

**NAME**

getkeycreatecon, setkeycreatecon – get or set the SELinux security context used for creating a new kernel keyrings

**SYNOPSIS**

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

```
int getkeycreatecon(char **con);
```

```
int getkeycreatecon_raw(char **con);
```

```
int setkeycreatecon(char *context);
```

```
int setkeycreatecon_raw(char *context);
```

**DESCRIPTION**

**getkeycreatecon()** retrieves the context used for creating a new kernel keyring. This returned context should be freed with **freecon(3)** if non-NULL. **getkeycreatecon()** sets *\*con* to NULL if no keycreate context has been explicitly set by the program (i.e. using the default policy behavior).

**setkeycreatecon()** sets the context used for creating a new kernel keyring. NULL can be passed to **setkeycreatecon()** to reset to the default policy behavior. The keycreate context is automatically reset after the next **execve(2)**, so a program doesn't need to explicitly sanitize it upon startup.

**setkeycreatecon()** can be applied prior to library functions that internally perform an file creation, in order to set an file context on the objects.

**getkeycreatecon\_raw()** and **setkeycreatecon\_raw()** behave identically to their non-raw counterparts but do not perform context translation.

**Note:** Signal handlers that perform a **setkeycreatecon()** must take care to save, reset, and restore the keycreate context to avoid unexpected behavior.

**Note:** Contexts are thread specific.

**RETURN VALUE**

On error -1 is returned. On success 0 is returned.

**SEE ALSO**

**selinux(8)**, **freecon(3)**, **getcon(3)**, **getexeccon(3)**