# Hacker Highschool
## SECURITY AWARENESS FOR TEENS

# LESSON 3
# PORTS AND PROTOCOLS

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES

# "License for Use" Information

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 Hacker Highschool web page at www.hackerhighschool.org/license.

The HHS Project is a learning tool and as with any learning tool, the instruction is the influence of the instructor and not the tool. ISECOM cannot accept responsibility for how any information herein is applied or abused.

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

# Table of Contents

# Contributors

Gary Axten, ISECOM

La Salle URL Barcelona

Kim Truett, ISECOM

Chuck Truett, ISECOM

Marta Barceló, ISECOM

Pete Herzog, ISECOM


# 3.1 Introduction

The text and exercises in this lesson try to impart a basic understanding of the ports and protocols in current use, as well as their relevance within the operating systems, Windows and Linux.

Additionally, you will have the opportunity to become familiar with a number of useful utilities which will allow you to properly understand the network capabilities of your computer system.

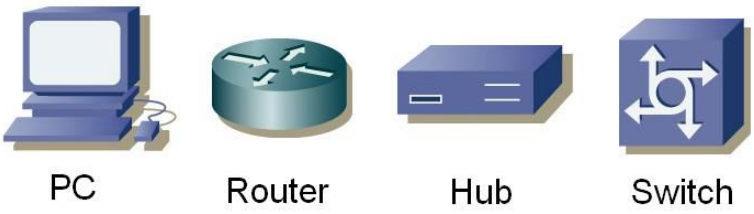
At the end of the lesson you should have a basic knowledge of:

- the concepts of networks
- IP addresses
- ports and protocols.

# 3.2 Basic concepts of networks

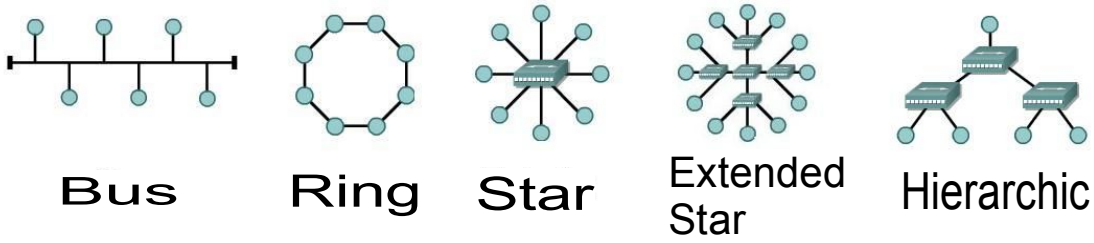
## 3.2.1 Devices

In order to understand the explanation of protocols and ports, it is necessary for you to become familiar with the icons that represent the most common devices that are seen in the basic schemes. These are:




## 3.2.2 Topologies

With these devices, local area networks (or LANs) can be created. In a LAN, computers can share resources, such as hard drives, printers and internet connections, and an *administrator* can control how these resources are shared. When a LAN is being designed, it is possible to choose any of the following physical topologies:




In a *bus* topology, all the computers are connected to a single means of transmission, and each computer can communicate directly with any of the others. In the *ring* configuration, each computer is connected to the following one, and the last one to the first, and each computer can only communicate directly with the two adjacent computers. In the *star* topology, none of the computers are directly connected with others. Instead they are connected through a central point and the device at that central point is responsible for relaying information from computer to computer. If several central points are connected to each other, an *extended star* topology is obtained. In a star or extended star topology, all the central points are *peers,* that is, each exchanges information on an equal basis. However, if you connect two star or extended star networks together using a central point which controls or limits the exchange of information between the two networks, then you have created a single, *hierarchical* network topology.

# 3.3 TCP/IP model

## 3.3.1 Introduction

TCP/IP was developed by the DoD (Department of Defense) of the United States and DARPA (Defense Advanced Research Project Agency) in the 1970s. TCP/IP was designed to be an open standard that anyone could use to connect computers together and exchange information between them. Ultimately, it became the basis for the Internet.

## 3.3.2 Layers

The TCP/IP model defines four totally independent layers into which it divides the process of communication between two devices. The layers through which it passes information between two devices are:


### 3.3.2.1 Application

The application layer is the layer nearest the end user. This is the layer that is in charge of translating data from applications into information that can be sent through the network.

The basic functions of this layer are:

- Representation
- Codification
- Dialog Control
- Application Management

### 3.3.2.2 Transport

The transport layer establishes, maintains and finishes virtual circuits for information transfer. It provides control mechanisms for data flow and allows broadcasting, and it provides mechanisms for the detection and correction of errors. The information that arrives at this layer from the application layer is divided into different segments. Information that comes to the transport layer from the internet layer is delivered back to the application layer through *ports*. (See **Section 3.3.5 Ports** for details on ports.)

The basic functions of this layer are:

- Reliability
- Flow Control
- Error Correction
- Broadcasting

### 3.3.2.3 Internet

This layer divides the segments of the transport layer into packets and sends the packets across the networks that make up the Internet. It uses *IP*, or *internet protocol* addresses to determine the location of the recipient device. It does not ensure reliability in the connections, because this is already taken care of by the transport layer, but it is responsible for selecting the best route between the originating device and the recipient device.

### 3.3.2.4 Network Access

This layer is in charge of sending information at both the LAN level and the physical level. It transforms all the information that arrives from the superior layers into basic information (bits) and directs it to the proper location. At this level, the destination of the information is determined by the *MAC*, or *media access control*, address of the recipient device.

## 3.3.3 Protocols

To be able to send information between two devices, both must speak the same language. This language is called the *protocol*.

The protocols that appear in the application layer of the TCP/IP model are:

- File Transfer Protocol (FTP)
- Hypertext Transfer Protocol (HTTP)
- Simple Mail Transfer Protocol (smtp)
- Domain Name Service (DNS)
- Trivial File Transfer Protocol (TFTP)

The protocols of the transport layer are:

- Transport Control Protocol (TCP)
- User Datagram Protocol (UDP)

The protocols of the internet layer are:

- Internet Protocol (IP)

The protocol most often used in the network access layer is:

- Ethernet

The protocols listed above and their associated ports will be described in the following sections.

### 3.3.3.1 Application layer protocols

*FTP* or *file transfer protocol* is used for the transmission of files between two devices. It uses TCP to create a virtual connection for the control of information, then creates another connection to be used for the delivery of data. The most commonly used ports are 20 and 21.

*HTTP* or *hypertext transfer protocol* is used to translate information into web pages. This information is distributed in a manner similar to that used for electronic mail. The most commonly used port is 80.

*SMTP* or *simple mail transfer protocol* is a mail service that is based on the FTP model. It transfers electronic mail between two systems and provides notifications of incoming mail. The most commonly used port is 25.

*DNS* or *domain name service* provides a means to associate a *domain name* with an ip address. The most commonly used port is 53.

*TFTP* or *trivial file transfer protocol* has the same functions as FTP but uses UDP instead of TCP. (See **Section 3.3.3.2** for details on the differences between UDP and TCP.) This gives it more speed, but less security and trustworthiness. The most commonly used port is 69.

### 3.3.3.2 Transport layer Protocols

There are two protocols which can be used by the transport layer to deliver information segments.

*TCP* or *transmission control protocol* establishes a logical connection between the final points of the network. It synchronizes and regulates the traffic with what is known as the "Three Way Handshake". In the "Three Way Handshake," the originating device sends an initial packet called a *SYN* to the recipient device. The recipient device sends an acknowledgment packet, called a *SYN/ACK*. The originating device then sends a packet called an *ACK*, which is an acknowledgment of the acknowledgment. At this point, both the originating device and the recipient device have established that there is a connection between the two and both are ready to send and receive data to and from each other.

*UDP* or *user datagram protocol* is a transport protocol which is not based on a connection. In this case, the originating device sends packets without warning the recipient device to expect these packets. It is then up to the recipient device to determine whether or not those packets will be accepted. As a result, UDP is faster that TCP, but it cannot guarantee that a packet will be accepted.

### 3.3.3.3 Internet layer Protocols

*IP* or *internet protocol* serves as a universal protocol to allow any two computers to communicate through any network at any time. Like UDP, it is *connectionless,* because it does not establish a connection with the remote computer. Instead, it is what is known as a *best effort* service, in that it will do whatever is possible to ensure that it works correctly, but its reliability is not guaranteed. The Internet Protocol determines the format for the packet headers, including the IP addresses of both the originating and the recipient devices.

## 3.3.4 IP Addresses

A domain name is the web address that you normally type into a web browser. That name identifies one or more IP addresses. For example, the domain name microsoft.com represents about a dozen IP addresses. Domain names are used in URLs to identify particular Web pages.

For example, in the URL http://www.pcwebopedia.com/index.html, the domain name is pcwebopedia.com.

Every domain name has a suffix that indicates which top level domain (TLD) it belongs to. There are only a limited number of such domains. For example:

.gov - Government agencies

.edu - Educational institutions

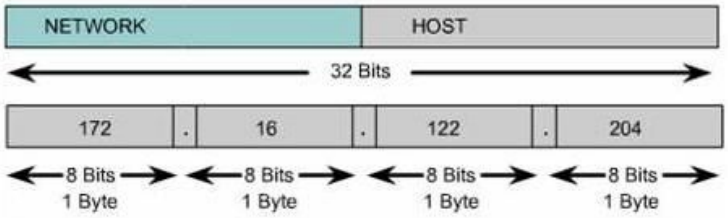
.org - Organizations (nonprofit)

.com - Commercial Business

.net - Network organizations

Because the Internet is based on IP addresses, not domain names, every Web server requires a Domain Name System (DNS) server to translate domain names into IP addresses.

IP Addresses are the identifiers that are used to differentiate between computers and other devices that are connected to a network. Each device must have a different IP address, so that there are no problems of mistaken identity within the network. IP addresses consist of 32 bits that are divided in four 8 bit octets which are separated by dots. Part of the IP address identifies the network, and the remainder of the IP address identifies the individual computers on the network.




There are both public and private IP addresses. Private IP addresses are used by private networks that have no connection with outside networks. IP addresses within a private network should not be duplicated within that network, but computers on two different – but unconnected – private networks could have duplicated IP addresses. The IP addresses that are defined by IANA, the Internet Assigned Numbers Authority, as being available for private networks are:

10.0.0.0 through 10.255.255.255

172.16.0.0 through 172.31.255.255

192.168.0.0. through 192.168.255.255

IP addresses are divided into classes based on what portion of the address is used to identify the network and what portion is used to identify the individual computers.

Depending on the size assigned to each part, more devices will be allowed within the network, or more networks will be allowed. The existing classes are:

| 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 |

- Class A: The first bit is always zero, so this class includes the addresses between 0.0.0.0 and 126.255.255.255. Note: the addresses of 127.x.x.x are reserved for the services of loopback or localhost.

- Class B: The first two bits of the first octet are '10', so this class includes the addresses between 128.0.0.0 and 191.255.255.255.

- Class C: The first three bits of the first octet are '110', so this class includes the addresses between 192.0.0.0 and 223.255.255.255.

- Class D: The first four bits of the first octet are '1110', so this class includes the addresses between 224.0.0.0 and 239.255.255.255. These addresses are reserved for group multicast implementations.

- The remaining addresses are used for experimentation or for possible future allocations.

At this time, the classes are not used to differentiate between the part of the address used to identify the network and the part used to identify the individual devices. Instead, a *mask* is used. In the mask, a '1' binary bit represents the part containing the network identification and a '0' binary bit represents the part that identifies the individual devices. Therefore, to identify a device, in addition to the IP address, it is necessary to specify a network mask:

| IP: 172.16.1.20 |
|---|
| Mask: 255.255.255.0 |

IP addresses 127.x.x.x are reserved to be used as loopback or local host addresses, that is, they refer directly back to the local computer. Every computer has a local host address of 127.0.0.1, therefore that address cannot be used to identify different devices. There are also other addresses that cannot be used. These are the *network address* and the *broadcast address*.

The *network address* is an address in which the part of the address which normally identifies the device is all zeros. This address cannot be used, because it identifies a network and can never be used to identify a specific device.

| IP: 172.16.1.0 |
|---|
| Mask: 255.255.255.0 |

The *broadcast address* is an address in which the part of the address which normally identifies the device is all ones. This address cannot be used to identify a specific device, because it is the address that is used to send information to all of the computers that belong to the specified network.

| IP: 172.16.1.255 |
| --- |
| Mask: 255.255.255.0 |

## 3.3.5 Ports

Both TCP and UDP use *ports* to exchange information with applications. A *port* is an extension of an address, similar to adding an apartment or room number to a street address. A letter with a street address will arrive at the correct apartment building, but without the apartment number, it will not be delivered to the correct recipient. Ports work in much the same way. A packet can be delivered to the correct IP address, but without the associated port, there is no way to determine which application should act on the packet.

Once the ports have been defined, it is possible for the different types of information that are sent to one IP address to then be sent to the appropriate applications. By using ports, a service running on a remote computer can determine what type of information a local client is requesting, can determine the protocol needed to send that information, and maintain simultaneous communication with a number of different clients.

For example, if a local computer attempts to connect to the website www.osstmm.org, whose IP address is 62.80.122.203, with a web server running on port 80, the local computer would connect to the remote computer using the *socket address :*

**62.80.122.203:80**

In order to maintain a level of standardization among the most commonly used ports, IANA has established that the ports numbered from 0 to 1024 are to be used for common services. The remaining ports – up through 65535 – are used for dynamic allocations or particular services.

The most commonly used ports – as assigned by the IANA – are listed here:

| Port Assignments | | |
| --- | --- | --- |
| **Decimals** | **Keywords** | **Description** |
| 0 | | Reserved |
| 1-4 | | Unassigned |
| 5 | rje | Remote Job Entry |
| 7 | echo | Echo |
| 9 | discard | Discard |
| 11 | systat | Active Users |
| 13 | daytime | 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 |

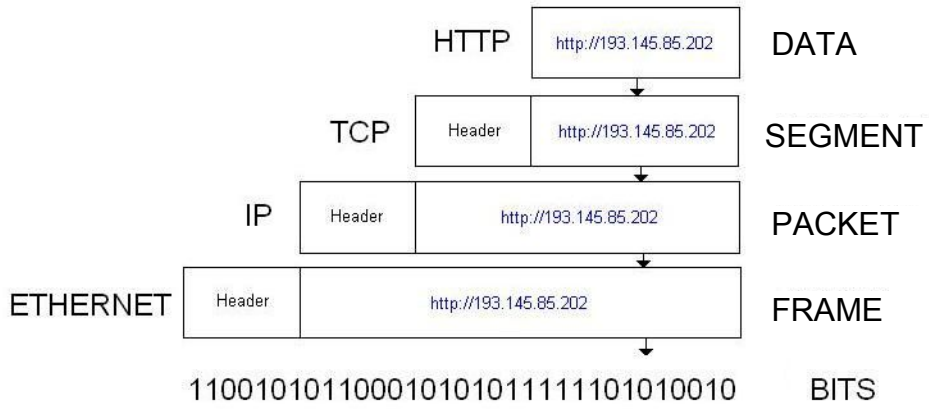
| Port Assignments | | |
|---|---|---|
| **Decimals** | **Keywords** | **Description** |
| 23 | telnet | Telnet |
| 25 | smtp | Simple Mail Transfer |
| 37 | time | Time |
| 39 | rlp | Resource Location Protocol |
| 42 | nameserver | Host Name Server |
| 43 | nicname | Who Is |
| 53 | domain | Domain 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 |
| 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 |

You can also refer to the Web page: *http://www.isecom.info/cgi-local/protocoldb/browse.dsp* for more detailed information on ports.

## 3.3.6 Encapsulation

When a piece of information – an e-mail message, for example – is sent from one computer to another, it is subject to a series of transformations. The application layer generates the data, which is then sent to the transport layer. The transport layer takes this information and adds a header to it. This header contains information, such as the IP addresses of the originating and recipient computers, that explains what must be done to the data in order to get it to the appropriate destination. The next layer adds yet another header, and so on. This recursive procedure is known as *encapsulation*.

Each layer after the first makes its data an encapsulation of the previous layer's data, until you arrive at the final layer, in which the actual transmission of data occurs. The following figure explains encapsulation in a graphic form:




When the encapsulated information arrives at its destination, it must then be de-encapsulated. As each layer receives information from the previous layer, it removes the unneeded information contained in the header placed there by the previous layer.

# 3.4 Exercises

## 3.4.1 Exercise 1: Netstat

Netstat

The Netstat command allows you to see the state of the ports on a computer. In order to execute it, you must open an MS-DOS window and type:

netstat

In the MS-DOS window, you will then see a list of the established connections. If you want to see the connections displayed in numeric form, type:

netstat - n

To see the connections and the active ports, type:

netstat - an

To see a list of other options, type:

netstat - h

In the Netstat output, the second and third columns list the local and remote IP addresses being used by the active ports. Why are the addresses of the remote ports different from the local addresses?

Next, using a web browser, open this web page:

http://193.145.85.202

then return to the MS-DOS prompt and run Netstat again. What new connection (or connections) appear?

Open another web browser and go to this web page:

http://193.145.85.203

Return to the MS-DOS prompt and run Netstat:

- Why does the protocol HTTP appear in several lines?

- What differences exist between each one of them?

- If there are several web browsers open, how does the computer know which information goes to which browser?

## 3.4.2 Exercise 2: Ports and Protocols

In this lesson, you learned that ports are used to differentiate between services.

Why is it that when a web browser is used, no port is specified?

What protocols are used?

Is it possible that one protocol gets used in more than one instance?

## 3.4.3 Exercise 3: My First Server

To perform this exercise, you must have the *Netcat* program. If you do not have it, you can download it from the page:

http://www.atstake.com/research/tools/network_utilities/

Once you have Netcat installed, open an MS-DOS window. Change to the Netcat directory and type:

nc - h

This displays the options that are available in Netcat. To create a simple server, type:

nc - l - p 1234

When this command executes, port 1234 is opened and incoming connections are allowed. Open a second MS-DOS window and type:

netstat – a

This should verify that there is a new service listening on port 1234. Close this MS-DOS window.

To be able to say that a server has been implemented, you must establish a client association. Open an MS-DOS window and type:

nc localhost 1234

With this command, a connection is made with the server that is listening to port 1234. Now, anything that is written in either of the two open MS-DOS windows can be seen in the other window.

Create a file named 'test', that contains the text, "Welcome to the Hacker Highschool server!" In an MS-DOS window, type:

nc - l - p 1234 > test

From another MS-DOS window, connect to the server by typing:

nc localhost 1234

When the client connects to the server, you should see the output of the file, 'test'.

To close the service, switch to the MS-DOS window in which it is running and press CTRL-C.

What protocol has been used to connect with the server?

Does Netcat allow you to change this? If so, how?

# Further Reading

You can find more information on ports and protocols by looking at the following links:

http://www.oreilly.com/catalog/fire2/chapter/ch13.html

http://www.oreilly.com/catalog/puis3/chapter/ch11.pdf

http://www.oreilly.com/catalog/ipv6ess/chapter/ch02.pdf

http://info.acm.org/crossroads/xrds1-1/tcpjmy.html

http://www.garykessler.net/library/tcpip.html

http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ip.htm

http://www.redbooks.ibm.com/redbooks/GG243376.html

Port Number references:

http://www.iana.org/assignments/port-numbers

http://www.isecom.info/cgi-local/protocoldb/browse.dsp