NAME

selinux_restorecon − restore file(s) default SELinux security contexts

SYNOPSIS

#include <selinux/restorecon.h>

int selinux_restorecon(const char ****pathname*,**
unsigned int *restorecon_flags***);**

DESCRIPTION

selinux_restorecon() restores file default security contexts on filesystems that support extended attributes (see **xattr**(7)), based on:

pathname containing a directory or file to be relabeled.
If this is a directory and the *restorecon_flags* **SELINUX_RESTORECON_RECURSE** has been set (for descending through directories), then **selinux_restorecon**() will write an SHA1 digest of specfile entries calculated by **selabel_get_digests_all_partial_matches**(3) to an extended attribute of *security.sehash* once the relabeling has been completed successfully (see the **NOTES** section for details).
These digests will be checked should **selinux_restorecon**() be rerun with the *restorecon_flags* **SELINUX_RESTORECON_RECURSE** flag set. If any of the specfile entries had been updated, the digest will also be updated. However if the digest is the same, no relabeling checks will take place.
The *restorecon_flags* that can be used to manage the usage of the SHA1 digest are:
SELINUX_RESTORECON_SKIP_DIGEST
SELINUX_RESTORECON_IGNORE_DIGEST

restorecon_flags contains the labeling option/rules as follows:

SELINUX_RESTORECON_SKIP_DIGEST Do not check or update any extended attribute *security.sehash* entries.

SELINUX_RESTORECON_IGNORE_DIGEST force the checking of labels even if the stored SHA1 digest matches the specfile entries SHA1 digest. The specfile entries digest will be written to the *security.sehash* extended attribute once relabeling has been completed successfully provided the **SELINUX_RESTORECON_NOCHANGE** flag has not been set.

SELINUX_RESTORECON_NOCHANGE don't change any file labels (passive check) or update the digest in the *security.sehash* extended attribute.

SELINUX_RESTORECON_SET_SPECFILE_CTX If set, reset the files label to match the default specfile context. If not set only reset the files "type" component of the context to match the default specfile context.

SELINUX_RESTORECON_RECURSE change file and directory labels recursively (descend directories) and if successful write an SHA1 digest of the specfile entries to an extended attribute as described in the **NOTES** section.

SELINUX_RESTORECON_VERBOSE log file label changes.

> Note that if **SELINUX_RESTORECON_VERBOSE** and **SELINUX_RESTORECON_PROGRESS** flags are set, then **SELINUX_RESTORECON_PROGRESS** will take precedence.

SELINUX_RESTORECON_PROGRESS show progress by outputting the number of files in 1k blocks processed to stdout. If the **SELINUX_RESTORECON_MASS_RELABEL** flag is also set then the approximate percentage complete will be shown.

SELINUX_RESTORECON_MASS_RELABEL generally set when relabeling the entire OS, that will then show the approximate percentage complete. The **SELINUX_RESTORECON_PROGRESS** flag must also be set.

SELINUX_RESTORECON_REALPATH convert passed-in *pathname* to the canonical pathname using **realpath**(3).

SELINUX_RESTORECON_XDEV prevent descending into directories that have a different device number than the *pathname* entry from which the descent began.

SELINUX_RESTORECON_ADD_ASSOC attempt to add an association between an inode and a specification. If there is already an association for the inode and it conflicts with the specification, then use the last matching specification.

SELINUX_RESTORECON_ABORT_ON_ERROR abort on errors during the file tree walk.

SELINUX_RESTORECON_SYSLOG_CHANGES log any label changes to **syslog**(3).

SELINUX_RESTORECON_LOG_MATCHES log what specfile context matched each file.

SELINUX_RESTORECON_IGNORE_NOENTRY ignore files that do not exist.

SELINUX_RESTORECON_IGNORE_MOUNTS do not read **/proc/mounts** to obtain a list of non-seclabel mounts to be excluded from relabeling checks.
Setting **SELINUX_RESTORECON_IGNORE_MOUNTS** is useful where there is a non-seclabel fs mounted with a seclabel fs mounted on a directory below this.

SELINUX_RESTORECON_CONFLICT_ERROR to treat conflicting specifications, such as where two hardlinks for the same inode have different contexts, as errors.

The behavior regarding the checking and updating of the SHA1 digest described above is the default behavior. It is possible to change this by first calling **selabel_open**(3) and not enabling the **SELABEL_OPT_DIGEST** option, then calling **selinux_restorecon_set_sehandle**(3) to set the handle to be used by **selinux_restorecon**(3).

If the *pathname* is a directory path, then it is possible to set directories to be excluded from the path by calling **selinux_restorecon_set_exclude_list**(3) with a **NULL** terminated list before calling **selinux_restorecon**(3).

By default **selinux_restorecon**(3) reads **/proc/mounts** to obtain a list of non-seclabel mounts to be excluded from relabeling checks unless the **SELINUX_RESTORECON_IGNORE_MOUNTS** flag has been set.

RETURN VALUE

On success, zero is returned. On error, −1 is returned and *errno* is set appropriately.

NOTES

1. To improve performance when relabeling file systems recursively (e.g. the *restorecon_flags* **SELINUX_RESTORECON_RECURSE** flag is set) **selinux_restorecon**() will write a calculated SHA1 digest of the specfile entries returned by **selabel_get_digests_all_partial_matches**(3) to an

extended attribute named *security.sehash* for each directory in the *pathname* path.

2. To check the extended attribute entry use **getfattr**(1)**,** for example:

```
getfattr -e hex -n security.sehash /
```

3. Should any of the specfile entries have changed, then when **selinux_restorecon**() is run again with the **SELINUX_RESTORECON_RECURSE** flag set, new SHA1 digests will be calculated and all files automatically relabeled depending on the settings of the **SELINUX_RESTORECON_SET_SPECFILE_CTX** flag (provided **SELINUX_RESTORECON_NOCHANGE** is not set).

4. **/sys** and in-memory filesystems do not support the *security.sehash* extended attribute and are automatically excluded from any relabeling checks.

5. By default **stderr** is used to log output messages and errors. This may be changed by calling **selinux_set_callback**(3) with the **SELINUX_CB_LOG** *type* option.

SEE ALSO

selabel_get_digests_all_partial_matches(3),
selinux_restorecon_set_sehandle(3),
selinux_restorecon_default_handle(3),
selinux_restorecon_set_exclude_list(3),
selinux_restorecon_set_alt_rootpath(3),
selinux_restorecon_xattr(3),
selinux_set_callback(3)