Сервис проксирующий запросы в КриптоПРО CSP

Назначение

Сервис предназначен для подписания документов из АРМ Поликлиника, запущеной в OS Linux под wine.

Текущая версия сервиса работает с КриптоПРО CSP 4.0, 5.0.11453, 5.0.12000

АРМ поликлиника направляет в сервис HTTP запросы, сервис взаимодействует с КриптоПРО CSP по протоколу CryptoAPI Lite

Сервис реализует методы:

- Получение списка сертификатов
- Подписание переданного массива байтов
- Верификация подписанного файла
- Получение публичного ключа сертификата

Требования

Сервис необходимо устанавливать на той рабочей станции, на которой запускается АРМ Поликлиника.

На рабочей станции должен быть установлен КриптоПРО CSP с графическими модулями

- "cprocsp-rdr-gui-gtk" "GUI for smart card and token support modules"
- "cprocsp-cptools-gtk" "cptools, GUI application for various CSP tasks"

Установка и настройка

Установка КриптоПРО

Установите на рабочей станции КриптоПРО CSP для Linux. В обязательном порядке нужно установить модули

- "cprocsp-rdr-gui-gtk" "GUI for smart card and token support modules"
- "cprocsp-cptools-gtk" "cptools, GUI application for various CSP tasks"

Эти модули позволят указывать пароль к контейнеру закрытого ключа через пользовательский интефейс КриптоПРО, а так же управлять контейнерами ключей через графический интерфейс.

Установите свою электронную подпись в в хранилище "Личное". Установка должна выполняться от имени пользователя, под которым запускается АРМ Поликлиника.

Установка сервиса

Для установки сервиса, скачайте установочный файл подходящий для вашей версии КриптоПРО с сайта http://dl.22m22.ru/

Установите бит исполнимого файла на скаченный файл и запустите файл на исполнение от имени пользователя, от которого будет запускаться АРМ Поликлиника.

Инсталлятор выполняет следующие действия:

- Создает папку $HOME/.local/bin
- Копирует в эту папку иполняемый файл сервиса cryptopro-ws
- Создает папку для конфигурационного файла запуска скрипта $HOME/.config/systemd/user
- Создает в этой папке конфигурационный файл запуска cryptopro-ws.service с таким содержимым

```
[Unit]
Description=cryptopro-ws
After=network.target

[Service]
Type=simple
```

```
ExecStart=/home/user/.local/bin/cryptopro-ws
Restart=always
RestartSec=5

[Install]
WantedBy=default.target
```

где user - это имя пользователя, от имени которого установлен сервис

- Включает автозапуск сервиса

```
systemctl --user enable cryptopro-ws
```

- Запускает сервис от имени текущего пользователя

```
systemctl --user start cryptopro-ws
```

- Выводит на экран статус сервиса

```
systemctl --user status cryptopro-ws
```

Дополнительные параметры запуска сервиса

В конфигурационном файле в строке запуска сервиса можно указать дополнительные параметры, изменяющие работу сервиса.

Список всех параметров можно посмотреть командой

```
~/.local/bin/cryptopro-ws -h
```

- -h показать справку по сервису
- -l адрес, который будет слушать сервис. Здается в формате < IPv4:port> или <[IPv6]:port>. IP можно не указыывать, тогда сервис будет слушать все доступные сетевые адреса. По умолчанию сервис запскается на порту 8000 и слушает адрес 127.0.0.1.
- -log.file полный путь до файла логов. В системе с systemd лучше не указывать этот параметр, тода логи будут писаться в системный журнал. Если вы задаете файло логов, то нужно самостоятельно позаботиться о его архивировании и обрезании.
  Если лог-файл не задан, то логи сервиса можно смотреть командой

```
journalctl -u cryptopro-ws
```

- -log.format формат записи логов. Допустимы значения text и json. По умлочанию text.
- -log.level уровень логгирования, допустимые значения: error, warning, info or debug. По умолчанию info.
- -response.headers json-объект дополнительных http-заголовоков, которые будут добавляться во все ответы сервиса, например

```
/home/user/.local/bin/cryptopro-ws -response.headers '{"Access-Control-Allow-Origin":"*"}'
```

- -tls.crt полный путь до файла сертификата в формате x509 для работы сервиса с использованием TLS шифрования трафика
- -tls.key полный путь до файла приватного ключа для работы сервиса с использованием TLS шифрования трафика

Настройка TLS

Сервис может запускаться в режиме использования TLS (transport layer security) подключений. Для этого сервису нужно указать файл приватного ключа и соотвествующего ему сертификата в формате x509 (см. параметры запуска выше).

Можно создать самоподписанный сертификат и использовать его в сервисе. АРМ Поликлиника не проверяет подлинность сертификатов.

Для создания самоподписанного сертификата выполните следующие действия.

- Установите в системе openssl, если он еще не установлен
- Создайте приватный ключ по алгоритму RSA, с длиной ключа не менее 2048 бит

```
openssl genrsa -out server.key 2048
```

- Установите права доступа к приватному ключу только для текущего пользователя

```
chmod 600 server.key
```

- Создайте публичный сертификат на основе приватного ключа

```
openssl req -new -x509 -sha256 -key server.key -out server.crt -days 3650
```

- Разместите созданые файлы в удобном месте и ипользуйте их в параметрах азпуска сервиса.

Описание API сервиса

Взаимодействие с сервисом выполняется по протоколу http (или https, в случае использования TLS). Ниже указаны примеры вызова методов сервиса при помощи утилиты командной строки curl. Данная утилита есть в поставке дистрибутива КриптоПРО, либо может быть усталновлена в операционной системе из стандартных репозиториев.

Получить список сертификатов

curl по умолчанию в астре не стоит, поэтому используем тот, который идет в поставке КриптоПРО

```
$ /opt/cprocsp/bin/amd64/curl http://localhost:8000/certs
```

Возвращает json-объкт со следующими полями

Поле	Тип	Описание
certs	[]Cert	список сертификатов
count	int	количество сертификатов в списке

Объект Cert

Поле	Тип	Описание
certs	[]Cert	список сертификатов
serialNumber	string	серийный номер сертификата
hash	string	отпечаток (SHA hash)
subjectKeyId	string	идентификатор открытого ключа субъекта сертификата
subjectStr	string	именованные данные владельца сертификата
subjectOidStr	string	данные владельца сертификата с OID идентификаторами
issuerStr	string	именованные данные издателя
issuerOidStr	string	данные издателя с OID идентификаторами
signatureAlgorithmOid	string	идентификатор алогоритма подписи
signatureAlgorithmName	string	алгоритм подписи
publicKeyAlgorithmOid	string	идентификатор алгоритма публичного ключа
publicKeyAlgorithmName	string	алгоритм публичного ключа
container	string	название контейнера
providerName	string	имя провайдера
providerType	int	тип провайдера
providerKeySpec	int	Спецификация закрытого ключа для извлечения.
providerFlag	int	Набор флагов, указывающих дополнительную информацию о

		поставщике
beginTime	DateTime	действует с
endTime	DateTime	действует до
pinCodeRequired	bool	признак, что контейнер закрытого ключа запаролен

Подписание файлов и проверка подписи

Подпись бывает

- прикрепленной - сервис вернет данные, включающие в себя как электронную подпись, так и исходные данные. Для извлечения исходных данных из такого файла нужно использовать средства криптопровайдера
- открепленной - сервис вернет только электронную подпись.

Сервис позволяет указать необязательный параметр pin - пароль доступа к контейнеру закрытого ключа. Если пароль будет передан в запросе, то КриптоПРО не будет показывать диалоговое окно запроса пароля.

Прикрепленная подпись

```
$ /opt/cprocsp/bin/amd64/curl http://localhost:8000/sign?sn=номер_сертификата -H "Content-Type: application/octetstream" --data-binary @файл_для_подписи -o файл_для_результата.sig
```

Файл файл_для_результата.sig будет содержать исходные данные и подпись.

Проверка прикрепленной подписи

```
$ /opt/cprocsp/bin/amd64/curl -X POST 'http://localhost:8000/verify' -F "sign=@файл_для_результата.sig" -o извлеченные_данные_из_подписанного_файла
```

Если подпись корректная, то код ответа 200 в извлеченные_данные_из_подписанного_файла будут записаны исходные данные, извлеченные из подписанного файла.

Открепленная подпись

При создании открепленной подписи сервис вернет бинарные данные, в которых будет только электронная подпись. Подписанные данные в ответ не включаются.

```
$ /opt/cprocsp/bin/amd64/curl http://localhost:8000/sign?sn=номер_сертификата&detached=1 -H "Content-Type: application/octetstream" --data-binary @файл_для_подписи -o файл_для_результата.sig
```

файл_для_результата.sig будет содержать только подпись

Проверка открепленной подписи

Для проверки открпленной подписи нужно передавать как файл подписи, так и исходный (подписанный) файл

```
$ /opt/cprocsp/bin/amd64/curl -X POST 'http://localhost:8000/verify' -F "sign=@файл_для_результата.sig" -F "data=@файл_для_подписи"
```

Если подпись корректная, то вернется пустой ответ, код ответа 200.

Получить публичный ключ при помощи curl

Публичный ключ возвращается в бинарном виде (application/octet-stream).

Публичный ключ можно получить по одному из следующих признаков

Название параметра	Поле в данных сертификата	Описание
sn	serialNumber	серийный номер сертификата
hash	hash	отпечаток (SHA hash)
di	subjectKeyId	идентификатор открытого ключа субъекта сертификата

```
$ /opt/cprocsp/bin/amd64/curl http://localhost:8000/pubkey?sn=номер_сертификата -o key.pub
```

или

```
$ /opt/cprocsp/bin/amd64/curl http://localhost:8000/pubkey?hash=отпечаток_сертификата -o key.pub
```

или

```
$ /opt/cprocsp/bin/amd64/curl http://localhost:8000/pubkey?id=ID_сертификата -o key.pub
```

Настройка АРМ Поликлиника

Установите wine. Запустите winecfg, перейдите на закладку "Библиотеки", добавьте библиотеку riched20 (она нужна для корректной работы АРМ Поликлиника)

- Запустите АРМ Поликлиника.
- Откройте меню "Настройки - Общие настройки".
- Включите модуль "ЭЦП".
- Если в дереве настроек нет пункта "Работа с ЭЦП", то закройте окно настроек и откройте его заново.
- В разделе "Работа с ЭЦП" выберите пункт меню "Программа работает в Linux под wine" и укажите адрес сервиса. Сервис работает по http протоколу, поэтому адрес должен быть в виде http://<host>:<port>/
- Нажмите кнопку "Тест". Вы должны увидеть список сертификтов из хранилища "Личное".

Если вы видите сертификаты, значит все настроено правильно.