

NAME

runcon — alter and consume SELinux context

SYNOPSIS

```
runcon
runcon context program [argument]...
runcon [-u user] [-r role] [-t type] [-l range] program [argument]...
    (with at least one of -urtl)
runcon -c [-u user] [-r role] [-t type] [-l range] program [argument]...
```

DESCRIPTION

Without *program*, writes the current SELinux security context, followed by a newline, to the standard output stream.

Otherwise, executes *program arguments* with a different context, if valid:

with <i>context</i>	<i>context</i> ,
with at least one of -urtl	getcon() with the specified fields altered,
with -c	the context computed to label a new object, from getcon() to the filesystem context of <i>program</i> , in the "process" security class, with fields overridden as specified by -urtl , if any.

OPTIONS

-c, --compute	Pre-compute context transition, cf. DESCRIPTION .
-u, --user= <i>user</i>	Set user (the first field) to <i>user</i> .
-r, --role= <i>role</i>	Set role (the second field) to <i>role</i> .
-t, --type= <i>type</i>	Set type (the third field) to <i>type</i> .
-l, --range= <i>range</i>	Set level-range (the fourth field) to <i>range</i> .

ENVIRONMENT

PATH In which *program* is searched when requested (except with **-c**), confer `execvp(3)`.

EXIT STATUS

127	<i>program</i> wasn't found.
126	<i>program</i> exists, but couldn't be executed for a different reason.
125	an error occurred in runcon (SELinux is not active, the final context or a field is invalid, &c.).
All others	returned by <i>program</i> , if executed, or 0 .

SEE ALSO

`runcon(1)`, `getcon(3)`, `getfscon(3)`, `security_check_context(3)`, `setexeccon(3)`, `selinux(8)`

STANDARDS

Compatible with the GNU system, which exits **1** for set-up errors and executes *program* from the **PATH** even with **-c**, but uses it verbatim as a **getfscon()** argument — this means that **runcon -c true** will fail unless there's a `true` file in the current working directory and, if there is, *its* context will be used for the computation, rather than `/bin/true`'s. This simply begs for a trojan horse scenario. Cf. *Bug#1013924: coreutils: runcon -c getfscon()s program verbatim but execve()s it; trojan moment?: <https://bugs.debian.org/1013924>*.