

## Lab 8 - Network Forensics

### Question 1

What service appears to be running on port 2200?

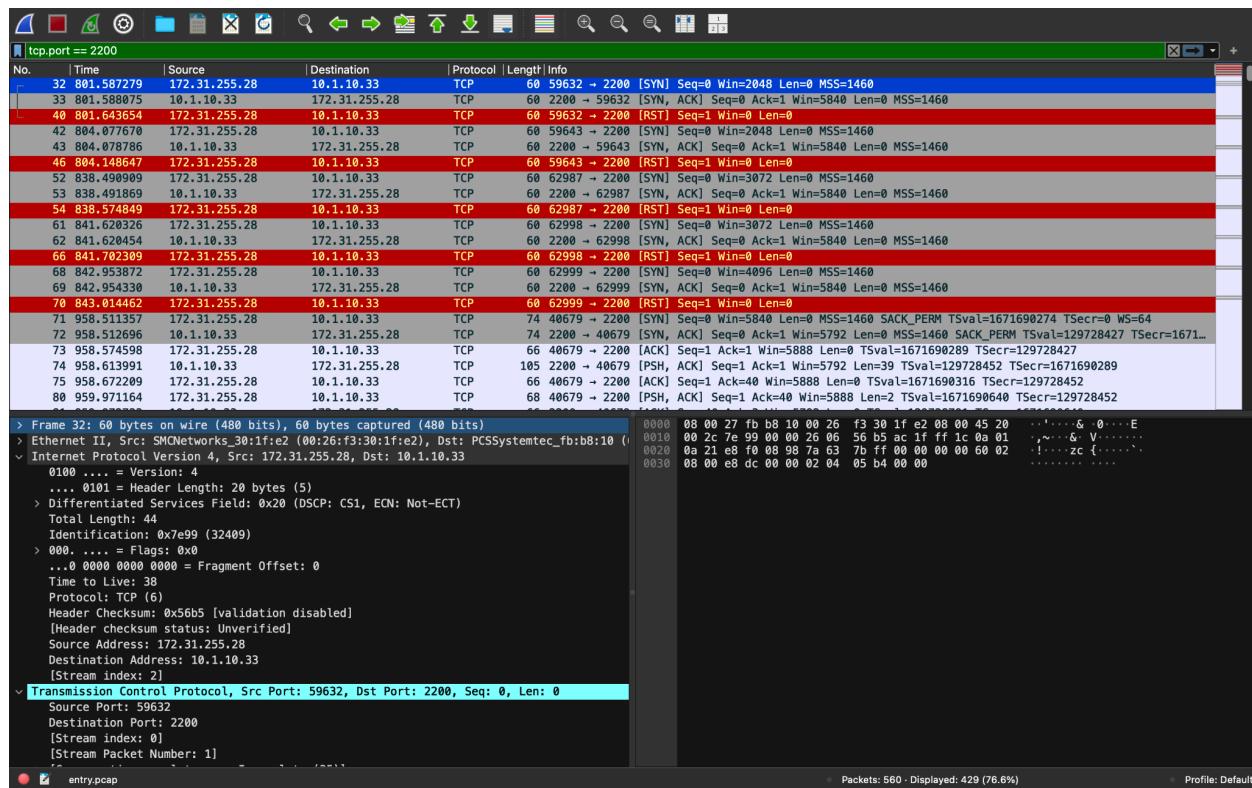
Choose one answer.

Industrial Control Interface

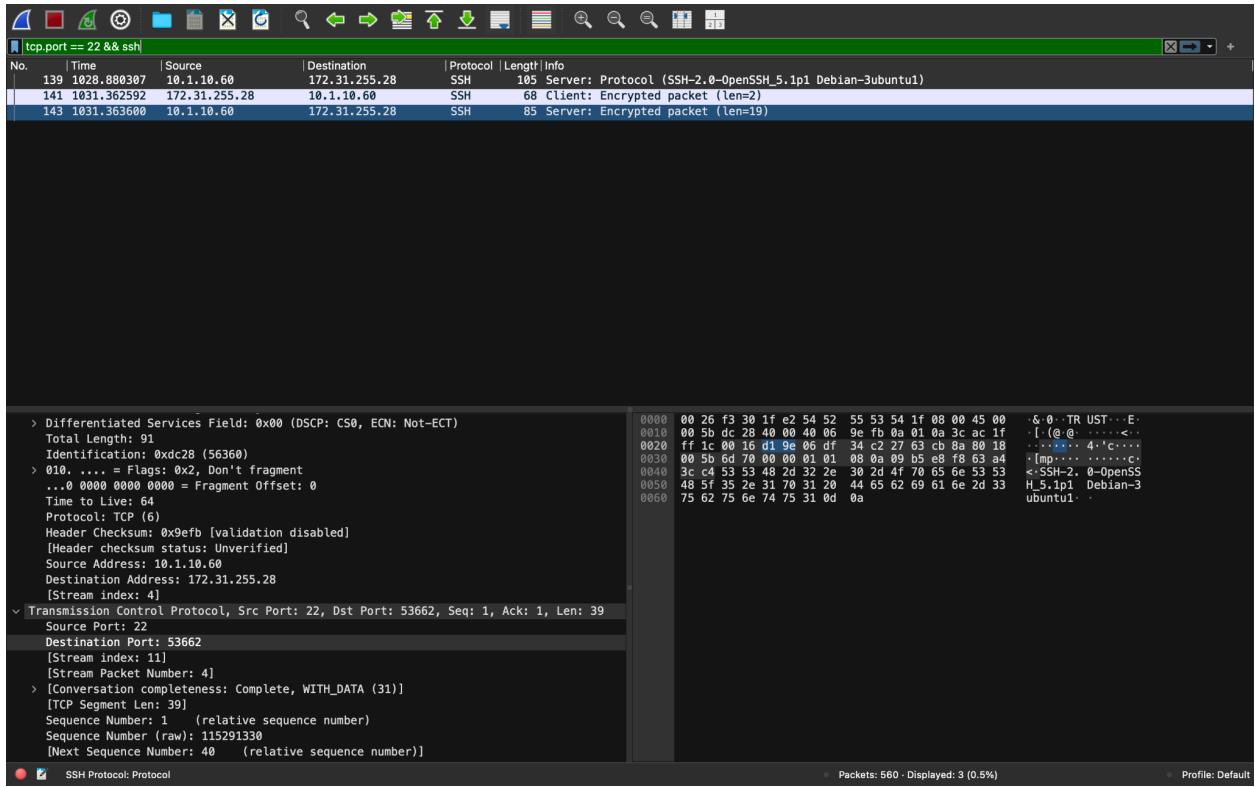
Rockwell Automation PPTP

Inter Carrier Interface

**Secure Shell**



I checked port 2200 to see what protocol was running on it using the filter 'tcp.port == 2200' to see what packets were running on that particular port. Typically SSH uses TCP as a transport protocol with a well known port for SSH traffic being on 22. It's possible that it could not be though and filtering by ssh reveals nothing but so does every other filter protocol.



When checking SSH protocols on port 22 you can see the destination of the request is 172.31.255.28 which is the same source as the TCP protocols on port 2200. All of these SSH protocols link directly into the TCP protocols. I can assume they are reliant on SSH.

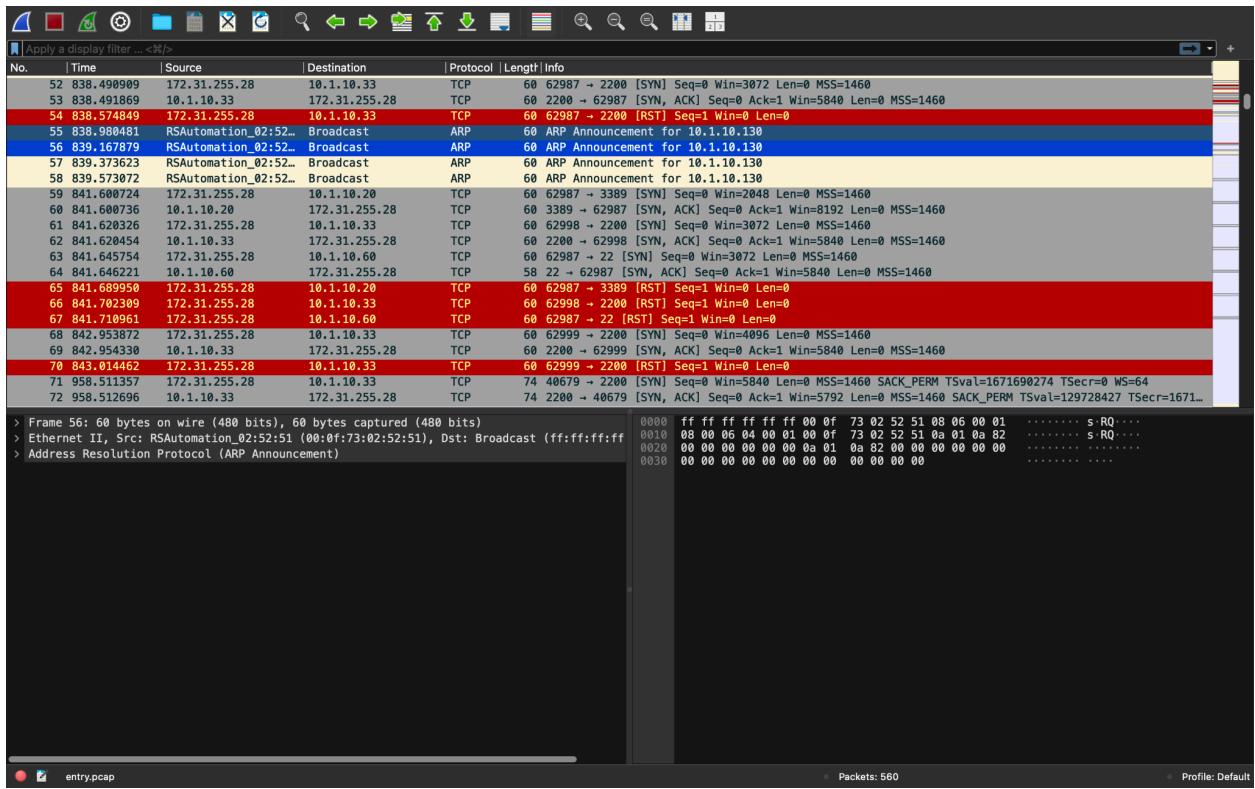
## Question 2

It appears that after running a scan, the attackers made connections to each of the open ports. Which of the following tools was most likely used to establish those connections?

Choose one answer.

- Nessus
- SSH
- Telnet
- Netcat**

My answer is going to make a lot of conjectures so just bear with me. Looking through the packets in entry.pcap you see a lot of TCP checks. This is indicative of the attackers searching for a network vulnerability. [Network Port Scanners](#) The SYN scanner in Nessus has a similar signature to what we see in the Wireshark file. You can see multiple SYN packets being sent to determine if the ports are open. If they are, the attackers received a SYN-ACK reply. You can also see that some of these connections are sending RST reset flags meaning that the server does not recognize the packets. Essentially a denial of service. After getting these scans we can assume the attacker knows the open ports for Netcat is a good option as it is specifically made for port interactions. Especially with tools like [Reverse Shell](#) which make this process straightforward.



## Question 3

Which IP address had port 2200 open?

Choose one answer.

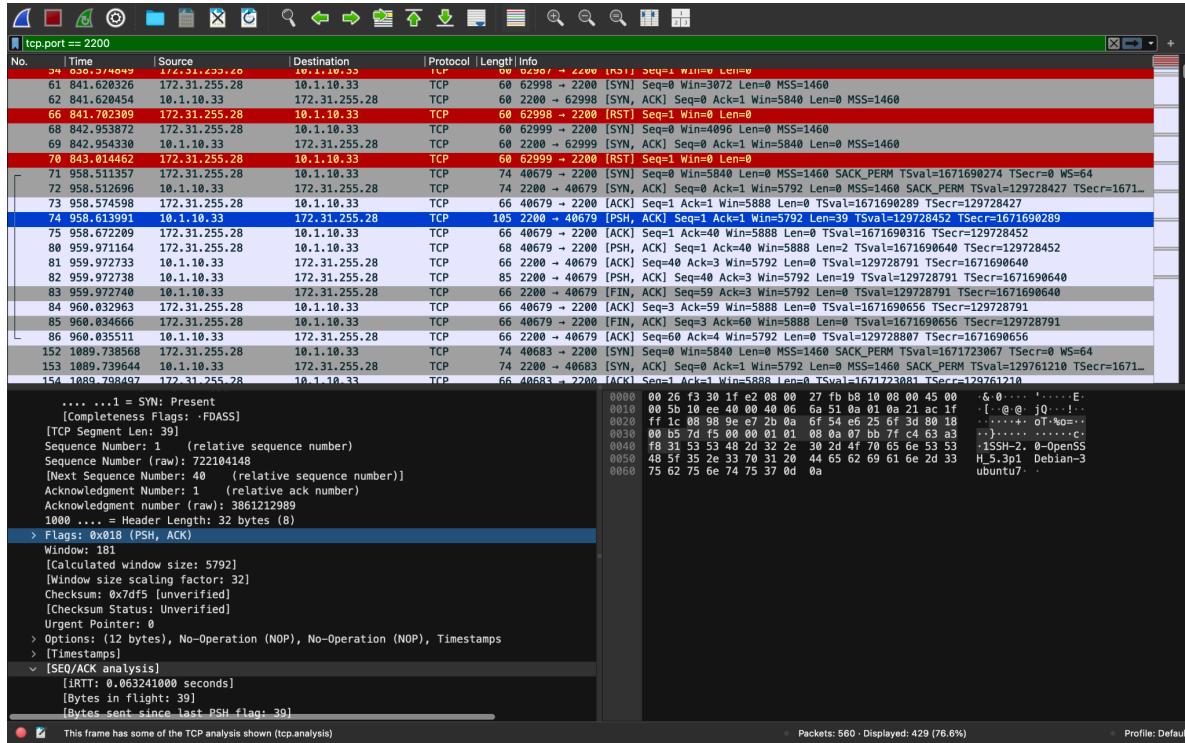
10.1.10.33

10.1.10.60

10.1.10.20

10.1.10.130

This question is pretty straightforward. We're just going to check TCP connections on port 2200. You can see any instance with port 2200 open has a source IP of 10.1.10.33. Thus 10.1.10.33 has port 2200 open.



## Question 4

Which version of SSH was the attacker using?

Choose one answer. **OpenSSH 5.1p1**

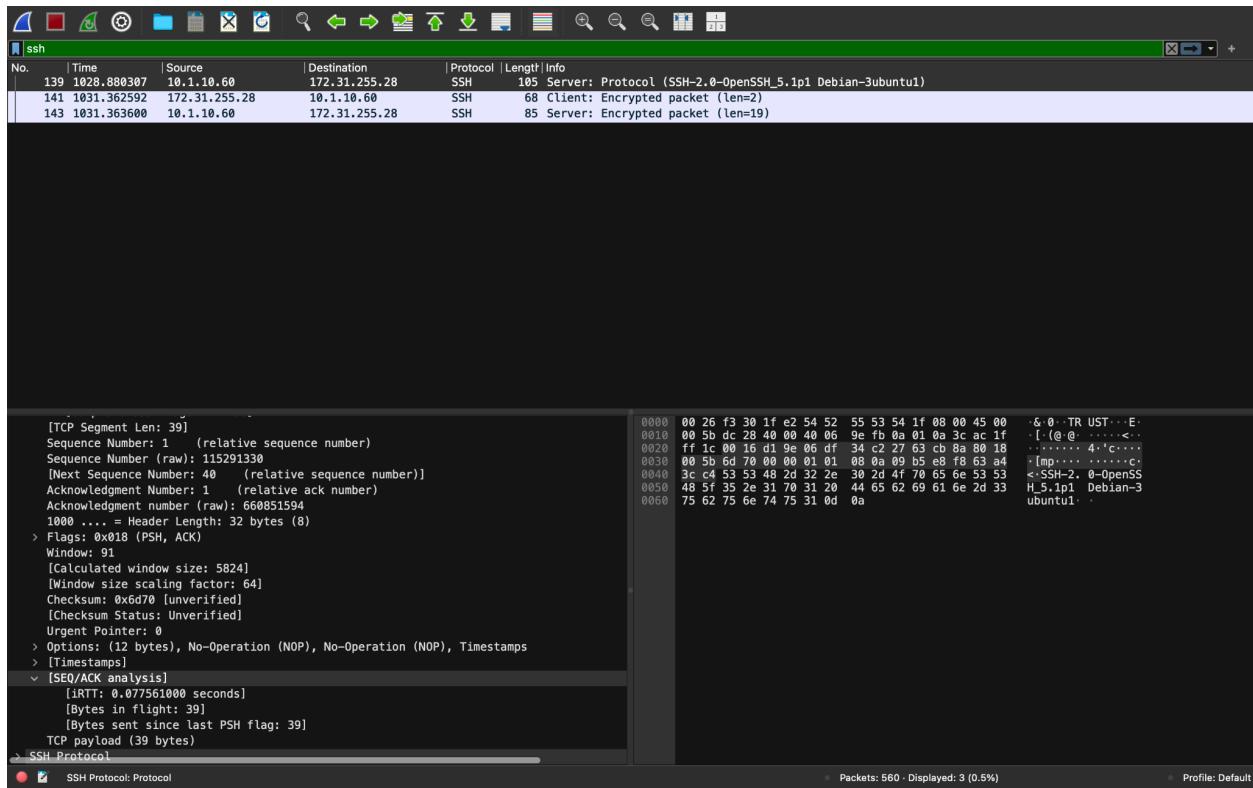
WinSSH

PuTTY

OpenSSH 5.3p1

OpenSSH 5.2

We can determine what version of SSH the attacker was using by checking the ssh filter. You can see that on packet 139 the attacker is using OpenSSH 5.1p1.



## Question 5

Which of the following IP addresses appear to be the same type of device?

Choose one answer.

- 10.1.10.13 and 10.1.10.29  
10.1.10.20 and 10.1.10.29  
**10.1.10.15 and 10.1.10.13**  
10.1.10.20 and 10.1.10.130

We can determine whether or not the IP Addresses are on the same kind of device based on the destination. In this case we can see both 10.1.10.15 and 10.1.10.13 are running on a GrandstreamN device.

The screenshot shows a Wireshark interface with the following details:

- Frame 135:** 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
- Ethernet II, Src: PCSystems\_1fb:8b:10 (00:0b:27:fb:b8:10), Dst: Grandstream\_N\_21:ad:07 (00:0b:82:21:ad:07)**
- Internet Protocol Version 4, Src: 10.1.10.33, Dst: 10.1.10.13**
- TCP: 100... = Version: 4  
.... 0101 = Header Length: 20 bytes (5)  
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
Total Length: 52  
Identification: 0x6556 (1622)  
> 010... = Flags: 0x2, Don't fragment  
0x0000 0x0000 0x0000 = Fragment Offset: 0  
Time to Live: 64  
Protocol: TCP (6)  
Header Checksum: 0x0c3f [validation disabled]  
[Header checksum status: Unverified]  
Source Address: 10.1.10.33  
Destination Address: 10.1.10.13  
[Stream Index: 2]  
[Stream Packet Number: 4]**
- Transmission Control Protocol, Src Port: 40456, Dst Port: 80, Seq: 1, Ack: 1, Len: 0**
- Source Port: 40456  
Destination Port: 80  
[Stream Index: 2]  
[Stream Packet Number: 4]**
- Packets: 1984 - Displayed: 34 (1.7%)**
- Profile: Default**

## Question 6

How many IP addresses had port 3389 open?

Choose one answer. 9

1

2

3

4

We can determine the IP addresses associated with port 3389 by doing a filter for ‘tcp.port == 3389’. After that we can go to Statistics > Endpoints. We get this result.

The screenshot shows the NetworkMiner interface with the 'TCP - 23' tab selected. The main pane displays a table of network traffic. The columns include Address, Port, Packets, Bytes, Total Packets, Percent Filtered, Tx Packets, Rx Bytes, and Rx Bytes. The data shows multiple entries for port 3389 from various IP addresses. The left sidebar shows 'Endpoint Settings' with 'Limit to display filter' checked, and a protocol list where 'Ethernet' and 'IPv6' are selected. Buttons for 'Copy' and 'Map' are also visible.

Endpoint Settings								
Address	Port	Packets	Bytes	Total Packets	Percent Filtered	Tx Packets	Rx Bytes	Rx Bytes
10.1.10.1	3389	2	134 bytes	2	100.00%	1	60 bytes	1
10.1.10.10	3389	2	134 bytes	2	100.00%	1	60 bytes	1
10.1.10.12	3389	2	134 bytes	2	100.00%	1	60 bytes	1
10.1.10.13	3389	2	134 bytes	2	100.00%	1	60 bytes	1
10.1.10.15	3389	2	134 bytes	2	100.00%	1	60 bytes	1
10.1.10.16	3389	2	134 bytes	2	100.00%	1	60 bytes	1
10.1.10.20	3389	4	280 bytes	4	100.00%	1	74 bytes	3
10.1.10.27	3389	2	134 bytes	2	100.00%	1	60 bytes	1
10.1.10.29	3389	2	134 bytes	2	100.00%	1	60 bytes	1
10.1.10.33	33521	2	134 bytes	2	100.00%	1	74 bytes	1
10.1.10.33	33891	2	134 bytes	2	100.00%	1	74 bytes	1
10.1.10.33	37832	2	134 bytes	2	100.00%	1	74 bytes	1
10.1.10.33	40709	1	74 bytes	1	100.00%	1	74 bytes	0
10.1.10.33	40731	1	74 bytes	1	100.00%	1	74 bytes	0
10.1.10.33	43177	2	134 bytes	2	100.00%	1	74 bytes	1
10.1.10.33	46009	4	280 bytes	4	100.00%	3	206 bytes	1
10.1.10.33	47139	2	128 bytes	2	100.00%	1	74 bytes	1
10.1.10.33	49390	2	134 bytes	2	100.00%	1	74 bytes	1
10.1.10.33	52225	2	134 bytes	2	100.00%	1	74 bytes	1
10.1.10.33	53526	2	134 bytes	2	100.00%	1	74 bytes	1
10.1.10.33	57181	2	134 bytes	2	100.00%	1	74 bytes	1
10.1.10.60	3389	2	128 bytes	2	100.00%	1	54 bytes	1
10.1.10.130	3389	2	148 bytes	2	100.00%	0	0 bytes	2

You can see that IP addresses: 10.1.10.1, 10.1.10.10, 10.1.10.12, 10.1.10.13, 10.1.10.15, 10.1.10.16, 10.1.10.20, 10.1.10.27, and 10.1.10.29 had port 3389 open at some point. Meaning that nine ports had port 3389 open.

## Question 8

Which of the following IP addresses did NOT have port 23 open?

Choose one answer. **None**

10.1.10.10

10.1.10.27

10.1.10.16

10.1.10.29

We can determine which of the following IP addresses did not have port 23 open in the same way we solved question six. We'll do a filter for 'tcp.port == 23'. Then we'll go to Statistics > Endpoints. Ultimately all of these IP addresses had port 23 open at some point.

Endpoint Settings									
		Ethernet - 12	IPv4 - 12	IPv6	TCP - 24	UDP			
Address	Port	Packets	Bytes	Total Packets	Percent Filtered	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
10.1.10.1	23	4	280 bytes	4	100.00%	1	74 bytes	3	206 bytes
10.1.10.10	23	2	134 bytes	2	100.00%	1	60 bytes	1	74 bytes
10.1.10.12	23	2	134 bytes	2	100.00%	1	60 bytes	1	74 bytes
10.1.10.13	23	4	280 bytes	4	100.00%	1	74 bytes	3	206 bytes
10.1.10.15	23	4	280 bytes	4	100.00%	1	74 bytes	3	206 bytes
10.1.10.16	23	6	421 bytes	6	100.00%	2	155 bytes	4	266 bytes
10.1.10.20	23	2	134 bytes	2	100.00%	1	60 bytes	1	74 bytes
10.1.10.27	23	4	280 bytes	4	100.00%	1	74 bytes	3	206 bytes
10.1.10.29	23	46	3 kB	46	100.00%	22	2 kB	24	2 kB
10.1.10.33	32827	4	280 bytes	4	100.00%	3	206 bytes	1	74 bytes
10.1.10.33	38933	1	74 bytes	1	100.00%	1	74 bytes	0	0 bytes
10.1.10.33	38935	1	74 bytes	1	100.00%	1	74 bytes	0	0 bytes
10.1.10.33	38983	2	134 bytes	2	100.00%	1	74 bytes	1	60 bytes
10.1.10.33	40864	4	280 bytes	4	100.00%	3	206 bytes	1	74 bytes
10.1.10.33	41987	2	134 bytes	2	100.00%	1	74 bytes	1	60 bytes
10.1.10.33	42013	2	128 bytes	2	100.00%	1	74 bytes	1	54 bytes
10.1.10.33	46469	2	134 bytes	2	100.00%	1	74 bytes	1	60 bytes
10.1.10.33	46799	4	280 bytes	4	100.00%	3	206 bytes	1	74 bytes
10.1.10.33	46823	42	3 kB	42	100.00%	21	1 kB	21	2 kB
10.1.10.33	55111	6	421 bytes	6	100.00%	4	266 bytes	2	155 bytes
10.1.10.33	57356	4	280 bytes	4	100.00%	3	206 bytes	1	74 bytes
10.1.10.33	60037	4	280 bytes	4	100.00%	3	206 bytes	1	74 bytes
10.1.10.60	23	2	128 bytes	2	100.00%	1	54 bytes	1	74 bytes
10.1.10.130	23	2	148 bytes	2	100.00%	0	0 bytes	2	148 bytes

Filter list for specific type

Help

Close

## Question 9

Which of the following ports was not included in the scan of the internal network?

Choose one answer.

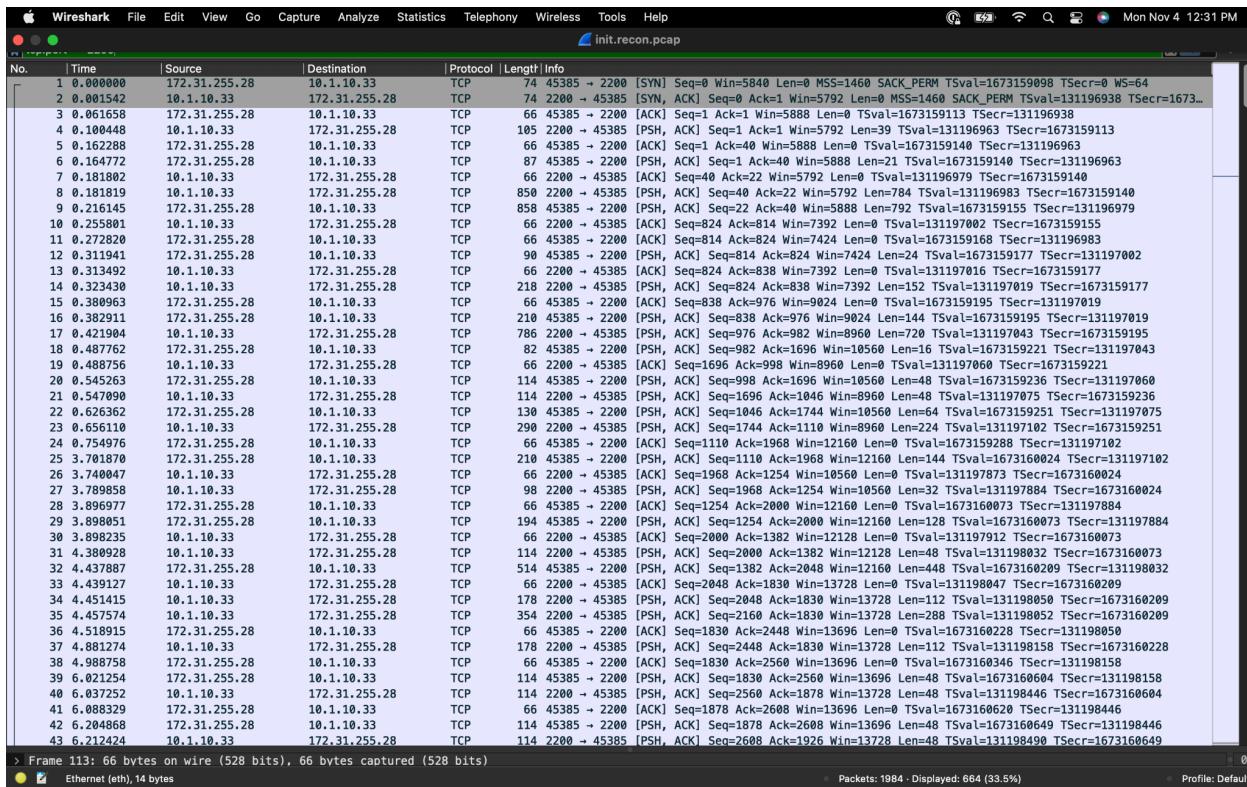
TCP 3389

TCP 80

TCP 23

**TCP 2200**

A common motif in this .pcap file is that any ports that have been scanned on the internal network have a packet with a RST, ACK modifier. TCP 2200 does not have any RST, ACK packets. This means that the port wasn't interacted with internally.



## Question 10

Approximately how long did the port scan take to complete?

Choose one answer.

27.1 seconds

4.0 seconds

3.5 seconds

17.1 seconds

We can see how long the port scans took to complete by seeing the time the TCP packets end and the SSDP protocols begin. In this case the packet at 1111 indicates the end of the scan at 27.134 seconds.

No.	Time	Source	Destination	Protocol	Length	Info
1074	25.801661	10.1.10.33	10.1.10.130	TCP	60	52491 → 80 [RST, ACK] Seq=1 Ack=1 Win=5840 Len=0
1075	25.801664	10.1.10.15	10.1.10.33	TCP	74	23 → 60037 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM Tsvl=1188955527 Tsec=13120...
1076	25.801667	10.1.10.13	10.1.10.33	TCP	74	23 → 32827 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM Tsvl=1810945899 Tsec=13120...
1077	25.801669	10.1.10.33	10.1.10.16	TCP	66	55111 → 23 [ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=963631296
1078	25.801908	10.1.10.33	10.1.10.29	TCP	66	46799 → 23 [ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=3733426
1079	25.801911	10.1.10.33	10.1.10.27	TCP	66	57356 → 23 [ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=117781539
1080	25.801914	10.1.10.33	10.1.10.15	TCP	66	60037 → 23 [ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=1188955527
1081	25.801917	10.1.10.33	10.1.10.13	TCP	66	32827 → 23 [ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=1810945899
1082	25.801919	10.1.10.33	10.1.10.60	TCP	66	56140 → 22 [ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=164666666
1083	25.802239	10.1.10.33	10.1.10.130	TCP	74	38933 → 23 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM Tsvl=131203384 Tsec=0 WS=32
1084	25.804641	10.1.10.33	10.1.10.1	TCP	74	40864 → 23 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM Tsvl=131203384 Tsec=0 WS=32
1085	25.804660	10.1.10.33	10.1.10.13	TCP	66	32827 → 23 [RST, ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=1810945899
1086	25.804675	10.1.10.33	10.1.10.15	TCP	66	60037 → 23 [RST, ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=188955527
1087	25.804691	10.1.10.33	10.1.10.16	TCP	66	55111 → 23 [RST, ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=963631296
1088	25.804707	10.1.10.33	10.1.10.27	TCP	66	57356 → 23 [RST, ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=117781539
1089	25.804723	10.1.10.33	10.1.10.29	TCP	66	46799 → 23 [RST, ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=3733426
1090	25.804739	10.1.10.33	10.1.10.60	TCP	66	56140 → 22 [RST, ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203384 Tsec=164666666
1091	25.819458	10.1.10.1	10.1.10.33	TCP	74	23 → 40864 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM Tsvl=186591961 Tsec=131203...
1092	25.819482	10.1.10.16	10.1.10.33	TELNET	81	No Echo, Do Negotiate About Window Size, Do Remote Flow Control, Will Echo, Will Suppress Go Ahe...
1093	25.819484	10.1.10.12	10.1.10.33	TCP	60	22 → 34318 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1094	25.819486	10.1.10.12	10.1.10.33	TCP	60	23 → 41987 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1095	25.819768	10.1.10.33	10.1.10.1	TCP	66	40864 → 23 [ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203389 Tsec=186591961
1096	25.819785	10.1.10.33	10.1.10.16	TCP	60	55111 → 23 [RST] Seq=1 Win=0 Len=0
1097	25.819942	10.1.10.33	10.1.10.1	TCP	66	40864 → 23 [ACK] Seq=1 Ack=1 Win=5856 Len=0 Tsvl=131203389 Tsec=186591961
1098	26.889758	10.1.10.33	10.1.10.130	TCP	74	38935 → 23 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM Tsvl=131203656 Tsec=0 WS=32
1099	26.891238	10.1.10.33	10.1.10.130	TCP	74	47795 → 22 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM Tsvl=131203656 Tsec=0 WS=32
1100	26.891244	10.1.10.33	10.1.10.130	TCP	74	40731 → 3389 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM Tsvl=131203656 Tsec=0 WS=32
1101	26.891247	10.1.10.33	10.1.10.130	TCP	74	55654 → 21 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM Tsvl=131203656 Tsec=0 WS=32
1102	26.891250	10.1.10.33	10.1.10.130	TCP	74	41871 → 443 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM Tsvl=131203656 Tsec=0 WS=32
1103	26.901891	10.1.10.33	10.1.10.1	TCP	74	49524 → 22 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM Tsvl=131203659 Tsec=0 WS=32
1104	27.015628	10.1.10.33	172.31.255.28	TCP	1514	2200 → 45385 [ACK] Seq=160 Ack=2918 Win=13728 Len=1448 Tsvl=131203687 Tsec=1673164818
1105	27.015646	10.1.10.33	172.31.255.28	TCP	730	2200 → 45385 [PSH, ACK] Seq=5608 Ack=2918 Win=13728 Len=664 Tsvl=131203687 Tsec=1673164818
1106	27.019253	10.1.10.33	172.31.255.28	TCP	562	2200 → 45385 [PSH, ACK] Seq=6272 Ack=2918 Win=13728 Len=496 Tsvl=131203688 Tsec=1673164818
1107	27.071961	172.31.255.28	10.1.10.33	TCP	66	45385 → 2200 [ACK] Seq=2918 Ack=5608 Win=18176 Len=0 Tsvl=1673165861 Tsec=131203687
1108	27.072107	10.1.10.33	172.31.255.28	TCP	178	2200 → 45385 [PSH, ACK] Seq=6768 Ack=2918 Win=13728 Len=112 Tsvl=131203701 Tsec=1673165863
1109	27.073479	172.31.255.28	10.1.10.33	TCP	66	45385 → 2200 [ACK] Seq=2918 Ack=6272 Win=21056 Len=0 Tsvl=1673165863 Tsec=131203687
1110	27.077515	172.31.255.28	10.1.10.33	TCP	66	45385 → 2200 [ACK] Seq=2918 Ack=6768 Win=23936 Len=0 Tsvl=1673165865 Tsec=131203688
1111	27.143727	172.31.255.28	10.1.10.33	TCP	66	45385 → 2200 [ACK] Seq=2918 Ack=6880 Win=23936 Len=0 Tsvl=1673165881 Tsec=131203701
1112	43.185266	10.1.10.1	239.255.255.250	SSDP	307	NOTIFY * HTTP/1.1
1113	43.186609	10.1.10.1	239.255.255.250	SSDP	379	NOTIFY * HTTP/1.1
1114	43.186620	10.1.10.1	239.255.255.250	SSDP	375	NOTIFY * HTTP/1.1
1115	43.186625	10.1.10.1	239.255.255.250	SSDP	355	NOTIFY * HTTP/1.1

Ethernet II, Src: SMNetworks 30:1f:e2 (00:26:f3:30:1f:e2), Dst: PCSSystemtec fb:b8:10 (00:00:27:fb:b8:10) Packets: 1984 Profile: Default

## Question 17

What web server appears to be running on the .130 device?

Choose one answer.

Apache 2.2.19

A-B WWW/0.1

Firefox/3.6.24

1763-L16BWA B/9.00

By using HTTP stream on .130 we can determine what webserver is running on the .130 device.

We'll use the filter "http.request.method == "GET"" then we'll go to Analyze > Follow > HTTP. Then we get this result: Firefox/3.6.24 for the user's web server.

```
GET /redirect.htm HTTP/1.1
Host: 10.1.10.130
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.24) Gecko/20111107 Ubuntu/10.04 (lucid) Firefox/3.6.24
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://10.1.10.130/

HTTP/1.0 200 OK
Server: A-B WWW/0.1
Expires: Thu, 01 Dec 1994 16:00:00 GMT
Content-Type: text/html
Content-Length: 137

<html><head><script type="text/javascript" language=JavaScript src="URLhdl.js"></script></head><body onload="parseQuery()"></body></html>
```

Client pkt, 2 server pkts, 1 turn.

Entire conversation (685 bytes) Show as ASCII No delta times Stream 1

Find: Case sensitive Find Next

Help Filter Out This Stream Print Save as... Back Close