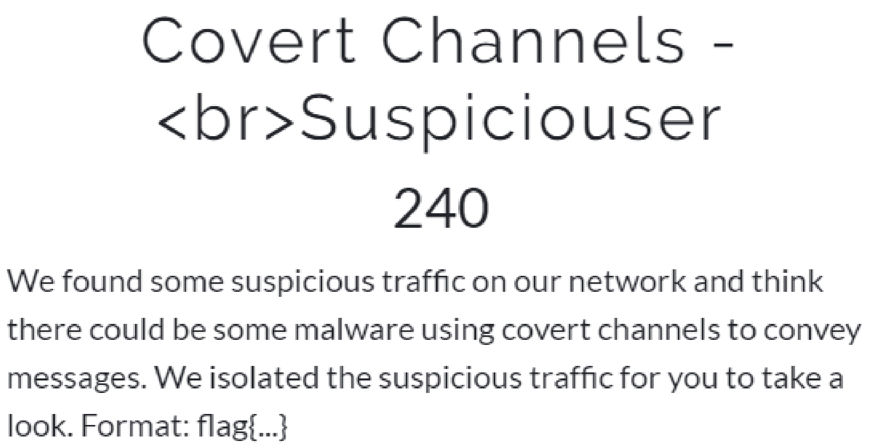
Covert Channel – Suspiciouser

Write up by:

John Antone

835 COS

854 CPT

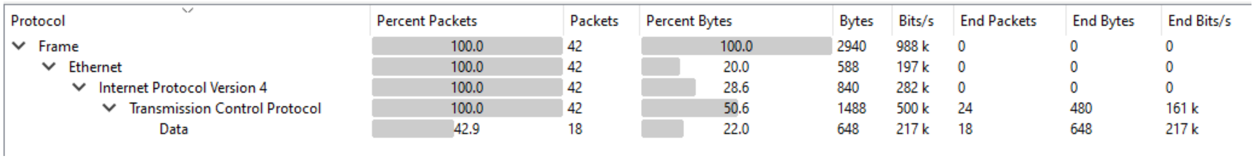


Covert Channels -
Suspiciouser

240

We found some suspicious traffic on our network and think there could be some malware using covert channels to convey messages. We isolated the suspicious traffic for you to take a look. Format: flag{...}

To start off with on this Pcap we look at the Protocol Hierarchy page:

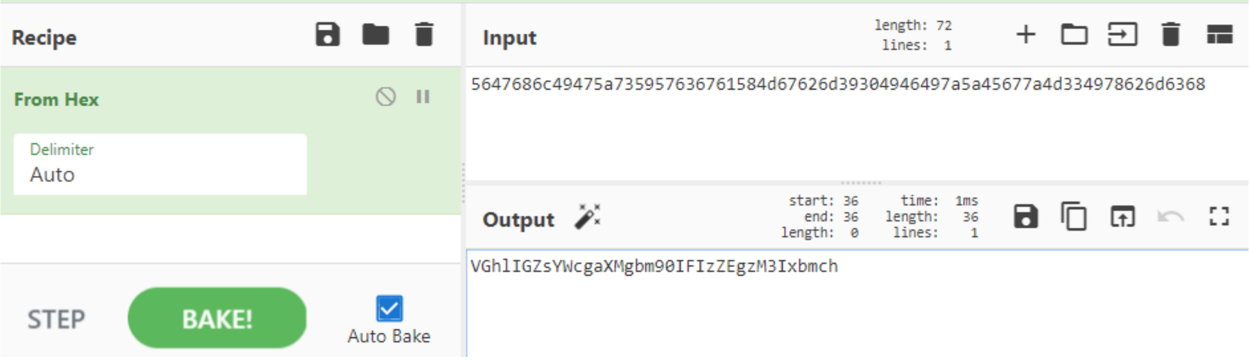


Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	42	100.0	2940	988 k	0	0	0
Ethernet	100.0	42	20.0	588	197 k	0	0	0
Internet Protocol Version 4	100.0	42	28.6	840	282 k	0	0	0
Transmission Control Protocol	100.0	42	50.6	1488	500 k	24	480	161 k
Data	42.9	18	22.0	648	217 k	18	648	217 k

We can see that this is all TCP packets with a Data Field on some of the packets.

To start off let us dig into that data field and see what is going on here. We see the following characters in all the data fields:
5647686c49475a735957636761584d67626d39304946497a5a45677a4d334978626d6368

Let's take this info to cyber chef and try to decode it. It looks like it was all in hex so here is the results of converting from Hex:



From Hex
Delimiter
Auto

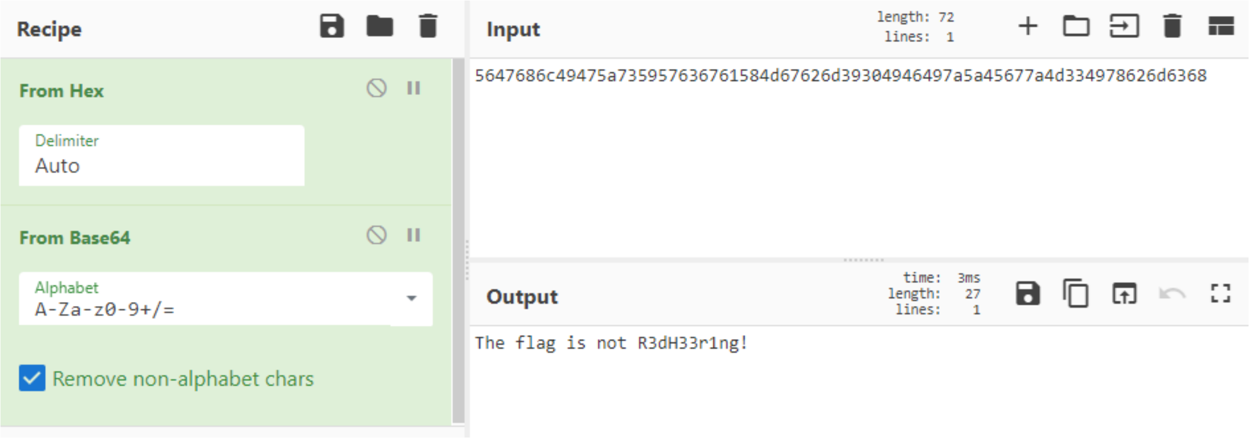
Input

```
length: 72
 lines: 1
5647686c49475a735957636761584d67626d39304946497a5a45677a4d334978626d6368
```

Output

```
 start: 36    time: 1ms
   end: 36  length: 36
length: 0    lines: 1
VGhlIGZsYWcgaXMgbm90IFIzZEgzM3Ixbmch
```

Next we see that it looks like it could be in base64, so let's decode that also:



From Hex
Delimiter
Auto

From Base64
Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

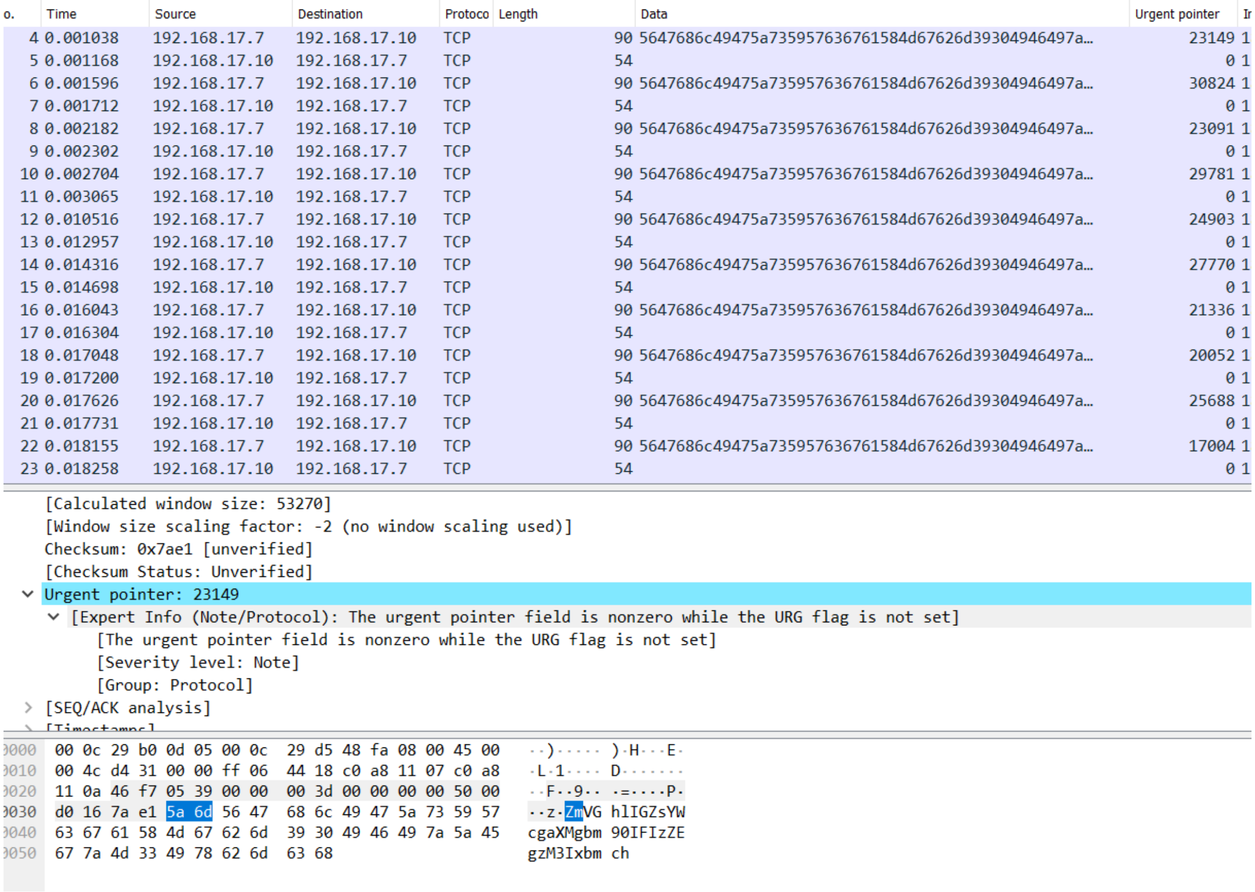
Input

```
length: 72
 lines: 1
5647686c49475a735957636761584d67626d39304946497a5a45677a4d334978626d6368
```

Output

```
  time: 3ms
length: 27
 lines: 1
The flag is not R3dH33r1ng!
```

So, we know that this is not the answer, we need to look into the packets a little bit more to see what else we can find that is not right. In the packet information we see that the Urgent Pointer filed is highlighted:



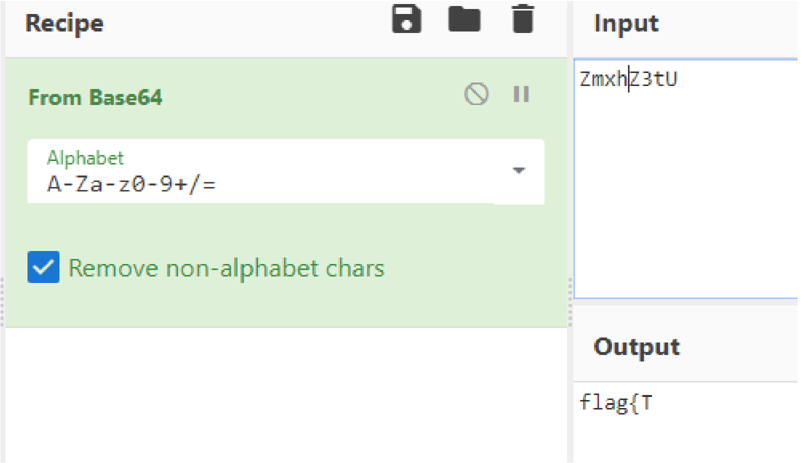
```
o.  Time      Source         Destination    Protoco Length        Data                                                   Urgent pointer Ir
  4 0.001038  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    23149 1
  5 0.001168  192.168.17.10  192.168.17.7   TCP             54                                                            0 1
  6 0.001596  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    30824 1
  7 0.001712  192.168.17.10  192.168.17.7   TCP             54                                                            0 1
  8 0.002182  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    23091 1
  9 0.002302  192.168.17.10  192.168.17.7   TCP             54                                                            0 1
 10 0.002704  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    29781 1
 11 0.003065  192.168.17.10  192.168.17.7   TCP             54                                                            0 1
 12 0.010516  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    24903 1
 13 0.012957  192.168.17.10  192.168.17.7   TCP             54                                                            0 1
 14 0.014316  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    27770 1
 15 0.014698  192.168.17.10  192.168.17.7   TCP             54                                                            0 1
 16 0.016043  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    21336 1
 17 0.016304  192.168.17.10  192.168.17.7   TCP             54                                                            0 1
 18 0.017048  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    20052 1
 19 0.017200  192.168.17.10  192.168.17.7   TCP             54                                                            0 1
 20 0.017626  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    25688 1
 21 0.017731  192.168.17.10  192.168.17.7   TCP             54                                                            0 1
 22 0.018155  192.168.17.7   192.168.17.10  TCP             90 5647686c49475a735957636761584d67626d39304946497a...    17004 1
 23 0.018258  192.168.17.10  192.168.17.7   TCP             54                                                            0 1

    [Calculated window size: 53270]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x7ae1 [unverified]
    [Checksum Status: Unverified]
  ⌄ Urgent pointer: 23149
    ⌄ [Expert Info (Note/Protocol): The urgent pointer field is nonzero while the URG flag is not set]
        [The urgent pointer field is nonzero while the URG flag is not set]
        [Severity level: Note]
        [Group: Protocol]
  > [SEQ/ACK analysis]
  > [Timestamps]

)000  00 0c 29 b0 0d 05 00 0c  29 d5 48 fa 08 00 45 00   ··)····· )·H···E·
)010  00 4c d4 31 00 00 ff 06  44 18 c0 a8 11 07 c0 a8   ·L·1···· D·······
)020  11 0a 46 f7 05 39 00 00  00 3d 00 00 00 00 50 00   ··F··9·· ·=····P·
)030  d0 16 7a e1 5a 6d 56 47  68 6c 49 47 5a 73 59 57   ··z·ZmVG hlIGZsYW
)040  63 67 61 58 4d 67 62 6d  39 30 49 46 49 7a 5a 45   cgaXMgbm 90IFIzZE
)050  67 7a 4d 33 49 78 62 6d  63 68                     gzM3Ixbm ch
```

We can now see that with adding a column with the urgent pointer we see that all of the pointers are different. In the first packet we see that the hex is 5a 6d or Zm

The second packet is 78 68 or xh, Third Packet is 5a 33 or Z3 and the 4th packet is 74 55 or tU.

We have ZmxhZ3tU

This looks like a base64 pattern so let’s look at cyberchef and see if that decodes to anything that might resemble a flag.



Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

Input

```
ZmxhZ3tU
```

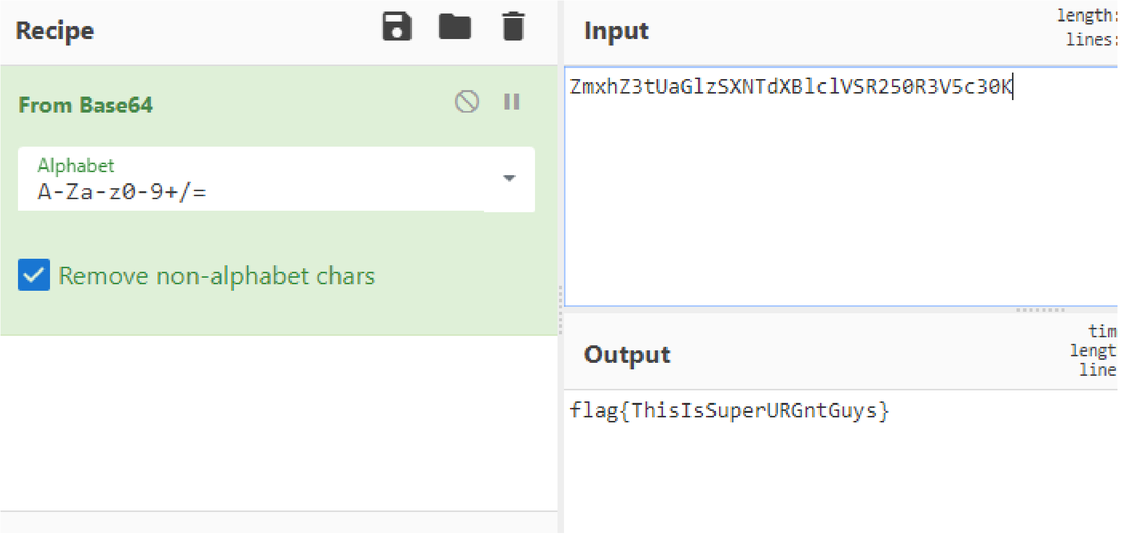
Output

```
flag{T
```

We have the start of our flag let’s pull the rest of the urgent pointers out to get the flag.

ZmxhZ3tUaGlzSXNTdXBlclVSR250R3V5c30K

This gives us our flag:



Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

Input

```
ZmxhZ3tUaGlzSXNTdXBlclVSR250R3V5c30K
```

Output

```
flag{ThisIsSuperURGntGuys}
```