Локальная аутентификация по Рутокен в Ubuntu/Debian (PAM,PKCS15)

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**

Введение

Pluggable Authentication Modules (PAM, подключаемые модули аутентификации) — это набор разделяемых библиотек, которые позволяют интегрировать различные низкоуровневые методы аутентификации в виде единого высокоуровневого API. Это позволяет предоставить единые механизмы для управления, встраивания прикладных программ в процесс аутентификации.

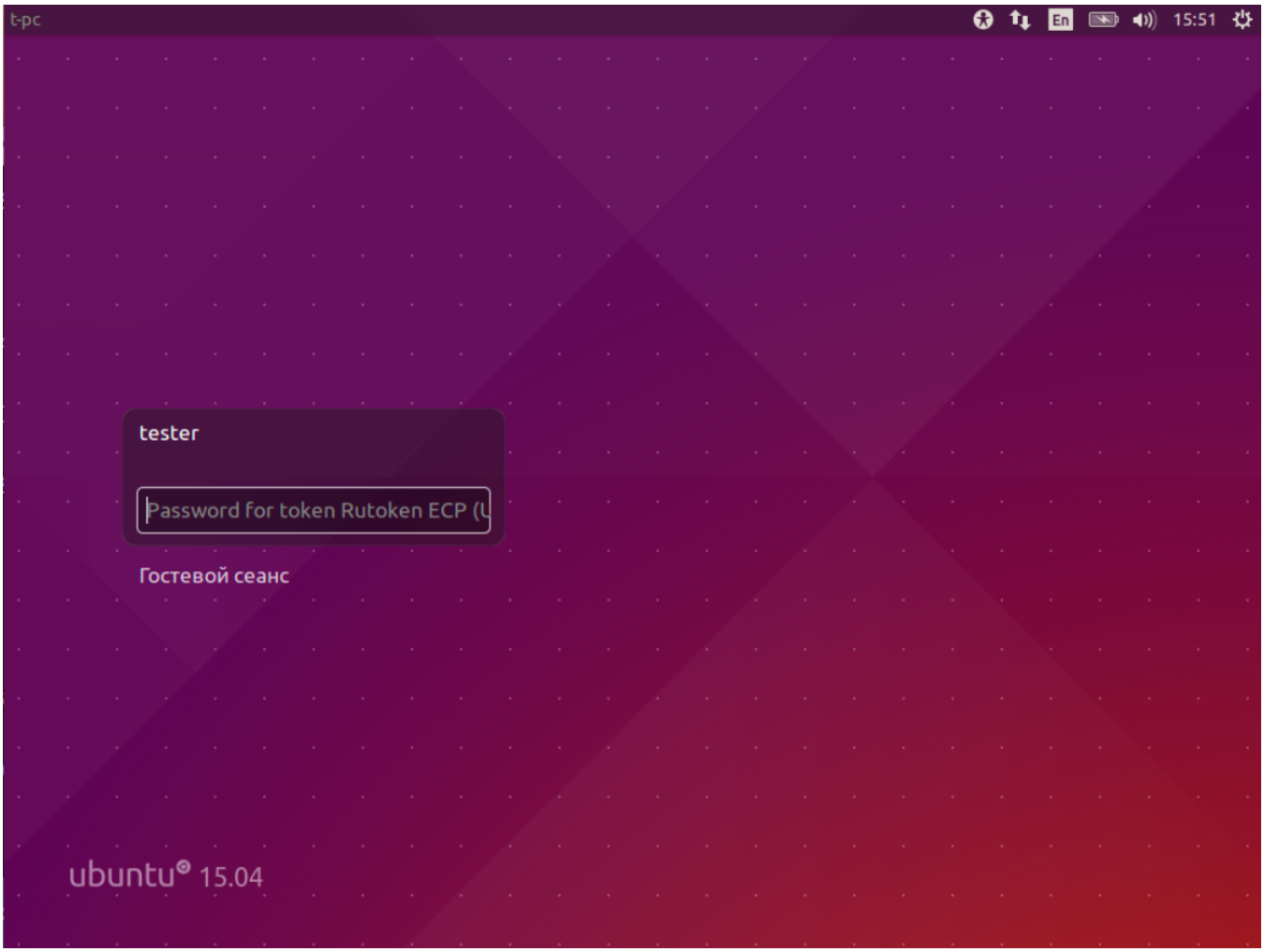
Для PAM существует проект pam_p11, развивающийся как часть OpenSC, позволяющий внедрить аутентификацию по токенам. Доступны два модуля аутентификации:

1. `pam_p11_openssh`: позволяет аутентифицировать пользователя по открытым ключам ssh в `~/.ssh/authorized_keys`
2. `pam_p11_opensc`: аутентификация по сертификатам из файла `~/.eid/authorized_certificates`

Проект OpenSC также предоставляет пакет pam_pkcs11, который представляет собой более гибкий набор модулей аутентификации. Мы рассмотрим конфигурацию pam_p11, как наиболее простого решения. Однако, pam_p11 не имеет понятия о цепочках сертификатов, списках отзыва и OCSP. При необходимости, pam_pkcs11 можно настроить руководствуясь инструкцией (на английском, ее источник). Также имеется инструкция для Gentoo (источник).
Общий порядок действий для настройки PAM следующий:

1. Сгенерировать на токене ключевую пару RSA (проверено, что работает для длины ключа 2048 бит, с 1024 были проблемы)
2. Если требуется сертификат, то с помощью OpenSSL или другого ПО сгенерировать сертификат и записать его на токен
3. Записать открытый ключ или сертификат в необходимый каталог

В итоге выглядит это так:



Предварительная подготовка

Демонстрация работы проводится на Ubuntu 18.04. Описанная последовательность действий актуальна также для других версий Ubuntu и систем, основанных на Debian.

Для конфигурации модуля PAM необходимо установить пакеты:

- `pcscd`
- OpenSC
- OpenSSL
- libpam-p11
- libengine-pkcs11-openssl

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

Пользователям Рутокен S также необходимо установить драйвер с нашего сайта.

Общий порядок действий

Настройка pam_p11

До начала работы с токеном стоит настроить модуль pam_p11:

1. Создать файл `/usr/share/pam-configs/p11` со следующим содержанием:

```
Name: Pam_p11
Default: yes
Priority: 800
Auth-Type: Primary
Auth: sufficient pam_p11_opensc.so /usr/lib/x86_64-linux-gnu/opensc-pkcs11.so
```

Если вы используете не Ubuntu 18.04, вам необходимо проверить местонахождение opensc-pkcs11.so. Он может находится, например, в `/usr/lib/opensc-pkcs11.so`. `find`

2. Обновить конфигурацию PAM:

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

3. В появившемся диалоге необходимо удостовериться, что выбран pam_p11. Если вы хотите отключить аутентификацию по паролям, то можно отключить Unix authentication.

Создание ключей на токене

4. Подготовим токен.

```
$ pkcs15-init -E
$ pkcs15-init --create-pkcs15 --so-pin "87654321" --so-puk ""
$ pkcs15-init --store-pin --label "User PIN" --auth-id 02 --pin "12345678" --puk "" --so-pin "87654321"
--finalize
```

В параметрах pin и so-pin можно указать желаемые пин-коды пользователя и администратора.

5. Создаем ключевую пару RSA длины 2048 бит с ID "45" (id стоит запомнить, он понадобится при создании сертификата). Аутентификация на токене происходит под сущностью пользователя.

```
$ pkcs15-init --generate-key rsa/2048 --auth-id 02 --id 45
< PIN >
```

6. Проверим сгенерированный ключ:

```
$ pkcs15-tool --list-keys
Using reader with a card: Aktiv Rutoken ECP 00 00
Private RSA Key [Private Key]
 Object Flags : [0x3], private, modifiable
 Usage : [0x4], sign
 Access Flags : [0x1D], sensitive, alwaysSensitive, neverExtract, local
 ModLength : 2048
 Key ref : 1 (0x1)
 Native : yes
 Path : 3f0010001000600200001
 Auth ID : 02
 ID : 45
```

Создание сертификата и импорт его на токен

7. Запускаем `openssl`

```
$ sudo openssl
```

8. Подгружаем модуль поддержки pkcs11:

```
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/x86_64-linux-gnu/opensc-pkcs11.so
(dynamic) Dynamic engine loading support
[Success]: SO_PATH:/usr/lib/x86_64-linux-gnu/engines-1.1/pkcs11.so
[Success]: ID:pkcs11
[Success]: LIST_ADD:1
[Success]: LOAD
[Success]: MODULE_PATH:/usr/lib/x86_64-linux-gnu/opensc-pkcs11.so
Loaded: (pkcs11) pkcs11 engine
OpenSSL>
```

Если вы используете не Ubuntu 18.04, вам необходимо проверить местонахождение pkcs11.so. Он может располагаться, например, в /usr /lib/openssl/engines/ . Если его найти не удается воспользуйтесь командой find

9. Создаем самоподписанный сертификат в PEM-формате:

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

Где 0:45 - это пара slot:id (который мы указывали в п.5). OpenSSL предложит ввести PIN-код и заполнить информацию о сертификате. Если у вас возникла ошибка, проверьте, не подключены ли другие USB-токены или считыватели смарт-карт к компьютеру.

10. Проверяем созданный сертификат. В текущем каталоге должен создаться файл самоподписанного сертификата с именем cert.pem. Примечание: если при создании сертификата в OpenSSL убрать ключ `-x509,` то на выходе получим заявку на сертификат.

```
verify -CAfile cert.pem cert.pem
cert.pem: OK
```

Выйдем из OpenSSL.

```
exit
```

11. Сохраняем сертификат на токен:

```
$ pkcs15-init --store-certificate cert.pem --auth-id 02 --id 45 --format pem
< PIN >
```

Занесение сертификата в список доверенных

12. Теперь нам необходимо прочитать с токена сертификат с нужным ID (в нашем случае - 45) и записать его в файл доверенных сертификатов:

```
mkdir ~/.eid
chmod 0755 ~/.eid
pkcs15-tool -r <certificate_id> > ~/.eid/authorized_certificates
chmod 0644 ~/.eid/authorized_certificates
```

Теперь при загрузке Ubuntu мы можем использовать токен для аутентификации.

Примечание

На стадии выбора пользователя информация о подключенном токене может не обновляться динамически. Если вы подключили токен и не видите поля ввода пин-кода, вам может понадобиться перенести фокус на "гостевой сеанс" и обратно на вашего пользователя.