IBM Software Group

TCP Packet Tracing – Part 1

Robert L Boretti Jr (robb@us.ibm.com)
Marvin Knight (knightm@us.ibm.com)
Advisory Software Engineers
24 May 2011

WebSphere® Support Technical Exchange

ON DEMAND BUSINESS™

Agenda

- **Main Focus - TCP Packet Tracing**
  - **What** is TCP - general *description*
  - **When** to capture - common *technical problems*
  - **Where** to capture - depends on *network topology*
  - **How** to capture - available *operating system* tools
  - Using **wireshark** to *analyze* a TCP Packet Trace
    - General **overview** – wireshark gui and packet panes
    - **Filter** a TCP connection
    - Basic breakdown of **TCP/IP communication flow**

What is TCP?

- **TCP (Transmission Control Protocol)** is a set of rules used along with the **Internet Protocol (IP)** to send data in the form of message units between computers over the Internet. While IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the packets that a message is divided into for efficient routing through the Internet.

- A **packet** is a **unit of data** that is routed between an *origin* and a *destination* on the Internet

- When you *send* or *receive* data (for example, a Web page), the message gets divided into these individual units of data. Each of these *packets* contains both the **sender's Internet address** and the **receiver's address.**

- Each of these packets is *separately* numbered.

TCP Packet Tracing!

When to Capture?

Where to Capture?

How to Capture?

When to Capture - *common technical problems*

- HTTP Header problems
  - Validation of **HTTP Header** information provided in **client requests** and **server responses** over TCP connections
  - Confirmation of proper **end of headers (0d 0a 0d 0a)** by both client and server
- Large file downloads or POST upload problems
  - **body data** "bytes" transmission verification (content-length)
  - **premature** connection closure or time-outs during data transmission
- General problems
  - TCP connect failures, premature connection closures and packet delays from **client side** or **server side**
- SSL handshake problems
  - **Client Hello** and **Server Hello** verification
  - **Cipher** selection and **key exchange** verification

Where to Capture - *depends on network topology*

- debugging communication between **client & web server**
  a) If direct connection exists (client**<--------->**web server)
     - Capture on *web server machine*
  b) If middle devices exist - firewall, proxy, load-balancer, etc (client**<----|---->**web server)
     - Capture on *web server machine*
     - Capture on *client machine*
     - Capture on *device* (if possible)
- debugging communication between **Plug-in & WebSphere Application Server**
  a) If direct connection exists (plug-in**<--------->**WebSphere)
     - Capture on *web server machine*
  b) If middle devices exist - firewall, proxy, load-balancer, etc (plug-in**<----|---->**WebSphere)
     - Capture on *web server machine*
     - Capture on *WebSphere machine*
     - Capture on *device* (if possible)

How to Capture - available *operating system* tools

- AIX - **iptrace**
  - Manpage -http://publib.boulder.ibm.com/infocenter/aix/v6r1/index.jsp?topic=/com.ibm.aix.cmds/doc/aixcmds3/iptrace.htm
- Linux® - **tcpdump**
  - Manpage - http://www.tcpdump.org/tcpdump_man.html
- Solaris - **snoop**
  - Manpage - http://docs.sun.com/app/docs/doc/819-2240/snoop-1m?a=view
- Microsoft® Windows® - **netmon** and **wireshark**
  - Download (netmon) - http://www.microsoft.com/downloads/en/details.aspx?FamilyID=983b941d-06cb-4658-b7f6-3088333d062f&displaylang=en%20
  - Download (wireshark) - http://www.wireshark.org/download.html

How to Capture - *iptrace examples*

- run iptrace on interface en1 to capture port 80 traffic from a single client IP to a server IP listening on port 80
  - **iptrace -a -i en1 -s clientip -b -d serverip -p 80 trace.out**
- to record packets coming in and going out to any host on every interface
  - **iptrace -a trace.out**
- to record packets coming in and going out from a server IP on any port for interface en0
  - **iptrace -a -i en0 -d serverip -b trace.out**

How to Capture - *tcpdump examples*

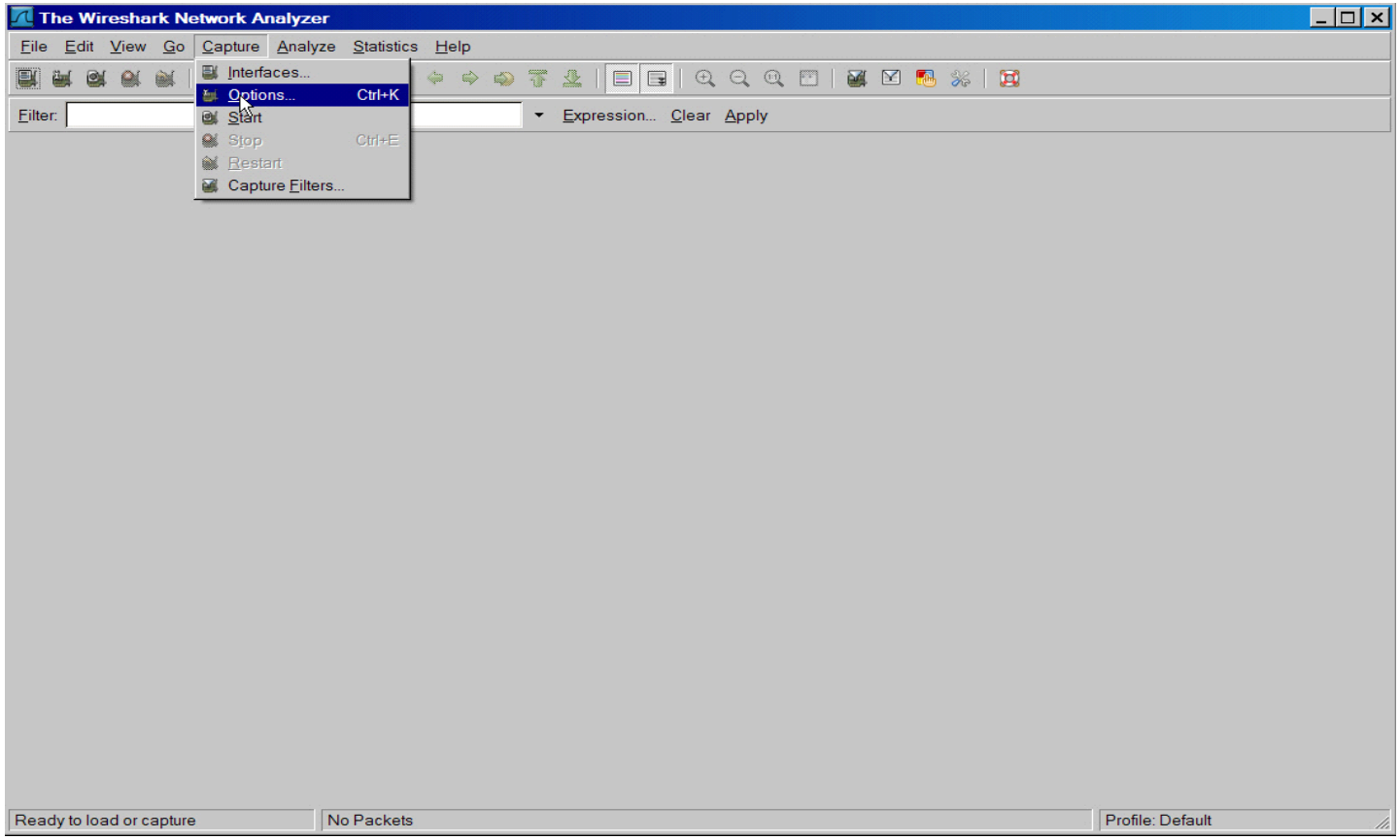
- A simple way to capture all packets on all interfaces to a binary file which is readable in wireshark
  - **tcpdump -s 2000 -w trace.out**
- To capture all packets on eth1 interface
  - **tcpdump -s 2000 -w trace.out -i eth1**
- To capture all packets on port 80 for a particular server IP
  - **tcpdump -s 2000 -w trace.out dst serverip and tcp port 80**

How to Capture - *snoop examples*

- To capture all packets on all interfaces in binary mode to a file
  - **snoop -o snoop.out**
- To capture all packets on interface hme0 in binary mode to a file
  - **snoop hme0 -o snoop.out**
- To capture all packets for a particular server IP in binary mode to a file
  - **snoop -o snoop.out serverip**

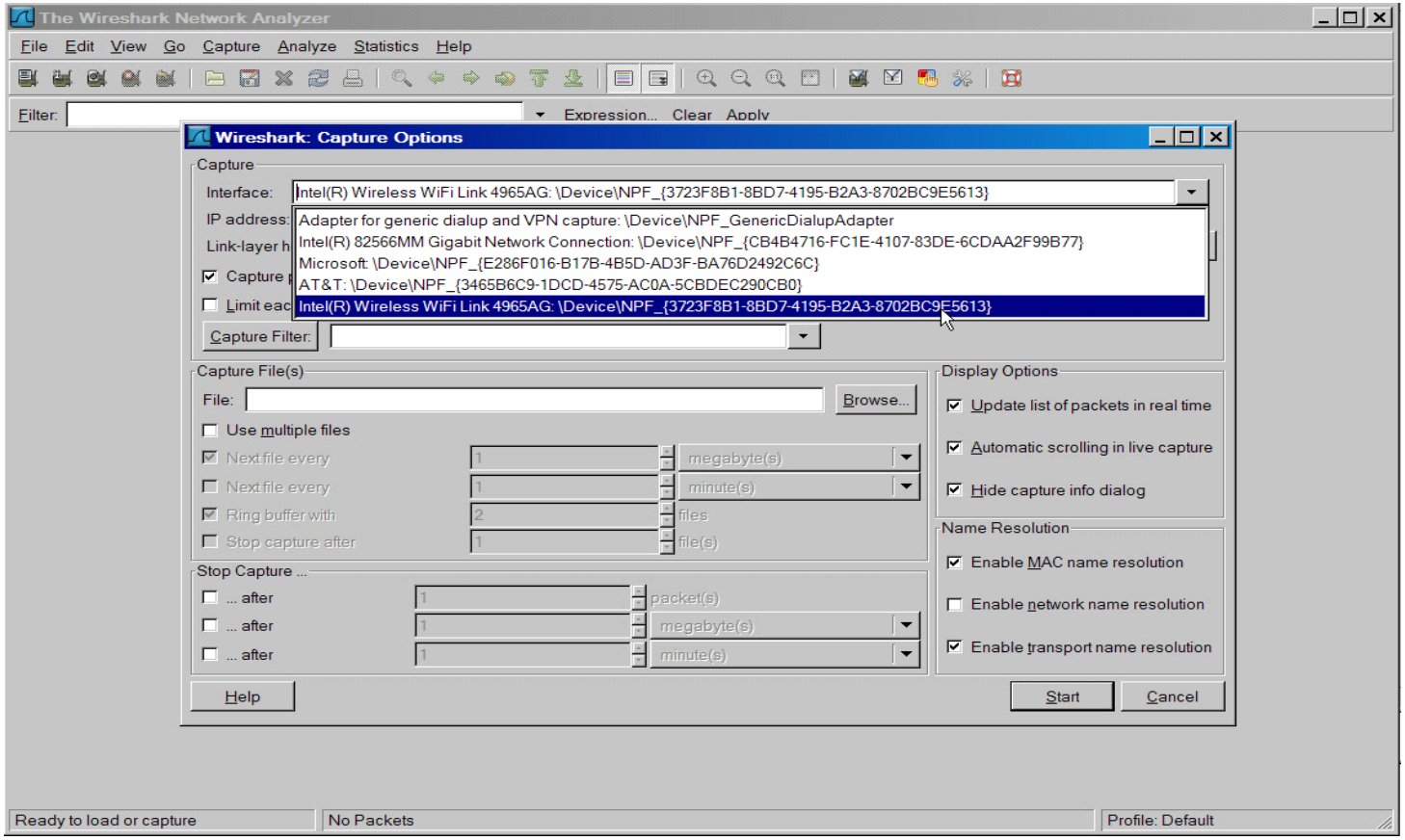
How to Capture - **Wireshark** *example*

- To begin capturing, select Capture -> Options



How to Capture - **Wireshark** *example*

- Then select the interface you wish to capture traffic on



How to Capture - **Wireshark** *example*

- Next, click “start” to begin capturing all traffic on the interface selected

The Wireshark Network Analyzer

File Edit View Go Capture Analyze Statistics Help

Filter:

Wireshark: Capture Options

Capture

Interface: Intel(R) Wireless WiFi Link 4965AG: \Device\NPF_{3723F8B1-8BD7-4195-B2A3-8702BC9E5613}

IP address: 192.168.0.10

Link-layer header type: Ethernet Buffer size: 1 megabyte(s) Wireless Settings

Capture packets in promiscuous mode

Limit each packet to 68 bytes

Capture Filter:

Capture File(s)

File: Browse...

Use multiple files

Next file every 1 megabyte(s)

Next file every 1 minute(s)

Ring buffer with 2 files

Stop capture after 1 file(s)

Display Options

Update list of packets in real time

Automatic scrolling in live capture

Hide capture info dialog

Name Resolution

Enable MAC name resolution

Enable network name resolution

Enable transport name resolution

Stop Capture ...

... after 1 packet(s)

... after 1 megabyte(s)

... after 1 minute(s)

Help Start Cancel

Ready to load or capture | No Packets | Profile: Default

How to Capture - **Wireshark** *example*

- If traffic is occuring on the interface, you will see TCP packets in the display window

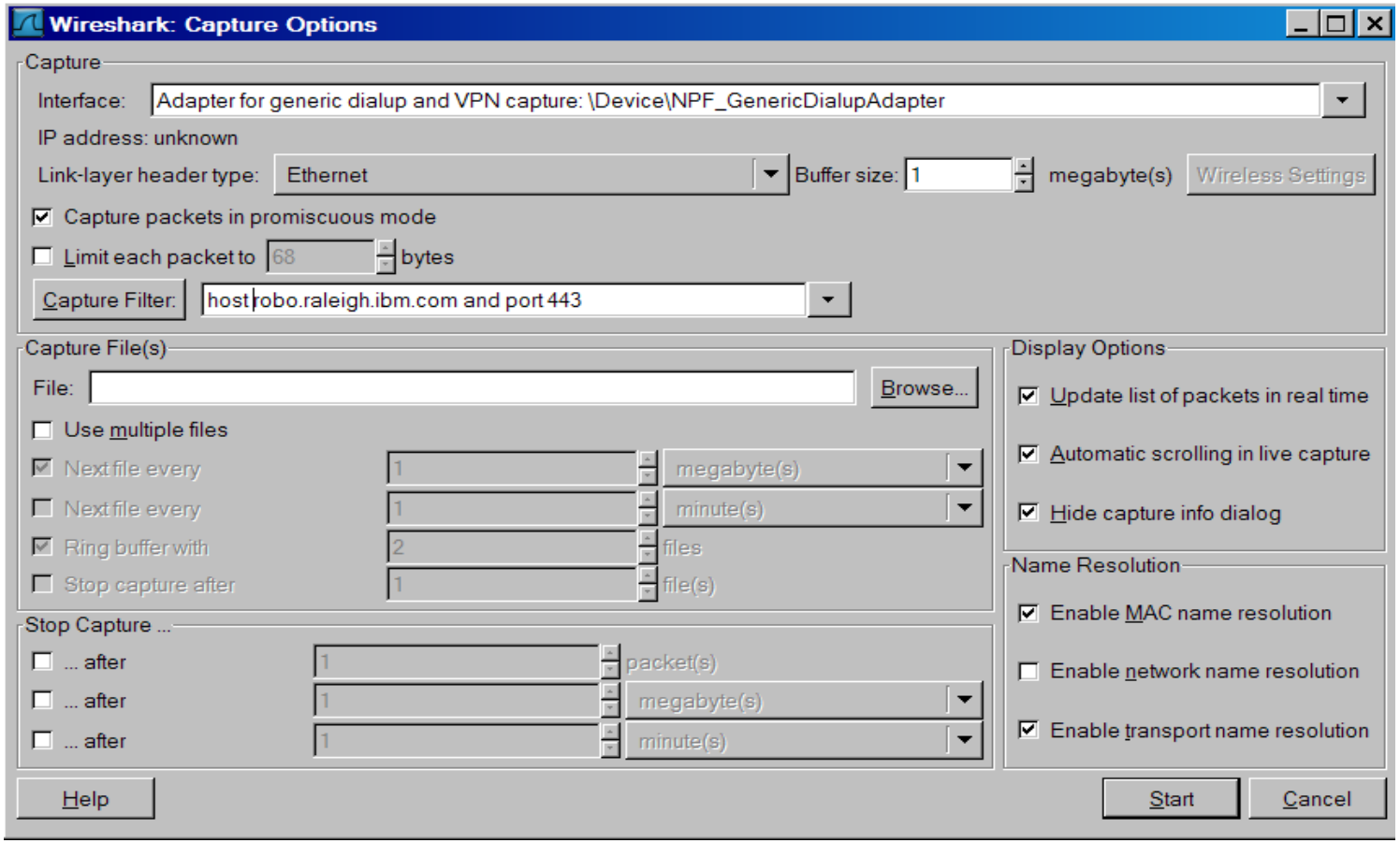
Edit View Go Capture Analyze Statistics Help

Expression... Clear Apply

	Time	Source	Destination	Protocol	Info
227	2011-05-11 16:31:17.751117	72.14.204.100	192.168.0.14	SSLv3	Server Hello, Change Cipher
228	2011-05-11 16:31:17.753315	192.168.0.14	72.14.204.100	SSLv3	Change Cipher Spec, Encrypt
229	2011-05-11 16:31:17.753341	192.168.0.14	72.14.204.100	SSLv3	[TCP Out-Of-Order] Change C
230	2011-05-11 16:31:17.753957	72.14.204.113	192.168.0.14	TCP	https > nssalertmgr [ACK] S
231	2011-05-11 16:31:17.754099	72.14.204.113	192.168.0.14	SSLv3	Server Hello, Change Cipher
232	2011-05-11 16:31:17.754628	192.168.0.14	72.14.204.113	SSLv3	Change Cipher Spec, Encrypt
233	2011-05-11 16:31:17.754649	192.168.0.14	72.14.204.113	SSLv3	[TCP Out-Of-Order] Change C
234	2011-05-11 16:31:17.757874	192.168.0.14	72.14.204.113	SSLv3	Application Data
235	2011-05-11 16:31:17.757904	192.168.0.14	72.14.204.113	SSLv3	[TCP Out-Of-Order] Applicat
236	2011-05-11 16:31:17.759979	72.14.204.138	192.168.0.14	TCP	https > camp [ACK] Seq=158
237	2011-05-11 16:31:17.761759	192.168.0.14	72.14.204.100	SSLv3	Application Data
238	2011-05-11 16:31:17.761787	192.168.0.14	72.14.204.100	SSLv3	[TCP Out-Of-Order] Applicat
239	2011-05-11 16:31:17.763820	72.14.204.138	192.168.0.14	SSLv3	Application Data, Applicati
240	2011-05-11 16:31:17.779484	72.14.204.138	192.168.0.14	TCP	http > nssagentmgr [SYN, AC
241	2011-05-11 16:31:17.779560	192.168.0.14	72.14.204.138	TCP	nssagentmgr > http [ACK] Se
242	2011-05-11 16:31:17.779584	192.168.0.14	72.14.204.138	TCP	[TCP Dup ACK 241#1] nssagen
243	2011-05-11 16:31:17.779893	192.168.0.14	72.14.204.138	HTTP	GET /tbproxy/getaccountinfo
244	2011-05-11 16:31:17.779917	192.168.0.14	72.14.204.138	HTTP	[TCP Out-Of-Order] GET /tbp
245	2011-05-11 16:31:17.790600	72.14.204.113	192.168.0.14	TCP	https > nssalertmgr [ACK] S
246	2011-05-11 16:31:17.799358	72.14.204.100	192.168.0.14	TCP	https > ctiprogramload [ACK
247	2011-05-11 16:31:17.813652	72.14.204.138	192.168.0.14	SSLv3	Application Data, Applicati
248	2011-05-11 16:31:17.819228	72.14.204.138	192.168.0.14	TCP	http > nssagentmgr [ACK] Se
249	2011-05-11 16:31:17.835488	72.14.204.138	192.168.0.14	HTTP	HTTP/1.1 200 OK (text/html
250	2011-05-11 16:31:17.846229	72.14.204.100	192.168.0.14	SSLv3	Application Data, Applicati
251	2011-05-11 16:31:17.891680	192.168.0.14	72.14.204.100	TCP	prchat-user > http [SYN] Se
252	2011-05-11 16:31:17.906081	72.14.204.113	192.168.0.14	SSLv3	Application Data, Applicati
253	2011-05-11 16:31:17.927039	192.168.0.14	72.14.204.100	TCP	prchat-user > http [SYN] Se
254	2011-05-11 16:31:17.929836	192.168.0.14	72.14.204.138	TCP	privatewire > https [ACK] S
255	2011-05-11 16:31:17.929865	192.168.0.14	72.14.204.138	TCP	[TCP Dup ACK 254#1] private
256	2011-05-11 16:31:17.930012	192.168.0.14	72.14.204.138	TCP	camp > https [ACK] Seq=1010

How to Capture - **Wireshark** *example*

- To limit the capture to a particular host and port, use the **host** and **port** *capture filter*



How to Capture - **Wireshark** *example*

- To stop the packet capture, select Capture -> Stop

Intel(R) Wireless WiFi Link 4965AG: Capturing - Wireshark

File Edit View Go Capture Analyze Statistics Help

Interfaces...
Options... Ctrl+K
Start
Stop Ctrl+E
Restart
Capture Filters...

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
5	2010-05	129.42.56.216	192.168.0.10	SSLv3	Server Hello, Certifica
6	2010-05-16	192.168.0.10	129.42.56.216	SSLv3	Client Key Exchange, Ch
7	2010-05-16 19:49:59.962571	129.42.56.216	192.168.0.10	TCP	https > syscomlan [ACK]
8	2010-05-16 19:49:59.965985	129.42.56.216	192.168.0.10	SSLv3	Change Cipher Spec, Enc
9	2010-05-16 19:50:00.092576	192.168.0.10	129.42.56.216	SSLv3	Application Data
10	2010-05-16 19:50:00.126765	129.42.56.216	192.168.0.10	SSLv3	Application Data
11	2010-05-16 19:50:00.127673	129.42.56.216	192.168.0.10	SSLv3	Application Data
12	2010-05-16 19:50:00.127706	192.168.0.10	129.42.56.216	TCP	syscomlan > https [ACK]
13	2010-05-16 19:50:01.333420	192.168.0.10	129.42.56.216	TCP	syscomlan > https [FIN,
14	2010-05-16 19:50:01.363593	129.42.56.216	192.168.0.10	TCP	https > syscomlan [FIN,
15	2010-05-16 19:50:01.363637	192.168.0.10	129.42.56.216	TCP	syscomlan > https [ACK]
16	2010-05-16 19:50:01.868706	192.168.0.10	129.42.56.216	TCP	rdrmshc > https [SYN] S
17	2010-05-16 19:50:01.978761	129.42.56.216	192.168.0.10	TCP	https > rdrmshc [SYN, A
18	2010-05-16 19:50:01.978797	192.168.0.10	129.42.56.216	TCP	rdrmshc > https [ACK] S
19	2010-05-16 19:50:01.979187	192.168.0.10	129.42.56.216	SSL	Client Hello
20	2010-05-16 19:50:02.081509	129.42.56.216	192.168.0.10	SSLv3	Server Hello, Certifica
21	2010-05-16 19:50:02.082060	192.168.0.10	129.42.56.216	SSLv3	Client Key Exchange, Ch
22	2010-05-16 19:50:02.183232	129.42.56.216	192.168.0.10	TCP	https > rdrmshc [ACK] S
23	2010-05-16 19:50:02.183685	129.42.56.216	192.168.0.10	SSLv3	Change Cipher Spec, Enc
24	2010-05-16 19:50:02.185809	192.168.0.10	129.42.56.216	SSLv3	Application Data
25	2010-05-16 19:50:02.286376	129.42.56.216	192.168.0.10	SSLv3	Application Data
26	2010-05-16 19:50:02.425924	192.168.0.10	129.42.56.216	TCP	rdrmshc > https [ACK] S
27	2010-05-16 19:50:09.293679	192.168.0.10	129.42.56.216	TCP	rdrmshc > https [RST, A

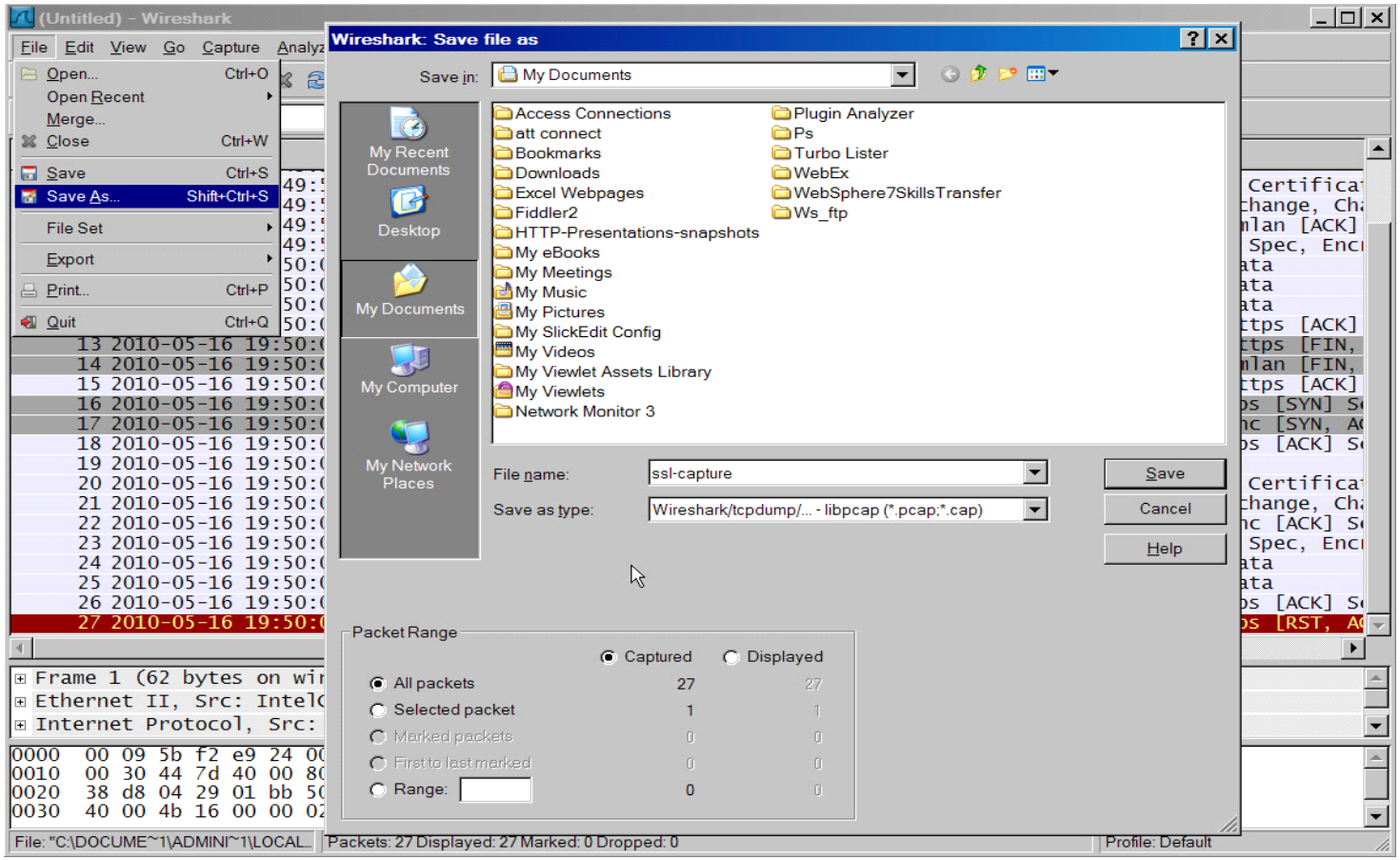
```
⊞ Frame 1 (62 bytes on wire, 62 bytes captured)
⊞ Ethernet II, Src: IntelCor_6d:dc:bb (00:1f:3b:6d:dc:bb), Dst: Netgear_f2:e9:24 (00:09:5b:f2:e9:24)
⊞ Internet Protocol, Src: 192.168.0.10 (192.168.0.10), Dst: 129.42.56.216 (129.42.56.216)

0000  00 09 5b f2 e9 24 00 1f  3b 6d dc bb 08 00 45 00   ..[..$.. ;m....E.
0010  00 30 44 7d 40 00 80 06  3b 96 c0 a8 00 0a 81 2a   .0D}@... ;......*
0020  38 d8 04 29 01 bb 50 09  27 67 00 00 00 00 70 02   8..)..P. 'g....p.
0030  40 00 4b 16 00 00 02 04  05 b4 01 01 04 02         @.K..... ......
```

Intel(R) Wireless WiFi Link 4965AG: <liv... | Packets: 27 Displayed: 27 Marked: 0 | Profile: Default

How to Capture - **Wireshark** *example*

- To save the packet capture, select File -> Save As



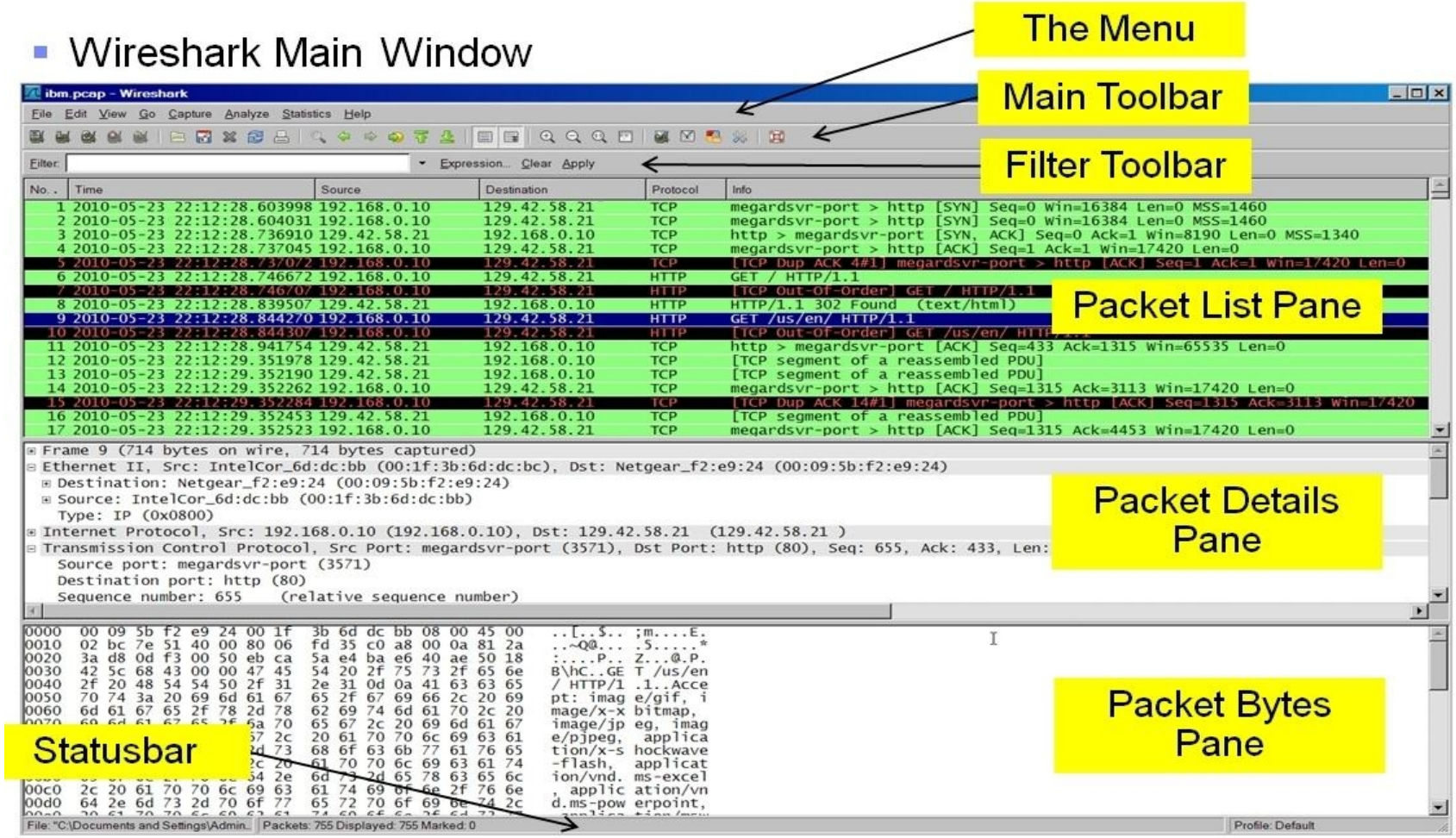
Using **Wireshark** to *analyze* a TCP Packet Trace “capture” file

Before we begin..

- Before attempting to *analyze* a TCP Packet Trace, it is essential to understand the **BASICS** behind a ***TCP connection***
  - A TCP connection consists of...
    - communication between two *network devices*, which have unique **MAC addresses**. A *MAC address* represents the physical address of a network device.
    - communication between two **IP addresses**, a *source* IP address and a *destination* IP address. The *IP address* represents the logical address on the Internet Protocol network.
    - communication between two **Ports**, a *randomly assigned port* of the device that initiates the connection and a *designated listening port* (e.g. 80, 443, 9080) of the device that accepts the connection.

General Overview - *wireshark gui and packet panes*

- Wireshark Main Window



General Overview - *wireshark gui and packet panes*

- The **packet list pane** - displays a summary of each packet captured. By clicking on packets in this pane you control what is displayed in the *packet details pane* and *packet bytes pane*.
  - **No.** - The number of the packet in the capture file. This number won't change even if a display filter is used.
  - **Time** - The timestamp of the packet.
  - **Source** - The address where this packet is coming from.
  - **Destination** - The address where this packet is going to.
  - **Protocol** - The protocol name in a short abbreviated version.
  - **Info** - Additional information about the packet content.

No.	Time	Source	Destination	Protocol	Info
4717	2011-05-05 17:28:26.284149	9.37.235.185	9.42.135.23	TCP	scol > 9081 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
4718	2011-05-05 17:28:26.284181	9.37.235.185	9.42.135.23	TCP	scol > 9081 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
4719	2011-05-05 17:28:26.284920	9.42.135.23	9.37.235.185	TCP	9081 > scol [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=146
4720	2011-05-05 17:28:26.284967	9.37.235.185	9.42.135.23	TCP	scol > 9081 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4721	2011-05-05 17:28:26.284984	9.37.235.185	9.42.135.23	TCP	[TCP Dup ACK 4720#1] scol > 9081 [ACK] Seq=1 Ack=1 Win=655
4722	2011-05-05 17:28:26.285898	9.37.235.185	9.42.135.23	TCP	scol > 9081 [PSH, ACK] Seq=1 Ack=1 Win=65535 Len=1323
4723	2011-05-05 17:28:26.285926	9.37.235.185	9.42.135.23	TCP	[TCP Out-Of-Order] scol > 9081 [PSH, ACK] Seq=1 Ack=1 Win=
4724	2011-05-05 17:28:26.300659	9.42.135.23	9.37.235.185	TCP	9081 > scol [ACK] Seq=1 Ack=1324 Win=65535 Len=0
4725	2011-05-05 17:28:26.321678	9.42.135.23	9.37.235.185	TCP	9081 > scol [ACK] Seq=1 Ack=1324 Win=65535 Len=1460
4726	2011-05-05 17:28:26.321705	9.42.135.23	9.37.235.185	TCP	9081 > scol [ACK] Seq=1461 Ack=1324 Win=65535 Len=1460
4727	2011-05-05 17:28:26.321740	9.37.235.185	9.42.135.23	TCP	scol > 9081 [ACK] Seq=1324 Ack=2921 Win=65535 Len=0
4728	2011-05-05 17:28:26.321758	9.37.235.185	9.42.135.23	TCP	[TCP Dup ACK 4727#1] scol > 9081 [ACK] Seq=1324 Ack=2921 W
4729	2011-05-05 17:28:26.322041	9.42.135.23	9.37.235.185	TCP	9081 > scol [ACK] Seq=2921 Ack=1324 Win=65535 Len=1460
4730	2011-05-05 17:28:26.322060	9.42.135.23	9.37.235.185	TCP	9081 > scol [ACK] Seq=4381 Ack=1324 Win=65535 Len=1460
4731	2011-05-05 17:28:26.322102	9.37.235.185	9.42.135.23	TCP	scol > 9081 [ACK] Seq=1324 Ack=5841 Win=65535 Len=0

General Overview - *wireshark gui and packet panes*

- The **packet details pane** - displays the packet selected in the *packet list pane* in more detail. This pane shows the protocols and protocol fields of the packet. The protocols and fields of the packet are displayed using a tree, which can be expanded and collapsed.

```
⊞ Frame 4723 (1377 bytes on wire, 1377 bytes captured)
⊞ Ethernet II, Src: Usi_ce:08:8c (00:1e:37:ce:08:8c), Dst: All-HSRP-routers_01 (00:00:0c:07:ac:01)
⊞ Internet Protocol, Src: 9.37.235.185 (9.37.235.185), Dst: 9.42.135.23 (9.42.135.23)
⊟ Transmission Control Protocol, Src Port: scol (1200), Dst Port: 9081 (9081), Seq: 1, Ack: 1, Len: 1323
    Source port: scol (1200)
    Destination port: 9081 (9081)
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1324    (relative sequence number)]
    Acknowledgement number: 1    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x18 (PSH, ACK)
    Window size: 65535
```

General Overview - *wireshark gui and packet panes*

- The **packet bytes pane** - displays the data from the packet selected in the *packet list pane*, and highlights the field selected in the *packet details pane*. As usual for a hexdump, the left side shows the offset in the packet data, in the middle the packet data is shown in a hexadecimal representation and on the right the corresponding ASCII characters are displayed.

```
0000  00 00 0c 07 ac 01 00 1e  37 ce 08 8c 08 00 45 00   ........ 7.....E.
0010  05 53 ab 56 40 00 80 06  c5 2e 09 25 eb b9 09 2a   .S.V@... ...%...*
0020  87 17 04 b0 23 79 8a 74  98 58 d5 f4 17 ea 50 18   ....#y.t .X....P.
0030  ff ff 56 cd 00 00 47 45  54 20 2f 73 6e 6f 6f 70   ..V...GE T /snoop
0040  2f 20 48 54 54 50 2f 31  2e 31 0d 0a 41 63 63 65   / HTTP/1 .1..Acce
0050  70 74 3a 20 69 6d 61 67  65 2f 67 69 66 2c 20 69   pt: imag e/gif, i
0060  6d 61 67 65 2f 6a 70 65  67 2c 20 69 6d 61 67 65   mage/jpe g, image
0070  2f 70 6a 70 65 67 2c 20  69 6d 61 67 65 2f 70 6a   /pjpeg,  image/pj
0080  70 65 67 2c 20 61 70 70  6c 69 63 61 74 69 6f 6e   peg, app lication
0090  2f 78 2d 73 68 6f 63 6b  77 61 76 65 2d 66 6c 61   /x-shock wave-fla
00a0  73 68 2c 20 61 70 70 6c  69 63 61 74 69 6f 6e 2f   sh, appl ication/
00b0  76 6e 64 2e 6d 73 2d 65  78 63 65 6c 2c 20 61 70   vnd.ms-e xcel, ap
00c0  70 6c 69 63 61 74 69 6f  6e 2f 76 6e 64 2e 6d 73   plicatio n/vnd.ms
00d0  2d 70 6f 77 65 72 70 6f  69 6e 74 2c 20 61 70 70   -powerpo int, app
```

General Overview - *wireshark gui and packet panes*

date / time of packet arrival

selected packet

MAC addresses of Src/Dst

IP addresses of Src/Dst

Src/Dst PORTS

ASCII data

ibm.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

No.	Time	Source	Destination	Protocol	Info
1	2010-05-23 22:12:28.603998	192.168.0.10	129.42.58.21	TCP	megardsvr-port > http [SYN] Seq=0 Win=16384 Len=0 MSS=1460
2	2010-05-23 22:12:28.604031	192.168.0.10	129.42.58.21	TCP	megardsvr-port > http [SYN] Seq=0 Win=16384 Len=0 MSS=1460
3	2010-05-23 22:12:28.736910	129.42.58.21	192.168.0.10	TCP	http > megardsvr-port [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1
4	2010-05-23 22:12:28.737045	192.168.0.10	129.42.58.21	TCP	megardsvr-port > http [ACK] Seq=1 Ack=1 Win=17420 Len=0
5	2010-05-23 22:12:28.737072	192.168.0.10	129.42.58.21	TCP	[TCP Dup ACK 4#1] megardsvr-port > http [ACK] Seq=1 Ack=1 Win=17
6	2010-05-23 22:12:28.746672	192.168.0.10	129.42.58.21	HTTP	GET / HTTP/1.1

```
Frame 6 (708 bytes on wire, 708 bytes captured)
Ethernet II, Src: IntelCor_6d:dc:bb (00:1f:3b:6d:dc:bc), Dst: Netgear_f2:e9:24 (00:09:5b:f2:e9:24)
Internet Protocol, Src: 192.168.0.10 (192.168.0.10), Dst: 129.42.58.21 (129.42.58.21)
Transmission Control Protocol, Src Port: megardsvr-port (3571), Dst Port: http (80), Seq: 1, Ack: 1, Len: 654
Hypertext Transfer Protocol
```

```
0110  2f 76 6e 64 2e 6d 73 2d  78 70 73 64 6f 63 75 6d   /vnd.ms- xpsdocum
0120  65 6e 74 2c 20 61 70 70  6c 69 63 61 74 69 6f 6e   ent, app lication
0130  2f 78 2d 6d 73 2d 78 62  61 70 2c 20 61 70 70 6c   /x-ms-xb ap, appl
0140  69 63 61 74 69 6f 6e 2f  78 2d 6d 73 2d 61 70 70   ication/ x-ms-app
0150  6c 69 63 61 74 69 6f 6e  2c 20 2a 2f 2a 0d 0a 41   lication , */*..A
0160  63 63 65 70 74 2d 4c 61  6e 67 75 61 67 65 3a 20   ccept-La nguage:
0170  65 6e 2d 75 73 0d 0a 55  41 2d 43 50 55 3a 20 78   en-us..U A-CPU: x
0180  38 36 0d 0a 41 63 63 65  70 74 2d 45 6e 63 6f 64   86..Acce pt-Encod
0190  69 6e 67 3a 20 67 7a 69  70 2c 20 64 65 66 6c 61   ing: gzi p, defla
01a0  74 65 0d 0a 55 73 65 72  2d 41 67 65 6e 74 3a 20   te..User -Agent:
01b0  4d 6f 7a 69 6c 6c 61 2f  34 2e 30 20 28 63 6f 6d   Mozilla/ 4.0 (com
01c0  70 61 74 69 62 6c 65 3b  20 4d 53 49 45 20 37 2e   patible;  MSIE 7.
01d0  30 3b 20 57 69 6e 64 6f  77 73 20 4e 54 20 35 2e   0; Windo ws NT 5.
01e0  31 3b 20 47 54 42 36 2e  34 3b 20 2e 4e 45 54 20   1; GTB6. 4; .NET
01f0  43 4c 52 20 32 2e 30 2e  35 30 37 32 37 3b 20 49   CLR 2.0. 50727; I
0200  6e 66 6f 50 61 74 68 2e  32 3b 20 2e 4e 45 54 20   nfoPath. 2; .NET
0210  43 4c 52 20 31 2e 31 2e  34 33 32 32 3b 20 2e 4e   CLR 1.1. 4322; .N
0220  45 54 20 43 4c 52 20 33  2e 30 2e 30 34 35 30 36   ET CLR 3 .0.04506
0230  2e 33 30 3b 20 4d 53 2d  52 54 43 20 4c 4d 20 38   .30; MS- RTC LM 8
0240  29 0d 0a 48 6f 73 74 3a  20 77 77 77 2e 69 62 6d   )..Host:  www.
0250  2e 63 6f 6d 0d 0a 43 6f  6e 6e 65 63 74 69 6f 6e   .com..Co nnection
0260  3a 20 4b 65 65 70 2d 41  6c 69 76 65 0d 0a 43 6f   : Keep-A live..Co
0270  6f 6b 69 65 3a 20 55 6e  69 63 61 4e 49 4f 44 49   okie: Un icaNIODI
0280  44 3d 6c 32 7a 64 31 59  37 6f 32 35 6a 2d 57 61   D=l2zd1Y 7o25j-Wa
0290  71 4b 4f 53 49 3b 20 69  62 6d 53 75 72 76 65 79   qKOSI; i  Survey
02a0  3d 31 32 38 39 36 36 37  32 36 35 33 32 38 3b 20   =1289667 265328;
02b0  49 42 4d 50 4f 4c 4c 43  4f 4f 4b 49 45 3d 22 22    POLLC OOKIE=""
02c0  0d 0a 0d 0a                                        ....
```

File: "C:\Documents and Settings\Admin... Packets: 755 Displayed: 755 Marked: 0 Profile: Default

Filter a TCP connection

Since TCP packet traces can contain many TCP packets from different *Mac addresses*, *IP addresses* and *ports*, it is important to understand how to ***filter a packet trace*** when needed ***to see the complete communication for a single client<----->server connection***

Filter a TCP connection

- Right-click on a packet of interest and select “follow TCP Stream”

ibm.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
644	2010-05-23 22:12:34.165827	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
645	2010-05-23 22:12:34.165896	192.168.0.10	129.42.58.21	TCP	cmc-port > http [ACK] Seq=502 Ack=16
646	2010-05-23 22:12:34.165932	192.168.0.10	129.42.58.21	TCP	[TCP Dup ACK 645#1] cmc-port > http
647	2010-05-23 22:12:34.166089	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
648	2010-05-23 22:12:34.166160	192.168.0.10	129.42.58.21	TCP	cmc-port > http [ACK] Seq=502 Ack=16
649	2010-05-23 22:12:34.166178	192.168.0.10	129.42.58.21	TCP	[TCP Dup ACK 648#1] cmc-port > http
650	2010-05-23 22:12:34.166421	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
651	2010-05-23 22:12:34.167367	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
652	2010-05-23 22:12:34.167418	192.168.0.10	129.42.58.21		> http [ACK] Seq=502 Ack=16
653	2010-05-23 22:12:34.167437	192.168.0.10	129.42.58.21		ACK 652#1] cmc-port > http
654	2010-05-23 22:12:34.167649	129.42.58.21	192.168.0.10		gment of a reassembled PDU]
655	2010-05-23 22:12:34.167715	192.168.0.10	129.42.58.21		> http [ACK] Seq=502 Ack=16
656	2010-05-23 22:12:34.167734	192.168.0.10	129.42.58.21		ACK 655#1] cmc-port > http
657	2010-05-23 22:12:34.168440	129.42.58.21	192.168.0.10		gment of a reassembled PDU]
658	2010-05-23 22:12:34.169322	129.42.58.21	192.168.0.10		gment of a reassembled PDU]
659	2010-05-23 22:12:34.169380	192.168.0.10	129.42.58.21		> http [ACK] Seq=502 Ack=17
660	2010-05-23 22:12:34.169403	192.168.0.10	129.42.58.21		ACK 659#1] cmc-port > http
661	2010-05-23 22:12:34.169541	129.42.58.21	192.168.0.10		gment of a reassembled PDU]
662	2010-05-23 22:12:34.169618	192.168.0.10	129.42.58.21		> http [ACK] Seq=502 Ack=17
663	2010-05-23 22:12:34.169652	192.168.0.10	129.42.58.21		ACK 662#1] cmc-port > http
664	2010-05-23 22:12:34.175916	129.42.58.21	192.168.0.10		gment of a reassembled PDU]
665	2010-05-23 22:12:34.176320	129.42.58.21	192.168.0.10		gment of a reassembled PDU]
666	2010-05-23 22:12:34.176381	192.168.0.10	129.42.58.21		> http [ACK] Seq=502 Ack=17
667	2010-05-23 22:12:34.176402	192.168.0.10	129.42.58.21		ACK 666#1] cmc-port > http
668	2010-05-23 22:12:34.176569	129.42.58.21	192.168.0.10		gment of a reassembled PDU]
669	2010-05-23 22:12:34.176637	192.168.0.10	129.42.58.21		> http [ACK] Seq=502 Ack=17
670	2010-05-23 22:12:34.176659	192.168.0.10	129.42.58.21		ACK 669#1] cmc-port > http

Mark Packet (toggle)
Set Time Reference (toggle)
Apply as Filter
Prepare a Filter
Conversation Filter
Colorize Conversation
SCTP
Follow TCP Stream
Follow UDP Stream
Follow SSL Stream
Copy
Export Selected Packet Bytes...
Decode As...
Print...
Show Packet in New Window

```
Frame 652 (54 bytes on wire, 54 bytes captured)
Ethernet II, Src: IntelCor_6d:dc:bc (00:1f:3b:6d:dc:bc), Dst: Netgear_f2:e9:24 (00:09:5b:f2:e9:24)
Internet Protocol, Src: 192.168.0.10 (192.168.0.10), Dst: 129.42.58.21  (129.42.58.21)
Transmission Control Protocol, Src Port: cmc-port (3576), Dst Port: http (80), Seq: 502, Ack: 166281, Len: 0

0000  00 09 5b f2 e9 24 00 1f  3b 6d dc bb 08 00 45 00   ..[..$.. ;m....E.
0010  00 28 81 62 40 00 80 06  fc b8 c0 a8 00 0a 81 2a   .(.b@... .......*
0020  3a d8 0d f8 00 50 23 0a  3a 01 ef de 4f e7 50 10   :....P#. :...O.P.
0030  44 0c 43 fa 00 00                                  D.C...
```

Filter a TCP connection

Filter displayed

Requests/Responses handled over this connection

Filter a TCP connection

Filter → (ip.addr eq **192.168.0.10** and ip.addr eq **129.42.58.21**) and (tcp.port eq **3576** and tcp.port eq **80**)

ibm.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: (ip.addr eq 192.168.0.10 and ip.addr Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
156	2010-05-23 22:12:31.624510	192.168.0.10	129.42.58.21	TCP	cmc-port > http [SYN] Seq=0 Win=16384 Len=0 MSS=1460
183	2010-05-23 22:12:31.706726	129.42.58.21	192.168.0.10	TCP	http > cmc-port [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1340
184	2010-05-23 22:12:31.706858	192.168.0.10	129.42.58.21	TCP	cmc-port > http [ACK] Seq=1 Ack=1 Win=17420 Len=0
185	2010-05-23 22:12:31.706888	192.168.0.10	129.42.58.21	TCP	[TCP Dup ACK 184#1] cmc-port > http [ACK] Seq=1 Ack=1 Win=17420 Len=0
189	2010-05-23 22:12:31.708611	192.168.0.10	129.42.58.21	HTTP	GET /images/icp/T147059A31886057/ceo-ideas-security.swf HTTP/1.1
190	2010-05-23 22:12:31.708641	192.168.0.10	129.42.58.21	HTTP	[TCP Out-Of-Order] GET /images/icp/T147059A31886057/ceo-ideas-security.swf HTTP/1.1
193	2010-05-23 22:12:31.809646	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
194	2010-05-23 22:12:31.809865	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
195	2010-05-23 22:12:31.809936	192.168.0.10	129.42.58.21	TCP	cmc-port > http [ACK] Seq=502 Ack=925 Win=16496 Len=0
196	2010-05-23 22:12:31.809963	192.168.0.10	129.42.58.21	TCP	[TCP Dup ACK 195#1] cmc-port > http [ACK] Seq=502 Ack=925 Win=16496 Len=0
197	2010-05-23 22:12:31.810165	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
198	2010-05-23 22:12:31.810767	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
199	2010-05-23 22:12:31.810824	192.168.0.10	129.42.58.21	TCP	cmc-port > http [ACK] Seq=502 Ack=2801 Win=17420 Len=0
200	2010-05-23 22:12:31.810847	192.168.0.10	129.42.58.21	TCP	[TCP Dup ACK 199#1] cmc-port > http [ACK] Seq=502 Ack=2801 Win=17420 Len=0
201	2010-05-23 22:12:31.810983	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
211	2010-05-23 22:12:31.911807	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
212	2010-05-23 22:12:31.911944	192.168.0.10	129.42.58.21	TCP	cmc-port > http [ACK] Seq=502 Ack=5481 Win=17420 Len=0
213	2010-05-23 22:12:31.911971	192.168.0.10	129.42.58.21	TCP	[TCP Dup ACK 212#1] cmc-port > http [ACK] Seq=502 Ack=5481 Win=17420 Len=0
214	2010-05-23 22:12:31.912159	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
215	2010-05-23 22:12:31.912637	129.42.58.21	[illegible]	[illegible]	[illegible] reassembled PDU]
216	2010-05-23 22:12:31.912696	192.168.0.	[illegible]	[illegible]	] Seq=502 Ack=8161 Win=17420 Len=0
217	2010-05-23 22:12:31.912715	192.168.0.	[illegible]	[illegible]	cmc-port > http [ACK] Seq=502 Ack=8161 Win=17420 Len=0
218	2010-05-23 22:12:31.912870	129.42.58.	[illegible]	[illegible]	assembled PDU]
219	2010-05-23 22:12:31.912934	192.168.0.10	129.42.58.21	TCP	cmc-port > http [ACK] Seq=502 Ack=9501 Win=17420 Len=0
220	2010-05-23 22:12:31.912953	192.168.0.10	129.42.58.21	TCP	[TCP Dup ACK 219#1] cmc-port > http [ACK] Seq=502 Ack=9501 Win=17420 Len=0
221	2010-05-23 22:12:31.913753	129.42.58.21	192.168.0.10	TCP	[TCP segment of a reassembled PDU]

```
⊞ Frame 189 (555 bytes on wire, 555 bytes captured)
⊞ Ethernet II, Src: IntelCor_6d:dc:bc (00:1f:3b:6d:dc:bc), Dst: Netgear_f2:e9:24 (00:09:5b:f2:e9:24)
⊞ Internet Protocol, Src: 192.168.0.10 (192.168.0.10), Dst: 129.42.58.21 (129.42.58.21)
⊞ Transmission Control Protocol, Src Port: cmc-port (3576), Dst Port: http (80), Seq: 1, Ack: 1, Len: 501
⊞ Hypertext Transfer Protocol

0000  00 09 5b f2 e9 24 00 1f  3b 6d dc bb 08 00 45 00   ..[..$.. ;m....E.
0010  02 1d 7f 50 40 00 80 06  fc d5 c0 a8 00 0a 81 2a   ...P@... .......*
                                                         :....P#. 8....._P.
                                                         D.....GE T /image
                                                         s/icp/T1 47059A31
                                                         886057/c eo-ideas
0060  2d 73 65 63 75 72 69 74  79 2e 73 77 66 20 48 54   -securit y.swf HT
```

MAC addresses of Src/Dst

IP addresses of Src/Dst

Src/Dst PORTS

Basic breakdown of TCP/IP communication flow

Now that you have seen how to filter a TCP connection, it is also important to have a *general* understanding of the ***sequence of events which take place over a connection***. This understanding is vital in ***"logically" following the conversation between a client and server***

Basic breakdown of TCP/IP communication flow

- Filter → (ip.addr eq **10.12.5.67** and ip.addr eq **92.168.185.2**) and (tcp.port eq **1403** and tcp.port eq **8080**)

Filter: (ip.port eq 1403 and tcp.port eq 8080) Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
49516	2010-03-18 08:45:31.806119275	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [SYN] Seq=0 Win=65535 Len=0 MSS=1460
49517	2010-03-18 08:45:31.806120941	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
49518	2010-03-18 08:45:31.826666556	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=1 Ack=1 Win=65535 Len=0
49519	2010-03-18 08:45:31.833256496	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49529	2010-03-18 08:45:31.841242625	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49541	2010-03-18 08:45:31.905884392	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [ACK] Seq=1 Ack=2629 Win=65535 Len=0
49542	2010-03-18 08:45:31.934601589	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49543	2010-03-18 08:45:31.942485898	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49544	2010-03-18 08:45:31.942511246	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [ACK] Seq=1 Ack=5549 Win=65535 Len=0
49547	2010-03-18 08:45:31.950379888	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49548	2010-03-18 08:45:31.971131343	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49549	2010-03-18 08:45:31.971150873	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [ACK] Seq=1 Ack=8469 Win=65535 Len=0
49552	2010-03-18 08:45:31.978916210	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49553	2010-03-18 08:45:31.986901417	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49554	2010-03-18 08:45:31.986919017	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [ACK] Seq=1 Ack=11389 Win=65535 Len=0
49557	2010-03-18 08:45:31.999581429	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49558	2010-03-18 08:45:32.007469767	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49559	2010-03-18 08:45:32.007488705	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [ACK] Seq=1 Ack=14309 Win=65535 Len=0
49562	2010-03-18 08:45:32.015353953	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49563	2010-03-18 08:45:32.023242349	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49564	2010-03-18 08:45:32.023261685	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [ACK] Seq=1 Ack=17229 Win=65535 Len=0
49567	2010-03-18 08:45:32.031135677	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49568	2010-03-18 08:45:32.039026767	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49569	2010-03-18 08:45:32.039052146	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [ACK] Seq=1 Ack=20149 Win=65535 Len=0
49572	2010-03-18 08:45:32.048202546	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49573	2010-03-18 08:45:32.056088039	10.12.5.67	92.168.185.2	TCP	[TCP segment of a reassembled PDU]
49574	2010-03-18 08:45:32.056107351	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [ACK] Seq=1 Ack=23069 Win=65535 Len=0

```
[Next sequence number: 1169    (relative sequence number)]
Acknowledgement number: 1    (relat
Header length: 20 bytes
⊞ Flags: 0x18 (PSH, ACK)
Window size: 65535
```

```
0000  00 1a 64 a8 85 a6 00 01  64 f9 1a 01 08 00 45 00   ..d..... d.....E.
0010  04 b8 46 60 40 00 79 06  2d d3 0a 0c 05 43 c0 a8   ..F`@.y. -....C..
0020  b9 15 05 7b 1f 90 bc 29  84 13 0b f3 12 7a 50 18   ...{...) .....zP.
0030  ff ff 17 ee 00 00 50 4f  53 54 20 2f 52 54 4d 41   ......PO ST /RTMA
0040  41 2f 52 54 4d 2f 53 74  6f 72 65 2f 73 74 6f 72   A/RTM/St ore/stor
0050  65 50 72 6f 6a 65 63 74  44 65 74 61 69 6c 73 2e   eProject Details.
0060  6a 73 70 3f 69 6e 69 74  69 61 74 69 76 65 49 64   jsp?init iativeId
0070  3d 43 4f 4d 36 30 33 37  20 48 54 54 50 2f 31 2e   =COM6037 HTTP/1.
0080  31 0d 0a 41 63 63 65 70  74 3a 20 69 6d 61 67 65   1..Accep t: image
0090  2f 67 69 66 2c 20 69 6d  61 67 65 2f 78 2d 78 62   /gif, im age/x-xb
00a0  69 74 6d 61 70 2c 20 69  6d 61 67 65 2f 6a 70 65   itmap, i mage/jpe
00b0  67 2c 20 69 6d 61 67 65  2f 70 6a 70 65 67 2c 20   g, image /pjpeg,
00c0  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 2d 73 68   applicat ion/x-sh
```

SYN and SYN,ACK

ACK

DATA

PSH,ACK

Basic breakdown of TCP/IP communication flow

- Filter → (ip.addr eq **10.12.5.67** and ip.addr eq **92.168.185.2**) and (tcp.port eq **1403** and tcp.port eq **8080**)

Filter: tcp.port eq 1403 and tcp.port eq 8080) Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
49794	2010-03-18 08:45:32.900991408	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=99281 Win=65535 Len=0
49796	2010-03-18 08:45:32.919159357	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=102201 Win=65535 Len=0
49797	2010-03-18 08:45:32.934966080	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=105121 Win=65535 Len=0
49798	2010-03-18 08:45:32.934967812	92.168.185.2	10.12.5.67	HTTP	Continuation or non-HTTP traffic
49799	2010-03-18 08:45:32.963526832	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=108041 Win=65535 Len=0
49800	2010-03-18 08:45:32.963529730	92.168.185.2	10.12.5.67	HTTP	Continuation or non-HTTP traffic
49801	2010-03-18 08:45:32.979362017	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=110961 Win=65535 Len=0
49802	2010-03-18 08:45:32.995036652	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=113881 Win=65535 Len=0
49803	2010-03-18 08:45:33.010811134	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=116801 Win=65535 Len=0
49804	2010-03-18 08:45:33.027090625	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=119721 Win=65535 Len=0
49806	2010-03-18 08:45:33.042757626	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=5
49809	2010-03-18 08:45:33.058533052	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=5
49810	2010-03-18 08:45:33.074307187	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=5
49811	2010-03-18 08:45:33.090081191	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=131401 Win=65535 Len=0
49812	2010-03-18 08:45:33.251524853	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=132242 Win=64694 Len=0
50584	2010-03-18 08:45:42.330023466	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [FIN, ACK] Seq=13224
50586	2010-03-18 08:45:42.350582851	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [ACK] Seq=51435 Ack=
50657	2010-03-18 08:45:43.417227640	10.12.5.67	92.168.185.2	TCP	prm-nm-np > http-alt [FIN, ACK] Seq=51435
50658	2010-03-18 08:45:43.417229392	92.168.185.2	10.12.5.67	TCP	http-alt > prm-nm-np [ACK] Seq=132243 Ack=51436 Win=65535 Len=0

```
Sequence number: 131401    (relative sequence number)
[Next sequence number: 132242    (relative sequence number)]
Acknowledgement number: 51435    (relative ack number)
Header length: 20 bytes
⊟ Flags: 0x18 (PSH, ACK)
```

```
0000  00 01 64 f9 1a 01 00 1a  64 a8 85 a6 08 00 45 00   ..d..... d.....E.
0010  03 71 35 2f 40 00 3c 06  7d 4b c0 a8 b9 15 0a 0c   .q5/@.<. }K......
0020  05 43 1f 90 05 7b 0b f5  13 c2 bc 2a 4c fd 50 18   .C...{.. ...*L.P.
0030  ff ff 00 00 00 00 6e 62  6f 78 3e 3c 69 6e 70 75   ......nb ox><inpu
0040  74 20 74 79 70 65 3d 62  75 74 74 6f 6e 20 63 6c   t type=b utton cl
0050  61 73 73 3d 62 75 74 74  6f 6e 2d 74 20 76 61 6c   ass=butt on-t val
0060  75 65 3d 22 50 72 65 76  69 6f 75 73 22 20 6f 6e   ue="Prev ious" on
0070  43 6c 69 63 6b 3d 22 6a  61 76 61 53 63 72 69 70   Click="j avaScrip
0080  74 3a 73 68 6f 77 70 72  65 76 69 6f 75 73 28 29   t:showpr evious()
0090  22 3e 3c 2f 73 70 61 6e  3e 3c 2f 54 44 3e 3c 54   "></span ></TD><T
00a0  44 20 76 61 6c 69 67 6e  3d 6d 69 64 64 6c 65 20   D valign =middle
00b0  61 6c 69 67 6e 3d 72 69  67 68 74 20 3e 3c 73 70   align=ri ght ><sp
00c0  61 6e 20 63 6c 61 73 73  3d 62 75 74 74 6f 6e 62   an class =buttonb
00d0  6f 78 3e 3c 69 6e 70 75  74 20 74 79 70 65 3d 62   ox><inpu t type=b
```

FIN, ACK from server

ACK from client

FIN, ACK from client

ACK from server

Summary

- Provided a description of TCP
- Listed some common technical problems that would warrant gathering a TCP packet trace to help with debugging. *More on debugging will be covered in the* **TCP Packet Tracing - Part 2** *WebSphere Support Technical Exchange*
- Made suggestions on where to collect TCP packet tracing depending on the network topology
- Provided examples of the available operating system TCP packet capture tools
- Walked through the wireshark gui and packet panes
- Demonstrated how to Filter a TCP connection
- Finally, TCP/IP communication flow was explained

Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:
  http://www.ibm.com/software/websphere/support/supp_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:
  http://www.ibm.com/developerworks/websphere/community/
- Join the Global WebSphere Community:
  http://www.websphereusergroup.org
- Access key product show-me demos and tutorials by visiting IBM® Education Assistant:
  http://www.ibm.com/software/info/education/assistant
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:
  http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My Notifications emails:
  http://www.ibm.com/software/support/einfo.html

We Want to Hear From You!

Tell us about what you want to learn

Suggestions for future topics
Improvements and comments about our webcasts
We want to hear everything you have to say!

Please send your suggestions and comments to:
wsehelp@us.ibm.com

Questions and Answers