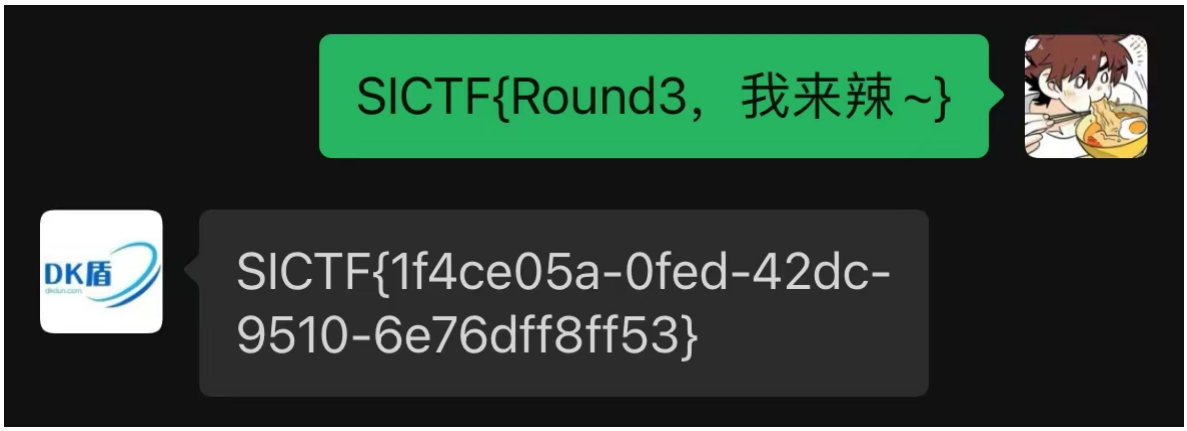
SICTF2024 别管 Writeup

Misc

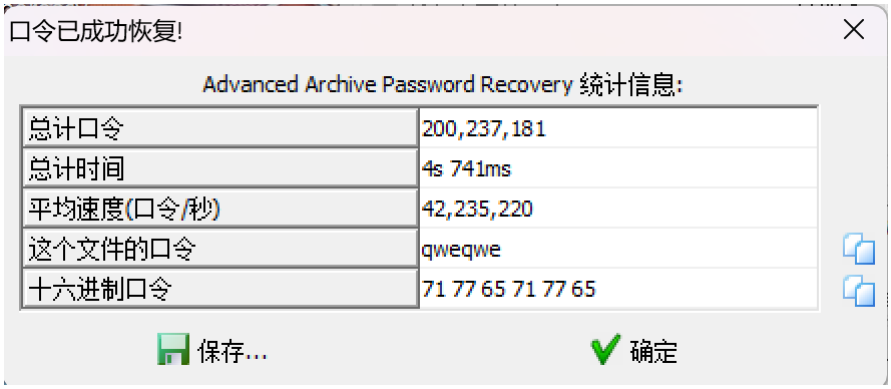
[签到]签到

扫码得到flag

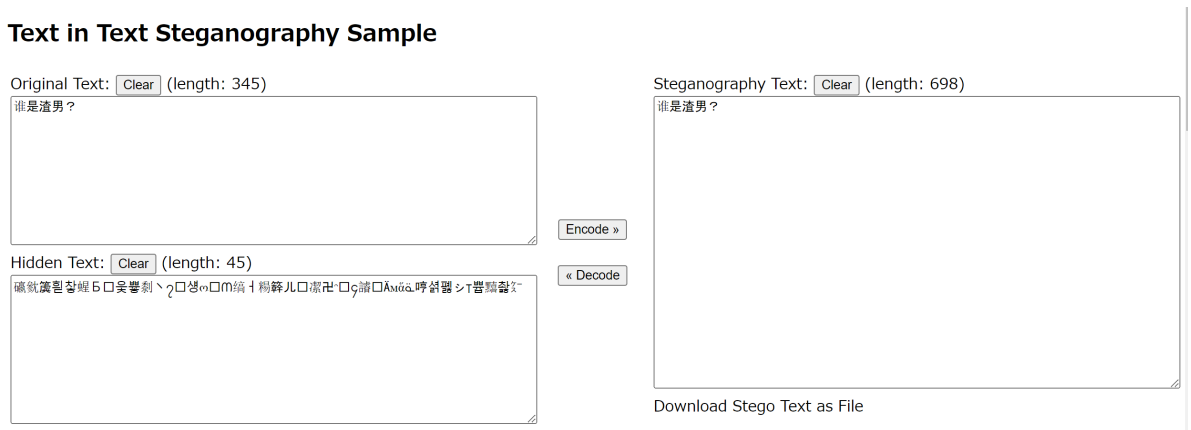


WHO?WHO?WHO

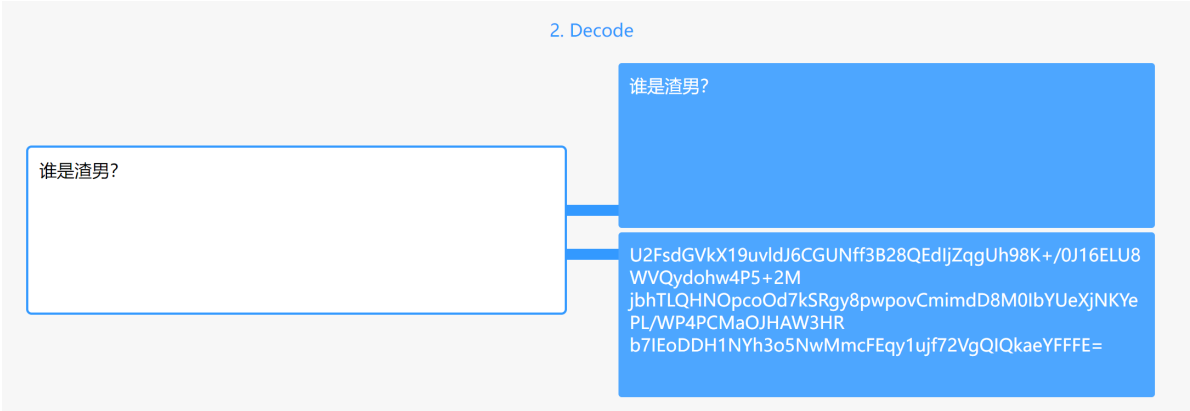
查看题目hint说压缩包密码为6位小写字母，进行密码爆破得到密码为qweqwe



解开压缩包里面有一个文本文件树木的dna里都是渣男.txt，查看文本询问了下队友St4rr是零宽字符，并给我了一个在线网站https://330k.github.io/misc_tools/unicode_steganography.html

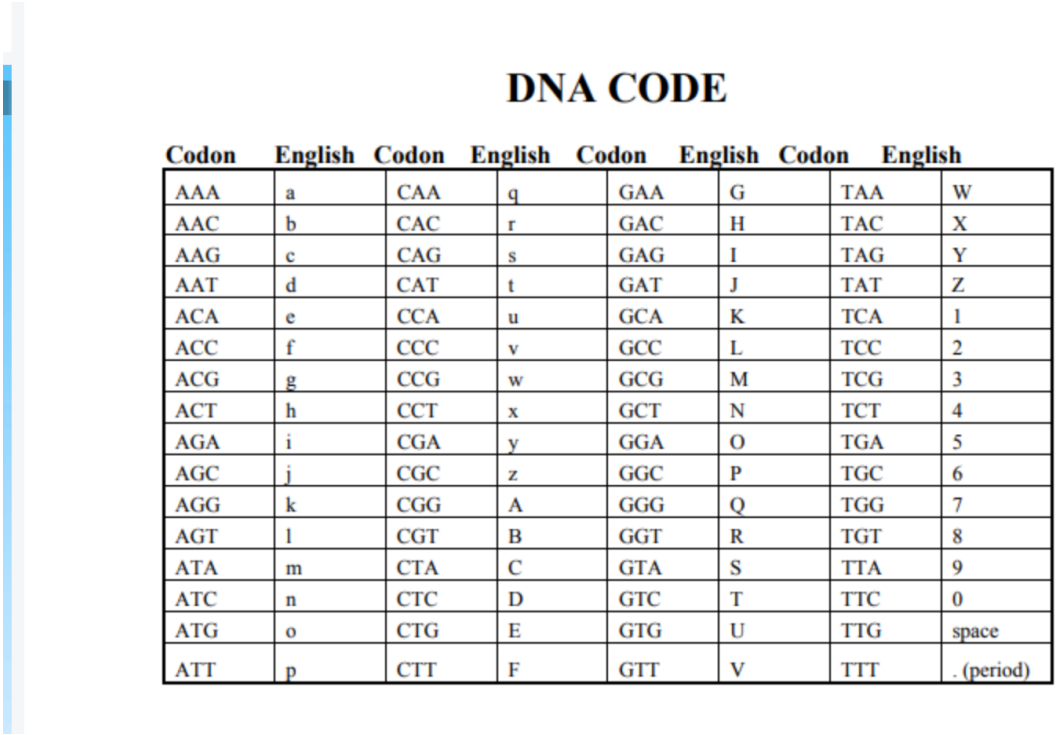


进行解密后发现得到了一串乱码，查询零宽字符概念后知晓必须与出题人的加解密网站一致才能正确解密，于是我在知乎上发现了另一个零宽字符在线解密网站https://yuanfux.github.io/zero-width-web/



得到了正确的解密密文
U2FsdGVkX19uvldJ6CGUNff3B28QEdIjZqgUh98K+/0J16ELU8WVQydohw4P5+2M
jbhTLQHNOpcoOd7kSRgy8pwpovCmimdD8M0IbYUeXjNKYePL/WP4PCMaOJHAW3HR
b7IEoDDH1NYh3o5NwMmcFEqy1ujf72VgQIQkaeYFFFE=，这是一串兔子密码，找到一个兔子密码解密网站http://www.esjson.com/rabbitEncrypt.html，解密得到

GTAGAGCTAGTCCTT{GGGTCACGGTTC_GGGTCACGGTTC_GAACGGTTC_GTAGTG_GCTTCA_GTAGACGTGGCGGTG_GTAGACTCA_TATGACCGG_GCTCGGGCT}，根据提示和高中生物知识，这是一串由核苷酸序列加密而成的密文，我找到了一张解密的 图片



DNA CODE

Codon	English	Codon	English	Codon	English	Codon	English
AAA	a	CAA	q	GAA	G	TAA	W
AAC	b	CAC	r	GAC	H	TAC	X
AAG	c	CAG	s	GAG	I	TAG	Y
AAT	d	CAT	t	GAT	J	TAT	Z
ACA	e	CCA	u	GCA	K	TCA	1
ACC	f	CCC	v	GCC	L	TCC	2
ACG	g	CCG	w	GCG	M	TCG	3
ACT	h	CCT	x	GCT	N	TCT	4
AGA	i	CGA	y	GGA	O	TGA	5
AGC	j	CGC	z	GGC	P	TGC	6
AGG	k	CGG	A	GGG	Q	TGG	7
AGT	l	CGT	B	GGT	R	TGT	8
ATA	m	CTA	C	GTA	S	TTA	9
ATC	n	CTC	D	GTC	T	TTC	0
ATG	o	CTG	E	GTG	U	TTG	space
ATT	p	CTT	F	GTT	V	TTT	. (period)

最终解密得到SICTF{Q1A0_Q1A0_GA0_SU_N1_SHUMU_SH1_ZHA_NAN}

日志分析2

随便翻一翻，攻击者IP、sql注入工具及版本号一目了然

10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL--%20ynh
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20ORDER%20BY%201--%20pBmu&mod=detail&id=12 HTTP/1.1 404 383 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL--%20BkKc&mod=detail&id=12 HTTP/1.1 404 383 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL--%20YcIH&mod=detail&id=12 HTTP/1.1 404 383 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL--%20sZoE&mod=detail&id=12 HTTP/1.1 404 383 - sqlmap
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL--%20fQSE&mod=detail&id=12 HTTP/1.1 404 38
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL--%20BKAR&mod=detail&id=12 HTTP/
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL--%20hEyB&mod=detail&id=
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL--%20DMZg&mod=
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL--%20oaJI
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view%20UNION%20ALL%20SELECT%20NULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL%2CNULL
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=5389&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=6307&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail.%27..%28%2C%28%29%22.&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%27qeaBZf%3C%27%22%3EToXTep&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%27%29%20AND%207301%3D7253%20AND%20%28%27vBZG%27%3D%27vBZG&id=12 HTTP/1.1 200 3
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%27%29%20AND%201219%3D1219%20AND%20%28%27hBDQ%27%3D%27hBDQ&id=12 HTTP/1.1 200
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%27%20AND%201451%3D5627%20AND%20%27GjKL%27%3D%27GjKL&id=12 HTTP/1.1 200 376 - sqlma
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%27%20AND%201219%3D1219%20AND%20%27IyXT%27%3D%27IyXT&id=12 HTTP/1.1 200 376 - sqlmap
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%25%27%20AND%201267%3D8459%20AND%20%27%25%27%3D%27&id=12 HTTP/1.1 200 376 - sqlmap
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%25%27%20AND%201219%3D1219%20AND%20%27%25%27%3D%27&id=12 HTTP/1.1 200 376 - sqlmap
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%29%20AND%206876%3D9233%20AND%20%285625%3D5625&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.1
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%29%20AND%201219%3D1219%20AND%20%287298%3D7298&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.1
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%20AND%201257%3D8368&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%20AND%201219%3D1219&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%20AND%209583%3D1466--%20rYpn&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%20AND%201219%3D1219--%20CPCW&id=12 HTTP/1.1 200 376 - sqlmap 1.2.4.18
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=%28SELECT%20%28CASE%20WHEN%20%288768%3D4675%29%20THEN%208768%20ELSE%208768%2A%28SE
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=%28SELECT%20%28CASE%20WHEN%20%285720%3D5720%29%20THEN%205720%20ELSE%205720%2A%28SE
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%27%29%20AND%20%28SELECT%202954%20FROM%28SELECT%20COUNT%28%2A%29%2CCONCAT%2
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%27%20AND%20%28SELECT%202954%20FROM%28SELECT%20COUNT%28%2A%29%2CCONCAT%280x7
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%25%27%20AND%20%28SELECT%202954%20FROM%28SELECT%20COUNT%28%2A%29%2CCONCAT%2
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%29%20AND%20%28SELECT%202954%20FROM%28SELECT%20COUNT%28%2A%29%2CCONCAT%280x7
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%20AND%20%28SELECT%202954%20FROM%28SELECT%20COUNT%28%2A%29%2CCONCAT%280x717a
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%20AND%20%28SELECT%202954%20FROM%28SELECT%20COUNT%28%2A%29%2CCONCAT%280x717a
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%27%29%20AND%203774%3DCAST%28%28CHR%28113%29%7C%7CCHR%28122%29%7C%7CCHR%281
10.11.35.95 - - [30/Sep/2019:10:27:30 +0800] GET /index.php?action=view&mod=detail%27%20AND%203774%3DCAST%28%28CHR%28113%29%7C%7CCHR%28122%29%7C%7CCHR%28122%
10.11.35.95 - - [30/Sep/2019:10:27:31 +0800] GET /index.php?action=view&mod=detail%25%27%20AND%203774%3DCAST%28%28CHR%28113%29%7C%7CCHR%28122%29%7C%7CCHR%281
10.11.35.95 - - [30/Sep/2019:10:27:31 +0800] GET /index.php?action=view&mod=detail%29%20AND%203774%3DCAST%28%28CHR%28113%29%7C%7CCHR%28122%29%7C%7CCHR%28122%

webshell连接工具和版本号

10.11.35.95 - - [30/Sep/2019:19:11:48 +0800] POST /images/avatar/web.php HTTP/1.1 200 322 - antSword/v2.1

攻击方法试了半天，还以为是什么高深的名词，最后得到是暴力破解。。。

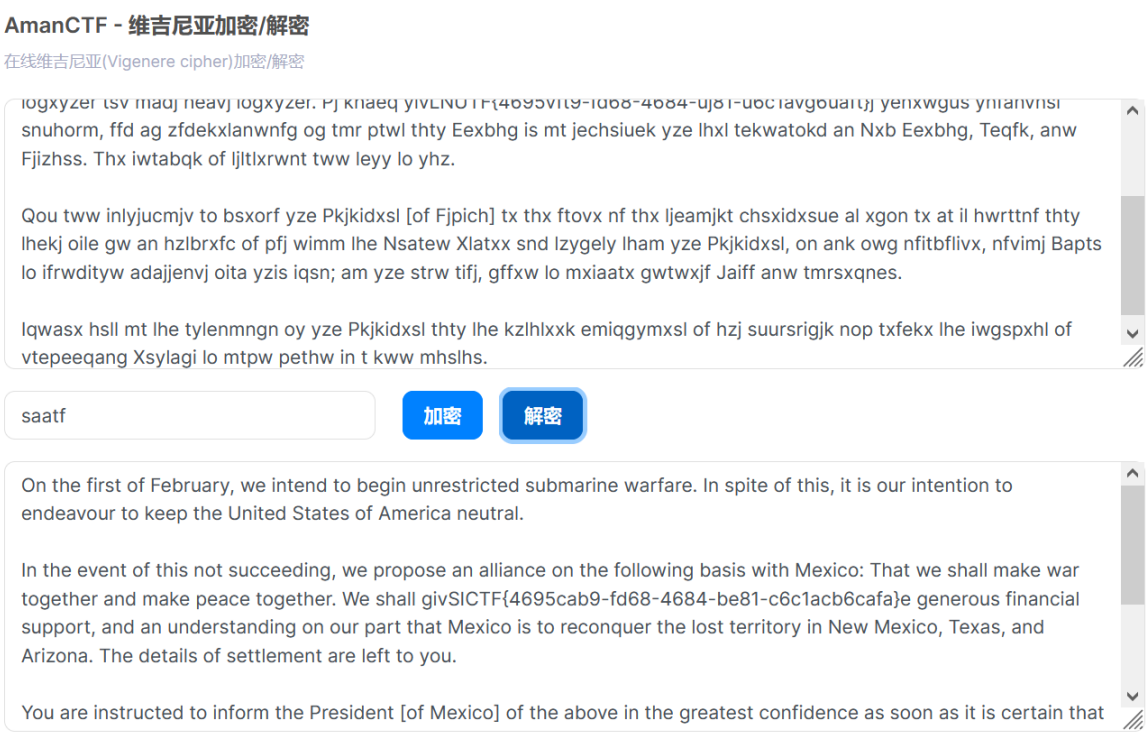
SICTF{10.11.35.95|暴力破解|sqlmap|1.2.4.18|蚁剑|2.1}

Crypto

[签到]Vigenere

签到题

关键是观察一下，试出密钥就行



签到，确信!

题目

```
from Crypto.Util.number import *
from enc import flag

m = bytes_to_long(flag)
def gen_keys(bits):
    while 1:
        p = getPrime(bits)
        q = sum([p**i for i in range(7)])
        if isPrime(q):
            r = getPrime(1024)
            n = p*q*r
            return p,n
p,n = gen_keys(512)
e = 65537
c = pow(m,e,n)
print(f"n = {n}")
print(f"e = {e}")
print(f"c = {c}")
'''
```

```
n =
836136162456319116861286371051644902828075763293460341214315292518684772182155287933860895112015763118269976283374309783736874052605573651608013652058484811313708758188642633519120768880706302409612800140669821799881678233565566380354485349606041893156954557139784964382658423443104900239477287726360304973672307139298982493920236263140916443471593866203879564131418962873061497821798786815065149134316152644789456924177009037763360205856123932945004603624719374588517429536563341148212164440864808904601696047910022085095300992777895030475433901354101953641388026407445643390767167004928831794554049549661553115091664705015893601009503741233466256104601616377757573695234982738003993852616871565564956695270878848510412690072300326401951388889794217589000771102628894168725696201279926438754589283276230432028759257560268367384539998403927235092980321749261750260100561377897610970184282900836522625949284813441781853562982776934226202077511569547221887643055702647128252604254519594406307852327934145919947591120396676275138133427777162367406370214163113252430285699973033413173945253458791885239489916984896677949120524362450639986373768741515538094245813760687198145322461792978512068625059524373012533136608762311362858772149490949954589976302357646350595280161490066137202871029418685172445098548756728874450997339099125988957437074204546239977401434072060903195675311441260900723314e = 65537c =9901744183419446581636823550814851552652879282998060853149162655806576725134936985605804849074322077308871320622426407567066959374032686829120831485688661470112475104398373409453344511101251825953979206020747750224164549189546236124495846375847163438062559170905259042012848525788342324478217168292530656109893179091887844263289515208661529362798918721839544393484493594915263606711521937352600990771989862643645680468343990645143505383299909851310529476700636056111137302461289268502424718207099571586091753768679937004117383142374000385844708269149464344983224307417975702599362662263256678145218384207330613359690712455806571875441617726198895188453486396728202127090302279999637445937151949285026069104527776873573614033404646237092067644786266390652682476817862879933305687452549301456541574678459748029511685529779653056108795644495442515066731075232130730326258404497646551885443146629498236191794065050199535063169471112533284663197357635908054343683637354352034115772227442563180462771041527246803861110504563589660801224223152060573760388045791699221007556911597792387829416892037414283131499832672222157450742460666013331962249415807439258417736128976044272555922344342725850924271905056434303543500959556998454661274520986141613977331669376614647269667276594163516040422089616099849315644424644920145900066426839607058422686565517159251903275091124418838917480242517812783383'''
```

用分圆多项式先分解出p

```
from Crypto.Util.number import *
P.<x, y> = PolynomialRing(ZZ)
R.<z> = PolynomialRing(ZZ)
z = R.gens()[0]
def calculate_eta_all(eta, aa, bb, m, k):
    eta_all = []
    for i in range(k):
        temp = eta**(aa**i)
        add = temp
        for _ in range((m-1)//k - 1):
            add = add**bb
            temp += add
        eta_all.append(temp)
    return eta_all
```

```
def calculate_irreducible_polynomial(eta_all, m):
    h = 1
    for i in range(k):
        h *= (y - eta_all[i].lift())

    d = sum([x**i for i in range(m)])
    f_irreducible = h % d

    return f_irreducible, d

def pad_polynomial_coefficients(f, m):
    tmp = f.list()
    while len(tmp) < m:
        tmp.append(0)
    return tmp

def Factoring_with_Cyclotomic_Polynomials(k, n):

    if k == 1:
        print('k = 1')
        a = 2
        while True:
            print('a =', a)
            p = gcd(int(pow(a, n, n)-1), n)
            if p > 2**20 and n % p == 0:
                return p
            a += 1

    Phi = cyclotomic_polynomial(k)
    Psi = (z**k-1)//(cyclotomic_polynomial(k))
    print('Cyclotomic_Polynomials Phi:', Phi)
    print('Psi:', Psi)
    m = 1
    while True:
        useful = False
        while not useful:
            m += k
            if not isPrime(m):
                continue

            aa = primitive_root(m)
            ff = x**m - 1
            Q = P.quo(ff)
            eta = Q.gens()[0]
            for bb in range(2, m):
                if (bb**((m-1)//k)-1)//(bb-1) % m:
                    continue
                eta_all = calculate_eta_all(eta, aa, bb, m, k)
                f_irreducible, d = calculate_irreducible_polynomial(eta_all, m)
                if f_irreducible.subs(y=0) in ZZ:
                    useful = True
                    break

        print(aa, bb)
        print(m)

        eta0 = eta_all[0]
```

```
        eta0_pow = []
        for i in range(2, k):
            eta0_pow_i = (eta0**i).lift().subs(x=z)
            constant_term = eta0_pow_i.list()[0]
            if constant_term != 0:
                dd = (d-1).subs(x=z)
                eta0_pow_i = eta0_pow_i - constant_term - constant_term * dd
            eta0_pow.append(eta0_pow_i)

        coefficients = []
        for i in range(k):

coefficients.append(pad_polynomial_coefficients(eta_all[i].lift().subs(x=z), m))

        A = matrix(QQ, coefficients)
        terget = [[-1]*k, [1] + [0]*(k-1)]
        for i in range(k-2):

terget.append(A.solve_left(vector(pad_polynomial_coefficients(eta0_pow[i], m))))

        B = matrix(QQ, terget)

        U.<w> = PolynomialRing(QQ)
        w = U.gens()[0]
        eta1 = U(list((B**-1)[1]))
        f = f_irreducible.subs(y=w)
        V = U.quo(f)
        eta1 = V(eta1)

        C = matrix(QQ, k, k)
        C[0, 0] = 1
        for i in range(1, k):
            tmp = eta1**i
            C[i] = pad_polynomial_coefficients(tmp, k)

        K.<s> = PolynomialRing(Zmod(n))
        f_modulo = f_irreducible.subs(y=s)
        K_quo = K.quo(f_modulo)

        f_ZZ = f_irreducible.subs(y=z)
        try:
            sigma = matrix(Zmod(n), C)
        except:
            continue
        while True:
            g = R.random_element(k - 1)
            try:
                kk, _, h = xgcd(f_ZZ, g)
                h = inverse_mod(int(kk), n) * h
                break
            except:
                continue
        g = g.subs(y=x)
        g_Q = K_quo(g)
        h_Q = K_quo(h)
        assert g_Q * h_Q == 1

        Psi_coefficients = Psi.coefficients()
```

```
        Psi_monomials = Psi.monomials()[::-1]
        if Psi_coefficients[0] < 0:
            yy = h_Q**(-Psi_coefficients[0])
        else:
            yy = g_Q**(Psi_coefficients[0])

        for i in range(1, len(Psi_monomials)):
            if Psi_coefficients[i] < 0:
                yy *= K_quo(list(vector(list(h_Q**(-Psi_coefficients[i]))) *
Psi_monomials[i](sigma)))
            else:
                yy *= K_quo(list(vector(list(g_Q**(Psi_coefficients[i]))) *
Psi_monomials[i](sigma)))
        yy = yy**n
        if gcd(yy[1], n) > 2**20:
            return gcd(yy[1], n)

k=7
n=836136162456319116861286371051644902828075763293460341214315292518684772182155
287933860895112015763118269976283374309783736874052605573651608013652058484811131
370875818864263351912076888070630240961280014066982179988167823356556638035448453
496060418931569545571397849643826584234431049002394772877263603049736723071392985
982493920236263140916443471593866203879564131418962873061497821798786815065149135
431615264478945692417700903776336020585612393294500460362471937458851742953656335
411482121644408648089046016960479100220850953009927778950304754339013541019536413
388026407445643390767167004928831794554049549661553115091664705015893601009503741
233466256104601616377757573695234982738003993852616871565564956695270878848510412
690072300326401951388889794217589000771102628894168725696201279926438754589283762
304320287592575602683673845399984039272350929803217492617502601005613778976109
701842829008365226259492848134417818535629827769342262020775115695472218876430557
026471282526042545195944063078523279341459199475911203966762751381334277716236
740637021416311325243028569997303341317394525345879188523948991698489667794912052
436245063998637376874151553809424581376068719814532246179297851206862505952437301
253313660876231136285877214949094995458997630235764635059528016149006613720287
102941868517244509854875672887445099733909912598895743707420454623997740143407206
090319567531144126090072331
pp = Factoring_with_Cyclotomic_Polynomials(k, n)
assert not n % pp
print('factor is found:', pp)
```

然后正常解RSA即可：

```
import gmpy2
from Crypto.Util.number import long_to_bytes
e = 65537
c =
990174418341944658163682355081485155265287928299806085314916265580657672513493698560580484907432207730887132062242640756706695937403268682912083148568866147011247510439837340945334451110125182595397920602074775022416454918954623612449584637584716343806255917090525904201284852578834232447821716829253065610989317909188784426328951520866152936279891872183954439348449359491526360671152193735260099077198986264364568046834399064514350538329990985131052947670063605611113730246128926850242471820709957158609175376867993700411738314237400038584470826914946434498322430741797570259936266226325667814521838420733061335969071245580657187544161772619889518845348639672820212709030227999963744593715194928502606910452777687735614033404646237092067644786266390652682476817862879933305687452549301456541574678459748029511685529779653056108795644495442515066731075232130730326258404497646551885443146629498236191794065050199535063169471112533284663197357635908054343683637354352034115772227442563180462771041527246803861110504563589660801224223152060573760388045791699221007556911597792387829416892037414283131499832672222157450742460666013331962249415807439258417736128976044272555922344342725850924271905056434303543500959556998454661274520986141613977331669376614647269667276594163516040422089616099849315644424644920145900066426839607058422686565517159251903275091124418838917480242517812783383
p=126829015671222220278622672495980835310426055339942919549630946921063178346006271705414824055696722631276799343671895359519031178525002782790009209546289 51
q=sum([p**i for i in range(7)])
n =
836136162456319116861286371051644902828075763293460341214315292518684772182155287933860895112015763118269976283374309783736874052605573651608013652058484811313708758188642633519120768880706302409612800140669821799881678233565566380354485349606041893156954557139784964382658423443104900239477287726360304973672307139298982493920236263140916443471593866203879564131418962873061497821798786815065149134316152644789456924177009037763360205856123932945004603624719374588517429536563341148212164440864808904601696047910022085095300992777895030475433901354101953641388026407445643390767167004928831794554049549661553115091664705015893601009503741233466256104601616377757573695234982738003993852616871565564956695270878848510412690072300326401951388889794217589000771102628894168725696201279926438754589283276230432028759257560268367384539998403927235092980321749261750260100561377897610970184282900836522625949284813441781853562982776934226202077511569547221887643055702647128252604254519594406307852327934145919947591120396676275138133427771623674063702141631132524302856999730334131739452534587918852394899169848966779491205243624506399863737687415155380942458137606871981453224617929785120686250595243730125331366087623113628587721494909499545899763023576463505952801614900661372028710294186851724450985487567288744509973390991259889574370742045462399774014340720609031956753114412609007233 1
r=n//p//q
phi=(p-1)*(q-1)*(r-1)
d=gmpy2.invert(e,phi)
m=pow(c,d,n)
print(long_to_bytes(m))
```

```
b'SICTF{d9428fc7-fa3a-4096-8ec9-191c0a4562ff}'
```

gggcccddd

题目

```
from Crypto.Util.number import *
from enc import flag

m = bytes_to_long(flag)

p = getPrime(512)
q = getPrime(512)
n = p*q
e = 65537
c1 = pow(m,e,n)
c2 = pow(233*m+9527,e,n)
print(f'n = {n}')
print(f'c1 = {c1}')
print(f'c2 = {c2}')
print(f'e = {e}')
"""
n =
714517843544880788325574408410671398875328208671609461464627655292620217564924155977594376450001987464388460664458351084386563179365118381988602102247387285025584207069475335448634288026547369704693130305843341335196447464987814619277627367691159332491959172070592971459655029556155994815755077389391884151915191
c1 =
602373050531823636860660008607559705431195494605857633667601830239690605297978213984511741458161543292584051436938727290682551550867342178836588064943711058897525987094460681591511662506355587749379246685062716243738719529829064595099045488335671174022678264777283679283851378578002562704285378820881104964841
c2 =
205635624489021368248826364689528951802539834493392269547383991633413322725718822097849964862501899121218709465779158816384154840435341610717823873589937129186787873980656889998107341892139046935145195949555224601517694795153230498219402854082280557713496709195875609525488767962526341049263670781777330762531
e = 65537
"""
```

相关明文攻击。但是这里的e太大，没法使用Franklin-Reiter算法，于是使用Half-GCD算法。

exp:

```
import sys
from Crypto.Util.number import long_to_bytes
def HGCD(a, b):
    if 2 * b.degree() <= a.degree() or a.degree() == 1:
        return 1, 0, 0, 1
    m = a.degree() // 2
    a_top, a_bot = a.quo_rem(x^m)
    b_top, b_bot = b.quo_rem(x^m)
    R00, R01, R10, R11 = HGCD(a_top, b_top)
    c = R00 * a + R01 * b
    d = R10 * a + R11 * b
    q, e = c.quo_rem(d)
    d_top, d_bot = d.quo_rem(x^(m // 2))
    e_top, e_bot = e.quo_rem(x^(m // 2))
    S00, S01, S10, S11 = HGCD(d_top, e_top)
```

```
    RET00 = S01 * R00 + (S00 - q * S01) * R10
    RET01 = S01 * R01 + (S00 - q * S01) * R11
    RET10 = S11 * R00 + (S10 - q * S11) * R10
    RET11 = S11 * R01 + (S10 - q * S11) * R11
    return RET00, RET01, RET10, RET11

def GCD(a, b):
    print(a.degree(), b.degree())
    q, r = a.quo_rem(b)
    if r == 0:
        return b
    R00, R01, R10, R11 = HGCD(a, b)
    c = R00 * a + R01 * b
    d = R10 * a + R11 * b
    if d == 0:
        return c.monic()
    q, r = c.quo_rem(d)
    if r == 0:
        return d
    return GCD(d, r)

P.<x> = PolynomialRing(Zmod(n))
sys.setrecursionlimit(500000)

n =
714517843544880788325574408410671398875328208671609461464627655292620217564924155977594376450001987464388460664458351084386563179365118381988602102247387285025584207069475335448634288026547369704693130305843341335196447464987814619277627367691159332491959172070592971459655029556155994815755077389391884151911
c1 =
602373050531823636860660008607559705431195494605857633667601830239690605297978213984511741458161543292584051436938727290682551550867342178836588064943711058897525987094460681591511662506355587749379246685062716243738719529829064595099045488335671174022678264777283679283851378578002562704285378820881104966841
c2 =
205635624489021368248826364689528951802539834493392269547383991633413322725718822097849964862501899121218709465779158816384154840435341610717823873589937129186787873980656889998107341892139046935145195949555224601517694795153230498219402854082280557713496709195875609525488767962526341049263670781777330762531
e = 65537
f = (x)^e - c1
g = (233* x+9527)^e - c2

km = GCD(f,g)
m = -km.monic()[0]
print(m)
print(long_to_bytes(int(m)))
```

```
65537 65537
32768 32767
16384 16383
8192 8191
4096 4095
2048 2047
1024 1023
512 511
256 255
128 127
64 63
32 31
16 15
8 7
4 3
2 1
116587369900739681441164092706025039667760954097630578268846176988820166911349004063514813098260958549 73
b'SICTF{45115fb2-84d6-4369-88c2-c8c3d72b4c55}'
```

SuperbRSA

题目

```
#user:mumu666
from Crypto.Util.number import *
p=getPrime(1024)
q=getPrime(1024)
n=p*q
e1=55
e2=200
m=bytes_to_long("flag")
assert(pow(m,5) < n)
c1 = pow(m, e1, n)
c2 = pow(m, e2, n)
print("n=",n)
print("c1=",c1)
print("c2=",c2)

n=
190068303581189023924324535958026755667308503528902469959206428119678212593880090498035131027505945241064717096412020198326824380273124688492999858326751917954171605533795808134107223590898725193720492292337324059930624642868888890846408787842090141658716968825648348963225080542317779670111956365644638062709983269361614490099884342491784771001273474067599321490107120913761837101356153752726718885412332754157371559533231334396445297098987918817951867758302178846630444959790678074187584552377013150196838024373231771254930761134197398274302823110180839761141581599254507467120646395693019256727421862942371131990 23
c1=
276245243658976720066605903875366763552720328374098965164676247771817997950424168480909517684516498439306387133611184795758628248588201187138612090081389226321683486308199743311842513053259894661221013008371261704678716150646764446208833447643781574516045641493770778735363586857160147826684394417412837449465273160781074676966630398315417741542529612480836572205781076576325382832502694868883931680720558621770570349864399879523171995953720198118660355479626037129047327185224203109006251809257919143284157354935005710902589809259500117996982503679601132486140677013625335552533104471327456798955341220640782369529
```

```
c2=
11734019659226247713821792108026989060106712358397514827024912309860741729438494
68948053187583328726845466985956871905389634647136075002795222663317355959406446
68504137375042678075994356796165220262411118872941381232011047188497443007696769
61585732810579953221056338076885840743126397063074940281522137794340822594577352
36161659870214347737914528468742770591383188549351261694450461247427840590927718
81188968824418124696794944592164314051394785481921528114411691761347500790733170
11232934250365454908280676079801770043968006983848495835089055956722848080915898
15135224221521007101133109876182803178630027677100183902l
```

明显是共模攻击，但是可以观察到e1和e2不互素，因此需要对算法稍作改进，exp:

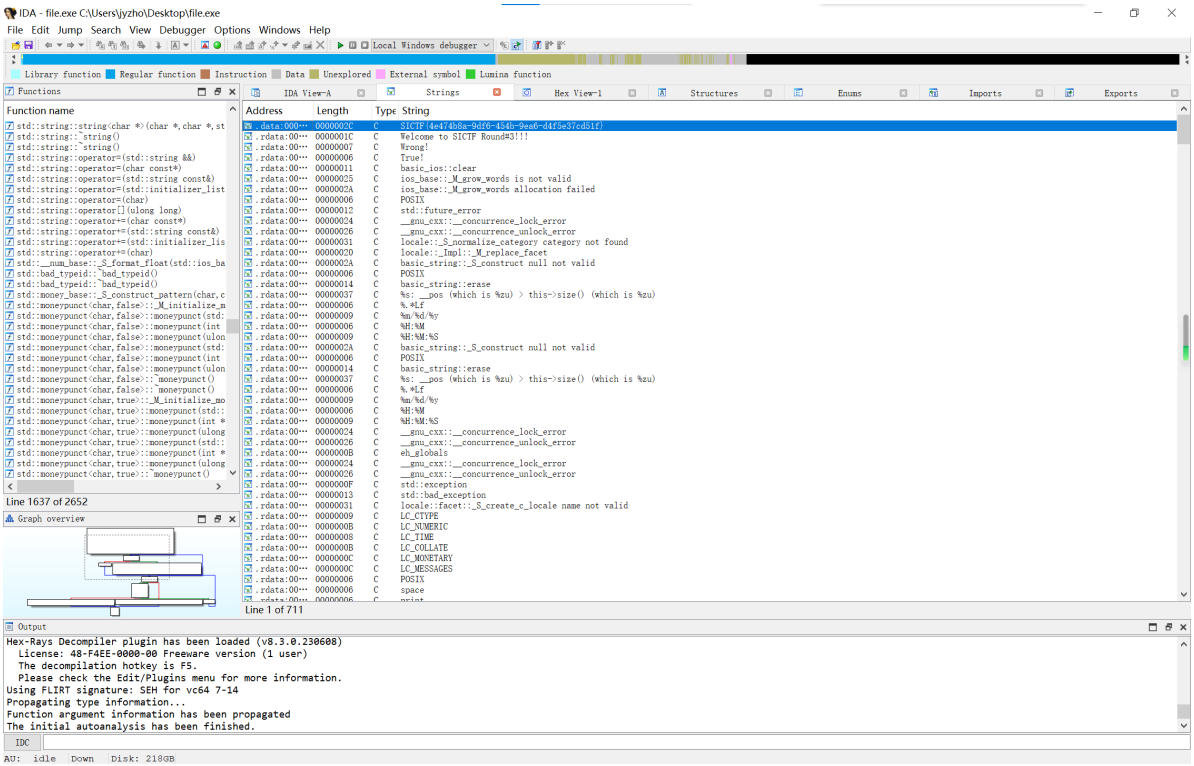
```
import gmpy2
from Crypto.Util.number import *

n=
19006830358118902392432453595802675566730850352890246995920642811967821259388009
04980351310275059452410647170964120201983268243802731246884929998583267519179541
71605533795808134107223590898725193720492292337324059930624642868888890846408787
84209014165871696882564834896322508054231777967011195636564463806270998326936161
44900998843424917847710012734740675993214901071209137618371013561537527267188854
12332754157371559533231334396445297098987918817951867758302178846630444959790678
07418758455237701315019683802437323177125493076113419739827430282311018083976114
15815992545074671206463956930192567274218629423711319902 3
c1=
27624524365897672006660590387536676355272032837409896516467624777181799795042416
84809095176845164984393063871336111847957586282485882011871386120900813892263216
83486308199743311842513053259894661221013008371261704678716150646764446208833447
64378157451604564149377077873536358685716014782668439441741283744946527316078107
46769666303983154177415425296124808365722057810765763253828325026948688839316807
20558621770570349864399879523171995953720198118660355479626037129047327185224203
10900625180925791914328415735493500571090258980925950011799698250367960113248614
06770136253355525331044713274567989553412206407823695 29
c2=
11734019659226247713821792108026989060106712358397514827024912309860741729438494
68948053187583328726845466985956871905389634647136075002795222663317355959406446
68504137375042678075994356796165220262411118872941381232011047188497443007696769
61585732810579953221056338076885840743126397063074940281522137794340822594577352
36161659870214347737914528468742770591383188549351261694450461247427840590927718
81188968824418124696794944592164314051394785481921528114411691761347500790733170
11232934250365454908280676079801770043968006983848495835089055956722848080915898
15135224221521007101133109876182803178630027677100183902l

e1 =55
e2 =200

g = gmpy2.gcd(e1, e2)
print(g)

l, x, y = gmpy2.gcdext(e1//g, e2//g)
print(x, y)

m = pow(c1, x, n)*pow(c2, y, n) % n
m = gmpy2.iroot(m, g)[0]
print(long_to_bytes(m))
```

```
5
11 -3
b'SICTF{S0_Great_RSA_Have_Y0u_Learned?}'
```

Reverse

Baby_C++

签到题，丢进IDA即可看到flag



[Game][Battle City]

~~根本不会reverse，但我会打游戏~~

解开压缩包开始进行坦克大战的经典游戏，打过第四关后即可得到flag

在assets文件夹中得到了win.png的图片，打开是个二维码

扫码得到

SICTF{Y0u_@Re_bat71e_C1ty_Ma5t3r}

Forensics

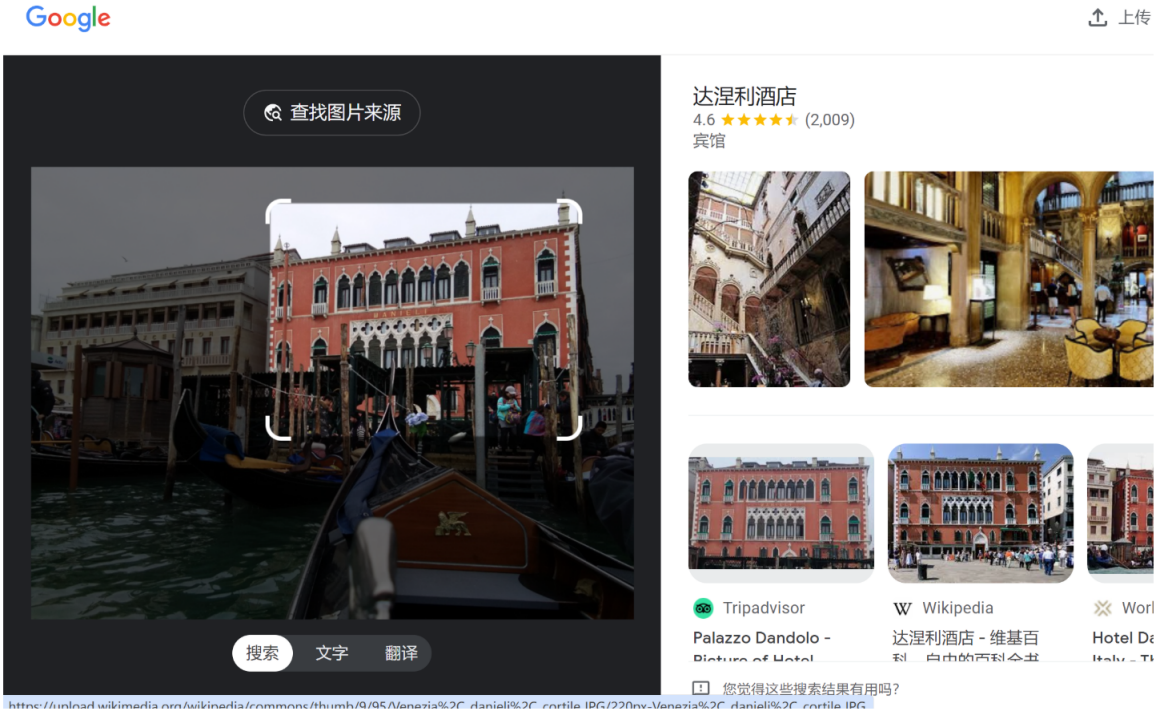
[签到]OSINT签到

找到了一张与所给图片一致的图片，根据文章内容我们得到此地为海南省 海口市 琼山区 红城湖公园。

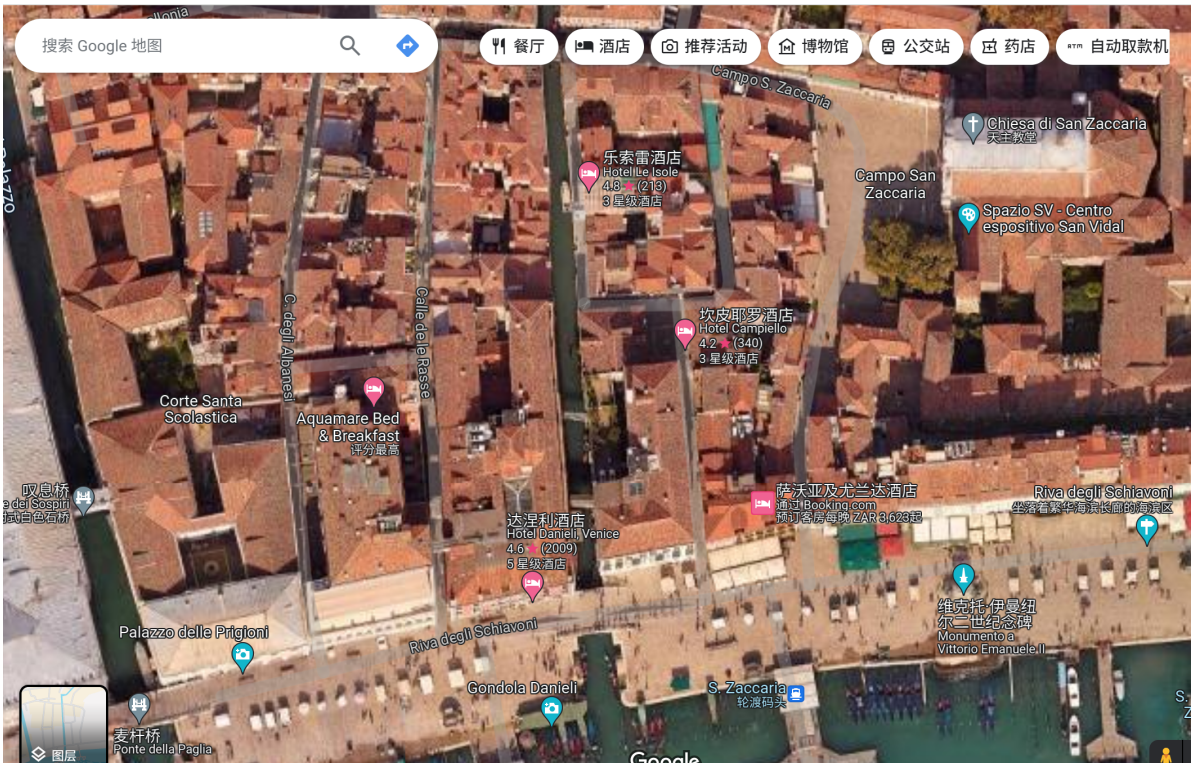
SICTF{海南省_海口市_琼山区_红城湖公园}

这才是签到

根据谷歌识图得到这里的大致位置是意大利的威尼斯



根据所在位置我们可以推测拍摄地点为Gondola Danieli



再根据提示的20m的比例尺地图和聊天记录中的小道提示~~（虽然我根本不觉得这叫小道）~~

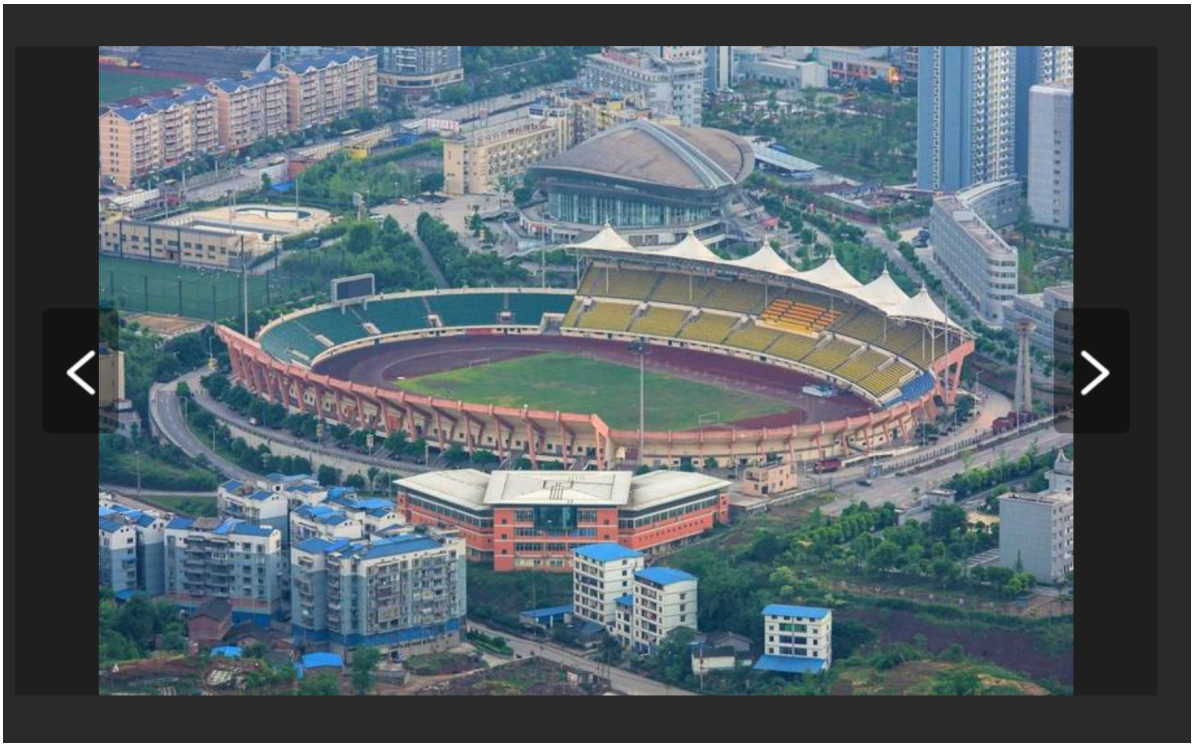
最终得到目的地是天主教堂ChiesadiSanZaccaria

SICTF{意大利_威尼斯_GondolaDanieli_ChiesadiSanZaccaria}

树木的压迫

放进百度图片里面识图（直接识别红色框框的局部）

找到一张极为相似的图片



点进文章里看是四川达州的体育中心，在百度地图里搜索一下得到

SICTF{四川省_达州市_通川区_凤凰大道376号_达州市体育中心}

真的签到

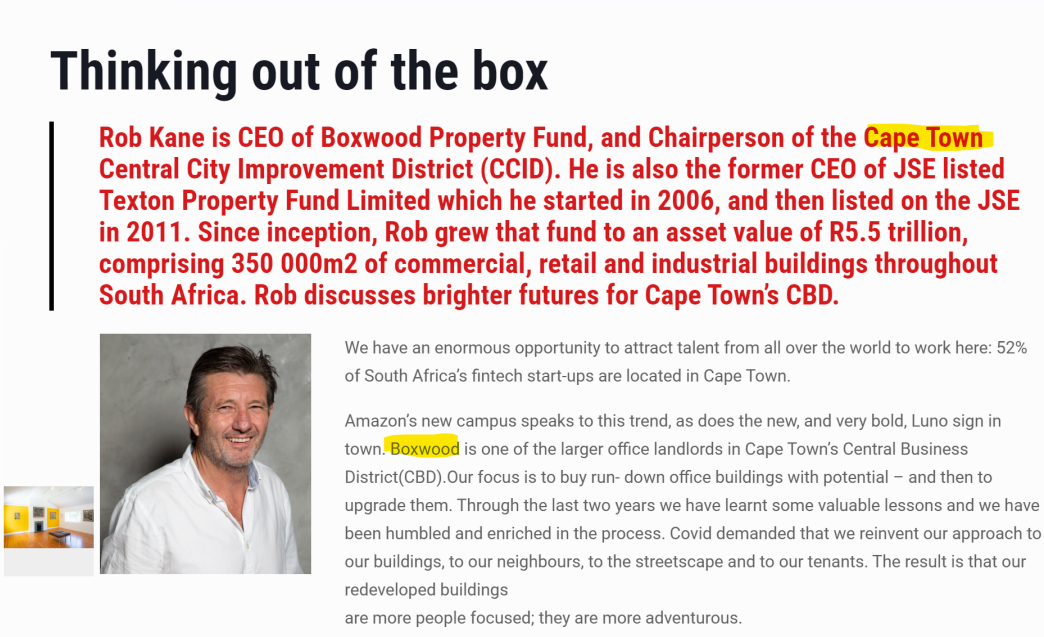
截取局部的标志性建筑摩天轮上去识图

得到一张极为相似的图片上面直接表明了地址，查询后得知这是广东珠海的摩天轮

SICTF{广东省_珠海市_斗门区_大信新都汇}

签退

这题应该是五道社工题里面最难的一道题，分析图中的信息我们可以得到一个有特点的景物蜘蛛侠和他的大大的蜘蛛丝，于是我们可以局部识图得到了一篇似乎是描述当地总裁新盖大楼并且搞了个蜘蛛侠logo的文章https://www.bigissue.org.za/thinking-out-of-the-box/



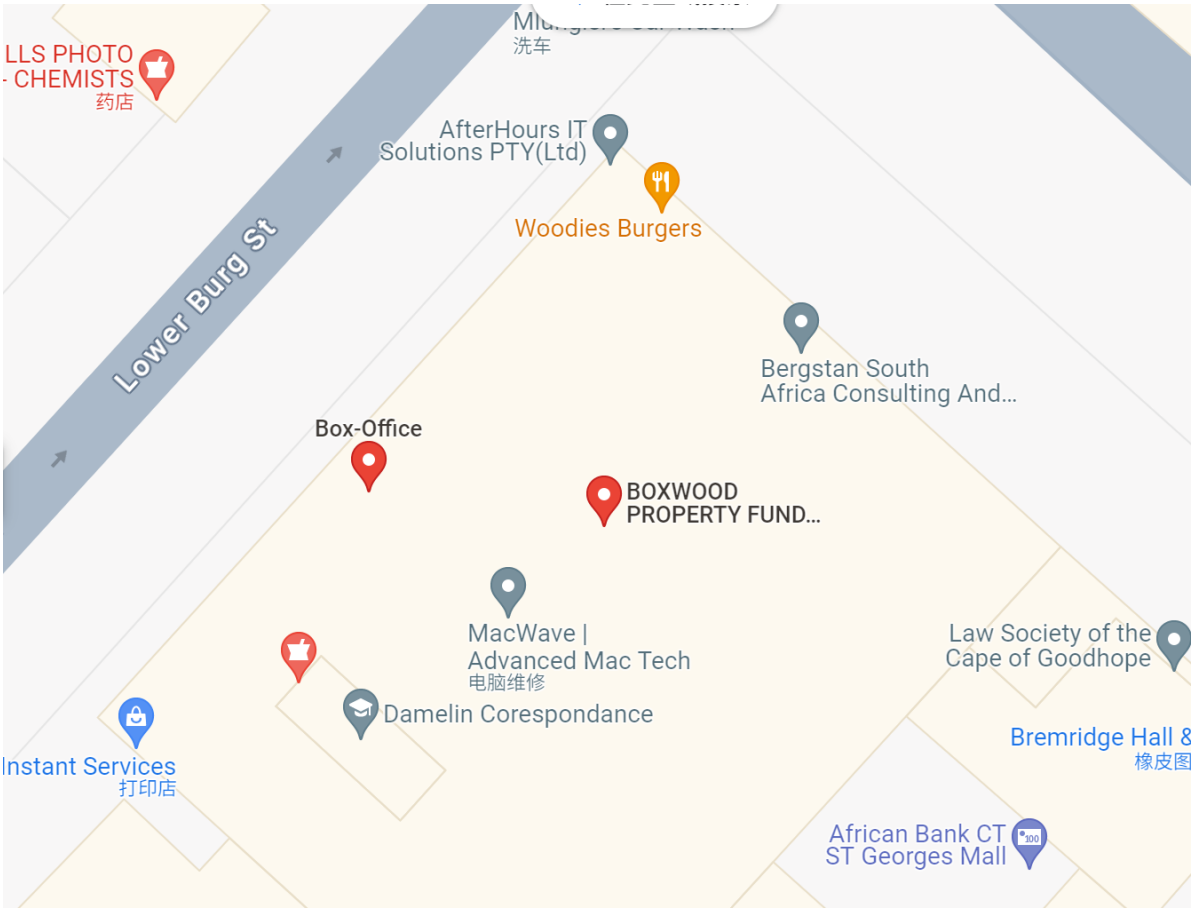
Thinking out of the box

Rob Kane is CEO of Boxwood Property Fund, and Chairperson of the Cape Town Central City Improvement District (CCID). He is also the former CEO of JSE listed Texton Property Fund Limited which he started in 2006, and then listed on the JSE in 2011. Since inception, Rob grew that fund to an asset value of R5.5 trillion, comprising 350 000m2 of commercial, retail and industrial buildings throughout South Africa. Rob discusses brighter futures for Cape Town's CBD.

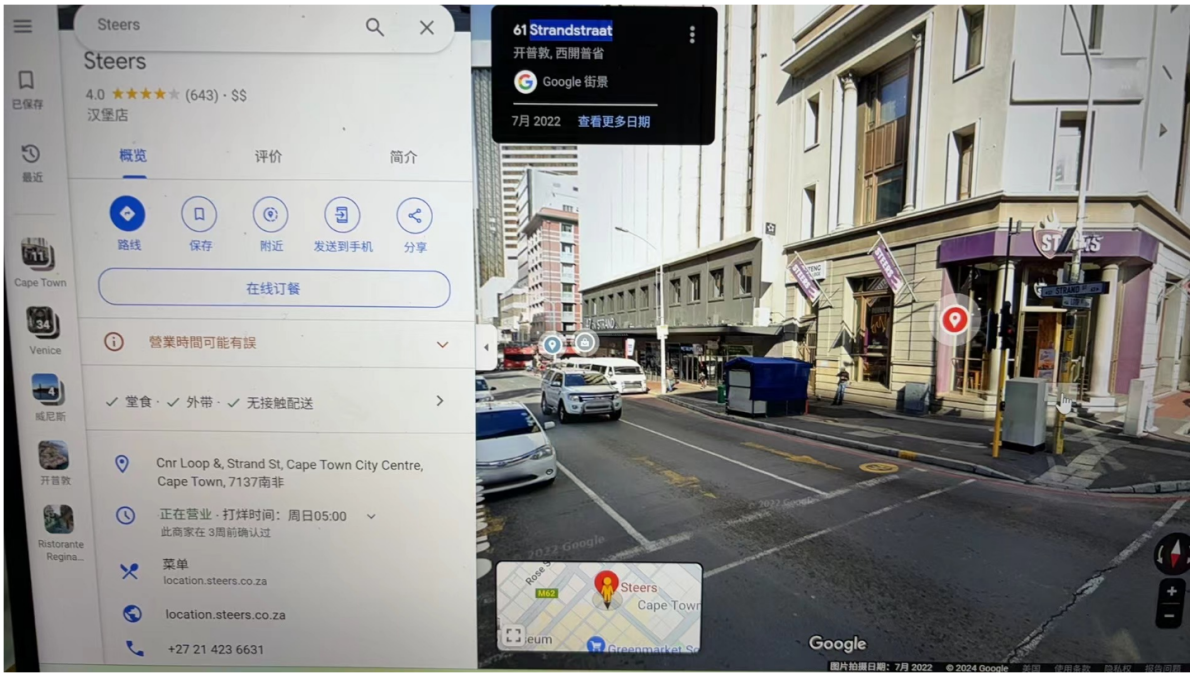
We have an enormous opportunity to attract talent from all over the world to work here: 52% of South Africa's fintech start-ups are located in Cape Town.

Amazon's new campus speaks to this trend, as does the new, and very bold, Luno sign in town. Boxwood is one of the larger office landlords in Cape Town's Central Business District(CBD).Our focus is to buy run- down office buildings with potential – and then to upgrade them. Through the last two years we have learnt some valuable lessons and we have been humbled and enriched in the process. Covid demanded that we reinvent our approach to our buildings, to our neighbours, to the streetscape and to our tenants. The result is that our redeveloped buildings
are more people focused; they are more adventurous.

里面有两个地名关键词，CapeTown和Boxwood，我们去谷歌地图搜索可以得知capetown是南非的一座城市，中文名开普敦，搜索找到了该地



我在地图上其实没有发现那家店铺的名字，但是我通过谷歌实景在这附近转了一圈又一圈，最终在不经意间发现了这家汉堡店的实景



街道名称地图上也有显示，我们猜测这就是拍摄地点，虽然他拍摄的角度是有一点斜着的，但还是在这条街道上的。最终我们得到SICTF{南非_开普敦_StrandSt_STEERS}，（店铺的名称是它实景的名称，不是地图上介绍的名称）