sql注入攻击linux下的xampp网站,从SQL注入到Getshell：记一次禅道系统的渗透

转载

Scifi-gamer 于 2021-05-14 16:34:49 发布
文章标签： sql注入攻击linux下的xampp网站
SQL注入分析

网上之前有过一个9.1.2的orderBy函数的分析，但是没想到9.2.1也存在此问题，(2018.3.2号看到目前最新版本是9.8.1)。

出问题的地方是此文件的orderBy函数：\lib\base\dao\dao.class.php

```
public function orderBy($order)
{
    if($this->inCondition and !$this->conditionIsTrue) return $this;

    $order = str_replace(array('|', '•', '_'), ' ', $order);

    /* Add "`" in order string. */
    /* When order has limit string. */
    $pos    = stripos($order, 'limit');
    $orders = $pos ? substr($order, 0, $pos) : $order;
    $limit  = $pos ? substr($order, $pos) : '';
    $orders = trim($orders);
    if(empty($orders)) return $this;
    if(!preg_match('/^(\w+\.)?(`\w+`|\w+)( +(desc|asc))?( *(, *(\w+\.)?(`\w+`|\w+)( +
(desc|asc))?)?)*$/i', $orders)) die("Order is bad request, The order is $orders");

    $orders = explode(',', $orders);
    foreach($orders as $i => $order)
    {
        $orderParse = explode(' ', trim($order));
        foreach($orderParse as $key => $value)
        {
            $value = trim($value);
            if(empty($value) or strtolower($value) == 'desc' or strtolower($value) ==
'asc') continue;

            $field = $value;
            /* such as t1.id field. */
            if(strpos($value, '.') !== false) list($table, $field) = explode('.', $field);
            if(strpos($field, '`') === false) $field = "`$field`";

            $orderParse[$key] = isset($table) ? $table . '.' . $field :  $field;
            unset($table);
        }
        $orders[$i] = join(' ', $orderParse);
        if(empty($orders[$i])) unset($orders[$i]);
    }
    $order = join(',', $orders) . ' ' . $limit;

    $this->sql .= ' ' . DAO::ORDERBY . " $order";
    return $this;
}
```

对于limit后未做严格的过滤与判断，然后拼接到了order by后面导致产生注入.

$order = join(',', $orders) . ' ' . $limit;

看了一下9.8.1的修补是对limit进行正则限制，但是事实上感觉此处正则是写了一个bug，比如正常调用orderBy($order)的时候，其中$order为abc desc limit 1,1的时候，进入$limit则是limit 1,1，导致匹配失败。

```
/* Add "`" in order string. */
/* When order has limit string. */
$pos    = stripos($order, 'limit');
$orders = $pos ? substr($order, 0, $pos) : $order;
$limit  = $pos ? substr($order, $pos) : '';
if($limit and !preg_match('/^[0-9]+ *(, *[0-9]+)?$/', $limit)) $limit = '';
```

如果想要造成前台注入(无需登录)的话，就得先看看禅道开放了哪些接口，看是否有调用orderBy函数。

\zentao\module\common\model.php

```
public function isOpenMethod($module, $method)
{
   if($module == 'user' and strpos('login|logout|deny|reset', $method) !== false) return
true;
   if($module == 'api'  and $method == 'getsessionid') return true;
   if($module == 'misc' and $method == 'ping')  return true;
   if($module == 'misc' and $method == 'checktable') return true;
   if($module == 'misc' and $method == 'qrcode') return true;
   if($module == 'misc' and $method == 'about') return true;
   if($module == 'misc' and $method == 'checkupdate') return true;
   if($module == 'misc' and $method == 'changelog') return true;
   if($module == 'sso' and $method == 'login')  return true;
   if($module == 'sso' and $method == 'logout') return true;
   if($module == 'sso' and $method == 'bind') return true;
   if($module == 'sso' and $method == 'gettodolist') return true;
   if($module == 'block' and $method == 'main') return true;

   if($this->loadModel('user')->isLogon() or ($this->app->company->guest and $this->app-
>user->account == 'guest'))
   {
       if(stripos($method, 'ajax') !== false) return true;
       if(stripos($method, 'downnotify') !== false) return true;
       if($module == 'tutorial') return true;
       if($module == 'block') return true;
       if($module == 'product' and $method == 'showerrornone') return true;
   }
   return false;
}
```

其中的if($module == 'block' and $method == 'main') return true;，也就是本次漏洞的主角，继续跟进。

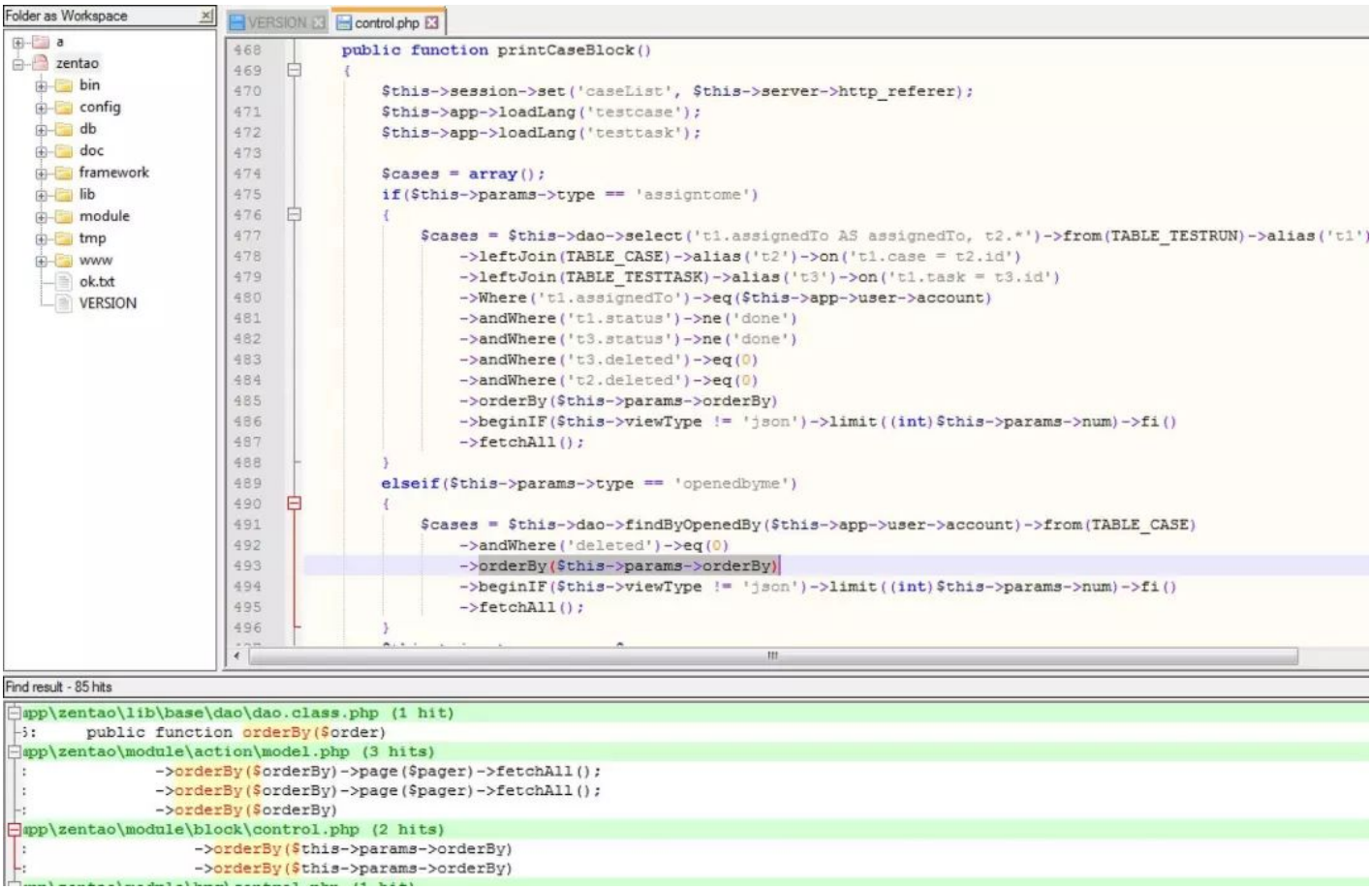
\zentao\module\block\control.php

```
class block extends control
{
    public function __construct($moduleName = '', $methodName = '')
    {
        parent::__construct($moduleName, $methodName);
        $this->selfCall = strpos($this->server->http_referer, common::getSysURL()) === 0
|| $this->session->blockModule;
        if($this->methodName != 'admin' and $this->methodName != 'dashboard' and !$this-
>selfCall and !$this->loadModel('sso')->checkKey()) die('');
    }
    public function main($module = '', $id = 0)
    {
        ...
       $mode = strtolower($this->get->mode);
       if($mode == 'getblocklist')
       {
           ...
       }
       elseif($mode == 'getblockform')
       {
           ...
       }
       elseif($mode == 'getblockdata')
       {
           $code = strtolower($this->get->blockid);

           $params = $this->get->param;
           $params = json_decode(base64_decode($params));
            ....
           $this->viewType    = (isset($params->viewType) and $params->viewType == 'json')
? 'json' : 'html';
           $this->params      = $params;
           $this->view->code  = $this->get->blockid;
           $this->view->title = $this->get->blockTitle;

           $func = 'print' . ucfirst($code) . 'Block';
           if(method_exists('block', $func))
           {
               $this->$func($module);
           }
           else
           {
               $this->view->data = $this->block->$func($module, $params);
           }
       }
    }
}
```

首先看__construct中，$this->selfCall是在验证referer的值，如果为真的话则后面的if将不会进入die语句里面

接下来跟进main函数，可以看到最后的$func = 'print' . ucfirst($code) . 'Block';，会对一些函数进行调用，与此同时，我们搜索orderBy的调用的时候可以发现printCaseBlock函数的存在

\zentao\module\block\control.php



所以前台注入的整个过程便比较清晰了，那么如何利用?

SQL注入利用

回过头来，因为禅道有windows直接的一键化安装程序，其数据库使用的也是root权限，导致可直接导出shell，但是如果没有这么高权限的时候，对于这个注入应该如何出数据。

禅道是支持多语句的，这也为后面的利用提供方便。

注入出数据库名和表段名后，当我想继续注入出用户账号密码的时候，意外地发现没有出数据。

sql = 'select 12345 from zt_user'

还是没有出数据，猜测是管理员改了表前缀，所以想去通过information_schema查询一下表名，但是意外地发现，也不能读取？难道被删了？但是我还是想知道一下表前缀。

请求的时候加了一个单引号，并且加上referer，看一下报错信息。

因为PDO的关系，SQL中的表名是%s替代的，所以未能够得到库名。

那么就利用报错去得到当前SQl语句里面查询的表名，比如利用polygon函数。

此注入点可以理解为limit后的注入点，因为使用多语句的话，报错效果不明显，所以就直接在limit后面进行注入。

上图为本地测试，但是limit的注入和mysql版本还有一些关系，目前网上的payload仅限于低版本才可报错注入出数据，很不幸运的是，目标使用的是高版本mysql。

那既然可以多语句，在不能用information_schema的情况下，可以通过下面语法来进行盲注：

show table status where name = 'xxx' and sleep(2)

写到py里面的payload是这样的

经过一番折腾发现，表前缀就是默认的zt_，但是为啥又不能够读取到用户数据呢?

仔细看到禅道里面的orderBy函数，发现做了过滤。

把下划线给过滤掉了，那这种在多语句下，可以利用mysql的预查询来绕过，值得注意的是，这个版本语法大小写敏感。

注入出admin密码的时候，惊喜地发现不能解开，无奈之下，只能先拿到一个普通账号。

Getshell

禅道在防止getshell方面还花了一点心思，曾经挖到一个可以任意写文件getshell(最新版本还存在这段代码)，不过需要的权限是管理员权限。

看了一下禅道里面人员组织架构情况，有研发、项目经理、产品经理，高层管理，系统管理员等角色，其中系统管理员虽然密码解不开，但是我们可以去解密一下高层管理的密码，因为这个角色的权限是可以修改某用户的用户组权限。在高层管理账号中，我们可以将一个普通账号修改为管理员。

接下来就是写文件Getshell：

/xampp/zentaopro/module/api/control.php

可以看到是进入了call_user_func_array，也就是我们可以任意实例化一个module方法，方法的参数也是可控的，可以通过,来分割参数。

/zentaopro/module/editor/model.php

在editor中是可以写一个文件的，filePath可控，fileContent也是可控的，这下就是可以任意写一个文件。

Exp:

但是问题又来了，前面报错里面得到的路径目录感觉像是做了权限(这里绕弯了，路径少加了一个www，所以以为是没权限写)，最终从数据库中的zt_file获取上传文件的路径，然后再将shell写入当中才得以结束。

总结

对于order by的漏洞如何进行防御的时候，我觉得上面代码在部分上有可取之处。

1、去掉limit部分，然后限制格式

2、然后循环对每个字段进行反引号的添加

整个过程就是自己在挖莫名其妙的坑，然后再一个个慢慢补上，希望能够对大家有用。