# JaCarta и Astra Linux Directory (ALD)

## Настройка двухфакторной аутентификации в домене Astra Linux Directory

Версия документа: 1.0

Листов: 165

# Аннотация

Настоящий документ описывает настройку двухфакторной аутентификации по смарт-картам и USB-токенам **JaCarta PKI** на основе цифровых сертификатов X.509 в домене **ALD (Astra Linux Directory).**

Данное решение позволяет отказаться от парольной аутентификации пользователя. Внедрение настоящего решения — это кардинальное снижение влияния человеческого фактора на безопасность системы.

# Оглавление

# Об Astra Linux

Инновационная операционная система класса Linux, обеспечивающая защиту информации, содержащей сведения, составляющие государственную тайну с грифом не выше «совершенно секретно». Разработаны и включены в состав операционной системы программные компоненты, расширяющие ее функциональность и повышающие уровень защищенности и удобства ее использования.


# Об Astra Linux Directory (ALD)

Домен **Astra Linux Directory (ALD)** предназначен для организации единого пространства пользователей (домена локальной вычислительной сети) в автоматизированных системах.

**ALD** использует технологии LDAP, Kerberos5, Samba/CIFS и обеспечивает:

- централизованное хранение и управление учетными записями пользователей и групп;
- сквозную аутентификацию пользователей в домене с использованием протокола Kerberos5;
- функционирование глобального хранилища домашних директорий, доступных по Samba/CIFS;
- автоматическую настройку всех необходимых файлов конфигурации UNIX, LDAP, Kerberos, Samba, PAM;
- поддержку соответствия БД LDAP и Kerberos;
- создание резервных копий БД LDAP и Kerberos с возможностью восстановления;
- интеграцию в домен входящих в дистрибутив СУБД, серверов электронной почты, веб-серверов, серверов печати и другие возможности.

# О JaCarta PKI

**JaCarta PKI** — это линейка PKI-токенов для строгой аутентификации пользователей в корпоративных системах, безопасного хранения ключевых контейнеров программных СКЗИ и цифровых сертификатов.


В среде **Astra Linux Directory (ALD)** электронные ключи **JaCarta PKI** могут использоваться для двухфакторной аутентификации пользователя в домене **ALD** и отказа от паролей. Кроме этого, с этими же электронными ключами можно выполнять различные сценарии внутри ОС, после аутентификации, таким как: электронная подпись, хранение ключевых контейнеров, доступ к веб ресурсам, проброс ключа в сессии MS Windows. Доступ к VDI сервисам, таким как vmware или citrix.

Подробный ход настройки двухфакторной аутентификации в домен **ALD** описан ниже.

# Ход настройки

## Описание демо стенда

- **Сервер — Astra Linux Smolensk** SE 1.5 4.2.0-23-generic, x86_64, с установленными пакетами:
  - **JaCarta IDProtect 6.37;**
  - libccid;
  - pcscd;
  - libpcsclite1;
  - krb5-pkinit;
  - libengine-pkcs11-openssl;
  - opensc.
- **Клиент — Astra Linux Smolensk** SE 1.5 4.2.0-23-generic, x86_64, с установленными пакетами:
  - **JaCarta IDProtect 6.37;**
  - libccid;
  - pcscd;
  - libpcsclite1;
  - krb5-pkinit.

**Настоящая инструкция предполагает, что ALD уже развернут, существует минимум один доменный пользователь, который может аутентифицироваться по паролю, время клиента и сервера совпадают.**

# Установка драйверов на сервер и клиент

Для обеспечения работы с картой **JaCarta PKI**, на клиенте и сервере установите следующие пакеты: **libccid**, **pcscd**, **libpcsclite1**. После установки этих обязательных пакетов, установите пакет драйверов **IDProtectClient**, который можно загрузить с официального сайта «Аладдин Р.Д.» в разделе **Поддержка –> Центр Загрузки –> JaCarta –> JaCarta PKI для Linux.**

Для обеспечения работы со смарт-картой подсистемы Kerberos добавочно к предустановленным пакетам **ald/kerberos** установите пакет **krb5-pkinit** на клиенте и сервере.

Для обеспечения возможности выпуска ключей и сертификатов на **JaCarta PKI**, на сервере также установите пакеты **libengine-pkcs11-openssl** и **opensc**.

# Установка и настройка центра сертификации на сервере

В качестве центра сертификации **(CA)** будет использован **OpenSSL.**

**OpenSSL** — криптографический пакет с открытым исходным кодом для работы с SSL/TLS. Позволяет создавать ключи RSA, DH, DSA и сертификаты X.509, подписывать их, формировать CSR и CRT.

Все настройки в руководстве выполняются для тестового домена EXAMPLE.RU. Примем, что сервер и клиент принадлежат домену EXAMPLE.RU, имя сервера – kdc, а клиента – PCclient. При настройке используйте имена вашего домена, сервера и клиента. Выполните следующие действия.

1. Создайте каталог CA командой `mkdir /etc/ssl/CA` и перейдите в него. В этом каталоге будут размещаться сгенерированные ключи и сертификаты.

2. Создайте ключ и сертификат CA:
   ```
   $ openssl genrsa -out cakey.pem 2048
   $ openssl req -key cakey.pem -new -x509 –days 365 -out cacert.pem
   ```
   В диалоге заполните необходимую информацию о вашем центре сертификации. В Common name указать EXAMPLE.RU.

3. Создайте ключ и сертификат KDC:
   ```
   $ openssl genrsa -out kdckey.pem 2048
   $ openssl req -new -out kdc.req -key kdckey.pem
   ```
   В диалоге заполните необходимую информацию о вашем сервере. В Common name указать kdc.

4. Установите переменные среды. Переменные среды устанавливаются в рамках сессии и не устанавливаются для других сессий, и не сохраняются после закрытия сессии.
   `export REALM=EXAMPLE.RU` - Ваш домен
   `export CLIENT=kdc` - имя Вашего сервера

5. Загрузите файл **pkinit_extensions** — http://dms.aladdin-rd.ru/970c5538-afbf-4a26-a7ef-d76550cbc435
   Содержимое файла **pkinit_extensions** приведено в приложении.
   Его следует положить в тот каталог, откуда вы выполняете команды.

6. Выпустите сертификат KDC:
   ```
   $ openssl x509 -req -in kdc.req -CAkey cakey.pem -CA cacert.pem -out kdc.pem -
   extfile pkinit_extensions -extensions kdc_cert –CAcreateserial –days 365
   ```

7. Файлы **kdc.pem, kdckey.pem, cacert.pem** перенесите в **/var/lib/krb5kdc/**

8. Создайте резервную копию файла `/etc/krb5kdc/kdc.conf`. Отредактируйте `/etc/krb5kdc/kdc.conf`, дополнив секцию `[kdcdefaults]` следующими записями:
   ```
   pkinit_identity = FILE:/var/lib/krb5kdc/kdc.pem,/var/lib/krb5kdc/kdckey.pem
   pkinit_anchors = FILE:/var/lib/krb5kdc/cacert.pem
   ```
   Первая запись задает ключи и сертификат сервера, а вторая указывает на корневой сертификат Центра Сертификации.

9. Для принятия изменений, выполните:
   ```
   /etc/init.d/krb5-admin-server restart
   /etc/init.d/krb5-kdc restart
   ```

# Подготовка смарт-карты. Выпуск ключей и сертификата пользователя

Убедитесь, что установлены пакеты **libengine-pkcs11-openssl** и **opensc.** Подключите устройство, которое следует подготовить.

Проинициализируйте устройство, установите ПИН код пользователя.

 Внимание! Инициализация устройства удалит все данные на JaCarta PKI без возможности восстановления.

Для инициализации необходимо воспользоваться утилитой **pkcs11-tool.**

```
pkcs11-tool --slot 0 --init-token --so-pin 00000000 --label 'JaCarta PKI'  --module
/lib64/libASEP11.so
```

где:

**`--slot 0`** — указывает в какой виртуальный слот подключено устройство. Как правило, это слот 0, но могут быть и другие значения – 1,2 и т.д.

**`--init-token`** – команда инициализации токена.

**`--so-pin 00000000`** – ПИН код администратора JaCarta PKI. По умолчанию имеет значение 00000000

**`--label 'JaCarta PKI'`** - метка устройства.

**`--module /lib64/libASEP11.so`** — указывает путь до библиотеки libASEP11.so. Устанавливается в рамках пакета idprotectclient см. раздел «Установка драйверов на сервер и клиент».

Для задания ПИН кода пользователя используйте команду:

```
pkcs11-tool --slot 0 --init-pin --so-pin 00000000 --login --pin 11111111 --module
/lib64/libASEP11.so
```

где:

**`--slot 0`** — указывает в какой виртуальный слот подключено устройство. Как правило, это слот 0, но могут быть и другие значения – 1,2 и т.д.

**`--init-pin`** – команда установки ПИН-кода пользователя.

**`--so-pin 00000000`** – ПИН код администратора JaCarta PKI. По умолчанию имеет значение 00000000

**`--login`** – команда логина

**`--pin 11111111`** – задаваемый ПИН код пользователя

**`--module /lib64/libASEP11.so`** — указывает путь до библиотеки libASEP11.so. Устанавливается в рамках пакета idprotectclient см. раздел «Установка драйверов на сервер и клиент».

Сгенерируйте ключи на устройстве, для этого введите следующую команду:

```
pkcs11-tool  --slot 0  --login --pin 11111111 --keypairgen  --key-type rsa:2048 --id 42
--label “test1 key” --module /lib64/libASEP11.so
```

где:

**--slot 0** — указывает в какой виртуальный слот подключено устройство. Как правило, это слот 0, но могут быть и другие значения – 1,2 и т.д.

**--login --pin 11111111** — указывает, что следует произвести логин под пользователем, с ПИН-кодом «11111111». Если у Вашей карты другой ПИН-код пользователя, укажите его.

**--keypairgen --key-type rsa:2048** — указывает, что должны быть сгенерированы ключи длиной 2048 бит.

**--id 42** — устанавливает атрибут CKA_ID ключа. CKA_ID может быть любым.

 Запомните это значение! Оно необходимо для дальнейших шагов подготовки устройства к работе.

**--label “test1 key”** — устанавливает атрибут CKA_LABEL ключа. Атрибут может быть любым.

**--module /lib64/libASEP11.so** — указывает путь до библиотеки libASEP11.so. Устанавливается в рамках пакета idprotectclient см. раздел «Установка драйверов на сервер и клиент».

Сгенерируйте запрос на сертификат с помощью утилиты openssl. Для этого введите следующие команды:

```
#openssl
```

```
OpenSSL> engine dynamic -pre SO_PATH:/usr/lib/ssl/engines/engine_pkcs11.so -pre
ID:pkcs11 -pre LIST_ADD:1 -pre LOAD -pre MODULE_PATH:/lib64/libASEP11.so
```

```
OpenSSL> req -engine pkcs11 -new -key 0:42 -keyform engine -out client.req -subj
"/C=RU/ST=Moscow/L=Moscow/O=Aladdin/OU=dev/CN=test1
(!Ваш_Пользователь!)/emailAddress=test1@mail.com"
OpenSSL>quit.
```

Обратите внимание на **-new -key 0:42**, где **0** — номер виртуального слота с устройством, **42** — атрибут CKA_ID сгенерированных раннее ключей.

Информацию, которую необходимо указать в запросе, следует задавать в поле
```
"/C=RU/ST=Moscow/L=Moscow/O=Aladdin/OU=dev/CN=test1
(!Ваш_Пользователь!)/emailAddress=test1@mail.com"
```

Необходимо установить переменные окружения:

```
$ export REALM=EXAMPLE.RU - Ваш домен
$ export CLIENT=test1 – имя Вашего пользователя
```

И выпустить сертификат на пользователя.
```
$ openssl x509 -CAkey cakey.pem -CA cacert.pem -req -in client.req -extensions
client_cert -extfile pkinit_extensions -out client.pem –days 365
```

Далее перекодируйте полученный сертификат из PEM в DER.

```
# openssl x509 -in client.pem -out client.cer -inform PEM -outform DER
```

Запишите полученный сертификат на токен.

```
pkcs11-tool --slot 0 --login --pin 11111111 --write-object client.cer  --type 'cert' --
label 'Certificate'  --id 42 --module /lib64/libASEP11.so
```

где:

**--slot 0** — указывает в какой виртуальный слот подключено устройство. Как правило, это слот 0, но могут быть и другие значения – 1,2 и т.д.

**--login --pin 11111111** — указывает, что следует произвести логин под пользователем, с ПИН кодом «11111111». Если у Вашей карты другой ПИН-код пользователя, укажите его.

**--write-object ./client.cer** — указывает, что необходимо записать объект и путь до него.

**--type 'cert'** — указывает, что тип записываемого объекта – сертификат.

**'cert' --label 'Certificate'** — устанавливает атрибут CKA_LABEL сертификата. Атрибут может быть любым.

**`--id 42`** — устанавливает атрибут CKA_ID сертификата. Должен быть указан тот же CKA_ID, что и для ключей.

**`--module /lib64/libASEP11.so`** — указывает путь до библиотеки libASEP11.so.

# Настройка клиента. Проверка работоспособности

Создайте на клиенте каталог **/etc/krb5/.** Скопируйте в **/etc/krb5/** сертификат CA **(cacert.pem)** с сервера.

Настройте `kerberos в /etc/krb5.conf`. Секцию `[libdefaults]` дополните следующими строками.

```
[libdefaults]
default_realm = EXAMPLE.RU
pkinit_anchors = FILE:/etc/krb5/cacert.pem
# для аутентификации по токену
pkinit_identities = PKCS11:/lib64/libASEP11.so
```

Выполните проверку:

```
kinit <username>
```

Когда появится строка запроса ПИН-кода к карте, введите его.
Для проверки того, что kerberos-тикет был успешно получен для пользователя, введите команду `klist`. Для удаления тикета — `kdestroy`.

Для входа в домен по смарт-карте на экране входа в ОС, вместо пароля введите ПИН-код от смарт-карты.

# Возможные проблемы и способы их устранения

Проблема: **JaCarta PKI** не отображается в **IDProtectClient.**

Решение:

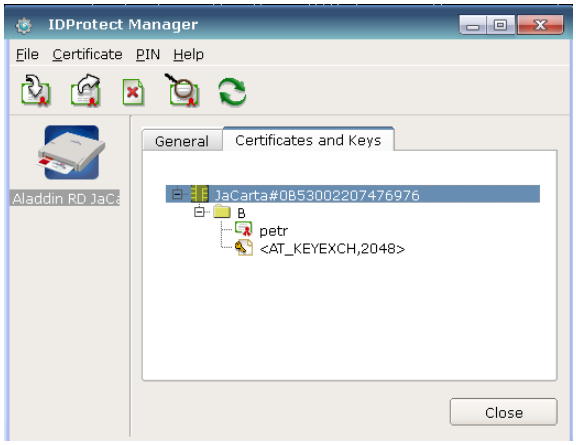
1. Убедитесь, что установлены пакеты **libccid**, **pcscd**, **libpcsclite1 .** Пакет драйверов **IDProtectClient** следует устанавливать только после установки этих пакетов.
2. Убедитесь, что демон **pcscd** запущен. Если нет, запустите его.
3. Если демон **pcscd** уже был запущен, перезапустите демон **pcscd**.

```
/etc/init.d/pcscd restart
```

Проблема: Не происходит аутентификации по сертификату.

Решение:

1. Проверьте, что утилита **IDProtect Manager** отображает ключи и сертификат в составе одного контейнера.




2. Убедитесь, что на сервере и клиенте установлен пакет **krb5-pkinit.**
3. Перезапустите на сервере службы **kdc**

```
/etc/init.d/krb5-admin-server restart
/etc/init.d/krb5-kdc restart
```

# Приложение

Содержимое файла **pkinit_extensions:**

```
[ kdc_cert ]
basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
keyUsage = nonRepudiation, digitalSignature, keyEncipherment, keyAgreement

#Pkinit EKU
extendedKeyUsage = 1.3.6.1.5.2.3.5

subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer

# Copy subject details

issuerAltName=issuer:copy

# Add id-pkinit-san (pkinit subjectAlternativeName)
subjectAltName=otherName:1.3.6.1.5.2.2;SEQUENCE:kdc_princ_name

[kdc_princ_name]
realm = EXP:0, GeneralString:${ENV::REALM}
principal_name = EXP:1, SEQUENCE:kdc_principal_seq

[kdc_principal_seq]
name_type = EXP:0, INTEGER:1
name_string = EXP:1, SEQUENCE:kdc_principals

[kdc_principals]
princ1 = GeneralString:krbtgt
princ2 = GeneralString:${ENV::REALM}

[ client_cert ]

# These extensions are added when 'ca' signs a request.

basicConstraints=CA:FALSE
```

```
keyUsage = digitalSignature, keyEncipherment, keyAgreement

extendedKeyUsage =  1.3.6.1.5.2.3.4
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer


subjectAltName=otherName:1.3.6.1.5.2.2;SEQUENCE:princ_name


# Copy subject details

issuerAltName=issuer:copy

[princ_name]
realm = EXP:0, GeneralString:${ENV::REALM}
principal_name = EXP:1, SEQUENCE:principal_seq

[principal_seq]
name_type = EXP:0, INTEGER:1
name_string = EXP:1, SEQUENCE:principals

[principals]
princ1 = GeneralString:${ENV::CLIENT}
```

# Контакты, техническая поддержка

## Офис (общие вопросы)

Адрес: 129226, Москва, ул. Докукина, д. 16, стр. 1, компания «Аладдин Р. Д.».

Телефоны: +7 (495) 223-00-01 (многоканальный), +7 (495) 988-46-40.

Факс: +7 (495) 646-08-82.

E-mail: aladdin@aladdin-rd.ru (общий).

Web: www.aladdin-rd.ru

Время работы: ежедневно с 10:00 до 19:00, кроме выходных и праздничных дней.

## Техподдержка

Служба техподдержки принимает запросы только в письменном виде через веб-сайт:

**www.aladdin-rd.ru/support/index.php**

Для оперативного решения вашей проблемы укажите используемый Вами продукт, его версию, подробно опишите условия и сценарии применения, по возможности, снабдите сообщение снимками экрана, примерами исходного кода.

# Регистрация изменений

| Версия | Изменения |
|---|---|
| 1.0 | Исходная версия документа |
| | |
| | |

Лицензии ФСТЭК России № 0037 и № 0054 от 18.02.03, № 2874 от 18.05.12
Лицензии ФСБ России № 12632 Н от 20.12.12, № 24530 от 25.02.14
Система менеджмента качества компании соответствует требованиям стандарта ISO/ИСО 9001-2011
Сертификат СМК ГОСТ Р ИСО 9001-2011 № РОСС RU.ИС72.К00082 от 10.07.15
Microsoft Silver OEM Hardware Partner, Microsoft Silver Cloud Platform Partner, Apple Developer




Тел. +7 (495) 223-00-01 Email: aladdin@aladdin-rd.ru Web: www.aladdin-rd.ru