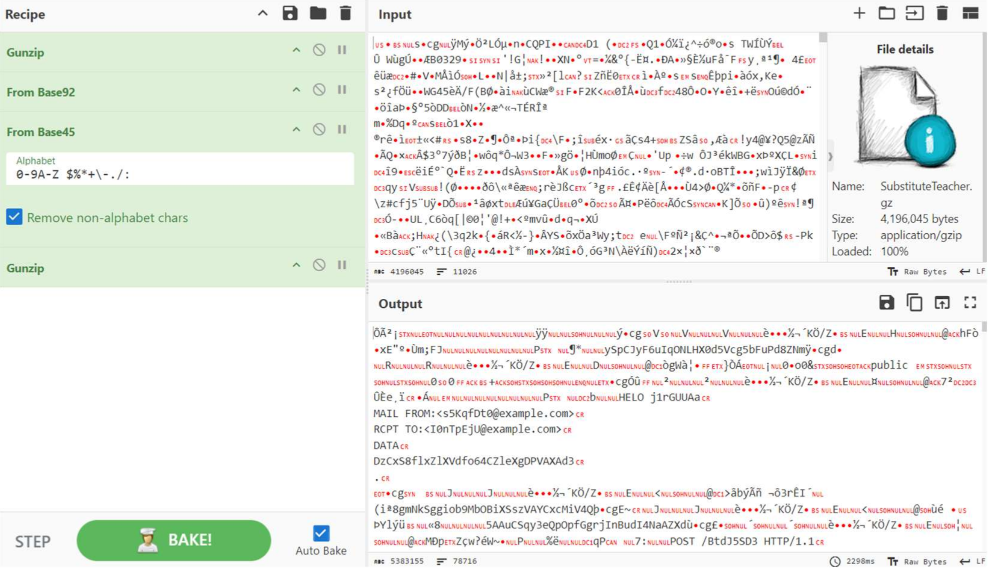
SUBSTITUTE TEACHER

Description

The year is 1992, a few weeks after the fall of the Soviet Union. Amidst the chaos, a group of 45 rogue operatives known as "The Teachers" were tasked with safeguarding classified files. Their mission? To ensure these secrets stayed hidden from prying eyes. To achieve this, they devised a series of intricate steps to obscure their plans.

Your mission is to recover the hidden flag from their encrypted communication. The operatives left all the tools you need in the provided file, but they didn't make it easy. They relied on meticulous precision, where every detail—big or small, uppercase or lowercase—could hold the key to unlocking their secrets. Can you decipher their layers of secrecy and reveal the hidden truth?

The hint referenced '1992' and '45 Rogue', which pointed to Base92 and Base45 encoding. The process involved: Gunzip > Base92 > Base45 > Gunzip. After following these steps, plaintext wording was revealed, allowing me to proceed with downloading and saving the file.



Recipe
- Gunzip
- From Base92
- From Base45
  - Alphabet: 0-9A-Z $%*+\-./:
  - ☑ Remove non-alphabet chars
- Gunzip

File details
Name: SubstituteTeacher.gz
Size: 4,196,045 bytes
Type: application/gzip
Loaded: 100%

Input

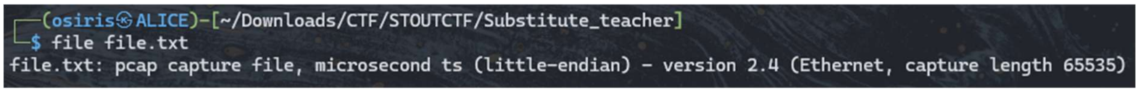
4196045 11026

Output

```
HELO j1rGUUAa
MAIL FROM:<s5KqfDt0@example.com>
RCPT TO:<I0nTpEjU@example.com>
DATA
DzCxS8flxZlXVdfo64CZleXgDPVAXAd3
.
POST /BtdJ5SD3 HTTP/1.1
```

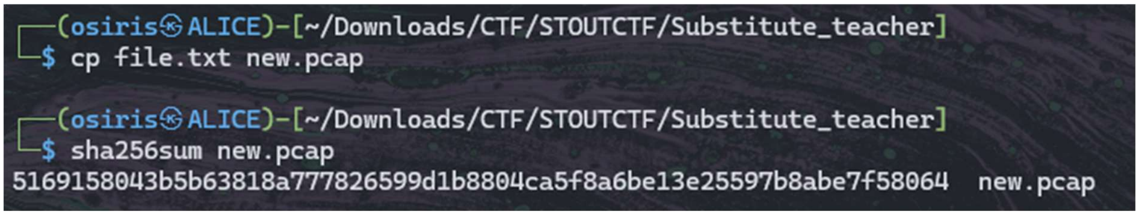
5383155 78716 2298ms

Checking the file type. It was pcap.



```
┌──(osiris㉿ALICE)-[~/Downloads/CTF/STOUTCTF/Substitute_teacher]
└─$ file file.txt
file.txt: pcap capture file, microsecond ts (little-endian) - version 2.4 (Ethernet, capture length 65535)
```

Rename the extension into pcap



```
┌──(osiris㉿ALICE)-[~/Downloads/CTF/STOUTCTF/Substitute_teacher]
└─$ cp file.txt new.pcap

┌──(osiris㉿ALICE)-[~/Downloads/CTF/STOUTCTF/Substitute_teacher]
└─$ sha256sum new.pcap
5169158043b5b63818a777826599d1b8804ca5f8a6be13e25597b8abe7f58064  new.pcap
```

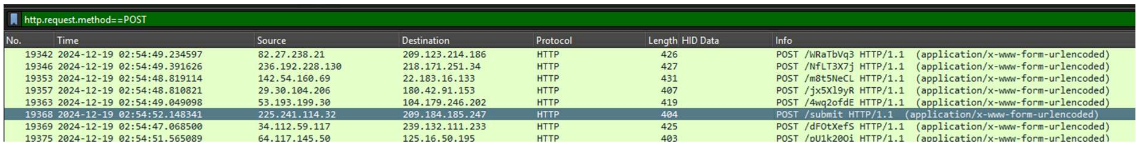
Hash (SHA256)
5169158043b5b63818a777826599d1b8804ca5f8a6be13e25597b8abe7f58064

HTTP

Filter

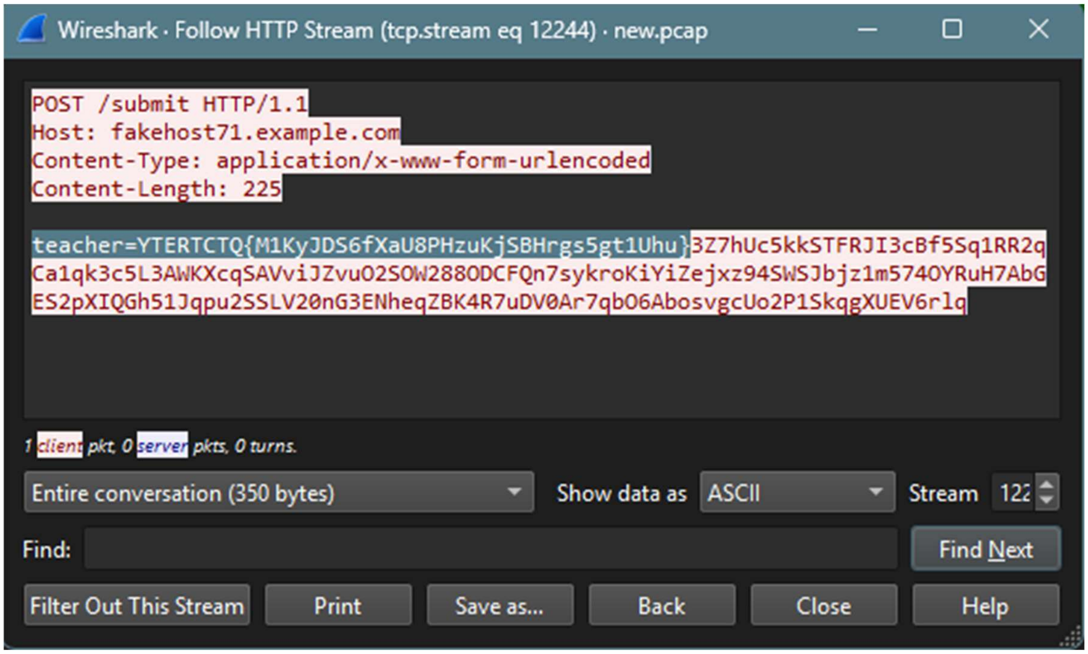
`http.request.method==POST`

Packet 19368:



```
http.request.method==POST
No.   Time                        Source           Destination       Protocol  Length HID Data  Info
19342 2024-12-19 02:54:49.234597  82.27.238.21     209.123.214.186   HTTP      426              POST /WRaTbVq3 HTTP/1.1  (application/x-www-form-urlencoded)
19346 2024-12-19 02:54:49.391626  236.192.228.130  218.171.251.34    HTTP      427              POST /NfLT3X7j HTTP/1.1  (application/x-www-form-urlencoded)
19353 2024-12-19 02:54:48.819114  142.54.160.69    22.183.16.133     HTTP      431              POST /m8t5NeCL HTTP/1.1  (application/x-www-form-urlencoded)
19357 2024-12-19 02:54:48.810821  29.30.104.206    180.42.91.153     HTTP      407              POST /jx5Xl9yR HTTP/1.1  (application/x-www-form-urlencoded)
19363 2024-12-19 02:54:49.049098  53.193.199.30    104.179.246.202   HTTP      419              POST /4wq2ofdE HTTP/1.1  (application/x-www-form-urlencoded)
19368 2024-12-19 02:54:52.148341  225.241.114.32   209.184.185.247   HTTP      404              POST /submit HTTP/1.1  (application/x-www-form-urlencoded)
19369 2024-12-19 02:54:47.068500  34.112.59.117    239.132.111.233   HTTP      425              POST /dFOtXefS HTTP/1.1  (application/x-www-form-urlencoded)
19375 2024-12-19 02:54:51.565089  64.117.145.50    125.16.50.195     HTTP      403              POST /pU1k20Oi HTTP/1.1  (application/x-www-form-urlencoded)
```

Right Click > Follow > HTTP Stream



Wireshark · Follow HTTP Stream (tcp.stream eq 12244) · new.pcap

```
POST /submit HTTP/1.1
Host: fakehost71.example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 225

teacher=YTERTCTQ{M1KyJDS6fXaU8PHzuKjSBHrgs5gt1Uhu}3Z7hUc5kkSTFRJI3cBf5Sq1RR2q
Ca1qk3c5L3AWKXcqSAVviJZvuO2SOW288ODCFQn7sykroKiYiZejxz94SWSJbjz1m574OYRuH7AbG
ES2pXIQGh51Jqpu2SSLV20nG3ENheqZBK4R7uDV0Ar7qbO6AbosvgcUo2P1SkqgXUEV6rlq
```

1 client pkt, 0 server pkts, 0 turns.

Entire conversation (350 bytes)

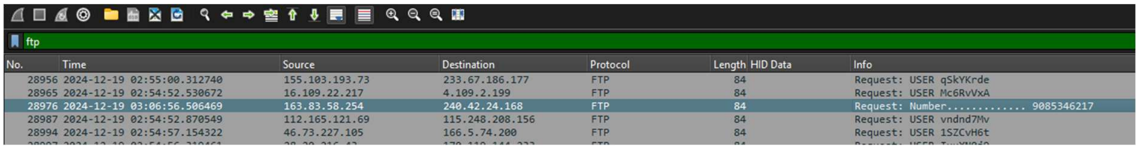
Show data as: ASCII

Stream: 122

Find:

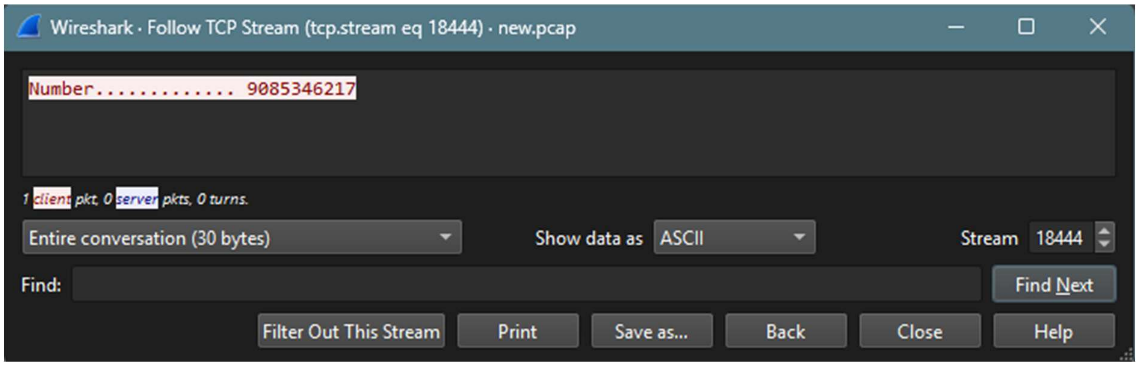
FTP

Packet: 28976



```
ftp
No.   Time                        Source           Destination       Protocol  Length HID Data  Info
28956 2024-12-19 02:55:00.312740  155.103.193.73   233.67.186.177    FTP       84               Request: USER qSkYKrde
28965 2024-12-19 02:54:52.530672  16.109.22.217    4.109.2.199       FTP       84               Request: USER Mc6RvVxA
28976 2024-12-19 03:06:56.506469  163.83.58.254    240.42.24.168     FTP       84               Request: Number............. 9085346217
28987 2024-12-19 02:54:52.870549  112.165.121.69   115.248.208.156   FTP       84               Request: USER vndnd7Mv
28994 2024-12-19 02:54:57.154322  46.73.227.105    166.5.74.200      FTP       84               Request: USER 1SZCvH6t
```

Right Click > Follow > Follow TCP Stream



Wireshark · Follow TCP Stream (tcp.stream eq 18444) · new.pcap

```
Number............. 9085346217
```

1 client pkt, 0 server pkts, 0 turns.

Entire conversation (30 bytes)

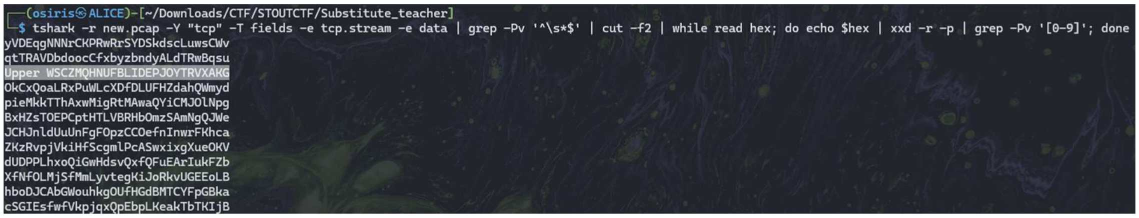
Show data as: ASCII

Stream: 18444

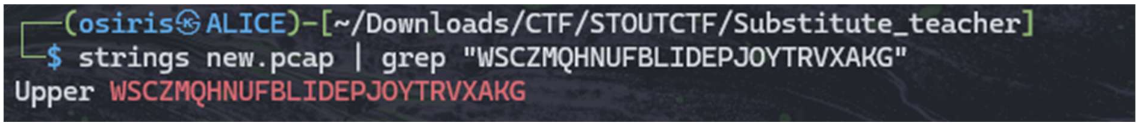
Find:

TCP

```
$
tshark -r new.pcap -Y "tcp" -T fields -e tcp.stream -e data | grep -Pv '^\s*$' |
cut -f2 | while read hex; do echo $hex | xxd -r -p | grep -Pv '[0-9]'; done
```



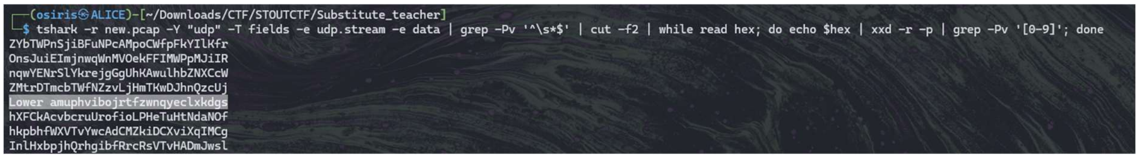
```
┌──(osiris㉿ALICE)-[~/Downloads/CTF/STOUTCTF/Substitute_teacher]
└─$ tshark -r new.pcap -Y "tcp" -T fields -e tcp.stream -e data | grep -Pv '^\s*$' | cut -f2 | while read hex; do echo $hex | xxd -r -p | grep -Pv '[0-9]'; done
yVDEqgNNNrCKPRwRrSYDSkdscLuwsCWv
qtTRAVDbdoocCfxbyzbndyALdTRwBqsu
Upper WSCZMQHNUFBLIDEPJOYTRVXAKG
OkCxQoaLRxPuWLcXDfDLUFHZdahQWmyd
pieMkkTThAxwMigRtMAwaQYiCMJOlNpg
BxHZsTOEPCptHTLVBRHbOmzSAmNgQJWe
JCHJnldUuUnFgFOpzCCOefnInwrFKhca
ZKzRvpjVkiHfScgmlPcASwxixgXueOKV
dUDPPLhxoQiGwHdsvQxfQFuEArIukFZb
XfNfOLMjSfMmLyvtegKiJoRkvUGEEoLB
hboDJCAbGWouhkgOUfHGdBMTCYFpGBka
cSGIEsfwfVkpjqxQpEbpLKeakTbTKIjB
```



```
┌──(osiris㉿ALICE)-[~/Downloads/CTF/STOUTCTF/Substitute_teacher]
└─$ strings new.pcap | grep "WSCZMQHNUFBLIDEPJOYTRVXAKG"
Upper WSCZMQHNUFBLIDEPJOYTRVXAKG
```

UDP

```
$
tshark -r new.pcap -Y "udp" -T fields -e udp.stream -e data | grep -Pv '^\s*$' |
cut -f2 | while read hex; do echo $hex | xxd -r -p | grep -Pv '[0-9]'; done
```



```
┌──(osiris㉿ALICE)-[~/Downloads/CTF/STOUTCTF/Substitute_teacher]
└─$ tshark -r new.pcap -Y "udp" -T fields -e udp.stream -e data | grep -Pv '^\s*$' | cut -f2 | while read hex; do echo $hex | xxd -r -p | grep -Pv '[0-9]'; done
ZYbTWPnSjiBFuNPcAMpoCWfpFkYIlKfr
OnsJuiEImjnwqWnMVOekFFIMWPpMJiIR
nqwYENrSlYkrejgGgUhKAwulhbZNXCcW
ZMtrDTmcbTWfNZzvLjHmTKwDJhnQzcUj
Lower amuphvibojrtfzwnqyeclxkdgs
hXFCkAcvbcruUrofioLPHeTuHtNdaNOf
hkpbhfWXVTvYwcAdCMZkiDCXviXqIMCg
InlHxbpjhQrhgibfRrcRsVTvHADmJwsl
```

DECRYPTING EVIDENCE

```
HTTP:
teacher=YTERTCTQ{M1KyJDS6fXaU8PHzuKjSBHrgs5gt1Uhu}

FTP:
Number............. 9085346217

TCP:
packet: 8764
Upper WSCZMQHNUFBLIDEPJOYTRVXAKG

UDP:
Lower amuphvibojrtfzwnqyeclxkdgs
```

After spending considerable time analyzing the encoded text and reviewing the description and hints, I began to understand that the process involved mapping encoded letters and numbers back to their original forms. The cipher relied on meticulous precision, where every detail—whether uppercase or lowercase—was important. For instance, the capital letter 'Y' in the encoded text corresponds to 'S' in the original mapping. Similarly, the capital 'T' remains 'T' after decoding, as observed when comparing the ciphered and standard alphabets. Made a script out of it to decode it.

Script

```
upper_cipher = "WSCZMQHNUFBLIDEPJOYTRVXAKG"
lower_cipher = "amuphvibojrtfzwnqyeclxkdgs"

standard_upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
standard_lower = "abcdefghijklmnopqrstuvwxyz"

numeric_key = "9085346217"

encoded_text = "YTERTCTQ{M1KyJDS6fXaU8PHzuKjSBHrgs5gt1Uhu}"

def decode(encoded, upper_map, lower_map, num_map):
    result = []
    for char in encoded:
        if char in upper_map:
            index = upper_map.index(char)
            result.append(standard_upper[index])
        elif char in lower_map:
            index = lower_map.index(char)
            result.append(standard_lower[index])
        elif char in num_map:
            index = num_map.index(char)
            result.append(str(index))
        else:
            result.append(char)
    return ''.join(result)

flag = decode(encoded_text, upper_cipher, lower_cipher, numeric_key)
print(flag)
```

Flag	STOUTCTF{E8YrQNB6mWaI2PGncYjBKGkyz3yl8Iec}