Created using OpenOffice™ 4

#SHAREorg

SHARE

Taming the Shark
Tips and Tricks on Using Wireshark
Hands On Labs

Matthias Burkhard
IBM Technical Support Services
IBM Germany

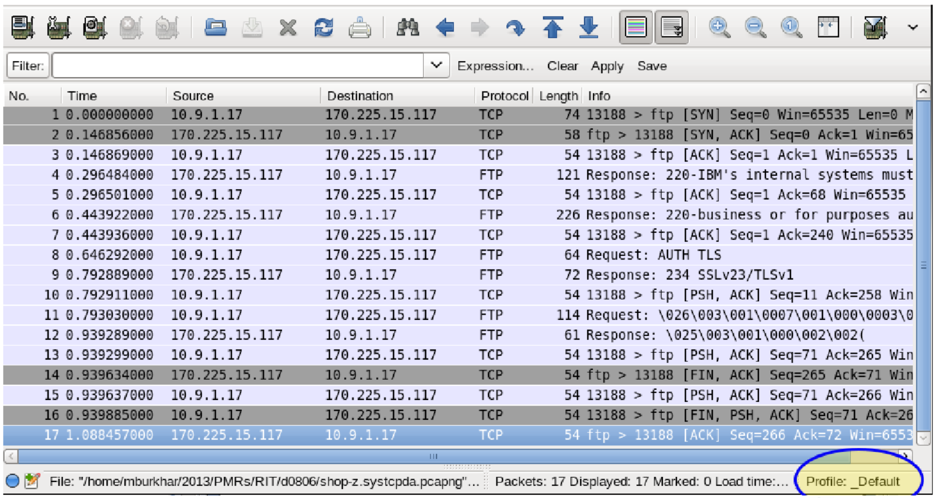
Thursday Aug. 15 2013 4:30-5:30 PM
Session 13282 Hynes Room 202

Twitter @mreede
Find us on Facebook at ip.wizards@groups.facebook.com
LinkedIn: de.linkedin.com/in/mreede/
IBM SmartCloud: Matthias Burkhard

SocialBusiness
IBMSmartCloud

SHARE in Boston

Why use different profiles?
A trace is a trace is a trace – isn't it?



```
No.  Time        Source          Destination     Protocol Length Info
   1 0.000000000 10.9.1.17       170.225.15.117  TCP        74 13188 > ftp [SYN] Seq=0 Win=65535 Len=0 M
   2 0.146856000 170.225.15.117  10.9.1.17       TCP        58 ftp > 13188 [SYN, ACK] Seq=0 Ack=1 Win=65
   3 0.146869000 10.9.1.17       170.225.15.117  TCP        54 13188 > ftp [ACK] Seq=1 Ack=1 Win=65535 L
   4 0.296484000 170.225.15.117  10.9.1.17       FTP       121 Response: 220-IBM's internal systems must
   5 0.296501000 10.9.1.17       170.225.15.117  TCP        54 13188 > ftp [ACK] Seq=1 Ack=68 Win=65535
   6 0.443922000 170.225.15.117  10.9.1.17       FTP       226 Response: 220-business or for purposes au
   7 0.443936000 10.9.1.17       170.225.15.117  TCP        54 13188 > ftp [ACK] Seq=1 Ack=240 Win=65535
   8 0.646292000 10.9.1.17       170.225.15.117  FTP        64 Request: AUTH TLS
   9 0.792889000 170.225.15.117  10.9.1.17       FTP        72 Response: 234 SSLv23/TLSv1
  10 0.792911000 10.9.1.17       170.225.15.117  TCP        54 13188 > ftp [PSH, ACK] Seq=11 Ack=258 Win
  11 0.793030000 10.9.1.17       170.225.15.117  FTP       114 Request: \026\003\001\0007\001\000\0003\0
  12 0.939289000 170.225.15.117  10.9.1.17       FTP        61 Response: \025\003\001\000\002\002(
  13 0.939299000 10.9.1.17       170.225.15.117  TCP        54 13188 > ftp [PSH, ACK] Seq=71 Ack=265 Win
  14 0.939634000 170.225.15.117  10.9.1.17       TCP        54 ftp > 13188 [FIN, ACK] Seq=265 Ack=71 Win
  15 0.939637000 10.9.1.17       170.225.15.117  TCP        54 13188 > ftp [PSH, ACK] Seq=71 Ack=266 Win
  16 0.939885000 10.9.1.17       170.225.15.117  TCP        54 13188 > ftp [FIN, PSH, ACK] Seq=71 Ack=26
  17 1.088457000 170.225.15.117  10.9.1.17       TCP        54 ftp > 13188 [ACK] Seq=266 Ack=72 Win=6553
```

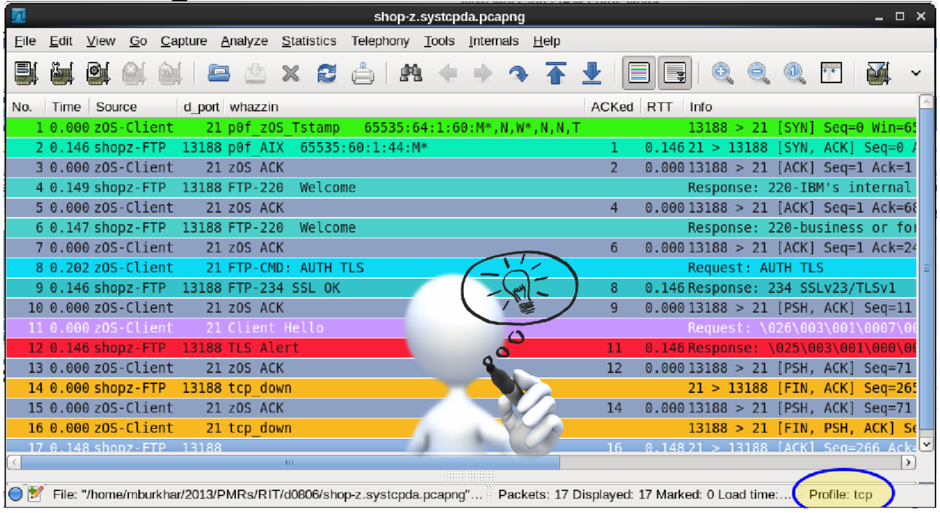
File: "/home/mburkhar/2013/PMRs/RIT/d0806/shop-z.systcpda.pcapng"... Packets: 17 Displayed: 17 Marked: 0 Load time:... Profile: _Default

2 Complete your sessions evaluation online at SHARE.org/BostonEval

SHARE in Boston

This is the standard layout that ships with wireshark. You find 6 columns in the Packet List pane and the colors are not very exciting.

Why use different profiles?
Coloring Rules! Show what's in there!


shop-z.systcpda.pcapng
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

```
No. Time  Source      d_port whazzin                                       ACKed RTT   Info
  1 0.000 zOS-Client     21 p0f_zOS_Tstamp   65535:64:1:60:M*,N,W*,N,N,T               13188 > 21 [SYN] Seq=0 Win=65
  2 0.146 shopz-FTP   13188 p0f_AIX  65535:60:1:44:M*                       1    0.146 21 > 13188 [SYN, ACK] Seq=0 A
  3 0.000 zOS-Client     21 zOS ACK                                         2    0.000 13188 > 21 [ACK] Seq=1 Ack=1
  4 0.149 shopz-FTP   13188 FTP-220  Welcome                                           Response: 220-IBM's internal
  5 0.000 zOS-Client     21 zOS ACK                                         4    0.000 13188 > 21 [ACK] Seq=1 Ack=68
  6 0.147 shopz-FTP   13188 FTP-220  Welcome                                           Response: 220-business or for
  7 0.000 zOS-Client     21 zOS ACK                                         6    0.000 13188 > 21 [ACK] Seq=1 Ack=24
  8 0.202 zOS-Client     21 FTP-CMD: AUTH TLS                                          Request: AUTH TLS
  9 0.146 shopz-FTP   13188 FTP-234 SSL OK                                  8    0.146 Response: 234 SSLv23/TLSv1
 10 0.000 zOS-Client     21 zOS ACK                                         9    0.000 13188 > 21 [PSH, ACK] Seq=11
 11 0.000 zOS-Client     21 Client Hello                                               Request: \026\003\001\0007\00
 12 0.146 shopz-FTP   13188 TLS Alert                                      11    0.146 Response: \025\003\001\000\00
 13 0.000 zOS-Client     21 zOS ACK                                        12    0.000 13188 > 21 [PSH, ACK] Seq=71
 14 0.000 shopz-FTP   13188 tcp_down                                                   21 > 13188 [FIN, ACK] Seq=265
 15 0.000 zOS-Client     21 zOS ACK                                        14    0.000 13188 > 21 [PSH, ACK] Seq=71
 16 0.000 zOS-Client     21 tcp_down                                                   13188 > 21 [FIN, PSH, ACK] Se
 17 0.148 shopz-FTP   13188                                                16    0.148 21 > 13188 [ACK] Seq=266 Ack=
```

File: "/home/mburkhar/2013/PMRs/RIT/d0806/shop-z.systcpda.pcapng"… Packets: 17 Displayed: 17 Marked: 0 Load time:… Profile: tcp

3 Complete your sessions evaluation online at SHARE.org/BostonEval

This is the same trace looked at using a more sophisticated profile.

The packet list contains information from deep in the packet. This save us zooming into each packet individually.

You can see that the source now contains a host name. This is achieved by coding a hosts file in the profile TCP's folder that wireshark can use to resolve ip addresses.

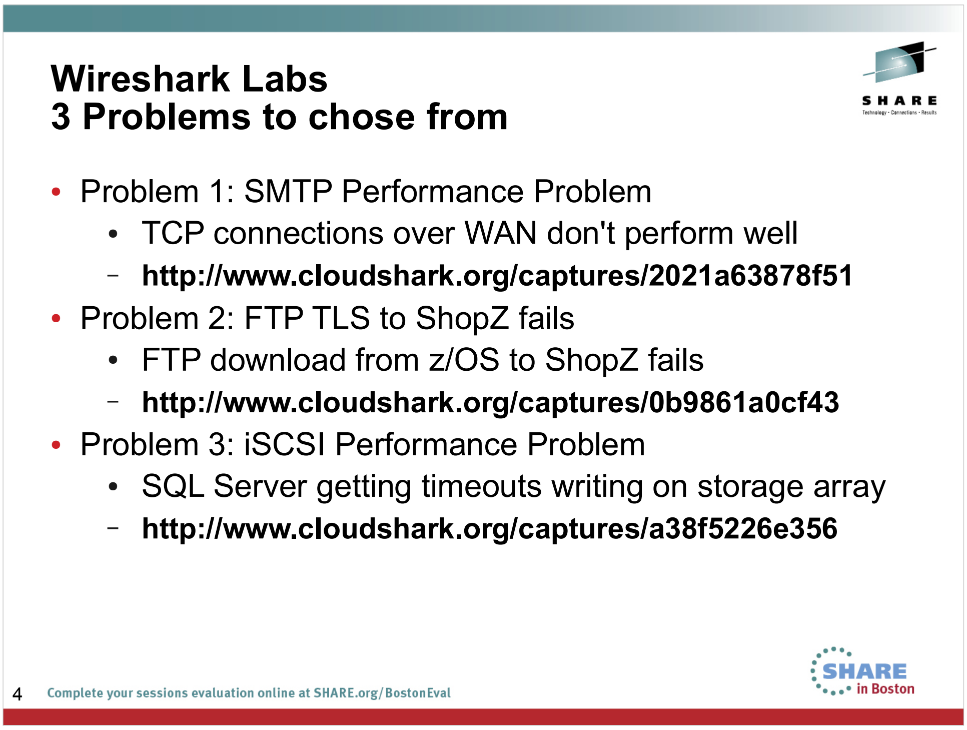
The most obvious difference are the different colors that are assigned via the coloring rules for certain packet content.

The coloring rule name, if chosen meaningful and added to the packet list, can help to guide the user through the trace.

The scenario here:

A z/OS client was trying to connect to shop-z to download some maintenance.
While this was running before, all of a sudden the FTP failed.

In this hands-on lab we will inspect the SYSTCPDA packet trace using wireshark and learn how to create a profile that helps us identify this problem faster.



Wireshark Labs
3 Problems to chose from

- Problem 1: SMTP Performance Problem
  - TCP connections over WAN don't perform well
  - **http://www.cloudshark.org/captures/2021a63878f51**
- Problem 2: FTP TLS to ShopZ fails
  - FTP download from z/OS to ShopZ fails
  - **http://www.cloudshark.org/captures/0b9861a0cf43**
- Problem 3: iSCSI Performance Problem
  - SQL Server getting timeouts writing on storage array
  - **http://www.cloudshark.org/captures/a38f5226e356**

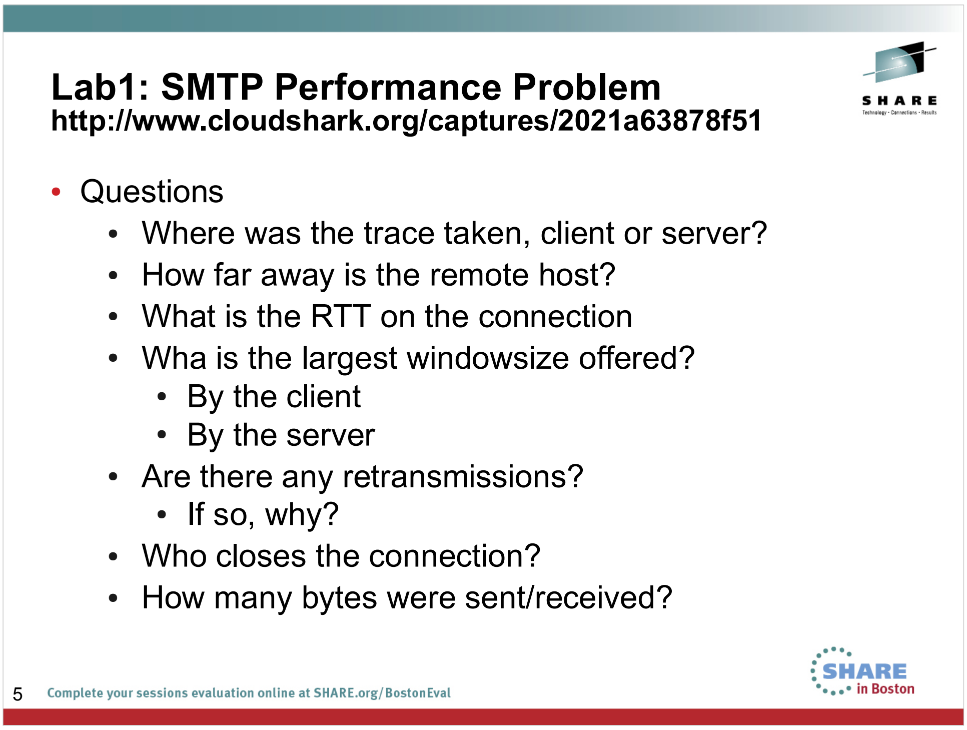
You have 3 Labs to chose from.
The intention is not to answer the questions as fast as possible but to figure out the best way to answer the questions by using various features and fucntions in wireshark to find your personal best fit in attacking network problems. Only when you know several options will you be able to chose theoptimal method.

All trace files are uploaded to www.cloudshark.org where the traces can be looked at using a browser based wireshark. So this would also work for iPad, Tablet, smartphones etc...

Problem number one is a SYSTCPDA packet Trace taken with TDSLINK, a trace tool for packet tracing on z/OS.

Problem number two is a SYSTCPDA Packet Trace converted using IPCS It shows a TLS negotiation error to ShopZ.

Problem number 3 is a tcpdump trace documenting a performance problem from an SQL Server to an iSCSI storage array.



Lab1: SMTP Performance Problem

http://www.cloudshark.org/captures/2021a63878f51

- Questions
  - Where was the trace taken, client or server?
  - How far away is the remote host?
  - What is the RTT on the connection
  - Wha is the largest windowsize offered?
    - By the client
    - By the server
  - Are there any retransmissions?
    - If so, why?
  - Who closes the connection?
  - How many bytes were sent/received?

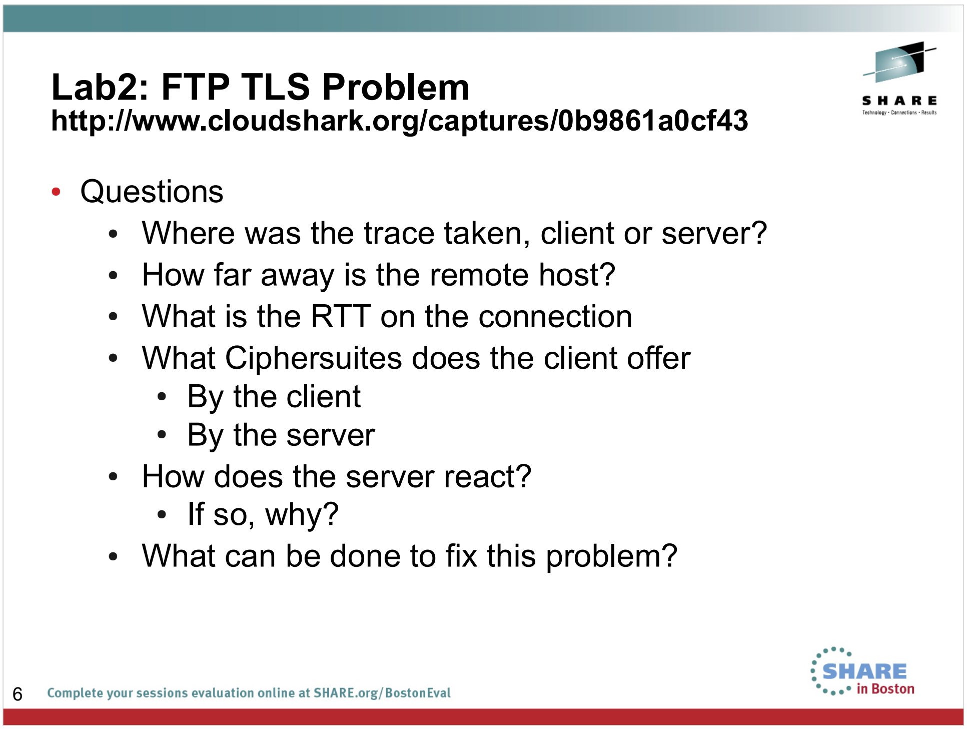
Hints:

Look at the ip.ttl field to identify who is the local and remote IP host.

Statistics → Flowgraph gives you a nice overview of the traffic

The windowsizes advertized by the TCP stack are derived by the TCP receivebuffer size used on the socket. In z/OS, the advertized windowsize is 2*TCPRECVBUF.

Retransmissions occur when the third duplicated ACK is received or when the retransmission timer pops. Try the tcp.analysis.flags filter to find suspicious packets.



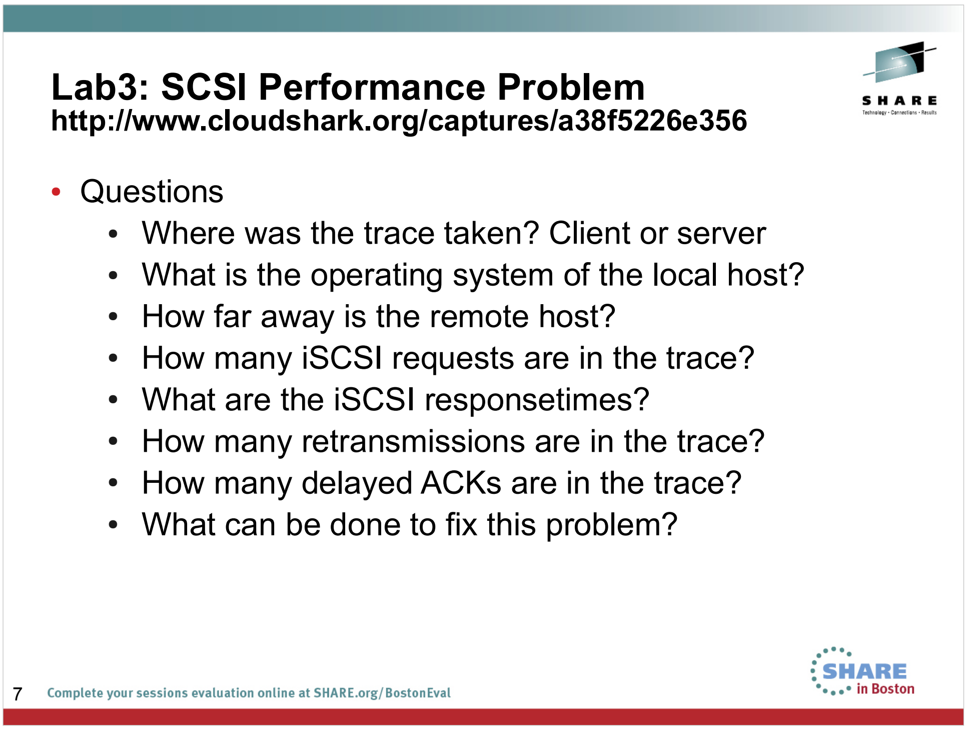
Lab2: FTP TLS Problem

http://www.cloudshark.org/captures/0b9861a0cf43

- Questions
  - Where was the trace taken, client or server?
  - How far away is the remote host?
  - What is the RTT on the connection
  - What Ciphersuites does the client offer
    - By the client
    - By the server
  - How does the server react?
    - If so, why?
  - What can be done to fix this problem?

FTP can be using TLS when a AUTH TLS command is sent to the server.
For wireshark to be able to interpret those packets, you need to use the 'Decode as' function.

Many encryption algorithms are not considered secure anymore as they have been compromised by now. Some servers insist on having current CipherSuites in use.



Lab3: SCSI Performance Problem

http://www.cloudshark.org/captures/a38f5226e356

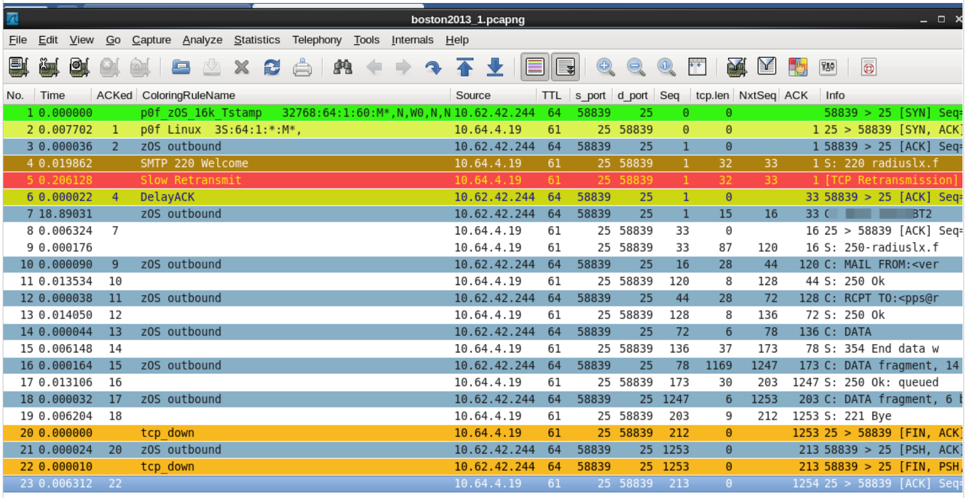
- Questions
  - Where was the trace taken? Client or server
  - What is the operating system of the local host?
  - How far away is the remote host?
  - How many iSCSI requests are in the trace?
  - What are the iSCSI responsetimes?
  - How many retransmissions are in the trace?
  - How many delayed ACKs are in the trace?
  - What can be done to fix this problem?

In TCP segments are acknowledged on a regular basis. Normal behaviour is that the receiver acknowledges 'every other' packet, which means we wait until we receive 2 segments before we send an 'empty' ACK. If the second segment does not arrive as the sender has no more data to send, the delay_ack timer pops and a so coalled 'delayed acknowledgment' is sent out.

The Nagle algorithm tries to reduce the number of packets by holding on to data in case the application has more to send.

In some scenarions, Nagle and delayed acknowledgements don't go well together. See the youtube video from Hansang Bae http://www.youtube.com/watch?v=2CMueBcQNtk

Session 13282 “Taming the Shark”
SMTP Performance

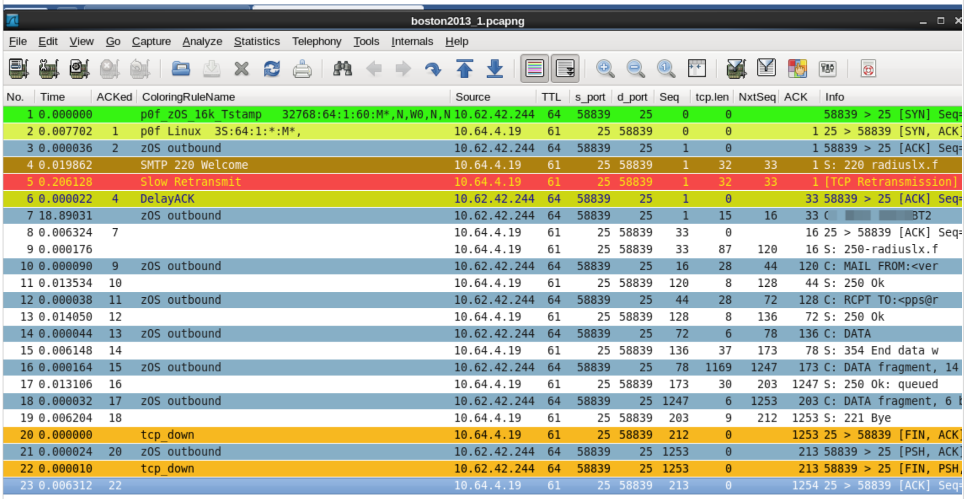


No.	Time	ACKed	ColoringRuleName	Source	TTL	s_port	d_port	Seq	tcp.len	NxtSeq	ACK	Info
1	0.000000		p0f_zOS_16k_Tstamp 32768:64:1:60:M*,N,W0,N,N	10.62.42.244	64	58839	25	0	0			58839 > 25 [SYN] Seq=
2	0.007702	1	p0f Linux 3S:64:1:*:M*,	10.64.4.19	61	25	58839	0	0		1	25 > 58839 [SYN, ACK]
3	0.000036	2	zOS outbound	10.62.42.244	64	58839	25	1	0		1	58839 > 25 [ACK] Seq=
4	0.019862		SMTP 220 Welcome	10.64.4.19	61	25	58839	1	32	33	1	S: 220 radiuslx.f
5	0.206128		Slow Retransmit	10.64.4.19	61	25	58839	1	32	33	1	[TCP Retransmission]
6	0.000022	4	DelayACK	10.62.42.244	64	58839	25	1	0		33	58839 > 25 [ACK] Seq=
7	18.89031		zOS outbound	10.62.42.244	64	58839	25	1	15	16	33	C [redacted] BT2
8	0.006324	7		10.64.4.19	61	25	58839	33	0		16	25 > 58839 [ACK] Seq=
9	0.000176			10.64.4.19	61	25	58839	33	87	120	16	S: 250-radiuslx.f
10	0.000090	9	zOS outbound	10.62.42.244	64	58839	25	16	28	44	120	C: MAIL FROM:<ver
11	0.013534	10		10.64.4.19	61	25	58839	120	8	128	44	S: 250 Ok
12	0.000038	11	zOS outbound	10.62.42.244	64	58839	25	44	28	72	128	C: RCPT TO:<pps@r
13	0.014050	12		10.64.4.19	61	25	58839	128	8	136	72	S: 250 Ok
14	0.000044	13	zOS outbound	10.62.42.244	64	58839	25	72	6	78	136	C: DATA
15	0.006148	14		10.64.4.19	61	25	58839	136	37	173	78	S: 354 End data w
16	0.000164	15	zOS outbound	10.62.42.244	64	58839	25	78	1169	1247	173	C: DATA fragment, 14
17	0.013106	16		10.64.4.19	61	25	58839	173	30	203	1247	S: 250 Ok: queued
18	0.000032	17	zOS outbound	10.62.42.244	64	58839	25	1247	6	1253	203	C: DATA fragment, 6 b
19	0.006204	18		10.64.4.19	61	25	58839	203	9	212	1253	S: 221 Bye
20	0.000000		tcp_down	10.64.4.19	61	25	58839	212	0		1253	25 > 58839 [FIN, ACK]
21	0.000024	20	zOS outbound	10.62.42.244	64	58839	25	1253	0		213	58839 > 25 [PSH, ACK]
22	0.000010		tcp_down	10.62.42.244	64	58839	25	1253	0		213	58839 > 25 [FIN, PSH,
23	0.006312	22		10.64.4.19	61	25	58839	213	0		1254	25 > 58839 [ACK] Seq=

8 Complete your sessions evaluation online at SHARE.org/BostonEval

Wireshark coloring rules can be used to highlight certain events.

Session 13282 “Taming the Shark”
SMTP Performance

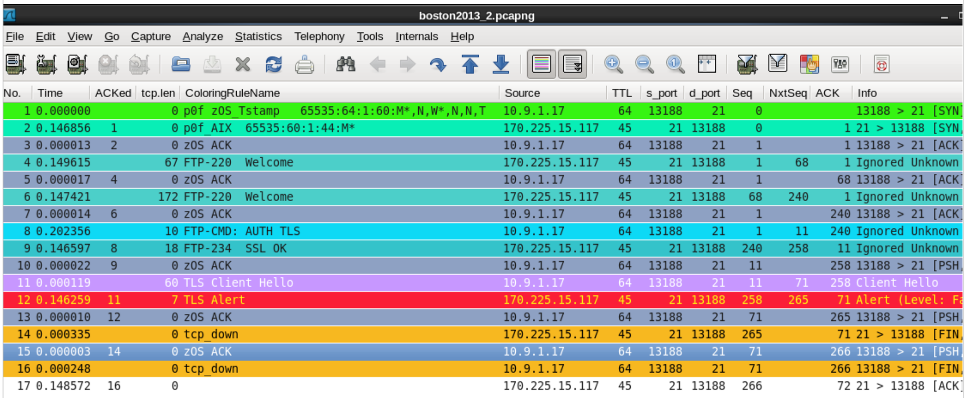


boston2013_1.pcapng

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

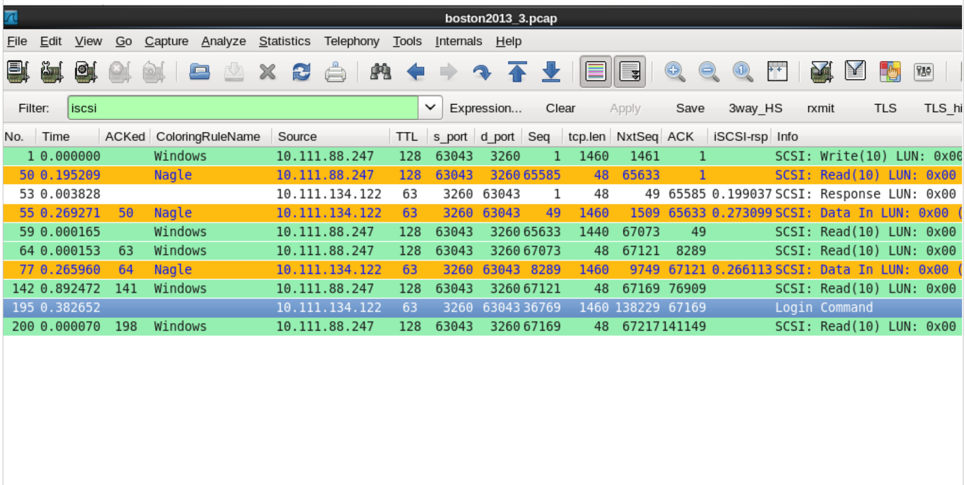
No.	Time	ACKed	ColoringRuleName	Source	TTL	s_port	d_port	Seq	tcp.len	NxtSeq	ACK	Info
1	0.000000		p0f_zOS_16k_Tstamp 32768:64:1:60:M*,N,W0,N,N	10.62.42.244	64	58839	25	0	0			58839 > 25 [SYN] Seq=
2	0.007702	1	p0f Linux 3S:64:1:*:M*,	10.64.4.19	61	25	58839	0	0		1	25 > 58839 [SYN, ACK]
3	0.000036	2	zOS outbound	10.62.42.244	64	58839	25	1	0		1	58839 > 25 [ACK] Seq=
4	0.019862		SMTP 220 Welcome	10.64.4.19	61	25	58839	1	32	33	1	S: 220 radiuslx.f
5	0.206128		Slow Retransmit	10.64.4.19	61	25	58839	1	32	33	1	[TCP Retransmission]
6	0.000022	4	DelayACK	10.62.42.244	64	58839	25	1	0		33	58839 > 25 [ACK] Seq=
7	18.89031		zOS outbound	10.62.42.244	64	58839	25	1	15	16	33	C[redacted]BT2
8	0.006324	7		10.64.4.19	61	25	58839	33	0		16	25 > 58839 [ACK] Seq=
9	0.000176			10.64.4.19	61	25	58839	33	87	120	16	S: 250-radiuslx.f
10	0.000090	9	zOS outbound	10.62.42.244	64	58839	25	16	28	44	120	C: MAIL FROM:<ver
11	0.013534	10		10.64.4.19	61	25	58839	120	8	128	44	S: 250 Ok
12	0.000038	11	zOS outbound	10.62.42.244	64	58839	25	44	28	72	128	C: RCPT TO:<pps@r
13	0.014050	12		10.64.4.19	61	25	58839	128	8	136	72	S: 250 Ok
14	0.000044	13	zOS outbound	10.62.42.244	64	58839	25	72	6	78	136	C: DATA
15	0.006148	14		10.64.4.19	61	25	58839	136	37	173	78	S: 354 End data w
16	0.000164	15	zOS outbound	10.62.42.244	64	58839	25	78	1169	1247	173	C: DATA fragment, 14
17	0.013106	16		10.64.4.19	61	25	58839	173	30	203	1247	S: 250 Ok: queued
18	0.000032	17	zOS outbound	10.62.42.244	64	58839	25	1247	6	1253	203	C: DATA fragment, 6 b
19	0.006204	18		10.64.4.19	61	25	58839	203	9	212	1253	S: 221 Bye
20	0.000000		tcp_down	10.64.4.19	61	25	58839	212	0		1253	25 > 58839 [FIN, ACK]
21	0.000024	20	zOS outbound	10.62.42.244	64	58839	25	1253	0		213	58839 > 25 [PSH, ACK]
22	0.000010		tcp_down	10.62.42.244	64	58839	25	1253	0		213	58839 > 25 [FIN, PSH,
23	0.006312	22		10.64.4.19	61	25	58839	213	0		1254	25 > 58839 [ACK] Seq=

Session 13282 “Taming the Shark”
FTP to Shop Z fails



No.	Time	ACKed	tcp.len	ColoringRuleName	Source	TTL	s_port	d_port	Seq	NxtSeq	ACK	Info
1	0.000000		0	p0f zOS_Tstamp 65535:64:1:60:M*,N,W*,N,N,T	10.9.1.17	64	13188	21	0			13188 > 21 [SYN]
2	0.146856	1	0	p0f_AIX 65535:60:1:44:M*	170.225.15.117	45	21	13188	0		1	21 > 13188 [SYN,
3	0.000013	2	0	zOS ACK	10.9.1.17	64	13188	21	1		1	13188 > 21 [ACK]
4	0.149615		67	FTP-220 Welcome	170.225.15.117	45	21	13188	1	68	1	Ignored Unknown
5	0.000017	4	0	zOS ACK	10.9.1.17	64	13188	21	1		68	13188 > 21 [ACK]
6	0.147421		172	FTP-220 Welcome	170.225.15.117	45	21	13188	68	240	1	Ignored Unknown
7	0.000014	6	0	zOS ACK	10.9.1.17	64	13188	21	1		240	13188 > 21 [ACK]
8	0.202356		10	FTP-CMD: AUTH TLS	10.9.1.17	64	13188	21	1	11	240	Ignored Unknown
9	0.146597	8	18	FTP-234 SSL OK	170.225.15.117	45	21	13188	240	258	11	Ignored Unknown
10	0.000022	9	0	zOS ACK	10.9.1.17	64	13188	21	11		258	13188 > 21 [PSH,
11	0.000119		60	TLS Client Hello	10.9.1.17	64	13188	21	11	71	258	Client Hello
12	0.146259	11	7	TLS Alert	170.225.15.117	45	21	13188	258	265	71	Alert (Level: Fa
13	0.000010	12	0	zOS ACK	10.9.1.17	64	13188	21	71		265	13188 > 21 [PSH,
14	0.000335		0	tcp_down	170.225.15.117	45	21	13188	265		71	21 > 13188 [FIN,
15	0.000003	14	0	zOS ACK	10.9.1.17	64	13188	21	71		266	13188 > 21 [PSH,
16	0.000248		0	tcp_down	10.9.1.17	64	13188	21	71		266	13188 > 21 [FIN,
17	0.148572	16	0		170.225.15.117	45	21	13188	266		72	21 > 13188 [ACK]

Session 13282 “Taming the Shark”
iSCSI Performance


boston2013_3.pcap
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save 3way_HS rxmit TLS TLS_hi

Filter: iscsi

No.	Time	ACKed	ColoringRuleName	Source	TTL	s_port	d_port	Seq	tcp.len	NxtSeq	ACK	iSCSI-rsp	Info
1	0.000000		Windows	10.111.88.247	128	63043	3260	1	1460	1461	1		SCSI: Write(10) LUN: 0x00
50	0.195209		Nagle	10.111.88.247	128	63043	3260	65585	48	65633	1		SCSI: Read(10) LUN: 0x00
53	0.003828			10.111.134.122	63	3260	63043	1	48	49	65585	0.199037	SCSI: Response LUN: 0x00
55	0.269271	50	Nagle	10.111.134.122	63	3260	63043	49	1460	1509	65633	0.273099	SCSI: Data In LUN: 0x00 (
59	0.000165		Windows	10.111.88.247	128	63043	3260	65633	1440	67073	49		SCSI: Read(10) LUN: 0x00
64	0.000153	63	Windows	10.111.88.247	128	63043	3260	67073	48	67121	8289		SCSI: Read(10) LUN: 0x00
77	0.265960	64	Nagle	10.111.134.122	63	3260	63043	8289	1460	9749	67121	0.266113	SCSI: Data In LUN: 0x00 (
142	0.892472	141	Windows	10.111.88.247	128	63043	3260	67121	48	67169	76909		SCSI: Read(10) LUN: 0x00
195	0.382652			10.111.134.122	63	3260	63043	36769	1460	138229	67169		Login Command
200	0.000070	198	Windows	10.111.88.247	128	63043	3260	67169	48	67217	141149		SCSI: Read(10) LUN: 0x00

Thank You for attending this session

Come to "Taming the Shark" - A wireshark Hands On Lab Session at 4:30PM R202

Thank You for your Time at SHARE

http://tinyurl.com/ipwizards

Matthias Burkhard IBM Germany

ip.wizards@groups.facebook.com

Your feedback is important:
This was session 13282 at SHARE in Boston 2013
Taming the shark

12 Complete your sessions evaluation online at SHARE.org/BostonEval

Created using OpenOffice 4

Join our Community in IBM Greenhouse to get the latest profiles and discuss tcpip related issues in the community.

IBM Greenhouse is an open platform to share knowledge and expertise based on IBM Connections. It is free, register here: https://greenhouse.lotus.com/