access private registry: x509: certificate signed by unknown authority #8849

New issue

Closed **hustcat** opened this issue on Oct 30, 2014 · 19 comments

hustcat commented on Oct 30, 2014

I setup docker-registry with nginx by following here.

I run 'docker login', get this error:

```
# docker login -u docker -p docker -e xx@xxx.com https://dev.registry.com
2014/10/30 11:12:08 Error response from daemon: Server Error: Post https://dev.registry.com
```

docker daemon's output:

```
[debug] server.go:1181 Calling POST /auth
[info] POST /v1.15/auth
[47687bb1] +job auth()
[debug] endpoint.go:109 Error unmarshalling the _ping RegistryInfo: json: cannot unmarshal
[debug] endpoint.go:113 Registry version header: '0.7.1'
[debug] endpoint.go:116 RegistryInfo.Version: "0.7.1"
[debug] endpoint.go:119 Registry standalone header: 'True'
[debug] endpoint.go:127 RegistryInfo.Standalone: true
[debug] endpoint.go:109 Error unmarshalling the _ping RegistryInfo: json: cannot unmarshal
[debug] endpoint.go:113 Registry version header: '0.7.1'
[debug] endpoint.go:116 RegistryInfo.Version: "0.7.1"
[debug] endpoint.go:119 Registry standalone header: 'True'
[debug] endpoint.go:127 RegistryInfo.Standalone: true
Server Error: Post https://dev.registry.com/v1/users/: x509: certificate signed by unknown
[47687bb1] -job auth() = ERR (1)
[error] server.go:1207 Handler for POST /auth returned error: Server Error: Post https://de
[error] server.go:110 HTTP Error: statusCode=500 Server Error: Post https://dev.registry.co
```

I checked the code. I think function Login may be need 'tlsConfig'
https://github.com/docker/docker/blob/master/registry/auth.go#L163

just like
https://github.com/docker/docker/blob/master/registry/registry.go#L49

```
# docker --version
Docker version 1.3.0, build c78088f



# curl --cacert ca.pem https://dev.registry.com/v1/_ping
true
# curl --cacert ca.pem -u docker:docker https://dev.registry.com/v1/users/
"OK"

# curl -u docker:docker https://dev.registry.com/v1/users/
curl: (60) Peer certificate cannot be authenticated with known CA certificates
More details here: http://curl.haxx.se/docs/sslcerts.html

curl performs SSL certificate verification by default, using a "bundle"
 of Certificate Authority (CA) public keys (CA certs). If the default
 bundle file isn't adequate, you can specify an alternate file
 using the --cacert option.
If this HTTPS server uses a certificate signed by a CA represented in
 the bundle, the certificate verification probably failed due to a
 problem with the certificate (it might be expired, or the name might
 not match the domain name in the URL).
If you'd like to turn off curl's verification of the certificate, use
 the -k (or --insecure) option.
```

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Subscribe

You're not receiving notifications from this thread.

16 participants

tiborvass commented on Nov 4, 2014 Contributor

@hustcat As of Docker 1.3.1, you can do `--insecure-registry dev.registry.com:5000` you can replace 5000 with whichever port your registry is listening on.

I'm closing this now, but let us know in the comments if this did not solve your issue.

tiborvass closed this on Nov 4, 2014

behemphi commented on Nov 5, 2014

I am leaving this here b/c it took me a few minutes to figure it out, and might save someone the time. The command would be:

```
%> docker --insecure-registry=docker-registry.example.com:8080 login https://docker-registry.example.com:8080
```

Thanks for getting the switch put in place for 1.3!

7

rhasselbaum commented on Jan 20, 2015

I am facing the same problem. The certificate validation works for the ping (and pushing/pulling), but not login.

The `--insecure-registry` flag is a workaround, not a fix. The certificate validation should work if the CA certificate is loaded into `/etc/docker/certs.d/<registry>` , but it doesn't.

15

cdub50 commented on Jan 20, 2015

I cant event get it to work by setting --insecure-registry I am on docker 1.3.2 on RedHat 7

[root@ip-10-2-20-209 ec2-user]# docker --insecure-registry=qa.docker.repo login https://qa.docker.repo
Username: qa
Password:
Email: qa@user.com
2015/01/19 14:26:40 Error response from daemon: Server Error: Post https://qa.docker.repo/v1/users/: x509: certificate signed by unknown authority

curl works fine when I use the generated ca.pem file.

curl --cacert /home/ec2-user/ca.pem -u qa:xxxxx https://qa.docker.repo/v1/users/
"OK"

mp88 commented on Jan 20, 2015

I'm having the same issue on docker version 1.3.2 and opensuse 13.1. I even tried to statically pass --cafile cacert.pem to every curl call (since I assumed docker internally just uses curl), however, this also did not help.

Any help would be much appreciated.

Thanks.
Mario

ghost commented on Jan 20, 2015

Before I found this issue, I opened #10150. They appear to be the same issue.

jeffutter commented on Jan 20, 2015

I seem to be having the same issue. Archlinux client 1.4.1 and the registry running from the official docker container. Anyone have any thoughts?

grimmy commented on Jan 20, 2015

If you've installed the cert globally (via ca-certificates) make sure you restart docker as it won't reload the global ssl certs. That said, mine still isn't working, but I ran into that at work :)

mp88 commented on Jan 20, 2015

Thank you grimmy, that did the trick on my end and it finally works. I did:

1. Get cacert.pem from http://curl.haxx.se/docs/caextract.html
2. Copy the cacert.pem file to /etc/pki/trust/anchors/
3. sudo update-ca-certificates
4. sudo systemctl docker stop
5. sudo systemctl docker start

mario

4

rhasselbaum commented on Jan 20, 2015

Thank you, that also worked for me. Equivalent steps on Ubuntu/Debian:

1. Copy CA cert to `/usr/local/share/ca-certificates` .
2. sudo update-ca-certificates
3. sudo service docker restart

There is still a bug here, though. The docs say to install the CA cert in `/etc/docker/certs.d/<registry>` , and clearly that isn't sufficient. In fact, after installing the certificate globally, I removed the one in `/etc/docker/certs.d` , restarted Docker, and it still worked.

8

Fandekasp referenced this issue on Jan 30, 2015

Misleading error message about certificates #10452 Closed

GaretJax commented on Jul 8, 2015 Contributor

+1 for reopening this, as **@rhasselbaum** mentioned

cjw296 commented on Sep 16, 2015

Has --insecure-registry gone away?

```
$ docker --version
Docker version 1.8.2, build 0a8c2e3

$ docker --insecure-registry
flag provided but not defined: --insecure-registry
See 'docker --help'.
```

What should we use now?

13

cdub50 commented on Sep 17, 2015

that goes in the docker config file you can check if its set by looking at
the docker process you should see the --insecure-registry flag

On Wed, Sep 16, 2015 at 3:01 AM, Chris Withers notifications@github.com
wrote:

> Has --insecure-registry gone away?
>
> $ docker --version
> Docker version 1.8.2, build `0a8c2e3`
>
> $ docker --insecure-registry
> flag provided but not defined: --insecure-registry
> See 'docker --help'.
>
> What should we use now?
>
> —
> Reply to this email directly or view it on GitHub
> #8849 (comment).

hchaithanya commented on Oct 9, 2015

I got the same error for docker pull command and I think the following should work.
Copy the SSL certificate which is the '.crt' file to the directory

sudo cp foo.crt /usr/share/ca-certificates/extra/foo.crt
Let Ubuntu add the '.crt' file's path relative to /usr/share/ca-certificates to /etc/ca-certificates.conf

sudo dpkg-reconfigure ca-certificates

2

jamiehannaford referenced this issue in **getcarina/getcarina.com** on Oct 9, 2015

Using RCS Private repositories #29 Closed

carloshpds commented on Nov 30, 2015

if your machine state is not important, so you can run `docker-machine rm <machine-name>` and create another one ;)

rezonant commented on Feb 16, 2016

If you use LetsEncrypt and you don't want to run anything without proper TLS, make sure to provide the full chain of the certificate including intermediates (ie REGISTRY_HTTP_TLS_CERTIFICATE=.../fullchain.pem) you may see green in Chrome while still getting this error from Docker.

Cheers!

10

iamqizhao referenced this issue in **grpc/grpc-go** on Jun 15, 2016

Conn.resetTransport failed to create client transport: connection error: desc = "transport: x509: certificate signed by unknown authority" with certificate generated by Let's encrypt #702 Closed

JazzDeben commented on Sep 16, 2016 • edited

On Ubuntu. If you experience error:

- x509: cannot validate certificate for [IP address or domain name] because it doesn't contain any IP SANs

On the Docker registry the certificate had to be compiled with the subjectAltName as described here:
https://docs.docker.com/engine/security/https/

Here is the code for convenience:
$ echo subjectAltName = IP:10.10.10.20,IP:127.0.0.1 > extfile.cnf
$ openssl x509 -req -days 365 -sha256 -in server.csr -CA ca.pem -CAkey ca-key.pem
-CAcreateserial -out server-cert.pem -extfile extfile.cnf

Note, I was able to check the subject alternative name is present in the certificate using the following command:
openssl x509 -in certificate.crt -text -noout

However, on Ubuntu 14 client (i.e. Docker Engine)
This error was followed suit by
x509: certificate signed by unknown authority

For people using Ubuntu 14.
The config file that is used for the Docker engine (that I want to use to connect to the Docker Registry):
/etc/default/docker

in there, you need to specify the docker options:
DOCKER_OPTS="--insecure-registry myinsecure.com:5000"

Then restart the daemon (add sudo if you user is not allowed to start a docker service):
$ [sudo] service docker restart

The value does not need to be a domain name, it simply has to match what you certificate is registered with; I have an IP address with a port and this works... (i.e. e.g. 100.100.100.100:100)

All this took me a day, so, I am posting this hoping that it will be useful to other people...

sallespro commented on Oct 5, 2016 • edited

@JazzDeben Thanks for your remarks ! very useful ! i am not sure how to do it with a Let's Encript certbot generated certificate.
i get this error in the registry server

```
tls: client didn't provide a certificate
```

Chrome complains about `ERR_BAD_SSL_CLIENT_AUTH_CERT`
if i include

```
  tls:
...
    clientcas:
      - /path/to/ca.pem
```

david-drinn commented on Oct 7, 2016 • edited

@cjw296 For RHEL7.2, I edited the file, `/usr/lib/systemd/docker.service` , and in the `ExecStart` line added the `--insecure-registry=your.docker.registry.com` .

```
< ExecStart=/usr/bin/dockerd
---
> ExecStart=/usr/bin/dockerd --insecure-registry=your.docker.registry.com
```

Then I ran `sudo systemctl daemon-reload` to pick up the configuration change, followed by `sudo systemctl restart docker` . And now it works.

To be honest, I'm still a systemd noob and there are probably better ways to do this more cleanly. But I struggled with this for too long, and wanted to post a workaround. Thanks to **@cdub50** for leading me in the right direction.

3

mrkz referenced this issue in **clearlinux/clear-config-management** on Oct 8, 2016

Error while downloading the clearlinux images for docker-hub #81 Closed

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Comment

Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About