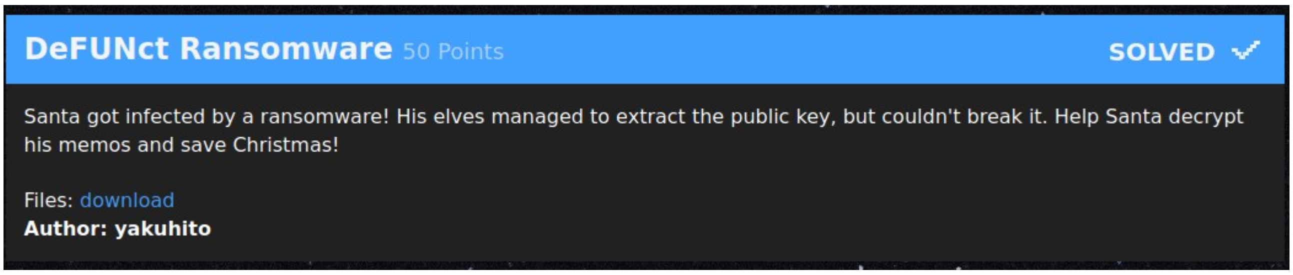
DeFUNct Ransomware



DeFUNct Ransomware 50 Points

SOLVED

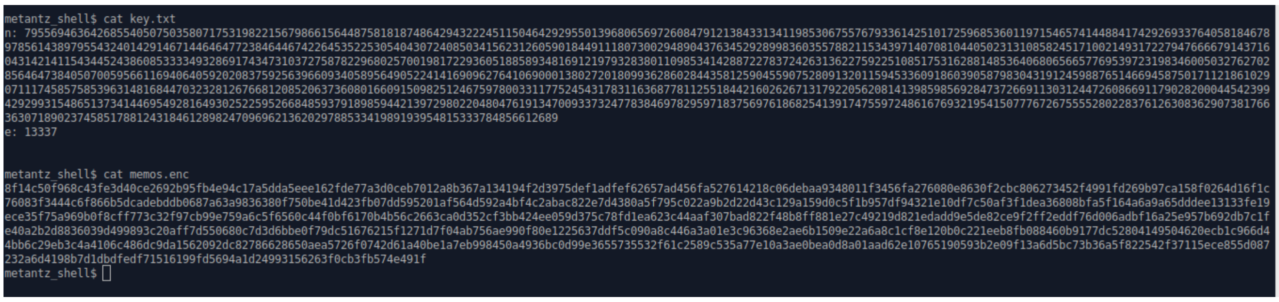
Santa got infected by a ransomware! His elves managed to extract the public key, but couldn't break it. Help Santa decrypt his memos and save Christmas!

Files: download

Author: yakuhito

We have 2 files for this challenge:

key.txt and **memocs.enc**:

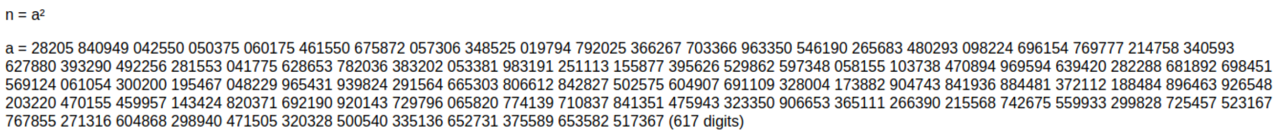


```
metantz_shell$ cat key.txt
n: 795569463642685540507503580717531982215679866156448758181874864294322245115046429295501396806569726084791213843313411985306755767933614251017259685360119715465741448841742926933764058184678
9785614389795543240142914671446464772384644674226453522530540430724085034156231260590184491118073002948904376345292899836035578821153439714070810440502313108582451710021493172279476666791437 16
0431421411543445243860853333493286917434731037275878229680257001981722936051885893481691219793283801109853414288722783724263136227592251085175316288148536406806566577695397231983460050327 62702
8564647384050700595661169406405920208375925639660934058956490522414169096276410690001380272018099362860284435812590455907528091320115945336091860390587983043191245988765146694587501711218 61029
0711174585758539631481684470323281267668120852063736080166091509825124675978003311775245431783116368778112551844216026267131792205620814139859856928473726691130312447260866911790282000445 42399
4292993154865137341446954928164930252259526684859379189859442139729802204804761913470093373247783846978295971837569761868254139174755972486167693219541507776726755552802283761263083629073 81766
363071890237458517881243184612898247096962136202978853341989193954815333784856612689
e: 13337


metantz_shell$ cat memos.enc
8f14c50f968c43fe3d40ce2692b95fb4e94c17a5dda5eee162fde77a3d0ceb7012a8b367a134194f2d3975def1adfef62657ad456fa527614218c06debaa9348011f3456fa276080e8630f2cbc806273452f4991fd269b97ca158f0264d16f1c
76083f3444c6f866b5dcadebddb0687a63a9836380f750be41d423fb07dd595201af564d592a4bf4c2abac822e7d4380a5f795c022a9b2d22d43c129a159d0c5f1b957df94321e10df7c50af3f1dea36808bfa5f164a6a9a65dddee13133fe19
ece35f75a969b0f8cff773c32f97cb99e759a6c5f6560c44f0bf6170b4b56c2663ca0d352cf3bb424ee059d375c78fd1ea623c44aaf307bad822f48b8ff881e27c49219d821edadd9e5de82ce9f2ff2eddf76d006adbf16a25e957b692db7c1f
e40a2b2d8836039d499893c20aff7d550680c7d3d6bbe0f79dc51676215f1271d7f04ab756ae990f80e1225637ddf5c090a8c446a3a01e3c96368e2ae6b1509e22a6a8c1cf8e120b0c221eeb8fb088460b9177dc52804149504620ecb1c966d4
4bb6c29eb3c4a4106c486dc9da1562092dc82786628650aea5726f0742d61a40be1a7eb998450a4936bc0d99e3655735532f61c2589c535a77e10a3ae0bea0d8a01aad62e10765190593b2e09f13a6d5bc73b36a5f822542f37115ece855d087
232a6d4198b7d1dbdfedf71516199fd5694a1d24993156263f0cb3fb574e491f
metantz_shell$ 
```

Ok it seems like RSA because the author gives us n and e.
Let’s try to find out what are the factors of n using the site https://www.alpertron.com.ar/ECM.HTM

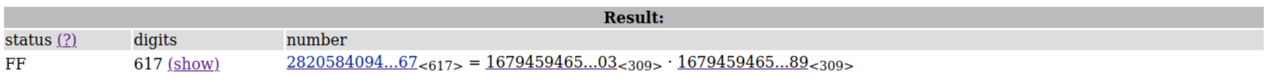
We see that $n = a^2$.



$n = a^2$

a = 28205 840949 042550 050375 060175 461550 675872 057306 348525 019794 792025 366267 703366 963350 546190 265683 480293 098224 696154 769777 214758 340593 627880 393290 492256 281553 041775 628653 782036 383202 053381 983191 251113 155877 395626 529862 597348 058155 103738 470894 969594 639420 282288 681892 698451 569124 061054 300200 195467 048229 965431 939824 291564 665303 806612 842827 502575 604907 691109 328004 173882 904743 841936 884481 372112 188484 896463 926548 203220 470155 459957 143424 820371 692190 920143 729796 065820 774139 710837 841351 475943 323350 906653 365111 266390 215568 742675 559933 299828 725457 523167 767855 271316 604868 298940 471505 320328 500540 335136 652731 375589 653582 517367 (617 digits)

But ‘a’ is not a prime number ! Let’s try to factorize ‘a’ using http://www.factordb.com just to use another site.



Result:		
status (?)	digits	number
FF	617 (show)	2820584094...67<617> = 1679459465...03<309> · 1679459465...89<309>

Ok this time the factors are two primes. We have

p =
1679459465097105285011471408501364447579364859002334943509203652966184664910387838884593403769625721766584714336724461050425691669300667640674587609544445423157230297272758960555944850647902479102165152696728090632087369569515902375008457798680996161107304944572478619713379001443617324249619360419080326395O3

q =
1679459465097105285011471408501364447579364859002334943509203652966184664910387838884593403769625721766584714336724461050425691669300667640674587609544445511813792910480405524843920120796121252379619305104906820721025144998836513427665103996523173354617886861358746087228514782733736695519462452625686010672 89

Now we can decrypt our message with **solve.py**.