



Architecture & Protocols

Lab 2

Report Practical Work

3^{ème} année - RTS

PETIT Alexandre
Arthur LAUMY

TP2 REPORT

I – Capture FTP Session

09-25 13:44:48.308339401	192.168.28.8	192.168.28.100	TCP	76 35270 -> 21 [SYN] Seq=0 Win=64248 Len=0 MSS=1460 SACK_PERM=1 TSval=3508209322 TSecr=0 WS=128
09-25 13:44:48.308854549	192.168.28.100	192.168.28.8	TCP	76 21 -> 35270 [SYN, ACK] Seq=8 Ack=1 Win=65168 Len=0 MSS=1460 SACK_PERM=1 TSval=3751468476 TSecr=3508209322
09-25 13:44:48.308935240	192.168.28.8	192.168.28.100	TCP	68 35270 -> 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3508209322 TSecr=3751468476
09-25 13:44:48.358104034	192.168.28.100	192.168.28.8	FTP	88 Response: 220 (vsFTPd 3.0.3)
09-25 13:44:48.358164997	192.168.28.8	192.168.28.100	TCP	68 35270 -> 21 [ACK] Seq=1 Ack=21 Win=64256 Len=0 TSval=3508209372 TSecr=3751468525
09-25 13:44:53.879905607	192.168.28.8	192.168.28.100	FTP	83 Request: USER tpresseau
09-25 13:44:53.879905630	192.168.28.100	192.168.28.8	TCP	68 21 -> 35270 [ACK] Seq=21 Ack=16 Win=65280 Len=0 TSval=3751474046 TSecr=3508205092
09-25 13:44:53.879905630	192.168.28.100	192.168.28.8	FTP	102 Response: 331 Welcome to the network.
09-25 13:44:53.879905630	192.168.28.8	192.168.28.100	TCP	68 35270 -> 21 [ACK] Seq=16 Ack=55 Win=64256 Len=0 TSval=3508205093 TSecr=3751474046
09-25 13:44:58.782890548	192.168.28.8	192.168.28.100	FTP	83 Request: PASS sethis
09-25 13:44:58.824548577	192.168.28.100	192.168.28.8	TCP	68 21 -> 35270 [ACK] Seq=55 Ack=31 Win=65280 Len=0 TSval=3751478992 TSecr=3508210796
09-25 13:44:59.106487223	192.168.28.100	192.168.28.8	FTP	91 Response: 230 Login successful.
09-25 13:44:59.106548624	192.168.28.8	192.168.28.100	TCP	68 35270 -> 21 [ACK] Seq=31 Ack=78 Win=64256 Len=0 TSval=3508211120 TSecr=3751479274
09-25 13:44:59.106669934	192.168.28.8	192.168.28.100	FTP	74 Request: SYST
09-25 13:44:59.106769572	192.168.28.100	192.168.28.8	TCP	68 21 -> 35270 [ACK] Seq=78 Ack=37 Win=65280 Len=0 TSval=3751479274 TSecr=3508211120
09-25 13:44:59.106860974	192.168.28.100	192.168.28.8	FTP	87 Response: 215 UNIX Type: L8
09-25 13:44:59.148392907	192.168.28.8	192.168.28.100	TCP	68 35270 -> 21 [ACK] Seq=37 Ack=97 Win=64256 Len=0 TSval=3508211162 TSecr=3751479274

Figure 1 – Capture Connection FTP

The connection between our computer and the computer with the IP address 192.168.28.100 is established using the TCP protocol. After this connection, we see an alternation between the TCP and FTP protocols. We can see that all these exchanges are not encrypted (the appearance of the password and the username). In fact, this protocol is so old that network security was not a real problem at the time.

The TCP protocol is used to manage requests to the storage server, identification management (user and password) and network connection management (ACK and synchronization). The FTP protocol is used for file transfer and listing.

09-25 13:45:55.911936760	192.168.28.8	192.168.28.100	FTP	76 Request: TYPE I
09-25 13:45:55.912399445	192.168.28.100	192.168.28.8	FTP	99 Response: 200 Switching to Binary mode.
09-25 13:45:55.912565504	192.168.28.8	192.168.28.100	FTP	95 Request: PORT 192,168,28,8,160,255
09-25 13:45:55.913114407	192.168.28.100	192.168.28.8	FTP	119 Response: 200 PORT command successful. Consider using PASV.
09-25 13:45:55.913274574	192.168.28.8	192.168.28.100	FTP	80 Request: RETR 1.png
09-25 13:45:55.914253405	192.168.28.100	192.168.28.8	TCP	76 20 -> 41215 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3751536081 TSecr=0 WS=128
09-25 13:45:55.914329960	192.168.28.8	192.168.28.100	TCP	76 41215 -> 20 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3508267927 TSecr=3751536081
09-25 13:45:55.914583839	192.168.28.100	192.168.28.8	TCP	68 20 -> 41215 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3751536081 TSecr=3508267927
09-25 13:45:55.914648470	192.168.28.100	192.168.28.8	FTP	134 Response: 150 Opening BINARY mode data connection for 1.png (48803 bytes).
09-25 13:45:55.918773357	192.168.28.100	192.168.28.8	FTP-DATA	14548 FTP Data: 14480 bytes (PORT) (RETR 1.png)
09-25 13:45:55.918848459	192.168.28.8	192.168.28.100	TCP	68 41215 -> 20 [ACK] Seq=1 Ack=14481 Win=55808 Len=0 TSval=3508267931 TSecr=3751536085
09-25 13:45:55.919205292	192.168.28.100	192.168.28.8	FTP-DATA	24684 FTP Data: 24616 bytes (PORT) (RETR 1.png)
09-25 13:45:55.919253302	192.168.28.8	192.168.28.100	TCP	68 41215 -> 20 [ACK] Seq=1 Ack=39097 Win=49152 Len=0 TSval=3508267932 TSecr=3751536086
09-25 13:45:55.919439936	192.168.28.100	192.168.28.8	FTP-DATA	1775 FTP Data: 1707 bytes (PORT) (RETR 1.png)
09-25 13:45:55.919504764	192.168.28.8	192.168.28.100	TCP	68 41215 -> 20 [ACK] Seq=1 Ack=40805 Win=64128 Len=0 TSval=3508267932 TSecr=3751536086
09-25 13:45:55.919563546	192.168.28.8	192.168.28.100	TCP	68 41215 -> 20 [FIN, ACK] Seq=1 Ack=40805 Win=64128 Len=0 TSval=3508267932 TSecr=3751536086
09-25 13:45:55.919755982	192.168.28.100	192.168.28.8	TCP	68 20 -> 41215 [ACK] Seq=40805 Ack=2 Win=64256 Len=0 TSval=3751536087 TSecr=3508267932
09-25 13:45:55.919791970	192.168.28.100	192.168.28.8	FTP	92 Response: 226 Transfer complete.
09-25 13:45:55.919837293	192.168.28.8	192.168.28.100	TCP	68 35270 -> 21 [ACK] Seq=117 Ack=383 Win=64256 Len=0 TSval=3508267932 TSecr=3751536081

Figure 2 – Capture Get FTP

221 2024-09-25 14:40:13.549441330	192.168.28.8	192.168.28.100	FTP	76 Request: TYPE I
222 2024-09-25 14:40:13.549590974	192.168.28.100	192.168.28.8	FTP	99 Response: 200 Switching to Binary mode.
224 2024-09-25 14:40:13.549763174	192.168.28.8	192.168.28.100	FTP	94 Request: PORT 192,168,28,8,140,99
225 2024-09-25 14:40:13.549944301	192.168.28.100	192.168.28.8	FTP	119 Response: 200 PORT command successful. Consider using PASV.
226 2024-09-25 14:40:13.550095060	192.168.28.8	192.168.28.100	FTP	83 Request: STOR test.txt
230 2024-09-25 14:40:13.550784048	192.168.28.100	192.168.28.8	FTP	90 Response: 150 Ok to send data.
234 2024-09-25 14:40:13.551391758	192.168.28.100	192.168.28.8	FTP	92 Response: 226 Transfer complete.

Figure 3 – Capture Send FTP

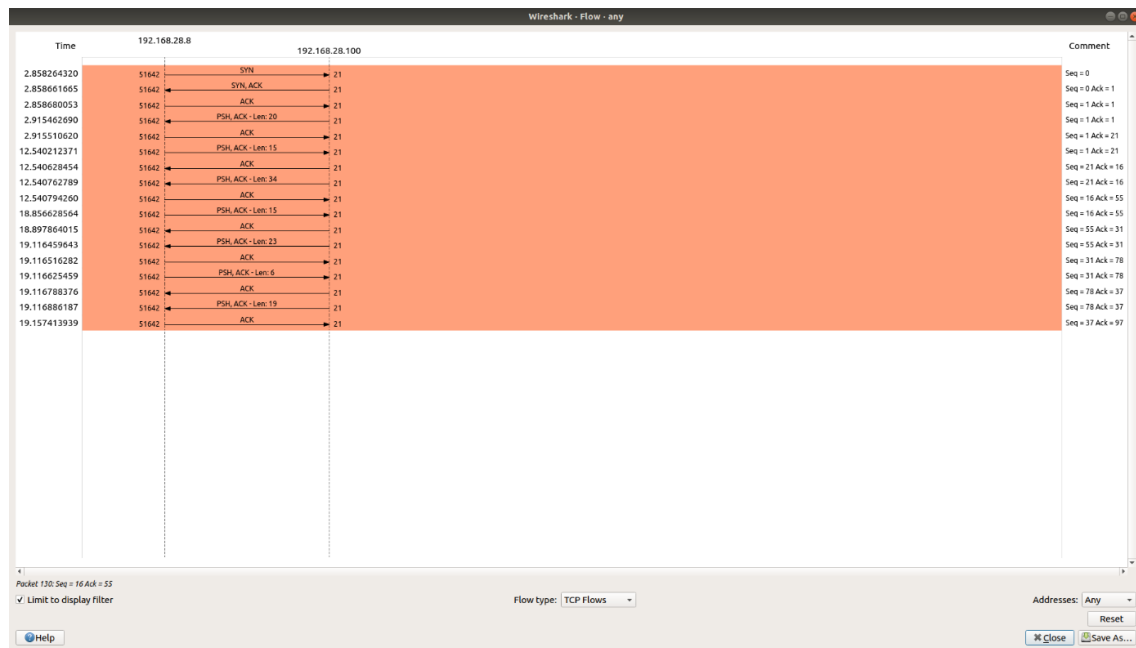
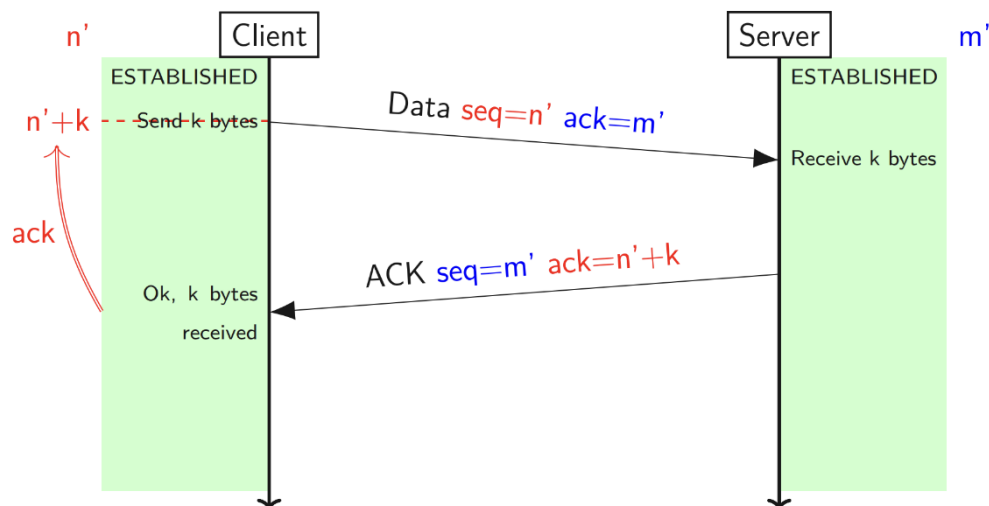


Figure 4 - Flow chart

You can see from the graph (obtained with Statistics > Flow Graph > TCP Flow) that the seq becomes the old ACK, clearly showing the next steps.

TCP sending data – simple ACK



93

Figure 5 – Course Scheme

In fact, we can see that when $ack=m'=21$, the following seq is equal to 21. Furthermore, len in the flowchart corresponds to k in the graph. Therefore, we see that $seq=21$ and the length of the file to be sent is 34, so ack becomes $m'+k=55$. There is therefore a sequence of steps in this protocol that respects the TCP protocol.

II – MTU (Maximum Transfer Unit)

We use the `ifconfig` command to determine the MTU.

```
enp0s31f6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.28.8 netmask 255.255.255.0 broadcast 192.168.28.255
  inet6 fe80::56bf:64ff:fe64:a688 prefixlen 64 scopeid 0x20<link>
  ether 54:bf:64:64:a6:88 txqueuelen 1000 (Ethernet)
  RX packets 22465 bytes 21124303 (21.1 MB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 12072 bytes 1666085 (1.6 MB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
  device interrupt 16 memory 0xef400000-ef420000
```

Figure 6 – MTU=1500

The MTU is now 1500. We then change it to 100 with the following command. We can see that the MTU has changed. We can note that the command need the administration rights.

```
tpreseau@d055-pc8:~$ ifconfig enp0s31f6 mtu 100
SIOCSIFMTU: Operation not permitted
tpreseau@d055-pc8:~$ sudo ifconfig enp0s31f6 mtu 100
[sudo] password for tpreseau:
tpreseau@d055-pc8:~$ ifconfig enp0s31f6
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST> mtu 100
  inet 192.168.28.8 netmask 255.255.255.0 broadcast 192.168.28.255
  ether 54:bf:64:64:a6:88 txqueuelen 1000 (Ethernet)
  RX packets 23534 bytes 21275715 (21.2 MB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 13127 bytes 1787456 (1.7 MB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
  device interrupt 16 memory 0xef400000-ef420000
```

Figure 7 – MTU=100

MTU stands for Maximum Transmission Unit. It is the maximum size of a packet that can be transmitted without being truncated. If it exceeds this size, the packet will be fragmented.

32	2024-09-25 15:07:55.276368894	192.168.28.8	192.168.28.7	ICMP	92 Echo (ping) request	id=0x2012, seq=2/512, ttl=64 (reply in 33)
33	2024-09-25 15:07:55.276525850	192.168.28.7	192.168.28.8	ICMP	92 Echo (ping) reply	id=0x2012, seq=2/512, ttl=64 (request in 32)

Figure 8– Protocol ICMP with Ping 50 bytes

We can see that when we ping 50 bytes there is no fragmentation for an MTU=100. We can see that the packet size is 92, which means that there is no fragmentation because the size is less than the MTU.

On the other hand, if we ping 80 bytes, we see that it is truncated because the packet size is 102. Thanks to the IPv4 protocol, we can see that the packets are truncated into two, one of size 42 and the other of size 60. So 80-byte ICMP packets are truncated.

120	2024-09-25 15:08:10.126178353	192.168.28.8	192.168.28.7	IPv4	114 Fragmented IP protocol (proto=ICMP 1, off=0, ID=c9ee) [Reassembled in #121]
121	2024-09-25 15:08:10.126187299	192.168.28.8	192.168.28.7	ICMP	42 Echo (ping) request id=0x2013, seq=1/256, ttl=64 (reply in 123)
122	2024-09-25 15:08:10.126427339	192.168.28.7	192.168.28.8	IPv4	114 Fragmented IP protocol (proto=ICMP 1, off=0, ID=7d25) [Reassembled in #123]
123	2024-09-25 15:08:10.126442344	192.168.28.7	192.168.28.8	ICMP	60 Echo (ping) reply id=0x2013, seq=1/256, ttl=64 (request in 121)

Figure 9– Protocol ICMP with Ping 80 bytes

1.	2024-09-25 15:14:49.569861711	192.168.28.100	192.168.28.8	FTP-DATA	33370 FTP Data: 33384 bytes (PORT) (RETR son.mp3)
1.	2024-09-25 15:14:49.569870552	192.168.28.8	192.168.28.100	TCP	66 49711 → 20 [ACK] Seq=1 Ack=18266521 Win=582916 Len=0 TSval=3513601515 TSecr=
1.	2024-09-25 15:14:49.570169728	192.168.28.100	192.168.28.8	FTP-DATA	31922 FTP Data: 31856 bytes (PORT) (RETR son.mp3)
1.	2024-09-25 15:14:49.570177709	192.168.28.8	192.168.28.100	TCP	66 49711 → 20 [ACK] Seq=1 Ack=18298377 Win=582912 Len=0 TSval=3513601515 TSecr=
1.	2024-09-25 15:14:49.570509544	192.168.28.100	192.168.28.8	FTP-DATA	13088 FTP Data: 13032 bytes (PORT) (RETR son.mp3)
1.	2024-09-25 15:14:49.570515027	192.168.28.8	192.168.28.100	TCP	66 49711 → 20 [ACK] Seq=1 Ack=18311489 Win=595584 Len=0 TSval=3513601516 TSecr=
1.	2024-09-25 15:14:49.570510740	192.168.28.100	192.168.28.8	FTP-DATA	1900 FTP Data: 1920 bytes (PORT) (RETR son.mp3)
1.	2024-09-25 15:14:49.570528146	192.168.28.8	192.168.28.100	TCP	66 49711 → 20 [ACK] Seq=1 Ack=18313330 Win=595664 Len=0 TSval=3513601516 TSecr=
1.	2024-09-25 15:14:49.571031369	192.168.28.100	192.168.28.8	FTP	90 Response: 226 Transfer complete.

Figure 10– Protocol TCP with MTU=1500

With the TCP protocol, we can see that the protocol does not respect the MTU. In fact, TCP is an old protocol, so it does not respect the MTU, which was introduced later with IPv4. As a result, packet sizes are very large, up to 32,000 despite an MTU of 1,500.

```

tpreseau@d055-pc8: ~
File Edit View Search Terminal Help
tpreseau@d055-pc8:~$ tracepath www.google.com
1?: [LOCALHOST] pmtu 1500
1: _gateway 0.266ms
1: _gateway 0.269ms
2: _gateway 0.372ms reached
Resume: pmtu 1500 hops 2 back 1
tpreseau@d055-pc8:~$

```

Figure 11– TracePath

All websites agreed that the best MTU is the 1500, it's the value by default.

III –TCP Window Size

The script is creating a server which is running and listening on 127.0.0.1 at port 9999. It is handling TCP request until we stop it.

1.	2024-09-25 16:30:31.060292755	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263825 Win=209280 Len=0 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060348659	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263825 Ack=1 Win=65536 Len=4 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060359712	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263829 Win=209280 Len=0 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060416896	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263829 Ack=1 Win=65536 Len=4 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060426705	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263833 Win=209280 Len=0 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060477134	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263833 Ack=1 Win=65536 Len=4 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060487997	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263837 Win=209280 Len=0 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060601058	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263837 Ack=1 Win=65536 Len=4 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060671193	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263841 Win=209280 Len=0 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060733167	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263841 Ack=1 Win=65536 Len=4 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060742796	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263845 Win=209280 Len=0 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060806192	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263845 Ack=1 Win=65536 Len=4 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.060879728	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263849 Win=209280 Len=0 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.061009435	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263849 Ack=1 Win=65536 Len=4 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.061028700	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263853 Win=209280 Len=0 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.061080161	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263853 Ack=1 Win=65536 Len=4 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.061089623	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263857 Win=209280 Len=0 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.061149599	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263857 Ack=1 Win=65536 Len=4 TSval=2921687473 TSecr=
1.	2024-09-25 16:30:31.061158986	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263861 Win=209280 Len=0 TSval=2921687474 TSecr=
1.	2024-09-25 16:30:31.061217129	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263861 Ack=1 Win=65536 Len=4 TSval=2921687474 TSecr=
1.	2024-09-25 16:30:31.061226454	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263865 Win=209280 Len=0 TSval=2921687474 TSecr=
1.	2024-09-25 16:30:31.061288432	127.0.0.1	127.0.0.1	TCP	70 54442 → 9999 [PSH, ACK] Seq=3263865 Ack=1 Win=65536 Len=4 TSval=2921687474 TSecr=
1.	2024-09-25 16:30:31.061297337	127.0.0.1	127.0.0.1	TCP	66 9999 → 54442 [ACK] Seq=1 Ack=3263869 Win=209280 Len=0 TSval=2921687474 TSecr=

Figure 12 - Window Capture without sleep

122	2024-09-25 16:33:58.072317704	127.0.0.1	127.0.0.1	TCP	2658 60452 → 9999 [PSH, ACK] Seq=106397 Ack=1 Win=65536 Len=2592 TSval=2921894482
123	2024-09-25 16:33:58.116305791	127.0.0.1	127.0.0.1	TCP	66 9999 → 60452 [ACK] Seq=1 Ack=106989 Win=6576 Len=0 TSval=2921894526 TSecr=29
124	2024-09-25 16:33:58.116329513	127.0.0.1	127.0.0.1	TCP	2146 60452 → 9999 [PSH, ACK] Seq=108989 Ack=1 Win=65536 Len=2688 TSval=2921894526
125	2024-09-25 16:33:58.168302317	127.0.0.1	127.0.0.1	TCP	66 9999 → 60452 [ACK] Seq=1 Ack=111069 Win=7552 Len=0 TSval=2921894576 TSecr=29
126	2024-09-25 16:33:58.168318382	127.0.0.1	127.0.0.1	TCP	2526 60452 → 9999 [PSH, ACK] Seq=111069 Ack=1 Win=65536 Len=2460 TSval=2921894576
127	2024-09-25 16:33:58.208302688	127.0.0.1	127.0.0.1	TCP	66 9999 → 60452 [ACK] Seq=1 Ack=113529 Win=6272 Len=0 TSval=2921894618 TSecr=29
128	2024-09-25 16:33:58.208317668	127.0.0.1	127.0.0.1	TCP	3822 60452 → 9999 [PSH, ACK] Seq=113529 Ack=1 Win=65536 Len=2956 TSval=2921894618
129	2024-09-25 16:33:58.252303972	127.0.0.1	127.0.0.1	TCP	66 9999 → 60452 [ACK] Seq=1 Ack=116485 Win=4864 Len=0 TSval=2921894662 TSecr=29
130	2024-09-25 16:33:58.252304685	127.0.0.1	127.0.0.1	TCP	2634 60452 → 9999 [PSH, ACK] Seq=116485 Ack=1 Win=65536 Len=2568 TSval=2921894662
131	2024-09-25 16:33:58.296292247	127.0.0.1	127.0.0.1	TCP	66 9999 → 60452 [ACK] Seq=1 Ack=119053 Win=2384 Len=0 TSval=2921894706 TSecr=29
132	2024-09-25 16:33:58.302305813	127.0.0.1	127.0.0.1	TCP	2370 [TCP Window Full] 60452 → 9999 [PSH, ACK] Seq=116303 Ack=1 Win=65536 Len=268
133	2024-09-25 16:33:58.552325889	127.0.0.1	127.0.0.1	TCP	66 [TCP ZeroWindow] 9999 → 60452 [ACK] Seq=1 Ack=121357 Win=0 Len=0 TSval=29218
134	2024-09-25 16:33:58.796308146	127.0.0.1	127.0.0.1	TCP	66 [TCP Keep-Alive] 60452 → 9999 [ACK] Seq=121356 Ack=1 Win=65536 Len=0 TSval=2
135	2024-09-25 16:33:58.796307689	127.0.0.1	127.0.0.1	TCP	66 [TCP ZeroWindow] 9999 → 60452 [ACK] Seq=1 Ack=121357 Win=0 Len=0 TSval=29218
136	2024-09-25 16:33:59.388308061	127.0.0.1	127.0.0.1	TCP	66 [TCP Keep-Alive] 60452 → 9999 [ACK] Seq=121356 Ack=1 Win=65536 Len=0 TSval=2
137	2024-09-25 16:33:59.388318382	127.0.0.1	127.0.0.1	TCP	66 [TCP ZeroWindow] 9999 → 60452 [ACK] Seq=1 Ack=121357 Win=0 Len=0 TSval=29218
138	2024-09-25 16:34:00.384301589	127.0.0.1	127.0.0.1	TCP	66 [TCP Keep-Alive] 60452 → 9999 [ACK] Seq=121356 Ack=1 Win=65536 Len=0 TSval=2
139	2024-09-25 16:34:00.384318613	127.0.0.1	127.0.0.1	TCP	66 [TCP ZeroWindow] 9999 → 60452 [ACK] Seq=1 Ack=121357 Win=0 Len=0 TSval=29218
140	2024-09-25 16:34:02.256304853	127.0.0.1	127.0.0.1	TCP	66 [TCP Keep-Alive] 60452 → 9999 [ACK] Seq=121356 Ack=1 Win=65536 Len=0 TSval=2
141	2024-09-25 16:34:02.256316711	127.0.0.1	127.0.0.1	TCP	66 [TCP ZeroWindow] 9999 → 60452 [ACK] Seq=1 Ack=121357 Win=0 Len=0 TSval=29218
142	2024-09-25 16:34:06.188304561	127.0.0.1	127.0.0.1	TCP	66 [TCP Keep-Alive] 60452 → 9999 [ACK] Seq=121356 Ack=1 Win=65536 Len=0 TSval=2
143	2024-09-25 16:34:06.188313575	127.0.0.1	127.0.0.1	TCP	66 [TCP ZeroWindow] 9999 → 60452 [ACK] Seq=1 Ack=121357 Win=0 Len=0 TSval=29218

Figure 13 - Window capture with sleep

Without the line 10 command, the window will continue to fill without stopping. If, on the other hand, the line is uncommented, the window fills rapidly until it reaches saturation, i.e. it can no longer respond, and packets are put on hold. To obtain this capture, we had to observe in the Loopback.

IV – Capturing a web session with Telnet

```

tpreseau@d055-pc8:~$ telnet www.ensea.fr 80
Trying 10.10.17.5...
Connected to enseaweb.ensea.fr.
Escape character is '^'.
GET / HTTP/1.0

HTTP/1.1 301 Moved Permanently
Date: Wed, 25 Sep 2024 14:57:42 GMT
Server: Apache/2.4.41 (Ubuntu)
Location: https:///
Content-Length: 295
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://"/>here</a>.</p>
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at www.ensea.fr Port 80</address>
</body></html>
Connection closed by foreign host.
tpreseau@d055-pc8:~$ telnet www.ensea.fr 80
Trying 10.10.17.5...
Connected to enseaweb.ensea.fr.
Escape character is '^'.
GET / HTTP/1.1
Host: www.ensea.fr

HTTP/1.1 301 Moved Permanently
Date: Wed, 25 Sep 2024 14:58:00 GMT
Server: Apache/2.4.41 (Ubuntu)
Location: https://www.ensea.fr/
Content-Length: 307
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://www.ensea.fr/">here</a>.</p>
<hr>
<address>Apache/2.4.41 (Ubuntu) Server at www.ensea.fr Port 80</address>
</body></html>
Connection closed by foreign host.
tpreseau@d055-pc8:~$

```

We use Telnet to connect to a site (here www.ensea.fr) and make an HTTP request using the following model GET / HTTP/1.1 then the following line: Host: www.ensea.fr.

The first '/' corresponds to retrieving the file or target from the root of the host. Note that the second line is not necessary if you are using HTTP version 1.0.

As a result, we will get a 301 error, which will return us to the host page, but in HTTPS, a more secure HTTP protocol.

6	2024-09-25 16:53:33	140198442	10.10.17.5	192.168.28.8	HTTP	551 HTTP/1.1 408 Request Timeout (text/html)
77	2024-09-25 16:54:09	368845286	10.10.17.5	192.168.28.8	HTTP	551 HTTP/1.1 408 Request Timeout (text/html)
139	2024-09-25 16:54:55	756911170	10.10.17.5	192.168.28.8	HTTP	551 HTTP/1.1 408 Request Timeout (text/html)
275	2024-09-25 16:57:15	274040162	192.168.28.8	10.10.17.5	HTTP	68 GET / HTTP/1.1
277	2024-09-25 16:57:15	289842708	10.10.17.5	192.168.28.8	HTTP	552 HTTP/1.1 408 Bad Request (text/html)
307	2024-09-25 16:57:43	208406008	192.168.28.8	10.10.17.5	HTTP	68 GET / HTTP/1.1
309	2024-09-25 16:57:43	223355796	10.10.17.5	192.168.28.8	HTTP	570 HTTP/1.1 301 Moved Permanently (text/html)
342	2024-09-25 16:58:05	279088228	192.168.28.8	34.107.221.82	HTTP	367 GET /canonical.html HTTP/1.1
344	2024-09-25 16:58:05	282282748	34.107.221.82	192.168.28.8	HTTP	364 HTTP/1.1 200 OK (text/html)
381	2024-09-25 16:58:05	288348093	192.168.28.8	34.107.221.82	HTTP	369 GET /success.txt?ip=4 HTTP/1.1
383	2024-09-25 16:58:05	291409071	34.107.221.82	192.168.28.8	HTTP	282 HTTP/1.1 200 OK (text/plain)
394	2024-09-25 16:58:10	272076838	192.168.28.8	10.10.17.5	HTTP	68 GET / HTTP/1.1
396	2024-09-25 16:58:10	287285924	10.10.17.5	192.168.28.8	HTTP	575 HTTP/1.1 301 Moved Permanently (text/html)
291	2024-09-25 16:57:35	296619536	192.168.28.8	192.168.28.250	DNS	72 Standard query 0xfa08 A www.ensea.fr
292	2024-09-25 16:57:35	296630856	192.168.28.8	192.168.28.250	DNS	72 Standard query 0x5221 AAAA www.ensea.fr
293	2024-09-25 16:57:35	296641374	192.168.28.250	192.168.28.8	DNS	111 Standard query response 0xfa08 A www.ensea.fr CNAME enseaweb.ensea.fr A 10.10.17
294	2024-09-25 16:57:35	296873947	192.168.28.250	192.168.28.8	DNS	123 Standard query response 0x5221 AAAA www.ensea.fr CNAME enseaweb.ensea.fr AAAA 20
295	2024-09-25 16:57:35	297328065	192.168.28.8	10.10.17.5	TCP	74 59876 - 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3943041361 TSecr=
296	2024-09-25 16:57:35	297768996	10.10.17.5	192.168.28.8	TCP	74 80 - 59876 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=121
297	2024-09-25 16:57:35	297618507	192.168.28.8	10.10.17.5	TCP	66 59876 - 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3943041361 TSecr=1215225899
298	2024-09-25 16:57:35	297689301	0.0.0.0:0.0.0.0	Spanning-tree:(for-bridges) 00	STP	119 MST_Root = 32768/0/00:14:c2:14:ff:00 Cost = 44000 Port = 0x800b
299	2024-09-25 16:57:37	750287484	Dell_05:50:af	Spanning-tree:(for-bridges) 00	STP	119 MST_Root = 32768/0/00:14:c2:14:ff:00 Cost = 44000 Port = 0x800b
300	2024-09-25 16:57:39	584792494	Dell_05:50:af	LLDP Multicast	LLDP	60 TTL = 120 System Name = SW-0055
301	2024-09-25 16:57:39	257834373	Dell_05:50:af	Spanning-tree:(for-bridges) 00	STP	119 MST_Root = 32768/0/00:14:c2:14:ff:00 Cost = 44000 Port = 0x800b
302	2024-09-25 16:57:40	332309440	Dell_64:a6:88	Vmware_a5:72:a5	ARP	42 Who has 192.168.28.250? Tell 192.168.28.8
303	2024-09-25 16:57:40	332478263	Vmware_a5:72:a5	Dell_64:a6:88	ARP	60 192.168.28.250 is at 00:50:56:a5:72:a5
304	2024-09-25 16:57:41	757507628	Dell_05:50:af	Spanning-tree:(for-bridges) 00	STP	119 MST_Root = 32768/0/00:14:c2:14:ff:00 Cost = 44000 Port = 0x800b
305	2024-09-25 16:57:42	434324853	192.168.28.8	10.10.17.5	TCP	62 59876 - 80 [FIN, ACK] Seq=1 Ack=17 Win=65152 Len=0 TSval=3943040498 TSecr=121522
306	2024-09-25 16:57:42	434607301	10.10.17.5	192.168.28.8	TCP	66 80 - 59876 [ACK] Seq=1 Ack=17 Win=65152 Len=0 TSval=1215233836 TSecr=3943040498
307	2024-09-25 16:57:43	208406008	192.168.28.8	10.10.17.5	HTTP	68 GET / HTTP/1.0
308	2024-09-25 16:57:43	208715480	10.10.17.5	192.168.28.8	TCP	66 80 - 59876 [ACK] Seq=1 Ack=19 Win=65152 Len=0 TSval=1215233810 TSecr=3943049272
309	2024-09-25 16:57:43	223355796	10.10.17.5	192.168.28.8	HTTP	570 HTTP/1.1 301 Moved Permanently (text/html)
310	2024-09-25 16:57:43	223589356	192.168.28.8	10.10.17.5	TCP	66 59876 - 80 [FIN, ACK] Seq=19 Ack=506 Win=64128 Len=0 TSval=3943049287 TSecr=1215
311	2024-09-25 16:57:43	223885856	10.10.17.5	192.168.28.8	TCP	66 80 - 59876 [ACK] Seq=506 Ack=20 Win=65152 Len=0 TSval=1215233825 TSecr=394304928

Figure 14 - Capture during HTTP

V – SSH Protocol

1.	2024-09-25 17:10:18	182484516	192.168.28.8	192.168.28.7	SSHv2	187 Client: Protocol (SSH-2.0-OpenSSH 7.6p1 Ubuntu-ubuntu0.7)
1.	2024-09-25 17:10:18	182809363	192.168.28.7	192.168.28.8	SSHv2	66 22 - 42674 [ACK] Seq=1 Ack=42 Win=65152 Len=0 TSval=2089196878 TSecr=1302696058
1.	2024-09-25 17:10:18	189858374	192.168.28.7	192.168.28.8	SSHv2	187 Server: Protocol (SSH-2.0-OpenSSH 7.6p1 Ubuntu-ubuntu0.7)
1.	2024-09-25 17:10:18	1898902635	192.168.28.8	192.168.28.7	SSHv2	66 42674 - 22 [ACK] Seq=42 Ack=42 Win=64256 Len=0 TSval=1302696066 TSecr=2089196885
1.	2024-09-25 17:10:18	190439768	192.168.28.7	192.168.28.8	TCP	66 42674 - 22 [ACK] Seq=42 Ack=1122 Win=64128 Len=0 TSval=1302696066 TSecr=208919688
1.	2024-09-25 17:10:18	191409140	192.168.28.8	192.168.28.7	SSHv2	1426 Client: Key Exchange Init
1.	2024-09-25 17:10:18	235442210	192.168.28.7	192.168.28.8	SSHv2	66 22 - 42674 [ACK] Seq=1122 Ack=1402 Win=64128 Len=0 TSval=2089196931 TSecr=1302696
1.	2024-09-25 17:10:18	235490657	192.168.28.8	192.168.28.7	SSHv2	114 Client: Diffie-Hellman Key Exchange Init
1.	2024-09-25 17:10:18	235678892	192.168.28.7	192.168.28.8	TCP	66 22 - 42674 [ACK] Seq=1122 Ack=1450 Win=64128 Len=0 TSval=2089196931 TSecr=1302696
1.	2024-09-25 17:10:18	243108953	192.168.28.7	192.168.28.8	SSHv2	518 Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypted packet (len=172)
1.	2024-09-25 17:10:18	253229752	192.168.28.8	192.168.28.7	SSHv2	82 Client: New Keys
1.	2024-09-25 17:10:18	255454507	192.168.28.7	192.168.28.8	TCP	66 22 - 42674 [ACK] Seq=1574 Ack=1450 Win=64128 Len=0 TSval=2089196991 TSecr=1302696
1.	2024-09-25 17:10:18	25565723	192.168.28.8	192.168.28.7	SSHv2	118 Client: Encrypted packet (len=44)
1.	2024-09-25 17:10:18	259535230	192.168.28.7	192.168.28.8	TCP	66 22 - 42674 [ACK] Seq=1574 Ack=1510 Win=64128 Len=0 TSval=2089196991 TSecr=1302696
1.	2024-09-25 17:10:18	259574213	192.168.28.7	192.168.28.8	SSHv2	110 Server: Encrypted packet (len=44)
1.	2024-09-25 17:10:18	26060742	192.168.28.8	192.168.28.7	SSHv2	134 Client: Encrypted packet (len=68)
1.	2024-09-25 17:10:18	33970207	192.168.28.7	192.168.28.8	TCP	66 22 - 42674 [ACK] Seq=1618 Ack=1578 Win=64128 Len=0 TSval=2089197035 TSecr=1302696
1.	2024-09-25 17:10:18	754543039	Dell_05:50:af	Spanning-tree:(for-bridges) 00	STP	119 MST_Root = 32768/0/00:14:c2:14:ff:00 Cost = 44000 Port = 0x800b
1.	2024-09-25 17:10:20	475379442	192.168.28.7	192.168.28.8	SSHv2	118 Server: Encrypted packet (len=52)
1.	2024-09-25 17:10:20	475836312	192.168.28.8	192.168.28.7	SSHv2	438 Client: Encrypted packet (len=372)
1.	2024-09-25 17:10:20	475886816	192.168.28.7	192.168.28.8	TCP	66 22 - 42674 [ACK] Seq=1678 Ack=1950 Win=64128 Len=0 TSval=2089199171 TSecr=1302696
1.	2024-09-25 17:10:20	481504006	192.168.28.7	192.168.28.8	SSHv2	118 Server: Encrypted packet (len=92)
1.	2024-09-25 17:10:20	524359567	192.168.28.8	192.168.28.7	TCP	66 42674 - 22 [ACK] Seq=1950 Ack=1722 Win=64128 Len=0 TSval=2089199196 TSecr=2089196
1.	2024-09-25 17:10:21	757271157	Dell_05:50:af	Spanning-tree:(for-bridges) 00	STP	119 MST_Root = 32768/0/00:14:c2:14:ff:00 Cost = 44000 Port = 0x800b
1.	2024-09-25 17:10:23	765118998	Dell_05:50:af	Spanning-tree:(for-bridges) 00	STP	119 MST_Root = 32768/0/00:14:c2:14:ff:00 Cost = 44000 Port = 0x800b
1.	2024-09-25 17:10:25	765708442	Dell_05:50:af	Spanning-tree:(for-bridges) 00	STP	119 MST_Root = 32768/0/00:14:c2:14:ff:00 Cost = 44000 Port = 0x800b
1.	2024-09-25 17:10:26	819001346	192.168.28.8	192.168.28.7	SSHv2	214 Client: Encrypted packet (len=148)
1.	2024-09-25 17:10:26	820384306	192.168.28.7	192.168.28.8	TCP	94 Server: Encrypted packet (len=26)
1.	2024-09-25 17:10:26	828426415	192.168.28.8	192.168.28.7	SSHv2	66 42674 - 22 [ACK] Seq=2098 Ack=1750 Win=64128 Len=0 TSval=1302704784 TSecr=2089205
1.	2024-09-25 17:10:26	828505142	192.168.28.8	192.168.28.7	SSHv2	178 Client: Encrypted packet (len=112)
1.	2024-09-25 17:10:26	871406031	192.168.28.7	192.168.28.8	TCP	66 22 - 42674 [ACK] Seq=1750 Ack=2210 Win=64128 Len=0 TSval=1302704852 TSecr=2089205
1.	2024-09-25 17:10:26	873760168	192.168.28.7	192.168.28.8	SSHv2	560 Server: Encrypted packet (len=508)
1.	2024-09-25 17:10:26	976328930	192.168.28.8	192.168.28.7	TCP	66 42674 - 22 [ACK] Seq=2210 Ack=2250 Win=64128 Len=0 TSval=1302704852 TSecr=2089205
1.	2024-09-25 17:10:26	976747922	192.168.28.7	192.168.28.8	TCP	66 42674 - 22 [ACK] Seq=2210 Ack=2294 Win=64128 Len=0 TSval=1302704852 TSecr=2089205
1.	2024-09-25 17:10:26	9767165374	192.168.28.8	192.168.28.7	SSHv2	1162 Client: Encrypted packet (len=1096)
1.	2024-09-25 17:10:26	977571088	192.168.28.7	192.168.28.8	TCP	66 22 - 42674 [ACK] Seq=2294 Ack=3306 Win=64128 Len=0 TSval=2089205673 TSecr=1302704
1.	2024-09-25 17:10:26	978755255	192.168.28.7	192.168.28.8	SSHv2	174 Server: Encrypted packet (len=108)
1.	2024-09-25 17:10:26	978827184	192.168.28.7	192.168.28.8	SSHv2	886 Server: Encrypted packet (len=820)
1.	2024-09-25 17:10:26	978836377	192.168.28.8	192.168.28.7	TCP	66 42674 - 22 [ACK] Seq=3306 Ack=3222 Win=64128 Len=0 TSval=1302704854 TSecr=2089205
1.	2024-09-25 17:10:27	852063966	192.168.28.7	192.168.28.8	SSHv2	150 Server: Encrypted packet (len=84)
1.	2024-09-25 17:10:27	892340362	192.168.28.8	192.168.28.7	TCP	66 42674 - 22 [ACK] Seq=3306 Ack=3306 Win=64128 Len=0 TSval=1302704968 TSecr=2089205

Figure 15 - SSH Capture

We can see an exchange of encrypted packets due to the wall command sending broadcast messages.