217's Blog

3 年多 ago

Olympic CTF 2014 Figure Crypting 200 mic Writeup

原題目 server https://raw2.github.com/csie217/ctf/master/olympic-ctf-2014/mic_server.py
不難的 crypto 一枚。python server 要求一個大質數 p 和基底 g，FLAG 是 `int(flag.encode('hex'),16)`，server 會回傳

```
( pow( g * FLAG, FLAG, p ) * FLAG + FLAG ) % p
```

我們已知 flag 的形式是 CTF{...} 這樣，所以 FLAG 是個奇數。對於一個質數 p，我們選兩個不同基底 g 和 -g，則

```
( pow( g * FLAG, FLAG, p ) * FLAG + FLAG ) % p + ( pow( -g * FLAG, FLAG, p ) * FLAG + FLAG ) % p
= ( pow( g * FLAG, FLAG, p ) * FLAG + FLAG - pow( g * FLAG, FLAG, p ) * FLAG + FLAG ) % p
= ( FLAG * 2 ) % p
```

由於 p 的範圍限制，我們沒辦法直接得到 FLAG。故選三個不同的質數，再用中國餘數定理組合起來

```
import os
import random

def gcd(a, b):
    while b:
        a, b = b, a % b
    return abs(a)

def check_prime(p):
    """Miller-Rabin test"""
    if p & 1 == 0:
        return False
    m = p - 1
    s = 0
    while m & 1 == 0:
        m >>= 1
        s += 1
    for j in range(100):
        a = random.randint(2, p - 2)
        if gcd(a, p) != 1:
            return False
        b = pow(a, m * (1 << s), p)
        if b in (0, 1, p - 1):
            continue
        for i in range(s):
            b = pow(b, 2, p)
            if b == 1:
                return False
            if b == p - 1:
                if i < s - 1:
                    break
                else:
                    return False
        else:
            return False
    return True

n1 = 2**107-1
p1 = 93350567636530487152013873049335L
```

```
q1 = 1435795156635843487341956505543369L
k1 = ((p1+q1)*pow(2,n1-2,n1))%n1

n2 = 1569275433846670190958947355801916604025588861116008628353
p2 = 384733450628775009296876869566896388560082127522333689371
q2 = 1292046548332604151018479737742990026174325571269708515885
k2 = ((p2+q2)*pow(2,n2-2,n2))%n2

n3 = 1292046548332604151018479737742990026174325571269708516021
p3 = 1141556213792204741466186809174330309943785447071641866426
q3 = 539840925477520043245085799373981523213927002400487030840
k3 = ((p3+q3)*pow(2,n3-2,n3))%n3

def ext_euclid(a, b) :
    if b == 0 :
        return (a, 1, 0)
    else :
        (d, xx, yy) = ext_euclid(b, a % b)
        x = yy
        y = xx - (a / b) * yy
        return (d, x, y)

def inverse(a, n):
    return ext_euclid(a, n)[1]

k = 3
a = [k1,k2,k3]
n = [n1,n2,n3]
N = []
b = []
n_product = reduce(lambda x, y: x * y, n, 1)

for i in xrange(0, k):
    N_term = 1
    for j in xrange(0, k):
        if i != j:
            N_term = N_term * n[j]
    N.append(N_term)
for i in xrange(0, k):
    b.append(inverse(N[i], n[i]))
x = 0
for i in xrange(0, k):
    x = x + a[i] * N[i] * b[i] % n_product
x = x%(n1*n2*n3)

print hex(x)[2:-1].decode('hex')
```

CTF{cf5246e06b13432b9e1116ddef226455}

Olympic CTF 2014 Figure Crypting 300 GuessGame Writeup →

- February 10, 2014 00:23
- Permalink