**Bloque 1: Enumeración con NMap Nv1**

Notas:

Hacer siempre los ataques de nmap con sudo

sudo nano /etc/hosts

192.168.8.1 ejemplo1

(cada vez que hagas un ping a ejemplo1 lo hace a esa ip para no tener que recordarla)

sudo nmap ejemplo1

en nmap los puertos pueden estar

open-closed-filtered-open|filtered(no fue capaz de saberlo)

**Ejercicio L89B1\_NMAPSTART: Una simple “puesta en marcha”**

sudo nmap 192.168.8.1-100 escanea el rango de IPs y te da las IP y sus puertos abiertos

Ahora meto todas las IP en el archivo /etc/hosts para darles un nombre

Para ver el SO: nmap -A -T4 <IP objetivo>

Deteccion de versiones: nmap -sV <IP objetivo>

• Escaneado de múltiples objetivos: nmap ... . P. ej.: nmap 192.168.2.1 192.168.2.100 scanme.nmap.org

• Escaneado de un rango de IPs: P. ej.: nmap 192.168.2.1-100

• Escaneado de una red completa: P. ej.: nmap 192.168.0.0/24

• Escaneado de una red completa excluyendo ciertos hosts: P. ej.: nmap 192.168.0.0/24 --exclude 192.168.2.10

• Identificar el sistema operativo de las máquinas activas: P. ej.: nmap -O 192.168.0.0/24

• Escanear sólo un rango de puertos: P. ej.: nmap -p 22-80 192.168.0.0/24

Análisis rápido (20 puertos principales) -> nmap -- top-ports 20 –open IPObjetivo

Análisis lento y detallado: sudo nmap -sS -A -sV -O –p – IPobjetivo

• -sS: Utiliza un tipo de análisis TCP SYN, potencialmente más capaz de evadir firewalls.

• -A: Detecta las versiones de servicios y sistema operativo, realiza un traceroute y ejecuta algunos scripts de análisis para completar la información que obtiene.

• -sV: Investiga puertos abiertos para determinar qué servicio y versión están ejecutando

• -O: Activa la detección de la versión y tipo del sistema operativo.

• -p -: Escanea un rango de puertos (- significa todos los puertos)

**Tratar con puertos**

§ nmap –p 80,443 scanme.nmap.org: Solo escanea los puertos 80 y 443 (útil si solo se quieren analizar los puertos típicos usados por un servidor web)

§ nmap –p 100-2000 scanme.nmap.org: Escaneo aleatorio de puertos del 100 a 2000.

§ nmap –p -2000 scanme.nmap.org: Ídem, pero del 1 a 2000.

§ nmap –p 100- scanme.nmap.org: Ídem, pero del 100 a 65536 (nada recomendable).

§ Puedes utilizar protocolos concretos asociados a cualquier número de puerto por utilizando letras (T: TCP, U: UDP, S: SCTP, P: IP). P. e.: nmap -p U:53,11,13,T:22-25,80,443,8080 scanme.nmap.org

§ También hay dos modificadores adicionales

• -F (Escaneo rápido): En lugar de 1000, escanea los 100 puertos más utilizados, aleatoriamente

• -r: En lugar de utilizar un orden de puertos aleatorio, utiliza uno secuencial ascendente

Estados de los puertos

§ Open: Hay un servicio esperando una conexión en este puerto.

§ Closed: Ningún servicio parece estar esperando conexiones en este puerto.

§ Filtered: Los paquetes NMap no se reciben en este puerto, por lo que su estado actual es desconocido.

§ Unfiltered: Los paquetes NMap se reciben en este puerto, pero alguna razón impide que NMap sepa en qué estado está el puerto.

§ Open/Filtered: NMap no puede decidir en cuál de estos dos estados está el puerto.

§ Closed/Filtered: Ídem al anterior.

**Uso del motor de scripting “de usar y tirar” (escaneo con scripts por defecto)**

agregando la opción -sC a cualquiera de los anteriores

**Bloque 2: Enumeración con NMap Nv2**

**Entender (un poco) el protocolo TCP**

**Tipos de descubrimiento de hosts avanzados de NMap**

--script=nombre de script y parámetros

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Tabla

Descripción generada automáticamente**

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

**Tipos de escaneo NMap y evasión de firewalls básica**

Usar el motor de scripts

**Interfaz de usuario gráfica

Descripción generada automáticamente**

Técnicas avanzadas de análisis de puertos

Imagen que contiene Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Escaneos más avanzados de Nmap**

opción -n se puede añadir a cualquiera de ellos para mejorar su velocidad

• Escaneo TCP SYN y UDP (requiere tener privilegios de root):

nmap -sS -sU -Pn 192.168.13.37

Este análisis TCP SYN y UDP tardará bastante tiempo, pero es discreto y sigiloso. Este comando comprobará alrededor de 2000 puertos TCP y UDP comunes (1000 TCP+1000 UDP de los más usados) para ver si están respondiendo. El parámetro -Pn se salta el escaneo vía ping y asume que el host está “vivo”. Esto puede ser útil cuando hay un firewall que podría estar impidiendo las respuestas ICMP.

• Escaneo TCP SYN y UDP para todos los puertos reservados (requiere tener privilegios de root): una variante del anterior. Ej.: nmap -sS -sU -PN -p 1-1024 192.168.13.37

• TCP Connect Scan (no requiere tener privilegios de root): Ej.: nmap -sT 192.168.13.37. Desaconsejado. Como dijimos, trata de usar un escaneo más sigiloso primero.

• Análisis rápido (no requiere tener privilegios de root): nmap -T4 -F 192.168.13.37. Usa opciones de temporización predefinidas para mejorar la velocidad de escaneo y escanea solo 100 puertos (- F).

Las opciones de temporización de NMap aparecen en esta tabla. Por defecto se utiliza T3. T0 es muy lento, pero también prácticamente indetectable.

Las opciones de temporización solo son un conjunto predefinido de valores para ciertos parámetros de NMap, que también se pueden ajustar individualmente para obtener más control (visto en el siguiente nivel).

Tabla

Descripción generada automáticamente

• Verbose: nmap -T4 -A -v 192.168.13.37. Con la opción -v añadida a la mayoría de los escaneos se obtiene una mejor visión de lo que está haciendo NMap; para algunos escaneos además da detalles adicionales que el informe de la salida no proporciona.

**Detección de tipo de sistema operativo sin "ruido”**

**Como -O es muy detectable, jugaremos con el tiempo de vida (TTL) de los paquetes que enviamos.**

Esto no requiere usar NMap, sino hacer ping al sistema de la siguiente forma:

ping -c 1 IPDestino.

Como resultado, si hay conexión, recibiremos respuestas que indican un valor ttl. Este valor se puede consultar en esta tabla para adivinar el tipo de sistema operativo que el objetivo tiene:

**Tabla

Descripción generada automáticamente**

**Formatos de salida NMap**

• Formato NMap (-oN fichero ): Equivalente a redirigir la salida NMap a un archivo, sigue la estructura de salida tradicional vista en teoría.

• Formato “Grepeable” (-oG fichero): Transforma la salida para que cada vez que un comando grep encuentre una coincidencia con la palabra que estamos buscando se muestre más información. También facilita su proceso mediante expresiones regulares.

• Formato XML (-oX fichero ): la salida tradicional contenida en una estructura XML. Aparte de ser mucho más sencillos de procesar por programa, estos archivos pueden ser importados directamente por Metasploit para utilizar los resultados de un análisis para comenzar operaciones de explotación.

**Bloque 3: Enumeración con NMap Nv3**

**NMap Scripting Engine (NSE)**

antes de usar locate por primera vez -> sudo updatedb

vamos a usar el comando locate para buscar archivos

locate \*.nse (busca todos los scripts de nmap)

sudo nmap --script-help=\*ftp\*

el script de geolocalizacion no funciona

hacer locate \*(cadena)\*.nse para encontrar los scripts

para ejecutar un script:

nmap –script nombre --script-args argumentos

**Técnicas de recopilación de información con NSE**

* Métodos de autenticación SSH:
  + sudo nmap -p22 --script ssh-auth-methods <IP>
* Fuerza bruta a servidores DNS
  + sudo nmap -p 80 --script dns-brute.nse <TARGET URL O IP>
* Buscar hosts en una IP:
  + nmap -p 80 --script hostmap-bfk.nse <TARGET URL O IP>
* Geolocalización con traceroute:
  + sudo nmap --traceroute --script traceroute-geolocation.nse -p 80 <TARGET URL O IP>

**Recopilación de información HTTP con NSE**

Métodos HTTP -> descubre qué métodos admite un servidor mandando una petición options

sudo nmap -p80 – script http-methods <IP>

Captura del banner HTTP -> pregunta al servicio qué es

nmap --script banner <IP>

Títulos de servicios HTTP -> agrega los títulos de las páginas web a los resultados de un análisis de NMap para que se pueda tener un mejor contexto de un host que ejecuta servicios HTTP

nmap -- script http-title -sV -p 80 <IP>

Exploración de registros WHOIS -> busca datos de contacto

nmap --script whois-ip <IP>

Cuentas de correo electrónico -> theHarvester

Imagen que contiene Tabla

Descripción generada automáticamente

**Detección de malware y vulnerabilidades con NSE**

*Detección CVE mediante NMap. Scripts de terceros*

nmap -Pn --script vuln <IP> (volcar a un fichero, salida muy grande)

Para utilizar cualquier script de terceros, debes seguir este procedimiento:

• Descargar la carpeta o el archivo .nse del script en la carpeta de scripts NSE (/usr/share/nmap/scripts). En nuestro caso concreto sería hacer git clone https://github.com/vulnersCom/nmap-vulners.git una vez estemos en dicha carpeta.

• Actualizar la base de datos de scripts nmap con nmap --script-updatedb Una vez hecho esto, puedes ejecutar el script como cualquier script estándar que ya hemos visto. Para el caso que nos ocupa, simplemente hay que hacer sudo nmap -sV --script=nmap-vulners/ . Nótese la información adicional que este script aporta y para qué puede servirnos.

**Bloque 4. Más allá de NMap. Otras técnicas de enumeración**

**Wazuh y el inventario de los agentes**

Puedes localizar servicios y puertos abiertos de las máquinas monitorizadas por el XDR Wazuh.

Arranca la MV Wazuh vista en laboratorios anteriores y busca alguna opción del agente que hemos puesto en marcha para ver si encuentras la información mencionada, para poder analizar qué devuelve.

**Escaneando en busca de archivos concretos**

Hay archivos concretos que son útiles desde el punto de vista de enumeración, y por lo tanto descargarlos es muy importante para estudiar un objetivo. La parte buena es que ni siquiera necesitas un programa concreto para hacerlo, solo un simple script que puedes desarrollar con cualquier lenguaje de programación. El siguiente es un ejemplo para descargar el archivo phpinfo.php del rango IP 156.35.90.0 - 156.35.99.255)

#!/bin/bash for ip\_address in 156.35.9{0..9}.{0..255}; do wget -t 1 -T 5 http://${ip\_address}/phpinfo.php; done&

Esta técnica se puede utilizar para automatizar búsquedas de archivos interesantes en un rango de IPs de cualquier empresa, desde archivos conocidos como robots.txt, index.html, etc. hasta archivos potencialmente más comprometedores como backup.zip, backup.rar, archivos conocidos de ciertos CMS...

**Endpoints de servicios en archivos JavaScript**

**Buckets de Amazon S3**

AWS Simple Storage Service (también conocido como Amazon S3) es utilizado por empresas que no desean instalar y administrar sus propios repositorios de almacenamiento para determinados datos.

Este servicio puede almacenar objetos y archivos de diferentes tipos en la nube (servidor virtual) en lugar de en una máquina física. Los espacios de almacenamiento de esta tecnología se denominan buckets. El creador de un bucket puede almacenar en él código fuente, certificados, contraseñas, bases de datos...cualquier tipo de contenido. Desafortunadamente, a veces no están adecuadamente protegidos, y si obtienes acceso completo a un bucket de S3 puedes descargar, cargar o modificar cualquiera de los archivos que encuentres. Como estos suelen ser archivos con contenido muy sensible, los efectos suelen ser terribles y con frecuencia se producen noticias relacionadas con buckets S3 como la origen de ataques.

Hay maneras sencillas de encontrar cuándo un sitio web utiliza buckets de S3:

• Mediante el uso de una herramienta que explore la URLs de un sitio web, como la herramienta photon anterior o la famosa Burp Suite. Estas herramientas pueden encontrar URLs que terminan en s3.amazonaws.com que pueden llevarnos a encontrar un potencial bucket sin protección.

• Mediante el uso de la base de datos de Google Hacking del laboratorio 2, buscando "S3" para obtener una lista de búsquedas de Google que puedan revelar esta información

• Al buscar directamente estas URL tú mismo en Google: site:amazonaws.com inurl: (Ej. site:amazonaws.com inurl:uniovi) Una vez encontrados, se puede utilizar el paquete awscli en Kali para interactuar con ellos si es posible.

**Cheatsheets-resumen de NMap**

**Imagen que contiene Diagrama

Descripción generada automáticamente**

**Captura de pantalla con letras

Descripción generada automáticamente con confianza media**