# Hacker Highschool

## SECURITY AWARENESS FOR TEENS



# LECCIÓN 3
# PUERTOS Y PROTOCOLOS

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM

## WARNING

The Hacker Highschool Project is a learning tool and as with any learning tool there are dangers. Some lessons if abused may result in physical injury. Some additional dangers may also exist where there is not enough research on possible effects of emanations from particular technologies. Students using these lessons should be supervised yet encouraged to learn, try, and do. However ISECOM cannot accept responsibility for how any information herein is abused.

The following lessons and workbooks are open and publicly available under the following terms and conditions of ISECOM:

All works in the Hacker Highschool Project are provided for non-commercial use with elementary school students, junior high school students, and high school students whether in a public institution, private institution, or a part of home-schooling. These materials may not be reproduced for sale in any form. The provision of any class, course, training, or camp with these materials for which a fee is charged is expressly forbidden without a license including college classes, university classes, trade-school classes, summer or computer camps, and similar. To purchase a license, visit the LICENSE section of the HHS web page at http://www.hackerhighschool.org/licensing.html.

The HHS Project is an open community effort and if you find value in this project we ask that you support us through the purchase of a license, a donation, or sponsorship.

## AVISO

El proyecto Hacker Highschool es una herramienta de aprendizaje, y como tal existen riesgos. El mal uso de algunas lecciones puede terminar en daño físico. Existen riesgos adicionales ya que no existen estudios suficientes sobre los posibles efectos de las emisiones en algunas tecnologías. Los estudiantes que sigan estas lecciones deberían ser supervisados y motivados a aprenderlas, probarlas y utilizarlas. No obstante, ISECOM no acepta responsabilidad alguna por el mal uso de la información presentada.

Las siguientes lecciones y cuadernos de trabajo son abiertos y accesibles al público bajo los siguientes términos y condiciones de ISECOM:

Todas las obras del proyecto Hacker Highschool se proporcionan para su uso no comercial con estudiantes de escuelas primarias, secundaria y bachillerato ya sea en centros públicos, instituciones privada, o educación en casa. Este material no puede ser reproducido para su venta bajo ningún concepto. Impartir cualquier clase, formación o actividad con estos materiales cobrando por ello está expresamente prohibido sin la adquisición de una licencia, incluyendo cursos en escuelas, clases universitarias, cursos comerciales, cursos de verano, campamentos de informática, y similares. Para adquirir una licencia, visite la sección LICENCIA en la página web de Hacker Highschool en www.hackerhighschool.org/licensing.html.

El proyecto HHS es resultado del esfuerzo de una comunidad abierta. Si encuentra útil este proyecto, le pedimos que nos apoye mediante la compra de una licencia, una donación o patrocinio.

# Índice de contenidos

## Contributors

Pete Herzog, ISECOM
Marta Barceló, ISECOM
Chuck Truett, ISECOM
Kim Truett, ISECOM
Gary Axten, ISECOM
Glenn Norman, ISECOM
Mario Platt
Marco Ivaldi, @ Mediaservice.net
Greg Playle, Serco
Alfonso Arjona, @alfonsoarjona.net
Adrián Crespo, madrid.crespo@gmail.com

ISECOM

## Introducción y Objetivos

En un remoto pasado, antes de que existiera Internet, las comunicaciones electrónicas eran un auténtico Vudú. Cada fabricante de equipos tenía su propia idea acerca de cómo debían hablar las máquinas a través de un cable. Y nadie consideraba siquiera la posibilidad de que un equipo Wang pudiera comunicarse con una máquina de Burroughs.

El mundo cambió cuando científicos y estudiantes experimentaron las bondades de usar un terminal para acceder a un ordenador central. Llegó el famoso PC de IBM, y rápidamente sus propietarios querían acceder al ordenador central desde su ordenador personal. En poco tiempo, los módems estaban haciendo conexiones de acceso telefónico y los usuarios estaban trabajando con emuladores de terminal. La creación de redes (networking) se había graduado en magia negra y los iniciados fueron llamados (ahora sí) Gurús.

Y cambió nuevamente de forma dramática cuando Internet, que comenzó como un proyecto militar privado, se hizo accesible al público. El networking siempre había sido local, es decir, limitado a una oficina o como mucho a un campus. ¿Cómo iban a hablar todos estos sistemas tan diferentes?

La respuesta fue "calzar" un sistema universal de direcciones en las redes existentes, el sistema que generalmente llamamos **Protocolo de Internet** (IP). Piensa en esto de la forma siguiente: imagina que tu amigo en el extranjero te envía un paquete. Este paquete puede viajar en avión, tren o automóvil, pero la realidad es que no necesitas saber el horario de la aerolínea o la ubicación de la estación de tren más cercana. El paquete, con el tiempo, llegará a tu domicilio que es en última instancia lo único que importa. Tu **dirección IP** es muy parecida a esto: los paquetes pueden viajar como los electrones, haces de luz o las ondas de radio, pero esos sistemas no importan. Lo único importante es tu dirección IP y la dirección IP del sistema con el que estás hablando.

Pero en el mundo real, hay algo que complica esta idea, y es que más de una persona puede estar viviendo en una misma dirección. En el mundo de las redes, eso es lo que ocurre cuando un servidor proporciona los recursos habituales de HTTP y HTTPS, así como de FTP. ¿Te has dado cuenta de la *P* que está al final, o cerca de este, en esas siglas? Eso siempre es un claro indicativo de protocolo, que sólo es otra manera de decir "un tipo de comunicación".

Esta lección te ayudará a entender cómo funcionan los protocolos y sus puertos en Windows, OSX y Linux. También te familiarizará con varias utilidades que exploran las capacidades de la red de tu sistema.

Al terminar esta lección deberías tener unos conocimientos básicos de:

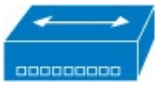
- El concepto de redes, y como tiene lugar la comunicación.
- Las direcciones IP, y
- Los puertos y protocolos.

## Conceptos básicos de redes

### Dispositivos

En el futuro, en tu carrera como hacker, verás una gran cantidad de diagramas de red. Es útil reconocer los símbolos más comunes:

**PC o Estación de trabajo**



**Hub**

**Switch**

**Router**

Un **hub** es similar a los viejos party-line telefónicos: Todo el mundo está en el mismo cable, y puede escuchar las conversaciones de los demás. Esto hace que una LAN se vuelva ruidosa rápidamente.

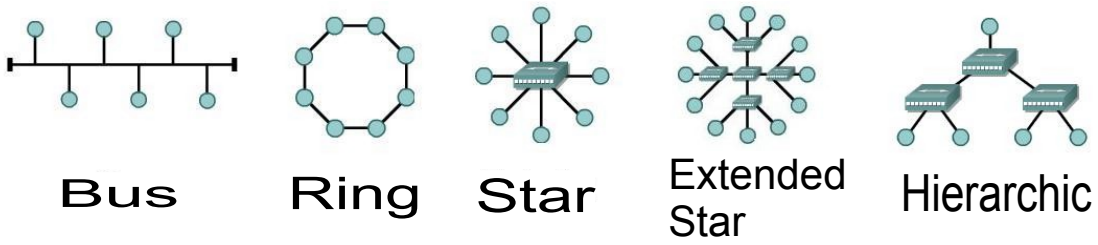
Un **switch** es mejor: filtra el tráfico de modo que sólo los dos computadores que estén hablando entre ellos puedan oir la conversación. Pero como ocurre con los hubs, sólo se usa en una LAN.

Un **router** se encuentre entre LANs y se usa para acceder a otras redes e Internet, utilizando direcciones IP. Examina los paquetes que se envían y decide cuál es la red a la que pertenecen los paquetes. Si el paquete pertenece a "otra" red, actúa como un agente de tráfico, y envía el paquete donde corresponda.

### Topología

Topología es otra forma de decir "la forma en la que las cosas se conectan". Las decisiones que tomemos con respecto a nuestra topología pueden afectarnos en el futuro, tanto positiva como negativamente, en función de las tecnologías que se utilicen, las limitaciones tecnológicas y físicas, el rendimiento y los requisitos de seguridad, el tamaño y la naturaleza de la organización, etcétera.

El punto de partida para la creación de redes es la red de área local (**LAN**). Una LAN permite compartir recursos a los ordenadores de un espacio físico común, como impresoras o discos, y a los administradores controlar ese acceso. Una estructura física de LAN puede parecerse a cualquiera de las siguientes topologías físicas:




En una topología de **bus**, todos los computadores están conectados a un único cable, y cada computador puede comunicarse directamente con cualquiera de los otros. Pero si se rompe el bus en cualquier parte, todo el mundo se encontrará desconectado.

En una configuración de **anillo (ring)**, cada ordenador está conectado con el siguiente, el último se conecta con el primero, y cada computador sólo puede comunicarse directamente con sus dos adyacentes. Ya que este diseño utiliza normalmente un anillo doble, hay cierta tolerancia a fallos.

La topología de anillo se usa raramente hoy en día. Pero los anillos se utilizan a menudo para un "campus", por lo general con dos anillos, y enviando el tráfico en direcciones opuestas, dada su fiabilidad.

En la topología de **estrella (star)**, ninguno de los computadores está conectado directamente con los otros. En lugar de eso, lo hace a través de un hub o switch que reenvía la información de un equipo a otro.

Si se conectan entre si varios hubs o switches, lo que obtendrás es una topología en **estrella extendida (extended star).**

En una estrella o estrella extendida, todos los puntos se denominan **pares (peers)**, es decir, que son esencialmente iguales. Esta es la topología de red más común hoy en día.

Sin embargo, si conectas dos estrellas o dos estrellas extendidas entre si usando un punto central que controle o limite el tráfico entre las dos redes, lo que tendrás será una topología de red **jerárquica**. Esta es la que se despliega habitualmente en las grandes empresas.

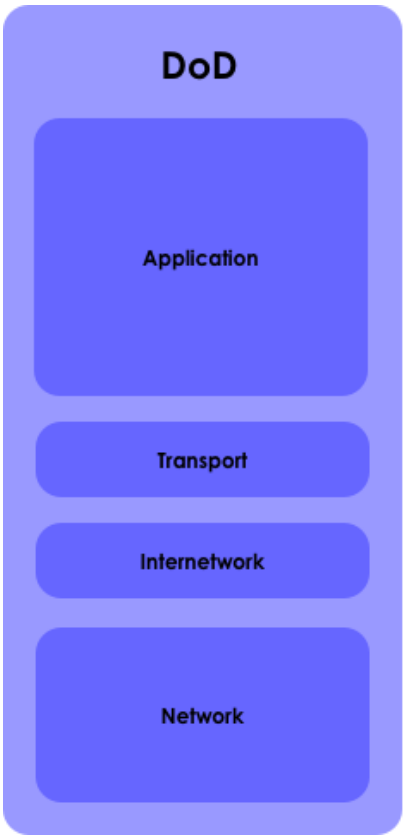
## El modelo TCP/IP

TCP/IP fue desarrollado por el **DOD** (Department of Defense – Departamento de Defensa) de los Estados Unidos y **DARPA** (Defense Advanced Research Project Agency - Agencia de Investigación de Proyectos Avanzados de Defensa) en los 70. Se diseñó para ser un estándar abierto que cualquiera pudiera usar para conectar computadores entre si e intercambiar información entre ellos. Esto se convirtió, finalmente, en la base de Internet.

En general, la forma más simple del modelo TCP / IP se llama **Modelo DoD**, y por ahí es por donde vamos a empezar.

### Capas

El modelo sencillo DoD define cuatro capas totalmente independientes, que dividen el proceso de comunicación entre dos dispositivos. Las capas por las que pasa la información entre dos dispositivos son:




**Aplicación (Application)**

La capa de aplicación es exactamente lo que con toda probabilidad piensas que es: la capa donde trabajan las aplicaciones como Firefox, Opera, clientes de correo electrónico, sitios de redes sociales, mensajería instantánea y aplicaciones de chat. En realidad sólo unas pocas aplicaciones acceden a Internet: algunas aplicaciones de oficina, por ejemplo, se conectan a galerías de imágenes en línea para que puedas añadir gráficos prediseñados. La capa de aplicación crea la carga útil que llevarán todas las otras capas. Una buena analogía es un sistema postal. La aplicación crea el paquete y lo envuelve con instrucciones sobre cómo debe ser utilizado el paquete. Luego, entrega el paquete a la sala de correo: la capa de transporte.

**Transporte (Transport)**

La capa de transporte establece las conexiones de red, que se llaman sesiones. En el mundo de Internet, el protocolo principal en la capa de transporte es TCP, el Protocolo de Control de Transmisión. TCP agrega otro "envoltorio" a la parte exterior del paquete, con instrucciones sobre qué paquete es (por ejemplo, 1 de 3), la forma de asegurarse de que el paquete llegó a su destino, y si el paquete está intacto.

Supongamos que vas a enviar una carta a tu madre. La carta puede ser corta o larga, pero es demasiado grande para enviarse a través de Internet de una sola pieza. Por eso, TCP rompe la carta en segmentos, trozos pequeños que se numeran consecutivamente, con un poco de código de comprobación de errores al final. Si un paquete se daña

durante el transporte, TCP pide una retransmisión. En el extremo receptor, TCP une la carta en el orden correcto y tu madre la recibe en su correo electrónico.

**Interred o red (Internetwork)**

Esta capa añade información acerca de las direcciones y puertos de origen y destino, además de donde empieza el **paquete** y donde termina. Es como una empresa de mensajería que entrega paquetes en la dirección correcta. No le importa si llegan todos los paquetes o si están intactos: eso es el trabajo de la capa de Transporte. El protocolo más importante en este nivel es, apropiadamente, **IP**, Internet Protocol. Esta es la capa que utiliza las direcciones IP para entregar los paquetes en el lugar adecuado y por la mejor ruta.

**Host-Red (Network Access)**

Esta capa es la red física de bajo nivel que usas para conectarte a Internet. Si usas un acceso telefónico a redes, lo sentimos, ya que estás usando una burda conexión PPP. Si tienes DSL puede que estés utilizando ATM o Metro Ethernet. Y si tienes Internet por cable estarás utilizando DOCSIS. No importa lo que uses, porque TCP/IP hace que todo funcione. La capa Host-Red consiste en el cable Ethernet y la tarjeta de red (NIC), o en la tarjeta inalámbrica y el punto de acceso. Maneja los unos y ceros de nivel más bajo (bits) a medida que van de un punto a otro.

**Aliments tu miente: Sorprenda a sus amigos por conocer la diferencia**

Mira "El modelo OSI" al final de esta lección para ver una versión alternativa del modelado de redes.

## Protocolos

Así que ahora ya estás conectado a Internet. Parece bastante simple, pero consideremos una situación habitual en la que te puedes encontrar: estás llevando a cabo una investigación inocente, pero importante, en Internet mientras que tu querido hermano o hermana se dedica a perder el tiempo viendo una película por streaming. ¿Por qué no se mezclan los dos tráficos? ¿Qué hace la red para distinguirlos?

La respuesta son los **protocolos**, que son como los idiomas que hablan los distintos tipos de tráfico. El tráfico Web utiliza un protocolo, las transferencias de archivos usan otro, y el correo electrónico otro diferente. Como todas las cosas digitales, en realidad los protocolos no utilizan nombres en el nivel de red: usan direcciones IP y **números de puerto**.

**Protocolos del nivel de Aplicación**

El **FTP** o *File Transfer Protocol* se utiliza para transmitir archivos entre dos dispositivos. Utiliza un puerto para entregar los datos, y otro para enviar señales de contro ("¡Archivo recibido! ¡Gracias!"). Los puertos más habituales son el 20 y 21.

El **HTTP** o H*yper-Text Transfer Protocol* se usa para páginas web. Este tráfico usa normalmente el puerto 80.

**SMTP** o *Simple Mail Transfer Protocol* es el protocolo que envía emails. El puerto habitual es el 25.

**DNS** o *Domain Name Service* es la forma en que un dominio cómo ISECOM.org se dirige hacia una dirección IP tal que 216.92.116.13. El puerto más utilizado es el 53.

### Protocolos de la capa de Transporte

**TCP** no es el único protocolo de la capa de Transporte: hay dos protocolos principales que se utilizan en esta capa para transferir datos.

TCP o **Transmission Control Protocol** establece una conexión logica (una **sesión**) entre dos hosts en una red. Establece esta conexión usando una negociación (handshake) de tres vías:

1. Cuando mi computador quiere conectarse al tuyo, envía un paquete **SYN**, que básicamente dice: "Vamos a sincronizar nuestros relojes para que podamos intercambiar datos usando marcas de tiempo (timestamps)"
2. Tu computador (si va a aceptar la conexión) responde con un paquete de confirmación **SYN/ACK**.
3. Mi computador cierra el trato mandando un **ACK**, y ya estaremos conectados. Pero esto sólo ocurre con TCP.

**UDP** o U*ser Datagram Protocol* es un protocolo de transporte al que ni siquiera le importa si tienes conexión. Es como el beber agua de una manguera: si atrapas el chorro beberás, y si no... pues no. Esto hace que UDP sea muy rápido, así que es útil para cosas como hacer streaming de vídeo, donde si pierde un paquete (frame) no importa mucho, o en juegos en línea, donde perder un fotograma tampoco importa (bueno, eso depende de a qué lado de la bala estés).

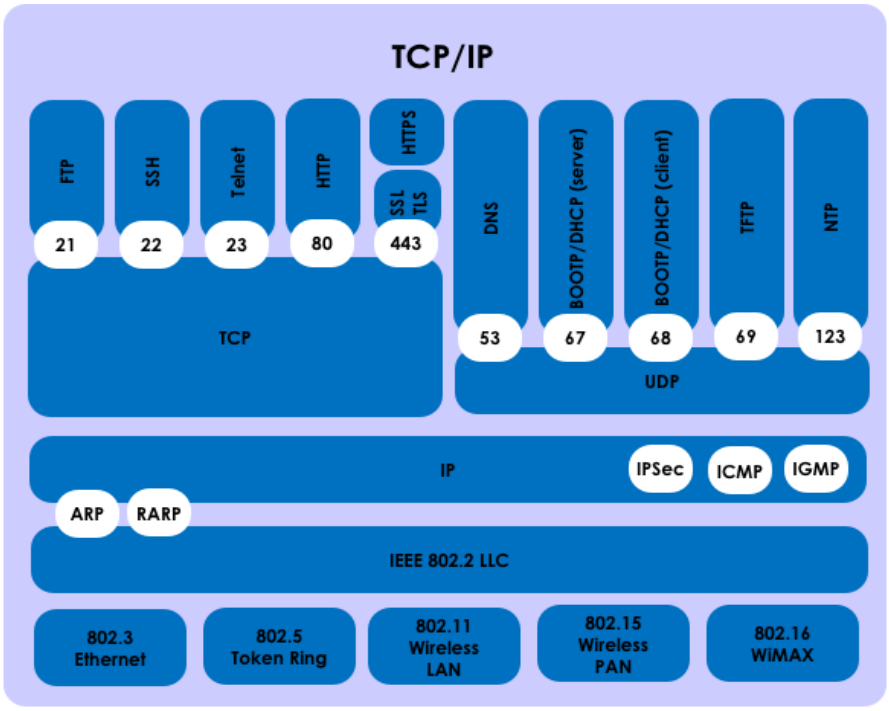
### Protocolos de la capa de Internet

**IP** o *Internet Protocol* actúa como un protocolo universal para permitir que dos computadores cualquiera se comuniquen a través de cualquier red en cualquier momento. Es como el cartero que entrega el correo, lo único que hace es entregar los paquetes en su dirección de destino.

### Protocolo de Mensajes de Control de Internet (ICMP)

**ICMP** es el protocolo que utilizan los dispositivos en red y los administradores de redes para resolver problemas y mantener la red. Incluye cosas como el PING (Packet Inter Net Groper) y otros comandos parecidos que sirven para probar la red e informar de errores.

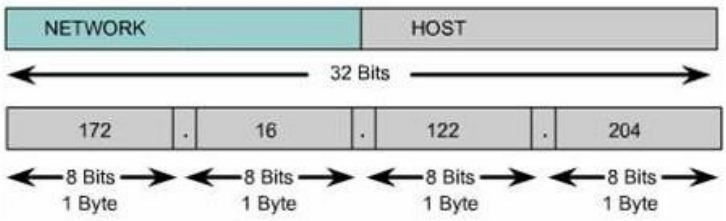
Puestos en conjunto , los puertos y protocolos se representan así:




## Direcciones IPv4

Los nombres de Domino son muy útiles para los seres humanos, porque somos buenos recordando nombres como ISECOM.org. Pero en la actualidad, las redes no los entienden; sólo comprenden números de dirección IP. Así que cuando quieres ir a ISECOM.org, tu computador hace una búsqueda rápida usando el **DNS** (Domain Name Service) para encontrar la dirección IP correspondiente.

Las direcciones IP son como las direcciones postales. Si quieres recibir correo, tienes que tener una. IPv4 consiste en 32 bits que se dividen en cuatro octetos (8 bit) que se separan por puntos. Parte de la dirección IP identifica la red, y el resto identifica individualmente a los computadores de esa red. Esto significa que hay $2^{32}$ (o 4.294.967.296 ) direcciones únicas en Internet bajo IPv4. Imagina que estas partes son como la ciudad/país (la red) y la calle (el host).




Volviendo a la analogía del servicio postal: IP es el furgón de reparto que "intenta" llevar el paquete a la oficina de correos correcta. TCP es el envoltorio exterior con la lista de cuantos paquetes tiene el envío, y cual es este. El nivel "host" de la dirección es la casa particular (computador) en la cual debe entregar el paquete.

Hay direcciones IP públicas y privadas (no enrutables). Las direcciones privadas se usan en redes privadas, y por un estándar industrial no pueden enviarse fuera de la red,

aunque los computadores de dos redes privadas diferentes (pero desconectadas entre ellas) pueden tener direcciones IP duplicadas. Las direcciones IP que ha definido IANA (Internet Assigned Numbers Authority) para utilizar en redes privadas son:

10.0.0.0 hasta 10.255.255.255 (Clase A)

172.16.0.0 hasta 172.31.255.255

192.168.0.0. hasta 192.168.255.255 (Clase B)

Las direcciones IP se dividen en clases basándose en qué parte de la dirección se usa para identificar la red, y que parte se usa para identificar computadores individuales.

| Class A | Network | Host | | |
|---|---|---|---|---|
| Octet | 1 | 2 | 3 | 4 |

| Class B | Network | | Host | |
|---|---|---|---|---|
| Octet | 1 | 2 | 3 | 4 |

| Class C | Network | | | Host |
|---|---|---|---|---|
| Octet | 1 | 2 | 3 | 4 |

| Class D | Host | | | |
|---|---|---|---|---|
| Octet | 1 | 2 | 3 | 4 |

Dependiendo del tamaño asignado a cada parte, se permitirán más dispositivos dentro de la red, o se permitirán más redes. Las clases vigentes son:

- Clase A: El primer bit siempre es cero, así que esta clase incluye las direcciones entre 0.0.0.0 (que, por convenio, nunca se usa) y 126.255.255.255. *Nota: la direcciones 127.x.x.x están reservadas para los servicios de loopback o localhost. Mira lo que significa esto más adelante.*

- Clase B: Los dos primeros bits del primer octeto son '10', así que esta clase incluye las direcciones entre 128.0.0.0 y 191.255.255.255.

- Clase C: Los tres primeros bits del primer octeto son '110', luego esta clase incluye las direcciones entre192.0.0.0 y 223.255.255.255.

- Clase D: Los cuatro primeros bits del primer octeto son '1110', por tanto esta clase incluye las direcciones entre 224.0.0.0 y 239.255.255.255. Estas direcciones están reservadas para implementaciones de grupos multicast.

- Las direcciones restantes se usan para experimentar o se reservan para posibles asignaciones en el futuro.

Llegados a este punto, vemos que las Clases no se usan para distinguir entra la parte de la dirección usada para identificar la red, y la parte que identifica dispositivos individuales. Para eso, usamos las *máscaras*. En la máscara, un bit a '1' representa la parte que contiene la identificación de la red, y un bit a '0' hace lo mismo para la parte que identifica los dispositivos individuales. Por eso para identificar un dispositivo, además de la dirección IP es necesario especificar la máscara de red:

| IP: 172.16.1.20 |
| --- |
| Máscara: 255.255.255.0 |

Las IP 127.x.x.x se reservan para usarse como direcciones de loopback o localhost, es decir, que se refieren directamente al computador local. Cada computador tiene como dirección localhost 127.0.0.1, y por tanto esa dirección no puede utilizarse para identificar un dispositivo diferente. Hay otras que tampoco pueden utilizarse, que son las de direcciones red (network) y las direcciones de multidifusión (broadcast).

La dirección de red es una en la cual la parte que se especificaría el dispositivo es todo ceros. Esta dirección no puede usarse, ya que identifica la red completa y por tanto no puede ser empleada para especificar un dispositivo en concreto.

| IP: 172.16.1.0 |
| --- |
| Máscara: 255.255.255.0 |

La dirección de multidifusión es aquella en la cual la parte que identifica un dispositivo, identifica a todos. Esta dirección tampoco puede usarse para identificar un equipo específico, porque es la dirección que se usa para enviar información a todos los computadores que se encuentran en una red concreta.

| IP: 172.16.1.255 |
| --- |
| Máscara: 255.255.255.0 |

**Alimenta tu Mente: IPv6**

El ICANN/IANA asignó los 8 bloques finales de direcciones IPv4 a los Registros Regionales de Internet en febrero de 2011.

IPv6 (Internet Protocol version 6) es una versión del Protocolo Internet (IP) desarrollado por la "Internet Engineering Task Force" (IETF), que está destinado a reemplazar IPv4 como el protocolo de comunicaciones dominante para el tráfico de Internet

IPv6 ofrece la mejor solución para el problema del agotamiento de direcciones IPv4 en Internet, mediante el uso de una dirección de 128 bits en lugar de los 32 bits de Ipv4.

La longitud de dirección ampliada que ofrece el IPv6 elimina la necesidad de utilizar técnicas tales como traducir direcciones de red o puertos (NAT / PAT).

Por convenio IPv6 utiliza 32 números hexadecimales, organizados en 8 cuartetos de 4 dígitos hexadecimales separados por dos puntos, para representar una dirección IPv6 de 128 bits. Por ejemplo:

2001:0 db8: 85a3: 0000:0000:8 A2E: 0370:7334

IPv6 incluye muchos tipos diferentes de direcciones, incluyendo las unicast y mulitcast

**IPv6 Address Types**

| Tipo de dirección | Propósito | Prefijo | Prefijo HEX sencillo |
|---|---|---|---|
| Global Unicast | Paquetes Unicast enviados a través de Internet. | 2000::/3 | 2 ó 3 |
| Local única | Estas direcciones se destinan exclusivamente a conjuntos de sitios cooperantes. Se introdujeron en IPv6 para reemplazar a las direcciones locales. Estas direcciones también proporcionan un número pseudoaleatorio de 40-bits que reduce el riesgo de conflictos de direcciones. | FD00::/8 | FD |
| Link Local | Es el prefijo de enlace local que ofrece IPV6. Este prefijo de dirección indica que la dirección sólo es válida en el enlace físico local. | FE80::/10 | FE80 |
| Multicast | Multicasts que permanecen en la subred local. | FF02::/16 | FF02 |

**Formato de direcciones**

Una dirección IPv6 se representa por 8 grupos de 16 bits en hexadecimal, separados por dos puntos (:). Por ejemplo:
2001:0db8:85a3:0000:0000:8a2e:0370:7334

Los números en hexadecimal no son sensible a mayúsculas/minúsculas.

Una dirección IPv6 puede abreviarse con las siguientes reglas:

1. Omitiendo los ceros a la izquierda en los valores de 16 bits.
2. Reemplazando uno o más grupos consecutivos de ceros por dos puntos dobles. Por ejemplo:

| Dirección | fe80 | : | 0000 | : | 0000 | : | 0000 | : | 0202 | : | b3ff | : | fe1e | : | 8329 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Despues de la Regla 1 | fe80 | | 0 | | 0 | | 0 | | 202 | | b3ff | | fe1e | | 8329 |
| Despues de la Regla 2 | fe80 | | | | | | | | 202 | | b3ff | | fe1e | | 8329 |

Estas son las representaciones en texto de estas direcciones:

fe80:0000:0000:0000:0202:b3ff:fe1e:8329

fe80:0:0:0:202:b3ff:fe1e:8329

fe80::202:b3ff:fe1e:8329

Otro ejemplo interesante es la dirección de loopback:

0:0:0:0:0:0:0:1

::1

**Implementación de una pila IP doble (Dual IP Stack)**

Significa que el host tiene tanto una dirección de IPv4 como una IPv6 asociada a cada tarjeta de red. Por tanto, puede enviar paquetes IPv4 a hosts IPv4 y paquetes IPv6 a hosts Ipv6.
El enfoque de doble pila puede ser un enfoque razonable para migrar una empresa a IPv6.

## Puertos

TCP y UDP usan los puertos para intercambiar información con las aplicaciones. Un puerto es una extensión de una dirección, algo parecido a añadir el número de casa o apartamento a la dirección de una calle. Una carta con la dirección de la calle llegará al edificio de apartamentos correcto, pero sin saber el número de apartamento no se podrá entregar en el buzón correcto. Los puertos trabajan de forma muy parecida. Un paquete puede entregarse en la dirección IP correcta, pero sin el puerto asociado no hay forma de saber qué aplicación debe utilizar ese paquete. El número de puerto es un valor de 16 bits, lo que significa que tiene valores decimales entre 0 y 65535 (2 elevado a 16).

Otra forma de verlo podría ser esto: cada computador es una oficina de correos y cada aplicación tiene su propio apartado de correos: dos aplicaciones no pueden tener el mismo apartado de correos. Por tanto, el número de puerto es como el apartado de correos.

Una vez que se han definido los puertos, ya es posible que los distintos tipos de información que se han enviado a una dirección IP lleguen a la aplicación adecuada. Usando puertos, un servicio que se ejecuta en un computador remoto puede determinar el protocolo que necesita para enviar esa información y mantener conexiones simultáneas con distintos clientes.

Por ejemplo, si un computador local intenta conectarse a la página web www.osstmm.org, cuya dirección IP es 62.80.122.203 con un servidor web corriendo en el puerto 80, el computador local podrá conectar al remoto usando esta **dirección de socket**:

**62.80.122.203:80**

Con objeto de mantener un nivel de estandarización en los puertos más comunes, IANA estableció que los puertos entre 0 y 1024 se usarán para servicios comunes, privilegiados o bien conocidos. El resto de puertos, hasta 65535, se usan para asignaciones dinámicas o servicios particulares.

Los puertos más usados (bien conocidos) asignados por IANA son estos:

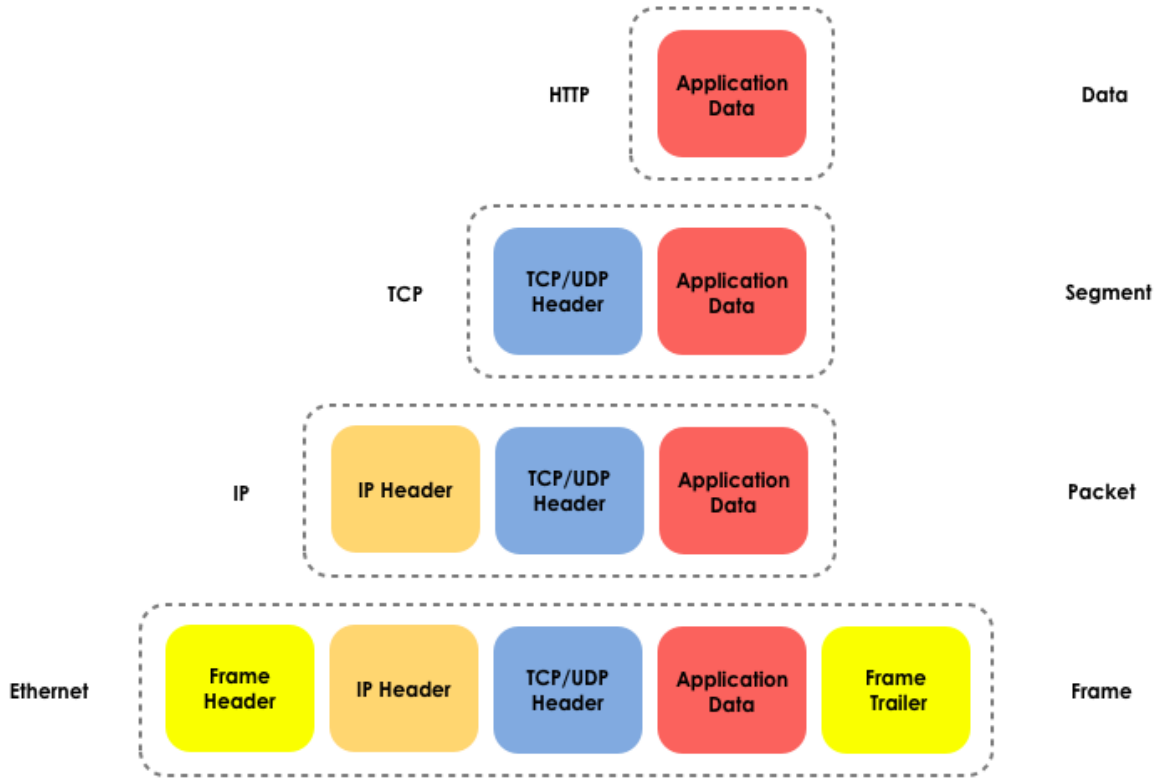
| Port Assignments | | |
|---|---|---|
| **Number** | **Keywords** | **Description** |
| 5 | rje | Remote Job Entry |
| 0 | | Reservado |
| 1-4 | | Sin asignar |
| 7 | echo | Eco (Echo) |
| 9 | discard | Discard |
| 11 | systat | Active Users |
| 13 | daytime | Fecha y hora actual (Daytime) |
| 15 | netstat | Who is Up or NETSTAT |
| 17 | qotd | Quote of the Day |
| 19 | chargen | Character Generator |
| 20 | ftp-data | File Transfer [Default Data] |
| 21 | ftp | File Transfer [Control] |
| 22 | ssh | SSH Remote Login Protocol |
| 23 | telnet | Telnet |
| 25 | smtp | Simple Mail Transfer |
| 37 | time | Time |
| 39 | rlp | Resource Location Protocol |
| 42 | nameserver | Host Name Server |
| 43 | nicname | Who Is |
| 53 | domain | Domain Name Server |
| 67 | bootps | Bootstrap Protocol Server |
| 68 | bootpc | Bootstrap Protocol Client |
| 69 | tftp | Trivial File Transfer |
| 70 | gopher | Gopher |
| 75 | | any private dial out service |
| 77 | | any private RJE service |
| 79 | finger | Finger |
| 80 | www-http | World Wide Web HTTP |
| 95 | supdup | SUPDUP |
| 101 | hostname | NIC Host Name Server |
| 102 | iso-tsap | ISO-TSAP Class 0 |
| 110 | pop3 | Post Office Protocol - Version 3 |
| 113 | auth | Authentication Service |

| Port Assignments | | |
|---|---|---|
| 117 | uucp-path | UUCP Path Service |
| 119 | nntp | Network News Transfer Protocol |
| 123 | ntp | Network Time Protocol |
| 137 | netbios-ns | NETBIOS Name Service |
| 138 | netbios-dgm | NETBIOS Datagram Service |
| 139 | netbios-ssn | NETBIOS Session Service |
| 140-159 | | Unassigned |
| 160-223 | | Reserved |

## Encapsulación

Cuando un elemento de información (un e-mail, por ejemplo) se envía de un computador a otro, se ve sometido a una serie de transformaciones. La capa de aplicación genera los datos, que se envían a la capa de transporte. La capa de transporte toma esta información y le añade una cabecera (header). Esta cabecera contiene información, como las direcciones IP de origen y destino, y explica que es lo que se debe hacer con el dato para entregarlo en el destino apropiado. La siguiente capa añadirá otra cabecera, y así sucesivamente. Este procedimiento se conoce como encapsulación.

Cada capa tras la primera encapsula los datos de la capa anterior, hasta que se llega a la última capa, en la cual se hace la transmisión de los datos. La siguiente imagen muestra la encapsulación de forma gráfica:



Cuando la información encapsulada llega a su destino, debe desencapsularse. Como cada capa pasa la información a la siguiente capa de la pila, se elimina la información innecesaria que contiene la cabecera colocada por la capa inferior.

El pedacito final de información de este gran esquema de direccionamiento es una dirección única que está en la tarjeta de red del computador: la "Media Access Controller" (MAC). Esta dirección se escribe normalmente mediante grupos de dos caracteres alfanuméricos separados por dos puntos (:) o un guión (-). Es la dirección física de la tarjeta de red, y no cambia. Por ejemplo:

**00-15-00-06-E6-BF**.

## Ejercicios

3.1. Usando los comandos que has aprendido en las lecciones 1 y 2, obtén tu dirección IP, máscara de red, servidor de DNS y la dirección MAC. Compárala con la de tus compañeros. ¿Qué es parecido, y qué diferente? Dado el esquema de direcciones IP que está utilizando la red ¿se trata de una red pública o privada?

3.2. netstat

El comando **netstat** te da estadísticas de tu red: con quién estás conectado, cuánto tiempo ha estado activada la red, y muchas cosas más. En Linux, Windows o OSX puedes abrir una terminal de línea de comandos (CLI) y escribir:

```
netstat
```

En la ventana de CLI, verás una lista con las conexiones establecidas. Si quieres verlas en forma numérica, escribe:

```
netstat -n
```

Para ver las conexiones y los puertos activos (escuchando o abiertos), escribe:

```
netstat -an
```

Para ver la lista de opciones, escribe:

```
netstat -h
```

En la salida de netstat, observa las columnas que muestran las IP local y remotas, y los puertos que están usando:

```
Proto Recv-Q Send-Q  Local Address       Foreign Address
(state)
tcp4    0      0  192.168.2.136.043  66.220.149.94.443
ESTABLISHED
```

Los puertos son los números que están detrás de las direcciones IP; pueden estar separados por un punto (.) o dos puntos (:). ¿Por qué los puertos utilizados por las direcciones remotas son distintos de los usados por las direcciones locales?

Abre varias páginas web en distintas ventanas o pestañas del navegador, y ejecuta netstat de nuevo.

Si hay varias pestañas abierta, ¿Cómo sabe el navegador que información va en cada ventana?

¿Por qué no hay un puerto de escucha cuando usamos un navegador?

¿Qué protocolos se usan?

¿Qué pasa si usamos un protocolo en más de una instancia?

3.3. Mi primer servidor

Para este ejercicio, necesitas el programa **netcat**. BackTrack lo incluye por defecto, OSX también, pero puedes bajarte instaladores con distintas versiones para varios sistemas operativos.

**1.** En una ventana CLI, escribe:

```
nc -h
```

Esto mostrará las opciones disponibles en netcat.

Para crear un servidor sencillo, escribe:

```
nc -l -p 1234 (Linux, Windows)
```

o

```
nc -l 1234 (Mac)
```

Acabas de arrancar un servidor que escucha en el puerto 1234.

**2.** Abre una segunda ventana CLI y escribe:

```
netstat -a
```

Esto verificará si hay un nuevo servicio escuchando en el puerto 1234.

Para comunicarte con el servidor ¡necesitas un cliente!
Escribe en tu segunda ventana CLI:

```
nc localhost 1234
```

Este comando crea una conexión con el servidor que escucha en el puerto 1234. Ahora, cualquier cosa que escribas en cualquiera de las dos ventanas aparecerá en la otra.

Considera las repercusiones que tiene esto. Discútelas en clase. ¿Alguien puede aprovechase de esta capacidad para atacar tu sistema?

Netcat envía todo el tráfico en claro. ¿Existe una alternativa segura?(Pista:cryptcat es una de ellas)

**3.** Detén tu servidor volviendo a la primera ventana CLI, y pulsando Control-C.

**4.** Ahora crea un archivo de texto que contenga la frase "¡Bienvenido al servidor de Hacker Highschool!" y llámalo *test*.

Una vez que hayas terminado, mira el siguiente comando y explícaselo al profesor: ¿qué hace cada parte? Luego, escribe en tu primera ventana CLI:

```
nc -l -p 1234 < test
```

Desde otra ventana CLI, conéctate al servidor escribiendo:

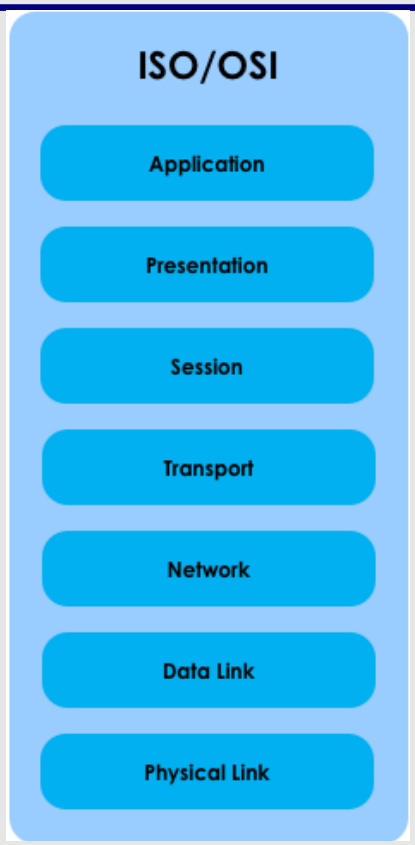
```
nc localhost 1234
```

Cuando el cliente se conecte al servidor, deberías ver la salida del archivo *test*.

¿Qué protocolo se ha usado para conectar al servidor?

¿Netcat permite cambiar esto? Si es así, ¿cómo?

## Alimenta tu Mente: El modelo OSI




El modelo OSI se desarrolló en los ochenta (casi diez años después del modelo TCP/IP) por ISO, la Organización de Estándares Internacionales. OSI significa Interconexión de Sistemas Abiertos (Open Systems Interconnection), y fue un intento de estandarizar la arquitectura de red por parte de una organización que no estaba involucrada realmente en el desarrollo de las redes.

El modelo OSI es un modelo por capas con un puñado de reglas sencillas. Las funciones similares se agrupan en una misma capa, y (por favor, no te olvides de esto) cada capa es atendida por su capa **inferior**, y sirve a la capa que se encuentra **sobre** ella.

Este modelo por capas es una buena idea, porque cada capa (en teoría) se encarga de sus propias comunicaciones y los nuevos desarrollos de cada capa no rompen nada de las otras. Esta caraterística explica por si mismas el boom de Internet que hemos visto desde el 2000, con nuevas aplicaciones y servicios apareciendo cada día.

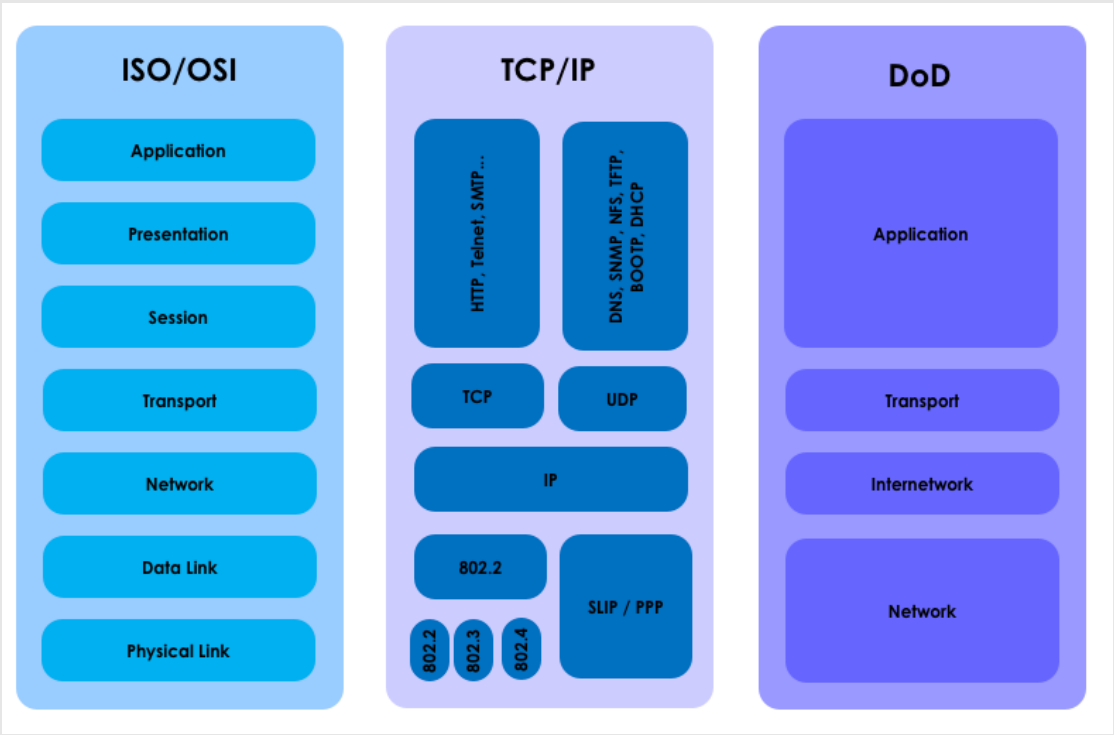
Aparte de las dos reglas del modelo OSI que hemos discutido previamente(las funciones similares están agrupadas, y cada capa recibe servicio de la capa inferior y sirve a la superior) este estándar tiene una regla estricta más. Cada capa implicada en la comunicación desde un computador, se comunica directamente con la misma capa del otro computador. Esto significa que cuando escribes www.google.com en tu navegador, hay una interacción directa entre en la interfaz de la capa 7 de tu computador (el navegador) y el servidor web de Google (que también está en la capa 7). Lo mismo ocurre con todas las demás capas.

Así que en primer lugar, vamos a definir las capas del modelo OSI y sus respectivas responsabilidades.

| | |
|---|---|
| Capa de aplicación (Application Layer) | Responsable de la interacción directa entre las aplicaciones y los interfaces de la aplicación de usuario. Por ejemplo, un navegador web como IE o Firefox |
| Capa de presentación (Presentation Layer) | Responsable de garantizar que los datos se intercambian de forma comprensible entra ambas partes. En todos los servicios que usen encriptación, esta debe tener lugar en la capa de presentación. |
| Capa de sesión (Session Layer) | Responsable del control de comunicación entre computadores. Básicamente, establece, controla y termina las conexiones que se dan entre dos computadores. |
| Capa de transporte (Transport Layer) | Proporciona una transferencia transparente de los datos entre dos computadores, ofreciendo servicios de transferencia de datos confiables a las capas superiores. Esto significa que es la responsable de ensamblar todos los datos que se envían en pequeños fragmentos para poder enviarlos a una red de datos. Si un paquete se pierde o no se recibe, la capa de transporte es la responsable de asegurarse de que sólo ese paquete es retransmitido y reensamblado en el orden correcto. |
| Capa de red (Network Layer) | Esta capa es responsable del direccionamiento de la conexión. No sólo debe asegurarse de que cada dirección en única en Internet, si no que además debe asegurarse de que cualquier camino disponible (ya sea bueno o malo) entrega la información donde tiene que ir, y de que nuestra información será enviada de salto en salto (hop) hasta que alcance su destino final. |
| Capa de enlace de datos (Data link layer) | La capa de enlace de datos fue diseñada para garantizar que la capa física pueda recuperarse de los errores que puedan ocurrir y hacer frente a los distintos medios de conexión. Básicamente "prepara" o "encaps ula" los datos de forma que puedan ser transmitidos por el medio físico necesario (ondas de radio, fibra óptica, cables de cobre). |
| Capa Física (Physical layer) | Esta capa define las especificaciones físicas y eléctricas de los dispositivos, y qué se necesita hacer para transmitir la información a través del medio elegido. En una conexión WiFi, sería enviar una señal de radio, en fibra óptica sería la luz que se emitirá, o en el caso de una conexión por cable de cobre, la electricidad que se envía por los hilos. |

Estas siete capas incluyen todo lo que se necesita para una comunicación confiable entre computadoras.

Así es como se ven los distintos modelos que hemos discutido, lado a lado:



Today's teens are in a world with major communication and productivity channels open to them and they don't have the knowledge to defend themselves against the fraud, identity theft, privacy leaks and other attacks made against them just for using the Internet. This is the reason for Hacker Highschool.

**The Hacker Highschool project is the development of security and privacy awareness learning materials for junior high and high school students.**

Hacker Highschool is a set of lessons and a practical means of making hackers. Beyond just providing cybersecurity awareness and critical Internet skills, we need to teach the young people of today how to be resourceful, creative, and logical, traits synonymous with hackers. The program contains free security and privacy awareness teaching materials and back-end support for teachers of accredited junior high, high schools, and home schooling. There are multiple workbooks available in multiple languages. These are lessons that challenge teens to be as resourceful as hackers, including safe Internet use, web privacy, researching on the internet, avoiding viruses and Trojans, legalities and ethics, and more.

**The HHS program is developed by ISECOM, a non-profit, open-source research group focused on security awareness and professional security development and accreditation.**

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM