down-bad [cryptography]

Friday, May 31, 2024 8:22 AM

Shiro and Yama added a certain logic gate for good measure to this really stupid algorithm...

Hint: Shiro is pretty small, and Yama has no padding
Flag: CyberBlitz{h4z3ls_lUVf33t_d0nt_Y0U}

Players given 3 files:

1. **down-bad_key.enc**
   462 digit key for RSA decryption (cubic root)
2. **file.enc**
   Ciphertext
3. **publickey.pem**
   public key (RSA)

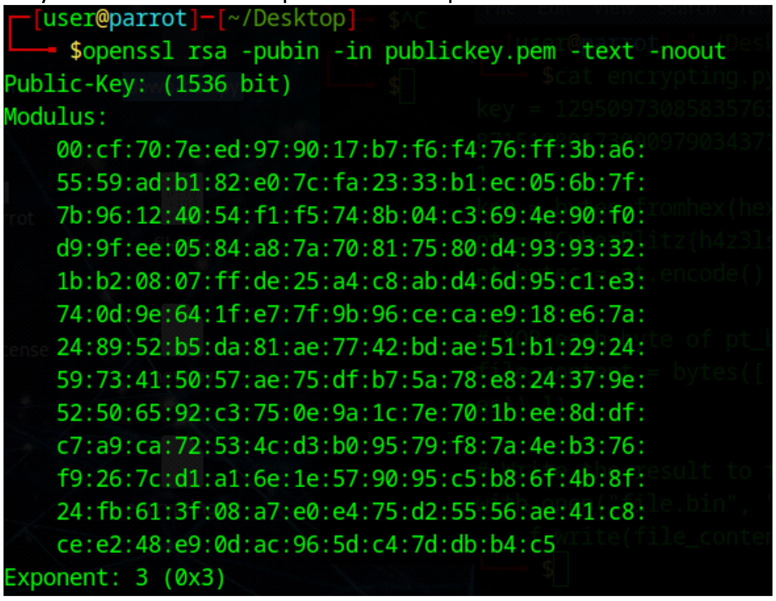
Suggested steps:

- Open publickey.pem and import key as RSA format

```
with codecs.open('publickey.pem') as fr:
   pk = fr.read()
   pk = RSA.importKey(pk)
```

```
print("exponent e: " + str(pk.e) + "\nvalue of n: " + str(pk.n))
```

- Players can check the exponent with OpenSSL as well



```
┌─[user@parrot]─[~/Desktop]
└──╼ $openssl rsa -pubin -in publickey.pem -text -noout
Public-Key: (1536 bit)
Modulus:
    00:cf:70:7e:ed:97:90:17:b7:f6:f4:76:ff:3b:a6:
    55:59:ad:b1:82:e0:7c:fa:23:33:b1:ec:05:6b:7f:
    7b:96:12:40:54:f1:f5:74:8b:04:c3:69:4e:90:f0:
    d9:9f:ee:05:84:a8:7a:70:81:75:80:d4:93:93:32:
    1b:b2:08:07:ff:de:25:a4:c8:ab:d4:6d:95:c1:e3:
    74:0d:9e:64:1f:e7:7f:9b:96:ce:ca:e9:18:e6:7a:
    24:89:52:b5:da:81:ae:77:42:bd:ae:51:b1:29:24:
    59:73:41:50:57:ae:75:df:b7:5a:78:e8:24:37:9e:
    52:50:65:92:c3:75:0e:9a:1c:7e:70:1b:ee:8d:df:
    c7:a9:ca:72:53:4c:d3:b0:95:79:f8:7a:4e:b3:76:
    f9:26:7c:d1:a1:6e:1e:57:90:95:c5:b8:6f:4b:8f:
    24:fb:61:3f:08:a7:e0:e4:75:d2:55:56:ae:41:c8:
    ce:e2:48:e9:0d:ac:96:5d:c4:7d:db:b4:c5
Exponent: 3 (0x3)
```

- Players must identify that file.enc is layered with Base64
  b'\xb4>k\xc8p\xbc\xe9\xb1\xa7\x83\xe8\xbd\xcb-%\x86\xc8\x07E\x8a[tQyp\xd1n{\xb9R\xf9\x9d\x18\xf6\xb0' ***[decoded from Base64]***

- A very important step here is the plain RSA attack. Players MUST know how to find cubic root by adding the key and n-value. This vulnerability stems from its low exponent value of 3.

  *Arguably, I could make it such that players will need to brute-force key + x * n-value; where x is a constant. e.g. key + 2 * n, key + 3 * n and so forth.*

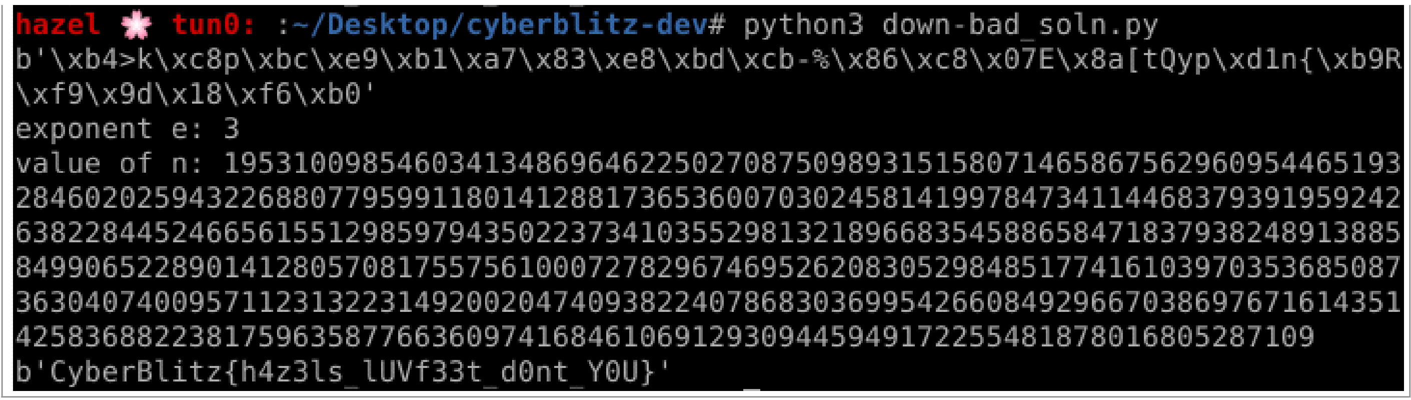
- Hex:

1295097308583576356017570235670474709437182172299949796102306392614257309287
1510801730909790343717206777660797494675328809965345887934044682722741193527
531
Convert to byte:
b'\xf7G\t\xad\x02\xfe\x85\xd8\xd3\xf9\x93\xd5\xffW\x16\xea\xbbX)\xdf\r\x12bJ\x04\x8e
\nK\xd7&\xa6\xc4(\xa3\xcdZ\xc6$\x89\x00\x1173\xef\xfd\xf1\xdcK\x887 \x9c\x92\xa9\xa3
\xe1a\xd0G\x8d\x04\xdb\xd0\xeb'

- Finally, XOR the byte-converted hex-string decoded from the plain RSA attack
  b'CyberBlitz{h4z3ls_lUVf33t_d0nt_Y0U}'

Full Script:

```
# solution for down-bad 05/26/2024
import gmpy2
import codecs
from Crypto.PublicKey import RSA
import base64
from itertools import *

with codecs.open('publickey.pem') as fr:
    pk = fr.read()
    pk = RSA.importKey(pk)

# file.enc
ct = "tD5ryHC86bGng+i9yy0lhsgHRYpbdFF5cNFue7lS+Z0Y9rA="
ct_decoded = base64.b64decode(ct)
print(ct_decoded)

#down-bad_key
with open('down-bad_key.enc', "r") as a:
    for line in a:
        key = line

print("exponent e: " + str(pk.e) + "\nvalue of n: " + str(pk.n))

# store them into mpz for accuracy
a1 = gmpy2.mpz(key)
a2 = gmpy2.mpz(pk.e)
a3 = gmpy2.mpz(pk.n)

# to find cubic root, use iroot
# need to test accordingly
y = a1 + a3
value, boolean = gmpy2.iroot(y, a2)
z = bytes.fromhex(hex(value)[2:])

# now XOR
xor_result = bytes(a ^ b for a, b in zip(ct_decoded, cycle(z)))
print(xor_result)
```



```
hazel 🌸 tun0: :~/Desktop/cyberblitz-dev# python3 down-bad_soln.py
b'\xb4>k\xc8p\xbc\xe9\xb1\xa7\x83\xe8\xbd\xcb-%\x86\xc8\x07E\x8a[tQyp\xd1n{\xb9R
\xf9\x9d\x18\xf6\xb0'
exponent e: 3
value of n: 1953100985460341348696462250270875098931515807146586756296095446519З
2846020259432268807795991180141288173653600703024581419978473411446837939195924 2
6382284452466561551298597943502237341035529813218966835458865847183793824891388 5
8499065228901412805708175575610007278296746952620830529848517741610397035368508 7
3630407400957112313223149200204740938224078683036995426608492966703869767161435 1
425836882238175963587766360974168461069129309445949172255481878016805287109
b'CyberBlitz{h4z3ls_lUVf33t_d0nt_Y0U}'
```