HTB x Uni CTF 2020 – Qualifications

Solver: Will Green (UAHDucky)

Challenge: Weak RSA

Category: Crypto

Intro:

In this challenge we were supplied with two files, flag.enc and pubkey.pem. Flag.enc is the encrypted file with the flag that we are trying to crack and pubkey.pem, which stores public key that was originally used to encrypt the flag.

Walkthrough:

This is a relatively simple CTF challenge that I have seen a few times in other CTFs. It's so common that there is a fairly well known python script called RsaCtfTool that is often used for these types of challenges. Running the script with the following command:

```
python3 RsaCtfTool.py –publickey pubkey.pem –uncipherfile flag.enc
```

Ran many different attacks on the pubkey.pem to try and extract the private key. After performing the wiener attack on the pubkey.pem, the flag was returned:

```
[*] Performing mersenne_primes attack on pubkey.pem.
[*] Performing comfact_cn attack on pubkey.pem.
[*] Performing wiener attack on pubkey.pem.

Results for pubkey.pem:

Unciphered data :
HEX : 0x000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000004854427b6231365f655f356d346c6c5f645f3
37175346c355f7733316e33725f34373734636b7d
INT (big endian) : 235739294664009754021141964585889552081020877168866603775
546741449217656885459234754557957518050
INT (little endian) : 22546574263662123411523113128044215139929103295256955
39888672246595316300700002664005866924193713034455256651439658814008019558
06604050152422026683983740372464444199982356534041823627217628568859818208
39225967889839162532460437518669392406390949281459764025707163465962902177848
02346752506910576671729456775168
STR : b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00HTB{b16
_e_5m4ll_d_3qu4l5_w31n3r_4774ck}'
vagrant@kali:~/RsaCtfTool$
```

Flag: HTB{b16_e_5m4ll_d_3qu4l5_w31n3r_4774ck}

With a quick google search, I was able to read up on the Wiener attack. Most of the math goes over my head but in basic terms, when a small value for d (ironic), is selected in the RSA cryptosystem (maybe to speed up the decryption time), the RSA system can be broken using the continued fraction method.