Obligatory RSA

Written by `@flaberpengu`. Challenge available on Github.

This is an easy cryptography challenge focused on RSA. The first thing to note is that we are given `n1, n2, d1, d2, e` but no designated ciphertext, which is unusual.

The first thing I tried was calculating `gcd(n1, n2)` using SageMath which we'd expect to be `1` if the provided cryptosystem was secure. However, it's not:

```
gcd(n1, n2) =
9925116240800973850976595132262541359576508947713626861810366840898037176246956650733841956083715917138800916619654906904656983178635430247237882300194413
```

If we suppose that this is a standard RSA implementation with `n1 = p1 * q1` and `n2 = p2 * q2`, then we have found `p1 = p2 = gcd(n1,n2)`. Thus, this allows us to calculate `q1, q2`:

```
q1 = n1 // p1 =
1300665137025800167292818837239186742274697860743918334884746468017851049240497302996261537873444052545321513568873221350585057258967890964421360094954092 1
q2 = n2 // p2 =
1091726392335855924478045243710499445239074732009021010168207888600063746430429406492055318644997017436076939694627316056186522639313547157941755331639928 3
```

Now, if we go to dcode.fr and input our `n1, p1, q1, e` values, we find that the `d1` calculated by the tool (let's call this `d1_1`) differs from the provided `d1` value - this is *very* interesting, as checking `p1 * q1` does indeed give us `n1` as expected.

```
d1 =
8884349598986987100155975488291807677985840444078039181856763960207317362328782175131534965057702372524522207496505003504551620730307846116816881936502574697358924513157014394471820304645739127041845908776426663089056607903982173516880580586601931514207043822509217130434335246902948050311394298614784866607 7
d1_1 =
8612603017783784866282597036904709734603734497992034423138970334479963771999708516542093046940339811142423426681299754203400433180874659277603508337293449002241292877976097949941140855166549497113176160859401322629690570930724175329539774866097923848187886640061344706995717308768148217946800862954208403955 3`
```

After some playing about, I tried decrypting `d1` using `n1, d1_1, e` on dcode. If we do this using the "plaintext as string" option, we find the flag: `csawctf{wH04m1_70d3Ny_7r4D1710n_48203915786490217 35}`. (The flag is presumably a callback to the title, referencing that nearly every CTF ever has some form of easy RSA attack).

Revision #4
Created 20 November 2025 10:07:31 by AFNOM
Updated 20 November 2025 10:24:34 by AFNOM