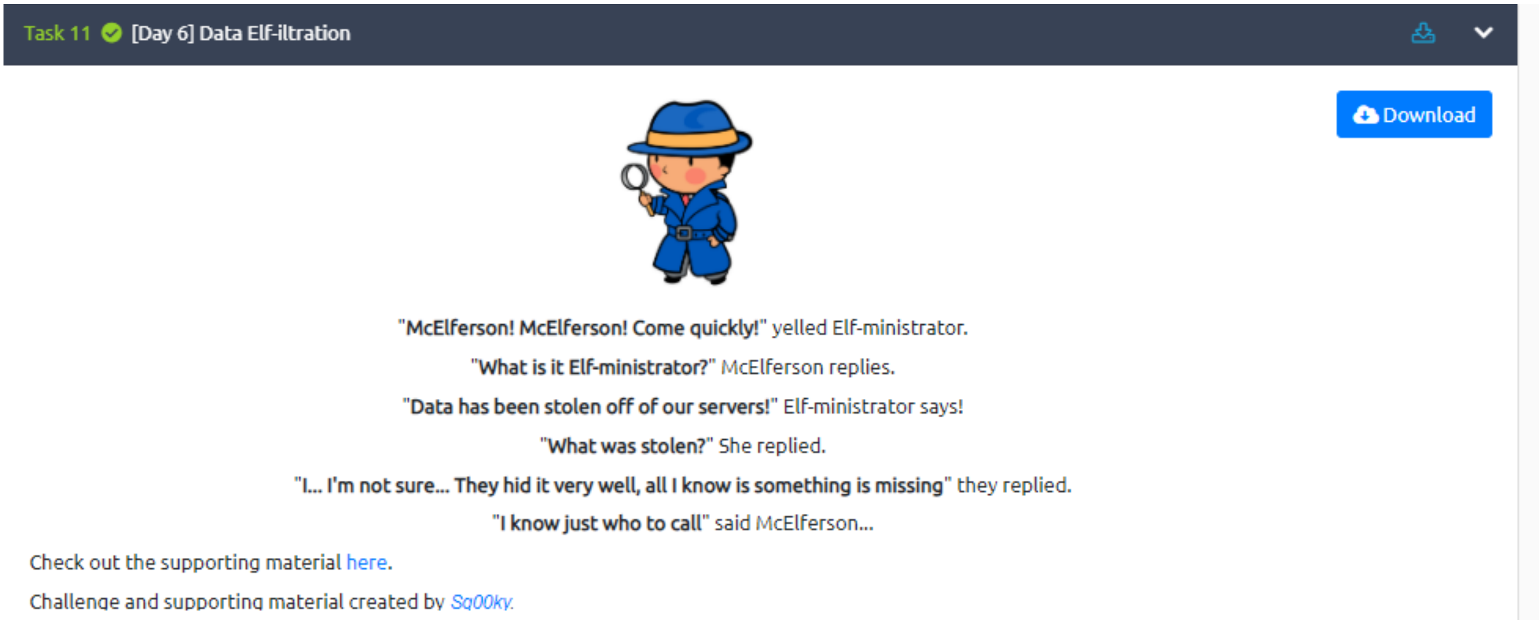
Day 6 - Data Elf-iltration

Scenario



Task 11 [Day 6] Data Elf-iltration

Download

"McElferson! McElferson! Come quickly!" yelled Elf-ministrator.

"What is it Elf-ministrator?" McElferson replies.

"Data has been stolen off of our servers!" Elf-ministrator says!

"What was stolen?" She replied.

"I... I'm not sure... They hid it very well, all I know is something is missing" they replied.

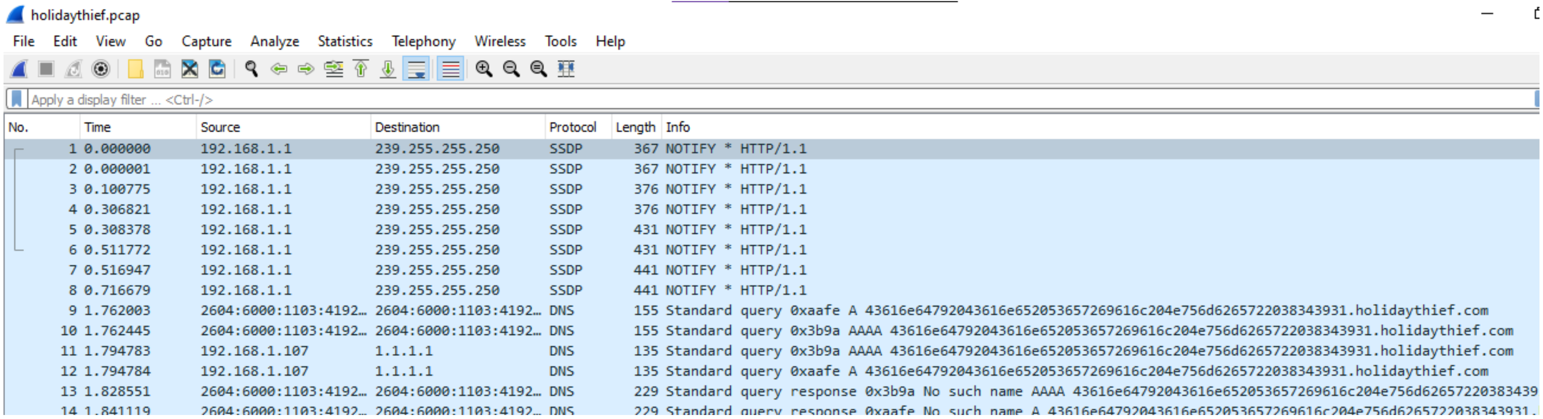
"I know just who to call" said McElferson...

Check out the supporting material here.

Challenge and supporting material created by Sq00ky.

it seems like the data has been stolen here, & we've the network capture packet file let's use the wireshark to examine it

the holidaythief.pcap file content

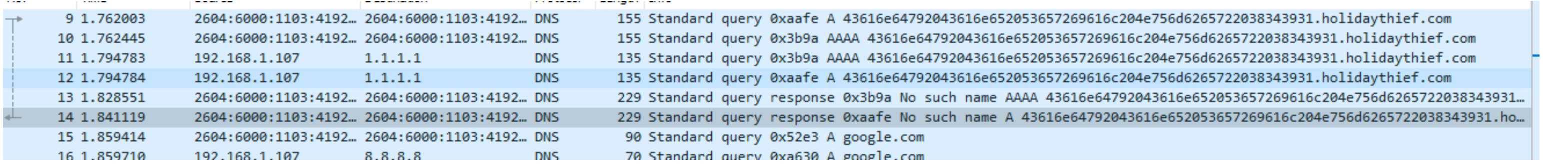


holidaythief.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time      Source                 Destination            Protocol Length Info
    1 0.000000  192.168.1.1            239.255.255.250        SSDP     367 NOTIFY * HTTP/1.1
    2 0.000001  192.168.1.1            239.255.255.250        SSDP     367 NOTIFY * HTTP/1.1
    3 0.100775  192.168.1.1            239.255.255.250        SSDP     376 NOTIFY * HTTP/1.1
    4 0.306821  192.168.1.1            239.255.255.250        SSDP     376 NOTIFY * HTTP/1.1
    5 0.308378  192.168.1.1            239.255.255.250        SSDP     431 NOTIFY * HTTP/1.1
    6 0.511772  192.168.1.1            239.255.255.250        SSDP     431 NOTIFY * HTTP/1.1
    7 0.516947  192.168.1.1            239.255.255.250        SSDP     441 NOTIFY * HTTP/1.1
    8 0.716679  192.168.1.1            239.255.255.250        SSDP     441 NOTIFY * HTTP/1.1
    9 1.762003  2604:6000:1103:4192…   2604:6000:1103:4192…   DNS      155 Standard query 0xaafe A 43616e64792043616e652053657269616c204e756d6265722038343931.holidaythief.com
   10 1.762445  2604:6000:1103:4192…   2604:6000:1103:4192…   DNS      155 Standard query 0x3b9a AAAA 43616e64792043616e652053657269616c204e756d6265722038343931.holidaythief.com
   11 1.794783  192.168.1.107          1.1.1.1                DNS      135 Standard query 0x3b9a AAAA 43616e64792043616e652053657269616c204e756d6265722038343931.holidaythief.com
   12 1.794784  192.168.1.107          1.1.1.1                DNS      135 Standard query 0xaafe A 43616e64792043616e652053657269616c204e756d6265722038343931.holidaythief.com
   13 1.828551  2604:6000:1103:4192…   2604:6000:1103:4192…   DNS      229 Standard query response 0x3b9a No such name AAAA 43616e64792043616e652053657269616c204e756d62657220383439
   14 1.841119  2604:6000:1103:4192…   2604:6000:1103:4192…   DNS      229 Standard query response 0xaafe No such name A 43616e64792043616e652053657269616c204e756d6265722038343931…
```

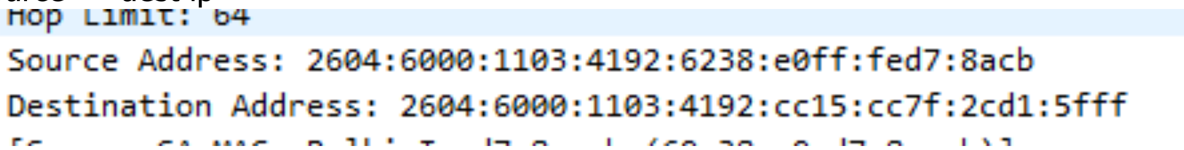
let's check the DNS queries & see whether the threat actor used this data exfiltration technique or not

if we can see here we found something suspicious here especially the dns query here ...holidaythief.com ?



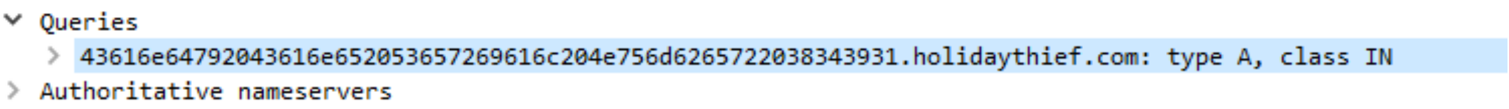
```
    9 1.762003  2604:6000:1103:4192…   2604:6000:1103:4192…   DNS      155 Standard query 0xaafe A 43616e64792043616e652053657269616c204e756d6265722038343931.holidaythief.com
   10 1.762445  2604:6000:1103:4192…   2604:6000:1103:4192…   DNS      155 Standard query 0x3b9a AAAA 43616e64792043616e652053657269616c204e756d6265722038343931.holidaythief.com
   11 1.794783  192.168.1.107          1.1.1.1                DNS      135 Standard query 0x3b9a AAAA 43616e64792043616e652053657269616c204e756d6265722038343931.holidaythief.com
   12 1.794784  192.168.1.107          1.1.1.1                DNS      135 Standard query 0xaafe A 43616e64792043616e652053657269616c204e756d6265722038343931.holidaythief.com
   13 1.828551  2604:6000:1103:4192…   2604:6000:1103:4192…   DNS      229 Standard query response 0x3b9a No such name AAAA 43616e64792043616e652053657269616c204e756d6265722038343931…
   14 1.841119  2604:6000:1103:4192…   2604:6000:1103:4192…   DNS      229 Standard query response 0xaafe No such name A 43616e64792043616e652053657269616c204e756d6265722038343931.ho…
   15 1.859414  2604:6000:1103:4192…   2604:6000:1103:4192…   DNS       90 Standard query 0x52e3 A google.com
   16 1.859710  192.168.1.107          8.8.8.8                DNS       70 Standard query 0xa630 A google.com
```

source -> dest ip



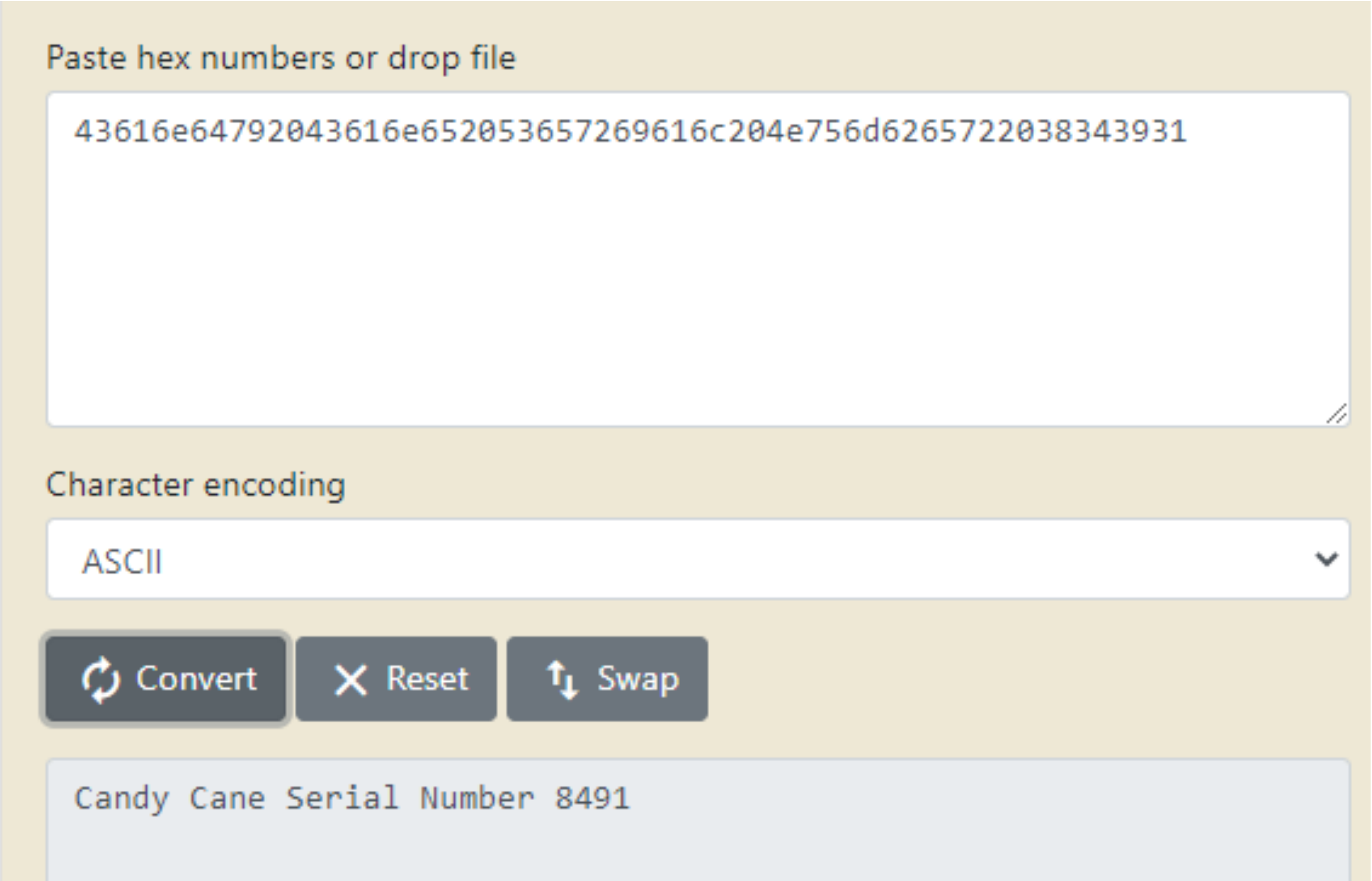
```
Hop Limit: 64
Source Address: 2604:6000:1103:4192:6238:e0ff:fed7:8acb
Destination Address: 2604:6000:1103:4192:cc15:cc7f:2cd1:5fff
```

dns query, the subdomain part seems quite weird here it's hex encoded form let's convert it to ascii



```
∨ Queries
  > 43616e64792043616e652053657269616c204e756d6265722038343931.holidaythief.com: type A, class IN
> Authoritative nameservers
```

& we've found the exfiltrated data here that passed to the threat actor using DNS data exfiltration technique



Paste hex numbers or drop file

```
43616e64792043616e652053657269616c204e756d6265722038343931
```

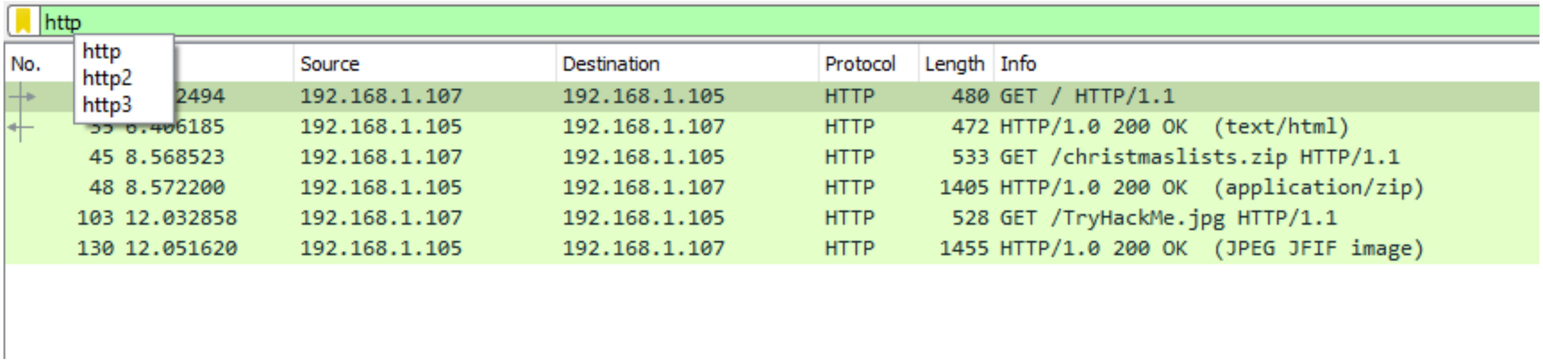
Character encoding

ASCII

```
Candy Cane Serial Number 8491
```

Question: What data was exfiltrated via DNS?
-Candy Cane Serial Number 8491

let's check the http protocol traffics & we've found a GET request on the christmaslists.zip file

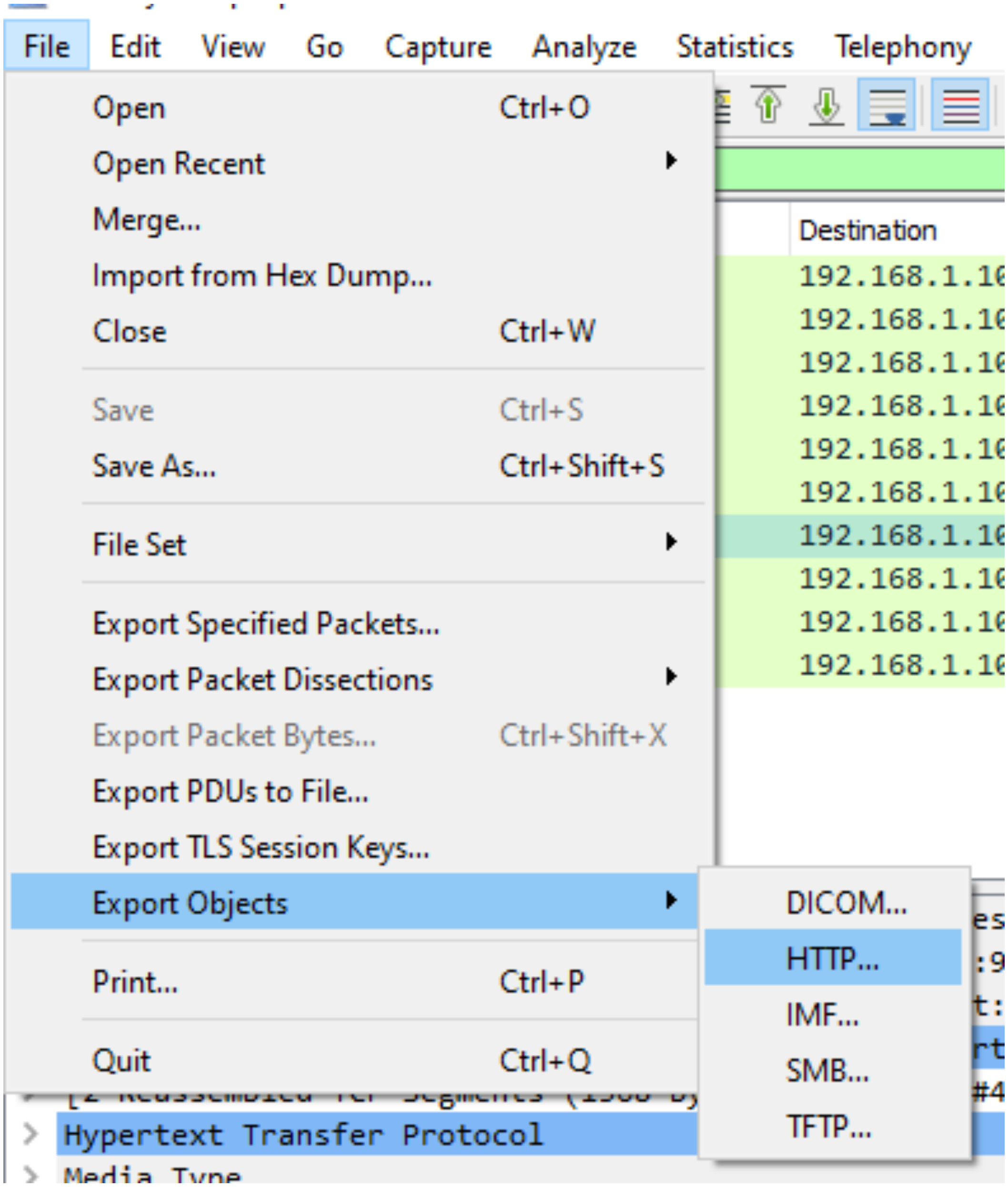


http

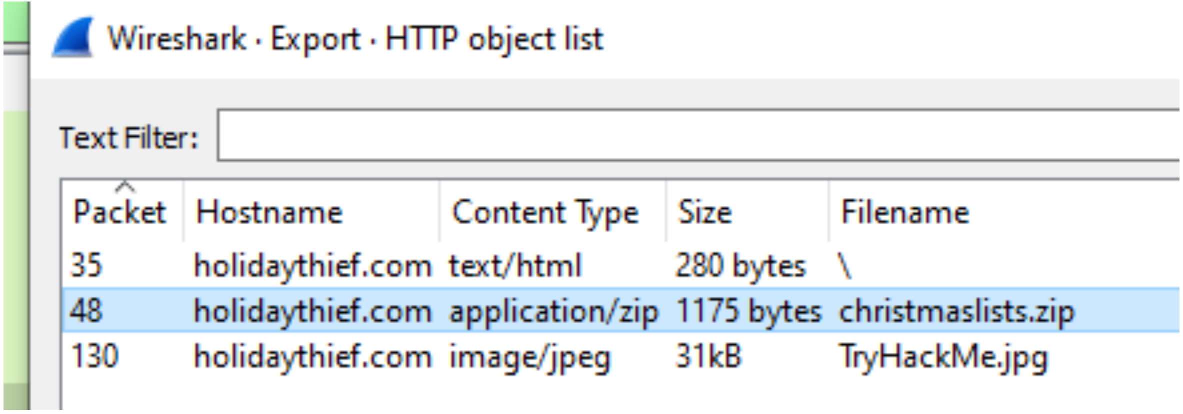
```
No.                 Source          Destination     Protocol  Length Info
         2494       192.168.1.107   192.168.1.105   HTTP         480 GET / HTTP/1.1
         6185       192.168.1.105   192.168.1.107   HTTP         472 HTTP/1.0 200 OK  (text/html)
   45 8.568523      192.168.1.107   192.168.1.105   HTTP         533 GET /christmaslists.zip HTTP/1.1
   48 8.572200      192.168.1.105   192.168.1.107   HTTP        1405 HTTP/1.0 200 OK  (application/zip)
  103 12.032858     192.168.1.107   192.168.1.105   HTTP         528 GET /TryHackMe.jpg HTTP/1.1
  130 12.051620     192.168.1.105   192.168.1.107   HTTP        1455 HTTP/1.0 200 OK  (JPEG JFIF image)
```

let's export the zip file and see the content

select export object -> HTTP

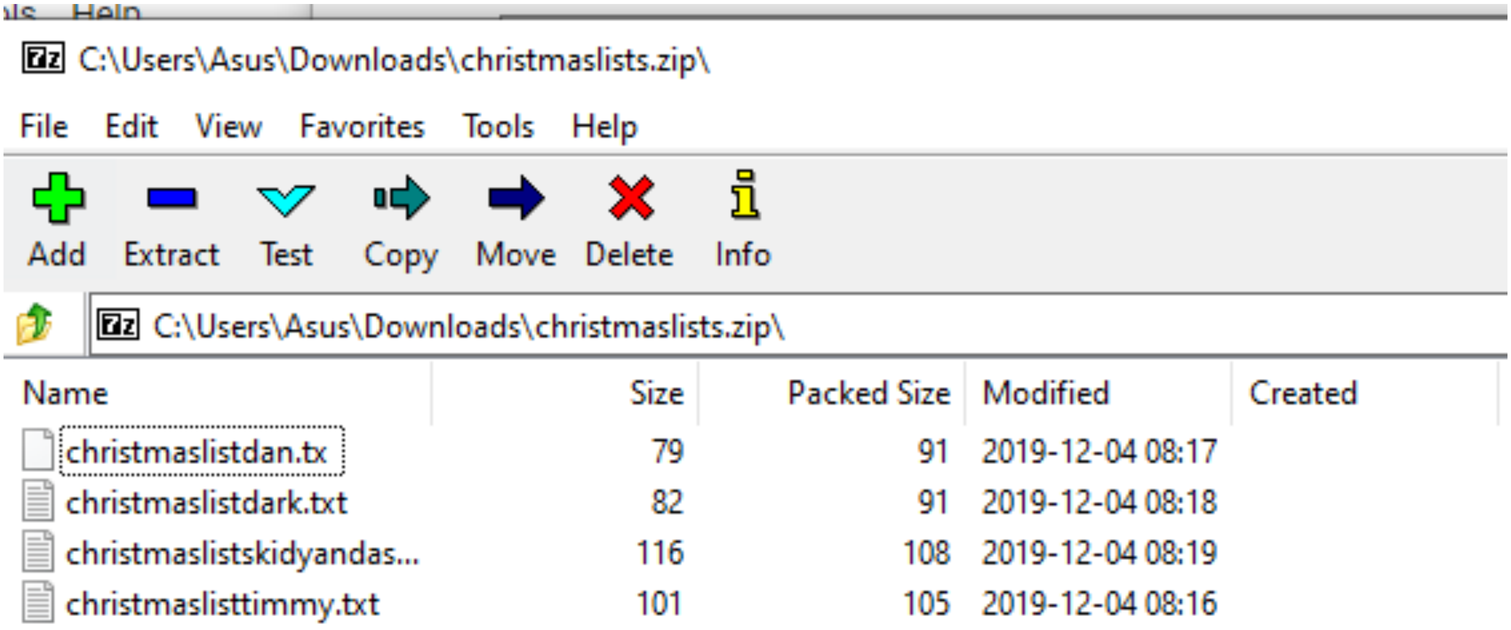


select the christmasist.zip to be exported



Packet	Hostname	Content Type	Size	Filename
35	holidaythief.com	text/html	280 bytes	\
48	holidaythief.com	application/zip	1175 bytes	christmaslists.zip
130	holidaythief.com	image/jpeg	31kB	TryHackMe.jpg

content of the zip file, now we want to find the item that timmy wanted for christmas

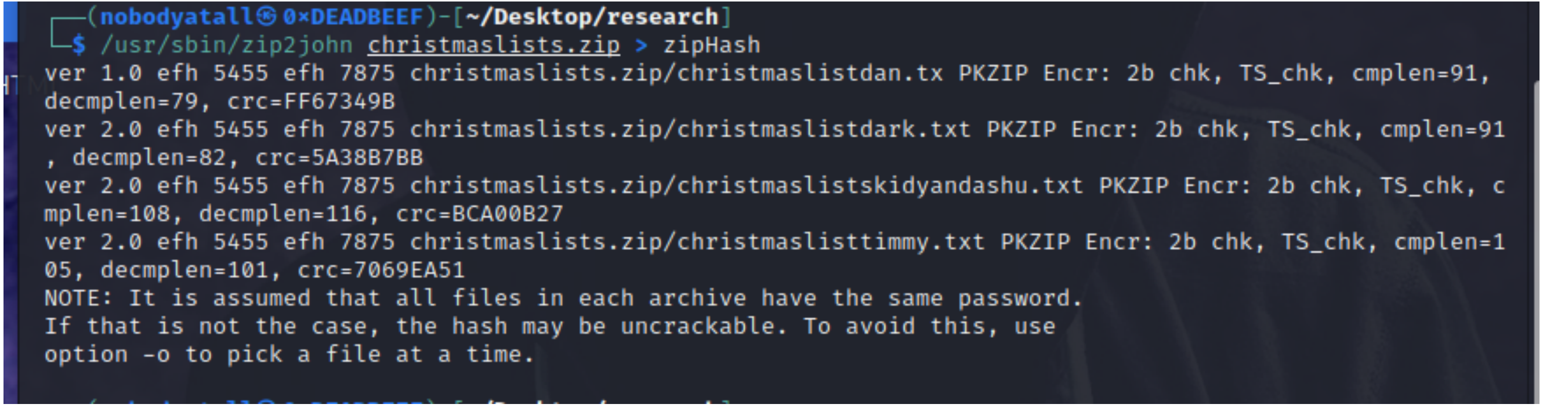


C:\Users\Asus\Downloads\christmaslists.zip\

Name	Size	Packed Size	Modified	Created
christmaslistdan.tx	79	91	2019-12-04 08:17	
christmaslistdark.txt	82	91	2019-12-04 08:18	
christmaslistskidyandas...	116	108	2019-12-04 08:19	
christmaslisttimmy.txt	101	105	2019-12-04 08:16	

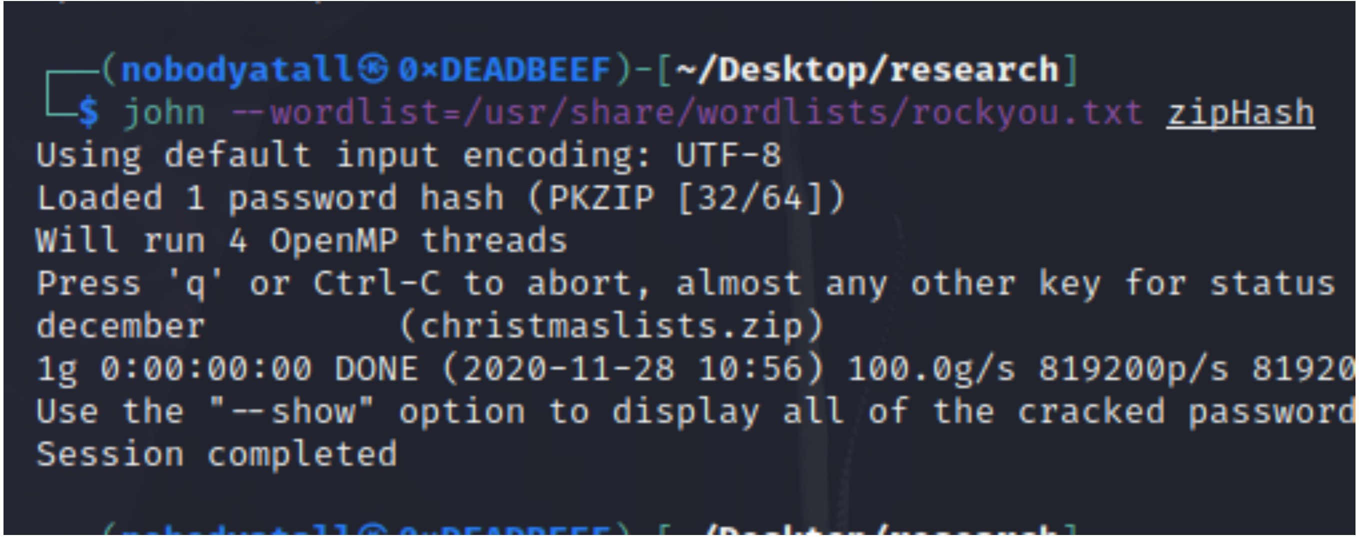
but it required credential to access the file, let's use john the ripper to crack it

extract the zip file hash



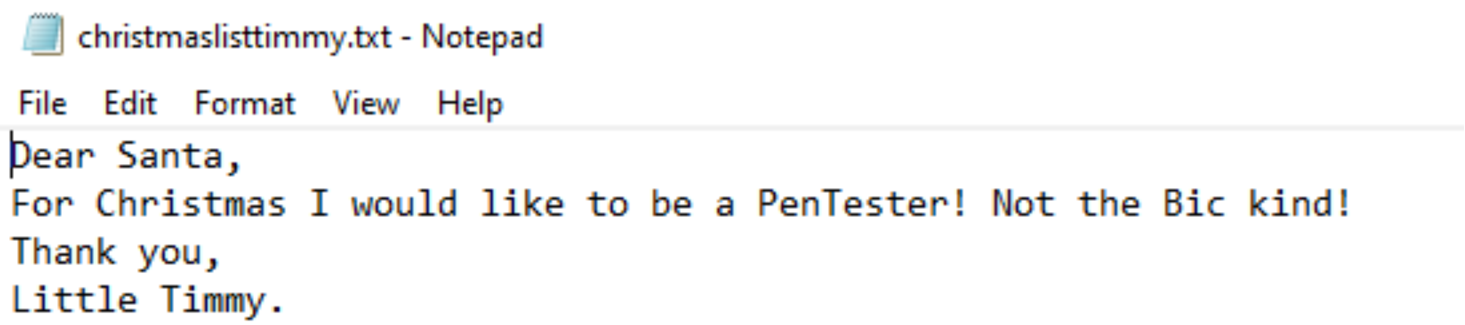
```
┌──(nobodyatall㉿0xDEADBEEF)-[~/Desktop/research]
└─$ /usr/sbin/zip2john christmaslists.zip > zipHash
ver 1.0 efh 5455 efh 7875 christmaslists.zip/christmaslistdan.tx PKZIP Encr: 2b chk, TS_chk, cmplen=91,
decmplen=79, crc=FF67349B
ver 2.0 efh 5455 efh 7875 christmaslists.zip/christmaslistdark.txt PKZIP Encr: 2b chk, TS_chk, cmplen=91
, decmplen=82, crc=5A38B7BB
ver 2.0 efh 5455 efh 7875 christmaslists.zip/christmaslistskidyandashu.txt PKZIP Encr: 2b chk, TS_chk, c
mplen=108, decmplen=116, crc=BCA00B27
ver 2.0 efh 5455 efh 7875 christmaslists.zip/christmaslisttimmy.txt PKZIP Encr: 2b chk, TS_chk, cmplen=1
05, decmplen=101, crc=7069EA51
NOTE: It is assumed that all files in each archive have the same password.
If that is not the case, the hash may be uncrackable. To avoid this, use
option -o to pick a file at a time.
```

now use john the ripper to crack it & we've found the credential for it



```
┌──(nobodyatall㉿0xDEADBEEF)-[~/Desktop/research]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt zipHash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
december         (christmaslists.zip)
1g 0:00:00:00 DONE (2020-11-28 10:56) 100.0g/s 819200p/s 81920
Use the "--show" option to display all of the cracked password
Session completed
```

checking timmy christmas item wishlist using the credential, seems like timmy want to be PenTester uh

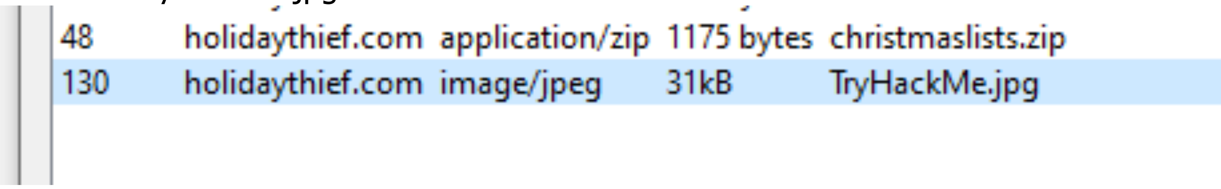


```
Dear Santa,
For Christmas I would like to be a PenTester! Not the Bic kind!
Thank you,
Little Timmy.
```

-PenTester

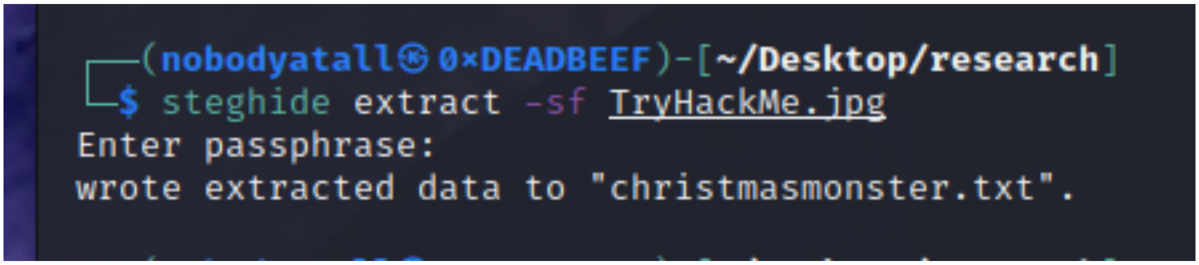
if we still remember that we can hide some files within an image file too using steganography technique

here the TryHackMe.jpg was kinda sus



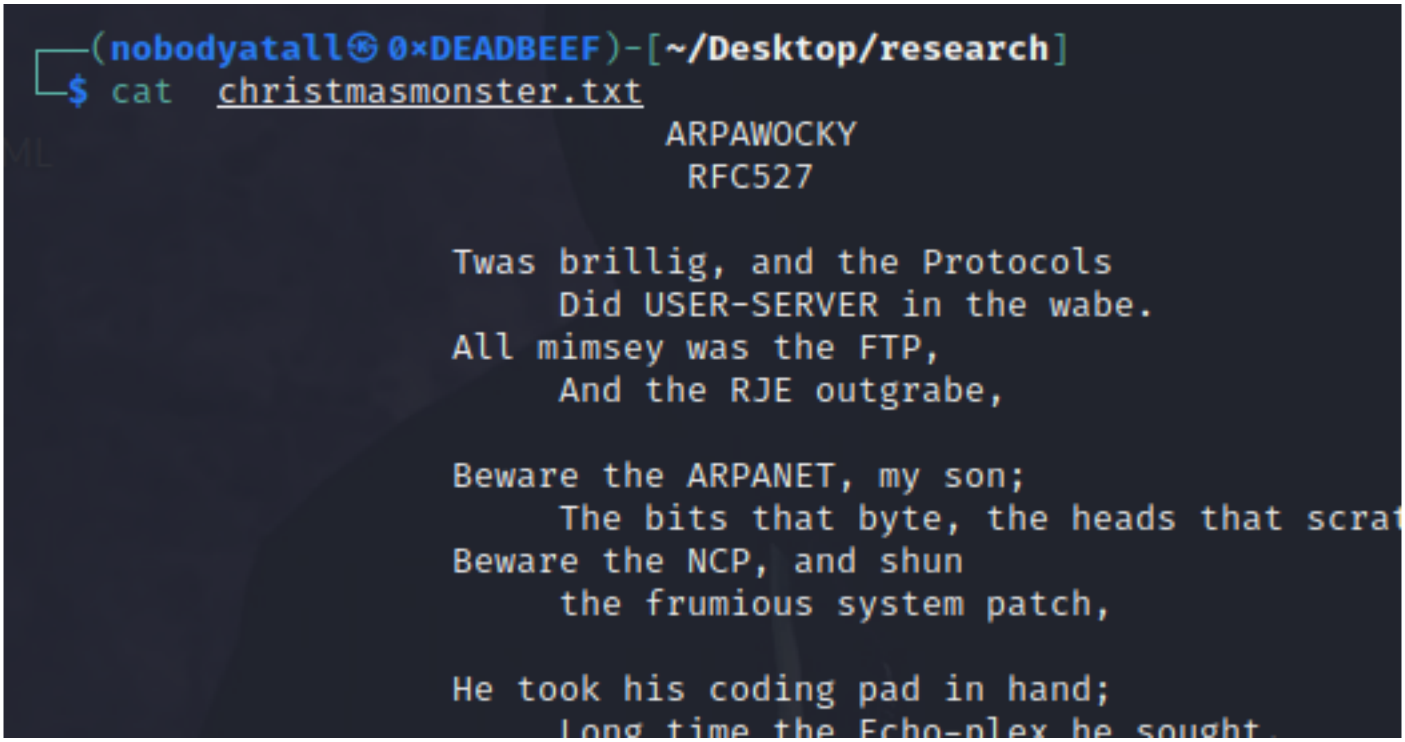
48	holidaythief.com	application/zip	1175 bytes	christmaslists.zip
130	holidaythief.com	image/jpeg	31kB	TryHackMe.jpg

let's use steghide to extract the content of the file without credentials & voila! we've extracted the christmasmonster.txt



```
┌──(nobodyatall㉿0xDEADBEEF)-[~/Desktop/research]
└─$ steghide extract -sf TryHackMe.jpg
Enter passphrase:
wrote extracted data to "christmasmonster.txt".
```

the content of the text file



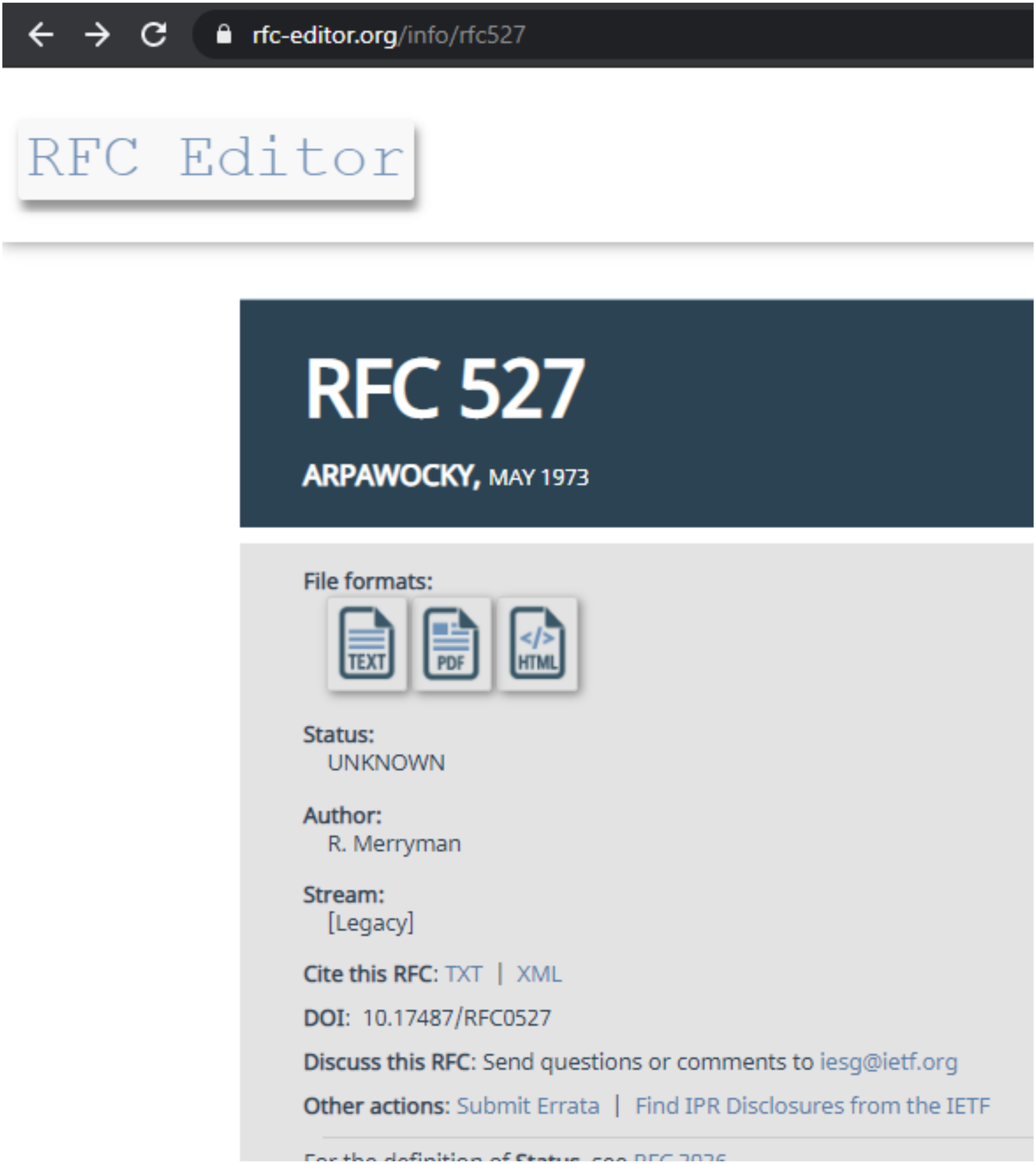
```
┌──(nobodyatall㉿0×DEADBEEF)-[~/Desktop/research]
└─$ cat  christmasmonster.txt
                              ARPAWOCKY
                               RFC527

                    Twas brillig, and the Protocols
                         Did USER-SERVER in the wabe.
                    All mimsey was the FTP,
                         And the RJE outgrabe,

                    Beware the ARPANET, my son;
                         The bits that byte, the heads that scrat
                    Beware the NCP, and shun
                         the frumious system patch,

                    He took his coding pad in hand;
                         Long time the Echo-plex he sought.
```

did some googleFu here & we found that the following text was belong to RFC 527



RFC 527

ARPAWOCKY, MAY 1973

File formats:

Status:
UNKNOWN

Author:
R. Merryman

Stream:
[Legacy]

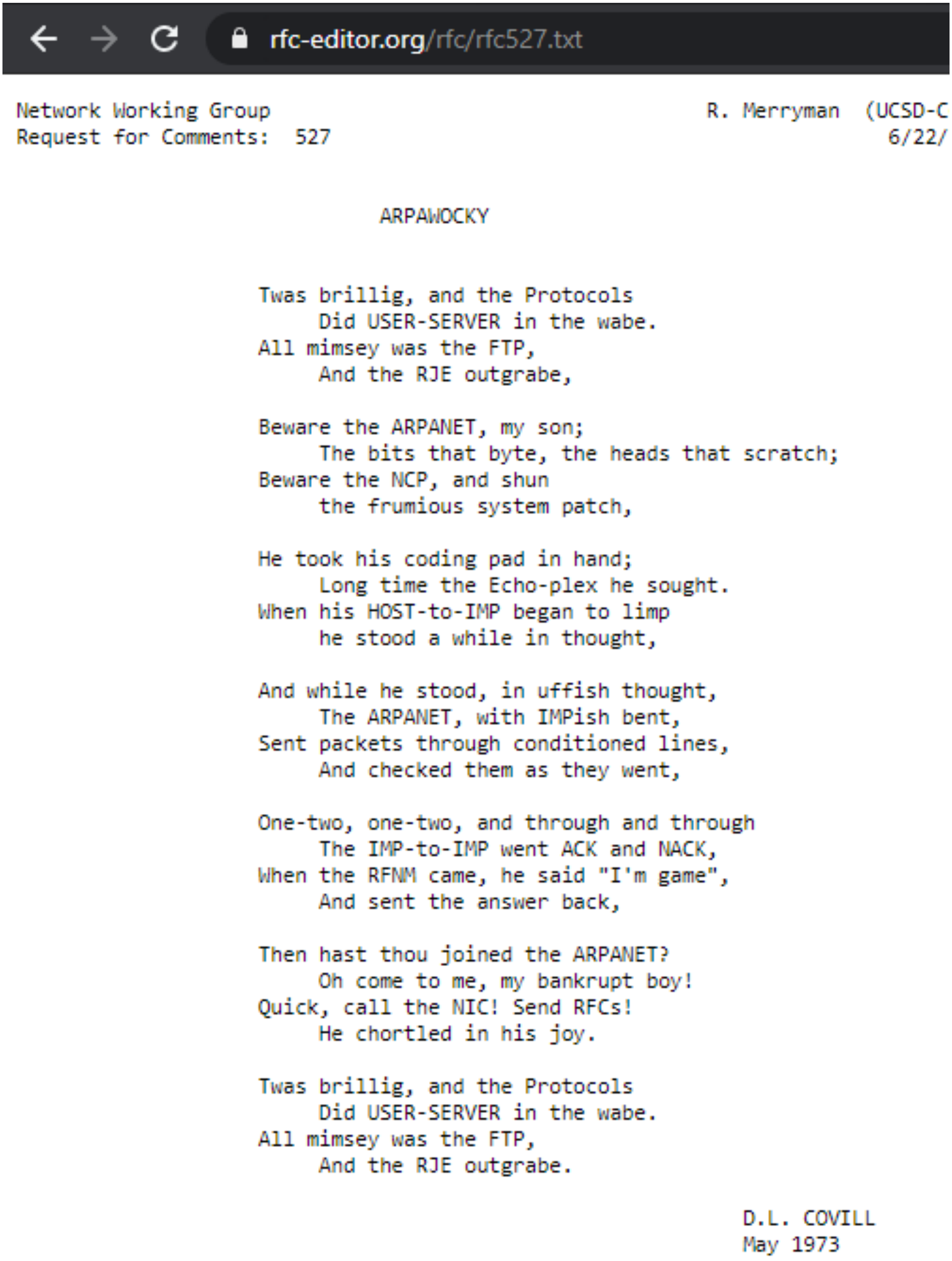
Cite this RFC: TXT | XML

DOI: 10.17487/RFC0527

Discuss this RFC: Send questions or comments to iesg@ietf.org

Other actions: Submit Errata | Find IPR Disclosures from the IETF

For the definition of **Status**, see RFC 2026



```
Network Working Group                                    R. Merryman  (UCSD-C
Request for Comments:  527                                             6/22/


                              ARPAWOCKY


                    Twas brillig, and the Protocols
                         Did USER-SERVER in the wabe.
                    All mimsey was the FTP,
                         And the RJE outgrabe,

                    Beware the ARPANET, my son;
                         The bits that byte, the heads that scratch;
                    Beware the NCP, and shun
                         the frumious system patch,

                    He took his coding pad in hand;
                         Long time the Echo-plex he sought.
                    When his HOST-to-IMP began to limp
                         he stood a while in thought,

                    And while he stood, in uffish thought,
                         The ARPANET, with IMPish bent,
                    Sent packets through conditioned lines,
                         And checked them as they went,

                    One-two, one-two, and through and through
                         The IMP-to-IMP went ACK and NACK,
                    When the RFNM came, he said "I'm game",
                         And sent the answer back,

                    Then hast thou joined the ARPANET?
                         Oh come to me, my bankrupt boy!
                    Quick, call the NIC! Send RFCs!
                         He chortled in his joy.

                    Twas brillig, and the Protocols
                         Did USER-SERVER in the wabe.
                    All mimsey was the FTP,
                         And the RJE outgrabe.

                                                            D.L. COVILL
                                                            May 1973
```

Question: What was hidden within the file?
-RFC527