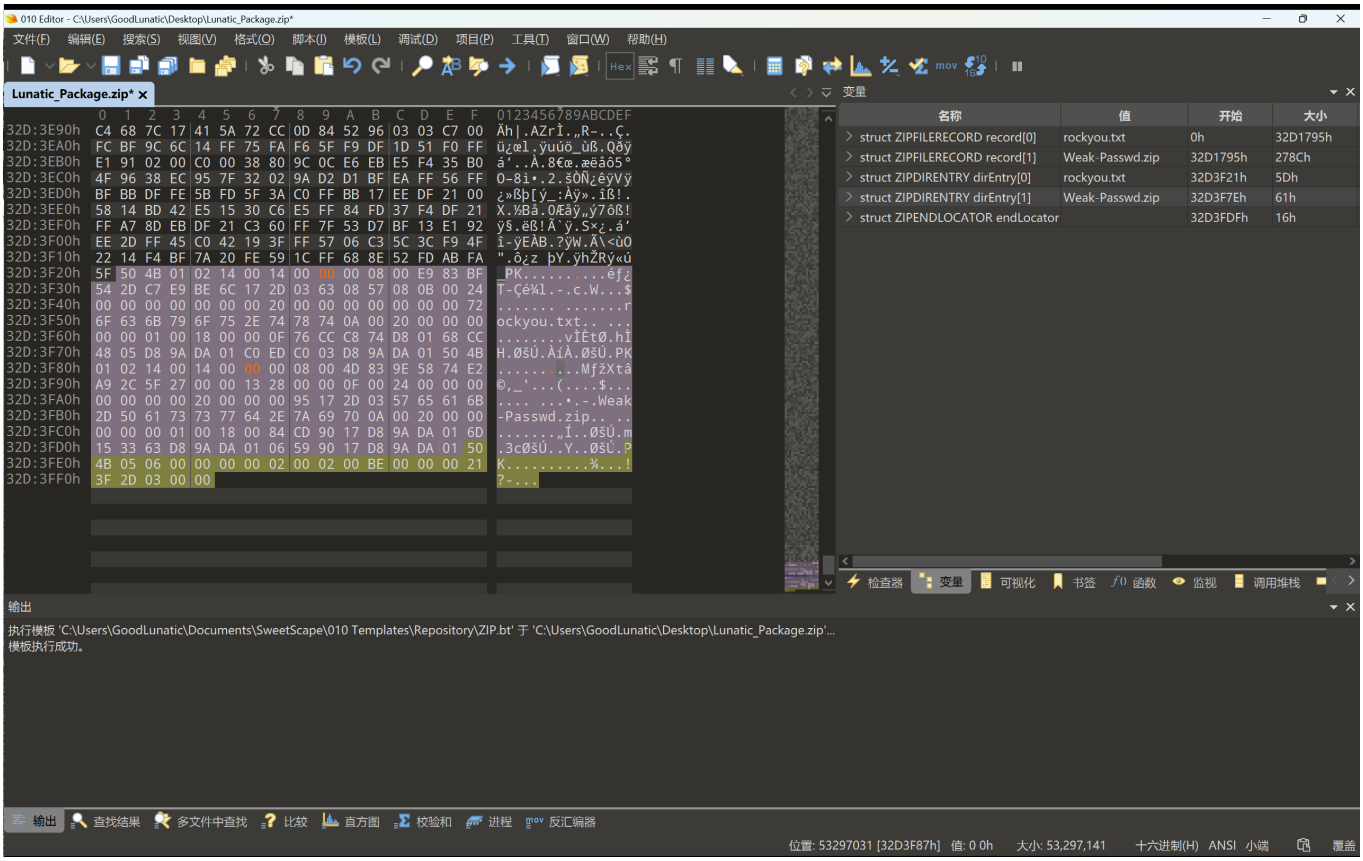
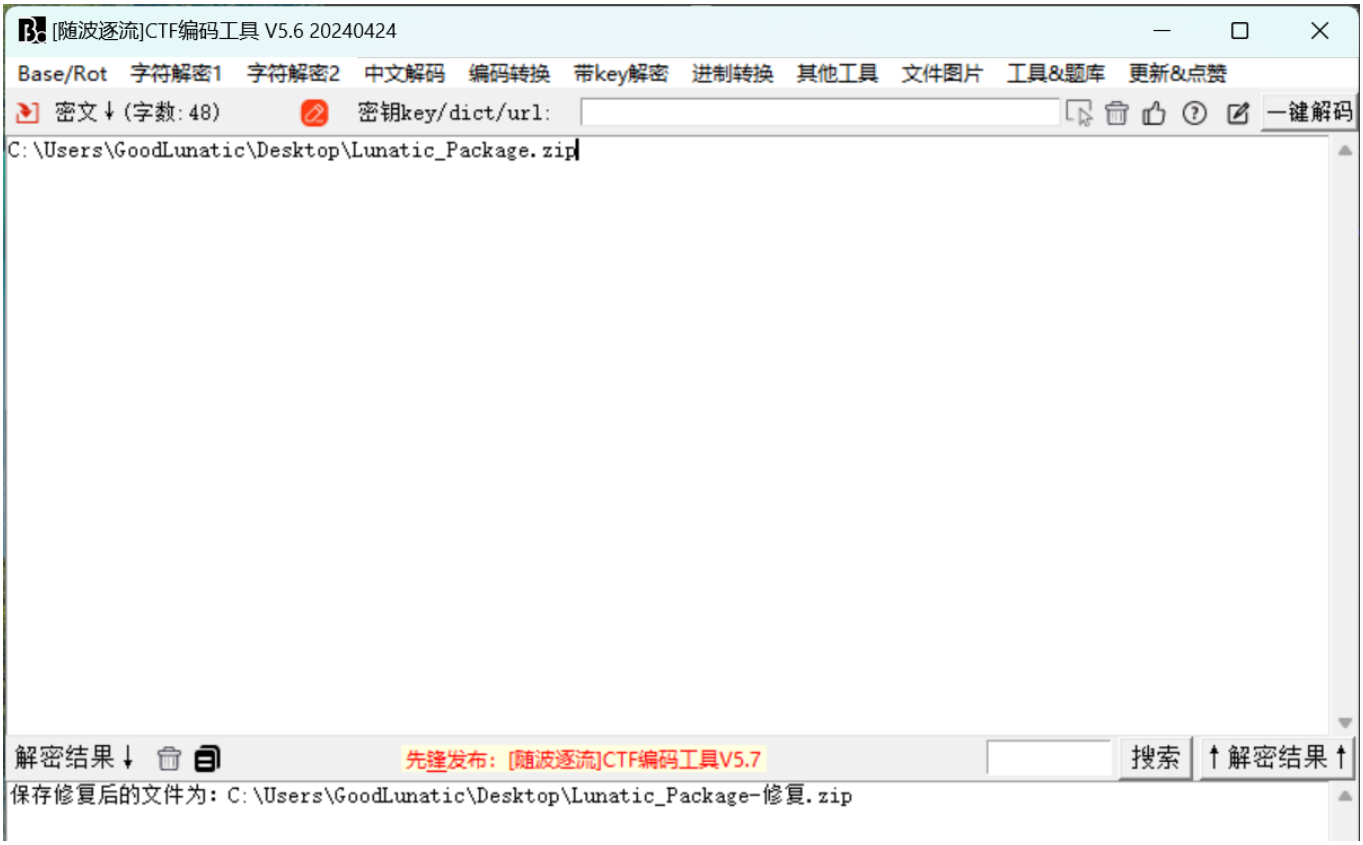
Lunatic_Package Writeup

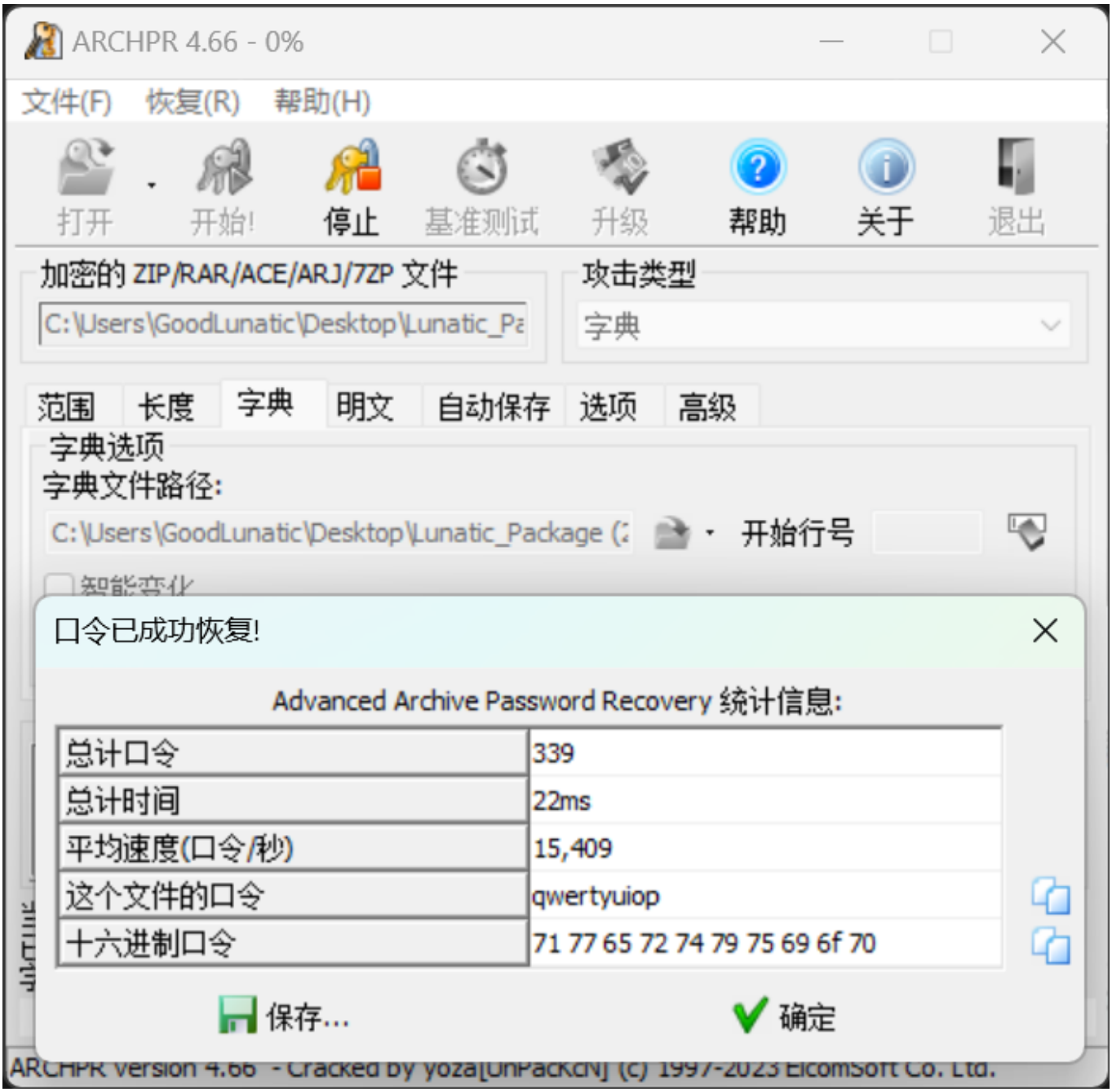
题目附件给了一个压缩包，使用010打开，发现压缩文件目录区存在 09 00
因此猜测就是压缩包伪加密，直接使用010将 09 00 改成 00 00 即可去除伪加密



如果懒得手改，也可以使用 高版本的随波逐流 直接改



解压压缩包，可以得到 rockyou.txt 和 Weak-Passwd.zip
如果有经验的话就会知道，rockyou 是一个非常有名的弱密码字典
因此这里考察的就是弱密码字典爆破压缩包



爆破可以得到压缩包密码：qwertyuiop

解压压缩包，得到 key.txt 和 Plaintext-Attack.zip

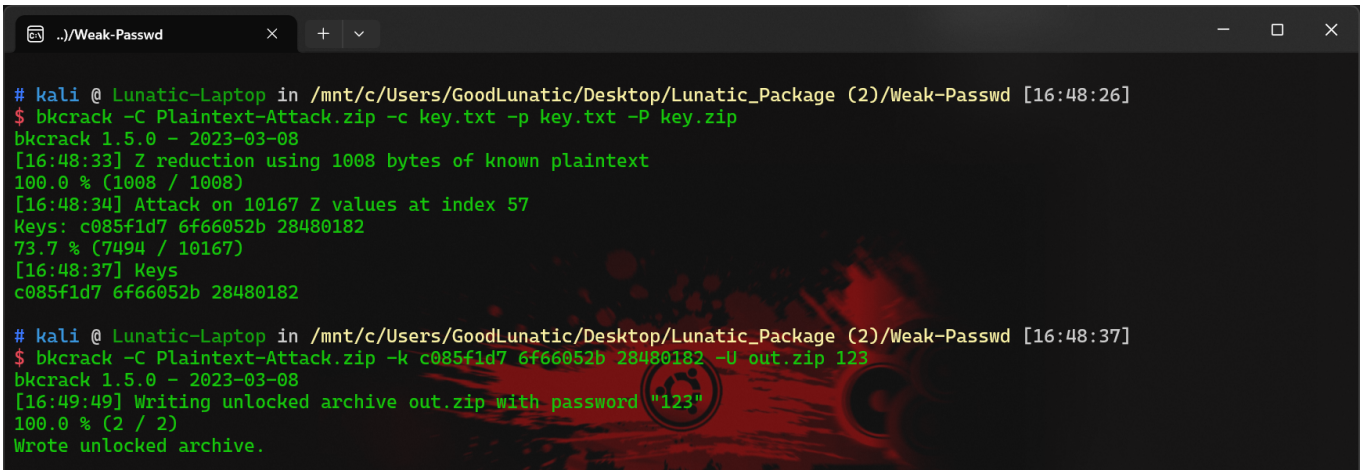
压缩包的名称和 store 的压缩方式就提示了这里考察的是明文攻击

然后发现文件压缩后的大小要大于原始大小，因此可以知道压缩的方式是仅存储

我们这里将 key.txt 进行压缩，压缩级别选择仅存储

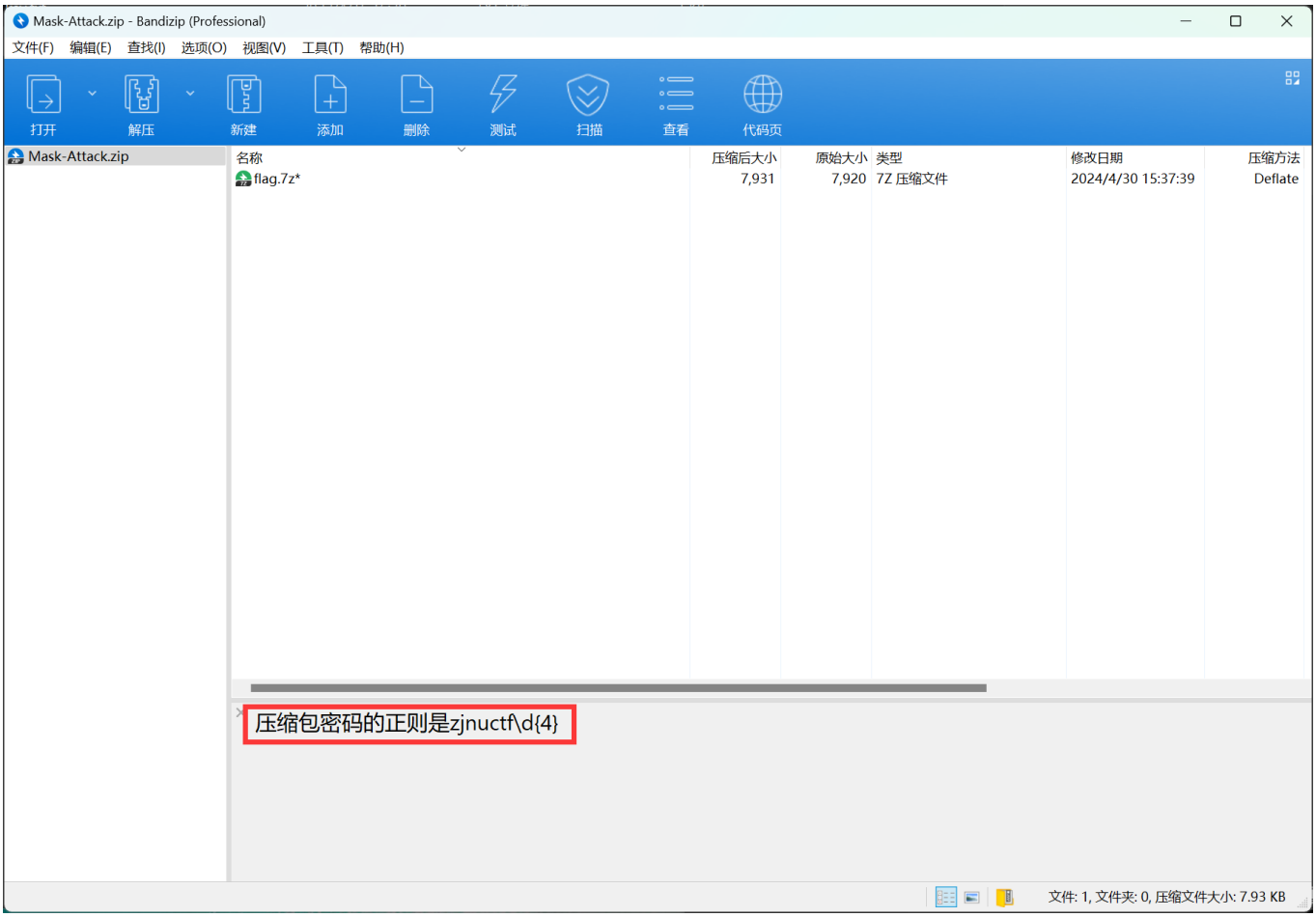
然后使用 bkcrack 进行明文攻击即可得到三段key：c085f1d7 6f66052b 28480182

使用 -U 参数把压缩包密码修改为123，然后导出新的压缩包 out.zip



使用密码 123 解压 out.zip 可以得到 Mask-Attack（掩码攻击）

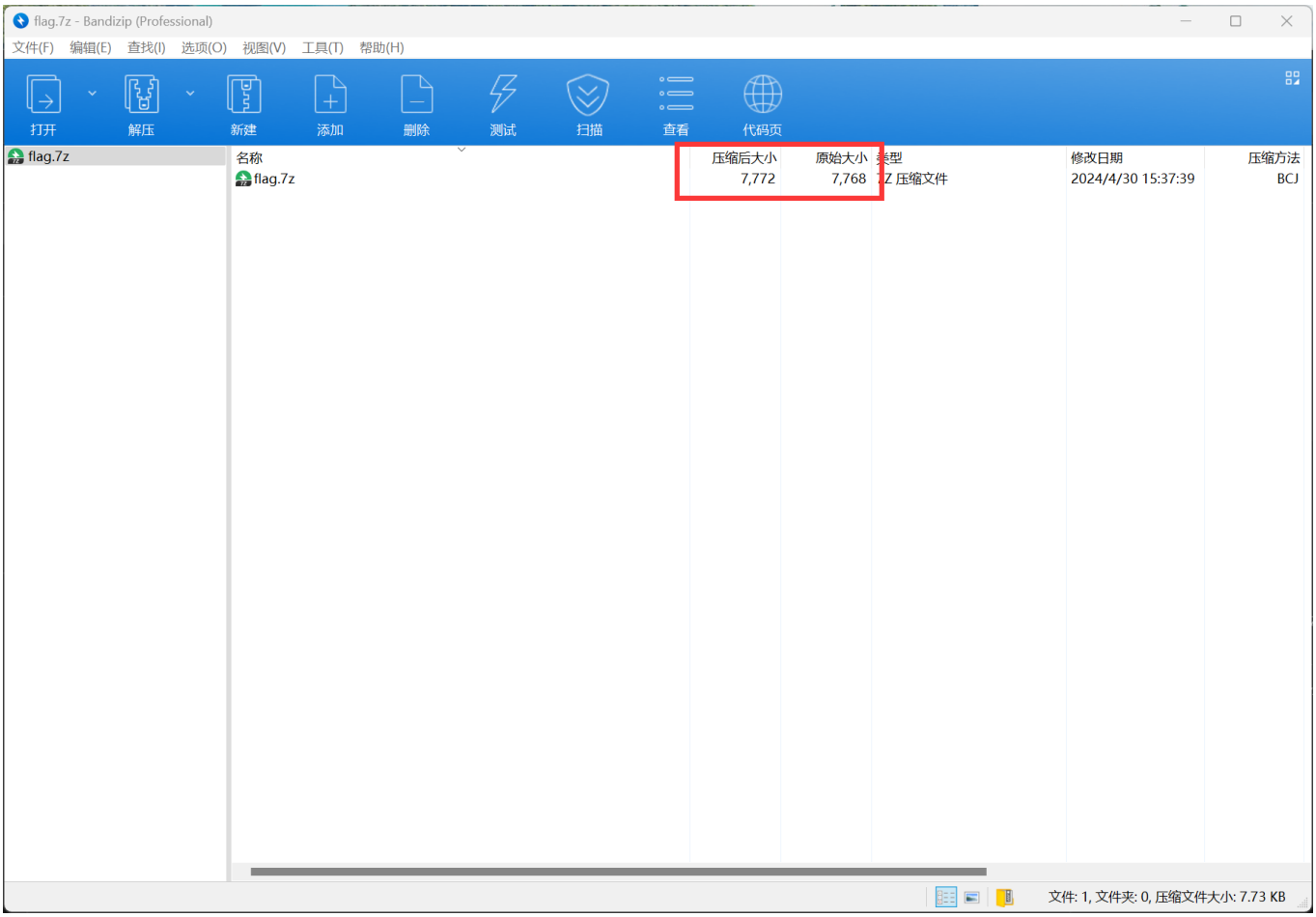
在压缩包的注释中发现注释：压缩包密码就是 zjnuctf+四位数字



直接使用爆破工具进行掩码爆破就可以得到解压密码：zjnuctf0512

解压压缩包，发现存在没有密码的压缩包套娃

Tips：这里根据压缩后的大小，可以猜测套的层数不是很多（为了降低新生赛难度特意设计的，因此这里也可以直接手动点击解套）



解压到最后一层可以得到一个 flag.txt 内容如下：

> Congratulations,but did you notice the suffix of package?
> Combine all of them plz.

> hint: Morse Code
> zip -> .
> 7z -> -
> tar.gz -> space

很明显，这里有一个压缩包后缀的隐写，然后隐写的内容是摩斯电码
最后提取出来的摩斯电码如下：

> --.. .--- -. ..- -.-. - ..-. ----.-- ..-. ..- -. -----.-

最后解码摩斯电码即可得到最后的 flag
Tips：这里的 %u7b 和 %u7d 分别就是 { 和 }

zjnuctf{fun}或zjnuctf{FUN}

附录：
出题脚本.py

```python
import zipfile
import tarfile
import py7zr
import os


def check_exist(directory_path):
    if os.path.exists(directory_path):
        print(f"[+]{directory_path}目录已存在")
    else:
        os.mkdir(directory_path)
        print(f"[+]{directory_path}目录创建成功")


def compress_7z(archive_file):
    check_exist("./tmp")
    output_file = "./tmp/flag.7z"
    output_file_name = "flag.7z"
    with py7zr.SevenZipFile(output_file, 'w') as archive:
        archive.write(archive_file)
    print(f"[+] {archive_file} ⟹ {output_file} 成功")
    os.remove(archive_file)
    os.system("mv ./tmp/flag.7z ./flag.7z")
    os.removedirs("./tmp")
```

```
    return output_file_name


def compress_zip(archive_file):
    check_exist("./tmp")
    output_file = "./tmp/flag.zip"
    output_file_name = "flag.zip"
    with zipfile.ZipFile(output_file, 'w') as zf:
        zf.write(archive_file)
    print(f"[+] {archive_file} == > {output_file} 成功")
    os.remove(archive_file)
    os.system("mv ./tmp/flag.zip ./flag.zip")
    os.removedirs("./tmp")
    return output_file_name


def compress_tar_gz(archive_file):
    check_exist("./tmp")
    output_file = "./tmp/flag.tar.gz"
    output_file_name = "flag.tar.gz"
    with tarfile.open(output_file, 'w:gz') as tar:
        tar.add(archive_file)
    print(f"[+] {archive_file} ⟹ {output_file} 成功")
    os.remove(archive_file)
    os.system("mv ./tmp/flag.tar.gz ./flag.tar.gz")
    os.removedirs("./tmp")
    return output_file_name


if __name__ == "__main__":
    morse_code = "--.. .--- -. ..- -·-. - ..-. ——.-- ..-. ..- -. ——·-"
    morse_code = morse_code[::-1]
    print(morse_code)
    length = len(morse_code)
    archive_file = "flag.txt"
    for i in range(length):
        if morse_code[i] == '.':
            archive_file = compress_zip(archive_file)
            print(archive_file)
        elif morse_code[i] == '-':
            archive_file = compress_7z(archive_file)
            print(archive_file)
        else:
            archive_file = compress_tar_gz(archive_file)
            print(archive_file)
```

解题脚本.py（提取压缩包后缀隐写的摩斯电码）

```
import zipfile
import tarfile
import py7zr
import os


def decompress_7z(archive_file):
    with py7zr.SevenZipFile(archive_file, 'r') as archive:
        file_list = archive.list()
        new_archive_file = file_list[0].filename
    with py7zr.SevenZipFile(archive_file, mode='r') as archive:
        archive.extractall("tmp/")
    os.remove(archive_file)
    os.system("mv tmp/* .")
    os.rmdir("tmp")
    return new_archive_file


def decompress_zip(archive_file):
    with zipfile.ZipFile(archive_file, 'r') as zip_ref:
        file_list = zip_ref.namelist()
        new_archive_file = file_list[0]
    os.mkdir("tmp")
    with zipfile.ZipFile(archive_file, 'r') as zip_ref:
        zip_ref.extractall(path="tmp/")
    os.remove(archive_file)
    os.system("mv tmp/* .")
    os.rmdir("tmp")
    return new_archive_file


def decompress_tar_gz(archive_file):
    with tarfile.open(archive_file, "r:gz") as tar_ref:
        file_list = tar_ref.getnames()
        new_archive_file = file_list[0]
    os.mkdir("tmp")
    with tarfile.open(archive_file, "r:gz") as tar:
        tar.extractall(path="tmp/")
    os.remove(archive_file)
    os.system("mv tmp/* .")
    os.rmdir("tmp")
    return new_archive_file
```

```python
if __name__ == "__main__":
    morse_code = []
    archive_file = "flag.7z"
    # archive_file = "flag.zip"
    # archive_file = "flag.tar.gz"
    # archive_file = decompress_7z(archive_file)
    # archive_file = decompress_zip(archive_file)
    # archive_file = decompress_tar_gz(archive_file)
    # print(archive_file)
    while True:
        if "7z" in archive_file:
            archive_file = decompress_7z(archive_file)
            morse_code.append("-")
        elif "tar.gz" in archive_file:
            archive_file = decompress_tar_gz(archive_file)
            morse_code.append(" ")
        elif "zip" in archive_file:
            archive_file = decompress_zip(archive_file)
            morse_code.append(".")
        else:
            break
print("后缀隐写内容如下:")
print("".join(morse_code))
```