AES 1 (125)

Description

Okay, so we got program that encrypt your name but with flag appended to it. If we see the source, it's using AES to encrypt with ECB mode. ECB mode encrpyt block per block and you can prepend string.

Write Up

So, this is the vulnerability. If we have string:

"raniaditakhairunnisa"

From the source, we know that size of the block is 16 and append \x00 if the length is not sufficient. ECB parse the string to:

```
plain     : raniaditakhairun nisa++++++++++++
encrypted : eb2e4a33b75674fc2c768134f341e6fd 7fde6885b657f8e2faf6ba3d611fa76c
```

If you can prepend some string, you can guess what is the first character:

Assume the first character is 'a'

First try 16 * 'a':

```
plain     : aaaaaaaaaaaaaaaa raniaditakhairun nisa++++++++++++
encrypted : 1a1cc93c9bd89061567b4faf6ec5188a eb2e4a33b75674fc2c768134f341e6fd 7fde6885b657f8e2faf
```

Next, check if the first character is 'a' with prepend 15 * 'a'

```
plain     : aaaaaaaaaaaaaaar aniaditakhairunn isa++++++++++++
encrypted : d6191e3f9408c69ed9b5f8f2d5a107da 5a9fae0a181657693a06bc950045d8f6 1b0910cbf7e9b51d7b9
```

Finally, check if the first block same as the second block. 1a1cc93c9bd89061567b4faf6ec5188a != d6191e3f9408c69ed9b5f8f2d5a107da

But what will happen if instead we prepend it with `'a' * 15 + 'r'`? We will get the same encrypted block. Then we continue comparing `'a' * 14 + 'r'` and `'a' * 14 + 'r' + [guess]` and so on.

I think from the explanation above, you can solve the problem. ;)

Flag: `COMPFEST9{Thi5_is_the_fl4g_4_y0u_dOnt_be_s4d}`