Date: Tue, 29 Nov 2022 13:22:58 +0100
From: Julien Pivotto <roidelapluie@...metheus.io>
To: oss-security@...ts.openwall.com
Subject: CVE-2022-46146 in Prometheus' exporter toolkit: bypass basic
 authentication

Hello everyone,

The exporter toolkit is a go library intended at Prometheus exporters.
It provides some features that are useful for Prometheus exporters,
which work by exposing HTTP servers to be exposed by the Prometheus
server.

One of those features is basic authentication. To achieve this,
Prometheus requires you to store a bcrypt hash into a file, web.yml.

While bcrypt is fine, it takes by design a lot of time and resources to
compare a password with a hash. To limit this impact, we have a built-in
cache that caches the good and bad answers.

Once a request comes, we check it against the cache and decide whether
to allow the request. We also check that the user is valid. However, the
key for that cache is predictable:

hex(username + hashed password + input password)

If you know the bcrypted password, you can poison the cache and use that
cached positive value in a subsequent query:

Request 1:

username = username+hashed password
password = "fakepassword"

Request 2:

username = username
password = bcrypt(fakepassword)+"fakepassword"

"fakepassword" is used as bcrypted password when a user does not exist.

The fact that we save unhappy tentatives and that we validate
non-existing users against "fakepassword" is to prevent side channel
attacks that could reveal if a user exists in a system or not.

Prometheus 2.37.4 and 2.40.4 are out, with this fix. We recommend all
the exporters that depend on the repository to upgrade.

CVE-2022-46146 was assigned to this security report in our exporter
toolkit:
https://github.com/prometheus/exporter-toolkit/security/advisories/GHSA-7rg2-cxvp-9p7p

We would like to thank Lei Wan for the responsible disclosure of this
bug.

Best regards,

--
Julien Pivotto
@roidelapluie