```
**********************************************************
    5058 Thu Jan 24 10:01:08 2019
new/usr/src/cmd/policykit/polkit-is-privileged.c
10136 smatch fix for policykit
**********************************************************
/***************************************************************************
 * CVSID: $Id$
 *
 * polkit-is-privileged.c : Determine if a user has privileges
 *
 * Copyright (C) 2006 David Zeuthen, <david@fubar.dk>
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
 *
 **************************************************************************/

/*
 * Copyright (c) 2018, Joyent, Inc.
 */

#ifdef HAVE_CONFIG_H
#  include <config.h>
#endif

#include <stdio.h>
#include <stdlib.h>
#include <getopt.h>
#include <dbus/dbus.h>

#include <libpolkit/libpolkit.h>

static void
usage (int argc, char *argv[])
{
        fprintf (stderr, "polkit-is-privileged version " PACKAGE_VERSION "\n");

        fprintf (stderr,
                 "\n"
                 "usage : %s -u <uid> -p <privilege> [-r <resource>]\n"
                 "        [-s <system-bus-connection-name>]", argv[0]);
        fprintf (stderr,
                 "\n"
                 "Options:\n"
                 "    -u, --user                     Username or user id\n"
                 "    -s, --system-bus-unique-name  Unique system bus connection
                 "    -r, --resource                 Resource\n"
                 "    -p, --privilege                Privilege to test for\n"
                 "    -h, --help                     Show this information and ex
                 "    -v, --verbose                  Verbose operation\n"
                 "    -V, --version                  Print version number\n"
                 "\n"
                 "Queries system policy whether a given user is allowed for a gi
                 "privilege for a given resource. The resource may be omitted.\n
                 "\n");
}

int
main (int argc, char *argv[])
{
        int rc;
        char *user = NULL;
        char *privilege = NULL;
        char *resource = NULL;
        char *system_bus_unique_name = NULL;
        static const struct option long_options[] = {
                {"user", required_argument, NULL, 'u'},
                {"system-bus-unique-name", required_argument, NULL, 's'},
                {"resource", required_argument, NULL, 'r'},
                {"privilege", required_argument, NULL, 'p'},
                {"help", no_argument, NULL, 'h'},
                {"verbose", no_argument, NULL, 'v'},
                {"version", no_argument, NULL, 'V'},
                {NULL, 0, NULL, 0}
        };
        LibPolKitContext *ctx = NULL;
        gboolean is_allowed;
        gboolean is_temporary;
        LibPolKitResult result;
        gboolean is_verbose = FALSE;
        DBusError error;
        DBusConnection *connection = NULL;

        rc = 1;

        while (TRUE) {
                int c;

                c = getopt_long (argc, argv, "u:r:p:s:hVv", long_options, NULL);

                if (c == -1)
                        break;

                switch (c) {
                case 's':
                        system_bus_unique_name = g_strdup (optarg);
                        break;

                case 'u':
                        user = g_strdup (optarg);
                        break;

                case 'r':
                        resource = g_strdup (optarg);
                        break;

                case 'p':
                        privilege = g_strdup (optarg);
                        break;

                case 'v':
                        is_verbose = TRUE;
                        break;

                case 'h':
                        usage (argc, argv);
                        rc = 0;
                        goto out;

                case 'V':
                        printf ("polkit-is-privileged version " PACKAGE_VERSION
                        rc = 0;
                        goto out;

                default:
                        usage (argc, argv);
                        goto out;
                }
        }

        if (user == NULL || privilege == NULL) {
                usage (argc, argv);
                return 1;
        }

        if (is_verbose) {
                printf ("user      = '%s'\n", user);
                printf ("privilege = '%s'\n", privilege);
                if (resource != NULL)
                        printf ("resource  = '%s'\n", resource);
        }

#ifdef POLKITD_ENABLED
        dbus_error_init (&error);
        connection = dbus_bus_get (DBUS_BUS_SYSTEM, &error);
        if (connection == NULL) {
                g_warning ("Cannot connect to system message bus");
                return 1;
        }
#endif /* POLKITD_ENABLED */

        ctx = libpolkit_new_context (connection);
        if (ctx == NULL) {
                g_warning ("Cannot get libpolkit context");
                goto out;
        }

        result = libpolkit_is_uid_allowed_for_privilege (ctx,
                                                         system_bus_unique_name,
                                                         user,
                                                         privilege,
                                                         resource,
                                                         &is_allowed,
                                                         &is_temporary,
                                                         NULL);
        switch (result) {
        case LIBPOLKIT_RESULT_OK:
                rc = is_allowed ? 0 : 1;
                break;

        case LIBPOLKIT_RESULT_ERROR:
                g_warning ("Error determing whether user is privileged.");
                break;

        case LIBPOLKIT_RESULT_INVALID_CONTEXT:
                g_print ("Invalid context.\n");
                goto out;

        case LIBPOLKIT_RESULT_NOT_PRIVILEGED:
                g_print ("Not privileged.\n");
                goto out;

        case LIBPOLKIT_RESULT_NO_SUCH_PRIVILEGE:
                g_print ("No such privilege '%s'.\n", privilege);
                goto out;

        case LIBPOLKIT_RESULT_NO_SUCH_USER:
```

```
                g_print ("No such user '%s'.\n", user);
                goto out;
        }

        if (is_verbose) {
                printf ("result %d\n", result);
                printf ("is_allowed %d\n", is_allowed);
        }

out:
        if (ctx != NULL)
                (void) libpolkit_free_context (ctx);
                libpolkit_free_context (ctx);

        return rc;
}
______unchanged_portion_omitted_
```