

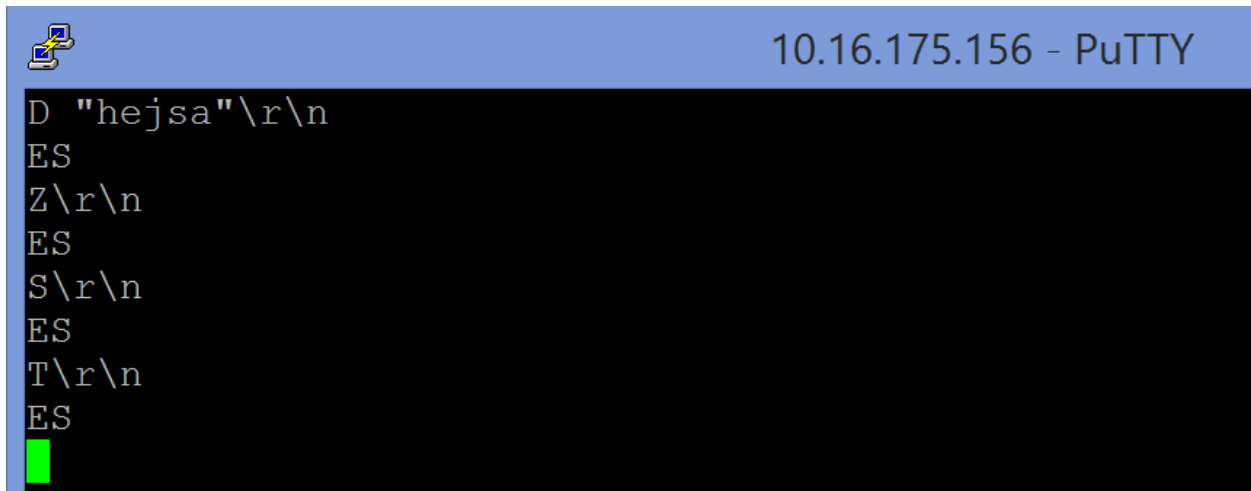
Test journal - Datakommunikation

Spørgsmål 1: Netstat -an -b finder hvor scale.exe er listening og hvilken port.

Eksempel på hvordan det ser ud herunder.

```
TCP    0.0.0.0:4567    0.0.0.0:0    LISTENING
[Scale.exe]
```

Spørgsmål 2: Efter ny Scale.exe er uploadet virker alle gamle kommandoer ikke, som vist herunder.



10.16.175.156 - PuTTY

```
D "hejsa"\r\n
ES
Z\r\n
ES
S\r\n
ES
T\r\n
ES
```

Til gengæld har funktionen RM20 nu fået lidt funktionalitet, med retursvar fra brugeren af vægten som vist herunder.

```
RM20 4 "Tester Key in" "aebler" "10"
RM20 B
RM20 A indtast
```

Vi sender først den øverste besked til serveren, og får et vente-signal tilbage, og når serverens bruger indtaster en besked og trykker på enter returneres RM 20 A, "hans besked".

Spørgsmål 3: Serveren tager imod multiple klienter, og virker som en fysisk vægt i det omfang at den returnerer "RM20 I" når den er optaget og dem der prøver at sende

kommandoer (Læs RM20 kommandoer da de simple kommandoer går igennem, som f.eks. S/T/Z etc. hvilket betyder at andre brugere kan nulstille vægten (simulator) mens en RM20 kommando er undervejs) undervejs bliver sat til at vente indtil nuværende bruger har overstået sin ordre.

Spørgsmål 4:

På figuren nedenfor kan man se de pakker vi modtager fra en specifik ip adresse. Vi har i dette tilfælde filtreret med udgangspunkt i en IP adresse, så vi således kan se de præcise pakker som bevæger sig imellem klienten og serveren, altså 2 end systems.

Filter: ip.addr == 10.16.168.108		Expression...		Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info
7894	90.370471000	10.16.168.108	10.16.175.156	TCP	81	22527->4567 [PSH, ACK] Seq=23 Ack=52 Win=64 Len=27
7896	90.370750000	10.16.168.108	10.16.175.156	TCP	56	22527->4567 [PSH, ACK] Seq=50 Ack=52 Win=64 Len=2
7897	90.370823000	10.16.175.156	10.16.168.108	TCP	54	4567->22527 [ACK] Seq=52 Ack=52 Win=256 Len=0
7898	90.370989000	10.16.175.156	10.16.168.108	TCP	59	4567->22527 [PSH, ACK] Seq=52 Ack=52 Win=256 Len=5
7905	90.422369000	10.16.168.108	10.16.175.156	TCP	54	22527->4567 [ACK] Seq=52 Ack=57 Win=64 Len=0
11755	134.306163000	10.16.168.108	10.16.175.156	TCP	56	[TCP Previous segment not captured] 22527->4567 [PSH, ACK] Seq=50 Ack=52 Win=64 Len=2
11756	134.306233000	10.16.175.156	10.16.168.108	TCP	66	[TCP Dup ACK 7898#1] 4567->22527 [ACK] Seq=57 Ack=52 Win=256 Len=0
11789	134.580025000	10.16.168.108	10.16.175.156	TCP	364	[TCP Retransmission] 22527->4567 [PSH, ACK] Seq=52 Ack=57 Win=64 Len=27
11790	134.580110000	10.16.175.156	10.16.168.108	TCP	66	4567->22527 [ACK] Seq=57 Ack=362 Win=254 Len=0 SLE=0
13428	152.687226000	10.16.175.156	10.16.168.108	TCP	54	4567->22527 [RST, ACK] Seq=57 Ack=362 Win=0 Len=0
16391	198.659590000	10.16.168.108	224.0.0.252	LLMNR	64	Standard query 0x6313 A wpad
16392	198.659785000	10.16.168.108	224.0.0.252	LLMNR	64	Standard query 0x6cd9 AAAA wpad
16404	199.069225000	10.16.168.108	224.0.0.252	LLMNR	64	Standard query 0x6cd9 AAAA wpad
16405	199.069389000	10.16.168.108	224.0.0.252	LLMNR	64	Standard query 0x6313 A wpad
21789	260.681834000	10.16.168.108	10.16.175.156	TCP	66	22563->4567 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=0
21792	260.683316000	10.16.175.156	10.16.168.108	TCP	66	4567->22563 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=0
21793	260.690636000	10.16.168.108	10.16.175.156	TCP	54	22563->4567 [ACK] Seq=1 Ack=1 Win=16384 Len=0
21794	260.701611000	10.16.168.108	10.16.175.156	TCP	75	22563->4567 [PSH, ACK] Seq=1 Ack=1 Win=16384 Len=21

Går vi så i detaljerne, kan vi se en specifik pakke overførelse, her er det klienten som har sendt en besked til serveren og derfra afventer en respons. Ud fra beskeden kan vi se at klienten har spurgt om at få returneret en vægt, da klienten har skrevet "S" igennem Telnet.

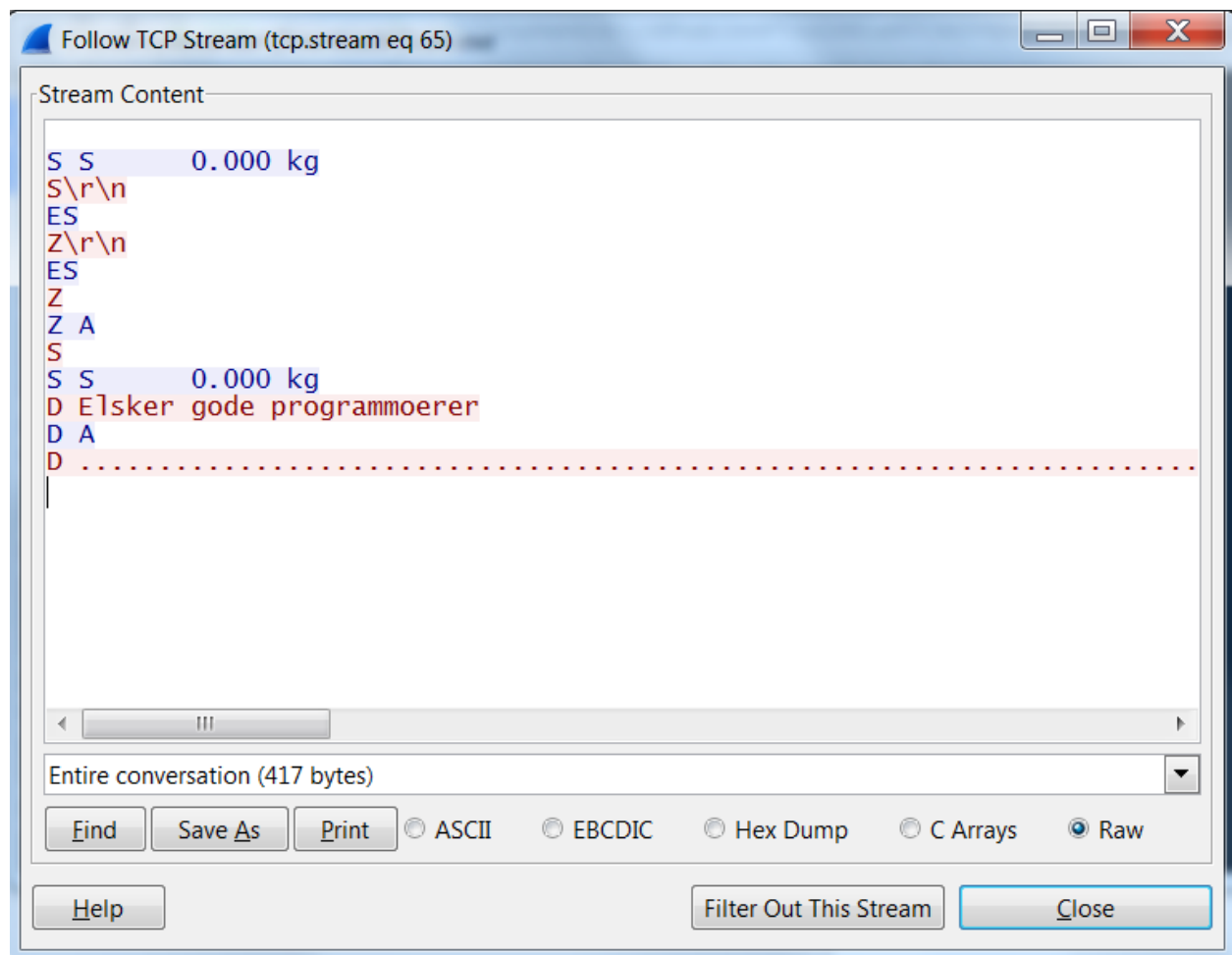
1460	24.7868930	10.16.168.108	10.16.175.156	TCP	57
0000	54 26 96 e0 cc df 60 57 18 98 6b e7 08 00 45 00	T&....`W ..k...E.			
0010	00 2b 4c d4 40 00 80 06 41 d0 0a 10 a8 6c 0a 10	.+L.@... A....l..			
0020	af 9c 58 23 11 d7 3a 4e 21 23 dc c0 0c 96 50 18	..X#...N !#....P.			
0030	00 40 37 91 00 00 53 0d 0a	.@7...S. .			

Vi kan herefter se at serveren returnere en besked tilbage til klienten, da vi ser i data at den siger "0.000 kg", derfra ved vi altså at serveren kunne modtage en besked og såfremt også returnere en besked tilbage til klienten.

1460	24.7868930	10.16.168.108	10.16.175.156	TCP	57
0000	60 57 18 98 6b e7 54 26 96 e0 cc df 08 00 45 00	`W..k.T&E.			
0010	00 3b 6e 76 40 00 80 06 20 1e 0a 10 af 9c 0a 10	.:nv@...			
0020	a8 6c 11 d7 58 23 dc c0 0c 96 3a 4e 21 26 50 18	.l..X#... :N!&P.			
0030	01 00 9d 41 00 00 53 20 53 20 20 20 20 20 30	...A..S S 0			
0040	2e 30 30 30 20 6b 67 0d 0a	.000 kg. .			

Vi kan også afslutningsvist til spørgsmålet se på TCP streamen, altså hvilke beskeder der har været sendt. Dette kan vi se på figuren nedenfor. Det skal siges at vi her følger alle beskeder der har blevet sendt frem og tilbage fra start til slut. Vi kan her se at klienten er markeret med magenta, og server svar er markeret med blå. Her ser vi at klienten har bedt om vægten og fået returneret den, yderligere har klienten også indskrevet en “display text”, hvor klienten har skrevet “Elsker gode programmoerer”, her har serveren så givet svar på at beskeden har gået igennem.

Således har vi altså kunne sniffe de pakker som er blevet sendt mellem klienten og serveren.



Spørgsmål 5: Herunder ser vi først porte brugt ved transmission til server og under den hvilke porte der blev brugt til svar.

```
Transmission Control Protocol, Src Port: 22563 (22563), Dst Port: 4567 (4567), Seq: 0, Ack: 1, Len: 3
Source Port: 22563 (22563)
Destination Port: 4567 (4567)

Transmission Control Protocol, Src Port: 4567 (4567), Dst Port: 22563 (22563), Seq: 1, Ack: 3, Len: 0
Source Port: 4567 (4567)
Destination Port: 22563 (22563)
```

IP adresserne benyttet står længere oppe i dokumentet men for oversigt kommer de her:

Server: 10.16.175.156 port: 4567

Klient: 10.16.168.108 port: 22563

Spørgsmål 6: Se eclipse zip-fil.

Spørgsmål 7:

Pros:

- Man kan tilgå serveren ved brug af en browser, man behøver altså ikke en særskilt klient.
- Kan give nemmere readouts geolokations mæssigt.

Cons:

- Kan blive DDoSet

Spørgsmål 8: Simulator af vægt er som udgangspunkt mangelfuld og fejlbehæftet.

Første udgave vi fik tilgang til kunne som start tage imod de mest simple kommandoer, altså ændre display på server fra klient og aflæse vægten plus få den til at lave Tare vægt. Alle andre kommandoer virkede ikke og returnerede som oftest "ES" i server respons.

Hvad der herefter blev sjovt at se var, serveren holdt op med at acceptere \r\n og kunne nu tilgås ved f.eks. for ændring af display besked skrive "D hvad end der nu skulle på display", hvis man tilføjede \r\n blev de nu en del af beskeden ligesom gåseøjne nu blev en del af display beskeden.

Når vi så kigger på den nye udgave scale.exe så virker ingen af de gamle syntakser, til gengæld kan man nu benytte dem uden \r\n, dvs. skrives S og "returtast" i telnet klienten returneres vægt, Z for reset og så videre hertil skal tilføjes at de simple kommandoer også kan benyttes selvom der er optaget på vægten, dvs. man kan benytte Z og nulstille vægten mens man er igang med en RM20 kommando fra anden bruger og server-brugeren er igang med at svare, vi er ikke helt indforståede med om det er meningen men som udgangspunkt udgår vi fra at det må være en fejl.

DW funktionen virker ikke i nogen af udgaverne, i den første returnerer den "ES" og i udgave #2 (scale.exe) får man "DW A" som svar.

Hvad der kunne ønskes var at alle funktioner var implementeret i simulator-vægten så man fik en idé om at det der blev sendt til server rent faktisk virkede og påvirkede server-side på en måde så eventuel fejlsøgning på klient-side er muligt.

Herudover at der ikke returneres noget eller at noget går så meget i stykker på server at \r\n ikke længere er nødvendigt og herudover rent faktisk laver fejl.