

فاز اول

بخش یک

در این بخش طبق استانداردهای گفته شده از آیتم های زیر برای سنجش میزان قدرت رمز استفاده میکنیم:

- طول رمز : مقداری بین 8 تا 20 کارکتر باشد.
- استفاده از حرف بزرگ
- استفاده از حرف کوچک
- استفاده از رقم
- استفاده از کارکتر های خاص مثل @
- عدم استفاده از عبارات موجود در لیست سیاه

رعایت هر کدام از شروط 1 امتیاز دارد و روال توصیف رمز به شکل زیر است:

- 1و2 امتیاز : ضعیف
- 3و4 امتیاز : متوسط
- 5و6 امتیاز : قوی

در پیاده سازی از کتابخانه re که برای بررسی عبارات منظم در رشته هاست استفاده میکنیم به خصوص متد سرچ آن که وجود یا عدم یک کارکتر یا کلمه در یک رشته را برمیگرداند . در ورودی رشته رمز را گرفته و با استفاده از تابع هر شش حالت را چک میکند و در صورت رعایت کردن هر کدام شمارنده را یک واحد افزایش میدهد. برای چک کردن هر آیتم هم یک فلگ بولین گذاشته شده تا در آخر نقاط قوت یا ضعف رمز را به کاربر اطلاع دهد. برای بلک لیست از یک دیکشنری رمز های معروف استفاده شده شامل 123، 11111، password و admin و اسم اشخاص یا اسم تیم فوتبال یا غیره است.

خروجی را برای سه حالت ضعیف، متوسط و قوی چک میکنیم:

رمز اول : 123

```
Run: Phase01-Part01 x
C:\Users\MMNazari1380\PycharmProjects\SecurityCourseProject01\venv\Scripts\python.exe
C:/Users/MMNazari1380/PycharmProjects/SecurityCourseProject01/Phase01-Part01.py
Input your password: 123

Password is Weak

Strength of your password:
It contains digit

Weakness of your password:
Length is less than 8
It does not contains lowercase alphabet
It does not contain uppercase alphabet
It does not contain special alphabet
It contains blacklist passwords

Process finished with exit code 0
```

رمز دوم : nazari12@

```
Run: Phase01-Part01 x
C:\Users\MMNazari1380\PycharmProjects\SecurityCourseProject01\venv\Scripts\python.exe
C:/Users/MMNazari1380/PycharmProjects/SecurityCourseProject01/Phase01-Part01.py
Input your password: nazari12@

Password is Medium

Strength of your password:
Length is between 8 and 20
It contains lowercase alphabet
It contains digit
It contains special alphabet

Weakness of your password:
It does not contain uppercase alphabet
It contains blacklist passwords

Process finished with exit code 0
```

رمز سوم: c#3MnA@34raB

```
Run: Phase01-Part01 x
C:\Users\MMNazari1380\PycharmProjects\SecurityCourseProject01\venv\Scripts\python.exe
C:/Users/MMNazari1380/PycharmProjects/SecurityCourseProject01/Phase01-Part01.py
Input your password: c#3MnA@34raB

Password is Strong

Strength of your password:
Length is between 8 and 20
It contains lowercase alphabet
It contains uppercase alphabet
It contains digit
It contains special alphabet
It does not contain blacklist passwords

Weakness of your password:

Process finished with exit code 0
```

بخش دو

در این قسمت ابتدا 3 مودکاری و رمز و لیست کارکترهای مورد استفاده و در نهایت k یعنی تعداد کارکترهایی که از اول رمز برای تابع رمزگشا مشخص است در مود 2 و 3 را از کاربر میگیریم. سپس با استفاده از کتابخانه `itertools` و متد `product` همه ی جایگشت های با تکرار با مجموعه کارکترهای داده شده برای مود اول فقط به اندازه طول رمز و برای مود دوم از 1 تا طول رمز -1 و برای مود آخر از 1 تا طول رمز - k را به ترتیب به دست می آوریم تا به رمز برسیم. در مود دوم و سوم فقط جایگشت های k به بعد را محاسبه میکنیم چونکه k کاراکتر اول رمز را داریم. در نهایت با استفاده از کتابخانه `time` زمان اجرای تابع را اندازه میگیریم. برای تست فقط از رمزهای 4 رقمی استفاده میکنیم به علت محدودیت زمانی:

رمز اول: 1234 با مود اول و فضای حالت اعداد

```
Run: Phase01-Part02 x
C:\Users\MMNazari1380\PycharmProjects\SecurityCourseProject01\
C:/Users/MMNazari1380/PycharmProjects/SecurityCourseProject01
Working modes:
1: Standard mode providing only length of password
2: Search mode providing only first character of password
3: Search mode providing only k character of password
Insert your mode: 1

Choose k: 0 for mode 1, 1 for mode 2, and k for mode 3
Insert k: 0

Character sets:
1: only digit
2: only digit and lowercase alphabet
3: only lowercase alphabet
4: any type of character
Insert your character set: 1

Insert your password: 1234

Cracked password is: 1234
Number of attempts: 1235
Time taken: 0.0

Process finished with exit code 0
```

رمز دوم: 12ab با مود اول و دوم و مجموعه حالت اعداد و حروف کوچک

```
Run: Phase01-Part02 x
C:\Users\MMNazari1380\PycharmProjects\SecurityCourseProject01\
C:/Users/MMNazari1380/PycharmProjects/SecurityCourseProject01/
Working modes:
1: Standard mode providing only length of passwprd
2: Search mode providing only first character of password
3: Search mode providing only k character of password
Insert your mode: 1

Choose k: 0 for mode 1, 1 for mode 2, and k for mode 3
Insert k: 0

Chatcter sets:
1: only digit
2: only digit and lowercase alphabet
3: only lowercase alphabet
4: any type of character
Insert your character set: 2

Insert your password: 12ab

Cracked password is: 12ab
Number of attempts: 49620
Time taken: 0.06250977516174316

Process finished with exit code 0
```

```
Run: Phase01-Part02 x
C:\Users\MMNazari1380\PycharmProjects\SecurityCourseProject01\
C:/Users/MMNazari1380/PycharmProjects/SecurityCourseProject01/
Working modes:
1: Standard mode providing only length of passwprd
2: Search mode providing only first character of password
3: Search mode providing only k character of password
Insert your mode: 2

Choose k: 0 for mode 1, 1 for mode 2, and k for mode 3
Insert k: 1

Chatcter sets:
1: only digit
2: only digit and lowercase alphabet
3: only lowercase alphabet
4: any type of character
Insert your character set: 2

Insert your password: 12ab

Cracked password is: 12ab
Number of attempts: 4296
Time taken: 0.015547752380371094

Process finished with exit code 0
```

رمز سوم : efghijkl با مود سوم و $k=4$ و مجموعه کاراکتر حروف کوچک

```
Run: Phase01-Part02 x
C:\Users\MMNazari1380\PycharmProjects\SecurityCourseProject01\
C:/Users/MMNazari1380/PycharmProjects/SecurityCourseProject01/
Working modes:
1: Standard mode providing only length of passwprd
2: Search mode providing only first character of password
3: Search mode providing only k character of password
Insert your mode: 3

Choose k: 0 for mode 1, 1 for mode 2, and k for mode 3
Insert k: 4

Chatcter sets:
1: only digit
2: only digit and lowercase alphabet
3: only lowercase alphabet
4: any type of character
Insert your character set: 3

Insert your password: efghijkl

Cracked password is: efghijkl
Number of attemps: 165242
Time taken: 0.4843602180480957

Process finished with exit code 0
```

رمز چهارم: mN@5 با مود اول و مجموعه کاراکتر همه حالت ها

```
Run: Phase01-Part02 x
C:\Users\MMNazari1380\PycharmProjects\SecurityCourseProject01\
C:/Users/MMNazari1380/PycharmProjects/SecurityCourseProject01
Working modes:
1: Standard mode providing only length of passwprd
2: Search mode providing only first character of password
3: Search mode providing only k character of password
Insert your mode: 1

Choose k: 0 for mode 1, 1 for mode 2, and k for mode 3
Insert k: 0

Chatcter sets:
1: only digit
2: only digit and lowercase alphabet
3: only lowercase alphabet
4: any type of character
Insert your character set: 4

Insert your password: mN@5

Cracked password is: mN@5
Number of attemps: 22498306
Time taken: 21.281206607818604

Process finished with exit code 0
```

فاز دوم

در این بخش برای رمزنگاری از کتابخانه cryptography و fernet استفاده میکنیم که با استفاده از یک کلید به روش متقارن AES میتوان فایل را رمزنگاری کرد. حال برای اینکه کلید را با استفاده از رمز شخصی بسازیم از کتابخانه Scrypt استفاده میکنیم. پارامترهای این الگوریتم شامل:

The salt: key derivation functions need random bits added to the password before it's hashed; these bits are called [the salt](#), which helps strengthen security and protect against dictionary and brute-force attacks

Length: The desired length of the key (32 in this case).

n: CPU/Memory cost parameter, must be larger than 1 and be a power of 2.

r: Block size parameter.

p: Parallelization parameter.

میباشد که مقادیر پیش فرض و استاندارد برای آن ها مقادیر زیر در نظر گرفته شده است:

سالت : به طول 16

طول کلید: 32

2^{14} : n

8 : r

1 : p

پس از تولید کلید توسط Scrypt، کلید و رمز به عنوان پارامتر به fernet داده میشود و با استفاده از آن تابع های رمزنگاری و رمزگشایی را میسازیم. متن اولیه فایل تکستی که میخواستیم رمز نگاری کنیم " This a test for file encryption and decryption " است.

تست رمزنگاری:

فایل قبل رمزنگاری

```
Phase01-Part01.py × Phase01-Part02.py × Phase02.py × data ×  
1 This a test for file encryption and decryption
```

```
Run: Phase02 ×  
C:\Users\MMNazari1380\PycharmProjects\Securit  
C:/Users/MMNazari1380/PycharmProjects/Securi  
Please Enter name of the file: data  
Enter password: mmnazari  
Enter 1 for encrypt and 2 for decrypt: 1  
File encrypted successfully  
  
Process finished with exit code 0
```

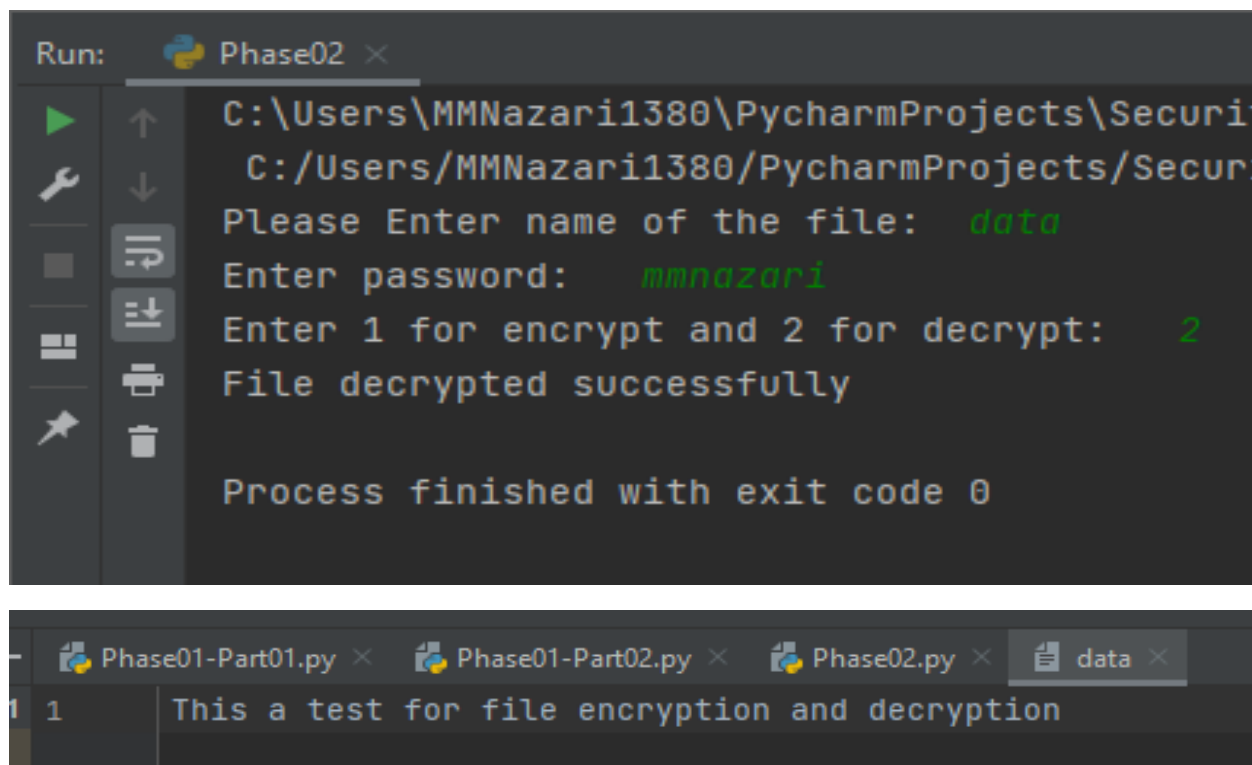
فایل را با رمز mmnazari رمزنگاری میکنیم و مشاهده میکنیم که مقادیر درون فایل data رمزگذاری شده و یک فایل جدید salt.salt برای ما ساخته شده.

```
Phase01-Part01.py × Phase01-Part02.py × Phase02.py × salt.salt × data ×  
1 gAAAAABLT3nqptQnmE1g3syTmwnmruK1j70zWQTN0V6_w8F5wxw9PDwxvwt9Gk1-2o-XXju49G6EkVutRZfpncLI0sLojVJLA9mNYwE3W3knJ7Xiah6TWVhnHt
```

```
Phase01-Part01.py × Phase01-Part02.py × Phase02.py × salt.salt ×  
1 EOT(ÜFFFF@Ü@w'EMSUBivTÑ
```

برای اینکه فایلی که رمزنگاری شده را بتوانیم رمزگشایی کنیم باید از همان رمز و سالت اولیه استفاده کنیم. به منظور رفع این مشکل از فلگ بولین saved_salt استفاده میکنیم.

تست رمزگشایی:



The image shows two screenshots from a PyCharm IDE. The top screenshot is a 'Run' console window titled 'Phase02'. It displays the execution of a Python script. The script prompts for a file name, password, and an action (1 for encrypt, 2 for decrypt). The user input is shown in green. The output shows the file was decrypted successfully. The bottom screenshot is an editor window showing a file named 'data'. The file contains the text 'This a test for file encryption and decryption'.

```
Run: Phase02 x
C:\Users\MMNazari1380\PycharmProjects\Securit
C:/Users/MMNazari1380/PycharmProjects/Secur
Please Enter name of the file: data
Enter password: mmnazari
Enter 1 for encrypt and 2 for decrypt: 2
File decrypted successfully

Process finished with exit code 0
```

Phase01-Part01.py x Phase01-Part02.py x Phase02.py x data x

```
1 1 This a test for file encryption and decryption
```

میدانیم در رمزنگاری متقارن باید کلید در رمزنگاری و رمزگشایی یکسان باشد. حال در کد بالا اگر یک فایل را رمزنگاری کردیم، باید سالت را تغییر ندهیم و از همان سالت استفاده کنیم تا بتوانیم فایل را رمزگشایی کنیم. برای اینکار مقدار saved_salt را در رمزنگاری False و در رمزگشایی True میگذاریم.