

3 CVE-2021-22922: Wrong content via metalink not discarded

Share:     

TIMELINE

 nyymi submitted a report to curl. May 30th (2 ye

Summary:

When compiled `--with-libmetalink` and used with `--metalink` curl does check the cryptographic hash of the downloaded files. However, the only indication the hash was incorrect is a message displayed to the user. The files with incorrect hashes are left to the disk as-is.

Since curl implements the hash validation and reports incorrect hashes there might be an expectation that files with incorrect hashes would not be kept either. Since the metalink can be used with insecure protocols such as http and ftp, the hash validation might be used as an actual way to verify the download integrity against tampering.

Steps To Reproduce:

1. Configure libcurl `--with-libmetalink` and build libcurl
2. Have metalinktest.xml with `<file name="testfile">` containing incorrect sha-256 hash for it.
3. Execute: `curl --metalink https://testsite/metalinktest.xml`

The following message will be displayed:

```
Metalink: validating (testfile) [sha-256] FAILED (digest mismatch)
```

Yet, the downloaded file `testfile` with incorrect hash mismatch is kept.

Fix

It might be more sensible to download the file to a temporary name first, verify the hash and only then store the file to final name if the hash is correct. If hash mismatch is found remove the temporary file.

Impact

Modified or tampered files are kept and possibly incorrectly assumed valid

 nyymi posted a comment. May 30th (2 ye

Ok, also having the digest missing altogether leads to the file being kept regardless of the following message being displayed:

```
Metalink: validating (testfile) FAILED (digest missing)
```

 bagder curl staff posted a comment. May 30th (2 ye

Thank you for your report!

We will take some time and investigate your reports and get back to you with details and possible follow-up questions as soon as we can!

 dugustafsson curl staff posted a comment. May 31st (2 ye

I might be missing something but from reading the RFC and the documents on metalinker.org it doesn't seem entirely clear how hash validation failures should be handled. The RFC states that the file must be rejected, but the interpretation of "rejected" is implementation specific. Do you have any references to anything more specific?

Removing the file might be the sane option, but let's first settle on what the required behavior is (if any).


 bagder curl staff posted a comment. Updated Jun 10th (2 ye

"Rejected" *could* possibly be considered to be fulfilled because curl says the file is a mismatch but it seems a little farfetched. I think leaving files around that doesn't match the hash is a potential security problem.

 bagder curl staff changed the status to Triaged. Jun 11th (2 ye

 bagder curl staff posted a comment. Jun 14th (2 ye

Did anyone spot if this bug existed all the time we supported metalink or did it slip in at some point afterward?

 bagder curl staff posted a comment. Jun 18th (about 1 y

First take on advisory:

Wrong content via metalink not discarded

Project curl Security Advisory, July 21th 2021 -

[Permalink](#)

VULNERABILITY

When curl is instructed to download content using the metalink feature, the contents is verified against a hash provided in the metalink XML file.

The metalink XML file points out to the client how to get the same content from a set of different URLs, potentially hosted by different servers and the client can then download the file from one or several of them. In a serial or parallel manner.

download. It should remove the contents and instead try getting the contents from another URL. This is not done, and instead such a hash mismatch is only mentioned in text and the potentially malicious content is kept in the file on disk.

There's a risk the user doesn't notice the message and instead assumes the file is fine.

We are not aware of any exploit of this flaw.

INFO

This flaw exists only in the curl tool. libcurl is not affected.

This flaw has existed in curl since commit [b5fde848bc3d](#) in curl 7.27.0, released on July 27, 2012.

The Common Vulnerabilities and Exposures (CVE) project has assigned the name CVE-2021-FFFFF to this issue.

[CWE-20](#): Improper Input Validation

Severity: Medium

AFFECTED VERSIONS

- Affected versions: curl 7.27.0 to and including 7.77.0
- Not affected versions: curl < 7.27.0 and curl >= 7.78.0

Also note that libcurl is used by many applications, and not always advertised as such.

THE SOLUTION

curl has completely removed the metalink feature as of 7.78.0

The fix for earlier versions is to rebuild curl with the metalink support switched off!

RECOMMENDATIONS

A - Upgrade curl to version 7.78.0

B - Make sure you do not use metalink with curl

C - Disable metalink in your build

TIMELINE

This issue was reported to the curl project on May 30, 2021.

This advisory was posted on Jul 21, 2021.

CREDITS

This issue was reported by Harry Sintonen. Patched by Daniel Stenberg.

Thanks a lot!

1 attachment:

[F1343735: CVE-2021-FFFFF.md](#)



nyymi posted a comment.

Jun 18th (about 1 y)

Looking ok, however as mentioned in the other issue in regards of metalink "Also note that libcurl is used by many applications, and not always advertised as such. could be omitted in this advisory as the issue only applies to curl command itself



bagder curl staff posted a comment.

Jun 18th (about 1 y)

Ah yes, thanks. I updated my local version.



bagder curl staff updated CVE reference to [CVE-2021-22922](#).

Jun 28th (about 1 y)



Jun 28th (about 1 year ago)

bagder curl staff changed the report title from metalink hash verification failure doesn't lead the downloaded file being rejected to CVE-2021-22922: Wrong content via metalink not disca



curl rewarded nyymi with a \$700 bounty.

Jun 30th (about 1 y)

The curl security team has decided to reward hacker @nyymi with the amount of 700 USD for finding and reporting this issue. Many thanks for your great work!



bagder curl staff closed the report and changed the status to **Resolved**.

Jul 21st (about 1 y)



bagder curl staff requested to disclose this report.

Jul 21st (about 1 y)



nyymi agreed to disclose this report.

Jul 21st (about 1 y)



This report has been disclosed.

Jul 21st (about 1 y)

