Локальная аутентификация по Рутокен ЭЦП в Fedora

Перед началом работы, установите следующие пакеты:

```
sudo dnf update
sudo dnf install ccid opensc pcsc-tools p11-kit nss-tools python3-tkinter rpmdevtools libsss_sudo krb5-pkinit
dialog openssl fedora-packager rpmdevtools gcc vim-common openssl-pkcs11 docbook-style-xsl openldap-devel
openssl-devel pam-devel pcsc-lite-devel pkgconf automake autoconf git libtool
```

Загрузите модуль librtpkcs11ecp.so и установите:

```
sudo rpm -i librtpkcs11ecp-X.X.X.X-X.x86_64.rpm
```

Установка pam_pkcs11

Для этого необходимо выполнить следующие действия:

```
git clone https://github.com/OpenSC/pam_pkcs11.git
cd pam_pkcs11
autoreconf -i
./configure --prefix=/usr/ && make && sudo make install
```

Для конфигурации pam_pkcs11 создайте папки /etc/pam_pkcs11/crls и /etc/pam_pkcs11/cacerts

```
sudo mkdir /etc/pam_pkcs11
sudo mkdir /etc/pam_pkcs11/crls
sudo mkdir /etc/pam_pkcs11/cacerts
```

После установки необходимых пакетов, вы можете воспользоваться графической утилитой для работы с Рутокенами в Linux для упрощённой настройки.

Создание ключей и сертификатов

Проверьте наличие libpkcs11.so по пути: `/usr/lib64/engines-3/`. Если ее нет, то для начала установите libpkcs11.so для того, чтобы OpenSSL смог общаться к Рутокеном.

Для этого скачайте архив библиотеки libp11-X.Y.Z.tar.gz.

Разархивируйте пакет и перейдите в новую папку.

```
tar xvzf libp11-X.Y.Z.tar.gz

cd libp11-X.Y.Z
```

Для установки, введите следующую команду:

```
./configure && make && sudo make install
```

Вы можете пропустить данный раздел, если у вас уже имеются необходимые ключи.
Если ключей нет, ниже команда для их созданию:

```
pkcs11-tool --module /usr/lib64/librtpkcs11ecp.so --keypairgen --key-type rsa:2048 -l --id 45
```

Параметр id задает идентификатор ключевой пары.

Создание сертификата и импорт его на токен через OpenSSL 3.x:

Для работы с pkcs11 engine необходимо сделать следующее:

Создать файл конфигурации engine.conf со следующим содержимым:

```
openssl_conf = openssl_init

[openssl_init]
engines = engine_section

[engine_section]
pkcs11 = pkcs11_section

[pkcs11_section]
engine_id = pkcs11
dynamic_path = /usr/lib64/engines-3/pkcs11.so
MODULE_PATH = /usr/lib64/librtpkcs11ecp.so
default_algorithms = ALL
```

При необходимости использовать pkcs11 engine указывать путь к файлу конфигурации engine.conf, например:

```
OPENSSL_CONF=/path/to/engine.conf openssl req -engine pkcs11 -x509 -new -key 0:45 -keyform engine -out cert.crt
-subj "/CN=test/C=RU/ST=Moscow/L=Moscow/O=Aktiv/OU=dev/emailAddress=testuser@mail.com"
```

Сохраните сертификат на токене:

```
pkcs11-tool --module /usr/lib64/librtpkcs11ecp.so -l -y cert -w cert.crt --id 45
```

Проверьте, что токен подключен и на нем сохранены сертификаты и ключи.

Добавление сертификата в список доверенных

Создайте базу данных доверенных сертификатов

```
sudo mkdir /etc/pam_pkcs11/nssdb

sudo chmod 0644 /etc/pam_pkcs11/nssdb

sudo certutil -d /etc/pam_pkcs11/nssdb -N #

sudo modutil -dbdir /etc/pam_pkcs11/nssdb/ -add p11-kit-trust -libfile /usr/lib64/pkcs11/p11-kit-trust.so
```

Выгрузите ваш сертификат с токена (если вы пользовались для получения сертификата вышеописанной инструкцией , то ID = 45):

```
pkcs11-tool --module=/usr/lib64/librtpkcs11ecp.so -l -r -y cert -d <ID> -o cert.crt
```

Добавьте сертификат в доверенные:

```
sudo cp cert.crt /etc/pki/ca-trust/source/anchors/ #    ,

sudo update-ca-trust force-enable

sudo update-ca-trust extract #
```

Настройка pam_pkcs11

Создайте текстовый файл /etc/pam_pkcs11/pam_pkcs11.conf со следующим содержимым:

```
pam_pkcs11 {
  nullok = false;
  debug = false;
  use_first_pass = false;
  use_authtok = false;
  card_only = false;
  wait_for_card = false;
  use_pkcs11_module = rutokenecp;

  # Aktiv Rutoken ECP
  pkcs11_module rutokenecp {
    module = /usr/lib64/librtpkcs11ecp.so;
    slot_num = 0;
    support_thread = true;
    ca_dir = /etc/pam_pkcs11/cacerts;
    crl_dir = /etc/pam_pkcs11/crls;
    cert_policy = signature;
  }

  use_mappers = digest;

  mapper_search_path = /usr/lib64/pam_pkcs11;

  mapper digest {
   debug = false;
   module = internal;
   algorithm = "sha1";
   mapfile = file:///etc/pam_pkcs11/digest_mapping;
  }

}
```

Регистрация модуля PAM PKCS11 для аутентификации в системе

Подключите модуль к системе авторизации PAM:

```
sudo nano /etc/pam.d/system-auth
#
sudo nano /etc/pam.d/password-auth
```

Перед первым использованием модуля pam_unix добавьте туда строку со следующим содержимым:

```
auth   sufficient             pam_pkcs11.so pkcs11_module=/usr/lib64/librtpkcs11ecp.so
```

Сохраните файл и узнайте поля вашего сертификата с помощью следующей команды:

```
sudo pkcs11_inspect
```

В результате отобразится сообщение:

```
[user@fedora ~]$ sudo pkcs11_inspect
PIN for token:
Printing data for mapper digest:
CB:13:CA:34:AC:04:CD:BF:A6:17:29:2F:C8:00:6A:D5:54:B8:0B:BB
```

Скопируйте строчку с описанием сертификата в файл /etc/pam_pkcs11/digest_mapping в формате:

```
<  pkcs11_inspect> -> <_>
```

Пример заполнения файла:

```
[user@fedora ~]$ sudo cat /etc/pam_pkcs11/digest_mapping
CB:13:CA:34:AC:04:CD:BF:A6:17:29:2F:C8:00:6A:D5:54:B8:0B:BB ->
user
```

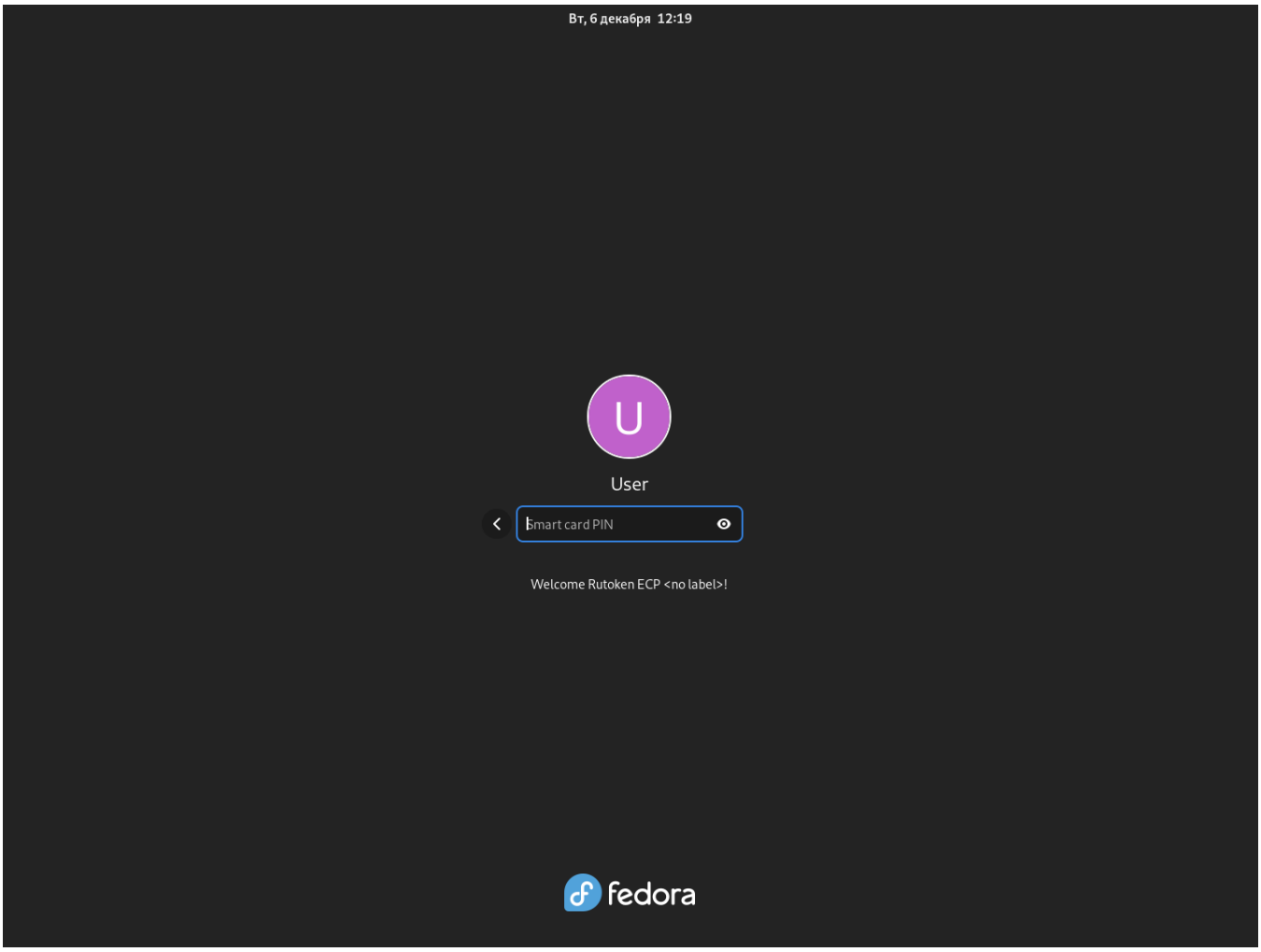
Попробуйте аутентифицироваться:

```
su <username>
```

Терминал должен запросить PIN код рутокена:

```
[user@fedora ~]$ su user
Smart card found.
  Rutoken ECP <no
label>!
Smart card PIN:
verifying certificate
Checking signature
[user@fedora ~]$
```

В окне экрана приветствия аналогично:



Настройка автоблокировки

В состав пакета libpam-pkcs11 входит утилита pkcs11_eventmgr, которая позволяет выполнять различные действия при возникновении событий PKCS#11.

Для того, чтобы аутентификация корректно работала на лок скрине. В настройках pkcs11_eventmgr нужно указать название сервиса, использующегося при аутентификации через лок скрин, чтобы сделать его доверенным. У каждой графической оболочки свое название данного сервиса. Узнать название вашей графической оболочки можно с помощью команды:

Название графической оболочки

```
echo $XDG_CURRENT_DESKTOP
```

Вот список соответствий названий графических оболочек и сервиса, используемого лок скрином. Данный список не является полным.

MATE → mate-screensaver
X-Cinnamon → cinnamon-screensaver
fly → <Отсутствует>
KDE → kde
GNOME → xdg-screensaver

Для настройки pkcs11_eventmgr служит файл конфигурации - /etc/pam_pkcs11/pkcs11_eventmgr.conf

Пример файла конфигурации представлен ниже:

```
pkcs11_eventmgr
{
    #
    daemon = true;

    #
    debug = false;

    #
    polling_time = 1;

    #  -
    # - 0
    expire_time = 0;

    #  pkcs11
    pkcs11_module = /usr/lib64/librtpkcs11ecp.so;

    #
    #  :
    event card_insert {
        #     (  )
        on_error = ignore ;

        action = "/bin/false";
    }

    #
    event card_remove {
        on_error = ignore;

        #
        action = "xdg-screensaver lock";
    }

    #
    event expire_time {
        #     (  )
        on_error = ignore;

        action = "/bin/false";
    }
}
```

После этого добавьте приложение pkcs11_eventmgr в автозагрузку и перезагрузите компьютер.

Для этого создайте папку ~/.config/autostart. И в данной директории создайте файл ~/.config/autostart/smartcard-screensaver.desktop

```
sudo mkdir ~/.config/autostart

sudo nano ~/.config/autostart/smartcard-screensaver.desktop
```

Содержание файла smartcard-screensaver.desktop должно быть следующим:

```
[Desktop Entry]
Type=Application
Name=Smart Card Screensaver
Comment=Application to lock screen on smart card removal.
Exec=/usr/bin/pkcs11_eventmgr daemon
```