NeilB

A blog about the Perl programming language

Addressing CPAN vulnerabilities related to checksums

By **Neil Bowers** on November 23, 2021 6:00 PM

This blog post addresses checksum and signature verification vulnerabilities affecting CPAN, the cpan client, and the cpanm client, which were published in a security advisory on 23rd November 2021. If you're not aware of this topic, you might like to start by reading the advisory. This post gives a high-level description of the issues, what has been done to address them, what is still left to do, and what you should do. If you have any questions on this, you can add comments here, or email the PAUSE admins (modules at perl dot org).

Before we dig into the details, we'll first give an overview of how the relevant parts of the CPAN ecosystem work.

If you're not interested in the details, skip to the section "What do you need to do?"

TL;DR: make sure your CPAN client uses https and a trusted mirror – such as cpan.org

How a CPAN module gets from the author to you

An author creates a new module and wants to share it on CPAN. They build a release, and upload the resulting tarball to PAUSE, where it goes into their author directory (my PAUSE id is NEILB and my author directory can be found on CPAN at https://www.cpan.org/authors/id/N/NE/NEILB). If the uploader has the relevant indexing permissions, PAUSE will update the CPAN Index. The Index is a big file which lists all modules on CPAN; here's the entry for the "enum" module, which I currently maintain:

```
enum     1.12      N/NE/NEILB/enum-1.12.tar.gz
```

It says "if you want enum, the latest version is 1.12 and it's found in the tarball enum-1.12.tar.gz at this path in the authors directory."

PAUSE also generates checksums for all files in author directories. Every directory has a file called CHECKSUMS (here's an example CHECKSUMS file for user GLASSER. After generating a new CHECKSUMS file, PAUSE signs it with GnuPG.

If you want to install a module, you can download the tarball and install it manually, but most people use a *CPAN client*. There are four main CPAN clients (cpan, cpanm, cpm, and cpanp) but we'll describe use of cpan, which is the client that's bundled with perl itself. To reduce confusion, we'll use "CPAN.pm" to refer to this client (that's the module that the cpan script uses).

To install the module, you run:

```
$ cpan Foo::Bar
```

The client makes sure it has a recent copy of the index, and looks for the module. If it's in the index, the client looks up the associated tarball and downloads it, along with the CHECKSUMS file that's in the author's directory on CPAN. The client generates a checksum for the downloaded file and compares it with the one in the CHECKSUMS file, and if they match, it goes ahead and installs it.

By default CPAN.pm doesn't check that the signature on the file is valid, but you can optionally enable that check as well. Since it's not enabled by default, the vast majority of us haven't been checking for a valid signature.

If you want to know more, you could start with a blog post on how PAUSE and CPAN work, and read a CPAN glossary which defines the terminology.

What issues did the advisory identify?

There were three issues identified, each of which has a separate CVE. This is a summary here; read the original advisory for the full story. Here are the CVEs: CVE-2020-16154, CVE-2020-16155, and CVE-2020-16156.

All three issues involve someone setting up a malicious CPAN mirror, and getting people to start using it.

The first issue is that the CHECKSUMS file only included the name of the tarball. For example, here's a snippet from the CHECKSUMS file in my author directory, as they appeared before the changes described below:

```
'Date-QuarterOfYear-0.04.tar.gz' => {
    'md5' => '8dd1a29e60af035a652aee121ab0919d',
    'md5-ungz' => '85ba26240f1396a147dcc09413a97db4',
    'mtime' => '2021-06-14',
    'sha256' => '1d13f55e5a1e84a7b05ce8cfd1d4cf9d9da80addd8f5f9fc845d2973daa0
    'sha256-ungz' => 'b4808c921b994656498ed74fbbba0677ff88d9624dabeabc4ef4f97
    'size' => 10763
},
```

There's no mention of the path there. Someone could create a malicious version of my module and upload a file with the same name to their PAUSE where it winds up in their author directory. Then they could upload copies of everything else in my author directory to PAUSE, so that their author directory looks the same as mine, except one file has been replaced with their malicious copy. Their author directory will have a CHECKSUMS file which looks the same as the one in my author directory, apart from different hashes for the one

About Neil Bowers

Perl hacker since 1992.

More info »

Search this blog

modified file. Plus it's been signed by PAUSE. They can set up their own CPAN mirror that swaps their author directory in place of mine, and if anyone downloads this tarball, even with the signing of CHECKSUMS validated, they get the malicious code but it looks like it’s coming from me.

The other two CVEs are for an issue that affects both CPAN.pm and cpanm. These again relate to someone setting up a malicious CPAN mirror. They can generate checksums for a directory and prepend an unsigned version of the checksum data to the CHECKSUMS file. Both CPAN.pm and cpanm will take the unsigned checksum data, ignoring the signed version that's also in the same CHECKSUMS file. It turned out that if you enabled the signature verification checks in CPAN.pm and cpanm, if an unsigned version of the CHECKSUM data has been injected above the signed data, then the clients would happily install the tarball for you anyway.

Addressing the issues

The first line of protection is to not use untrusted mirrors. In the early days of CPAN, when the internet was slower and more flakey, most of us used a local mirror, so a network of hundreds of CPAN mirrors sprang up. In 2021 they are largely irrelevant. Since 2019, the mirror list was truncated to just www.cpan.org. Most CPAN clients already default to this or to cpan.metacpan.org (which we also consider a trustworthy source).

If an attacker controls your network traffic, then they could inject malicious tarballs and checksums as well. Using https is the easiest thing you can do here.

CPAN.pm has defaulted to use only cpan.org since 2013 and with the truncation of the mirror list in 2019, any newly configured CPAN.pm clients are not vulnerable to a malicious mirror. The latest release of CPAN.pm will use https if at all possible.

PAUSE has been updated so that the CHECKSUMS files now include `cpan_path` for each file, which gives the relative path to the directory for which the checksums were generated. CPAN.pm will fail to validate if the paths don’t match so a signed CHECKSUMS file can’t be used from a different directory without detection.

CPAN.pm has been updated so that if you have configured it to validate the signature on CHECKSUMS, it will refuse to install a tarball if the associated CHECKSUMS file isn't signed.

The CPAN clients

In this section we'll summarise how the various CPAN clients work as of 2021-11-23, so you can make an informed decision on whether to continue using your current client, whether to upgrade, reconfigure, or to switch.

CPAN.pm / cpan 2.29

- It now ignores any previously configured urllist and only uses cpan.org. If you want it to honour the urllist parameter instead, you must set the new pushy_https parameter to false.
- It will try to use https, including use of curl or wget if the appropriate modules aren't installed. If none of those approaches work, it will fall back on http://www.cpan.org.
- It always looks at the CHECKSUMS file, and if you set `check_sigs` to true, it will require the CHECKSUMS file to have a valid signature.

cpanm 1.7044

- Defaults to using http://www.cpan.org (but can be configured to use https://www.cpan.org – see below).
- By default it doesn't look at the CHECKSUMS file.
- The `--verify` option tells it to validate the signature and content of the CHECKSUMS file.
- As of 2021-11-23, the latest version of cpanm (1.7044) doesn't address the issues raised, and so will install a distribution with unsigned CHECKSUMS content.

cpm 0.997007

- This is the most modern client, and as a result was already in good shape.
- By default cpm always uses https and downloads files from MetaCPAN's CPAN mirror, which we consider a trusted source. You can specify your own mirror in the command-line arguments, and even in your cpanfile.
- cpm's goal is speed, and as a result it doesn't look at the CHECKSUMS file, and there's no option to ask it to.

cpanp (CPANPLUS) 0.9914

- The default mirror is now www.cpan.org
- If the "md5" config option is set to "true", it will check that the checksums match. This defaults to true if Digest::SHA is available; this has been a core module since Perl 5.9.3. * There is no option to request verification of the signature.
- The default is to not use https, but if you set `prefer_bin` to true, and configure `hosts` for https, then it will use https://www.cpan.org as the source.
- A new version of CPANPLUS (0.9914) has been released which also checks the cpan_path field, and refuses to install if it doesn't match.

What do you need to do?

Use only https://www.cpan.org/ or https://cpan.metacpan.org/. Do not use any other mirror that you have not personally verified is trustworthy.

Here’s how to configure that in different CPAN clients:

- **CPAN.pm**: preferably upgrade to version 2.29, or configure your existing installation. In the CPAN shell, run the following command: "o conf urllist https://www.cpan.org", followed by "o conf commit" to save that change.
- **cpanm**: set the PERL_CPANM_OPT environment variable to ""--from https://www.cpan.org"

- **cpm**: it is already using https://cpan.metacpan.org, so you don't need to do anything. Check you haven't configured it to use a different mirror, and if you have, either update the config to explicitly use https://cpan.metacpan.org, or remove your configuration so it will use the default.
- **CPANPLUS**: set the `prefer_bin` option to true and make sure you've got curl installed. Then set the `hosts` parameter to `{ 'scheme' => 'https', 'path' => '/', 'host' => 'www.cpan.org' }`

If you insist on using an untrusted mirror, configure your CPAN client to verify CHECKSUMS. Here's how to do that in different CPAN clients:

- **CPAN.pm**: always verifies the checksums, but with the `check_sigs` option set to true ("o conf check_sigs true" then "o conf commit"), it will check the signature as well.
- **cpanm**: the --verify option tells it to verify the checksums, so add that to PERL_CPANM_OPT.
- **cpm**: doesn't verify the checksums, and there is currently no option to request that.
- **CPANPLUS**: as long as you're using Perl 5.10 or later, it will verify the checksums. There's no option for testing the signature.

What's left to do?

At the moment CPAN clients can't make https mandatory, because Perl doesn't support https out of the box. We've raised this on p5p, and this will hopefully lead to an RFC and support for https in the not-too-distant future.

Fixes are being worked on for cpanm, so unsigned CHECKSUMS will be fatal if you've given --verify, and also to support the cpan_path key in the CHECKSUMS files.

We're hoping that CPANPLUS will be updated to try and use https by default, as we've done with CPAN.pm.

Conclusion

The best way to ensure you're downloading the original versions of CPAN releases is to use a trusted CPAN mirror, and only talk to it over https.

If you are aware of any further potential security issues with PAUSE, or the CPAN clients, you can raise them via email to the PAUSE Admins private email list: pause-admin at perl dot org.

Acknowledgements

Firstly, thank you to Stig Palmquist, who identified these issues, and who has been patient and helpful as they were being addressed.

As ever with Perl, this has been a team effort of volunteers. Thank you to the CPAN client authors and others, who have worked on changes, and in some cases are still working. Thank you to everyone who read drafts of this post, and improved it with their input.

13 comments

13 Comments

kid51 | November 23, 2021 8:40 PM | Reply

> CPAN.pm: preferably upgrade to version 2.29, or if configure your existing installation.

Something seems to be missing between "if" and "configure" in that sentence.

Oodler 577 | November 23, 2021 10:29 PM | Reply

Not sure if it's still the case but at one point cpan (or "perl -MCPAN -e shell") presented all known mirrors in alphanumeric sort ordering, making it trivial to get an untrusted mirror ranked "high" and likely to be selected by naming it appropriately. I always thought it was an accident waiting to happen. So if this is still the case, at least for CPAN.pm, there should be some randomization thrown in or a tier of "official" mirrors.

Toby Inkster | November 24, 2021 9:54 AM | Reply

A while back, I started providing PGP/GPG signatures for all my CPAN and BackPAN releases here.

Obviously, this doesn't help with people using the CPAN client to automatically download and install packages, but might be useful if anybody wants to manually ensure they have an untampered version of a particular package.

Tom Wyant | November 24, 2021 4:37 PM | Reply

Thank you very much for the update.

Is it adequate for users of Mini-CPAN to ensure that their `~/.minicpanrc` specifies `remote: https//www.cpan.org/` (or other trusted server)? Assuming that there is no other reason to mistrust the Mini-CPAN repository, of course.

Aristotle | November 25, 2021 1:48 AM | Reply

If the mirror is trustworthy and so is the connection to it, does verification of the PAUSE-signed `CHECKSUMS` serve any remaining purpose? What threat not already covered by the use of a trustworthy origin and channel can be averted by signature verification?

As far as I can reason it out, (corrected) signature verification is both useful and important only when using an untrusted mirror and/or untrusted connection (in which case verifying the signature as being signed by the PAUSE private key provides proof that neither the mirror nor any hop on the route has tampered with the bits).

Robert Rothenberg | November 26, 2021 9:22 AM | Reply

If a malicious person managed to get write access to the filesystem of a trusted mirror, then the PAUSE-signed CHECKSUMS add a safety check to ensure that the files were not tampered with.

Robert Rothenberg | November 26, 2021 9:29 AM | Reply

A related aside: it would be nice to have CPAN authors register GnuPG keys with PAUSE, and an option to require their distributions be signed by their keys.

Any change to an author's keys or uploads that aren't signed could be flagged.

The various CPAN tools could be modified to check a distribution against the author's keys.

Aristotle replied to comment from Robert Rothenberg | November 26, 2021 10:52 AM | Reply

> If a malicious person managed to get write access to the filesystem of a trusted mirror

Indeed – and I had thought of that but dismissed it: I reasoned it would require the filesystem of either of the two trusted mirrors to be exposed in a way that that allows to gain write access to it without gaining shell on the mirror – since (I reasoned) shell access would ultimately expose the private key one way or another, anyway.

But as I went to write down my reasoning for this reply I realised my mistake – the PAUSE private key exists on PAUSE but not the trusted mirrors. The mirrors cannot sign the `CHECKSUMS`.

So the answer to my question is that verifying the signature protects against compromise of a mirror in general, because it proves that the checksums haven't been tampered after leaving PAUSE.

The upshot is that "make sure your CPAN client uses https and a trusted mirror" is not a substitute for (properly patched) signature verification, and in fact is unnecessary given (properly patched) mandatory signature verification. It is merely a partial mitigation for absent or vulnerable signature verification.

Aristotle replied to comment from Robert Rothenberg | November 26, 2021 11:03 AM | Reply

Yes – if authors signed their distributions themselves, this would verify the origin of the bits all the way to the source, rather than just up to PAUSE, which would be a worthwhile increase in trust. The only problem is the usual web of trust question: if the point is not to have to trust PAUSE then you can't source authors' keys from PAUSE, so where do you get them?

Robert Rothenberg replied to comment from Aristotle | November 26, 2021 12:05 PM | Reply

PAUSE signatures means that you trust that this is what was uploaded to PAUSE. But it's possible that a malicious person stole an author's credentials to upload something.

Author signatures means that you trust that the author has approved this code.

There's always the possibility that a malicious person has stolen PAUSE credentials *and* an author's key-signing credentials. It's not foolproof.

As an added safety, we could add a scheme for multiple signatures to be added. So another person can review code and submit their signature to PAUSE somehow.

Neil Bowers replied to comment from kid51 | November 26, 2021 12:35 PM | Reply

Thanks Jim - now fixed.

Neil Bowers replied to comment from Aristotle | November 26, 2021 12:38 PM | Reply

> If the mirror is trustworthy and so is the connection to it, does verification of the PAUSE-signed CHECKSUMS serve any remaining purpose?

Marginal benefit, I'd say. It's an additional check that you're getting the expected file.

I've heard anecdotally that the checksums once identified a case where an rsync had been interrupted and result in a truncated file.

nhorne | January 1, 2022 10:48 PM | Reply

I have a local mirror which downloads from https://cpan.org, so it's trusted. I then mount the mirror using NFS, so the entry in MyConfig.pm for urllist starts with "`file://foo/bar`". Even though I know it's trusted I still get:

```
Warning: checksum file '/mnt/CPAN/authors/id/G/GB/GBARR/CHECKSUMS'
not conforming.

The cksum does not contain the key 'cpan_path' for 'CPAN-
DistnameInfo-0.12.tar.gz'.
Proceed nonetheless? [no]
```

How can I handle this scenario?

Leave a comment

Sign in to comment.