Redes de Información - Trabajo Práctico

Tema 4 - OpenVas/Greenbone

Tabla de Contenidos

Requisitos

- Docker
- Dockerfile de la máquina vulnerable

Instalación

Para poder usar los comandos de docker tal cual como están se deben seguir los siguientes pasos para no tener que usar `sudo` en linux.

Esto es opcional, en caso de no querer hacerlo se debe anteponer `sudo` antes de los comandos de docker.

Configuración de red virtual

Primero se debe crear una red virtual de docker, dentro de esta se encontrarán nada más que 3 hosts. 2 de ellos son máquinas vulnerables y el otro es el que contiene los servicios de OpenVas.

Para crear la red:

```
docker network create tpe-redes --attachable --subnet 10.0.0.0/24
```

En caso de que esa subred se encuentre en uso se deberá cambiar el `10.0.0.0/24` por otra que sí esté disponible.

Configuración de contenedores vulnerables

Se recomienda una terminal distinta para cada contenedor.

Metasploitable

Metasploitable es una máquina virtual de Ubuntu diseñada para ser *bastante* vulnerable, se utiliza para probar distintas herramientas de seguridad. Se utilizará la siguiente versión dockerizada.

Para crear el contenedor y asignarlo a la red creada previamente ejecutar el siguiente comando:

```
docker run \
    -it \
    --rm \
    --network tpe-redes \
    --ip="10.0.0.2" \
    --name metasploitable \
    --hostname metasploitable2 \
    tleemcjr/metasploitable2 \
    sh -c "/bin/services.sh && bash"
```

Esto levanta el contenedor en la red creada previamente y le asigna una ip estática. Luego levanta todos los servicios vulnerables y queda en la terminal del contenedor.

Contenedor casero

Esta image es muy simple, está utilizando una versión vieja de ubuntu y cuenta con servicio ssh en el puerto 2222 al cual se lo puede acceder con el usuario/contraseña `admin:admin`.

Primero hay que crear la imagen, para ello hay que correr el siguiente comando desde el directorio raíz del proyecto donde se encuentra el Dockerfile.

Si no cuenta con el Dockerfile se lo puede obtener de la siguiente manera (en Linux):

```
# Crear y moverse al directorio del proyecto
mkdir tpe-redes
cd tpe-redes
# Descargar el archivo Dockerfile al directorio actual
wget https://gist.githubusercontent.com/santire/80d206de725b0861bc5cb38b2cc87583/raw/9bc8bc2cb1f03039fa7ca32dd7b
```

Y luego:

```
docker build -t vulnerable-casero .
```

Para crear el contenedor y asignarlo a la red:

```
docker run \
    -it \
    --rm \
    -d \
    --network tpe-redes \
    --ip="10.0.0.3" \
    --name vulnerable \
    --hostname vulnerable-casero \
    vulnerable-casero
```

Esto levanta el contenedor en el fondo.

Para entrar al contenedor hay que correr:

```
docker exec -it vulnerable bash
```

Configuración de contenedor OpenVas/Greenbone

OpenVas es un escáner y se utiliza como parte del sistema de manejo de vulnerabilidades de Greenbone. Se utiliza una imagen de docker pre configurada que ya cuenta con todo el sistema de manejo de vulnerabilidades de Greenbone utilizando OpenVas.

Crear un volumen de docker

Primero hay que crear un volumen de docker para persistir los datos si se cierra el contenedor.

```
docker volume create tpe-redes-openvas
```

Crear el contenedor

Luego, para crear el componente y asignarlo a la red:

```
docker run \
    -it \
    --detach \
    --publish 8080:9392 \
    --network tpe-redes \
    --ip="10.0.0.4" \
    --volume tpe-redes-openvas:/data \
    --name openvas \
    --hostname openvas \
    immauss/openvas
```

El `--publish 8080:9392` es para poder acceder al puerto 9392 del contenedor a través del puerto `8080` del host. De esta manera se puede acceder a la interfaz web del Greenbone Security Assistant accediendo a http://localhost:8080 desde el navegador de la maquina host.

Nota: Este acceso es por http, por lo cual no es muy seguro ya que las credenciales se encuentran visibles para cualquiera que intercepte el paquete. Se puede acceder utilizando https, pero para eso hay que tenerlo configurado con certificados. Para la demo se utiliza http ya que es una demo en un ambiente de prueba. La funcionalidad es la misma.

La primera vez que se levanta el contenedor no se podrá acceder a la interfaz hasta que se terminen de crear y actualizar las bases de vulnerabilidades, esto puede tardar bastante tiempo (~30min).

El contenedor se genera con el usuario administrador `admin` con contraseña `admin` por defecto.

Por defecto el contenedor sincroniza las bases de vulnerabilidades cada vez que se reinicia.

Para ver el estado de esta sincronización se puede correr lo siguiente:

```
docker logs -f openvas
```

Validar configuración de red

Luego, se puede ingresar al contenedor de la siguiente manera:

```
docker exec -it openvas bash
```

Desde la terminal del contenedor:

```
apt-get update && apt-get install nmap net-tools iputils-ping vim -y
```

Para instalar algunas herramientas básicas. Luego se puede verificar que toda la configuración inicial se haya hecho correctamente corriendo los siguientes comandos:

```
ifconfig
```

Debería presentar el siguiente resultado

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.4  netmask 255.255.255.0  broadcast 10.0.1.255
        ether 02:42:0a:00:01:04  txqueuelen 0  (Ethernet)
        RX packets 8066  bytes 10990814 (10.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2833  bytes 6302693 (6.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 870  bytes 603168 (589.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 870  bytes 603168 (589.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Se puede validar que las dos máquinas vulnerables estén activas haciéndoles un `ping` a sus respectivas IPs, o de la siguiente manera con nmap:

```
nmap -sP 10.0.0.0/24
```

Que debería devolver lo siguiente:

```
root@openvas:/# nmap -sP 10.0.0.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-14 21:28 UTC
Nmap scan report for 10.0.0.1
Host is up (0.000039s latency).
MAC Address: 02:42:A8:AD:C7:E5 (Unknown)
Nmap scan report for metasploitable.tpe-redes (10.0.0.2)
Host is up (0.000010s latency).
MAC Address: 02:42:0A:00:01:02 (Unknown)
Nmap scan report for vulnerable.tpe-redes (10.0.0.3)
Host is up (0.000019s latency).
MAC Address: 02:42:0A:00:01:03 (Unknown)
Nmap scan report for openvas (10.0.0.4)
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 15.02 seconds
```

De este resultado se puede ignorar la fecha, tiempos de latencia y direcciones MAC, pero es importante que para los hosts creados previamente les corresponda su IP y hostname.

Repositorio de Github

En el repositorio se encuentran todos los documentos junto con el Dockerfile

Link al repositorio