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) =
99251162408009738509765951322625413595765089477136268618103668408980371762469566507338419560837159171388009166196549069046569831786354302472378823001944 13
```

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 =
13006651370258001672928188372391867422746978607439183348847464680178510492404973029962615378734440525453215135688732213505850572589678909644213600949540921
q2 = n2 // p2 =
10917263923358559244780452437104994452390747320090210101682078886000637464304294064920553186449970174360769396946273160561865226393135471579417553316399283
```

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 =
88843495989869871001559754882918076779858404440780391818567639602073173623287821751315349650577023725245222074965050035045516207303078461168168819365025746973589245131570143944718203046457391270418459087764266630890566079039821735168805805866019315142070438225092171304343352469029480503113942986147848666077
d1_1 =
86126030177837848662825970369047097346037344979920344231389703344799637719997085165420930469403398111424234266812997542034004331808746592776035083372934490022412928779760979499411408551665494971131761608594013226296905709307241753295397748660979238481878866400613447069957173087681482179468008629542084039553`
```

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_4820391578649021735}`. (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 2025-11-20 10:07:31 UTC by AFNOM
Updated 2025-11-20 10:24:34 UTC by AFNOM