SQL注入你真的了解过报错注入吗?

原创

山与路 于 2020-04-08 18:34:29 发布 3026 收藏 1

分类专栏： CTF

本文链接：https://blog.csdn.net/a1309525802/article/details/105394272

SQL-2

1.为什么要去理解该知识点

2.原理

3.自己的理解和实践

4.CTF题目案例

声明一点以下可能说的不是很友好,所以如有敏感词汇请评论我,作者会改

日常吐槽:SQL注入知识很多,让我头很疼,但是慢慢学我也发现其实sql并没有想象的那么难,只是思维要放开一点,不能局限,而且学sql注入真的不能心急,想我以前学sql注入真的就是心急的不得了,看完视频也不去尝试,搞到最后几乎没学到什么,不会的真的可以去找度娘,现在很多大佬都会分享这些知识,所以慢慢学一定能学好,心急吃不了热豆腐

为什么要去理解该知识点

今天要讲讲sql注入中常见的注入类型:报错注入,报错注入在CTF还OK,而且也挺麻烦的,所以大佬们大多都是用脚本来进行爆破,或者用神器sqlmap,我丢这玩意真好用,但是对于菜鸡的我还是不怎么会用,只会简单的语句

原理

这里要补充一下昨天sql绕过知识点,我也是今天在搜索sql_puzzing时看到的 先借那些大佬的话:看到隔壁表哥的博客发现了这种绕过

```
SELECT * FROM `user` WHERE id=1=0=1
```

id	username	password
2	lisi	lisi
3	wangwu	wangwu

很神奇,为什么不会输出id=1的值而且为什么这样的语法没有错
经过老夫反复观察和实验才发现mysql语句其实真的很像php这类编程语言一样,语法也相似
上面的sql语句其实和下面的php语句执行时是一样的吧(其实我也不确定)

```
$pd=(1==0==1)
If(($pd==true)
SELECT * FROM `user` WHERE id=1
Else
SELECT * FROM `user` WHERE id!=1
```

这样一看应该就明白很多了吧,主要是身为菜鸡的我只知道sql的=,却忘了他还有应该!=
其实跟疑惑的是什么,这个绕过有什么用,那作用就大了
1.首先它可以无限延展,主要=,'没被过滤那用它还真好用
2.有时你发现你输入and,or什么的页面没有发生什么变化,所以就判断没有注入其实不然,因为它可能把and,or替换为#或者'',所以你就会发现页面没有什么变化,这时这种方法岂不是就OK
步入今天的话题

报错注入

有些Sql真好,知道我哪里出问题了就马上报出了让你知道,像我这样的直男,连自己错了都不知道错那里,所以sql要是个姑娘那可就真的好,既然你告诉我错哪里了那我就改好吧, 分为二类: 1.xpath语法错误 2.concat+rand()+group_by()导致主键重复

xpath语法错误

extractvalue函数

理解其漏洞那就必然要理解函数的作用: 函数原型：extractvalue(xml_document,Xpath_string) 正常语法：extractvalue(xml_document,Xpath_string); 第一个参数：xml_document是string格式，为xml文档对象的名称 第二个参数：Xpath_string是xpath格式的字符串 作用：从目标xml中返回包含所查询值的字符串 漏洞在于第二个参数,字符串漏洞其实有多,因为第二个参数只要xpath的格式字符串所以可以使用concat函数形成新的字符串,concat就有执行漏洞,xpath语法字符串有一个特点就是报错会将XPath_string打印出来

```
select extractvalue(1,concat('~',(select database())))
```

```
1  select extractvalue(1,concat('~',(select database())))
```

信息 状态

```
select extractvalue(1,concat('~',(select database())))
> 1105 - XPATH syntax error: '~test'
> 时间: 0s
```


网上有很多大佬都是这样的代码,但是你知道的CTF你懂的我也懂,不把他过滤我就感觉没意思,直接过滤掉看你怎么办,转十六进制,没事你很聪明,但是我还是可以过滤掉,没办法了怎么办其实问题不大,并不是只有才行,这个函数的漏洞主要在于报错会将XPath_string打印出来,而且上面的原理在于以开头的内容不是xml格式的语法,concat过滤马上找等价的,过滤了没事,只要开头的内容不是xml格式的语法都行
注意:extractvalue()能查询字符串的最大长度为32,所以要用substring进行截取

updatexml函数

updatexml()函数与extractvalue()类似，是更新xml文档的函数。 语法updatexml(目标xml文档，xml路径，更新的内容) 其实updatexml于extractvalue等价,所以漏洞也类似 3.

concat+rand()+group_by()导致主键重复

无疑这个知识点是报错注入的核心知识点,而且不易懂,但是能我好像懂了 必要函数: 1.Count 2.Rand 3.Group by 4.Floor Count无疑很简单就是计数 Rand编程语言都知道的产生随机数 Group by XX 通过XX进行分组 Floor编程语言都知道的向下取整,例如floor(1.1111)=1 明白函数的作用那就请看大屏幕 ``` select count(*) from user group by floor(rand(0)*2); ``` 看到代码先不着急,慢慢分析,理解原理以后绕过简单 将上面代码白话文说就是 将user表中先通过floor(rand(0)*2)这个key进行分组,然后查user表中与key相等的然后+1 疑惑点: 1.floor(rand(0)*2)得到的key是什么 2.Group by 分组后是然后进行统计的 一步步分析 1.解: 0

```
select count(*) from user group by id
```

id	username	password
1	zhangshan	zhangshan
2	lisi	lisi
3	wangwu	wangwu

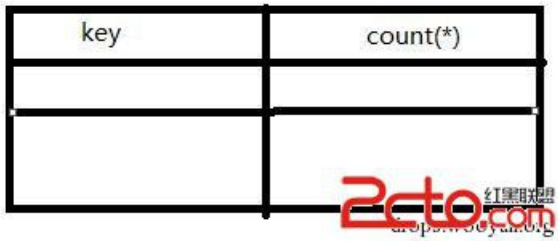
表先给你看,免得说我无图无真相
首先通过id分组,就可以分为

id
1
2
3

所以key={id=1,id=2,id=3},这是全部执行完后的结果
为什么要展示这个还不是防止你看的稀里糊涂
开始执行第一个语句,首先虚拟表为空
查询第一行数据,也就是select user表
发现id=1,然后select * from 虚拟表 where id=1
看虚拟表有没有id=1
没有OK ! insert into count=1 到虚拟表中
如果有,那就update count+=1
以此类推
这个很容易理解吧
Ok现在我们走向难一点的那个

```
select count(*) from user group by floor(rand(0)*2);
```

根据上面的思路来
我们现在知道key={0,1}
查询第一行数据,此时假设floor(rand(0)*2)=1,也就是key=1
因为没有任何一列等于key,所以msql会将key设为单独的一列,也就是floor(rand(0)*2)变成了数值而不是类名
所以呢所以呢,继续像上面那样做
看虚拟表中有没有key(1)
没有ok! Insert 1 into count=1到虚拟表
…
这样的逻辑好像没错啊,那为什么会报错是我哪里有问题吗
嗯,是的,没错,你错了,只错在一点,floor(rand(0)*2)会多次执行
在查询时执行一次,在插入的时候其实也执行了一次,而且虚拟表有一个特征insert的key都是主链,也就是唯一,表中只允许出现一次
执行的插入语句不是上面这条,而是insert floor(rand(0)*2) into count=1
这怎么会报错,别急慢慢来,首先为了方面表 floor(rand(0)*2)设为x
在查询时x=1,虚拟表不存在x=1的key 所以执行insert,此时不管x变成1还是0
虚拟表都不存在这些值,所以不会报错
但是下一次就不一定这么幸运了,假设虚拟表中存储了x=1,在查询时x=0,虚拟表不存在x=0,所以insert,此时x变成1 ,变成insert x into count=1 ,因为虚拟表已经存在x=1,x=1又是主链所以必然会报错
没图没意思



第一次看不懂那就多看几遍.慢慢来慢慢分析

现在分析一些自己提出的问题

1.只能是floor?
2.Rand括号里的只能是0吗?
3.只能count和group by联合?
4.只能让floor(rand(0)*2)的取值在0-1吗,就不能再大一点吗?
1.解:那不一定,向下取整也是可行的,只要让key=二个相邻的整数就行了
2.解:不是,rand是随机数,rand(0)输出的种子会有规律可寻,而rand()无规律可循,那到时看你是非洲人还是欧洲人了,0也可以换成其他的,只要你找其规律请随意
3.解:目前是的,因为其他聚合函数我反正试了报了其他错误
4.只要让key=二个相邻的整数就行了,别让key的值太多,不然真的很麻烦

CTF题目案例

因为报错注入遇到的CTF比较少,所以所以所以.
请查考这位大佬的writeup,原题我也没找到