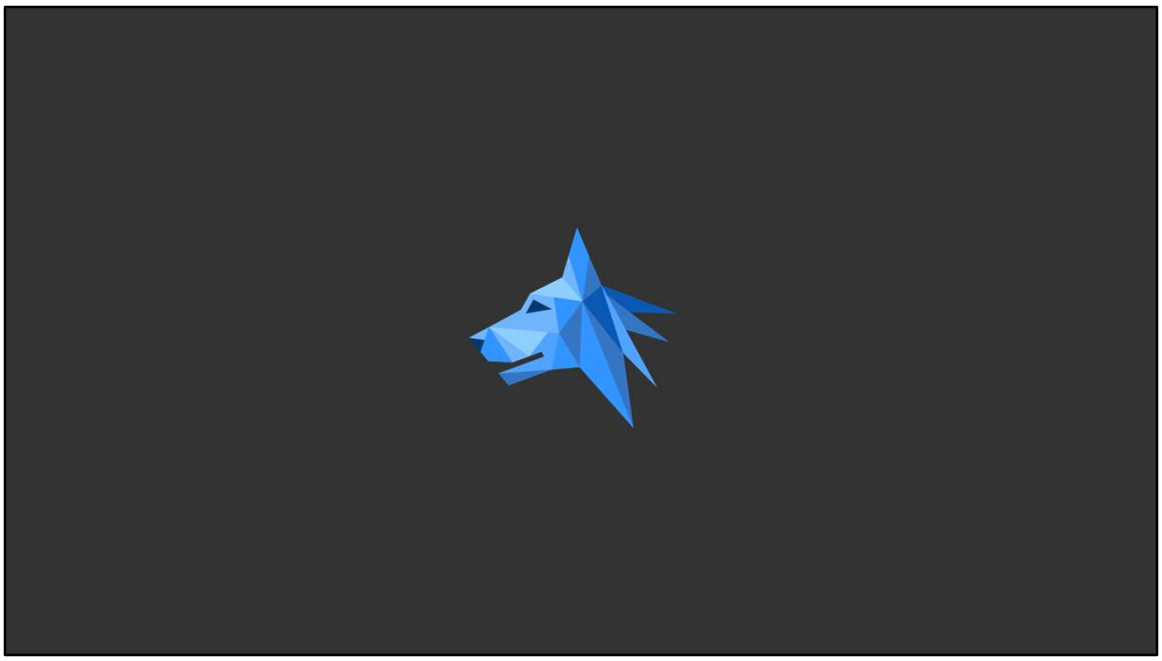


SecureSet and Flatiron School have joined forces with a mission of enabling people to pursue careers they love.

We are tackling the cyber skills gap together, using our collective strengths to provide transformative cybersecurity education.

NET100

OSI Layers 3 and 4

OSI LAYER 3 - TCP/IP INTERNET LAYER

Transport and routing of packets across network boundaries; end-point to end-point protocols.

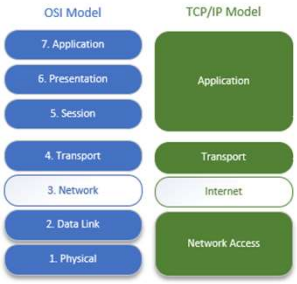
How Internet Protocol IP relates to the Mail Service:

- Unique Address for source and destination
- Routing: how to forward/route data
- Start thinking about how to create a map of addresses

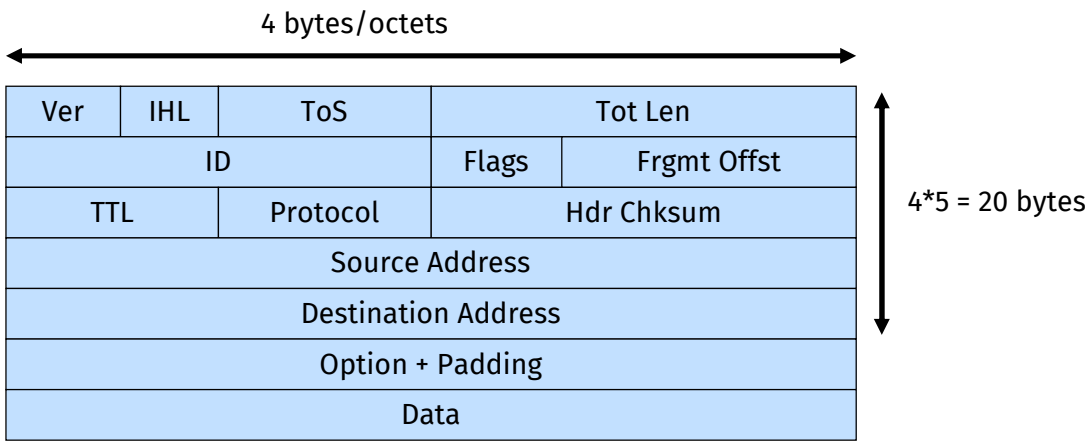
Global Addressability

TCP/IP uses “IP address”

Information Labeled as: **Packet**



IP PACKET STRUCTURE



IP PACKET DIAGRAM

Ver: 4
IHL: 5 (20 bytes)
ToS: 00
Total Len: 00 34 (52 bytes)
ID: 72 D2
Flags: 4

Fragment Offset: 0 00
TTL: 80
Protocol: 06
Checksum: 00
Source Address: C0 A8 00 5A
Dest Address: AC D9 02 0E

Ethernet Frame (MAC – MAC – TYPE) IP Packet

```
0000  78 8a 20 ba 81 c5 00 05  1b ad da 5d 08 00 45 00
0010  00 34 72 d2 40 00 80 06  00 00 c0 a8 00 5a ac d9
0020  02 0e d3 11 01 bb ad 38  39 9d 00 00 00 00 80 02
0030  ff ff 70 10 00 00 02 04  05 b4 01 03 03 08 01 01
0040  04 02
```

flatironschool.com

Destination MAC Address: 78-8a-20-ba-81-c5
Source MAC Address: 00-05-1b-ad-da-5d
EtherType: 08 00 (IPv4 See: https://en.wikipedia. g/wiki/EtherType)

Ver: 4 indicates IPv4. What would it be for IPv6?

Note: IHL is 5, which corresponds to 20 bytes (why?) Two hex characters is a byte. So, 45 00 00 34 72 d2 40 00 80 06 00 00 c0 a8 00 5a ac d9 02 0e is 20 bytes.

Total Length: 00 34 (52 bytes, IP header and IP payload)

ID: 72 D2 If there are multiple packets in this IP stream (fragmentation), then all related packets will have the same ID.

Flags: 4

Fragment Offset: 0 00

TTL: 80

Protocol: 06 (Indicates TCP, if this were a UDP packet it would be 17)

Checksum: 00

Source Address: C0 A8 00 5A (IP Address see subsequent slides)

Destination Address: AC D9 02 0E (IP Address see subsequent slides)

IPV4 ADDRESS (REVIEW)

32-bit number

Usually displayed as 4 “octets”

- 192.168.0.57
- Each octet is a number between 0 and 255
  - Why?

Numerical label assigned to each connected device or node, used for IP routing

With 8 bits, we can display 28, or 256, numbers total. Since we start at 0, this is 0-255.

IPV6 ADDRESS (REVIEW)

128-bit number

Usually expressed as 8 sixteen-bit numbers separated by a colon (using Hex)

- 2001:0DB8:AC10:FE01:0000:0000:0000:0000
- 2001:0DB8:AC10:FE01:::: (all zeroes can be left off)

https://en.wikipedia.org/wiki/IPv6_address

ARP & RARP

Address Resolution Protocol (ARP)

- Helps to bridge information between OSI Layer 3 and 2.
- “I know the IP address; I need to know the MAC address”
- More later - this is more interesting/useful/risky than it would seem at first

Reverse Address Resolution Protocol (RARP)

- Just like it sounds - “I know the MAC Address, what is the IP?”
- Obsolete protocol, but comes up occasionally in discussion

IP ROUTING PROTOCOL GOALS

Build Routing Table (dynamically) to all subnets in a network

If multiple routes pick the best

Remove invalid routes

If an alternate route is made available (though another neighbor) add it to the routing table

Converge fast

Prevent loops

Routing protocols basically break down into Dynamic and Static, under Dynamic- Distance vector and Link-state. Each routing protocol uses different metrics to decide what is the best route.

IP ROUTING

Components of a Routing Table

- Subnet Number
- Forwarding interface
- Next hop address
- Metric

Directly Connected routes are learned first w/o a routing protocol

Next-hop router is usually the one who told a router about a subnet

Routing protocols basically break down into Dynamic and Static, under Dynamic- Distance vector and Link-state. Each routing protocol uses different metrics to decide what is the best route.

.

NETWORK LAYER UTILITIES

Domain Name System (DNS)

- Resolve name into IP address

ICMP (ping)

- Echo request / reply

DHCP (address pool, MAC independent)

- LAN Broadcast requesting IP address
- IP Address+ Subnet + Default GW + other Server IP + file download (diskless workstations)
- Configuration done by subnet rather than by host
- Conditional granting
- Discuss IP helper

ICMP (ping) is a utility and not a service ie: no port

DNS: Domain Name Service: Domain names to IPs and vice versa
ARP: Address Resolution Protocol: Explain how ARP works if you have not already. A host/node wants to send traffic to some destination but only has the IP...ARP broadcast requests that whichever entity has the IP address please respond with their MAC address.
ICMP: ping/traceroute, its is a support protocol that is used by network devices to send error messages and operational information.
DHCP: Provide IP info to DHCP configured machines

The Transport Layer

OSI LAYER 4 - TCP/IP TRANSPORT LAYER

Role / Responsibilities of Transport Protocol

Segment (divide) data from applications into manageable size.

Multiplex - provide delivery of packets to different applications (hint - ports)

Manage connections (or not)

Flow control (or not)

Reliable delivery confirmation (or not)

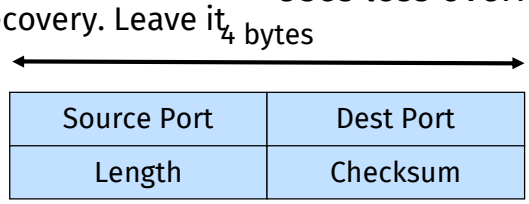
Information Labeled as: **Segment**

TRANSPORT LAYER PROTOCOLS – UDP/TCP

- User Datagram Protocol (UDP)
  - Streamlined / Fast / Unidirectional
- Transmission Control Protocol (TCP)
  - Acknowledgements and reliability
  - Retransmission

UDP

- Connectionless
- Unreliable
- No Flow Control:
  - No wait for ACK
- No Reassembly but supports Data transfer:
  - No reordering, no recovery. Leave it to higher layers
- Performs Segmentation
  - No ordering information, so data may get jumbled
- Allows Multiplexing:
  - Socket just changes transport protocol value
- Uses less overhead

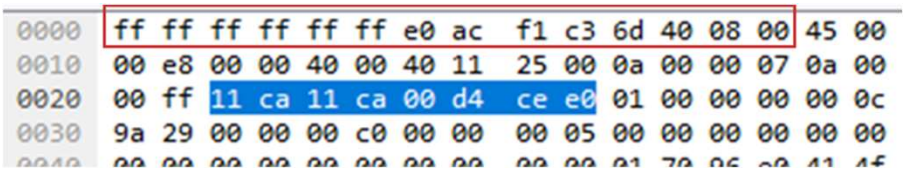


This is a ”Send and Pray” protocol.

UDP HEADER

Ethernet Header (red)
IP Header (45 00 ... 00 ff)
UDP Header (in blue)

0x11CA (Source Port)
0x11CA (Destination Port)
0x00D4 (Length)
0xCEE0 (Checksum)



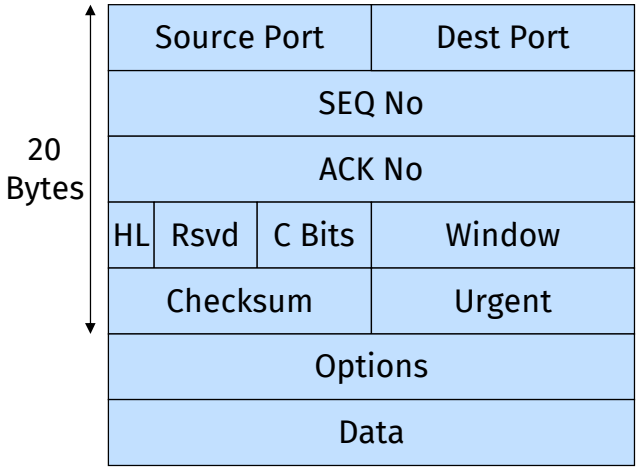
Is this IPv4 or 6?

0x11CA = 4554 (Decimal)
0x00D4 = 212 (Decimal)

Checksum (error checking)

TCP FEATURES

- Multiplexing
- Error Recovery
- Flow Control
- Connection oriented
- End-to-end ordered data transfer
- Segmentation/Reassembly



TCP HEADER

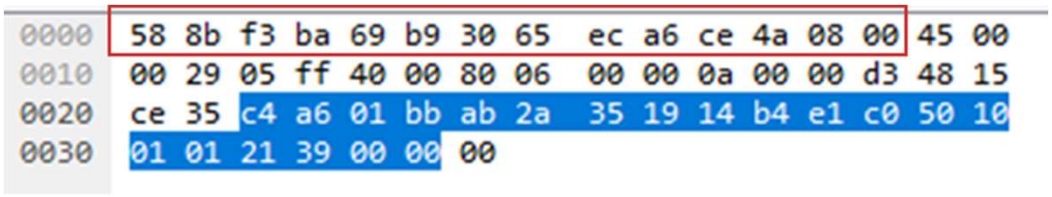
Ethernet Header (red)

IP Header (45 00 … CE 35)

TCP Header (in blue)

TCP Header Fields:

- 0xC4A6 (Source port)
- 0x01BB (Destination Port)
- 0xAB2A 3519 (Seq Number)
- 0x14B4 E1C0 (Ack Number)
- 0x5 (Length – 4 bits)
- 0x010 (Flags - 12 bits)
- 0x0101 (Windows Size)
- 0x2139 (TCP Checksum)
- 0x0000 (Urgent Pointer)



Is this IPv4 or 6?

CONNECTION ORIENTED

Connection Establishment

- Initialize SEQ # and ACK # and Port # agreement
- 3-way handshake (SYN, SYN+ACK, ACK)

Connection Termination

- (ACK+FIN, ACK, ACK+FIN, ACK)

Remember that:

- Connection-oriented =/= Reliable

Control Bits (6):

- (URG) Urgent pointer
- (PSH) Push
- (SYN) Synchronize
- (ACK) Acknowledgment
- (RST) Reset
- (FIN) No more data from sender

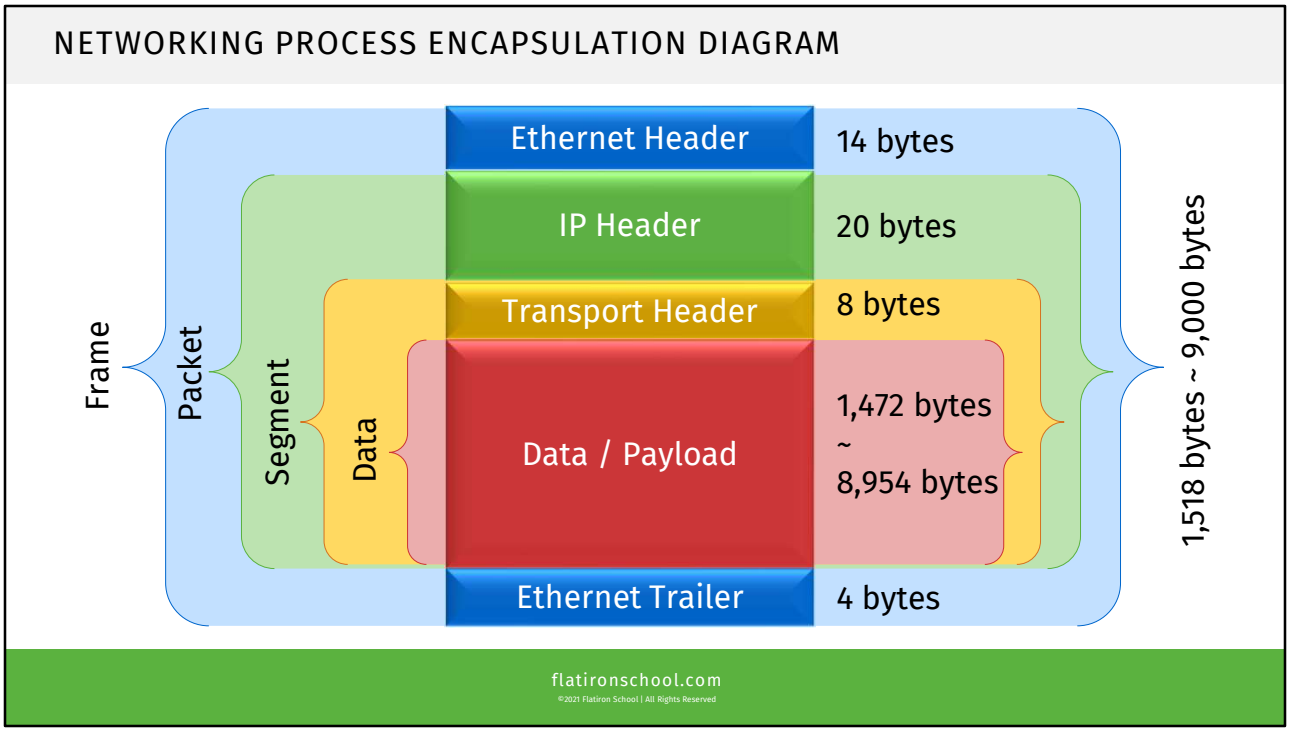
DATA ENCAPSULATION

Layers do not care about payload details

- Each lower layer generally treats higher layer as payload

Remember specific terms used for the PDU (Protocol Data Unit)

- Segment = (L4) Transport
- Packet = (L3) Network
- Frame = (L2) Data link
- Bits = (L1) Physical



Standard Frames: 1500 byte packets (1518 byte total frame)
Jumbo Frames: 8982 byte packets (9000 byte total frame)

https://en.wikipedia.org/wiki/Ethernet_frame

Network Address Translation (NAT)

NAT

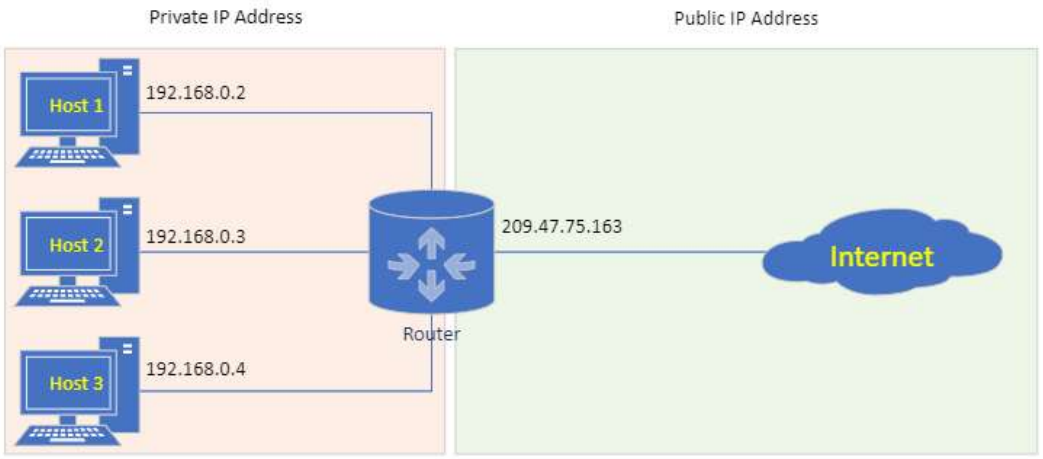
NAT provides a way to manage IP addresses for these billions of devices, without giving each one a unique IP address.

HOW DOES NAT WORK?

Computers on a local network have private IP addresses, and the router (which connects the local network to the internet) advertises a single public IP address.

These private IP addresses must be unique on the local network but can be reused on a different local network.

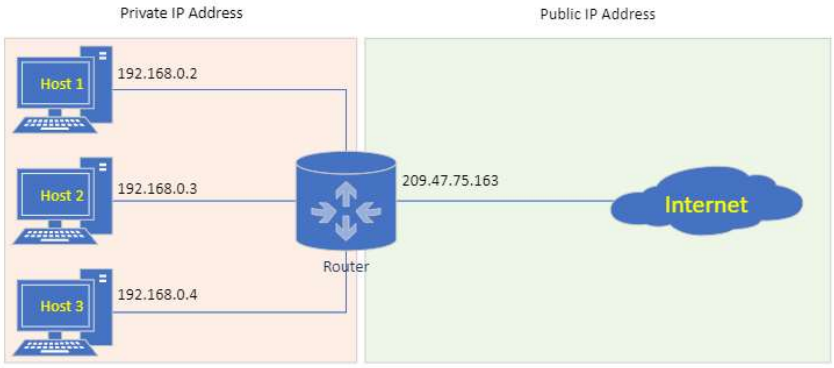
HOW DOES NAT WORK?



flatironschool.com

NAT - SECURITY

NAT also provides a measure of security by limiting visibility of the devices inside the local network to the rest of the internet.



NAT TABLE (HANDLING PORTS)

The router handles traffic from many (dozens? hundreds?) of devices...

Furthermore, it handles traffic from many different source ports on each device.

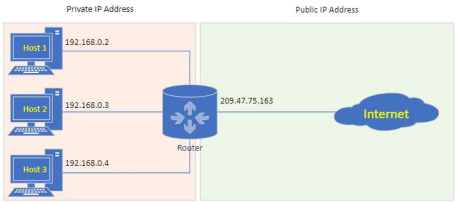
How does it handle routing packets back to the source device?

That is, what if multiple devices on the network use the same source port?

NAT TABLE (HANDLING PORTS)

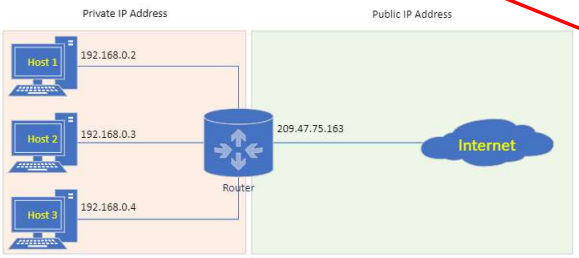
The router maintains a table of IP addresses and source ports.

Local IP	Local Src Port	Router IP	Router Src Port	Destination IP:Port (Internet)
192.168.0.2	50542	209.47.75.163	47534	98.138.219.231:443
192.168.0.3	33571	209.47.75.163	55555	72.19.57.109:25
192.168.0.4	50542	209.47.75.163	39314	132.157.222.9:22
192.168.0.3	27796	209.47.75.163	25486	58.48.48.161:110
192.168.0.3	22258	209.47.75.163	52214	8.8.8.8:53
192.168.0.4	37489	209.47.75.163	49494	98.138.219.231:443



NAT TABLE (HANDLING PORTS)

Local IP	Local Src Port	Router IP	Router Src Port	Destination IP:Port (Internet)
192.168.0.2	50542	209.47.75.163	47534	98.138.219.231:443
192.168.0.3	33571	209.47.75.163	55555	72.19.57.109:25
192.168.0.4	50542	209.47.75.163	39314	132.157.222.9:22
192.168.0.3	27796	209.47.75.163	25486	58.48.48.161:110
192.168.0.3	22258	209.47.75.163	52214	8.8.8.8:53
192.168.0.4	37489	209.47.75.163	49494	98.138.219.231:443



What happens if two different internal IP addresses use the same source port?

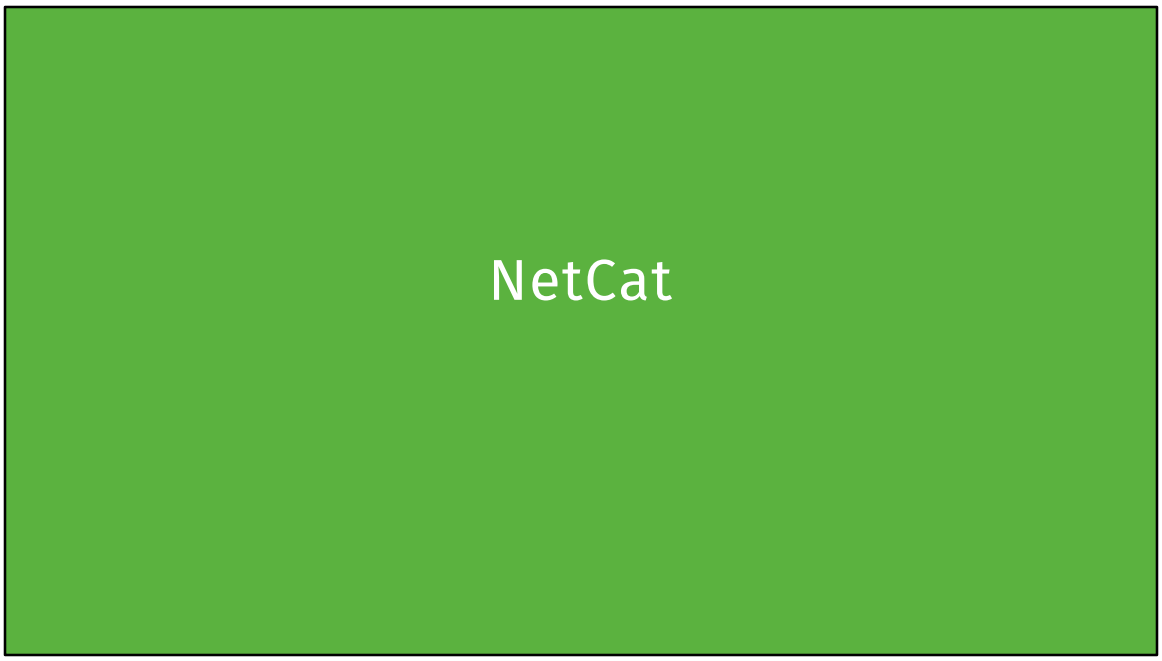
GATEWAY

Border device that controls traffic flow (routes) outward from a network segment

Default device – when other routes are not appropriate (e.g. internal networks)

May have other services (DNS, DHCP, FW, Proxy), as well

Reference:https://whatismyipaddress.com/gateway


NetCat

NETCAT OVERVIEW

What is it?

Simple tool with many uses (and abuses)

Basically lets you use TCP or UDP to make a connection between two hosts (TCP is default)

Capabilities

Simple “chat” between hosts

Send / Receive data

File transfers

Banner grabs

etc ...

Note: You will be using the chat functionality in lab - adding more later...

NCAT MODES

Listen Mode

Waits for incoming connection

- --listen or –l

Defaults to port 31337

Privileged user need to bind to a port <1024

- --keep-open (-k) allows for multiple TCP connections

“Server mode”

Connect Mode

Initiates connection (or sends UDP traffic)

- -C

“Client mode”

Listen Mode allows netcat to wait for an incoming connection on a specific port (31337 by default, if no port provided). Note that in order to listen on a port < 1024, you must be running with privileges (sudo).

The TCP listener only allows 1 connection at a time and exists after the client disconnects
--keep-open (-k) allows multiple TCP connections to be established simultaneously. Multiple TCP connections can exist and the list of IPs on that port make up a “client list.” Any outgoing communication will be sent to all connections, but incoming will be on a per client basis.

A listener will only communicate with 1 UDP client, whichever is the first to send. The listener does not maintain a list of clients.

The connect mode initiates a communication (defined as a socket: IP, Port and Protocol). This can be UDP or TCP connections.

NETCAT BASIC USAGE

nc <host> [<port>]

Connect mode, by default

Acts like a simple Web Browser

nc --listen <host> [<port>]

nc –l <host> [<port>]

Acts like a simple Web Server

flatironschool.com

Notice the double dash (--listen) for the listen or use a single dash (-l) with l

Note that for ncat to act like a simple Web Server, an html file must already exist. So, for example, assume that example.html already exits. The file has the following contents:

```
HTTP/1.0 200 OK

<html>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

Then the command nc –l localhost 8080 < example.html will set netcat to listen on port 8080 and then serve the example.html page out to the connection when request. To test this, open a web browser on the same machine and put “http://localhost:8080” into the URL field. Why does this work?

NCAT ADDITIONAL OPTIONS

Transport Protocols

- --udp (-u) for UDP mode

IP Protocols

- -4 forces IPv4
- -6 forces IPv6
- -p port
- -v verbose mode
- -e Program to execute on a successful connection