SHARKY CTF

Cryptography : Beware my big exponent

Value : 299 points (41 solves)

Description : I made my own certificate on my local network.
Have a look, communication between my client and my server is sooo secure!

Creator: Fratso

File : beware_my_big_exponent.pcapng

Solution :

With a ‘strings beware_my_big_exponent.pcapng’ we can see that there is ciphertext.

On wireshark, we can see TCP and TLSv1.2 packets. We find easily the TLS Certificate.
[on my wireshark there is no tls filter, it’s ssl that we must use but it’s quiet the same]

We can extract it in .pem format, so we have public TLS certificate.

By the challenge’s name, I was obliged to search how to crack RSA private key with huge exponent ‘e’. I found that pdf that I recommand to read, it is very instructive :
http://cacr.uwaterloo.ca/techreports/2004/cacr2004-01.pdf

On it we can look at Boneh & Durfee attack. I’ve used a tool found on github to perform the attack to obtain d, the private exponent with sage :https://github.com/mimoo/RSA-and-LLL-attacks

```
=== running algorithm ===
=== solution found ===
solx:
x:
170929497848127003170928217764342890439398152725314152110567313264204137648814146389745739167904159374630471868458625857392523969683430092676488388573817772
soly:
y: -
167553850445244983058754709588811635736746597308890464708838270746046124457745257455679429642493915266069105629620089997134118705603728517675417005196152209932938346075959981163156448435069257365747801127596265020036929801252789804443590839017361423919187010639863200492189226064555071536593312817687219041343
private key found:
129431424106040453249635737173991509953899998924389589333004388245848286772653040429595686597084583836686964956046705433699036771477323707748262940749332 49
=== 2.2510740757 seconds ===
```

Then, I lose a lot of time because I failed a copy-paste on my hexa strings …
But I just had to use a python script like pycryptodome to construct the private RSA key.

```
from Crypto.PublicKey import RSA
import os
n =
0x00de508237659bf9ddfea3171e51b7bab7be61ca6fc8842d607030f2b836fb2fe9ad33c4e88d96362a69cebaa0c5e3646447a051ce15e
6f81222f37e02655bed041f21ace12c690d50caad1c1a2d429d1b15d85016d51bcd5816c157a20ce517142858f1c8a83d84c5464eb1b4d
5dee0fc618924b95769717e10e60ead9341454698360b88c23bee8b5c19e2cb3f81cc8020c236024339ac2d74042b94764ddfd0dce6c1d
a291d2b28b1875d9e0c35a1883962bc178b697a3713a133729a4510510a48f0cbd8780c7818f25571073b2d3924ae1c67c5be217b6829
f4ff6cf3dcbe63195d7ae9bc8618ae2ab4749be54b0db559152d025fb14d136575c0d84afd9d
e =
0x0092cbd92005563daed06c4b010fbc53dd98c63711dad7b4712bad8ba6bec38ace7f3ef48e491c88e46f38b4b3c443d6809976838fdda
a023724045cf042b21325be66939840068b569a7366cec013ceecfe9d3b63b6817bcfe6d14d72a86992189880aa139237366dd76b197ad
130aec9806056e755b6c7ea97c412dc82268cf6cb95b68749778b79e676d8dfea67f79bebf950b118d61aca718e57644462659071c2eef9
a75fbf2d6fea2d54b4c658651568a958ee9c2ac1542f0b02a00787af1ecfbcf8b5cac1f2f34215feaf674c55eab4f9d289fcfa098947af3c17e
1e1aea3f028ca077ca35b821995301ffde713364d9aac9a3c9ee481a8fcb5d598b6c1
d =
129431424106040453249635737173991509953899998924389589333004388245848286772653040429595686597084583836686964
9560467054336990367714773237077482629407493324 9

key = RSA.construct((n, e, d))
private_key = key.export_key()
file_out = open("private_RSA.pem", "wb")
file_out.write(private_key)
file_out.close()
```

And it’s done, I just insert this private key into wireshark on ‘Preferences> Protocol > SSL’ to have packets in clear. Then, with a simple string you can have the flag.

Flag : shkCTF{Publ1c_3xp_t0o_b1G_6a8ef56ab89cb}