Copyright Notice:  
All images and screenshots used in this document are the property of CBT Nuggets. Used for educational purposes only. This document is licensed under a Creative Commons Attribution 4.0 International License.

All Transport Layer Protocols: **ICMP** #1, **TCP** #6, **UDP** #17, IPSec: **ESP** #50, **AH** #51, **IKE** #500 (UDP)

Anycasts are useful in Load Balancing groups, like how DNS Servers can be all the same IP address in a group, so like the closest to route to- using routing protocols like RIP, OSPF, or BGP.

**In this lab, I learned to recognize network traffic on Wireshark using all of the common protocols on the application layers corresponding to their network and transport layer ports:**

FTP (insecure, unpreferred): The application that needs to transfer a file uses the 49152-65535 range of private/unassigned dynamic ports to connect to the FTP Server at TCP port 21 to establish a session for data transfer to whichever port the OS selects. A SYN, SYN-ACK, ACK process takes place between the FTP Server and Client at whichever used dynamic/private port. TCP Port 20 is then used for transferring data/files between the client and the server, under the terms established in the session initiation, which included what dynamic or other assigned ports on the client could be used to accept the reception of FTP data/files, likely different from the one used for session initiation. Network Traffic was then observed on Wireshark reflecting file transfers through these protocol/ports.

SFTP: TCP Port 22 establishes the FTP session and data transfer over a Secure Shell (SSH) session at the same port as SSH: ‘TCP Port 22’.

Telnet (plaintext transfer, usually used for serial connections in some routers, but undesirable to use for its insecurity): TCP Port 23

SMTP: TCP Port 25. This is the unencrypted way to send mail through any Mail application.

SSMTP: TCP Port 445

DNS: UDP Port 53. The operating system (windows, Linux, etc.) on the Endpoint/User/Server requesting the DNS Server record from the DNS Server will use available Dynamic/Private Ports that are not assigned between 49152-65535 range of ports to connect to the assigned DNS Server, which is listening at port 53 to each of those connections for the respective applications in that port range.

DHCP: UDP Port 67/68- This assigns the Subnet Mask, DNS Servers, and Default Gateways to hosts(mostly just ipv4)- depending on the configuration. At port 68, the hosts send a DHCP Discover Broadcast to the network, and when the DHCP receives the DISCOVER Packet at port 67, it checks to make sure the client is allowed to get a DHCP Offer, along with other authentication on other ports- and then sends a DHCP Offer back to the sender on which IP Addresses, default gateway, and other information it can accept from said user/sender. When the sender establishes the DNS Servers, IP Address for itself, and default gateway, it will send a DHCP Request to the DHCP Server from its own port 68 to the DHCP Server at port 67. If the DHCP Server accepts that choice of provided options for that sender/user’s PC, it will send an ACK request out port 67 on its end to the receiving 68 on the user’s end, when the user is now free to change its IP address, Default Gateway, DNS Servers, or other information to the allowed values from now until the record refreshes again (Lease Time). There are smart ways to set Lease Time to best support the business model of devices going in and out of a particular room/department.

TFTP (Trivial FTP): UDP Port 69, oft used to perform direct out-of-band management on different brands of routers for patch updates on a Serial Connection like RS232 or USB. A bit insecure and undesirable for that purpose- but is still in occasional use.

HTTP: TCP Port 80. Used for requesting & retrieving web pages from a web server- the unencrypted version. The Source port is chosen on the host/client in the aforementioned dynamic private range that is also an unassigned port, and the application sends to port 80 on the web server GET and other requests for resources, oft of the web page variety. Port 80 sends it back to the respective application port or a set of ports that application is using for that particular web page (they’re large and complex, web pages today). Remote Desktop Connections over a network can be established on port 443, the secure version of port 80. 443 is quite popular…

HTTPS: TCP Port 443. Same use as HTTP, except, it’s encrypted, and uses Transport Layer Security (different versions like 1.1, 1.2, 1.4) Protocol- which allows for Certificates to be installed on user devices and the server end to do a secure exchange of information, and only to the users/server allowed and verified- a ‘Digital Signature’ process of sorts.

SIP (Session Initiation Protocol): UDP Port 5060/TCP Port 5061. Used in VoIP phones over the network.

NTP (Network Time Protocol): UDP Port 123, is a well known port that always sends/receives from 123 to 123 on all devices. It’s important for time to be kept on a network, especially for firewalls and security.

**Network/User/File Management & Monitoring Ports:**

SMB: TCP Port 445. Server Message Block is used between windows servers and machines to transfer files and communicate between themselves- including any printers connected to them- take Print Shares on the network for example.

LDAP: TCP Port 587. Used to retrieve Active directory object information (users, devices, etc.) and use it to authenticate users’ access to resources or devices (i.e. printers, scanners, fax) across the network. This is the unencrypted non-secure version.

SNMP: UDP Port 161/162. This protocol is used to manage the network, including configurations for network devices- ensuring all is going as it should. 161 is the port on the network devices and users that listen for instructions/automation/instrumentation from the SNMP Server, which sends those changes out and receives updates on port 162. The SNMP Agents are on each of those network devices and users.

Syslog: UDP Port 514- a log collection system for various PC, Network, and other devices- using a system of these severity levels (UDP is light on bandwidth & more ‘raw’):

1. **Emergency (0)**: Indicates that the system is unusable. Example: Kernel panic or complete system failure.
2. **Alert (1)**: Requires immediate action. Example: Loss of a primary internet connection.
3. **Critical (2)**: Denotes critical conditions. Example: Hardware failures like disk malfunctions.
4. **Error (3)**: Represents error conditions. Example: Application crashes or service interruptions.
5. **Warning (4)**: Highlights potential issues. Example: Low disk space or high CPU usage.
6. **Notice (5)**: Normal but significant events. Example: Successful system startup or shutdown.
7. **Information (6)**: General informational messages. Example: User login/logout events.
8. **Debug (7)**: Detailed debugging information. Example: Application-level debugging logs.

LDAPSecure: TCP Port 636

SQL Server: TCP Port 1433 – used for updating database information in SQL over the network.

RDP: TCP Port 3389- Remote Desktop connections over the network- 443 is more often used for this session, kind of like FTP over SSH, for security reasons. 443 is a more suitable port security and bandwidth wise on many networks, depending on configuration, which is easy to adjust, of course, down to the DSCP bits for bandwidth/performance, or the switches along the path… this could have use in medical facilities where users often need secure remote desktop support for things on PCs that can’t risk any insecure exposure, say, it could be a machine that has critical medical data or control over machines while people are getting operated on.

RADIUS: UDP Port 1812 for Authentication, UDP Port 1813 for Accounting

TACACS+: TCP Port 49

**Routing Protocols use different ports for sending route updates**:

RIPv2: UDP Port 520  
OSPF: Network Layer Protocol @ Port #89  
BGP: TCP Port 179 – used over the internet, by ISPs, and organization network Internet Boundaries  
EIGRP: Transport Layer Protocol @ Port #88 – ‘Runs on top of IP at layer 3’ – hence, why it’s both link-state and distance-vector. A Hybrid Protocol.

I did a project on implementing each of these routing protocols on a live network. Quite fascinating, really- the different uses for them in different places!