

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: 79556946364268554050750358071753198221567986615644875818187486429432224511504642929501396806569726084791213843313411985306755767933614251017259685360119715465741448841742926933764058184678
978561438979554324014291467144646477238464467422645352253054043072408503415623126059018449111807308294890437634529289983603557882115343971407081044050231310858245171062149317227947666679143716
6431421411543445243608053334920869174347310372587822668025700198172293605188589348169121079328801109853414288722783724263186227592251085175318681485364068065665776953972319834609532762702
856464738405070059566116040640592020837592563966093405895640052241416080627641069000138027201809936286028443581250645506752809132011504533660918663985798304319124598876514660458750171121861029
071117458575853963148168447032328126768812085206373688016609158982512467597880331177524543178311636877811255184421602626713173220562081413985985692847372669113031244726886691179028208044542399
4292993154865137341469549281649302522595268485937918985944213972980220480476191347009337324778384697829597183756976186825413917475597248616769321954150777672675555280228376126388362907381766
363071890237458517881243184612898247096962136202978853341989193954815333784856612689
e: 13337

metantz_shell$ cat memocs.enc
0114c50f968c43fe3d48ce2692095fb4e94c17a5d5e5ee162fde77a3d0ceb7012a8b367a134104fd23975def1adfe62657ad456fa527614218c06debaa9348011f3456fa276080e0630f2cbc06273452f4991fd269b97ca158f0264d16f1c
76083f3444c6f866b5dcadeb8a63a9836380f750be41d423fb07dd595201af564d592a4bf4c2abac822e7d4380a5f795c022a9b2d2d43c129a159d0c5f1b957df94321e10df7c50af3f1dea36800bfa5f164a6a9a65dddee13133fe19
e0c35f75a969b0f8cfff773c32f97cb99e759a6c5f6560c44f0bf6170b4b56c2663ca0d352cf3bb424ae059d375c78fd1ea623c44aaf307bad822f48b8ff881e27c49219d821edadd9e5de82ce9f2ff2eddf76d006adbf16a25e957b692db7c1f
e40a2b2d88368039d499893c20aff7d550680c7d3d6bb0ef79dc51676215f1271d7f04ab756ae990f80e1225637ddfd5c090a8c446a3a01e3c96368e2ae6b1509e22a6a8c1cf8e120b0c221eeb8fb088460b9177dc52804149504620ecb1c966d4
4bb6c29eb3c4a106c486dc9da1562092dc82786628650aaa5726f0742d61a40be1a7eb998450a4936bc0d99e365573532f61c2589c535a77e10a3ae0bea0d8a01aad62e10765190593b2e09f13a6d5bc73b36a5f822542f37115ec855d087
232a6d4198b7d1dbdfedf71516199fd5694a1d24993156263f0c3fb574e91f
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 another <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

$p =$

```
16794594650971052850114714085013644475793648590023349435092036529661846649103878
38884593403769625721766584714336724461050425691669300667640674587609544445423157
23029727275896055594485064790247910216515269672809063208736956951590237500845779
868099616110730494457247861971337900144361732424961936041908032639503
```

q =

16794594650971052850114714085013644475793648590023349435092036529661846649103878
38884593403769625721766584714336724461050425691669300667640674587609544445511813
79291048040552484392012079612125237961930510490682072102514499883651342766510399
652317335461788686135874608722851478273373669551946245262568601067289

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