Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'cockpit-tls.8' command

$ man cockpit-tls.8

COCKPIT-TLS(8) cockpit-tls COCKPIT-TLS(8)

NAME

cockpit-tls - TLS proxy for Cockpit web service

SYNOPSIS

cockpit-tls [--help] [--port PORT] [--no-tls] [--idle-timeout SECONDS]

DESCRIPTION

The cockpit-tls program is a TLS terminating HTTP proxy for cockpit-ws(8). It manages a set of isolated cockpit-ws instances, one per TLS client certificate, plus one for TLS without a client certificate, and one for unencrypted HTTP. With that, one session cannot tamper with another one through possible security vulnerability exploits.

Users or administrators should never need to start this program as it automatically started by systemd(1) via socket activation.

TRANSPORT SECURITY

To specify the TLS certificate the web service should use, simply drop a file with the extension .cert in the /etc/cockpit/ws-certs.d directory. If there are multiple files in this directory, then the highest priority one is chosen after sorting.

The .cert file should contain at least two OpenSSL style PEM blocks. First one or more BEGIN CERTIFICATE blocks for the server certificate and intermediate certificate authorities and a second one containing a BEGIN PRIVATE KEY or similar. The key must not be encrypted.

If there is no TLS certificate, a self-signed certificate is

automatically generated using sscg (if available) or openssl and stored in the 0-self-signed.cert file.

When enrolling into a FreeIPA domain, an SSL certificate is requested from the IPA server and stored in 10-ipa.cert.

To check which certificate cockpit-ws will use, run the following command.

```
$ sudo /usr/libexec/cockpit-certificate-ensure --check
```

Or, on Debian-based systems:

```
$ sudo /usr/lib/cockpit/cockpit-certificate-ensure --check
```

If using certmonger to manage certificates, following command can be used to generate a certificate/key pair:

```
CERT_FILE=/etc/cockpit/ws-certs.d/50-certmonger.crt
KEY_FILE=/etc/cockpit/ws-certs.d/50-certmonger.key
getcert request -f ${CERT_FILE} -k ${KEY_FILE} -D $(hostname --fqdn)
```

OPTIONS

--help

Show help options.

--port PORT

Serve HTTP requests on PORT instead of port 9090. Usually Cockpit is started on demand by systemd socket activation, and this option has no effect. Update the ListenStream directive cockpit.socket file in the usual systemd manner.

--no-tls

Don't use TLS. Certificates will not be read, and https connections denied. Then cockpit-tls will only manage a single cockpit-ws instance, and thus not do anything different than running cockpit-ws --no-tls directly. Only use this for debugging or testing.

--idle-timeout SECONDS

If greater than 0, exit if no connections have happened for the given number of seconds, i. e. the server is idle. If not given, the default is 90.

ENVIRONMENT

The cockpit-tls program expects the RUNTIME_DIRECTORY environment variable to be set to an empty directory (preferably in /run/) that is only accessible by the system user under which it is running. This contains the Unix sockets for communicating with the cockpit-ws instances, and in the future, state information about client certificates. This variable is normally set by the cockpit.service systemd unit.

In addition, cockpit-tls will use the XDG_CONFIG_DIRS environment variable from the XDG basedir spec[1] to find its certificates and the cockpit.conf(5) configuration file.

BUGS

Please send bug reports to either the distribution bug tracker or the upstream bug tracker[2].

AUTHOR

Cockpit has been written by many contributors[3].

SEE ALSO

cockpit-ws(8) , cockpit.conf(5) , systemd(1)

NOTES

1. XDG basedir spec
   https://specifications.freedesktop.org/basedir-spec/basedir-spec-latest.html
2. upstream bug tracker
   https://github.com/cockpit-project/cockpit/issues/new
3. contributors
   https://github.com/cockpit-project/cockpit/graphs/contributors