Локальная аутентификация по Рутокен с PAM и librtpkcs11ecp

Введение

В данной инструкции описывается, как заставить модуль `pam_p11` работать с библиотекой `librtpkcs11ecp.so`.

Проверка модели устройства

1. Подключите USB-токен к компьютеру.
2. Для определения названия модели USB-токена откройте **Терминал** и введите команду:

```
$ lsusb
```

В результате в окне Терминала отобразится название модели USB-токена:

```
[dmitrieva@localhost ~]$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 004: ID 0a89:0030 Aktiv Rutoken ECP
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

Убедитесь, что используете: **Aktiv Rutoken ECP**

Стенд

Нам понадобится Рутокен ЭЦП, отформатированный через Панель управления Рутокен. В качестве дистрибутива использовалась Ubuntu.

Общий порядок действий

1. Устанавливаем необходимые пакеты:

```
$ sudo apt-get install opensc  libpam-p11 libengine-pkcs11-openssl
```

При установке `opensc` также устанавливаются пакеты `libccid` и `pcscd`, а при установке `libpam-p11` – пакет `libp11-2`.

2. Устанавливаем библиотеку PKCS#11 Рутокен, предварительно загрузив установочный пакет с сайта rutoken.ru.

3. Создаем файл описания модуля PKCS#11

```
$ sudo nano /usr/share/p11-kit/modules/Rutoken.module
```

добавляем в редакторе строку:

module:/usr/lib/librtpkcs11ecp.so

и сохраняем файл.

4. Аналогично действиям в приведенной выше статье создаем файл `/usr/share/pam-configs/p11`, с единственным отличием – укажем путь к нашей библиотеке:

```
Name: Pam_p11
Default: yes
Priority: 800
Auth-Type: Primary
Auth: sufficient pam_p11_opensc.so /usr/lib/librtpkcs11ecp.so
```

5. Выполняем команду:

```
$ sudo pam-auth-update
```

В появившемся диалоге выбираем `Pam_p11`.

Использование существующего сертификата

⚠ Если вы желаете использовать сертификат RSA, который уже записан на ваш токен/смарт-карту, то на данном этапе перейдите к указаниям в приложении к данной инструкции (в самом низу данной страницы).

6. Переходим к созданию ключевой пары:

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

```
qwe@ubuntu:~$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so --keypairgen --key-type rsa:2048 -l --id 45
Using slot 0 with a present token (0x0)
Logging in to "Rutoken ECP <no label>".
Please enter User PIN:
Key pair generated:
Private Key Object; RSA
  label:
  ID:         45
  Usage:      decrypt, sign, unwrap
warning: PKCS11 function C_GetAttributeValue(ALWAYS_AUTHENTICATE) failed: rv = CKR_ATTRIBUTE_TYPE_INVALID (0x12)

Public Key Object; RSA 2048 bits
  label:
  ID:         45
  Usage:      encrypt, verify, wrap
qwe@ubuntu:~$
```

Утилита `pkcs11-tool` входит в состав `opensc`.

Параметры, задаваемые в этой строке:

`--module <arg>`	путь к библиотеке pkcs11 (обязательный параметр)
`--keypairgen`	генерация ключевой пары
`-- key-type <arg>`	задает тип и длину ключа. В нашем случае тип – rsa, длина - 2048 бит (с длиной ключа 1024 бит возникают проблемы)
`-l`	запрос PIN-кода токена до каких-либо операций с ним (обязательный параметр)
`--id <arg>`	определяет id создаваемого объекта (понадобится при создании сертификата)

7. Переходим к созданию сертификата.

Запускаем `openssl` и подгружаем модуль поддержки pkcs11:

```
$ openssl
OpenSSL>  engine dynamic -pre SO_PATH:/usr/lib/x86_64-linux-gnu/engines-1.1/pkcs11.so -pre ID:pkcs11 -pre
LIST_ADD:1  -pre LOAD -pre MODULE_PATH:/usr/lib/librtpkcs11ecp.so
```

```
qwe@ubuntu:~$ openssl
OpenSSL> engine dynamic -pre SO_PATH:/usr/lib/engines/engine_pkcs11.so -pre ID:pkcs11 -pre LIST_ADD:1 -pre LOAD -pre MODULE_PATH:/usr/lib/librtpkcs11ecp.so
(dynamic) Dynamic engine loading support
[Success]: SO_PATH:/usr/lib/engines/engine_pkcs11.so
[Success]: ID:pkcs11
[Success]: LIST_ADD:1
[Success]: LOAD
[Success]: MODULE_PATH:/usr/lib/librtpkcs11ecp.so
Loaded: (pkcs11) pkcs11 engine
OpenSSL>
```

8. Создаем сертификат в PEM-формате:

```
OpenSSL>  req -engine pkcs11 -new -key 0:45 -keyform engine -x509 -out cert.pem  -text
```

```
OpenSSL> req -engine pkcs11 -new -key 0:45 -keyform engine -x509 -out cert.pem -text
engine "pkcs11" set.
PKCS#11 token PIN:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:RU
State or Province Name (full name) [Some-State]:Russia
Locality Name (eg, city) []:Moscow
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Aktiv
Organizational Unit Name (eg, section) []:Aktiv
Common Name (e.g. server FQDN or YOUR name) []:qwe
Email Address []:qwe
OpenSSL>
```

Здесь:

`-key`	указывает закрытый ключ (в нашем случае 0:45 – слот:ID ключа)
`-x509`	выдает самоподписанный сертификат

9. Конвертируем сертификат PEM в CRT:

```
OpenSSL> x509 -in cert.pem -out cert.crt -outform DER
```

10. Закрываем `openssl`. Теперь сохраняем сертификат CRT на Рутокен:

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

```
qwe@ubuntu:~$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -l -y cert -w cert.crt --id 45
Using slot 0 with a present token (0x0)
Logging in to "Rutoken ECP <no label>".
Please enter User PIN:
Created certificate:
Certificate Object, type = X.509 cert
  label:
  ID:         45
```

Здесь:

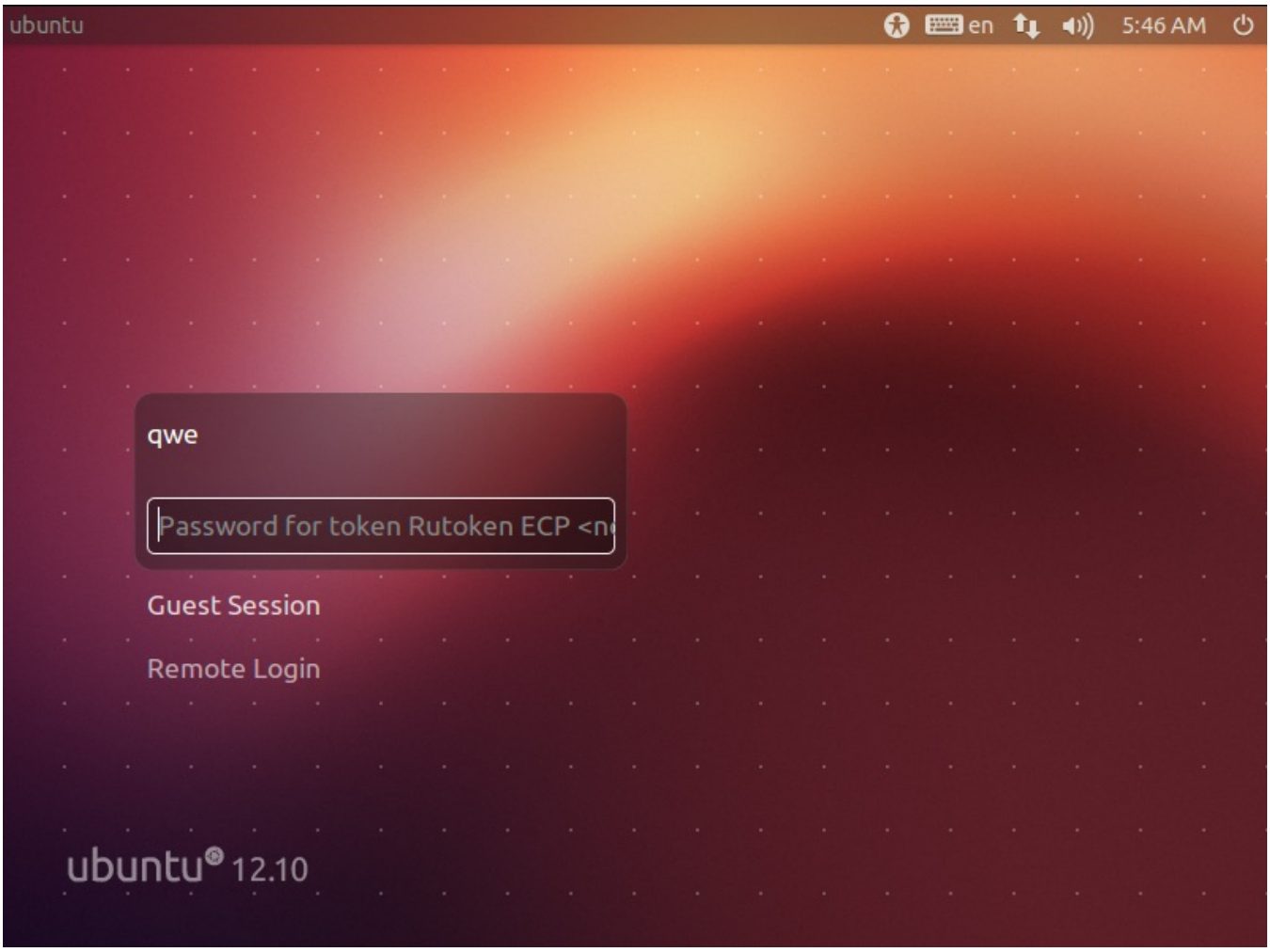
`-y <arg>`	тип объекта (может быть `cert, privkey, pubkey, data`)

`-w <arg>`	записать объект на токен

11. Остается только добавить сертификат PEM в список доверенных:

```
$ mkdir ~/.eid
$ chmod 0755 ~/.eid
$ less ~/cert.pem >> ~/.eid/authorized_certificates
$ chmod 0644 ~/.eid/authorized_certificates
```

12. На этом настройка закончена. После перезапуска ОС окно входа в систему будет выглядеть так:



Приложение: настройка аутентификации с имеющимся сертификатом

Если у вас уже имеется выписанная на токен ключевая пара RSA с привязанным к ней сертификатом, то вы можете использовать их для аутентификации в Ubuntu.

Рекомендуемая длина ключа RSA - не ниже 2048 бит.

1. Выполните пп.1-4 основной инструкции.
2. Необходимо узнать ID сертификата, записанного на токен.

   ```
   pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -O
   ```

3. Добавим сертификат в список доверенных:

```
$ mkdir ~/.eid
$ chmod 0755 ~/.eid
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so --type cert -r --id ID__ | openssl x509 -inform der
>>  ~/.eid/authorized_certificates
$ chmod 0644 ~/.eid/authorized_certificates
```

4. Выполните перезагрузку.