NetworkManager

From ArchWiki

NetworkManager (http://projects.gnome.org/NetworkManager/) is a program for providing detection and configuration for systems to automatically connect to network. NetworkManager's functionality can be useful for both wireless and wired networks. For wireless networks, NetworkManager prefers known wireless networks and has the ability to switch to the most reliable network. NetworkManager-aware applications can switch from online and offline mode. NetworkManager also prefers wired connections over wireless ones, has support for modem connections and certain types of VPN. NetworkManager was originally developed by Red Hat and now is hosted by the GNOME project.

Related articles

- Network configuration
- Wireless network configuration
- Category:Network managers

Warning: By default, Wi-Fi passwords are stored in clear text. See section #Encrypted Wi-Fi passwords

Contents

Base install

NetworkManager can be installed with the package networkmanager (https://www.archlinux.org/packages/?name=networkmanager), available in the official repositories.

> **Note:** You must ensure that no other service that wants to configure the network is running; in fact, multiple networking services will conflict. You can find a list of the currently running services with `systemctl --type=service` and then stop them.

VPN support

NetworkManager VPN support is based on a plug-in system. If you need VPN support via NetworkManager, you have to install one of the following packages from the official repositories:

- networkmanager-openconnect (https://www.archlinux.org/packages/?name=networkmanager-openconnect)
- networkmanager-openvpn (https://www.archlinux.org/packages/?name=networkmanager-openvpn)
- networkmanager-pptp (https://www.archlinux.org/packages/?name=networkmanager-pptp)
- networkmanager-vpnc (https://www.archlinux.org/packages/?name=networkmanager-vpnc)

From the AUR:

- networkmanager-l2tp (https://aur.archlinux.org/packages/networkmanager-l2tp/)

PPPoE / DSL support

Install rp-pppoe (https://www.archlinux.org/packages/?name=rp-pppoe) for PPPoE / DSL connection support.

Graphical front-ends

To configure and have easy access to NetworkManager, most users will want to install an applet. This GUI front-end usually resides in the system tray (or notification area) and allows network selection and configuration of NetworkManager. Various applets exist for different types of desktops.

GNOME

GNOME's network-manager-applet (https://www.archlinux.org/packages/?name=network-manager-applet) works in all environments.

To store authentication details for connections (Wireless/DSL) install and configure GNOME Keyring.

Be aware that after enabling the tick-box option `Make available to other users` for a connection, NetworkManager stores the password in plain-text, though the respective file is accessible only to root (or other users via `nm-applet`). See NetworkManager#Encrypted Wi-Fi passwords.

KDE

KDE 4

Install kdeplasma-applets-plasma-nm (https://www.archlinux.org/packages/?name=kdeplasma-applets-plasma-nm) applet.

> **Note:** The older KNetworkManager kdeplasma-applets-networkmanagement (https://aur.archlinux.org/packages/kdeplasma-applets-networkmanagement/) is available from the AUR, but is considered as legacy.

Activate the applet in the KDE System Tray (https://userbase.kde.org/Plasma/SystemTray): right click somewhere in the System Tray outside of the service icons, then in the Display page, activate the network management checkbox. The applet can allow you to store encrypted WiFi passwords in KDE Wallet and will prompt you to do this. As for GNOME, enabling `all users may connect to this network` for a connection lets NetworkManager store the password in plain text.

If you have both the KDE Plasma widget and GNOME's `nm-applet` installed and do not want to start `nm-applet` when using KDE, add the following line to `~/.config/autostart/nm-applet.desktop`:

```
NotShowIn=KDE
```

See Userbase page (http://userbase.kde.org/NetworkManagement) for more info.

KDE 5

Install plasma-nm (https://www.archlinux.org/packages/?name=plasma-nm) applet.

Xfce

While network-manager-applet (https://www.archlinux.org/packages/?name=network-manager-applet) works in Xfce, but in order to see notifications, including error messages, `nm-applet` needs an implementation of the Freedesktop desktop notifications specification (see the Galapago Project (http://www.galago-project.org/specs/notification/0.9/index.html)) to display them. To enable notifications install xfce4-notifyd (https://www.archlinux.org/packages/?name=xfce4-notifyd), a package that provides an implementation for the specification.

Without such a notification daemon, `nm-applet` outputs the following errors to stdout/stderr:

```
(nm-applet:24209): libnotify-WARNING **: Failed to connect to proxy
** (nm-applet:24209): WARNING **: get_all_cb: couldn't retrieve
system settings properties: (25) Launch helper exited with unknown
return code 1.
** (nm-applet:24209): WARNING **: fetch_connections_done: error
fetching connections: (25) Launch helper exited with unknown return
code 1.
** (nm-applet:24209): WARNING **: Failed to register as an agent:
(25) Launch helper exited with unknown return code 1
```

`nm-applet` will still work fine, though, but without notifications.

If `nm-applet` is not prompting for a password when connecting to new wifi networks, and is just disconnecting immediately, you probably need to install gnome-keyring (https://www.archlinux.org/packages/?name=gnome-keyring).

Openbox

To work properly in Openbox, the GNOME applet requires the xfce4-notifyd (https://www.archlinux.org/packages/?name=xfce4-notifyd) notification daemon for the same reason as in XFCE and the gnome-icon-theme (https://www.archlinux.org/packages/?name=gnome-icon-theme) package to be able to display the applet in the systray.

If you want to store authentication details (Wireless/DSL) install and configure gnome-keyring.

`nm-applet` installs the autostart file at `/etc/xdg/autostart/nm-applet.desktop`. If you have issues with it (e.g. `nm-applet` is started twice or is not started at all), see Openbox#autostart or [1] (https://bbs.archlinux.org/viewtopic.php?pid=993738) for solution.

Other desktops and window managers

In all other scenarios it is recommended to use the GNOME applet. You will also need to be sure that the gnome-icon-theme (https://www.archlinux.org/packages/?name=gnome-icon-theme) package is installed to be able to display the applet.

To store connection secrets install and configure GNOME Keyring.

In order to run `nm-applet` without a systray, you can use trayer (https://www.archlinux.org/packages/?name=trayer) or stalonetray (https://www.archlinux.org/packages/?name=stalonetray). For example, you can add a script like this one in your path:

nmgui

```
#!/bin/sh
nm-applet    2>&1 /dev/null &
stalonetray  2>&1 /dev/null
killall nm-applet
```

When you close the *stalonetray* window, it closes `nm-applet` too, so no extra memory is used once you are done with network settings.

Command line

nmcli

The networkmanager (https://www.archlinux.org/packages/?name=networkmanager) package contains nmcli (http://manpages.ubuntu.com/manpages/maverick/man1/nmcli.1.html) since version 0.8.1.

Examples:

- To connect to a wifi network:

```
nmcli dev wifi connect <name> password <password>
```

- To connect to a wifi on the `wlan1` wifi interface:

```
nmcli dev wifi connect <name> password <password> iface wlan1 [profile name]
```

- To disconnect an interface:

```
nmcli dev disconnect iface eth0
```

- To reconnect an interface marked as disconnected:

```
nmcli con up uuid <uuid>
```

- To get a list of UUIDs:

```
nmcli con show
```

- To see a list of network devices and their state:

```
nmcli dev
```

- To turn off wifi:

```
nmcli r wifi off
```

nmtui

nmtui is a graphic frontend for *networkmanager* using curses. It is useful for configuring and managing networks without X. The networkmanager (https://www.archlinux.org/packages/?name=networkmanager) package contains *nmtui* since version 0.9.10.

nmcli-dmenu

Alternatively there is `networkmanager-dmenu-git` (https://aur.archlinux.org/packages/networkmanager-dmenu-git/) which is a small script to manage NetworkManager connections with *dmenu* instead of `nm-applet`. It provides all essential features such as connect to existing NetworkManager wifi or wired connections, connect to new wifi connections, requests passphrase if required, connect to existing VPN connections, enable/disable networking, launch *nm-connection-editor* GUI.

Configuration

NetworkManager will require some additional steps to be able run properly. Make sure you have configured `/etc/hosts` as described in Network configuration#Set the hostname section.

Enable NetworkManager

NetworkManager is controlled via `NetworkManager.service`. Once the NetworkManager daemon is started, it will automatically connect to any available "system connections" that have already been configured. Any "user connections" or unconfigured connections will need *nmcli* or an applet to configure and connect.

> **Note:** NetworkManager will print meaningless warnings (FS#34971 (https://bugs.archlinux.org/task/34971)) to your system log, when NetworkManager-dispatcher.service and ModemManager.service (https://www.archlinux.org/packages/?name=modemmanager) are not enabled. You may enable both to suppress the messages.

Enable NetworkManager Wait Online

If you have services which fail if they are started before the network is up, you may use `NetworkManager-wait-online.service` in addition to `NetworkManager.service`. This is, however, rarely necessary because most networked daemons start up okay, even if the network has not been configured yet.

In some cases, the service will still fail to start successfully on boot due to the timeout setting in `/usr/lib/systemd/system/NetworkManager-wait-online.service` being too short. Change the default timeout from 30 to a higher value.

Set up PolicyKit permissions

See General troubleshooting#Session permissions for setting up a working session.

With a working session, you have several options for granting the necessary privileges to NetworkManager:

- *Option 1.* Run a Polkit authentication agent when you log in, such as `/usr/lib/polkit-gnome/polkit-gnome-authentication-agent-1` (part of `polkit-`

gnome (https://www.archlinux.org/packages/?name=polkit-gnome)). You will be prompted for your password whenever you add or remove a network connection.
- *Option 2.* Add yourself to the `wheel` group. You will not have to enter your password, but your user account may be granted other permissions as well, such as the ability to use sudo without entering the root password.
- *Option 3.* Add yourself to the `network` group and create the following file:

```
/etc/polkit-1/rules.d/50-org.freedesktop.NetworkManager.rules
```

```
polkit.addRule(function(action, subject) {
  if (action.id.indexOf("org.freedesktop.NetworkManager.") == 0 && subject.isInGroup("network"))
    return polkit.Result.YES;
  }
});
```

All users in the `network` group will be able to add and remove networks without a password. This will not work under systemd if you do not have an active session with *systemd-logind*.

Network services with NetworkManager dispatcher

There are quite a few network services that you will not want running until NetworkManager brings up an interface. Good examples are NTPd and network filesystem mounts of various types (e.g. **netfs**). NetworkManager has the ability to start these services when you connect to a network and stop them when you disconnect. To activate the feature you need to start the `NetworkManager-dispatcher.service`.

Once the feature is active, scripts can be added to the `/etc/NetworkManager/dispatcher.d` directory. These scripts must be **owned by root**, otherwise the dispatcher will not execute them. For added security, set group ownership to root as well:

```
# chown root:root scriptname
```

Also, the script must have **write permission for owner only**, otherwise the dispatcher will not execute them:

```
# chmod 755 scriptname
```

The scripts will be run in alphabetical order at connection time, and in reverse alphabetical order at disconnect time. They receive two arguments: the name of the interface (e.g. `eth0`) and the status (*up* or *down* for interfaces and *vpn-up* or *vpn-down* for vpn connections). To ensure what order they come up in, it is common to use numerical characters prior to the name of the script (e.g. `10_portmap` or `30_netfs` (which ensures that the *portmapper* is up before NFS mounts are attempted).

Warning: If you connect to foreign or public networks, be aware of what services you are starting and what servers you expect to be available for them to connect to. You could make a security hole by starting the wrong services while connected to a public network

Avoiding the dispatcher timeout

If the above is working, then this section is not relevant. However, there is a general problem related to running dispatcher scripts which take longer to be executed. Initially an internal timeout of three seconds only was used. If the called script did not complete in time, it was killed. Later the timeout was extended to about 20 seconds (see the Bugtracker (https://bugzilla.redhat.com/show_bug.cgi?id=982734) for more information). If the timeout still creates the problem, a work around may be to modify the dispatcher service file `/usr/lib/systemd/system/NetworkManager-dispatcher.service` to remain active after exit:

```
/etc/systemd/system/NetworkManager-dispatcher.service
```

```
.include /usr/lib/systemd/system/NetworkManager-dispatcher.service
[Service]
RemainAfterExit=yes
```

Now start and enable the modified `NetworkManager-dispatcher` service.

Warning: Adding the `RemainAfterExit` line to it will prevent the dispatcher from closing. Unfortunately, the dispatcher **has** to close before it can run your scripts again. With it the dispatcher will not time out but it also will not close, which means that the scripts will only run once per boot. Therefore, do not add the line unless the timeout is definitely causing a problem.

Start OpenNTPD

Install the networkmanager-dispatcher-openntpd (https://www.archlinux.org/packages/?name=networkmanager-dispatcher-openntpd) package.

Mount remote folder with sshfs

As the script is run in a very restrictive environment, you have to export `SSH_AUTH_SOCK` in order to connect to your SSH agent. There are different ways to accomplish this, see this message (https://bbs.archlinux.org/viewtopic.php?pid=1042030#p1042030) for more information. The example below works with GNOME Keyring, and will ask you for the password if not unlocked already. In case NetworkManager connects automatically on login, it is likely *gnome-keyring* has not yet started and the export will fail (hence the sleep). The `UUID` to match can be found with the command `nmcli con status` or `nmcli con list`.

```
#!/bin/sh
USER='username'
```

```
REMOTE='user@host:/remote/path'
LOCAL='/local/path'

interface=$1 status=$2
if [ "$CONNECTION_UUID" = "uuid" ]; then
  case $status in
    up)
      export SSH_AUTH_SOCK=$(find /tmp -maxdepth 1 -type s -user "$USER" -name 'ssh')
      su "$USER" -c "sshfs $REMOTE $LOCAL"
      ;;
    down)
      fusermount -u "$LOCAL"
      ;;
  esac
fi
```

Use dispatcher to connect to a VPN after a network connection is established

In this example we want to connect automatically to a previously defined VPN connection after connecting to a specific Wi-Fi network. First thing to do is to create the dispatcher script that defines what to do after we are connected to the network.

1. Create the dispatcher script:

```
/etc/NetworkManager/dispatcher.d/vpn-up
```

```
#!/bin/sh
VPN_NAME="name of VPN connection defined in NetworkManager"
ESSID="Wi-Fi network ESSID (not connection name)"

interface=$1 status=$2
case $status in
  up|vpn-down)
    if iwgetid | grep -qs ":\"$ESSID\""; then
      nmcli con up id "$VPN_NAME"
    fi
    ;;
  down)
    if iwgetid | grep -qs ":\"$ESSID\""; then
      if nmcli con status id "$VPN_NAME" | grep -qs activated; then
        nmcli con down id "$VPN_NAME"
      fi
    fi
    ;;
esac
```

If you would like to attempt to automatically connect to VPN for all Wi-Fi networks, you can use the following definition of the ESSID: `ESSID=$(iwgetid -r)`. Remember to set the script's permissions accordingly.

If you require and tick the `nm-applet` option to *Make the VPN connection available to all users*, trying to connect may still fail and NetworkManager will complain about 'no valid VPN secrets', because of the way VPN secrets are stored (http://developer.gnome.org/NetworkManager/0.9/secrets-flags.html), which brings us to step 2:

2. Either edit the VPN connection configuration file to make NetworkManager

store the secrets by itself rather than inside a keyring that will be inaccessible for root (https://bugzilla.redhat.com/show_bug.cgi?id=710552): open up `/etc/NetworkManager/system-connections/`*`name of your VPN connection`* and change the `password-flags` and `secret-flags` from `1` to `0`.

Alternatively put the password directly in the configuration file adding the section `vpn-secrets`:

```
[vpn]
....
password-flags=0

[vpn-secrets]
password=your_password
```

Note: It may now be necessary to re-open the NetworkManager connection editor and save the VPN passwords/secrets again.

Proxy settings

NetworkManager does not directly handle proxy settings, but if you are using GNOME, you could use proxydriver (http://marin.jb.free.fr/proxydriver/) wich handles proxy settings using NetworkManager's informations. You can find the package for proxydriver (https://aur.archlinux.org/packages/proxydriver/) in the AUR.

In order for *proxydriver* to be able to change the proxy settings, you would need to execute this command, as part of the GNOME startup process (System -> Preferences -> Startup Applications):

```
xhost +si:localuser:your_username
```

See: Proxy settings.

Disable NetworkManager

It might not be obvious, but the service automatically starts through *dbus*. To completely disable it you can mask the service with *systemctl*:

```
systemctl mask NetworkManager
systemctl mask NetworkManager-dispatcher
```

Testing

NetworkManager applets are designed to load upon login so no further configuration should be necessary for most users. If you have already disabled your previous network settings and disconnected from your network, you can now test if NetworkManager will work. The first step is to start `NetworkManager.service`.

Some applets will provide you with a `.desktop` file so that the NetworkManager applet can be loaded through the application menu. If it does not, you are going to either have to discover the command to use or logout and login again to start the applet. Once the applet is started, it will likely begin polling network connections with for auto-configuration with a DHCP server.

To start the GNOME applet in non-xdg-compliant window managers like awesome:

```
nm-applet --sm-disable &
```

For static IP addresses, you will have to configure NetworkManager to understand them. The process usually involves right-clicking the applet and selecting something like 'Edit Connections'.

Troubleshooting

Some fixes to common problems.

No prompt for password of secured Wi-Fi networks

When trying to connect to a secured Wi-Fi network, no prompt for a password is shown and no connection is established. This happens when no keyring package is installed. An easy solution is to install `gnome-keyring (https://www.archlinux.org/packages/?name=gnome-keyring)`. If you want the passwords to be stored in encrypted form, follow GNOME Keyring to set up the *gnome-keyring-daemon*.

No traffic via PPTP tunnel

PPTP connection logins successfully; you see a ppp0 interface with the correct VPN IP address, but you cannot even ping the remote IP address. It is due to lack of MPPE (Microsoft Point-to-Point Encryption) support in stock Arch pppd. It is recommended to first try with the stock Arch `ppp (https://www.archlinux.org/packages/?name=ppp)` as it may work as intended.

To solve the problem it should be sufficient to install `ppp-mppe (https://aur.archlinux.org/packages/ppp-mppe/)` from the AUR.

See also WPA2_Enterprise#MS-CHAPv2.

Network management disabled

When NetworkManager shuts down but the pid (state) file is not removed, you will see a `Network management disabled` message. If this happens, remove the file manually:

```
# rm /var/lib/NetworkManager/NetworkManager.state
```

Customizing resolv.conf

See the main page: resolv.conf. Also make sure that NetworkManager uses dhcpcd (https://www.archlinux.org/packages/?name=dhcpcd) and not dhclient (https://www.archlinux.org/packages/?name=dhclient). If you want to use dhclient (https://www.archlinux.org/packages/?name=dhclient), you may try the networkmanager-dispatch-resolv (https://aur.archlinux.org/packages/networkmanager-dispatch-resolv/) package from the AUR.

DHCP problems with dhclient

If you have problems with getting an IP address via DHCP, try to add the following to your `/etc/dhclient.conf`:

```
interface "eth0" {
  send dhcp-client-identifier 01:aa:bb:cc:dd:ee:ff;
}
```

Where `aa:bb:cc:dd:ee:ff` is the MAC address of this NIC. The MAC address can be found using the `ip link show *interface*` command from the iproute2 (https://www.archlinux.org/packages/?name=iproute2) package.

Hostname problems

It depends on the NetworkManager plugins used, whether the hostname is forwarded to a router on connect. The generic "keyfile" plugin does not forward the hostname in default configuration. To make it forward the hostname, add the following to `/etc/NetworkManager/NetworkManager.conf`:

```
[keyfile]
hostname=your_hostname
```

The options under `[keyfile]` will be applied to network connections in the default `/etc/NetworkManager/system-connections` path.

Another option is to configure the DHCP client, which NetworkManager starts automatically, to forward it. NetworkManager utilizes dhclient (https://www.archlinux.org/packages/?name=dhclient) in default and falls back to dhcpcd (https://www.archlinux.org/packages/?name=dhcpcd), if the former is not installed. To make *dhclient* forward the hostname requires to set a non-default option, *dhcpcd* forwards the hostname by default.

First, check which DHCP client is used (*dhclient* in this example):

```
# journalctl -b | egrep "dhclient|dhcpcd"
```

```
...
Nov 17 21:03:20 zenbook dhclient[2949]: Nov 17 21:03:20 zenbook dhclient[2949]: Bound to *:546
Nov 17 21:03:20 zenbook dhclient[2949]: Listening on Socket/wlan0
Nov 17 21:03:20 zenbook dhclient[2949]: Sending on   Socket/wlan0
Nov 17 21:03:20 zenbook dhclient[2949]: XMT: Info-Request on wlan0, interval 1020ms.
Nov 17 21:03:20 zenbook dhclient[2949]: RCV: Reply message on wlan0 from fe80::126f:3fff:fe0c:2d
```

Configure dhclient to push the hostname to the DHCP server

Copy the example configuration file:

```
# cp /usr/share/dhclient/dhclient.conf.example /etc/dhclient.conf
```

Take a look at the file - there will only really be one line we want to keep and *dhclient* will use it's defaults (as it has been using if you didn't have this file) for the other options. This is the important line:

```
/etc/dhclient.conf
```

```
send host-name = pick-first-value(gethostname(), "ISC-dhclient");
```

Force an IP address renewal by your favorite means, and you should now see your hostname on your DHCP server.

Configure NetworkManager to use dhcpcd

Install `dhcpcd (https://www.archlinux.org/packages/?name=dhcpcd)` and tell NetworkManager about it:

```
/etc/NetworkManager/NetworkManager.conf
```

```
dhcp=dhcpcd
```

Then restart `NetworkManager.service`.

Missing default route

On at least one KDE4 system, no default route was created when establishing wireless connections with NetworkManager. Changing the route settings of the wireless connection to remove the default selection "Use only for resources on this connection" solved the issue.

3G modem not detected

See USB 3G Modem#Network Manager.

Switching off WLAN on laptops

Sometimes NetworkManager will not work when you disable your Wi-Fi adapter with a switch on your laptop and try to enable it again afterwards. This is often a problem with *rfkill*. Install `rfkill (https://www.archlinux.org/packages/?name=rfkill)` from the official repositories and use:

```
$ watch -n1 rfkill list all
```

to check if the driver notifies *rfkill* about the wireless adapter's status. If one identifier stays blocked after you switch on the adapter you could try to manually unblock it with (where X is the number of the identifier provided by the above output):

```
# rfkill event unblock X
```

Static IP address settings revert to DHCP

Due to an unresolved bug, when changing default connections to a static IP address, `nm-applet` may not properly store the configuration change, and will revert to automatic DHCP.

To work around this issue you have to edit the default connection (e.g. "Auto eth0") in `nm-applet`, change the connection name (e.g. "my eth0"), uncheck the "Available to all users" checkbox, change your static IP address settings as desired, and click **Apply**. This will save a new connection with the given name.

Next, you will want to make the default connection not connect automatically. To do so, run `nm-connection-editor` (**not** as root). In the connection editor, edit the default connection (e.g. "Auto eth0") and uncheck "Connect automatically". Click **Apply** and close the connection editor.

Cannot edit connections as normal user

See #Set up PolicyKit permissions.

Forget hidden wireless network

Since hidden networks are not displayed in the selection list of the Wireless view, they cannot be forgotten (removed) with the GUI. You can delete one with the following command:

```
# rm /etc/NetworkManager/system-connections/SSID
```

This works for any other connection.

VPN not working in GNOME

When setting up OpenConnect or vpnc connections in NetworkManager while using GNOME, you'll sometimes never see the dialog box pop up and the following error appears in `/var/log/errors.log`:

```
localhost NetworkManager[399]: <error> [1361719690.10506] [nm-vpn-connection.c:1405] get_secrets
```

This is caused by the GNOME NM Applet expecting dialog scripts to be at `/usr/lib/gnome-shell`, when NetworkManager's packages put them in `/usr/lib/networkmanager`. As a "temporary" fix (this bug has been around for a while now), make the following symlink(s):

- For OpenConnect:
  `ln -s /usr/lib/networkmanager/nm-openconnect-auth-dialog /usr/lib/gnome-shell/`
- For VPNC (i.e. Cisco VPN):
  `ln -s /usr/lib/networkmanager/nm-vpnc-auth-dialog /usr/lib/gnome-shell/`

This may need to be done for any other NM VPN plugins as well, but these are the two most common.

Unable to connect to visible European wireless networks

WLAN chips are shipped with a default regulatory domain. If your access point does not operate within these limitations, you will not be able to connect to the network. Fixing this is easy:

1. Install `crda` (https://www.archlinux.org/packages/?name=crda)
2. Uncomment the correct Country Code in `/etc/conf.d/wireless-regdom`
3. Reboot the system, because the setting is only read on boot

Automatic connect to VPN on boot is not working

The problem occurs when the system (i.e. NetworkManager running as the root user) tries to establish a VPN connection, but the password is not accessible because it is stored in the Gnome keyring of a particular user.

A solution is to keep the password to your VPN in plaintext, as described in step (2.) of #Use dispatcher to connect to a VPN after a network connection is established.

You do not need to use the dispatcher described in step (1.) to auto-connect anymore, if you use the new "auto-connect VPN" option from the `nm-applet` GUI.

dhcpcd repetitively refusing leases

An occurrence of dhcpcd repetitively refusing leases has been reported, while spouting a large quantity of the following log messages :

```
dhcpcd[25188]: wlan0: NAK: from 10.1.0.1
dhcpcd[25188]: wlan0: soliciting a DHCP lease
dhcpcd[25188]: wlan0: offered 10.2.0.159 from 10.2.0.1
dhcpcd[25188]: wlan0: ignoring offer of 10.1.0.197 from 10.1.0.1
dhcpcd[25188]: wlan0: NAK: from 10.1.0.1
dhcpcd[25188]: wlan0: soliciting a DHCP lease
dhcpcd[25188]: wlan0: offered 10.2.0.159 from 10.2.0.1
dhcpcd[25188]: wlan0: ignoring offer of 10.1.0.197 from 10.1.0.1
```

This problem seems to be solved by switching to dhclient (https://www.archlinux.org/packages/?name=dhclient) instead of dhcpcd (https://www.archlinux.org/packages/?name=dhcpcd) (with NetworkManager, set the `dhcp` option to `dhclient` in `/etc/NetworkManager/NetworkManager.conf`).

Systemd Bottleneck

Over time the log files (`/var/log/journal`) can become very large. This can have a big impact on boot performance when using NetworkManager, see: Systemd#Boot time increasing over time.

Tips and tricks

Encrypted Wi-Fi passwords

By default, NetworkManager stores passwords in clear text in the connection files at `/etc/NetworkManager/system-connections/`. To print the stored passwords, use the following command:

```
# grep -H '^psk=' /etc/NetworkManager/system-connections/*
```

The passwords are accessible to the root user in the filesystem and to users with access to settings via the GUI (e.g. `nm-applet`).

If it is preferable to save the passwords in encrypted form instead of clear text, this can be achieved by storing them in a keyring which NetworkManager then queries for the passwords. A suggested keyring daemon is GNOME Keyring or (for KDE specifically) KDE Wallet. The keyring daemon has to be started and the keyring needs to be unlocked for the following to work.

Furthermore, NetworkManager needs to be configured not to store the password for all users. Using GNOME `nm-applet`, right click the applet, go to `Edit connections`, select a network connection, click `edit`, select the `General` tab and untick `Make available to other users`. Using KDE's kdeplasma-applets-plasma-nm (https://www.archlinux.org/packages/?name=kdeplasma-applets-plasma-nm), click the applet, click on the top right `Settings` icon, double click on a network connection, in the `General settings` tab, untick `all users may connect to this network`. If the option is ticked, the passwords will still be stored in clear text, even if a keyring daemon is running.

If the option was selected previously and you un-tick it, you may have to use the `reset` option first to make the password disappear from the file. Alternatively, delete the connection first and set it up again.

The downside of using the keyring is that the connections have to be set up for each user.

Sharing internet connection over Wi-Fi

You can share your internet connection (e.g.: 3G or wired) with a few clicks using nm. You will need a supported Wi-Fi card (Cards based on Atheros AR9xx or at least AR5xx are probably best choice).

Ad-hoc

- Install the dnsmasq (https://www.archlinux.org/packages/?name=dnsmasq) package to be able to actually share the connection.
- Custom dnsmasq.conf may interfere with NetworkManager (not sure about this, but i think so).
- Click on applet and choose "Create new wireless network".
- Follow wizard (if using WEP, be sure to use 5 or 13 character long password, different lengths will fail).
- Settings will remain stored for the next time you need it.

Real AP

Support of infrastructure mode (which is needed by Android phones as they intentionally do not support ad-hoc) is added by NetworkManager as of late 2012.

See Fedora's wiki (https://fedoraproject.org/wiki/Features/RealHotspot).

Checking if networking is up inside a cron job or script

Some *cron* jobs require networking to be up to succeed. You may wish to avoid running these jobs when the network is down. To accomplish this, add an **if** test for networking that queries NetworkManager's *nm-tool* and checks the state of networking. The test shown here succeeds if any interface is up, and fails if they are all down. This is convenient for laptops that might be hardwired, might be on wireless, or might be off the network.

```
if [ $(nm-tool|grep State|cut -f2 -d' ') == "connected" ]; then
    #Whatever you want to do if the network is online
else
    #Whatever you want to do if the network is offline - note, this and the else above are optio
fi
```

This useful for a cron.hourly script that runs *fpupdate* for the F-Prot virus scanner signature update, as an example. Another way it might be useful, with a little modification, is to differentiate between networks using various parts of the output from *nm-tool*; for example, since the active wireless network is denoted with an asterisk, you could grep for the network name and then grep for a literal asterisk.

Automatically unlock keyring after login

GNOME

1. Right click on the `nm-applet` icon in your panel and select Edit Connections and open the Wireless tab
2. Select the connection you want to work with and click the Edit button
3. Check the boxes “Connect Automatically” and “Available to all users”

Log out and log back in to complete.

Note: The following method is dated and known not to work on at least one machine!

- In `/etc/pam.d/gdm` (or your corresponding daemon in `/etc/pam.d`), add these lines at the end of the "auth" and "session" blocks if they do not exist already:

```
auth            optional        pam_gnome_keyring.so
session         optional        pam_gnome_keyring.so  auto_start
```

- In `/etc/pam.d/passwd`, use this line for the 'password' block:

```
password    optional    pam_gnome_keyring.so
```

Next time you log in, you should be asked if you want the password to be unlocked automatically on login.

KDE

Note: See https://wiki.gnome.org/Projects/GnomeKeyring/Pam/Manual for reference, and if you are using KDE with KDM, you can use pam-keyring-tool (https://aur.archlinux.org/packages/pam-keyring-tool/) from the AUR.

Put a script like the following in `~/.kde4/Autostart`:

```
#!/bin/sh
echo PASSWORD | /usr/bin/pam-keyring-tool --unlock --keyring=default -s
```

Similar should work with Openbox, LXDE, etc.

SLiM login manager

See Slim#SLiM and Gnome Keyring.

KDE and OpenConnect VPN with password authentication

kdeplasma-applets-plasma-nm (https://www.archlinux.org/packages/?name=kdeplasma-applets-plasma-nm) now supports configuring username and password for OpenConnect VPN connections. Open your VPN connection, accept the certificate, and connection fields will appear. If not, see the instructions below. Now enter the correct username and password.

Troubleshooting

While you may type both values at connection time, kdeplasma-applets-plasma-nm (https://www.archlinux.org/packages/?name=kdeplasma-applets-plasma-nm) 0.9.3.2-1 and above are capable of retrieving OpenConnect username and password directly from KWallet.

Open "KDE Wallet Manager" and look up your OpenConnect VPN connection under "Network Management|Maps". Click "Show values" and enter your credentials in key "VpnSecrets" in this form (replace *username* and *password* accordingly):

```
form:main:username%SEP%username%SEP%form:main:password%SEP%password
```

Next time you connect, username and password should appear in the "VPN secrets" dialog box.

Ignore specific devices

Sometimes it may be desired that NetworkManager ignores specific devices and does not try to configure addresses and routes for them.You can quickly and easily ignore devices by MAC or interface-name by using the following in /etc/NetworkManager/NetworkManager.conf :

```
[keyfile]
unmanaged-devices=mac:00:22:68:1c:59:b1;mac:00:1E:65:30:D1:C4;interface-name:eth0
```

After you have put this in, restart NetworkManager, and you should be able to configure interfaces without NetworkManager altering what you have set.

Connect faster

Enable DNS Caching

DNS requests can be sped up by caching previous requests locally for subsequent lookup. NetworkManager has a plugin to enable DNS caching using dnsmasq, but it is not enabled in the default configuration. It is, however, easy to enable using the following instructions.

Start by installing dnsmasq (https://www.archlinux.org/packages/?name=dnsmasq). Then, edit /etc/NetworkManager/NetworkManager.conf and add the following line under the [main] section:

```
dns=dnsmasq
```

Now restart NetworkManager or reboot. NetworkManager will automatically start dnsmasq and add 127.0.0.1 to `/etc/resolv.conf`. The actual DNS servers can be found in `/var/run/NetworkManager/dnsmasq.conf`. You can verify dnsmasq is being used by doing the same DNS lookup twice with dig and verifying the server and query times.

Enable IPv6 Privacy Extensions

NetworkManager does not honour the settings placed in `/etc/sysctl.d/40-ipv6.conf` when following Ipv6#Privacy_extensions. This can be verified by running `$ ip -6 addr show [device]` after rebooting: no `scope global` **`temporary`** address appears.

In order to enable IPv6 Privacy Extensions for NetworkManager-managed connections, edit as root the desired connection keyfile in `/etc/NetworkManager/system-connections/` and append to its `[ipv6]` section the key-value pair `ip6-privacy=2`:

/etc/NetworkManager/system-connections/*example_connection*

```
...
[ipv6]
method=auto
ip6-privacy=2
```

Source: Fedora wiki (https://fedoraproject.org/wiki/Tools/NetworkManager/IPv6#IPv6_Privacy_Extensions)

Retrieved from "https://wiki.archlinux.org/index.php?title=NetworkManager&oldid=352661"
Category: Network managers

- This page was last modified on 25 December 2014, at 08:59.
-