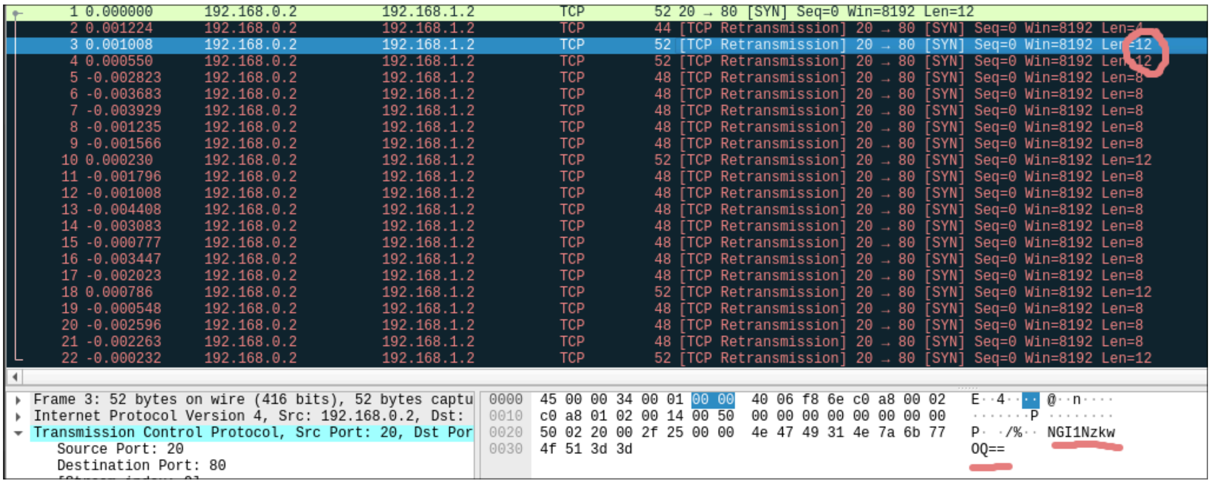
# When I opened the PCAP file, I saw some base64 valid codes. We can see only the ones with length=12 are valid, like :

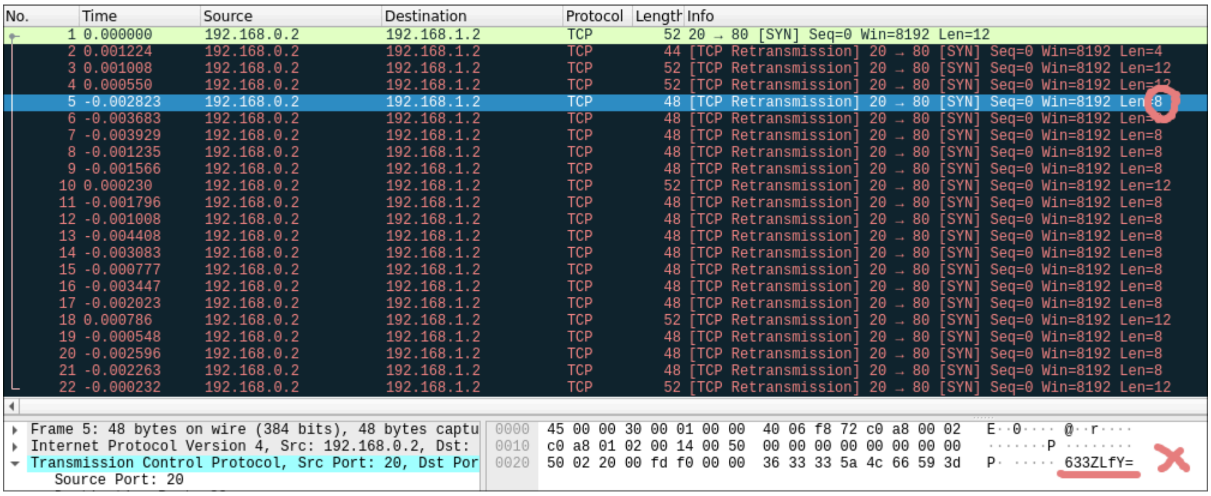


```
 1 0.000000   192.168.0.2   192.168.1.2   TCP   52 20 → 80 [SYN] Seq=0 Win=8192 Len=12
 2 0.001224   192.168.0.2   192.168.1.2   TCP   44 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=4
 3 0.001008   192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
 4 0.000550   192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
 5 -0.002823  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
 6 -0.003683  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
 7 -0.003929  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
 8 -0.001235  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
 9 -0.001566  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
10 0.000230   192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
11 -0.001796  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
12 -0.001008  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
13 -0.004408  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
14 -0.003083  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
15 -0.000777  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
16 -0.003447  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
17 -0.002023  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
18 0.000786   192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
19 -0.000548  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
20 -0.002596  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
21 -0.002263  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
22 -0.000232  192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
```

```
▸ Frame 3: 52 bytes on wire (416 bits), 52 bytes captu
▸ Internet Protocol Version 4, Src: 192.168.0.2, Dst:
▾ Transmission Control Protocol, Src Port: 20, Dst Por
     Source Port: 20
     Destination Port: 80
```

```
0000  45 00 00 34 00 01 00 00  40 06 f8 6e c0 a8 00 02   E··4···· @··n····
0010  c0 a8 01 02 00 14 00 50  00 00 00 00 00 00 00 00   ·······P ········
0020  50 02 20 00 2f 25 00 00  4e 47 49 31 4e 7a 6b 77   P· ·/%·· NGI1Nzkw
0030  4f 51 3d 3d                                        OQ==
```

# The following “base64” codes are not valid (length=8, not because we cannot find “==” at the end because the algorithm works without “=” or “==”, these are just a padding indicator, not part of the content. Padding is optional in many cases, but not always.)



```
No.  Time        Source        Destination   Protocol Length Info
 1 0.000000   192.168.0.2   192.168.1.2   TCP   52 20 → 80 [SYN] Seq=0 Win=8192 Len=12
 2 0.001224   192.168.0.2   192.168.1.2   TCP   44 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=4
 3 0.001008   192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
 4 0.000550   192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
 5 -0.002823  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
 6 -0.003683  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
 7 -0.003929  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
 8 -0.001235  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
 9 -0.001566  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
10 0.000230   192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
11 -0.001796  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
12 -0.001008  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
13 -0.004408  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
14 -0.003083  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
15 -0.000777  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
16 -0.003447  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
17 -0.002023  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
18 0.000786   192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
19 -0.000548  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
20 -0.002596  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
21 -0.002263  192.168.0.2   192.168.1.2   TCP   48 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=8
22 -0.000232  192.168.0.2   192.168.1.2   TCP   52 [TCP Retransmission] 20 → 80 [SYN] Seq=0 Win=8192 Len=12
```

```
▸ Frame 5: 48 bytes on wire (384 bits), 48 bytes captu
▸ Internet Protocol Version 4, Src: 192.168.0.2, Dst:
▾ Transmission Control Protocol, Src Port: 20, Dst Por
    Source Port: 20
```

```
0000  45 00 00 30 00 01 00 00  40 06 f8 72 c0 a8 00 02   E··0···· @··r····
0010  c0 a8 01 02 00 14 00 50  00 00 00 00 00 00 00 00   ·······P ········
0020  50 02 20 00 fd f0 00 00  36 33 33 5a 4c 66 59 3d   P······ 633ZLfY=
```

# We can run the following command to extract all the strings :

```
strings myNetworkTraffic.pcap
```

# The output :

```
ezF0X3c0cw==
fQ==
NGI1NzkwOQ==
XzM0c3lfdA==
633ZLfY=
k7ZdzLM=
.
.
.
PRH9csM=
cGljb0NURg==
```

# We are going to decode the following :

ezF0X3c0cw==
fQ==
NGI1NzkwOQ==
XzM0c3lfdA==
bnRfdGg0dA==
YmhfNHJfZA==
cGljb0NURg==

# After decode, we get the strings of the CTF and we have to arrange them.

THE FLAG : picoCTF{1t_w4snt_th4t_34sy_tbh_4r_d4b57909}
~Z4que