Strict domain name filtering leads to SSRF(Serve

Done #I18MKC Task c0d1M4x Opened this issue 2020-0

Test environment

OS：windows
ERMEB version：3.1.0+
download time: 2020/1/18

Code analysis

The vulnerable code is in file `/crmeb/app/admin/controller/store/CopyTaobao.php` line 108 `get_request_contents()` function.

```
/**
 * 获取货源,并解析出对应的商品参数
 * @return json
 */
public function get_request_contents()
{
    list($link) = UtilService::postMore([
        ['link', '']
    ], $this->request, suffix: true);
    $url = $this->checkurl($link);
    if ($url === false) return JsonService::fail($this->errorInfo);
    $this->errorInfo = true;
    $html = $this->curl_Get($url, time_out: 60);
    if (!$html) return JsonService::fail( msg: '商品HTML信息获取失败');
    $html = $this->Utf8String($html);
    preg_match( pattern: '/<title>([^<>]*)<\/title>/', $html, &matches: $title);
    //商品标题
    $this->productInfo['store_name'] = isset($title[1]) ? str_replace(['-淘宝网', '-tmall.com天
    $this->productInfo['store_info'] = $this->productInfo['store_name'];
    try {
```

in this function,it will call `checkurl()` function in line 113.The `checkurl()` function in line 275 restricts the use of http (s) to access the address, so other protocols are not used, but in line 280, it only needs the link to contain the words `1688` and `offer`,so it's easy to bypass.

```
 * @return string
 */
public function checkurl($link)
{
    $link = strtolower($link);
    if (!$link) return $this->setErrorInfo( msg: '请输入链接地址');
    if (substr($link, start: 0, length: 4) != 'http') return $this->setErrorInfo( msg: '链接地址必须以http开头');
    $arrLine = explode( delimiter: '?', $link);
    if (!count($arrLine)) return $this->setErrorInfo( msg: '链接地址有误(ERR:1001)');
    if (!isset($arrLine[1])) {
        if (strpos($link, needle: '1688') !== false && strpos($link, needle: 'offer') !== false) return trim($arrLine
        else if (strpos($link, needle: 'item.jd') !== false) return trim($arrLine[0]);
        else return $this->setErrorInfo( msg: '链接地址有误(ERR:1002)');
    }
    if (strpos($link, needle: '1688') !== false && strpos($link, needle: 'offer') !== false) return trim($arrLine[0]
    if (strpos($link, needle: 'item.jd') !== false) return trim($arrLine[0]);
    $arrLineValue = explode( delimiter: '&', $arrLine[1]);
    if (!is_array($arrLineValue)) return $this->setErrorInfo( msg: '链接地址有误(ERR:1003)');
```

line 280 code is like this,and it will return the link.

```
if (strpos($link, '1688') !== false && strpos($link, 'offer') !== false) return trim($arrLine[0]);
```

Then the `curl_Get()` function will be called to access the link address in line 116.

```
public function get_request_contents()
{
    list($link) = UtilService::postMore([
        ['link', '']
    ], $this->request, suffix: true);
    $url = $this->checkurl($link);
    if ($url === false) return JsonService::fail($this->errorInfo);
    $this->errorInfo = true;
    $html = $this->curl_Get($url, time_out: 60);
    if (!$html) return JsonService::fail( msg: '商品HTML信息获取失败');
```

Status
Done

Assignees
Not set

Projects
CRMEB开源商城PHP版

Pull Requests
None yet
Successfully merging a pull request will close this issue.

Duration (hours)
0

Planed to start - Planed to end
Unscheduled - Unscheduled

Top level
Not Top

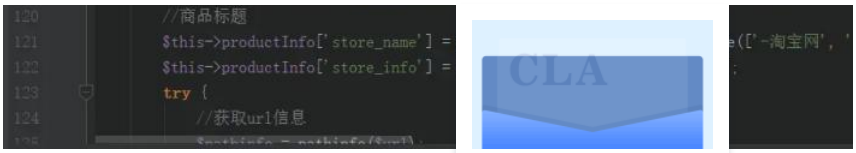
Priority
Not specified

Labels
Not set

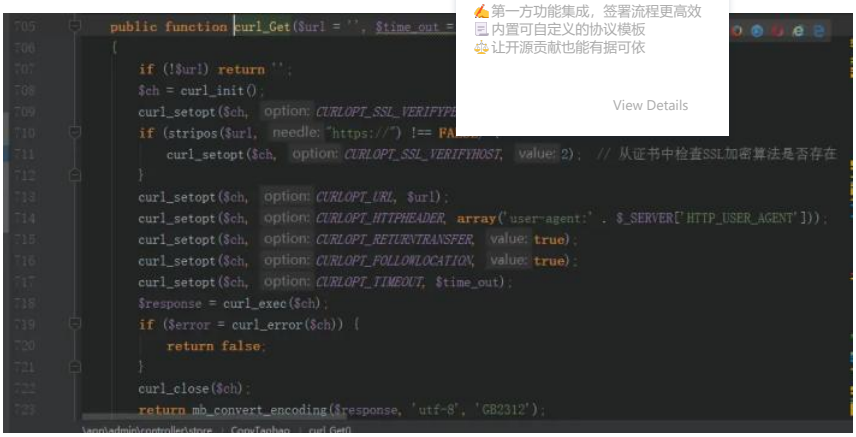
Milestones
No related milestones

Branches
No related branch

参与者（1）
C



The `curl_Get()` function is in line 705 and the code is like this.



it will filter domain name in line 134,but it should filter in call `curl_Get()` function before.And it not,so cause the SSRF.

Vulnerability test

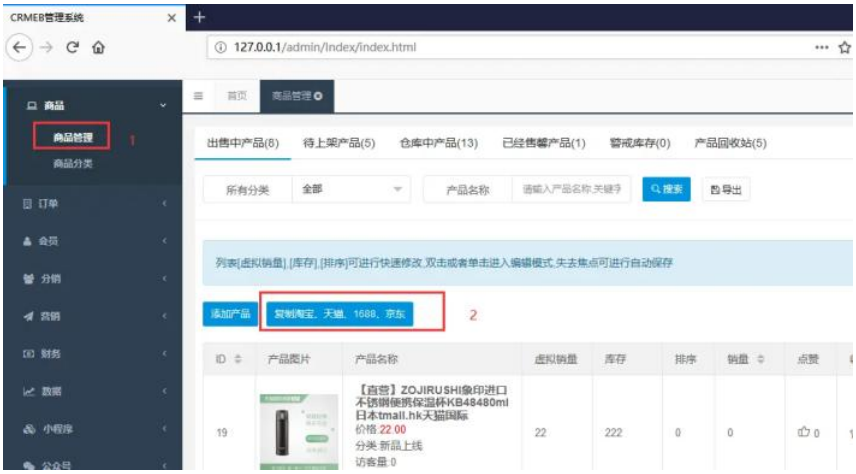
Create two web server

use the python environment to create two web server like this.

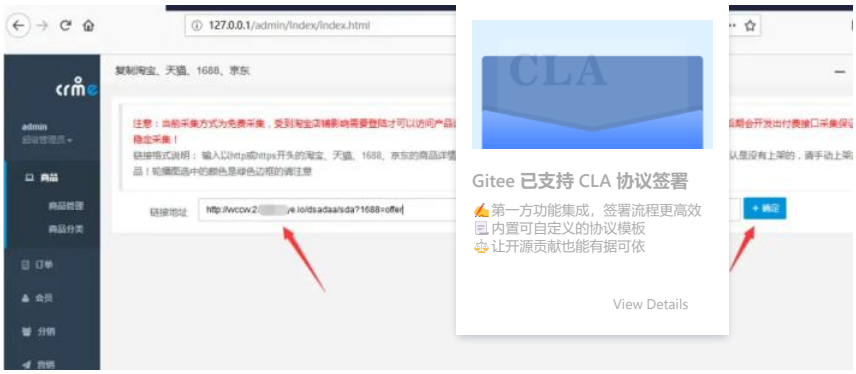
```
python2 -m SimpleHTTPServer 9999
python3 -m http.server
```

Exploit

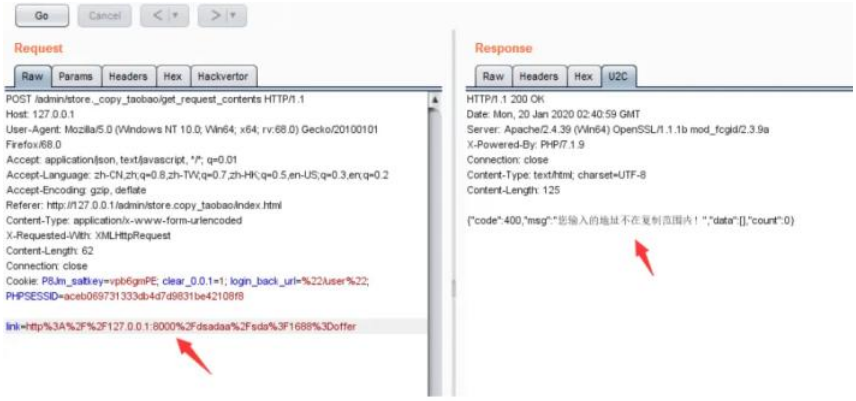
Login to the background, and then operate as follows.



Arbitrary input characters, while capturing packets using burpsuite



Modify the value of the parameter `link` .

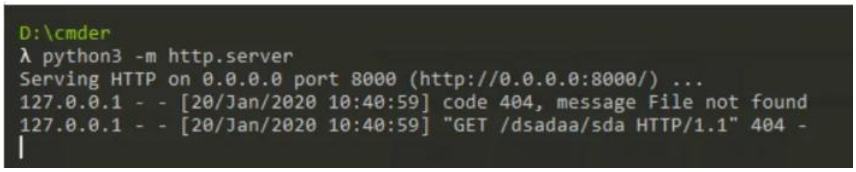


payload

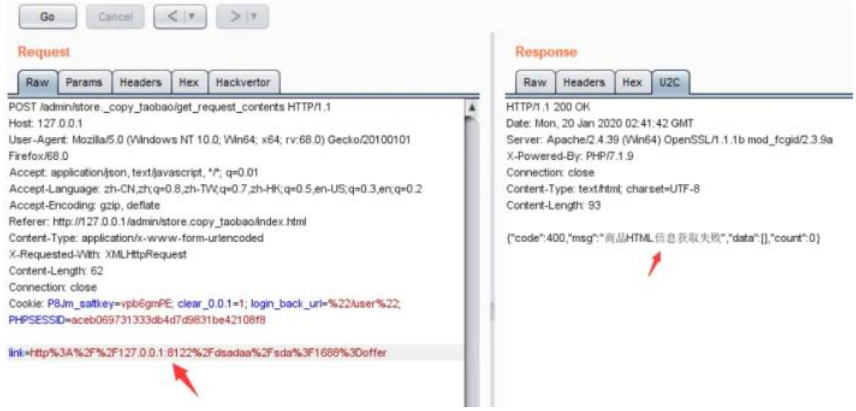
the payload for parameter `link` .

```
link=http%3A%2F%2F127.0.0.1:8000%2Fdsadaa%2Fsda%3F1688%3Doffer
```

and the python web server will receive the request.

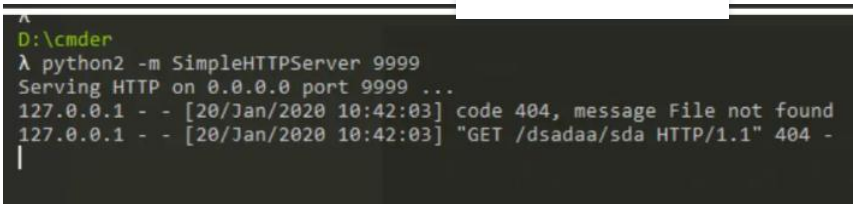


you can use this vulnerability scan the server open port with http(s) protocol.Information returned when scanning a closed port,such as `8122` port.



Information returned when scanning an open port.

also it can receive some request in port 9999



You can use this vulnerability to attack or scan the open port of the intranet server and collect information of other intranet servers.

Solution

Domain filtering should be performed first, followed by URL requests.

c0d1M4x created 任务 3 years ago

Expand operation logs

Sign in to comment