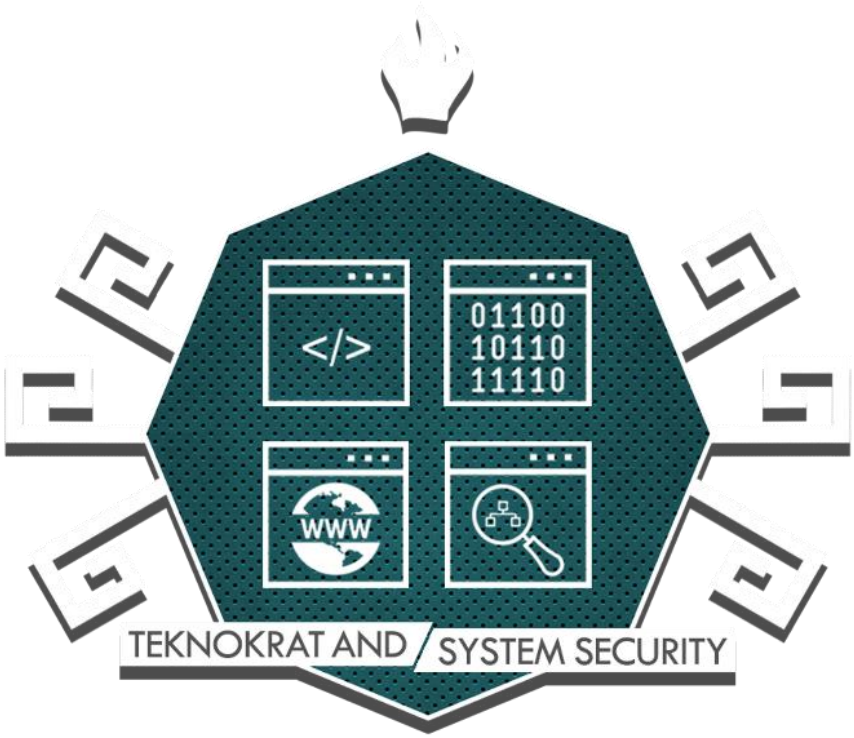
CTF Writeups By

TENESYS

"TWCTF 2nd 2016"

#fredrica



Judul : Welcome!!

Nilai : 10 pt

Kategori : Misc

Problem:
The flag is "`TWCTF{Welcome_To_TW_MMACTF!!}`".

Karena ini adalah challenge contoh, maka langsung saja submit flag yang diberikan yaitu **`TWCTF{Welcome_To_TW_MMACTF!!}`.**

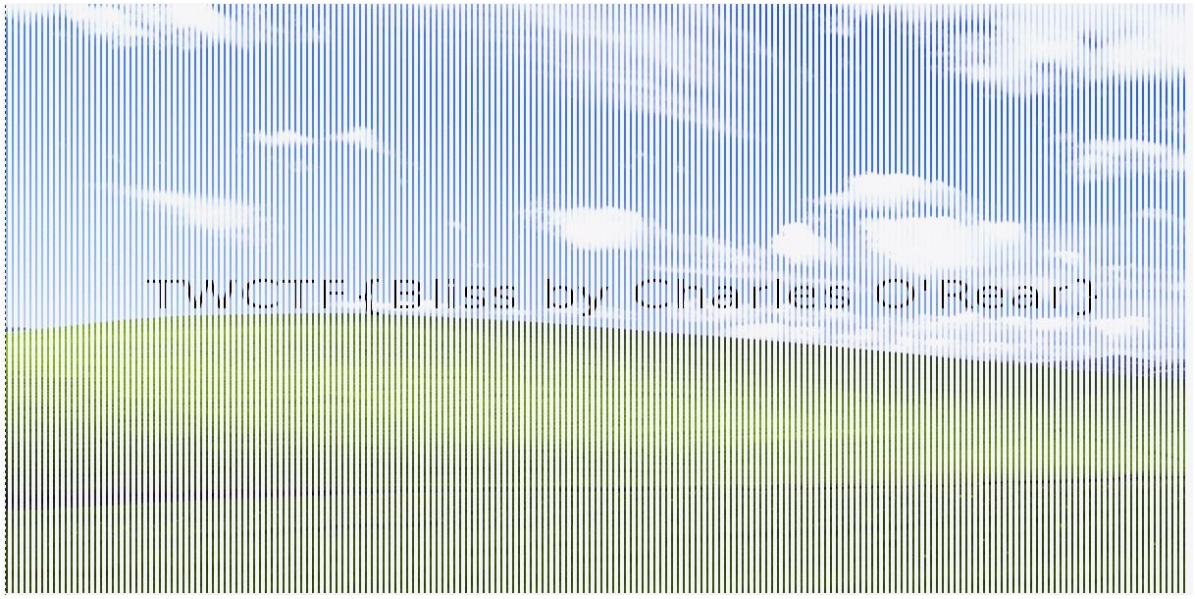
Judul : Glance

Nilai : 50 pt

Kategori : Misc

Problem: I saw this through a gap of the door on a train.

Challenge berupa image gif yang berisi flag. Untuk melihat flag tersebut dapat dilakukan dengan mengekstrak gif menjadi gambar per-frame,



Dan didapatlah flagnya yaitu **`TWCTF{Bliss by Charless O'Rear}`**.

Judul : Make a Palindrome!

Nilai : 50 pt

Kategori : PPC

Problem:

Your task is to make a palindrome string by rearranging and concatenating given words.

```
Input Format: N <Word_1> <Word_2> ... <Word_N>
Answer Format: Rearranged words separated by space.
Each words contain only lower case alphabet characters.

Example Input: 3 ab cba c
Example Answer: ab c cba
```

You have to connect to ppc1.chal.ctf.westerns.tokyo:31111(TCP) to answer the problem.

```
$ nc ppc1.chal.ctf.westerns.tokyo 31111
```

- Time limit is 3 minutes.
- The maximum number of words is 10.
- There are 30 cases. You can get flag 1 on case 1. You can get flag 2 on case 30.
- samples.7z Server connection examples.

Kita diminta untuk terkoneksi ke service *ppc1.chal.ctf.westerns.tokyo* pada port *31111*, kemudian akan muncul soal yang meminta kita untuk menyusun kata sehingga menjadi *polindrom*, seperti contoh diatas. Soal tersebut dapat diselesaikan dengan script seperti dibawah ini,

```
from pwn import *
import sys
import itertools
import numpy
import time

def permute(sequence, index=0):
    length = len(sequence)

    if index > length:
        raise StopIteration

    if index == length:
        yield sequence
    else:
        for i in range(index, length):
            sequence[i], sequence[index] = sequence[index], sequence[i]
            for permutation in permute(sequence, index + 1):
                yield permutation
            sequence[index], sequence[i] = sequence[i], sequence[index]
```

```
def palindrom(n):
        return str(n) == str(n)[::-1]

r = remote('ppc1.chal.ctf.westerns.tokyo',31111)
start = time.clock()
for soal in range(1):
        r.recvuntil("Input: 5 ")
        prob = r.recvline().strip()
        print "[-] Soal ke-1: 5 Words"
        r.recv()
        prob = prob.split(" ")
        pal = prob

        jawab=""
        for aa in permute(pal):
                ab = ' '.join(aa)
                ac = ab.replace(" ","")
                if(palindrom(ac)==True):
                        jawab = ab
                        break
        print "[+] Jawab: "+ jawab
        elapsed = (time.clock() - start)
        print "[+] Waktu: "+str(elapsed)
        r.sendline(str(jawab))

"Flag Pertama = TWCTF{Charisma_School_Captain}"

for soal2 in range(29):
        r.recvuntil("Input: ")
        prob = r.recvline().strip()
        r.recvuntil("Answer: ")
        prob = prob.split(" ")
        print "\n[-] Soal ke-"+str(soal2+2)+": "+str(prob.pop(0))+" Words"
        pal = prob
        jawab=""
        for aa, bb in itertools.izip(permute(pal),permute([pal[(len(pal) + (~i, i)[i%2])
// 2] for i in range(len(pal))])):
                ab = ' '.join(aa)
                bc = ' '.join(bb)
                ac = ab.replace(" ","")
                bd = bc.replace(" ","")
                if palindrom(ac)==True:
                        jawab = ab
                        r.sendline(str(jawab))
                        print "[+] Jawab: "+ jawab
                        elapsed = (time.clock() - start)
                        print "[+] Waktu: "+str(elapsed)
                        break
                if palindrom(bd)==True:
                        jawab = bc
                        r.sendline(str(jawab))
                        print "[+] Jawab: "+ jawab
                        elapsed = (time.clock() - start)
                        print "[+] Waktu: "+str(elapsed)
                        break

print r.recv()
print r.recv()
print r.recv()
"Flag Kedua: TWCTF{Hiyokko_Tsuppari}"
```

Didapatlah flag pertama adalah **TWCTF{Charisma_School_Captain}** dan yang kedua adalah **TWCTF{Hiyokko_Tsuppari}.**

Judul : Twin Primes

Nilai : 50 pt

Kategori : Crypto

Problem:

Decrypt it.
twin-primes.7z

Soal berupa file kompresi dengan ekstensi *.7z*. setelah diekstrak, kita mendapatkan 4 buah file, yang pertama adalah script python dengan nama *encrypt.py* dan 3 buah file lainnya tidak berektensi akan tetapi berisi teks,

encrypt.py

```
from Crypto.Util.number import *
import Crypto.PublicKey.RSA as RSA
import os

N = 1024

def getTwinPrime(N):
    while True:
        p = getPrime(N)
        if isPrime(p+2):
            return p

def genkey(N = 1024):
    p = getTwinPrime(N)
    q = getTwinPrime(N)
    n1 = p*q
    n2 = (p+2)*(q+2)
    e = long(65537)
    d1 = inverse(e, (p-1)*(q-1))
    d2 = inverse(e, (p+1)*(q+1))
    key1 = RSA.construct((n1, e, d1))
    key2 = RSA.construct((n2, e, d2))
    if n1 < n2:
        return (key1, key2)
    else:
        return (key2, key1)

rsa1, rsa2 = genkey(N)

with open("flag", "r") as f:
    flag = f.read()
padded_flag = flag + "\0" + os.urandom(N/8 - 1 - len(flag))

c = bytes_to_long(padded_flag)
c = rsa1.encrypt(c, 0)[0]
c = rsa2.encrypt(c, 0)[0]

with open("key1", "w") as f:
    f.write("%d\n" % rsa1.n)
    f.write("%d\n" % rsa1.e)
with open("key2", "w") as f:
    f.write("%d\n" % rsa2.n)
    f.write("%d\n" % rsa2.e)
```

```
with open("encrypted", "w") as f:
    f.write("%d\n" % c)
```

encrypted

```
799121918959101457219662381738573787902720810846980080262970656425850862601067451387549602917729057581965036680273080328376113703625538076776653886608646389553997359461588232197473814093168933387310612445984932255675457901006254198813821117657462166810122853176982835828997315039334310994861158360921942021353083436483743873041137930504615667001502454726301993228898980822809160120694874130422219777980859273807511102467898227385692258661541523855521114884742758967823874518625364978366560792838200286811127807705487129483792318953671423504404199354115840294337218877979799671179261043996910599391737365184733763892 9
```

key1

```
194026437680279672944806953610372276496375145612804613527084201921973289935127108520878719863491843834420315449452639664774466855871680251547750601787828970979939498008459032188909752757254166992584629200979864249360885411127909588752113361882491072807536614676195110796490702486595362822672679286692652529351844486389978775937819301038664169495856865415096424940485542420041008633152204300749971455319291282008857582740378753495390186693362634698032772810486571981148444132367546805498744727535288664346860487998333815420188763622298426052135008697093616570000441825733088255502379991394420404221079318575068978109516553765
```

key2

```
194026437680279672944806953610372276496375145612804613527084201921973289935127108520878719863491843834420315449452639664774466855871680251547750601787828970979939498008459032188909752757254166992584629200979864249360885411127909588752113361882491072807536614676195110796490702486595362822672679286692652529357574188671723145935466781041001290273392560689409874128167797443399949716651095556804014673244873975418524868057703008950633150839654450998467966738905392320963293379345531703349669197397492241574949069875012089172754014231783160960425531160246267389657034543342990940680603153790486530477470655757947009682859 65537
```

Setelah dianalisa, file encrypted ternyata di enkripsi dengan *RSA*, dengan public key *1024* bits. Untuk mendapatkan flag kita harus mendekripsi file *encrypted*, kali ini kami menggunakan bantuan *Python GUI*,

```
import gmpy2
def num_to_str(num):
        res = ""
        while num > 0:
                res = chr(num % 256) + res
                num = num / 256
        return res
p =
109839168287920364771652233739542245893972429420400471787477887103169099491804762856071669374751286279860451783039232642710981517010937585802203565874477414469934412741906018847402147404957765188018616912003220542453809516059524224015255036266232001320821428611494617812180060212800300789614856560253120304701
q =
176645945799298135110721766377313792982812334295271987596634864064777954683139799946630491521527848390622912482826980130051166690303653228530141163053890146954290070312482492552495214917023922382112893625586133272913759418717134953590760109002220865007673751773346439753002517112721944238066505389966935631251
p2 =
109839168287920364771652233739542245893972429420400471787477887103169099491804762856071669374751286279860451783039232642710981517010937585802203565874477414469934412741906018847
```

```
021474049577651880186169120032205424538095160595242240152550362662320013208214286114946178121800602128003007896148565602531203047 03
q2 =
176645945799298135110721766377313792982812334295271987596634864064777954683139799946630491521527848390622912482826980130051166690303653228530141163053890146954290070312482492552495214917023922382112893625586133272913759418717134953590760109002220865007673751773346439753002517112721944238066505389966935631253

t = (p-1)*(q-1)
t2 = (p2-1)*(q2-1)
d = gmpy2.invert(65537,t)
d2 = gmpy2.invert(65537,t2)

c =
799121918959101457219662381738573787902720810846980080262970656425850862601067451387549602917729057581965036680273080328376113703625538076776653886608646389553997359461588232197473814093168933387310612445984932255675457901006254198813821117657462166810122853176982835828997315039334310994861158360921942021353083436483743873041137930504615667001502454726301993228898980822809160120694874130422219777980859273807511102467898227385692258661541523855521114884742758967823874518625364978366560792838200286811127807705487129483792318953671423504404199354115840294337218877979799671179261043996910599391737365184733763892 9
n =
194026437680279672944806953610372276496375145612804613527084201921973289935127108520878719863491843834420315449452639664774466855871680251547750601787828970979939498008459032188909752757254166992584629200979864249360885411127909588752113361882491072807536614676195110796490702486595362822672679286692652529351844486389978775937819301038664169495856865415096424940485542420041008633152204300749971455319291282008857582740378753495390186693362634698032772810486571981148444132367546805498744727535288664346860487998333815420188763622298426052135008697093616570000441825733088255502379991394420404221079318575068978109512 9
n2 =
194026437680279672944806953610372276496375145612804613527084201921973289935127108520878719863491843834420315449452639664774466855871680251547750601787828970979939498008459032188909752757254166992584629200979864249360885411127909588752113361882491072807536614676195110796490702486595362822672679286692652529357574188671723145935466781041001290273392560689409874128167797443399949716651095556804014673244873975418524868057703008950633150839654450984679667389053923209632933793455317033496691973974922415749490698750120891727540142317831609604255311602462673896570345433429909406806031537904865304774706557579470096828 59
m = pow(c,d2,n2)
m = pow(m,d,n)

print num_to_str(m)
```

Dan didapat flag yaitu **`TWCTF{3102628d7059fa267365f8c37a0e56cf7e0797ef}`**.