Judul:

Close to you

Filosofi deskripsi:

- Mungkin jarak yang terbentang adalah penguat sebuah ikatan,
- Dan kedekatan tanpa celah justru meruntuhkan pertahanan,

Judul dan kalimat diatas merupakan petunjuk kerentanan utama di challenge ini, yaitu **Close Prime**.

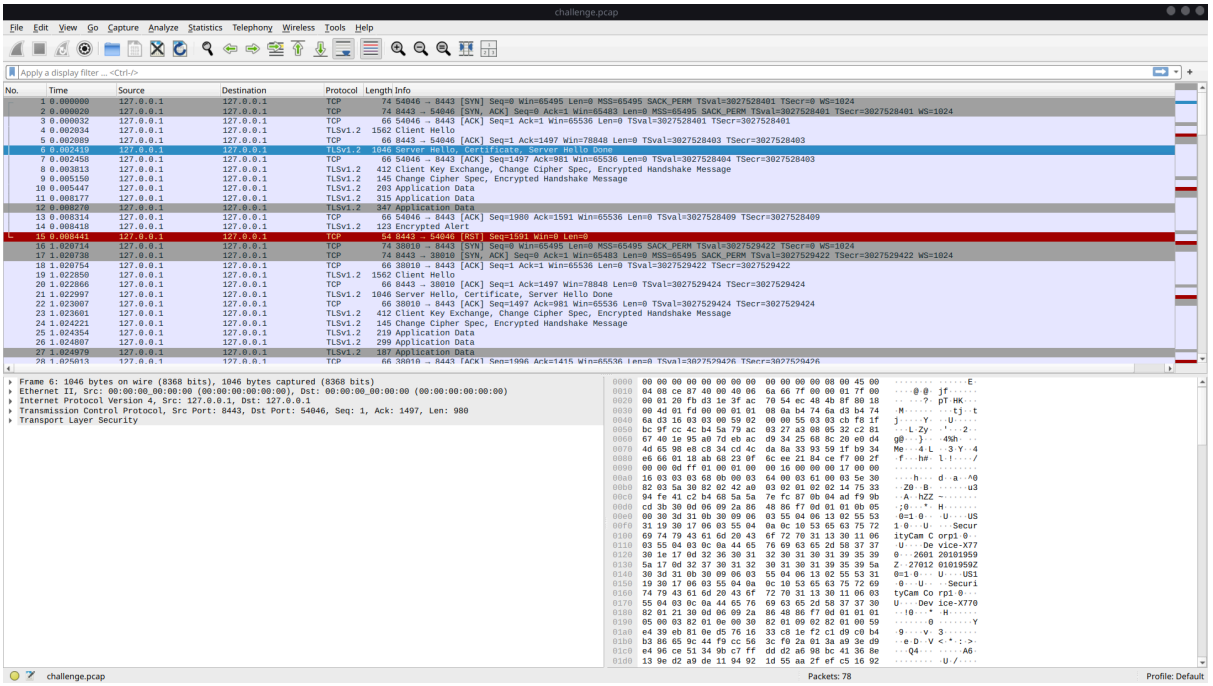
- Jangan melihat dirinya hanya dari luarnya,
- Lihatlah dia dari dalam hatinya.

Kalimat diatas merupakan petunjuk untuk menyelesaikan challenge ini kita perlu **melihat isi file yang diberikan lebih dalam**.

Selain sebagai petunjuk challenge, tentu saja kalimat ini juga mempunyai maknanya sendiri.

Pada challenge ini dilampirkan file .pcap, ini merupakan rekaman komunikasi antara client dengan server dengan protokol HTTPS, untuk bisa mengerti apa yang mereka bicarakan kita perlu melakukan dekripsi protokol HTTPS menjadi HTTP, dan untuk melakukannya kita perlu memiliki key yang valid. Berikut langkah langkahnya:

Buka file .pcap yang dilampirkan, lalu cari paket dengan protokol **TLSv1.2** dan info: **Server Hello, Certificate, Server Hello Done,** atau kita juga bisa menggunakan filter: tls.handshake.type == 11 untuk mencarinya. Lalu buka paket tersebut.



Selanjutnya cari modulus dan public exponent yang digunakan untuk enkripsi HTTPS, ini akan menjadi bahan kita kedepannya untuk membuat key yang valid. Berikut lokasinya:

> Transport Layer Security

> TLSv1.2 Record Layer: Handshake Protocol: Certificate

> Handshake Protocol: Certificate

> Certificates (865 bytes)

> Certificate [...]

> signedCertificate

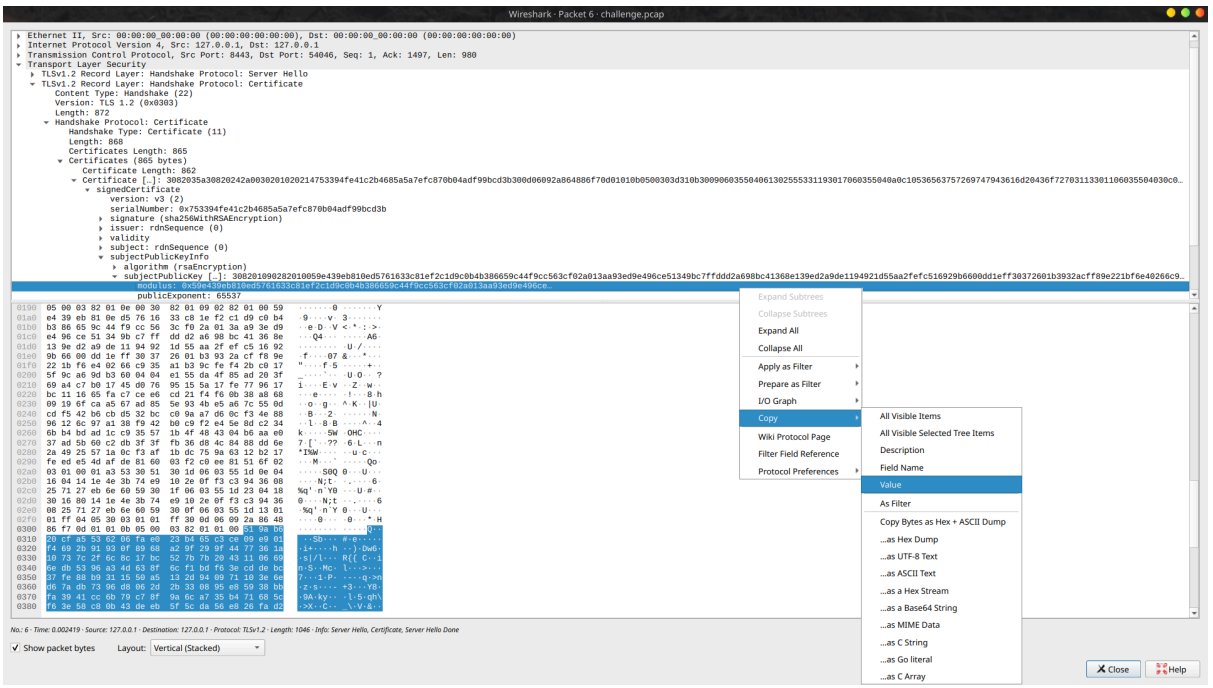
> subjectPublicKeyInfo

> subjectPublicKey

> **modulus**

> **publicExponent**

Setelah ditemukan, klik kanan lalu copy value yang ada pada modulus dan public Exponent.



modulus:

59e439eb810ed5761633c81ef2c1d9c0b4b386659c44f9cc563cf02a013aa93ed9e496ce51349b
c7ffddd2a698bc41368e139ed2a9de1194921d55aa2fefc516929b6600dd1eff30372601b3932a
cff89e221bf6e40266c935a1b39cfef42bc0175f9ca69db3600404e155da4f85ad203f69a4c7b01
745d07695155a17fe779617bc111665fac7cee6cd21f4f60b38a86809196fcaa567ad855e934be
5a67c550dcdf542b6cbd532bcc09aa7d60cf34e8896126c97a138f942b0c9f2e45e8dc2346bb4b
dad1cc935571b4f484304b6aae037ad5b60c2db3f3ffb36d84c8488dd6e2a4925571a0cf3af1bd
c759a6312b217feede54dafde816003f2c0ee81516f

publicExponent:

65537

Sesuai deskripsi soal, kerentanan yang ada disini yaitu Close Prime, dimana 2 bilangan prima yang digunakan terlalu berdekatan sehingga menjadi mudah untuk ditebak. Gunakan script berikut untuk membuat key yang dapat digunakan untuk mendekripsi protokol HTTPS yang ada pada file .pcap yang diberikan. Script berikut akan mengkonversi hex menjadi integer, lalu menjalankan Fermat attack untuk mengetahui p dan q dan membuat key yang valid.

Solver.py:

```
import sys
from math import isqrt
from Crypto.PublicKey import RSA
hex_n = str(input("Masukkan modulus: "))
n = int(hex_n, 16)
e = int(input("Masukkan eksponen: "))

print(f"[*] Target Modulus (N): {str(n)[:30]}...")
print("[*] Menjalankan Fermat Factorization...")
a = isqrt(n) + 1
count = 0
p = 0
q = 0
while True:
    val = a*a - n
    if val >= 0:
        b = isqrt(val)
        if b*b == val:
            # Ketemu!
            p = a - b
            q = a + b
            print(f"[+] Faktor ditemukan dalam {count} iterasi!")
            break
    a += 1
    count += 1
    if count % 1000000 == 0:
        print(f"    ... iterasi ke-{count}")

print("[*] Menghitung Private Key...")
phi = (p - 1) * (q - 1)
d = pow(e, -1, phi)

key = RSA.construct((n, e, d, p, q))
output_filename = "hacked_private.pem"

with open(output_filename, "wb") as f:
    f.write(key.export_key())

print(f"\n[SUKSES] Private Key tersimpan di: {output_filename}")
print("Gunakan file ini di Wireshark untuk mendekripsi HTTPS.")
```

Lalu masukkan file output dari script tersebut ke wireshark dengan cara:

> Tab ‘Edit’

> Preferences…

Pada window yang baru, klik:

> Protocols

> TLS

> RSA key list : Edit

Pada window yang baru, masukkan parameter:

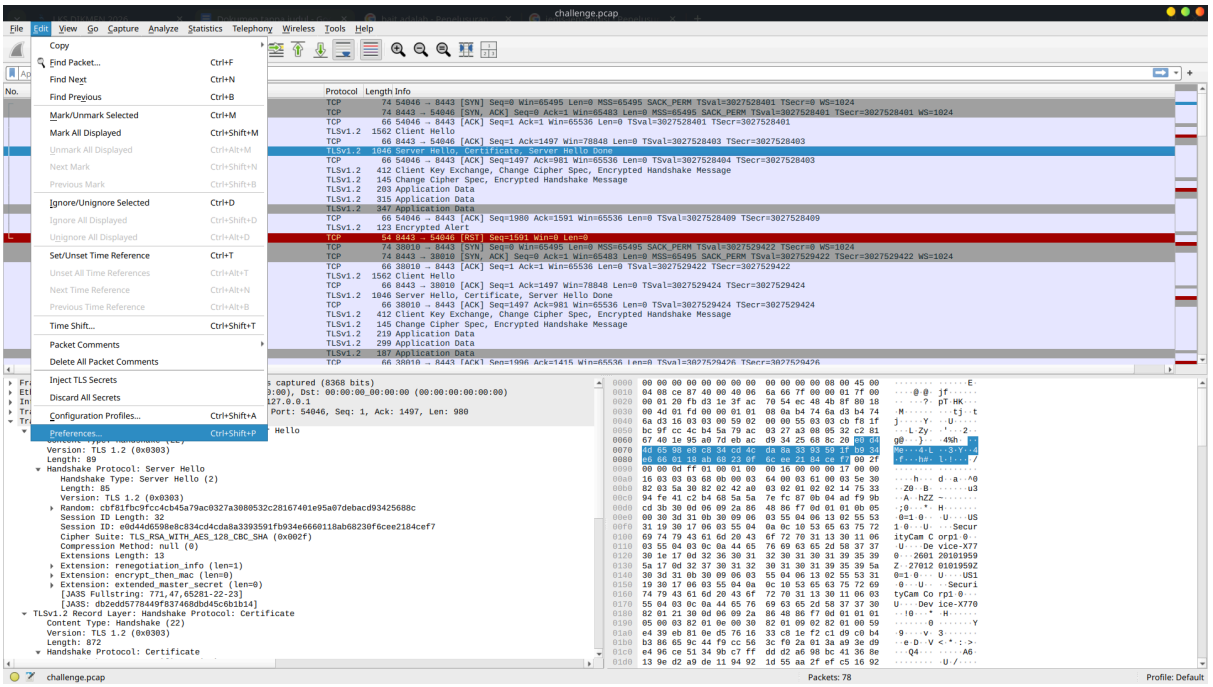
IP address : 0.0.0.0

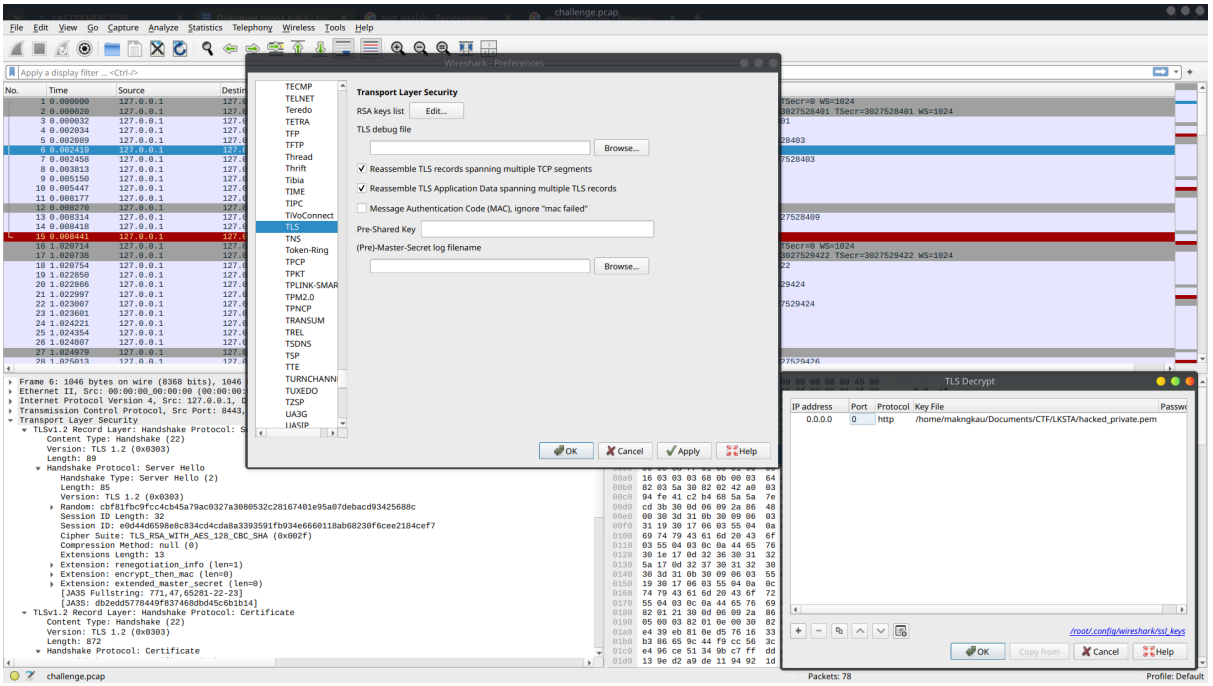
Port : 0

Protocol : http

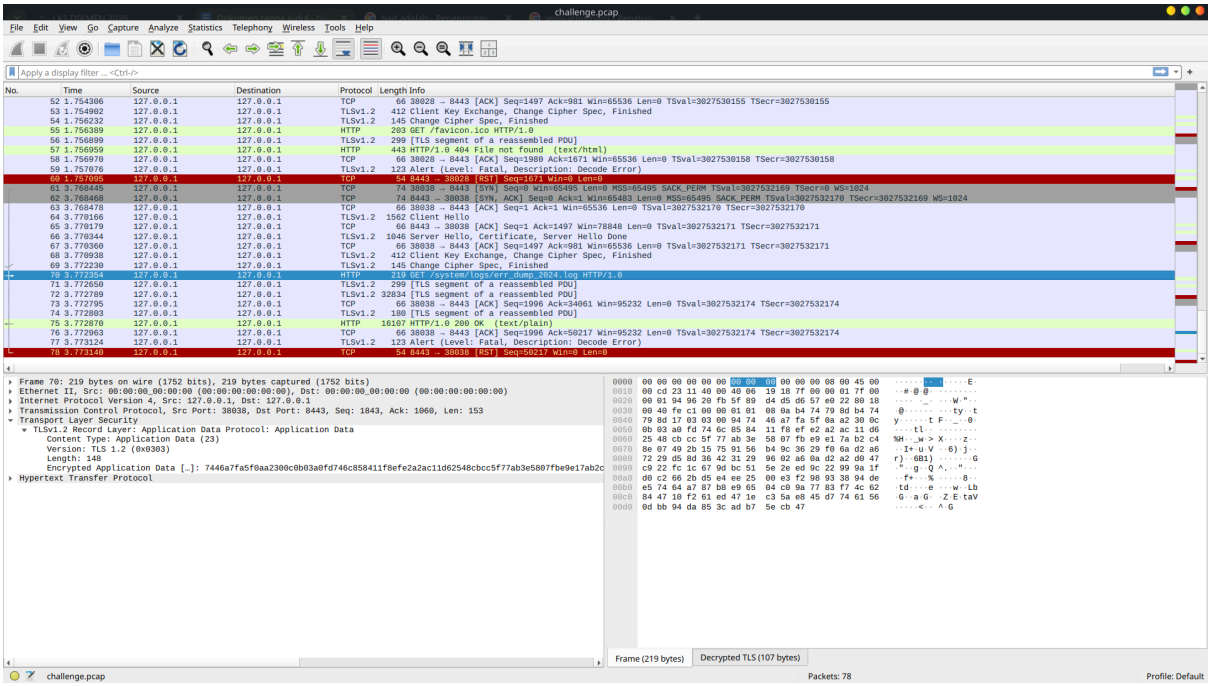
Key file : <lokasi file output dari solver>

Password : <kosong>





Lalu klik ‘OK’ pada kedua tab, dan protokol HTTPS yang sebelumnya berwarna ungu akan berubah menjadi hijau. Artinya kita berhasil melakukan dekripsi HTTPS menjadi HTTP.



Selanjutnya kita tinggal ekspor file di dalam file .pcap yang telah di dekripsi dengan:

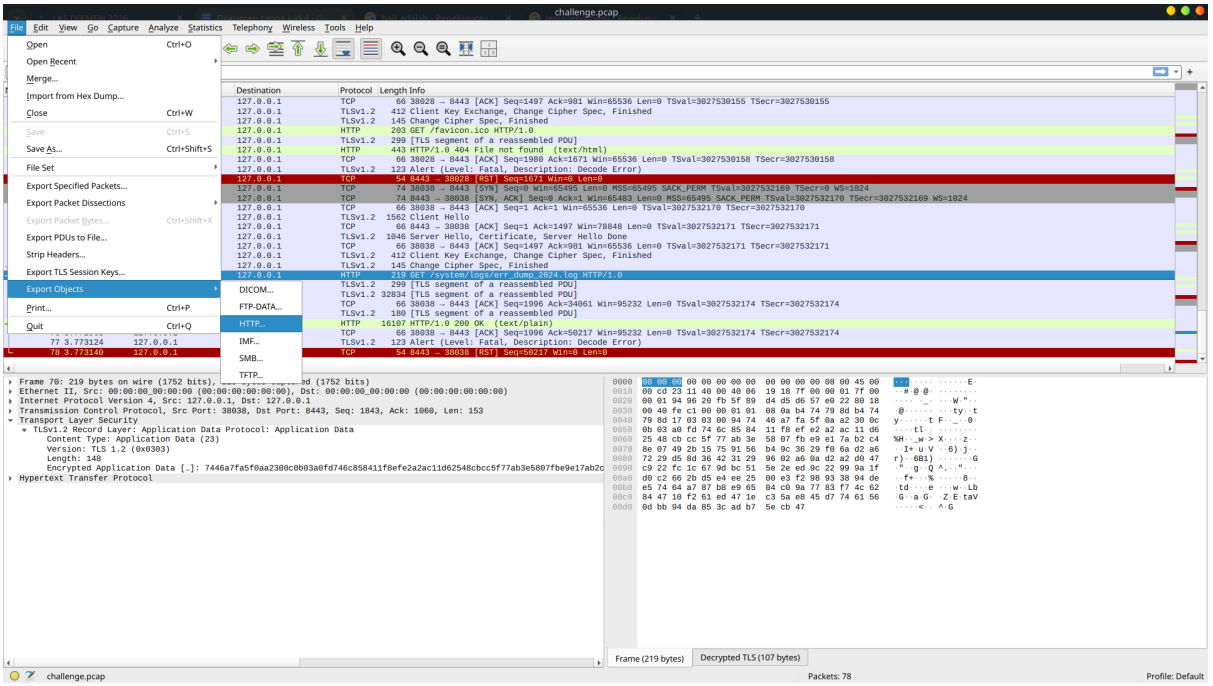
> Tab ‘file’

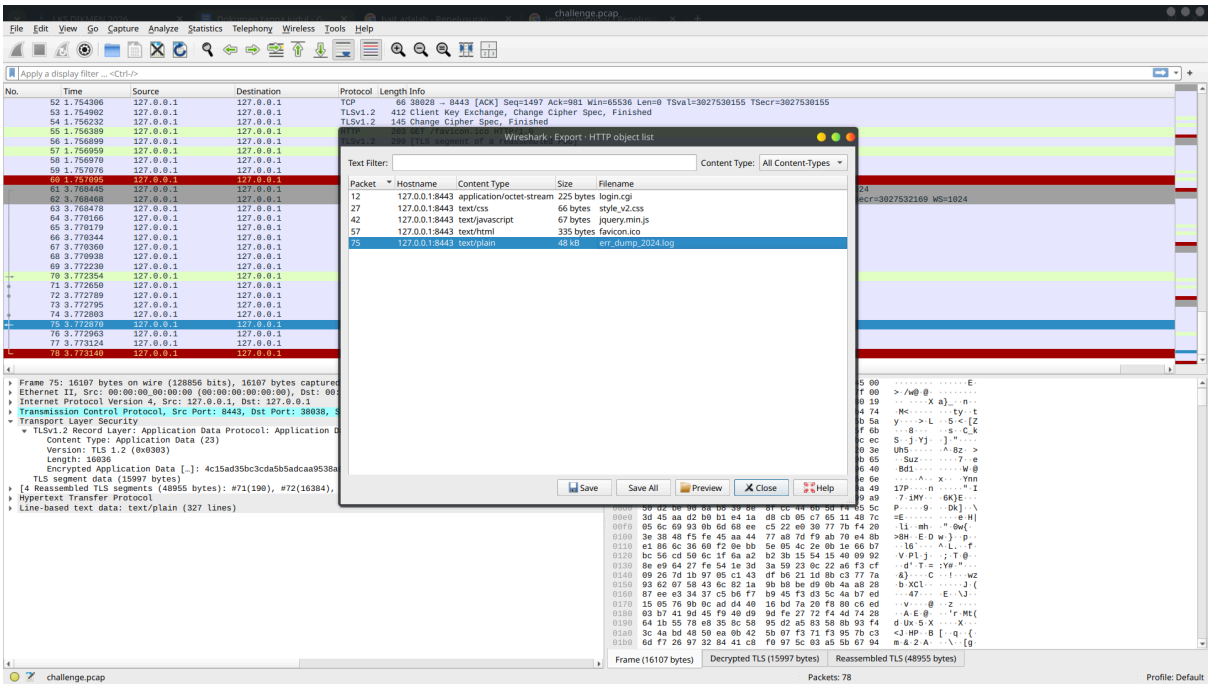
> Export Objects

> HTTP…

> pilih err_dump_2024.log

> Save





File tersebut sebenarnya bukanlah file log, melainkan file PNG. Kita bisa melihatnya ketika membuka file tersebut sebagai teks, maka akan muncul header PNG, atau kita juga bisa menggunakan command ‘file’ di linux untuk melihat ekstensi asli file tersebut. Dan akhirnya flag ditemukan.

Flag:

LKS{HTTPS_SNYA_S1K444T}