

攻防世界misc——picture2

原创

[Captain杰派罗](#)



于 2022-01-13 17:05:39 发布



2008



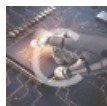
收藏

分类专栏: [攻防世界WP](#) 文章标签: [网络安全](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_45770420/article/details/122477875

版权



[攻防世界WP 专栏收录该内容](#)

32 篇文章 0 订阅

订阅专栏


```

import java.io.FileOutputStream;
import java.util.Base64;

/**
 * @Author: Jack Jparrow
 * @Date: 2022-01-13 15:07:51
 * @LastEditTime: 2022-01-13 15:13:36
 * @LastEditors: Jack Jparrow
 * @Description: base64转字符, 发现是zip文件, 直接保存出错, 转byte, 再写出为zip文件
 */

public class picture2 {

    public static void main(String[] args) {
        try {
            String str = "S1ADBBQAAQAAADkw10xs4x98WgAAAE4AAAAEAAAAY29kZePegfAPrkdnhMG2gb86/AHHP50GMqCrR9s21bP43S
qmesL+oQGo501jz4zIctqxIsTHV25+1mTE7vFc9g15IUif7f1/rHIpHq17nqKPb+2M6nRLuwH8mb/w1BLAQI/ABQAAQAAADkw10xs4x98WgAAAE
4AAAAEACQAAAAAAAAIAAAAAAAAAABjb2RlCgAgAAAAAAAAABAgAAAFvDg4Xa0wE8gAmth9rTATyACa2H2tMBUEsFBgAAAAABAAEAVgAAAHwAAADcAF
tQeXRob24gMi43XQ0KPj4+IKh9qH2ofQ0KDQpUcmFjZWJhY2sgKG1vc3QgcmVjZW50IGNhbGwgbGFzdCk6DQogIEZpbGUgIjxweXNoZWxsIzA+Ii
wgbGluZSxlcBpbIA8bW9kdWx1Pg0KICAgIKh9qH2ofQ0KwMvyb0Rpdm1zaW9uRXJyb3I6IKh9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofa
h9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofah9qH2ofa
h9qH2ofah9qH2ofah9qH2ofah9qH2ofSA8LSBwYXNzd29yZCA7KQ0KPj4+IAA=";

            Base64.Decoder de = Base64.getDecoder();

            // byte[] b = new String(de.decode(str)).getBytes();// 转String文件会出错
            byte[] c = de.decode(str);

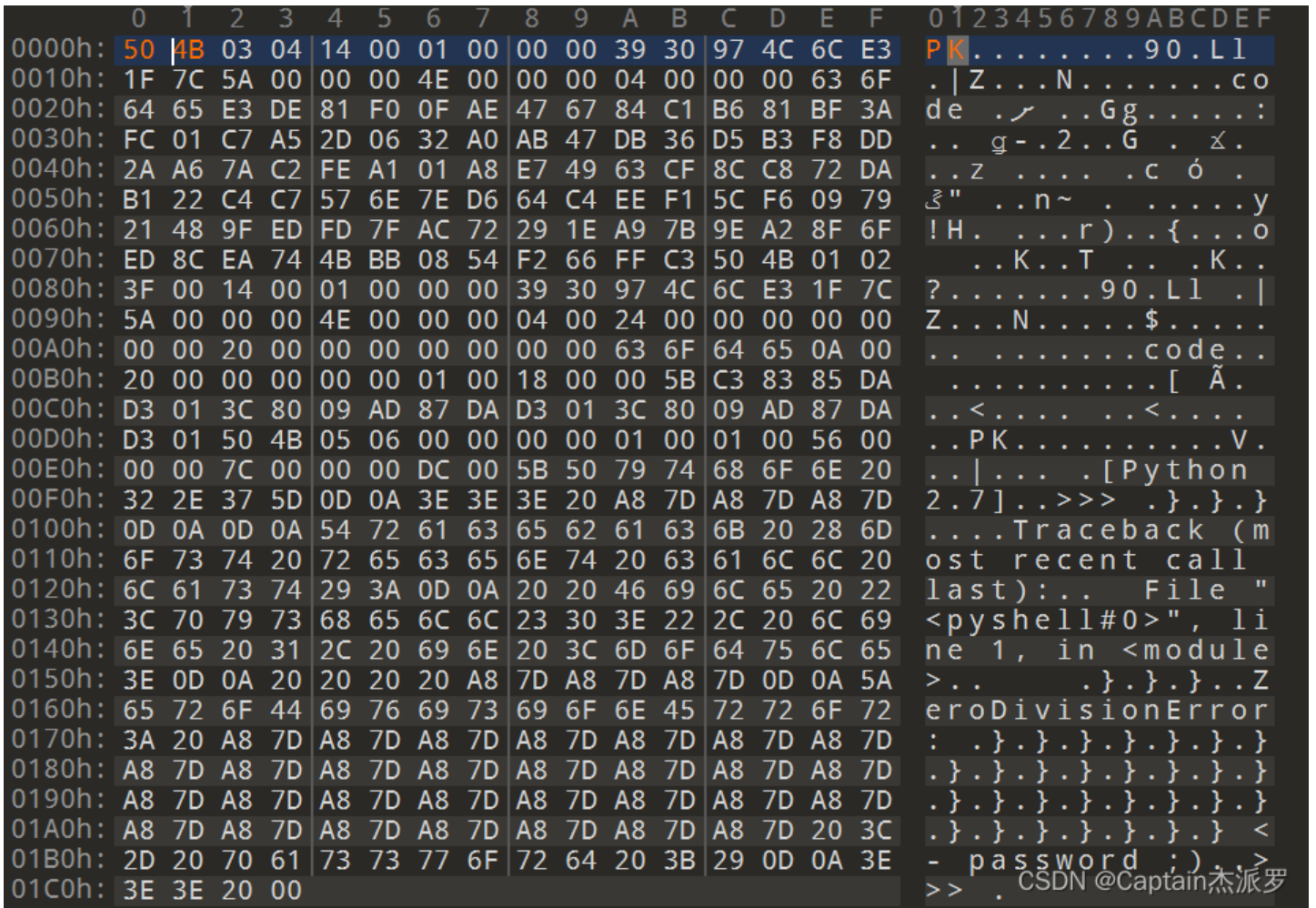
            FileOutputStream out = new FileOutputStream("D:\\Desktop\\3.zip");

            out.write(c);
            out.flush();
            out.close();

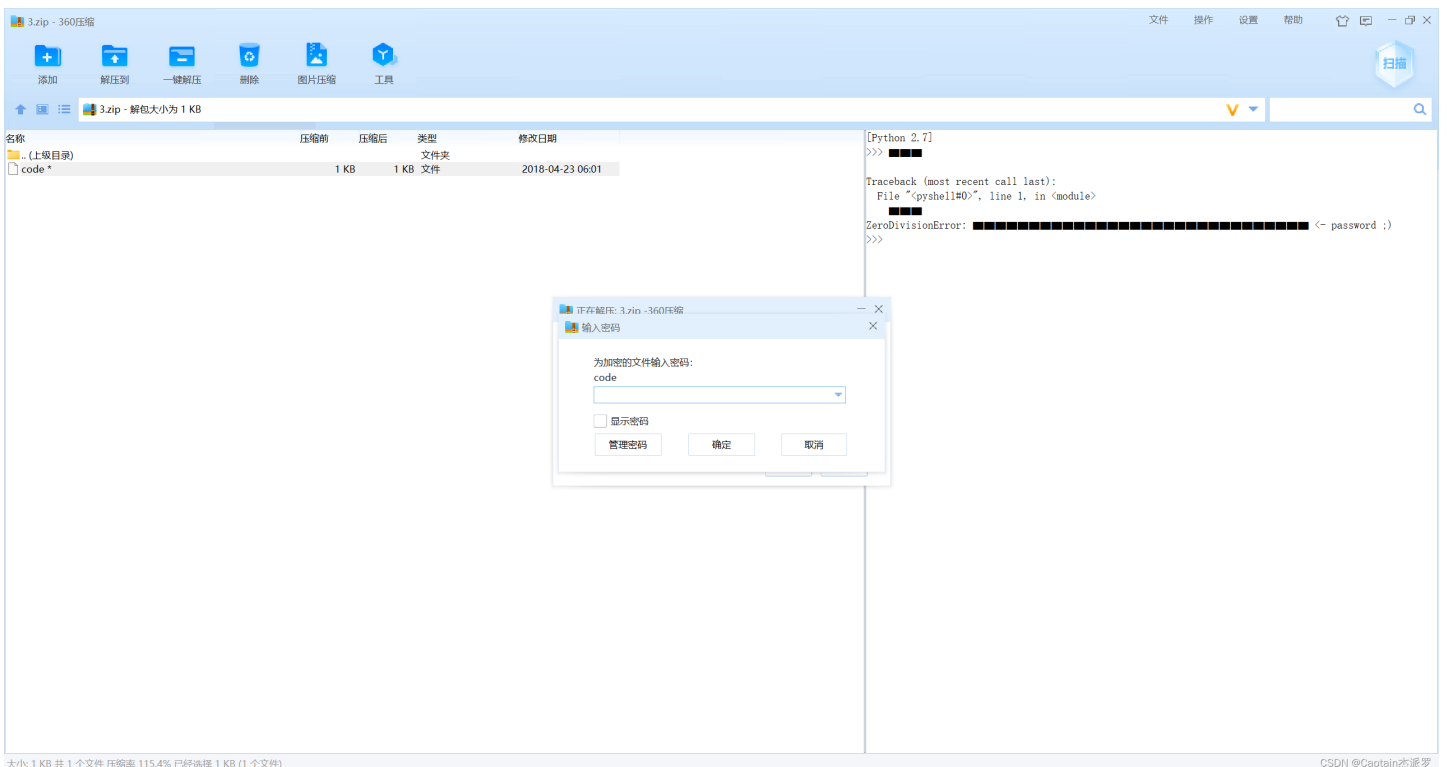
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

010把前面改成504B



解压，要密码



根据提示，密码是py2.7的一个报错提示

经过百度，得知该提示为integer division or modulo by zero

python中异常的介绍

每个异常都是一些类的实例，这些实例可以被引发，并且可以用很多种方法进行捕捉，使得程序可以捉住错误并且对其进行处理

```
>>> 1/0

Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    1/0
ZeroDivisionError: integer division or modulo by zero
```

CSDN @Captain杰派罗

解压得到加密过的字符串

```
begin 644 key.txt
G0TE30TY[,C,X.$%&,C@Y,T5".#5%0C%"-#,Y04)&1C8Q-S,Q.49]
end
```

根据其特征begin xxxx，得知为UUencode编码，解码得到flag：

UUencode

UUencode

```
G0TE30TY[,C,X.$%&,C@Y,T5".#5%0C%"-#,Y04)&1C8Q-S,Q.49]
```

字符集 utf8(unicode编码)

编码

解码

```
CISCN{2388AF2893EB85EB1B439ABFF617319F}
```

CSDN @Captain杰派罗

分割线

Uuencode的算法很简单，编码时它将3个字符顺序放入一个24位的缓冲区，缺字符的地方补零，然后将缓冲区截断成为4个部分，高位在先，每个部分6位，用下面的64个字符重新表示：

```
"`!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNPOQRSTUVWXYZ[]^_`"
```

在文件的开头有 "begin xxx

被编码的文件名"，在文件的结尾有 "end"，用来标志Uue文件的开始和结束。编码时，每次读取源文件的45个字符，不足45个的用 "NULL" 补足为3的整数倍（如：23补为24），然后输入目标文件一个ASCII为："32+实际读取的字符数"的字符作为每一行的开始。读取的字符编码后输入目标文件，再输入一个"换行符"。如果源文件被编码完了，那么输入 "` (ASCII为96)" 和一个"换行符"表示编码结束。

解码时它将4个字符分别转换为4个6位字符后，截取有用的后六位放入一个24位的缓冲区，即得3个二进制代码。 @Captain杰派罗