

NAME

context_new, context_str, context_free, context_type_get, context_type_set, context_range_get, context_range_set, context_role_get, context_role_set, context_user_get, context_user_set – Routines to manipulate SELinux security contexts

SYNOPSIS

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

```
context_t context_new(const char *context_str);
```

```
const char * context_str(context_t con);
```

```
void context_free(context_t con);
```

```
const char * context_type_get(context_t con);
```

```
const char * context_range_get(context_t con);
```

```
const char * context_role_get(context_t con);
```

```
const char * context_user_get(context_t con);
```

```
int context_type_set(context_t con, const char *type);
```

```
int context_range_set(context_t con, const char *range);
```

```
int context_role_set(context_t con, const char *role);
```

```
int context_user_set(context_t con, const char *user);
```

DESCRIPTION

These functions allow an application to manipulate the fields of a security context string without requiring it to know the format of the string.

context_new() returns a new context initialized to a context string.

context_str() returns a pointer to the string value of the **context_t**, valid until the next call to **context_str()** or **context_free()** for the same **context_t**.

context_free() frees the storage used by a context.

context_type_get(), **context_range_get()**, **context_role_get()**, **context_user_get()** get a pointer to the string value of a context component.

Note: Values returned by the get functions are only valid until the next call to a set function or **context_free()** for the same **context_t** structure.

context_type_set(), **context_range_set()**, **context_role_set()**, **context_user_set()** set a context component.

RETURN VALUE

On failure **context_*_set()** functions return non-zero and 0 on success.

The other functions return NULL on failure and non-NULL on success.

On failure *errno* is set appropriately.

SEE ALSO

selinux(8)