Assignment 0x01 - Intro & Cryptography

Start Assignment

Due 27 Mar by 23:55 **Points** 20 **Submitting** a file upload **File types** pdf
Available 28 Feb at 21:00 - 30 Jun at 23:59 4 months

Part I - Intro

You may start this part after completing workshop 0x01.

Instructions:

- Download the question files from here: assignment01.tar.gz (https://myuni.adelaide.edu.au/courses/71400/files/10261716/download?download_frd=1)
- Please ensure you provide the details of what you did (command executed, script written, screenshot of any results, etc), and not just provide the answer!
- Upload your assignment as PDF by the due date

1. (1 point) There is a secret passphrase embedded in the file called "text" in the folder Q01. The secret can be found **following** the line that begins with the word "**And**" and ends with "**it**". Use grep with regular expression to locate this line and the line following it. What is the passphrase?

2. (1 point) There is a file called "here" that contains passphrases. Find the passphrase that occur exactly **14 times**.

3. (1 point) There are lots of files here. What is the **content** of the file whose SHA256 sum is a92536e3c31979736460be6e6729147f974411ef193629999b022b96f5682450?

4. (1 point) Generate a list of passwords from the source file words.txt. Use "l33t" conversion so that a=>4, e=>3, i=>1, and o=>0. For example "hello world" becomes "h3ll0 w0rld". Calculate the SHA256 hash of the generated file and provide that as the answer.

5. (1 point) There is a file encrypted using gpg (Gnu Privacy Protection), using the command gpg -c --batch --passphrase <pass>. Unfortunately we have forgotten the password. Try the "l33t" converted password list from Question 4 to brute-force the encrypted file**.** This may take a few minutes. **Please provide both the correct password and the content of the decrypted file**.

6. (1 point) Find the secret hidden in the file under Q06.

7. (1 point) There are lots of sub-directories and files under Q07. Find the file containing the secret. The file has size of exactly **47** bytes long.

8. (1 point) The secret is in Q08 folder.

9. (1 point) Execute ./a.out and try to guess the secret. **The answer is NOT "xxxx redacted xxxx"**.

10. (1 point) There is an encoded secret in Q10 folder.

Part II - Cryptography

You may start this part after completing workshop 0x02.

11. (2 points) This file (**secret2021.enc** **(https://myuni.adelaide.edu.au/courses/71400/files/10261726/download?download_frd=1)**) has been "encrypted" using this simple crypter code below . "Decrypt" the file and find the secret. Is this a good encryption? What is the key space?

```python
#!/usr/bin/python3

import argparse
import os
import sys
import random

def mycrypto (filename):
    with open(filename, 'r') as f, open(filename + '.enc', 'w') as o:
        blob = f.read()
        for b in blob:
            key = random.randrange(255)
            x = ord(b) ^ key
            o.write(chr(x))

def main():
    parser = argparse.ArgumentParser(description='Encrypt (?) a file')
    parser.add_argument('filename', metavar='filename', type=str, help='file to encrypt')
    parser.add_argument('--seed', metavar='seed',type=int,default=2021,help='seed')
    args = parser.parse_args()

    if not os.path.isfile(args.filename):
        print('The file  does not exist')
        sys.exit()

    random.seed(args.seed)
    mycrypto(args.filename)

if __name__== "__main__":
    main()
```

12. (2 points) A short, plaintext message has been encrypted using Textbook RSA (using the public exponent) with the following parameters:

```
n=0x9B51C20306EDE535C8FCAADBC3F3515E52A0D005703DD449BEC66B23E2932313
p=0xC5A047A7C52ED3A2875F7D76C47B555F
q=0xC93268355C09197BBF1659B5522FFACD
e=0x010001
```

```
d=0x0D067636BAC6088AD2281E4BFFCACFEFEF9BC1A69FB9E701063DFBAAB436E4C1
encrypted_message=0x13121ff7d7be2301a4db5801d6d142e9bb3fbef7f4c73c14f647d5f43ebc8db3
```

Notes

- The numbers above are all in hexadecimal
- Use the sample code from workshop
- You can use the following helper functions in Python3 to byte-wise convert between string and integer.

```python
import binascii

# convert string to integer using
def string_to_int(string):
    return int.from_bytes(binascii.a2b_qp(string),byteorder='big')

# convert into back to string
def int_to_string(number):
    bin = number.to_bytes((number.bit_length() + 7) // 8, byteorder='big')
    return binascii.b2a_qp(bin).decode("utf-8")
```

13. (2 points) Find the message hidden inside this **encrypted bitmap image** ⤓ **(https://myuni.adelaide.edu.au/courses/71400/files/10261398/download?download_frd=1) .** ⤓ **(https://myuni.adelaide.edu.au/courses/71400/files/10261387/download?download_frd=1)** You don't need to decrypt.

14. (2 points) Decipher this text. What is the cipher, and what is the key?

```
OM CAL MIT ZTLM GY MODTL, OM CAL MIT CGKLM GY MODTL, OM CAL MIT AUT GY COLRGD, OM CAL MIT AUT GY YGGSOLIF
TLL, OM CAL MIT THGEI GY ZTSOTY, OM CAL MIT THGEI GY OFEKTRWSOMB, OM CAL MIT LTALGF GY SOUIM, OM CAL MIT
 LTALGF GY RAKQFTLL, OM CAL MIT LHKOFU GY IGHT, OM CAL MIT COFMTK GY RTLHAOK, CT IAR TXTKBMIOFU ZTYGKT W
L, CT IAR FGMIOFU ZTYGKT WL, CT CTKT ASS UGOFU ROKTEM MG ITAXTF, CT CTKT ASS UGOFU ROKTEM MIT GMITK CAB -
OF LIGKM, MIT HTKOGR CAL LG YAK SOQT MIT HKTLTFM HTKOGR, MIAM LGDT GY OML FGOLOTLM AWMIGKOMOTL OFLOLMTR G
F OML ZTOFU KTETOXTR, YGK UGGR GK YGK TXOS, OF MIT LWHTKSAMOXT RTUKTT GY EGDHAKOLGF GFSB.
```

15. (2 points) Download this **shadow file** ⤓ **(https://myuni.adelaide.edu.au/courses/71400/files/10261397/download?download_frd=1)** (/etc/shadow) from an old Linux system. Crack the password for the user account called "yoda" (the last entry). Please list the password and describe how you get it in the answer.

- Use hashcat and the rockyou.txt (found under /usr/share/wordlists). You can use other tools if you like.
- The $1$ prefix of the hash means it's MD5