



***Rocky Enterprise Linux 9.2 Manual Pages on command 'cockpit\_ws\_selinux.&cockpit'***

***\$ man cockpit\_ws\_selinux.&cockpit***

cockpit\_ws\_selinux(8) SELinux Policy cockpit\_ws cockpit\_ws\_selinux(8)

NAME

cockpit\_ws\_selinux - Security Enhanced Linux Policy for the cockpit\_ws processes

DESCRIPTION

Security-Enhanced Linux secures the cockpit\_ws processes via flexible mandatory access control.

The cockpit\_ws processes execute with the cockpit\_ws\_t SELinux type.

You can check if you have these processes running by executing the ps command with the -Z qualifier.

For example:

```
ps -eZ | grep cockpit_ws_t
```

ENTRYPOINTS

The cockpit\_ws\_t SELinux type can be entered via the cockpit\_ws\_exec\_t file type.

The default entrypoint paths for the cockpit\_ws\_t domain are the following:

lowing:

```
/usr/libexec/cockpit-ws, /usr/libexec/cockpit-tls, /usr/share/cock?
```

pit/motd/update-motd, /usr/libexec/cockpit-wsinstance-factory

## PROCESS TYPES

SELinux defines process types (domains) for each process running on the system

You can see the context of a process using the `-Z` option to `ps`

Policy governs the access confined processes have to files. SELinux cockpit\_ws policy is very flexible allowing users to setup their cockpit\_ws processes in as secure a method as possible.

The following process types are defined for cockpit\_ws:

cockpit\_ws\_t

Note: `semanage permissive -a cockpit_ws_t` can be used to make the process type `cockpit_ws_t` permissive. SELinux does not deny access to permissive process types, but the AVC (SELinux denials) messages are still generated.

## BOOLEANS

SELinux policy is customizable based on least access required. cockpit\_ws policy is extremely flexible and has several booleans that allow you to manipulate the policy and run cockpit\_ws with the tightest access possible.

If you want to allow all domains to execute in `fips_mode`, you must turn on the `fips_mode` boolean. Enabled by default.

```
setsebool -P fips_mode 1
```

## MANAGED FILES

The SELinux process type `cockpit_ws_t` can manage files labeled with the following file types. The paths listed are the default paths for these file types. Note the processes UID still need to have DAC permissions.

cluster\_conf\_t

```
/etc/cluster(/.*)?
```

cluster\_var\_lib\_t

```
/var/lib/pcsd(/.*)?
```

```
/var/lib/cluster(/.*)?
```

```
/var/lib/openais(/.*)?
```

```
/var/lib/pengine(/.*)?
```

/var/lib/corosync(/.\*)?  
/usr/lib/heartbeat(/.\*)?  
/var/lib/heartbeat(/.\*)?  
/var/lib/pacemaker(/.\*)?

#### cluster\_var\_run\_t

/var/run/crm(/.\*)?  
/var/run/cman\_\*  
/var/run/rsctmp(/.\*)?  
/var/run/aisexec.\*  
/var/run/heartbeat(/.\*)?  
/var/run/pcsd-ruby.socket  
/var/run/corosync-qnetd(/.\*)?  
/var/run/corosync-qdevice(/.\*)?  
/var/run/corosync.pid  
/var/run/cpglockd.pid  
/var/run/rgmanager.pid  
/var/run/cluster/rgmanager.sk

#### cockpit\_var\_lib\_t

/var/lib/cockpit(/.\*)?

#### cockpit\_var\_run\_t

/var/run/cockpit(/.\*)?  
/var/run/cockpit-ws(/.\*)?

#### krb5\_keytab\_t

/var/kerberos/krb5(/.\*)?  
/etc/krb5.keytab  
/etc/krb5kdc/kadm5.keytab  
/var/kerberos/krb5kdc/kadm5.keytab

#### root\_t

/sysroot/ostree/deploy/\*-atomic/deploy(/.\*)?  
/  
/initrd

#### systemd\_passwd\_var\_run\_t

/var/run/systemd/ask-password(/.\*)?

`/var/run/systemd/ask-password-block(/.*)?`

## FILE CONTEXTS

SELinux requires files to have an extended attribute to define the file type.

You can see the context of a file using the `-Z` option to `ls`

Policy governs the access confined processes have to these files.

SELinux cockpit\_ws policy is very flexible allowing users to setup their cockpit\_ws processes in as secure a method as possible.

The following file types are defined for cockpit\_ws:

`cockpit_ws_exec_t`

- Set files with the `cockpit_ws_exec_t` type, if you want to transition an executable to the `cockpit_ws_t` domain.

Paths:

`/usr/libexec/cockpit-ws,`            `/usr/libexec/cockpit-tls,`  
`/usr/share/cockpit/motd/update-motd,` `/usr/libexec/cockpit-wsin?`  
`stance-factory`

Note: File context can be temporarily modified with the `chcon` command.

If you want to permanently change the file context you need to use the `semanage fcontext` command. This will modify the SELinux labeling data? base. You will need to use `restorecon` to apply the labels.

## COMMANDS

`semanage fcontext` can also be used to manipulate default file context mappings.

`semanage permissive` can also be used to manipulate whether or not a process type is permissive.

`semanage module` can also be used to enable/disable/install/remove policy modules.

`semanage boolean` can also be used to manipulate the booleans

`system-config-selinux` is a GUI tool available to customize SELinux policy settings.

## AUTHOR

This manual page was auto-generated using `sepolicy manpage`.

## SEE ALSO

selinux(8), cockpit\_ws(8), semanage(8), restorecon(8), chcon(1), sepol?

icy(8), setsebool(8)

cockpit\_ws

21-04-16

cockpit\_ws\_selinux(8)