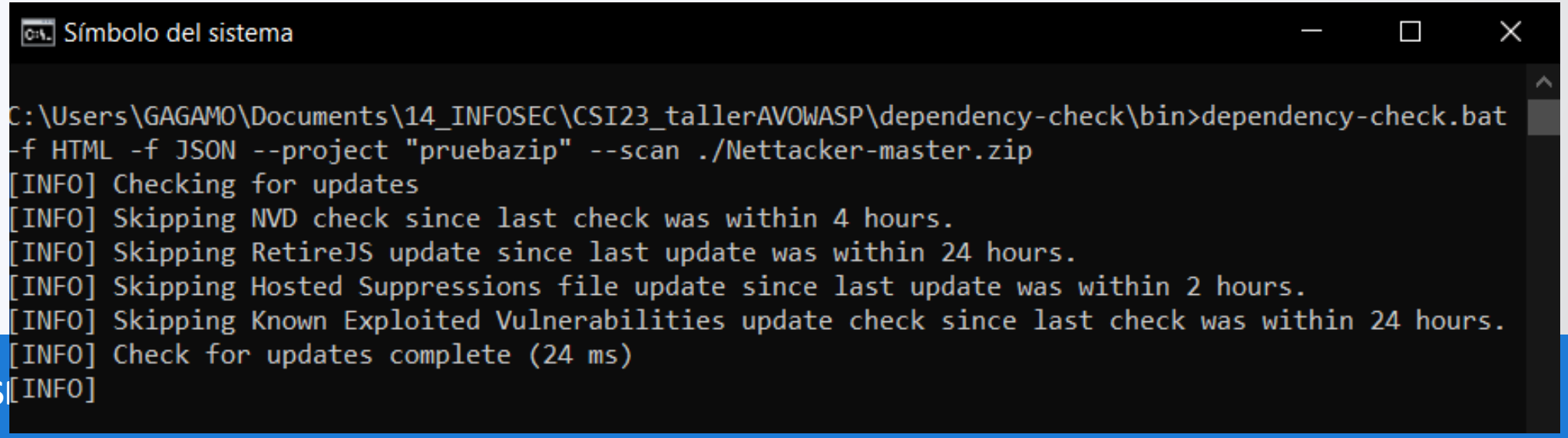
DEPENDENCY-CHECK

Ejecución

Importante: Es obligatoria la conexión a internet.

Target: https://github.com/OWASP/Nettacker/

```
bin>dependency-check.bat -f HTML -f JSON --project “Prueba" --scan
                    "C:\Users\ruta\al\archivo.zip"
```



Bibliografía adicional

Información adicional

- ¿Te interesa proponer un OWASP Project?. https://owasp.org/www-pdf-archive/PROJECT_LEADER-HANDBOOK_2014.pdf
- Penetration Testing Framwork. http://www.vulnerabilityassessment.co.uk/Penetration%20Test.html
- Glosario de NIST. https://csrc.nist.gov/glossary
- *Engineering Trustworthy Secure Systems*. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v1r1.pdf
- Penetrarion Testing Execution Standard. Technical Guides. http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines
- 2023 CWE Top 25 de Debilidades de software más peligrosas. https://cwe.mitre.org/top25/archive/2023/2023_top25_list.html

Agradecimientos

Muchas gracias por su atención.

¿Alguna pregunta?


OWASP
TM