Pràctica 3.- Anàlisi de paquets.

Objectiu i descripció general.

L'objectiu d'esta pràctica és l'anàlisi dels paquets que viatgen per la xarxa. L'alumne haurà d'analitzar l'estructura i definició de les trames capturades i comprovar l'equivalència amb les especificacions dels diferents protocols estudiats en teoria. Apunteu respostes i comentaris als exercicis plantejats per tal de poder recordar-ho posteriorment.

Desenvolupament de la pràctica.

1.- nmap.

La ferramenta *nmap* permet realitzar un escàner d'un ordinador o d'una xarxa per a determinar quins ordinadors estan operatius i quins serveis ofereixen. Les tècniques que pot utilitzar són molt variades, però ací ens centrarem en les més bàsiques.

1.1.- L'opció -sT determina els serveis TCP que ofereix un ordinador o xarxa. Executa *nmap -sT localhost* i determina quins serveis TCP ofereix el teu ordinador.
1.2.- Així mateix, executa *nmap -sT slabii.uv.es* i determina els serveis TCP que ofereix eixe ordinador.
1.3.- L'opció –O, que pot tardar a executar-se un parell de minuts, permet esbrinar el sistema operatiu que executa un ordinador remot. Executeu *nmap -O slabii.informat.uv.es* i indiqueu el sistema operatiu de l'ordinador.
1.4.- Executeu ara *nmap -O laetitia.irobot.uv.es* i indiqueu la resposta obtinguda.

2.- ethereal

Ethereal és una interfície gràfica de l'analitzador de protocols de xarxa tcpdump[1]. El programa pot funcionar en mode promiscu i capturar totes les trames de la xarxa o bé en mode no promiscu. Encara que posseeix moltes opcions d'inici, el més normal és executar-ho sense cap tipus d'opció i configurar-ho mentre s'usa.

La interfície d'usuari és bastant simple i intuïtiva. Per a capturar tràfic de xarxa, n'hi ha prou amb polsar *Capture* i indicar que capturarem per la interfície *Eth0* i posar sense marcar la casella de *Capture packets in promiscuous mode*. Apareixeran uns comptadors, i si executem alguns dels comandaments vistos amb anterioritat, podrem observar com augmenten. Quan es tinguen capturades algunes trames, detindre el procés de captura[2].

Analitzeu cada paquet simplement punxant en la finestra superior sobre cadascuna de les trames capturades. Justament en la finestra inferior s'observen amb detall l'estructura de les trames, des de MAC (Ethernet), IP i nivells superiors.

[1] Consulteu les pàgines del manual per a obtindre més informació de tcpdump.
[2] Una gran part del tràfic capturat correspon a tràfic broadcast, raó per la qual el nombre de paquets capturats pot arribar a ser, en un moment donat, bastant elevat.

Comproveu els continguts de la capçalera de la trama Ethernet, són els que s'han vist en teoria? En la finestra més inferior es troba la informació en hexadecimal, en la qual pot ser més senzill mirar els diferents camps.

Passeu ara a IP i analitzar la capçalera del protocol. Hi ha 20 bytes a analitzar. Comproveu tots ells i expliqueu que significa cadascú.

Torneu una altra vegada a capturar paquets i executeu el comandament *traceroute –t 16 slabii.informat.uv.es*. Expliqueu ara la resposta.

Executeu novament el mode de captura i, utilitzant ara el comandament *ping*, forçar l'enviament de missatges ICMP. Intenteu que els paquets no arriben al seu destí amb un TTL baix (opció *–t* del comandament *ping*). Analitzeu els resultats obtinguts.

Genereu ara amb el comandament *ping* paquets IP que s'hagen de fragmentar (opció *–s* del comandament *ping*) i intenteu predir tots els fragments que es produiran i la seua grandària. Ara, amb l'analitzador, observeu les trames Ethernet i comproveu si la predicció anà correctament. Observeu les capçaleres i analitzeu si la informació que contenen els camps de fragmentació coincideix amb la que esperàveu.

Realitzeu ara una connexió FTP a *glup.irobot.uv.es* tal i com s'ha vist amb anterioritat. Tanqueu després la connexió i analitzeu algunes de les trames rebudes.

Treballarem ara en mode promiscu. Per a això cal marcar la casella de *Capture packets in promiscuous mode,* començar la captura i, quan es tinguen un cert nombre de paquets, detindre-la. Comproveu que ara els paquets tenen adreces d'origen i destí diferents de la del nostre ordinador.

Com la captura produeix una gran quantitat de paquets, l'analitzador posseeix l'opció d'introduir filtres[3]. Crear filtres i comprovar el seu funcionament de manera que permeten:

- Capturar els paquets amb destí a *www.uv.es*.
- Capturar els paquets amb origen o destí el port 53 udp.
- Capturar tots els paquets IP de longitud menor de 200 bytes.
- Capturar tots els paquet el camp DF dels quals estiga a 1.

[3] En l'apèndix A podeu trobar una explicació dels filtres que teniu al vostre abast així com de la seua sintaxi.

Apèndix A: Filtres en ethereal.

En ethereal és possible construir filtres que determinen si un determinat paquet serà o no capturat. En el cas que no s'utilitze cap filtre, tots els paquets són capturats.

Els filtres es construeixen per mitjà d'expressions que consisteixen en una o més primitives. Les primitives, usualment, estan formades per un identificador (nom o número), precedides per un o més qualificadors. Hi ha tres tipus diferents de qualificadors:

- De tipus: Identifiquen un nom o adreça, els seus possibles valors són *host*, *net* i *port*. Per exemple, host glup.uv.es, net 147.156, port 20. Si no hi ha cap qualificador de tipus, s'assumeix que el tipus és host.

- D'adreça: Identifiquen una adreça particular de transferència, açò és, un origen o destí. Els seus valors possibles són *src*, *dst*, *src or dst* i *src and dst*. Per exemple, src glup.uv.es, dst net 147.156, src or dst port ftp-data. Si no s'indica cap qualificador d'adreça, es pren el qualificador d'adreça per defecte (src or dst).

- De protocol: Identifiquen un protocol particular. Els seus valors possibles són *ether, fddi, tr, ip, ip6, arp, rarp, decnet, tcp* i *udp*. Per exemple, ether src glup.uv.es, arp net 147.156, tcp port 21. Si no s'especifica cap protocol, tots els protocols que siguen consistents amb la identificació de tipus són capturats.

Com pot veure's per l'explicació dels diferents tipus de qualificadors, sempre estan presents, encara que siga per defecte, els tres tipus de qualificadors. Així, l'expressió ip 147.156.222.65 és equivalent a ip src or dst host 147.156.222.65. [4]

Poden construir-se filtres més complexos gràcies a la combinació de primitives per mitjà de la utilització de parèntesi i/o les paraules *and*, *or* i *not*., sent la prioritat de *not* major que la d'*and* i *or*, la prioritat de la qual entre si és igual. Així, per exemple, host glup.uv.es and not port ftp indica que es capturen tots els paquets l'origen o el destí dels quals siga glup.uv.es excepte aquells el port d'origen o destí dels quals és el de ftp (port 21).

Un llistat de les primitives més utilitzades es troba en la taula següent:

Primitiva	**Descripció**
dst host <ordinador>	Veritat si el camp destí del paquet és l'<ordinador>
src host <ordinador>	Veritat si el camp origen del paquet és l'<ordinador>
host <ordinador>	Veritat si el camp origen o destí del paquet és l'<ordinador>
ether dst <ordinador>	Veritat si l'adreça ethernet de destí és l'<ordinador>
ether src <ordinador>	Veritat si l'adreça ethernet d'origen és l'<ordinador>

[4] Si es desitja especificar en cap moment una adreça ethernet en el filtre, es deu fer amb el format XX:XX:XX:XX:XX:XX.

ether host <ordinador>	Veritat si les adreces ethernet d'origen o destí són l'<ordinador>
gateway <ordinador>	Veritat si el paquet utilitza com a passarel·la (gateway) l'<ordinador>
dst net <xarxa>	Veritat si l'adreça de destí del paquet correspon a una adreça de la <xarxa>
src net <xarxa>	Veritat si l'adreça d'origen del paquet correspon a una adreça de la <xarxa>
net <xarxa>	Veritat si les adreces d'origen o destí del paquet corresponen a una adreça de la <xarxa>
dst net <xarxa> mask <màscara>	Veritat si l'adreça de destí del paquet correspon a una adreça de la <xarxa> de màscara <màscara>
src net <xarxa> mask <màscara>	Veritat si l'adreça d'origen del paquet correspon a una adreça de la <xarxa> de màscara <màscara>
net <xarxa> mask <màscara>	Veritat si les adreces d'origen o destí del paquet corresponen a adreces de la <xarxa> de màscara <màscara>
dst net <xarxa>/<longitud>	Veritat si l'adreça de destí del paquet correspon a una adreça de la <xarxa> la màscara de la qual s'indica per <longitud>
src net <xarxa>/<longitud>	Veritat si l'adreça d'origen del paquet correspon a una adreça de la <xarxa> la màscara de la qual s'indica per <longitud>
net <xarxa>/<longitud>	Veritat si les adreces d'origen o destí del paquet corresponen a adreces de la <xarxa> la màscara de la qual s'indica per <longitud>
dst port <port>	Veritat si el paquet té com a destí el port <port>
src port <port>	Veritat si el paquet té com a origen el port <port>
port <port>	Veritat si el paquet té com a origen o destí el port <port>
less <longitud>	Veritat si el paquet té una longitud menor o igual que <longitud>
greater <longitud>	Veritat si el paquet té una longitud major o igual que <longitud>
ether broadcast	Veritat si el paquet és un paquet ethernet broadcast.
ip broadcast	Veritat si el paquet és un paquet IP broadcast.
ether multicast	Veritat si el paquet és un paquet ethernet multicast.
ip multicast	Veritat si el paquet és un paquet IP multicast.

A més de les expressions anteriors, hi ha expressions del tipus *<expressió 1> <operador> <expressió 2>*, on *<operador>* és <, >, <=, >=, =, *!=* i *<expressió 1>* i *<expressió 2>* són expressions aritmètiques compostes per constants senceres (expressades amb la sintaxi de C), els operadors +, -, *, /, &, |, i un accés especial a les dades del paquet.

Per a accedir a les dades d'un paquet s'utilitza la sintaxi *protocol [desplaçament : grandària]*, on *protocol* és un dels protocols favorits (*ether*, *fddi*, *tr*, *ip*, *arp*, *rarp*, *tcp*, *udp*, *icmp* o *ip6*), *desplaçament* és el desplaçament, en bytes, des del començament de les dades

del protocol especificat, i *grandària* són els bytes a analitzar. Així, ip[0] & 0x0F != 5 indica tots els paquets que contenen opcions IP (camp IHL de valor diferent de 5), mentre que ip[6 : 2] & 0x1FFF = 0 indica només datagrames no fragmentats o el primer fragment dels datagrames fragmentats.

Práctica 3.- Análisis de paquetes.

Objetivo y descripción general.

El objetivo de esta práctica es el análisis de los paquetes que viajan por la red. El alumna deberá analizar la estructura y definición de las tramas capturadas así como también comprobar la equivalencia con las especificaciones de los diferentes protocolos estudiados en teoría. Apuntad las respuestas y comentarios sobre los ejercicios planteados para su estudio posterior.

Desarrollo de la práctica.

1.- nmap.

La herramienta *nmap* permite realizar un escáner de un ordenador o una red para determinar que ordenadores están operativos y que servicios ofrecen. Las técnicas que puede utilizar son muy variadas, pero aquí nos centraremos en las más básicas.

1.1.- La opción -sT determina los servicios TCP que ofrece un ordenador o red. Ejecuta *nmap -sT localhost* y determina que servicios TCP ofrece tu ordenador.
1.2.- De igual forma, ejecuta *nmap -sT slabii.informat.uv.es* y determina los servicios TCP que ofrece ese ordenador.
1.3.- La opción –O, que puede tardar en ejecutarse un par de minutos, permite averiguar el sistema operativo que ejecuta un ordenador remoto. Ejecutar *nmap -O slabii.informat.uv.es* e indicar el sistema operativo del ordenador.
1.4.- Ejecutar *nmap -O laetitia.irobot.uv.es* e indicar la respuesta obtenida.

2.- ethereal

Ethereal es un interfaz grafico del analizador de protocolos de red tcpdump[5]. El programa puede funcionar en modo promiscuo y capturar todas las tramas de la red o bien en modo no promiscuo. Aunque posee muchas opciones de inicio, lo más normal es ejecutarlo sin ningún tipo de opción y configurarlo durante su uso.

El interfaz de usuario es bastante simple e intuitivo. Para capturar trafico de red, basta con pulsar *Capture* e indicar que vamos a capturar por la interfaz *eth0* y poner sin marcar la casilla de *Capture packets in promiscuous mode*. Aparecerán unos contadores, y si ejecutamos algunos de los comandos vistos con anterioridad, podremos observar que varían dichos contadores. Cuando se tengan capturadas algunas tramas, detener el proceso de captura[6].

Analizar cada paquete simplemente pinchando en la ventana superior cada una de

[5] Consultar las páginas de manual para más información sobre tcpdump.
[6] Una gran parte del tráfico capturado corresponde a tráfico broadcast, por lo que en ciertos momentos el número de paquetes capturado puede ser elevado.

las tramas capturadas. Justamente en la ventana inferior se observan con detalle la estructura de las tramas, desde MAC (Ethernet), IP y niveles superiores.

Comprobar los contenidos de la cabecera de la trama Ethernet, ¿son los que se han visto en teoría? En la ventana más inferior se encuentra la información en hexadecimal, en la cual puede ser más sencillo mirar los diferentes campos.

Pasar ahora a IP y analizar la cabecera de dicho protocolo. Hay 20 bytes a analizar. Comprobar todos ellos y explicar que significa cada uno.

Volver otra vez a capturar paquetes y ejecutar el comando *traceroute –t 16 slabii.informat.uv.es*, explicando la respuesta.

Ejecutar nuevamente el modo de captura y, utilizando ahora el comando *ping*, forzar el envío de mensajes ICMP. Intentar que los paquetes no lleguen a su destino con un TTL bajo (opción *–t* del comando *ping*). Analizar los resultados obtenidos.

Generar ahora con el comando *ping* paquetes IP que se hayan de fragmentar (opción *–s* del comando *ping*) e intentar predecir cuantos fragmentos se producirán y de que tamaño. Ahora, con el analizador, observar las tramas Ethernet y comprobar si fue correcta la predicción. Observar las cabeceras y analizar si la información que contienen los campos de fragmentación coincide con la que era de esperar.

Realizar ahora una conexión FTP a glup.irobot.uv.es tal y como se ha visto con anterioridad. Cerrar dicha conexión y analizar algunas de las tramas recibidas.

Vamos a proceder ahora a trabajar en modo promiscuo. Para ello, marcar la casilla de *Capture packets in promiscuous mode*. Empezar la captura y cuando se tengan un cierto número de paquetes detener la misma. Comprobar que ahora los paquetes tienen direcciones de origen y destino distintas a la de nuestro ordenador.

Como la captura produce una gran cantidad de paquetes, el analizador posee la opción de introducir filtros[7]. Crear filtros y comprobar su funcionamiento de forma que permitan:

- Capturar los paquetes con destino a www.uv.es.
- Capturar los paquetes con origen o destino el puerto 53 udp.
- Capturar todos los paquetes ip de longitud menor de 200 bytes.
- Capturar todos los paquete cuyo campo DF esté a 1.

[7] En el apéndice A se encuentra una explicación de los filtros y su sintaxis.

Apéndice A: Filtros en ethereal.

En ethereal es posible construir filtros que determinen si un determinado paquete van a ser o no capturado. En caso de que no se utilice ningún filtro, todos los paquetes son capturados.

Los filtros se construyen mediante expresiones que consisten en una o más primitivas. Las primitivas, usualmente, consisten en un identificador (nombre o número), precedidas por uno o más calificadores. Existen tres tipos diferentes de calificadores:

- De tipo: Identifican un nombre o dirección, sus posibles valores son *host*, *net* y *port*. Por ejemplo, host glup.uv.es, net 147.156, port 20. Si no existe ningún calificador de tipo, se asume que el tipo es host.

- De dirección: Identifican una dirección particular de transferencia, esto es, un origen o destino. Sus valores posibles son *src*, *dst*, *src or dst* y *src and dst*. Por ejemplo, src glup.uv.es, dst net 147.156, src or dst port ftp-data. Si no se indica ningún calificador de dirección, se toma el calificador de dirección por defecto (src or dst).

- De protocolo: Identifican un protocolo particular. Sus valores posibles son ether, fddi, tr, ip, ip6, arp, rarp, decnet, tcp y udp. Por ejemplo, ether src glup.uv.es, arp net 147.156, tcp port 21. Si no se especifica ningún protocolo, todos los protocolos que sean consistentes con la identificación de tipo son capturados.

Como puede verse por la explicación de los diferentes tipos de calificadores, siempre están presentes, aunque sea por defecto, los tres tipos de calificadores. Así, la expresión ip 147.156.222.65 es equivalente a ip src or dst host 147.156.222.65.[8]

Pueden construirse filtros más complejos mediante la combinación de primitivas mediante la utilización de paréntesis y/o las palabras *and*, *or* y *not*., siendo la prioridad de *not* mayor que la de *and* y *or*, cuya prioridad entre si es igual. Así, por ejemplo, host glup.uv.es and not port ftp indica que se capturen todos los paquetes cuyo origen o destino es glup.uv.es excepto aquellos cuyo puerto de origen o destino es el de ftp (puerto 21).

Un listado de las primitivas más utilizadas se encuentra en la siguiente tabla:

Primitiva	**Descripción**
dst host <ordenador>	Verdad si el campo destino del paquete es el <ordenador>
src host <ordenador>	Verdad si el campo origen del paquete es el <ordenador>
host <ordenador>	Verdad si el campo origen o destino del paquete es el <ordenador>

[8] Si en algún momento se desea especificar una dirección ethernet en el filtro, esta debe especificarse como XX:XX:XX:XX:XX:XX.

Primitiva	Descripción
ether dst <ordenador>	Verdad si la dirección ethernet de destino es el <ordenador>
ether src <ordenador>	Verdad si la dirección ethernet de origen es el <ordenador>
ether host <ordenador>	Verdad si la dirección ethernet de origen o destino es el <ordenador>
gateway <ordenador>	Verdad si el paquete utiliza como pasarela (gateway) el <ordenador>
dst net <red>	Verdad si la dirección de destino del paquete corresponde a una dirección de la <red>
src net <red>	Verdad si la dirección de origen del paquete corresponde a una dirección de la <red>
net <red>	Verdad si las dirección de origen o destino del paquete corresponde a una dirección de la <red>
dst net <red> mask <mascara>	Verdad si la dirección de destino del paquete corresponde a una dirección de la <red> de máscara <mascara>
src net <red> mask <mascara>	Verdad si la dirección de origen del paquete corresponde a una dirección de la <red> de máscara <mascara>
net <red> mask <mascara>	Verdad si las dirección de origen o destino del paquete corresponde a una dirección de la <red> de máscara <mascara>
dst net <red>/<longitud>	Verdad si la dirección de destino del paquete corresponde a una dirección de la <red> cuya máscara se indica por <longitud>
src net <red>/<longitud>	Verdad si la dirección de origen del paquete corresponde a una dirección de la <red> cuya máscara se indica por <longitud>
net <red>/<longitud>	Verdad si las dirección de origen o destino del paquete corresponde a una dirección de la <red> cuya máscara se indica por <longitud>
dst port <puerto>[9]	Verdad si el paquete tiene como destino el puerto dado por <puerto>
src port <puerto>	Verdad si el paquete tiene como origen el puerto dado por <puerto>
port <puerto>	Verdad si el paquete tiene como origen o destino el puerto dado por <puerto>
less <longitud>	Verdad si el paquete tiene una longitud menor o igual que <longitud>
greater <longitud>	Verdad si el paquete tiene una longitud mayor o igual que <longitud>
ether broadcast	Verdad si el paquete es un paquete ethernet broadcast.
ip broadcast	Verdad si el paquete es un paquete IP broadcast.

[9] Esta expresión y las dos siguientes pueden ir precedidas de tcp o udp, para indicar que solo se desea el puerto correspondiente al protocolo tcp o udp.

Primitiva	Descripción
ether multicast	Verdad si el paquete es un paquete ethernet multicast.
ip multicast	Verdad si el paquete es un paquete IP multicast.

Además de las expresiones anteriores, existen expresiones del tipo <*expresión 1*> <*operador*> <*expresión 2*>, donde <*operador*> es <, >, <=, >=, =, *!=* y <*expresión 1*> y <*expresión 2*> son expresiones aritméticas compuestas por constantes enteras (expresadas con la sintaxis de C), los operadores +, -, *, /, &, |, y un acceso especial a los datos del paquete.

Para acceder a los datos de un paquete se utiliza la sintaxis *protocolo [desplazamiento : tamaño]*, donde *protocolo* es uno de los protocolos validos (*ether*, *fddi*, *tr*, *ip*, *arp*, *rarp*, *tcp*, *udp*, *icmp* o *ip6*), *desplazamiento* es el desplazamiento, en bytes, desde el comienzo de los datos del protocolo especificado, y *tamaño* son los bytes a analizar. Así, ip[0] & 0x0F != 5 indica todos los paquetes que contienen opciones IP (campo IHL de valor distinto de 5), mientras que ip[6 : 2] & 0x1FFF = 0 indica solo datagramas no fragmentados o el primer fragmento de los datagramas fragmentados.