Entorno y contexto

En este laboratorio utilizaremos dos aplicaciones deliberadamente vulnerables para recorrer cada fase de la metodología de hacking ético. OWASP Juice Shop estará disponible en el puerto 3000, y DVWA (Damn Vulnerable Web Application) en el puerto 8080. El objetivo es familiarizarse con las herramientas de pentesting más comunes.

Preparación del entorno

Antes de nada, en Windows debemos:

Abrir PowerShell como Administrador e instalar WSL2 con Ubuntu 22.04

wsl --install -d Ubuntu-22.04

Cómo funciona

- `wsl --install -d Ubuntu-22.04`: descarga e instala la distribución Ubuntu 22.04 en WSL2.
- El flag `-d` especifica la distro a instalar; requiere permisos de administrador.

Instalación de herramientas en Ubuntu

Ya en la terminal de Ubuntu:

sudo apt update

Cómo funciona

- Actualiza la lista de paquetes disponibles y sus versiones desde los repositorios configurados.

sudo apt install -y nmap gobuster sqlmap ffuf jq

Cómo funciona

- `apt install`: instala paquetes listados.
- `-y`: responde automáticamente “sí” a las confirmaciones.
- **nmap**: escáner de puertos y servicios.
- **gobuster**: descubre directorios y ficheros.
- **sqlmap**: automatiza pruebas de inyección SQL.
- **ffuf**: realiza fuzzing de rutas o parámetros.
- **jq**: procesa salidas en formato JSON.

Reconocimiento de OWASP Juice Shop

nmap -sC -sV -oN juice_nmap.txt localhost -p3000

Cómo funciona

- `-sC`: ejecuta scripts básicos de detección (versiones, cabeceras).
- `-sV`: intenta identificar la versión del servicio en cada puerto.
- `-oN juice_nmap.txt`: guarda el resultado en texto plano en `juice_nmap.txt`.
- `localhost -p3000`: escanea sólo el puerto 3000 de la máquina local.

curl -I http://localhost:3000/

Cómo funciona

- `curl -I`: solicita sólo las cabeceras HTTP (HEAD request).
- Permite ver el código de estado, políticas de seguridad y metadatos sin descargar el cuerpo de la respuesta.

Pruebas de inyección y autenticación en Juice Shop

sqlmap -u "http://localhost:3000/rest/products/search?q=1" --batch --dbs

Cómo funciona

- `-u`: URL objetivo con parámetro vulnerable (`q=1`).
- `--batch`: ejecuta sin pedir interacción.
- `--dbs`: enumera las bases de datos si encuentra inyección.

curl -X POST http://localhost:3000/rest/user/login \

-H "Content-Type: application/json" \

-d '{"email":"admin@juice-sh.op","password":"admin123"}'

Cómo funciona

- `-X POST`: indica método POST.
- `-H "Content-Type: application/json"`: fija el tipo de contenido del cuerpo.
- `-d '{…}'`: envía las credenciales por defecto.
- La respuesta incluye el JWT que autoriza llamadas posteriores.

Despliegue de DVWA

Asegúrate de tener este `docker-compose.yml` en tu carpeta de laboratorio:

```
services:
  dvwa:
    image: vulnerables/web-dvwa
    ports:
      - "8080:80"
    restart: unless-stopped
```

Cómo funciona

- Define un servicio llamado `dvwa` usando la imagen oficial.
- Mapea el puerto interno `80` al `8080` de tu máquina.
- `restart: unless-stopped` reinicia el contenedor tras caídas o reinicios del sistema.

Levántalo con:

```
docker-compose up -d
```

Cómo funciona

- `up`: crea y arranca los contenedores del servicio definido.
- `-d`: modo “detached”, ejecuta en segundo plano.

Reconocimiento y fuzzing en DVWA

```
nmap -Pn -p80 -oN dvwa_nmap.txt localhost
```

Cómo funciona

- `-Pn`: asume que el host está activo (sin ping).
- `-p80`: escanea el puerto 80 donde corre DVWA.
- `-oN dvwa_nmap.txt`: guarda el informe en `dvwa_nmap.txt`.

ffuf -u "http://localhost:8080/FUZZ" \

-w /usr/share/wordlists/dirb/common.txt

Cómo funciona

- `-u "…/FUZZ"`: la palabra clave `FUZZ` se reemplaza por cada entrada del wordlist.
- `-w`: ruta al wordlist de rutas comunes.
- Permite descubrir directorios o scripts no documentados en la aplicación.