Mark Seliternikov

picoCTF - b00tl3gRSA2 [400 points]

Challenge details:

b00tl3gRSA2 | 400 points

Tags: Category: Cryptography

AUTHOR: INVISIBILITY

Description

In RSA d is a lot bigger than e, why don't we use d to encrypt instead of e?

Connect with `nc jupiter.challenges.picoctf.org 19566`.

Hints

1

What is e generally?

When using netcat this is the output:

```
┌──(kali㉿kali)-[~]
└─$ nc jupiter.challenges.picoctf.org 19566
c: 140285621783533311392565572530362871208699937569520735101694690889904627917349414254488425806970893023716710431870753084775070633323235689544017845650193358753013622479280420354843815068864852382139363039585329060550444595819314451025218134947942672624551466817711876027637105146986444364108550045898992353 48
n: 566331460826517011192151463857720869924331492816913865158785411491698651520179131640834846398400446966052791278505025084658834549264061102743886643185462898974615562372660993854929739518635825956301925431390755544107374233713925419512414185752602514510120689407856012686786064692019557359310151901995605254 43
e: 153505837467724311256502107267170507245309132831830231787855166543151728727351909523899327271940820298533069628932713819733578542397833001650096927377612039764020331521130079089070106609842541374768062343335256713988589916091523444007647761520938651279319555079437521086919724071318389654643389231045495265 93
```

When I was researching RSA vulnerabilities I've stumbled opon a vulnerability that you can infer when you encounter a ridiculously large e. Which is called "Wiener's attack". The most common value for a key is '65537' Which is supposed to be the encrypting key! But in this challenge we were told that it is switched… So before trying the "Wiener's attack" I've tried this value:

```
from codecs import decode

c = 92581661842194997673146264837844267526984732189953013264002310661497579463649319964245723551217180757837022291327400207927242395889272732091015637979986747394220803855216329725668184936947656514589448500279502136103117549734622125970921213239411759711203754619233275583433311810902973728996489448020817939 35
n = 10581753558515819172434957925829438183338080938189539860905970544931387937581841729860181932992905951734399320810354156735117922655821115116508588817500348435611983923915643690545618457646967349151451535160147048681087452013724841552818324923285536890565304647821219032945611767344302001402317939520969933500 01
e = 58647944597093260265247871696889189791017152740872889568898113753077315723451674192244287708027728288577229127026640529027829822028379444340897732825436788385078866404707092272859181869121674746663446714950760776973773356624117990883162808836194344984618881144853830559206116825438331712395645328410886227841

text = decode(hex(pow(c, 65537, n))[2:], 'hex')

print(str(text)[2: -1])
```

The result:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3 test.py
picoCTF{bad_1d3a5_2438125}
```

IT ACTUALLY WORKED!

Source of where I've learned of this attack's existence: https://github.com/JohnHammond/ctf-katana