

Some days ago I decided to give the NetworkManager a new try on my Gentoo system. Month ago I gave up using it as I had some trouble getting a stable wireless lan connection working and decided to wait for a newer version. I had some minor problems to get it running as I had to rebuild the hal and the dbus package because nm-applet has kept telling me, that my wireless-device is not able to perform network scans or able to create a wireless connection.

After removing all `/etc/init.d/net.ath*` and `/etc/init.d/net.eth*` files I was able to establish network connections by using the NetworkManager. But there is an other problem I got:

NetworkManager used dhcp to determine and set the host and domainname of the system. As I use my notebook within different networks this is mostly a useful feature but brings some problems with it. As I am using Xauthority, a change of the hostname causes in a broken authority which means that applications will not find any MIT-MAGIC-COOKIE-entry in the users `.Xauthority` file that fits the host anymore. If you try to start a new application afterwards, you will get a

No protocol specified.

message on your terminal. One way to fix this is disabling Xauthority, which is not a good idea as it will disable all authentication to your Xorg session. An other way is adding a new entry to your `~/.Xauthority` file using the `xauth` command which I do automatically within a startup script after the NetworkManager has established a new connection. Here we go:

Using `xauth list` shows a list of all xauth entries within my `~/.Xauthority` file. As network connections are allways established during my Xorg-session there is an entry like this:

```
localhost.localdomain/unix:0 MIT-MAGIC-COOKIE-1 dc586be78216f3e2a2183a5c6a3dc5eb
```

This is the entry created by `xdm` (`gdm`) on session startup before any network connection has been established. The rightmost value is the cookie for session authentication. This cookie has to be used as a new entry with the new host and domainname. Adding such a cookie is straightforward. Just copy the cookie value and create a new cookie using

Hostname changes by NetworkManager while Xorg running

Friday, 11 December 2009 09:28 - Last Updated Thursday, 31 December 2009 11:39

```
xauth add "$(hostname)/unix:0" MIT-MAGIC-COOKIE-1
dc586be78216f3e2a2183a5c6a3dc5eb
```

This is the way doing it manually. As I want this automatically during connection creation, I added a new script file to `/etc/NetworkManager/dispatcher.d/99hostname` and made it executable. NetworkManager automatically executes the scripts within the `/etc/NetworkManager/dispatcher.d` directory on connection creation or releasing. The script I uses looks like this:

```
#!/bin/bash

# generate hostname and auth-type
DISPLAYNAME="$(hostname)/unix:0"
AUTHTYPE="MIT-MAGIC-COOKIE-1"

# read userid of user on DISPLAY=:0
USERID=$(eval "echo $(ck-list-sessions
| awk "
  /unix-user =/ {
    user = $3;
    next;
  };

  /x11-display = ':0'/ {
    print user;
    next;
  };
  )" )

if [ "$USERID" == "" ]; then
  echo "No user on :0 found"
  exit 0
fi
echo "User-ID:  $USERID"

USERNAME=$(cat /etc/passwd
| awk "
  /^([:]*):([:]*):$USERID/ {
    split($0, a, ":");
    print a[1]; next;
  }")

if [ "$USERNAME" == "" ]; then
  echo "Evaluating username failed"
  exit 0
```

```
fi
echo "User-Name: $USERNAME"

echo "Getting magic cookie for $USERNAME"
DEFAULTCOOKIE=$(su "$USERNAME" -c
"xauth list localhost.localdomain/unix:0 | cut -f 5 -d" ")

echo "Calling xauth for $USERNAME"
su "$USERNAME" -c
"xauth add "$DISPLAYNAME" "$AUTHTYPE" "$DEFAULTCOOKIE"
```

It mainly tries to find out, which user is currently logged on to the display :0 using ConsoleKit. Afterwards it uses su to read out the ~/.Xauthority entry of the user logged on and extract the cookie of the entry. Afterwards it added the entry using a new hostname to the users .Xauthority file.

The script is just a first draft I use and has some issues which might cause the script to fail on your site:

1. It uses /etc/passwd to get the username of the userid returned by ConsoleKit. This will not work on LDAP or other network based authentication services or services that do not use the /etc/passwd-file for authentication.
2. It depends on running ConsoleKit sessions on :0 created by gdm or xdm. If the user has no ConsoleKit-session running, it will not work.
3. If the user has more than one ConsoleKit session on :0 running, the script will currently fail in the version above as it will return multiple user ids. To cope with this, you have to fix the awk-script to return only one user id and terminate afterwards
4. Might have several other problems I do not yet know.