


```
|____Extensions____|
+-----+
```

libpolkit-gobject-1 enveloppe l'API D-Bus PolicyKit utilisant GObject. Cependant, un mécanisme peut également utiliser l'API D-Bus ou la commande pkcheck pour vérifier les autorisations.

La librairie libpolkit-agent-1 fournit une abstraction du système d'authentification natif, par exemple pam et également des facilités d'enregistrement et de communication avec le service D-Bus PolicyKit.

Les extensions PolicyKit et les backends d'autorité sont implémentés en utilisant libpolkit-backend-1

Agents d'authentification

Un agent d'authentification est utilisé pour que l'utilisateur d'une session prouve qu'il est l'utilisateur ou un utilisateur administratif. Pour s'intégrer avec le reste de la session utilisateur, les agents d'authentification sont censés être fournis par la session utilisateur que l'utilisateur utilise. Par exemple, un agent d'authentification peut ressembler à :

```
+-----+
|____Authenticate_____[X]_|
+-----+
|_____|
|_[Icon]__L'authentification_est_requise_pour_lancer_____|
|____des_tests_ATA_SMART_____|
|_____|
|____Une_application_tente_d'effectuer_une_action_____|
|____qui_necessite_des_privileges._L'authentification_|
|____super_utilisateur_est_requis_pour_effectuer_____|
|____cette_action._____|
|_____|
|____Password_for_root:_[_____]_|
|_____|
|_[V]_Details:_____|
|_Drive:___ATA_INTEL_SSDSA2MH08_(045C)_____|
|_Device:___/dev/sda_____|
|_Action:___org.fd.devicekit.disks.drive-ata-smart-selftest_|
|_Vendor:___The_DeviceKit_Project_____|
|_____|
|_____[Cancel]_[Authenticate]_|
+-----+
```

Si le système est configuré sans un compte root il peut vous autoriser à sélectionner un utilisateur administratif :

```
+-----+
|____Authenticate_____[X]_|
+-----+
|_____|
|_[Icon]__Authentication_is_required_to_run_ATA_SMART_____|
|____self_tests_____|
|_____|
|____An_application_is_attempting_to_perform_an_____|
|____action_that_requires_privileges._Authentication_|
|____as_one_of_the_users_below_is_required_to_____|
|____perform_this_action._____|
|_____|
```

```
| _____ [Face]_Patrick_Bateman_(bateman) _____ [V] |
| _____ |
| _____ Password_for_bateman:_[ _____ ] _____ |
| _____ |
| [V]_Details: _____ |
| _____ Drive:___ATA_INTEL_SSDSA2MH08_(045C) _____ |
| _____ Device:___/dev/sda _____ |
| _____ Action:___org.fd.devicekit.disks.drive-ata-smart-selftest _____ |
| _____ Vendor:___The_DeviceKit_Project _____ |
| _____ |
| _____ [Cancel]_[Authenticate] _____ |
+-----+
```

Les applications qui tournent pas sous un environnement de bureau peuvent ne pas avoir d'agent d'authentification associé avec lui. De telles applications peuvent utiliser le type `PolkitAgentTextListener` ou `pktyagent` pour que l'utilisateur puisse s'authentifier en utilisant une interface textuelle.

Déclarer des actions

Un mécanisme doit déclarer un jeu d'actions pour utiliser PolicyKit. Les actions correspondent aux opérations que les clients peuvent demander et sont définis dans des fichiers XML que le mécanisme install dans `/usr/share/polkit-1/actions`.

Les action PolicyKit sont en namespace et peuvent seulement contenir les caractères "[a-z][0-9].-". Chaque fichier XML peut contenir plus d'une action mais toutes les actions doivent être dans le même espace de nom et le fichier doit être nommé après l'espace de nom et doit avoir l'extension `.policy`. Le fichier XML doit avoir la déclaration doctype suivante :

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

L'élément `policyconfig` doit être présent exactement une fois. Les éléments qui peuvent être utilisés dans `policyconfig` incluent :

vendor Le nom du projet ou vendeur qui fournis les action dans le document XML.

vendor_url Une URL du projet ou vendeur qui fournis les action dans le document XML.

icon_name Un icône représentant le projet ou vendeur qui fournis les actions.

action Déclare une action. Le nom de l'action est spécifiée en utilisant l'attribut `id` et peuvent seulement contenir les caractères "[a-z][0-9].-".

Les éléments qui peuvent être inclus dans les actions sont :

description Une description de l'action

message Le message affiché à l'utilisateur quand les accreditifs sont demandés.

defaults Cet élément est utilisé pour spécifier des autorisation implicites au client

Les éléments qui peuvent être utilisés dans `defaults` incluent :

allow_any Autorisation implicite qui s'applique à tout client.

allow_inactive Autorisation implicite qui s'applique aux clients dans les sessions inactives dans les consoles locales.

allow_active Autorisation implicite qui s'applique aux clients dans les sessions actives dans les consoles locales.

Chaque élément `allow_any`, `allow_inactive`, et `allow_active` peuvent contenir les éléments suivants :

no non autorisé

yes autorisé

auth_self Authentification par le propriétaire de la session d'où vient le client requise

auth_admin Authentification par un utilisateur administratif est requis

auth_self_keep Comme **auth_self** mais l'autorisation est conservé pour une période brève.

auth_admin_keep Comme **auth_admin** mais l'autorisation est conservée pour une brève période.

annotate Utilisé pour annoter une action avec une paire de clé/valeur. La clé est spécifiée en utilisant l'attribut `clé` et la valeur est spécifiée en utilisant la valeur attribut. Cet élément peut apparaître 0 ou plusieurs fois.

vendor Utilisé pour remplacer le vendeur sur une base par action

vendor_url Utilisé pour remplacer l'URL vendeur sur une base par action

icon_name Utilisé pour remplacer le nom de l'icône sur une base par action

Les éléments localization, description et message peuvent exister 0 ou plusieurs fois avec différents attributs xml `:lang`

Pour lister les actions installées, utiliser la commande `pkaction`

Annotations connues

org.freedesktop.policykit.exec.path est utilisé par le programme `pkexec` fournis par PolicyKit.

org.freedesktop.policykit.impl (sa valeur est une chaîne contenant une liste séparée par un espace d'identifiants d'action). peut être utilisé pour définir des actions méta. La manière dont elles fonctionnent et que si un sujet est autorisé pour une action spécifiée, il est également autorisé pour toutes les autres action de l'annotation. Une utilisation typique est en définissant un shell avec un simple bouton lock qui devrait débloquer plusieurs actions pour des mécanismes distincts.

org.freedesktop.policykit.owner peut être utilisé pour définir un jeu d'utilisateurs qui peuvent demander si un client est autorisé à effectuer cette action. Si cette annotation n'est pas spécifiée, seul root peut vérifier si un client tournant sous un utilisateur différent est autorisé pour une action. La valeur de cette annotation est une chaîne contenant une liste séparée par des espaces d'entrées `PolkitIdentity`. par exemple `"unix-user :42 unix-user :colord"`. Une utilisation typique est pour les processus qui tournent sous un utilisateur système au lieu de root.