Network Traffic Analysis

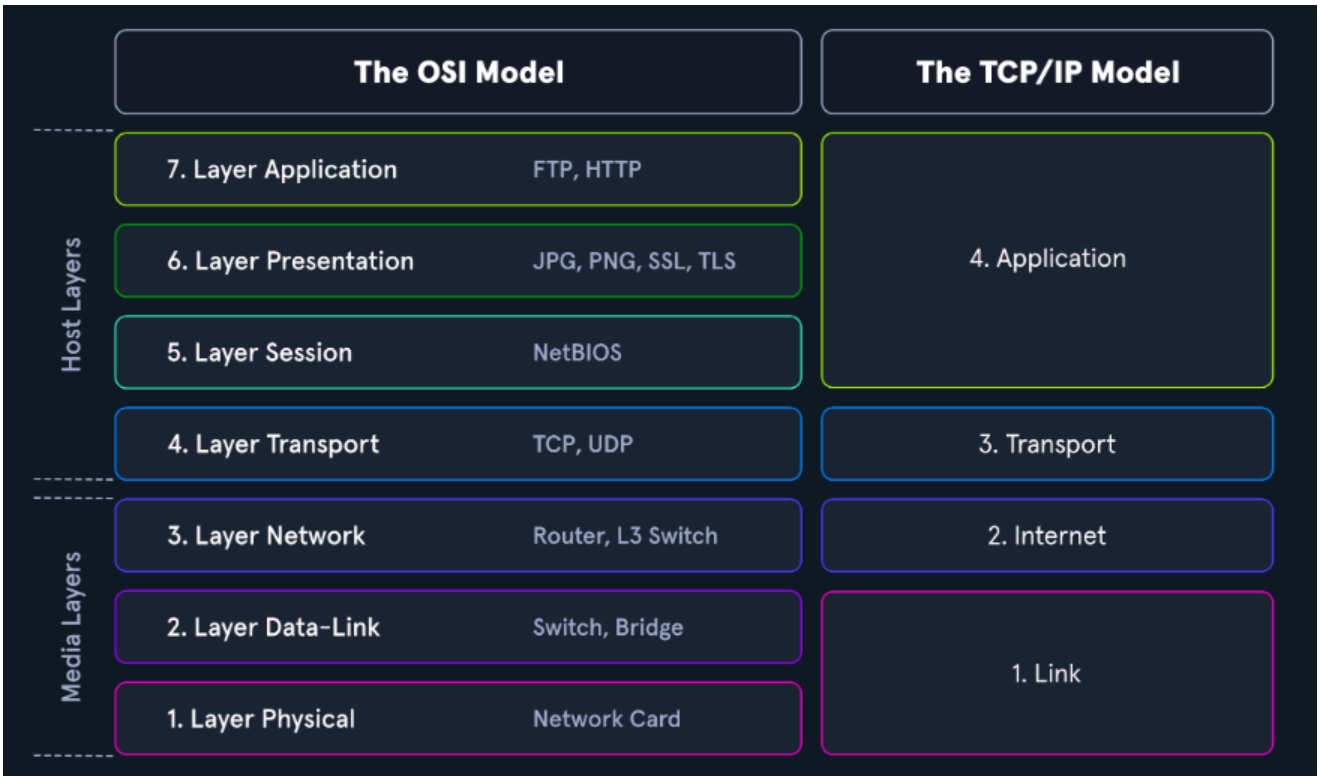
Network Traffic Analysis (NTA) se puede describir como el acto de examinar el tráfico de red para personalizar los puertos y protocolos comunes utilizados, establecer una línea de base para nuestro entorno, monitorear y responder a las amenazas y garantizar la mayor comprensión posible de la red de nuestra organización.

Por ejemplo, si detectamos muchos paquetes **SYN** en puertos que nunca (o rara vez) utilizamos en nuestra red, podemos concluir que probablemente se trate de alguien que intenta determinar qué puertos están abiertos en nuestros hosts. Acciones como esta son indicadores típicos de un **escaneo de puertos**.

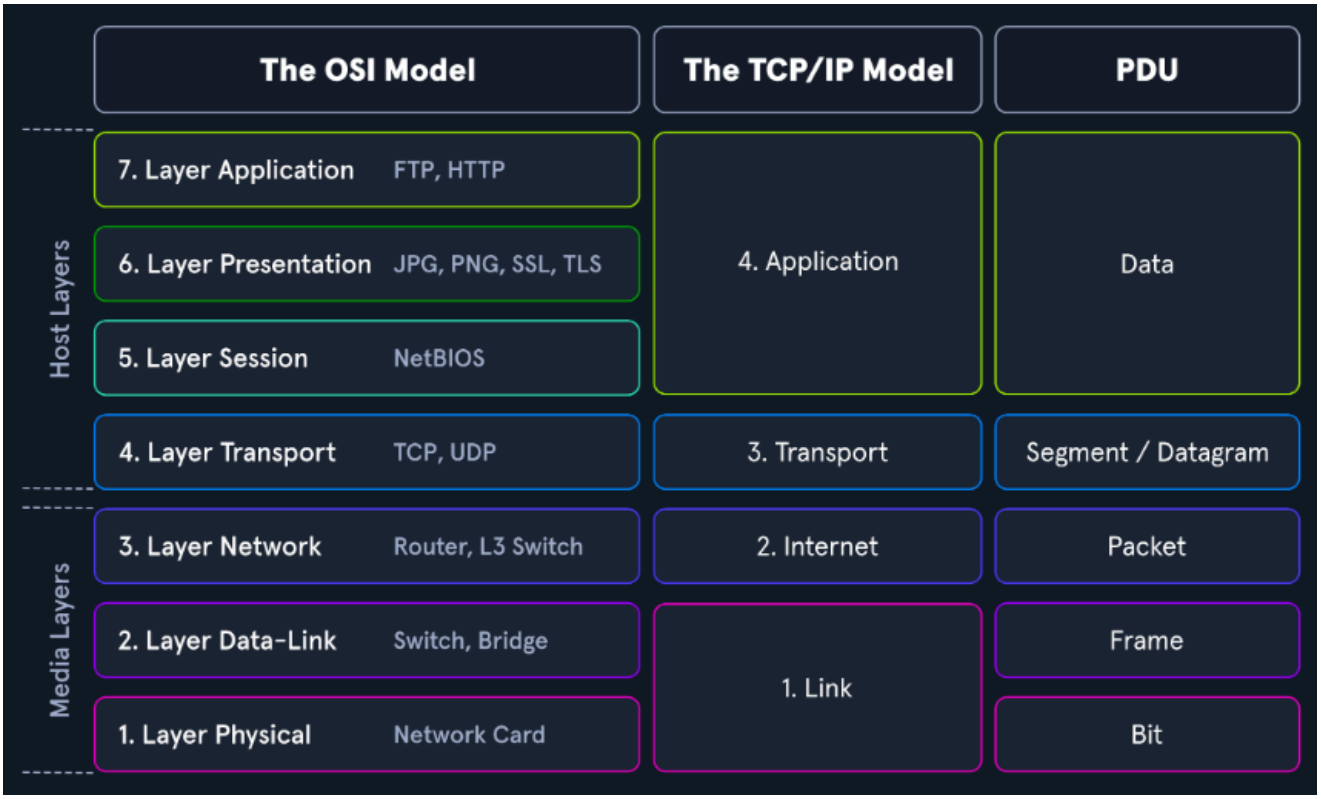
Algunas de las herramientas que se suelen utilizar en el ámbito de análisis de redes son:

Tool	Description
tcpdump	tcpdump is a command-line utility that, with the aid of LibPcap, captures and interprets network traffic from a network interface or capture file.
Tshark	TShark is a network packet analyzer much like TCPDump. It will capture packets from a live network or read and decode from a file. It is the command-line variant of Wireshark.
Wireshark	Wireshark is a graphical network traffic analyzer. It captures and decodes frames off the wire and allows for an in-depth look into the environment. It can run many different dissectors against the traffic to characterize the protocols and applications and provide insight into what is happening.
NGrep	NGrep is a pattern-matching tool built to serve a similar function as grep for Linux distributions. The big difference is that it works with network traffic packets. NGrep understands how to read live traffic or traffic from a PCAP file and utilize regex expressions and BPF syntax. This tool shines best when used to debug traffic from protocols like HTTP and FTP.
tcpick	tcpick is a command-line packet sniffer that specializes in tracking and reassembling TCP streams. The functionality to read a stream and reassemble it back to a file with tcpick is excellent.
Network Taps	Taps (Gigamon, Niagra-taps) are devices capable of taking copies of network traffic and sending them to another place for analysis. These can be in-line or out of band. They can actively capture and analyze the traffic directly or passively by putting the original packet back on the wire as if nothing had changed.
Networking Span Ports	Span Ports are a way to copy frames from layer two or three networking devices during egress or ingress processing and send them to a collection point. Often a port is mirrored to send those copies to a log server.
Elastic Stack	The Elastic Stack is a culmination of tools that can take data from many sources, ingest the data, and visualize it, to enable searching and analysis of it.
SIEMS	SIEMS (such as Splunk) are a central point in which data is analyzed and visualized. Alerting, forensic analysis, and day-to-day checks against the traffic are all use cases for a SIEM.

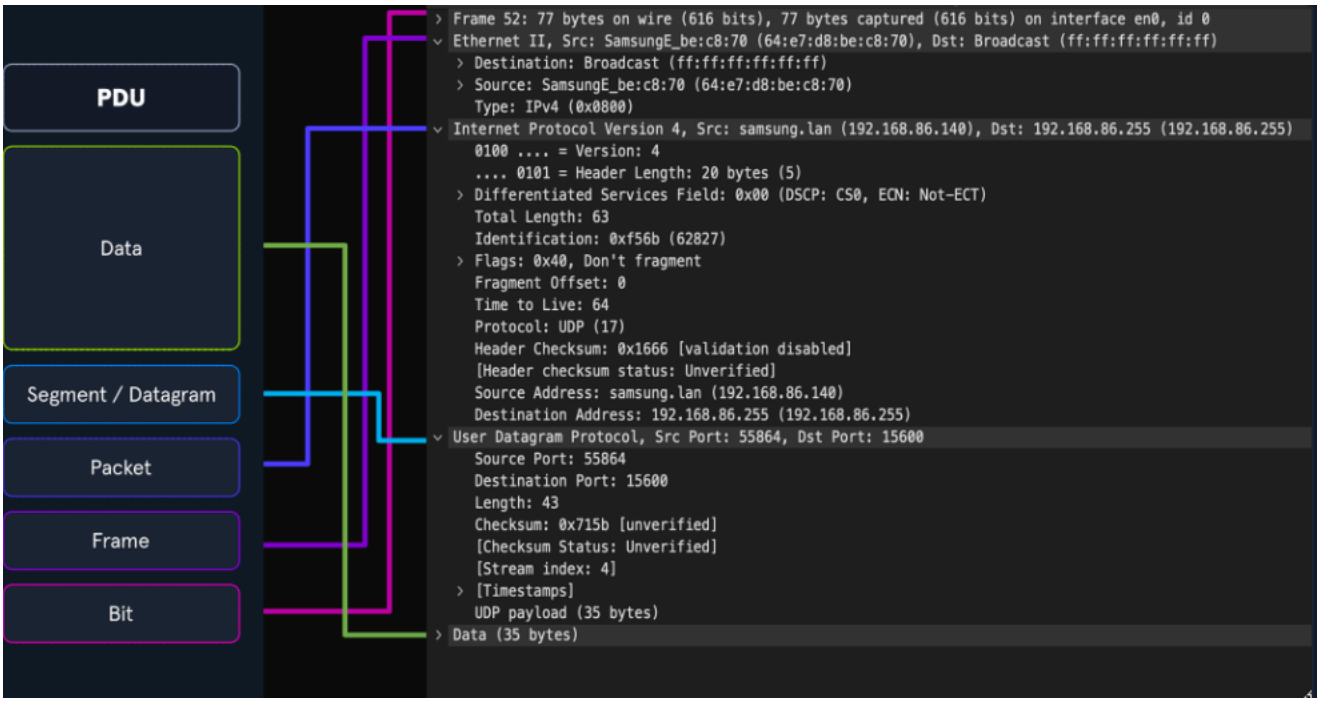
Networking Primer - Layers 1-4



Para entenderlo de una manera más aplicada y sencilla usaremos el Protocol Data Units (PDU):



Como se sitúa en Wireshark las capas de la red:



MAC-Addressing

El direccionamiento MAC se utiliza en las comunicaciones de capa dos (capa de enlace de datos o capa de enlace, según el modelo utilizado) entre hosts. Esto funciona mediante la comunicación de host a host dentro de un dominio de difusión. Si el tráfico de capa dos necesita cruzar una interfaz de capa tres, esa PDU se envía a la interfaz de salida de capa tres y se enruta a la red correcta.

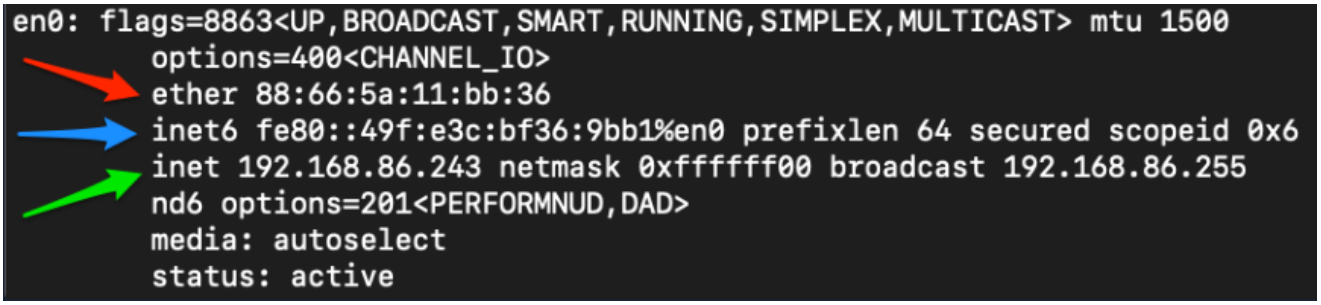
```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=400<CHANNEL_IO>
        ether 88:66:5a:11:bb:36
        inet6 fe80::49f:e3c:bf36:9bb1%en0 prefixlen 64 secured scopeid 0x6
        inet 192.168.86.243 netmask 0xffffff00 broadcast 192.168.86.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

IP Addressing

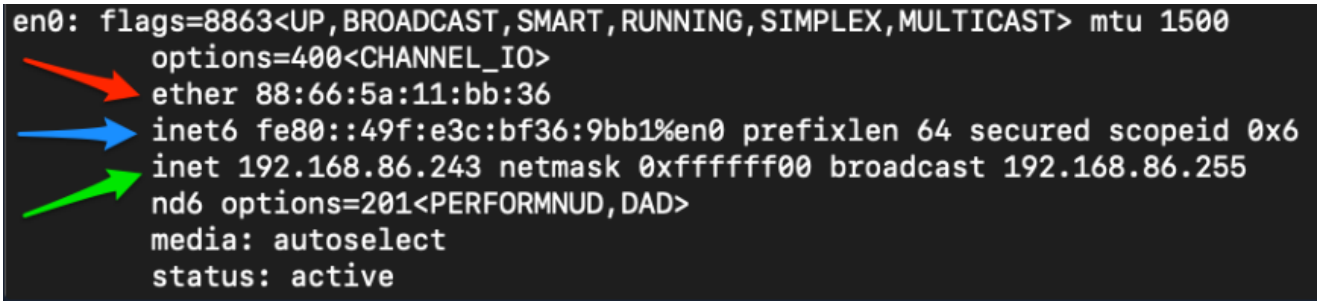
El Protocolo de Internet (IP) se desarrolló para transmitir datos de un host a otro a través de los límites de la red. El protocolo IP se encarga del enrutamiento de paquetes, la encapsulación de datos y la fragmentación y reensamblado de datagramas al llegar al host de destino.

Una dirección IPv4 se compone de un número de 32 bits y cuatro octetos, representado en formato decimal. En nuestro ejemplo, vemos la dirección 192.168.86.243. Cada octeto de una dirección IP puede representarse mediante un número del 0 al 255. Al examinar una PDU, encontramos direcciones IP en la capa

tres (Red) del modelo OSI y en la capa dos (Internet) del modelo TCP-IPm podemos ver la dirección ipv4 con la flecha verde.



IPv6 nos proporciona un espacio de direcciones mucho mayor que puede utilizarse para cualquier propósito de red. IPv6 es una dirección de 128 bits y 16 octetos representados en formato hexadecimal. Podemos ver un ejemplo de una dirección IPv6 abreviada en la imagen a continuación, con la flecha azul.



TCP / UDP, Transport Mechanisms

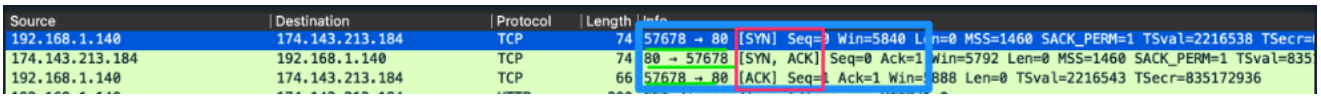
Characteristic	TCP	UDP
Transmission	Connection-oriented	Connectionless. Fire and forget.
Connection Establishment	TCP uses a three-way handshake to ensure that a connection is established.	UDP does not ensure the destination is listening.
Data Delivery	Stream-based conversations	packet by packet, the source does not care if the destination is active
Receipt of data	Sequence and Acknowledgement numbers are utilized to account for data.	UDP does not care.
Speed	TCP has more overhead and is slower because of its built-in functions.	UDP is fast but unreliable.

TCP Three-way Handshake

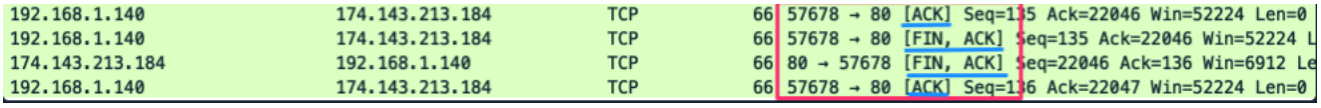
1. El cliente envía un paquete con el indicador SYN activado junto con otras opciones negociables en el encabezado TCP.
2. El servidor responderá con un paquete TCP que incluye un indicador SYN establecido para la negociación del número de secuencia y un indicador ACK

establecido para reconocer el paquete SYN anterior enviado por el host.

3. El cliente responderá con un paquete TCP con un indicador ACK establecido aceptando la negociación.



TCP Session Teardown



El servidor responde con un acuse de recibo del FIN y envía su propio FIN. Finalmente, el cliente confirma la finalización de la sesión y cierra la conexión.

Networking Primer - Layers 5-7

HTTP

El Protocolo de transferencia de hipertexto (HTTP) es un protocolo de capa de aplicación sin estado que se aloja en el puerto 80. HTTP permite la transferencia de datos en texto claro entre un cliente y un servidor a través de TCP.

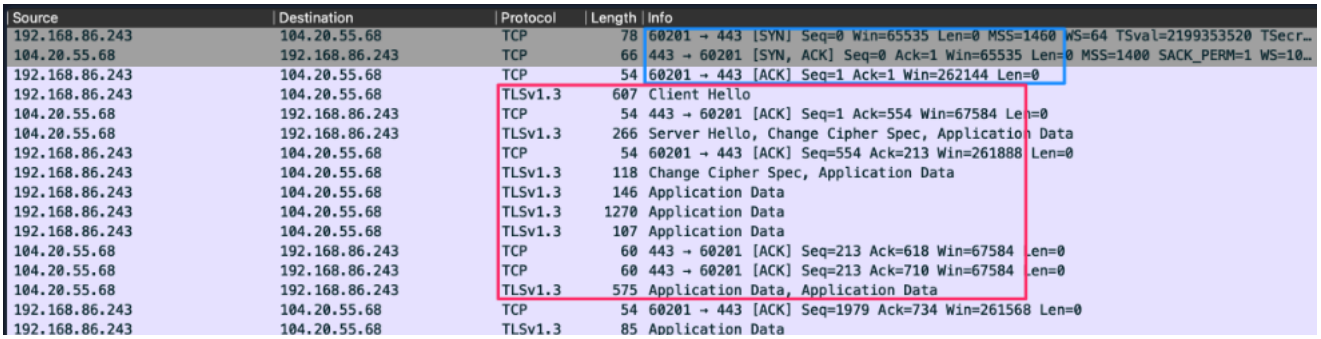
HTTP Methods

Method	Description
HEAD	`required` is a safe method that requests a response from the server similar to a Get request except that the message body is not included. It is a great way to acquire more information about the server and its operational status.
GET	`required` Get is the most common method used. It requests information and content from the server. For example, `GET http://10.1.1.1/Webserver/index.html` requests the index.html page from the server based on our supplied URI.
POST	`optional` Post is a way to submit information to a server based on the fields in the request. For example, submitting a message to a Facebook post or website forum is a POST action. The actual action taken can vary based on the server, and we should pay attention to the response codes sent back to validate the action.
PUT	`optional` Put will take the data appended to the message and place it under the requested URI. If an item does not exist there already, it will create one with the supplied data. If an object already exists, the new PUT will be considered the most up-to-date, and the object will be modified to match. The easiest way to visualize the differences between PUT and POST is to think of it like this; PUT will create or update an object at the URI supplied, while POST will create child entities at the provided URI. The action taken can be compared with the difference between creating a new file vs. writing comments about that file on the same page.
DELETE	`optional` Delete does as the name implies. It will remove the object at the given URI.
TRACE	`optional` Allows for remote server diagnosis. The remote server will echo the same request that was sent in its response if the TRACE method is enabled.
OPTIONS	`optional` The Options method can gather information on the supported HTTP methods the server recognizes. This way, we can determine the requirements for interacting with a specific resource or server without actually requesting data or objects from it.
CONNECT	`optional` Connect is reserved for use with Proxies or other security devices like firewalls. Connect allows for tunneling over HTTP. (`SSL tunnels`)

HTTPS

HTTP Secure (HTTPS) es una modificación del protocolo HTTP diseñada para utilizar la Seguridad de la Capa de Transporte (TLS) o la Capa de Sockets Seguros (SSL) en aplicaciones más antiguas para la seguridad de datos. TLS se utiliza como mecanismo de cifrado para proteger las comunicaciones entre un cliente y un servidor.

TLS Handshake Via HTTPS



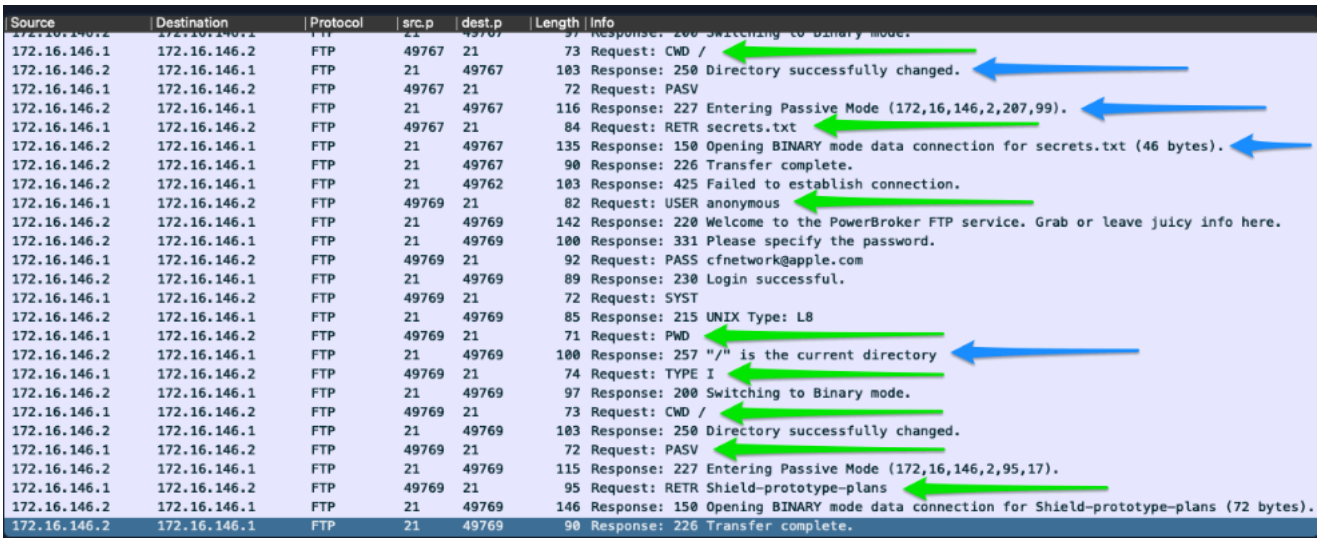
Source	Destination	Protocol	Length	Info
192.168.86.243	104.20.55.68	TCP	78	60201 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2199353520 TSecr…
104.20.55.68	192.168.86.243	TCP	66	443 → 60201 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM=1 WS=10…
192.168.86.243	104.20.55.68	TCP	54	60201 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
192.168.86.243	104.20.55.68	TLSv1.3	607	Client Hello
104.20.55.68	192.168.86.243	TCP	54	443 → 60201 [ACK] Seq=1 Ack=554 Win=67584 Len=0
104.20.55.68	192.168.86.243	TLSv1.3	266	Server Hello, Change Cipher Spec, Application Data
192.168.86.243	104.20.55.68	TCP	54	60201 → 443 [ACK] Seq=554 Ack=213 Win=261888 Len=0
192.168.86.243	104.20.55.68	TLSv1.3	118	Change Cipher Spec, Application Data
192.168.86.243	104.20.55.68	TLSv1.3	146	Application Data
192.168.86.243	104.20.55.68	TLSv1.3	1270	Application Data
192.168.86.243	104.20.55.68	TLSv1.3	107	Application Data
104.20.55.68	192.168.86.243	TCP	60	443 → 60201 [ACK] Seq=213 Ack=618 Win=67584 Len=0
104.20.55.68	192.168.86.243	TCP	60	443 → 60201 [ACK] Seq=213 Ack=710 Win=67584 Len=0
104.20.55.68	192.168.86.243	TLSv1.3	575	Application Data, Application Data
192.168.86.243	104.20.55.68	TCP	54	60201 → 443 [ACK] Seq=1979 Ack=734 Win=261568 Len=0
192.168.86.243	104.20.55.68	TLSv1.3	85	Application Data

FTP

El Protocolo de Transferencia de Archivos (FTP) es un protocolo de capa de aplicación que permite la transferencia rápida de datos entre dispositivos informáticos. FTP se puede utilizar desde la línea de comandos, un navegador web o un cliente FTP gráfico como FileZilla, FTP utiliza los puertos 20 y 21.

FTP puede funcionar en dos modos: activo y pasivo. El modo activo es el predeterminado, lo que significa que el servidor escucha el comando de control PORT del cliente, indicando el puerto a utilizar para la transferencia de datos. El modo pasivo nos permite acceder a servidores FTP ubicados detrás de firewalls o un enlace habilitado para NAT que hace imposibles las conexiones TCP directas.

FTP Command & Response Examples



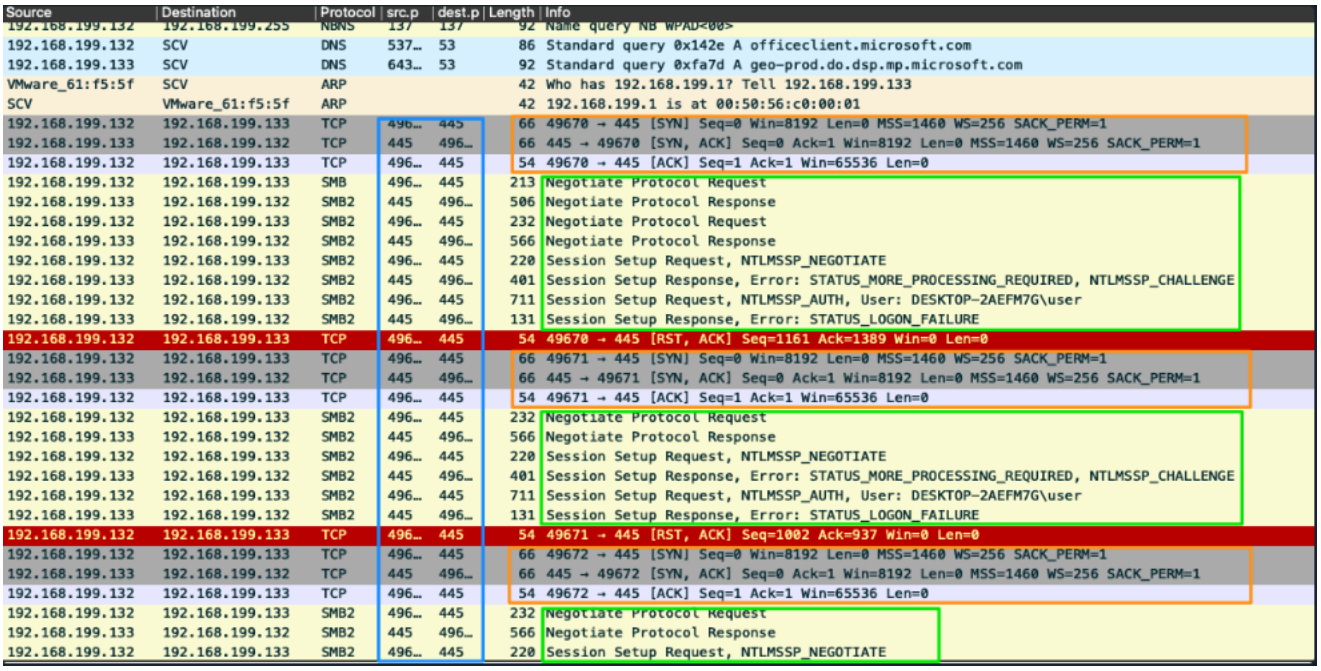
Source	Destination	Protocol	src.p	dest.p	Length	Info
172.16.146.1	172.16.146.2	FTP	49767	21	73	Request: CWD /
172.16.146.2	172.16.146.1	FTP	21	49767	103	Response: 250 Directory successfully changed.
172.16.146.1	172.16.146.2	FTP	49767	21	72	Request: PASV
172.16.146.2	172.16.146.1	FTP	21	49767	116	Response: 227 Entering Passive Mode (172,16,146,2,207,99).
172.16.146.1	172.16.146.2	FTP	49767	21	84	Request: RETR secrets.txt
172.16.146.2	172.16.146.1	FTP	21	49767	135	Response: 150 Opening BINARY mode data connection for secrets.txt (46 bytes).
172.16.146.2	172.16.146.1	FTP	21	49767	90	Response: 226 Transfer complete.
172.16.146.2	172.16.146.1	FTP	21	49762	103	Response: 425 Failed to establish connection.
172.16.146.1	172.16.146.2	FTP	49769	21	82	Request: USER anonymous
172.16.146.2	172.16.146.1	FTP	21	49769	142	Response: 220 Welcome to the PowerBroker FTP service. Grab or leave juicy info here.
172.16.146.2	172.16.146.1	FTP	21	49769	100	Response: 331 Please specify the password.
172.16.146.1	172.16.146.2	FTP	49769	21	92	Request: PASS cfnetwork@apple.com
172.16.146.2	172.16.146.1	FTP	21	49769	89	Response: 230 Login successful.
172.16.146.1	172.16.146.2	FTP	49769	21	72	Request: SYST
172.16.146.2	172.16.146.1	FTP	21	49769	85	Response: 215 UNIX Type: L8
172.16.146.1	172.16.146.2	FTP	49769	21	71	Request: PWD
172.16.146.2	172.16.146.1	FTP	21	49769	100	Response: 257 "/" is the current directory
172.16.146.1	172.16.146.2	FTP	49769	21	74	Request: TYPE I
172.16.146.2	172.16.146.1	FTP	21	49769	97	Response: 200 Switching to Binary mode.
172.16.146.1	172.16.146.2	FTP	49769	21	73	Request: CWD /
172.16.146.2	172.16.146.1	FTP	21	49769	103	Response: 250 Directory successfully changed.
172.16.146.1	172.16.146.2	FTP	49769	21	72	Request: PASV
172.16.146.2	172.16.146.1	FTP	21	49769	115	Response: 227 Entering Passive Mode (172,16,146,2,95,17).
172.16.146.1	172.16.146.2	FTP	49769	21	95	Request: RETR Shield-prototype-plans
172.16.146.2	172.16.146.1	FTP	21	49769	146	Response: 150 Opening BINARY mode data connection for Shield-prototype-plans (72 bytes).
172.16.146.2	172.16.146.1	FTP	21	49769	90	Response: 226 Transfer complete.

FTP commands

Command	Description
USER	specifies the user to log in as.
PASS	sends the password for the user attempting to log in.
PORT	when in active mode, this will change the data port used.
PASV	switches the connection to the server from active mode to passive.
LIST	displays a list of the files in the current directory.
CWD	will change the current working directory to one specified.
PWD	prints out the directory you are currently working in.
SIZE	will return the size of a file specified.
RETR	retrieves the file from the FTP server.
QUIT	ends the session.

SMB

El Bloque de Mensajes del Servidor (SMB) es un protocolo ampliamente utilizado en entornos empresariales de Windows que permite compartir recursos entre hosts a través de arquitecturas de red comunes. SMB es un protocolo orientado a la conexión que requiere la autenticación del usuario desde el host hasta el recurso para garantizar que tenga los permisos correctos para usarlo o realizar acciones. Desde los cambios modernos, SMB ahora admite el transporte TCP directo a través del puerto 445, NetBIOS a través del puerto TCP 139 e incluso el protocolo QUIC.



Source	Destination	Protocol	src.p	dest.p	Length	Info
192.168.199.132	192.168.199.255	NBNS	137	137	92	Name query NB WPAD<00>
192.168.199.132	SCV	DNS	537…	53	86	Standard query 0x142e A officeclient.microsoft.com
192.168.199.133	SCV	DNS	643…	53	92	Standard query 0xfa7d A geo-prod.do.dsp.mp.microsoft.com
VMware_61:f5:5f	SCV	ARP			42	Who has 192.168.199.1? Tell 192.168.199.133
SCV	VMware_61:f5:5f	ARP			42	192.168.199.1 is at 00:50:56:c0:00:01
192.168.199.132	192.168.199.133	TCP	496…	445	66	49670 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.199.133	192.168.199.132	TCP	445	496…	66	445 → 49670 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.199.132	192.168.199.133	TCP	496…	445	54	49670 → 445 [ACK] Seq=1 Ack=1 Win=65536 Len=0
192.168.199.132	192.168.199.133	SMB	496…	445	213	Negotiate Protocol Request
192.168.199.133	192.168.199.132	SMB2	445	496…	506	Negotiate Protocol Response
192.168.199.132	192.168.199.133	SMB2	496…	445	232	Negotiate Protocol Request
192.168.199.133	192.168.199.132	SMB2	445	496…	566	Negotiate Protocol Response
192.168.199.132	192.168.199.133	SMB2	496…	445	220	Session Setup Request, NTLMSSP_NEGOTIATE
192.168.199.133	192.168.199.132	SMB2	445	496…	401	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
192.168.199.132	192.168.199.133	SMB2	496…	445	711	Session Setup Request, NTLMSSP_AUTH, User: DESKTOP-2AEFM7G\user
192.168.199.133	192.168.199.132	SMB2	445	496…	131	Session Setup Response, Error: STATUS_LOGON_FAILURE
192.168.199.132	192.168.199.133	TCP	496…	445	54	49670 → 445 [RST, ACK] Seq=1161 Ack=1389 Win=0 Len=0
192.168.199.132	192.168.199.133	TCP	496…	445	66	49671 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.199.133	192.168.199.132	TCP	445	496…	66	445 → 49671 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.199.132	192.168.199.133	TCP	496…	445	54	49671 → 445 [ACK] Seq=1 Ack=1 Win=65536 Len=0
192.168.199.132	192.168.199.133	SMB2	496…	445	232	Negotiate Protocol Request
192.168.199.133	192.168.199.132	SMB2	445	496…	566	Negotiate Protocol Response
192.168.199.132	192.168.199.133	SMB2	496…	445	220	Session Setup Request, NTLMSSP_NEGOTIATE
192.168.199.133	192.168.199.132	SMB2	445	496…	401	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
192.168.199.132	192.168.199.133	SMB2	496…	445	711	Session Setup Request, NTLMSSP_AUTH, User: DESKTOP-2AEFM7G\user
192.168.199.133	192.168.199.132	SMB2	445	496…	131	Session Setup Response, Error: STATUS_LOGON_FAILURE
192.168.199.132	192.168.199.133	TCP	496…	445	54	49671 → 445 [RST, ACK] Seq=1002 Ack=937 Win=0 Len=0
192.168.199.132	192.168.199.133	TCP	496…	445	66	49672 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.199.133	192.168.199.132	TCP	445	496…	66	445 → 49672 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
192.168.199.132	192.168.199.133	TCP	496…	445	54	49672 → 445 [ACK] Seq=1 Ack=1 Win=65536 Len=0
192.168.199.132	192.168.199.133	SMB2	496…	445	232	Negotiate Protocol Request
192.168.199.133	192.168.199.132	SMB2	445	496…	566	Negotiate Protocol Response
192.168.199.132	192.168.199.133	SMB2	496…	445	220	Session Setup Request, NTLMSSP_NEGOTIATE

Tcpdump Fundamentals

Tcpdump es un rastreador de paquetes de línea de comandos que puede capturar e interpretar directamente tramas de datos desde un archivo o una interfaz de red. Fue diseñado para usarse en cualquier sistema operativo tipo Unix y tenía un gemelo de Windows llamado WinDump. Cabe resaltar que necesitamos ser administradores del sistema para poder utilizar Tcpdump.

Traffic Captures with Tcpdump

Los comandos más utilizados en tcpdump son:

Switch Command	Result
D	Will display any interfaces available to capture from.
i	Selects an interface to capture from. ex. -i eth0
n	Do not convert addresses (i.e., host addresses, port numbers, etc.) to names.
e	Will grab the ethernet header along with upper-layer data.
X	Show Contents of packets in hex and ASCII.
XX	Same as X, but will also specify ethernet headers. (like using Xe)
v, vv, vvv	Increase the verbosity of output shown and saved.
c	Grab a specific number of packets, then quit the program.
s	Defines how much of a packet to grab.
S	change relative sequence numbers in the capture display to absolute sequence numbers. (13248765839 instead of 101)
q	Print less protocol information.
r file.pcap	Read from a file.
w file.pcap	Write into a file

Tcpdump Packet Filtering

Helpful TCPDump Filters

Filter	Result
host	`host` will filter visible traffic to show anything involving the designated host. Bi-directional
src / dest	`src` and `dest` are modifiers. We can use them to designate a source or destination host or port.
net	`net` will show us any traffic sourcing from or destined to the network designated. It uses / notation.
proto	will filter for a specific protocol type. (ether, TCP, UDP, and ICMP as examples)
port	`port` is bi-directional. It will show any traffic with the specified port as the source or destination.
portrange	`portrange` allows us to specify a range of ports. (0-1024)
less / greater "< >"	`less` and `greater` can be used to look for a packet or protocol option of a specific size.
and / &&	`and` `&&` can be used to concatenate two different filters together. for example, src host AND port.
or	`or` allows for a match on either of two conditions. It does not have to meet both. It can be tricky.
not	`not` is a modifier saying anything but x. For example, not UDP.

Host Filter

```
[!bash!]$ sudo tcpdump -i eth0 host 172.16.146.2
```

Este filtro se suele usar cuando queremos examinar solo un host o servidor específico. Con él, podemos identificar con quién se comunica este host o servidor y de qué manera.

Source/Destination Filter

```
[!bash!]$ sudo tcpdump -i eth0 src host 172.16.146.2
```

Nos permiten trabajar con las direcciones de la comunicación.

Utilizing Source With Port as a Filter

Podeos especificar el puerto que queremos ver (también podemos hacerlo por rango de puertos, ejem: 0-203):

```
[!bash!]$ sudo tcpdump -i eth0 tcp src port 80
```

Using Destination in Combination with the Net Filter

Este filtro puede utilizar el nombre o número de protocolo común para cualquier protocolo IP, IPv6 o Ethernet. Ejemplos comunes serían tcp[6], udp[17] o icmp[1] (especificándolo con: proto <número>). En los resultados a continuación, utilizaremos tanto el nombre común (arriba) como el número de protocolo (abajo).

```
[!bash!]$ sudo tcpdump -i eth0 dest net 172.16.146.0/24
```

Utilizing Greater

Podemos usar el modificador "greater 500" para mostrar solo paquetes con 500 bytes o más. Obtuvo una respuesta única en ASCII.

```
[!bash!]$ sudo tcpdump -i eth0 greater 500
```

AND Filter

El modificador AND nos mostrará cualquier cosa que cumpla con ambos requisitos. Por ejemplo, el host 10.12.1.122 y el puerto TCP 80 buscarán cualquier cosa del host de origen que contenga tráfico TCP o UDP del puerto 80.

```
[!bash!]$ sudo tcpdump -i eth0 host 192.168.0.1 and port 23
```

OR Filter

Tenemos una combinación de diferentes orígenes y destinos, junto con múltiples tipos de protocolo. Si usáramos el modificador OR (o ||), podríamos solicitar tráfico de un host específico o solo tráfico ICMP, por ejemplo. Volvamos a ejecutarlo y agreguemos un OR.

```
[!bash!]$ sudo tcpdump -r sus.pcap icmp or host 172.16.146.1
```

NOT Filter

Excluye lo que queramos

```
[!bash!]$ sudo tcpdump -r sus.pcap not icmp
```

Analysis with Wireshark

Wireshark es un analizador de tráfico de red gratuito y de código abierto, similar a tcpdump, pero con una interfaz gráfica. Wireshark es multiplataforma y capaz de capturar datos en tiempo real desde diversos tipos de interfaz (como Wi-Fi, USB y Bluetooth) y guardar el tráfico en varios formatos. Wireshark permite al usuario una inspección de paquetes de red mucho más profunda que otras herramientas.

TShark VS. Wireshark (Terminal vs. GUI)

TShark es ideal para equipos con poco o ningún entorno de escritorio y permite transferir fácilmente la información de captura recibida a otra herramienta mediante la línea de comandos. Wireshark es una opción de interfaz gráfica de usuario con numerosas funciones para la captura y el análisis de tráfico.

(yo voy a usar whireshark)

En este caso particular recomiendo ver los el módulo en esta parte de HTB, ya que viene bastante compactado y bien explicado.