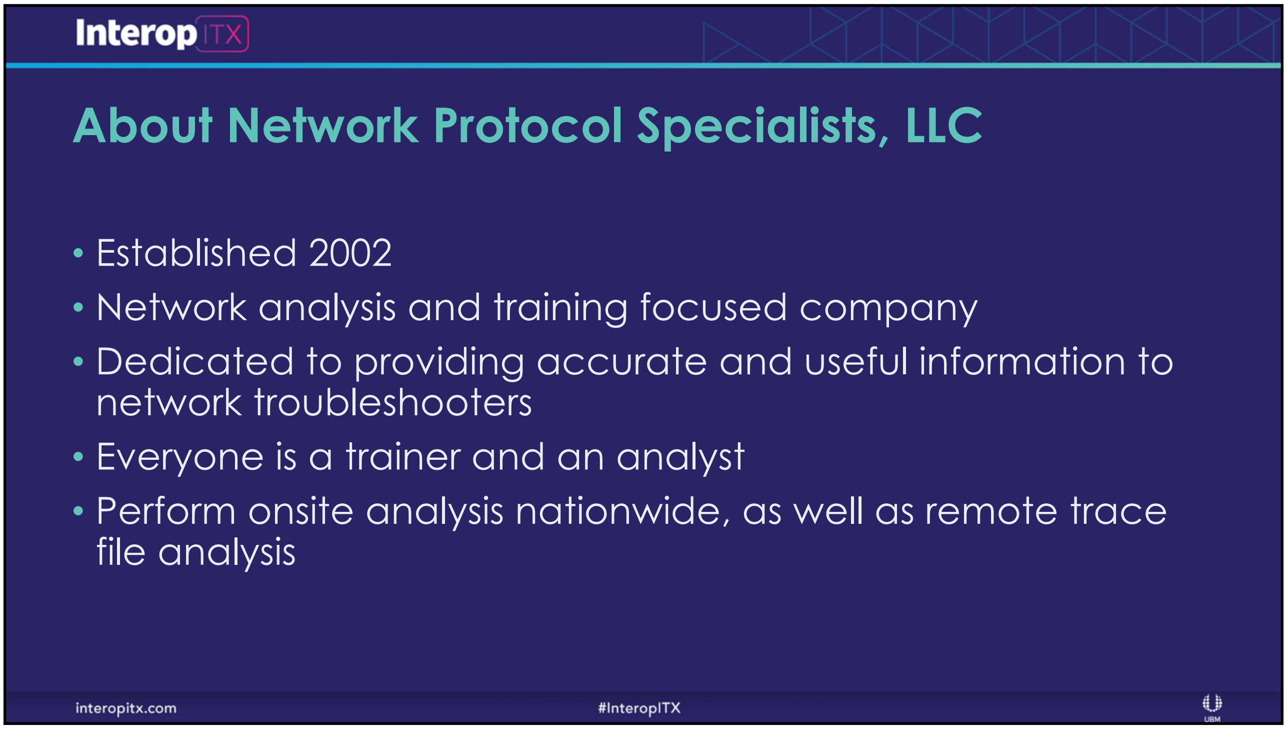
InteropITX APRIL 30 – MAY 4, 2018
THE MIRAGE | LAS VEGAS

Deep Dive Packet Analysis with Wireshark®

Mike Pennacchi
mike@nps-llc.com

interopitx.com #InteropITX

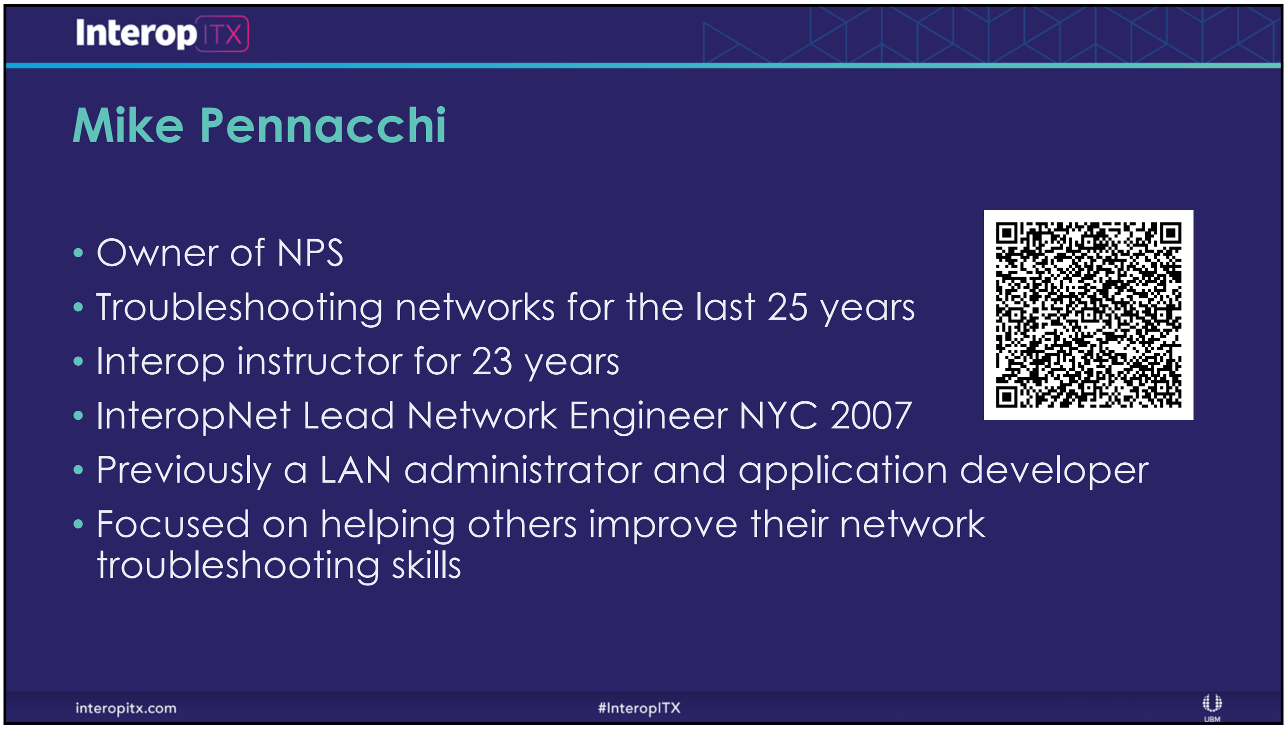


InteropITX

About Network Protocol Specialists, LLC

- Established 2002
- Network analysis and training focused company
- Dedicated to providing accurate and useful information to network troubleshooters
- Everyone is a trainer and an analyst
- Perform onsite analysis nationwide, as well as remote trace file analysis

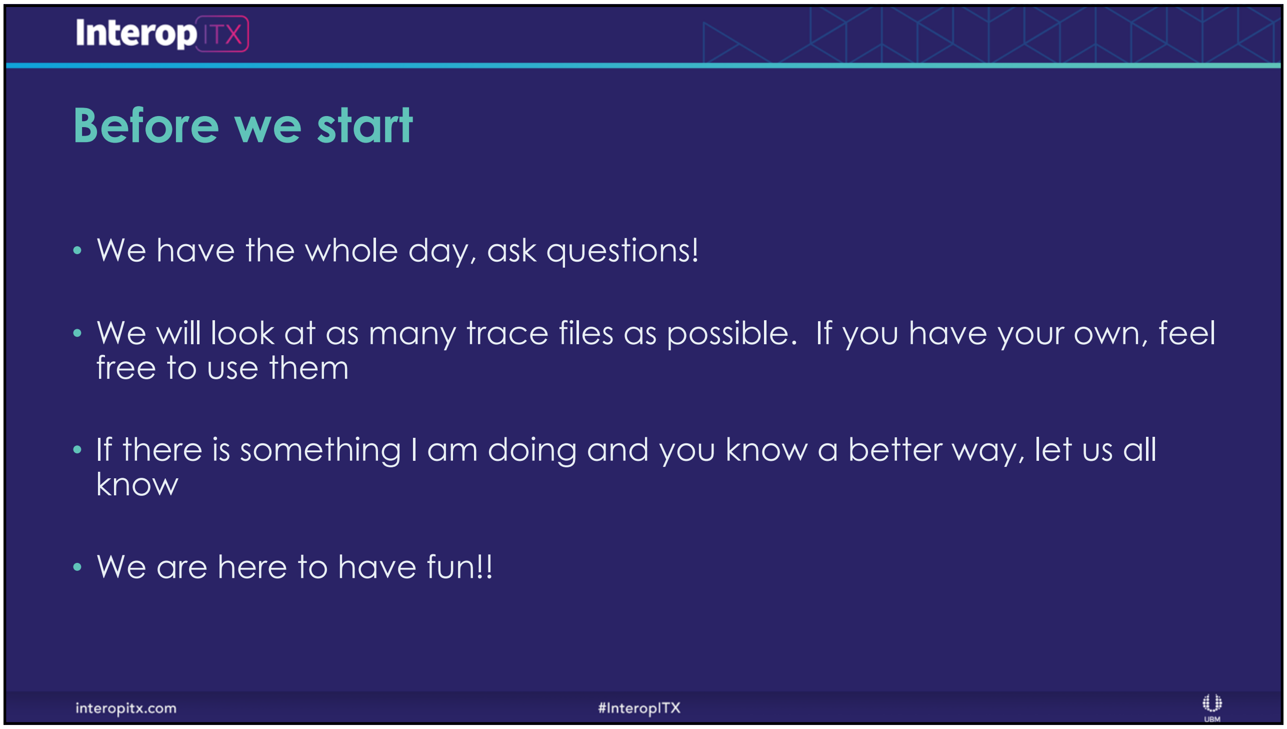
interopitx.com #InteropITX



Mike Pennacchi

- Owner of NPS
- Troubleshooting networks for the last 25 years
- Interop instructor for 23 years
- InteropNet Lead Network Engineer NYC 2007
- Previously a LAN administrator and application developer
- Focused on helping others improve their network troubleshooting skills

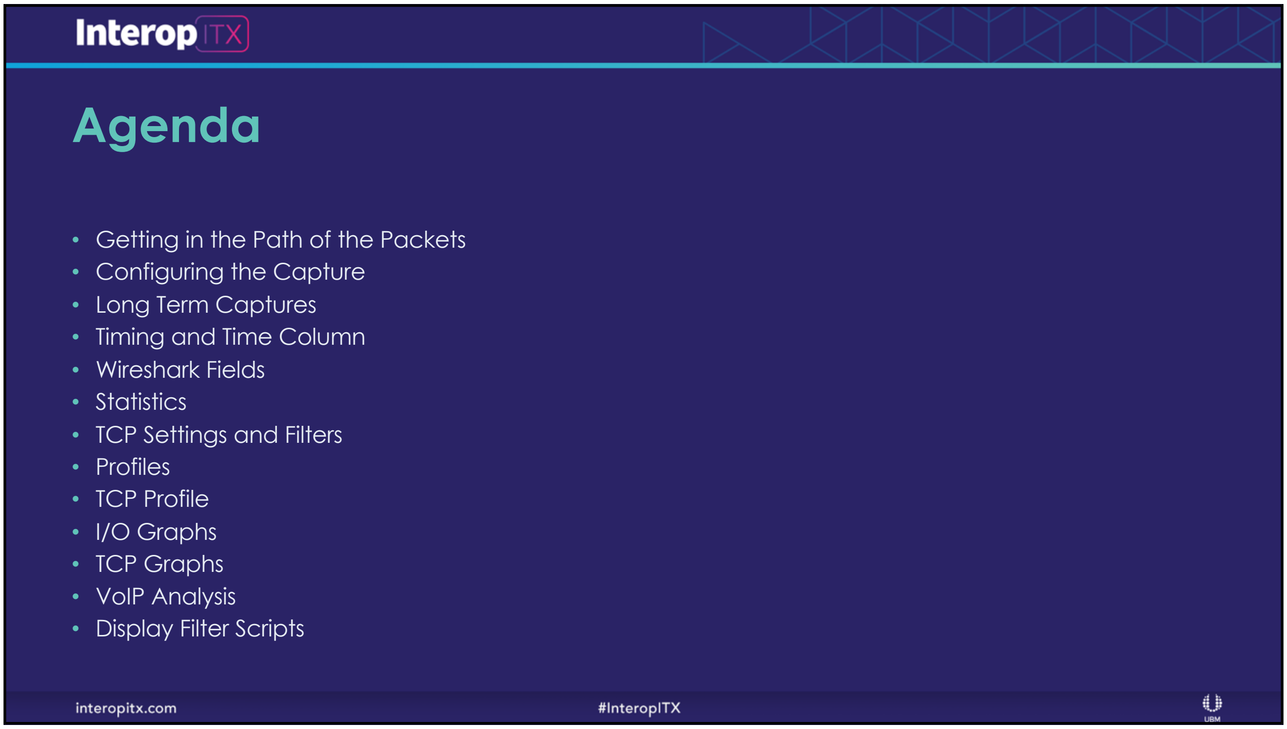
interopitx.com #InteropITX



Before we start

- We have the whole day, ask questions!
- We will look at as many trace files as possible. If you have your own, feel free to use them
- If there is something I am doing and you know a better way, let us all know
- We are here to have fun!!

interopitx.com #InteropITX



Interop ITX

Agenda

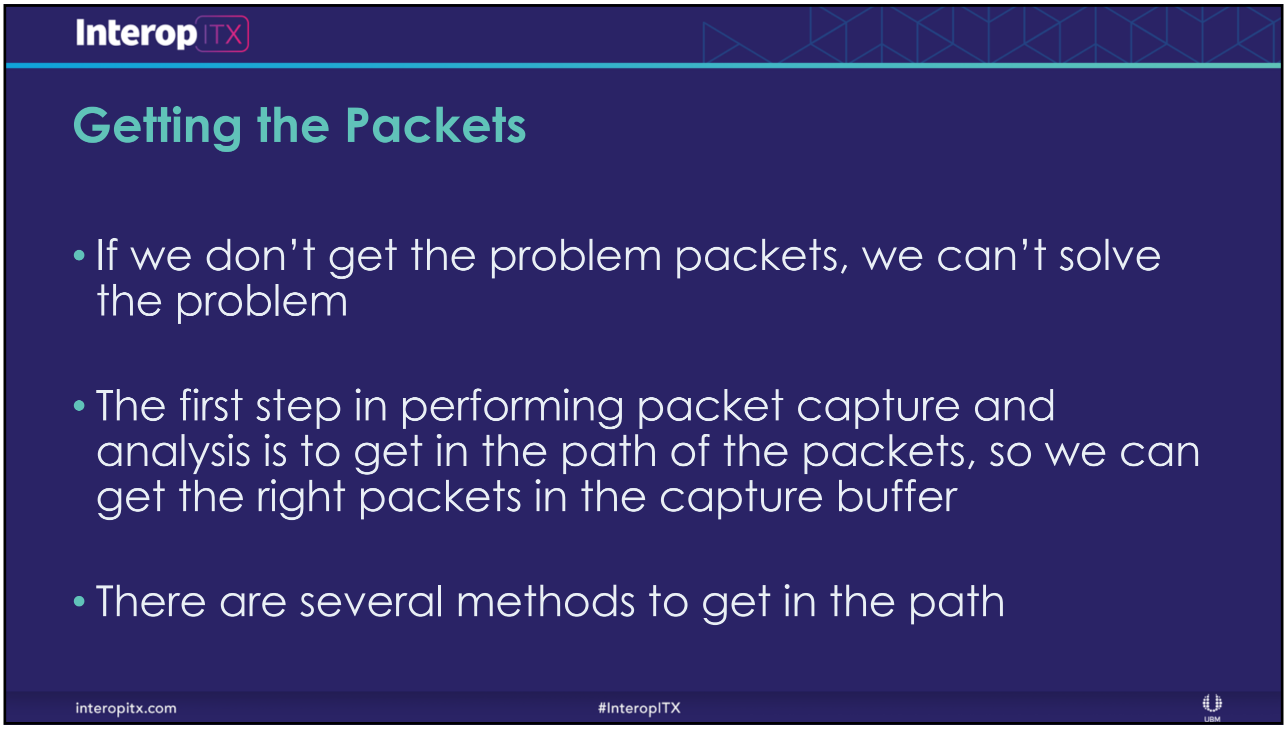
- Getting in the Path of the Packets
- Configuring the Capture
- Long Term Captures
- Timing and Time Column
- Wireshark Fields
- Statistics
- TCP Settings and Filters
- Profiles
- TCP Profile
- I/O Graphs
- TCP Graphs
- VoIP Analysis
- Display Filter Scripts

interopitx.com #InteropITX

Interop ITX APRIL 30 – MAY 4, 2018
THE MIRAGE | LAS VEGAS

Getting in the Path of the Packets

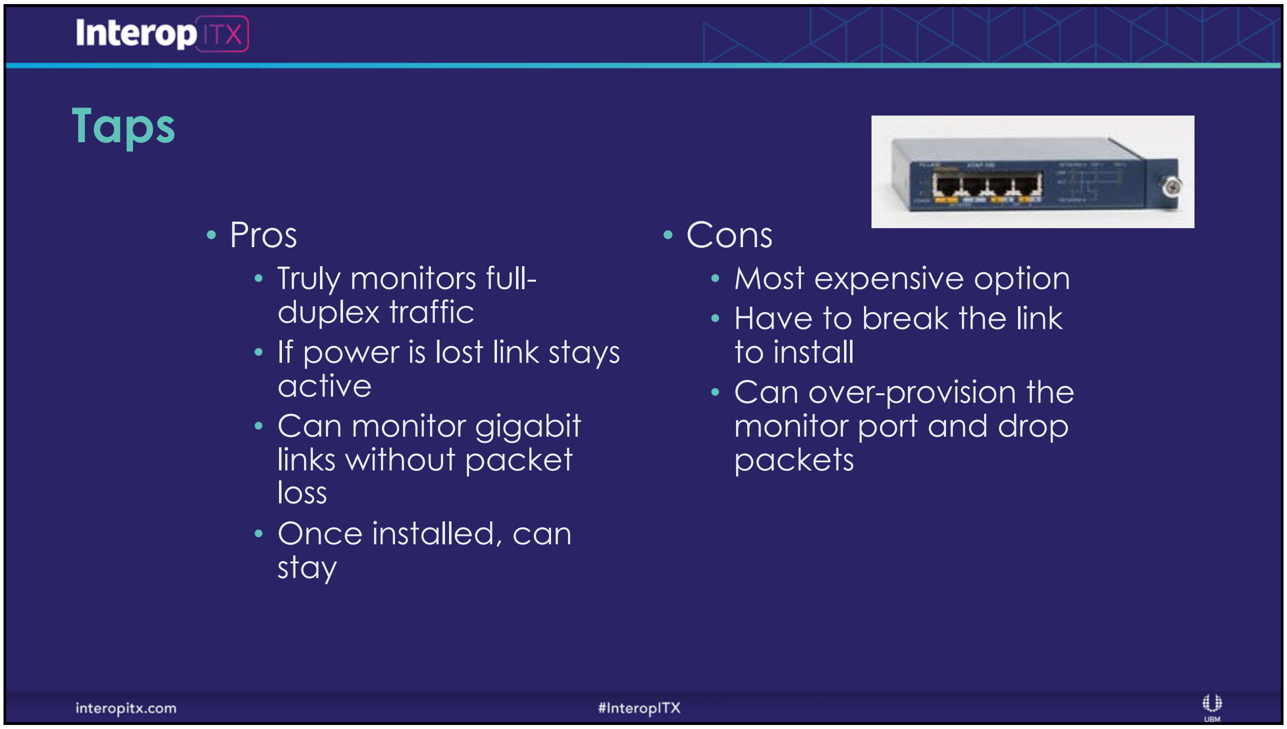
interopitx.com #InteropITX



Getting the Packets

- If we don't get the problem packets, we can't solve the problem
- The first step in performing packet capture and analysis is to get in the path of the packets, so we can get the right packets in the capture buffer
- There are several methods to get in the path

interopitx.com #InteropITX



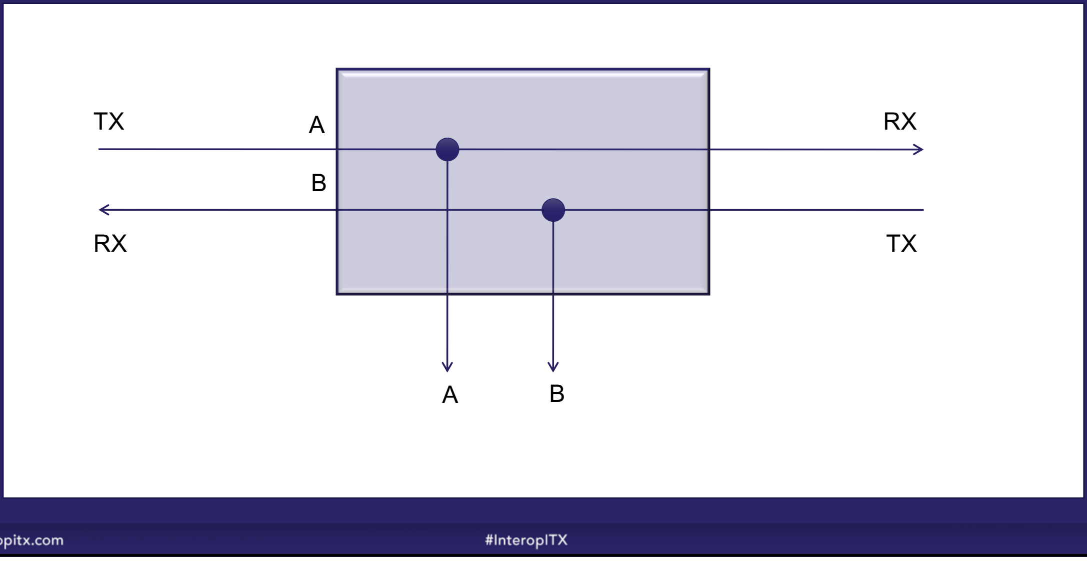
Taps

- Pros
  - Truly monitors full-duplex traffic
  - If power is lost link stays active
  - Can monitor gigabit links without packet loss
  - Once installed, can stay
- Cons
  - Most expensive option
  - Have to break the link to install
  - Can over-provision the monitor port and drop packets

interopitx.com #InteropITX

Interop ITX

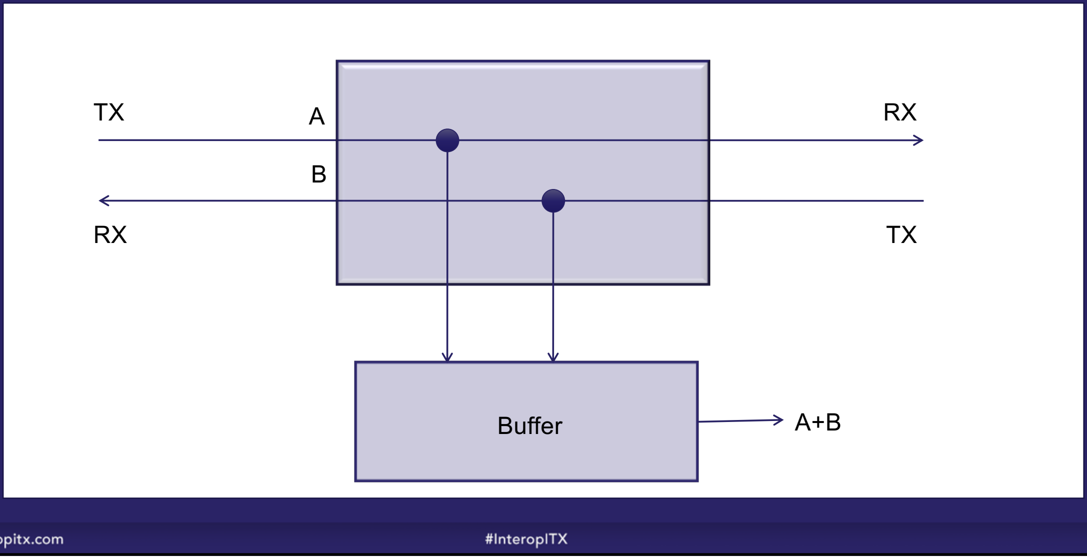
Tap Operation - Breakout



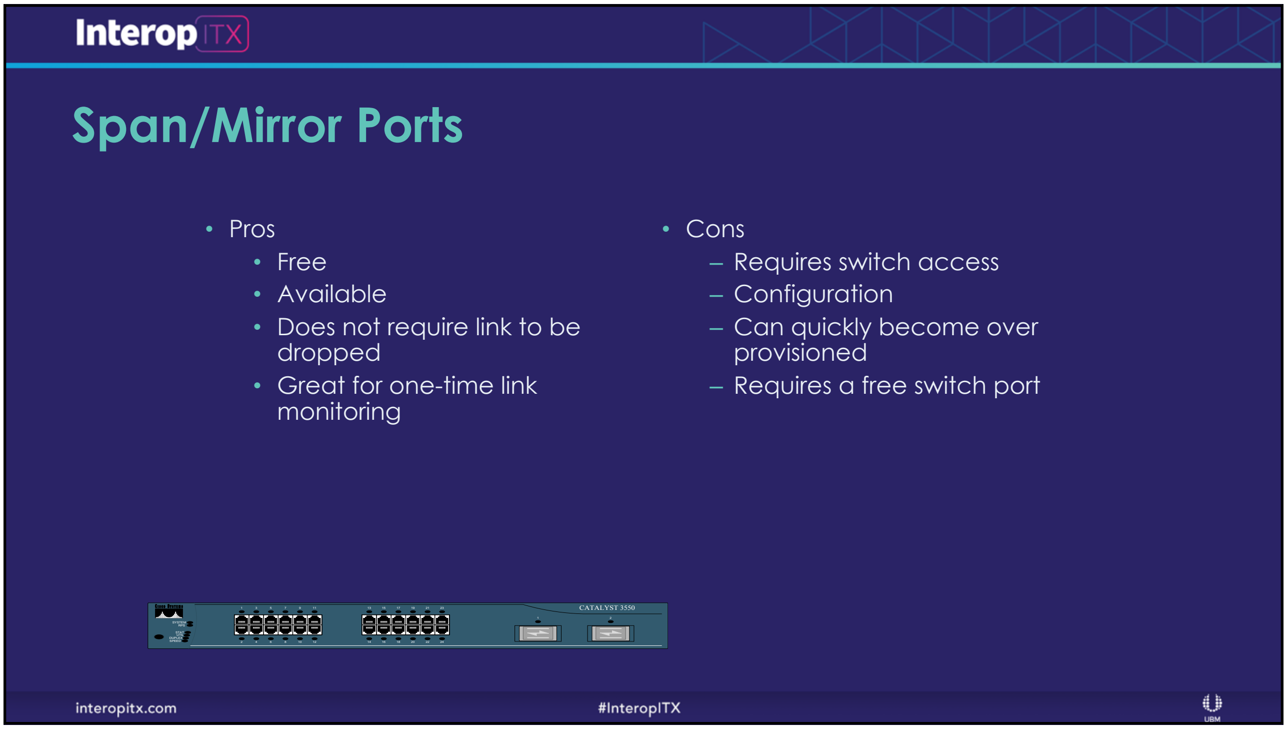
interopitx.com #InteropITX

Interop ITX

Tap Operation - Aggregation



interopitx.com #InteropITX



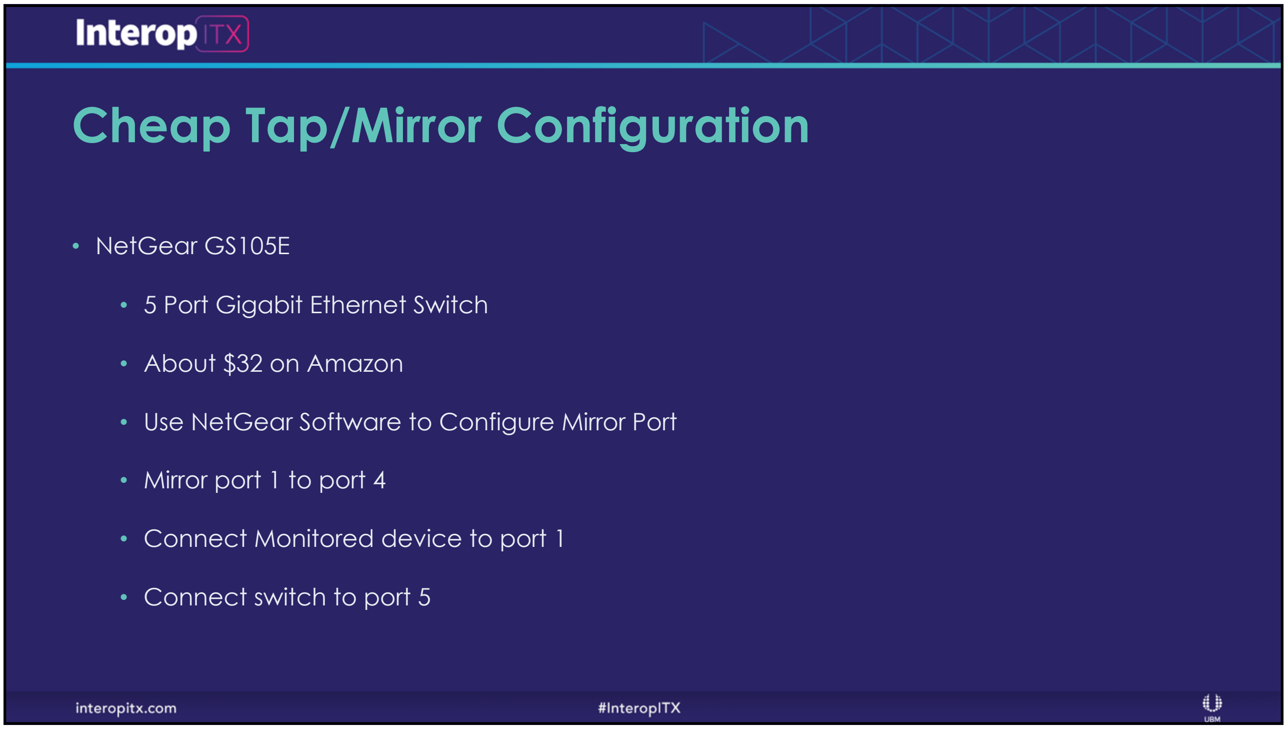
Interop ITX

Span/Mirror Ports

- Pros
  - Free
  - Available
  - Does not require link to be dropped
  - Great for one-time link monitoring
- Cons
  - – Requires switch access
  - – Configuration
  - – Can quickly become over provisioned
  - – Requires a free switch port

CATALYST 3550

interopitx.com #InteropITX



Interop ITX

Cheap Tap/Mirror Configuration

- NetGear GS105E
  - 5 Port Gigabit Ethernet Switch
  - About $32 on Amazon
  - Use NetGear Software to Configure Mirror Port
  - Mirror port 1 to port 4
  - Connect Monitored device to port 1
  - Connect switch to port 5

interopitx.com #InteropITX

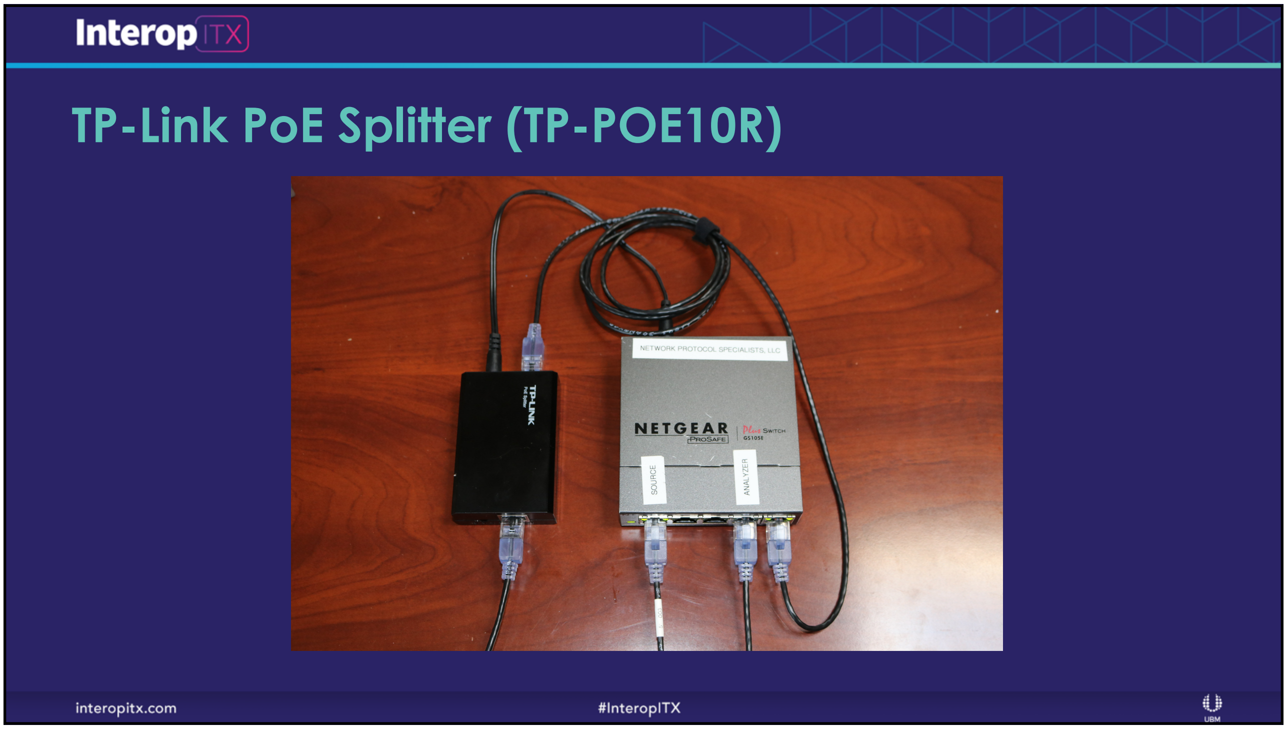
Interop ITX

NetGear GS105E

interopitx.com

#InteropITX

UBM



Interop ITX

TP-Link PoE Splitter (TP-POE10R)

interopitx.com

#InteropITX

UBM

Interop ITX

Capture to Disk Appliance

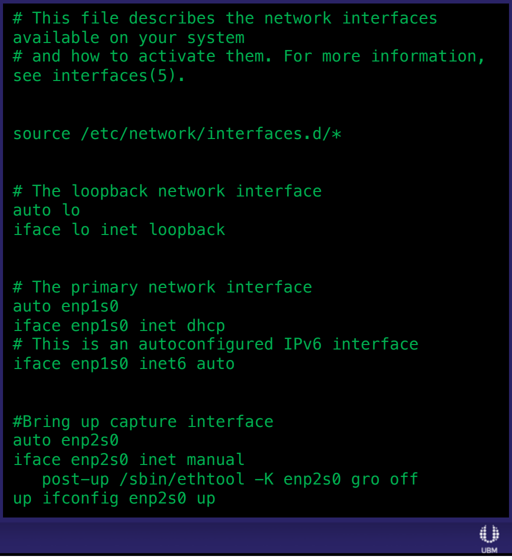
- Build Linux device to capture packets over very long periods of time
- In this case, I connected the computer to a in-line tap on my Internet connection
- Packets may be processed on the Linux box or downloaded to your PC

interopitx.com #InteropITX

Interop ITX

Generic Receive Offload (GRO) is Evil

- When it comes to packet capture, we don't want the NIC to reassemble packets for us
- I found that Ubuntu (the OS on my capture to disk appliance) automatically reassembled the packets
- After much research, I discovered that disabling GRO solved this problem



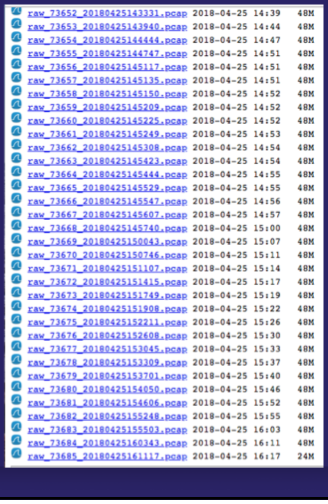
```
# This file describes the network interfaces
available on your system
# and how to activate them. For more information,
see interfaces(5).


source /etc/network/interfaces.d/*


# The loopback network interface
auto lo
iface lo inet loopback


# The primary network interface
auto enp1s0
iface enp1s0 inet dhcp
# This is an autoconfigured IPv6 interface
iface enp1s0 inet6 auto


#Bring up capture interface
auto enp2s0
iface enp2s0 inet manual
   post-up /sbin/ethtool -K enp2s0 gro off
up ifconfig enp2s0 up
```

interopitx.com #InteropITX

Interop ITX

Accessing the Files

- I setup the web server on the Ubuntu box to access the trace file directory
- This allows me to easily access the trace files from any computer
- It is a good idea to secure the web server, so only you can access the files



```
raw_73652_20180425143331.pcap 2018-04-25 14:39   48M
raw_73653_20180425143940.pcap 2018-04-25 14:44   48M
raw_73654_20180425144444.pcap 2018-04-25 14:47   48M
raw_73655_20180425144747.pcap 2018-04-25 14:51   48M
raw_73656_20180425145117.pcap 2018-04-25 14:51   48M
raw_73657_20180425145135.pcap 2018-04-25 14:51   48M
raw_73658_20180425145150.pcap 2018-04-25 14:52   48M
raw_73659_20180425145209.pcap 2018-04-25 14:52   48M
raw_73660_20180425145225.pcap 2018-04-25 14:52   48M
raw_73661_20180425145249.pcap 2018-04-25 14:53   48M
raw_73662_20180425145308.pcap 2018-04-25 14:54   48M
raw_73663_20180425145423.pcap 2018-04-25 14:54   48M
raw_73664_20180425145444.pcap 2018-04-25 14:55   48M
raw_73665_20180425145529.pcap 2018-04-25 14:55   48M
raw_73666_20180425145547.pcap 2018-04-25 14:56   48M
raw_73667_20180425145607.pcap 2018-04-25 14:57   48M
raw_73668_20180425145740.pcap 2018-04-25 15:00   48M
raw_73669_20180425150043.pcap 2018-04-25 15:07   48M
raw_73670_20180425150746.pcap 2018-04-25 15:11   48M
raw_73671_20180425151107.pcap 2018-04-25 15:14   48M
raw_73672_20180425151415.pcap 2018-04-25 15:17   48M
raw_73673_20180425151749.pcap 2018-04-25 15:19   48M
raw_73674_20180425151908.pcap 2018-04-25 15:22   48M
raw_73675_20180425152211.pcap 2018-04-25 15:26   48M
raw_73676_20180425152608.pcap 2018-04-25 15:30   48M
raw_73677_20180425153045.pcap 2018-04-25 15:33   48M
raw_73678_20180425153309.pcap 2018-04-25 15:37   48M
raw_73679_20180425153701.pcap 2018-04-25 15:40   48M
raw_73680_20180425154050.pcap 2018-04-25 15:46   48M
raw_73681_20180425154606.pcap 2018-04-25 15:52   48M
raw_73682_20180425155248.pcap 2018-04-25 15:55   48M
raw_73683_20180425155503.pcap 2018-04-25 16:03   48M
raw_73684_20180425160343.pcap 2018-04-25 16:11   48M
raw_73685_20180425161117.pcap 2018-04-25 16:17   24M
```

interopitx.com #InteropITX

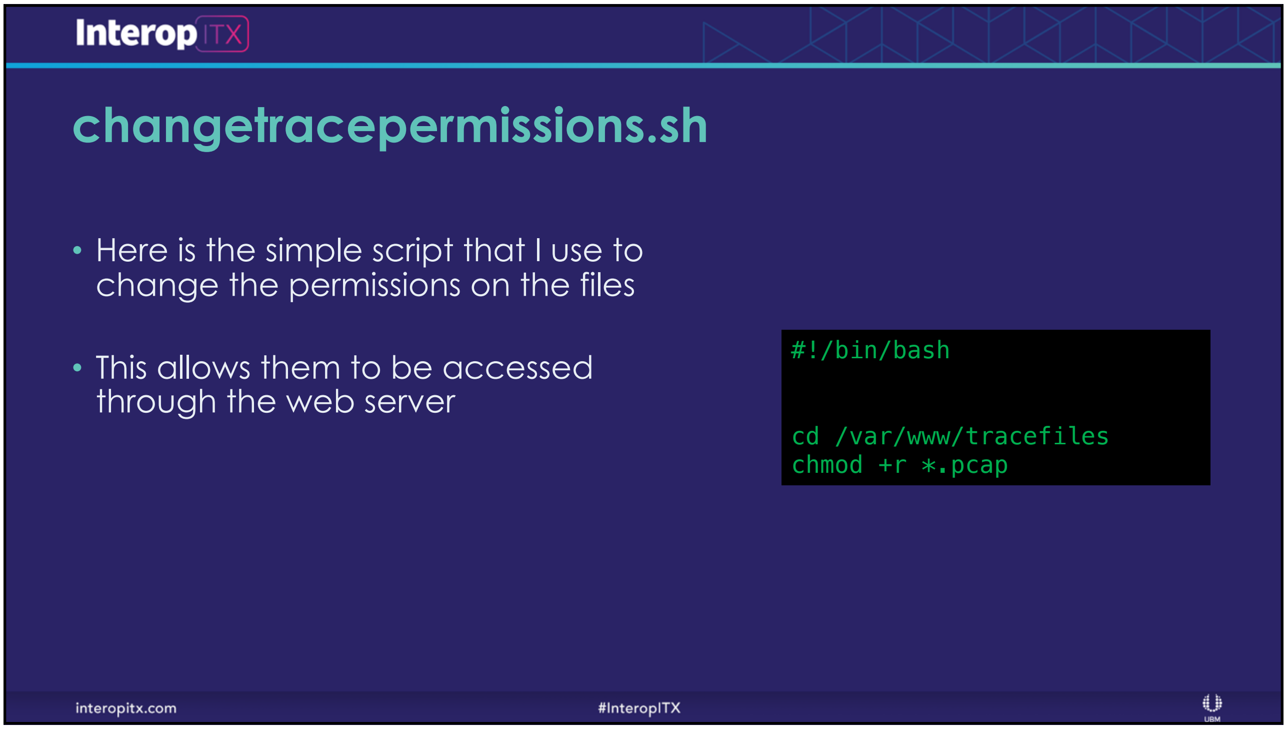
Interop ITX

Permissions are a Problem

- I found that on Ubuntu, when using tshark or dumpcap, the files can only be downloaded by the root user
- To overcome this, I added a cron job that updates the permissions on the files in the directory
- May not be the best method, but it works

```
*/1 * * * * /usr/bin/changetracepermissions.sh >/dev/null 2>&1
```

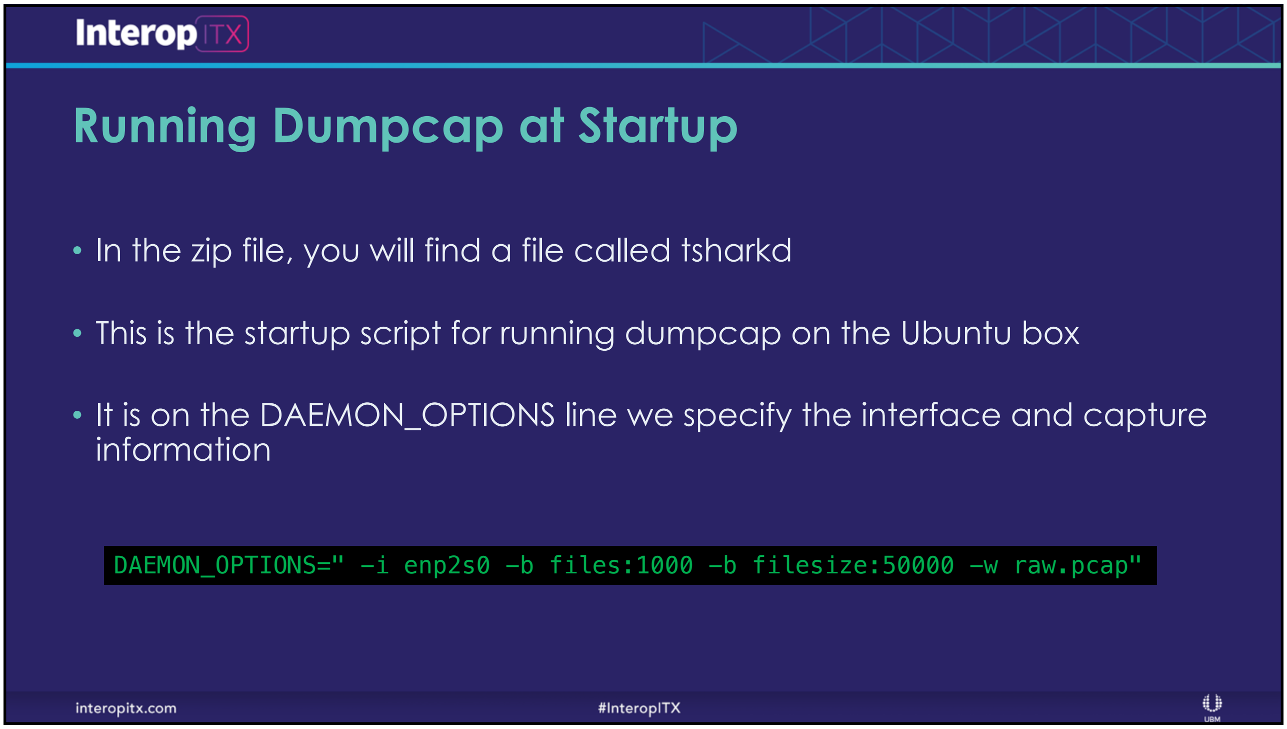
interopitx.com #InteropITX



changetracepermissions.sh

- Here is the simple script that I use to change the permissions on the files
- This allows them to be accessed through the web server

```
#!/bin/bash


cd /var/www/tracefiles
chmod +r *.pcap
```



Running Dumpcap at Startup

- In the zip file, you will find a file called tsharkd
- This is the startup script for running dumpcap on the Ubuntu box
- It is on the DAEMON_OPTIONS line we specify the interface and capture information

```
DAEMON_OPTIONS=" -i enp2s0 -b files:1000 -b filesize:50000 -w raw.pcap"
```

Interop ITX APRIL 30 – MAY 4, 2018
THE MIRAGE | LAS VEGAS

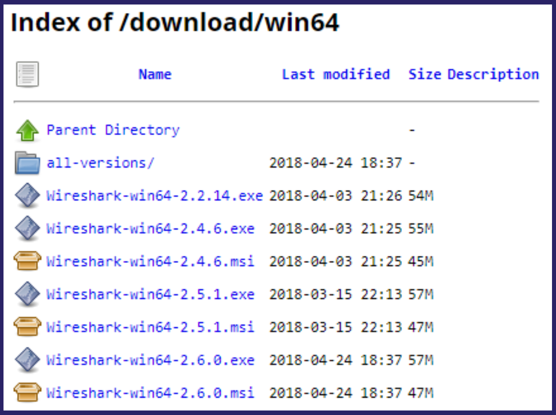
Wireshark®

interopitx.com #InteropITX

Interop ITX

Download

www.wireshark.org/download.html



Index of /download/win64

```
               Name                 Last modified    Size Description
Parent Directory                                     -
all-versions/                       2018-04-24 18:37 -
Wireshark-win64-2.2.14.exe          2018-04-03 21:26 54M
Wireshark-win64-2.4.6.exe           2018-04-03 21:25 55M
Wireshark-win64-2.4.6.msi           2018-04-03 21:25 45M
Wireshark-win64-2.5.1.exe           2018-03-15 22:13 57M
Wireshark-win64-2.5.1.msi           2018-03-15 22:13 47M
Wireshark-win64-2.6.0.exe           2018-04-24 18:37 57M
Wireshark-win64-2.6.0.msi           2018-04-24 18:37 47M
```

interopitx.com #InteropITX

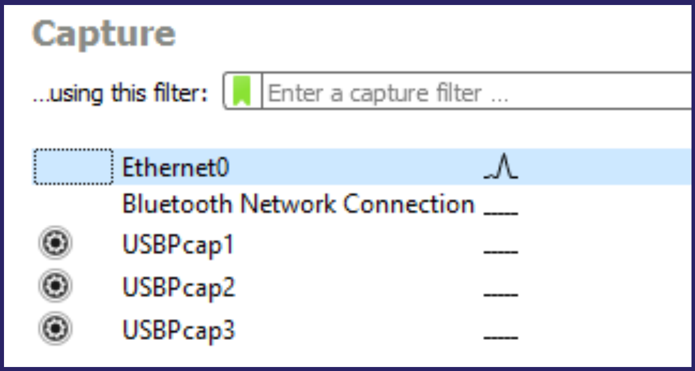
Interop ITX

Install

- During the Wireshark installation, two components are installed
  - Wireshark – Application for configuring the capture filters, setting capture parameters, displaying frames, decoding frames, producing graphs, tables, and statistics
  - Winpcap – Drivers used to capture packets off the NDIS interface

Interop ITX

Setup – Select Interface Card



Capture

...using this filter: Enter a capture filter ...

Ethernet0
Bluetooth Network Connection
USBPcap1
USBPcap2
USBPcap3

Interop ITX

Capture Options

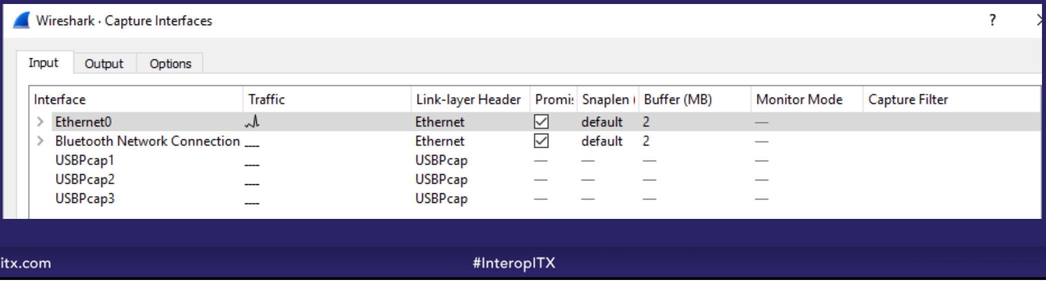
- Select Interface
- Packet Slicing
- Buffer Size
- Capture Filters
- Ring Buffer

interopitx.com #InteropITX

Interop ITX

Capture Options

- Capture Options allows you to
  - Improve the performance of the analyzer
  - Configure Capture Filters
  - Slice Packets
  - Divide the captured packets over multiple trace files



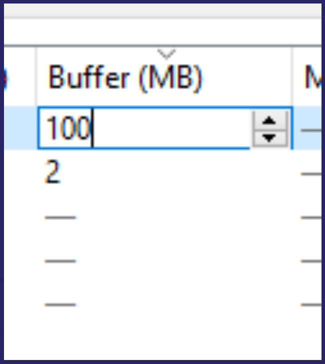
Interface	Traffic	Link-layer Header	Promi:	Snaplen (	Buffer (MB)	Monitor Mode	Capture Filter
Ethernet0		Ethernet	☑	default	2	—	
Bluetooth Network Connection		Ethernet	☑	default	2	—	
USBPcap1		USBPcap	—	—	—	—	
USBPcap2		USBPcap	—	—	—	—	
USBPcap3		USBPcap	—	—	—	—	

interopitx.com #InteropITX

Interop ITX

Buffer Size

- Not the capture buffer size
- Used to control the Kernel Memory allocated to the Wireshark process
- Increasing will significantly reduce packet loss during high speed captures
- I like using 100 megabytes



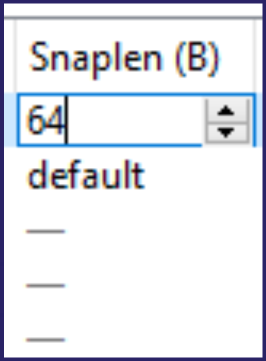
Buffer (MB)
100
2
—
—
—

interopitx.com #InteropITX

Interop ITX

Packet Slicing

- Maximize the number of packets in the capture buffer by capturing fewer bytes per packet
- Slice of confidential data, such as VoIP



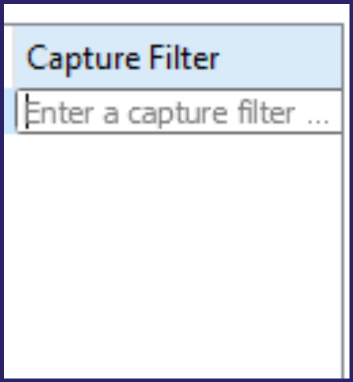
Snaplen (B)
64
default
—
—
—

interopitx.com #InteropITX

InteropITX

Capture Filter

- Helps to reduce the number of packets in the capture buffer
- Uses tcpdump filter format
- Not the same as the Display Filter format

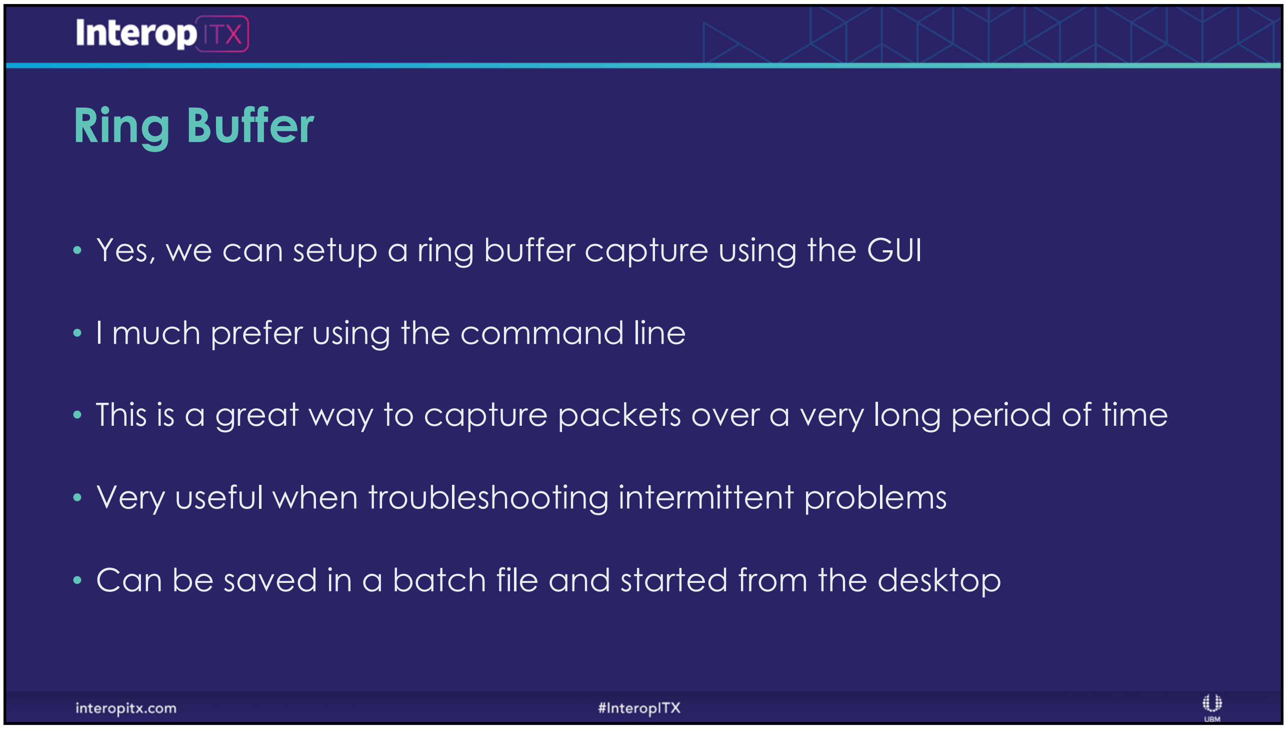


interopitx.com #InteropITX

InteropITX

Filter Type	Filter
TCP Port	tcp port [*port number*]
IP Address	host [ip address]
Ethernet Address Both Directions	ether host [*0020af123456*]
Ethernet Address Source	ether src host [*0020af123456*]
Ethernet Address Destination	ether dst host [*0020af123456*]
Address Resolution Protocol	arp
Internet Protocol (IP)	ip
IP Subnet	net *192.168.0.0/24*
From IP Subnet	src net *192.168.0.0/24*
To IP Subnet	dst net *192.168.0.0/24*
Ethernet Broadcasts	ether broadcast
Ethernet Multicasts	ether multicast
IP Broadcasts	ip broadcast
TCP SYN and FIN Packets	tcp[tcpflags] & (tcp-syn|tcp-fin) != 0

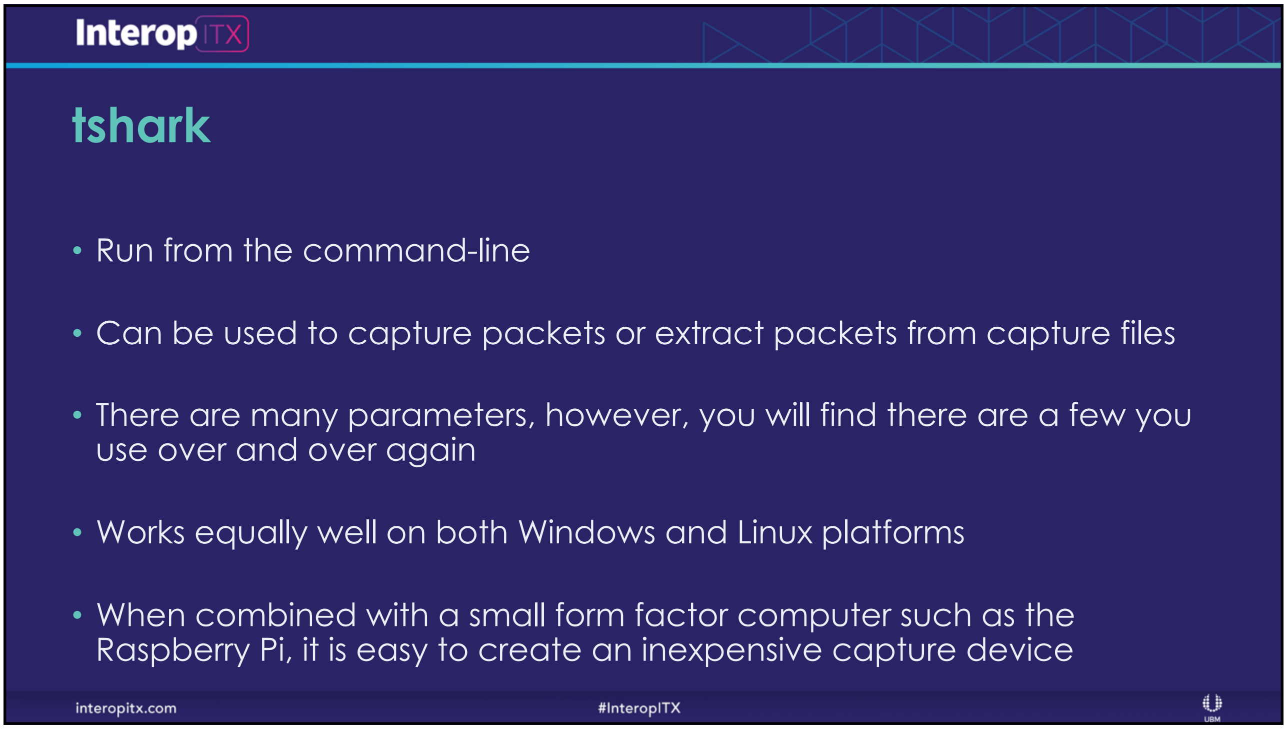
interopitx.com #InteropITX



Ring Buffer

- Yes, we can setup a ring buffer capture using the GUI
- I much prefer using the command line
- This is a great way to capture packets over a very long period of time
- Very useful when troubleshooting intermittent problems
- Can be saved in a batch file and started from the desktop

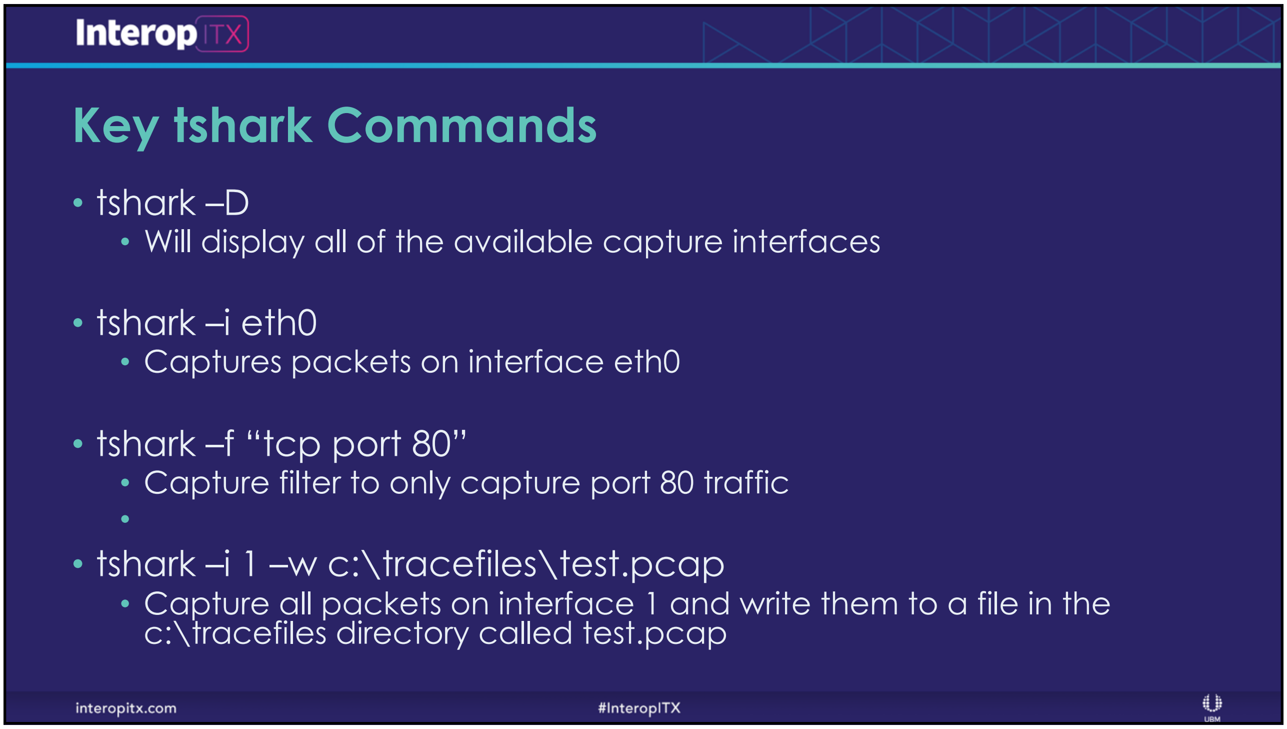
interopitx.com #InteropITX



tshark

- Run from the command-line
- Can be used to capture packets or extract packets from capture files
- There are many parameters, however, you will find there are a few you use over and over again
- Works equally well on both Windows and Linux platforms
- When combined with a small form factor computer such as the Raspberry Pi, it is easy to create an inexpensive capture device

interopitx.com #InteropITX

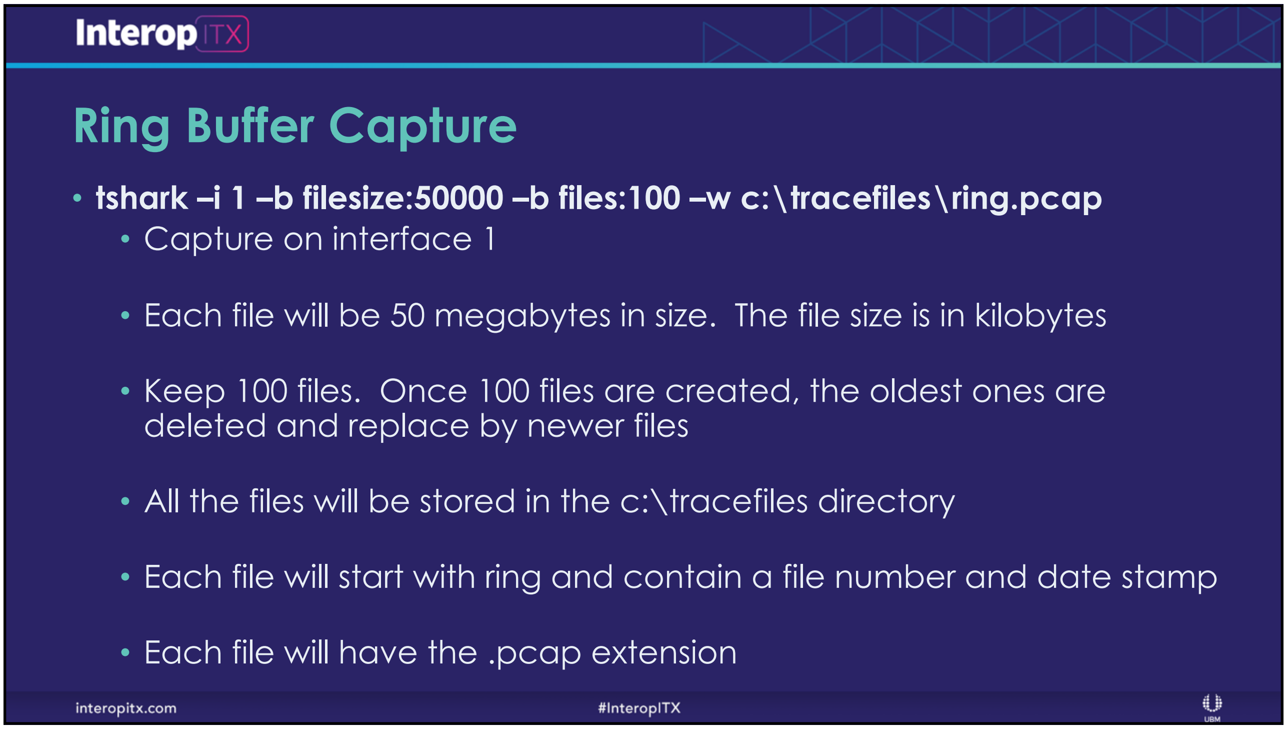


Interop ITX

Key tshark Commands

- tshark –D
  - Will display all of the available capture interfaces
- tshark –i eth0
  - Captures packets on interface eth0
- tshark –f "tcp port 80"
  - Capture filter to only capture port 80 traffic
  -
- tshark –i 1 –w c:\tracefiles\test.pcap
  - Capture all packets on interface 1 and write them to a file in the c:\tracefiles directory called test.pcap

interopitx.com #InteropITX UBM



Interop ITX

Ring Buffer Capture

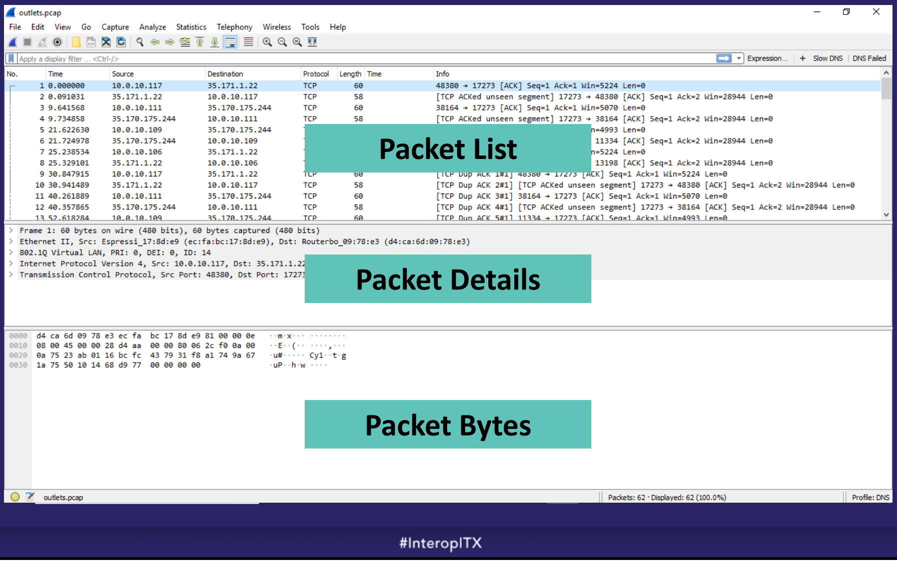
- **tshark –i 1 –b filesize:50000 –b files:100 –w c:\tracefiles\ring.pcap**
  - Capture on interface 1
  - Each file will be 50 megabytes in size. The file size is in kilobytes
  - Keep 100 files. Once 100 files are created, the oldest ones are deleted and replace by newer files
  - All the files will be stored in the c:\tracefiles directory
  - Each file will start with ring and contain a file number and date stamp
  - Each file will have the .pcap extension

interopitx.com #InteropITX UBM

Getting those Intermittent Problems

- When using the ring buffer, you can get the packets that are going across the wire when those "grey" problems occur
- What is a "grey" problem?
  - It is a problem that does not occur on a regular basis
  - It damages your creditability, since it impacts the customer and you can't seem to fix it
  - It is the kind of problem we hope will just go away, but never does

The Three Panes

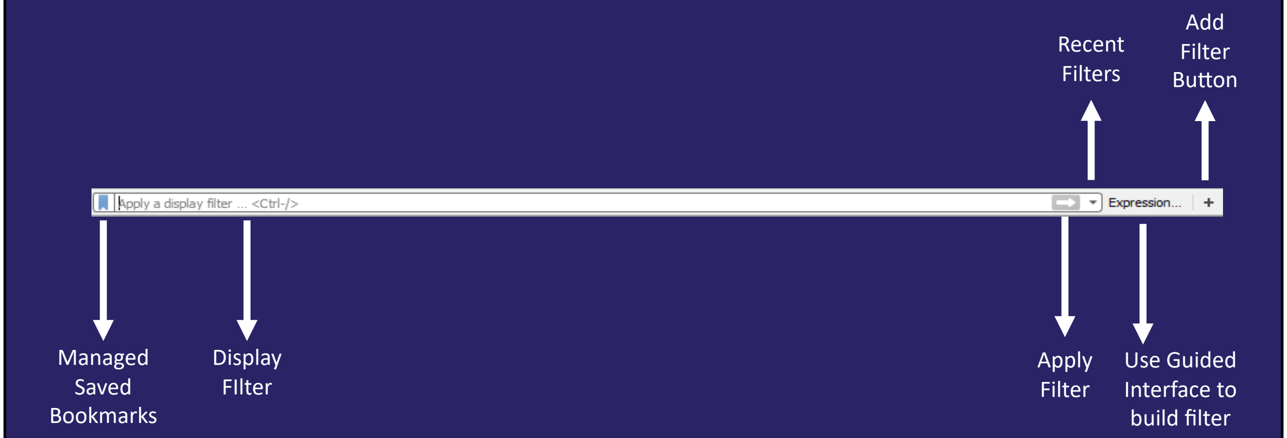

outlets.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+ Slow DNS
DNS Failed
Packet List
Packet Details
Packet Bytes
outlets.pcap
Packets: 62 · Displayed: 62 (100.0%)
Profile: DNS

```
No.   Time         Source           Destination      Protocol Length Time  Info
    1 0.000000     10.0.10.117      35.171.1.22      TCP        60         48380 → 17273 [ACK] Seq=1 Ack=1 Win=5224 Len=0
    2 0.091031     35.171.1.22      10.0.10.117      TCP        58         [TCP ACKed unseen segment] 17273 → 48380 [ACK] Seq=1 Ack=2 Win=28944 Len=0
    3 9.641568     10.0.10.111      35.170.175.244   TCP        60         38164 → 17273 [ACK] Seq=1 Ack=1 Win=5070 Len=0
    4 9.734858     35.170.175.244   10.0.10.111      TCP        58         [TCP ACKed unseen segment] 17273 → 38164 [ACK] Seq=1 Ack=2 Win=28944 Len=0
    5 21.622630    10.0.10.109      35.170.175.244                                                   n=4993 Len=0
    6 21.724978    35.170.175.244   10.0.10.109                                                      11334 [ACK] Seq=1 Ack=2 Win=28944 Len=0
    7 25.238534    10.0.10.106      35.171.1.22                                                      n=5224 Len=0
    8 25.329101    35.171.1.22      10.0.10.106                                                      13198 [ACK] Seq=1 Ack=2 Win=28944 Len=0
    9 30.847915    10.0.10.117      35.171.1.22      TCP        60         [TCP Dup ACK 1#1] 48380 → 17273 [ACK] Seq=1 Ack=1 Win=5224 Len=0
   10 30.941489    35.171.1.22      10.0.10.117      TCP        58         [TCP Dup ACK 2#1] [TCP ACKed unseen segment] 17273 → 48380 [ACK] Seq=1 Ack=2 Win=28944 Len=0
   11 40.261889    10.0.10.111      35.170.175.244   TCP        60         [TCP Dup ACK 3#1] 38164 → 17273 [ACK] Seq=1 Ack=1 Win=5070 Len=0
   12 40.357865    35.170.175.244   10.0.10.111      TCP        58         [TCP Dup ACK 4#1] [TCP ACKed unseen segment] 17273 → 38164 [ACK] Seq=1 Ack=2 Win=28944 Len=0
   13 52.618284    10.0.10.109      35.170.175.244   TCP        60         [TCP Dup ACK 5#1] 11334 → 17273 [ACK] Seq=1 Ack=1 Win=4993 Len=0
```

```
> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: Espressi_17:8d:e9 (ec:fa:bc:17:8d:e9), Dst: Routerbo_09:78:e3 (d4:ca:6d:09:78:e3)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 14
> Internet Protocol Version 4, Src: 10.0.10.117, Dst: 35.171.1.22
> Transmission Control Protocol, Src Port: 48380, Dst Port: 17273
```

```
0000  d4 ca 6d 09 78 e3 ec fa  bc 17 8d e9 81 00 00 0e   ··m·x··· ········
0010  08 00 45 00 00 28 d4 aa  00 00 80 06 2c f0 0a 00   ··E··(·· ····,···
0020  0a 75 23 ab 01 16 bc fc  43 79 31 f8 a1 74 9a 67   ·u#····· Cy1··t·g
0030  1a 75 50 10 14 68 d9 77  00 00 00 00               ·uP··h·w ····
```

The Filter Bar



Recent Filters

Add Filter Button

Managed Saved Bookmarks

Display FIlter

Apply Filter

Use Guided Interface to build filter

interopitx.com #InteropITX

Customizing Wireshark

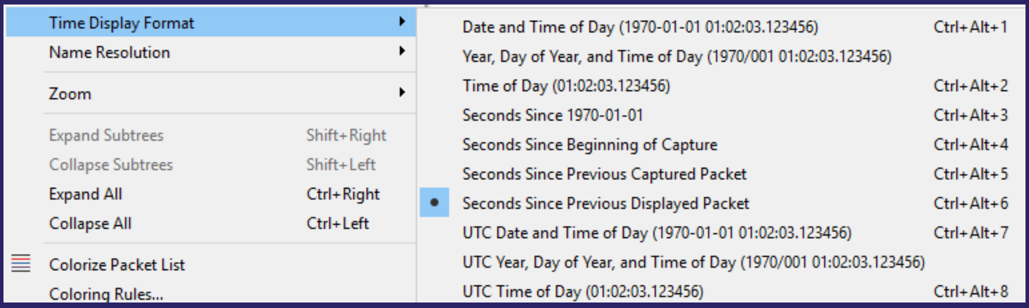
- Now that we have the packets, it is time to customize Wireshark to meet our needs
- Out of the box, it is a good analyzer
- Through the use of profiles, columns, and filters, we will make it a great analyzer
- The extent of customization will depend on your environment and applications
- Profiles can be moved from one machine to another, thereby allowing the work of one person to be used by others

interopitx.com #InteropITX

Interop ITX

Setting the Time Column

- One of the first things I customize is the time column
  1. Seconds Since Previous Displayed Packet (Ctl+Alt+6)
  2. Time of Day (Ctl+Alt+2)
  3. Set Time Reference (toggle)

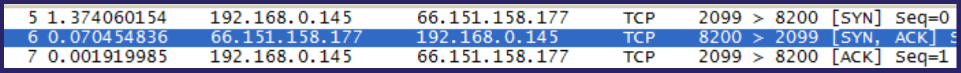

Time Display Format
Name Resolution
Zoom
Expand Subtrees Shift+Right
Collapse Subtrees Shift+Left
Expand All Ctrl+Right
Collapse All Ctrl+Left
Colorize Packet List
Coloring Rules...
Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+1
Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
Time of Day (01:02:03.123456) Ctrl+Alt+2
Seconds Since 1970-01-01 Ctrl+Alt+3
Seconds Since Beginning of Capture Ctrl+Alt+4
Seconds Since Previous Captured Packet Ctrl+Alt+5
● Seconds Since Previous Displayed Packet Ctrl+Alt+6
UTC Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+7
UTC Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
UTC Time of Day (01:02:03.123456) Ctrl+Alt+8

interopitx.com #InteropITX

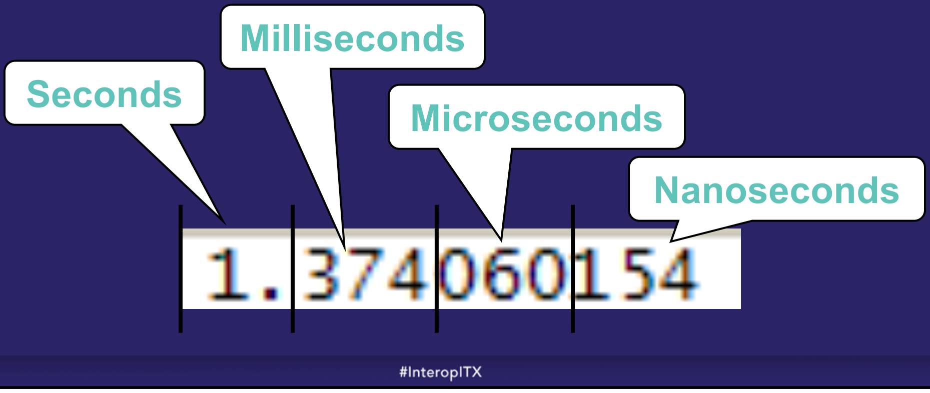
Interop ITX

Analyze – Reading the Time

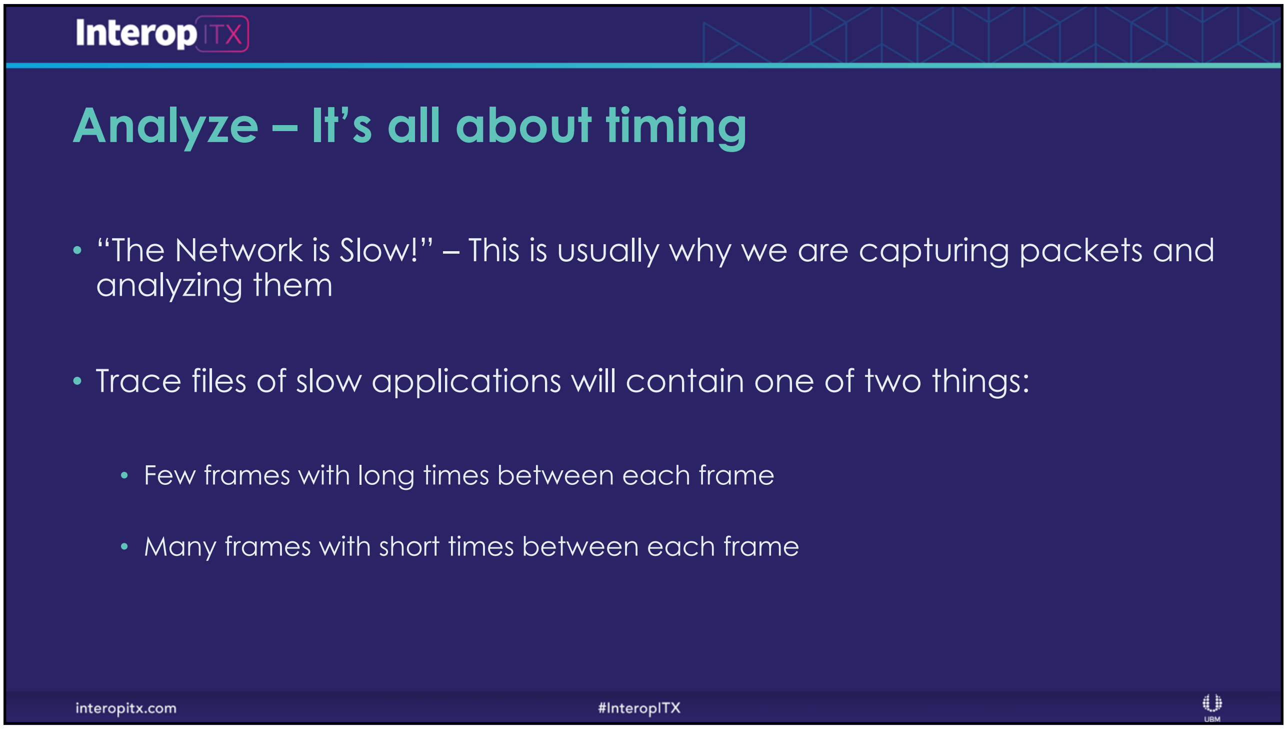
TCP Three-way Handshake



```
 5 1.374060154    192.168.0.145      66.151.158.177     TCP     2099 > 8200 [SYN] Seq=0
 6 0.070454836    66.151.158.177     192.168.0.145      TCP     8200 > 2099 [SYN, ACK] S
 7 0.001919985    192.168.0.145      66.151.158.177     TCP     2099 > 8200 [ACK] Seq=1
```



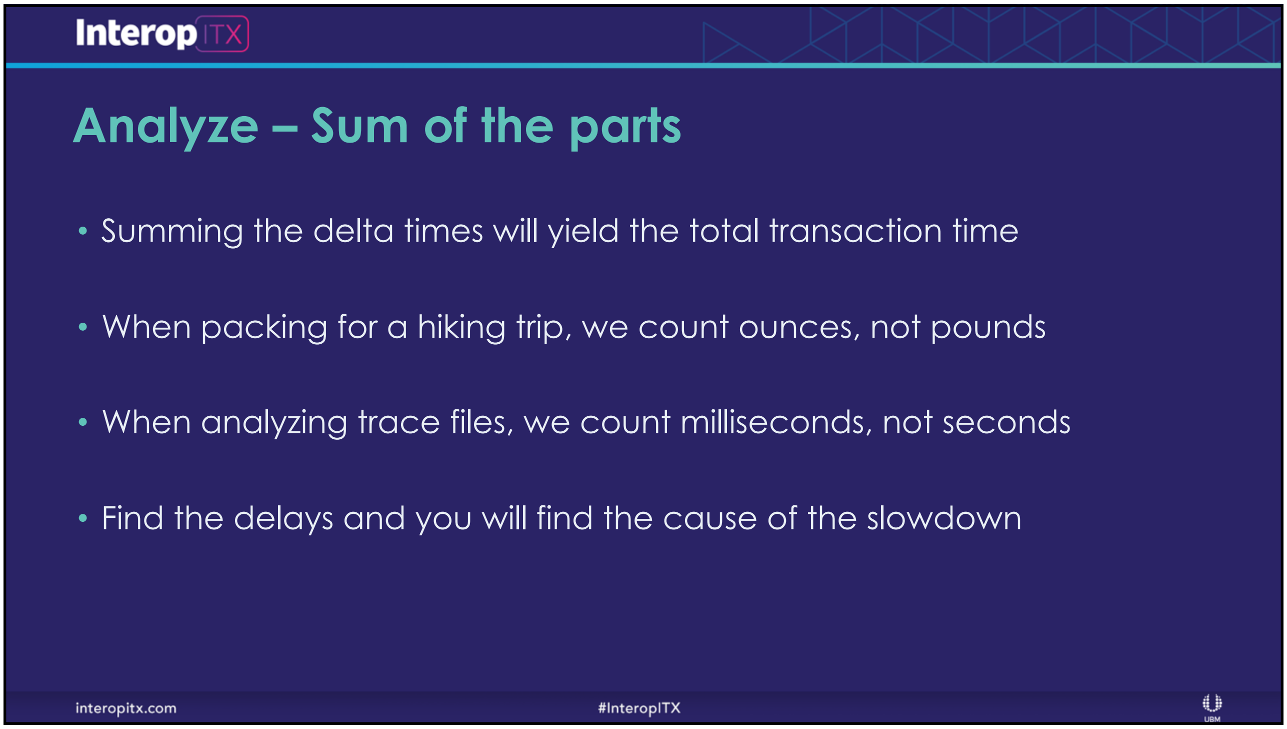
interopitx.com #InteropITX



Analyze – It's all about timing

- "The Network is Slow!" – This is usually why we are capturing packets and analyzing them
- Trace files of slow applications will contain one of two things:
  - Few frames with long times between each frame
  - Many frames with short times between each frame

interopitx.com #InteropITX



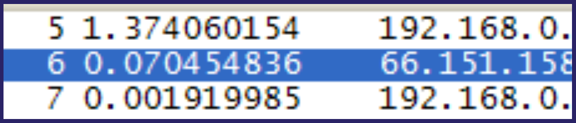
Analyze – Sum of the parts

- Summing the delta times will yield the total transaction time
- When packing for a hiking trip, we count ounces, not pounds
- When analyzing trace files, we count milliseconds, not seconds
- Find the delays and you will find the cause of the slowdown

interopitx.com #InteropITX

Interop ITX

Analyze – TCP Three Way Handshake



```
5 1.374060154    192.168.0.
6 0.070454836    66.151.158
7 0.001919985    192.168.0.
```

- Frame 5 – TCP SYN – Start of handshake, we don't care about the delta time
- Frame 6 – TCP SYN/ACK – Response from server. Represents round trip time between client and server. This took 70.454 milliseconds
- Frame 7 – TCP ACK – Sent by client. This took 1.919 milliseconds

Interop ITX

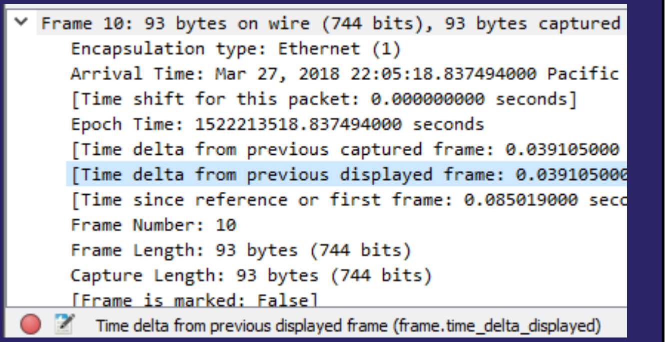
Every Field has a Name

- We can
  - Filter on those names
  - Create columns on those names
  - Export the content of those fields

Interop ITX

Field Names

- Click on a field
- Look at the status bar at the bottom of the window
- You will see the description and field name displayed
- We will use these later when creating profiles



```
⌄ Frame 10: 93 bytes on wire (744 bits), 93 bytes captured
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 27, 2018 22:05:18.837494000 Pacific
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1522213518.837494000 seconds
    [Time delta from previous captured frame: 0.039105000
    [Time delta from previous displayed frame: 0.03910500
    [Time since reference or first frame: 0.085019000 seco
    Frame Number: 10
    Frame Length: 93 bytes (744 bits)
    Capture Length: 93 bytes (744 bits)
    [Frame is marked: False]
```

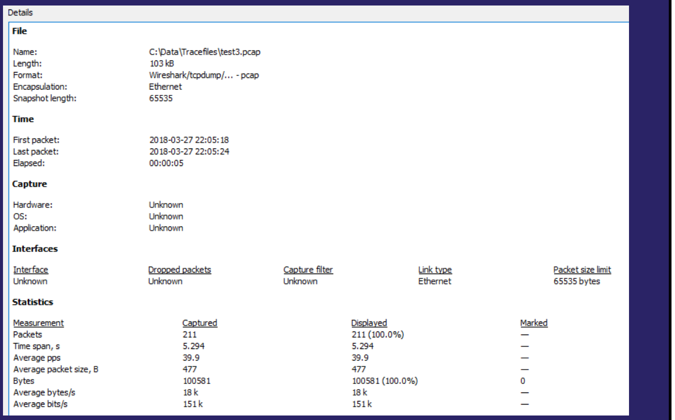
Time delta from previous displayed frame (frame.time_delta_displayed)

interopitx.com #InteropITX UBM

Interop ITX

Statistics – Capture File Properties

- Gives us a great overview of the trace file
- Shows stats on both captured and displayed packets
- Great for throughput measurements



Details

File

Name:	C:\Data\Tracefiles\test3.pcap
Length:	103 kB
Format:	Wireshark/tcpdump/... - pcap
Encapsulation:	Ethernet
Snapshot length:	65535

Time

First packet:	2018-03-27 22:05:18
Last packet:	2018-03-27 22:05:24
Elapsed:	00:00:05

Capture

Hardware:	Unknown
OS:	Unknown
Application:	Unknown

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	65535 bytes

Statistics

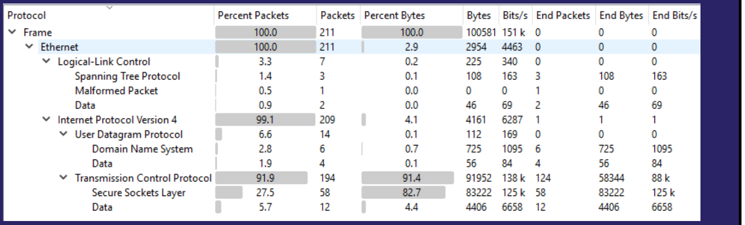
Measurement	Captured	Displayed	Marked
Packets	211	211 (100.0%)	—
Time span, s	5.294	5.294	—
Average pps	39.9	39.9	—
Average packet size, B	477	477	—
Bytes	100581	100581 (100.0%)	0
Average bytes/s	18 k	18 k	—
Average bits/s	151 k	151 k	—

interopitx.com #InteropITX UBM

Interop ITX

Statistics – Protocol Hierarchy

- Provides a breakdown of the protocols found in the trace file
- Good way to find unexpected protocols



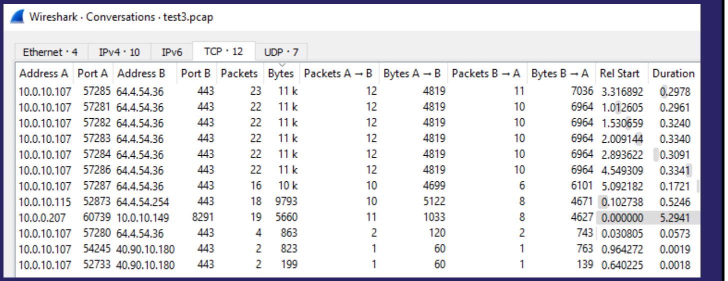
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	211	100.0	100581	151 k	0	0	0
Ethernet	100.0	211	2.9	2954	4463	0	0	0
Logical-Link Control	3.3	7	0.2	225	340	0	0	0
Spanning Tree Protocol	1.4	3	0.1	108	163	3	108	163
Malformed Packet	0.5	1	0.0	0	0	1	0	0
Data	0.9	2	0.0	46	69	2	46	69
Internet Protocol Version 4	99.1	209	4.1	4161	6287	1	1	1
User Datagram Protocol	6.6	14	0.1	112	169	0	0	0
Domain Name System	2.8	6	0.7	725	1095	6	725	1095
Data	1.9	4	0.1	56	84	4	56	84
Transmission Control Protocol	91.9	194	91.4	91952	138 k	124	58344	88 k
Secure Sockets Layer	27.5	58	82.7	83222	125 k	58	83222	125 k
Data	5.7	12	4.4	4406	6658	12	4406	6658

interopitx.com #InteropITX

Interop ITX

Statistics - Conversations

- Details each of the conversations in the trace file
- Very useful for documenting application dependencies
- Can be used to drill down into the trace



Wireshark · Conversations · test3.pcap

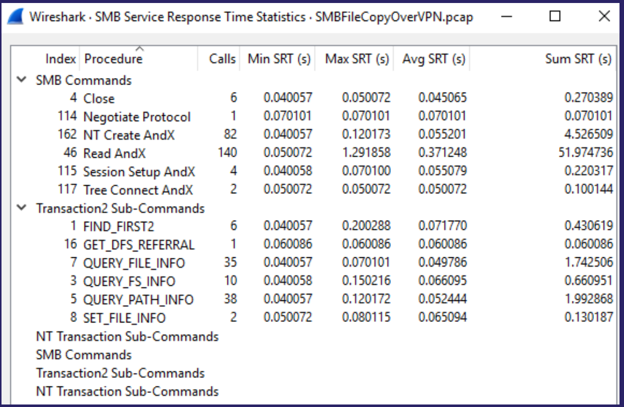
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
10.0.10.107	57285	64.4.54.36	443	23	11 k	12	4819	11	7036	3.316892	0.2978
10.0.10.107	57281	64.4.54.36	443	22	11 k	12	4819	10	6964	1.012605	0.2961
10.0.10.107	57282	64.4.54.36	443	22	11 k	12	4819	10	6964	1.530659	0.3240
10.0.10.107	57283	64.4.54.36	443	22	11 k	12	4819	10	6964	2.009144	0.3340
10.0.10.107	57284	64.4.54.36	443	22	11 k	12	4819	10	6964	2.893622	0.3091
10.0.10.107	57286	64.4.54.36	443	22	11 k	12	4819	10	6964	4.549309	0.3341
10.0.10.107	57287	64.4.54.36	443	16	10 k	10	4699	6	6101	5.092182	0.1721
10.0.10.115	52873	64.4.54.254	443	18	9793	10	5122	8	4671	0.102738	0.5246
10.0.0.207	60739	10.0.10.149	8291	19	5660	11	1033	8	4627	0.000000	5.2941
10.0.10.107	57280	64.4.54.36	443	4	863	2	120	2	743	0.030805	0.0573
10.0.10.107	54245	40.90.10.180	443	2	823	1	60	1	763	0.964272	0.0019
10.0.10.107	52733	40.90.10.180	443	2	199	1	60	1	139	0.640225	0.0018

interopitx.com #InteropITX

Interop ITX

Statistics – Service Response Time

- Calculates the Minimum, Maximum, and Average response times for each of the SMB calls
- Capturing on both ends of the WAN allows you to determine the impact of the WAN on response time



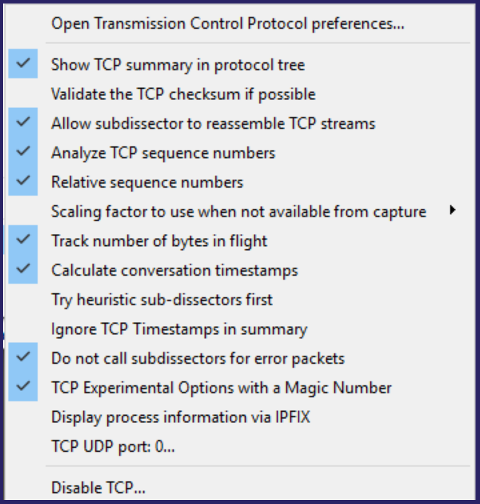
Wireshark · SMB Service Response Time Statistics · SMBFileCopyOverVPN.pcap

Index	Procedure	Calls	Min SRT (s)	Max SRT (s)	Avg SRT (s)	Sum SRT (s)
SMB Commands						
4	Close	6	0.040057	0.050072	0.045065	0.270389
114	Negotiate Protocol	1	0.070101	0.070101	0.070101	0.070101
162	NT Create AndX	82	0.040057	0.120173	0.055201	4.526509
46	Read AndX	140	0.050072	1.291858	0.371248	51.974736
115	Session Setup AndX	4	0.040058	0.070100	0.055079	0.220317
117	Tree Connect AndX	2	0.050072	0.050072	0.050072	0.100144
Transaction2 Sub-Commands						
1	FIND_FIRST2	6	0.040057	0.200288	0.071770	0.430619
16	GET_DFS_REFERRAL	1	0.060086	0.060086	0.060086	0.060086
7	QUERY_FILE_INFO	35	0.040057	0.070101	0.049786	1.742506
3	QUERY_FS_INFO	10	0.040058	0.150216	0.066095	0.660951
5	QUERY_PATH_INFO	38	0.040057	0.120172	0.052444	1.992868
8	SET_FILE_INFO	2	0.050072	0.080115	0.065094	0.130187
NT Transaction Sub-Commands						
SMB Commands						
Transaction2 Sub-Commands						
NT Transaction Sub-Commands						

Interop ITX

TCP Settings

- There are a number of changes we can make to the TCP settings in Wireshark to give us greater visibility into what is going on
- While the default settings are good, there are some better settings



Open Transmission Control Protocol preferences...

✓ Show TCP summary in protocol tree

Validate the TCP checksum if possible

✓ Allow subdissector to reassemble TCP streams

✓ Analyze TCP sequence numbers

✓ Relative sequence numbers

Scaling factor to use when not available from capture ▸

✓ Track number of bytes in flight

✓ Calculate conversation timestamps

Try heuristic sub-dissectors first

Ignore TCP Timestamps in summary

✓ Do not call subdissectors for error packets

✓ TCP Experimental Options with a Magic Number

Display process information via IPFIX

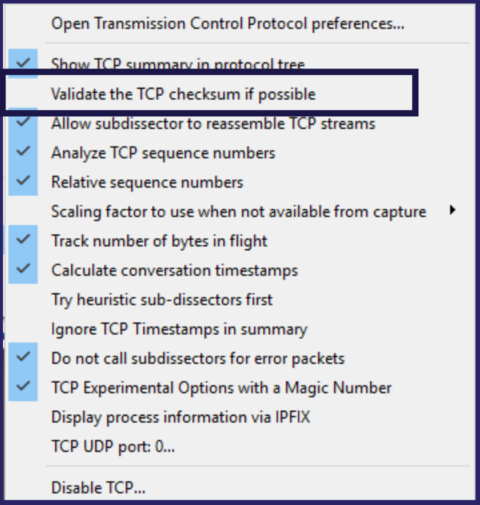
TCP UDP port: 0...

Disable TCP...

Interop ITX

TCP Settings – TCP Checksum

- This is disabled by default
- Enable if you are not capturing on one of the endpoints
- Leave disabled if you are capturing on one of the endpoints
- Why? TCP Checksum offloading will cause every packet transmitted by the device to show up with a bad TCP Checksum



Open Transmission Control Protocol preferences...

✓ Show TCP summary in protocol tree

Validate the TCP checksum if possible

✓ Allow subdissector to reassemble TCP streams

✓ Analyze TCP sequence numbers

✓ Relative sequence numbers

Scaling factor to use when not available from capture ▸

✓ Track number of bytes in flight

✓ Calculate conversation timestamps

Try heuristic sub-dissectors first

Ignore TCP Timestamps in summary

✓ Do not call subdissectors for error packets

✓ TCP Experimental Options with a Magic Number

Display process information via IPFIX

TCP UDP port: 0...

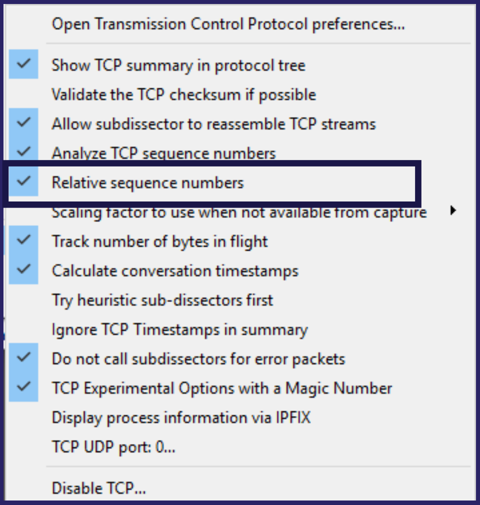
Disable TCP...

interopitx.com #InteropITX

Interop ITX

TCP Settings – Relative Sequence Numbers

- This is enabled by default
- TCP sequence numbers do not start at 1, contrary to what you might see in Wireshark
- Relative sequence numbers make life easier
- If you are trying to find the same sequence number in two traces captured in different locations, disable this setting



Open Transmission Control Protocol preferences...

✓ Show TCP summary in protocol tree

Validate the TCP checksum if possible

✓ Allow subdissector to reassemble TCP streams

✓ Analyze TCP sequence numbers

✓ Relative sequence numbers

Scaling factor to use when not available from capture ▸

✓ Track number of bytes in flight

✓ Calculate conversation timestamps

Try heuristic sub-dissectors first

Ignore TCP Timestamps in summary

✓ Do not call subdissectors for error packets

✓ TCP Experimental Options with a Magic Number

Display process information via IPFIX

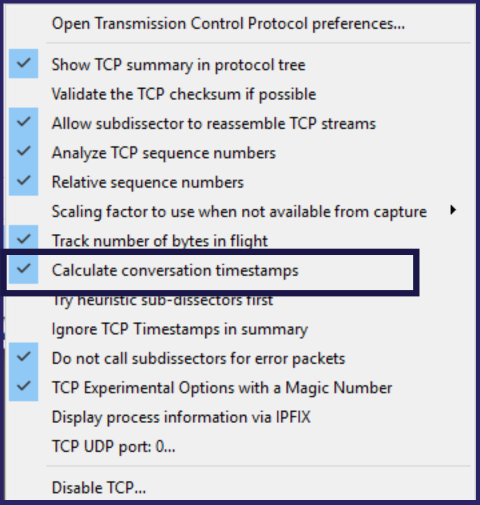
TCP UDP port: 0...

Disable TCP...

interopitx.com #InteropITX

TCP Settings – Calculate conversation timestamps

- This is disabled by default
- When enabled, we can use the field tcp.time to measure the time between two TCP frames
- Very useful when trying to find slow response times



Open Transmission Control Protocol preferences...

✓ Show TCP summary in protocol tree

Validate the TCP checksum if possible

✓ Allow subdissector to reassemble TCP streams

✓ Analyze TCP sequence numbers

✓ Relative sequence numbers

Scaling factor to use when not available from capture ▸

✓ Track number of bytes in flight

✓ Calculate conversation timestamps

Try heuristic sub-dissectors first

Ignore TCP Timestamps in summary

✓ Do not call subdissectors for error packets

✓ TCP Experimental Options with a Magic Number

Display process information via IPFIX

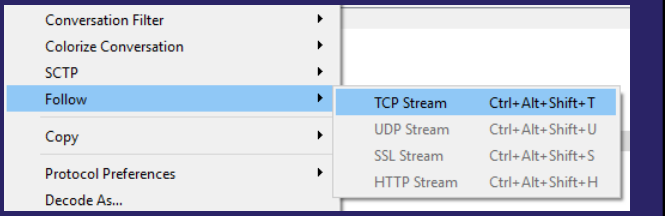
TCP UDP port: 0...

Disable TCP...

interopitx.com #InteropITX

Follow TCP Stream

- Select any frame that is part of a conversation of interest
- Right click on the frame
- Select Follow TCP Stream
- Wireshark creates a filter on the IP address pair and port numbers
- The data portion of the conversation will be assembled into a text window

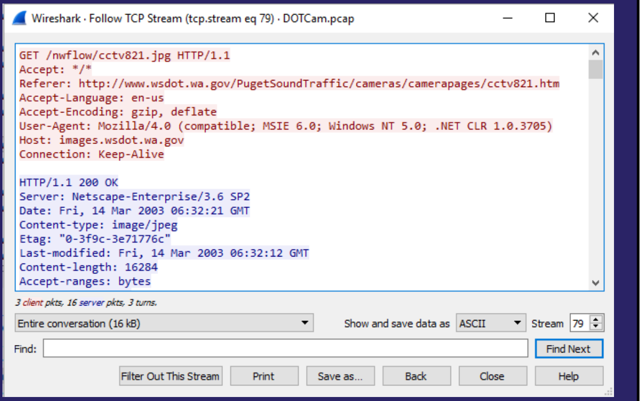

Conversation Filter ▸
Colorize Conversation ▸
SCTP ▸
Follow ▸
Copy ▸
Protocol Preferences ▸
Decode As...
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
SSL Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H

interopitx.com #InteropITX

Interop ITX

Follow TCP Stream

- In addition to creating a filter on the TCP socket, the data in the TCP portion of the packet is reassembled
- Great way to focus on a specific conversation and see the data that is transferred between the devices


Wireshark · Follow TCP Stream (tcp.stream eq 79) · DOTCam.pcap

```
GET /nwflow/cctv821.jpg HTTP/1.1
Accept: */*
Referer: http://www.wsdot.wa.gov/PugetSoundTraffic/cameras/camerapages/cctv821.htm
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET CLR 1.0.3705)
Host: images.wsdot.wa.gov
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: Netscape-Enterprise/3.6 SP2
Date: Fri, 14 Mar 2003 06:32:21 GMT
Content-type: image/jpeg
Etag: "0-3f9c-3e71776c"
Last-modified: Fri, 14 Mar 2003 06:32:12 GMT
Content-length: 16284
Accept-ranges: bytes
```

3 client pkts, 16 server pkts, 3 turns.

Entire conversation (16 kB) ▾ Show and save data as ASCII ▾ Stream 79

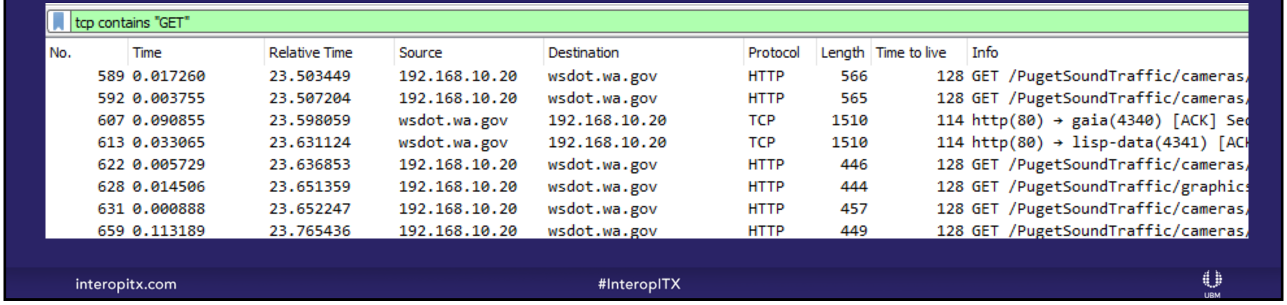
Find:

interopitx.com #InteropITX

Interop ITX

TCP Contains

- This filter is very useful in application environments
- If a certain call or file is having problems in the application, you can search for the call or file name using the TCP contains filter



tcp contains "GET"

```
No.     Time        Relative Time  Source          Destination     Protocol  Length  Time to live  Info
    589 0.017260    23.503449      192.168.10.20   wsdot.wa.gov    HTTP       566         128 GET /PugetSoundTraffic/cameras/
    592 0.003755    23.507204      192.168.10.20   wsdot.wa.gov    HTTP       565         128 GET /PugetSoundTraffic/cameras/
    607 0.090855    23.598059      wsdot.wa.gov    192.168.10.20   TCP       1510         114 http(80) → gaia(4340) [ACK] Se
    613 0.033065    23.631124      wsdot.wa.gov    192.168.10.20   TCP       1510         114 http(80) → lisp-data(4341) [ACK
    622 0.005729    23.636853      192.168.10.20   wsdot.wa.gov    HTTP       446         128 GET /PugetSoundTraffic/cameras/
    628 0.014506    23.651359      192.168.10.20   wsdot.wa.gov    HTTP       444         128 GET /PugetSoundTraffic/graphics
    631 0.000888    23.652247      192.168.10.20   wsdot.wa.gov    HTTP       457         128 GET /PugetSoundTraffic/cameras/
    659 0.113189    23.765436      192.168.10.20   wsdot.wa.gov    HTTP       449         128 GET /PugetSoundTraffic/cameras/
```

interopitx.com #InteropITX

Interop ITX

Profiles

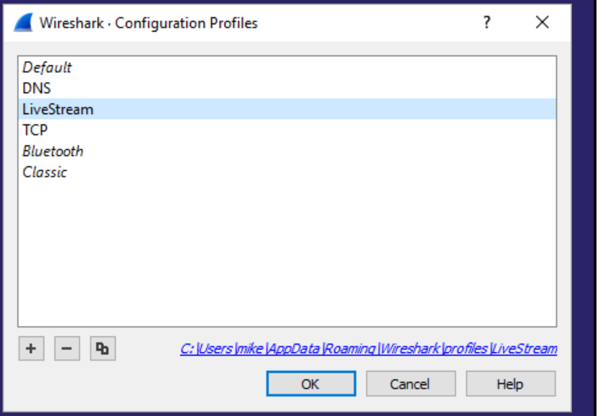
- Profiles are the key to customizing Wireshark
- Separate profiles can be created for protocol and applications
- Profiles can be copied to create subsets of existing profiles
- Profiles can be copied from one computer to another

interopitx.com #InteropITX

Interop ITX

Creating a Profile

- Edit – Configuration Profiles
- Click the + button
- Enter a name for the Profile
- Press Enter
- Wireshark will reload the trace using the default settings
- Any changes to colorization, filters, and columns will be applied to the new profile


Wireshark · Configuration Profiles

Default
DNS
LiveStream
TCP
Bluetooth
Classic

C:\Users\mike\AppData\Roaming\Wireshark\profiles\LiveStream

interopitx.com #InteropITX

Interop ITX

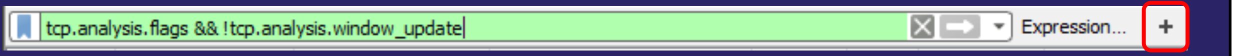
The Most Powerful TCP Filter

- I've got to give credit to Laura Chappell for this one
  - tcp.analysis.flags && !tcp.analysis.window_update
- This will display all TCP frames that have been flagged as having an issue

interopitx.com #InteropITX

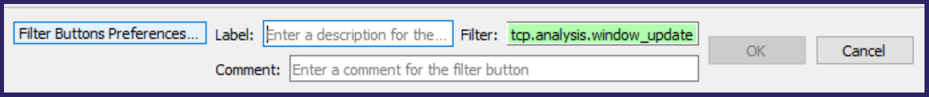
Interop ITX

The Most Powerful Button on the Screen



tcp.analysis.flags && !tcp.analysis.window_update

- Now that you have created the most powerful filter, it is time to make it really easy to use
- Clicking the Plus button on the filter line will create a new button that is assigned to that filter
- This filter button is specific to this profile



Filter Buttons Preferences... Label: Enter a description for the... Filter: tcp.analysis.window_update

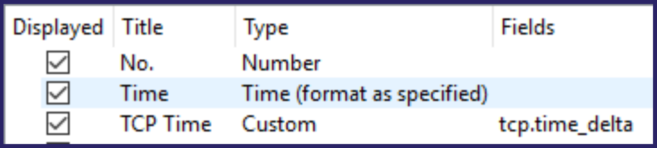
Comment: Enter a comment for the filter button

interopitx.com #InteropITX

InteropITX

Add TCP Time Column

- Edit → Preferences → User Interface → Columns
- Click + button
- Field Type: Custom
- Field Name: tcp.time_delta
- Click on Title and change to TCP Time

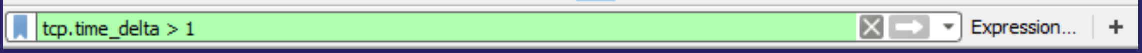


Displayed	Title	Type	Fields
☑	No.	Number	
☑	Time	Time (format as specified)	
☑	TCP Time	Custom	tcp.time_delta

interopitx.com #InteropITX

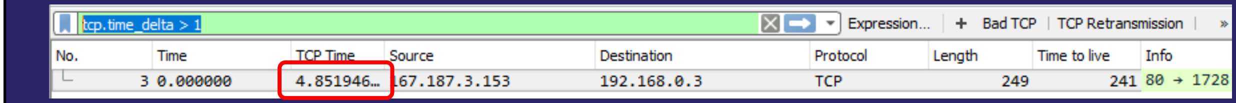
InteropITX

Finding TCP Delays



tcp.time_delta > 1

- We must configure the TCP protocol to calculate TCP conversation timestamps
- This will show all frames where it took TCP longer than 1 second to respond
- Great for finding slow response times



tcp.time_delta > 1

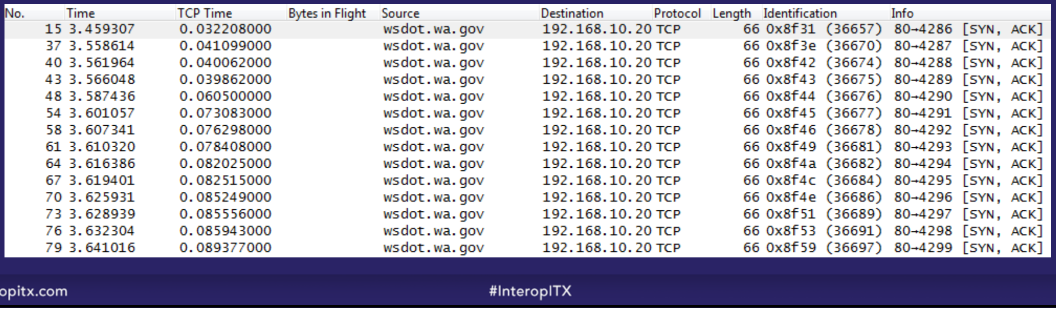
No.	Time	TCP Time	Source	Destination	Protocol	Length	Time to live	Info
3	0.000000	4.851946…	67.187.3.153	192.168.0.3	TCP	249	241	80 → 1728

interopitx.com #InteropITX

Interop ITX

Finding Established Connections

- When determining our dependencies, we need to know which TCP connections have been established and with which servers
- tcp.flags.syn==1 && tcp.flags.ack==1

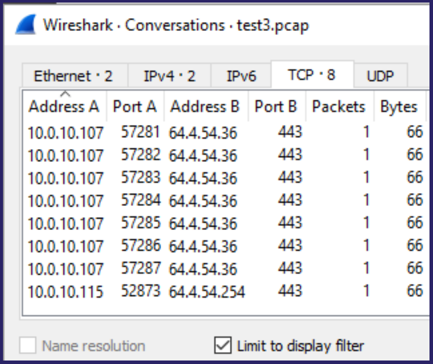


No.	Time	TCP Time	Bytes in Flight	Source	Destination	Protocol	Length	Identification	Info
15	3.459307	0.032208000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f31 (36657)	80→4286 [SYN, ACK]
37	3.558614	0.041099000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f3e (36670)	80→4287 [SYN, ACK]
40	3.561964	0.040062000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f42 (36674)	80→4288 [SYN, ACK]
43	3.566048	0.039862000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f43 (36675)	80→4289 [SYN, ACK]
48	3.587436	0.060500000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f44 (36676)	80→4290 [SYN, ACK]
54	3.601057	0.073083000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f45 (36677)	80→4291 [SYN, ACK]
58	3.607341	0.076298000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f46 (36678)	80→4292 [SYN, ACK]
61	3.610320	0.078408000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f49 (36681)	80→4293 [SYN, ACK]
64	3.616386	0.082025000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f4a (36682)	80→4294 [SYN, ACK]
67	3.619401	0.082515000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f4c (36684)	80→4295 [SYN, ACK]
70	3.625931	0.085249000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f4e (36686)	80→4296 [SYN, ACK]
73	3.628939	0.085556000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f51 (36689)	80→4297 [SYN, ACK]
76	3.632304	0.085943000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f53 (36691)	80→4298 [SYN, ACK]
79	3.641016	0.089377000		wsdot.wa.gov	192.168.10.20	TCP	66	0x8f59 (36697)	80→4299 [SYN, ACK]

Interop ITX

Finding Established Connections

- Once this filter has been applied, we can go to Statistics → Conversations
- Check the Limit to display filter box
- You will now see a listing of all established connections, the server and the TCP ports



Wireshark · Conversations · test3.pcap

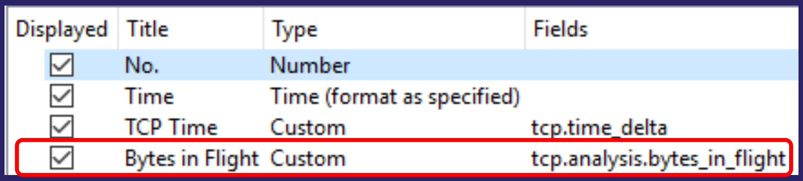
Address A	Port A	Address B	Port B	Packets	Bytes
10.0.10.107	57281	64.4.54.36	443	1	66
10.0.10.107	57282	64.4.54.36	443	1	66
10.0.10.107	57283	64.4.54.36	443	1	66
10.0.10.107	57284	64.4.54.36	443	1	66
10.0.10.107	57285	64.4.54.36	443	1	66
10.0.10.107	57286	64.4.54.36	443	1	66
10.0.10.107	57287	64.4.54.36	443	1	66
10.0.10.115	52873	64.4.54.254	443	1	66

□ Name resolution ☑ Limit to display filter

Interop ITX

TCP Bytes in Flight

- When troubleshooting TCP performance problems, it is helpful to know the number of unacknowledged bytes that are hanging out on the wire
- This can be accomplished by adding a column for TCP Bytes in Flight

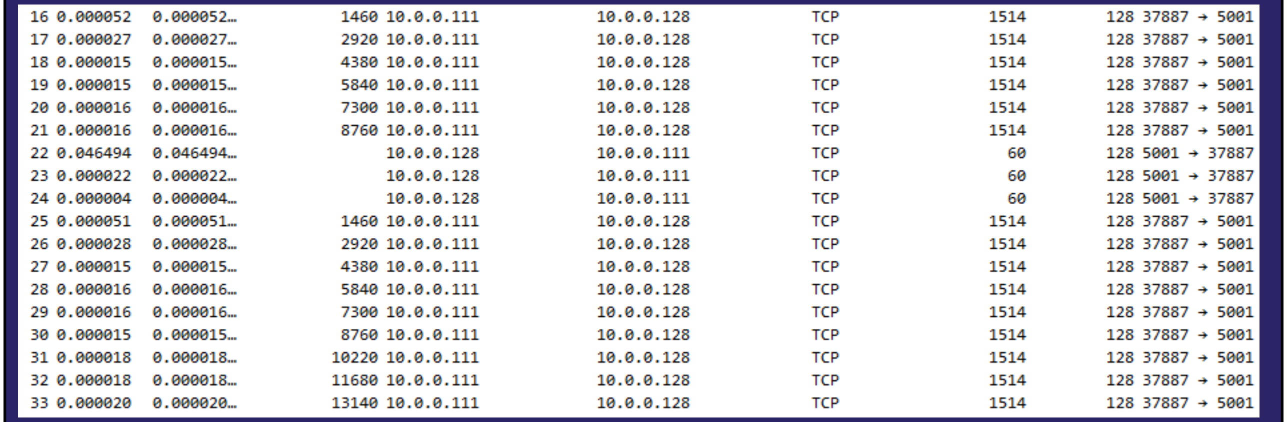


Displayed	Title	Type	Fields
☑	No.	Number	
☑	Time	Time (format as specified)	
☑	TCP Time	Custom	tcp.time_delta
☑	Bytes in Flight	Custom	tcp.analysis.bytes_in_flight

interopitx.com #InteropITX

Interop ITX

TCP Bytes in Flight

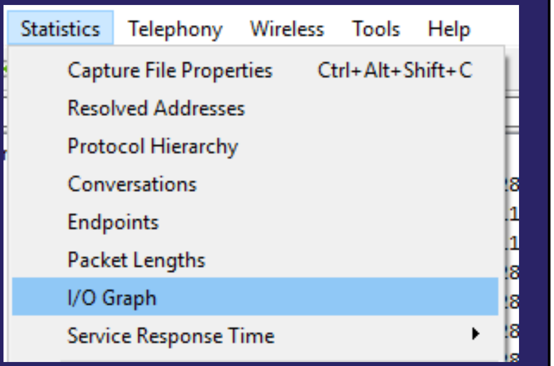


```
16 0.000052  0.000052…       1460 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
17 0.000027  0.000027…       2920 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
18 0.000015  0.000015…       4380 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
19 0.000015  0.000015…       5840 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
20 0.000016  0.000016…       7300 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
21 0.000016  0.000016…       8760 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
22 0.046494  0.046494…            10.0.0.128        10.0.0.111        TCP            60       128 5001 → 37887
23 0.000022  0.000022…            10.0.0.128        10.0.0.111        TCP            60       128 5001 → 37887
24 0.000004  0.000004…            10.0.0.128        10.0.0.111        TCP            60       128 5001 → 37887
25 0.000051  0.000051…       1460 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
26 0.000028  0.000028…       2920 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
27 0.000015  0.000015…       4380 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
28 0.000016  0.000016…       5840 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
29 0.000016  0.000016…       7300 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
30 0.000015  0.000015…       8760 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
31 0.000018  0.000018…      10220 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
32 0.000018  0.000018…      11680 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
33 0.000020  0.000020…      13140 10.0.0.111        10.0.0.128        TCP          1514       128 37887 → 5001
```

interopitx.com #InteropITX

I/O Graphs

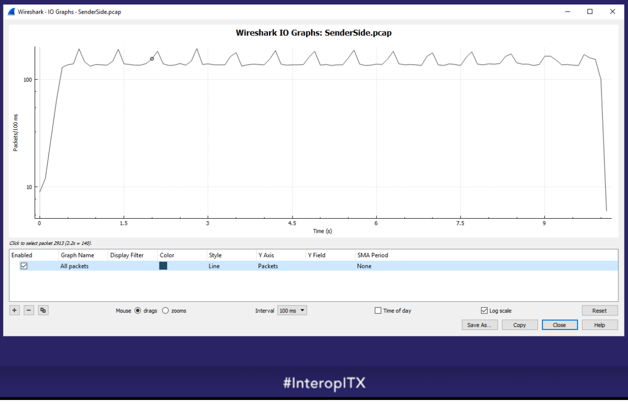
- Wireshark can also assist in measuring how much bandwidth was being used by a particular application or client
- These graphs can be accessed from the Statistics dropdown menu



interopitx.com #InteropITX

I/O Graphs

- This graph measures the bandwidth in the trace file over time. The X and Y axis can be modified to display packets per second or bits per second


Wireshark · IO Graphs · SenderSide.pcap
Wireshark IO Graphs: SenderSide.pcap
100
10
Packets/100 ms
0
1.5
3
4.5
6
7.5
9
Time (s)

Click to select packet 2913 (2.2s = 140).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
☑	All packets		■	Line	Packets		None

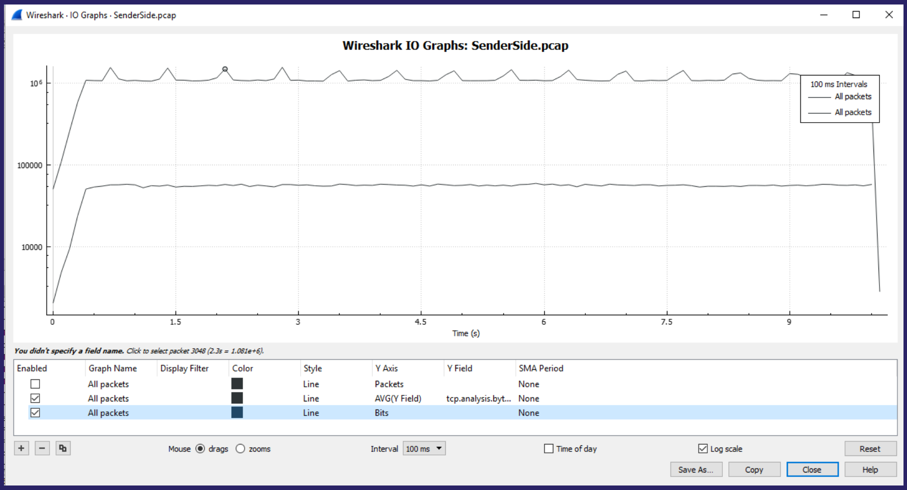
Mouse ● drags ○ zooms Interval 100 ms ☐ Time of day ☑ Log scale

+
−
Reset
Save As...
Copy
Close
Help

interopitx.com #InteropITX

Interop ITX

IO Graph and TCP Bytes in Flight


Wireshark · IO Graphs · SenderSide.pcap
Wireshark IO Graphs: SenderSide.pcap
10⁶
100000
10000
0
1.5
3
4.5
6
7.5
9
Time (s)
100 ms Intervals
All packets
All packets

You didn't specify a field name. Click to select packet 3048 (2.3s = 1.081e+6).

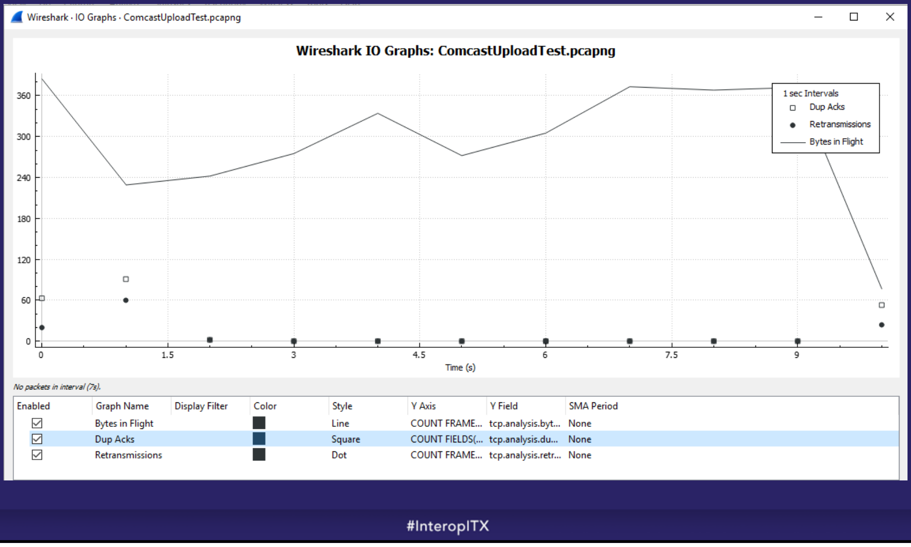
Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
☐	All packets		■	Line	Packets		None
☑	All packets		■	Line	AVG(Y Field)	tcp.analysis.byt...	None
☑	All packets		■	Line	Bits		None

Mouse ● drags ○ zooms Interval 100 ms ☐ Time of day ☑ Log scale

interopitx.com #InteropITX

Interop ITX

IO Graph and TCP Dup Ack and Retransmissions



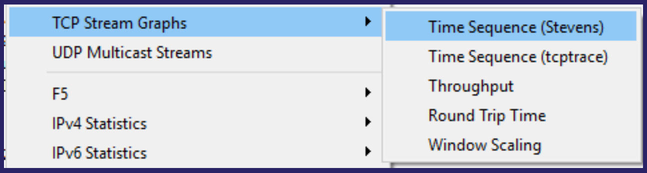
No packets in interval (7s).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
☑	Bytes in Flight		■	Line	COUNT FRAME...	tcp.analysis.byt...	None
☑	Dup Acks		■	Square	COUNT FIELDS(...	tcp.analysis.du...	None
☑	Retransmissions		■	Dot	COUNT FRAME...	tcp.analysis.retr...	None

interopitx.com #InteropITX

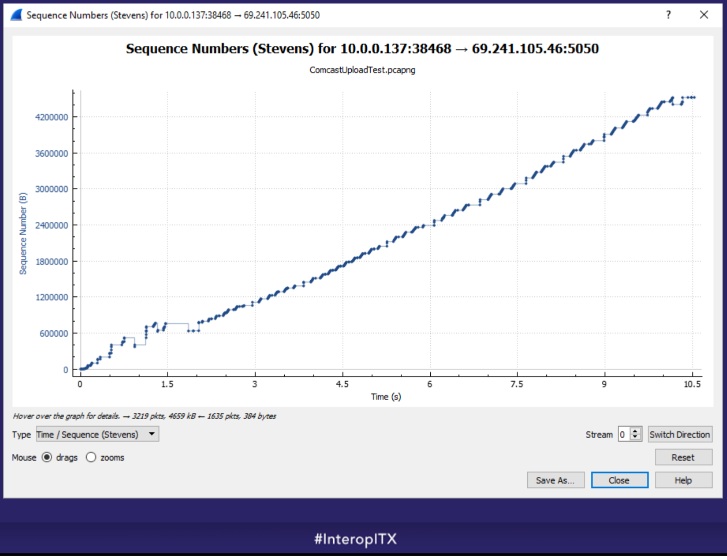
Stevens Graph

- Another good graph to use when troubleshooting in Wireshark is the Stevens Graph
- It simply graphs the sequence numbers in a TCP connection over time. If large gaps of time are experienced in the trace they will be easier to see



interopitx.com #InteropITX

Stevens Graph


Sequence Numbers (Stevens) for 10.0.0.137:38468 → 69.241.105.46:5050

Sequence Numbers (Stevens) for 10.0.0.137:38468 → 69.241.105.46:5050

ComcastUploadTest.pcapng

Hover over the graph for details. → 3219 pkts, 4659 kB ← 1635 pkts, 384 bytes

Type: Time / Sequence (Stevens)

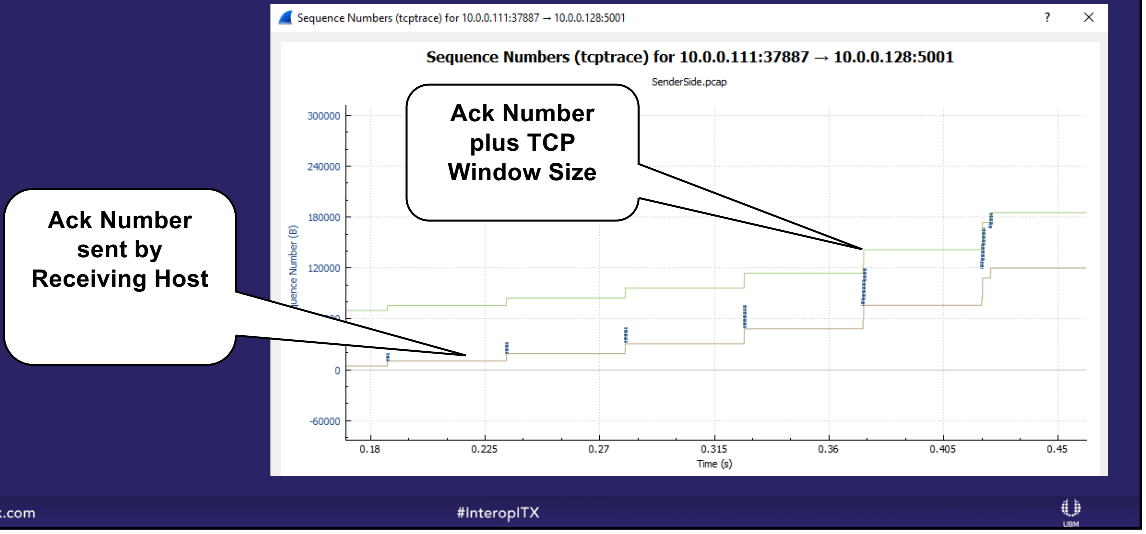
Stream: 0

Mouse: ● drags ○ zooms

interopitx.com #InteropITX

Interop ITX

TCPTrace Graph



Sequence Numbers (tcptrace) for 10.0.0.111:37887 → 10.0.0.128:5001

Sequence Numbers (tcptrace) for 10.0.0.111:37887 → 10.0.0.128:5001

SenderSide.pcap

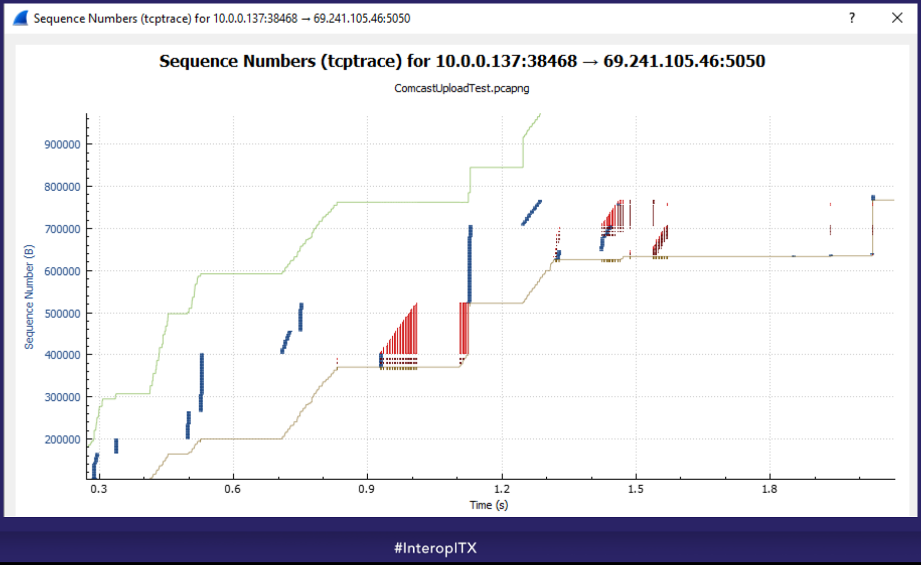
Ack Number plus TCP Window Size

Ack Number sent by Receiving Host

interopitx.com #InteropITX

Interop ITX

TCPTrace Graph – This can't be good



Sequence Numbers (tcptrace) for 10.0.0.137:38468 → 69.241.105.46:5050

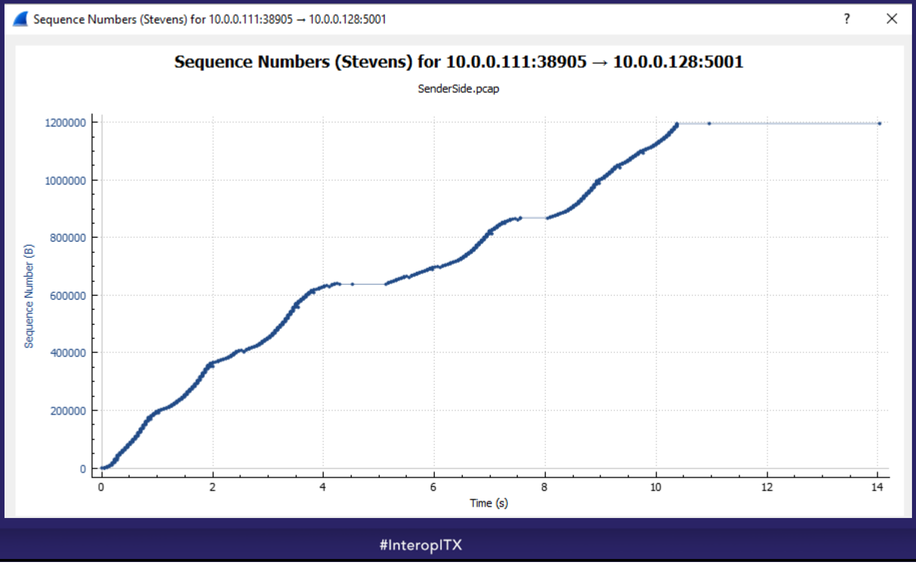
Sequence Numbers (tcptrace) for 10.0.0.137:38468 → 69.241.105.46:5050

ComcastUploadTest.pcapng

interopitx.com #InteropITX

Interop ITX

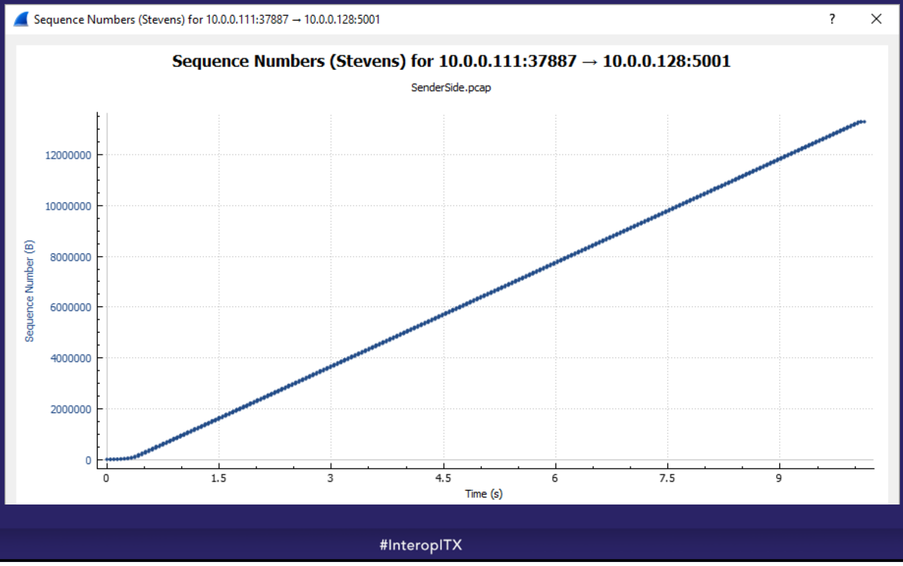
Stevens Graph – Bad Throughput



interopitx.com #InteropITX

Interop ITX

Stevens Graph – Good Throughput

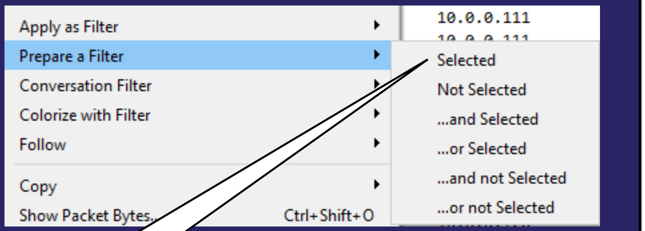


interopitx.com #InteropITX

Interop ITX

Right Clicking on a Field

- Allows us to quickly create a filter without typing
- Find field names quickly
- Can be used to combine multiple filters together to create complex filter expressions



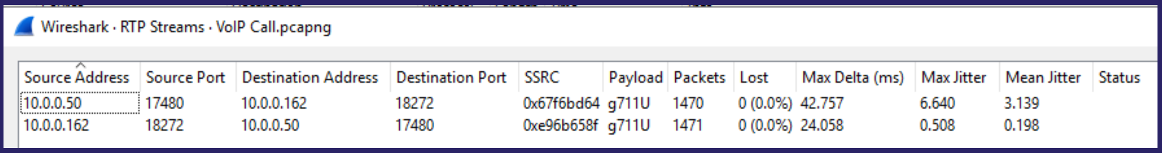
Click to create the filter

interopitx.com #InteropITX

Interop ITX

Analyzing VoIP Traffic

- Wireshark has a number of tools for analyzing and decoding VoIP traffic
- This can assist us in determining why VoIP calls are failing or are poor quality



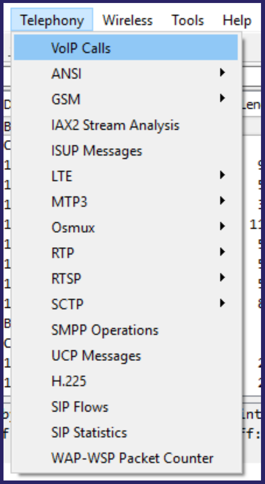
Wireshark · RTP Streams · VoIP Call.pcapng

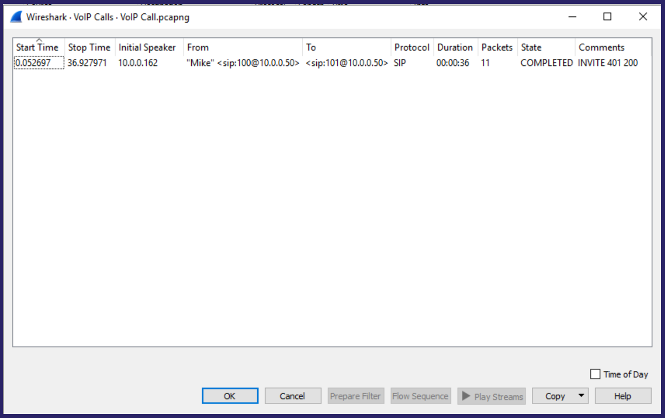
Source Address	Source Port	Destination Address	Destination Port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter	Mean Jitter	Status
10.0.0.50	17480	10.0.0.162	18272	0x67f6bd64	g711U	1470	0 (0.0%)	42.757	6.640	3.139	
10.0.0.162	18272	10.0.0.50	17480	0xe96b658f	g711U	1471	0 (0.0%)	24.058	0.508	0.198	

interopitx.com #InteropITX

Interop ITX

View the VoIP Calls




Wireshark · VoIP Calls · VoIP Call.pcapng

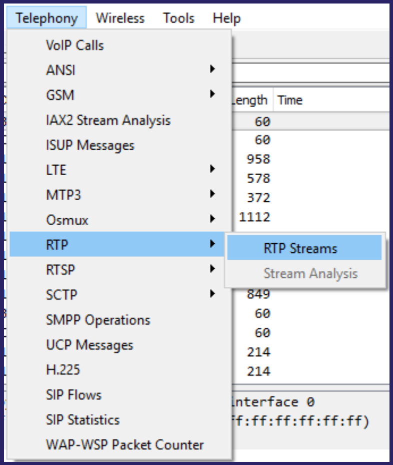
Start Time	Stop Time	Initial Speaker	From	To	Protocol	Duration	Packets	State	Comments
0.052697	36.927971	10.0.0.162	"Mike" <sip:100@10.0.0.50>	<sip:101@10.0.0.50>	SIP	00:00:36	11	COMPLETED	INVITE 401 200

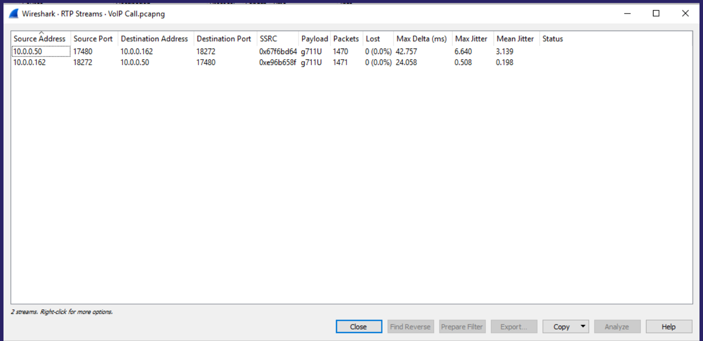
□ Time of Day

interopitx.com #InteropITX

Interop ITX

View RTP Streams




Wireshark · RTP Streams · VoIP Call.pcapng

Source Address	Source Port	Destination Address	Destination Port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter	Mean Jitter	Status
10.0.0.50	17480	10.0.0.162	18272	0x67f6bd64	g711U	1470	0 (0.0%)	42.757	6.640	3.139	
10.0.0.162	18272	10.0.0.50	17480	0xe96b658f	g711U	1471	0 (0.0%)	24.058	0.508	0.198	

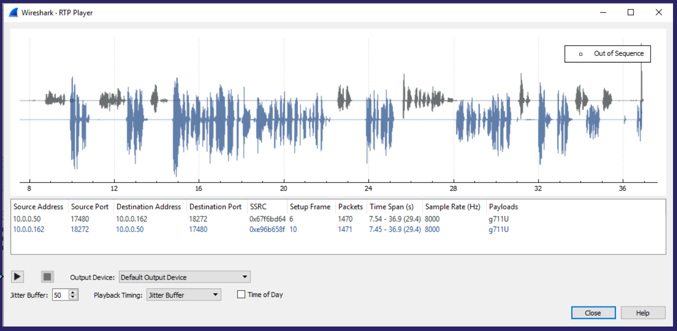
2 streams. Right-click for more options.

interopitx.com #InteropITX

Interop ITX

Play Back Stream

- From VoIP Calls, click Play Streams
- Click the Play Button



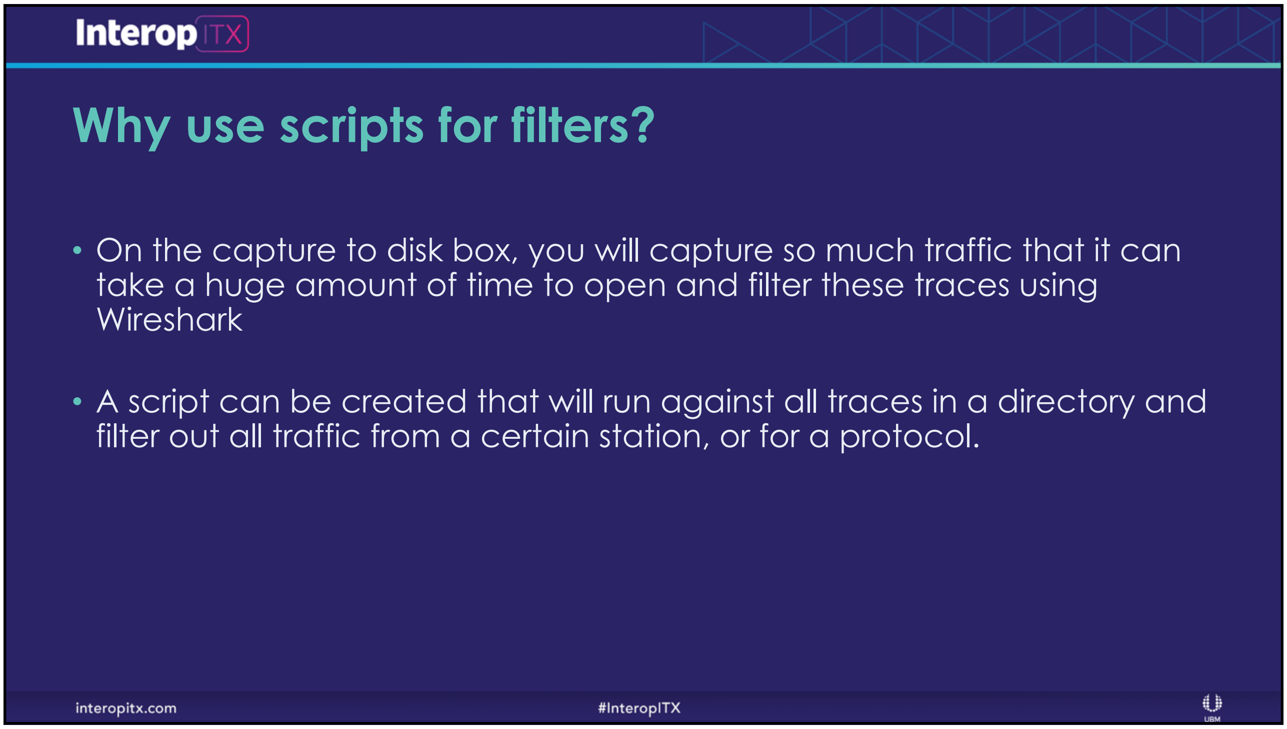
Source Address	Source Port	Destination Address	Destination Port	SSRC	Setup Frame	Packets	Time Span (s)	Sample Rate (Hz)	Payloads
10.0.0.50	17480	10.0.0.162	18272	0x67f6bd64	6	1470	7.54 - 36.9 (29.4)	8000	g711U
10.0.0.162	18272	10.0.0.50	17480	0xe96b658f	10	1471	7.45 - 36.9 (29.4)	8000	g711U

interopitx.com #InteropITX

Interop ITX APRIL 30 – MAY 4, 2018
THE MIRAGE | LAS VEGAS

Display Filter Scripts

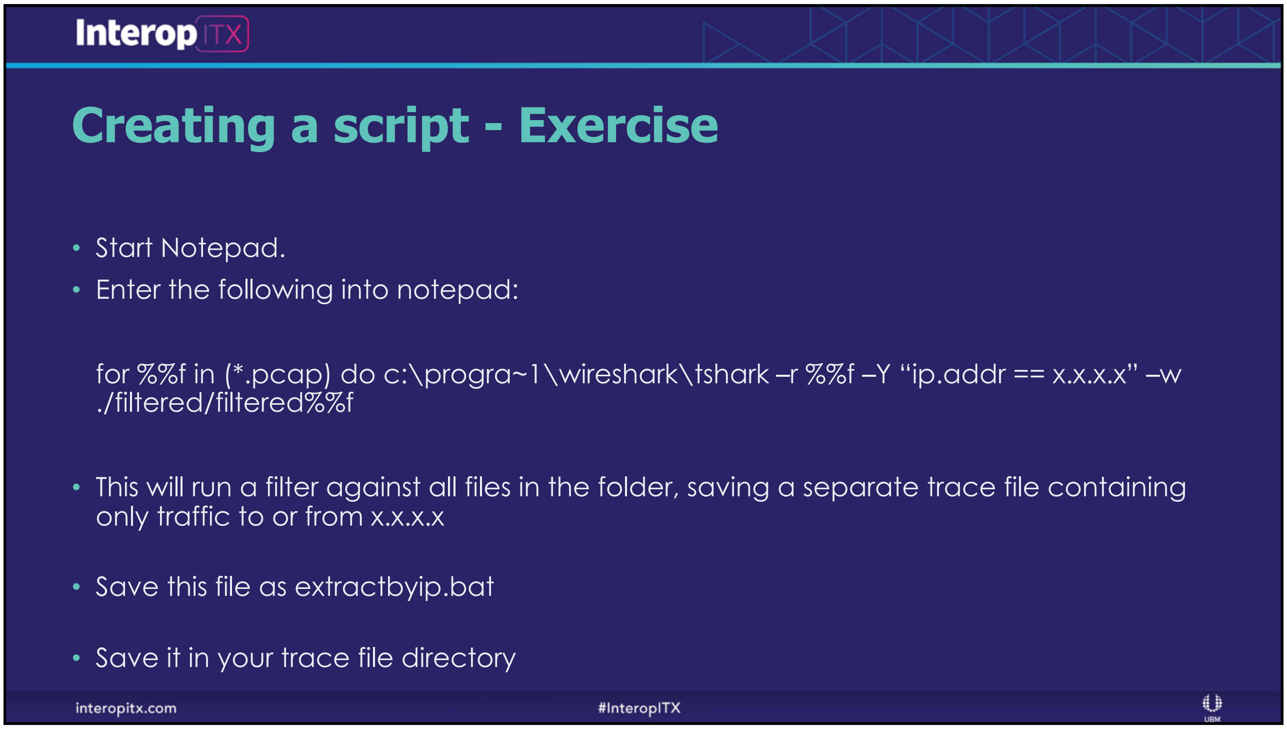
interopitx.com #InteropITX



Why use scripts for filters?

- On the capture to disk box, you will capture so much traffic that it can take a huge amount of time to open and filter these traces using Wireshark
- A script can be created that will run against all traces in a directory and filter out all traffic from a certain station, or for a protocol.

interopitx.com #InteropITX



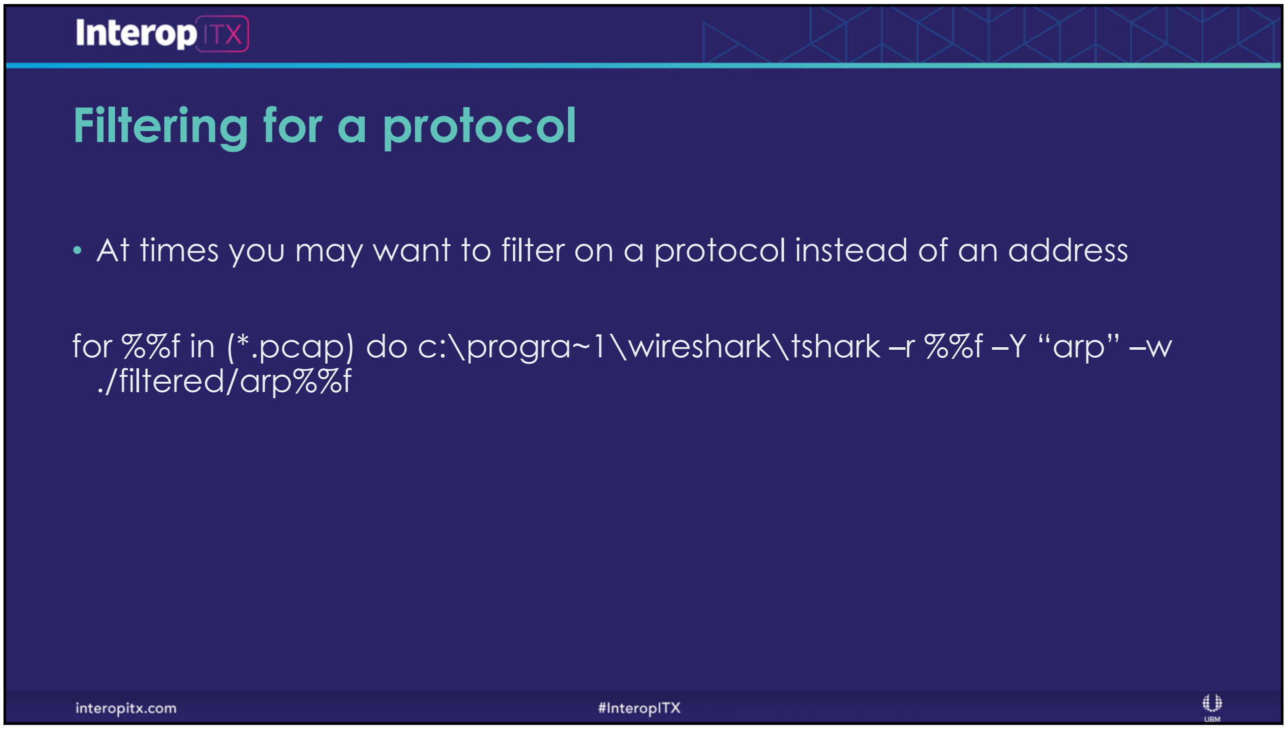
Creating a script - Exercise

- Start Notepad.
- Enter the following into notepad:

```
for %%f in (*.pcap) do c:\progra~1\wireshark\tshark –r %%f –Y “ip.addr == x.x.x.x” –w
./filtered/filtered%%f
```

- This will run a filter against all files in the folder, saving a separate trace file containing only traffic to or from x.x.x.x
- Save this file as extractbyip.bat
- Save it in your trace file directory

interopitx.com #InteropITX



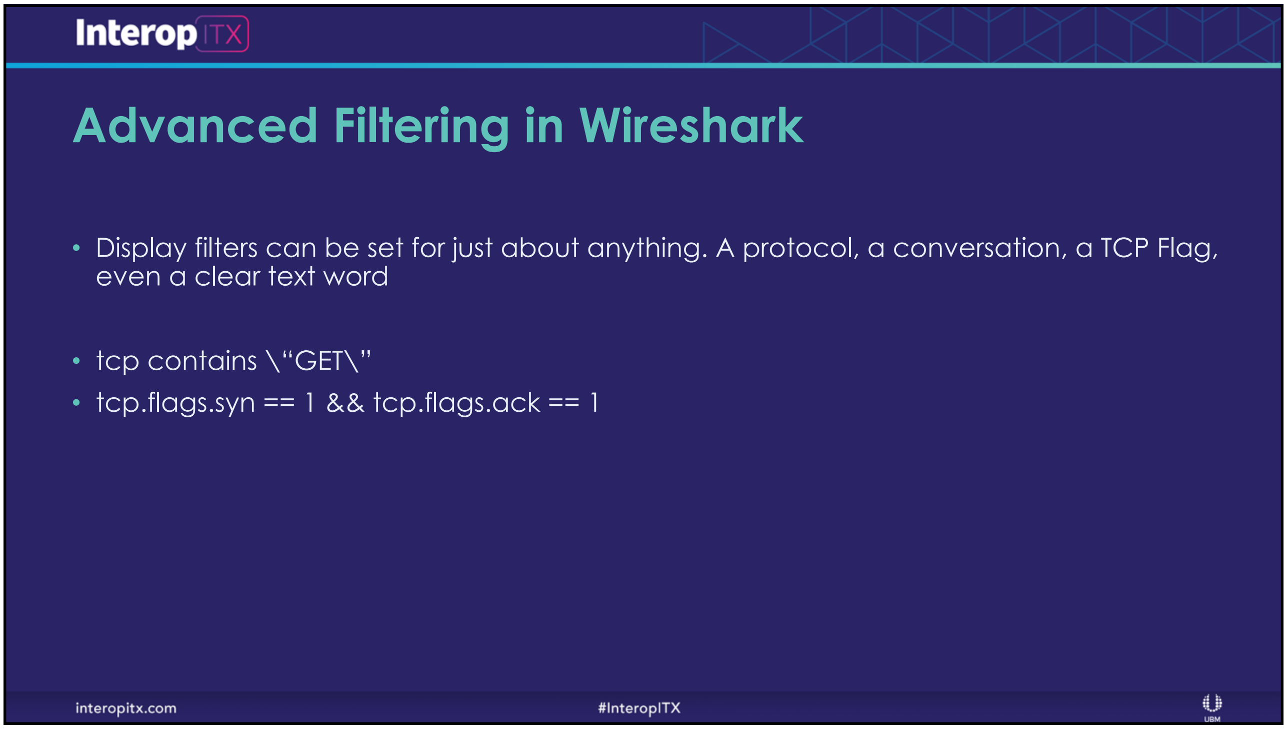
Interop ITX

Filtering for a protocol

- At times you may want to filter on a protocol instead of an address

```
for %%f in (*.pcap) do c:\progra~1\wireshark\tshark –r %%f –Y "arp" –w
  ./filtered/arp%%f
```

interopitx.com #InteropITX

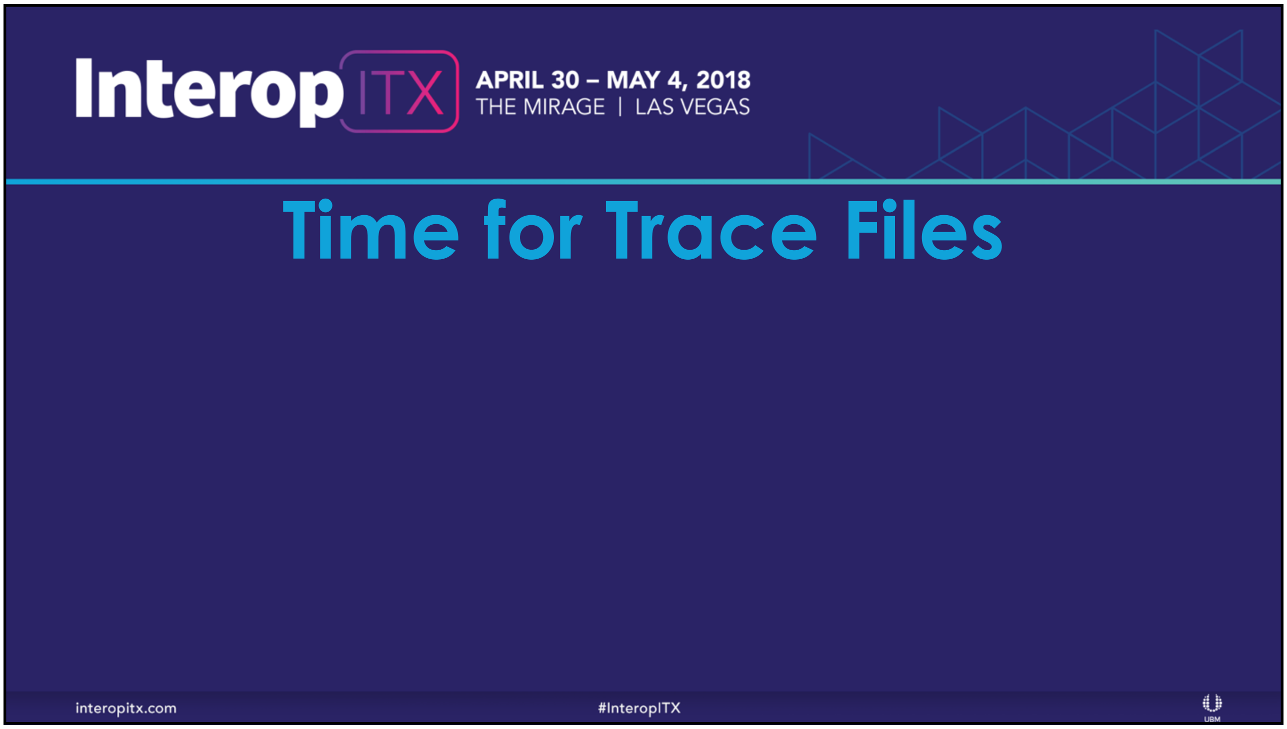


Interop ITX

Advanced Filtering in Wireshark

- Display filters can be set for just about anything. A protocol, a conversation, a TCP Flag, even a clear text word
- tcp contains \"GET\"
- tcp.flags.syn == 1 && tcp.flags.ack == 1

interopitx.com #InteropITX

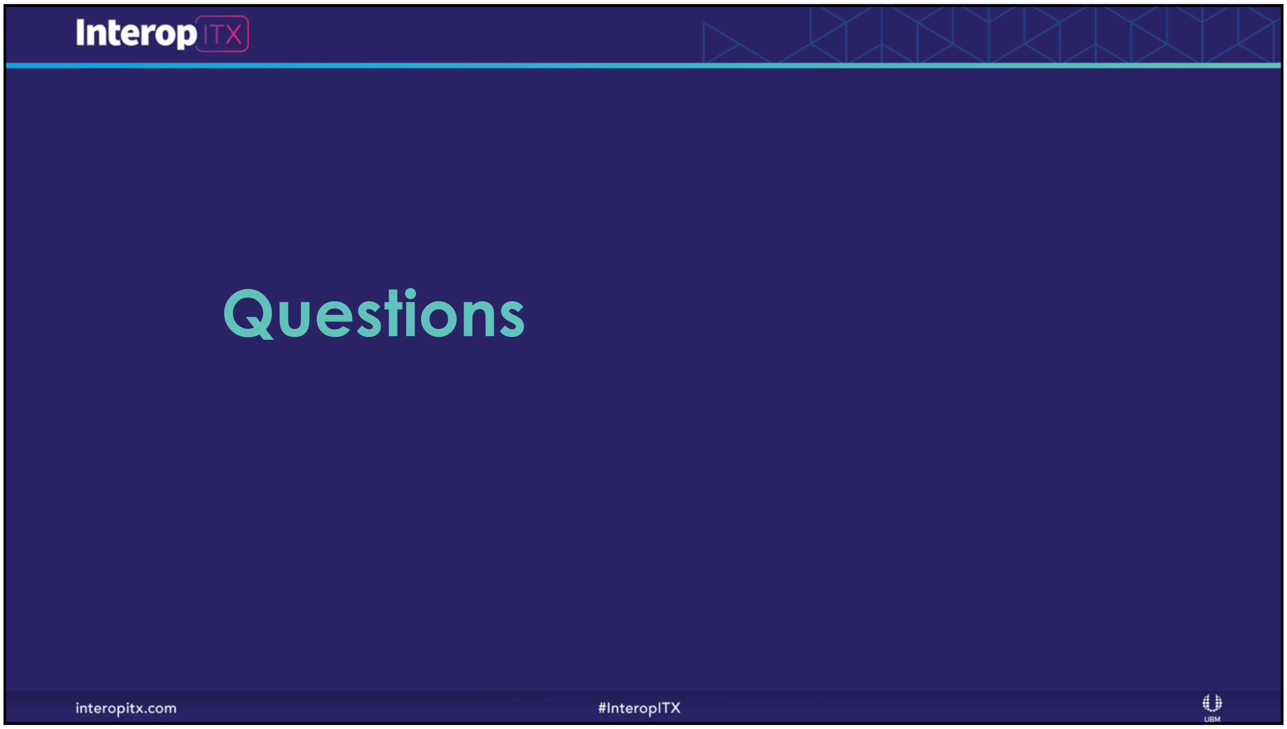


Interop ITX

APRIL 30 – MAY 4, 2018
THE MIRAGE | LAS VEGAS

Time for Trace Files

interopitx.com #InteropITX



Interop ITX

Questions

interopitx.com #InteropITX