Login und Authentifizierung

Login in lokalen Linux-Rechner

Die Linux-Pam und die verschiedenen Session-Manager können mit GnuPG-Card umgehen.

- http://www.schiessle.org/howto/poldi.php

Die dort verwendeten Optionen für poldi-ctrl scheinen in der wheezy-Version noch nicht zu existieren. Folgendes Information hilft weiter:

- http://walter.silvergeeks.com/rechner/howto/how-to-anmeldung-mit-smart-card-oder-usb-token-am-lokalen-linux-system/

Einrichten des Poldi-Moduls:

(alle Kommandos als Root oder mit sudo)

Das Packet muss installiert werden:

```
apt-get install libpam-poldi
```

Unter Debian Jessie findet poldi das Binary von scdaemon nicht mehr. Ein

```
ln -s /usr/lib/gnupg2/scdaemon
/usr/bin/scdaemon
```

als Root schafft einfache Abhilfe.

Ein

```
poldi-ctrl -d
```

testet, ob die Karte erkannt wird.

Mit

```
poldi-ctrl -s
```

finden wir die Seriennummer der Karte heraus (`gpg –card-status` würde auch funktionieren).

Jetzt müssen wir unter /etc/poldi/localdb/users eine Kartei der User anlegen, für die die Authentifizierung mit der jeweiligen Karte eingerichtet werden soll. Z.B. mit:

```
echo "`poldi-ctrl -s` USERNAME" | tee -a /etc/poldi/localdb/users
```

Unter /etc/poldi/localdb/keys/ muss jetzt für jede Karte eine Datei mit der Seriennummer als Dateiname angelegt werden, die den public Key der Karte enthält. Das geht einfach mit:

```
poldi-ctrl -s > serialno.txt
poldi-ctrl -k | tee -a /etc/poldi/localdb/keys/$(cat serialno.txt)
```

Damit sollte Poldi fertig konfiguriert sein.

Einrichten der Authentifizierung

Nach der grundlegenden Einrichtung der Smartcard, kann das Pam-Poldi-Modul zur /etc/pam.d/common-auth hinzugefügt werden. Alle Authentifikationen, die auf die common-auth zurückgreifen (u.a. xscreensaver, sudo) können dann auch via smartcard authentifizieren. Wenn das Modul am Anfang der common-auth hinzugefügt wird und alle anderen Einträge bleiben, ist die Gnupg Card ein **zusätzliches** Authentifizierungsmittel. Bitte immer testen, so lange noch eine offene Konsole verfügbar ist, um die Änderungen ggf. zu korrigieren. Sonst sperrt mensch sich leicht aus.

```
--- a/pam.d/common-auth
+++ b/pam.d/common-auth
@@ -13,6 +13,8 @@
 # pam-auth-update to manage selection of other modules.  See
 # pam-auth-update(8) for details.

+# with smartcard
+auth sufficient pam_poldi.so
 # here are the per-package modules (the "Primary" block)
 auth   [success=1 default=ignore]      pam_unix.so nullok_secure
 # here's the fallback if no module succeeds
```

Login und Authentifizierung

Login in lokalen Linux-Rechner

Die Linux-Pam und die verschiedenen Session-Manager können mit GnuPG-Card umgehen.

- http://www.schiessle.org/howto/poldi.php

Die dort verwendeten Optionen für poldi-ctrl scheinen in der wheezy-Version noch nicht zu existieren. Folgendes Information hilft weiter:

- http://walter.silvergeeks.com/rechner/howto/how-to-anmeldung-mit-smart-card-oder-usb-token-am-lokalen-linux-system/

Einrichten des Poldi-Moduls:

(alle Kommandos als Root oder mit sudo)

Das Packet muss installiert werden:

```
apt-get install libpam-poldi
```

Unter Debian Jessie findet poldi das Binary von scdaemon nicht mehr. Ein

```
ln -s /usr/lib/gnupg2/scdaemon /usr/bin/scdaemon
```

als Root schafft einfache Abhilfe.

Ein

```
poldi-ctrl -d
```

testet, ob die Karte erkannt wird.

Mit

```
poldi-ctrl -s
```

finden wir die Seriennummer der Karte heraus (`gpg –card-status` würde auch funktionieren).

Jetzt müssen wir unter /etc/poldi/localdb/users eine Kartei der User anlegen, für die die Authentifizierung mit der jeweiligen Karte eingerichtet werden soll. Z.B. mit:

```
echo "`poldi-ctrl -s` USERNAME" | tee -a /etc/poldi/localdb/users
```

Unter /etc/poldi/localdb/keys/ muss jetzt für jede Karte eine Datei mit der Seriennummer als Dateiname angelegt werden, die den public Key der Karte enthält. Das geht einfach mit:

```
poldi-ctrl -s > serialno.txt
poldi-ctrl -k | tee -a /etc/poldi/localdb/keys/$(cat serialno.txt)
```

Damit sollte Poldi fertig konfiguriert sein.

Einrichten der Authentifizierung

Nach der grundlegenden Einrichtung der Smartcard, kann das Pam-Poldi-Modul zur /etc/pam.d/common-auth hinzugefügt werden. Alle Authentifikationen, die auf die common-auth zurückgreifen (u.a. xscreensaver, sudo) können dann auch via smartcard authentifizieren. Wenn das

Modul am Anfang der common-auth hinzugefügt wird und alle anderen Einträge bleiben, ist die Gnupg Card ein **zusätzliches** Authentifizierungsmittel. Bitte immer testen, so lange noch eine offene Konsole verfügbar ist, um die Änderungen ggf. zu korrigieren. Sonst sperrt mensch sich leicht aus.

```
--- a/pam.d/common-auth
+++ b/pam.d/common-auth
@@ -13,6 +13,8 @@
 # pam-auth-update to manage selection of other modules.  See
 # pam-auth-update(8) for details.

+# with smartcard
+auth sufficient pam_poldi.so
 # here are the per-package modules (the "Primary" block)
 auth   [success=1 default=ignore]      pam_unix.so nullok_secure
 # here's the fallback if no module succeeds
```

Nutzung für ssh

Leider hat sich die Interaktion zwischen gnome-keyring-daemon, gpg-agent und ssh-agent in Debian stretch wiederum geändert. Hinweise dazu gibt es hier:

- https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=850982
- https://blog.josefsson.org/2017/06/19/openpgp-smartcard-under-gnome-on-debian-9-0-stretch/

In Stretch ist jetzt die user-Session von systemd verantwortlich für den Start des gpg-agent. Die verschiedenen Anleitungen, den gnome-keyring-daemon daran zu hindern, sich für ssh verantwortlich zu fühlen, funktionieren in unseren Setups (unter Mate und Xfce4 allerdings nicht. Der entscheidende Hinweise steht unter `/usr/share/doc/gnupg-agent/README.Debian`

Für xfce und mate muss daher ein

```
SSH_AUTH_SOCK=/run/user/$(id -u)/gnupg/S.gpg-agent.ssh
```

in die ~/.bashrc geschrieben werden. Damit wird die Umgebungsvariable gesetzt, mit dem ssh dann den gpg-agent findet, der für die Authentifizierung zuständig sein soll.

- http://www.ozonesolutions.com/programming/2014/04/pgp-smart-card-ssh-login-gpg-agent-ubuntu/
- http://www.bootc.net/archives/2013/06/09/my-perfect-gnupg-ssh-agent-setup/

- https://grepular.com/Smart_Cards_and_SSH_Authentication

Wir müssen ssh-Support im gpg-agent konfigurieren:

```
echo "enable-ssh-support" >> ~/.gnupg/gpg-agent.conf
```

Gleichzeitig soll verhindert werden, dass der ssh-agent beim Starten einer Desktop-Umgebung startet. Dazu editieren wir die `/etc/X11/Xsession.options` und kommentieren

```
#use-ssh-agent
```

aus.

Konflikt mit Gnome-Keyring Manager

In Gnome-Desktops (auch Mate, Cinnamon), gibt es einen Konflikt zwischen dem gpg-agent und dem gnome-keyring Manager, in dessen Folge der gpg-Agent nicht startet.

Um dem Gnome-Keyring-Manager abzugewöhnen, sich für ssh und gpg verantwortlich zu fühlen, benennen wir die Dateien:

```
/etc/xdg/autostart/gnome-keyring-ssh.desktop
/etc/xdg/autostart/gnome-keyring-gpg.desktop
```

jeweils um, in dem wir z.B. ein `.disable` anhängen. Wir könnten sie theoretisch auch löschen.

In manchen Howtos wird empfohlen

```
apt-get remove libpam-gnome-keyring
```

zu deinstallieren. Das scheint nicht zwingend notwendig.

Auch ein

```
gconftool-2 --set -t bool /apps/gnome-keyring/daemon-component/ssh true
```

wird in anderen Howtos genannt, scheint aber nicht in allen Umgebungen nötig zu sein.

ssh über mehrere Server (ForwardAgent)

- http://livecipher.blogspot.co.uk/2013/02/ssh-agent-forwarding.html

Damit das funktioniert, muss der ssh-client so konfiguriert werden, dass er „agent-forwarding“ untestützt:

- /etc/ssh/ssh_config (sytemweit)
- /home/user/.ssh/config (per user)

```
Host *
```

```
    ForwardAgent yes
```

Eine ausführliche Beschreibung findet sich hier:

- http://www.unixwiz.net/techtips/ssh-agent-forwarding.html

Innerhalb von screen-sessions

Hier steht was dazu:

- http://www.bootc.net/archives/2013/06/09/my-perfect-gnupg-ssh-agent-setup/

Aber mit debian-wheezy sollte das out of the box funktinieren.

Konflikt mit Gnome-Keyring Manager

In Gnome-Desktops (auch Mate, Cinnamon), gibt es einen Konflikt zwischen dem gpg-agent und dem gnome-keyring Manager, in dessen Folge der gpg-Agent nicht startet.

Um dem Gnome-Keyring-Manager abzugewöhnen, sich für ssh und gpg verantwortlich zu fühlen, benennen wir die Dateien:

```
/etc/xdg/autostart/gnome-keyring-ssh.desktop
/etc/xdg/autostart/gnome-keyring-gpg.desktop
```

jeweils um, in dem wir z.B. ein `.disable` anhängen. Wir könnten sie theoretisch auch löschen.

In manchen Howtos wird empfohlen

```
apt-get remove libpam-gnome-keyring
```

zu deinstallieren. Das scheint nicht zwingend notwendig.

Auch ein

```
gconftool-2 --set -t bool /apps/gnome-keyring/daemon-component/ssh true
```

wird in anderen Howtos genannt, scheint aber nicht in allen Umgebungen nötig zu sein.

ssh über mehrere Server (ForwardAgent)

- http://livecipher.blogspot.co.uk/2013/02/ssh-agent-forwarding.html

Damit das funktioniert, muss der ssh-client so konfiguriert werden, dass er „agent-forwarding“ untestützt:

- /etc/ssh/ssh_config (sytemweit)
- /home/user/.ssh/config (per user)

```
Host *
      ForwardAgent yes
```

Eine ausführliche Beschreibung findet sich hier:

- http://www.unixwiz.net/techtips/ssh-agent-forwarding.html

Innerhalb von screen-sessions

Hier steht was dazu:

- http://www.bootc.net/archives/2013/06/09/my-perfect-gnupg-ssh-agent-setup/

Aber mit debian-wheezy sollte das out of the box funktinieren.