二阶注入

1. 二次注入介绍

二次注入的概念

二次注入也有人称它为 SQL 二阶注入。
二次注入漏洞是一种在 Web 应用程序中广泛存在的安全漏洞形式。相对于一次注入漏洞而言，二次注入漏洞更难以被发现，但是它却具有与一次注入攻击漏洞相同的攻击威力。
简单的说，二次注入是指已存储（数据库、文件）的用户输入被读取后再次进入到 SQL 查询语句中导致的注入。
网站对我们输入的一些重要的关键字进行了转义，但是这些我们构造的语句已经写进了数据库，可以在没有被转义的地方使用
可能每一次注入都不构成漏洞，但是如果一起用就可能造成注入。

普通注入与二次注入的区别：

普通注入：
在 http 后面构造语句，是立即直接生效的
一次注入很容易被扫描工具扫描到
二次注入：
先构造语句（有被转义字符的语句）
我们构造的恶意语句存入数据库
第二次构造语句（结合前面已经存入数据库的语句，成功。因为系统没有对已经存入数据库的数据做检查）
二次注入更加难以被发现
二次注入可以理解为，攻击者构造的恶意数据存储在数据库后，恶意数据被读取并进入到 SQL 查询语句所导致的注入。防御者可能在用户输入恶意数据时对其中的特殊字符进行了转义处理，但在恶意数据插入到数据库时被处理的数据又被还原并存储在数据库中，当 Web 程序调用存储在数据库中的恶意数据并执行 SQL 查询时，就发生了 SQL 二次注入。

二次注入的条件

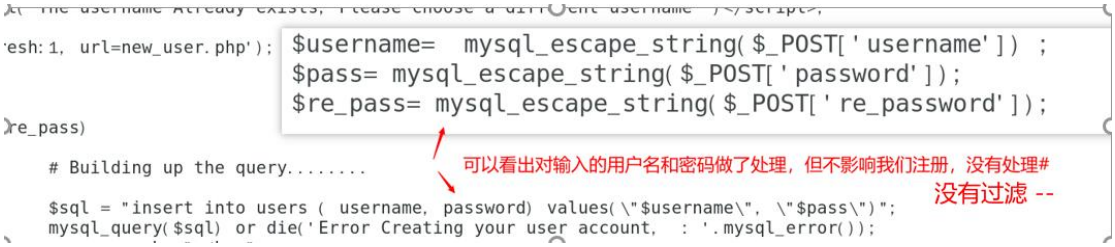
(1)用户向数据库插入恶意语句(即使后端代码对语句进行了转义，如 mysql_escape_string、mysql_real_escape_string 转义)
(2）数据库对自己存储的数据非常放心，直接取出恶意数据给用户

二次注入步骤

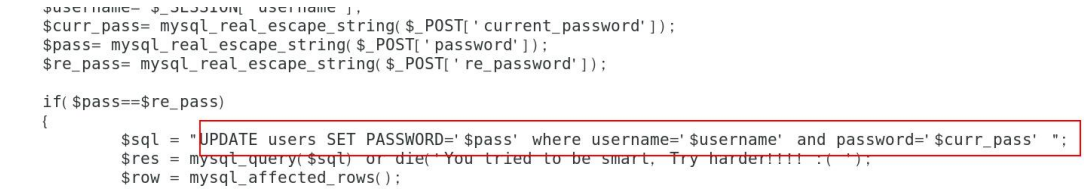
二次注入，可以概括为以下两步：
第一步：插入恶意数据
进行数据库插入数据时，对其中的特殊字符进行了转义处理，在写入数据库的时候又保留了原来的数据。
第二步：引用恶意数据
开发者默认存入数据库的数据都是安全的，在进行查询时，直接从数据库中取出恶意数据，没有进行进一步的检验的处理。

2. 二次注入代码分析

首先看注册的代码 cat login_create.php



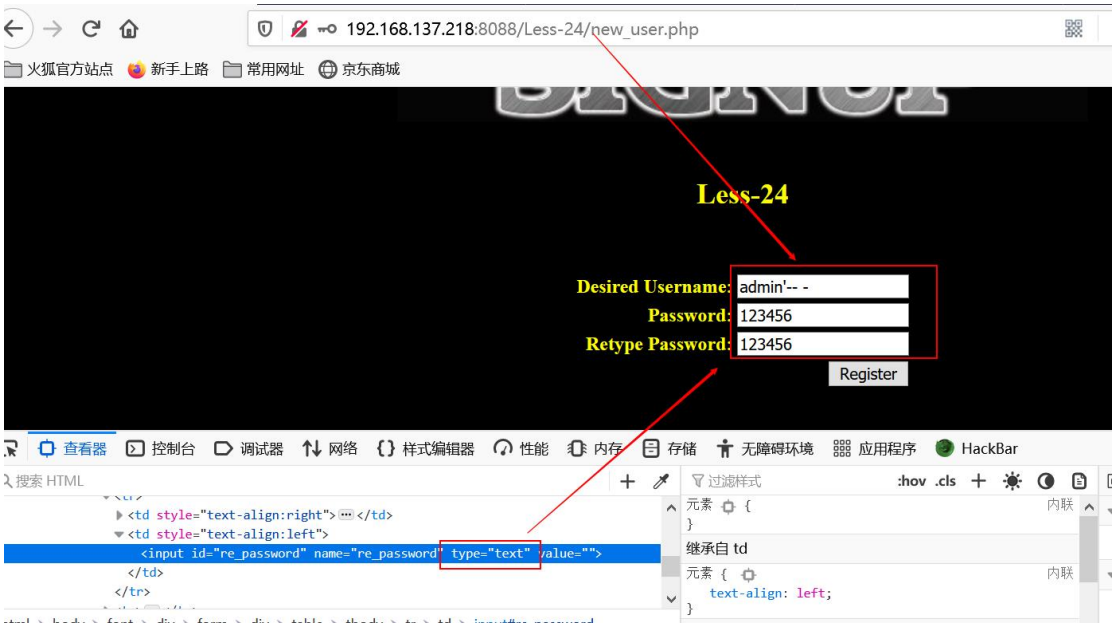
我们可以利用- -和#做很多事情
再来看修改密码的代码 pass_change.php



这里如果修改的 username 是 admin’ -- - 那么实际上我们修改的是 admin 的 password
也就是说我们注册了一个 admin’-- -这样的一个用户，但修改可以修改 admin 的密码

3. 二次注入利用

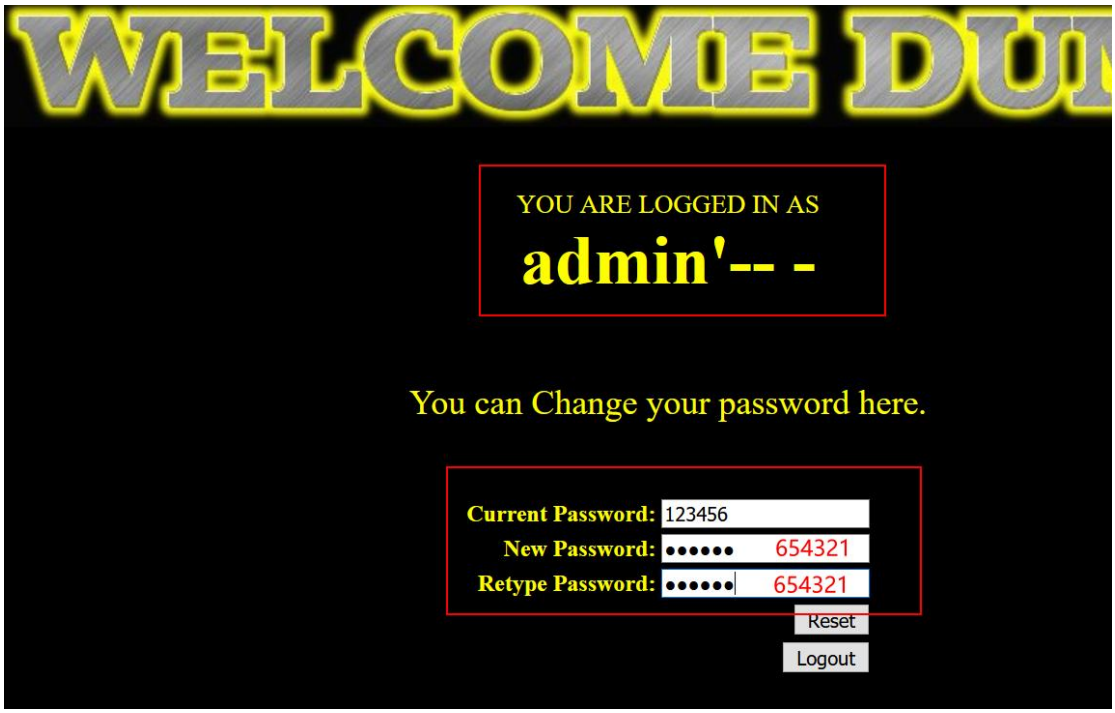
第一步：注册一个 admin’-- -,注意，后面这个-表示前面是一个注释，因为注释必须是空格才能生效



进入到数据库查看

id	username	password
1	Dumb	Dumb
2	Angelina	I-kill-you
3	Dummy	p@ssword
4	secure	crappy
5	stupid	stupidity
6	superman	genious
7	batman	mob!le
8	admin	admin
9	admin1	admin1
10	admin2	admin2
11	admin3	admin3
12	dhakkan	dumbo
14	admin4	admin4
15	admin'-- -	123456

第二步，用 admin’-- -用户登录，修改 admin’-- -的密码




```
mysql> select * from users;
+----+------------+------------+
| id | username   | password   |
+----+------------+------------+
|  1 | Dumb       | Dumb       |
|  2 | Angelina   | I-kill-you |
|  3 | Dummy      | p@ssword   |
|  4 | secure     | crappy     |
|  5 | stupid     | stupidity  |
|  6 | superman   | genious    |
|  7 | batman     | mob!le     |
|  8 | admin      | 654321     |
|  9 | admin1     | admin1     |
| 10 | admin2     | admin2     |
| 11 | admin3     | admin3     |
| 12 | dhakkan    | dumbo      |
| 14 | admin4     | admin4     |
| 15 | admin'--  - | 123456     |
+----+------------+------------+
```

修改的是admin的密码为654321

4. 二次注入危害

(1) 检测需要严密审查代码，很难自动化
(2) 二次注入漏洞更难以被发现，但是它却具有与一次注入攻击漏洞相同的攻击威力。

5、二次注入的流量分析

流量检测的时候要多监控注册这种流量，看注册的用户名密码中是否有特殊字符如：’ -- - #等