

Вход в Linux по сертификату на JaCarta

Версия ПО: GNU/Linux, ID Protect для linux, OpenSSL, libpam-pkcs11, XCA

Токены: Любые

Проблема:

Настроить вход на рабочую станцию, используя сертификат с закрытым ключом, сгенерированным на стороне смарт-карты.

Решение:

Настройку входа в систему осуществляем при помощи **OpenSSL** и его графического интерфейса **XCA**. Работа на стенде с Debian testing.

Устанавливаем пакеты **pcscd**, **libccid**, **openssl**, **xca**, **libengine-pkcs11-openssl**, **libengine-pkcs11-openssl1.1**, **libpam-pkcs11**

*Замечание: пакет **libengine-pkcs11-openssl** в репозиториях представлен в двух версиях. Устанавливались оба.*

Драйвер для JaCarta

PKI: <https://www.aladdin-rd.ru/support/downloads/989e9d58-8600-4d97-a039-e209c7c8fa8e>

1. Настройка ЦС.

Запускаем xca от имени root.

Замечание: вероятно, можно это делать и без привилегий суперпользователя. Но стенд настраивался именно так.

Переходим на вкладку сертификаты, создаем новый для сервера CA.

X Certificate and Key management <2>

Создание x509 сертификата

Источник | Владелец | Расширения | Область применения ключа | Netscape | Дополнительно

Подписанный запрос

- Использовать подписанный запрос на сертификат
- Копировать расширения из запроса
- Изменить владельца в запросе

Подписание

- Создать самоподписанный сертификат с серийным номером 1
- Use this Certificate for signing

Алгоритм подписи

SHA 256

Шаблон для нового сертификата

[default] CA

Применить расширения | Применить владельца | Применить все

OK | Отмена

На вкладке "Источник" выбираем самоподписанный сертификат, алгоритм подписи SHA 256.

Далее идём на вкладку "Владелец":

Создание x509 сертификата



Источник Владелец Расширения Область применения ключа Netscape Дополнительно

Distinguished name

| | | | |
|---------------------|----------------------|------------------------|---------------------------|
| Внутреннее имя | <input type="text"/> | organizationName | <input type="text"/> |
| countryName | <input type="text"/> | organizationalUnitName | <input type="text"/> |
| stateOrProvinceName | <input type="text"/> | commonName | <u>wks-dbn.aladdin.ru</u> |
| localityName | <input type="text"/> | emailAddress | <input type="text"/> |

| Тип | Содержание |
|-----|------------|
|-----|------------|

Добавить
Удалить

Закрытый ключ


wks-dbn.aladdin.ru (RSA:2048 bit) Отображать уже использованные ключи **Создать новый ключ**

OK Отмена

Заполняем необходимые поля, в поле **commonName** указываем имя хоста.

Здесь же нажимаем "Создать новый ключ".

Новый ключ



Введите название и задайте тип и размер нового ключа

Свойства ключа

| | |
|-------------|---|
| Имя ключа | <input type="text" value="wks-dbn.aladdin.ru"/> |
| Тип ключа | RSA |
| Длина ключа | 2048 bit |

Remember as default

Создать Отмена

Нажимаем "Создать".

Далее переходим на вкладку "Расширения".

Создание x509 сертификата

Источник Владелец **Расширения** Область применения ключа Netscape Дополнительно

X509v3 Basic Constraints

Тип Центр Сертификации

Длина пути Critical

Key identifier

Subject Key Identifier

Authority Key Identifier

Период действия

Не раньше, чем

Не позже, чем

Временной диапазон

Полночь Local time Нет четко определенного срока

X509v3 Subject Alternative Name

X509v3 Issuer Alternative Name

X509v3 CRL Distribution Points

Authority Information Access

В поле "Тип" указываем "Центр сертификации". При необходимости изменяем другие параметры.

Далее нажимаем кнопку "ОК" и убеждаемся, что новый сертификат появился в списке.

Устанавливаем указатель на этот сертификат, нажимаем "Экспорт"

Certificate export

Name

Имя файла

PEM Text format with headers

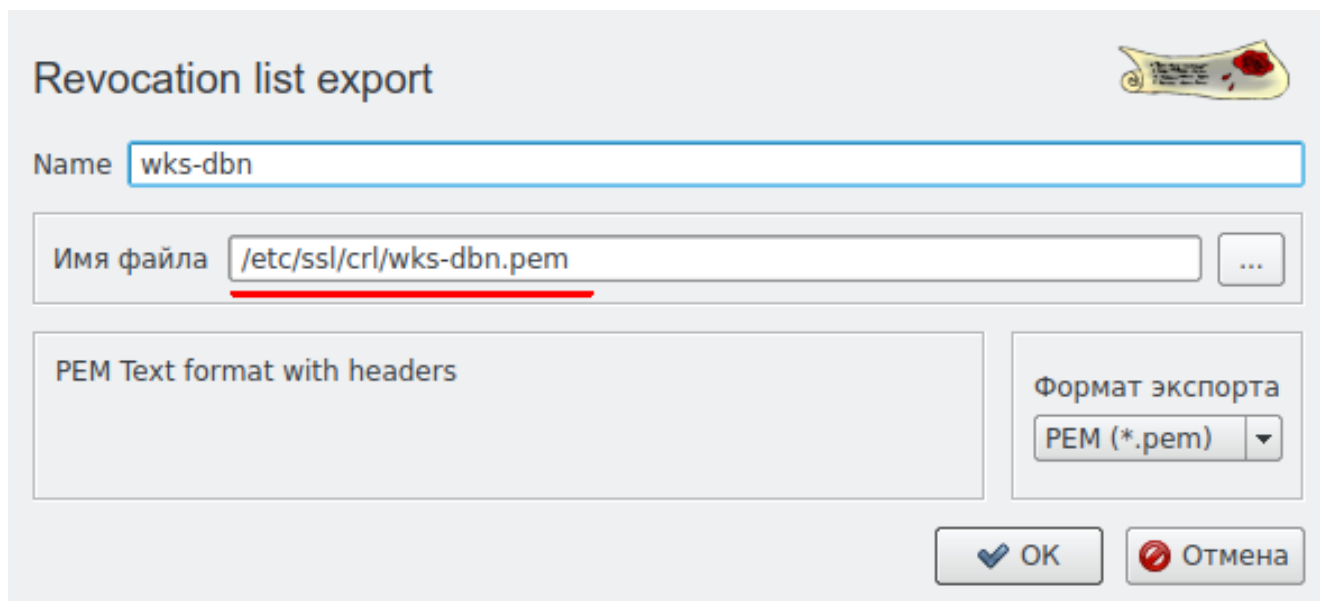
Формат экспорта

Указываем каталог **/etc/ssl**.

Далее ПКМ на сертификате: ЦС -> Создать CRL.

Переходим на вкладку "Списки отозванных сертификатов".

Выделяем появившийся список, нажимаем "Экспорт"



Revocation list export

Name

Имя файла

PEM Text format with headers

Формат экспорта
PEM (*.pem)

OK Отмена

Каталог для экспорта **/etc/ssl/crl**.

Далее создаем хэши для наших данных.

Для этого понадобится скрипт **make_hash_link.sh**. Содержимое можно взять здесь: https://github.com/OpenSC/pam_pkcs11/blob/master/tools/pkcs11_make_hash_link

Создаем файл `make_hash_link.sh`, делаем его исполняемым:

```
sudo chmod +x make_hash_link.sh
```

Далее последовательно выполняем:

```
sudo ./make_hash_link.sh /etc/ssl
```

```
sudo ./make_hash_link.sh /etc/ssl/crl
```

2. Шаблон сертификата.

Возвращаемся в интерфейс ХСА, делаем шаблон для своего сертификата. Переходим на вкладку "Шаблоны", нажимаем "Новый шаблон"

Edit XCA template



Владелец Расширения Область применения ключа Netscape Дополнительно

Distinguished name

| | | | |
|---------------------|---------------------------------------|------------------------|-----------------------------------|
| Внутреннее имя | <input type="text" value="SC-logon"/> | organizationName | <input type="text" value="ARDS"/> |
| countryName | <input type="text" value="RU"/> | organizationalUnitName | <input type="text"/> |
| stateOrProvinceName | <input type="text" value="Moscow"/> | commonName | <input type="text"/> |
| localityName | <input type="text" value="Moscow"/> | emailAddress | <input type="text"/> |

| Тип | Содержание |
|-----|------------|
|-----|------------|

Закрытый ключ

Отображать уже использованные ключи

На вкладке "Владелец" заполняем необходимые поля.

На вкладке "Расширения" указываем "Конечный пользователь"



Владелец Расширения Область применения ключа Netscape Дополнительно

X509v3 Basic Constraints

Тип: Конечный пользователь

Длина пути: Critical

Key identifier

Subject Key Identifier

Authority Key Identifier

Период действия

Не раньше, чем:

Не позже, чем:

Временной диапазон

Полночь Local time Нет четко определенного срока

X509v3 Subject Alternative Name:

X509v3 Issuer Alternative Name:

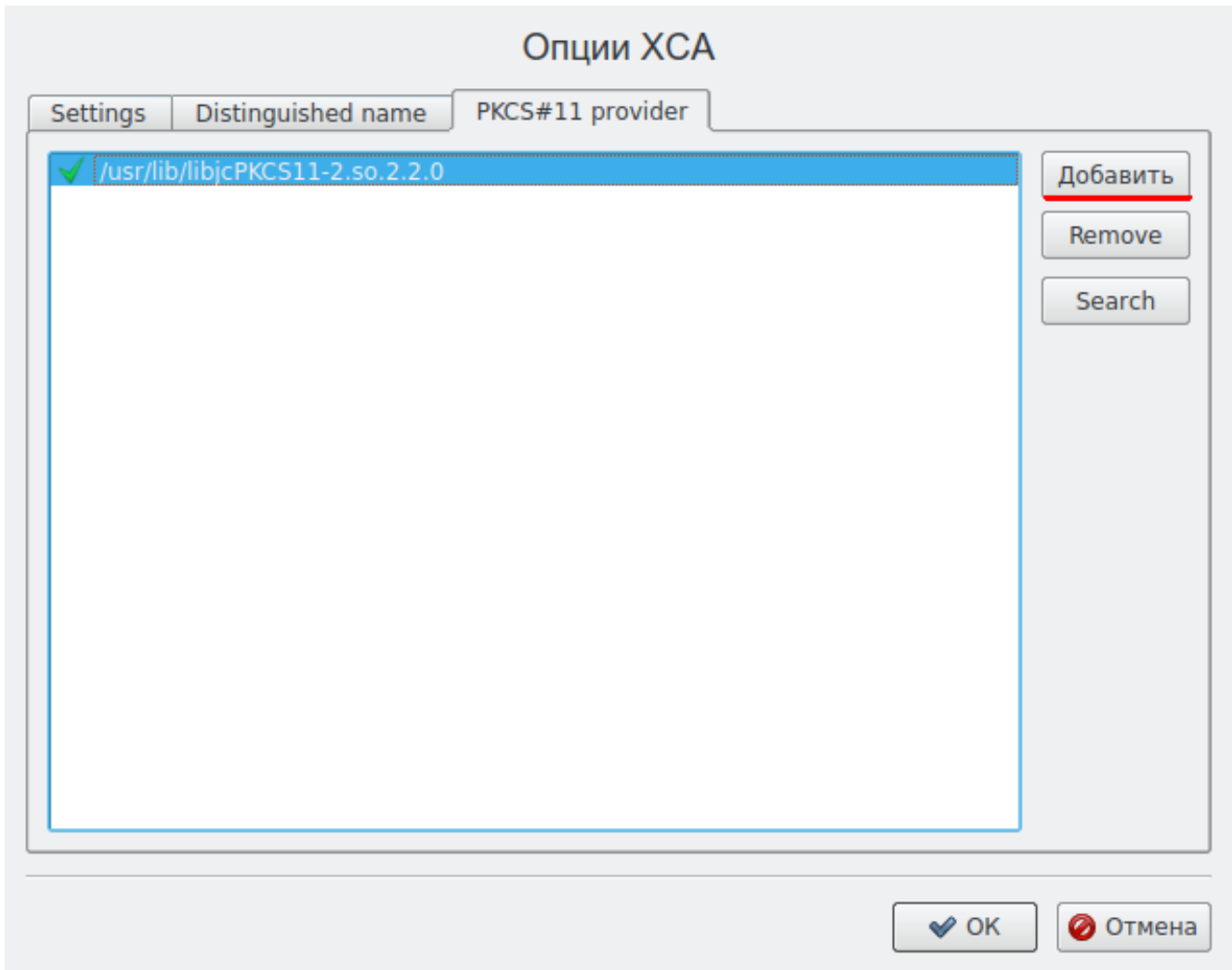
X509v3 CRL Distribution Points:

Authority Information Access:

Нажимаем "OK"

3. Настройка модуля для смарт-карт.

Для того, чтобы можно было выпускать сертификаты с ключевой парой и сертификатами прямо на токене, необходимо добавить модуль **pkcs11** в настройках. В меню Файл -> Опции, вкладка PKCS#11 provider.



Необходимо добавить библиотеку **/usr/lib/libPKCS11.so.2**.

Замечание: пути могут меняться в зависимости от дистрибутива и версий ПО.

4. Выпуск сертификата.

Выпускаем сертификат для пользователя.

Вкладка "Сертификаты", кнопка "Новый сертификат", вкладка "Источник"

Создание x509 сертификата



Источник | Владелец | Расширения | Область применения ключа | Netscape | Дополнительно

Подписанный запрос

- Использовать подписанный запрос на сертификат
- Копировать расширения из запроса
- Изменить владельца в запросе

Показать запрос

Подписание

- Создать самоподписанный сертификат с серийным номером 1
- Use this Certificate for signing wks-dbn

Алгоритм подписи: SHA 256

Шаблон для нового сертификата: SC-logon

Применить расширения | Применить владельца | Применить все

OK | Отмена

Выбираем "Use this Certificate for signing", алгоритм подписи SHA 256, выбираем созданный ранее шаблон. Нажимаем "Применить всё".

Идём на вкладку "Владелец":

Создание x509 сертификата



Источник Владелец Расширения Область применения ключа Netscape Дополнительно

Distinguished name

| | | | |
|---------------------|----------------------|------------------------|----------------------|
| Внутреннее имя | <input type="text"/> | organizationName | ARDS |
| countryName | RU | organizationalUnitName | <input type="text"/> |
| stateOrProvinceName | Moscow | commonName | test |
| localityName | Moscow | emailAddress | <input type="text"/> |

| Тип | Содержание |
|-----|------------|
|-----|------------|

Добавить
Удалить

Закрытый ключ


wks-dbn.aladdin.ru (RSA:2048 bit) Отображать уже использованные ключи Создать новый ключ

OK Отмена

Заполняем необходимые поля. В поле **commonName** указываем имя нужного пользователя.

Далее нажимаем "Создать новый ключ".

Новый ключ



Введите название и задайте тип и размер нового ключа

Свойства ключа

| | |
|-------------|---|
| Имя ключа | test |
| Тип ключа | <u>My token #0C50000427129613 (RSA Key of 1024 - 2048 bits)</u> |
| Длина ключа | 2048 bit |

Remember as default

Создать Отмена

В поле "Тип ключа" выбираем наш токен, алгоритм RSA. Нажимаем "Создать". После ввода пин-кода будет сгенерирована ключевая пара.

Далее нажимаем "ОК", соглашаемся, чтобы сертификат был сохранён на токене.

Теперь этот токен можно использовать для входа в систему.

5. Настройка pam.d и pam_pkcs11.

Для возможности входа по токену в pam.d необходимо добавить модуль pam_pkcs11.

Настраивается модуль в файле /etc/pam_pkcs11/pam_pkcs11.conf. Если файла нет, пример его можно взять по ссылке: https://github.com/OpenSC/pam_pkcs11/blob/master/etc/pam_pkcs11.conf.example.in

В секцию **pam_pkcs11** добавляем модуль JaCarta:

```
pam_pkcs11_module JaCarta {  
    module = /usr/lib/libjcpkcs11-2.so;  
    description = "JaCarta PKCS#11 module";  
    slot_num = 0;  
    support_threads = true;  
    ca_dir = /etc/ssl;  
    crl_dir = /etc/ssl/crl;  
    cert_policy = ca,signature; }  
}
```

Замечание: пути могут меняться в зависимости от дистрибутива и версий ПО.

Меняем параметр:

```
use_pkcs11_module = JaCarta;
```

Меняем файл pam.d. Находим в каталоге /etc/pam.d файл вашего менеджера, например lightdm. В начало файла вставим строку:

```
auth sufficient pam_pkcs11.so config_file=/etc/pam_pkcs11/pam_pkcs11.conf
```

*Замечание: если мы хотим исключить другие методы аутентификации, меняем директиву **sufficient** на **required**. Не рекомендуется экспериментировать на файлах `mina login`, `common-auth` и т.д.*

Далее при подключенном токене при входе в систему ввести пин-код (некоторые DM выдают явное сообщение, что нужно авторизоваться именно на токене) и войти в систему.

ID статьи: 227

Последнее обновление: 18 Oct, 2017

Ревизия: 1

JaCarta -> Вход в Linux по сертификату на JaCarta

<https://kbp-6.aladdin-rd.ru/index.php?View=entry&EntryID=227>