NTU Computer Security HW0

tags: **NTU**

Let's meet at class

- core idea: meet in the middle attack
- crack target: `hint = keys[0] ^ keys[1] ^ keys[2] ^ keys[3] ^ keys[4]`
- *hint* was known, according to the meet in the middle attack, we can reduce the key space down to $10^{5\times3}+10^{5\times2}$ (craft table for `hint ^ key[0] ^ key[1]` and `keys[2] ^ keys[3] ^ keys[4]`)

```
python.exe .\solver.py
b"FLAG{enCrypTIon_wI7H_A_kEy_i5_N0t_secur3_7Hen_h0w_ab0u7_f1ve_Keys}\x8f...\r\xb4"
```

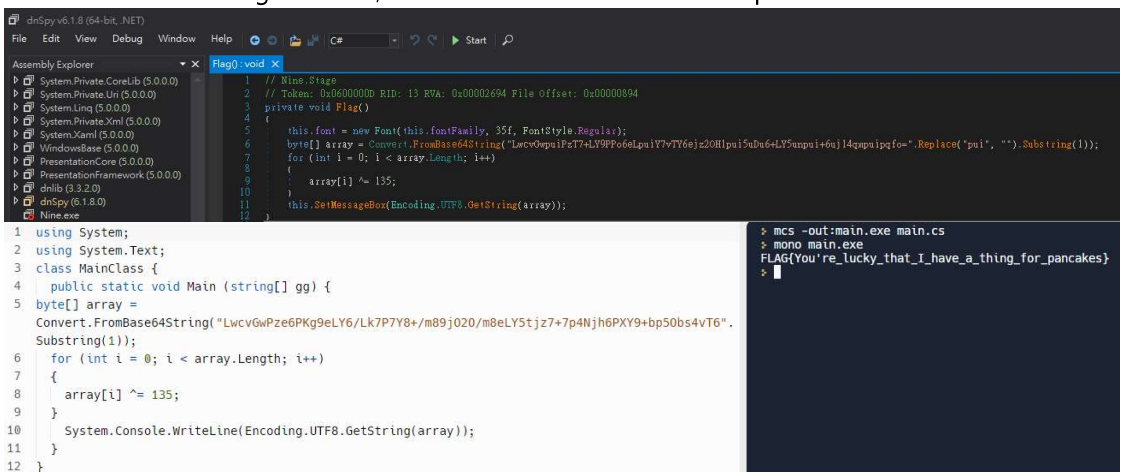
Welcome

- core idea: read the binary and dump flag from data section

```
$ strings dmp|grep flag
flag{CS2022Fall_is_good}
```

Nine - revenge

- core idea: find the flag function, extract it and execute it as a separated code



PyScript

- core idea: since python and js both have `eval` function, we can find a switch expression that evalute only on python or js
- final result: `[] == 0` will produce `true` on js but `false` on python, which can be used to craft array index 0 and 1 to evalutate different expressions

```
$ python.exe .\uploader.py
Here is your Flag: FLAG{w3lc0m3_t0_th3_w0r1d_0f_CTF!}
```

Under Development

- after i discover a better solution for PyScript on stackoverflow, i decided to use it as template
  - ref: https://stackoverflow.com/questions/73688464/the-web-type-ctf-problem-includes-node-and-python/73689708#73689708 (https://stackoverflow.com/questions/73688464/the-web-type-ctf-problem-includes-node-and-python/73689708#73689708)
- core idea
  - poke flask debug console via python script
  - return flag partial infomation via comparison
    - flag len
    - flag character
      if the comparison fail, script return garbage. otherwise, it will return flag1
  - keep leak the info until as the flag fully recovered

```
guessing flag len ... 0
...<skipped>
guessing flag len ... 82
sec_flg_len:82
current flag: <
...<skipped>
class="string">&#39;FLAG{f14sk_d36ug_m0d3_i5_r3a1ly_d4ng3r0u5}\n&#39;</span>
```