[ÍNDICE 1](#_Toc2081862)

[1. Acceso a la documentación de proyecto- Esta el el HGF 22](#_Vmware_workstation_player)

[2. Usuarios de validación de QC 22](#_Herramienta_de_kali)

[3. [Metodologia] Reporte incidencias a OSS/BSS 24](#_Configuración_entorno_de)

## Vmware workstation player

Se descarga la versión de Windows y siguiente, siguiente, siguiente

Ahora descargamos la maquina virtual de kali Linux, que va a ser la maquina virtual del atacante, la cual importamos en vmware

Usuario:kali

Passwd:kali

## Herramienta de kali para documentar

Cherrytree

Esta en cherrytree site:github.com

Vamos para la descarga a <https://www.giuspen.net/cherrytree/>

sudo apt install cherrytree

sudo apt update

sudo apt install cherrytree

## Configuración entorno de trabajo

echo $$ --- obtenemos el pid del proceso actual

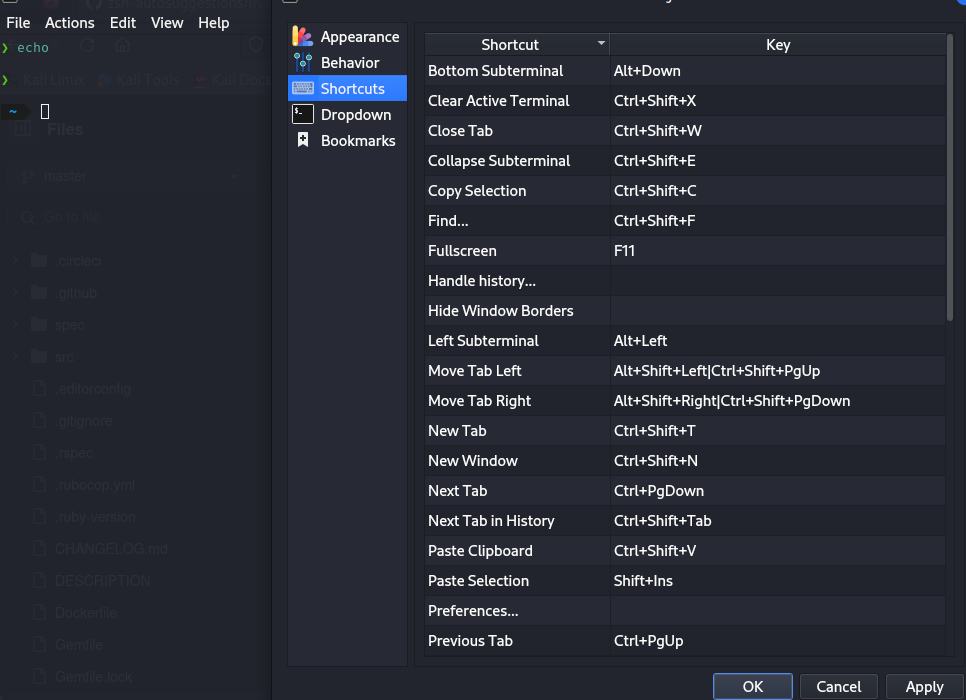
cd /proc/$$ nos dice el directorio de la shell que se esta ejecutando

si hacemos ls –l exe---- Nos dice el tipo de shell de consola de comandos que estamos utilizando

ls –l /proc/$$/exe -------- Nos dice la shel que se esta ejecutando

editamos el fichero .zshrc com emacs - .zshrc es nuestro fichero de configuración, el que personalizamos

Si en un terminal kali hacemos: file / preferences / shorcuts, podemos modificar las combinaciones de teclas del mismo



## Recopilacion avanzada de información

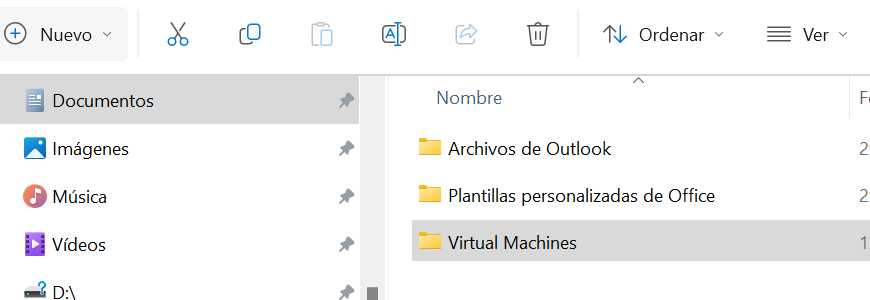
Vulnhub – recurso para descargar maquinas virtuales vulnerables

<https://www.vulnhub.com/>

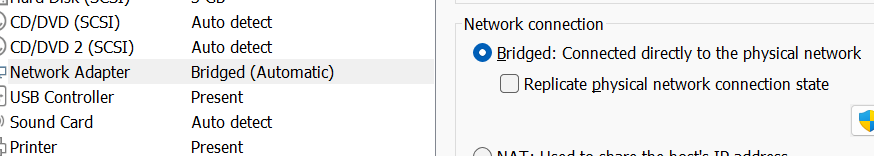
Descargamos la Troll1

Usu:

Las máquinas virtuales se crean en /documentos/virtual machines



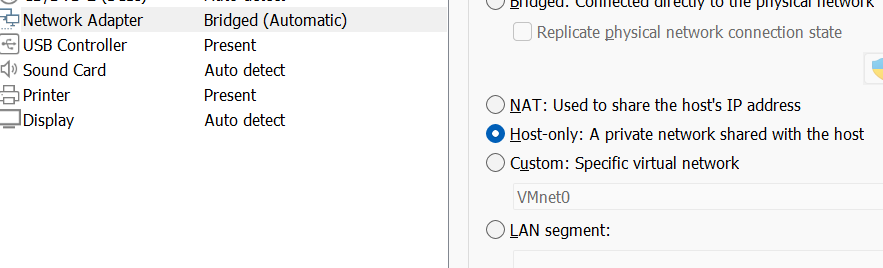
Si editamos los detalles de esta maquina vemos que esta la red en modo Bridge, lo cual la hace vulnerable y no tiene sentido exponerla a nuestra red local, con lo que haremos una red virtual dentro de VMWare



E instalaremos una herramienta de seguridad que se llama snort

## Red privada virtual dentro de VMWare

Formaremos una red de todas nuestras máquinas virtuales. Utilizaremos la red que ya viene creada en modo host only, ya que no tenemos la versión host.



NOTA: En la nueva configuración yo he puesto custo y he cogido vmnet1 (Host-only) para las dos maquinas, la nuestra y la troll

## Herramienta de seguridad que se llama snort – Intercepta el tráfico que pasa por la máquina

Ponemos en google snort download

Snort es un IDS = sistema de detección de intrusos en red, libre y gratuito

Se coloca en el punto central de nuestra red que queremos monitorizar, es decir, el host

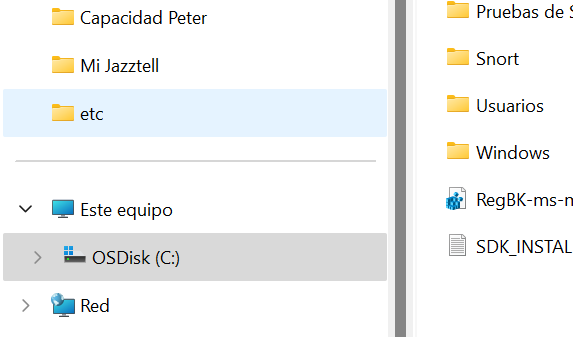
El adaptador de red Host Only se conoce como **vmnet1**: 192.168.244.1

Nos instalamos en binaries el: [Snort\_2\_9\_20\_Installer.x64.exe](https://www.snort.org/downloads/snort/Snort_2_9_20_Installer.x64.exe)

Para interceptar el *trafico que pase por esta maquina*, snort nos advierte que hace falta también una librería que se llama N PCAP, la cual forma parte de la suite de herramientas de Nmap.

Nos descargamos el instaler de ncap para Windows <https://npcap.com/>

Ahora vemos que en c: se ha creado un directorio snort:



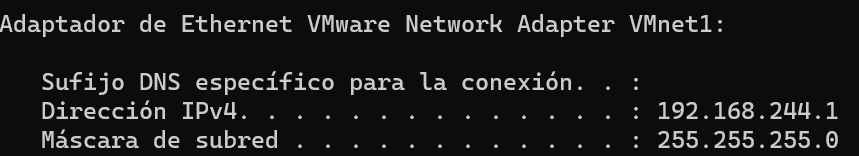
Ahora dentro del directorio etc, donde esta snort.conf hay que hacer algunos cambios:

# Setup the network addresses you are protecting

ipvar HOME\_NET 192.168.20.0/24 --- Esto hay que cambiarlo por el host que nos da ipconfig para vmnet1: 192.168.244.1

Y ponemos ipvar HOME\_NET 192.168.244.0/24

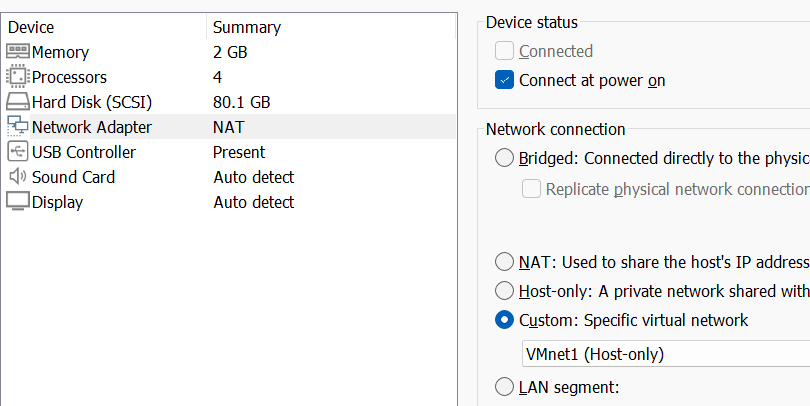
El 0 es porque la máscara de subred termina en 0



## Ejecución de snort en nuestras máquinas virtuales

Ahora arrancamos nuestras dos máquinas en VMware: troll1 y la del curso

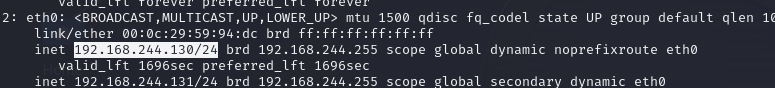
Ahora metemos la maquina de kali (no la troll) en la red virtual que acabamos de crear, yéndonos a settings y en Networkadapter le metemos **vmnet1 (**El adaptador de red Host Only se conoce como **vmnet1**: 192.168.244.1)



Ahora nos vamos al terminal y ponemos: sudo dhclient -- para que nuestro servidor dhcp nos de una nueva dirección ip

Si ejecutamos : ip a

Vemos que la dirección de red esta dentro del rango que le ha asignado nuestro servidor a nuestra maquina **kali**



En la pantalla hay que buscar por inet

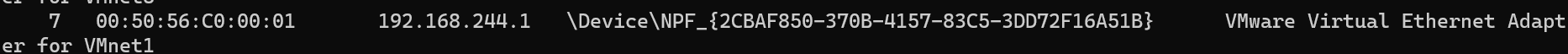
Ahora, para ejecutar snort nos vamos a Windows, c:/snort/bin

Esto lo hacemos desde la powershell, pero antes hay que decirle cual es la interfaz dentro de nuestra máquina **host** por la que vamos a recibir el trafico de red, en nuestro caso **vmnet1: 192.168.244.1**

Para ver las interfaces que tenemos disponibles ponemos

./snort -W

A nosotros de los resultados nos interesa es la vmnet1:



Aquí sale un 7, que es la interfaz por la que tiene que capturar el tráfico de red

Hacemos ahora: ./snort -i 7 -A console -c c:\snort\etc\snort.conf

NOTA: en el comando anterior cuidado con los guiones largos

Le estamos diciendo interfaz 7,

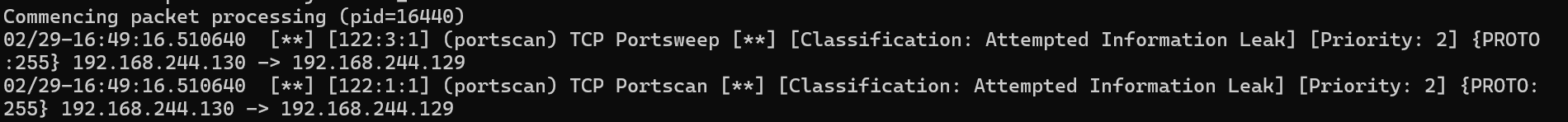
De esta manera snort ya esta escuchando todo el trafico que se intercambia a través de la intenfaz **vmnet1 que es el host-only 192.168.244.1**

Para **probar** snort nos vamos a nuestra maquina kali Linux (no trol) y le decinmos que escanee *todo el segmento de red* para ver que maquinas hay disponibles:

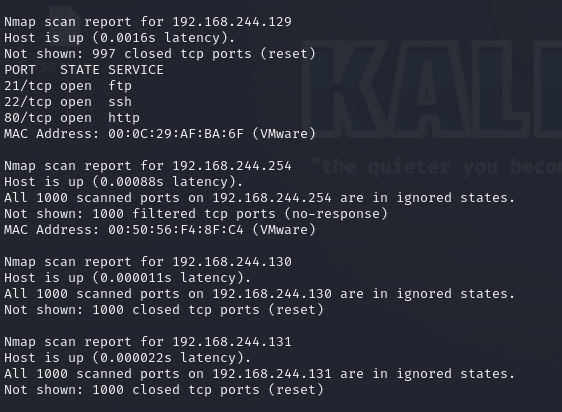
sudo nmap -sS -n 192.168.244.0/24

Lo que queremos es localizar nuestra máquina *troll*

Ahora en el resultado de la consola (La pantalla de powershell, la del **administrador**) vemos las líneas que contienen portscan (Escaneo de puertos)



Esto nos dice que hay un flujo de información entre la maquina .**130** (la kali después de que hiciésemos arriba el dhclient) a la .**129** (que debe ser la troll ya que en el cuadro siguiente pone host is up en la .129 y además pone VMware)

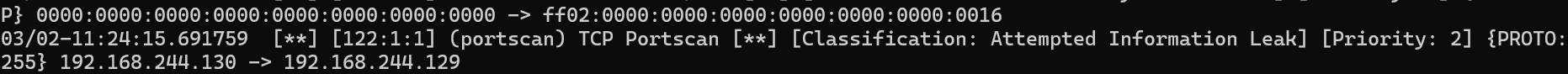


## Reconocimiento avanzado de host o nodos que se encuentran en una red

* La maquina troll es la máquina *vulnerable* 192.168.244.129
* La máquina *kali* es la atacante (La *nuestra*) 192.168.244.130
* Las dos máquinas se encuentran en una *subred virtual* con un **vmnet1** que es el host-only 192.168.244.1 (nuestro nuestro *host* Windows donde está el **administrador**).

Este es el host donde tenemos el *snort*, desde donde como administradores monitorizamos toda la actividad de la infraestructura de red que hemos montado.

Nuestro objetivo es utilizar desde nuestra máquina kali Linux, como *hackers*, técnicas de reconocimiento de información, de puertos, servicios, sin que la máquina que tiene el snort nos *detecte* y **emita ninguna alerta**.

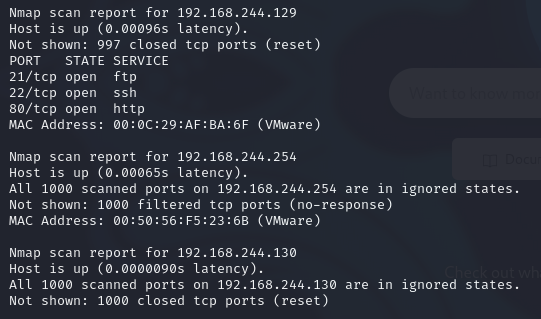


Ahora vemos que cuando el administrador ha activado el monitoreo de red con el snort en el host Windows **vmnet1:** 192.168.244.1 se ha levantado una alerta *portscan* correspondiente al comando *nmap* que ha lanzado el hacker desde la máquina kali Linux.

(probar en mejorar nuestro virus metiendo código para que no nos detecten nuestros comandos antes de ejecutar nuestra powershell de reverse tcp) (Quien detecta la ejecución de la reverse\_tcp, el antivirus o el sniffer del host de la empresa?)

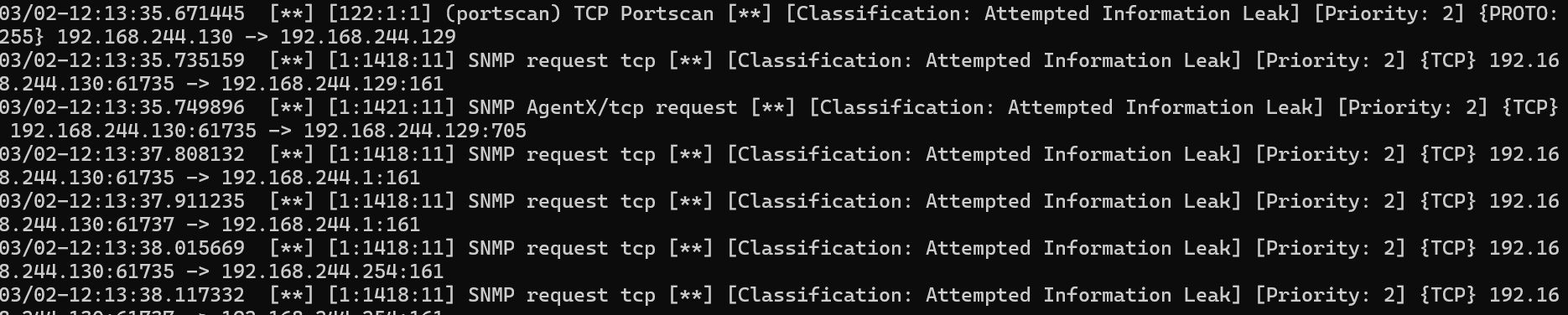
## Escaneo con nmap

Si hacemos nmap -n 192.168.244.0/24 (Para solo escanear nuestra red que nos da nuestro ip config y que hemos configurado enl el snort.conf)



Nos dice que la troll es la acabada en .129 con 3 puertos abiertos y que en la nuestra 130 se han escaneado 1000 puertos tipo tcp closed.

Sin embargo si miramos en el snort de la maquina Windows del administrador se ve que tiene una serie de alertas de escaneo de puertos y los SNMP:



## Escaneo avanzado de host

Vamos a ver que *host* están conectados a una *red*

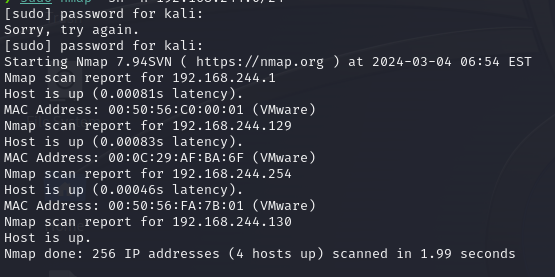
Hemos visto que si hacemos un nmap como el anterior el sistema snort los localiza viendo un “portscanning”, luego:

Ponemos en google nmap host Discovery

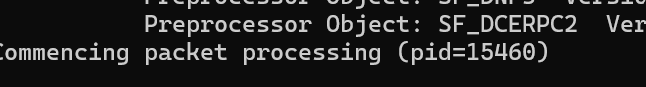
Si en vez de querer localizar puertos como en el capitulo anterior solo queremos localizar *host* y su mac entonces es mas indetectable el nmap, utilizando la opción -sn. Hacemos

sudo nmap -sn -n 192.168.244.0/24

Vemos que solo escanea host, no puertos:



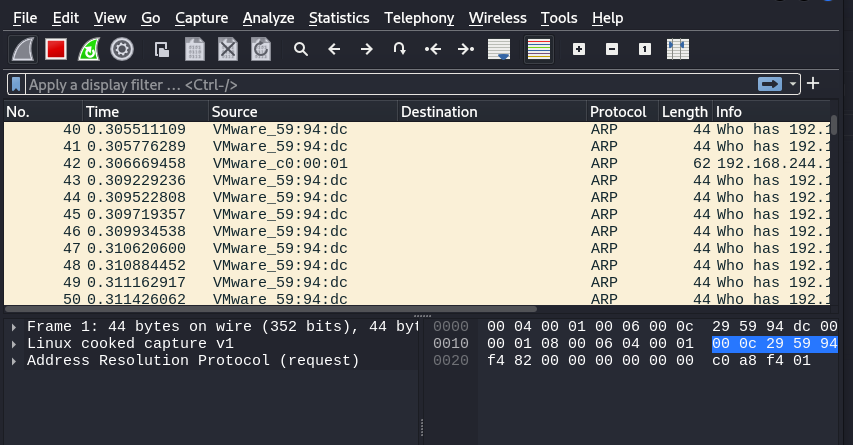
Y vemos que el snort no ha levantado alarmas:



Ponemos también wireshark para ver el trafico de red que se genera. Para arrancar wireshark le metemos como filtro “any”

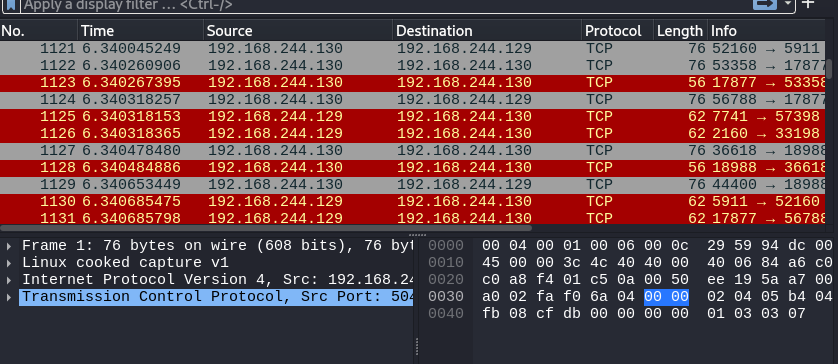
Wireshark es un analizador de paquetes de red, una utilidad que captura todo tipo de información que pasa a través de una conexión

Con el comando sudo nmap -sn -n 192.168.244.0/24 todo el trafico detectado por wireshark solo hace un escaneo de *puertos* y nos saca paquetes de *protocolo* **arp**, no tcp, con lo cual snort no ha localizado nada



Si hacemos la orden inicial de *puertos y host* nmap -n 192.168.244.0/24

Wireshark nos localiza de todo (arp y tcp):



## Escaneo avanzado de host II parte

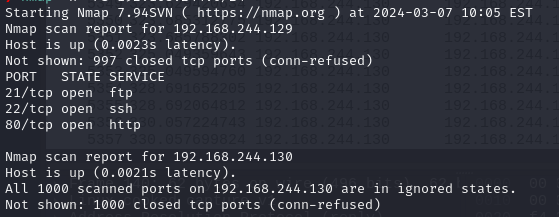
Hay otras opciones de nmap como: nmap -**PS** <port\_list>

nmap -n -PS 192.168.244.0/24

-n es para que no haga resolución DNS reversa

Esto manda un paquete *TCP vacío* con un SYN flag puesto a On

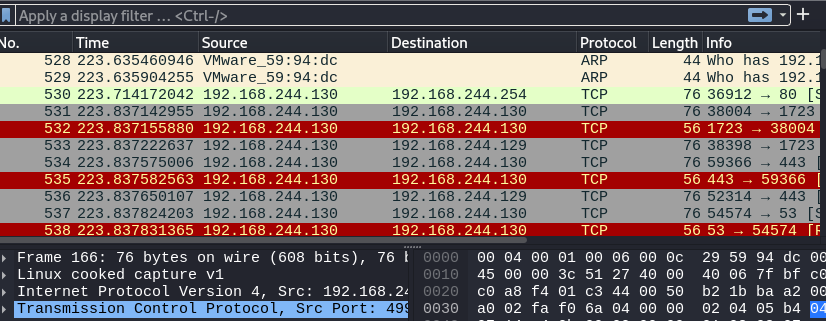
De esta manera verifica que la máquina está activa



Vemos que nos ha detectado la maquina troll1 192.168.244.129 con sus puertos activos pero snort ha detectado nuestro escaneo:



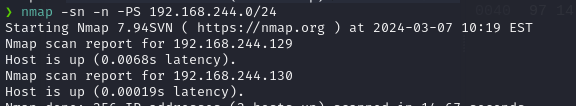
Y vemos que wiresharq también ha capturado un monton de escaneo de puertos TCP. Recordemos que Wireshark es un analizador de paquetes de red, una utilidad que captura todo tipo de información que pasa a través de una conexión



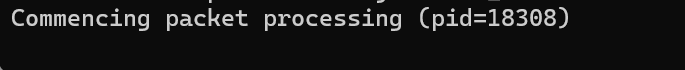
Ahora si metemos el -sn para que no haga *escaneo de puertos*:

nmap -sn -n -PS 192.168.244.0/24

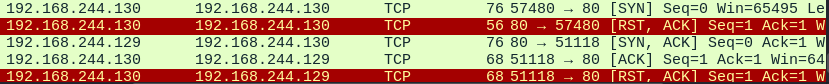
Vemos que nos muestra los dos host encontrados



Pero snort no encuentra amenazas:



Y luego wireshark ve la llamada del 130 y la respuesta del 129, habiendo *filtrado por tcp*



**NOTA**: El escaneo -PS es una de las *mejores maneras* de hacer un nmap

NOTA: Un **ACK**, en comunicaciones entre computadores, es un mensaje que el destino de la comunicación envía al origen de esta para confirmar la recepción de un mensaje

Si se recibe un paquete SYN/ACK esto indica que el *puerto* está en escucha (*abierto*), mientras que si se recibe un **RST (reset)** indica que no hay nada escuchando en el puerto.

El *protocolo* **ARP** es una parte integral del conjunto de protocolos de Internet (TCP/IP). Su función principal radica en mapear direcciones IP a direcciones físicas de hardware, como las direcciones MAC en redes locales. En términos sencillos, ARP se encarga de encontrar la dirección física asociada a una dirección IP específica.

Una vez que hemos localizado los *host* o **nodos** que hay conectados a una red hay que ver que *puertos* tienen abiertos, sin ser detectados por sniffers como snort.

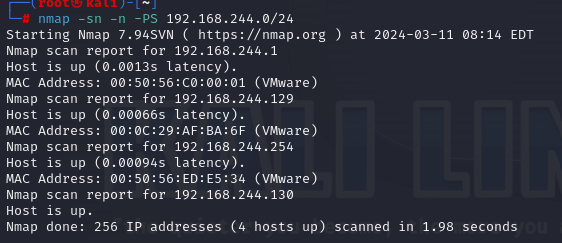
Posteriormente habrá que ver que *servicios* se están exponiendo

## Escaneo avanzado de puertos

Abrimos snort: ./snort -i 7 -A console -c c:\snort\etc\snort.conf

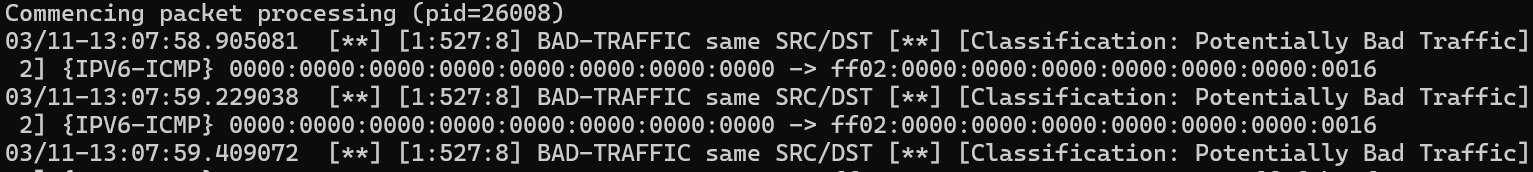
Hacemos primero un escaneo de host, no de puertos:

nmap -sn -n -PS 192.168.244.0/24



Vemos que detecta el nuestro 130 y el trol 129

**NOTA**: Todas las señales que detecta smort de bad trafic tienen que ver con las maquinas virtuales, no con el escaneo de nmap



NOTA: Este escaneo de **PS** es el mejor ya que no es detectado por snort

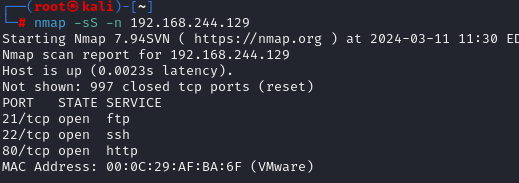
Entonces, nuestra máquina a atacar es la troll, tiene la IP según el nmap anterior: 192.168.244.129 y lo que vamos a hacer es ver que puertos tiene abiertos:

nmap **-T0** -sS -n 192.168.244.129 (Que es la ip de la máquina troll)

El **-T0** es el modo sililoso

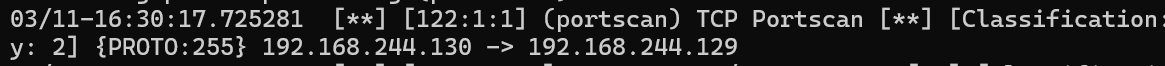
-sS hace que nos centremos solo en puertos abiertos y no cerrados

NOTA: el –n es para que no haga resolución DNS reversa



Vemos que hay 3 puertos abiertos tcp.

En *snort* vemos una traza de portscan, con lo cual este nmap de puertos no es muy valido



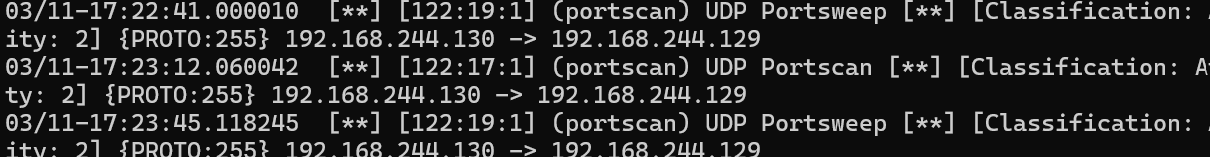
NOTA: Para tener el teclado querty hacemos: ./inicio.sh

O bien: sudo setxkbmap es;

Si escaneamos en vez de puertos TCP puertos **UDP** (U) se tiene:

nmap -s**U** -n 192.168.244.129 (Que es la ip de la máquina troll)

Tenemos en snort un portscan de UDP:



Cq tipo de escaneo de puertos es interceptado ya que según se ve en wireshark se lanzan mogollon de peticiones TCP

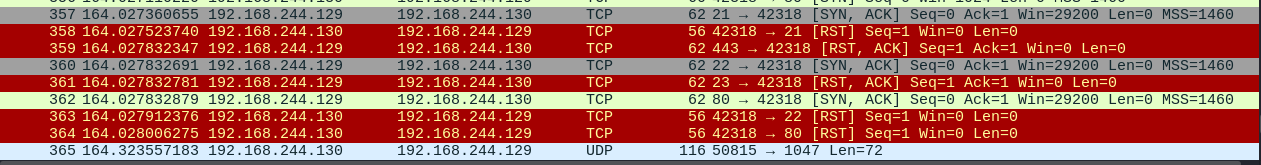
Para restringir tanto mandar peticiones que pille el snort vamos a lanzar peticiones solo de **5** puertos (es el máximo de puertos a escanear antes de que snort salte) más utilizados por nmap. Esta es la mejor manera de escanear puertos:

nmap -sS -n -top-ports 5 192.168.244.129

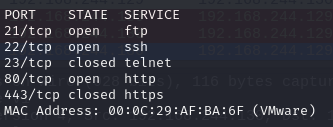
Entonces arrancamos snort con: snort: ./snort -i 7 -A console -c c:\snort\etc\snort.conf

(Cada vez que se haga un escaneo conviene cerrar snort)

Vemos que wireshark ha capturado las comunicaciones y también snort



Ha escaneado los 5 primeros puertos, sin que snort lo capture:



## Evasión de escaneos con nmap - Fragmentacion

Si ponemos en google nmap firew All evasión, son herramientas de seguridad orientadas a la evasión. Coge los paquetes de un escaneo TCP y los divide en otros mas pequeños, de bloques de 8 o 16 bytes. De esta manera nos saltamos los firewalls

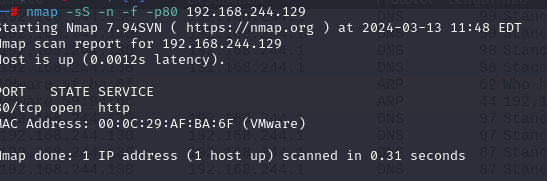
NOTA: Es relativamente fácil detectarlo

Hacemos: nmap -sS -n -f -p80 192.168.244.129

NOTA: con el –f le decimos que fragmente la información en paquetes de 8 bytes

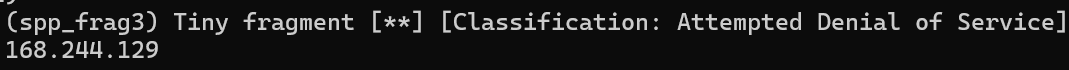
De esta manera escaneamos un puerto en particular , el 80 que como vimos en la imagen de arriba era un tcp open

Tenemos como salida

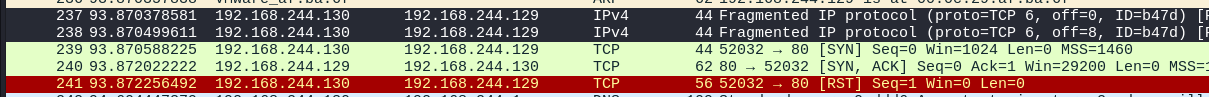


Que nos indica que esta abierto

Y el snort si que ha visto esto y lo interpreta como ataque de denegación de servicio.



Para hacer ataques DDOS el paquete hacemos ataques a 1 solo puerto.

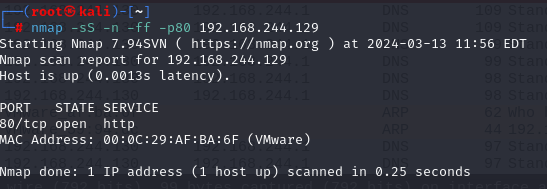


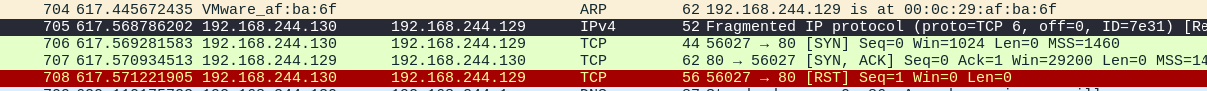
Y con el parámetro –f vemos que no solo utiliza TCP sino también paquetes IPv4, vemos por tanto que ha **fragmentado** nuestro paquete en varios frames.

Arrancamos snort

./snort -i 7 -A console -c c:\snort\etc\snort.conf

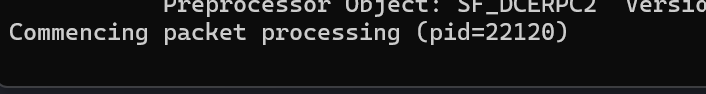
Si queremos fragmentar en paquetesw de 16 bytes hacemos con 2 f. Hacemos fragmentación en paquetes: nmap -sS -n -ff -p80 192.168.244.129





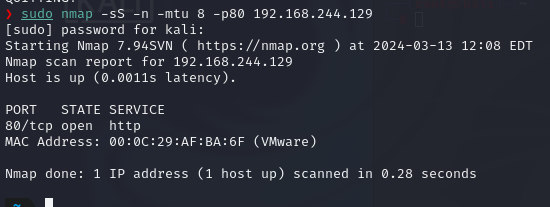
Vemos que hay un ipv4 y solo 2 TCP, porque al ser de 16 bytes no se necesitan mas envios

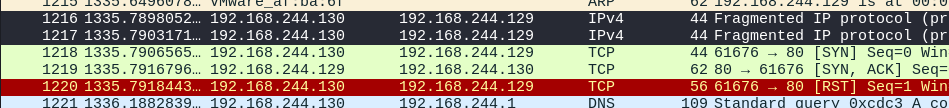
En este caso de doble fragmentación snort **no detecta** el escaneo



Si queremos prescindir de la opción –ff podemos utilizar la opción **–mtu**, en la cual le decimos que queremos que fragmente en paquetes de minimo de 8 bytes y múltiplo de 8

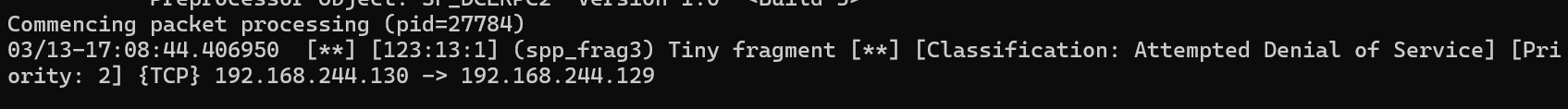
nmap -sS -n -mtu 8 -p80 192.168.244.129 Esto es lo mismo que utilizar *un solo* -f





Lanza los IPv4 y los TCP con fragmentos de 8 bytes

Y snort los detecta rápido



Con lo cual no es un escaneo valido

## Evasión de escaneos con nmap – Utilizando señuelos – Decoy scan

Hace que nmap haga un *spoofing* en la dirección que nmap realiza el escaneo, esto es Decoy scan

Lanzamos escaneos a esa máquina pero modificando la dirección IP de origen

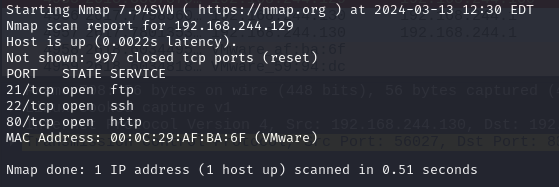
De esta manera el IDS o (Sniffer) va a ver que hay varias máquinas intentando atarcarle pero no va a saber cual es la maquina que le ataca realmente.

Vara ello utilizamod la opción –D y una serie de direcciones IP que utilizamos como señuelo

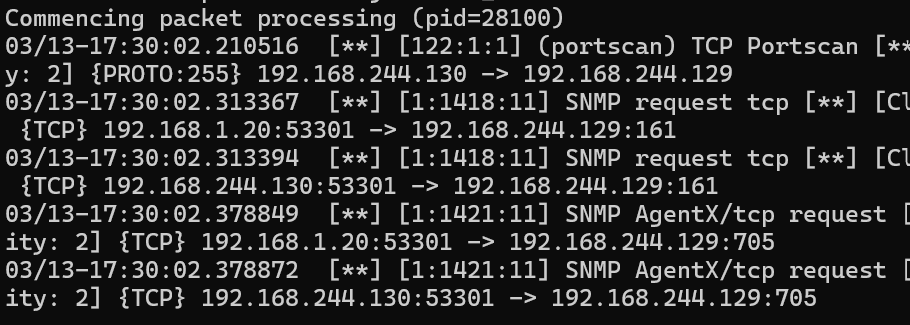
nmap -sS -n -D 192.168.1.20,ME 192.168.244.129

El ME indica cual es la dirección mia que realiza el escaneo

Estas direcciones ip deben corresponder para ser mas realistas al nmap de host que hemos hemos anteriormente, es decir, host que estén en esta red



Aquí vemos que snort emite una serie de alertas:



Pero vemos que en dichas alertas se ha colado la dirección ficticia 192.168.1.20, el cual era el señuelo. Es útil en entornos reales para confundir a los administradores

## Evasión de escaneos con nmap – Spoofing de la identidad del atacante con nmap

El spoofing es manipular la dirección origen desde la que estamos realizando un nmap a la máquina objetivo.

Escaneamos con la opción -S nuestra máquina troll:

nmap -PN -S 192.168.211.1 -n -e eth0 192.168.244.129

192.168.211.1 – Este debe ser un host que este activo en nuestra red, visto previamente con el escaneo de host

Esto en esta versión de nmap no funciona

Lo que hace es omitir nuestra dirección origen y poner otra dirección.

Esto es útil para los firewall que restringen conexiones en base a la dirección IP origen

## Creación de programas de Python en Linux para usar nmap

Se hace con emacs <nombre\_archivo>

Para poner el editor en Python mode se hace

Emacs <nombre\_archivo**.py**>

Para salvar el archivo hacemos ctrl+x ; ctrl +c

Mirar: detección y evasión de defensas, anonimato

## Detección y evasión de defensas

Análisis estático de firmas

Concepto de *lista negra* – Indicamos explícitamente aquello que queremos bloquear o detectar. Se le indica que conceptos se quieren detectar.

Concepto de *lista Blanca -* Indicamos aquello que queremos permitir y el resto lo bloqueamos

Se basa en detectar en el flujo formas de malware y se genera una firma, que es un patrón o caracteres (Los *primeros* bytes del binario). De esta forma nuestra herramienta de seguridad busca si esos bytes que ya habíamos identificado en un código malicioso aparecen en ese código binario:

* Si aparecen se bloquea ese código binario y emite una alerta de que se ha detectado un elemento malicioso. Si nosotros ofuscamos u ocultamos esos bytes que busca la herramienta nos lo podremos saltar.

Análisis heurístico de firmas

Establece en vez de unos bytes determinados un *sistema de reglas* que queremos localizar en los payloads.

* Nombre del proceso que ejecuta
* Tipo de proceso que ejecuta
* Si escribe en memoria o en disco

Se trata de un análisis basado en el comportamiento del código.Esto es más potente que el análisis statico de firmas ya que si realiza unos comportamientos esperados, aunque sea un nuevo virus y no esté fichado lo localizaremos.

Análisis dinámico

Coge el binario (por si hemos cifrado el payload), lo ejecuta y espera un tiempo. Analiza las acciones que ha hecho en un periodo de tiempo y emplea las dos tecnicas anteriores (heurística y statico) para ver si es un código legítimo. Este método consume mucho tiempo y recursos.

Tipos de herramientas de seguridad

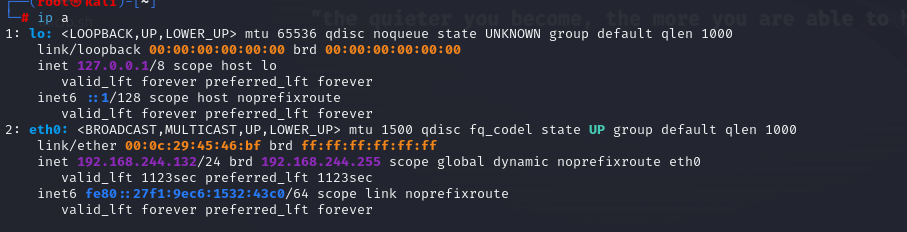


## Detección y evasión de defensas – Balanceador

## Resolución de la máquina Tr0ll1

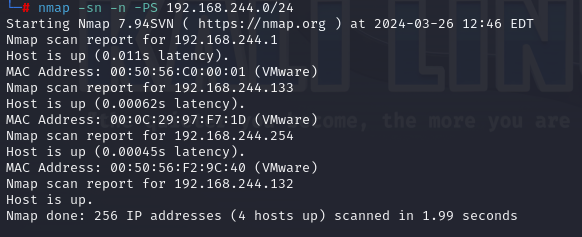
Para poner el teclado querty hacemos: sudo setxkbmap es

* Abrimos snort: ./snort -i 7 -A console -c c:\snort\etc\snort.conf
* Hacemos un ip a para ver que ip tenemos nosotros: 132



* Hacemos primero un escaneo de host, para ver el host que queremos atacar

nmap -sn -n -PS 192.168.244.0/24 (este escaneo no lo detecta snort)



* Hacemos un escaneo de puertos.

nmap -sS -n 192.168.244.133 (Que es la ip de la máquina **troll**. Pero si lo detecta snort)



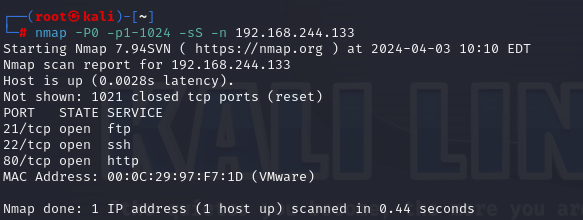
Con esto ya sabemos que hay abiertos 3 puertos 21,22,80

Ahora intentamos un escaneo de puertos sobre un rango menor

nmap -T0 -p1-1024 -sS -n 192.168.244.133

o si se quiere más rápido y que no lo detecte snort:

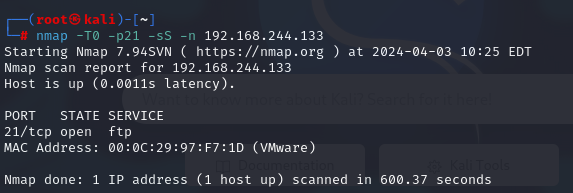
nmap -p21-22 -sS -n 192.168.244.133



<https://nmap.org/idlescan-es.html>

hay que usar la opción "-p-" si sólo se quieren escanear los puertos bastante conocidos ("well know ports") además de los puertos 1-1024

NOTA: Un escaneo con -T0 tarda aproximadamente 10 minutos por puerto escaneado, asi que lo mejor es limitarlo a un rango tipo 21-80 y dejarlo por la noche:

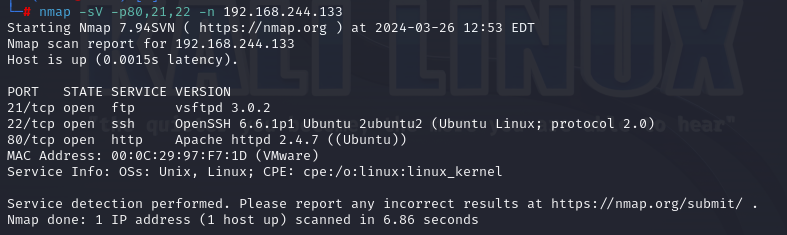


En este caso de escaneo silencioso vemos que snort **no ha detectado nuestra intrusión**

nmap -T0 –p21-80 -sS -n 192.168.244.133

* Ver que *servicios* están corriendo en esos puertos y sus versiones:

nmap -sV -p80,21,22 -n 192.168.244.133



(Hacerlo con snort para que no lo pille aunque sea fraccionando)

nmap -sV -n -ff -p80,21,22 192.168.244.133 (Asi no lo pilla, pero claro, la anterior búsqueda de puertos si que la pillo)

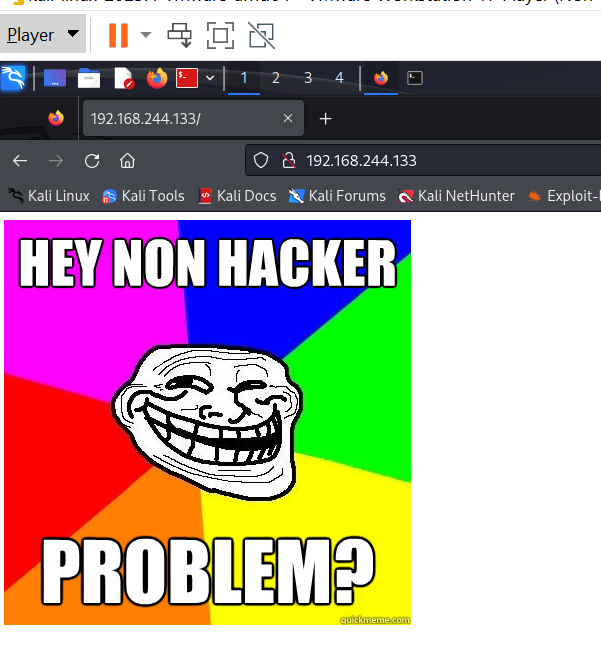
* Lo siguiente es ver si los servicios y versiones anteriores pueden tener alguna vulnerabilidad pública

Si vemos el puerto 80, que esta abierto y es un http apache, se podría suponer que este dando servicio tb a alguna aplicación web

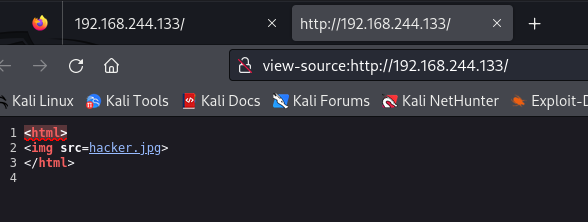
Si ponemos en un navegador web:

http:// 192.168.244.133 (que es la dirección de la maquina troll)

vemos que si hay alguna aplicación corriendo



Podriamos tb ver el código fuente de la maquina con botón derecho + ver código fuente, pero en este caso no hay nada porque solo hay una imagen



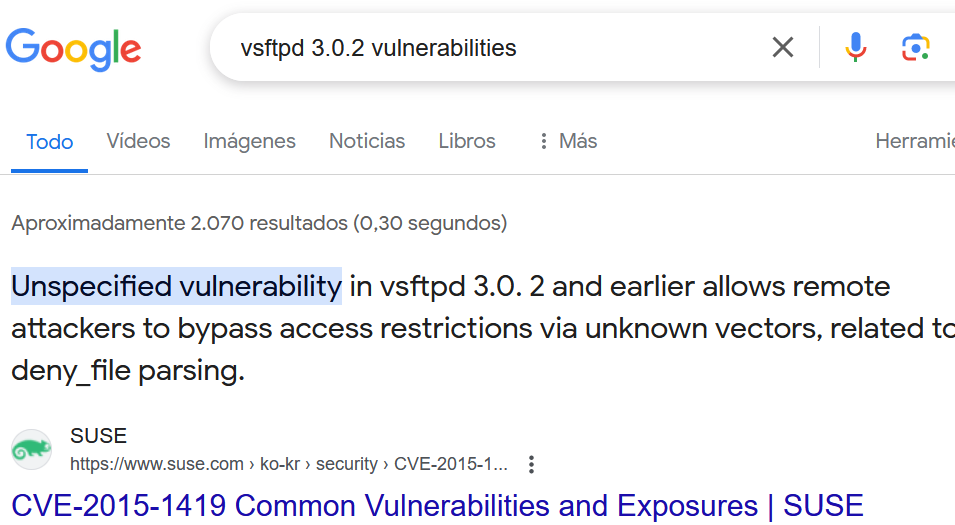
* Lo siguiente es ir buscando en google las frases

Apache httpd 2.4.7 vulnerabilities o cve (exploit)

OpenSSH 6.6.1p1 vulnerabilities

vsftpd 3.0.2

Para ver si existen en internet vulnerabilidades asociadas:



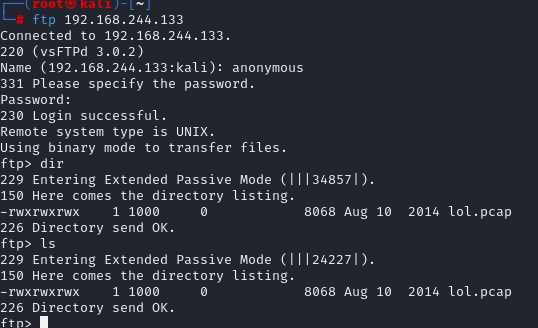
* Otra de las cosas que vemos es que tiene el ftp abierto en el puerto 21, y el servicio ftp de manera general suele incorporar un acceso anónimo:

Hacemos un: ftp 192.168.244.133

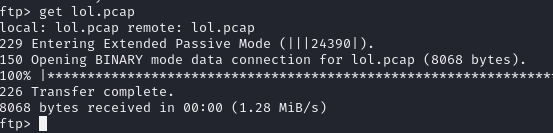
De usuario ponemos : anonymous

Passwd: la dejamos vacia

Y vemos que si hacemos un ls vemos que hay un fichero: lol.pcap



Este tipo de ficheros suelen ser de análisis de red de herramientas como wireshark, con lo que haremos un get y nos lo traemos:



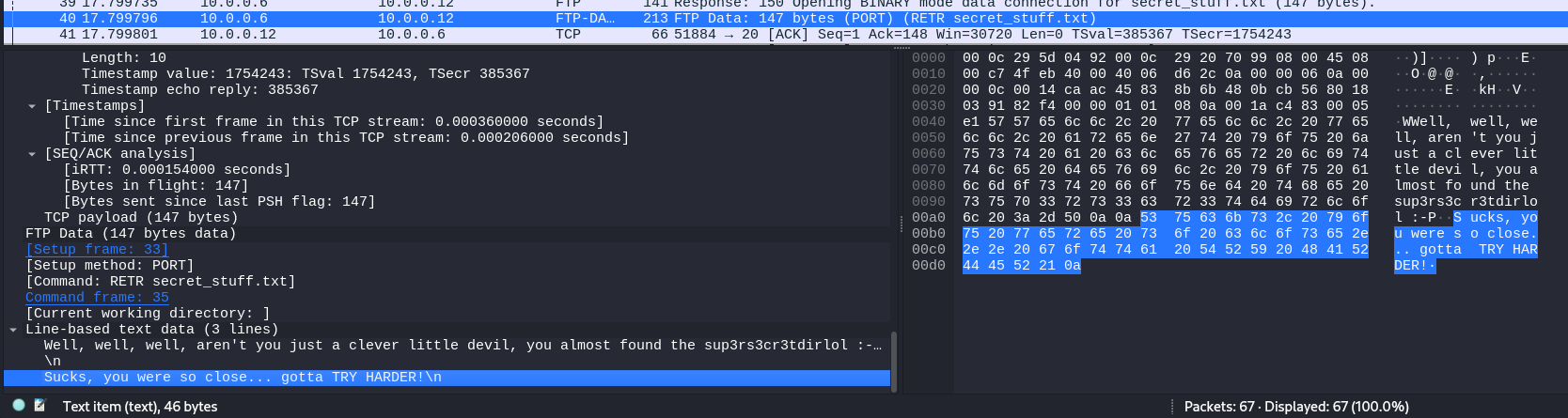
Cerramos el ftp y analizamos el fichero extraido haciendo: wireshark lol.pcap



Dentro del pcap vemos que se han hecho conexiones ftp y normalmente se tienen pistas para avanzar. Lo que podemos buscar son transferencias de datos, los mensajes encabezados por “FTP data”



En ese ftp data tenemos:



Un mensaje en el que nos dice que estamos cerca pero que nada, que nos falta algo. Dice que hemos estado so close(cerca) de encontrar el supersecret dir lol:

Hacemos un copy de “Line-based text data (3 lines)>”

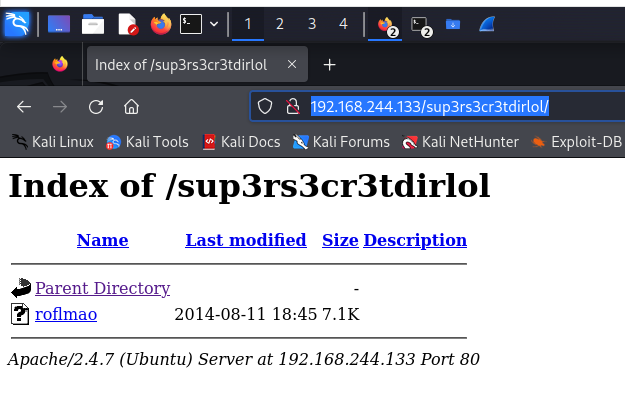
”””Well, well, well, aren't you just a clever little devil, you almost found the sup3rs3cr3tdirlol :-P\n

Sucks, you were so close... gotta TRY HARDER!\n”””

Asi que vamos a probar via web a encontrar el directorio sup3rs3cr3tdirlol

Si ponemos en un navegador web:

http://192.168.244.133/sup3rs3cr3tdirlol



Accedemos a /home/kali/Downloads/

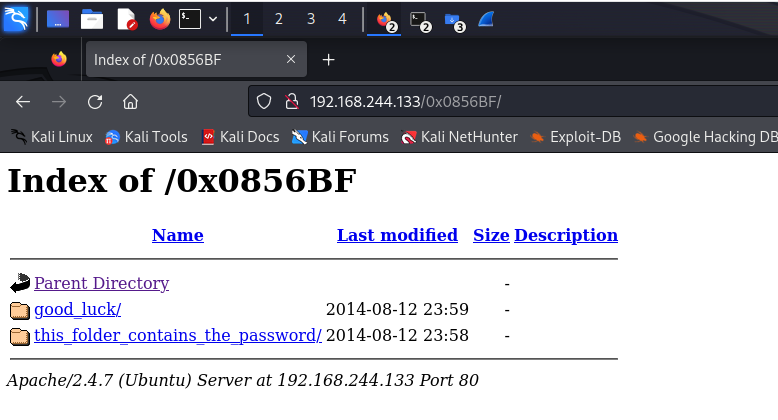
Ya que el roftmao es un ejecutable

Lo ejecutamos y nos dice “Find address 0x0856BF to proceed”

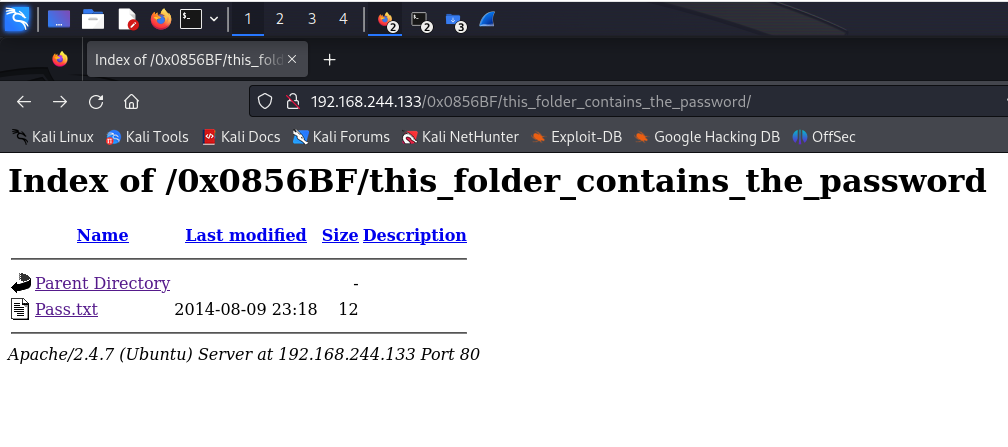
Añadimos esta cadena a nuestra dirección web

<http://192.168.244.133/0x0856BF/>

Y tenemos:

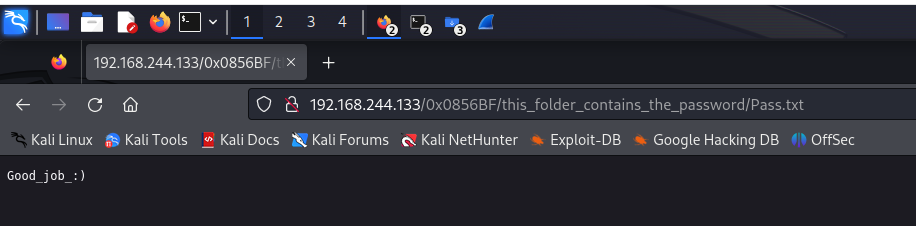


Vamos a la carpeta que contiene la password:



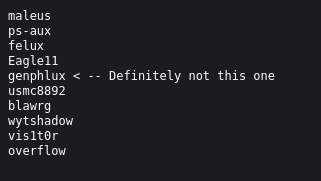
Abrimos el Pass.txt

y tenemos: Good\_job\_:)



Lo metemos en la máquina troll por ssh con los usuarios contenidos en el otro fichero “users.txt”

En: [http://192.168.244.133/0x0856BF/good\_luck/which\_one\_lol.txt>users.txt](http://192.168.244.133/0x0856BF/good_luck/which_one_lol.txt%3eusers.txt)



Y asi probamos por fuerza bruta user y password:

ssh genphlux@192.168.244.133

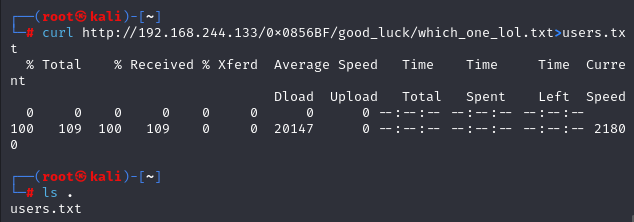
siendo: ssh <usuario>@<maquina>

Para descargarnos este fichero utilizamos **curl.**

cURL (Client URL) es una herramienta de línea de comandos, que permite transferir datos hacia o desde un servidor sin interacción del usuario

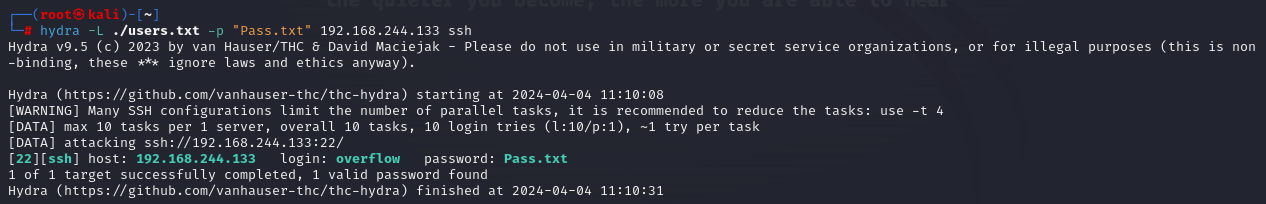
curl [http://192.168.244.133/0x0856BF/good\_luck/which\_one\_lol.txt>users.txt](http://192.168.244.133/0x0856BF/good_luck/which_one_lol.txt%3eusers.txt)

con esto nos hemos descargado el fichero al directorio de la máquina, al igual que paso con el pcap.

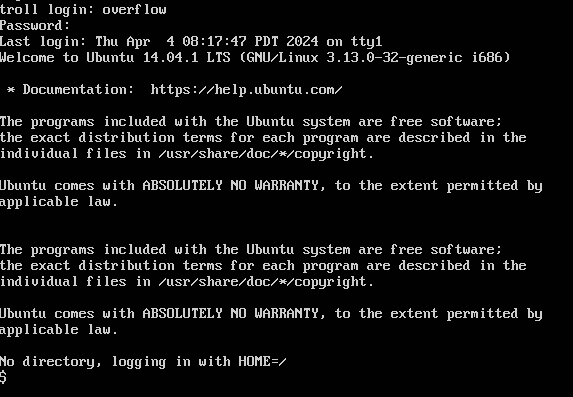
****

Para hacer el ataque a la pagina web de fuerza bruta tenemos una herramienta que es Hydra:

hydra -L ./users.txt -p "Pass.txt" 192.168.244.133 ssh

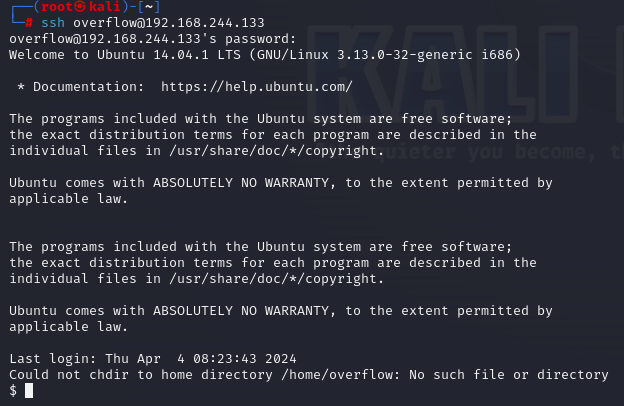


Y ahora hay que conertarse a la maquina troll



O nos logamos con:

[overflow@192.168.244.133](mailto:overflow@192.168.244.133)



Si estuviésemos en una auditoria de seguridad ya estaríamos en la fase de postexplotación, ya que ya hemos conseguido comprometer la máquina.

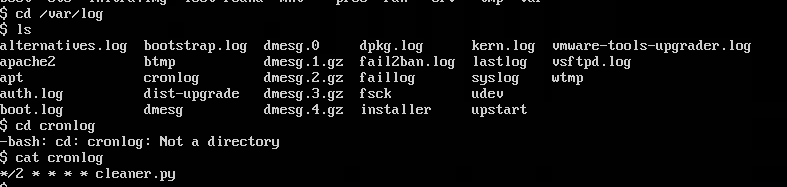
Ahora vemos que hay algún proceso cron que nos cierra la sesión, con lo cual podemos o hacerle un kill o bien sabiendo que la versión en la imagen anterior es Ubuntu 14.04.1 ver si hay algún sploit que pueda hacernos escalar privilegios en la maquina

Overflow Pass.txt

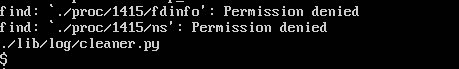
sudo setxkbmap es;

Vamos a ver si hay algún cron corriendo para lo cual exploramos el directorio de los logs:

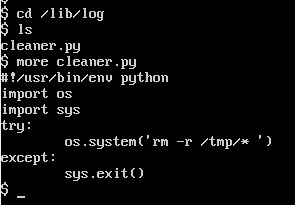
cd /var/log



Vemos que hay una tarea en ejecución periódica que es cleaner.py



Vamos a /lib/log y hacemos un cat de cleaner.py para ver lo que hace



Y vemos quien tiene permisos para ejecutar este fichero:

Hacemos ls –la cleaner.py



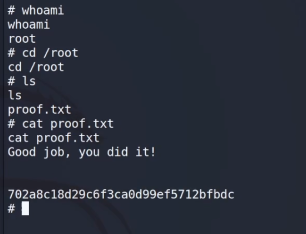
Y vemos que tienen privilegios de ejecución 777, con lo cual si lo editamos podemos ejecutar cualquier comando.

Podemos hacer una reverse Shell que haga ejecutar en la maquina troll con privilegios de root:

Hacemos en el pc trol un py con una reserse shell



Y desde nuestro pc Linux hacer un net



Se mira en el prof.txt y esta el hash que es lo que se entrega para resolver la máquina.

Aprovechando que no tenemos el Linux actualizado, Tambien hay publicado en internet, buscando la versión de Linux, un exploit que sirve para elevar privilegios:

“”Apport 2.14.1 (Ubuntu 14.04.2) - Local Privilege Escalation””

Hay un exploit 37292 que esta en Python y sirve para escalar privilegios

Lo podemos copiar, meter en nuestro pc de troll y hacer una reverse Shell con esto.

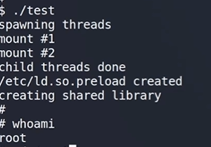
Nos vamos a tmp que es el único directorio donde podemos escribir



Compilamos: gcc expl.c –o text (Metiendole dentro el código del exploit)

Ejecutamos ./test

Y vemos que nos convierte en root



Hacemos un cd /root/

Cat proof.txt y todo resuelto

NOTA: ir y resolver otras maquinas virtuales:

<https://www.vulnhub.com/>

Buscar “difficulty easy”

**NOTA**:

* Cerrar maquina con shutdown
* Como editor de texto utilizamos *nano*

## Tomar control de otro equipo con anonimato sin exponer mi ip

hackear equipo con anonimato sin exponer mi ip

## Como se hace una intrusion solo por ip?

se buscan los puertos abiertos a la escucha... se trata de adivinar que programa está escuchando, se investiga sobre el programa encontrado y se busca como procesa los datos de entrada, *se busca un bug en el programa que te permita usar esa entrada de datos a tu favor, se explota la entrada de datos* y se obtiene lo que se pueda por ella (por lo general explotar más bugs hasta que logres "salir del programa" para poder acceder al windows)

por eso es que los exploit hacen un .exe, ese es un "hueco" a la pc victima que te permite saltarte todos esos pasos... y es dificil conseuir un programa que esté solo a la escucha, porque a estas alturas poco lo hacen (por razones practicas que no explicaré y por razones lógicas como evitar estos ataques)

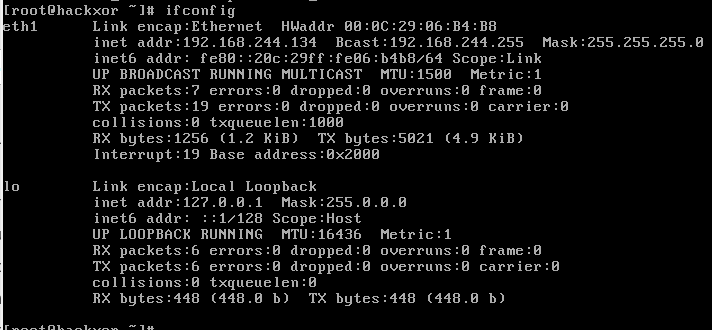
[Hakear pc sin aceso fisico solo con ip (elhacker.net)](https://forum.elhacker.net/hacking/hakear-pc-sin-aceso-fisico-solo-con-ip/)

## Resolver Hackxor: 1

Interpretas a un hacker profesional de sombrero negro contratado para rastrear a otro hacker por cualquier medio posible. Comience revisando su correo electrónico en wraithmail y vea qué tan lejos puede llegar en la madriguera del conejo.

Los sitios web clave de este juego son http://wraithmail:8080 http://cloaknet:8080 http://gghb:8080 y http://hub71:8080 por lo que si no tienes ganas de rastrear a tu objetivo, puedes hackearlo en cualquier orden. Cada sitio web se introducirá adecuadamente a través de la trama.

* logging in with username:root pass:hackxor
* typing ifconfig



Vemos que su ip en nuestra red es 192.168.211.130

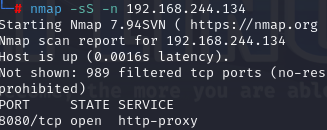
Y el nuestro de kali es 192.168.244.129

* Hacemos un escaneo de host nmap -sn -n -PS 192.168.244.0/24

Ahora vemos puertos y servicios

nmap -sS -n 192.168.244.134

Y vemos que tienen el http proxy abierto en el puerto 8080



Deberiamos poder hacer un http://192.168.244.134:8080/

Pero no tenemos explorador. Como somos root tenemos privilegios para instalar un explorador en kali:

* Configure your hosts file (/etc/hosts on linux) to redirect the following domains to the IP:
  + <http://wraithmail:8080> and login with username:algo password:smurf

si hacemos <http://wraithmail:8080> desde otro pc con reverse shell

En eth1 tenemos: 192.168.244.134

wraithmail, wraithbox, cloaknet, GGHB, hub71, utrack

metemos en /etc/hosts la línea

192.168.244.134 wraithmail

Y asi nos saltamos la redirección automática de http a https que esta haciendo firefox

* + <http://cloaknet:8080>
  + <http://gghb:8080>
  + <http://hub71:8080>

Que es lo que hace el /etc/hosts:

Este fichero se utiliza para obtener una relación entre un nombre de máquina y una dirección IP: en cada línea de /etc/hosts se especifica una dirección IP y los nombres de máquina que le corresponden.

De ahí tenemos las ip de los servidores.

Nos conectamos en mozilla desde nuestra mauina kali a:

<http://wraithmail:8080/>

<http://cloaknet:8080>

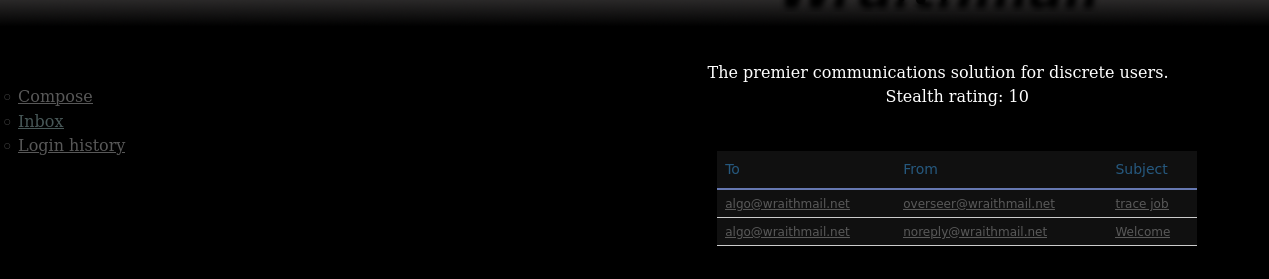
<http://gghb:8080>

<http://hub71:8080>

Se nos muestra la pantalla:



Metemos el usuario y password: username:algo password:smurf



Interpretas a un hacker profesional de sombrero negro contratado para rastrear a otro hacker por cualquier medio posible. Comience revisando su correo electrónico en wraithmail y vea qué tan lejos puede llegar en la madriguera del conejo. Cada sitio web se introducirá adecuadamente a través de la trama.

Vemos que ha habido un mail del usuario *overseer*

En /var/spool/ están los directorios de correos en Linux

Si en la pagina del gestor de email de nuestra maquina Linux hacemos un botón derecho viewpagesource nos encontramos con la lines:

<http://wraithbox:8080/htmlisland.jsp?messageid=9660695031371369539>'

Este es el mensaje enviado por overseer

Para ver el contenido de un mensaje hacemos: postcat -q ID

postcat -q 9660695031371369539

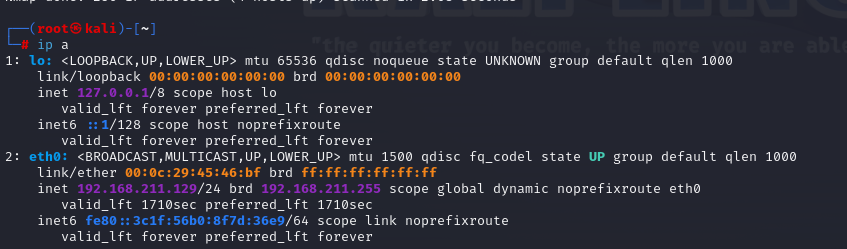
## Resolver KIOPTRIX\_level 2 Adquirir privilegios de root

Methodology

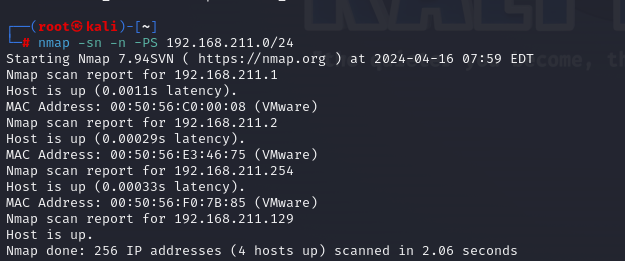
* Network Scanning
* Enumeration
* Exploitation
* Gaining root access
* Hacemos primero un escaneo de host, para ver el host que queremos atacar

Ip a

Tenemos nuestra maquina 192.168.211.129/



nmap -sn -n -PS 192.168.211.0/24



Luego la ip de la maquina es 192.168.211.2

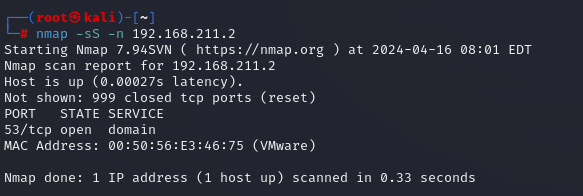
* Escaneamos puertos y servicios

nmap -sS -n 192.168.211.2

Realizaremos un escaneado de los puertos para ver los servicios que se peuden encontrar:

nmap -A -Pn -n -p- 192.168.211.2

nmap -sS -n 192.168.211.2

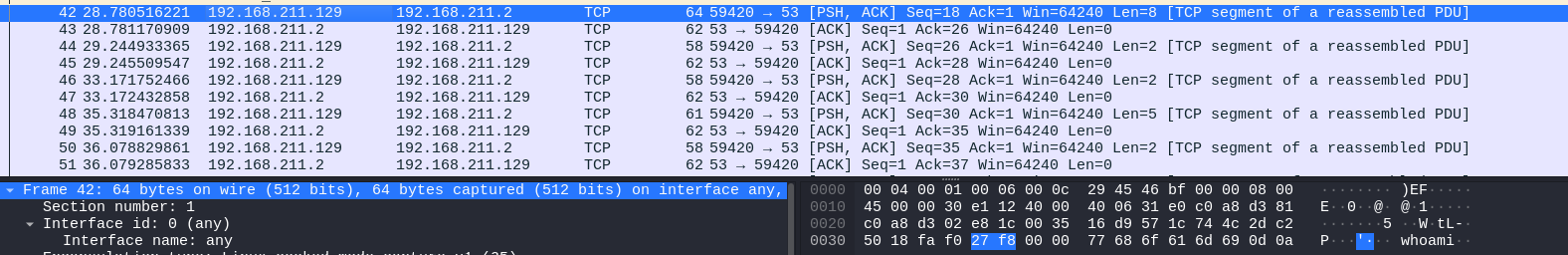


Tenemos un servicio tcp domain abierto, el puerto 53, lo9 cual es para dns, it is for DNS you want it open if you want DNS to work, con lo cual esto no nos vale de nada, es como si la máuina destino no tuviese nada abierto

1. Escriba "telnet + dirección IP o nombre de host + número de puerto" (por ejemplo, *telnet www.synology.com 1723* o *telnet 10.17.xxx.xxx 5000*) para ejecutar el comando telnet y probar el estado del puerto.
2. Si el puerto está abierto, un mensaje indicará **Connected to 10.17.xxx.xxx**.
3. 

Luego hacemos: telnet 192.168.211.2 53

Si hacemos un telnet 192.168.211.2 53 y enviamos un whoami se ve:



* Una comunicación del 192.168.211.129 (nuestro host) al 192.168.211.2 con un mensaje en la esquina de abajo a la derecha de ‘whoami’

Comando nc

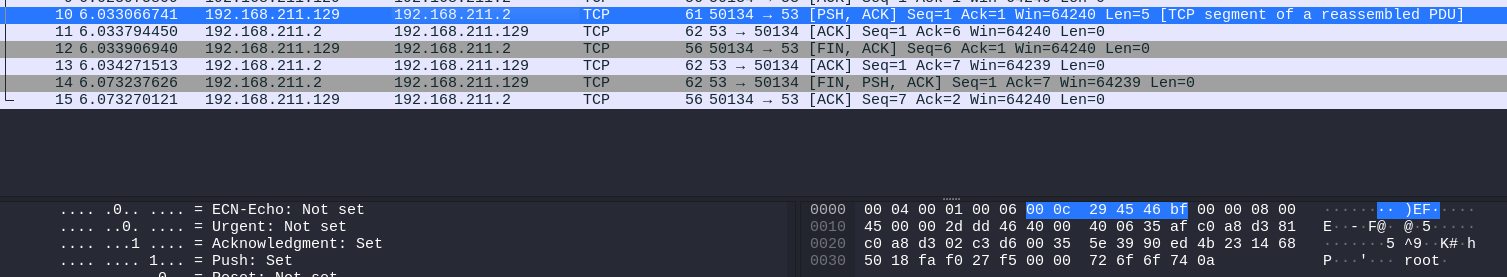
O ejecutamos: nc maquina\_destino puerto\_destino

**nc** crea un socket para conectarse al puerto indicado de la máquina destino

**-c comando** / **-e ejecutable** ejecuta un comando/programa una vez iniciada la conexión cuyas entrada y salida estándar están redirigidas a la conexión establecida

nc 192.168.211.2 53 -c whoami

vemos lo que tiene wireshark:



Es una respuesta ‘root’

Ahora sólo hace falta escalar privilegios y hacerse con el control de la máquina.

Vamos a ver si hay algún cron corriendo para lo cual exploramos el directorio de los logs:

cd /var/log

Lo mejor es hacerlo con un **NC** ya que tarda más en romper la conexión:

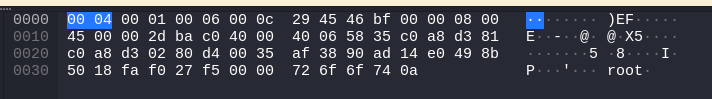
Se puede hacer:

* un ls -la
* Hacer un cd /root/ y ver que hay en ese directorio
* Y hacer un pwd para comprobar que estábamos en el directorio root

Un **ACK**, en comunicaciones entre computadores, es un mensaje que el destino de la comunicación envía al origen de esta para confirmar la recepción

nc 192.168.211.2 53 -c whoami

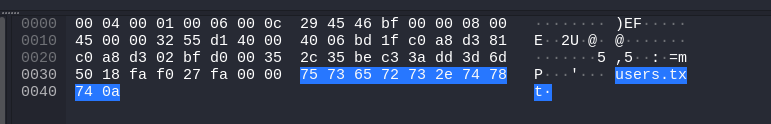
Se tiene la respuesta:



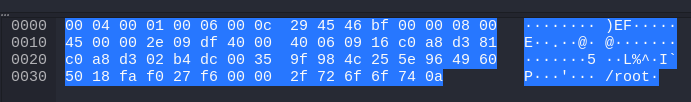
nc 192.168.211.2 53 -c cd root

nc 192.168.211.2 53 -c ls

Tenemos la respuesta:



nc 192.168.211.2 53 -c pwd



Después de esta demo necesitamos hacer un ls de /

nc 192.168.211.2 53 -c 'cd ..'

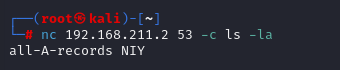
nc 192.168.211.2 53 -c ls

nc 192.168.211.2 53 -c pwd

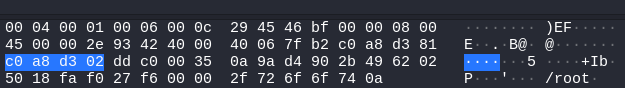
**NOTA: Si los puertos estuviesen cerrados hacemos:** nmap -A 192.168.211.2

Vamos a ver, hacemos

nc 192.168.211.2 53 -c ls -la



nc 192.168.211.2 53 -c pwd

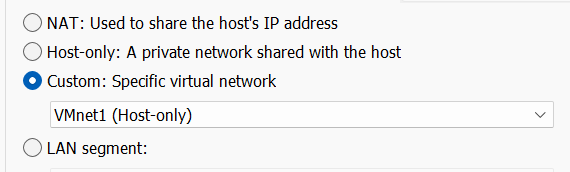


To see a list of which programs are listening on which TCP port:

sudo netstat -plnt

## Resolver Earth

**NOTA**: Antes de todo debemos crear una **red privada virtual** con host only tanto en nuestra maquina como ene l objetivo:



This CTF is coming in four stages:

1. Getting target information
2. Getting user password
3. Connecting to the target system
4. Getting root account

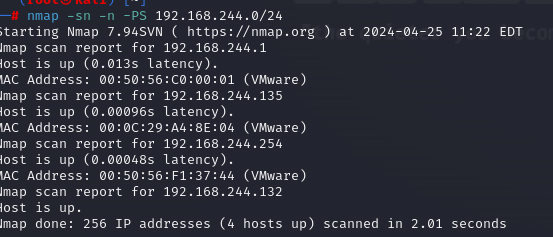
El objetivo de este juego es: There are two flags on the box: a user and root flag which include an md5 hash

* Para saber cual es nuestro host Hacemos primero un: Ip a

Tenemos nuestra maquina 192.168.244.132/

Luego escaneamos los host:

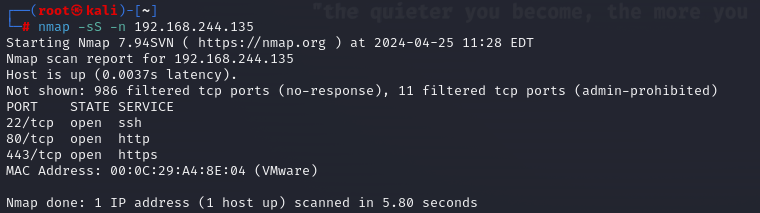
nmap -sn -n -PS 192.168.244.0/24



Luego nuestra maquina objetivo es la 192.168.244.135

Luego escaneamos puertos y servicios

nmap -sS -n 192.168.244.135



Y tenemos tres puertos tcp abiertos: ssh, http, https

Utilizamos el http y https con el alias, pero antes vamos a utilizar la herramienta:

**Dirb – Domain Brute-Force Tool**

**Dirb** es un escáner de contenido web, también conocido como una herramienta de fuerza bruta para el descubrimiento de ficheros y directorios existentes en un portal web.

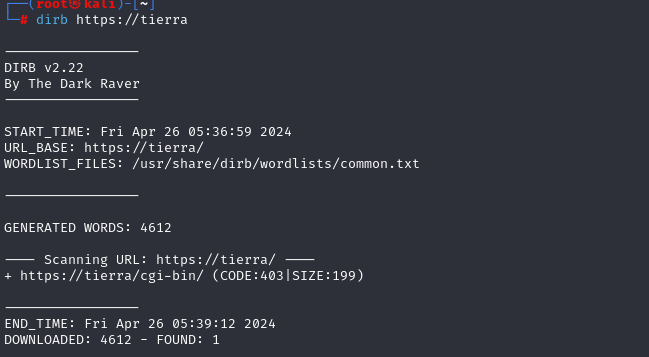
El objetivo de esta herramienta es tratar de descubrir rutas que el administrador ha dejado públicas de forma descuidada o simplemente para tratar de hacerse una idea de la estructura web del servidor.

Hacemos: dirb <http://earth.local>

O

<https://earth.local> para SSL

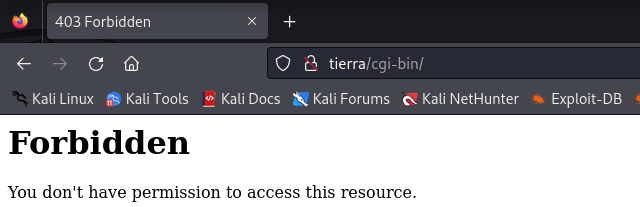
Y se tiene 1 resultado:



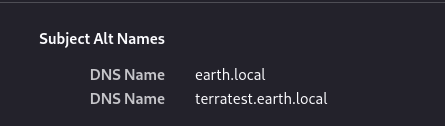
Hacemos ahora según esto:

<https://tierra/cgi-bin/>

Tenemos un error 403 forbidden



Vamos a ver el certificado SSL del sitio al que intentamos acceder



Añadimos estos dos nombres al /etc/hosts

earth.local

terratest.earth.local

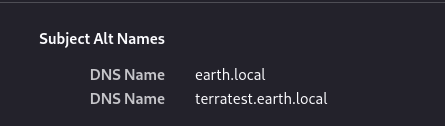
Nano /etc/hosts



Ahora ya podemos meter en el explorador direcciones

**NOTA**: En otros proyectos nos daban los nombres de los servidores, pero en este no nos lo dan, con lo cual sacamos sus nombres de la pantalla de certificados

asi nos saltamos la redirección automática de http a https que esta haciendo Firefox y nos quitamos el forbidden 403



Volvemos a hacer un:

dirb <http://earth.local>

dirb <http://terratest.earth.local/>

Y sacamos dos direcciones:

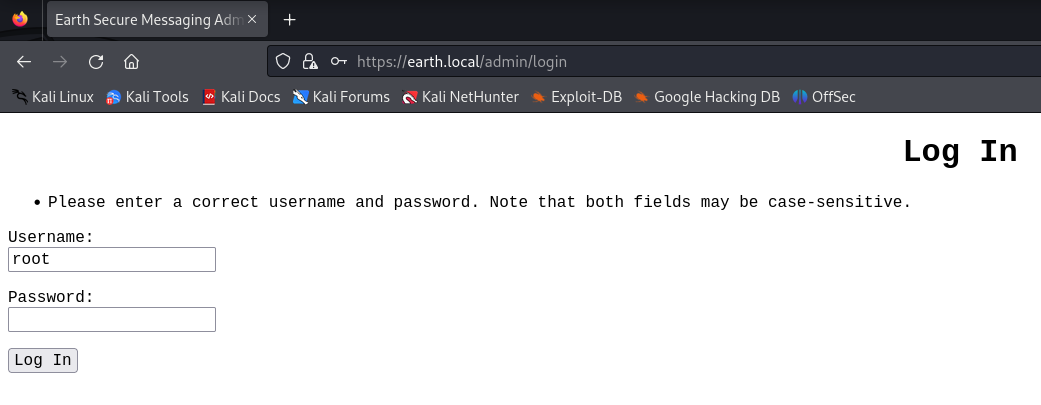
<http://earth.local/admin>

<http://earth.local/cgi-bin/>

<http://terratest.earth.local/admin>

<http://terratest.earth.local/cgi-bin/>

La primera de las direcciones nos manda a:



Hacemos ahora el escaneo dirb de los https:

dirb <https://earth.local>

dirb <https://terratest.earth.local/>

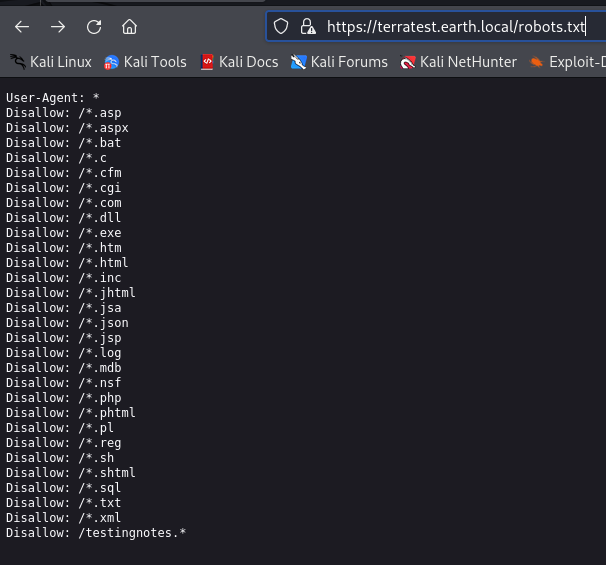
Y se tiene:

<https://terratest.earth.local/index.html>

<https://terratest.earth.local/robots.txt>

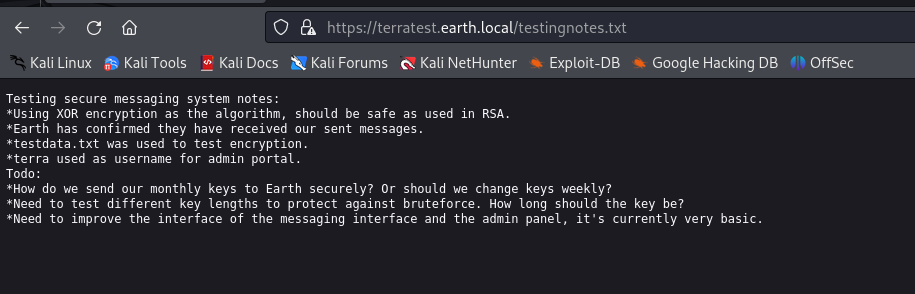
Para que sirve el archivo ‘robots.txt’. Un archivo robots.txt indica a los rastreadores de los buscadores a qué URLs de tu sitio pueden acceder. Se utiliza para evitar que las solicitudes que recibe tu sitio lo sobrecarguen; no es un mecanismo para impedir que una página web aparezca en Google. Si quieres que una página web no aparezca en Google, bloquea la indexación con noindex o protege la página con una contraseña.

Al visualizar robots.txt tenemos:



Si vemos la ultima línea hacemos:

https://terratest.earth.local/testingnotes.txt

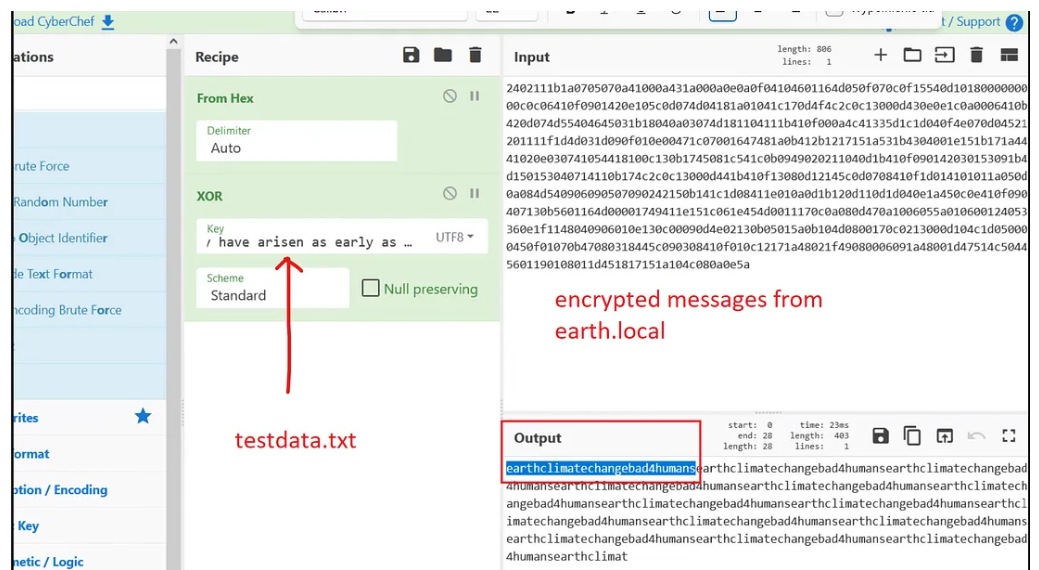


Se tiene que el usuario para entrar como Admin es: terra

Vemos como desencryptar una cadena XOR:

So lets go to CyberChef and see what the messages are all about!

<https://gchq.github.io/CyberChef/>



Hacemos nano ./testdata.txt -----

Esta es la key para meter en la web de traducción:

‘According to radiometric dating estimation and other evidence, Earth formed over 4.5 billion years ago. Within the first billion years of Earth's history, life appeared in the oceans and began to affect Earth's atmosphere and surface, leading to the proliferation of anaerobic and, later, aerobic organisms. Some geological evidence indicates that life may have arisen as early as 4.1 billion years ago.’

Le metemos la ultima línea de la web de eath de mensagges en input:

‘’

Y se tiene en output:

‘earthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclima’

Si tenemos una lista de password de usuarios nos puede servir para utilizar la herramienta de fuerza bruta para webs hydra.

**SSH (Secure Shell o Secure Socket Shell)** es un protocolo de red que permite una conexión segura a una computadora a través de una red no segura. Por defecto corre en el *puerto 22.*

Para conectarnos al servidor SSH debemos escribir en nuestro terminal el siguiente comando: ssh usuario@[IP]:[puerto]

Hacemos: ssh root@192.168.244.135

Pero claro, sin password de root no nos dejara acceder a la máquina

“como hackear un host linux con servicio ssh abierto”

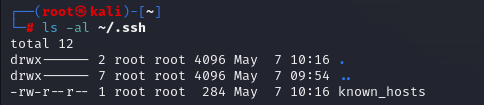
NOTA: ssh/known\_hosts es una base de datos conteniendo las máquinas a las que el usuario se ha conectado. Cualquier nuevo hosts es añadido al fichero del usuario

<https://campusvirtual.ull.es/ocw/pluginfile.php/2172/mod_resource/content/0/perlexamples/node27.html#:~:text=ssh%2Fknown_hosts%20una%20base%20de,el%20usuario%20se%20ha%20conectado>.

<https://docs.github.com/es/authentication/connecting-to-github-with-ssh/checking-for-existing-ssh-keys>

$ ls -al ~/.ssh

# Lists the files in your .ssh directory, if they exist



**Hashing**

Supongamos que el atacante ha descubierto nuestra clave y ha entrado en nuestro servidor: ahora puede hacer cat ~/.ssh/known\_hosts y deducir que otros servidores solemos visitar.

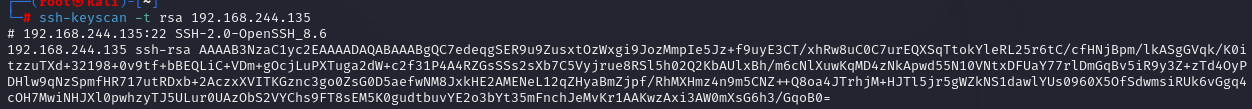
Podemos utilizar la utilidad **ssh-keyscan***:*

**La cual** recupera llaves públicas SSH

ssh-keyscan -t rsa <IP> -p <PORT>

Como tenemos abierto el puerto 22 del servicio ssh en la máquina 192.168.244.135, hacemos:

ssh-keyscan -t rsa 192.168.244.135



¿Cómo saber interpretar lo que responde?

Descifrar una clave rsa