

초심자를 위한

Python

4주차

Index

1.준비물

2.압축파일 패스워드 크래커 제작

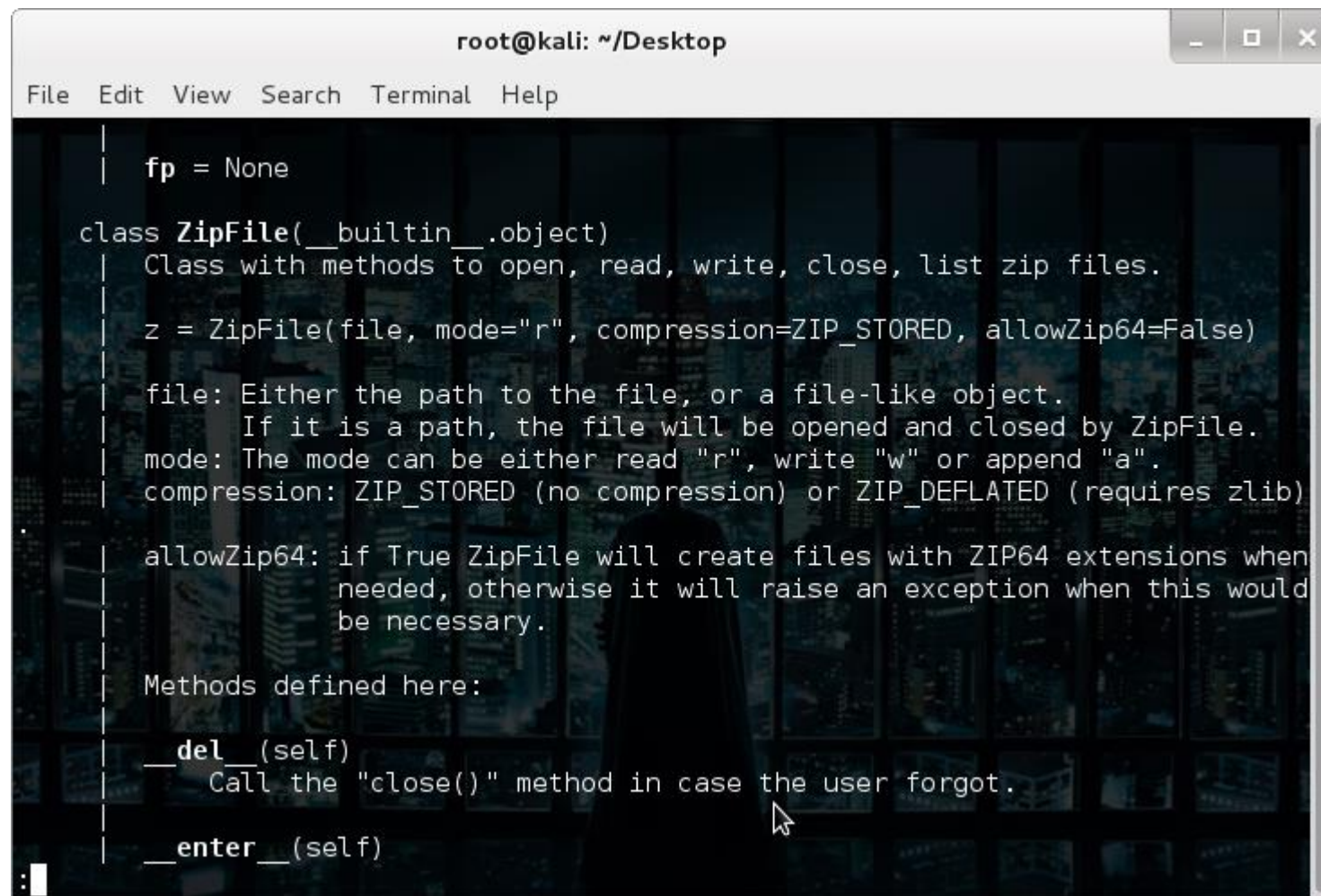
준비물

python 2.7.x 버전이 설치되어있는 Linux

*권장 사항 : Kali Linux 1.1.0 v486 (python 2.7.3)

압축파일 패스워드 크래커 제작

ZipFile 라이브러리 확인해보기



```
root@kali: ~/Desktop
File Edit View Search Terminal Help

fp = None

class ZipFile(__builtin__.object)
  Class with methods to open, read, write, close, list zip files.

  z = ZipFile(file, mode="r", compression=ZIP_STORED, allowZip64=False)

  file: Either the path to the file, or a file-like object.
        If it is a path, the file will be opened and closed by ZipFile.
  mode: The mode can be either read "r", write "w" or append "a".
  compression: ZIP_STORED (no compression) or ZIP_DEFLATED (requires zlib)

  allowZip64: if True ZipFile will create files with ZIP64 extensions when
              needed, otherwise it will raise an exception when this would
              be necessary.

  Methods defined here:

  __del__(self)
    Call the "close()" method in case the user forgot.

  __enter__(self)
```

리눅스 터미널에서..

- 1) 'python' 명령어 입력 후 idle로 들어가기
- 2) help('zipfile') 입력
- 3) pageDown키 등을 통해서 화면을 내리다보면 'ZipFile' 라이브러리 확인 가능

ZipFile 라이브러리 확인해보기

ZipFile 클래스의 'extractall()' 메소드

압축을 해제하기 위해 사용되는 메소드.
패스워드를 인자값으로 입력받을 수 있다.

ex) `extractall(pwd='password')`

```
extractall(self, path=None, members=None, pwd=None)  
    Extract all members from the archive to the current working  
    directory. 'path' specifies a different directory to extract to.  
    'members' is optional and must be a subset of the list returned  
    by namelist().
```

기본 설정

리눅스 바탕화면에 실습을 위해 필요한 파일(파이썬 코드 등)을
위치하도록 할 것이기 때문에 터미널에서 `cd Desktop` 명령으로
경로를 바탕화면에 있도록 해야합니다.



기본 설정

리눅스 바탕화면에 files.zip 파일을 옮겨주세요

***VM-Ware : 드래그 혹은 ctrl+c / ctrl+v**

그래도 안된다면 리눅스 상에서 직접 다운로드 받을 것

첫번째 코드 작성

```
root@kali:~/Desktop# vi unzip.py
```

vi 사용법 ;

i : 입력모드

esc : 명령모드

esc + :wq 입력 : 저장 후 종료

unzip.py

```
import zipfile  
zFile=zipfile.ZipFile("files.zip")  
zFile.extractall(pwd="secret")
```

첫번째 코드 실행

```
root@kali:~/Desktop# python unzip.py
```

바탕화면에 압축이 풀린 파일들을 확인할 수 있습니다.

확인이 되셨으면, 다음 코드를 작성하기 위해 압축인 풀린 파일들을 지워주세요.

두번째 코드 작성

vi unzip_2.py

unzip_2.py

```
import zipfile
zFile=zipfile.ZipFile("files.zip")

try:
    zFile.extractall(pwd="oranges")
except Exception, e:
    print e
```

두번째 코드 실행

```
root@kali:~/Desktop# python unzip_2.py  
( 'Bad password for file', <zipfile.ZipInfo object at 0xb75442cc>)
```

패스워드가 틀릴경우 에러 메시지를 출력하도록 함
(예외처리 이용..)

세번째 코드 작성

vi unzip_3.py

unzip_3.py

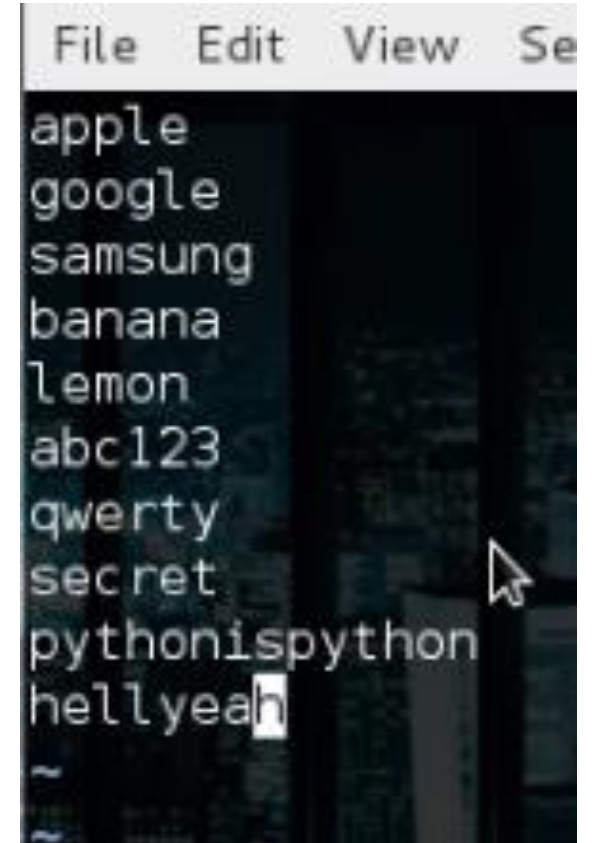
```
import zipfile
zFile=zipfile.ZipFile('files.zip')
passFile=open('dictionary.txt')
for line in passFile.readlines():
    password=line.strip('\n')
    try:
        zFile.extractall(pwd=password)
        print '[+]Password : '+password+'\n'
        exit(0)
    except Exception, e:
        pass
```

세번째 코드 작성 + 사전 만들기

vi dictionary.txt

dictionary.txt

```
apple  
google  
abc123  
qwerty  
secret  
pythonispython
```

A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'View', and 'Se'. The terminal displays the contents of a file named 'dictionary.txt'. The text is as follows:

```
apple  
google  
samsung  
banana  
lemon  
abc123  
qwerty  
secret  
pythonispython  
hellyeah  
~  
~
```

A mouse cursor is visible on the right side of the terminal, pointing towards the 'secret' line.

세번째 코드 실행

```
root@kali:~/Desktop# python unzip_3.py  
[+]Password = secret
```

사전파일을 이용하여 압축을 풀기 위해 사전 파일에 등록되어있는
비밀번호를 대입하도록 함.

찾은 비밀번호가 무엇인지 출력하도록 함.

네번째 코드 작성

vi unzip_4.py

unzip_4.py

①

```
import zipfile
```

```
def extractFile(zFile, password):
```

```
    try:
```

```
        zFile.extractall(pwd=password)
```

```
        return password
```

```
    except:
```

```
        return
```

③

```
if __name__ == '__main__':
```

```
    main( )
```

②

```
def main( ):
```

```
    zFile=zipfile.ZipFile('files.zip')
```

```
    passFile=open('dictionary.txt')
```

```
    for line in passFile.readlines( ):
```

```
        password=line.strip('\n')
```

```
        guess=extractFile(zFile, password)
```

```
        if guess:
```

```
            print '[+]Password : '+password+'\n'
```

```
            exit(0)
```

* 작성 순서 : ①->②->③

네번째 코드 실행

```
root@kali:~/Desktop# python unzip_4.py  
[+] Password : secret
```

프로그램을 함수(메소드) 단위로 새롭게 작성.

마지막 코드 작성

`vi unzip_final.py`

실습 코드 `unzip_final.py` 참고

(코드가 너무 길어요)

마지막 코드 실행

```
root@kali:~/Desktop# python unzip_final.py  
usage%prog -f <zipfile> -d <dictionary>  
root@kali:~/Desktop# python unzip_final.py -f files.zip -d dictionary.txt  
[+] Password : secret
```

Thread의 사용으로 조금 더 빠른 패스워드 크래킹 활성화

옵션을 주는 방식으로 어떠한 압축파일, 사전 파일이라도 사용 가능하도록 함.

Thank you.