NAME
pkcheck − Check whether a process is authorized

SYNOPSIS
pkcheck [−−**version**] [−−**help**]

pkcheck [−−**list−temp**]

pkcheck [−−**revoke−temp**]

pkcheck −−**action−id** *action* {−−**process** { *pid* | *pid,pid−start−time* | *pid,pid−start−time,uid* } |
−−**system−bus−name** *busname*} [−−**allow−user−interaction**] [−−**enable−internal−agent**]
[−−**detail** *key value*...]

DESCRIPTION
pkcheck is used to check whether a process, specified by either −−**process** (see below) or
−−**system−bus−name**, is authorized for *action*. The −−**detail** option can be used zero or more times to pass
details about *action*. If −−**allow−user−interaction** is passed, **pkcheck** blocks while waiting for
authentication.

The invocation **pkcheck** −−**list−temp** will list all temporary authorizations for the current session and
pkcheck −−**revoke−temp** will revoke all temporary authorizations for the current session.

This command is a simple wrapper around the PolicyKit D−Bus interface; see the D−Bus interface
documentation for details.

RETURN VALUE
If the specified process is authorized, **pkcheck** exits with a return value of 0. If the authorization result
contains any details, these are printed on standard output as key/value pairs using environment style
reporting, e.g. first the key followed by a an equal sign, then the value followed by a newline.

KEY1=VALUE1
KEY2=VALUE2
KEY3=VALUE3
...

Octects that are not in [a−zA−Z0−9_] are escaped using octal codes prefixed with \. For example, the
UTF−8 string *føl,* will be printed as *f\303\270l\54\344\275\240\345\245\275*.

If the specificied process is not authorized, **pkcheck** exits with a return value of 1 and a diagnostic message
is printed on standard error. Details are printed on standard output.

If the specificied process is not authorized because no suitable authentication agent is available or if the
−−**allow−user−interaction** wasn't passed, **pkcheck** exits with a return value of 2 and a diagnostic message
is printed on standard error. Details are printed on standard output.

If the specificied process is not authorized because the authentication dialog / request was dismissed by the
user, **pkcheck** exits with a return value of 3 and a diagnostic message is printed on standard error. Details
are printed on standard output.

If an error occured while checking for authorization, **pkcheck** exits with a return value of 127 with a
diagnostic message printed on standard error.

If one or more of the options passed are malformed, **pkcheck** exits with a return value of 126. If stdin is a
tty, then this manual page is also shown.

NOTES
Do not use either the bare *pid* or *pid,start−time* syntax forms for −−**process**. There are race conditions in
both. New code should always use *pid,pid−start−time,uid*. The value of *start−time* can be determined by
consulting e.g. the **proc**(5) file system depending on the operating system. If fewer than 3 arguments are
passed, **pkcheck** will attempt to look up them up internally, but note that this may be racy.

If your program is a daemon with e.g. a custom Unix domain socket, you should determine the *uid*

parameter via operating system mechanisms such as PEERCRED.

AUTHENTICATION AGENT

pkcheck, like any other PolicyKit application, will use the authentication agent registered for the process in question. However, if no authentication agent is available, then **pkcheck** can register its own textual authentication agent if the option **−−enable−internal−agent** is passed.

AUTHOR

Written by David Zeuthen <davidz@redhat.com> with a lot of help from many others.

BUGS

Please send bug reports to either the distribution or the polkit−devel mailing list, see the link **http://lists.freedesktop.org/mailman/listinfo/polkit-devel** on how to subscribe.

SEE ALSO

polkit(8), **pkaction**(1), **pkexec**(1), **pkttyagent**(1)