8

Server-Side request forgery in New-Subscription feature of the calendar app

Share:

TIMELINE

foobar7 submitted a report to Nextcloud. Oct 24th (4 ye

CVSS

8.5 High CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:L/A:N

Description

The "New Subscription" functionality of the official Calendar app allows authenticated users to direct the server to perform arbitrary external requests, and then displays the full response to the user. The requests can be directed to external websites as well as the internal network.

An attacker can use this vulnerability to exfiltrate data from the internal network that is not protected by other mechanisms, or to perform actions in the name of server in the internal network which do not contain other protection mechanisms (reduced security is common in internal networks).

An attacker can also use this issue to scan the internal network as well as external websites for further vulnerabilities.

A user account is required, but it does not require extended permissions.

POC

```
GET /nextcloud/nextcloud/index.php/apps/calendar/v1/proxy?url=http%3A%2F%2Flocalhost%2Fsecret HTTP/1.1
```

```
HTTP/1.1 200 OK
[...]
secret
```

The "secret" file is an example of a file located in the internal network that is not protected by further authentication. Its content is displayed in full and as-is (the s is true for more complex files).

Solution

The given URL should be verified before making a request. If it is pointing to a private address, the request should not be performed. To reduce the impact of poter misuse of the host as a scanner, requests could be throttled.

Additionally, the impact of the issue could be reduced by processing the response server-side instead of client-side. Instead of displaying the entire HTTP respon the client, the response could be parsed and validated to ensure that it is in a calendar format. This way, non-calendar/ics information from internal networks wou leak to an attacker.

Impact

exfiltrate data from the internal network and perform actions in the name of the server in the internal network

BOT: posted a comment. Oct 24th (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 to you to not disclose this issue to any other party.

rullzer posted a comment. Oct 24th (4 ye

Hi @foobar7,

Good catch. I'm in contact with out calendar engineers to see how we can fix this.
There is already work on the way to cache calendars server side: https://github.com/nextcloud/server/pull/10059/files

A possible quick fix (that might do the trick). If check the actual response type of the proxy request.
Also filtering out localhost as server would make sense.

We'll look into this and get back to you.

Cheers,
--Roeland

nickvergessen Nextcloud staff changed the status to Triaged. Oct 24th (4 ye

foobar7 posted a comment. Oct 24th (4 ye

Hi @rullzer,

thanks for the very quick response!

Checking the response type sounds like a good solution to severely reduce the impact. And filtering out private network ranges is a good additional defense if you use-cases allow for it (just the type filter would still allow (port-)scans of internal networks, stealing of internal calendars, and some other (less likely) attacks).

Best,
Foobar7

foobar7 posted a comment. Updated Nov 28th (4 ye

Hi @rullzer,

Foobar7

PS: I forgot to mention this in my initial report, but owncloud is affected by this as well. You might want to coordinate the release of a fix with them or hold back det information in the release notes.

foobar7 posted a comment. Nov 29th (4 ye

fyi: They had trouble reproducing this over at owncloud because of a failing CSRF check. I rechecked it at nextcloud and could confirm that simply visiting the URL in the initial report doesn't seem to work.

I'm glad that you were still able to reproduce it, but for completeness sake: You can initiate the correct request - including a CSRF token - in the web interface:

- Open the calendar -> click on "New Subscription"
- Enter the URL -> click on "Create"
- This will lead to an error message in the web interface, but the request will have been sent, and a response will be received. You can read the response using an interception proxy (eg Burp).

You can alternatively use the attached python script which sends the correct request (including CSRF token) and returns the contents of the internal file.

1 attachment:
F383116: nextcloud_ssrf.py

rullzer posted a comment. Dec 4th (4 ye

Hi,

Sorry for the delay. We had a company hackweek and are very busy with the upcomming releases. I'll check on your other tickets as well.

This should be fixed with the latest calendar release (1.6.4). Could you verify that?

Cheers,
--Roeland

foobar7 posted a comment. Dec 6th (4 ye

Hi @rullzer ,

thanks for getting back to me on this!

I can confirm that the fix severely reduces the impact of the issue. I couldn't bypass it to read any files that do not start with `BEGIN:VCALENDAR` or `BEGIN:VCARD` (an otherwise parse properly; the extension of the file doesn't matter, but it's highly unlikely that non-vcalendar/vcard files would start with these strings).

It is still possible to perform a portscan of the internal network and to perform other requests to the internal network from the given server (see eg here) though. / ICS files could of course still be taken from the internal network.

Because of this, I would still suggest to additionally filter out private network ranges - and optionally provide a configuration option (with a security notice in the documentation) to enable it if absolutely required.

Best,
Foobar7

rullzer posted a comment. Dec 18th (4 ye

Hi @foobar7,

Sorry for the delay here. I was traveling with limited internet access.
I'll talk with our dav/calendar developers to discuss this further.

Cheers,
--Roeland

rullzer posted a comment. Jan 10th (4 ye

Hi,

So the background job already does this internal network test:
https://github.com/nextcloud/server/blob/master/apps/dav/lib/BackgroundJob/RefreshWebcalJob.php#L227L239

But lets see if we can add it to calendar directly as well

Cheers,
--Roeland

foobar7 posted a comment. Updated Jan 10th (4 ye

Hi @rullzer,

reusing existing code sounds good to me :)

The code may need some adapting though, I found at least two bypasses.

`http://[::]/` is accepted by the filter, and will be retrieved as valid localhost URL. So with the above exploit and a secret.ics file hosted at localhost, the following will retrieve the file even with the filter in place:

python nextcloud_ssrf.py http://192.168.0.104/nextcloud/nextcloud/ admin [pass] http://[::]/secret.ics

Casing can also be used to bypass the filter and eg `http://LocalHost/` will be retrieved by the calendar app:

python nextcloud_ssrf.py http://192.168.0.104/nextcloud/nextcloud/ admin [pass] http://LocalHost/secret.ics foobar7 posted a comment. Feb 19th (4 ye

Hi @rullzer,

any update on this?

Best,
Foobar7

foobar7 posted a comment. Mar 19th (4 ye

Hi @rullzer,

I set a preliminary disclosure date for **April the 23rd** for the same issue in ownCloud.

As the issues can be exploited using the same POC, I wanted to let you know in case you are planning to add the internal network check from the background job t calendar as well.

If you need a bit more time to release a full fix, please let me know an approximate timeline and I'll postpone the disclosure accordingly.

Best,
Foobar7

foobar7 posted a comment. Apr 17th (4 ye

Hi @rullzer, hi @nickvergessen,

I'd like to remind you of the public disclosure date next week, April the 23rd.

If you have an approximate timeline for the development of the remainder of the fix, please let me know before then and I'll postpone the disclosure accordingly.

Best,
Foobar7

foobar7 posted a comment. Updated Apr 24th (4 ye

Hi @rullzer ,

the owncloud team asked for an extension, so I'm postponing the disclosure. I'll keep you posted.

It would still be good for me to know if you are planning on further working on this issue (in which case an ETA would be appreciated), or if you want to keep the sta is (in which case I would not object to marking the report as resolved, but would ask you to hold off on disclosure for now). If you are not sure yet, an ETA on when y will make a decision would also be appreciated.

Best,
Foobar7

georgehrke posted a comment. Apr 24th (4 ye

Hi @foobar7,

I'm sorry for the late response and the rather suboptimal communication for this issue.
We backported the fix from the Nextcloud server to the Calendar app.
It is available on the master and stable1.6 branches:

- master: https://github.com/nextcloud/calendar/commit/0cdb36e7b3c726bc0c758f6501becb319af1c3ca
- stable1.6: https://github.com/nextcloud/calendar/commit/f6d5acf744558e1c309a41da02805bc3ad5f64bb

Both have been released in calendar versions 1.7.0 (Nextcloud 16) and 1.6.5 (Nextcloud 14/15) respectively.
This covers all Nextcloud versions currently supported.

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)

foobar7 posted a comment. Updated Apr 24th (4 ye

Hi @georg_xa7pd,

thanks for getting back to me on this!

The fix looks mostly OK to me. You still have the problem that the filter can be bypassed (see this comment), but I would consider that to be a separate issue as it's affecting code taken from core (though it now also affects the calendar app).

Do you want me to open a separate report for this so that you can keep track of it?

I'm not sure if you coordinate disclosure with owncloud, but if possible, I'd like to give them a heads-up about your advisory.

Would it be possible to hold off on it to give them time to fix the issue as well? If so, do you want to set a specific date?

If you want to fix the filter bypass, it might also be a good idea to hold off on the advisory until that is fixed as well.

You can credit me as:

Name: Tim Coen
Website: https://security-consulting.icu/blog/

foobar7 posted a comment. Oct 31st (3 ye

Hi @georg_xa7pd,

any update on this? As the main issue has been fixed, I think keeping this report open seems unnecessary.

If you want to keep track of the broader issue of the SSRF filter bypass, I'm happy to open a new report, just let me know.

Best,
Foobar7

nickvergessen Nextcloud staff closed the report and changed the status to **Resolved**. Nov 12th (3 ye

Thanks a lot for your report again. This has been resolved in all supported releases back in July. If you can still bypass something please feel free to open a new issu

We are going to credit you with the following information:

- Name: Tim Coen
- Website: https://security-consulting.icu/blog/

nickvergessen Nextcloud staff updated the severity from High to Medium (5.0). Nov 12th (3 ye

nickvergessen Nextcloud staff changed the report title from **Calendar App: SSRF** to **Server-Side request forgery in New-Subscription feature of the calendar app**. Nov 12th (3 ye

Nextcloud rewarded foobar7 with a **$100** bounty. Nov 13th (3 ye

nickvergessen Nextcloud staff requested to disclose this report. Nov 13th (3 ye

foobar7 agreed to disclose this report. Dec 12th (3 ye

This report has been disclosed. Dec 12th (3 ye