LABORATORIOS

Cliente - Laboratorio Windows 10:

https://acortar.link/DIvXL

Metasploitable 3-Laboratorio Windows Server 2008:

https://acortar.link/Ug155

Metasploitable 2 – Laboratorio Linux:

https://acortar.link/0H7gx

Descargar más laboratorios:

https://www.hackthebox.eu/
https://www.vulnhub.com/
https://lab.pentestit.ru/

Instalar Metasploitable 2

Vamos a nueva y ponemos el nombre que queramos, pero en tipo ponemos Linux y en versión Ubuntu (64).

En la siguiente ventana nos pide que le pongamos el tamaño de la RAM, podemos dejar solo 1 GB y le damos a next. En Disco duro marcamos el que dice “Usar un archivo de disco duro virtual existente” y buscamos metasploitable2, si no está en la lista pues hay que darle a Añadir y ya lo buscamos entre los ficheros del laboratorio. Una vez que lo tengamos le damos a crear. Ahora hay que configurar las redes y ponerla como solo anfitrión, esto permitirá que los laboratorios estén aislados. Para configurar la red marcamos la máquina virtual → Configuración → Red → Adaptador sólo-anfitrión.

netdiscover

La sintaxis de netdiscover es:

sudo netdiscover -i [interfaz] -r [Rango]

Ejemplo:

sudo netdiscover -i eth0 -r 192.168.56.0/24

sudo → Se utiliza para dar permisos de administrador.

-i → Para indicar la interfaz, en mi caso eth0, también puede ser wlan0 si estás desde wifi.

-r → Aquí ponemos el rango, Ejemplo: 192.168.56.0/24, /16, /8.*

Ahora hay que poner a correr Wireshark para ver cómo detecta nuestro ataque, para ello solo hay que seleccionar la interfaz que vamos a usar, le damos y empezará ya a escanear. Vamos a ejecutar el comando anterior en la terminal. Tarda un poco en dar resultados, solo hay que esperar.

El comando para iniciar el método pasivo es: **sudo netdiscover -i eth0 -p**

Recopilación de información gracias a servicios

Gracias a ciertos servicios también podemos descubrir la existencia de las máquinas. Una de las herramientas que vamos a utilizar para esto es nbtscan.

Sintaxis: **sudo nbtscan [RANGO]**

Ejemplo: **sudo nbtscan 192.168.56.0/24**

Como puedes ver, ha hecho un barrido NetBIOS y me muestra aquellas máquinas que tengan un servicio NetBIOS funcionando. Esto es peligroso ya que los sistemas operativos de Windows tienen el servicio NetBIOS funcionando por defecto. Este protocolo es increíble, pero la contra es que estamos mandando paquetes y pueden descubrirnos. Conceptos de Análisis de Puertos y Vulnerabilidades Un análisis de puertos lo único que hace es intentar establecer o finalizar una conexión con un puerto determinado. Gracias a los tipos de respuestas que vamos a obtener de la máquina remota, estas

herramientas pueden evaluar si el puerto está abierto, cerrado, filtrado… Además del protocolo TCP, también tenemos que tener en cuenta el UDP. Esto se debe a que no es lo mismo evaluar una empresa que se dedica a hacer muchos tipos de servicios de telemarketing, que seguramente tenga servicios de VOIP habilitados. Si por ejemplo hiciéramos una auditoría de caja blanca sabríamos qué protocolo van a tener los servicios, en cambio si no lo sabemos habría que hacer un escaneo básico del protocolo UDP, no de todos los puertos, pero sí de los más importantes.

Lo que nos permite detectar un escaneo de puertos es:

- Detectar sistemas encendidos o procesos que se están ejecutando en la red. (Lo que hemos hecho con NetBIOS)
- Descubrir qué programas o qué aplicaciones están funcionando en dichos puertos.
- Determinar el sistema operativo.
- Descubrir más direcciones IP.
- Identificar Banners. TCP Connect El tipo de análisis de puertos más viejo y a su vez más seguro es el TCP Connect. TCP Connect se basa en las 3 banderas(estados) de conexión en redes.

    - La primera bandera sería la SYN, que es cuando una máquina le comunica al servidor que si quiere conectar a un puerto.
    - La segunda bandera, en caso de que estuviera abierto el puerto, sería la respuesta, que es el ACK CONNECT, que dice si tienes permiso o no. La tercera bandera sería una vez que nos han dado el permiso, establecer la conexión contra dicho puerto. Si lo piensas, que me esté respondiendo un SYN-ACT significa que el puerto está abierto así que el último paso no sería necesario.

En eso es en lo que se basa el siguiente punto. TCP SYN TCP SYN se asemeja al escaneo de puertos TCP Connect, sin embargo, al comprobar que recibe respuesta, en vez de establecer la conexión completamente con un ACK Connect aborta dicha conexión haciendo que sortee algunas medidas de seguridad simplemente por el hecho de que ha obviado el último paso. TCP Null y TCP FIN Mientras que TCP SYN y TCP Connect se pueden usar en sistemas operativos Windows y Linux, TCP Null sólo se puede usar con Linux o Unix que cumplan el estándar de comunicaciones RFC(Request For Comments, Peticiones de comentarios). Como indica su nombre(nulo, vacío) enviará una petición, pero completamente vacía.

En el caso de que no se obtuviese ninguna respuesta por parte del objetivo, significa que está abierto o filtrado por un firewall. En el caso de que devolviese un error a nuestro escáner de vulnerabilidades, significa que está cerrado. TCP Fin, al igual que pasaba con TCP Null, es para uso en S.O Linux o Unix. Imitando el funcionamiento de TCP Null, que enviaba una llamada “vacía”, TCP Fin realiza dicha llamada simplemente poniendo el valor de “Fin de conexión”, por lo que si el objetivo no responde dicho puerto estaría abierto, en cambio si se registrase error estaría filtrado o cerrado. TCP FIN, radica en el hecho de que si nosotros mandamos una bandera FIN y nos dice que ha recibido el FIN de conexión significa que ese puerto pudiese estar abierto, en cambio si nos manda un error es que no está abierto. Con el TCP Null pasa lo mismo solo que en vez de mandar una bandera Fin manda una bandera Null, es decir, solo ceros. Si la respuesta fuera similar, se podría decir que está abierto. TCP XMAS y TCP ACK TCP XMAS es una técnica de exploración de puertos parecida al FIN Scan, ya que también se obtiene como resultado un paquete de Reset(RST) si el puerto está cerrado. Para el caso de este tipo de exploración de puertos, se envían paquetes o solicitudes del tipo FIN, URG y PUSH al host que se está explorando. No es recomendable usar este tipo de exploración de puertos con Sistemas Microsoft ya que la información que devolverá será un poco confusa y poco válida. XMAS Scan está pensado para trabajar únicamente con sistemas operacionales que tengan implementaciones de TCP/IP con respecto al documento RFC793.

Este tipo de exploración es recomendable llevarlo a la práctica en sistemas de tipo UNIX, LINUX Y *.BSD. TCP ACK, al contrario que los anteriores análisis de puertos que te listan los puertos abiertos, lo que hace es buscar los puertos que están siendo filtrados por alguna clase de firewall. Al enviar una petición ACK Active, ésta obtiene una respuesta de resetear la conexión (RST). En el caso de que no haya respuesta es que está siendo filtrado por un firewall.

Otra explicación que quizás te resulte más sencilla es que de TCP XMAS se podría decir que es parecido a un análisis FIN, la diferencia es que no está orientado al nivel de capa de redes. El paquete PUSH está implementado para pasarlo en la petición directamente a la capa de aplicación, como podría ser una aplicación Web, por lo tanto aquellos servicios que están trabajando a nivel de red no van a dar una respuesta correcta ya que esto está orientado para la capa de aplicación. Va a mandar un paquete FIN, URG(Urgente) y un paquete PUSH, que lo que hace es mandarlo a la pila de protocolos del nivel de aplicación. Aquí lo que estamos intentando hacer es aprovecharse de fallos de

diseño que pueda tener el servicio. El tipo de análisis de puertos ARK no está diseñado para saber que puestos están abiertos o cerrados, solo buscar que puertos están filtrados tras una puerta de enlace, cortafuegos, etc.. UDP El protocolo UDP también cuenta con sus métodos para poder analizar los puertos que están en uso, y por lo tanto abiertos. El análisis consiste en el simple envío de una cabecera sin datos. Dependiendo del error que reciba dicha petición se listará como abierto y/o filtrado, o cerrado.

El protocolo UDP no te va a responder, dependiendo del error que te devuelve va a determinar si existe un puerto abierto o no. Es muy sencillo, porque el protocolo UDP no tiene respuestas del estado de conexión como el TCP. El protocolo UDP no premia que el paquete se haya podido corromper y es más costoso en cuanto a tiempo determinar si está abierto o no. Un análisis de puertos UDP se suele lanzar cuando sabes que la empresa está utilizando algún tipo de servicios que puede estar funcionando sobre este tipo de protocolos. (Servicios de telefonía VOIP, Streaming, Transferencia de ficheros, servicios DNS..) Conceptos de análisis de vulnerabilidades Es la segunda parte de la fase de escaneo y tiene como objetivo el identificar si un sistema es débil o susceptible a ser afectado o atacado de alguna manera (Hardware, Software, Telecomunicaciones, Humanos…) Lo que hay que comprobar es:

Identificación de vulnerabilidades en Versiones de Aplicación y Sistemas operativos.

Gestión de parches (Patch Management) Identificar Vulnerabilidades Tecnológicas y Humanas. Configuraciones por defecto. Vulnerabilidades Técnicas y Funcionales A raíz del tipo de puerto y del servicio que esté funcionando se hará una batería de pruebas relacionadas a ese tipo de aplicación.

Esa batería de pruebas lo que va a hacer es identificar, a raíz de la versión, si está en la base de datos catalogado con algún tipo de exploit público que se haya reportado, se va a gestionar los parches que se tendrían que instalar en la aplicación, se va a hacer una batería de pruebas de configuración para ver si hay fallos de tecnología, configuración humana o configuración de por defecto, y además, se va a hacer una prueba de baterías tanto técnicas como funcionales como por ejemplo, ver si es vulnerable con XSS, SQL Injection. Se hará una petición legítima sin intención de explotar una vulnerabilidad y luego a partir de ahí, viendo la respuesta, se empezarán a hacer las peticiones maliciosas.

Clases de vulnerabilidades:

- Configuraciones de usuario o vendedor de software, que a menudo vienen de forma predeterminada.
- Aplicación. Errores de codificación que resulta en desbordamientos de buffer, inyecciones SQL, XSS, etc…
- Diseño. Fallos en el protocolo o arquitectura de aplicaciones.
- Host. Sistemas operativos y servicios.
- Dispositivos como Routers, Switches, Balanceadores de Carga, Firewalls, etc..
- Aplicaciones Cliente/Servidor, Bases de datos, etc.
- Humanos. Administradores de redes, desarrolladores, empleados, etc.. Aspectos Importantes
- Las herramientas de análisis de vulnerabilidades se basan en Plugins, por lo que es importante tenerlas siempre actualizadas. Yo me hice un Script que una vez al mes busca actualizaciones automáticamente.
- Configurar de forma adecuada el perfil del análisis de vulnerabilidades según la información recolectada en las fases pasadas. Algunas herramientas como Nessus necesitan configurarse al milímetro ya que tiene tantos Plugins que es una pérdida de tiempo tener encendidos todos los Plugins de Linux si estoy analizando una máquina basada en Windows.

Clasificación de las vulnerabilidades Dependiendo del software a utilizar puede variar, pero nosotros las definiremos para el informe de auditoría en:

- Bajas
- Medias
- Altas
- Críticas

Bajas. Son vulnerabilidades relacionadas con aspectos de la configuración de un sistema, rutas, etc. La cual probablemente podría ser utilizada para violar la seguridad del sistema, sin embargo no constituyen por si una vulnerabilidad ya que para ser explotada requiere de un conjunto de criterios que no necesariamente sería conseguido de forma directa por un atacante. Para la parte de recolección en una red, este tipo de vulnerabilidad es informativa, pero nos puede dar datos interesantes para la posterior explotación. Se suele abandonar la lectura de estas vulnerabilidades y siempre nos solemos ir a las

medias, altas y críticas. Esto es un grave error ya que a veces nos dan información muy relevante y pueden determinar que tipo de ataque llevar a cabo.

Medias. Este tipo de vulnerabilidades no son solo el objetivo final en ningún ataque, sino que dan pie o base a otras vulnerabilidades más críticas que comprometen el sistema. Un ejemplo es el uso de escalar privilegios.

Altas. Son un tipo de vulnerabilidades que pueden ser usadas para obtener acceso a recursos que deberían estar protegidos en el host remoto. Estas vulnerabilidades como tal comprometen el sistema afectado, ya que si son explotadas por un atacante este conseguirá el control parcial o total del sistema, además de que podrá ver y cambiar información confidencial, y ejecutar comandos y programas en el equipo afectado. Algunos fallos de protección pueden ser debidos a fallos en el diseño de un servicio ya que usan una versión obsoleta y te permite reproducir una vulnerabilidad que ya se ha listado de forma pública, también pueden ser fallos de configuración ya que se han dejado datos de acceso en recursos compartidos.

Críticas. Son similares a las vulnerabilidades de tipo alta, sin embargo las críticas suelen ser más peligrosas y requieren de la evaluación y corrección por parte de los administradores de informática de forma inmediata. Un ejemplo es poder acceder a la raíz de una máquina. Estas vulnerabilidades son tan serias que en cuanto se redactan en el informe, es lo primero que hay que intentar reproducir y si se da el caso de que realmente es una vulnerabilidad hay que avisar lo antes posible a la empresa que le estas haciendo la auditoría.

Nmap

Cuando hablamos de análisis de puertos una de las herramientas más importantes es Nmap. **Nmap** ya viene incluida en Kali Linux y se utiliza por la terminal.

Parámetros relacionados con el tipo de escaneo

Aquí vas a ver la relación de nmap con la teoría anterior. Los parámetros relacionados con el tipo de escaneo son:

- -sT → Realiza un escaneo de puertos mediante el método TCP Connect.
- -sS → Realiza un escaneo de puertos mediante el método TCP SYN.
- -sN→ Realiza un escaneo de puertos mediante el método TCP Null.
- -sF→ Realiza un escaneo de puertos mediante el método TCP FIN.
- -sA→ Realiza un escaneo de puertos mediante el método TCP ACK.
- -sU→ Realiza un escaneo de puertos mediante el método UDP.
- -p→ Indica a nmap que utilice el rango de puertos que le indiquemos, por ejemplo -p 1-30000 hará que compruebe ese rango de puertos. Por lo general, nmap suele escanear los 10.000 primeros puertos a no ser que le indiquemos lo contrario. Yo, personalmente, recomiendo indicarle el rango ya que existe la seguridad por oscuridad. La seguridad por oscuridad lo que intenta es ocultar servicios por lo que tendríamos que cambiar el rango predeterminado de puertos. El rango de puertos está entre 0 y 65535.

El 90% de las ocasiones vamos a trabajar con TCP SYN, ya que es uno de los más efectivos a no ser que haya alguna medida perimetral que esté bloqueando el análisis de puertos, es muy extraño tener que recurrir a otros tipos de análisis de puertos.

Parámetros de obtención de información del servicio en funcionamiento • -sV → Con este comando, nmap nos permite poder descubrir que tipo de servicio está funcionando en la máquina. (s→ Scanner; V → Version). • --version-all → Este parámetro debe de ir junto a -sV ya que se asegura de comprobar todos los servicios que conoce nmap para comprobar que servicio y versión ofrece el puerto abierto.

Parámetros para obtención de información del sistema operativo:

- -O → Con este parámetro busca que sistema operativo está utilizando nuestro objetivo.

- --osscan-guess: Si quieres un análisis más completo del S.O entonces debes incluir este parámetro junto a -O. Parámetros de evasión de detección

• -f → Fragmenta la petición en el tamaño que le indiquemos nosotros con el --mtu. Esto hace que la detección de nuestro análisis de vulnerabilidades sea mucho más difícil.
• --mtu: Funciona con el parámetro -f. Aquí indicaremos en qué tamaño queremos fragmentar la petición, el tamaño debe ser en fragmentos de 8 bits. Por ejemplo --mtu=32 lo fragmenta en trozos de 32 bits.
• -Pn → En ciertas ocasiones ya vamos a tener asegurado que la máquina esté encendida con el truco que vimos anteriormente de ARP. Este parámetro evita hacer “Ping” al objetivo para chequear si está o no “online”, esto es interesante ya que el “Ping” puede darnos problemas para analizar a nuestro objetivo porque algunas máquinas lo bloquean por que lo traducen en actividad sospechosa.
• -n → En ciertas ocasiones también tenemos el nombre del host, por lo tanto no hace falta realizar una resolución DNS directa/reversa.
• -D [direcciónIP] → Este parámetro nos permite suplantar una IP, Esto lo hace malformando el paquete TCP que está enviando para el análisis de puertos y en vez de venir la IP de origen de la máquina auditora, va a venir la IP que le hayamos indicado nosotros.
• -Tx → Establece el tiempo de espera entre conexiones a cada uno de los puertos. Va del 0 al 5 y cuanto más alto más rápido.

Parámetros para añadir más información:

• -V → El parámetro Verbose, indica a la aplicación que nos devuelva más información de la que presenta normalmente, cuantos más verboses se ponga, más detallado será el informe de nmap, por ejemplo -vvvv.
• -d → Este parámetro es el de depuración, hace que nmap nos muestre en pantalla las peticiones que realiza para ejecutar su análisis. Al igual que verbose, cuantos más pongamos más información nos dará. Ejemplo -dddd.
• -oX → Guardará el resultado de nuestro análisis en XML. Este parámetro se usaría así: -oX [url]. Ejemplo: -oX /home/jotta/Escritorio

Práctica

Ahora vamos a hacer uso de nmap, pero antes vamos a hacerlo todo de 0 como si de verdad estuviéramos en una auditoría.

Los pasos son:

1. Ver si hay alguna máquina encendida con el comando:

sudo netdiscover -i eth0 -r 192.168.56.0/24

Como vemos tenemos la de Metasploitable.

2. Empezar el análisis con **nmap**. Vamos a intentar hacerlo lo más sencillo posible.

Vamos a hacer un análisis de puertos mediante el método TCP SYN, así que en el comando tendrá que ir como parámetro -sS, además, como ya tenemos la IP de la máquina y sabemos que está encendida no nos hace falta hacer un “Ping” así que usaremos el parámetro -Pn y como tenemos el nombre del host, tampoco nos hace falta hacer una resolución DNS por eso también pondremos el comando -n.

El comando que vamos a usar quedaría así:

sudo nmap -sS -Pn -n 192.168.56.102

Para ver cómo se comporta también vamos a ejecutar Wireshark. Solo tienes que seleccionar la interfaz que vas a usar y darle al icono de la aleta de tiburón de color azul, en mi caso la interfaz es eth0. Una vez hecho esto ejecutamos el comando de nmap en la terminal.

Este es el resultado, como puedes ver no ha tardado nada y nos ha mostrado todos los puertos abiertos. ¿Quieres ver como se comporta?

Aquí he seleccionado un paquete cualquiera, para hacer esto es tan sencillo como hacerle clic derecho a alguno de la lista → seguir → Flujo TCP. Lo que podemos leer en esta captura es que se ha intentado hacer conexión mediante TCP SYN y nos la ha rechazado con un RST (reset), lo que significa que ese puerto no está abierto. Si por ejemplo cogemos uno que sí ha establecido conexión como el 445 podemos ver lo siguiente. Aquí ha intentado establecer conexión, la ha establecido y después ha cerrado la conexión. Este puerto estaría abierto. Este método es un poco agresivo porque intenta establecer conexión con el puerto como unas 10 veces, si queremos que solo lo intente una vez es tan sencillo como poner el comando -max-retries=1

sudo nmap -sS -Pn -n -max-retries=1 192.168.56.102

Ahora, se puede dar el caso de que la máquina que estoy analizando tenga seguridad por oscuridad, para comprobarlo ponemos el parámetro -p-

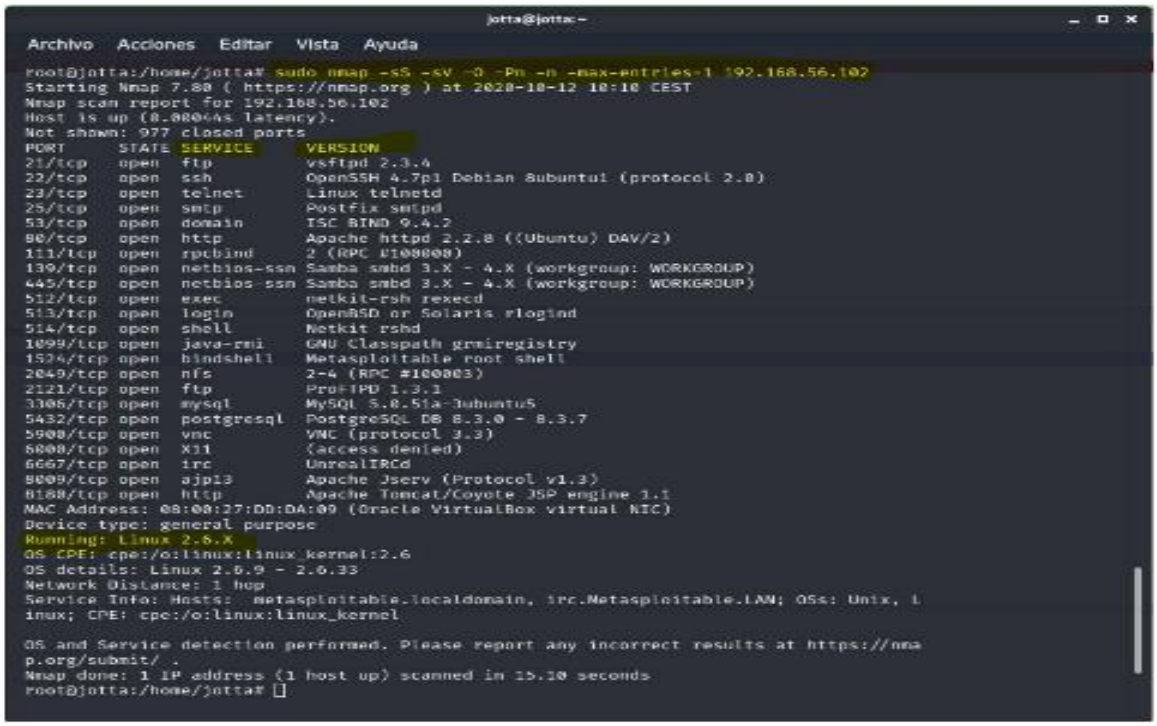
sudo nmap -sS -Pn -n -p- -max-retries=1 192.168.56.102

En la primera captura podemos ver que en teoría hay 23 puertos los que hay abiertos y en la segunda hemos encontrado más de 23 puertos, puedes comprobarlo comparando las listas. Estos podrían estar trabajando de forma oculta porque tienen fallos de configuración, son servicios críticos…

Ahora, como ya hemos hecho el análisis de puertos, también vamos a hacer el de servicios añadiendo -sV y también un análisis del tipo de S.O que está funcionando con el parámetro -O.

sudo nmap -sS -sV -O -Pn -p- -max-retries=1 192.168.56.102

Este procedimiento va a tardar un poco más ya que nmap está haciendo una batería muy básica de peticiones para determinar qué servicio está funcionando.



Como puedes ver, nos dice el servicio, la versión y el sistema operativo. Algunos servicios como el ftp que está en el puerto 21 funciona en la capa de

redes, pero otro como el 80(http) funciona en la de aplicación. Es sencillo ver que tipo de función está corriendo en un puerto determinado, para eso usamos la herramienta netcat

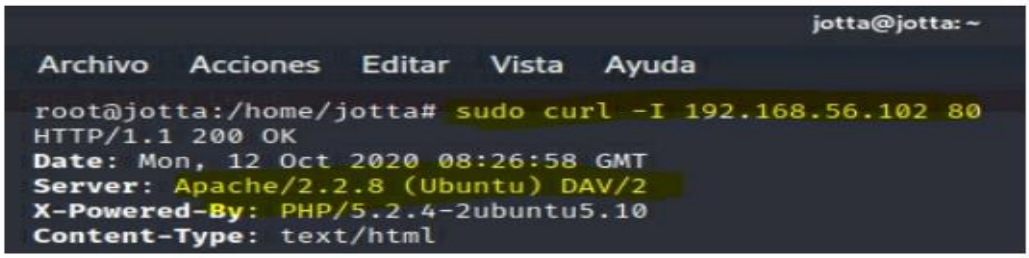
sudo nc 192.168.56.102 21

Si te fijas, en el momento en el que me he conectado ya me dice la versión con la que está trabajando.

Este comando para servicios que trabajan en la capa de aplicación ya no funciona. Para estos casos utilizo curl, un navegador por consola, acompañado del parámetro I.

El parámetro -I hace una petición head, esta petición lo que hace es preguntarle a la aplicación que servicio está corriendo.

```
sudo curl -I 192.168.56.102 80
```



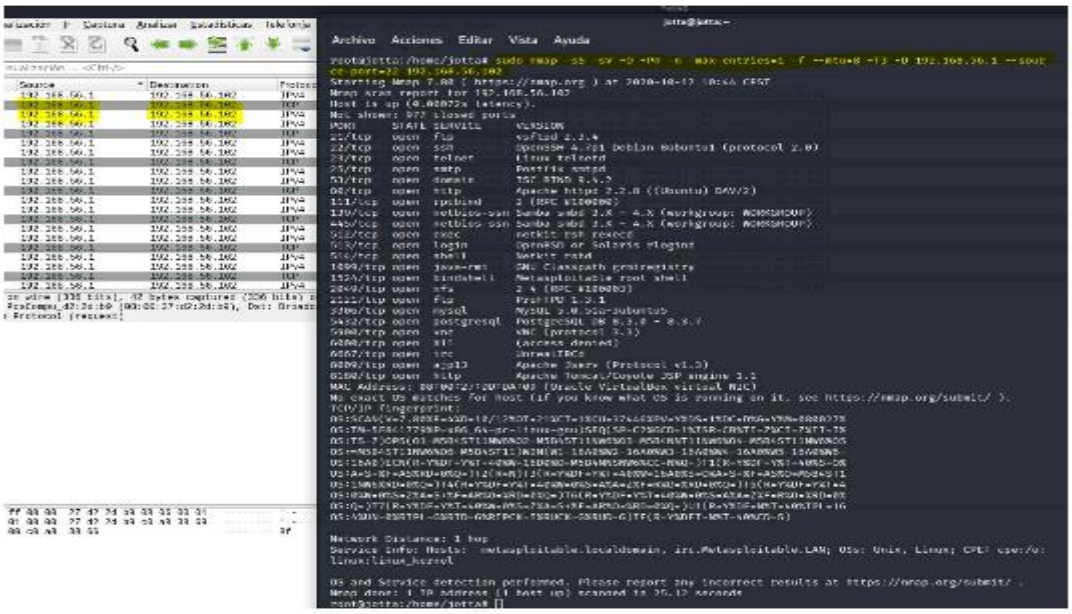
Aquí no solo me está diciendo que servicio está corriendo y su versión sino también el sistema operativo.

Ahora, vamos a añadir más parámetros a nuestro comando de **nmap** pero, **parámetros de evasión**.

Vamos a decirle que vamos a **fraccionarlo** en fragmentos de 8 bits con el parámetro -f --mtu=8, le vamos a poner un **Timer** de 3 con -T3, vamos a **suplantar la IP** de la puerta de enlace con -D 192.168.56.1 y vamos a decir que el origen de las conexiones van a ser **por el puerto** 22 con --source-port=22.

```
sudo nmap -sS -sV -O -Pn -max-entries=1 -f --mtu=8 -T3 -D 192.168.56.1 --source-port=22 192.168.56.102
```

Como puedes ver ya ha crecido un poco más el comando.



Y como resultado vemos que ha estado suplantando la IP que hemos puesto. Son los mismos resultados con la diferencia de que hemos intentado evadir la seguridad para que no nos pillaran.

Un punto que no he comentado antes son los scripts de **nmap**. Estos scripts no pueden evadir medidas IDS/IPS sino que están pensados para agilizar el proceso de una auditoría tras haber descubierto dichos puertos abiertos y sus respectivos servicios funcionando.

Algunos de estos scripts son:

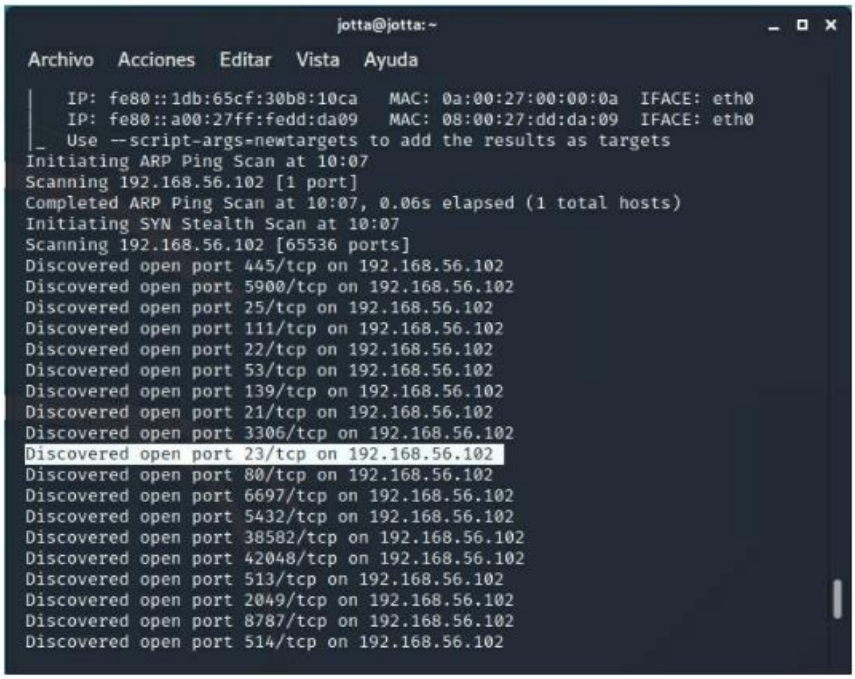
- auth. Se utiliza para identificar los métodos de autenticación en el objetivo.
- default. Ejecuta los scripts por defecto.
- discovery. Se utiliza para describir objetivos.
- intrusive. Ejecuta scritps agresivos que pueden afectar al objetivo (no lo recomiendo).
- safe. Ejecuta scripts “seguros” que no son muy agresivos.

- vuln. Realiza un análisis de vulnerabilidades contra exploits conocidos.

Antes de ejecutar el comando quiero que veas una cosa, por eso como hemos hecho en puntos anteriores vamos a ejecutar Wireshark. Una vez que ya está corriendo vamos a ejecutar el siguiente comando, esto suele tardar unos 10 min. Acuérdate de sustituir la IP que yo tengo puesta por la de tu víctima.

sudo nmap -sS -sV -p 0-65535 -T4 -O -v -n -Pn --scriptauth,discovery,exploit,vuln 192.168.56.102

Como puedes ver, nos ha salido este puerto.

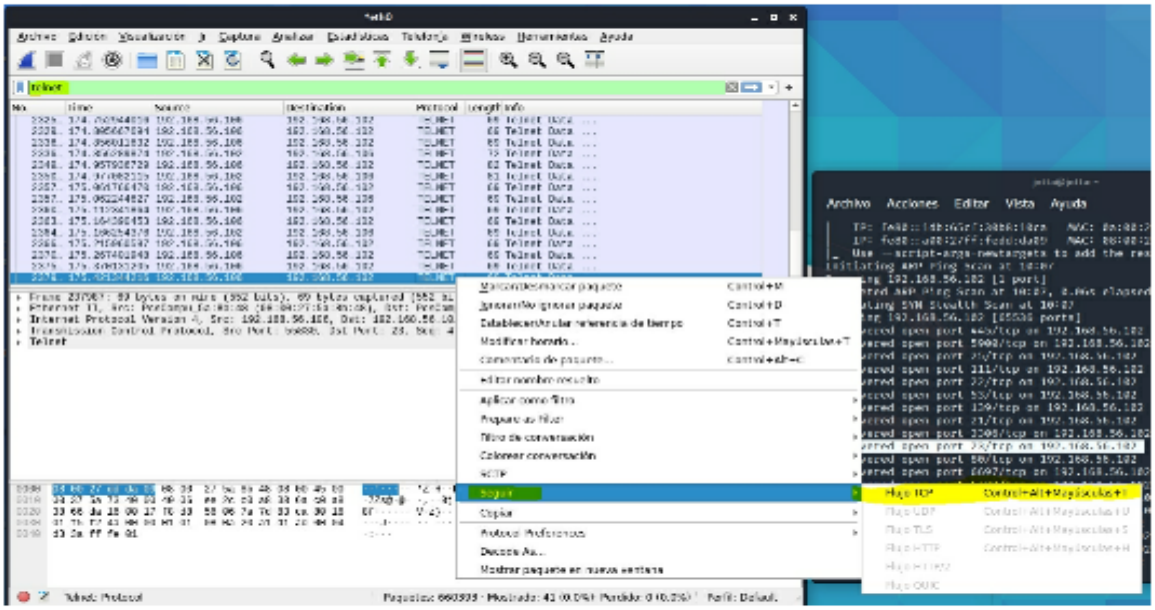


Esta es la primera vulnerabilidad que aparece en el servicio. El puerto 23 hace referencia a Telnet. Con Telnet tu puedes ver el tráfico que está corriendo en esa máquina, ya sean credenciales, comandos, etc.

Para comprobarlo vamos a Wireshark y buscamos Telnet.

Yo he parado el sniffeo para que no siga buscando.

Buscamos Telnet en Wireshark → Vamos al último paquete → Clic derecho → Seguir → Flujo TCP



Y se nos abrirá una ventana, esperamos a que cargue y nos mostrará los resultados.

En este caso lo que se puede hacer es decirle al cliente que migre ese servicio a uno seguro.

Si seguimos analizando la lista se puede ver el puerto 25. El puerto 25 utiliza un servicio smtp, este servicio se encarga de realizar el proceso de envío de correos electrónicos.

En el **puerto 53** llegamos a los servicios dns, estos servicios son interesantes ya que pueden contener subdominios que están utilizándose en la máquina.

En el **puerto 80** llegamos al servicio de aplicación web, se puede ver que este canal no es seguro ya que utiliza el protocolo http. Esto permite que si tienen alguna página de login, el atacante pueda ver todas las credenciales solo con sniffar el tráfico.

En el **puerto 111** está corriendo el servicio rpcbind, este servicio se encarga de informar que servicios de protocolo de comandos remotos están corriendo en la máquina.

En el **puerto 139 y 445** está corriendo el servicio Samba.

En el **puerto 1524** tenemos un bindshell, esto es muy peligroso ya que con poner un simple comando nos podemos conectar a la máquina.

sudo nc 192.168.56.102 1524

Ya estaría dentro de la máquina Metasploitable con el usuario root. Para saber más sobre los puertos te recomiendo echarle un ojo a esta web, no hace falta que los sepas todos, puedes tenerla como referencia

https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers