宽字节注入

一、宽字节注入基础

（1）、宽字节概念

1、单字节字符集：所有的字符都使用一个字节来表示，比如 ASCII 编码(0-127)
2、多字节字符集：在多字节字符集中，一部分字节用多个字节来表示，另一部分（可能没有）用单个字节来表示。
3、宽字节注入是利用 mysql 的一个特性，使用 GBK 编码的时候，会认为两个字符是一个汉字
PHP 中编码为 GBK，函数执行添加的是 ASCII 编码，MYSQL 默认字符集是 GBK 等

（2）、php 中的宽字节

addslashes()函数
1、addslashes() 函数返回在预定义字符之前添加反斜杠的字符串。
2、预定义字符：单引号（'），双引号（"），反斜杠（\），NULL
3、实例

```
<?php
$ss=addslashes('aiyou"bu"cuoo');
echo($ss);
?>
```

运行结果：aiyou\"bu\"cuoo

%DF':会被 PHP 当中的 addslashes 函数转义为“ %DF\'”，\在 URL 里是“%5C"，那么也就是说，“%DF'"会被转成“%DF%5C%27，之后再数据库查询语句进行 GBK 多字节编码，即一个中文占用两个字节，一个英文同样占用两个字节且在汉字编码范围内两个编码为一个汉字。然后 MySQL 服务器会对查询语句进行 GBK 编码即%df%5c 转换成汉字"運",单引号逃逸出来，从而绕过转义造成注入漏洞。

现在基本都会将 mysql 的连接配置设置为：

[set character_set_client=binary]

涉及到的其他函数
涉及到其他函数
mysql_real_escape_string()函数转义 SQL 语句中使用的字符串中的特殊字符
mysql_escape_string()转义一个字符串

数据库使用的是 GBK 编码,PHP 编码为 UTF8 就可能出现宽字节注入，原因是为了防止发生 SQL 注入,会调用上面所介绍的几种函数,将单引号或双引号进行转义操作,在单或双引号前加上斜杠(\）。当数据库使用的是宽字节编码会将两个连在一起的字符会被当做是一个汉字,而在 PHP 使用的 UF8 编码则认为是两个独立的字符。

二、宽字节注入代码分析

两个条件都要有
Less-33 源代码分析

```
$fp=fopen('result.txt','a');
fwrite($fp,'ID:'.$id."\n");
fclose($fp);

// connectivity

mysql_query("SET NAMES gbk");
$sql="SELECT * FROM users WHERE id='$id' LIMIT 0,1";
$result=mysql_query($sql);
$row = mysql_fetch_array($result);
```

```
function check_addslashes($string)
{
    $string= addslashes($string);
    return $string;
}
```

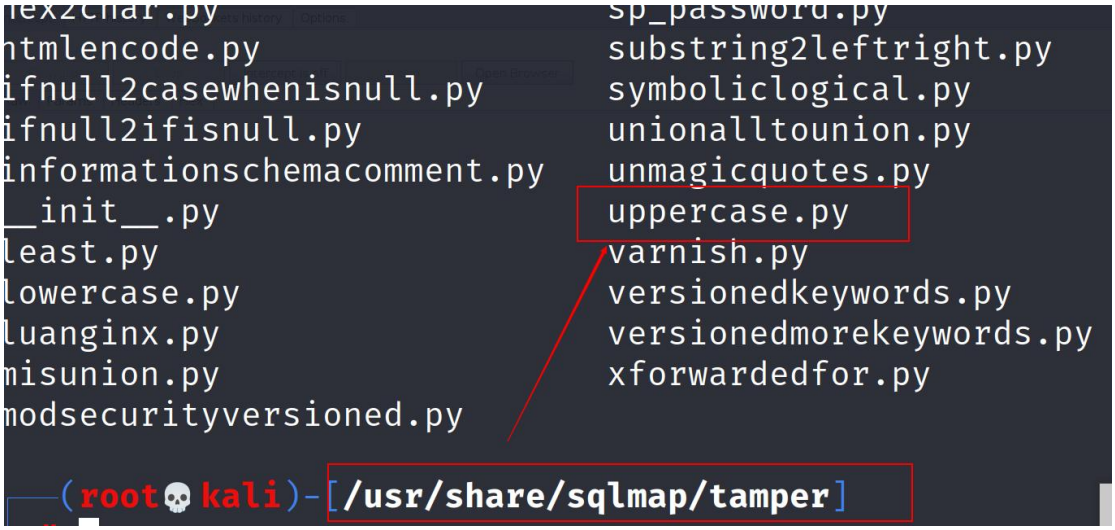
两个条件都要有

三、宽字节手动注入演示

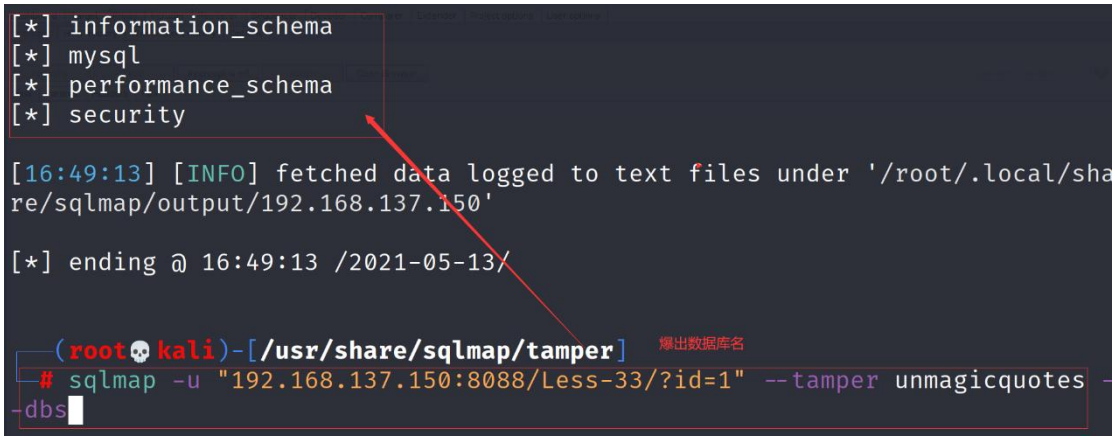
Less-33
Payload
http://192.168.137.150:8088/Less-33/?id=10000%df%27%20union%20select%201,2,3%20--+
注入后结果

四、sqlmap 进行宽字节注入

用 sqlmap 进行宽自节注入，这些需要一个脚本
脚本名：unmagicquotes.py
脚本位置：



作用：宽字符绕过
用法：sqlmap -u "192.168.137.150:8088/Less-33/?id=1" --tamper unmagicquotes -dbs
Sqlmap 测试结果



五、流量分析

%df’ 或者 %bf’ 宽字节标志流量
其他常见 sql 注入流量，如
联合注入流量：union select
盲注流量：and if sleep
报错函数流量：updatexml extractvalue floor
数据库元信息流量：concat database() information_schema 等