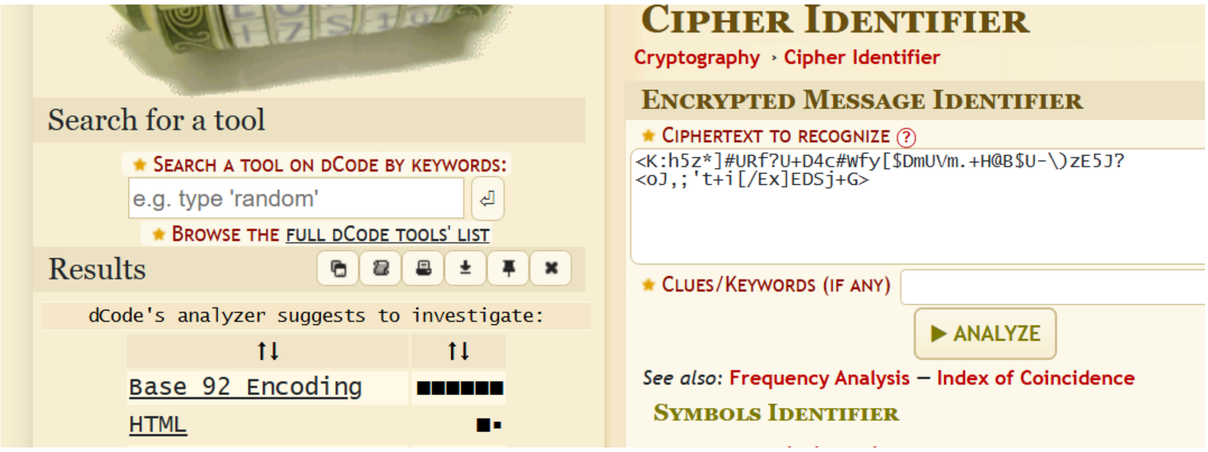
Substitute Teacher

After the 3 hints, there are 4 steps. Gunzip, base92, base45, gunzip. (This is after I understand the challenge description)

And this is before I understand. So I tried identifying it first but putting partial ciphertext into dcode identifier.



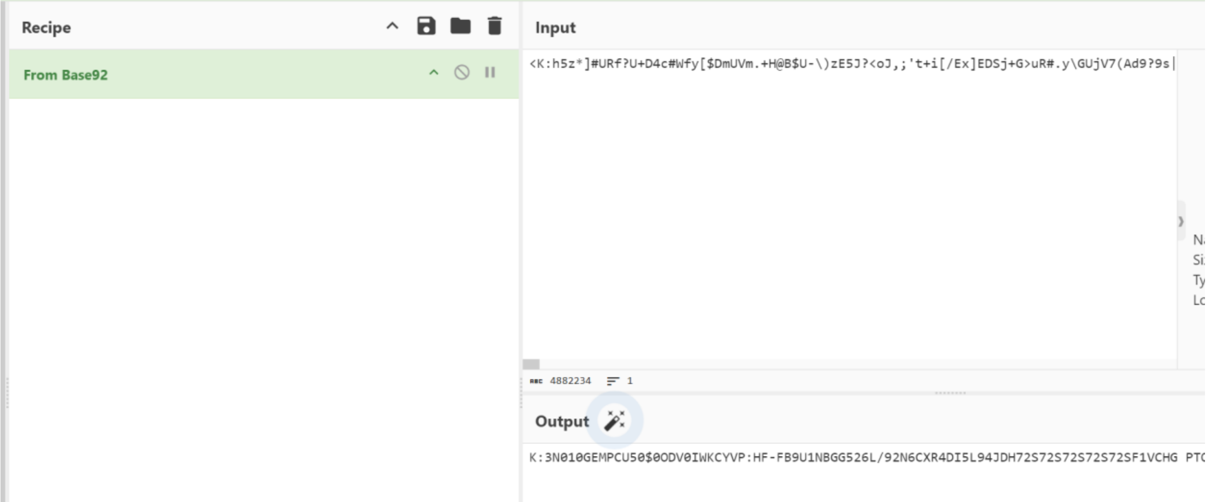
Search for a tool
★ Search a tool on dCode by keywords:
e.g. type 'random'
★ Browse the full dCode tools' list
Results
Cipher Identifier
Cryptography › Cipher Identifier
Encrypted Message Identifier
★ Ciphertext to recognize ?
★ Clues/Keywords (if any)
▶ ANALYZE
See also: Frequency Analysis – Index of Coincidence
Symbols Identifier

```
<K:h5z*]#URf?U+D4c#Wfy[$DmUVm.+H@B$U-\)zE5J?
<oJ,;'t+i[/Ex]EDSj+G>
```

dCode's analyzer suggests to investigate:

↑↓	↑↓
Base 92 Encoding	■■■■■■
HTML	■▪

Here I know its base92. Then I use cyberchef.

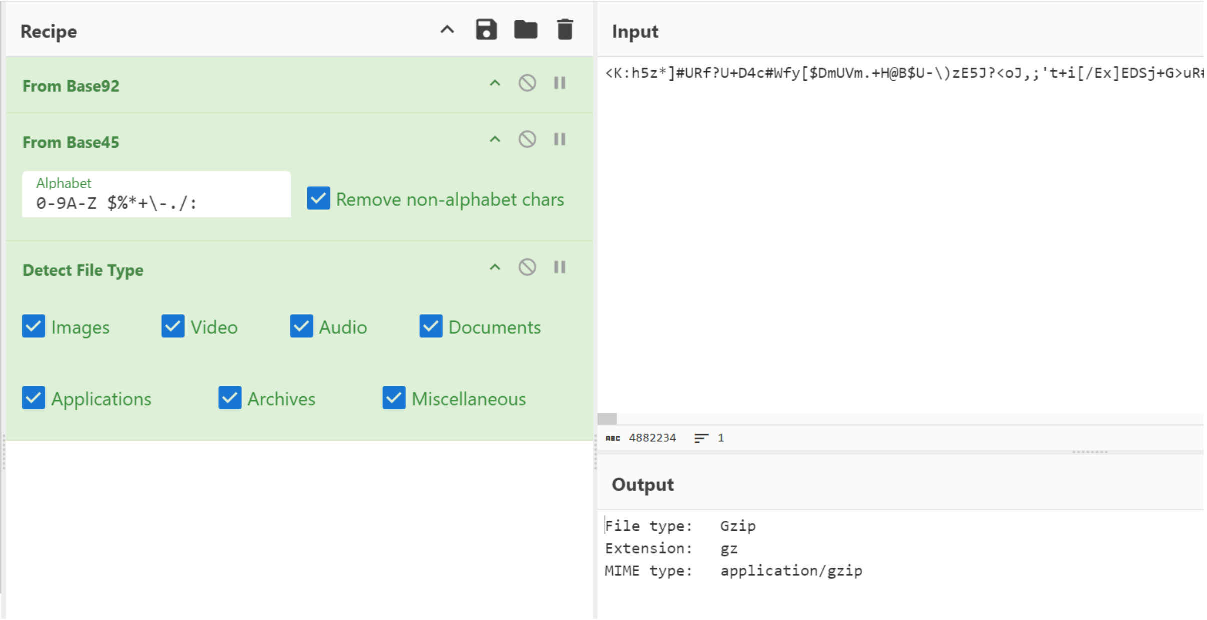


```
<K:h5z*]#URf?U+D4c#Wfy[$DmUVm.+H@B$U-\)zE5J?<oJ,;'t+i[/Ex]EDSj+G>uR#.y\GUjV7(Ad9?9s
```

```
4882234   1
```

```
K:3N010GEMPCU50$0ODV0IWKCYVP:HF-FB9U1NBGG526L/92N6CXR4DI5L94JDH72S72S72S72S72SF1VCHG PTC
```

U see the magic wand? I was like DAMN is this itt?? It gave base45 decode and detect file type. You don't know how happy I am to get the pcap file.



From Base92

From Base45

Alphabet
`0-9A-Z $%*+\-./:`

☑ Remove non-alphabet chars

Detect File Type

☑ Images ☑ Video ☑ Audio ☑ Documents

☑ Applications ☑ Archives ☑ Miscellaneous

Input

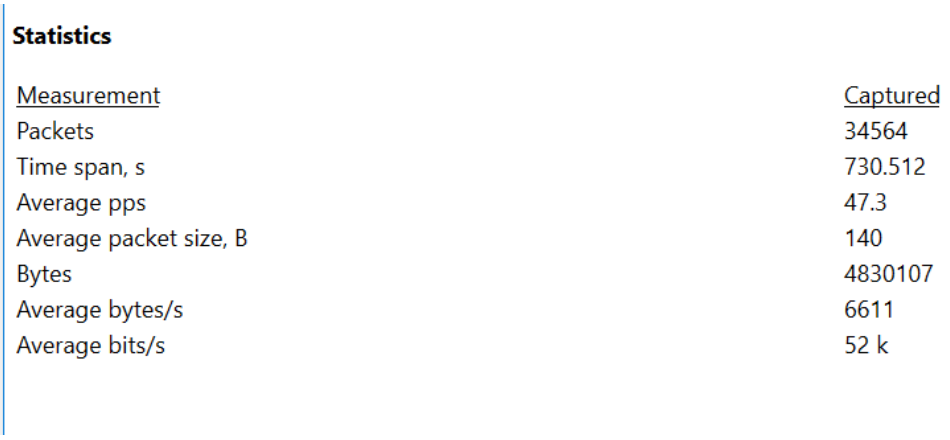
```
<K:h5z*]#URf?U+D4c#Wfy[$DmUVm.+H@B$U-\)zE5J?<oJ,;'t+i[/Ex]EDSj+G>uR
```

Output

```
File type:   Gzip
Extension:   gz
MIME type:   application/gzip
```

Back to the present. Now I got a pcap file after the 4 steps.

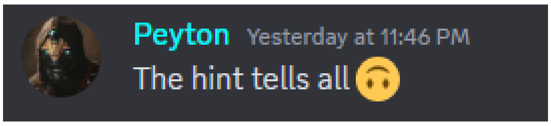
THERE ARE 30K PACKETS BROO



Statistics

Measurement	Captured
Packets	34564
Time span, s	730.512
Average pps	47.3
Average packet size, B	140
Bytes	4830107
Average bytes/s	6611
Average bits/s	52 k

Lets deduce the hints and challenge description



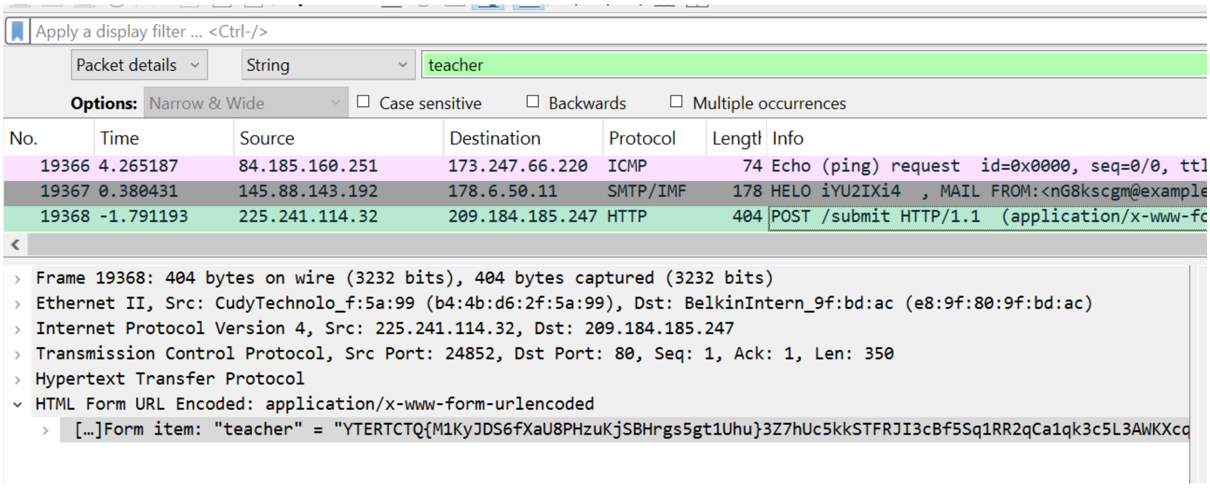
Peyton Yesterday at 11:46 PM
The hint tells all 🙃

Thanks to the hints, there are four parts to find to get the flag. One in each, HTTP, FTP, TCP, UDP.

It mentions about uppercase and lower case. It being the key to get flag. So, one ciphertext, three key. 3 keys? Which cipher use 3 keys? Welp just read this writeup.

Lastly the title mention about substitute teacher. So we know the cipher is substitute. For the teacher? Here.

First, CTRL+F teacher



Apply a display filter ... <Ctrl-/>

Packet details | String | teacher

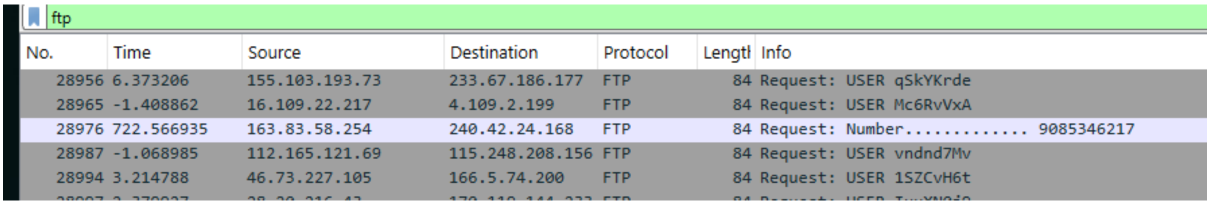
Options: Narrow & Wide ☐ Case sensitive ☐ Backwards ☐ Multiple occurrences

```
No.   Time      Source          Destination     Protocol  Lengtl Info
19366 4.265187  84.185.160.251  173.247.66.220  ICMP       74 Echo (ping) request  id=0x0000, seq=0/0, ttl
19367 0.380431  145.88.143.192  178.6.50.11     SMTP/IMF  178 HELO iYU2IXi4  , MAIL FROM:<nG8kscgm@example
19368 -1.791193 225.241.114.32  209.184.185.247 HTTP      404 POST /submit HTTP/1.1  (application/x-www-fc
```

```
> Frame 19368: 404 bytes on wire (3232 bits), 404 bytes captured (3232 bits)
> Ethernet II, Src: CudyTechnolo_f:5a:99 (b4:4b:d6:2f:5a:99), Dst: BelkinIntern_9f:bd:ac (e8:9f:80:9f:bd:ac)
> Internet Protocol Version 4, Src: 225.241.114.32, Dst: 209.184.185.247
> Transmission Control Protocol, Src Port: 24852, Dst Port: 80, Seq: 1, Ack: 1, Len: 350
> Hypertext Transfer Protocol
v HTML Form URL Encoded: application/x-www-form-urlencoded
  > […]Form item: "teacher" = "YTERTCTQ{M1KyJDS6fXaU8PHzuKjSBHrgs5gt1Uhu}3Z7hUc5kkSTFRJI3cBf5Sq1RR2qCa1qk3c5L3AWKXcq
```

You see that? It’s a ciphertext bro. Now we got one in http.

Second, ftp



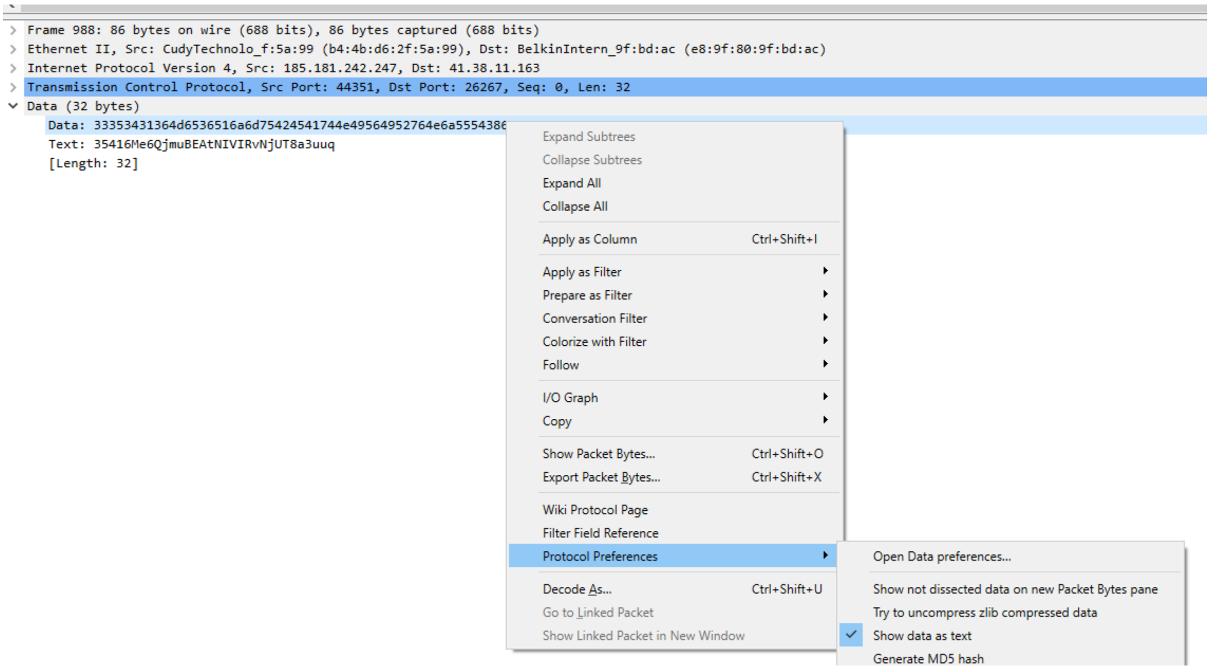
ftp

```
No.   Time       Source          Destination     Protocol Lengtl Info
28956 6.373206   155.103.193.73  233.67.186.177  FTP      84 Request: USER qSkYKrde
28965 -1.408862  16.109.22.217   4.109.2.199     FTP      84 Request: USER Mc6RvVxA
28976 722.566935 163.83.58.254   240.42.24.168   FTP      84 Request: Number............. 9085346217
28987 -1.068985  112.165.121.69  115.248.208.156 FTP      84 Request: USER vndnd7Mv
28994 3.214788   46.73.227.105   166.5.74.200    FTP      84 Request: USER 1SZCvH6t
```

Just scroll and you see the one and only.

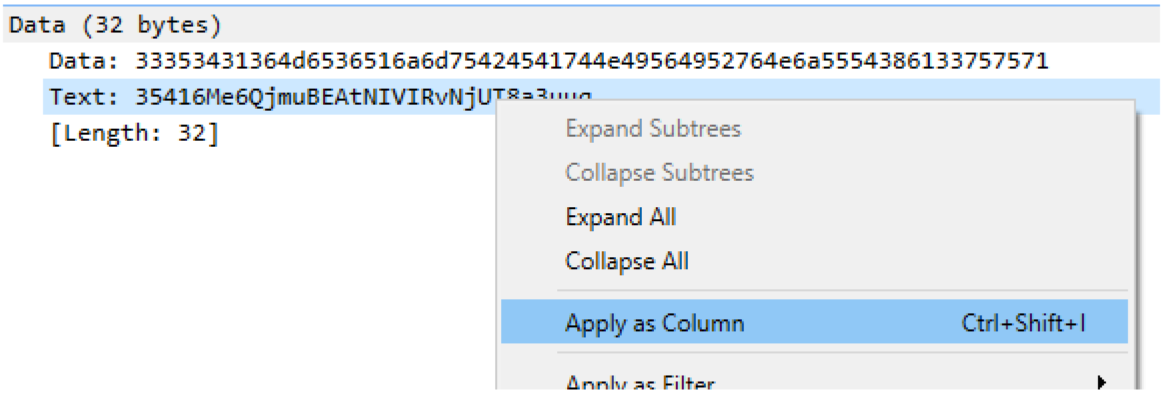
Third, TCP

We want to prepare the colomn first. Right click the Data -> protocol reference -> Show data as text

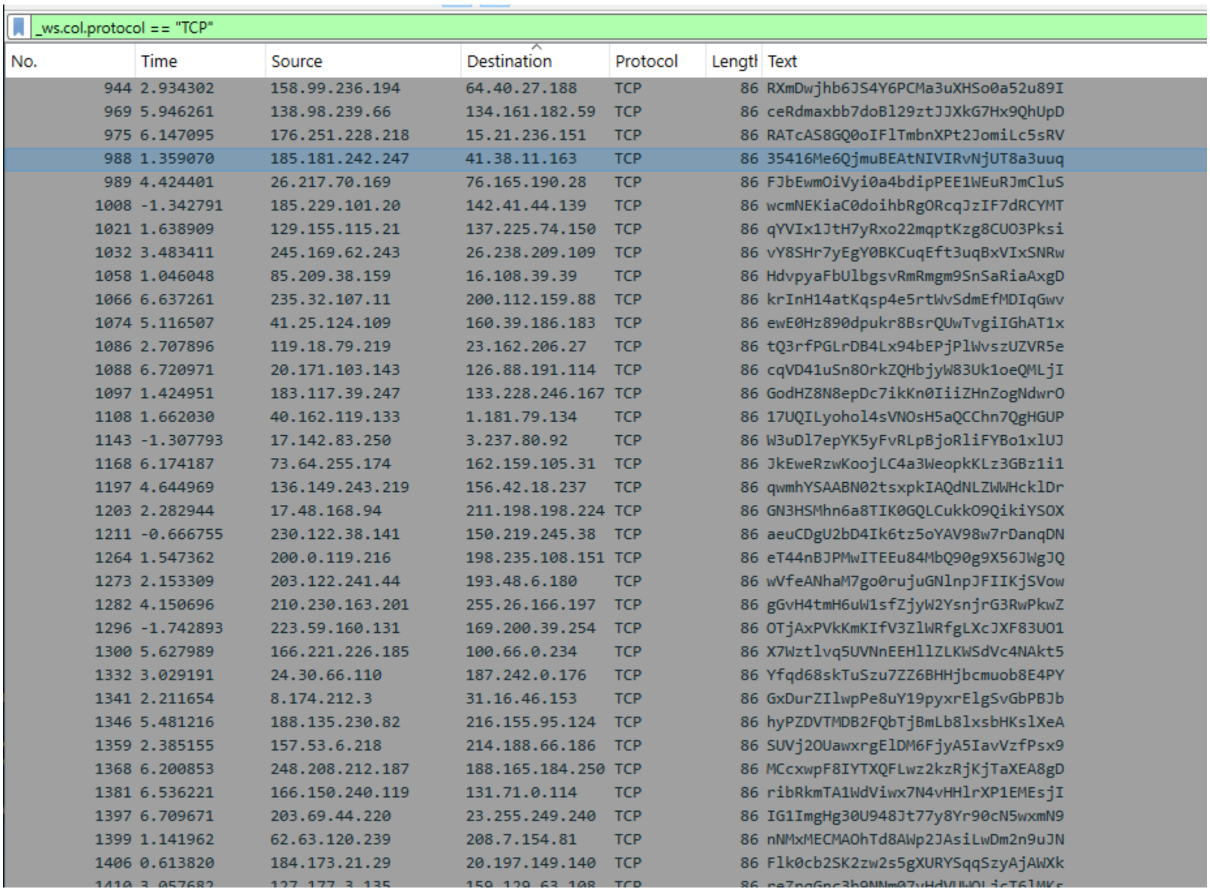


```
> Frame 988: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
> Ethernet II, Src: CudyTechnolo_f:5a:99 (b4:4b:d6:2f:5a:99), Dst: BelkinIntern_9f:bd:ac (e8:9f:80:9f:bd:ac)
> Internet Protocol Version 4, Src: 185.181.242.247, Dst: 41.38.11.163
> Transmission Control Protocol, Src Port: 44351, Dst Port: 26267, Seq: 0, Len: 32
v Data (32 bytes)
    Data: 33353431364d6536516a6d75424541744e49564952764e6a5554386
    Text: 35416Me6QjmuBEAtNIVIRvNjUT8a3uuq
    [Length: 32]
```

Then right click apply as column



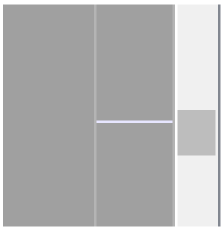
```
Data (32 bytes)
    Data: 33353431364d6536516a6d75424541744e49564952764e6a5554386133757571
    Text: 35416Me6QjmuBEAtNIVIRvNjUT8a3uuq
    [Length: 32]
```



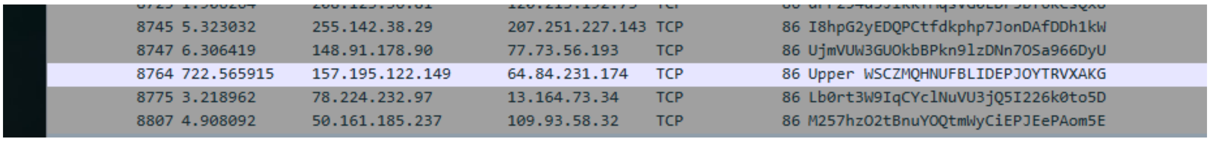
_ws.col.protocol == "TCP"

```
No.     Time        Source            Destination       Protocol  Lengtl Text
  944  2.934302   158.99.236.194    64.40.27.188      TCP       86 RXmDwjhb6JS4Y6PCMa3uXHSo0a52u89I
  969  5.946261   138.98.239.66     134.161.182.59    TCP       86 ceRdmaxbb7doBl29ztJJXkG7Hx9QhUpD
  975  6.147095   176.251.228.218   15.21.236.151     TCP       86 RATcAS8GQ0oIFlTmbnXPt2JomiLc5sRV
  988  1.359070   185.181.242.247   41.38.11.163      TCP       86 35416Me6QjmuBEAtNIVIRvNjUT8a3uuq
  989  4.424401   26.217.70.169     76.165.190.28     TCP       86 FJbEwmOiVyi0a4bdipPEE1WEuRJmCluS
 1008 -1.342791   185.229.101.20    142.41.44.139     TCP       86 wcmNEKiaC0doihbRgORcqJzIF7dRCYMT
 1021  1.638909   129.155.115.21    137.225.74.150    TCP       86 qYVIx1JtH7yRxo22mqptKzg8CUO3Pksi
 1032  3.483411   245.169.62.243    26.238.209.109    TCP       86 vY8SHr7yEgY0BKCuqEft3uqBxVIxSNRw
 1058  1.046048   85.209.38.159     16.108.39.39      TCP       86 HdvpyaFbUlbgsvRmRmgm9SnSaRiaAxgD
 1066  6.637261   235.32.107.11     200.112.159.88    TCP       86 krInH14atKqsp4e5rtWvSdmEfMDIqGwv
 1074  5.116507   41.25.124.109     160.39.186.183    TCP       86 ewE0Hz890dpukr8BsrQUwTvgiIGhAT1x
 1086  2.707896   119.18.79.219     23.162.206.27     TCP       86 tQ3rfPGLrDB4Lx94bEPjPlWvszUZVR5e
 1088  6.720971   20.171.103.143    126.88.191.114    TCP       86 cqVD41uSn8OrkZQHbjyW83Uk1oeQMLjI
 1097  1.424951   183.117.39.247    133.228.246.167   TCP       86 GodHZ8N8epDc7ikKn0IiiZHnZogNdwrO
 1108  1.662030   40.162.119.133    1.181.79.134      TCP       86 17UQILyohol4sVNOsH5aQCChn7QgHGUP
 1143 -1.307793   17.142.83.250     3.237.80.92       TCP       86 W3uDl7epYK5yFvRLpBjoRliFYBo1xlUJ
 1168  6.174187   73.64.255.174     162.159.105.31    TCP       86 JkEweRzwKoojLC4a3WeopkKLz3GBz1i1
 1197  4.644969   136.149.243.219   156.42.18.237     TCP       86 qwmhYSAABN02tsxpkIAQdNLZWWHcklDr
 1203  2.282944   17.48.168.94      211.198.198.224   TCP       86 GN3HSMhn6a8TIK0GQLCukkO9QikiYSOX
 1211 -0.666755   230.122.38.141    150.219.245.38    TCP       86 aeuCDgU2bD4Ik6tz5oYAV98w7rDanqDN
 1264  1.547362   200.0.119.216     198.235.108.151   TCP       86 eT44nBJPMwITEEu84MbQ90g9X56JWgJQ
 1273  2.153309   203.122.241.44    193.48.6.180      TCP       86 wVfeANhaM7go0rujuGNlnpJFIIKjSVow
 1282  4.150696   210.230.163.201   255.26.166.197    TCP       86 gGvH4tmH6uW1sfZjyW2YsnjrG3RwPkwZ
 1296 -1.742893   223.59.160.131    169.200.39.254    TCP       86 OTjAxPVkKmKIfV3ZlWRfgLXcJXF83UO1
 1300  5.627989   166.221.226.185   100.66.0.234      TCP       86 X7Wztlvq5UVNnEEHllZLKWSdVc4NAkt5
 1332  3.029191   24.30.66.110      187.242.0.176     TCP       86 Yfqd68skTuSzu7ZZ6BHHjbcmuob8E4PY
 1341  2.211654   8.174.212.3       31.16.46.153      TCP       86 GxDurZIlwpPe8uY19pyxrElgSvGbPBJb
 1346  5.481216   188.135.230.82    216.155.95.124    TCP       86 hyPZDVTMDB2FQbTjBmLb8lxsbHKslXeA
 1359  2.385155   157.53.6.218      214.188.66.186    TCP       86 SUVj2OUawxrgElDM6FjyA5IavVzfPsx9
 1368  6.200853   248.208.212.187   188.165.184.250   TCP       86 MCcxwpF8IYTXQFLwz2kzRjKjTaXEA8gD
 1381  6.536221   166.150.240.119   131.71.0.114      TCP       86 ribRkmTA1WdViwx7N4vHHlrXP1EMEsjI
 1397  6.709671   203.69.44.220     23.255.249.240    TCP       86 IG1ImgHg30U948Jt77y8Yr90cN5wxmN9
 1399  1.141962   62.63.120.239     208.7.154.81      TCP       86 nNMxMECMAOhTd8AWp2JAsiLwDm2n9uJN
 1406  0.613820   184.173.21.29     20.197.149.140    TCP       86 Flk0cb2SK2zw2s5gXURYSqqSzyAjAWXk
 1410  3.057682   127.177.3.135     159.129.63.108    TCP       86 re7ngGnc3h9NNm07vHdVUWOLjcT6lMKs
```

Here you want to find all lowercase or all uppercase as the challenge description said about it. Then find the odd one. Looking at the scroll button there's and odd one tho. At first I thought I found the cheesy wy but the challenge creator said that is the intended solution.



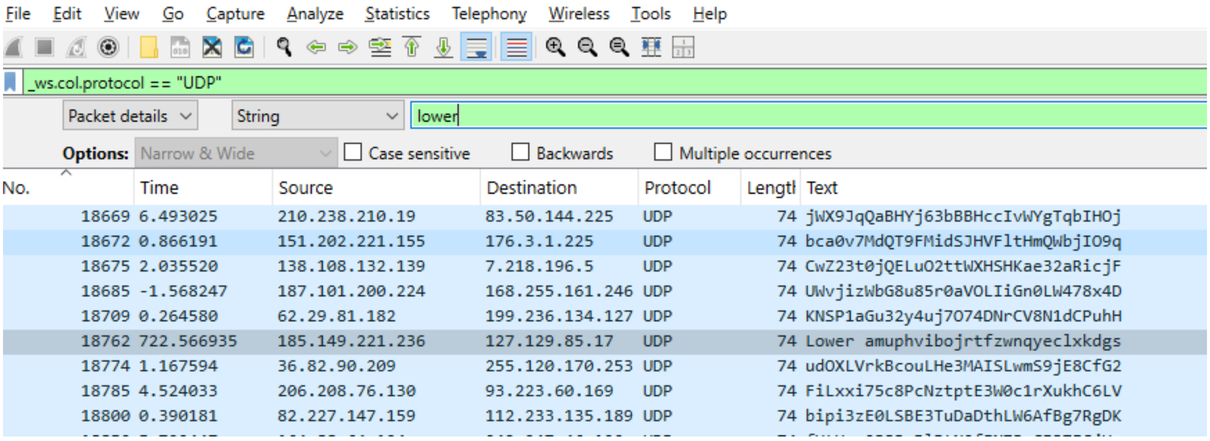
*the scroll bar. Got white line



```
8745 5.323032    255.142.38.29    207.251.227.143 TCP    86 I8hpG2yEDQPCtfdkphp7JonDAfDDh1kW
8747 6.306419    148.91.178.90    77.73.56.193    TCP    86 UjmVUW3GUOkbBPkn9lzDNn7OSa966DyU
8764 722.565915  157.195.122.149  64.84.231.174   TCP    86 Upper WSCZMQHNUFBLIDEPJOYTRVXAKG
8775 3.218962    78.224.232.97    13.164.73.34    TCP    86 Lb0rt3W9IqCYclNuVU3jQ5I226k0to5D
8807 4.908092    50.161.185.237   109.93.58.32    TCP    86 M257hzO2tBnuYOQtmWyCiEPJEePAom5E
```

So now we got Upper, so other one ofcourse lower

Fourth, find Lower in udp



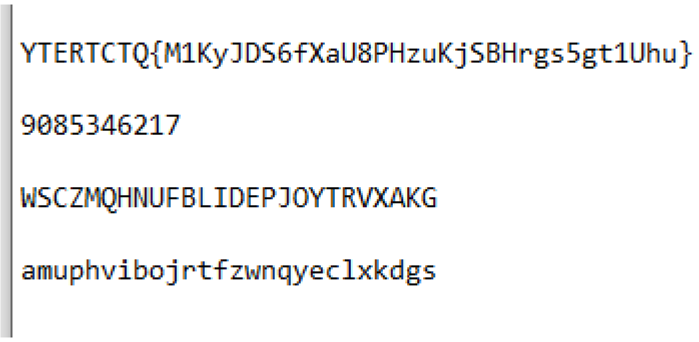
_ws.col.protocol == "UDP"

Packet details | String | lower

Options: Narrow & Wide □ Case sensitive □ Backwards □ Multiple occurrences

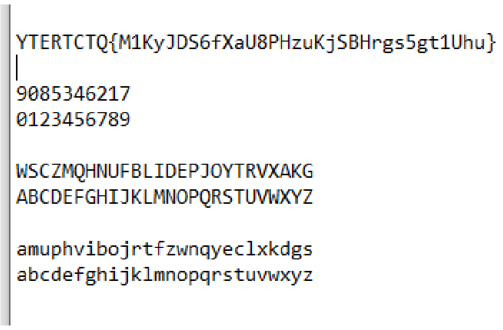
```
No.      Time        Source           Destination      Protocol  Length Text
   18669 6.493025    210.238.210.19   83.50.144.225    UDP       74 jWX9JqQaBHYj63bBBHccIvWYgTqbIHOj
   18672 0.866191    151.202.221.155  176.3.1.225      UDP       74 bca0v7MdQT9FMidSJHVFltHmQWbjIO9q
   18675 2.035520    138.108.132.139  7.218.196.5      UDP       74 CwZ23t0jQELuO2ttWXHSHKae32aRicjF
   18685 -1.568247   187.101.200.224  168.255.161.246  UDP       74 UWvjizWbG8u85r0aVOLIiGn0LW478x4D
   18709 0.264580    62.29.81.182     199.236.134.127  UDP       74 KNSP1aGu32y4uj7O74DNrCV8N1dCPuhH
   18762 722.566935  185.149.221.236  127.129.85.17    UDP       74 Lower amuphvibojrtfzwnqyeclxkdgs
   18774 1.167594    36.82.90.209     255.120.170.253  UDP       74 udOXLVrkBcouLHe3MAISLwmS9jE8CfG2
   18785 4.524033    206.208.76.130   93.223.60.169    UDP       74 FiLxxi75c8PcNztptE3W0c1rXukhC6LV
   18800 0.390181    82.227.147.159   112.233.135.189  UDP       74 bipi3zE0LSBE3TuDaDthLW6AfBg7RgDK
```

Not we got all four parts



```
YTERTCTQ{M1KyJDS6fXaU8PHzuKjSBHrgs5gt1Uhu}

9085346217

WSCZMQHNUFBLIDEPJOYTRVXAKG

amuphvibojrtfzwnqyeclxkdgs
```

After 10 minutes staring at this note, and comparing YTERTCTQ to STOUTCTF, I got like this.



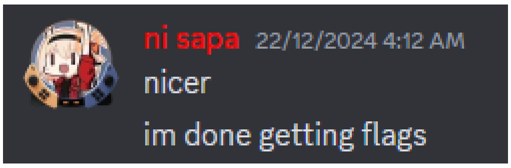
```
YTERTCTQ{M1KyJDS6fXaU8PHzuKjSBHrgs5gt1Uhu}

9085346217
0123456789

WSCZMQHNUFBLIDEPJOYTRVXAKG
ABCDEFGHIJKLMNOPQRSTUVWXYZ

amuphvibojrtfzwnqyeclxkdgs
abcdefghijklmnopqrstuvwxyz
```

Does it make sense to you? For me yes. For example, we take a look at letter Y. uppercase Y is in the same number position for S in alphabetical order. Same with other flag format. Letter E is at O position. So now its confirm my deduction is correct.

I manually map this and not using any script cause im a noob. Eventually, I got:

STOUTCTF{E8YrQNB6mWaI2PGncYjBKGkyz3yl8Iec}

Solved the hardest challenge at 4am and im still hoarding till now (im writing this writeup rn when Im doing malware challenge). Only Peyton know about this :P



ni sapa 22/12/2024 4:12 AM

nicer

im done getting flags

All flag got except malware and the last web. This one is valuable. I cant let anyone know I got this flag.

*OSIRIS got it when Im struggling doing malware :’(. Its okay ill try again next time.