HACKING 310

Lesson 5: Network Security (Part 1)

Review

Lesson 4

- Active foot-printing
- Port scanning with NMAP and Netcat
- Banner grabbing, SNMP, DNS
- Web application enumeration

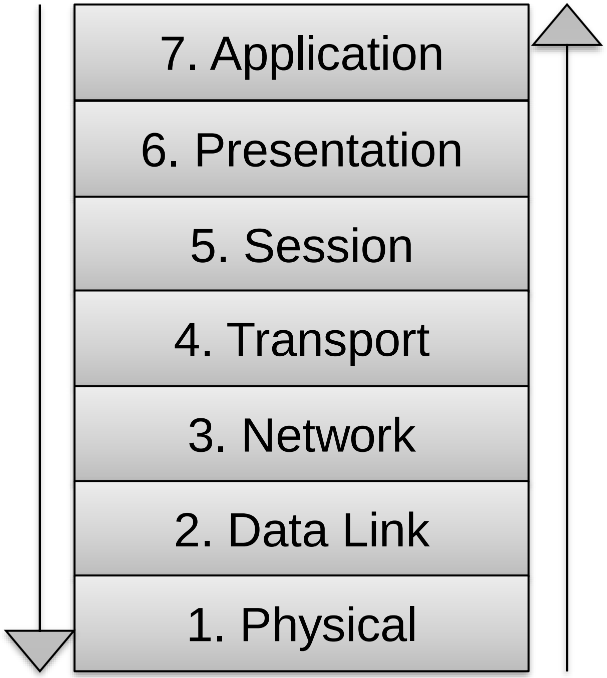
Objectives

After this lesson, you will be able to:

- Describe network technologies and their relevance to security.
  - OSI and TCP/IP models
  - TCP, UDP, IP, Ports, Sockets
  - Network services: DNS, DHCP
- Understand how NMAP scans work from a network level.
- Be able to perform network traffic capture with Wireshark or TCPDump.

OSI Model

- Open Systems Interconnection model
- Abstraction – each layer is encapsulated in the lower layers
- Layered organization of communications
- Useful for conceptualizing the flow of data
- Gives us a framework for talking about networking, how different parts of the stack interact


7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data Link
1. Physical

OSI Model

Application
- Communication related pieces of a software application

Presentation
- Format conversion, encryption, compression

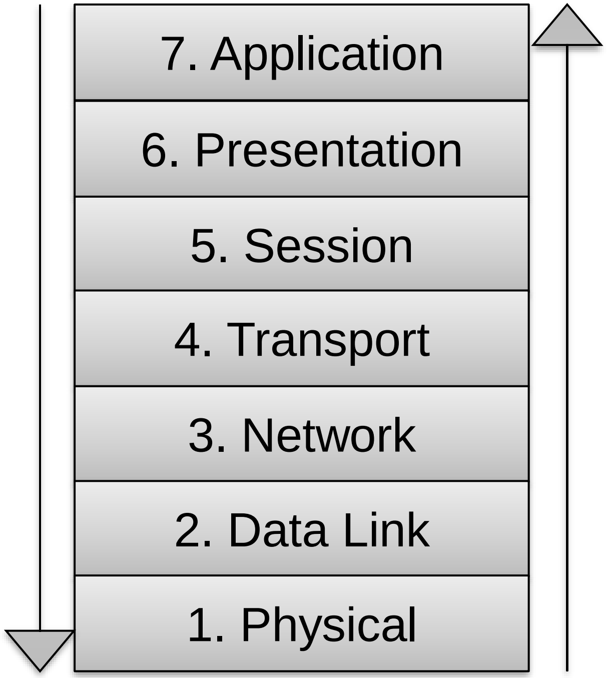
Session
- Sockets, RPC

Transport
- TCP, UDP

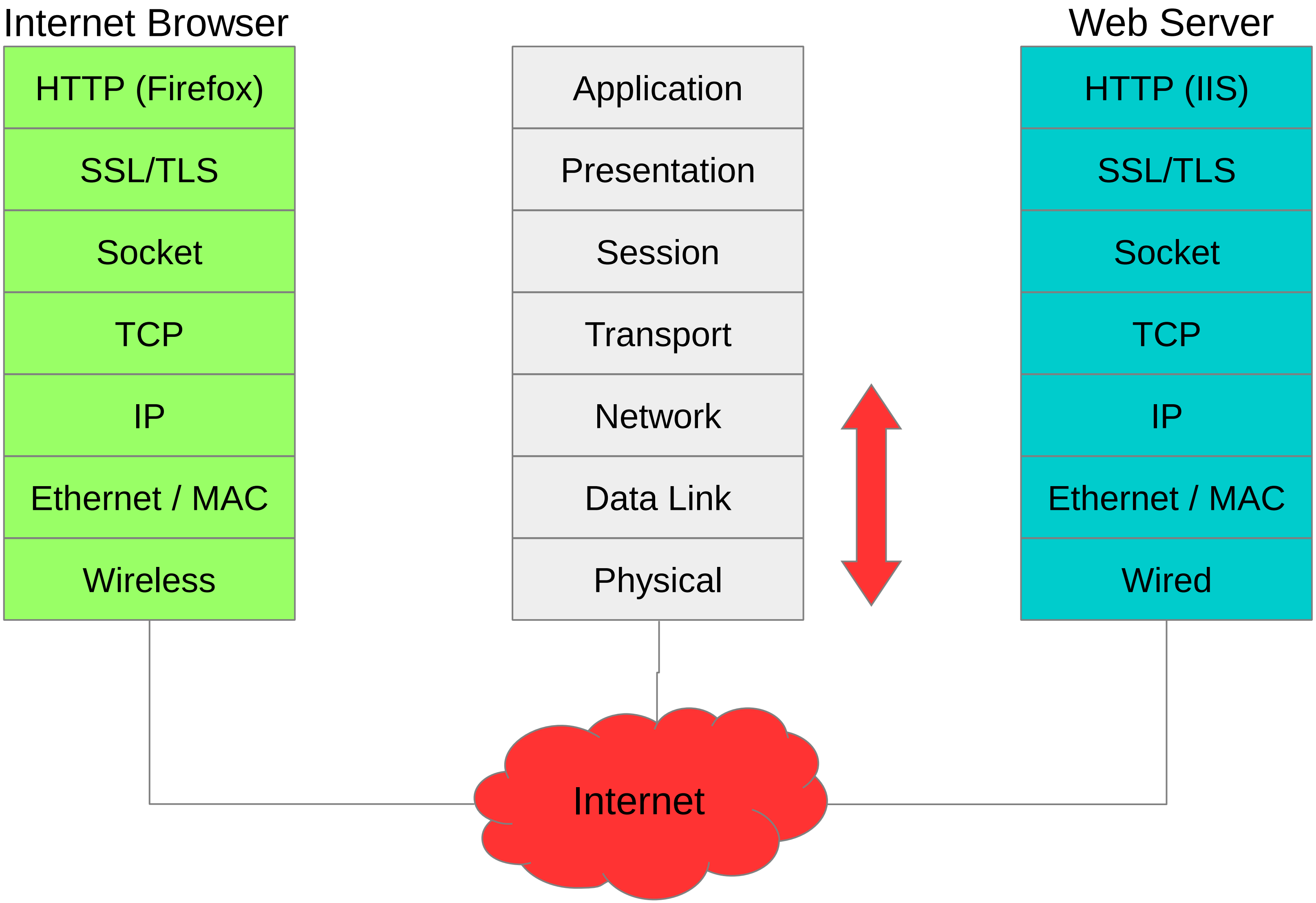
Network
- IP, routing

Data Link
- Ethernet, WiFi, MAC

Physical
- Topology, cabling, electrical



OSI Model



Internet Browser		Web Server
HTTP (Firefox)	Application	HTTP (IIS)
SSL/TLS	Presentation	SSL/TLS
Socket	Session	Socket
TCP	Transport	TCP
IP	Network	IP
Ethernet / MAC	Data Link	Ethernet / MAC
Wireless	Physical	Wired

TCP/IP Model

Application

- Encompasses Application/Presentation/Session functionality from OSI model.
- HTTP, FTP, DNS, SNMP, and other protocols operate here.

Transport

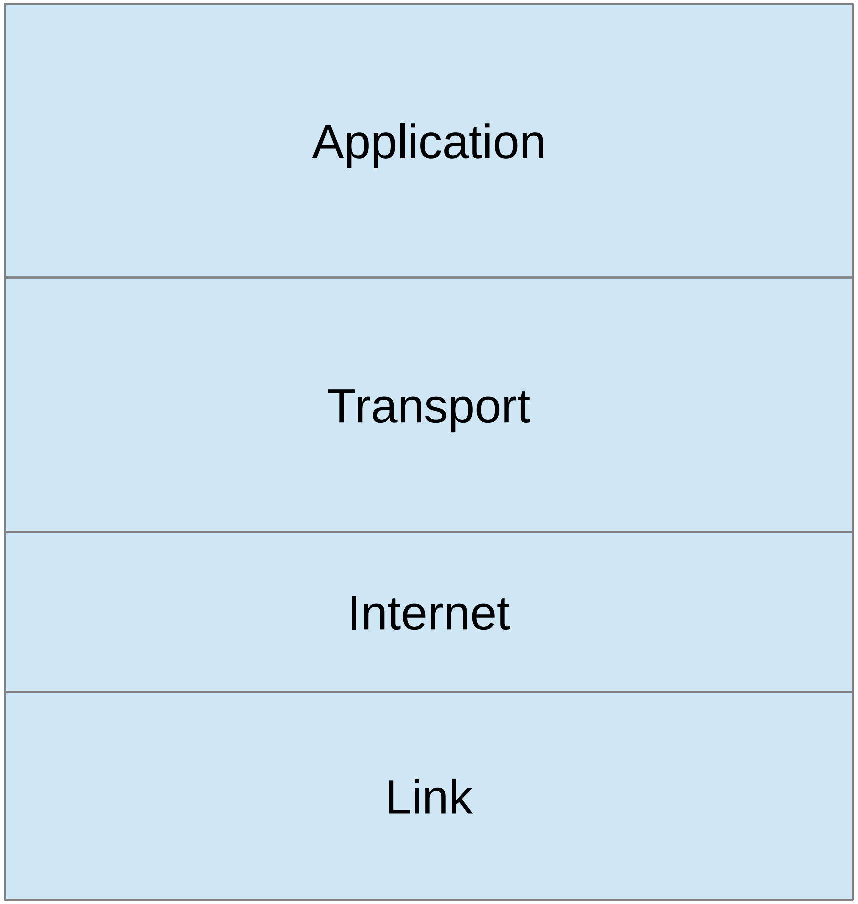
- Agnostic to structure of user data
- Responsible for getting data where it needs to go, independent of network below it.
- TCP, UDP, error control, segmentation

Internet

- Agnostic to transport structures
- Addressing (IP), packet routing

Link

- Local network connections
- MAC addressing, VPNs
- (Physical/Data Link from OSI)



Application

Transport

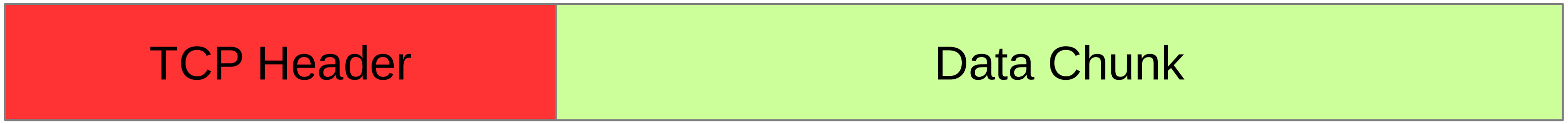
Internet

Link

Transmission Control Protocol (TCP)

Ensures that data sent over the underlying network is:

- Reliably delivered (re-transmit lost data)
- Received in the correct order
- Is not corrupted on it’s way to the recipient
- Operates on “segments”
- Application layer is broken up into chunks of a particular size
- A segment consist of a TCP header and data



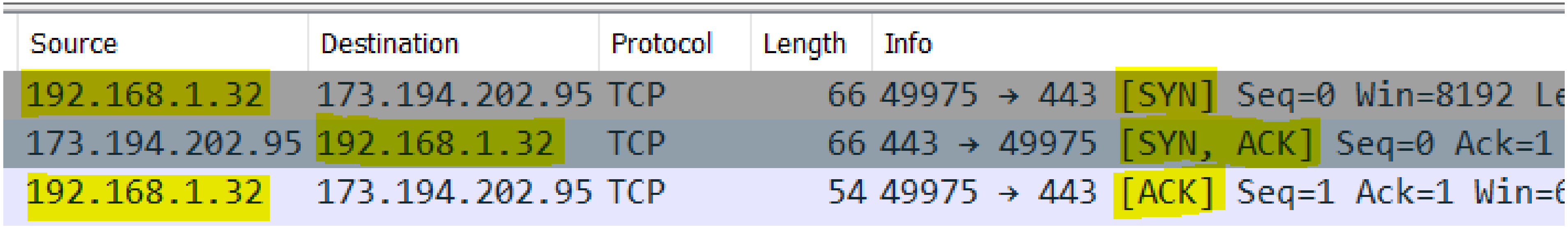
TCP Header

Data Chunk

Transmission Control Protocol (TCP)

- Connection Oriented
- 3-way handshake to establish connection
  1. SYN – Client sends SYN to server
  2. SYN-ACK – Server replies with SYN-ACK with chosen sequence number
  3. ACK – Client sends ACK to server

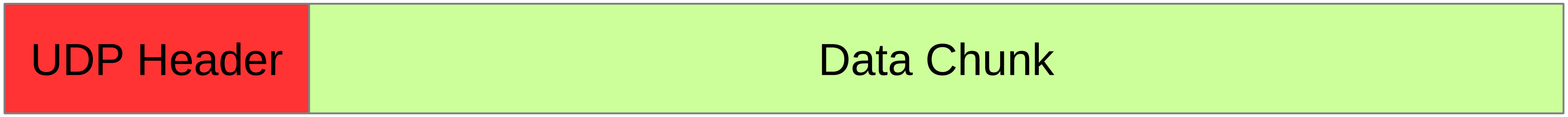
  TCP connection established!
- 4-way handshake to terminate connection
  1. FIN – Client sends FIN to close its side of the connection
  2. ACK – Server acknowledges FIN – **Client Disconnected!**
  3. FIN – Server sends FIN to close its side
  4. ACK – Client acknowledges FIN – **Server Disconnected!**



Source	Destination	Protocol	Length	Info
192.168.1.32	173.194.202.95	TCP	66	49975 → 443 [SYN] Seq=0 Win=8192 Le
173.194.202.95	192.168.1.32	TCP	66	443 → 49975 [SYN, ACK] Seq=0 Ack=1
192.168.1.32	173.194.202.95	TCP	54	49975 → 443 [ACK] Seq=1 Ack=1 Win=

User Datagram Protocol (UDP)

- TCP is a connection-oriented protocol where UDP is connectionless
- Application layer is exposed to network unreliability
- No ordering
- It does have checksums
- Greatly reduced overhead
- Good for broadcasting
- Operates on “datagrams”
- Very small header, 8 bytes



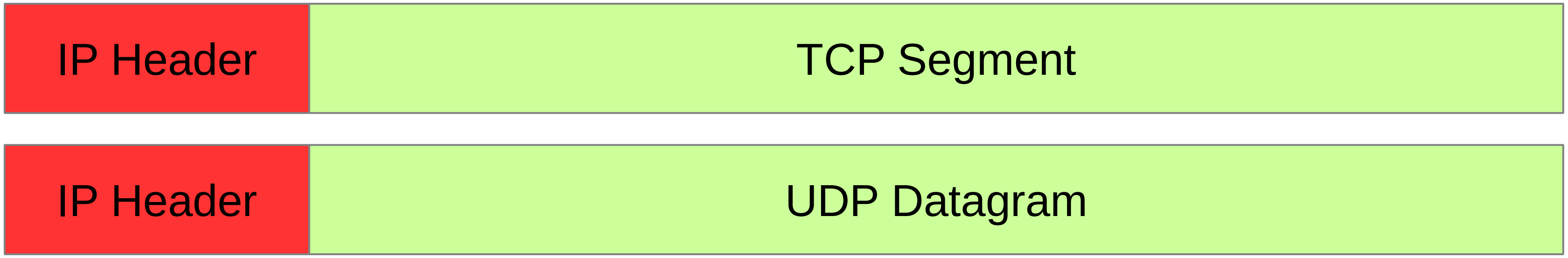
Internet Protocol (IP)

IP mainly provides two things:

- Structure of an IP datagram
    - This is different than a UDP datagram, though it may encapsulate a UDP datagram!
- Addressing
    - Address format
    - Private address ranges
    - Concept of subnetworks
- Two versions of IP are in use:
  - IPv4
  - IPv6

Internet Protocol (IP)

- Structure of an IP datagram
- Header (Different between IPv4 and IPv6)
  - Source/Destination addresses
  - Payload length
  - Checksum
  - Various flags and metadata
- Data from Transport layer



Internet Protocol (IP)

Addressing

- IPv4
  - 32 bits long
  - <8 bits>.<8 bits>.<8 bits>.<8 bits>
  - 0.0.0.0 through 255.255.255.255
  - Example: 10.152.20.5
- IPv6
  - 128 bits long
  - 8 groups of 16 bits, colon separated (<16 bits>:<16 bits>:<16 bits>: …)
  - Each 16-bit group is represented by 4 hex characters
  - Example: 2001:0:9d38:6abd:381a:19ca:3f57:fedf

Internet Protocol (IP)

Addressing (cont’d)

- Loopback (self)
  - IPv4: 127.0.0.1
  - IPv6: ::1
- Private Networks – RFC 1918
  - 192.168.0.0 – 192.168.255.255
  - 172.16.0.0 – 172.31.255.255
  - 10.0.0.0 – 10.255.255.255

Internet Protocol (IP)

Addressing (cont'd)

- Subnetting
  - Breaks a network's address space into smaller chunks
  - Mechanism is a subnet mask (ex: 255.255.255.0)
    - This describes how many bits of the IP are used for network addressing and how many are used for host addressing.

```
IP:           192.168.1.27 = 11000000.10101000.00000001.00011011
Subnet Mask: 255.255.255.0 = 11111111.11111111.11111111.00000000
```

- So what network is this machine on?
  - Bitwise AND the IP with the Subnet Mask

```
Network:       192.168.1.0 = 11000000.10101000.00000001.00000000
```

- Networks are also written in CIDR notation: 192.168.1.0/24
  - Network of 192.168.1.0 with subnet mask of 255.255.255.0

Internet Protocol (IP)

Addressing (cont'd)

- Conventions within a subnetwork
  - First IP in the subnet is reserved to address the subnetwork (192.168.1.0)
  - Second IP is typically the gateway router (192.168.1.1)
  - Last IP is typically the broadcast address (192.168.1.255)

Ports

- Identify an endpoint of some communication within a system
- Multiple concurrent connections from different programs on the same system
- Widely used concept in many protocols
- Represented by 16 bits; 0 - 65535 possible ports
  - Reserved: 0
  - Well Known: 1 - 1024
  - Ephemeral: 49152 - 65535
- A number of ports have been standardized as “well known port numbers”

Port	Service
TCP 20/21	FTP
TCP 22	SSH/SFTP
TCP 23	Telnet
TCP 25	SMTP
TCP/UDP 53	DNS
UDP 67/68	DHCP
TCP 80/443	HTTP/HTTPS
TCP/UDP 134/138/139/445	NetBIOS/Samba

Sockets

- In general, a socket is a combination of IP address, port, and a transport protocol.
    - For example, these are distinct sockets:
        - 192.168.1.32, TCP port 80
        - 192.168.1.32, TCP port 21
        - 192.168.1.32, UDP port 80
        - 10.100.50.7, UDP port 80
    - Raw sockets (raw data wrapped in IP header, no transport layer)
    - Local vs Remote
    - Socket Pair: local + remote socket
        - 192.168.1.32:42395 <-> 12.23.34.56:80
            - Local IP: 192.168.1.32
            - Local Port: 42395
            - Remote IP: 12.23.34.56
            - Remote Port: 80

NMAP

TCP SYN Scan (-sS)

- Sends a SYN packet to the target IP/port. Depending on the response we can tell the state of the port:
  - SYN/ACK – Port is open
  - RST (Reset) – Port is closed
  - No response – Port is filtered (could be open or closed)

TCP Connect Scan (-sT)

- Tries to establish a full connection (completed TCP handshake) to the target IP/port.

UDP Scan (-sU)

- Sends an empty UDP datagram to target IP/port (non-empty for specific services)
  - If we get a response we know it's open (not always true, application dependent)
  - If we get a specific 'port unreachable' error we know it's closed!
  - Else... we don't really know much and give up after a while (SLOW!)

There are more and you will learn more about them in this week's engagements!

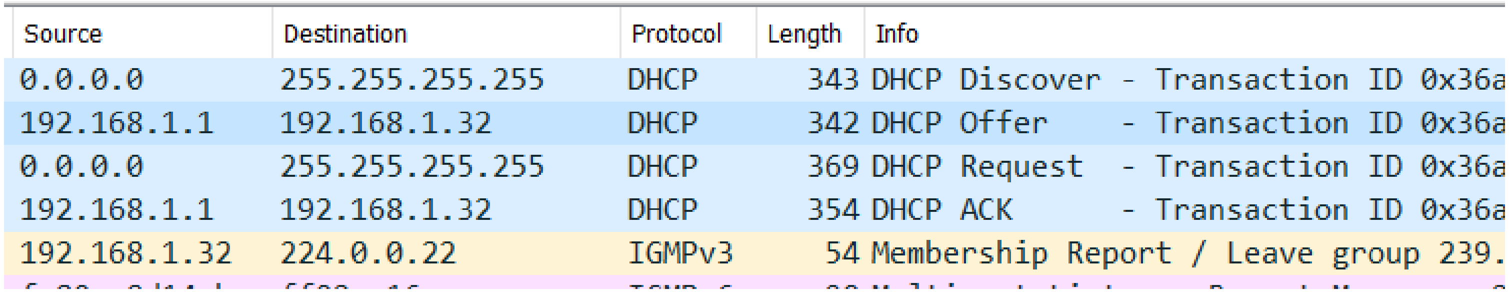
Dynamic Host Configuration Protocol (DHCP)

Allows a host to automatically retrieve network configuration settings:

- IP Address
- Subnet Mask
- DNS server
- Default gateway

How it works:

1. Host connects to network and broadcasts DHCPDISCOVER message to 255.255.255.255:67 (UDP)
2. DHCP server reserves an IP for the host and sends back a DHCPOFFER with the details
3. Host accepts the offer by broadcasting a DHCPREQUEST for the IP offered in step #2.
4. DHCP server sends back a DHCPACK with the IP lease duration and potentially other info, completing the process.



Source	Destination	Protocol	Length	Info
0.0.0.0	255.255.255.255	DHCP	343	DHCP Discover - Transaction ID 0x36a
192.168.1.1	192.168.1.32	DHCP	342	DHCP Offer - Transaction ID 0x36a
0.0.0.0	255.255.255.255	DHCP	369	DHCP Request - Transaction ID 0x36a
192.168.1.1	192.168.1.32	DHCP	354	DHCP ACK - Transaction ID 0x36a
192.168.1.32	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.

Notice the last entry and DHCP response – changes from 0.0.0.0 to 192.168.1.32

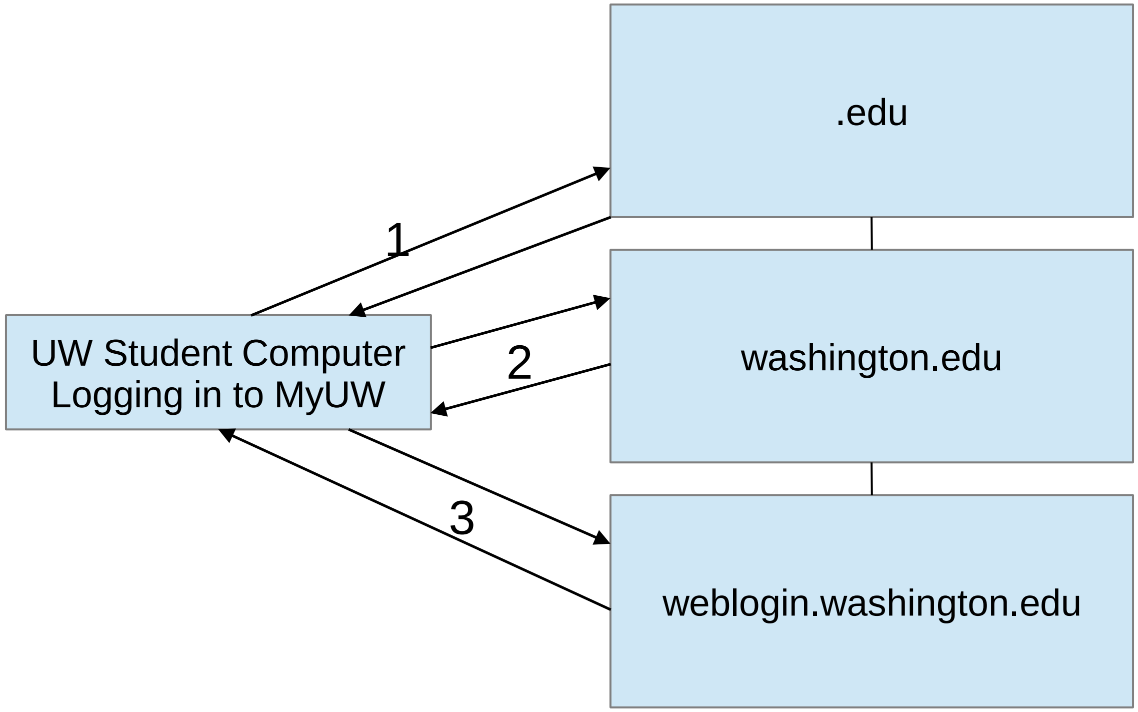
Domain Name System (DNS)

- A lookup system for domain-related information
- Mostly used for resolving a domain name to an IP address (“forward lookup”), but provides other info too:
    - “Reverse lookup” (IP to domain)
    - E-mail related server lookup (SMTP)
    - Other DNS servers
- A DNS zone is an administrative “chunk” of a domain space, often per domain/sub-domain
    - Example zones:
        - washington.edu
        - seattle.washington.edu
        - tacoma.washington.edu
        - bothell.washington.edu
        - compsci.tacoma.washington.edu
- DNS servers maintain mapping data for DNS zones in a zone file

Domain Name System (DNS)

- Zone files consist of a number of records of various types, including:
  - Start of Authority (SOA)
    - Required
    - Indicates person responsible for the name server
    - The authoritative name server for the zone
    - Other important metadata like time-to-live (TTL)
  - A record
    - Maps an IPV4 address to a domain name
  - AAAA record
    - Maps an IPV6 address to a domain name
  - MX record
    - Mail servers that handle incoming mail for a domain
  - NS record
    - Name servers on record for the domain that are authoritative
  - PTR record (Reverse DNS)
    - Maps a hostname to an IP address

Domain Name System (DNS)



- DNS is hierarchical
- DNS root zones for top-level domains (.edu, .com, etc.)
- With no prior info, host queries the root (1) and moves down the hierarchy until a name server can answer it (1 or 2 or 3)

Packet Capture

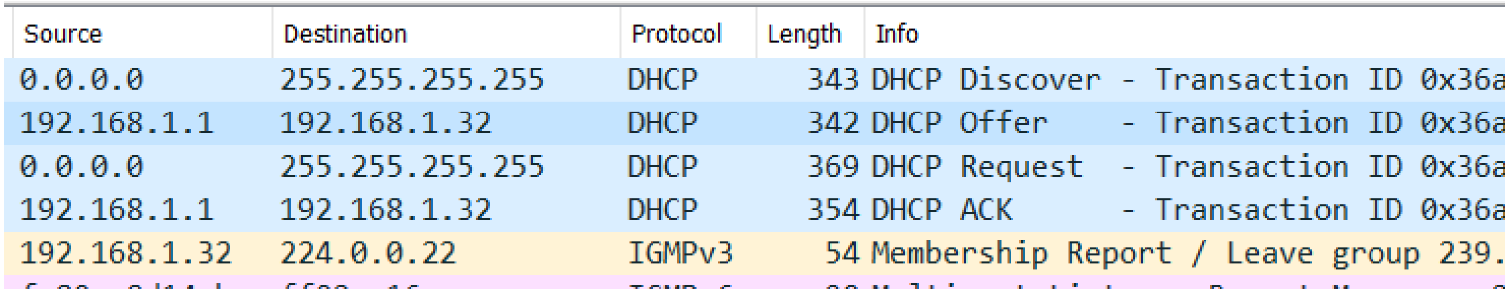
- Captured in the pcap file standard
- Wireshark
  - Great filtering application build on top of packet capture libraries/drivers
  - Packet capture driver installs on machine, allows capturing traffic to/from a network interface
  - Great for analyzing information and what’s going on in a protocol
  - We'll be using this throughout the course for various things
  - Included in Kali!
- Other tools are out there
  - tcpdump/libpcap
  - winpcap
  - Command line tools or libraries if you need programmatic access to captured data

Demo: Wireshark / tcpdump

Lab

1. Open Wireshark, choose network interface (eth0) and start capturing traffic.
2. Open a terminal and run the following commands to release the machine's DHCP configuration and then request an IP.
   - sudo dhclient -r eth0
   - sudo dhclient eth0
3. Stop the Wireshark capture.
4. In the Wireshark filter bar, type 'DHCP' to filter for DHCP protocol traffic.
5. At this point, only DHCP traffic should be displayed.
6. Identify the various DHCP messages discussed in the lecture (for example, DHCPDISCOVER).

You should see four things: DHCP Discover, DHCP Offer, DHCP Request, and DHCP ACK



Source	Destination	Protocol	Length	Info
0.0.0.0	255.255.255.255	DHCP	343	DHCP Discover - Transaction ID 0x36a
192.168.1.1	192.168.1.32	DHCP	342	DHCP Offer - Transaction ID 0x36a
0.0.0.0	255.255.255.255	DHCP	369	DHCP Request - Transaction ID 0x36a
192.168.1.1	192.168.1.32	DHCP	354	DHCP ACK - Transaction ID 0x36a
192.168.1.32	224.0.0.22	IGMPv3	54	Membership Report / Leave group 239.