3.2.1 Descripción General

Soluciones de seguridad Protego

Como parte del proceso de recopilación de información de pruebas de penetración, participará en el reconocimiento activo de la red interna de Pixel Paradise. Esto es lo que sabemos sobre la red.

La red Pixel Paradise incluye lo siguiente:

- Terminales de usuario: equipos de escritorio, portátiles, dispositivos móviles y consolas de juegos utilizados por los empleados.
- Terminales relacionados con las instalaciones: cámaras de vigilancia, un sistema de alarma, sensores y actuadores de control de clima, sistemas de control de iluminación y sistemas de VoIP.
- Dispositivos intermedios: enrutadores y conmutadores que conectan terminales y otras redes.
- Puntos de acceso inalámbricos: dispositivos que proporcionan acceso Wi-Fi a los empleados.
- Cortafuegos: dispositivos que protegen la red de amenazas externas y acceso no autorizado.
- Servidores: servidores en las instalaciones que proporcionan almacenamiento de archivos, acceso a bases de datos y otros servicios, como correo electrónico.
- Infraestructura de la nube: Amazon Web Services

Reconocimiento activo

Como se mencionó anteriormente en este módulo, con cada paso de la fase de recopilación de información, el objetivo es recopilar información adicional sobre el objetivo. El proceso de recopilación de esta información se denomina *enumeración.* Entonces, hablemos sobre el tipo de enumeración que normalmente haría en una prueba de penetración. En un ejemplo anterior, observamos la enumeración de hosts expuestos a Internet por h4cker.org. La enumeración externa de hosts suele ser una de las primeras cosas que hace en una prueba de penetración. Determinar los hosts conectados a Internet de una red de destino puede ayudarlo a identificar los sistemas más expuestos. Obviamente, un dispositivo al que se puede acceder públicamente a través de Internet está abierto a ataques de agentes maliciosos en todo el mundo. Después de identificar esos sistemas, debe identificar qué servicios son accesibles. Un servidor debe estar detrás de un cortafuegos, lo que permite una exposición mínima a los servicios que ejecuta. A veces, sin embargo, se exponen servicios inesperados. Para determinar si una red está ejecutando alguno de estos servicios, puede ejecutar un escaneo de puertos para enumerar los servicios que se ejecutan en los hosts expuestos.

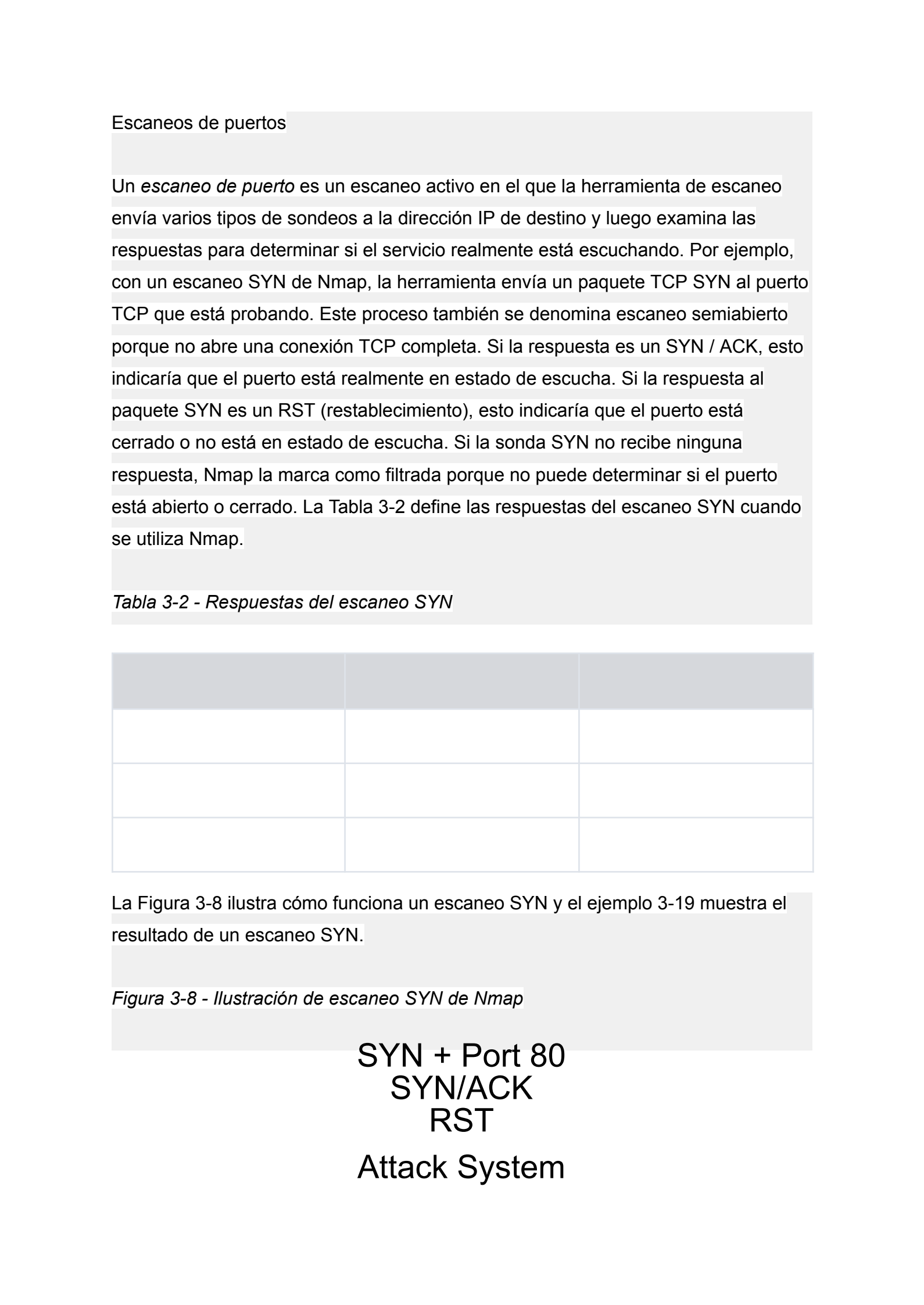
Escaneos de puertos

Un *escaneo de puerto* es un escaneo activo en el que la herramienta de escaneo envía varios tipos de sondeos a la dirección IP de destino y luego examina las respuestas para determinar si el servicio realmente está escuchando. Por ejemplo, con un escaneo SYN de Nmap, la herramienta envía un paquete TCP SYN al puerto TCP que está probando. Este proceso también se denomina escaneo semiabierto porque no abre una conexión TCP completa. Si la respuesta es un SYN / ACK, esto indicaría que el puerto está realmente en estado de escucha. Si la respuesta al paquete SYN es un RST (restablecimiento), esto indicaría que el puerto está cerrado o no está en estado de escucha. Si la sonda SYN no recibe ninguna respuesta, Nmap la marca como filtrada porque no puede determinar si el puerto está abierto o cerrado. La Tabla 3-2 define las respuestas del escaneo SYN cuando se utiliza Nmap.

Tabla 3-2 - Respuestas del escaneo SYN

La Figura 3-8 ilustra cómo funciona un escaneo SYN y el ejemplo 3-19 muestra el resultado de un escaneo SYN.

Figura 3-8 - Ilustración de escaneo SYN de Nmap



Ejemplo 3-19 - Salida de muestra de escaneo SYN de Nmap

```
|--[root@websploit]--[~]

|---- #nmap -sS 192.168.88.251

Starting Nmap 7.80 ( https://nmap.org )

Nmap scan report for 192.168.88.251

Host is up (0.00011s latency).

Not shown: 992 closed ports

PORT     STATE SERVICE

22/tcp   open  ssh

80/tcp   open  http

139/tcp  open  netbios-ssn

445/tcp  open  microsoft-ds

3306/tcp open  mysql

8888/tcp open  sun-answerbook

9000/tcp open  cslistener

9090/tcp open  zeus-admin

MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

El ejemplo 3-19 muestra cómo ejecutar un escaneo TCP SYN mediante Nmap especificando la opción -sS en un host con la dirección IP 192.168.88.251. Como puede ver, este sistema tiene varios puertos abiertos. En algunas situaciones, querrá utilizar las diferentes opciones de Nmap en sus escaneos para obtener los resultados que busca. Las siguientes secciones analizan algunas de las opciones y tipos de escaneo más comunes disponibles en Nmap.

3.2.2 Tipos de análisis de Nmap

Las siguientes secciones cubren algunas de las opciones de escaneos de Nmap más comunes utilizadas para escenarios específicos, incluidas las siguientes:

- Escaneo de conexión TCP (-sT)
- Escaneo de UDP (-sU)
- Escaneo FIN de TCP (-sF)
- escaneos de detección de host (-sn)
- Opciones de tiempo (-T 0-5)

Escaneo de conexión TCP (-sT)

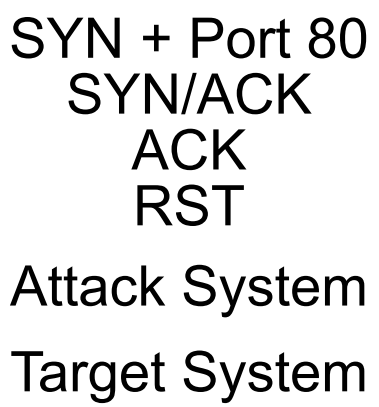
Un escaneo de conexión TCP realmente utiliza el mecanismo de red del sistema operativo subyacente para establecer una conexión TCP completa con el dispositivo de destino que se está escaneando. Dado que crea una conexión completa, crea más tráfico (y, por lo tanto, tarda más en ejecutarse). Este es el tipo de escaneo predeterminado que se utiliza si no se especifica ningún tipo de escaneo con el comando nmap. Sin embargo, generalmente se debe usar solo cuando un escaneo SYN no es una opción, como cuando un usuario que ejecuta el comando nmap no tiene privilegios de paquetes sin procesar en el sistema operativo porque muchos de los tipos de escaneo de Nmap dependen sobre la escritura de paquetes sin procesar. Esta sección ilustra cómo funciona un escaneo de conexión de TCP y

proporciona un ejemplo de un escaneo de un sistema Kali Linux. La Tabla 3-3 define las respuestas del escaneo de conexión de TCP.

Tabla 3-3 - Respuestas de escaneo de conexión de TCP

La Figura 3-9 ilustra cómo funciona un escaneo de conexión TCP.

Figura 3-9 - Ilustración de escaneo de TCP Connect



El ejemplo 3-20 muestra el resultado de un escaneo de conexión TCP completo.

Ejemplo 3-20 - Salida de muestra de escaneo de conexión TCP de Nmap

```
|--[root@websploit]--[~]

|--- #nmap -sT 192.168.88.251

Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 12:48
EDT
```

```
Nmap scan report for 192.168.88.251

Host is up (0.00024s latency).

Not shown: 992 closed ports

PORT     STATE SERVICE

22/tcp   open  ssh

80/tcp   open  http

139/tcp  open  netbios-ssn

445/tcp  open  microsoft-ds

3306/tcp open  mysql

8888/tcp open  sun-answerbook

9000/tcp open  cslistener

9090/tcp open  zeus-admin

MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)



Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
```

El resultado del ejemplo 3-20 muestra los resultados de un escaneo de conexión TCP de Nmap. Como puede ver, los resultados indican que varios puertos TCP están escuchando en el dispositivo de destino y estos resultados son muy similares al resultado que se muestra en el ejemplo 3-19.

Un escaneo de conexión TCP completo requiere que el escáner envíe un paquete adicional por escaneo, lo que aumenta la cantidad de ruido en la red y puede activar alarmas que un escaneo medio abierto no activaría. Las herramientas de seguridad y el sistema objetivo subyacente tienen más probabilidades de registrar una conexión TCP completa, y los sistemas de detección de intrusiones (IDS) tienen más probabilidades de activar alarmas en varias conexiones TCP desde el mismo host.

CONSEJO Nmap escanea solo los 1000 puertos más comunes para cada protocolo. Puede especificar puertos adicionales para escanear mediante la opción -p. Puede obtener información adicional sobre las especificaciones del puerto y el orden de escaneo en *https://nmap.org/book/man-port-specification.html*. Omar Santos también ha creado una hoja de referencia de Nmap que incluye todas las opciones y está disponible en su repositorio de GitHub, *https://github.com/The-Art-of-Hacking/h4cker/blob/master/cheat_sheets/NMAP_cheat_sheet.md*.

Escaneo UDP (-sU)

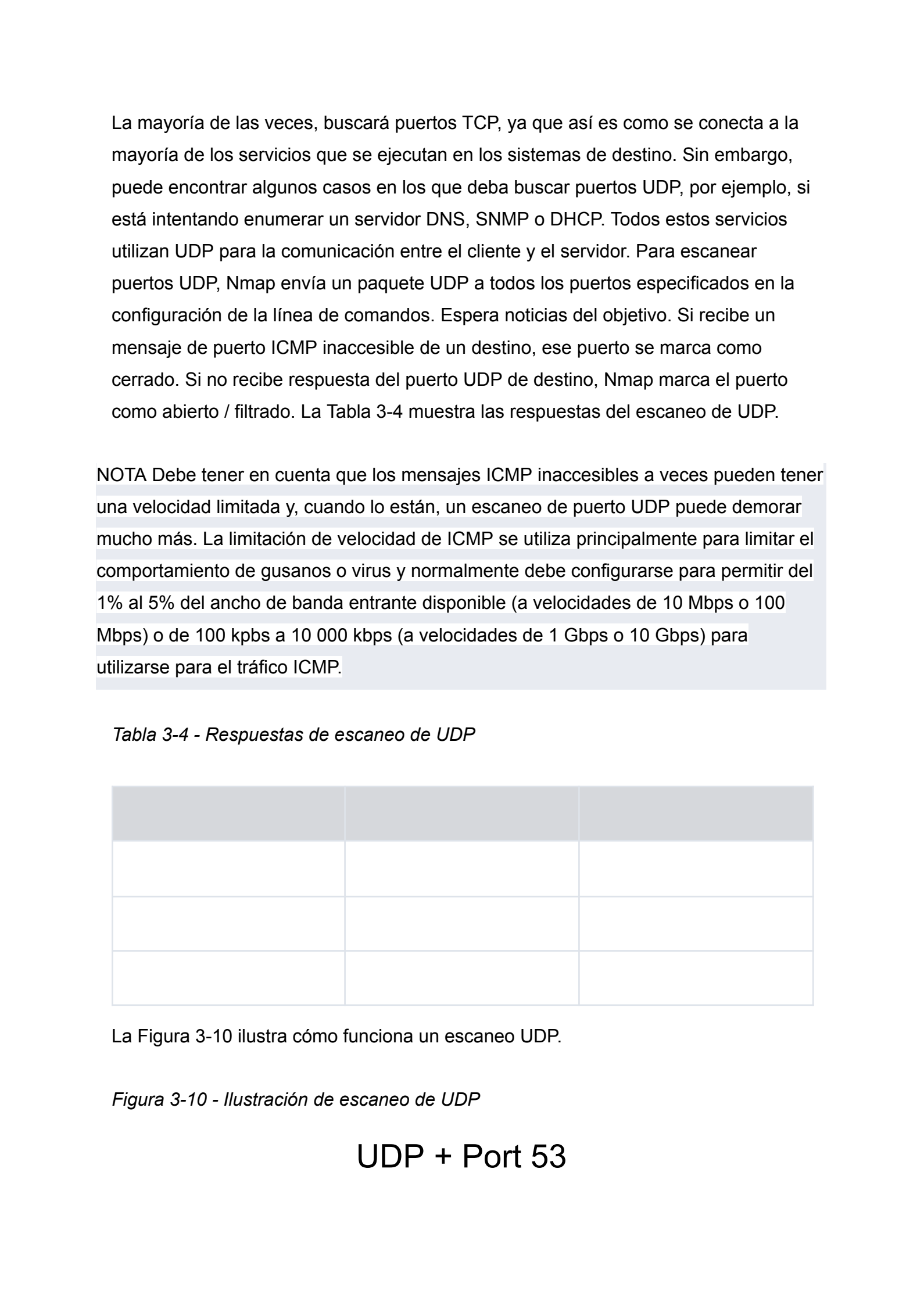
La mayoría de las veces, buscará puertos TCP, ya que así es como se conecta a la mayoría de los servicios que se ejecutan en los sistemas de destino. Sin embargo, puede encontrar algunos casos en los que deba buscar puertos UDP, por ejemplo, si está intentando enumerar un servidor DNS, SNMP o DHCP. Todos estos servicios utilizan UDP para la comunicación entre el cliente y el servidor. Para escanear puertos UDP, Nmap envía un paquete UDP a todos los puertos especificados en la configuración de la línea de comandos. Espera noticias del objetivo. Si recibe un mensaje de puerto ICMP inaccesible de un destino, ese puerto se marca como cerrado. Si no recibe respuesta del puerto UDP de destino, Nmap marca el puerto como abierto / filtrado. La Tabla 3-4 muestra las respuestas del escaneo de UDP.

NOTA Debe tener en cuenta que los mensajes ICMP inaccesibles a veces pueden tener una velocidad limitada y, cuando lo están, un escaneo de puerto UDP puede demorar mucho más. La limitación de velocidad de ICMP se utiliza principalmente para limitar el comportamiento de gusanos o virus y normalmente debe configurarse para permitir del 1% al 5% del ancho de banda entrante disponible (a velocidades de 10 Mbps o 100 Mbps) o de 100 kpbs a 10 000 kbps (a velocidades de 1 Gbps o 10 Gbps) para utilizarse para el tráfico ICMP.

Tabla 3-4 - Respuestas de escaneo de UDP

La Figura 3-10 ilustra cómo funciona un escaneo UDP.

Figura 3-10 - Ilustración de escaneo de UDP



UDP + Port 53 Data
Attack System
Target System

El resultado del ejemplo 3-21 muestra los resultados de un escaneo de UDP de Nmap en el puerto 53 del destino 192.168.88.251. Como puede ver, los resultados indican que este puerto está abierto.

Ejemplo 3-21 - Salida de muestra de escaneo de Nmap UDP

```
|--[root@websploit]--[~]

|--- # nmap -sU -p 53 192.168.88.251

Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 13:12
EDT

Nmap scan report for 192.168.88.251

Host is up (0.00057s latency).

PORT   STATE  SERVICE

53/udp open domain

MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Escaneo FIN de TCP (-sF)

Hay momentos en los que un filtro de red o cortafuegos puede detectar un escaneo de sincronización. En tal caso, debe emplear un tipo diferente de paquete en un escaneo de puertos. Con el escaneo TCP FIN, se envía un paquete FIN a un puerto de destino. Si el puerto está realmente cerrado, el sistema de destino devuelve un paquete RST. Si no se recibe nada del puerto de destino, puede considerar el puerto abierto porque el comportamiento normal sería ignorar el paquete FIN. La Tabla 3-5 muestra las respuestas del escaneo FIN de TCP.

NOTA un escaneo FIN de TCP no es útil cuando se analizan sistemas basados en Windows, ya que responden con paquetes RST, independientemente del estado del puerto.

Tabla 3-5 - Respuestas de escaneo FIN de TCP

La Figura 3-11 ilustra cómo funciona un escaneo TCP FIN.

Figura 3-11 - Ilustración de escaneo de TCP FIN

FIN + Port 80

No Response

Attack System

Target System

Port is likely open.

FIN + Port 80

RST

Attack System

Target System

Port is likely closed.

El ejemplo 3-22 muestra el resultado de un escaneo FIN de TCP en el puerto 80 del destino. La salida muestra los resultados de un escaneo FIN de TCP de Nmap, que especifica el puerto 80 en el destino. La respuesta del destino indica que el puerto está abierto / filtrado.

Ejemplo 3-22 - Salida de muestra de escaneo FIN de Nmap TCP

```
|--[root@websploit]--[~]

|--- # nmap -sF -p 80 192.168.88.251

Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 13:15
EDT

Nmap scan report for 192.168.88.251

Host is up (0.00045s latency).

PORT    STATE           SERVICE

80/tcp open|filtered http

MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds

|--[root@websploit]--[~]

|--- #
```

escaneo de detección de host (-sn)

un escaneo de detección de host es uno de los tipos más comunes de escaneos utilizados para enumerar hosts en una red porque puede usar diferentes tipos de mensajes ICMP para determinar si un host está en línea y responde en una red.

NOTA El valor predeterminado para la opción de escaneo -sn es enviar un paquete de solicitud de eco ICMP al destino, un TCP SYN al puerto 443, un TCP ACK al puerto 80 y una solicitud de marca de tiempo ICMP. Esto está documentado en *https://nmap.org/book/man-host-discovery.html*. Si el objetivo responde al eco de ICMP o a los paquetes mencionados anteriormente, se considera activo.

El ejemplo 3-23 muestra un ejemplo de un escaneo de ping de la subred 192.168.88.0/24. Se trata de un escaneo de detección de host muy básico que se puede realizar para determinar qué dispositivos de una red están activos. un escaneo de este tipo para la detección de host de una subred completa a veces se denomina *barrido de ping*.

Ejemplo 3-23 - Escaneo de detección de host de Nmap

```
|--[root@websploit]--[~]

|---- #nmap -sn 192.168.88.0/24

Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 14:32
EDT

Nmap scan report for 192.168.88.1

Host is up (0.00045s latency).

MAC Address: E0:55:3D:E9:61:74 (Cisco Meraki)
```

```
Nmap scan report for 192.168.88.12

Host is up (0.00094s latency).

MAC Address: 0E:64:AF:27:9C:44 (Unknown)

Nmap scan report for 192.168.88.14

Host is up (0.0092s latency).

MAC Address: 00:B8:B3:FD:BF:C2 (Cisco Systems)

Nmap scan report for 192.168.88.24

Host is up (0.0033s latency).

MAC Address: 00:E1:6D:E5:43:C2 (Cisco Systems)

Nmap scan report for 192.168.88.32

Host is up (0.00046s latency).

MAC Address: BE:38:F5:2D:6C:C0 (Unknown)

Nmap scan report for 192.168.88.231

Host is up (0.00061s latency).

MAC Address: FE:82:8C:A3:D2:3C (Unknown)

Nmap scan report for 192.168.88.251

Host is up (0.00040s latency).

MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)

Nmap scan report for 192.168.88.71
```

```
Host is up.

Nmap scan report for 192.168.88.225

Host is up.

Nmap done: 256 IP addresses (11 hosts up) scanned in 2.45
seconds

|--[root@websploit]--[~]

|--- #
```

Opciones de sincronización (-T 0-5)

El escáner de Nmap proporciona seis plantillas de temporización que se pueden especificar con la opción -T y el número de plantilla (de 0 a 5) o el nombre. Las plantillas de sincronización de Nmap le permiten determinar cuán agresivo será un escaneo, mientras deja que Nmap elija los valores de sincronización exactos. Estas son las opciones de tiempo:

- -T0 (Paranoico) : muy lento, se usa para la evasión de IDS
- -T1 (Furtivo) : bastante lento, se usa para la evasión de IDS
- -T2 (Educado) : se ralentiza para consumir menos ancho de banda, se ejecuta aproximadamente 10 veces más lento que el valor predeterminado
- -T3 (Normal) : predeterminado, un modelo de tiempo dinámico basado en la capacidad de respuesta del objetivo
- -T4 (Agresivo) : supone una red rápida y confiable y puede abrumar a los objetivos
- -T5 (Demente) : muy agresivo; probablemente abrumará a los objetivos o perderá los puertos abiertos

NOTA El modo normal es el modo Nmap predeterminado. Si utiliza este modo (especificando - T3), no verá ninguna diferencia con respecto a un escaneo normal.

Puede encontrar información adicional sobre las opciones de sincronización y el rendimiento de Nmap en *https://nmap.org/book/man-performance.html*.

3.2.3 Práctica - Tipos de análisis de Nmap

Pregunta 1

Un técnico de ciberseguridad está utilizando la herramienta Nmap para realizar un escaneo de puertos con el comando:

```
nmap -sn 172.16.50.0/24
```

¿Qué tipo de escaneo de puertos está utilizando el técnico?

Escaneo de conexión TCP

escaneo de detección de host

Escaneo UDP

Escaneo FIN de TCP

done

Enviar

Mostrar retroalimentación

Pregunta 2

Un atacante está utilizando la herramienta Nmap para realizar un escaneo de puertos con el comando:

```
nmap -sn -T0 172.18.12.0/24
```

¿Cuál es el propósito de usar la opción -T0?

realizar el escaneo de manera agresiva

evitar ser detectado por IDS o IPS

ejecutar el escaneo unas 10 veces más lento que el predeterminado

detener el escaneo después de un período de tiempo especificado por T0

done

Enviar

Mostrar retroalimentación

3.2.4 Tipos de enumeración

Esta sección abarca las técnicas de enumeración que deben realizarse en la fase de recopilación de información de una prueba de penetración. Aprenderá cómo y cuándo deben utilizarse estas técnicas de enumeración. Esta sección también incluye ejemplos de cómo realizar estos tipos de enumeración mediante Nmap, así como un escaneo profundo de la elaboración de paquetes con Scapy.

- Enumeración de host
- Enumeración de usuarios
- Enumeración de grupos
- Enumeración de recursos compartidos de red
- Ejemplos de enumeración de SMB adicionales
- Enumeración de páginas web / Enumeración de aplicaciones web
- Enumeración de servicios
- Exploración de la enumeración a través de la elaboración de paquetes

Enumeración de host

La enumeración de hosts es una de las primeras tareas que debe realizar en la fase de recopilación de información de una prueba de penetración. *La enumeración de hosts* se realiza interna y externamente. Cuando se realiza de manera externa, generalmente desea limitar las direcciones IP que está escaneando solo a las que forman parte del alcance de la prueba. Esto reduce la posibilidad de escanear inadvertidamente una dirección IP que no está autorizado a probar. Al realizar una enumeración de host interna, normalmente se escanea la subred o subredes completas de direcciones IP utilizadas por el destino. La enumeración de hosts generalmente se realiza mediante una herramienta como Nmap o Masscan; sin embargo, los escáneres de vulnerabilidades también realizan esta tarea como parte de sus pruebas automatizadas. El ejemplo 3-23, anteriormente en este módulo, muestra un ejemplo de escaneo de ping de Nmap que se utiliza para la enumeración de hosts en la red 192.168.88.0/24. En versiones anteriores de Nmap, la opción de escaneo de ping de Nmap era -sP (no -sn).

Enumeración de usuarios

Recopilar una lista válida de usuarios es el primer paso para descifrar un conjunto de credenciales. Cuando tenga el nombre de usuario, puede iniciar intentos de fuerza bruta para obtener la contraseña de la cuenta. La *enumeración de usuarios* se realiza cuando se obtiene acceso a la red interna. En una red de Windows, puede hacerlo mediante la manipulación del protocolo Server Message Block (SMB), que utiliza el puerto TCP 445. La Figura 3-12 ilustra cómo funciona una implementación típica de una PYME.

Figura 3-12 - Ilustración de mensaje de SMB

SMB_COM_NEGOTIATE (Request)
SMB_COM_NEGOTIATE (Response)
SMB_COM_SESSION_SETUP_ANDX (Request)
SMB_COM_SESSION_SETUP_ANDX (Response)

SMB_COM_TREE_CONECT_ANDX (Request)
SMB_COM_TREE_CONECT_ANDX (Response)
SMB Client
SMB Server

La información contenida en las respuestas a estos mensajes le permite revelar información sobre el servidor:

- SMB_COM_NEGOTIATE: Este mensaje permite al cliente indicarle al servidor qué protocolos, indicadores y opciones le gustaría usar. La respuesta del servidor también es un mensaje SMB_COM_NEGOTIATE. Esta respuesta se transmite al cliente sobre qué protocolos, indicadores y opciones prefiere. Esta información se puede configurar en el propio servidor. Una configuración incorrecta a veces revela información que puede utilizar en las pruebas de penetración. Por ejemplo, el servidor puede estar configurado para permitir mensajes sin firmas. Puede determinar si el servidor utiliza mecanismos de autenticación a nivel de recurso compartido o de usuario y si el servidor permite contraseñas de texto sin formato. La respuesta del servidor también proporciona información adicional, como la hora y la zona horaria que utiliza el servidor. Esta es información necesaria para muchas tareas de pruebas de penetración.
- SMB_COM_SESSION_SETUP_ANDX: después de que el cliente y el servidor hayan negociado los protocolos, los indicadores y las opciones que usarán para la comunicación, comienza el proceso de autenticación. La autenticación es la función principal del mensaje SMB_COM_SESSION_SETUP_ANDX. La información enviada en este mensaje incluye el nombre de usuario, la contraseña y el dominio del cliente. Si esta información no está cifrada, es fácil detectarla directamente fuera de la red. Incluso si está cifrada, si el mecanismo utilizado no es suficiente, la información puede revelarse mediante herramientas como Lanman y NTLM en el caso de las implementaciones de Microsoft Windows. En el siguiente ejemplo, se muestra el uso de este mensaje con el guion smb-enum-users.nse:
  `nmap --script smb-enum-users.nse <host>`

El ejemplo 3-24 muestra los resultados del guion Nmap smb-enum-users ejecutado contra el destino 192.168.88.251. Como puede ver, los resultados indican que el guion pudo enumerar los usuarios configurados en este destino de Windows.La línea resaltada revela el usuario enumerado por Nmap (derek).

Ejemplo 3-24 - Enumeración de usuarios de PYMES

```
|--[root@websploit]--[~]

|--- #nmap  --script smb-enum-users.nse 192.168.88.251

Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-22 11:14
EDT

Nmap scan report for 192.168.88.251

Host is up (0.012s latency).

Not shown: 992 closed ports

PORT     STATE SERVICE

22/tcp   open  ssh

80/tcp   open  http

139/tcp  open  netbios-ssn

445/tcp  open  microsoft-ds

3306/tcp open  mysql

8888/tcp open  sun-answerbook

9000/tcp open  cslistener

9090/tcp open  zeus-admin

Host script results:

| smb-enum-users:

|     VULNHOST-1\derek (RID: 1000)
```

```
|       Full name:

|       Description:

|_      Flags:          Normal user account

Nmap done: 1 IP address (1 host up) scanned in 0.81 seconds
```

Enumeración de grupos

Para un evaluador de penetración, la *enumeración de grupo* es útil para determinar los roles de autorización que se utilizan en el entorno de destino. El guion de Nmap NSE para enumerar grupos de SMB es smb-enum-groups. Este guion intenta extraer una lista de grupos de una máquina Windows remota. También puede revelar la lista de usuarios que son miembros de esos grupos. La sintaxis de los comandos es la siguiente:

```
nmap --script smb-enum-groups.nse -p445 <host>
```

El ejemplo 3-25 muestra el resultado de muestra de este comando que se ejecuta en el servidor de Windows en 192.168.56.3. Este ejemplo utiliza credenciales conocidas para recopilar información.

Ejemplo 3-25 - Enumeración de grupos de SMB

```
|--[root@websploit]--[~]

|--- # nmap --script smb-enum-groups.nse --script-args
smbusername=vagrant,smbpass=vagrant 192.168.56.3

Starting Nmap 7.91 ( https://nmap.org )

Nmap scan report for 192.168.56.3
```

```
Host is up (0.0062s latency).

Not shown: 979 closed ports

PORT      STATE SERVICE

22/tcp    open  ssh

135/tcp   open  msrpc

139/tcp   open  netbios-ssn

445/tcp   open  microsoft-ds

3306/tcp  open  mysql

3389/tcp  open  ms-wbt-server

MAC Address: 08:00:27:1B:A4:60 (Oracle VirtualBox virtual NIC)

Host script results:

| smb-enum-groups:

|   Builtin\Administrators (RID: 544): Administrator, vagrant,
sshd_server

|   Builtin\Users (RID: 545): vagrant, sshd, sshd_server,
leia_organa,

luke_skywalker, han_solo, artoo_detoo, c_three_pio,
ben_kenobi, darth_

vader, anakin_skywalker, jarjar_binks, lando_calrissian,
boba_fett,

jabba_hutt, greedo, chewbacca, kylo_ren
```

```
|     Builtin\Guests (RID: 546): Guest, ben_kenobi

|     Builtin\Power Users (RID: 547): boba_fett

|     Builtin\Print Operators (RID: 550): jabba_hutt

|     Builtin\Backup Operators (RID: 551): leia_organa

|     Builtin\Replicator (RID: 552): chewbacca

|     Builtin\Remote Desktop Users (RID: 555): greedo

|     Builtin\Network Configuration Operators (RID: 556):
anakin_skywalker

|     Builtin\Performance Monitor Users (RID: 558):
lando_calrissian

|     Builtin\Performance Log Users (RID: 559): jarjar_binks

|     Builtin\Distributed COM Users (RID: 562): artoo_detoo

|     Builtin\IIS_IUSRS (RID: 568): darth_vader

|     Builtin\Cryptographic Operators (RID: 569): han_solo

|     Builtin\Event Log Readers (RID: 573): c_three_pio

|     Builtin\Certificate Service DCOM Access (RID: 574):
luke_skywalker

|_    VAGRANT-2008R2\WinRMRemoteWMIUsers__ (RID: 1003): <empty>

Nmap done: 1 IP address (1 host up) scanned in 0.81 seconds

|--[root@websploit]--[~]

|--- #
```

El resultado resaltado en el ejemplo 3-25 muestra los grupos y usuarios enumerados en el host de destino. En Windows, el identificador relativo (RID) es un número de longitud variable asignado a objetos y se convierte en parte del identificador de seguridad (SID) del objeto que identifica de forma exclusiva una cuenta o un grupo dentro de un dominio. Para obtener más información sobre los diferentes números de RID, consulte *https://docs.microsoft.com/en-us/troubleshoot/windows-server/identity/security-identifiers-in-windows*.

Enumeración de recursos compartidos de red

Identificar los sistemas en una red que comparten archivos, carpetas e impresoras es útil para desarrollar una superficie de ataque de una red interna. El guion de NSE Nmap smb-enum-participas utiliza la llamada a procedimiento remoto de Microsoft (MSRPC) para la *enumeración de recursos compartidos de red*. La sintaxis del guion Nmap smb-enum-Share.nse es la siguiente:

```
nmap --script smb-enum-shares.nse -p 445 <host>
```

El ejemplo 3-26 muestra la enumeración de recursos compartidos de SMB.

Ejemplo 3-26 - Enumeración de recursos compartidos de SMB

```
|--[root@websploit]--[~]

|--- # nmap --script smb-enum-shares.nse -p 445 192.168.88.251

Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-22 11:27
EDT

Nmap scan report for 192.168.88.251

Host is up (0.0011s latency).
```

```
PORT    STATE SERVICE

445/tcp open  microsoft-ds


Host script results:

| smb-enum-shares:

|   account_used: guest

|   \\192.168.88.251\IPC$:

|     Type: STYPE_IPC_HIDDEN

|     Comment: IPC Service (Samba 4.9.5-Debian)

|     Users: 1

|     Max Users: <unlimited>

|     Path: C:\tmp

|     Anonymous access: READ/WRITE

|     Current user access: READ/WRITE

|   \\192.168.88.251\print$:

|     Type: STYPE_DISKTREE

|     Comment: Printer Drivers

|     Users: 0
```

```
|      Max Users: <unlimited>

|      Path: C:\var\lib\samba\printers

|      Anonymous access: <none>

|      Current user access: <none>

|    \\192.168.88.251\secret_folder:

|      Type: STYPE_DISKTREE

|      Comment: Extremely sensitive information

|      Users: 0

|      Max Users: <unlimited>

|      Path: C:\secret_folder

|      Anonymous access: <none>

|_     Current user access: <none>


Nmap done: 1 IP address (1 host up) scanned in 0.39 seconds

|--[root@websploit]--[~]

|--- #
```

Ejemplos de enumeración de SMB adicionales

El sistema utilizado en los ejemplos anteriores (con la dirección IP 192.168.88.251) ejecuta Linux y Samba. Sin embargo, no es fácil determinar que se trata de un sistema Linux a partir de los resultados de escaneos anteriores. Una manera fácil de realizar enumeraciones y huellas digitales adicionales de las aplicaciones y el sistema operativo que se ejecutan en un host es mediante el comando nmap -sC. La opción - sC ejecuta los guiones o scripts de NSE más comunes según los puertos abiertos en el sistema de destino.

NOTA Puede localizar los scripts de NSE instalados en Kali Linux y Parrot OS simplemente con el comando locate * .nse. El sitio *https://nmap.org/book/man-nse.html* incluye una explicación detallada de la NSE y cómo crear nuevos guiones con el lenguaje de programación Lua.

El ejemplo 3-27 muestra el resultado del comando nmap -sC iniciado en el sistema Linux en 192.168.88.251, que ejecuta Samba.

Ejemplo 3-27 - Ejecución de los guiones predeterminados de Nmap NSE

```
|--[root@websploit]--[~]

|--- # nmap -sC 192.168.88.251

Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-21 17:38
EDT

Nmap scan report for 192.168.88.251

Host is up (0.00011s latency).

Not shown: 992 closed ports

PORT     STATE SERVICE

22/tcp   open  ssh

| ssh-hostkey:
```

```
|   2048 d0:0c:83:4d:7f:84:2c:60:96:9f:df:26:da:d2:11:9a (RSA)

|   256 e2:aa:69:ab:a3:e6:0f:13:c5:5a:65:f2:d5:16:8c:3e
(ECDSA)

|_  256 21:4b:27:7b:6e:a6:d4:33:86:60:cb:39:3b:48:9c:0b
(ED25519)

80/tcp   open  http

|_http-title: WebSploit Mayhem

139/tcp  open  netbios-ssn

445/tcp  open  microsoft-ds

3306/tcp open  mysql

| mysql-info:

|   Protocol: 10

|   Version: 5.5.47-0ubuntu0.14.04.1

|   Thread ID: 3

|   Capabilities flags: 63487

|   Some Capabilities: InteractiveClient,

DontAllowDatabaseTableColumn, FoundRows, IgnoreSigpipes,

Support41Auth, ODBCClient, ConnectWithDatabase, LongPassword,

SupportsTransactions, IgnoreSpaceBeforeParenthesis,

Speaks41ProtocolOld, Speaks41ProtocolNew, SupportsCompression,
```

```
SupportsLoadDataLocal, LongColumnFlag,
SupportsMultipleResults,

SupportsMultipleStatments, SupportsAuthPlugins

|   Status: Autocommit

|   Salt: b_60.4ZH=52:15ajmhBP

|_  Auth Plugin Name: mysql_native_password

8888/tcp open  sun-answerbook

9000/tcp open  cslistener

9090/tcp open  zeus-admin

MAC Address: 1E:BD:4F:AA:C6:BA (Unknown)

Host script results:

|_clock-skew: mean: 17s, deviation: 0s, median: 17s

|_nbstat: NetBIOS name: VULNHOST-1, NetBIOS user: <unknown>,
NetBIOS

MAC: <unknown> (unknown)

| smb-os-discovery:

|   OS: Windows 6.1 (Samba 4.9.5-Debian)

|   Computer name: vulnhost-1

|   NetBIOS computer name: VULNHOST-1\x00

|   Domain name: ohmr.org
```

```
|   FQDN: vulnhost-1.ohmr.org
|_  System time: 2022-06-21T21:38:40+00:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-06-21T21:38:40
|_  start_date: N/A
Nmap done: 1 IP address (1 host up) scanned in 28.77 seconds
|--[root@websploit]--[~]
|--- #
```

Las líneas resaltadas en el ejemplo 3-27 muestran detalles sobre la versión de Samba que se ejecuta en el sistema (versión de Samba 4.9.5). También puede ver

que aunque el SO está marcado como Windows 6.1, el sistema operativo correcto es Debian. El ejemplo 3-28 muestra la salida del comando samba -V en el sistema de destino (vulnhost-1), que confirma que el analizador pudo determinar la versión correcta de Samba.

Ejemplo 3-28 - Confirmación de resultados del escaneo en el sistema de destino

```
omar@vulnhost-1:~$ sudo samba -V

Version 4.9.5-Debian

omar@vulnhost-1:~$
```

También puede utilizar herramientas como enum4linux para enumerar los recursos compartidos de Samba, incluidas las cuentas de usuario, los recursos compartidos y otras configuraciones. El ejemplo 3-29 muestra el resultado de la herramienta enum4linux después de iniciarse en el sistema de destino (192.168.88.251).

Ejemplo 3-29 - Enumeración de información adicional con enum4linux

```
|-- [root@websploit]--[~]

|--- # enum4linux 192.168.88.251

Starting enum4linux v0.8.9 (
http://labs.portcullis.co.uk/application/enum4linux/ )

 ==========================

|    Target Information    |

 ==========================
```

```
Target ..... 192.168.88.251

RID Range ..... 500-550,1000-1050



Username ..... ''

Password ..... ''

Known Usernames .. administrator, guest, krbtgt, domain
admins, root, bin, none

 ============================================================

|    Enumerating Workgroup/Domain on 192.168.88.251    |

 ============================================================

[+] Got domain/workgroup name: WORKGROUP

 ================================================

|    Nbtstat Information for 192.168.88.251    |

 ================================================

Looking up status of 192.168.88.251

  VULNHOST-1       <00> -         B <ACTIVE>  Workstation
Service

  VULNHOST-1       <03> -         B <ACTIVE>  Messenger
Service

  VULNHOST-1       <20> -         B <ACTIVE>  File Server
Service
```

```
..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>  Master Browser

  WORKGROUP       <00> - <GROUP> B <ACTIVE>  Domain/Workgroup
Name

  WORKGROUP       <1d> -         B <ACTIVE>  Master Browser

  WORKGROUP       <1e> - <GROUP> B <ACTIVE>  Browser Service

Elections

  MAC Address = 00-00-00-00-00-00

 =========================================

|    Session Check on 192.168.88.251    |

 =========================================

[+] Server 192.168.88.251 allows sessions using username '',
password ''

 ================================================

|    Getting domain SID for 192.168.88.251    |

 ================================================

Domain Name: WORKGROUP

Domain Sid: (NULL SID)

[+] Can't determine if host is part of domain or part of a
workgroup

 =========================================

|     OS information on 192.168.88.251    |
```

```
 ========================================

Use of uninitialized value $os_info in concatenation (.) or
string at ./enum4linux.pl line 464.

[+] Got OS info for 192.168.88.251 from smbclient:

[+] Got OS info for 192.168.88.251 from srvinfo:

    VULNHOST-1     Wk Sv PrQ Unx NT SNT Samba 4.9.5-Debian

    platform_id     : 500

    os version      : 6.1

    server type     : 0x809a03

 ================================

|    Users on 192.168.88.251    |

 ================================

index: 0x1 RID: 0x3e8 acb: 0x00000010 Account: derek Name:
Desc:

user:[derek] rid:[0x3e8]

 ============================================

|    Share Enumeration on 192.168.88.251    |

 ============================================

    Sharename       Type      Comment

    ---------       ----      -------
```

```
    print$          Disk      Printer Drivers
secret_folder   Disk      Extremely sensitive information
    IPC$            IPC       IPC Service (Samba 4.9.5-Debian)
SMB1 disabled -- no workgroup available
[+] Attempting to map shares on 192.168.88.251
//192.168.88.251/print$ Mapping: DENIED, Listing: N/A
//192.168.88.251/secret_folder Mapping: DENIED, Listing: N/A
=====================================================
|    Password Policy Information for 192.168.88.251    |
 =====================================================
[+] Attaching to 192.168.88.251 using a NULL share
[+] Trying protocol 139/SMB...
[+] Found domain(s):
    [+] VULNHOST-1
    [+] Builtin
[+] Password Info for Domain: VULNHOST-1
    [+] Minimum password length: 5
    [+] Password history length: None
    [+] Maximum password age: 37 days 6 hours 21 minutes
```

```
    [+] Password Complexity Flags: 000000

        [+] Domain Refuse Password Change: 0

        [+] Domain Password Store Cleartext: 0

        [+] Domain Password Lockout Admins: 0

        [+] Domain Password No Clear Change: 0

        [+] Domain Password No Anon Change: 0

        [+] Domain Password Complex: 0

    [+] Minimum password age: None

    [+] Reset Account Lockout Counter: 30 minutes

    [+] Locked Account Duration: 30 minutes

    [+] Account Lockout Threshold: None

    [+] Forced Log off Time: 37 days 6 hours 21 minutes

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled

Minimum Password Length: 5

 ================================

|    Groups on 192.168.88.251    |

 ================================

[+] Getting builtin groups:
```

```
[+] Getting builtin group memberships:

[+] Getting local groups:

[+] Getting local group memberships:

[+] Getting domain groups:

[+] Getting domain group memberships:

=============================================================

|    Users on 192.168.88.251 via RID cycling (RIDS: 500-550, 1000-1050)    |

=============================================================

[I] Found new SID: S-1-22-1

[I] Found new SID: S-1-5-21-2226316658-154127331-1048156596

[I] Found new SID: S-1-5-32

[+] Enumerating users using SID S-1-5-21-2226316658-154127331-1048156596 and logon username '', password ''

<output omitted for brevity>

S-1-5-21-2226316658-154127331-1048156596-501 VULNHOST-1\nobody (Local User)

S-1-5-21-2226316658-154127331-1048156596-513 VULNHOST-1\None (Domain Group)

S-1-5-21-2226316658-154127331-1048156596-1000 VULNHOST-1\derek (Local User)
```

```
<output omitted for brevity>

[+] Enumerating users using SID S-1-22-1 and logon username
'', password ''

S-1-22-1-1000 Unix User\omar (Local User)

S-1-22-1-1001 Unix User\derek (Local User)

[+] Enumerating users using SID S-1-5-32 and logon username
'', password ''

<output omitted for brevity>

==============================================

|    Getting printer info for 192.168.88.251    |

==============================================

No printers returned.

|--[root@websploit]--[~]

|--- #
```

Hay una implementación de enum4linux basada en Python llamada enum4linux-ng que se puede descargar de https://github.com/cddmp/enum4linux-ng.

El ejemplo 3-30 muestra un ejemplo de enumeración de SMB con enum4linux-ng.

Ejemplo 3-30 - Enumeración con enum4linux-ng

```
|--[root@websploit]--[~/enum4linux-ng]
```

```
|--- # ./enum4linux-ng.py -As 192.168.88.251

ENUM4LINUX - next generation


 ==========================

|    Target Information    |

 ==========================

[*] Target ........... 192.168.88.251

[*] Username ......... ''

[*] Random Username .. 'opaftohf'

[*] Password ......... ''

[*] Timeout .......... 5 second(s)


 =====================================

|    Service Scan on 192.168.88.251    |

 =====================================

[*] Checking LDAP

[-] Could not connect to LDAP on 389/tcp: connection refused

[*] Checking LDAPS

[-] Could not connect to LDAPS on 636/tcp: connection refused
```

```
[*] Checking SMB

[+] SMB is accessible on 445/tcp

[*] Checking SMB over NetBIOS

[+] SMB over NetBIOS is accessible on 139/tcp

 ========================================

|    SMB Dialect Check on 192.168.88.251    |

 ========================================

[*] Check for legacy SMBv1 on 445/tcp

[+] Server supports dialects higher SMBv1

 ========================================

|    RPC Session Check on 192.168.88.251    |

 ========================================

[*] Check for null session

[+] Server allows session using username '', password ''

[*] Check for random user session

[+] Server allows session using username 'opaftohf', password
''

[H] Rerunning enumeration with user 'opaftohf' might give more
results

 ====================================================
```

```
|    Domain Information via RPC for 192.168.88.251     |
 =====================================================
[+] Domain: WORKGROUP
[+] SID: NULL SID
[+] Host is part of a workgroup (not a domain)
 =================================================
|    OS Information via RPC on 192.168.88.251     |
 =================================================
[+] The following OS information were found:
server_type_string = Wk Sv PrQ Unx NT SNT Samba 4.9.5-Debian
platform_id        = 500
os_version         = 6.1
server_type        = 0x809a03
os                 = Linux/Unix (Samba 4.9.5-Debian)
 =======================================
|    Users via RPC on 192.168.88.251    |
 =======================================
[*] Enumerating users via 'querydispinfo'
[+] Found 2 users via 'querydispinfo'
```

```
[*] Enumerating users via 'enumdomusers'
[+] Found 2 users via 'enumdomusers'
[+] After merging user results we have 2 users total:
'1000':
  username: derek
  name: ''
  acb: '0x00000010'
  description: ''
'1001':
  username: omar
  name: ''
  acb: '0x00000010'
  description: ''
 ============================================
|    Groups via RPC on 192.168.88.251     |
 ============================================
[*] Enumerating local groups
[+] Found 0 group(s) via 'enumalsgroups domain'
[*] Enumerating builtin groups
```

```
[+] Found 0 group(s) via 'enumalsgroups builtin'

[*] Enumerating domain groups

[+] Found 0 group(s) via 'enumdomgroups'

 ========================================

|    Shares via RPC on 192.168.88.251    |

 ========================================

[*] Enumerating shares

[+] Found 3 share(s):

IPC$:

  comment: IPC Service (Samba 4.9.5-Debian)

  type: IPC

print$:

  comment: Printer Drivers

  type: Disk

secret_folder:

comment: Extremely sensitive information

  type: Disk

[*] Testing share IPC$

[-] Could not check share: STATUS_OBJECT_NAME_NOT_FOUND
```

```
[*] Testing share print$

[+] Mapping: DENIED, Listing: N/A

[*] Testing share secret_folder

[+] Mapping: DENIED, Listing: N/A

 ===========================================

|    Policies via RPC for 192.168.88.251    |

 ===========================================

[*] Trying port 445/tcp

[+] Found policy:

domain_password_information:

  pw_history_length: None

  min_pw_length: 5

  min_pw_age: none

  max_pw_age: 49710 days 6 hours 21 minutes

  pw_properties:

  - DOMAIN_PASSWORD_COMPLEX: false

  - DOMAIN_PASSWORD_NO_ANON_CHANGE: false

  - DOMAIN_PASSWORD_NO_CLEAR_CHANGE: false

  - DOMAIN_PASSWORD_LOCKOUT_ADMINS: false
```

```
  - DOMAIN_PASSWORD_PASSWORD_STORE_CLEARTEXT: false

  - DOMAIN_PASSWORD_REFUSE_PASSWORD_CHANGE: false

domain_lockout_information:

  lockout_observation_window: 30 minutes

  lockout_duration: 30 minutes

  lockout_threshold: None

domain_logoff_information:

  force_logoff_time: 49710 days 6 hours 21 minutes

 ============================================

|    Printers via RPC for 192.168.88.251     |

 ============================================

[+] No printers returned (this is not an error)

Completed after 0.70 seconds

|--[root@websploit]--[~/enum4linux-ng]

|--- #
```

Las líneas resaltadas en el ejemplo 3-30 muestran los usuarios enumerados, la versión de Samba y las carpetas compartidas. También puede utilizar herramientas simples como smbclient para enumerar los recursos compartidos y otra información de un sistema que ejecuta SMB, como se muestra en el ejemplo 3-31.

Ejemplo 3-31 - Enumeración con smbclient

```
|--[root@websploit]

|--- #smbclient -L \\\192.168.88.251

        Sharename       Type       Comment

        ---------       ----       -------

        print$                          Disk       Printer Drivers

        secret_folder                Disk       Extremely
sensitive information

        IPC$            IPC        IPC Service (Samba
4.9.5-Debian)

SMB1 disabled -- no workgroup available

|--[root@websploit]--[~/enum4linux-ng]

|--- #
```

Enumeración de páginas web / Enumeración de aplicaciones web

Una vez que haya identificado que un servidor web se está ejecutando en un host de destino, el siguiente paso es echar un vistazo a la aplicación web y comenzar a trazar el mapa de la superficie de ataque realizando *enumeración de páginas web* o, a menudo, denominado *Enumeración de aplicaciones web*. Puede trazar un mapa de la superficie de ataque de una aplicación web de diferentes maneras. La práctica herramienta Nmap tiene un script NSE disponible para forzar las rutas de directorios y archivos de las aplicaciones web. Armado con una lista de archivos y directorios

conocidos utilizados por aplicaciones web comunes, analiza el servidor para cada uno de los elementos de la lista. En función de la respuesta del servidor, puede determinar si existen esas rutas. Esto es útil para identificar elementos como la página del administrador predeterminada de Apache o Tomcat que comúnmente se dejan en los servidores web y pueden ser posibles rutas de acceso. La sintaxis de http-enum NSE es la siguiente:

```
nmap -sV --script=http-enum <target>
```

El ejemplo 3-32 muestra los resultados de ejecutar en el host con la dirección IP 192.168.88.251.

Ejemplo 3-32 - Ejemplo de salida de secuencia de comandos http-enum de Nmap

```
|--[root@websploit]--[~]

|--- #nmap -sV --script=http-enum -p 80 192.168.88.251

Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-22 11:53
EDT

Nmap scan report for 192.168.88.251

Host is up (0.0011s latency).



PORT   STATE SERVICE VERSION

80/tcp open  http    nginx 1.17.2

| http-enum:

|   /admin/: Possible admin folder

|   /admin/index.html: Possible admin folder
```

```
|_  /s/: Potentially interesting folder

|_http-server-header: nginx/1.17.2

Service detection performed. Please report any incorrect
results at

https://nmap.org/submit/ .

Nmap done: 1 IP address (1 host up) scanned in 8.54 seconds

|--[root@websploit]--[~]

|--- #
```

El resultado resaltado en el ejemplo 3-32 muestra varios directorios / carpetas enumerados y la versión del servidor web que se utiliza (Nginx 1.17.2). Este es un buen lugar para comenzar a atacar una aplicación web.

Otra herramienta de enumeración de servidores web de la que deberíamos hablar es Nikto. Nikto es un escáner de vulnerabilidades web de código abierto que ha existido durante muchos años. No es tan sólido como los escáneres de vulnerabilidades web comerciales; sin embargo, es muy útil para ejecutar un guion rápido para enumerar información sobre un servidor web y las aplicaciones que aloja. Debido a la velocidad a la que trabaja Nikto para escanear un servidor web, es muy ruidoso. Proporciona varias opciones de escaneo, incluida la capacidad de autenticarse en una aplicación web que requiere un nombre de usuario y una contraseña. El ejemplo 3-33 muestra el resultado de un escaneo de Nikto que se ejecuta en el mismo host que en el ejemplo 3-32 (192.168.88.251). El resultado del ejemplo 3-33 muestra resultados similares al guion de Nmap utilizado en el ejemplo 3-32.

Ejemplo 3-33 - Muestra de Nikto Scan

```
|--[root@websploit]--[~]

|--- #nikto -h 192.168.88.251

- Nikto v2.1.6

---------------------------------------------------------------------
---------

+ Target IP:           192.168.88.251

+ Target Hostname:    192.168.88.251

+ Target Port:        80

---------------------------------------------------------------------
---------

+ Server: nginx/1.17.2

+ The anti-clickjacking X-Frame-Options header is not present.

+ The anti-clickjacking X-Frame-Options header is not present.

+ The X-XSS-Protection header is not defined. This header can
hint to

the user agent to protect against some forms of XSS

+ The X-Content-Type-Options header is not set. This could
allow the

user agent to render the content of the site in a different
fashion
```

```
to the MIME type

+ No CGI Directories found (use '-C all' to force check all
possible

dirs)

+ OSVDB-3092: /admin/: This might be interesting...

+ /admin/index.html: Admin login page/section found.

+ /wp-admin/: Admin login page/section found.

+ /wp-login/: Admin login page/section found.

+ 7916 requests: 0 error(s) and 7 item(s) reported on remote
host

+ End Time:           2021-06-22 11:57:59 (GMT-4) (15
seconds)

---------------------------------------------------------------
-------------

+ 1 host(s) tested

|--[root@websploit]--[~]

|--- #
```

CONSEJO Ninguna herramienta es perfecta. Se recomienda que se familiarice con el comportamiento y los resultados de las diferentes herramientas. El Módulo 10 abarca

varias herramientas adicionales que se pueden utilizar para la enumeración y el reconocimiento.

Enumeración de servicios

La enumeración de servicios es el proceso de identificación de los servicios que se ejecutan en un sistema remoto y es un enfoque principal de lo que hace Nmap como escáner de puertos. La discusión anterior en este módulo destaca los diversos tipos de escaneo y cómo se pueden utilizar para omitir los filtros. Cuando está conectado a un sistema que está en un segmento de red conectado directamente, puede ejecutar algunos guiones adicionales para seguir enumerando. Un escaneo de puertos toma la perspectiva de un usuario remoto con credenciales. El guion de NSE Nmap smb-enum-processing enumera los servicios en un sistema Windows y lo hace mediante las credenciales de un usuario que tiene acceso para leer el estado de los servicios en ejecución. Esta es una herramienta útil para consultar de forma remota un sistema de Windows para determinar la lista exacta de servicios en ejecución. La sintaxis de los comandos es la siguiente:

```
nmap --script smb-enum-processes.nse --script-args
smbusername=<username>, smbpass=<password> -p445 <host>
```

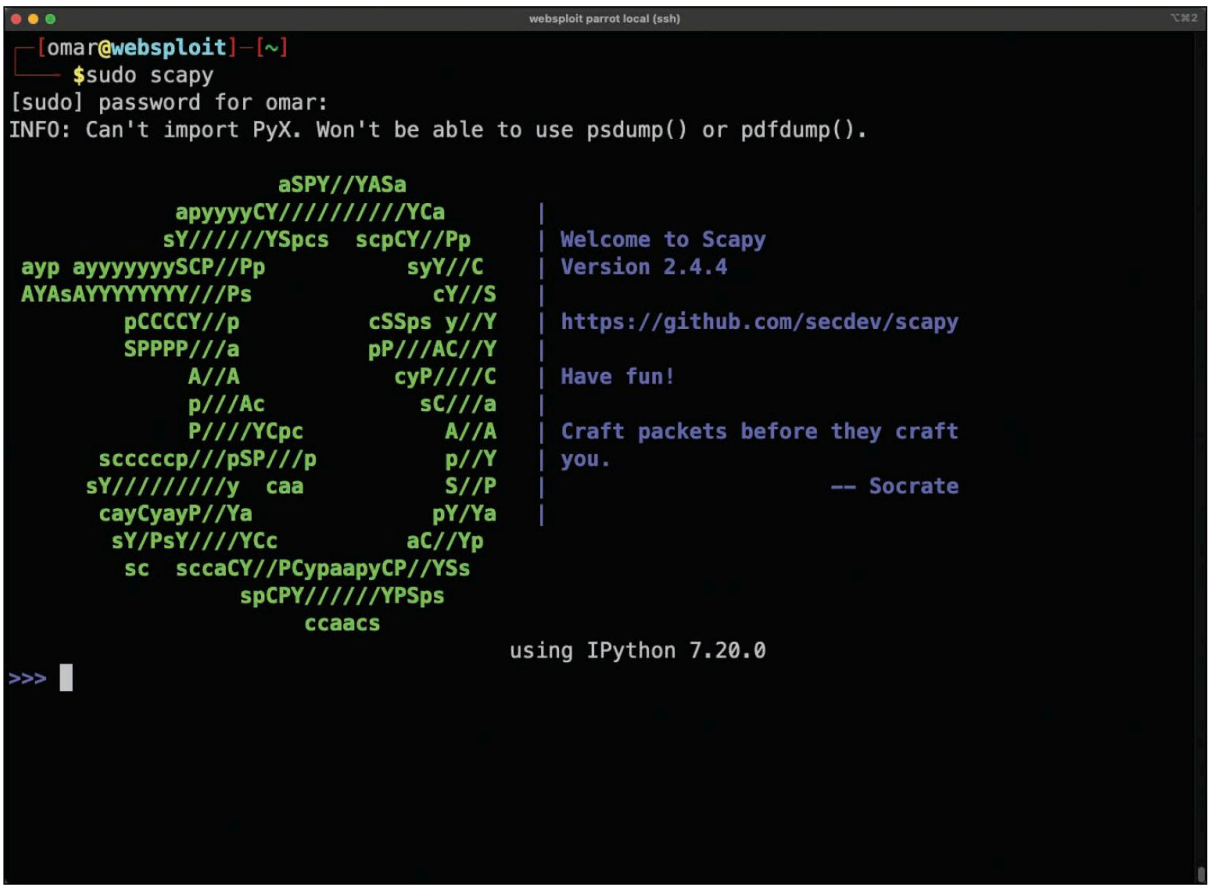
Exploración de la enumeración a través de la elaboración de paquetes

Cuando se trata de la enumeración a través de la elaboración y la generación de paquetes, Scapy es una de las herramientas y marcos favoritos de los evaluador de penetraciones. Scapy es un ecosistema o marco muy completo basado en Python para la generación de paquetes. En esta sección, se analizan algunas de las formas simples en que puede utilizar esta herramienta para realizar un reconocimiento básico de la red.

NOTA Scapy debe ejecutarse con permisos de root para poder modificar paquetes.

Iniciar el shell interactivo de Scapy es tan fácil como escribir sudo scapy desde una ventana de terminal, como se ilustra en la Figura 3-13.

Figura 3-13- Inicio de scapy desde la línea de comandos



El ejemplo 3-34 muestra lo fácil que es comenzar a crear paquetes. En este ejemplo, se crea un paquete ICMP simple con malicious_payload como carga útil que se envía al host de destino 192.168.88.251.

Ejemplo 3-34 - Creación de un paquete ICMP simple con Scapy

```
>>> send(IP(dst="192.168.88.251")/ICMP()/"malicious_payload")
.
Sent 1 packets.
```

El ejemplo 3-35 muestra el paquete ICMP recibido por el sistema de destino (192.168.88.225/vulnhost-1). La herramienta de captura de paquetes tshark se utiliza para capturar el paquete ICMP diseñado.

Ejemplo 3-35 - Recopilación de un paquete creado con tshark

```
omar@vulnhost-1 ~ % sudo tshark host 192.168.78.142

Capturing on 'eth0'

     1 0.000000000 192.168.78.142 ? 192.168.88.251 ICMP 60
Echo (ping) request  id=0x0000, seq=0/0, ttl=63

     2 0.000026929 192.168.88.251 ? 192.168.78.142 ICMP 59
Echo (ping) reply    id=0x0000, seq=0/0, ttl=64 (request in 1)
```

Scapy admite una gran cantidad de protocolos. Puede usar la función ls () para enumerar todos los formatos y protocolos disponibles, como se muestra en el ejemplo 3-36.

Ejemplo 3-36 - La función Scapy ls ()

```
>>> ls()

AH           : AH

AKMSuite     : AKM suite

ARP          : ARP

ASN1P_INTEGER : None

ASN1P_OID    : None
```

```
ASN1P_PRIVSEQ : None

ASN1_Packet : None

ATT_Error_Response : Error Response

ATT_Exchange_MTU_Request : Exchange MTU Request

ATT_Exchange_MTU_Response : Exchange MTU Response

ATT_Execute_Write_Request : Execute Write Request

ATT_Execute_Write_Response : Execute Write Response

ATT_Find_By_Type_Value_Request : Find By Type Value Request

ATT_Find_By_Type_Value_Response : Find By Type Value Response

ATT_Find_Information_Request : Find Information Request

ATT_Find_Information_Response : Find Information Response

ATT_Handle : ATT Short Handle

ATT_Handle_UUID128 : ATT Handle (UUID 128)

ATT_Handle_Value_Indication : Handle Value Indication

ATT_Handle_Value_Notification : Handle Value Notification

ATT_Handle_Variable : None

ATT_Hdr    : ATT header

ATT_Prepare_Write_Request : Prepare Write Request

<output omitted for brevity>
```

Puede usar la función ls () para mostrar todas las opciones y campos de un protocolo o formato de paquete específico soportado por Scapy, como se muestra en el ejemplo 3-37. Este ejemplo muestra los campos disponibles para el protocolo TCP.

Ejemplo 3-37 - Listado de los campos de la capa 4 de TCP en Scapy

```
>>> ls(TCP)
sport          : ShortEnumField                      = (20)
dport          : ShortEnumField                      = (80)
seq            : IntField                            = (0)
ack            : IntField                            = (0)
dataofs        : BitField  (4 bits)                  = (None)
reserved       : BitField  (3 bits)                  = (0)
flags          : FlagsField  (9 bits)                = (<Flag 2
(S)>)
window         : ShortField                          = (8192)
chksum         : XShortField                         = (None)
urgptr         : ShortField                          = (0)
options        : TCPOptionsField                     = (b'')
```

El ejemplo 3-38 muestra los campos del paquete DNS que Scapy puede modificar.

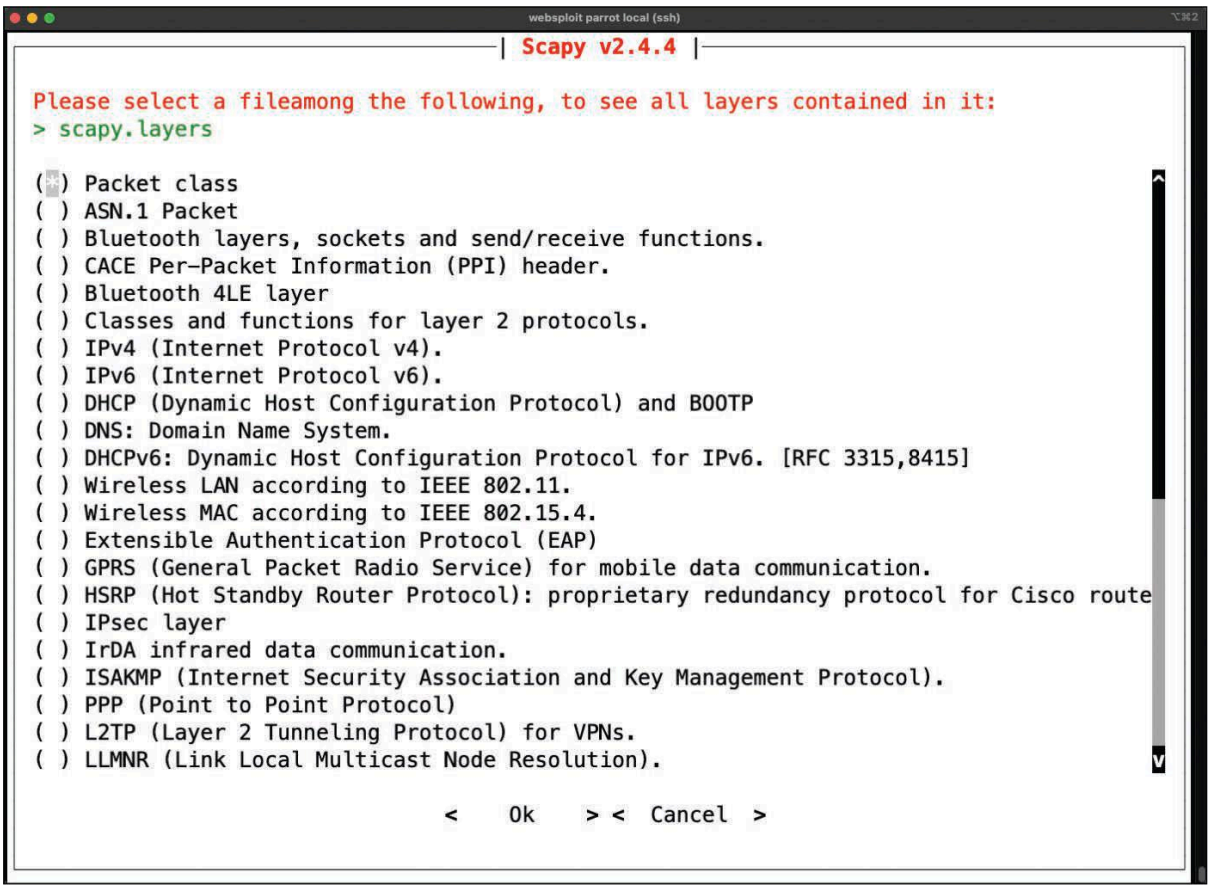
Ejemplo 3-38 - Lista de los campos de paquetes DNS disponibles en Scapy

```
>>> ls(DNS)
length        : ShortField (Cond)                   = (None)
id            : ShortField                          = (0)
qr            : BitField  (1 bit)                   = (0)
opcode        : BitEnumField  (4 bits)              = (0)
aa            : BitField  (1 bit)                   = (0)
tc            : BitField  (1 bit)                   = (0)
rd            : BitField  (1 bit)                   = (1)
ra            : BitField  (1 bit)                   = (0)
z             : BitField  (1 bit)                   = (0)
ad            : BitField  (1 bit)                   = (0)
cd            : BitField  (1 bit)                   = (0)
rcode         : BitEnumField  (4 bits)              = (0)
qdcount       : DNSRRCountField                     = (None)
ancount       : DNSRRCountField                     = (None)
nscount       : DNSRRCountField                     = (None)
```

```
arcount       : DNSRRCountField                   = (None)

qd            : DNSQRField                        = (None)

an            : DNSRRField                        = (None)

ns            : DNSRRField                        = (None)

ar            : DNSRRField                        = (None)
```

Puede utilizar la función explore () para desplazarse por las capas y los protocolos de Scapy. Después de ejecutar la función explore (), se mostrará la pantalla de la Figura 3-14.

Figura 3-14 - Uso de la función explore () en Scapy



Puede utilizar la función explore () con cualquier formato de paquete o protocolo. El ejemplo 3-39 muestra los paquetes contenidos en scapy.layers.dns con la función explore (“dns”).

Ejemplo 3-39 - Uso de la función explore (“dns”) para mostrar los tipos de paquetes en scapy. layers.dns

```
>>> explore("dns")

Packets contained in scapy.layers.dns:

Class                          |Name

-------------------------------|------------------------------

DNS                            |DNS

DNSQR                          |DNS Question Record
```

```
DNSRR                          |DNS Resource Record

DNSRRDLV                       |DNS DLV Resource Record

DNSRRDNSKEY                    |DNS DNSKEY Resource Record

DNSRRDS                        |DNS DS Resource Record

DNSRRMX                        |DNS MX Resource Record

DNSRRNSEC                      |DNS NSEC Resource Record

DNSRRNSEC3                     |DNS NSEC3 Resource Record

DNSRRNSEC3PARAM                |DNS NSEC3PARAM Resource Record

DNSRROPT                       |DNS OPT Resource Record

DNSRRRSIG                      |DNS RRSIG Resource Record

DNSRRSOA                       |DNS SOA Resource Record

DNSRRSRV                       |DNS SRV Resource Record

DNSRRTSIG                      |DNS TSIG Resource Record

EDNS0TLV                       |DNS EDNS0 TLV

InheritOriginDNSStrPacket|
```

Puede utilizar Scapy como escáner de muchas maneras diferentes. Omar Santos tiene varios ejemplos de guiones (scripts) de Python para realizar escaneo de redes y sistemas con Scapy en su repositorio de GitHub; consulte *https://github.com/The-Art-of-Hacking/h4cker/blob/master/python_ruby_and_bash*. Sin embargo, puede realizar un escaneo TCP SYN simple en cualquier puerto,

como se muestra en el ejemplo 3-40. En este ejemplo, se envía un paquete SYN del puerto TCP 445 al host con la dirección IP 192.168.88.251. El resultado indica que recibió una respuesta, pero no especifica cuál fue la respuesta real.

Ejemplo 3-40 - Envío de un paquete TCP SYN con Scapy

```
>>> ans, unans =
sr(IP(dst='192.168.88.251')/TCP(dport=445,flags='S'))

Begin emission:

Finished sending 1 packets.

....*

Received 5 packets, got 1 answers, remaining 0 packets

>>>
```

El ejemplo 3-41 muestra la captura de paquetes en el host de destino (192.168.88.251).

Ejemplo 3-41 - Captura de paquetes de TCP en el host de destino

```
omar@vulnhost-1 ~ % sudo tshark host 192.168.78.142

Running as user "root" and group "root". This could be
dangerous.

Capturing on 'eth0'
```

```
     1 0.000000000 192.168.78.142 ? 192.168.88.251 TCP 60 20 ?
445 [SYN] Seq=0 Win=8192 Len=0

     2 0.000033735 192.168.88.251 ? 192.168.78.142 TCP 58 445
? 20 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

     3 0.001065273 192.168.78.142 ? 192.168.88.251 TCP 60 20 ?
445 [RST] Seq=1 Win=0 Len=0
```

3.2.5 Práctica - Exploración de la enumeración a través de la elaboración de paquetes con Scapy

¿Qué herramienta puede utilizar un atacante para generar paquetes modificados?

Nmap

Scapy

ExifTool

Recon-ng

done

Enviar

Mostrar retroalimentación

3.2.6 Práctica de laboratorio - Enumeración con Nmap

Tarea de soluciones de seguridad de Protego

Nmap es una herramienta de reconocimiento activa clásica para evaluador de penetraciones. Todo evaluador de penetración debe estar familiarizado con él porque tiene una flexibilidad sin precedentes. Nmap es una herramienta de línea de comandos y Zenmap es una versión de GUI. Si bien Zenmap hace que el conocimiento de la amplia gama de opciones de Nmap sea menos crucial, Zenmap no puede reemplazar la fluidez en la línea de comandos de Nmap.

Pronto comenzaremos a realizar reconocimientos activos en los sistemas de nuestros clientes. Practique sus habilidades de Nmap en esta práctica de laboratorio para poder participar plenamente mientras nuestro equipo trabaja en la interacción de Pixel Paradise.

En esta práctica de laboratorio se cumplirán los siguientes objetivos:

- Investigar Nmap
- Realizar escaneos básicos de Nmap

descriptionPráctica de laboratorio - Enumeración con Nmap

Responda las siguientes preguntas después de completar la práctica de laboratorio.

Verificación de habilidades

Construya el comando nmap que creó el siguiente resultado seleccionando de los cuadros desplegables.

```
   Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-10 16:44
UTC
Nmap scan report for 10.6.6.11
Host is up (0.000098s latency).

PORT     STATE SERVICE VERSION
8888/tcp open  http    nginx 1.18.0

Service detection performed. Please report any incorrect
results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.39 seconds
```

done

Enviar

Mostrar comentarios

Encuesta de laboratorio

Cuéntenos su experiencia con el laboratorio indicando su nivel de acuerdo con las siguientes afirmaciones.

Me siento seguro acerca de las habilidades que practiqué en este laboratorio.

Totalmente de acuerdo

done

Realizar esta práctica de laboratorio fue una buena inversión de mi tiempo.

Totalmente de acuerdo

done

done

Enviar

Mostrar retroalimentación

3.2.7 Inspección y escucha de paquetes

Como evaluador de penetración, puede utilizar herramientas como Wireshark, tshark y tcpdump para recopilar capturas de paquetes para la inspección y la escucha. Cualquiera que haya estado involucrado con las redes o la seguridad ha utilizado estas herramientas en algún momento para capturar y analizar el tráfico en una red. Para un evaluador de penetración, estas herramientas pueden ser convenientes para realizar un reconocimiento pasivo. Por supuesto, este tipo de reconocimiento requiere una conexión física o inalámbrica con el objetivo. Si le preocupa que lo detecten, probablemente sea mejor que intente una conexión inalámbrica porque no requeriría estar dentro del edificio. Muchas veces, la señal inalámbrica de una empresa se filtra más allá de sus paredes físicas. Esto le da al

evaluador de penetración la oportunidad de recopilar información sobre el objetivo y posiblemente obtener acceso a la red para detectar el tráfico. El módulo 5, “Aprovechamiento de las redes cableadas e inalámbricas”, trata este tema en profundidad.

3.2.8 Práctica - Inspección de paquetes y escuchas

¿Qué herramientas se pueden utilizar para realizar un reconocimiento pasivo? (Escoja tres opciones.)

Nikto

Nmap

tshark

tcpdump

Wireshark

enum4linux

done

Enviar

Mostrar retroalimentación

3.2.9 Práctica de laboratorio - Elaboración de paquetes con Scapy

Tarea de soluciones de seguridad de Protego

Scapy le permite crear y enviar paquetes que tienen los valores que desea en los campos de datos del protocolo. Por ejemplo, puede crear paquetes que hayan manipulado direcciones IP de origen y destino, indicadores TCP y números de puerto, y muchos otros valores que especifique. Lo usamos todo el tiempo en Protego.

Realice esta práctica de laboratorio para aprender los conceptos básicos de Scapy. Luego, puede ayudarnos con nuestro reconocimiento activo de Pixel Paradise.

En esta práctica de laboratorio, usará Scapy, una herramienta de manipulación de paquetes basada en Python, para crear paquetes personalizados. Estos paquetes personalizados se utilizarán para realizar el reconocimiento en un sistema de destino.

- Parte 1: Investigar la herramienta Scapy.
- Parte 2: Usar Scapy para rastrear el tráfico de red.
- Parte 3: Crear y enviar un paquete ICMP.
- Parte 4: Crear y enviar paquetes TCP SYN.

description Práctica de laboratorio - Elaboración de paquetes con Scapy

Seleccione reproducir para ver una demostración de la práctica de laboratorio.

Responda las siguientes preguntas después de completar la práctica de laboratorio.

Verificación de habilidades

Desea enviar un paquete al puerto HTTP conocido en el host 192.168.99.17 con el indicador TCP SYN establecido. ¿Qué opciones usará con la función send() de Scapy para crear este paquete? (Escoja tres opciones.)

```
IP(dst="192.168.99.17")
```

```
TCP(flags="SN")
```

```
TCP(dport=80)
```

```
TCP(dport=8080)
```

```
TCP(flags="S")
```

done

Enviar

Mostrar retroalimentación

Encuesta de laboratorio

Cuéntenos su experiencia con el laboratorio indicando su nivel de acuerdo con las siguientes afirmaciones.

Me siento seguro acerca de las habilidades que practiqué en este laboratorio.

Totalmente de acuerdo

done

Realizar esta práctica de laboratorio fue una buena inversión de mi tiempo.

Totalmente de acuerdo

done

done

Enviar

Mostrar retroalimentación

3.2.10 Práctica de laboratorio - Rastreo de redes con Wireshark

Tarea de soluciones de seguridad de Protego

Wireshark y tcpdump son herramientas indispensables para capturar el tráfico en una red. Permiten la inspección detallada del tráfico de red, incluidas las contraseñas, los hash, los archivos y los intercambios de paquetes con subprocesos completos. Wireshark usa una GUI para capturas de paquetes, mientras que tcpdump es una herramienta de línea de comandos. Realmente necesita conocerlos a ambos para trabajar en Protego.

Familiarícese con tcpdump y Wireshark en la práctica de laboratorio. Practique sus habilidades para poder ayudarnos con la participación de Pixel Paradise.

En esta práctica de laboratorio, utilizará la utilidad de Linux tcpdump para capturar y guardar el tráfico de red. Luego utilizará Wireshark para investigar la captura de tráfico.

- Prepare el host para capturar el tráfico de red.
- Capture y guarde el tráfico de red.
- Ver y analizar la captura de paquetes.

description Práctica de laboratorio - Rastreo de redes con Wireshark

Demostración en video

Seleccione reproducir para ver una demostración de la práctica de laboratorio.

Responda las siguientes preguntas después de completar la práctica de laboratorio.

Verificación de habilidades

¿Qué hará esta declaración de tcpdump?

```
sudo tcpdump -i eth0 -s 0 -w packetdump.pcap
```

Con

raíz

done

privilegios, podrá

capturar

262144

done

bytes de datos y guardar un archivo

llamado packetdump.pcap al

directorio de trabajo actual

done

.

done

Enviar

Mostrar retroalimentación

Encuesta de laboratorio

Cuéntenos su experiencia con el laboratorio indicando su nivel de acuerdo con las siguientes afirmaciones.

Me siento seguro acerca de las habilidades que practiqué en este laboratorio.

Totalmente de acuerdo

done

Realizar esta práctica de laboratorio fue una buena inversión de mi tiempo.

Totalmente de acuerdo

done

done

Enviar

Mostrar retroalimentación