CTFSHOW easyrsa1-6 Writeup

原创

abtgu 于 2020-09-17 15:51:01 发布 1054 收藏 5

分类专栏： CTF Python 密码学 文章标签： 信息安全

本文链接：https://blog.csdn.net/weixin_43790779/article/details/108562895


CTF 同时被 3 个专栏收录

22 篇文章 1 订阅

订阅专栏


Python

7 篇文章 0 订阅

订阅专栏


密码学

10 篇文章 2 订阅

订阅专栏

easyrsa1

题目： 无

解题思路： 题中给了e，n，c，可以分解n得到p，q，在线分解大整数网址http://www.factordb.com/index.php 。脚本如下：

```
import gmpy2
import binascii

e = 65537
n = 145592552973435810546140653225991179080734761646499106530184 7
c = 693803710579142461926067606861522332256595033663193320650 09
p = 120114705943853078683536519456 7
q = 121211263707786291719219191384 1

phi = (p-1)*(q-1)
d = gmpy2.invert(e,phi)
m = gmpy2.powmod(c,d,n)

print(binascii.unhexlify(hex(m)[2:]))
```

easyrsa2

题目： 无

解题思路： 两组数中e相同，n，c不同，求出n1与n2的最大公因数即为p，之后就可以得到q和d，从而求解m。

```
import gmpy2
import binascii

e = 65537
n1 = 236865639255375777530472290407542829533522217241544953906873588777753801476051524555379885634907169438725175
932128583261468115111033118657530183291093146237022070738828842513725532259861120068271113515010449722392722006 1
687171632526541611503889080511482931511195031918318959128382179323799904442788793453683581352674875961296310337 7
803089900662509399569819785571492828112437312659229879806168758843603248823629821851053775458651933952183988482 1
639500392484872704538882884275403055428241799517344120449853648665321248037460081397630818867813614883046665754 5
668041180650509496342540117551041686492960122055615856944374 7
c1 = 162748414223789761394460782826898119391141740806482454071194519203564908810413303814740022407058841033519066
268223118999758008468042420949530307806120512284890464831921964658872099401924927986346298101532948372474782399 1
513714172478886306703290044871781158393304147301058706003793357846922086994952763485999282741595204008663847963 5
394220963433914645270685990469462793090372128599313033355074551460013903265506685316654932452938390098324686683 9
082028266498406639905140322799006803222638222217347807850588823874958323798064369840500568924792290134220414283 3
875409505180847943212126302482358445768662608278731750064815

n2 = 222576053205255840781808890735232239739241929843538471371646051869566296759389295853863923276720655243381764
024964140140838164465088605308877425833388803174788625123066330616015104049600951439413208471605620505240728602 1
177252247849474221364389002744399218336267897042604676563094664433909314913914338875279493280695658988450356917 5
226850419271095336798456238899009883100793515744579945854481430194879360765346236418019384644095257242811629393 1
644024982610660773393048752122508979184204278140001427512828059806320898671085253354880189400916986098909952524 1
300707372585039607627202718342229768466756571202219905428971 1
c2 = 274260069544183655946955370283109837594864191540910697615784037797812391200739875362346111265979620991886698
548047191139336279775362447953764680251042041503946183211801884903058067524981757692685836354168313577723932200 2
741820145944286109172066259843766755795255913189902403644721138554935991439893850589677849639263080528599197595 7
059275354309424631848916894100780590904746826948864200222306576611579938759316009327638246187734200772736171062 9
766019517992201887539917434686340471042016649701719642458611653591571296514714177502654987063632819569077425999 0
189286665844641289108474834973710730426105047318959307995062

p = gmpy2.gcd(n1,n2)
q = n1 // p
phi = (p-1)*(q-1)

d = gmpy2.invert(e,phi)
m = gmpy2.powmod(c1,d,n1)

print(binascii.unhexlify(hex(m)[2:]))
```

easyrsa3

题目： 无

解题思路： 已知n,e1,e2,c1,c2，求m，考查共模攻击。

共模攻击详细知识点请看 。

```
import gmpy2
import binascii

n = 159444754310880532855802297963099560665215201072768179690795509195866505354592425430361433608657807300447330269454885113908189474407675426589562723803893881123720847606897771413923702538507353075784459882897146473328679355250104821977242284575921501849798194637117530585695206512051136903970031461059724084528549485122237029573034065773487173487531068683569956161168677247642762343916788996627742724198418766521261276846837528805684076050836066888841200549639749307572759134479081857122045771942748343683232391430088875542647460683377094653191068866186438499615510923778431840672176159032290680101172728346024692935710
e1 = 797
c1 = 111575932649208254457700163571419961243685298997507452566844501890702881811074230448461655932180134650538396614015954172366579208741138399744718834930998463970027212705900594149811016866687215483306304689513539105646964455095569569552320593866257258830381033990280105667320740113255436506729828842369519044101410777289292614770836890951615969792139614947166375029803582989443166368293091697943243947422851753776018264732760067950725185108507349417031944179265664469802625124295902536435610982758529704619130261080906084915073003653916390815553161665269322337875660538273553490223965637696972782395771845036272441709300
e2 = 521
c2 = 669927435185333002311784039645037594879768240959567056099989882603837804015785993988802186133843135017219396105431448747696503022838137265973319755159773039427536081146240185398840400692271003905358647124437628201948769130786574162199197753907360136889283422719128666380923658672919687627700583849531863936557563898913757279284331091522003947672268455455333711693032367182922052856257316929590149643785832773050499279975372446576016180582072357808766873758170468215899102850214374444543577545829690767140718492168331737121672921405638129247414166802780160032718744337585839457701539410881327377464142718441188754684 9

s = gmpy2.gcdext(e1,e2)
m1 = gmpy2.powmod(c1,s[1],n)
m2 = gmpy2.powmod(c2,s[2],n)

m = (m1*m2)%n

print(binascii.unhexlify(hex(m)[2:]))
```

easyrsa4

题目： 无

解题思路： 题中e=3相对于n，c来说极小，故可知是**低加密指数攻击**。

①

当$m < n$时，$c = m^e$，直接将c开三次方即可得到m。

```
import gmpy2
import binascii

e = 3
n = 189700537286166093664582860677312887490222649591584037583579859153933831179636938275688099257706793537656248108049043822788455264989814223463194179384348615582913667385420791651697362325586878217099373465034807562814897758594392546144724250175540511777251430681221859615526706462752290095315286785482518734210766916508275078298592993002726832239592676612886016198459544663651340775476998197344653213457584169572656821758642272735062507073117757979798340909070208630994679071199579678941722227477621516745009373563920297414877818366750215020226517547121383368598844556881961208526891778071894547257376536558816394575476 1
c = 150409620528139732054476072280993764527079006992643377862720337847060335153837950368208902491767027770946661

i = 0
while True:
    if gmpy2.iroot((c+i*n),3)[1] == True:
        m = gmpy2.iroot((c+i*n),3)[0]
        break
    i += 1

print(binascii.unhexlify(hex(m)[2:]))
```

easyrsa5

题目： 无

解题思路： 题中e很大，故可知是低解密指数攻击。

可以使用破解脚本：求出d的值，文件下载地址https://github.com/pablocelayes/rsa-wiener-attack

（注意，这里要将破解脚本和rsa-wiener-attack的py文件放在同一个目录下）

```
import gmpy2
import binascii
import RSAwienerHacker

e = 284100478693161642327695712452505468891794410301906465434604643365855064101922252698327584524956955373553355814138784402605517536436009073372339264422522610010012877243630454889127160056358637599704871937659443985644871453345576728414422489075791739731547285138648307770775155312545928721094602949588237119345
n = 468459887279781789188886188573017406548524570309663876064881031936564733341508945283407498306248145591559137207097347130203582813352382018491852922849186827279111555223982032271701972642438224730082216672110316142528108239708171781850491578433309964093293907697072741538649347894863899103340030347858867705231
c = 350429162418561525458539070186062788413426454598897326594935655762503536409897624028778814302849485850451243934994919418665502401195173255808119461832488053305530748068788500746791135053620550583421369214031040191188956883213974500055288799870361839225786458401670096126619033993124192536949283773989393928 27

d = RSAwienerHacker.hack_RSA(e,n)
m = gmpy2.powmod(c,d,n)

print(binascii.unhexlify(hex(m)[2:]))
```

easyrsa6

题目：

```
import gmpy2,libnum
from Crypto.Util.number import getPrime
from secret import flag

e = 0x10001
p = getPrime(1024)
q = gmpy2.next_prime(p)
n = p * q
print("n =",n)
m = libnum.s2n(flag)
c = pow(m,e,n)
print("c =", c)

# n = 26737417831000820542131903300607349805884383394154602685589253691058592906354935906805134188533804962897170
211026684453428204518730064406526279112572388086653330354347467824800159214965211971007509161988095657918569122896
964026831303423482648738347983551251763397375408443800189322573267198507765491780971966509718019598298918977829537
998195402581811869718871223297465323483102168188464976445205532183633361948554980093398383691146494536181013219
993473678005819599335967344570817623787467063715992156686864599065530070188122976580153538036264096067074602109
052163626469403557376798899123990142375025293738042883042705630
# c = 18343406988553647441155363755415469675162952205929092244387144604220598930987120971635625205531679665588524
624774972379282080365368504475385813836796957675346369136362299791881988434459126442243685599469468046961707420163
849755187402196540739689823324440860766040276525600017446640429559755587590377841083082073283783044180553080312
093936655426279610008234238497453986740658015049273023492032325305925499263982266317509342604959809805578180715
819784421086649380350482836529047761222588878122181300629226379468397199620669975860711741390226214613560571952382
040172091951384219283820044879575505273602318856695503917257
```

解题思路： 因为p和q很相近，所以可以使用yafu分解n。

```
import gmpy2
import binascii
from Crypto.Util.number import getPrime

e = 0x10001
n = 26737417831000820542131903300607349805884383394154602685589253691058592906354935906805134188533804962897170211026684453428204518730064406526279112572388086653330354347467824800159214965211971007509161988095657918569122896402683130342348264873834798355125176339737540844380018932257326719850776549178097196650971801959829891897782953799819540258181186971887122329746532348310216818846497644520553218363336194855498009339838369114649453618101321999347367800581959933596734457081762378746706371599215668686459906553007018812297658015353803626409606707460210905216362646940355737679889912399014237502529373804288304270563
c = 18343406988553647441155363755415469675162952205929092244387144604220598930987120971635625205531679665588524624774972379282080365368504475385813836796957675346369136362299791881988434459126442243685599469468046961707420163849755187402196540739689823324440860766040276525600017446640429559755587590377841083082073283783044180553080312093936655426279610008234238497453986740658015049273023492032325305925499263982266317509342604959809805578180718157819784421086649380350482836529047761222588878122181300629226379468397199620669975860711741390226214613560571952382040172091951384219283820044879575505273602318856695503917257

p = 16351580300008134123346207756475416525496048953685071026135530571368556329633228535709249310011384460304092516906466456358002541299972005777192095326848477328093991873851763091694212058332799432146216954444966602498816759741414883574323734121841401305035622951591529495243732143584175671896386802091721473858801
q = 16351580300008134123346207756475416525496048953685071026135530571368556329633228535709249310011384460304092516906466456358002541299972005777192095326848477328093991873851763091694212058332799432146216954444966602498816759741414883574323734121841401305035622951591529495243732143584175671896386802091721473851631
phi = (p-1)*(q-1)
d = gmpy2.invert(e,phi)
m = gmpy2.powmod(c,d,n)

print(binascii.unhexlify(hex(m)[2:]))
```