MyWebUniversity
python
Red Hat Enterprise Linux Manual Pages
FPDF Library PDF generator
FPDF
Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'pkexec.1' command

$ man pkexec.1

```
PKEXEC(1)                pkexec                PKEXEC(1)

NAME

    pkexec - Execute a command as another user

SYNOPSIS

    pkexec [--version] [--disable-internal-agent] [--help]

    pkexec [--user username] PROGRAM [ARGUMENTS...]

DESCRIPTION

    pkexec allows an authorized user to execute PROGRAM as another user. If

    PROGRAM is not specified, the default shell will be run. If username is

    not specified, then the program will be executed as the administrative

    super user, root.

RETURN VALUE

    Upon successful completion, the return value is the return value of

    PROGRAM. If the calling process is not authorized or an authorization

    could not be obtained through authentication or an error occured,

    pkexec exits with a return value of 127. If the authorization could not

    be obtained because the user dismissed the authentication dialog,

    pkexec exits with a return value of 126.

AUTHENTICATION AGENT

    pkexec, like any other polkit application, will use the authentication

    agent registered for the calling process or session. However, if no

    authentication agent is available, then pkexec will register its own

    textual authentication agent. This behavior can be turned off by
```

passing the --disable-internal-agent option.

SECURITY NOTES

Executing a program as another user is a privileged operation. By default the action to check for (see the section called ?ACTION AND AUTHORIZATIONS?) requires administrator authentication. In addition, the authentication dialog presented to the user will display the full path to the program to be executed so the user is aware of what will happen.

The environment that PROGRAM will run it, will be set to a minimal known and safe environment in order to avoid injecting code through LD_LIBRARY_PATH or similar mechanisms. In addition the PKEXEC_UID environment variable is set to the user id of the process invoking pkexec. As a result, pkexec will not by default allow you to run X11 applications as another user since the $DISPLAY and $XAUTHORITY environment variables are not set. These two variables will be retained if the org.freedesktop.policykit.exec.allow_gui annotation on an action is set to a nonempty value; this is discouraged, though, and should only be used for legacy programs.

Note that pkexec does no validation of the ARGUMENTS passed to PROGRAM. In the normal case (where administrator authentication is required every time pkexec is used), this is not a problem since if the user is an administrator he might as well just run pkexec bash to get root. However, if an action is used for which the user can retain authorization (or if the user is implicitly authorized) this could be a security hole. Therefore, as a rule of thumb, programs for which the default required authorization is changed, should never implicitly trust user input (e.g. like any other well-written suid program).

ACTION AND AUTHORIZATIONS

By default, the org.freedesktop.policykit.exec action is used. To use another action, use the org.freedesktop.policykit.exec.path annotation on an action with the value set to the full path of the program. In addition to specifying the program, the authentication message, description, icon and defaults can be specified. If the org.freedesktop.policykit.exec.argv1 annotation is present, the action will only be picked if the first argument to the program matches the value of the annotation.

Note that authentication messages may reference variables (see the section called ?VARIABLES?), for example $(user) will be expanded to the value of the user variable.

WRAPPER USAGE

To avoid modifying existing software to prefix their command-line invocations with pkexec, it's possible to use pkexec in a she-bang wrapper[1] like this:

```
#!/usr/bin/pkexec /usr/bin/python
import os
import sys
print "Hello, I'm running as uid %d"%(os.getuid())
for n in range(len(sys.argv)):
    print "arg[%d]=`%s'"%(n, sys.argv[n])
```

If this script is installed into /usr/bin/my-pk-test, then the following annotations

```
[...]
<annotate key="org.freedesktop.policykit.exec.path">/usr/bin/python</annotate>
<annotate key="org.freedesktop.policykit.exec.argv1">/usr/bin/my-pk-test</annotate>
[...]
```

can be used to select the appropriate polkit action. Be careful to get the latter annotation right, otherwise it will match any pkexec invocation of /usr/bin/python scripts.

VARIABLES

The following variables are set by pkexec. They can be used in authorization rules and messages shown in authentication dialogs:

program

Fully qualified path to the program to be executed. Example: ?/bin/cat?

command_line

The requested command-line (do not use this for any security checks, it is not secure). Example: ?cat /srv/xyz/foobar?

user

The user name of the user to execute the program as. Example: ?davidz?

user.gecos

The full name of the user to execute the program as. Example: ?David Zeuthen?

user.display

A representation of the user to execute the program as that is suitable for display in an authentication dialog. Is typically set to a combination of the user name and the full name. Example: ?David Zeuthen (davidz)?

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), polkitd(8), pkaction(1), pkcheck(1), pkttyagent(1)

NOTES

1. she-bang wrapper

   http://en.wikipedia.org/wiki/Shebang_(Unix)

polkit May 2009 PKEXEC(1)