

# IS IA-1

## Implementation of Wireshark

Presented by:

Om Topale - 16O1O12OO55

Atharva Bet - 16O1O12OO59

Deven Sharma - 16O1O12OO49

Sudhanshu Shrirao -16O1O12OO51

# Introduction

- Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.
- You could think of a network packet analyzer as a measuring device for examining what's happening inside a network cable, just like an electrician uses a voltmeter for examining what's happening inside an electric cable (but at a higher level, of course).
- In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, that has changed. Wireshark is available for free, is open source, and is one of the best packet analyzers available today.

# Purpose

- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- QA engineers use it to verify network applications
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals

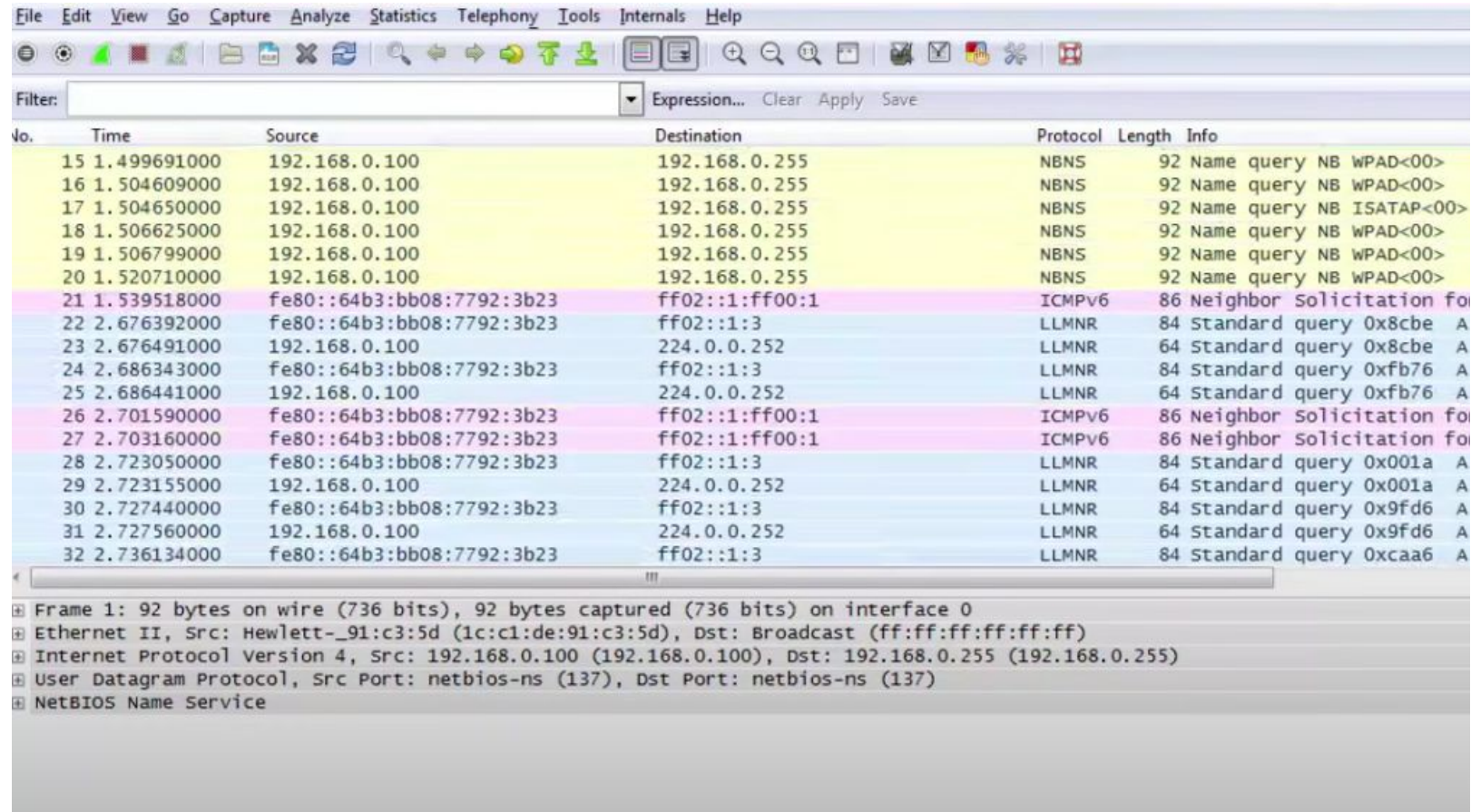
# Features

- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and many other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.
- ...and a lot more!

# Packet Sniffer

- Packet sniffer is a basic tool for observing network packet exchanges in a computer. As the name suggests, a packet sniffer captures (“sniffs”) packets being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured packets.
- A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself.
- The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must “understand” the structure of all messages exchanged by protocols.

# Packet Sniffer



The image shows a screenshot of the Wireshark network protocol analyzer. The main window displays a list of captured packets. The first 20 packets are NetBIOS Name Service (NBNS) queries from 192.168.0.100 to 192.168.0.255. Packets 21 and 27 are ICMPv6 Neighbor Solicitation messages. Packets 22 through 32 are LLMNR (Link-Local Multicast Name Resolution) queries from fe80::64b3:bb08:7792:3b23 to 224.0.0.252. The bottom pane shows the details of the first packet (Frame 1), which is an Ethernet II frame containing an Internet Protocol Version 4 packet, which in turn contains a User Datagram Protocol packet for NetBIOS Name Service.

No.	Time	Source	Destination	Protocol	Length	Info
15	1.499691000	192.168.0.100	192.168.0.255	NBNS	92	Name query NB WPAD<00>
16	1.504609000	192.168.0.100	192.168.0.255	NBNS	92	Name query NB WPAD<00>
17	1.504650000	192.168.0.100	192.168.0.255	NBNS	92	Name query NB ISATAP<00>
18	1.506625000	192.168.0.100	192.168.0.255	NBNS	92	Name query NB WPAD<00>
19	1.506799000	192.168.0.100	192.168.0.255	NBNS	92	Name query NB WPAD<00>
20	1.520710000	192.168.0.100	192.168.0.255	NBNS	92	Name query NB WPAD<00>
21	1.539518000	fe80::64b3:bb08:7792:3b23	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation fo
22	2.676392000	fe80::64b3:bb08:7792:3b23	ff02::1:3	LLMNR	84	Standard query 0x8cbe A
23	2.676491000	192.168.0.100	224.0.0.252	LLMNR	64	Standard query 0x8cbe A
24	2.686343000	fe80::64b3:bb08:7792:3b23	ff02::1:3	LLMNR	84	Standard query 0xfb76 A
25	2.686441000	192.168.0.100	224.0.0.252	LLMNR	64	Standard query 0xfb76 A
26	2.701590000	fe80::64b3:bb08:7792:3b23	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation fo
27	2.703160000	fe80::64b3:bb08:7792:3b23	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation fo
28	2.723050000	fe80::64b3:bb08:7792:3b23	ff02::1:3	LLMNR	84	Standard query 0x001a A
29	2.723155000	192.168.0.100	224.0.0.252	LLMNR	64	Standard query 0x001a A
30	2.727440000	fe80::64b3:bb08:7792:3b23	ff02::1:3	LLMNR	84	Standard query 0x9fd6 A
31	2.727560000	192.168.0.100	224.0.0.252	LLMNR	64	Standard query 0x9fd6 A
32	2.736134000	fe80::64b3:bb08:7792:3b23	ff02::1:3	LLMNR	84	Standard query 0xcaa6 A

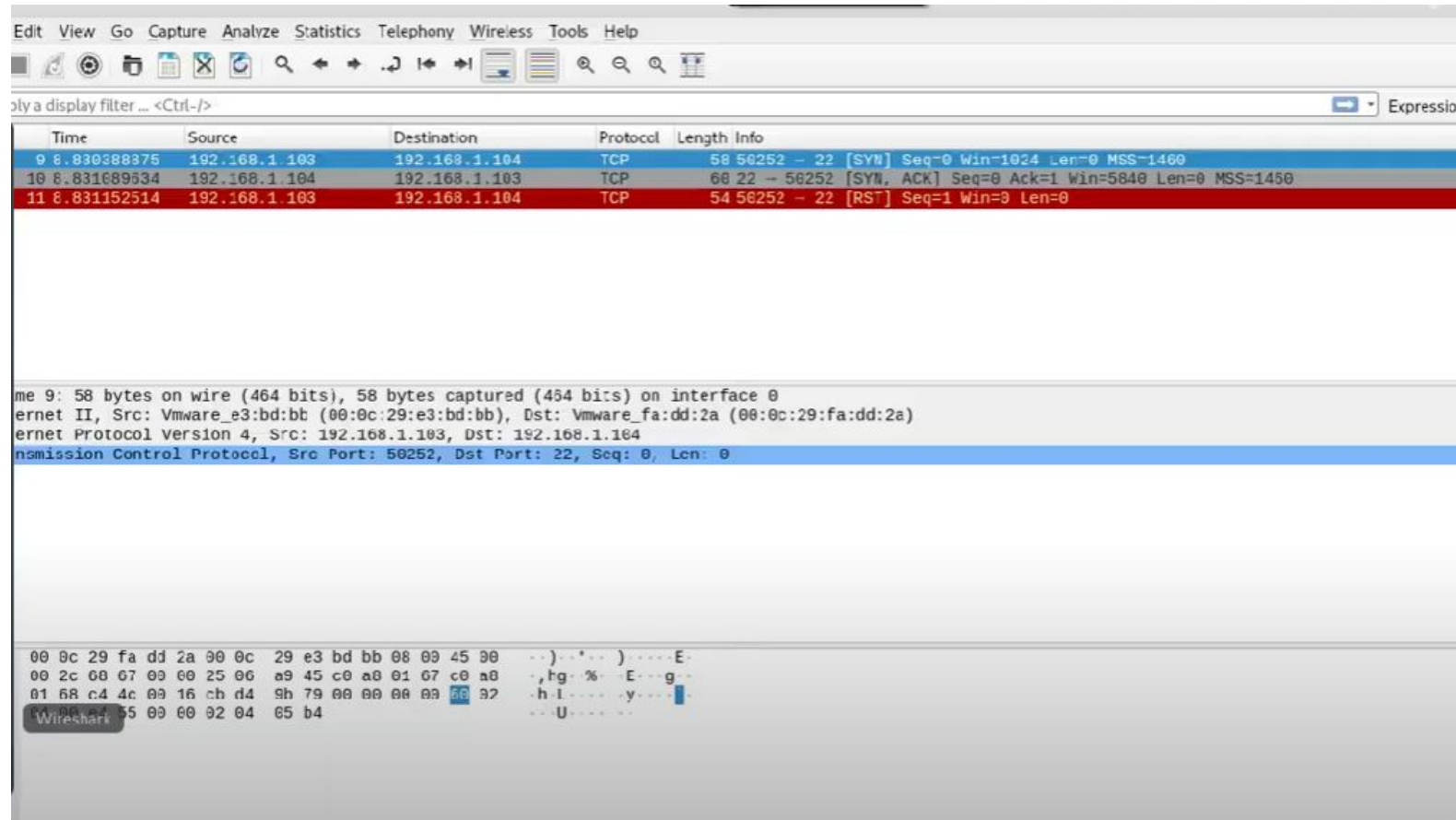
Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0  
Ethernet II, Src: Hewlett\_91:c3:5d (1c:c1:de:91:c3:5d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
Internet Protocol Version 4, Src: 192.168.0.100 (192.168.0.100), Dst: 192.168.0.255 (192.168.0.255)  
User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)  
NetBIOS Name Service

# Port Scan

- Packet sniffer is a basic tool for observing network packet exchanges in a computer. As the name suggests, a packet sniffer captures (“sniffs”) packets being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured packets.
- A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself.
- The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer must “understand” the structure of all messages exchanged by protocols.



# Port Scan





# Port Scan

*eth0						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
ip.addr == 192.168.1.10						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.090600900	192.168.1.103	14.49.100.37	TCP	78	54460 → 80 [ACK] Seq=1 Ack=1 Win=5093 Len=0 TSval=4191317101 TSecr=474
2	0.096283568	14.49.100.37	192.168.1.103	TCP	1514	80 → 54460 [ACK] Seq=7241 Ack=1 Win=59 Len=1448 TSval=47463029 TSecr=4
3	0.098201963	192.168.1.103	14.49.100.37	TCP	78	[TCP Dup ACK 1#1] 54460 → 80 [ACK] Seq=1 Ack=1 Win=5093 Len=0 TSval=41
4	0.022617929	14.49.100.37	192.168.1.103	TCP	1514	80 → 54460 [ACK] Seq=8589 Ack=1 Win=59 Len=1448 TSval=47463029 TSecr=4
5	0.023104320	192.168.1.103	14.49.100.37	TCP	78	[TCP Dup ACK 1#2] 54460 → 80 [ACK] Seq=1 Ack=1 Win=5093 Len=0 TSval=41
6	0.024283506	14.49.100.37	192.168.1.103	TCP	1514	[TCP Fast Retransmission] 80 → 54460 [ACK] Seq=1 Ack=1 Win=59 Len=1448
7	0.024318499	192.168.1.103	14.49.100.37	TCP	66	54460 → 80 [ACK] Seq=1 Ack=10137 Win=5048 Len=0 TSval=4191317126 TSecr
8	0.028218552	14.49.100.37	192.168.1.103	TCP	1514	80 → 54460 [ACK] Seq=19137 Ack=1 Win=59 Len=1448 TSval=47463932 TSecr=
9	0.030721554	14.49.100.37	192.168.1.103	TCP	1514	80 → 54460 [ACK] Seq=11585 Ack=1 Win=59 Len=1448 TSval=47463932 TSecr=
10	0.030747966	192.168.1.103	14.49.100.37	TCP	66	54460 → 80 [ACK] Seq=1 Ack=13633 Win=5093 Len=0 TSval=4191317133 TSecr
11	0.031281900	14.49.100.37	192.168.1.103	TCP	1514	80 → 54460 [ACK] Seq=13033 Ack=1 Win=59 Len=1448 TSval=47463932 TSecr=
12	0.036805563	14.49.100.37	192.168.1.103	TCP	1514	80 → 54460 [ACK] Seq=14481 Ack=1 Win=59 Len=1448 TSval=47463933 TSecr=
▶ Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0						
▶ Ethernet II, Src: Vmware_e3:bd:bb (00:0c:29:e3:bd:bb), Dst: Realtek_82:56:a0 (00:1e:a5:56:a0)						
▶ Internet Protocol Version 4, Src: 192.168.1.103, Dst: 14.49.100.37						
▶ Transmission Control Protocol, Src Port: 54460, Dst Port: 80, Seq: 1, Ack: 1, Len: 0						
0000	00 1e a6 4b 55 a0 00 0c	29 e3 bd bb 08 03 45 90	..KV.. )....E			
0010	00 40 ae b7 49 00 40 06	57 9b c0 a0 01 07 0e 31	..@..@..W...g 1			
0020	64 25 d4 bc 00 50 5d 5c	df cd 51 07 04 67 b0 10	dx...P]\..Q..g			
0030	13 e5 34 98 00 00 01 01	08 0a f9 d2 6c 6d 02 d4	..4.....lm..			
0040	3a 72 01 01 05 0a 51 07	0a 0f 51 07 20 af	:r....Q..Q..			

# Traffic Analyser

- Traffic analyzers are tools that we use to analyze the network traffic coming in and out of a specific host computer.
- Sniffer grabs all this information, and then the sniffer's going to do one of two things. It's either going to save it into a file or it's going to make a live feed directly into the traffic analyzer. The traffic analyzer really just reads pcap data and then — here's where the term comes from — analyzes it in a way that we can look at it.

# Protocol Analyser

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.254	255.255.255.255	UDP	215	32822 → 7423 Len=173
2	2.967285	192.168.0.254	255.255.255.255	UDP	215	32822 → 7423 Len=173
3	5.430885	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.32? Tell 192.168.0.1
4	5.430886	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.33? Tell 192.168.0.1
5	5.430886	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.34? Tell 192.168.0.1
6	5.430886	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.35? Tell 192.168.0.1
7	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.36? Tell 192.168.0.1
8	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.37? Tell 192.168.0.1
9	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.38? Tell 192.168.0.1
10	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.39? Tell 192.168.0.1
11	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.40? Tell 192.168.0.1
12	5.430887	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.41? Tell 192.168.0.1
13	5.430888	50:6a:03:8a:e9:03	ff:ff:ff:ff:ff:ff	ARP	60	Who has 192.168.0.42? Tell 192.168.0.1

> Frame 1: 215 bytes on wire (1720 bits), 215 bytes captured (1720 bits) on interface 0  
 > Ethernet II, Src: 50:6a:03:8a:e9:04, Dst: ff:ff:ff:ff:ff:ff  
 > Internet Protocol Version 4, Src: 192.168.0.254, Dst: 255.255.255.255  
 > User Datagram Protocol, Src Port: 32822 (32822), Dst Port: 7423 (7423)  
 > Data (173 bytes)

```

0000  ff ff ff ff ff ff 50 6a 03 8a e9 04 08 00 45 00  .....Pj .....E.
0010  00 c9 00 00 40 00 40 11 78 7e c0 a8 00 fe ff ff  ....@.@. x~.....
0020  ff ff 80 36 1c ff 00 b5 b4 96 4b 41 4e 4e 4f 55  ...6.... .KANNOU
0030  25 4e 00 00 00 00 00 50 6a 03 8a e9 04 43 47 33  %N....P j....CG3
  
```

# Protocol Analyser

*Wireless Network Connection						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
149	42.702265	192.168.0.1	239.255.255.250	SSDP	381	NOTIFY * HTTP/1.1
150	42.702266	192.168.0.1	239.255.255.250	SSDP	326	NOTIFY * HTTP/1.1
151	42.702266	192.168.0.1	239.255.255.250	SSDP	317	NOTIFY * HTTP/1.1
152	42.702266	192.168.0.1	239.255.255.250	SSDP	391	NOTIFY * HTTP/1.1
153	44.518629	192.168.0.7	68.105.28.11	DNS	79	Standard query 0xe934 A www.astonmartin.com
154	44.545736	192.168.0.7	68.105.28.11	DNS	79	Standard query 0xe934 A www.astonmartin.com
155	44.554682	68.105.28.11	192.168.0.7	DNS	95	Standard query response 0xe934 A www.astonmartin.com A 213.199.132.134
156	44.555390	192.168.0.7	213.199.132.134	TCP	66	51605 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=256 SACK_PERM=1
157	44.555680	192.168.0.7	68.105.28.11	DNS	79	Standard query 0xf5d4 A www.astonmartin.com
158	44.560447	68.105.28.11	192.168.0.7	DNS	95	Standard query response 0xe934 A www.astonmartin.com A 213.199.132.134
159	44.560484	192.168.0.7	68.105.28.11	ICMP	123	Destination unreachable (Port unreachable)
160	44.568163	68.105.28.11	192.168.0.7	DNS	95	Standard query response 0xf5d4 A www.astonmartin.com A 213.199.132.134
161	44.568613	192.168.0.7	68.105.28.11	DNS	79	Standard query 0xf5d4 A www.astonmartin.com
> Frame 156: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0 > Ethernet II, Src: cc:3d:82:1e:51:4f, Dst: 50:6a:03:8a:e9:03 > Internet Protocol Version 4, Src: 192.168.0.7, Dst: 213.199.132.134 > Transmission Control Protocol, Src Port: 51605 (51605), Dst Port: 80 (80), Seq: 0, Len: 0						
0000	50 6a 03 8a e9 03 cc 3d 82 1e 51 4f 08 00 45 00	Pj.....=..QO..E.				
0010	00 34 71 86 40 00 80 06 6e 40 c0 a8 00 07 d5 c7	.4q.@... n@.....				
0020	84 86 c9 95 00 50 dd e1 1e ed 00 00 00 00 00 02	.....P.. .....				
0030	20 00 6d 5c 00 00 02 04 05 b4 01 03 03 00 01 01	..m^..... .....				
0040	04 07	..				



# Protocol Analyser

Wireless Network Connection

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1266	48.440859	72.21.81.200	192.168.0.7	TCP	50	[TCP Window Update] 80 → 51631 [ACK] Seq=1 Ack=342 Win=131070 Len=0
1267	48.440893	192.168.0.7	64.185.181.238	TCP	54	51620 → 80 [ACK] Seq=1028 Ack=36681 Win=65536 Len=0
1268	48.440918	192.168.0.7	64.185.181.238	HTTP	397	GET /sitefinity/Navigation/amworldgaydon.jpg HTTP/1.1
1269	48.440974	192.168.0.7	64.185.181.238	HTTP	396	GET /sitefinity/Navigation/amworldmedia.jpg HTTP/1.1
1270	48.441085	192.168.0.7	64.185.181.238	HTTP	394	GET /sitefinity/Navigation/qnavbadge2.png HTTP/1.1
1271	48.441816	72.21.81.200	192.168.0.7	TCP	1514	[TCP segment of a reassembled PDU]
1272	48.441817	72.21.81.200	192.168.0.7	TCP	1514	[TCP segment of a reassembled PDU]
1273	48.441849	192.168.0.7	72.21.81.200	TCP	54	51631 → 80 [ACK] Seq=342 Ack=2921 Win=65536 Len=0
1274	48.443386	72.21.81.200	192.168.0.7	TCP	1514	[TCP segment of a reassembled PDU]
1275	48.443387	72.21.81.200	192.168.0.7	TCP	1514	[TCP segment of a reassembled PDU]
1276	48.443441	192.168.0.7	72.21.81.200	TCP	54	51631 → 80 [ACK] Seq=342 Ack=5841 Win=65536 Len=0
1277	48.444535	72.21.81.200	192.168.0.7	TCP	1514	[TCP segment of a reassembled PDU]
1278	48.444536	72.21.81.200	192.168.0.7	TCP	1514	[TCP segment of a reassembled PDU]

> Frame 1277: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0

> Ethernet II, Src: 50:6a:03:8a:e9:03, Dst: cc:3d:82:1e:51:4f

> Internet Protocol Version 4, Src: 72.21.81.200, Dst: 192.168.0.7

> Transmission Control Protocol, Src Port: 80 (80), Dst Port: 51631 (51631), Seq: 5841, Ack: 342, Len: 1460

```

0000  cc 3d 82 1e 51 4f 50 6a 03 8a e9 03 08 00 45 00  ..QOPj .....E.
0010  05 dc a8 da 40 00 3a 06 37 b5 48 15 51 c8 c0 a8  ....@.:. 7.H.Q...
0020  00 07 00 50 c9 af c7 1b ee ad 5b 5b 85 42 50 10  ...P.... ..[.BP.
0030  ff ff d6 c2 00 00 e2 d2 70 f0 a0 88 c0 bc 9c 28  ....p.....(
0040  2f d6 0e af 1a 82 b6 26 4c 67 85 02 9b 86 0c 9a  /.....& lg.....
0050  04 f5 41 11 c0 65 41 53 5e cf 35 e1 06 8a c5 bd  ..A..eAS ^.5.....
0060  ba 45 c0 9f 8d 06 0d fd e8 9e 96 91 c0 d1 1a eb  .E.....
0070  fb f0 09 87 00 93 84 ff 00 0a 0a 50 6f 37 ae cb  .......Po7...
0080  b6 b5 ea 90 47 5f ca 07 91 34 1b 56 9f b6 10 59  ....G...4.1...9
  
```

# Conclusion

We have successfully carried out a port scan, analysed traffic on a network and sniffed packets using Wireshark.