

Wireshark

To record the traffic between the client and server we used the filter tcp.port == 1335

1. Login session explained:

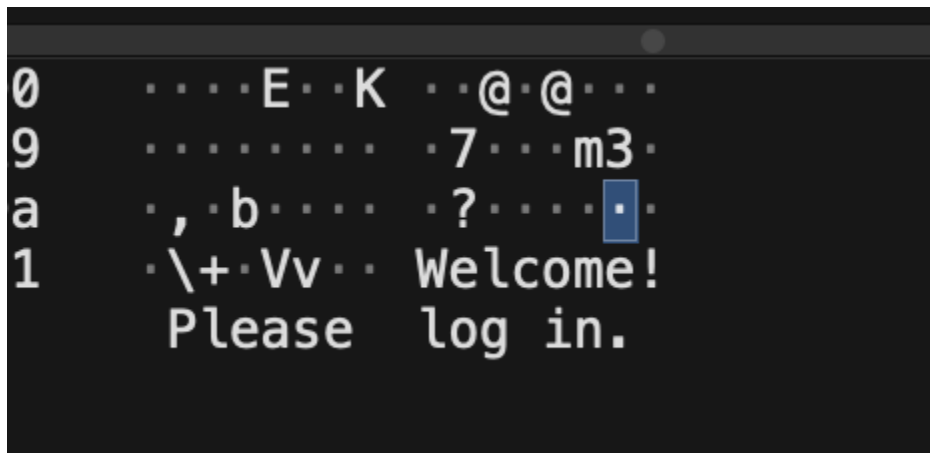
tcp.port == 1335							
No.	Time	Source	Destination	Protocol	Length	Info	
23	5.748276	127.0.0.1	127.0.0.1	TCP	68	55738 → 1335 [SYN] Seq=0 Win=65535 Len=0 MS	
24	5.748468	127.0.0.1	127.0.0.1	TCP	68	1335 → 55738 [SYN, ACK] Seq=0 Ack=1 Win=655	
25	5.748492	127.0.0.1	127.0.0.1	TCP	56	55738 → 1335 [ACK] Seq=1 Ack=1 Win=408256 L	
26	5.748506	127.0.0.1	127.0.0.1	TCP	56	[TCP Window Update] 1335 → 55738 [ACK] Seq=	
27	5.748554	127.0.0.1	127.0.0.1	TCP	79	1335 → 55738 [PSH, ACK] Seq=1 Ack=1 Win=408	
28	5.748583	127.0.0.1	127.0.0.1	TCP	56	55738 → 1335 [ACK] Seq=1 Ack=24 Win=408256	
103	26.118961	127.0.0.1	127.0.0.1	TCP	68	55738 → 1335 [PSH, ACK] Seq=1 Ack=24 Win=40	
104	26.119021	127.0.0.1	127.0.0.1	TCP	56	1335 → 55738 [ACK] Seq=24 Ack=13 Win=408256	
105	26.119077	127.0.0.1	127.0.0.1	TCP	81	1335 → 55738 [PSH, ACK] Seq=24 Ack=13 Win=4	
106	26.119099	127.0.0.1	127.0.0.1	TCP	56	55738 → 1335 [ACK] Seq=13 Ack=49 Win=408192	

Packets explained by (No. column):

23-25 - Three way handshake this is where the the connection between the client and server is established.

26 - Tcp Windows Update - Internal tcp implementation, likely server freed up buffer space and can receive info and is letting customer know.

27 - The first message from the server in the login process, server sent the Welcome message



28 - Client sent acknowledgement packet for packet 27

103 - Client sends the credentials in the format: 0 prefix indicating it is auth message and username and password separated by a comma and \ at the end as postfix, can be seen in the payload marked below.

```

02 00 00 00 45 00 00 40 00 00 40 00 40 06 00 00 .....E..@..@..@..
7f 00 00 01 7f 00 00 01 d9 ba 05 37 f0 2c 9c 62 .....7..b
85 6d 33 30 80 18 18 eb fe 34 00 00 01 01 08 0a ..m30.....4.....
56 77 0a 13 01 5c 2b f1 30 20 6e 6f 61 6d 2c 31 Vw... \+ 0 noam,1
32 33 34 5c 234\

```

104 - acknowledgement from server for packet 103

105 - Successful login message from server to client:

```

10 7f 00 00 01 7f 00 00 01 05 37 d9 ba 85 6d 33 30 .....7...m30
20 f0 2c 9c 6e 80 18 18 eb fe 41 00 00 01 01 08 0a ..,n....A.....
30 01 5c 7b 84 56 77 0a 13 48 69 20 6e 6f 61 6d 2c \{Vw.. Hi noam,
40 20 67 6f 6f 64 20 74 6f 20 73 65 65 20 79 6f 75 good to see you
50 2e

```

106 - Acknowledgement from the client to server.

2. Command - factors command:

tcp.port == 1335						
No.	Time	Source	Destination	Protocol	Length	Info
39	7.956894	127.0.0.1	127.0.0.1	TCP	64	55633 → 1335 [PSH, ACK] Seq=1 Ack=1 Win=637
40	7.956949	127.0.0.1	127.0.0.1	TCP	56	1335 → 55633 [ACK] Seq=1 Ack=9 Win=6379 Len
41	7.957041	127.0.0.1	127.0.0.1	TCP	97	1335 → 55633 [PSH, ACK] Seq=1 Ack=9 Win=637
42	7.957061	127.0.0.1	127.0.0.1	TCP	56	55633 → 1335 [ACK] Seq=9 Ack=42 Win=6377 Le

39 - Command sent from client to server - 3 indicating the command is factors than space and int to calculate for, end with postfix \, the command sent here is **factors**:

10251

```

02 00 00 00 45 00 00 3c 00 00 40 00 40 06 00 00 .....E.<..@..@..
7f 00 00 01 7f 00 00 01 d9 51 05 37 92 37 a8 57 .....Q.7.7.W
bf f5 88 87 80 18 18 ea fe 30 00 00 01 01 08 0a .....0.....
2e 57 94 fe 06 73 dc e7 33 20 31 30 32 35 31 5c .W...s. 3 10251\

```

40 - ack from server for packet 39.

41 - response from server - sent response for the command sent.

```

Data: 746865207072696d652066616374617273206166203130323531206172653a20332c
000 02 00 00 00 45 00 00 5d 00 00 40 00 40 06 00 00 .....E..]..@..@..
010 7f 00 00 01 7f 00 00 01 05 37 d9 51 bf f5 88 87 .....7.Q.....
020 92 37 a8 5f 80 18 18 eb fe 51 00 00 01 01 08 0a ..7.....Q.....
030 06 76 86 4d 2e 57 94 fe 74 68 65 20 70 72 69 6d ..v.M.W.. the prim
040 65 20 66 61 63 74 6f 72 73 20 6f 66 20 31 30 32 e factor s of 102
050 35 31 20 61 72 65 3a 20 33 2c 20 31 37 2c 20 36 51 are: 3, 17, 6
060 37 7

```

42 - ack from client to server for packet 41.