Writeup - The Nack

Paweł Lasek

June 16, 2016

Contents

1 Description of task

"Mysterious traffic", or in other words, we have a dump of network traffic and we're asked to analyze what it says. Provided is a pcap file with dump of traffic.

2 Solution

This wasn't a particularly hard task if one knows his TCP. Short analysis of the dump in Wireshark showed that the file contains only SYN and NACK packets, as one side constantly tries to connect to a closed port.

Quick look at sample of SYN packets showed that each SYN packet had data associated with it (used to be rare, nowadays more popular method of lowering latency to first byte). Some quick shell play with tshark gave us the data:

```
tshark -r ce6e1a612a1da91648306ace0cf7151e6531abc9.pcapng \
 -Y 'tcp.connection.syn' -T fields -e data

474f41540147494638
474f41540139614e02
474f415401e100a100
474f41540100ffffff
```

```
474f415401000000ff
474f415401ffffffff
474f415401ff21ff0b
474f4154014e455453
474f41540143415045
474f415401322e3003
..............
```

As we can easily recognize, the first half of the data field is always `47 4F 41 54 01`. However, better to verify that:

```
tshark -r ce6e1a612a1da91648306ace0cf7151e6531abc9.pcapng \
 -Y 'tcp.connection.syn' -T fields -e data \
 | cut -b -10 | uniq

474f415401
```

So, we have found a common "header". Which coincidentally matches the theme of the CTF, saying `"GOAT\x01"`. Time to extract usable data out.

```
tshark -r ce6e1a612a1da91648306ace0cf7151e6531abc9.pcapng \
 -Y 'tcp.connection.syn' -T fields -e data \
 | cut -b 11- | tr -d '\n' | xxd -r -p >temp
```

Some people might have already noticed the first packet starts with what is essentially beginning of GIF header, something confirmed after looking over the file with `file` as well image viewer `feh`. The image it shows is the infamous ROFLCOPTER. Still, no flag... but something blinks in the image.

The obvious thing was to check for data hidden in short-lived frame. Let's split the gif file into frames:

```
convert -coalesce temp out%05d.pgm
```

In the resulting outXXXX.pgm files, we find in file `out00016.pgm`, we find the flag: `TUCTF{this_transport_layer_is_a_syn}`

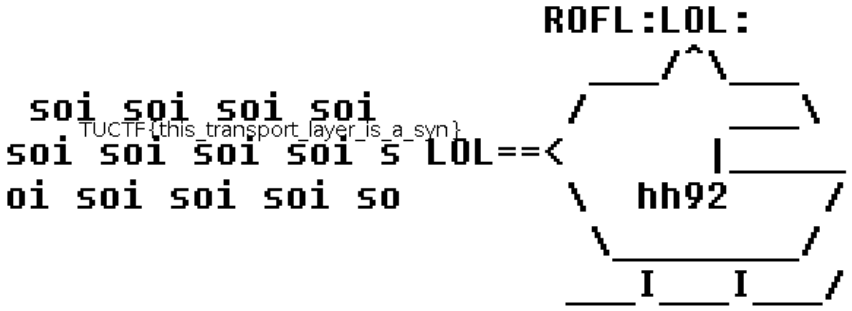

Figure 1: Flag frame