Ports and Protocols

*Resources used:*

*DionTraining UDemy Course - “Ports and Protocols”*

*Professor Messer on Youtube - “Common Network Ports”*

*Comptia A+ Exam Textbook*

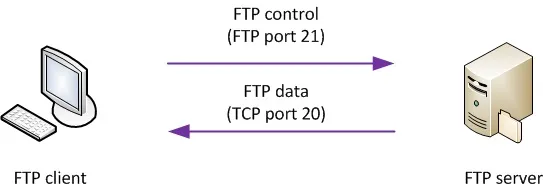
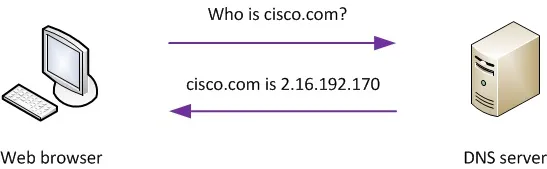
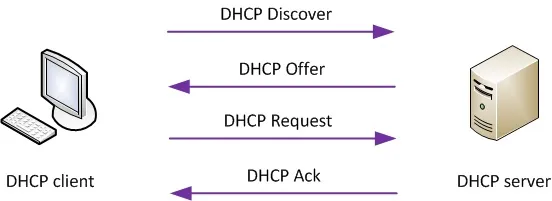
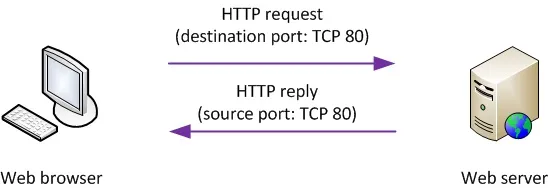
*LiveOverflow on Youtube - “What is a Protocol?”*

*How to Network Article - “Port Numbers and Network Protocols”:* [*https://tinyurl.com/rybftjme*](https://tinyurl.com/rybftjme)

Vocabulary

1. Port: Logical communication endpoint that exists on a computer or server
   1. Ex: Web server has port80 is open (“inbound port”)
      1. Inbound Port: logical communication opening on a server that is listening for a connection from a client “your computer/server is listening for a connection”
      2. Outbound Port: logical communication opening created on a client in order to call out to a server that is listening for a connection “computer sends out booty call for whatever server is listening”
         1. A computer wants to access a website? It’s going to make an outbound request to that web server using a high number port ID (he used example like 51223) (I didn’t know ports could go that high)
   2. Port Numbers are also used to track the session associated with the protocol.
2. Well-Known Ports
   1. Ports 0 to 1023 are considered well-known and assigned by IANA
      1. Commonly used and recognized ports
      2. A well-known port number helps to uniquely identify a Web server, for example. Even though the Web server might use a random port number, it usually uses port number 80 as a convention so other devices can know how to reach it. If the service used a non-standard port number, a client device could reach it only if it was specifically informed about the port number used. Well-known port numbers are usually those below 1024.
3. Registered Ports (“Temporary Ports”, “Ephemeral Ports”)
   1. Ports 1024 to 49151 are considered registered and are usually assigned to proprietary protocols
      1. Have to be used by vendors for their own proprietary protocols and each vendor is going to register them with IANA prior to using them
         1. EX: Microsoft has a SQL server, which uses Port 1433
         2. EX: RDP, Port 3389
   2. Usually used as source ports in a two-way communication process. They are determined in real-time by the client workstation and are usually numbers above 1024.
4. Dynamic or Private Ports
   1. Ports 49152 to 65536 can be used by any application without being registered with IANA
      1. This is usually used by your client (like previous web server example above)
      2. Whenever your client wants a temporary outbound connection, this is the range they will use
         1. Commonly used for GAMING, INSTANT MESSAGING AND CHAT

**WELL KNOWN PORT TIME**

* **File Transfer Protocol (FTP); Ports 20, 21 -** Provides insecure file transfers between client and server on a computer network. Data transmitted in the clear; “no encryption”. Not safe for transferring sensitive files.
  + tcp/20 (active mode data) for data transfers; tcp/21 (control) used to control data transfer
  + Authentication with a username and password
  + Full-featured functionality, such as list, add, delete, etc
  + A user who wants to access an FTP server usually uses a dedicated FTP client application (like FileZilla) that initiates a connection to the server. The FTP client sends a session request on port 21, and after the session to the server is initialized data is transferred using port 20.
  + 
* **Secure Shell (SSH); Port 22 -** Provides secure remote control of another machine using a text-based environment. Best known for its use as a remote login capability and it is a cryptographic network protocol. Uses encryption, so it's safe to use over the internet.
  + Encrypted communication link, tcp/22
  + Command line remote control service - much like Telnet (looks and acts the same as telnet)
  + One of the most commonly used Telnet and SSH software client is a free software utility called PuTTY, which allows a user to initiate a Telnet or SSH session with a network device and obtain control of the command line interface (CLI).
* **Secure File Transfer Protocol (SFTP); Port 22 -** Encrypted FTP.
* **Telnet; Port 23 -** Provides insecure remote control of another machine using a text-based environment. Works a lot like how SSH does; Telnet came out many years before SSH - but telnet is NOT secure. “Telnet provides us with remote access via the command prompt.”
  + tcp/23 - “text based front end”; main difference between SSH and Telnet is that all data on Telnet will be sent in the “clear” - dangerous, dangerous, dangerous!
  + One of the most commonly used Telnet and SSH software client is a free software utility called PuTTY, which allows a user to initiate a Telnet or SSH session with a network device and obtain control of the command line interface (CLI).
* **Simple Mail Transfer Protocol (SMTP); Port 25 -** Provides the ability to send emails over the network.
  + Millions of email servers on the interwebs, and so many of them use SMTP to communicate with each other
  + Server to server email; also used to send mail from a device to a mail server - so client to mail server as well, not just mail server to mail server
  + tcp/25
  + Common to use SMTP to SEND emails, but use another protocol to receive mail
* **Domain Name Service (DNS); Port 53 -** Converts domain names to IP addresses, and IP addresses to domain names.
  + udp/53
  + Usually multiple DNS servers are in production - very critical resources
  + In order to communicate with the Web server, the client must have its IP address, and this is where the DNS service comes into play by translating the name into a Layer 3 address.The client computer knows where to send the DNS request because it is configured with a DNS server address, which is either entered manually or received via DHCP.
  + The DNS server listens for requests on port 53 (both TCP and UDP).
  + The DNS service doesn’t work using a single server because it would not be capable of managing all the name-to-IP mappings on the Internet. Instead, DNS uses a collection of servers that work together to provide name-to-IP resolution.
  + 
* **Dynamic Host Configuration Protocol (DHCP); Ports 67, 68 -** Automatically provides network parameters, such as assigned IP address, subnet mask, default gateway and the DNS server. Helps simplify the administration of networks.
  + Receive wifi from a coffee shop or hotel? It was given to you by the DHCP! IP addresses are assigned in real time from a pool; each system is given a lease and must review at set intervals. “Only using this IP address for a certain amount of time, then the IP goes back into the pool for it to be used.”
  + Automated configuration of IP address, subnet mask, gateway, etc…requires a DHCP server
  + udp/67, udp/68
  + DHCP reservations - so certain devices can receive the same IP every time they’re booted up via recognizing their MAC address
  + 
* **Hypertext Transfer Protocol (HTTP); Port 80 -** Used for insecure web browsing.
  + Main way of communication between our web browsers and the web servers
  + When a client PC (Web browser) makes a request, the type of request will determine the destination port number used. Websites are often listening on TCP port 80, a well-known port identifying the HTTP protocol, and they respond to request packets using TCP 80 as the source port, as depicted in Figure 5.1 below. This makes life easier for Web clients because if there were no standards regarding this approach, the communication between users and servers would be random.
  + 
* **Post Office Protocol Version Three (POP3); Port 110** - Used for receiving inbound or incoming email. Uses a ‘store and forward’ method of communication. Older method of receiving emails, but still heavily used by many people today.
  + To receive emails from an email server
  + tcp/110
  + Wasn’t built for multiple clients (which is troublesome for modern day usage; since we all access for example, gmail on our computers and our phones)
* **Network Basic Input/Output System (NetBIOS); Port 137, 139 -** Used for file or printer sharing in a Windows network. Provides service for allowing applications on a separate computer to communicate over LAN to share files and printers.
  + udp/137 name services (to find devices on your network)
  + tcp/139 session services (to set up a session and transfer data between devices)
  + Most modern devices don’t use NetBIOS; will use SMB
* **Internet Mail Application Protocol (IMAP); Port 143 -** Newer method of retrieving incoming emails which improves upon the older POP3. Server keeps everything synchronized across multiple devices.
  + tcp/143
  + Includes management of email inbox from multiple clients (suck it, POP3)
  + IMAP is a more evolved protocol that was developed in the last few years, as it offers more functionality and flexibility. It can manage all email messages directly on the mail server, unlike POP3 which must download the messages in order to manage them locally. IMAP offers access to emails from everywhere, not just on the local workstation, and it can also transparently synchronize the local email client with the e-mail server. Although IMAP offers extra functionality, all of these features make IMAP use more resources on the server.
* **Simple Network Management Protocol (SNMP);** Ports 161, 162 - Used to collect data about network devices and monitor their status. Collect and organize information about all the managed devices on an IP network (routers, switches, VoIP phones, etc). Can change the device’s behavior and monitor the uptime, downtime and other states of any given device.
  + Gather statistics from network devices
  + udp/161 to perform queries
  + udp/162 send alerts “traps”
  + V1 - in the clear, V2 - encrypted, V3 - additional security, authentication and encryption of all SNMP data
  + SNMP has evolved during the years and has now reached version 3 (SNMPv3). Network designers and engineers should demand that every environment use SNMPv3, not the older, unsecure SNMP versions (1 and 2), because of the advanced security features it presents.
* **Lightweight Directory Access Protocol (LDAP); Port 389 -** Used to provide directory services to your network. Ex: looking someone up in Outlook - it’s a directory.
  + tcp/389
  + Commonly used in Microsoft Active Directory
* **Hypertext Transfer Protocol Secure (HTTPS); Port 443 -** Use SSL or TLS method to encrypt website browsing.
  + The encryption used by the HTTPS protocol is accomplished using the Transport Layer Security/Secure Sockets Layer (TLS/SSL) mechanism.
* **Server Message Block (SMB); Port 445 -** Used for Windows file and printer sharing services. Oftentimes operates with NetBIOS. NetBIOS handles the authentication of files, and SMB handles the transfer.
  + Many operating systems have their own method of transferring files and information between devices using the same OS. Windows uses SMB to do so.
  + CIFS (another name). If you’re using an older Windows machine, you’re probably using NetBIOS over TCP/IP
* **Remote Desktop Protocol (RDP); Port 3389 -** Provides graphical remote control of another client or server. Proprietary protocol developed by Microsoft. Similar to telnet and SSH, but provides a graphical user interface (we can actually see what we’re doing, versus just using a command line).
  + tcp/3389
  + Can connect to an entire desktop or just an application
  + There are clients on almost any OS, not just Windows

Brain Dump Notes

Be aware of the “openings” you have on your network/computer. Server will keep an open inbound port open for the next user who wants to use it.

Services have port numbers that they use so other devices can communicate and use those services. Both the server and the client need to know what port number to use to communicate.

Ports can be assigned any number 0-65,655.

Protocol is a set of conventional rules…set of rules of how computers or programs should behave. Think about this; when we load up a website, for example, Neopets, the web browser has to send a request to the Neopet’s server - this is a string of text that looks like mumbo jumbo, like, GET/test HTTP/1.1, blah blah blah. But the server receives and UNDERSTANDS that gibberish, and sends a response back to the web browser in the same ‘language’. This understanding between the web browser and the server is thanks to the HTTP protocol! It’s the ‘standard’ they can recognize and use. You can even find the HTTP rulebook online!

TCP - 3 way handshake

We’re on one computer, using one internet connection- so when the computer receives data, which application should receive this data? That’s what the ports are for! Why do we have the complex sequence of packets back and forth? Why not just send data with additional port information? HEY! That’s what UDP is for! No 3 way handshake for UDP. Also, UDP is super old. Hadn’t needed an update since 1980 (JESUS LMAO), this is because UDP is SUPER simple compared to TCP. If we were to use UDP packets for HTTP, we would send out a UDP request and…wait…and wait…”Hey, did I send anything out? Does the server exist?” and if we did receive a response back, how can we be sure it’s the correct server? How can we be sure that it’s not a hacker spoofing a UDP response packet? That’s why TCP is handy, because it would use the 3-way handshake SYN-ACK to confirm all that information for us.

The TCP header contains port numbers but they are really being used by the Session Layer of the OSI reference model (i.e., bundled in the Application Layer in the TCP/IP protocol suite). Even though the same port number can be used in both TCP and UDP, the services they identify can be completely different. For example, TCP port 80 identifies a different service/protocol than UDP port 80 does.

Port numbers allow network-connected devices to transmit data using transmission protocols. Without port numbers, network traffic would be indecipherable.