

**NAME**

selinux\_file\_context\_verify – Compare the SELinux security context on disk to the default security context required by the policy file contexts file

**SYNOPSIS**

```
#include <selinux/selinux.h>
```

```
int selinux_file_context_verify(const char *path, mode_t mode);
```

**DESCRIPTION**

**selinux\_file\_context\_verify()** compares the context of the specified *path* that is held on disk (in the extended attribute), to the system default entry held in the file contexts series of files.

The *mode* may be zero.

Note that the two contexts are compared for "significant" differences (i.e. the user component of the contexts are ignored) as shown in the **EXAMPLE** section.

**RETURN VALUE**

If the contexts significantly match, 1 (one) is returned.

If the contexts do not match 0 (zero) is returned and *errno* is set to either **ENOENT** or **EINVAL** for the reasons listed in the **ERRORS** section, or if *errno* = 0 then the contexts did not match.

On failure -1 is returned and *errno* set appropriately.

**ERRORS****ENOTSUP**

if extended attributes are not supported by the file system.

**ENOENT**

if there is no entry in the file contexts series of files or *path* does not exist.

**EINVAL**

if the entry in the file contexts series of files or *path* are invalid, or the returned context fails validation.

**ENOMEM**

if attempt to allocate memory failed.

**FILES**

The following configuration files (the file contexts series of files) supporting the active policy will be used (should they exist) to determine the *path* default context:

*contexts/files/file\_contexts* - This file must exist.

*contexts/files/file\_contexts.local* - If exists has local customizations.

*contexts/files/file\_contexts.homedirs* - If exists has users home directory customizations.

*contexts/files/file\_contexts.subs* - If exists has substitutions that are then applied to the 'in memory' version of the file contexts files.

**EXAMPLE**

If the files context is:

```
unconfined_u:object_r:admin_home_t:s0
```

and the default context defined in the file contexts file is:

```
system_u:object_r:admin_home_t:s0
```

then the actual strings compared are:

:object\_r:admin\_home\_t:s0 and :object\_r:admin\_home\_t:s0

Therefore they will match and **selinux\_file\_context\_verify()** will return 1.

**SEE ALSO**

**selinux(8)**