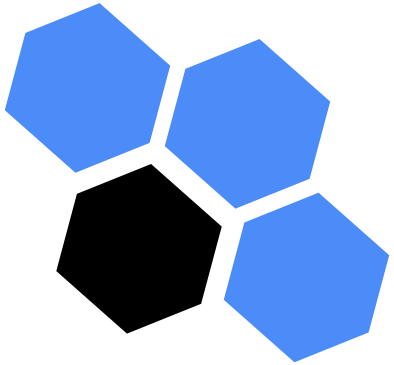
ZIP File

Bruteforce Attack



2021. 07. 27

SCP 이예준

이 프로젝트를 하게 된 이유

JBU CTF

ZIP 파일 비밀번호 뚫는 문제..
(캡처본 없음.. ㅎㅋ)

python™

목차

- ZIP File
- Brute force
- Zip File Brute force
  1) dictionary.py + unlock.py
  2) static_StartEnd_unlock_zip.py
  3) oneclick_unlock_zip.py

- 환경 : winodws 10
- 언어 : Python
- 기간 : 약 1-2주,,

ZIP File

ZIP (파일 포맷)

데이터 압축, 보관을 위한 파일형식
여러 개의 파일을 크기를 줄이고 하나로 묶어 저장

Brute force

brute force 미국식 영국식 웹수집

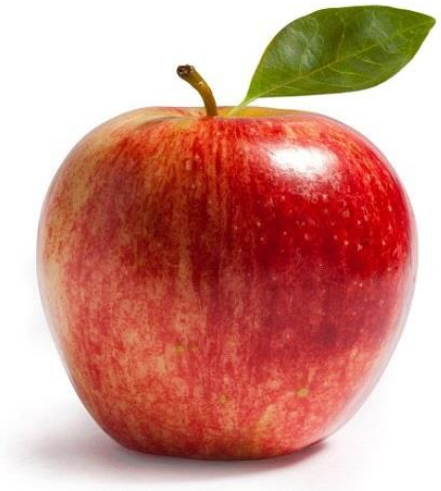
명사 (컴퓨터) 억지 기법 ((무차별 대입해 억지로 문제를 푸는))

웹수집

⇒ **조합 가능한 모든 경우의 수를 다 대입해보는 것.**

⇒ **효율과는... 거리가 멀다.**

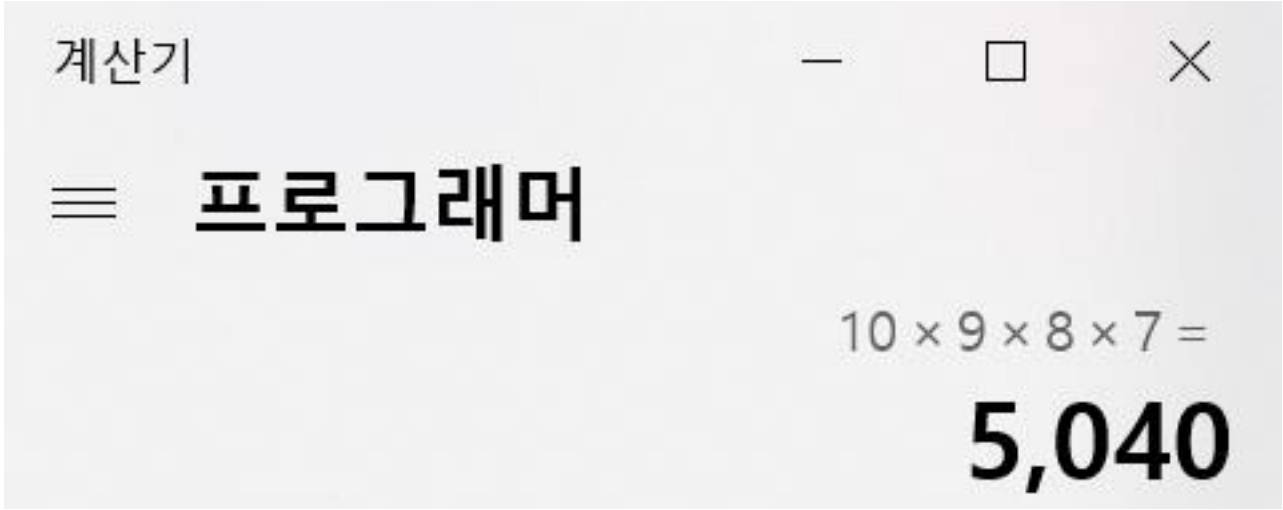
Brute force



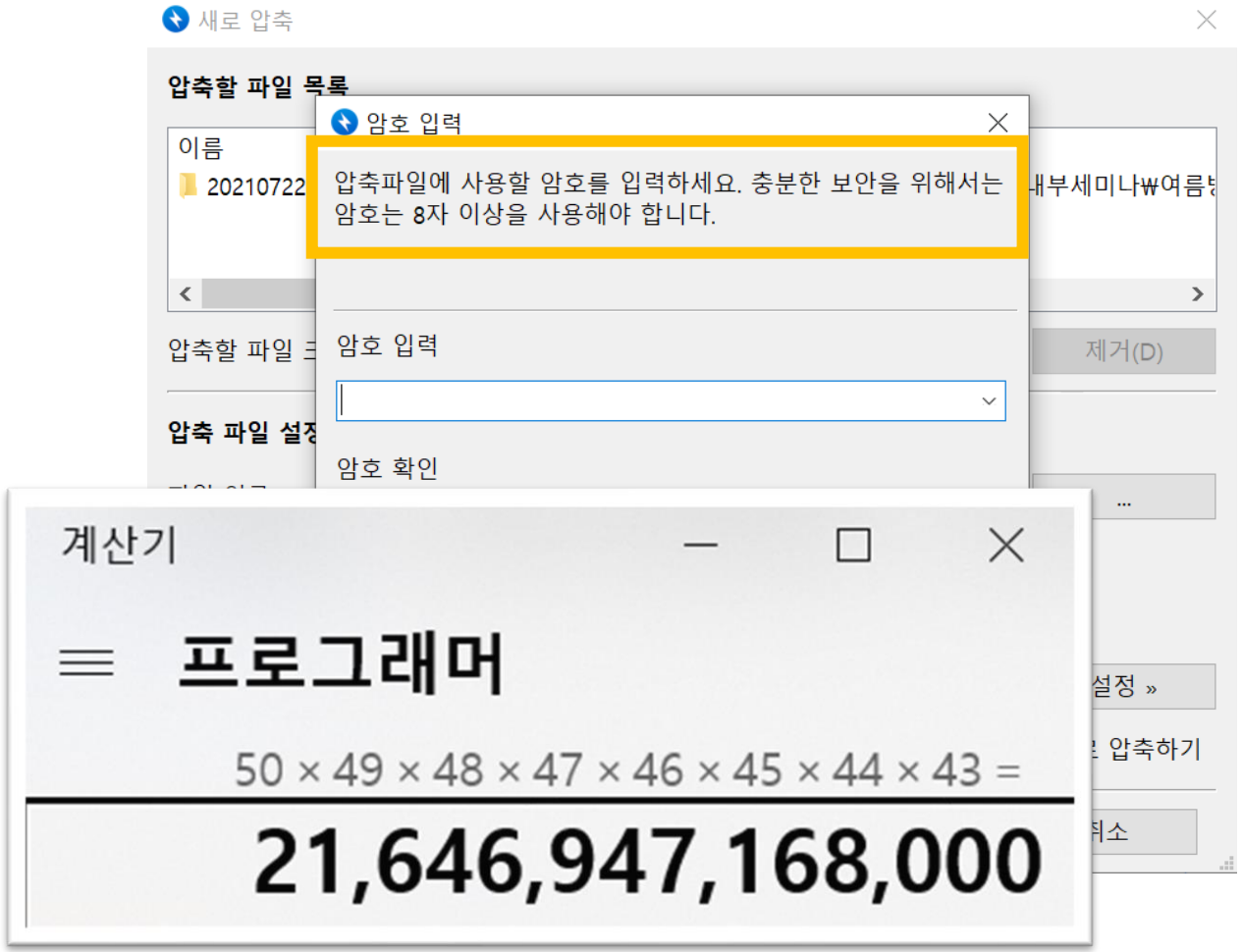
Brute force

0001, 0002, 0003 · · · 9997, 9998, 9999

⇒ 중복빼고.. 10*9*8*7 ,,,



ZIP File Brute force Attack



ZIP File Brute force Attack

파이썬 표준 라이브러리 zipfile 모듈

=> ZIP파일 압축 또는 압축해제 가능 !!

메소드명	설명
ZipFile객체명 = zipfile.ZipFile(file_or_pathname, mode, compression여부)	처리 대상 file 또는 pathname을 ZipFile "ZipFile객체명"으로 정의함. - mode : "r"(읽기) 또는 "w"(쓰기) - compression : ZIP_STORED(압축안함; default), ZIP_DEFLATED(압축)
ZipFile객체명.write(filename)	filename을 ZipFile로 정의한 객체명에 zip file 작성
ZipFile객체명.extractall(path=None)	- path=None 일 경우에는 현재 디렉터리(cwd)로 ZipFile객체명에 있는 모든 파일 및 디렉터리들을 unzip - path명이 기술되어 있는 경우에는 해당 path명으로 ZipFile객체명에 있는 모든 파일 및 디렉터리들을 unzip

사진 출처 : https://blog.naver.com/youndok/222052231633

ZIP File Brute force Attack

Ver 1

dictionary.py
unlock_zip.py

Ver 2

static_StartEnd_unlock_zip.py

Ver 3

oneclick_unlock_zip.py

ZIP File Brute force Attack

<dictionary.py>

```python
#-*- coding: utf-8 -*-
from itertools import product

chars = input('[*] Input : brute force dictionary chars >> ')   # 조합할 문자들 입력
f = open('brute_force.txt','w')
for length in range(5,6):   # 만들고 싶은 사전의 자릿수. 5자리 -> range(5,6)
    to_attempt = product(chars, repeat=length)
    for attempt in to_attempt:
        brute = ''.join(attempt)
        f.write(brute+'\n')
print('[*] Complete : brute force dictionary! ')
f.close()
```

사전 파일 생성

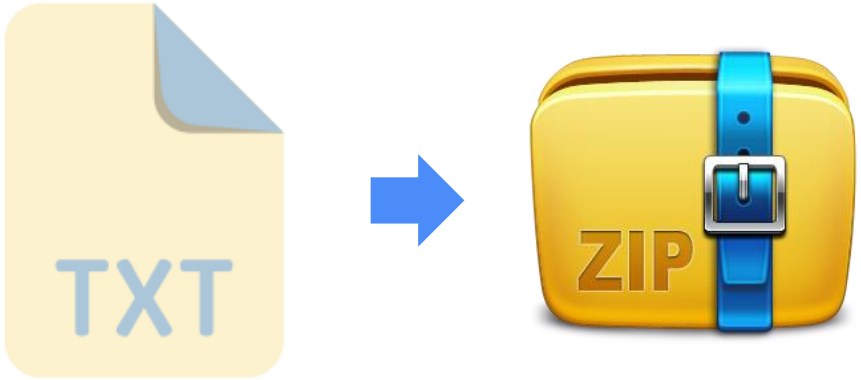
Ver 1 : dictionary.py

<test.py>

```python
from itertools import product
a = product('123',repeat=3)
for i in a:
    print(''.join(i))
```

```
C:₩Users₩yejun₩Desktop>python test.py
111
112
113
121
122
123
131
132
133
211
212
213
221
```

ZIP File Brute force Attack



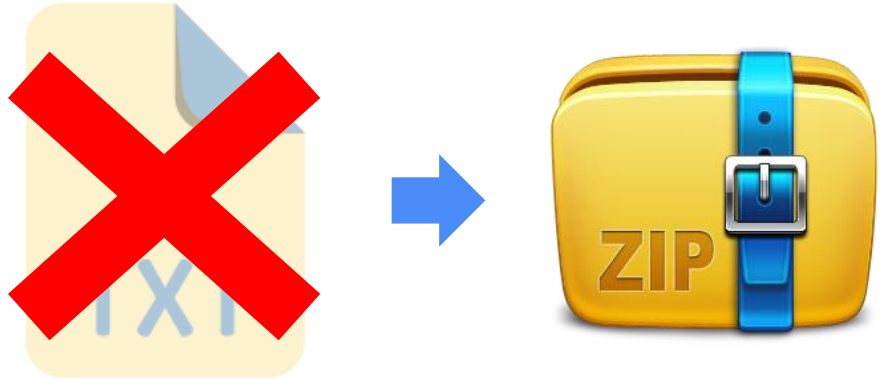

```
C:₩Users₩yejun₩unlock_zip>python "1-2)unlock_zip.py"
zip file path >> ./test.zip
dictionary file path >> ./brute_force.txt
[*] start : brute force attack!
[*] find password! : 25885
```

<unlock_zip.py>

```python
from zipfile import *

def unzipfile(filename, password):
    try:
        filename.extractall(pwd = password.encode())
        print("[*] find password! : {0}".format(password))
        exit()
    except:
        pass

def main():
    i = input('zip file path >> ')  # 비밀번호가 잠긴 zip파일 경로지정
    if i.find('.zip') == -1:    # zip확장자가 아니면 종료
        print('It is no zip file.')
        exit()
    filename = ZipFile(i)  # 기본 모드 "r"
    li = input('dictionary file path >> ')  # 사전파일 경로지정
    pass_list = open(str(li),'r')   # 파일 모두 읽기
    print('[*] start : brute force attack!')
    for line in pass_list:  # 라인 읽기
        password = line.strip('\n')
        #print(password)
        unzipfile(filename, password)

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

Ver 1 : unlock_zip.py

ZIP File Brute force Attack

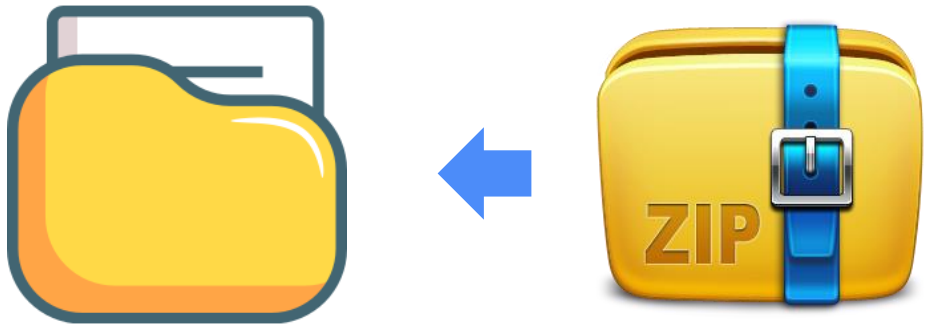



```
C:₩Users₩yejun₩unlock_zip>python "2)static_StartEnd_unlock_zip.py"
zip file path >> test.zip
static_start_char >> 25
static_end_char >>
dictionary chars >> 1234567890
password min_length >> 5
password max_length >> 5
[*] start : brute force attack!
[*] find password! : 25885
```

```python
#-*- coding: utf-8 -*-
from itertools import product
from zipfile import *

def unzipfile(filename, password):
    try:
        #print(password)
        filename.extractall(pwd = password.encode())
        print("[*] find password! : {0}".format(password))
        return 1
    except:
        pass


def main():

    i = input('zip file path >> ')
    if i.find('.zip') == -1:
        print('It is no zip file.')
        exit()
    filename = ZipFile(i)

    static_start_char = input('static_start_char >> ')  # 앞에서 부터 확실한 비밀번호 입력
    static_end_char = input('static_end_char >> ')  # 뒤에서 부터 확실한 비밀번호 입력
    static_char_len = len(static_start_char) + len(static_end_char) # 고정된 비밀번호 개수
    chars = input('dictionary chars >> ')
    min_length = int(input('password min_length >> ')) - static_char_len    # 고정된 값을 제외
    max_length = int(input('password max_length >> ')) - static_char_len

    print('[*] start : brute force attack!')
    for length in range(min_length,max_length+1):
        to_attempt = product(chars, repeat=length)
        for attempt in to_attempt:
            password = ''.join(attempt)
            ret = unzipfile(filename, static_start_char + password + static_end_char)
            if ret == 1:
                exit()

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

Ver 2 : static_StartEnd_unlock_zip.py

ZIP File Brute force Attack

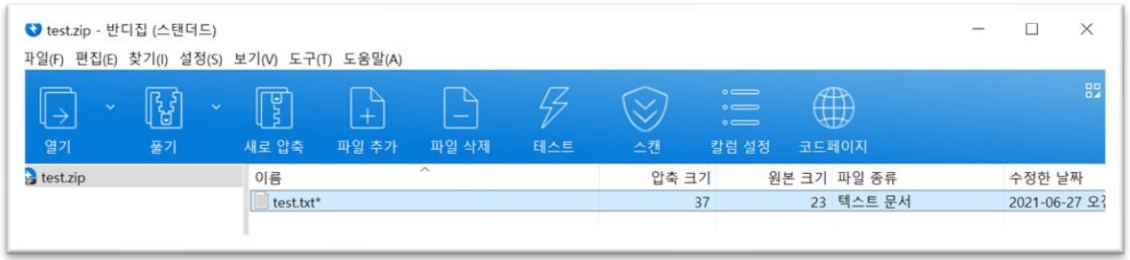


```
C:₩Users₩yejun₩unlock_zip>python "3)oneclick_unlock_zip.py"
[*] Input : zip file path >> test.zip
[*] Input : Guess_password (If you don't know : [spacebar] ) >>   8 5
[*] Input : bruteforce chars >> 1234567890
[*] Start : bruteforce Attack!
[*] Find password! : 25885
```

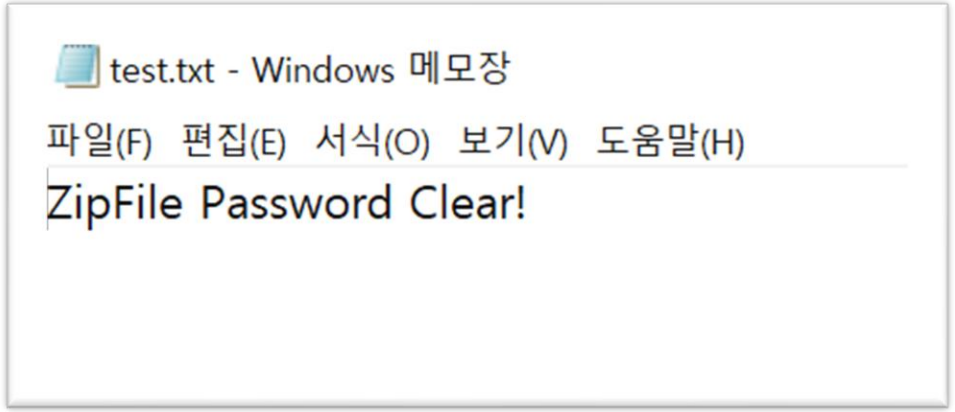
Ver 3 : oneclick_unlock_zip.py

```python
def main():
    i = input('[*] Input : zip file path >> ')
    if i.find('.zip') == -1:
        print('It is no zip file.')
        exit()
    filename = ZipFile(i)

    static_char = str(input("[*] Input : Guess_password (If you don't know : [spacebar] ) >> "))
    static_char_len = len(static_char)

    static_true = []    #공백이 아닌 자리의 인덱스를 저장
    static_false = []   #공백의 자리의 인덱스를 저장 -> 이 부분만 대입하기 위해
    index = 0
    while(index < static_char_len):
        if static_char[index] == ' ':
            static_false.append(index)
        else:
            static_true.append(index)
        index = index+1

    static_true_len = len(static_true)
    static_false_len = len(static_false)

    chars = input('[*] Input : bruteforce chars >> ')

    print('[*] Start : bruteforce Attack!')

    for length in range(static_false_len,static_false_len+1):   # 만들어지는 비밀번호 자리수 == 공백의 수
        to_attempt = product(chars, repeat=length)
        for attempt in to_attempt:
            password_list = []
            for char in attempt:
                password_list += char

            for s_word in static_true:  # 공백이 아니었던 자리를 insert를 이용해서 채워준다.
                password_list.insert(s_word,static_char[s_word])

            password = ''.join(password_list)
            ret = unzipfile(filename, password) # 비밀번호 해제 시도
            if ret == 1:
                exit()

    print('[*] Exit : bruteforce Attack is failure.')
```

ZIP File Brute force Attack



test.zip : 5자리 숫자로 이루어진 zip파일



Q & A