Installing Step by Step

(Version 0.15)

Contents

This HowTo describes the installation and setup of the “Fingerprint GUI” project. It was tested on *Ubuntu 8.04, 8.10, 9.04, 9.10 and 10.04 Desktop*, *Lubuntu 10.04* and *Fedora 10 and 12* (32bit versions) and *Ubuntu 8.10 9.04 and 10.04 Desktop* (64bit version) each new installed with default settings. It is applicable for GDM/Gnome desktop systems only and **can not be used as a HowTo for KDE systems**. It should show the principles of installing and configuring the system and provide enough information needed for deployment in other Linux distributions. In the chapter **"Other Linux distributions"**, my experiences with these distributions are described. I didn't have the time to solve all those problems. That should be the responsibility of the distributors or of experienced users. Please contact me if you have the system installed properly to such a distribution or if you experience a bug.

1 Installing Executables and Libraries

After downloading the “fingerprint-gui-x.y-<arch>.tar.gz” package please unpack it into some directory (`tar -xzf fingerprint-gui-x.y-<arch>.tar.gz`). Then change to this directory and become “root”. The command for installation is “`./install.sh [--uninstall]`”. If you have a device from UPEK Inc. or SGS Thomson you will need the proprietary driver library “libbsapi.so” from UPEK Inc. In this case you will be prompted for installing this library. If you chose “Yes” the “libbsapi.so” file (for your architecture) will be copied to “/usr/lib/” and “ldconfig” will be called then. If you have a device from other vendors you will not be prompted for installing “libbsapi.so”. Please have a look at the libfprint homepage (http://reactivated.net/fprint/wiki/Main_Page) for a list of supported devices.

1.1 Installing required Libraries

When executing “./install.sh” as root it will probably print a list of missing libraries. Use your package manager to install the required packages and their dependencies. Below is a list of packages to install:

Ubuntu 10.04 Desktop:

- libfakekey0
- libfprint0 (**IMPORTANT**: Since version 0.14 fingerprint-gui requires libfprint0 version 0.1.0~pre2-1 that is not part of the Ubuntu repository yet. Please install this version from https://launchpad.net/~fingerprint/+archive/fprint .
- libqca2
- libqca2-plugin-ossl
- libqt4-xml

Fedora 12:

- libfakekey-0.1.3
- libfprint-0.1.0-14.pre2
- qt-x11-1:4.6.2-16
- qca2-2.0.2-2
- qca-ossl-2.0.0-0.8.beta3

1.2 Installing executables

If all required libraries are installed the “./install.sh” script will copy the executables and some other files to the following locations:

- “fingerprint-gui” and “fingerprint-identifier” to /usr/local/bin/,
- “fingerprint-suid”, “fingerprint-helper” and “fingerprint-plugin” to /usr/local/lib/fingerprint-gui/,
- A “Fingerprint GUI” entry in the “System Settings” menu,
- The plugin “pam_fingerprint-gui.so” to /lib/security/ (/lib64/security/ in Fedora 64bit),
- In case of a detected device from UPEK Inc. or SGS Thomson your choice of “Yes” to the appropriate prompt the library “libbsapi.so” to “/usr/lib”, a configuration file “upek.cfg” to “/etc”, an udev-rules file “91-fingerprint-gui-upek.rules” to “/etc/udev/rules.d” and create a directory “/var/upek_data”.

1.3 Creating a “plugdev” group

While installation the “./install.sh” script will check your “/etc/group” file for the existence of a group named “plugdev”. If it doesn't exist you'll get a warning. In this case please create this group and make all desktop users being members of this group or make sure all users have r/w access to the fingerprint scanner device by a proper setup of your “udev” configuration.

1.4 Uninstalling other Fingerprint Solutions

Because fingerprint-gui can conflict with other fingerprint PAM modules these must be uninstalled. Please make sure there is no “libpam-fprint”, “libpam-fprintd” or ”libpam-thinkfinger” installed.

IMPORTANT: On Fedora 12 you need to uninstall “gdm-plugin-fingerprint” and disable the fingerprint authentication in “system | administration | authentication”.

1.5 Special preparations for Lubuntu

The default display manager (lxdm) of Lubuntu doesn't work with fingerprint login. Please install “gdm” and make it the default display manager. If you want to use the default screensaver (xscreensaver) of Lubuntu please change settings of file “/etc/pam.d/xscreensaver” instead of “/etc/pam.d/gnome-screensaver” below. The setup for an embedded keyboard command is not required in this case.

2 Acquiring Fingerprints

Now you should be able to call “fingerprint-gui” from the command line or use the “Fingerprint GUI” entry in the “System Settings” menu. Acquiring fingerprints should be self-explanatory in the “fingerprint-gui” program. Your fingerprints are stored in a “/var/lib/fingerprint-gui/<your_username>/” directory, where only you have access to. If you give the “--debug” argument to “fingerprint-gui” a lot of debug output is given to syslog (or /var/log/auth.log).

After some users have registered their fingerprints you can test the fingerprint identification by calling “fingerprint-identifier” as root (execute “sudo fingerprint.identifier –debug”). This application can identify your users and print their login names to stdout.

3 Setting up Fingerprint Authentication

You need root permissions to make changes to your PAM configuration. First of all make a copy of your “/etc/pam.d/common-auth” file and name it “/etc/pam.d/common-auth.fingerprint”. Edit this file like follows:

- insert a line “`auth sufficient pam_fingerprint-gui.so --debug`” **as the first**

line;

- find the line containing “pam_unix.so” and add the argument “try_first_pass” to the call of “pam_unix.so”;

The distributions differ slightly with regard to the filenames and their contents:

Ubuntu 10.04 Desktop:

“/etc/pam.d/common-auth.fingerprint” is a copy of “/etc/pam.d/common-auth”. The changed lines in question read:

```
“auth sufficient  pam_fingerprint-gui.so  --debug”
“auth [success=1 default=ignore]    pam_unix.so try_first_pass nullok_secure”
```

Fedora 12:

“/etc/pam.d/common-auth.fingerprint” is a copy of “/etc/pam.d/system-auth-ac”. The changed lines in question read:

```
“auth sufficient  pam_fingerprint-gui.so --debug”
“auth sufficient  pam_unix.so nullok try_first_pass”
```

If you're finished setting up your “common-auth.fingerprint” file you can setup the services for fingerprint authentication now. It is assumed you have at least one fingerprint registered for your user account and one for root. Also make sure there is set a password for root (sudo passwd root).

The following settings will change the existing reference to “common-auth” (“system-auth” in Fedora) to the new “common-auth.fingerprint” for the PAM services.

IMPORTANT NOTE: The following settings can lock access to your system completely if something goes wrong. So please open a secure tty (ctrl-alt-F2) and login as root there. This way you're able to undo the changes made in “/etc/pam.d/”.

3.1 Configuring “su”

Edit the file “/etc/pam.d/su” and change the line “@include common-auth” to “@include common-auth.fingerprint” (on Ubuntu) or “auth include system-auth” to “auth include common-auth.fingerprint” (on Fedora).

Ubuntu:

```
...
#@include common-auth
@include common-auth.fingerprint
@include common-account
@include common-session
```

Fedora:

```
...
#auth  required  pam_wheel.so  use_uid
auth  include  common-auth.fingerprint
#auth  include  system-auth
account  sufficient  pam_succeed_if.so uid = 0  use_uid  quiet
...
```

Then open a terminal window and call “su”. A password prompt should appear in the terminal **and** the system should open a GUI widget requesting a finger swipe with the message “Authenticating root” in it's status bar. If you can become root by swiping the finger registered for root it works. You

should also be able to become root by ignoring this GUI widget and typing root's password at the prompt.

3.2 Configuring “login”

IMPORTANT: On Fedora 12 SELinux denies access to the user's fingerprint data in “/var/lib/fingerprint-gui/...” while login. Currently I'm not able to setup a SELinux policy for fingerprint-gui. If you can be of assistance about this please contact me. If not, set your SELinux mode to “permissive” at least while testing login.

Edit the file “/etc/pam.d/login” and change the line “@include common-auth” to “@include common-auth.fingerprint” (on Ubuntu) or “auth include system-auth” to “auth include common-auth.fingerprint” (on Fedora). Then change to a secure tty (e.g. ctrl-alt-F3), type the username and press enter. The password prompt should appear along with a message “Type your password or swipe your finger”. You should be able to login with a finger swipe and with typing the password as well.

3.3 Configuring “sudo”

Edit the file “/etc/pam.d/sudo” and change the line “@common-auth” to “@common-auth.fingerprint” (on Ubuntu) or “auth include system-auth” to “auth include common-auth.fingerprint” (on Fedora). Make sure your login name is in the sudoers file. Then open a terminal window and call “sudo gnome-terminal”. After swiping your finger the gnome-terminal should open with root permissions.

3.4 Configuring “gdm”

In order to be able to login into a desktop session you need to configure your gdm (probably with gdmsetup). Disable “autologin”, “timed login” and “userlist”. Use the command (this is one line!) to disable the userlist:

```
sudo gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.defaults --type bool --set /apps/gdm/simple-
greeter/disable_user_list true
```

Then double check you have a root session on a secure tty open (for undoing the changes if something goes wrong).

On Ubuntu edit the file “/etc/pam.d/gdm” and change the line “@include common-auth” to “@include common-auth.fingerprint”.

On Kubuntu edit the file “/etc/pam.d/kdm” and change the line “@include common-auth” to “@include common-auth.fingerprint” and move this line to the beginning of the file. Then start “System settings | Advanced" and open the “Convenience” tab. Disable “Enable Auto-login” and “Focus password” and set “Previous” as the default user for login. You can then login with your fingerprint after pressing <enter> in the kdm greeter.

On Fedora edit the file “/etc/pam.d/gdm-password” and change the line “auth substack system-auth” to “auth substack common-auth.fingerprint”.

If there is a line reading “auth requisite pam_nologin.so” **comment this line out or remove it**. Now logout from your gnome session. The gdm greeter should show a login prompt **and** the GUI widget requesting a finger swipe below. You should be able to login with fingerprint and with name/password as well.

3.5 Configuring "gnome-screensaver"

Gnome-screensaver needs a plugin to display the fingerprint GUI widget to the user while unlocking. To start this plugin with the gnome-screensaver-dialog open the gconf-editor, find the "apps | gnome-screensaver" entry and **enable** the "embedded_keyboard_enabled" item. Then invoke the string "/usr/local/lib/fingerprint-gui/fingerprint-plugin -d" as the "/apps/gnome-screensaver/embedded_keyboard_command" and close gconf-editor. **This step needs to be taken by every user who wants to unlock his/her gnome-screensaver by fingerprint on that machine!**

Then edit the file "/etc/pam.d/gnome-screensaver" change the line "@include common-auth" to "@include common-auth.fingerprint" (on Ubuntu) or "auth include system-auth" to "auth include common-auth.fingerprint" (on Fedora). Double check you have a root session on a secure tty open (for undoing the changes if something goes wrong) before testing. You can now lock your screen and should be able to unlock it with a fingerswipe or with your password.

For setting up the screensaver in Lubuntu please refer to "Special preparations for Lubuntu" above.

4 Exporting Fingerprint Data and Testing PAM Settings

With "fingerprint-gui" ("Settings" Tab) users can export their fingerprint data (bir files) and test the PAM settings of the current machine for proper setup for fingerprint authentication.

With the "Export now" button all data stored for this user (in /var/lib/fingerprint-gui/<username>/) are exported to a file "Fingerprints.tar.gz" in the user's home directory.

To test for proper PAM settings the "Test" button can be used. First chose the PAM service to be tested then click the "Test" button. In case of proper settings the fingerprint-helper widget will appear and after a finger swipe the message "Authentication successful" will appear in the text field below. If nothing happens the PAM settings might be invalid. You can press <enter> to abort the test in this case.

5 Password Store

There are applications that need a password for encrypting or decrypting something on your system. Probably gnome-keyring is the most widespread of such applications. Also an **encrypted home directory** needs a password to decrypt when a user logs in. These applications sometimes get their key for decrypting (e.g. for the password safe) by querying the PAM session environment for the password given by the user at login. But when the user was logged in with a fingerprint there is no password stored in the PAM session environment. So the application will prompt the user for a password when needed (e.g. if a wireless WPA connection has to be established by the Gnome Network Manager or if you want to access your email account with Evolution) even if the user was logged in already.

Since version 0.11 of Fingerprint GUI there is a solution: You can use some removable media (USB stick) to save your (encrypted) password there. If the media is connected to your machine while you login with your fingerprint the "pam_fingerprint-gui.so" module can decrypt the password and send it to the PAM session environment.

PLEASE READ CAREFULLY NOW AND USE THIS FEATURE ONLY IF YOU UNDESTAND HOW IT WORKS!

If you use the "Password" tab of "fingerprint-gui" you can chose a directory on some removable media, then type your login password twice and click the "Save" button. The removable media must be mounted and you must have write permission there. This is where "fingerprint-gui" creates a

subdirectory ".fingerprints" and writes a file "<username>@<machinename>.xml" containing the encrypted password. The key for decrypting this password, the path for the "<username>@<machinename>.xml" file and the UUID of the removable media are saved in a file "/var/lib/fingerprint-gui/<username>/config.xml" (probably on your local HDD).

When you login using your fingerprint the "pam_fingerprint-gui.so" module reads the "/var/lib/fingerprint-gui/<username>/config.xml" file, finds the "<username>@<machinename>.xml" file on the removable media (if it is connected and has the given UUID), mounts it, decrypts the password and saves it to the PAM session environment where gnome-keyring or other permitted applications can read it. This avoids your system asking for the password again.

In case of a fingerprint login to a session with an encrypted user home a message "!!!ERROR: FOUND ENCRYPTED HOMEDIR BUT NO PASSWORD!!!" will appear in the gdm greeter and the login by fingerprint will fail, when the external media keeping the encrypted password could not be found.

PLEASE NOTE THE FOLLOWING RESTRICTIONS:

- Do not use this feature if someone other then you has root permissions on this machine. This is because root can connect to the machine via telnet, ssh or something like this, mount the external media, find the "<username>@<machinename>.xml" file, read the "/var/lib/fingerprint-gui/<username>/config.xml" file and decrypt your password.

- Do not connect the removable media if it isn't needed. The "pam_fingerprint-gui.so" module only needs it while login is in progress. It mounts the partition with the given UUID containing the "<username>@<machinename>.xml" file and unmounts it immediately after it has read the file.

- Do never leave the removable media and the computer at the same location unattended. Someone could copy both files and decrypt your password later.

- You don't need to type your password any more so you can use a very long and strong password now. But do not forget your password! You would not be able to unlock your login-keyring any more if your removable media gets lost or corrupted.

- If you change your login password on this machine you need to use "fingerprint-gui" again and save the new password to the removable media.

This is how I use this feature for myself:

My USB stick has 3 partitions: One "vfat" (/dev/sdb1) to keep files to be transferred to other machines, one "luks_crypto" (/dev/sdb2) partition to keep my secret data and a very small (3MB) "ext2" (/dev/sdb3) partition to hold the "<username>@<machinename>.xml" file. Corresponding entries in /etc/fstab ensure that the partitions sdb2 and sdb3 are not automatic mounted. Needless to say that I'm the only person who has root access to my notebook.

While booting my notebook I connect the USB stick until I'm logged in with my fingerprint, then remove the stick immediately and reconnect it only (and only as long as needed!) if I want to copy

something from or to it. Because I don't need to invoke my password any more I use a very strong and cryptic login password.

6 Troubleshooting

6.1 Gdm Greeter doesn't show the Fingerprint GUI Widget or needs a long time (up to 20 seconds) to show it

This behavior was seen on Fedora 12 with SELinux set to “enforcing”. Please set the system default of SELinux to “permissive” (or help me setting up SELinux rules that can be installed with Fingerprint GUI).

6.2 Fingerprint-gui Error “Could not open fingerprint device”

On some systems the file “/etc/udev/rules.d/40-libfprint0.rules” (or something like this in “/lib/udev/rules.d”) installed by the “libfprint” package doesn't work properly. It should help to rename this file so it is invoked at a later time. In all known cases renaming it to “91-libfprint0.rules” solved the problem. **You should also make sure your fingerprint scanner hardware has an entry in this file**.

6.3 Login on a secure tty hangs with “OK” Message

If you try to login on a secure tty the prompt “Swipe your finger or type your password” appears. If you swipe the finger the message “OK” appears and then nothing happens. In this case the “uinput” device doesn't work. Make sure the “uinput” module is loaded (“lsmod | grep uinput”), the device exists in “/dev/input/uinput”, “/dev/misc/uinput” or “/dev/uinput” and you have write permission to it. On Ubuntu add a line “uinput” to the file “/etc/modules” and restart.

6.4 You have a fingerprint device from UPEK/SGS Thomson and get some `"ABSOpen() failed..."` error message in /var/log/auth.log

This is probably a problem with the proprietary UPEK driver (libbsapi.so). Maybe your device needs the "NVM emulation". Please have a look into this document:
http://www.n-view.net/Appliance//fingerprint/BSAPIUsageonLinux.pdf
and try to setup the emulation for your device.

6.5 Password can not be saved to removable media

If you find an entry reading:

```
"AES128-CBC not supported! Provider (libqca-ossl.so) not installed?"
```

in the log files, the plugin library for encryption is missing. Install the “libqca2-plugin-ossl” package (Ubuntu) or a similar encryption plugin.

In other cases make sure the media is removable, contains a valid partition and is mounted with read/write permission.

7 Known Limitations

7.1 Applications that don't use PAM for prompting a password

The normal way to use PAM for authentication is to let the PAM system prompt the user for a username and/or a password. PAM uses then a callback function of the calling application for prompting something in it's own style. If called back by PAM the application can decide how it wants to prompt for name or password; if not called back, PAM has performed the authentication in another way (fingerprint, smart card, iris scanner or whatever). Maybe they didn't understand that or had another reason not to use that mechanism, the developers of some applications decided to prompt for password or username before calling PAM. In this case the “pam_fingerprint-gui.so” plugin is called at a time where the password is already known by the PAM stack and therefore exits immediately. Fingerprint authentication is not possible then.

7.2 Missing XAUTHORITY environment variable

When calling PAM some applications don't have a XAUTHORITY variable in their environment. “pam_fingerprint-gui.so” tries hard to find the “MIT Magic Cookie” to be used to connect to the current display but in some cases it fails. I guess this is in several KDE applications the reason for not being able to show the fingerprint widget. Maybe I'll find some better solution in a later version.

7.3 Other Linux distributions

Debian 4.0

I didn't find any way to install libfprint. There is neither a package available nor do the sources compile without errors. Didn't want to waste more time with it.

SuSE 11.1 (gnome edition)

The gdm used in SuSE behaves totally strange. It doesn't allow to show the fingerprint widget. Maybe it's only some setting to be changed or the original source installation of gdm to be used. Neither found any useful documentation about it nor had the time to try a fresh compiled gdm from sources. I gave up!

Slackware

Slackware might need someone who has enough spare time to make it “PAM aware”. Not me!

So if you are interested to bring Fingerprint GUI to work on some other distributions first read the “Hacking” document of this project for hints about how it works. If you need further information about it contact me. If you managed to make it up and running write a HowTo and let me know.

Ubuntu and Fedora users should have no serious problems; so have fun with it!