# Forensics CTF 8

**Platform:** picoCTF 2019

**Challenge Name:** WebNet1

**Category:** Forensics

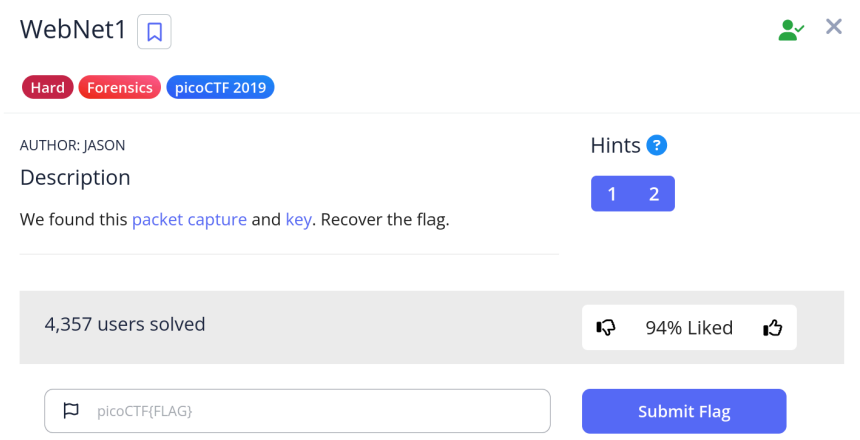
**Difficulty:** Hard

**Submitted By:** Gurleen Kaur Brar

## Objective

The goal was to analyze encrypted traffic in a packet capture (PCAP) file and extract a hidden flag using a provided RSA private key. The challenge involved configuring Wireshark to decrypt TLS traffic and inspecting reassembled SSL packets.

## Challenge Description




## Files and Tools Used

- **Files Provided:**
  - `capture.pcap` – a packet capture file
  - `picopico.key` – RSA private key for decrypting TLS

- **Tools Used:**
  - Kali Linux
  - Wireshark
  - RSA Key configuration panel in Wireshark

# Step-by-Step Process

## Step 1: View the RSA Key

Verified that the provided `picopico.key` was a standard PEM-encoded private RSA key.

```
cat picopico.key
```

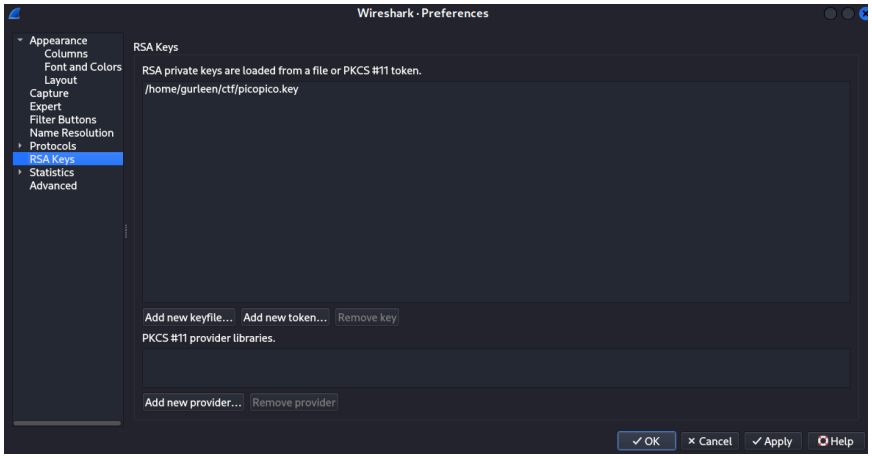
```
┌──(gurleen㉿kali)-[~/ctf]
└─$ cat picopico.key
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCwKlFPNKjseJF5
puCJU5×38XcT1eQge5zOKNahAlYudvGVOEs61TnIgvcER4ko8i3OCwak2/atcGk3
oz9jFKep7XFEYNP31IwwD9j/YazlKy4DRLGObOyIZUU1f2WRA7Uhf0POQXsDT1oU
X32jMKZkQSSDW4MRZd9trJYdO2TrcEPMsBiZQlFlvgnNwl3QlawozTHLAJKI36j1
cPwSMMeNca1e0Zi1s7R5IxfhpNXOBF0FmxiWvmeOHbaspyHg8UEmGBrkd4k4wXSK
GQvrc8QjycP4ScEdquxJiYnDT8iEbAq70/7f/5NIN1DE9YoGJqKYjTS9nRPB4Yvj
JN/SJnhvAgMBAAECggEACCnd3LrG/TZVH3sROqvqO1CwQPYPfUXdLVyNHab7EWon
pc+XBOHurJENG2CpRYF7h+nQ5ADhfIYSCicBf/jsEB7VueJ20CxEVtHVL3h6R6Bp
oHMle0Em8OcofuMpdL/kO+om3T8BkVSzCvCl5NMTUuAF7iRmfX7oDLALwM0IzzQv
2un+2UmT15rgAZfl3IL1PGvJhbhLxfeeyPE9MBy1SqBjQ9rNFn8sQv959J6BHz4b
EpK//ErtNP2yh7oiVBBgKEQ1gEuOjQC/4oxoqCFfZaf9XNRCxB/zY1nUprvJyz09
NMQWNF2EmvmBVGfoTxmuut5N0GbVr2UyHxWMKm2sOQKBgQDpb2+AWgWlGtetuLKJ
fJs8dnd6LhnafbKCOXMOT68qMBRoTpBtVTLRVSNvWCm8m4TTEazX4+ZA+bJFwUFw
aATDmHcr6lMI3tNKrcsnY2F7o5I4z6mwuRuSeszq/ndxZqCzwCu4nKixh3cznp7j
JiElNG0d8Lu5eQgmVAK1AhWXfQKBgQDBMa9ga7VJUP4pzcHnWAoi34OpfjvQYeGl
IKL3AKO4OedaHdH9qid41PQHnL7O3xzN669SkLZ5s0d88A/LFLk4oZNMKdkSTQIQ
+AMbXH01HGFvnCOuPg/FbNp1wS7zJEg5u5HFQWyMPNJLr/hZ6g2Yp+UGpAcGTwM/
RCPVAPhLWwKBgQDAB0OaOnPaVjKGXiHAqBirrGiswa/S5QQrzEaxxys5cUPYaoi0
6BldysPTnJr45JZna2rcTkXjvYTBjTDf3zHMFWgzYBfefC8kh8NPK5nNs8ldorbd
AemEnjBkP+DSELKyK6vLulOrdtzAQgRCp+MsT+xTbO2ArefeX826SXSpoQKBgC2v
nDOHBQXje1dTawlUToFUrgQE8AwlOYEdKKyUoCLOvqEW8DO2a0MtyM+MB6tQI7Wm
iH1T73L0LHGlK3bw3aRAwV5/fu/O+jAdFk8AHjPTFE+acu2fi4c6aKb0GjAxYksU
yjIFeK/pKinv4SESMkjpW0WowGiDgtcRPBAA/LaFAoGAfEM1rfM0v3UmB7PS6u0m
P3ckP2CFCdaryXPfC52GBcJ3Q46YpsQvLTVotM+teHvTjNw2jwwZxIl4NenGSEj3
KDhQoOiQC9BrDD+DB4I9+T9nxT3g7R6MrgITghB4We7TVhL/PljnJTyDqpjNA4kY
TveAJPv6Xq1ERt5PUtX3BqQ=
-----END PRIVATE KEY-----
```

## Step 2: Load the PCAP into Wireshark

Opened `capture.pcap` in Wireshark.

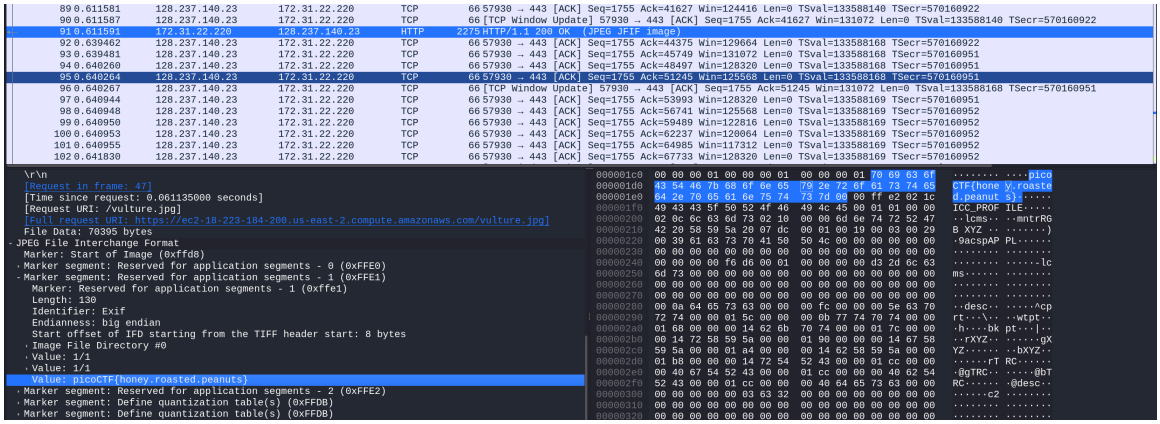
Then configured Wireshark to decrypt the TLS stream using the provided RSA key:

1. Navigated to `Edit → Preferences`
2. Selected `Protocols → TLS`
3. Added the key to the **RSA Keys list** for port 443 traffic




## Step 3: Locate the Flag in SSL Stream

With decryption enabled, located a JPEG payload (JFIF marker) in the reassembled SSL packets. In the decoded view, the flag was visible as ASCII inside the JPEG segment.




# Flag Submitted


The flag was extracted from decrypted SSL content and successfully submitted.