

2. ZIP暴力破解

```
import zipfile
import threading
import optparse
```

- `zipfile`：用于处理 ZIP 文件的模块。
- `threading`：用于创建和管理线程的模块，以便可以同时尝试多个密码。
- `optparse`：用于解析命令行参数的模块。

```
def extractFile (zipFile,password):
    try:
        zipFile.extractall(pwd=str(password).encode())
        print("password:", password)
    except:
        pass
```

定义一个函数并且设置两个参数zipFile和password

并尝试用password参数解压zip，如果成功打印密码，如果失败啥也不做，

(password后面有encode的原因是密码传递给 extractall 方法时，使用 .encode() 方法将其转换为字节串，因为 extractall 需要字节串作为密码。)

```
def main():
    parser = optparse.OptionParser("usage%prog -f <ZIP目录> -d <字典目录>")
```

使用说明

```
        parser.add_option("-f", dest="zipFile", type="string",
help="指定zip文件")
        parser.add_option("-d", dest="dictionary", type="string",
help="指定字典文件")
        (options, args) = parser.parse_args() #解析命令行参数
```

命令行参数-f 指定zip文件 -d 指定字典文件,并解析命令行参数将结果存储在options和args中

python复制代码

```

        if options.zipFile is None or options.dictionary is None:
            print(parser.usage) #如果命令行参数为空，则打印帮助信息
            exit(0)
    ...

```

如果缺少参数则打印使用方法并退出

```

```python
 else :
 zname = options.zipFile
 dname = options.dictionary

```

如果没有缺少则将参数存储在变量当中

```

 zFile = zipfile.ZipFile(zname)
 passFile = open(dname)
 for line in passFile.readlines():
 password = line.strip('\n') #strip()函数用于移除字符串头尾指
 定的字符（默认为空格或换行符）或字符序列
 t = threading.Thread(target=extractFile,args=
 (zFile,password))
 t.start()

```

使用 `zipfile.ZipFile` 打开指定的 ZIP 文件，并将其存储在 `zFile` 变量中，打开指定的字典文件（`passFile` 变量），循环读取字典每一行的内容，并去掉行尾的换行符，并为每一个密码创建一个线程，将 `zFile` 和 `password` 作为参数传递。

```

if __name__ == '__main__':
 main()

```

初始化运行

## 最终代码(注释由CodeGeeX编写)

```

import zipfile
import threading
import optparse

def extractFile (zipFile,password):
 try:
 zipFile.extractall(pwd=str(password).encode())
 print("password:", password)

```

```

except:
 print('error')
def main():
 global zname, dname
 parser = optparse.OptionParser("usage%prog -f <ZIP目录> -d <字典目录>")
 parser.add_option("-f", dest="zipFile", type="string", help="指定zip文件")
 parser.add_option("-d", dest="dictionary", type="string", help="指定字典文件")
 (options, args) = parser.parse_args() #解析命令行参数
 if options.zipFile is None or options.dictionary is None:
 print(parser.usage) #如果命令行参数为空，则打印帮助信息
 exit(0)
 else:
 zname = options.zipFile
 dname = options.dictionary
 threads = [] # 创建一个线程列表
 zFile = zipfile.ZipFile(zname, 'r')
 passFile = open(dname)
 for line in passFile.readlines():
 password = line.strip('\n') #strip()函数用于移除字符串头尾指定的字符（默认为空格或换行符）或字符序列
 t = threading.Thread(target=extractFile,args=(zFile,password))
 threads.append(t) #将线程添加到线程列表中
 t.start()
 for t in threads:
 t.join() #等待所有线程执行完毕
if __name__ == '__main__':
 main()

```

## 尝试破解

7500多个密码破解用来9秒多速度还可以

```

D:\pythoncode>unzip.py -f zip2.zip -d 常用密码.txt
password: 123

```