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 *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 PolicyKit application, will use the authentication agent registered for the calling process. 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 required authorization (See the section called “REQUIRED 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:

[IMAGE][1]

```
+----------------------------------------------------------+
|                     Authenticate                     [X] |
+----------------------------------------------------------+
|                                                          |
|  [Icon]  Authentication is needed to run `/bin/bash'     |
|          as the super user                               |
|                                                          |
|          An application is attempting to perform an      |
|          action that requires privileges. Authentication |
|          as the super user is required to perform this   |
|          action.                                         |
|                                                          |
|          Password for root: [_________________________]  |
|                                                          |
| [V] Details:                                             |
|  Command: /bin/bash                                      |
|  Run As:  Super User (root)                              |
|  Action:  org.freedesktop.policykit.exec                 |
|  Vendor:  The PolicyKit Project                          |
|                                                          |
|                                  [Cancel] [Authenticate] |
+----------------------------------------------------------+
```

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 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.

REQUIRED AUTHORIZATIONS

By default, the *org.freedesktop.policykit.exec* authorization is required unless an action definition file is present for the program in question. To require another authorization, it can be specified using the *org.freedesktop.policykit.exec.path* annotation on an action (See the section called “EXAMPLE” for details).

EXAMPLE

To specify what kind of authorization is needed to execute the program /usr/bin/pk−example−frobnicate as another user, simply write an action definition file like this

```
<?xml version="1.0" encoding="UTF−8"?>
<!DOCTYPE policyconfig PUBLIC
 "−//freedesktop//DTD PolicyKit Policy Configuration 1.0//EN"
 "http://www.freedesktop.org/standards/PolicyKit/1/policyconfig.dtd">
<policyconfig>

  <vendor>Examples for the PolicyKit Project</vendor>
  <vendor_url>http://hal.freedesktop.org/docs/PolicyKit/</vendor_url>

  <action id="org.freedesktop.policykit.example.pkexec.run−frobnicate">
    <description>Run the PolicyKit example program Frobnicate</description>
    <description xml:lang="da">Kør PolicyKit eksemplet Frobnicate</description>
    <message>Authentication is required to run the PolicyKit example program Frobnicate (user=$(user), program=$(prog
    <message xml:lang="da">Autorisering er påkrævet for at afvikle PolicyKit eksemplet Frobnicate (user=$(user), progra
    <icon_name>audio−x−generic</icon_name>
    <defaults>
      <allow_any>no</allow_any>
      <allow_inactive>no</allow_inactive>
      <allow_active>auth_self_keep</allow_active>
    </defaults>
    <annotate key="org.freedesktop.policykit.exec.path">/usr/bin/pk−example−frobnicate</annotate>
  </action>

</policyconfig>
```

and drop it in the /usr/share/polkit−1/actions directory under a suitable name (e.g. matching the namespace of the action). Note that in addition to specifying the program, the authentication message, description, icon and defaults can be specified. Note that occurences of the strings $(user), $(program) and $(command_line) in the message will be replaced with respectively the user (of the form "Real Name (username)" or just "username" if there is no real name for the username), the binary to execute (a fully−qualified path, e.g. "/usr/bin/pk−example−frobnicate") and the command−line, e.g. "pk−example−frobnicate foo bar". For example, for the action defined above, the following authentication dialog will be shown:

[IMAGE][2]

```
+−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−+
|                     Authenticate                     [X] |
+−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−+
|                                                          |
|  [Icon]  Authentication is required to run the PolicyKit |
|          example program Frobnicate                      |
|                                                          |
|          An application is attempting to perform an      |
|       action that requires privileges. Authentication |
|       is required to perform this action.             |
|                                                       |
|       Password: [__________________________________] |
|                                                       |
| [V] Details:                                          |
| Command: /usr/bin/pk−example−frobnicate               |
| Run As:  Super User (root)                            |
| Action:  org.fd.pk.example.pkexec.run−frobnicate      |
| Vendor:  Examples for the PolicyKit Project           |
|                                                       |
|                          [Cancel] [Authenticate]      |
+−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−+
```

If the user is using the da_DK locale, the dialog looks like this:

[IMAGE][3]

```
+−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−+
|                 Autorisering                   [X]   |
+−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−+
|                                                       |
| [Icon]  Autorisering er påkrævet for at afvikle       |
|       PolicyKit eksemplet Frobnicate                  |
|                                                       |
|       Et program forsøger at udføre en handling der   |
|       kræver privilegier. Autorisering er påkrævet.   |
|                                                       |
|       Kodeord: [___________________________________] |
|                                                       |
| [V] Detaljer:                                         |
| Bruger:   Super User (root)                           |
| Program:  /usr/bin/pk−example−frobnicate              |
| Handling: org.fd.pk.example.pkexec.run−frobnicate     |
| Vendor:   Examples for the PolicyKit Project          |
|                                                       |
|                        [Annullér] [Autorisering]      |
+−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−+
```

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), such as with pk−example−frobnicate above, 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).

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), **pkcheck**(1), **pkttyagent**(1)

NOTES

1. /usr/share/gtk-doc/html/polkit-1/pkexec-bash.png
2. /usr/share/gtk-doc/html/polkit-1/pkexec-frobnicate.png
3. /usr/share/gtk-doc/html/polkit-1/pkexec-frobnicate-da.png