Packet Analysis

UB NetSec

- Syllabus: https://ubnetdef.org/courses/netsec/
- Ran by an Alumni: Chris Crawford
  - he does a lot of Packet Analysis stuff
  - really smart!
  - **@bashasaurusrex is TA**

Taught Differently… SCRUM

- five meetings a week, set call on Google Hangouts/ Zoom or another platform
- what you got done
- what you plan on doing
- what you are stuck on

This really helps learning, do something everyday really builds knowledge quickly than large 2-4 hour stretches.

Learning is self paced!!!

Trello

- since it is SCRUM, you use Trello
- complete X card per week
- each card is a small task, such as
  - install VirtualBox
  - install WireShark
  - listen on X port

You build up the necessary technical skills to build a packet analysis environment, the class is really self contained.

Documentation and Reports

- after building isolated environment you get to use Wireshark, Bro, and Snort to look into packets
- you find cool stuff
- then you write a really detailed report on what happened telling the “Story” of the intrusion
- ever task need documentation, extremely particular, but this builds a **super** useful skill - extremely key eye

Overview:

- What is packet analysis
- Network basics relating to packet analysis
- Packet Sniffers
- Wireshark
- Working with Captured Packets
- Security Applications

Packet Analysis

- Describes the process of capturing and interpreting live data as its flows across a network
- Packet sniffer - tool used to capture raw network traffic
- Packet analysis can help with the following:
  - Understanding network characteristics
  - Learning who is on a network
  - Determining who or what is utilizing available bandwidth
  - Identifying peak network usage times
  - Identifying malicious activity
  - Finding unsecured and bloated applications

Review Network Basics

- TCP, IP, ARP, DHCP all are “rules” that define how packets should be routed, how to initiate a connection, and how to acknowledge receipt of data
- Protocols address a wide variety of issues:
  - **Connection initiation** - client/server side?
  - **Negotiation of connection characteristics** - encrypted?
  - **Data formatting** - how is data in packet organized?
  - **Error detection and correction** - What happens in the event that apacket takes too long to reach its destination?
  - **Connection termination:** How does one host signify to the other that communication has ended?

What is a packet?

What is a packet?

The unit of data routed between origin and destination on a network

Packets are constructed in such a way that layers for each protocol used for a particular connection are wrapped around the packets, like the layers of skin on an onion.

IP Header Contains:

- Controls and flags
- Source and destination IP address

TCP Header Contains:

source/destination port

- SEQ # and ACK # flags among others
- Data

IP Header

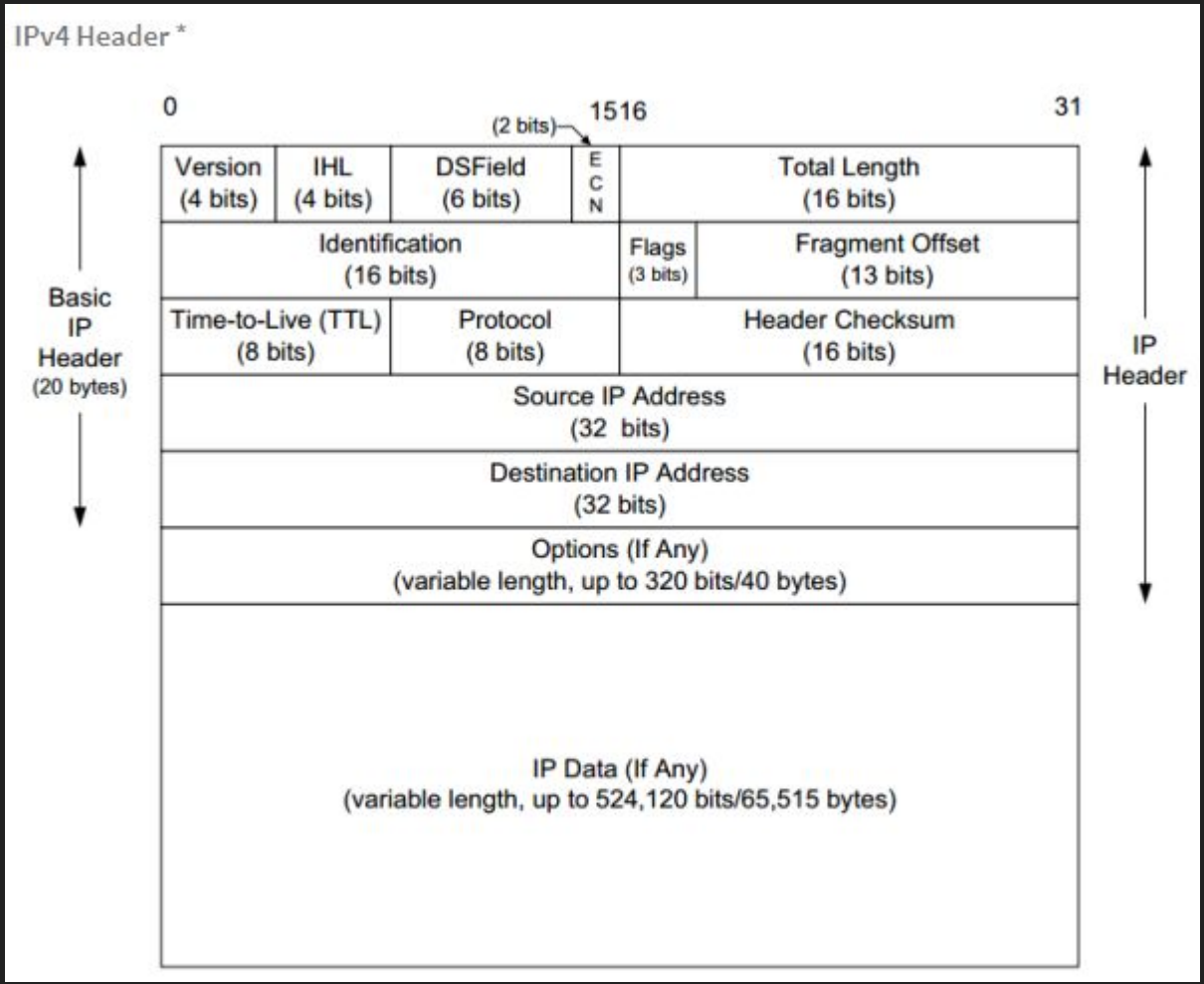
IP Version - v4 or v6

Time to Live/Hop Limit - the # of hops a packet is permitted to travel before being discarded by a router. When router sees that TTL = 0 for an incoming packet, packet is discarded and ICMP response is sent back.

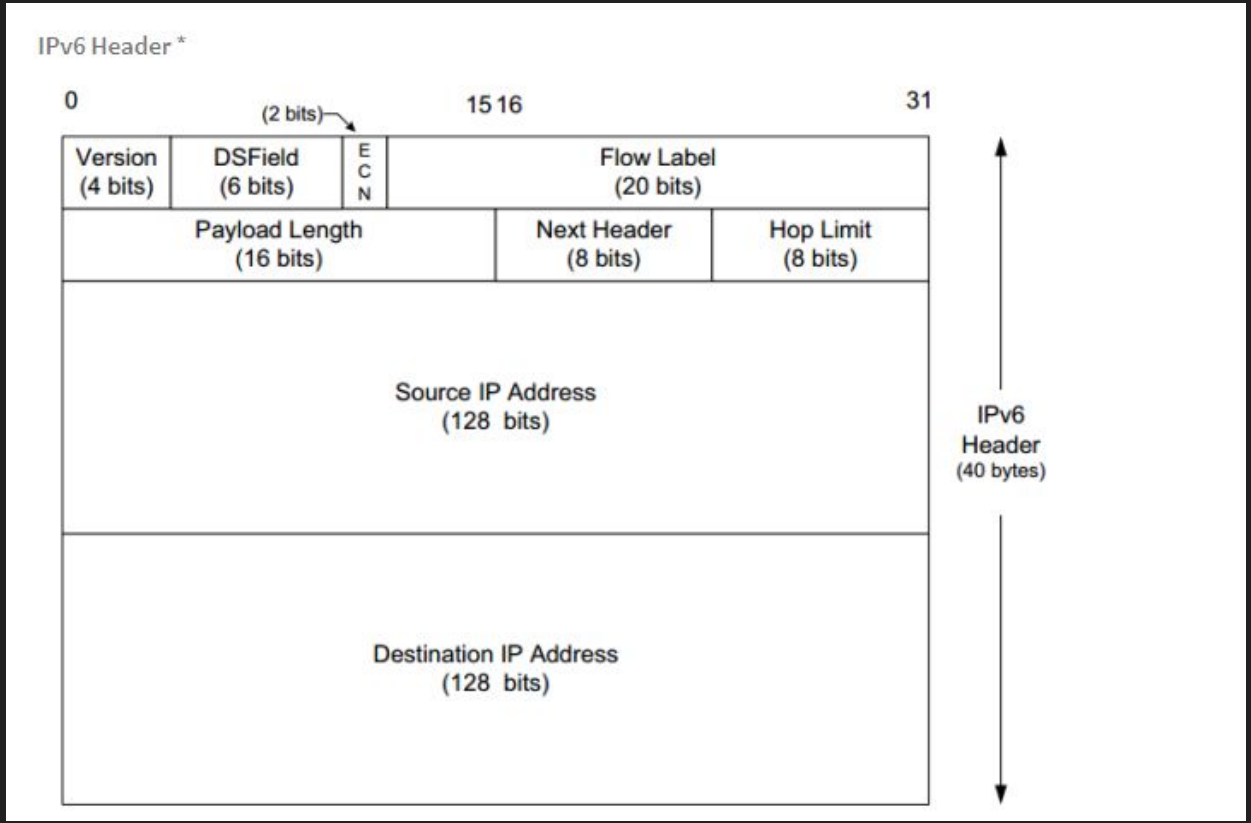
Protocol contains a number indicating the type of data found in the payload portion of the datagram. The most common values are 17 (for UDP) and 6 (for TCP).

Source Address/Destination Address

IPv4 Header


IPv4 Header *
0
1516
31
(2 bits)
Version (4 bits)
IHL (4 bits)
DSField (6 bits)
ECN
Total Length (16 bits)
Identification (16 bits)
Flags (3 bits)
Fragment Offset (13 bits)
Time-to-Live (TTL) (8 bits)
Protocol (8 bits)
Header Checksum (16 bits)
Source IP Address (32 bits)
Destination IP Address (32 bits)
Options (If Any) (variable length, up to 320 bits/40 bytes)
IP Data (If Any) (variable length, up to 524,120 bits/65,515 bytes)
Basic IP Header (20 bytes)
IP Header

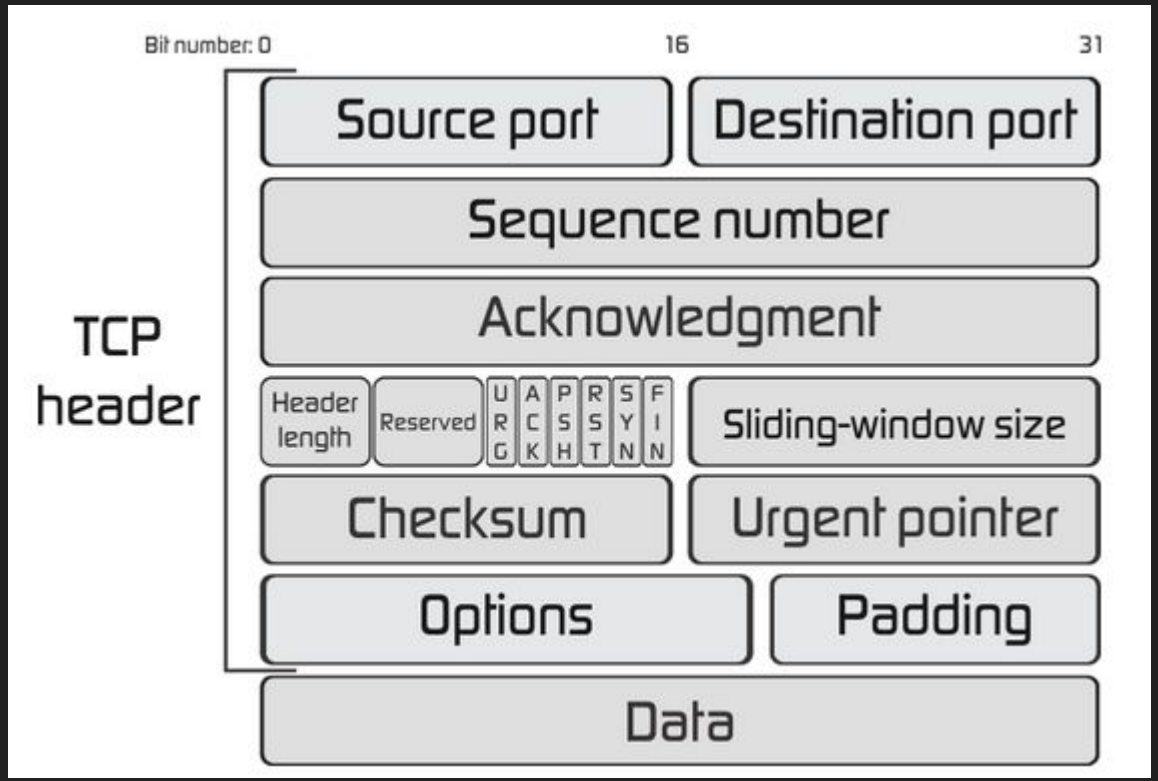
IPv6 Header


IPv6 Header *
0
15 16
31
(2 bits)
Version (4 bits)
DSField (6 bits)
E C N
Flow Label (20 bits)
Payload Length (16 bits)
Next Header (8 bits)
Hop Limit (8 bits)
Source IP Address (128 bits)
Destination IP Address (128 bits)
IPv6 Header (40 bytes)

TCP Header

- TCP is the primary transport protocol used to provide reliable, full-duplex connections
- Source and destination TCP port numbers are the communication endpoints for sending and receiving devices.
- Sequence numbers mark the ordering of a group of messages.
- Control flags indicate a particular connection state or provide additional information.

TCP Header


Bit number: 0
16
31
TCP header
Source port
Destination port
Sequence number
Acknowledgment
Header length
Reserved
URG
ACK
PSH
RST
SYN
FIN
Sliding-window size
Checksum
Urgent pointer
Options
Padding
Data

Packet Sniffing

Collection → Conversion → Analysis

Collection - packet sniffer collects raw binary data from the wire.

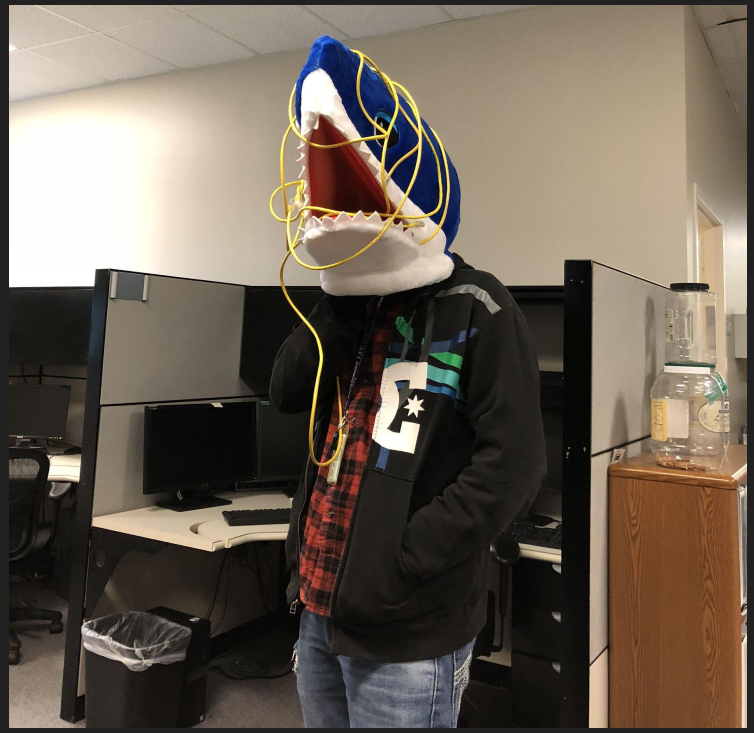
Conversion - raw binary data is converted into a readable form.

Analysis - Sniffer analyzes converted binary data and verifies the protocol of the captured network data based on the info extracted, and begins analysis of the protocols specific features

Before you go sniffing...

Ensure that you have the permission to capture packets from the network you are connected to. (Corporate policies or applicable law might prohibit capturing data from the network)

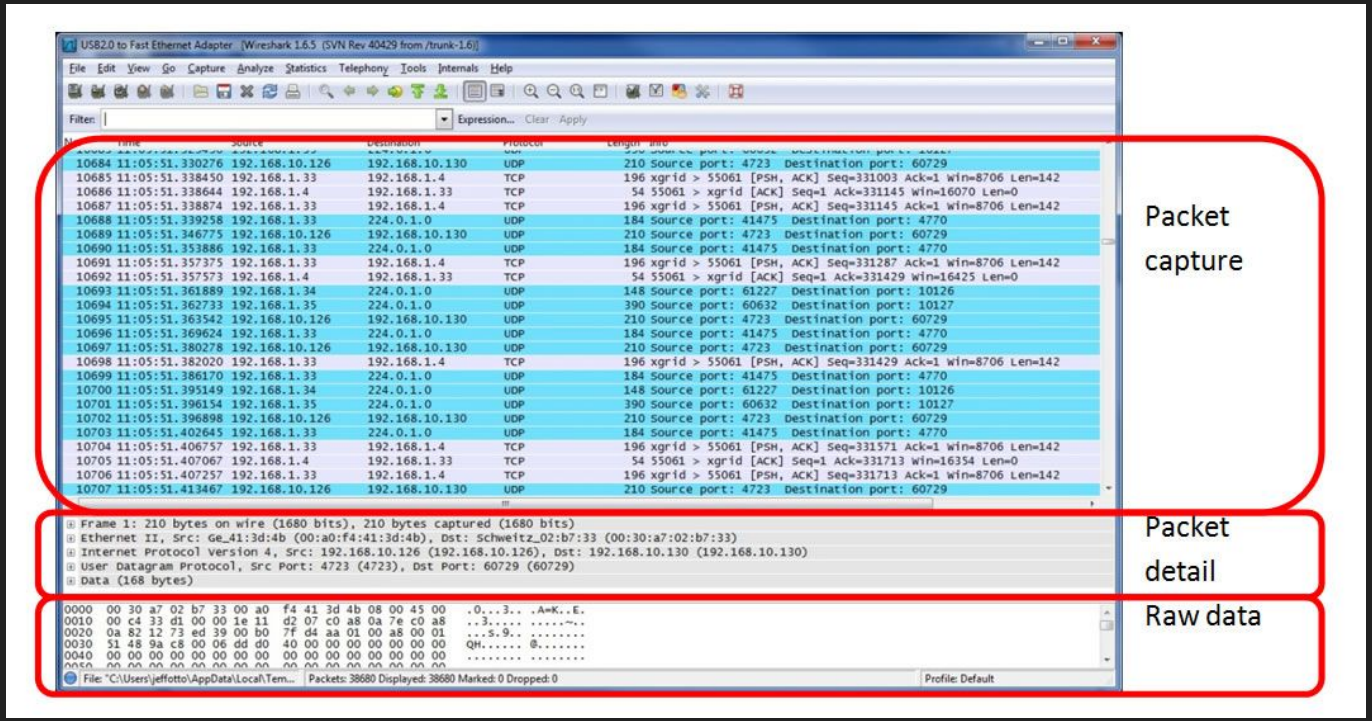
Wireshark



What is wireshark?

- Wireshark is a free and open source packet analyzer.
- Lets you see what is happening on your network at a microscopic level.
- Useful for:
  - Network troubleshooting and analysis
  - Software and communications protocol development
- A headache that you agreed to deal with

Wireshark output



Output - cont

- The output of a packet capture tells us:
  - Source of traffic
  - Destination of traffic
  - Protocol
  - Length in bytes
  - Additional info
- Promiscuous mode - promiscuous mode refers to the special mode of Ethernet hardware, in particular network interface cards (NICs), that allows a NIC to receive all traffic on the network, even if it is not addressed to this NIC

Filters

Wireshark’s filter functionality make it a very useful application. There are two ways to filter in wireshark.

- Display Filter - filters packets AFTER they have been captured. Display filter can be changed on the fly.
- Capture Filter - determines what wireshark will capture even before you initiate a capture. Useful to reduce the size of a raw packet capture.

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src==151.101.193.140 && ip.dst==192.168.1.145

No.	Time	Source	Destination	Protocol	Length	Info
18	5.056628	reddit.com	192.168.1.145	TCP	66	https(443) → 57539 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=512
21	5.084889	reddit.com	192.168.1.145	TCP	60	https(443) → 57539 [ACK] Seq=1 Ack=518 Win=30720 Len=0
22	5.086098	reddit.com	192.168.1.145	TLSv1.2	1514	Server Hello
23	5.086198	reddit.com	192.168.1.145	TCP	1514	https(443) → 57539 [ACK] Seq=1461 Ack=518 Win=30720 Len=1460 [TCP segment of a reassembled PDU]
25	5.086352	reddit.com	192.168.1.145	TLSv1.2	1076	Certificate, Certificate Status, Server Key Exchange, Server Hello Done
29	5.111425	reddit.com	192.168.1.145	TLSv1.2	296	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
30	5.111160	reddit.com	192.168.1.145	TLSv1.2	120	Application Data
31	5.111161	reddit.com	192.168.1.145	TLSv1.2	275	Application Data
37	5.142991	reddit.com	192.168.1.145	TCP	60	https(443) → 57539 [ACK] Seq=4472 Ack=1052 Win=31744 Len=0

Display Filter

Capture Filter

Wireshark · Capture Filters

Name	Filter
Non-HTTP and non-SMTP to/from www.wireshark.org	not port 80 and not port 25 and host www.wireshark.org
No ARP and no DNS	not arp and port not 53
HTTP TCP port (80)	tcp port http
TCP or UDP port 80 (HTTP)	port 80
UDP only	udp
TCP only	tcp
IPX only	ipx
IPv6 address 2001:db8::1	host 2001:db8::1
IPv6 only	ip6
IPv4 address 192.0.2.1	host 192.0.2.1
IPv4 only	ip
No ARP	not arp
No Broadcast and no Multicast	not broadcast and not multicast
Ethernet type 0x0806 (ARP)	ether proto 0x0806
Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00

OK Cancel Help

```
Frame 21: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
  Interface id: 0 (\Device\NPF_{5F39EC36-74A9-465C-B46A-73529DEE3577})
  Encapsulation type: Ethernet (1)
  Arrival Time: Nov  1, 2018 12:54:10.525367000 Eastern Daylight Time
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1541091250.525367000 seconds
  [Time delta from previous captured frame: 0.027922000 seconds]
  [Time delta from previous displayed frame: 0.028261000 seconds]
  [Time since reference or first frame: 5.084889000 seconds]
  Frame Number: 21
  Frame Length: 60 bytes (480 bits)
  Capture Length: 60 bytes (480 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp]
  [Coloring Rule Name: TCP]
  [Coloring Rule String: tcp]
Ethernet II, Src: Cisco-Li_89:51:f0 (c0:c1:c0:89:51:f0), Dst: AsustekC_e0:10:0b (18:31:bf:e0:10:0b)
Internet Protocol Version 4, Src: reddit.com (151.101.193.140), Dst: 192.168.1.145 (192.168.1.145)
```

```
0000  18 31 bf e0 10 0b c0 c1  c0 89 51 f0 08 00 45 00   ·1····· ··Q···E·
0010  00 28 b2 36 40 00 39 06  74 6e 97 65 c1 8c c0 a8   ·(·6@·9· tn·e····
0020  01 91 01 bb e0 c3 9e 0a  f3 b4 ab 3b 10 b9 50 10   ····· ···;··P·
0030  00 3c 64 3a 00 00 00 00  ec 7a e9 2e               ·<d:···· ·z··
```

Video Demonstration - Basic Packet Capture

- Wireshark main screen
- Select interface
- Begin capture
- Background packet traffic - other open tabs, OneDrive, etc
- Reddit.com
- Lots of packets
- Can set up wireshark so that it resolves names of packet destinations/sources
- Can look at Conversations to get a better idea of what is happening and to pinpoint certain communications
    - Easy way to apply a display filter, just select the conversation you want to see.

Basic Packet Capture

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr==151.101.193.140 && ip.addr==192.168.1.145

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
17	5.030531	192.168.1.145	reddit.com	TCP	66	57539 → https(443) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
18	5.056628	reddit.com	192.168.1.145	TCP	66	https(443) → 57539 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=512
19	5.056696	192.168.1.145	reddit.com	TCP	54	57539 → https(443) [ACK] Seq=1 Ack=1 Win=65536 Len=0
20	5.056967	192.168.1.145	reddit.com	TLSv1.2	571	Client Hello
21	5.084889	reddit.com	192.168.1.145	TCP	60	https(443) → 57539 [ACK] Seq=1 Ack=518 Win=30720 Len=0
22	5.086098	reddit.com	192.168.1.145	TLSv1.2	1514	Server Hello
23	5.086198	reddit.com	192.168.1.145	TCP	1514	https(443) → 57539 [ACK] Seq=1461 Ack=518 Win=30720 Len=1460 [TCP segment of a reassembled PDU]
24	5.086245	192.168.1.145	reddit.com	TCP	54	57539 → https(443) [ACK] Seq=518 Ack=2921 Win=65536 Len=0
25	5.086352	reddit.com	192.168.1.145	TLSv1.2	1076	Certificate, Certificate Status, Server Key Exchange, Server Hello Done
26	5.087048	192.168.1.145	reddit.com	TLSv1.2	147	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
27	5.091377	192.168.1.145	reddit.com	TLSv1.2	147	Application Data
28	5.091813	192.168.1.145	reddit.com	TLSv1.2	322	Application Data
29	5.111425	reddit.com	192.168.1.145	TLSv1.2	296	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
30	5.118160	reddit.com	192.168.1.145	TLSv1.2	120	Application Data
31	5.118161	reddit.com	192.168.1.145	TLSv1.2	275	Application Data
32	5.118317	192.168.1.145	reddit.com	TCP	54	57539 → https(443) [ACK] Seq=972 Ack=4472 Win=65536 Len=0
33	5.119069	192.168.1.145	reddit.com	TLSv1.2	92	Application Data
34	5.119146	192.168.1.145	reddit.com	TLSv1.2	96	Application Data
37	5.142991	reddit.com	192.168.1.145	TCP	60	https(443) → 57539 [ACK] Seq=4472 Ack=1052 Win=31744 Len=0

Basic Packet Capture

*Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr==151.101.193.140 && ip.addr==192.168.1.145

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
17	5.030531	192.168.1.145	reddit.com	TCP	66	57539 → https(443) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
18	5.056628	reddit.com	192.168.1.145	TCP	66	https(443) → 57539 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=512
19	5.056696	192.168.1.145	reddit.com	TCP	54	57539 → https(443) [ACK] Seq=1 Ack=1 Win=65536 Len=0
20	5.056967	192.168.1.145	reddit.com	TLSv1.2	571	Client Hello
21	5.084889	reddit.com	192.168.1.145	TCP	60	https(443) → 57539 [ACK] Seq=1 Ack=518 Win=30720 Len=0
22	5.086098	reddit.com	192.168.1.145	TLSv1.2	1514	Server Hello
23	5.086198	reddit.com	192.168.1.145	TCP	1514	https(443) → 57539 [ACK] Seq=1461 Ack=518 Win=30720 Len=1460 [TCP segment of a reassembled PDU]
24	5.086245	192.168.1.145	reddit.com	TCP	54	57539 → https(443) [ACK] Seq=518 Ack=2921 Win=65536 Len=0
25	5.086352	reddit.com	192.168.1.145	TLSv1.2	1076	Certificate, Certificate Status, Server Key Exchange, Server Hello Done
26	5.087048	192.168.1.145	reddit.com	TLSv1.2	147	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
27	5.091377	192.168.1.145	reddit.com	TLSv1.2	147	Application Data
28	5.091813	192.168.1.145	reddit.com	TLSv1.2	322	Application Data
29	5.111425	reddit.com	192.168.1.145	TLSv1.2	296	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
30	5.118160	reddit.com	192.168.1.145	TLSv1.2	120	Application Data
31	5.118161	reddit.com	192.168.1.145	TLSv1.2	275	Application Data
32	5.118317	192.168.1.145	reddit.com	TCP	54	57539 → https(443) [ACK] Seq=972 Ack=4472 Win=65536 Len=0
33	5.119069	192.168.1.145	reddit.com	TLSv1.2	92	Application Data
34	5.119146	192.168.1.145	reddit.com	TLSv1.2	96	Application Data
37	5.142991	reddit.com	192.168.1.145	TCP	60	https(443) → 57539 [ACK] Seq=4472 Ack=1052 Win=31744 Len=0

Connection being established

TCP connection being established, my computer sent a SYN to reddit to synchronize the connection and the sequence number is going to be 0.

Next line(18) shows reddit acknowledging (ACK) my SYN with sequence # of 0, and asks for the next sequence # of 1, which can be seen in the next line. Reddit also sends me its SYN bit with its own sequence number.

3 way handshake / TCP Handshake

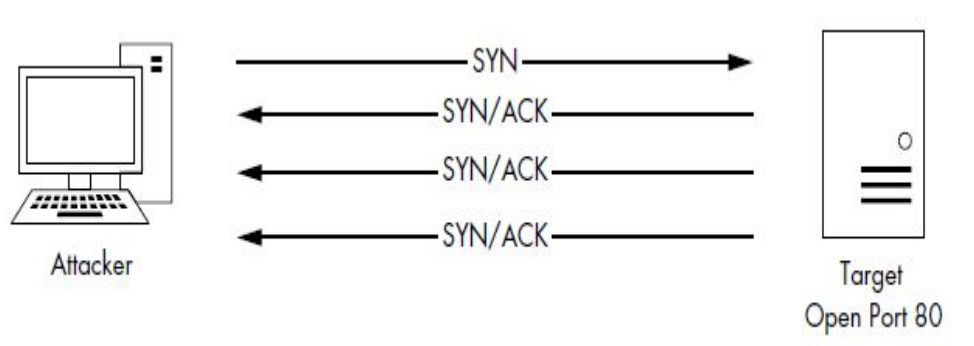
Security Applications

Reconnaissance:

- SYN Scan: aka half open scan. A fast, reliable, and quiet method to determine which ports are open on a target host. Used in conjunction with nmap, a port scanning tool.
- Attacker sends a TCP SYN packet to a range of ports on the target, as if trying to establish a channel for normal communication on the ports
- Once this packet is received by the target, one of several things may happen, as shown in the next slide.

SYN SCAN

If a service on the target's machine is listening on a port that receives the SYN packet, it will reply to the attacker with a TCP SYN/ACK packet, the second part of the TCP handshake. Now the attacker knows that port is open and a service is listening on it. Under normal circumstances, a final TCP ACK would be sent to complete the connection handshake. In this case, however, the attacker doesn't want that to happen since they won't be communicating with the host further at this point, so the attacker doesn't attempt to complete the TCP handshake.



SYN SCAN

If no service is listening on a scanned port, the attacker will not receive a SYN/ACK. Depending on the configuration of the target's operating system, the attacker could receive an RST packet in return, indicating that the port is closed. Alternatively, the attacker may receive no response at all. No response could mean that the port is filtered by an intermediate device, such as a firewall or the host itself. On the other hand, it could just be that the response was lost in transit. Thus, while this result typically indicates that the port is closed, it is ultimately inconclusive.