DEFKTHON CTF: Crypto 200

Description:

> RSA!!

Write-up

The provided `encr02.7z` archive contains two files: `enc` and `rsa`. The former is (from its name) clearly an encrypted message. The filename and contents of the latter hint that the message is encrypted with RSA. The `rsa` file is just a JSON file, with base64-encoded values for properties such as `primeP` and `primeQ` and a more interesting one: `privateExponent`. Unless this is a red herring, this private-key value can be used to decrypt the encrypted message.

Given that the `rsa` file is a "structured" JSON file, lead me to believe that it was exported from some application. Searching for the properties on Google, quickly showed that it came from Keyczar, "an open source cryptographic toolkit designed to make it easier and safer for developers to use cryptography in their applications". Knowing this, the decryption would be straightforward: read the private key into Keyczar and use that to decrypt the message.

Actually, it required an additional step, because of (correct me if I'm incorrect) key encapsulation. Apparently, only the first 261 characters of the encrypted message could be decrypted using the RSA key (I figured out it was 261 by encrypting several strings, and these always resulted in a 261-character-long string). The decryption of the first 261 characters of the encrypted message results in following JSON string:

```
{"hmacKey": {"hmacKeyString": "L_YKADwsSRdPYYioRV-xjn0cWHCKnBIc1j8EmJzRrJY", "size": 256}, "aesKeyString":
```

This JSON data referencing an AES-key can then be used in Keyczar to decrypt the rest of the message, which eventually results in finding the flag: `the flag is hipsteralert`. The following snippet shows how this challenge can be solved with just a handful of lines of Python code (note that it requires the `python-keyczar` package).

```
with open('rsa', 'r') as f:
    rsa = f.read()
with open('enc', 'rb') as f:
    enc = f.read()
import keyczar.keys
key = keyczar.keys.RsaPrivateKey.Read(rsa)
rsaDecrypted = key.Decrypt(enc[:261]) # use RSA to decrypt first 261 ( = len(key.Encrypt('foo')) ) characte
aesKey = keyczar.keys.AesKey.Read(rsaDecrypted)
print aesKey.Decrypt(enc[261:]) # use AES to decrypt the rest of enc
```

Other write-ups and resources

- none yet