

# A lost message WRITEUP

---

The challenge consists in a pdf and an image, the image consists in a series of names taken from the *NEAPOLITAN SMORFIA*, in the pdf we have a message containing some informations about key interpretation and the ciphertext: "The message is as follows:

0x727f00340e075a6b3a69146f2d3e3a67403c343e101d052b1a58623d3c1a0e53087c00245b6e00771d1f1005316e08693e24000714" , followed by an obvious reference to a final ROT encoding:

"But that's not all: she has to follow the passages of an ancient manuscript always found on the island; this manuscript is written in Latin and was written by a young emperor who also mysteriously passed from that island a few years ago."

In the *NEAPOLITAN SMORFIA* each number corresponds to a name, so by exploiting this we can obtain a sequence of numbers: 93, 42, 51, 59, 9, 51, 15, 88, 34, 51, 2, 92, 24, 51, 32, 93, 26, 41, 51, 24, 35, 11, 95, 24, 4, 95, 62, 51, 27, 41, 51, 88, 30, 41, 51, 11, 92, 93, 2, 43, 51, 24, 35, 51, 40, 93, 9, 51, 13, 32, 92, 2, 9 , by extracting bytes from the numbers and by bruteforcing a single-byte XOR we find out that xoring the bytes with key 6c we obtain a meaningful sentence:

1F\_We\_c4N\_n0t\_L1vE\_r0g3th3R\_eE\_4rE\_g01nG\_t0\_D1e\_aL0ne . if we use this as a key to perform a XOR with the given ciphertext we will find an high entropy sentence, by applying it ROT with key -6 and after apply (with a little script) ROT to numbers we obtain a meaningful sentence that will be our flag: w3\_weR3\_n0t\_SupP0seD\_t0\_le4vE\_Kat3\_We\_h4Ve\_To\_g0\_b4ck .

Recipe

From Charcode

Delimiter  
Space

Base  
10

XOR

Key  
6c

HEX

Scheme  
Standard

☐ Null preserving

XOR

1f\_We\_c4N\_n0t\_L1vE\_t0g3th3R\_wE\_4rE\_g01nG\_t0\_D1e\_aL0ne

Key  
727f00340e075a6b3a69146f2d3e3a67403c343e101d0...

HEX

Scheme  
Standard

☐ Null preserving

ROT13

☒ Rotate lower case  
chars

☒ Rotate upper case  
chars

Amount  
-6

Input

length: 151  
lines: 1

93 42 51 59 9 51 15 88 34 51 2 92 24 51 32 93 26 41 51 24 35 11 95 24 4 95 62 51 27 41  
51 88 30 41 51 11 92 93 2 43 51 24 35 51 40 93 9 51 13 32 92 2 9

Output

start: 53    time: 1ms  
end: 53    length: 53  
length: 0    lines: 1

w9\_weR9\_n6t\_SupP6seD\_t6\_le0vE\_Kat9\_We\_h0Ve\_To\_g6\_b0ck

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> stringa="w9_weR9_n6t_SupP6seD_t6_le0vE_Kat9_We_h0Ve_To_g6_b0ck"
>>> stringa2=""
>>> for i in stringa:
...     if i.isnumeric():
...         stringa2+=str((int(i)-6)%10)
...     else:
...         stringa2+=i
...
>>> stringa2
'w3_weR3_n0t_SupP0seD_t0_le4vE_Kat3_We_h4Ve_To_g0_b4ck'
>>> []
```