Fireside Fridays

IP Transports

Thanks to our sponsors!

ACTIVE COUNTERMEASURES

Antisyphon Training

Lab requirements for this section

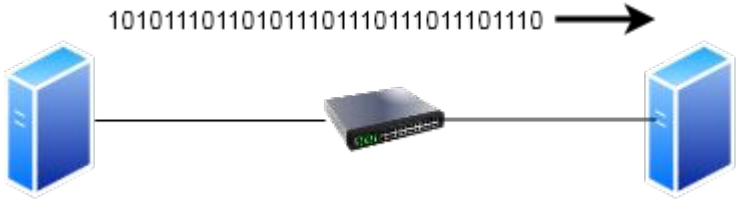
- Today is just lecture
- No lab setup needed

IP review

- In this section we'll cover some of the basics of IP communications
- Not complete coverage, just get us all on the same page
- If you want to deep dive, I have a class for that
  - Next class is May 5th!

https://www.antisyphontraining.com/course/getting-started-in-packet-decoding-with-chris-brenton/

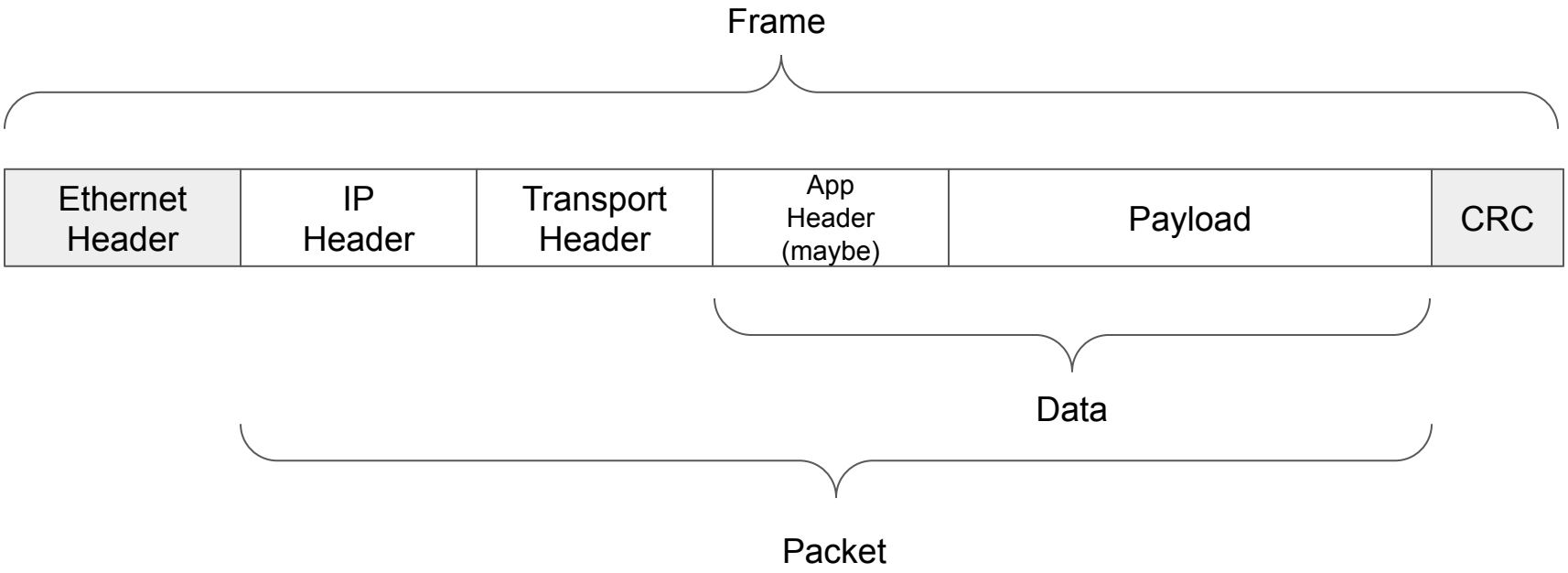
It's all about the binary




```
    10.55.182.100.14291 > 10.233.233.5.80: Flags [S], cksum 0x9309 (correct),
seq
2643678933, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0

        0x0000:  4500 0034 0a86 4000 7f06 3cb4 0a37 b664  E..4..@...<..7.d
        0x0010:  0ae9 e905 37d3 0050 9d93 56d5 0000 0000  ....7..P..V.....
        0x0020:  8002 faf0 9309 0000 0204 05b4 0103 0308  ................
        0x0030:  0101 0402                                ....
```

Anatomy of a transmission



Decoders simply offset and measure based on RFC standards

Spoiler: It's all offset and measurement

- Last slide showed the order of headers
- This is defined by the RFCs
- RFCs also define fields in the headers
- Identifying an attribute is simply a matter of offsetting and measuring the right value

Identifying the transport protocol

- Offset to the beginning of the IP header
- Offset to byte 9 within the IP header
- Read the next 8 bits (1 byte) of data
- Compare value to assigned number list
- Print assigned protocol label
- IPv6 "next header" is byte 6 (but may not be the transport)

https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

ICMP

- Protocol 1 (byte 9 of IP header)
- Maintenance protocol
- Not originally designed to transmit data
- System to system checks
- Handle network and some host errors
- Supports unicast, broadcast and multicast
- Lightweight & efficient but unreliable

ICMP and firewalls

- ICMP messages can be:
  - Query/response based like ping
  - Error reporting like Type 3 unreachables
- Query/response can be handled statefully
- Error reporting requires stateful inspection
  - Decode original packet stored in the payload
  - Match this original packet against the state table

ICMP error example

```
07:37:07.730636 IP (tos 0x0, ttl 250, id 61550, offset 0, flags [none], proto ICMP (1), length 96)
    24.220.6.168 > 10.55.200.11: ICMP host 8.26.204.25 unreachable - admin prohibited filter, length 76
       IP (tos 0x0, ttl 121, id 11575, offset 0, flags [none], proto UDP (17), length 97)
    10.55.200.11.56963 > 8.26.204.25.53: 5075 [1au][|domain]
        0x0000:  4500 0060 f06e 0000 fa01 de67 18dc 06a8  E..`.n.....g....
        0x0010:  0a37 c80b 030d 0ae8 0000 0000 4500 0061  .7..........E..a
        0x0020:  2d37 0000 7911 6ddf 0a37 c80b 081a cc19  -7..y.m..7......
        0x0030:  de83 0035 004d 5214 13d3 0000 0001 0000  ...5.MR.........
        0x0040:  0000 0001 0474 7366 650e 7472 6166 6669  .....tsfe.traffi
        0x0050:  6373 6861 7069 6e67 0364 7370 026d 7009  cshaping.dsp.mp.
```

ICMP type 3, code 13 is Admin prohibited

In between red brackets is exposed ICMP header
In between blue brackets is original UDP packet
In between green brackets is 40 bytes of payload from original UDP packet

SI firewall should decode the blue section and compare it to state table

Can you hide data in ICMP?

```
student@packetdecode:~/lab2$ tshark -r weird-ping.pcap -T fields -Y data.data -e "data.data"
 | xxd -r -p  | head -20
dir
dir
 Volume in drive C is OS
 Volume Serial Number is AA6E-E1EA

 Directory of c:\temp

05/27/2021  12:26 AM    <DIR>          .
05/27/2021  12:26 AM    <DIR>          ..
10/15/2003  11:32 AM            25,122 about-nls.txt
07/21/2016  06:55 PM    <DIR>          all-hands
10/15/2003  11:32 AM               608 bugs.txt
02/26/2008  11:26 AM            23,574 build-h-bomb.txt
10/15/2003  11:33 AM            18,318 copying.txt
10/15/2003  11:33 AM               912 credits.txt
10/15/2003  01:44 PM           980,992 cygiconv-2.dll
08/11/2003  01:15 AM            37,888 cygintl-2.dll
08/11/2003  03:39 AM           134,656 cygjpeg-62.dll
10/15/2003  01:45 PM           185,344 cygmcrypt-4.dll
10/15/2003  01:45 PM           134,656 cygmhash-2.dll
student@packetdecode:~/lab2$
```

Some of the oldest C2 channels are based on ICMP

UDP

- Lightweight
- Connectionless (no concept of "state")
- Unreliable but can be built in at app layer
- Uses ICMP for error reporting
- Uses "ports" to support multiple services
- Supports unicast, broadcast and multicast

UDP comms can be unidirectional

```
04:21:42.061717 IP 172.18.0.1.40395 > 172.18.0.2.514: SYSLOG user.notice, length: 210
04:21:42.061839 IP 172.18.0.1.40395 > 172.18.0.2.514: SYSLOG user.notice, length: 207
04:21:42.061968 IP 172.18.0.1.40395 > 172.18.0.2.514: SYSLOG user.notice, length: 210
04:21:42.062084 IP 172.18.0.1.40395 > 172.18.0.2.514: SYSLOG user.notice, length: 237
04:21:42.062132 IP 172.18.0.1.40395 > 172.18.0.2.514: SYSLOG user.notice, length: 201
04:21:42.062177 IP 172.18.0.1.40395 > 172.18.0.2.514: SYSLOG user.notice, length: 211
04:21:42.062222 IP 172.18.0.1.40395 > 172.18.0.2.514: SYSLOG user.notice, length: 212
04:21:42.062266 IP 172.18.0.1.40395 > 172.18.0.2.514: SYSLOG user.notice, length: 221
```

Lack of ICMP error and it's "assumed" packet was delivered and accepted

UDP comms can be bidirectional

```
15:48:42.685367 IP 10.0.2.15.55180 > 75.75.75.75.53: 15701+ A? config.teams.micro
soft.com. (44)
15:48:42.700835 IP 75.75.75.75.53 > 10.0.2.15.55180: 15701 3/0/0 CNAME config.tea
ms.trafficmanager.net., CNAME s-0005.s-msedge.net., A 52.113.194.132 (135)
15:49:42.819756 IP 10.0.2.15.62299 > 75.75.75.75.53: 38640+ A? ctldl.windowsupdat
e.com. (41)
15:49:42.837319 IP 75.75.75.75.53 > 10.0.2.15.62299: 38640 5/0/0 CNAME audownload
.windowsupdate.nsatc.net., CNAME au.download.windowsupdate.com.hwcdn.net., CNAME
cds.d2s7q6s2.hwcdn.net., A 205.185.216.42, A 205.185.216.10 (198)
15:52:20.105650 IP 10.0.2.15.62865 > 75.75.75.75.53: 60440+ A? wpad.hsd1.fl.comca
st.net. (42)
15:52:20.122593 IP 75.75.75.75.53 > 10.0.2.15.62865: 60440 NXDomain 0/1/0 (94)
```

UDP and firewalls

- Vulnerable to spoofing if static filters are used
  - Fix the source port at 53 and blast away
- Can be handled quite nicely with stateful filtering
  - Static filter handles first packet
  - Stateful handles replies
  - But not all services expect replies
- Usually no stateful inspection implementation
- State table timeout usually set around 30 seconds

TCP

- More overhead than UDP
- Built for reliability (to 1980 standards)
- Connection oriented ("state" is maintained)
- Has built in error reporting
- Has built in flow control
- Unicast communications only
- Like UDP, supports multiple services via ports

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.)																															
⋮	⋮																																
60	480																																

TCP flags

- Used to control connection state
- Six low order bits of byte 13
- Used to identify connection establishment all the way through connection closing
- Designed to improve transmission reliability

Flag descriptions

- URG - bit 32 - Indicates that there is data to process in the urgent field of TCP header
- ACK - bit 16 - Indicates that there is data in the acknowledgement field. Should always be on after the first packet in a session
- PSH - bit 8 - Tell the destination to push data from the receipt buffer to the listening app

Flags (continued)

- RST - bit 4 - Indicate a closed port or that a current session has failed irrevocably
- SYN - bit 2 - Used to start a session. Set in the first packet sent by each host.
- FIN - bit 1 - Used to gracefully close a session. Sent once by each end of the connection.

TCP and static firewalls

- Pattern match on TCP flags to enumerate state
- SYN=1 interpreted as connection establishment
- SYN=0 interpreted as established state
- Static filter vulnerable to packet crafting
  - SYN=1 & FIN=1
  - Unsolicited ACK=1

TCP and stateful firewalls

- Like with UDP, dramatic security improvement
- Static filters used to screen first packet in session
- State table used to screen everything else
- Not as susceptible to packet crafting
  - SYN/FIN not passed unless there is a rule to permit connection establishment
  - Not fooled like static filters

TCP and stateful inspection firewalls

- Used with FTP to see data negotiation over command channel
- Not needed by other protocols
- Can be used to inspect the payloads
  - But it is usually simple RegEx pattern matching
  - Not as advanced as proxies or monitoring tools

QUIC

- Built on top of UDP, but effectively a transport
- Assumes HTTP and TLS on all connections
- Improved congestion control and multiplexing
- Error correct when supporting multiple flows
  - Remove head of line problems with TCP
- Connection migration between networks
- Uses UDP port 443

Why do we need QUIC?

- Most Internet traffic is TCP-HTTP-TLS based
  - Handshaking for each creates overhead
  - QUIC designed to replace these connections
- QUIC focused on optimizing this configuration
  - Reduce the overhead involved
  - No TCP three packet handshake
  - Credential and privacy info can be cached
  - Optimize content delivery and data recovery

QUIC is still a work in progress

- QUIC is still changing
  - Updates a recent at Oct 2024
  - RTP support, multipath extensions, event definitions, etc.
- Originally developed by Google
  - Has since gained wider support
- Slowly adopted for streaming content
  - Video & audio
  - Mobile apps

QUIC and firewalls

- Limitations are similar to UDP
  - Traffic controlled at the port and IP level
- Arguably less control as you can't manipulate streams
  - Can't limit to internal hosts only
  - Can't specify key or cryptology levels
  - Basically, you are flying kind of blind

QUIC and security

- Privacy improvements
  - Essentially a tunnel protocol
  - No useful plaintext data
  - No one can see what you are doing
- Major blow to corporate security
  - Blue team cannot see what systems are doing
  - Could be normal or malicious, can't tell
  - Many sites are choosing to disable support

Next week on Fireside Fridays!

- Packet filtering firewalls
- Both packet filtering and proxies
- Next week is just lecture
- The week after is hands on testing

Wrap up

- Thank you for attending!
- Certs & videos should be out by Monday
- If you have any lingering questions, the Discord channel will remain active
  - Also a good chance to socialize with others in the class
  - Have other tips and tricks? Please share with others :-)
- We appreciate you sharing your time with us!