



EXCELENCIA

PROYECTOS DE INNOVACIÓN E INVESTIGACIÓN APLICADAS CON CENTROS:

“INCORPORACIÓN DE CONTENIDOS TRANSVERSALES SOBRE CIBERSEGURIDAD EN LOS CICLOS SUPERIORES DE LA FAMILIA DE INFORMÁTICA Y COMUNICACIONES”

Laboratorio de escaneo de vulnerabilidades

1º CFGS Administración de Sistemas informáticos en Red

Índice

1. Introducción.....	3
2. Instalación y configuración de DockerDescarga de imagen.....	3
3. Descarga de las imágenes docker necesarias.....	4
4. Creación del escenario Docker-Compose del laboratorio.....	5
5. Parar y volver a lanzar el escenario multicontenedor.....	8
6. Acceso a Nesus y escaneos de vulnerabilidades.....	8
7. Eliminando todos los rastros.....	13
8. Webgrafía.....	13

1. Introducción

En resumen lo que vamos a hacer es crear un escenario multi-contenedor docker-compose.

Un escenario multicontenedor es un conjunto de máquinas virtuales conectadas entre sí a través de una red virtual. Podemos redireccionar los puertos necesarios a nuestra máquina anfitriona, por lo que podemos acceder a estas máquinas virtuales desde el navegador y también a través de un terminal.

Por lo tanto, lo primero es instalar Docker en el caso de que no lo tengamos.

Podemos seguir los siguientes tutoriales de Docker:

<https://docs.docker.com/desktop/install/windows-install/>

<https://docs.docker.com/engine/install/ubuntu/>

o buscar otro tutorial, hay infinidad de ellos.

2. Instalación y configuración de Docker. Descarga de imágenes

Lo primero es instalar Docker en el caso de que no lo tengamos.

Podemos seguir los siguientes tutoriales de Docker:

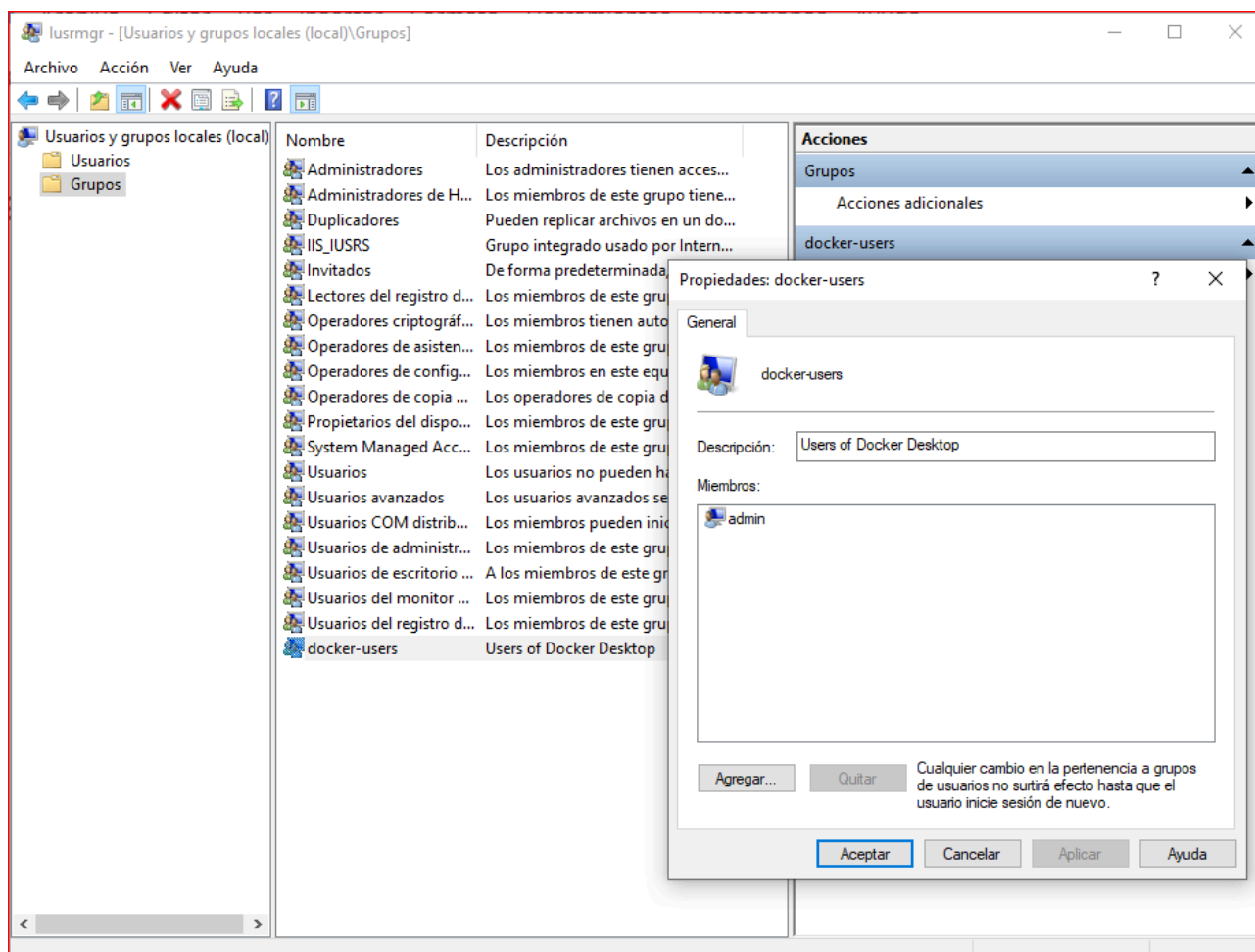
<https://docs.docker.com/desktop/install/windows-install/>

<https://docs.docker.com/engine/install/ubuntu/>

o buscar otro tutorial, hay infinidad de ellos.

Descargamos el instalador de Docker para nuestro Sistema Operativo. Es posible que tengas que autenticarte con una cuenta de docker o Gmail para finalizar la instalación..

Dentro de la configuración de usuarios y grupos, añadimos nuestro usuario al grupo Docker-users (por ejemplo podemos hacerlo abriendo el administrador de usuarios y grupos desde terminar: lusrmgr.msc en windows). Accedemos a la información del grupo Docker-users y desde ahí podemos dar al botón de **Agregar** y buscamos y añadimos nuestro usuario.



En linux podemos usar el comando **adduser mi_usuario docker** desde terminal

3. Descarga de las imágenes docker necesarias

Podemos descargar las imágenes necesarias antes de la creación del escenario multicontenedor. Desde Windows PowerShell, después de haber iniciado Docker Desktop, ejecutamos:

```
docker pull jmmedinac03/bwapp_examen ; docker pull  
jmmedinac03/nessus_plugins ; docker pull kalilinux/kali-rolling ; docker pull
```

mariadb:10 ;docker pull wordpress:5.4 (Si lo hacemos todo en una sola línea no tendremos que esperar a que finalice uno para descargar el siguiente)

4. Creación del escenario Docker-Compose del laboratorio.

Lo primero que tenemos que hacer es crear una carpeta nueva, yo la voy a llamar LaboratoriolESVJP

Nos situamos dentro de esta carpeta y creo un archivo con nombre **docker-compose.yml** con el siguiente contenido: **¡¡¡¡OJO ¡¡¡¡** La línea **command: ["sh", "-c", "apt-get update && apt-cache search kali-linux && apt-get -yf install net-tools iputils-ping kali-linux-headless kali-linux-large && apt-get -yf install && apt -y upgrade"]** es una sola línea.):

```
---
version: '3.3'

services:

  bwapp:
    image: jmmedinac03/bwapp_examen
    ports:
      - "${LISTEN_PORT:-8081}:80"
      #para acceder e inicializar la máquina: http://localhosts:8081/install.php
      #despues haz login
      #usuario bee
      #contraseña bug
    networks:
      - laboratorio-net

  nessus:
    # Nessus Vulnerability Scanner
    image: jmmedinac03/nessus_plugins
    # image: tenable/nessus:latest-ubuntu
    # restart: always
    # código de activacion nessus A2AA-KWWR-ZRSM-RW79-LBPH
    # acceso a la máquina por https://localhost:8834
    # creado usuario:usuario passwd:usuario
    ports:
      - 8834:8834
    networks:
      - laboratorio-net

  kali:
    image: kalilinux/kali-rolling
    restart: unless-stopped
    command: ["sh", "-c", "apt-get update && apt-cache search kali-linux && apt-get -yf install net-tools
iputils-ping kali-linux-headless kali-linux-large && apt-get -yf install && apt -y upgrade"]
    stdin_open: true
    tty: true
    networks:
      - laboratorio-net

  WPdb:
    image: mariadb:10
    volumes:
      - WPdata:/var/lib/mysql
```




























```
environment:
  - MYSQL_ROOT_PASSWORD=secret
  - MYSQL_DATABASE=wordpress
  - MYSQL_USER=manager
  - MYSQL_PASSWORD=secret
networks:
  - laboratorio-net
WPweb:
  image: wordpress:5.4
  depends_on:
    - WPdb
  volumes:
    - ./target:/var/www/html
  environment:
    - WORDPRESS_DB_USER=manager
    - WORDPRESS_DB_PASSWORD=secret
    - WORDPRESS_DB_HOST=WPdb
    - WORDPRESS_DB_NAME=wordpress
  ports:
    - 8080:80
  networks:
    - laboratorio-net
```

```
volumes:
  data:
    WPdata:
networks:
  laboratorio-net:
    driver: bridge
```

Comprobamos que estamos en el Windows Power Shell en la carpeta donde está nuestro docker-compose.yml e Iniciamos la creación del escenario con el siguiente comando: **docker compose up -d**

```
Network labpp3asir_laboratorio-net Created
Volume "labpp3asir_WPdata" Created
Container labpp3asir-bwapp-1 Started
Container labpp3asir-WPdb-1 Started
Container labpp3asir-kali-1 Started
Container labpp3asir-nessus-1 Started
Container labpp3asir-WPweb-1 Started
PS C:\Users\admin\docker\labPP3ASIR>
```

Podemos comprobar que se han creado todos los elementos en Docker Desktop o bien ejecutando **docker compose ps** desde el terminal.

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	 labpp3asir		Running (5/5)	19.39%		2 minutes ago	  
<input type="checkbox"/>	 nessus-1	50c91823ft jmedinac03/nessus	Running	0.59%	8834:8834 	14 minutes ago	  
<input type="checkbox"/>	 bwapp-1	104019681 jmedinac03/bwapp	Running	0.08%	8081:80 	14 minutes ago	  
<input type="checkbox"/>	 WPdb-1	06264198e mariadb:10	Running	0.02%		14 minutes ago	  
<input type="checkbox"/>	 kali-1	cba02c741 kalilinux/kali-rolling	Running	18.7%		2 minutes ago	  
<input type="checkbox"/>	 WPweb-1	1bc0e8ee2 wordpress:5.4	Running	0%	8080:80 	14 minutes ago	  

Como podemos ver hemos creado 5 máquinas virtuales:

- **Nessus** es nuestra herramienta de escaner de vulnerabilidades. Podemos acceder a ella desde un navegador web <https://localhost:8834> (Es posible que nos salga una advertencia de seguridad. Le damos a continuar). Accedemos a la máquina con usuario:**usuario** y contraseña:**usuario**.

- **bwapp**: Es una máquina que se ha hecho deliberadamente insegura para poder practicar. Para acceder a ella primero procedemos a la instalación desde un navegador en la dirección

<http://localhost:8081/install.php> y

después de hacer click sobre el

mensaje “Click here to install

bWAPP.” podemos acceder ya desde el login: <http://localhost:8081/login.php> con usuario:**bee** y contraseña: **bug** . Nos va a servir para ver en ella un gran número de vulnerabilidades.

```
PS C:\Users\admin\docker\labPP3ASIR> docker-compose exec kali bash
(root@cba02c741ff8)-[/]
```

- **Kali**: Es una distribución de linux. Tenerla dentro de la red nos permite poder ejecutar comandos, inspeccionar, e incluso probar ataques en la red. Para acceder al terminal de la máquina de Kali-linux, escribimos en el terminal (siempre situados en la carpeta del laboratorio): **docker-compose exec kali**

bash. Se conectará a la MV y nos cambiará el prompt.

- **WordPress**: está compuesto por dos máquinas virtuales, una contiene la Base de Datos (WPdb) y otra la interfaz web(WPweb). Podemos acceder a ella, también desde el navegador web, en la dirección <http://localhost:8080/> . Si la vamos a utilizar tendremos primero que efectuar el proceso de instalación. Después de unos minutos que tarda en instalarse, podemos acceder con el usuario y contraseña que hemos puesto.



Esta máquina también nos va a servir para ver las vulnerabilidades que contiene.

5. Parar y volver a lanzar el escenario multicontenedor

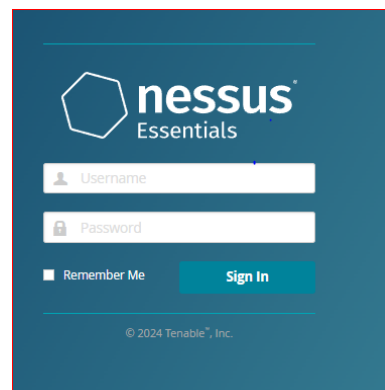
Si necesitamos parar nuestro escenario para continuar en otro momento debemos ejecutar el comando **docker-compose stop**.

Para reanudarlo ejecutaremos **docker-compose start**.

Recordamos que para realizar cualquier acción sobre los contenedores o escenario tenemos que estar en el terminal en la carpeta del escenario.

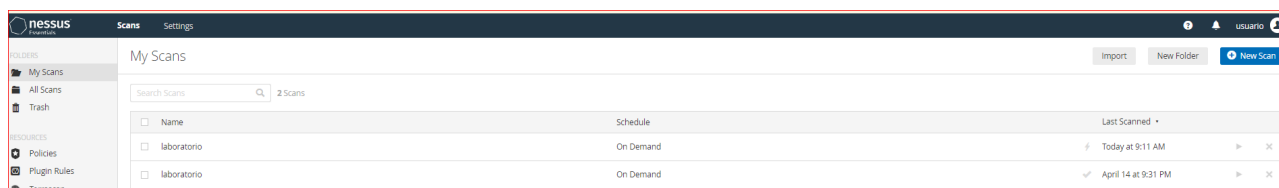
6. Acceso a Nessus y escaneos de vulnerabilidades.

Como hemos comentado, para acceder a nuestra máquina virtual de Nessus ponemos en nuestro navegador web <https://localhost:8834>.



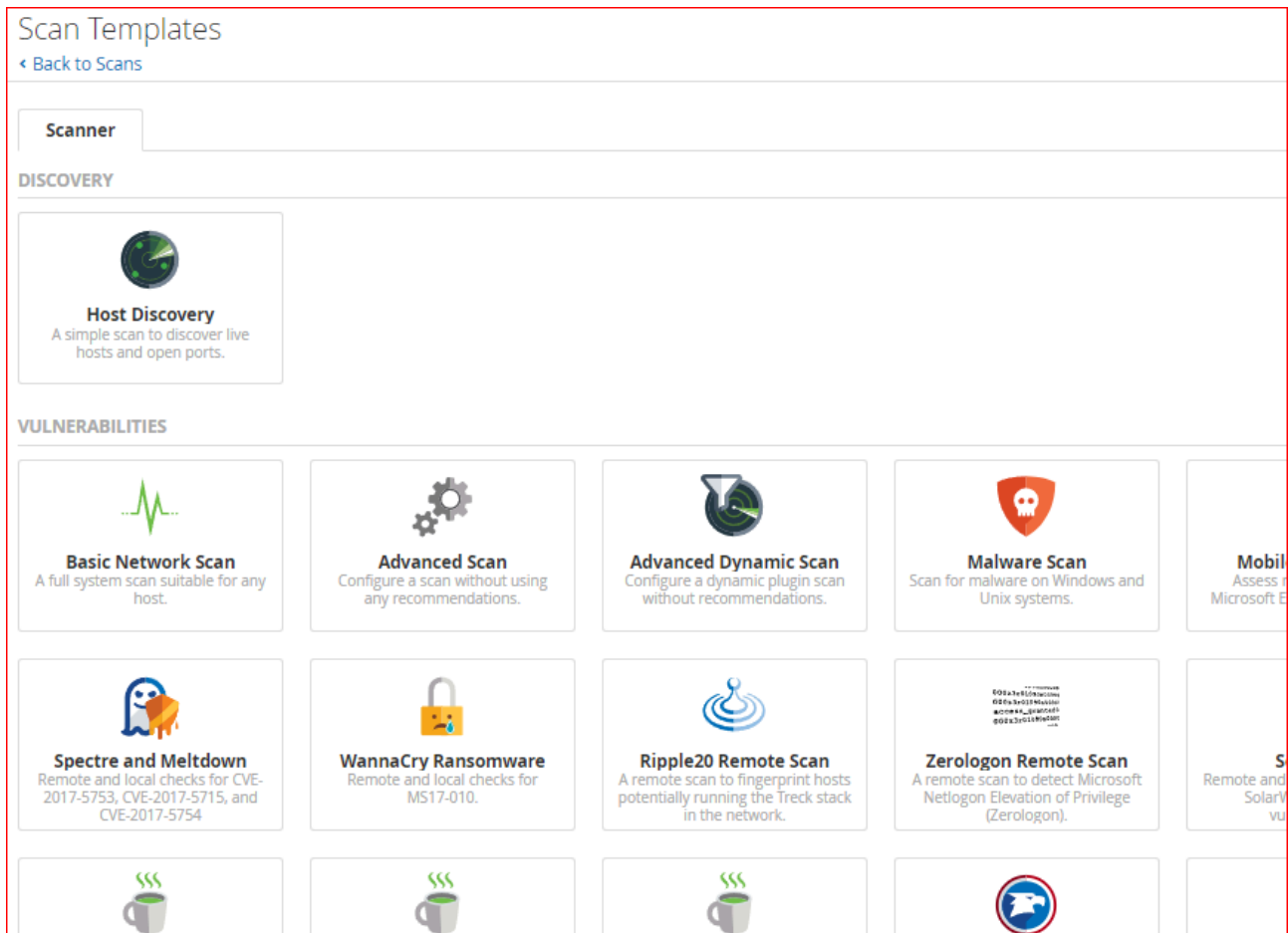
Usamos el usuario: usuario y la contraseña:usuario

Una vez dentro de la interfaz web nos aparecen los escaneos que hemos realizado.



En la parte superior, a la derecha tenemos el botón **New Scan** para hacer un nuevo escaneo.

Tenemos gran variedad de escaneos diferentes, algunos de ellos están disponibles en la versión de prueba y otros sólo en la de pago.



Nosotros vamos a hacer dos, el primero es un descubrimiento de los equipos que hay en la red. En éste nos va a aparecer información sobre los equipos conectados a la red y los puertos que tienen abiertos, que en gran medida van a darnos información de las debilidades que pueden tener presentes.

Debemos de saber los datos de la red virtual creada. Para ello en un terminal de Windows

PowerShell ejecutamos el comando: **docker network inspect labpp3asir_laboratorio-net** (Si tu carpeta no se llama labpp3asir, tendrás que modificar el comando y poner ahí el nombre de tu carpeta).

Este comando nos informa de los detalles de la red virtual creada.

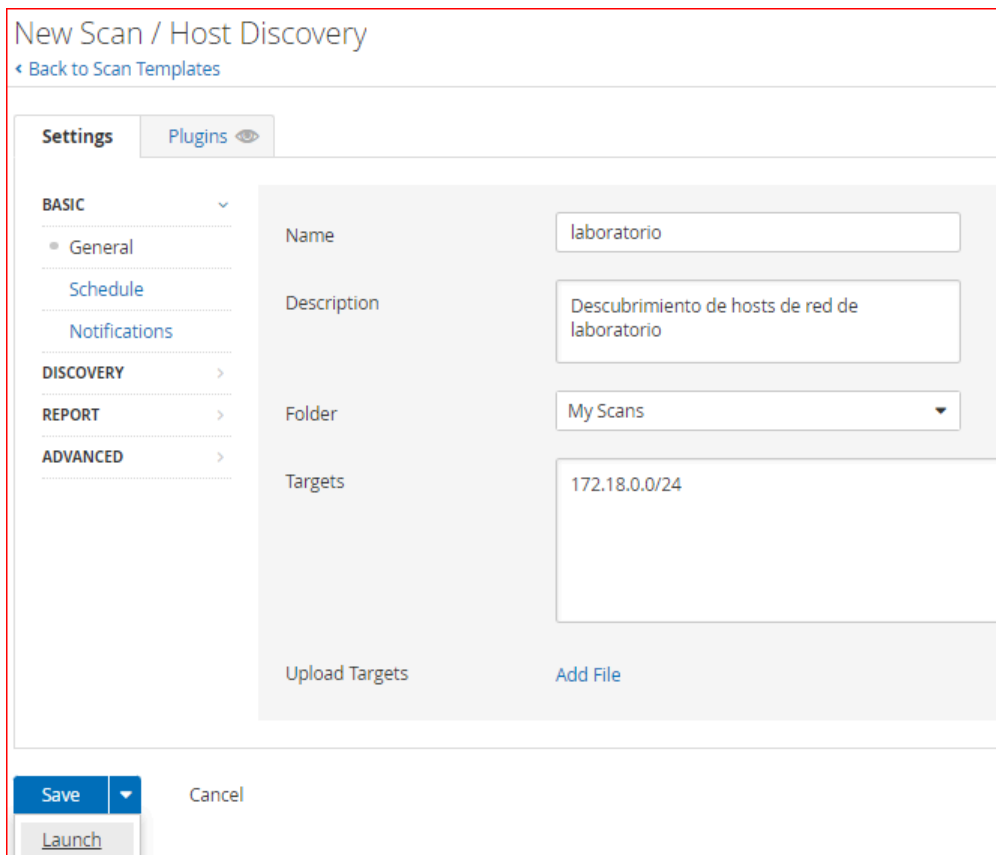
En este caso podemos observar que los datos de nuestra red son 172.18.0.0/16. (Comprueba si los datos de tu red son los mismos porque pueden cambiar).

```

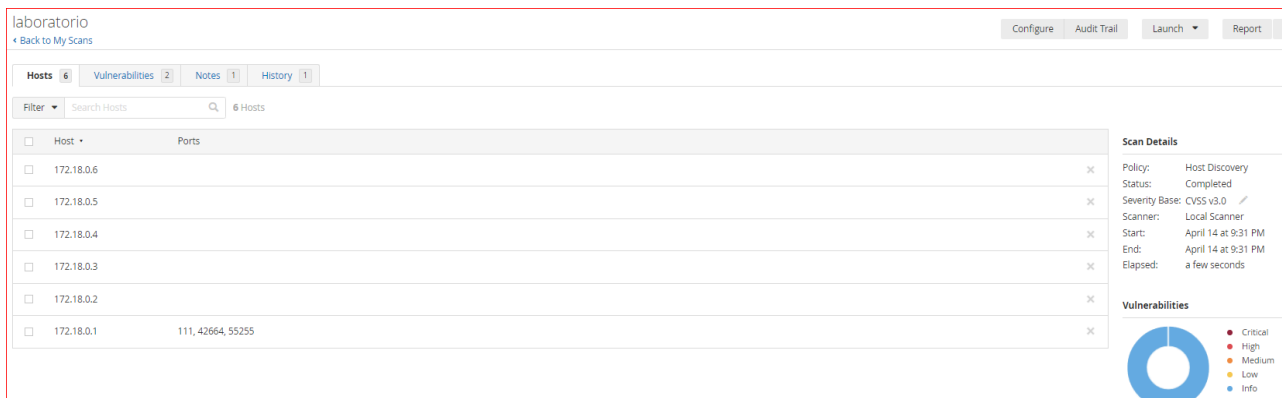
},
  "1bc0e8ee2d1298a63b4686420fb0cc09ee47495e",
  "Name": "labpp3asir-wPweb-1",
  "EndpointID": "b41f2ecdb511832e3f2ad7",
  "MacAddress": "02:42:ac:12:00:06",
  "IPv4Address": "172.18.0.6/16",
  "IPv6Address": ""
},
  "50c91823fb64b6c1e4edb7f14dbdb29c68bbe98",
  "Name": "labpp3asir-nessus-1",
  "EndpointID": "fb8eb27025b821f1e00a49",
  "MacAddress": "02:42:ac:12:00:04",
  "IPv4Address": "172.18.0.4/16",
  "IPv6Address": ""

```

Esos son los datos que tendremos que poner en el campo Target de el escaneo de descubrimiento de redes.




El resultado es el que se muestra, donde vemos los equipos presentes con sus direcciones correspondientes.



Host	Ports
<input type="checkbox"/> 172.18.0.6	
<input type="checkbox"/> 172.18.0.5	
<input type="checkbox"/> 172.18.0.4	
<input type="checkbox"/> 172.18.0.3	
<input type="checkbox"/> 172.18.0.2	
<input type="checkbox"/> 172.18.0.1	111, 42664, 55255

Scan Details
Policy: Host Discovery
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: April 14 at 9:31 PM
End: April 14 at 9:31 PM
Elapsed: a few seconds

Vulnerabilities


- Critical
- High
- Medium
- Low
- Info

Vemos cómo además de tener información de los **Hosts**, también tenemos en otras pestañas con información acerca de las **Vulnerabilidades**, **Avisos** e **Historial**

Hacemos para finalizar un **Escaneo Básico de Red** que en este caso, ya nos van a aparecer vulnerabilidades en las diferentes máquinas virtuales que tenemos.

New Scan / Basic Network Scan

Back to Scan Templates

Settings Credentials Plugins

BASIC

- General
- Schedule
- Notifications

DISCOVERY

ASSESSMENT

REPORT

ADVANCED

Name: Escaneo Básico Laboratorio

Description: Escaneo básico del laboratorio

Folder: My Scans

Targets: 172.18.0.0/16

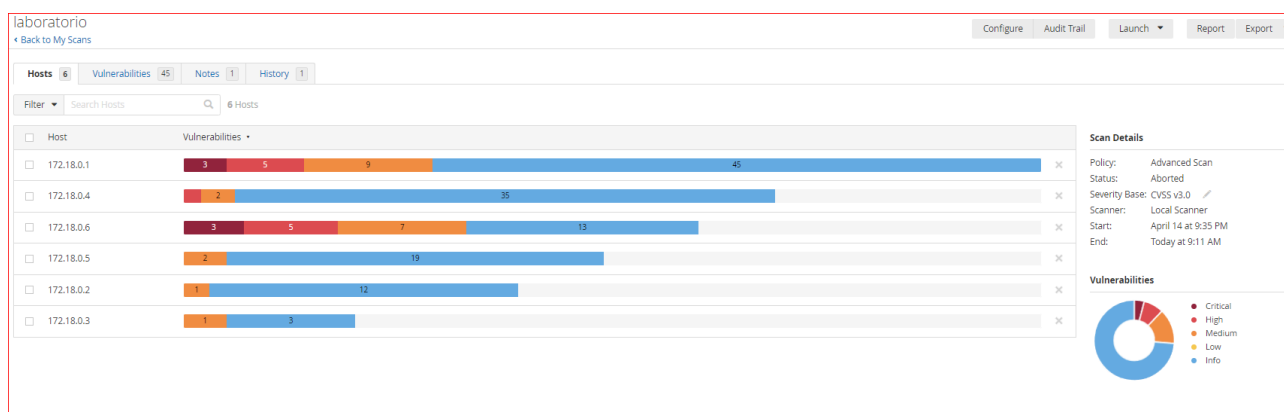
Upload Targets Add File

Save Cancel

Launch

Posteriormente podríamos hacer escaneos avanzados sobre hosts en concreto, escáner de malware, etc... como hemos dicho hay una gran variedad de operaciones disponibles

En la siguiente imagen vemos todas las vulnerabilidades encontradas. De modo gráfico podemos verlo en la parte derecha con diferentes colores que van desde el azul que son sólo informativas a las de color rojo fuerte que se tratan de vulnerabilidades críticas.



En la pestaña de Vulnerabilidades vamos a encontrar más detalles de todas las vulnerabilidades encontradas.

laboratorio

[Back to My Scans](#)

Hosts 6 Vulnerabilities 45 Notes 1 History 1

Filter Search Vulnerabilities 45 Vulnerabilities

Sev	CVSS	VPR	Name	Family	Count
MIXED	14 PHP (Multiple Issues)	CGI abuses	28
HIGH	7.5	4.4	Ubuntu 20.04 LTS / 22.04 LTS / 22.10 / 23.04 : FreeType vulnerability (USN-6062-1)	Ubuntu Local Security Checks	1
MEDIUM	6.5	4.0	IP Forwarding Enabled	Firewalls	5
MEDIUM	5.3		web.config File Information Disclosure	CGI abuses	2
MIXED	6 SSL (Multiple Issues)	General	10
INFO	2 HTTP (Multiple Issues)	Web Servers	12
INFO	2 Apache HTTP Server (Multiple Issues)	Web Servers	6
INFO	2 PHP (Multiple Issues)	Web Servers	6
INFO	2 TLS (Multiple Issues)	Service detection	4

Podemos también obtener los detalles de cada una de ellas, pulsando sobre la que queramos.

laboratorio / Plugin #175487

[Back to Vulnerabilities](#)

Hosts 6 Vulnerabilities 45 Notes 1 History 1

HIGH Ubuntu 20.04 LTS / 22.04 LTS / 22.10 / 23.04 : FreeType vulnerability (USN-6062-1)

Description

The remote Ubuntu 20.04 LTS / 22.04 LTS / 22.10 / 23.04 host has packages installed that are affected by a vulnerability as referenced in the USN-6062-1 advisory.

- An integer overflow vulnerability was discovered in FreeType in tt_hvadvance_adjust() function in src/truetype/ttgxvar.c. (CVE-2023-2004)

Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

Solution

Update the affected packages.

See Also

<https://ubuntu.com/security/notices/USN-6062-1>

Output

```
- Installed package : libfreetype6_2.10.1-2ubuntu0.2
Fixed package      : libfreetype6_2.10.1-2ubuntu0.3
```

To see debug logs, please visit individual host

Port	Hosts
N/A	172.18.0.4

Como vemos, podemos obtener información de la descripción de la vulnerabilidad, indicaciones de cómo se puede solucionar y enlaces de páginas web donde podemos encontrar más información.

7. Eliminando todos los rastros.

Eliminamos el escenario multicontenedor desde un terminal: **docker compose down**.

Borramos las imágenes que hemos descargado para el laboratorio:

```
docker image rm jmmedinac03/bwapp_examen ; docker image rm  
jmmedinac03/nessus_plugins ; docker image rm kalilinux/kali-rolling ; docker image  
rm mariadb:10 ;docker image rm wordpress:5.4
```

8. Webgrafía

Aparte de las páginas web de docker: <https://www.docker.com/> y Nessus <https://es-la.tenable.com/products/nessus>