The Hidden Traffic By Payampardaz

by Mojtaba Mollaei

I analyzed the provided pcapng file and initially observed a lot of seemingly normal traffic. To isolate suspicious communications, I began applying filters.

Step 1: Filter Out Common Traffic

First, I excluded known DNS and public IPs to reduce noise:

```
((!(ip.dst == 142.250.190.78)) && !(ip.dst == 8.8.4.4) && !(ip.dst ==
1.1.1.1) && !(ip.dst == 4.2.2.4))
```

Then, I removed retransmitted TCP packets and traffic to a local machine:

```
((!(ip.dst == 142.250.190.78) && !(ip.dst == 8.8.4.4) && !(ip.dst ==
1.1.1.1) && !(ip.dst == 4.2.2.4)) && !(ip.dst == 192.168.1.141))
```

This significantly reduced the noise and made the analysis more manageable.

Fake Flags

I encountered decoy flags like:

```
flag[THE_REAL_ONE_IS_ENCRYPTED]
```

Clearly, the real flag must be hidden or encrypted. Time to dig deeper.

Suspicious DNS Queries

Amidst the DNS traffic, one query stood out — a request to `170.39.217.204` for:

```
LNwvTIPmGafeyAgQABkd8g==fake.key.goodvibe.com
```

This domain is clearly abnormal, hinting at exfiltration or encoded data. I filtered it specifically:

```
ip.dst == 170.39.217.204
```

Key and Ciphertext Extraction

Among the DNS queries, I found a `.key.goodvibe.com` domain:

```
UEB5YW1wYXJkYXotQ08qIQ==.key.goodvibe.com
```

Base64 decoding the first part gives:

```
P@yampardaz-CO*!  ← (the AES key)
```

Other related queries ended with `.data.goodvibe.com`, which likely contain encrypted flag data.

Assembling the Encrypted Data

Each query to `.data.goodvibe.com` followed a pattern:

```
<number>-<ciphertext>
```

By sorting these packets based on the number prefix and concatenating the ciphertexts, I reconstructed the full ciphertext:

```
_gb7kJVJxMhUx8fq-gE9WBEd_9W3ftbDH1arovtydFAz0wxs_UFXDSeThTv8DeFs
```

Decryption

Using AES in CBC mode with PKCS5 padding and the 128-bit key (`P@yampardaz-CO*!`), I decrypted the ciphertext.

Result:

```
flag[Payampardaz-CTF-2025]|99
```