

RACTF2020 Writeup – Mysterious Masquerading Message

We found a file that looks to be like an ssh private key... but it doesn't seem quite right. Maybe you can shed some light on it?

100 Points

For this challenge an SSH private key file called `id_rsa` was provided. Reading the file produced this output:

```
tbbq yhpx:)

-----BEGIN OPENSSH PRIVATE KEY-----
SWYgeW91IGFyZSByZWZkaW5nIHRoaXMsIHRoZW4geW91IHByb2JhYmx5IGZ
pZ3VyZWQgb3V0IHRoYXQgaXQgd2Fzbid0IGFjdHVhbGx5IGFuIFNTSCBrZX
kgYnV0IGEGZGlzZ3Vpc2UuIFNvIHlvdSB0eXZlIG1hZGUgaXQgdGhpcyBmY
XIgYW5kIGZvciB0aGF0IEkgc2F5IHdldGwgZG9uZS4gSXQgd2Fzbid0IHZl
cnkgaGFyZCwgZGhhdCBJJGtub3csIGJldCBuZXZlcnRoZWxlcnMgeW91IGh
dmUgc3RpbGwgbWZkaW5nIHRoaXMsIHRoZW4geW91IHByb2JhYmx5IGZpZ3V0I
GFubm95aW5nIHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpcyYwY5kIHLvdSB3
YW50IHRvIHJlYWQgdGhlIHdod2xlIHRoZW5nIHRvIGNoZWNrIGZvciBjbHV
lcYwgYnV0IHLvdSBjYwY5IGZpbmQgYW55LiBZb3UgYXJlIHNoYXJ0aW5nIH
RvIGdlldCBmcnVzdHJhdGVkIGF0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0ICJ3ZWxsIGRvbmUgeW9
1IGhhdmdUgZ290IHRoXMGZmFyIi4gWW91IHNoYXJ0IHNoYW1taW5nIGRlc2
tzLCBhbmQgc29vbiB0aGUgbW9uaXRvciB3aWxsIGZvbGxvdy4gWW91IGFyZ
SB3b25kZXJpbmcgd2hldmUgdGhpcyBpcyBnb2luZyBhbmQgcmlvdzZlZaW5n
IGl0J3MgY29taW5nIHRvIHRoZSBldmUgdGhlc2YgdGhlIHhhdmdFncmFwaCwgYw5
kIHLvdSBtaWdodCBub3QgaGF2ZSBzZWVuIGFueXRoaW5nLiBjJGhhdmdUgZ2
l2ZW4geW91IHNoYXJlIGZpZ3V0IHRoZSBwZXJzb24gd2hvIG1hZGUgdGhpc
yBhcyB0aGV5IHNoaWxsIGhhdmdVUjJ3QgbWVudGlvbmVkJGFueXRoaW5nIHRv
IGRvIHdpdGggdGhlIGNoYXJsZW5nZSwgZXhjZXB0IC
```

Straight away, it was clear this was not a valid id_rsa file. The first thing I did was investigate the first line:

tbbq yhpx:)

Putting the first 8 letters through Cyber Chef and applying the ROT13 transformation recipe, the output was simply **good luck**. Nice troll there. So next was to look at the actual

Private Key. Putting this into Cyber Chef and applying the From Base64 recipe, the following output was produced:

```
If you are reading this, then you probably figured out that it wasn't actually an SSH key but a disguise. So you have made it this far and for that I say well done. It wasn't very hard, that I know, but nevertheless you have still made it here so congrats. Now you are probably reading this and thinking about annoying the person who made this, and you want to read the whole thing to check for clues, but you cant find any. You are starting to get frustrated at the person who made this as they still haven't mentioned anything to do with the challenge, except "well done you have got this far". You start slamming desks, and soon the monitor will follow. You are wondering where this is going and realising it's coming to the end of the paragraph, and you might not have seen anything. I have given you some things, although you will need something else as well good luck.  
696e656564746f6f70656e6c6f636b73  
696e697469616c69736174696f6e3132
```

Not much, however 2 hex strings at the end. Converting these to ASCII strings gives us:

```
ineedtoopenlocks
```

```
initialisation12
```

Ok, so save these for later. I then went and had a look at the binary at the end of the file. Converting this to hex produced the following string:

Output

```
90988c9befe5ea3f5a91effe03060a8714dfc20088415570b394ce9cd32be718
```

This hex produced no readable ASCII text. After a couple of hours of pondering, I decided to look at the length of the string. 64 characters, which equates to 32 bytes. A quick look on google for where 32 byte strings may be used, and I was pretty much able to find it out straight away: AES!

This can be calculated as follows:




String length – 32 bytes

Each byte = 8 bits



32 bytes * 8 bits = 256 bits, which means the binary is encrypted using AES-256.

Chuckling the string into cyber chef, I used the AES Decrypt recipe. For the key and IV, I used the 2 hex values located in the fake ssh private key. This gave us the flag **ractf{3Asy_F1aG_0n_aEs_rAcTf}**

Recipe



AES Decrypt



Key

696e656564746f6f70656e6c6f636b73

HEX ▾

IV

696e697469616c69736174696f6e3132

HEX ▾

Mode

CBC

Input

Hex

Output

Raw

GCM Tag

HEX ▾

Input

90988c9befe5ea3f5a91effe03060a8714dfc20088415570b394ce9cd32be718

Output

ractf{3Asy_F1aG_0n_aEs_rAcTf}