Inside the TCP Handshake

Inside the TCP Handshake

Betty DuBois
Chief Detective
Network Detectives, LLC
betty@netdetect.co

Tracefiles are available for download at:
www.tinyurl.com/sharkfest2015

Agenda

- Goals of the TCP handshake
- Beginning sequence numbers
- Options

Let’s Go Live!

1. Start a Wireshark capture
2. Using your favorite FTP client:

   - ftp://ftp.FreeBSD.org/pub/FreeBSD/
   - User: anonymous
   - Password: whatever

3. Click on any of the documents, let it load and then stop your capture.
4. Right click on any ftp packet, and “follow the TCP stream”
5. Or use “Betty_LionClient.pcapng” as example

Goals of the Handshake

- Is destination port open?
- Notification of opened ephemeral port
- Notification of each sides beginning sequence #
- Notification of each sides receive window size
- Option negotiation

Is the Port Open?

- Is destination port open?
- Notification of opened ephemeral port

Beginning Sequence #’s

- Each side will give their starting sequence number
- They will be different on each side
- The TCP stack uses them for byte count
- Wireshark will show relative numbers so it looks as if both sides start at zero.
- The numbers are relative to the source IP and source port (i.e. socket)
- The beauty is using them to see how deep you are into the data transfer at any given point

Option Negotiation

- Silence means NO
- MSS
- Window Scaling
- SACK
- Timestamps
- Vendor Specific Options

Silence means NO

- There is not a negative ACK/NACK
- So if a host does not support an option:
  - There is no request from the client
  - Or
  - There is no mention of the option in the server's response
- See Owen - Windows7client.pcapng

Maximum Segment Size

How much TCP Data can fit in a single packet?
Implementation is that lowest number wins

Ethernet standard frames. No jumbo frames, no 802.1q tags.

Minimum Frame = 64 Maximum Frame = 1518

On Wireshark, this displays as 60-1514, because the CRC is gone

1518	Max Size
-6	DA
-6	SA
-2	ET
-4	CRC
1500	MTU
-20	IP
-20	TCP
1460	MSS

DLC = 18 bytes

IP = 20 – 60 bytes (20 is default)
TCP = 20 – 60 bytes (20 is default)

Window Scaling

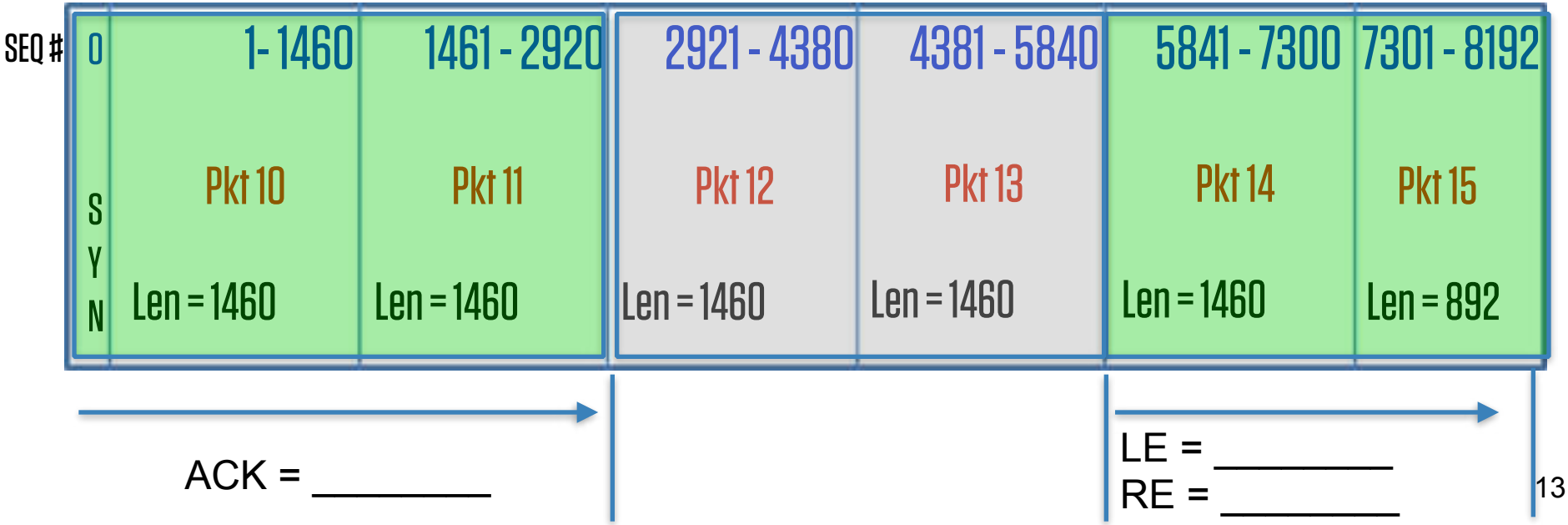
- Both sides must support, but do not have to agree on amount
- Simply a way to take advantage of bigger buffers

Selective Ack - SACK

- Both sides must support
- ACK field is always cumulative data
- SACK field is for the data after missing segments
- Room for 3 SACK sections in the options section
- Once data is sacked it can be flushed from the sender's TCP window

SACK Example

- Example is of an FTP file transfer with 8k block sizes
- 1460 + 1460 +1460 +1460 + 1460 + 892 = 8192



Timestamp

- Both sides must support
- Goals:
  - More granular Round Trip Time (RTT) measurements
  - Tie-breaker when sequence number wraps aka Protect Against Wrapped Sequence (PAWS)
- Start at a random number
- Increment by milliseconds
- RFCs
  - 1323, 3522
- Use “Betty_LionClient.pcapng” for example

Vendor Specific Options

- Some vendors use options to perform auto-discovery between their systems.
- Riverbed Steelheads are the example used here.
- Csh-wan.cap
  - TCP SYN from csh-lan is SYN+ for auto discovery
  - TCP SYN/ACK++ from the ssh-wan which says “There might be a SH in my path, but it might not be the last one.
  - TCP SYN/ACK+ from the ssh-wan after the TCP handshake between SSH LAN and server has been setup. It has the IP address of the SSH in the TCP options.
  - No ACK, that is not done. This is a pre-setup TCP session part of the Connection Pool and is now been converted into an inner channel part between the two SHs on TCP port 7800

Questions???