Описание работы утилиты stunnel в режиме клиента на Windows 10 и режиме сервера на MAC OS X

В данном документе описан процесс подготовки к работе и функционирования утилиты stunnel в режиме клиента в среде ОС Windows 10 и режиме сервера в среде OS X El Capitan.

Утилита stunnel предназначена для шифрования трафика между приложениями, в которых данный функционал не был реализован. Возможна работа в режиме клиента и сервера. Stunnel, работающий в режиме клиента, принимает незашифрованный трафик от некоего клиентского ПО по указанному ip-адресу и порту, затем зашифровывает и передаёт его на сервер. Если сервер поддерживает TLS (IIS, TrustedTLS и тд) – серверная часть stunnel'а не требуется. В режиме сервера stunnel принимает на заранее определённый в конфигурационном файле ip-адрес и порт зашифрованный трафик, затем расшифровывает его и передает соответствующему ПО на сервере.

В имеющемся в Mac OS X веб-сервере apache по умолчанию отсутствует функционал, реализующий соединение, зашифрованное по алгоритмам ГОСТ. Для обхода данного ограничения организуется работа утилиты stunnel в режиме сервера. Шифрованный трафик, приходящий на заранее указанный порт, будет расшифрован и направлен в открытом виде на порт, выделенный для Apache по умолчанию.

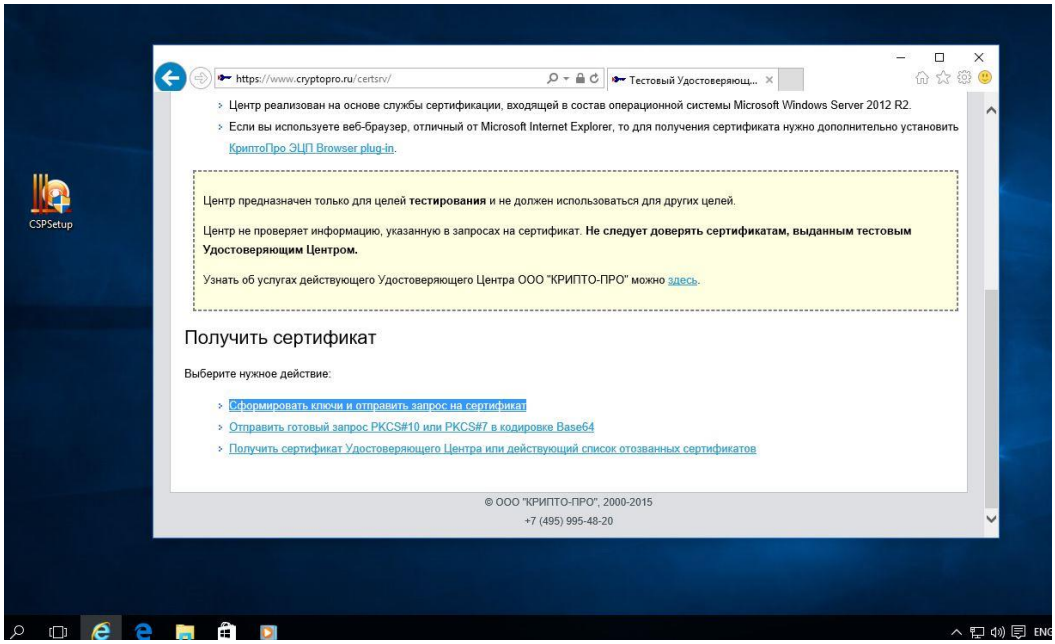
На Windows 10 утилита stunnel настроена в режиме клиента. Незашифрованный трафик поступает на заранее выбранный порт, затем шифруется и передаётся на указанный порт компьютера с Mac OS X и серверной частью stunnel.

В данном примере доступ осуществляется по конкретному сертификату клиента, присутствующему в хранилище TrustedUsers на сервере. Для выпуска сертификатов используется тестовый УЦ ООО "КРИПТО-ПРО" https://www.cryptopro.ru/certsrv/ Возможны варианты, когда проверка сертификата производится только при его наличии, либо не производится вовсе - условия задаются опцией verify в конфигурационном файле stunnel.conf.

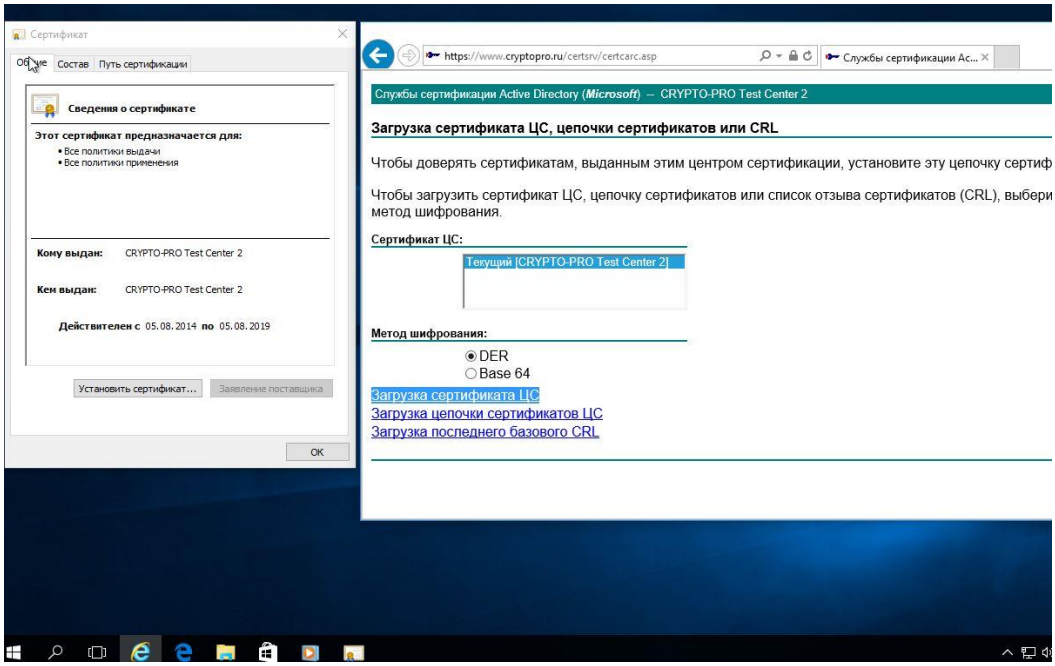
Настройка клиентской части stunnel в среде Windows 10

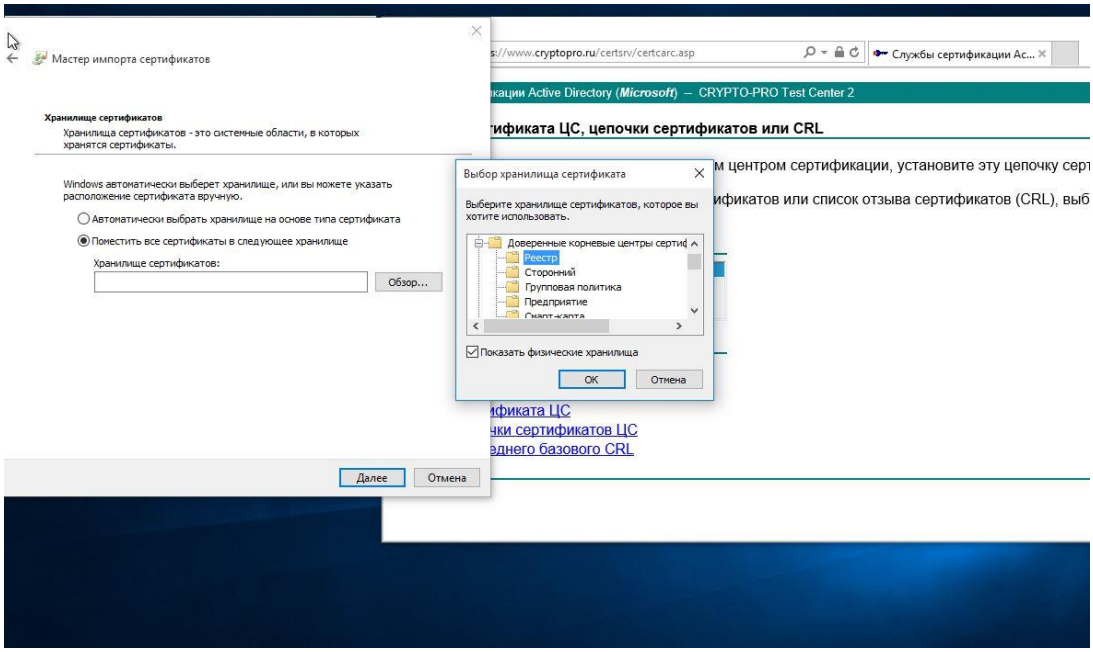
Для работы в клиентском режиме потребуется установленное ПО КриптоПро CSP с действующей лицензией, сконфигурированная утилита stunnel, а также клиентский сертификат с объектным идентификатором в области использования ключа “проверка подлинности клиента”

После скачивания дистрибутива КриптоПро CSP с официального сайта и его последующей установки стандартным способом, производится выпуск клиентского сертификата. Сам сертификат может быть выпущен различными способами, но в рассматриваемом примере будет использоваться интерфейс тестового Удостоверяющего Центра ООО "КРИПТО-ПРО" https://www.cryptopro.ru/certsrv.

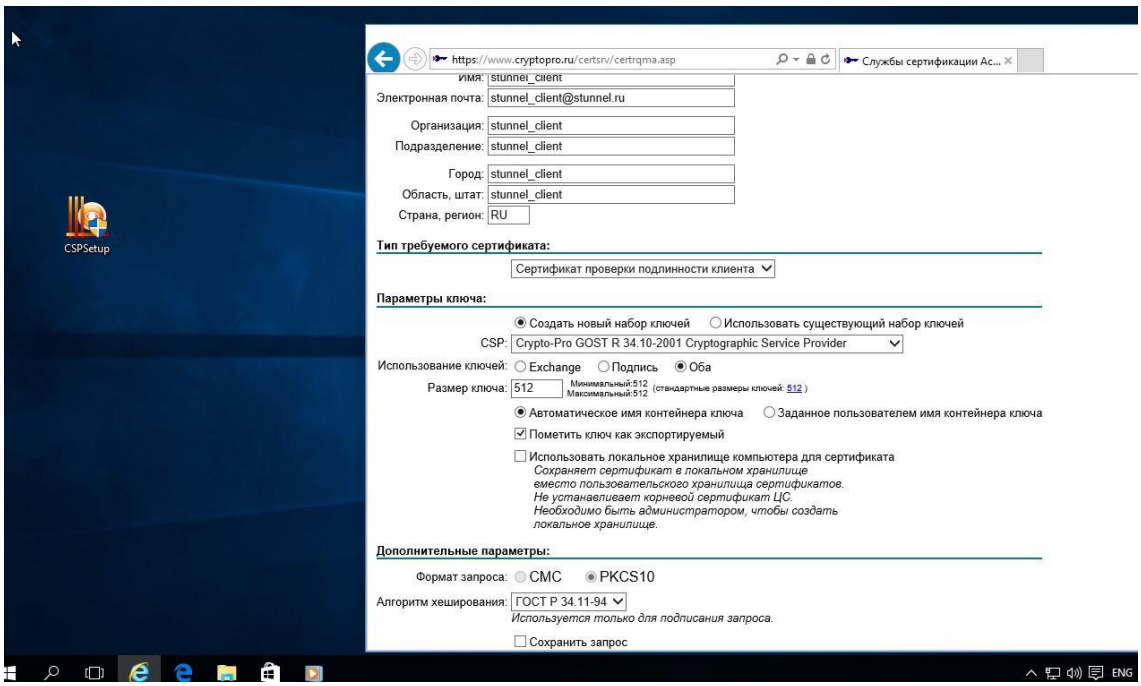


Далее отображен процесс скачивания корневого сертификата удостоверяющего центра и установки его в хранилище “Доверенные корневые центры сертификации”.

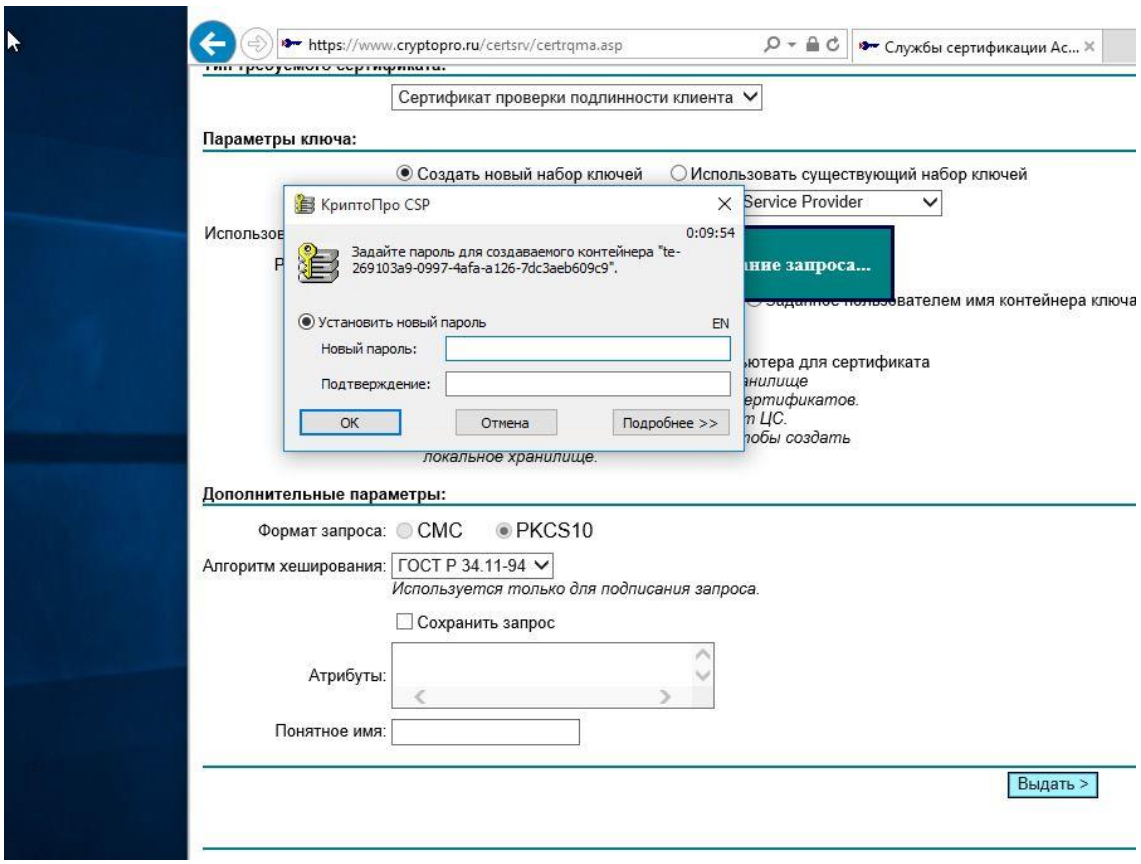




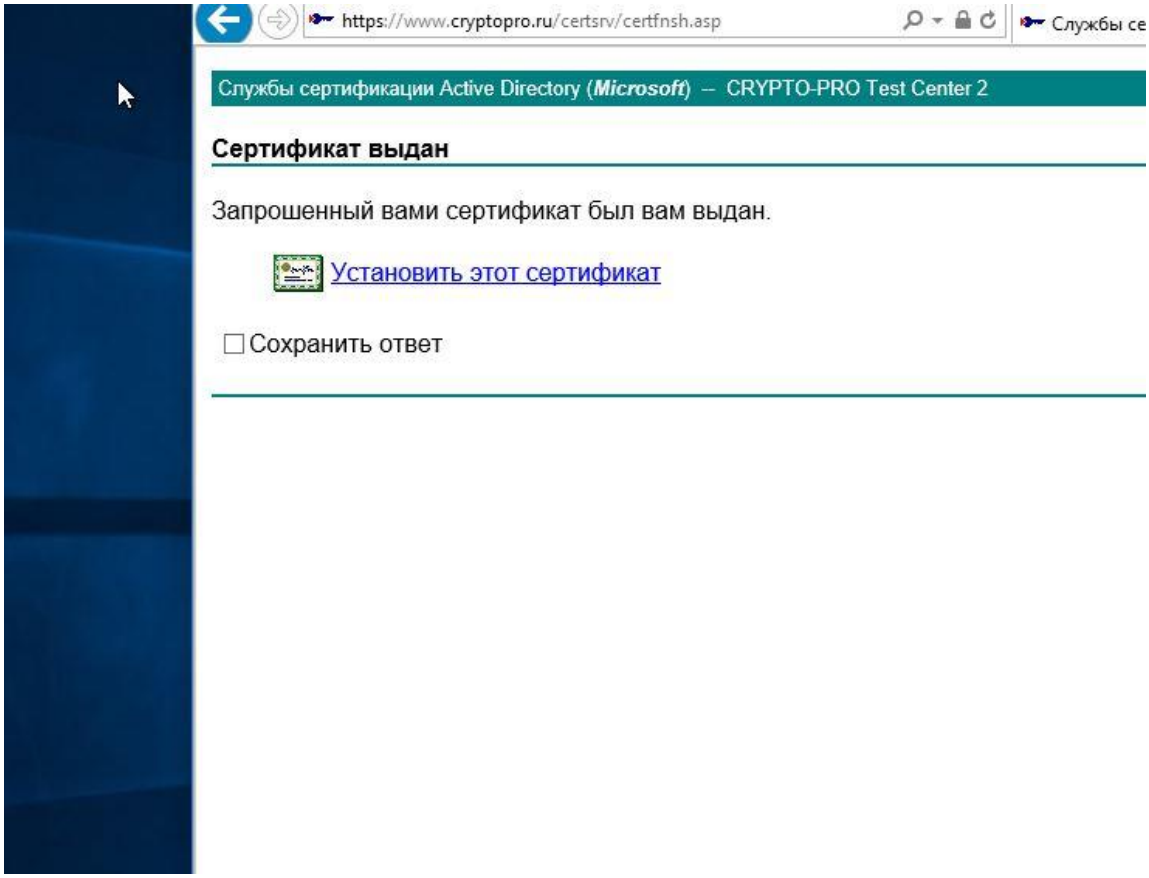
При выпуске сертификата клиента необходимо выбрать тип “Сертификат проверки подлинности клиента”



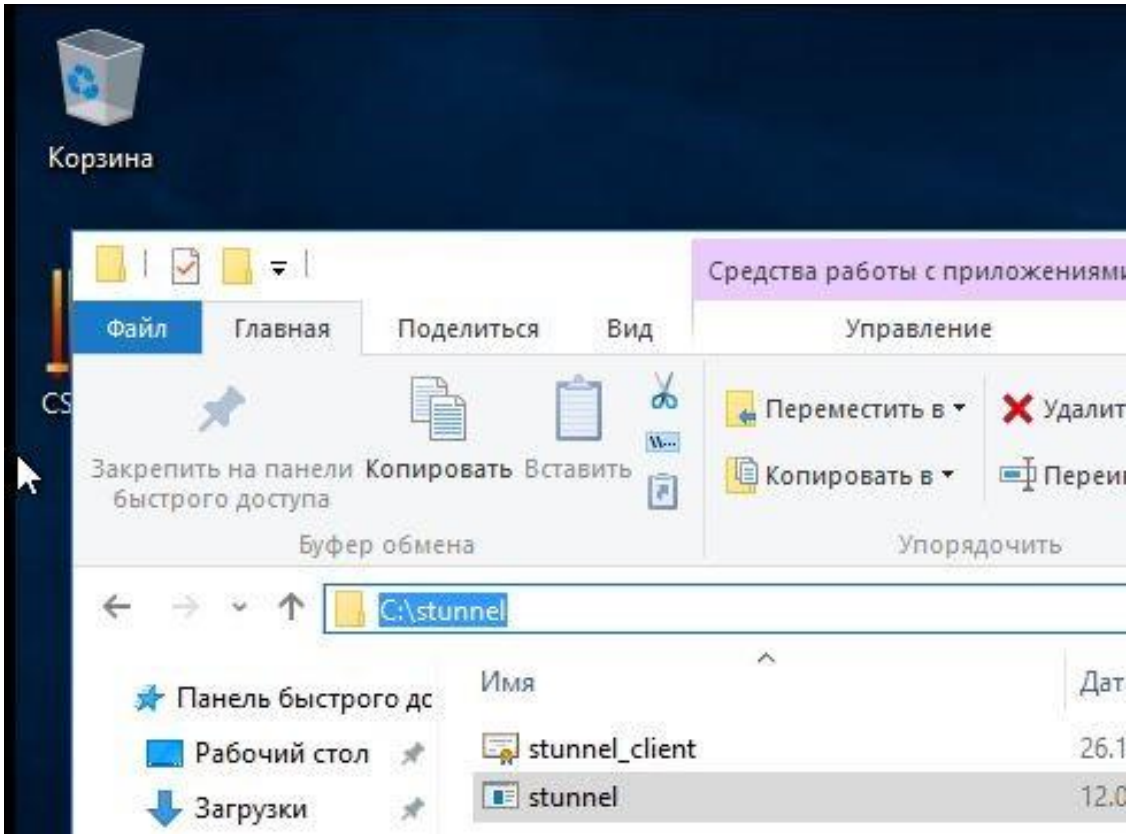
При необходимости задаётся пароль на контейнер.



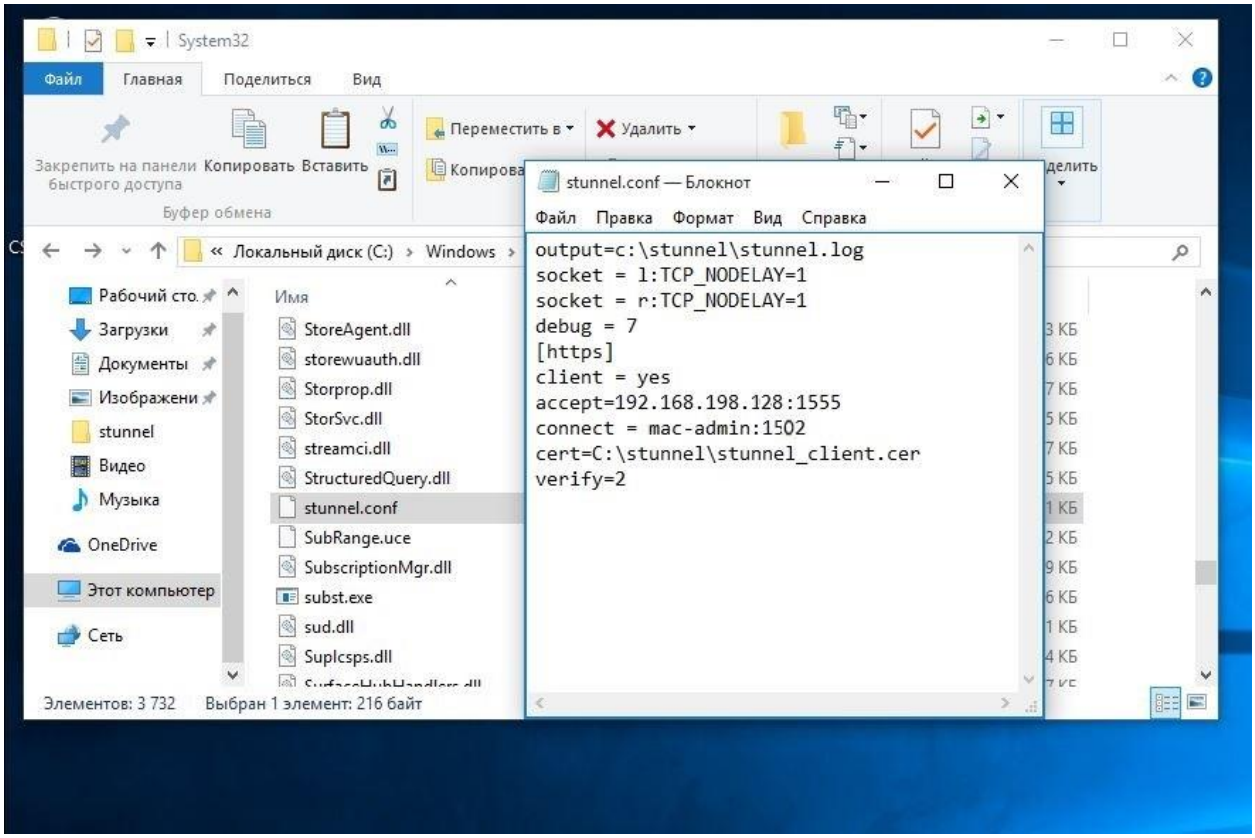
Производится установка сертификата.



Создаётся рабочая папка, в которую производится экспорт созданного сертификата, а также копирование утилиты stunnel версии, соответствующей разрядности операционной системы (https://www.cryptopro.ru/products/other/stunnel).



В папке C:\Windows\system32 формируется файл конфигурации stunnel.conf



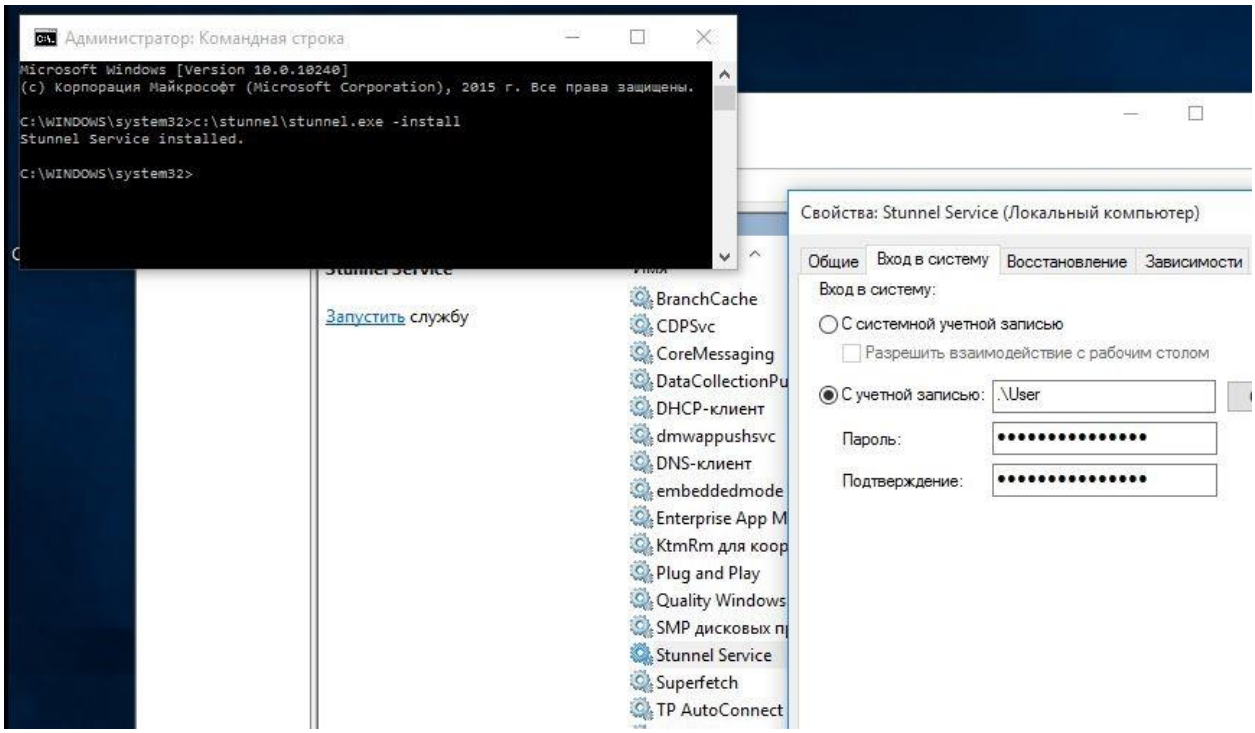
Содержимое конфигурационного файла stunnel.conf клиентской части:

```
output=C:\stunnel\stunnel.log

socket=l:TCP_NODELAY=1

socket=r:TCP_NODELAY=1

debug=7

[https]

Accept=192.168.198.128:1555

Connect=mac-admin:80

Cert= C:\stunnel\stunnel_client.cer

Verify=2
```

Краткое описание наиболее часто используемых параметров конфигурации.

Опция	Описание
Pid (в unix системах)	Путь к файлу, в котором будет храниться идентификатор процесса
output	Путь к лог-файлу
socket	Опции для конфигурирования принимающих, локальных, удалённых сокетов.
debug	Уровень протоколирования (для экономии места на диске рекомендуется устанавливать значение 1)
client	Работа в режиме клиента
accept	Адрес и порт для приёма незашифрованного трафика
connect	Адрес и порт сервера, на который передаётся зашифрованный трафик
cert	Путь к файлу сертификата клиента
verify	Возможные варианты проверки сертификата удалённого сервера 0. Не проверять сертификат сервера 1. Проверять сертификат при его наличии 2. Проверять сертификат всегда 3. Проверять наличие данного сертификата в хранилище TrustedUsers
pincode	PIN-код контейнера

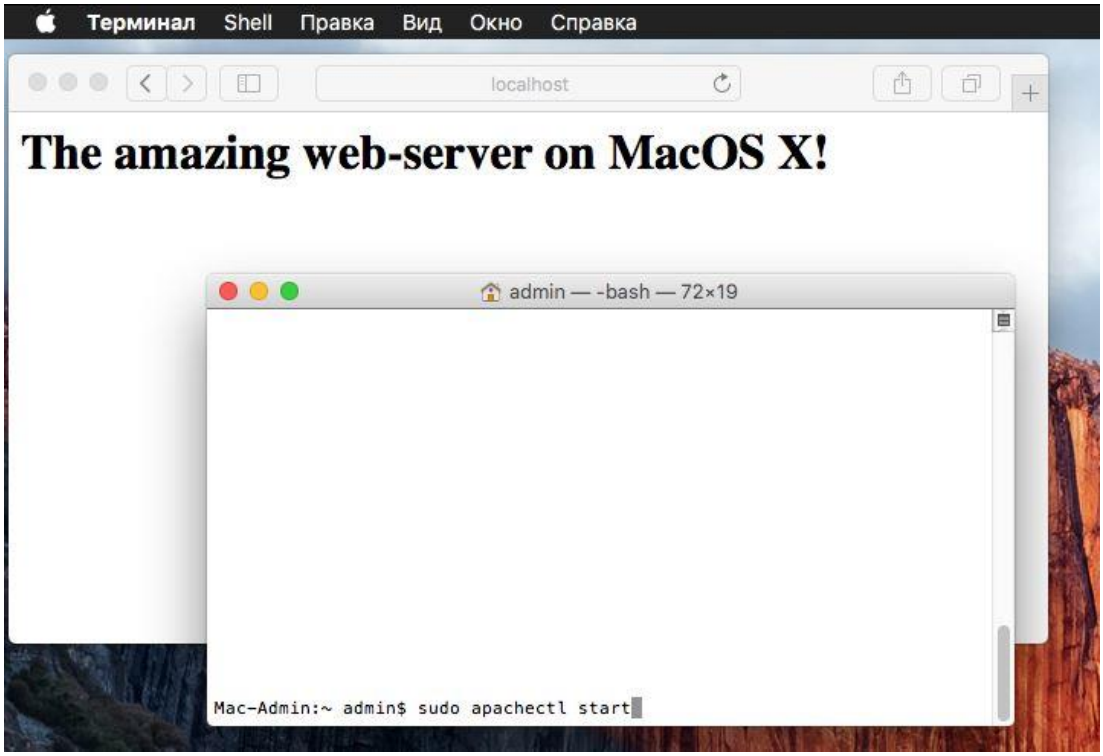
Далее в командной строке, запущенной с правами администратора, производится установка службы stunnel, после чего в оснастке services.msc указывается учётная запись, под которой будет работать служба. Производится запуск самой службы.



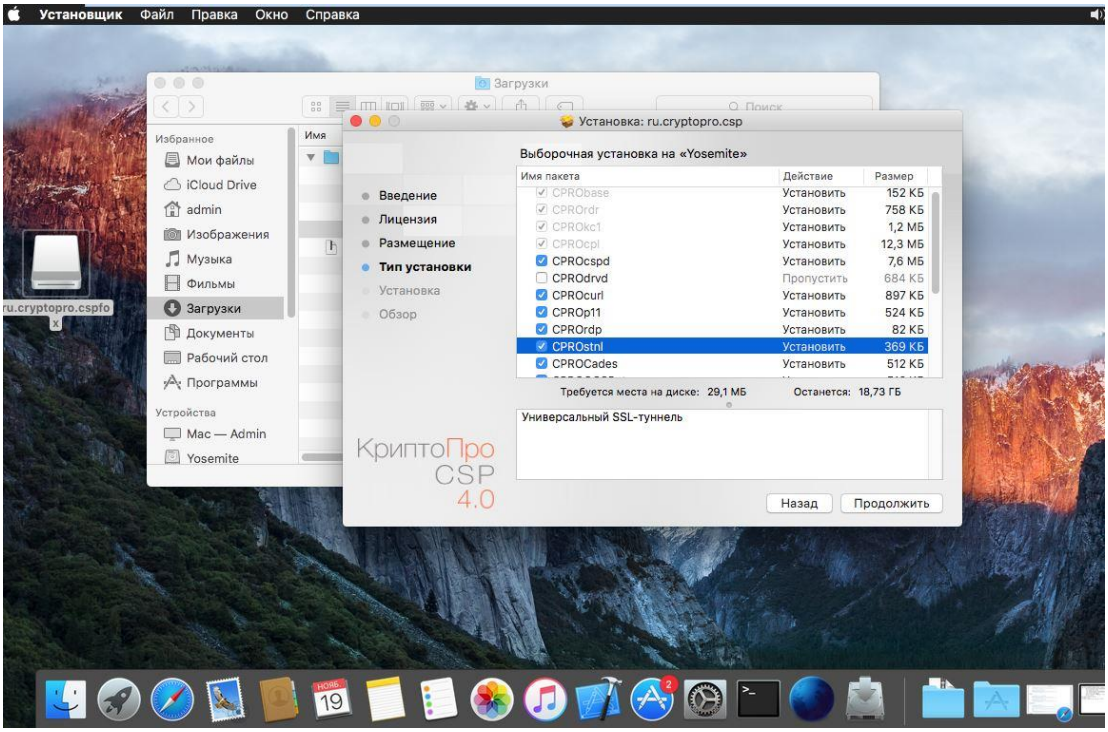
При наличии PIN на контейнере потребуется, либо сохранить его в памяти средствами CSP(протестировать контейнер, поставив галку “запомнить пароль”), или указать в явном виде в файле конфигурации, использовав опцию pincode.

Настройка серверной части stunnel в среде Mac OS X

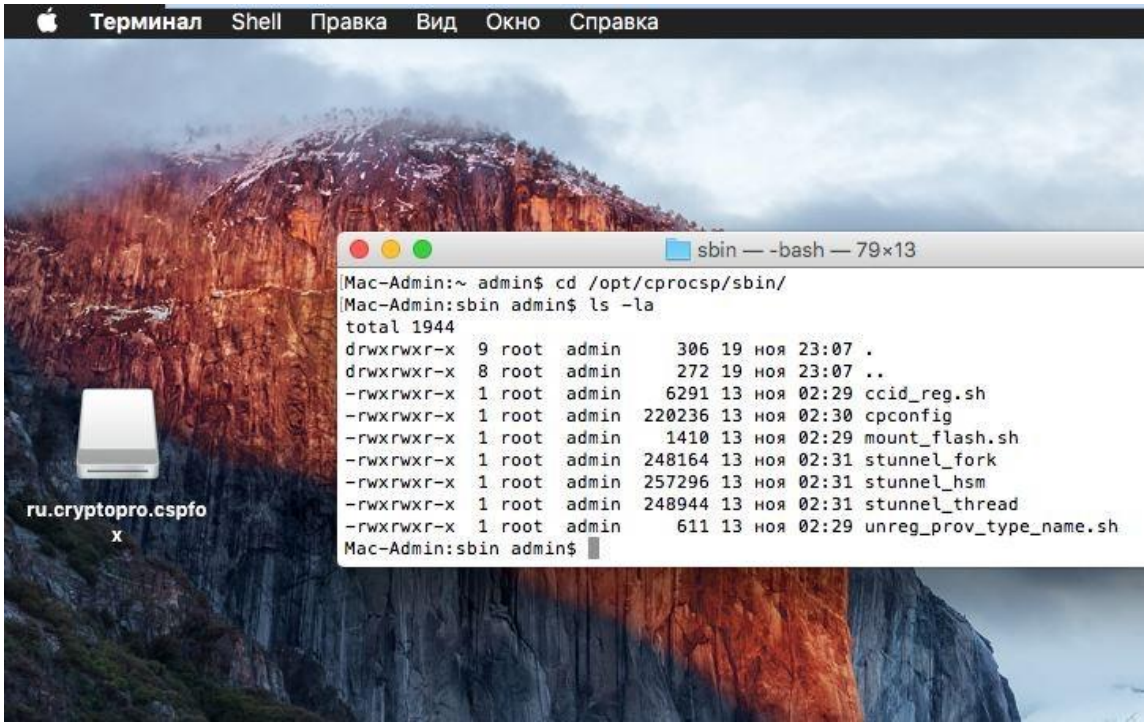
Производится запуск имеющегося в системе веб-сервера Apache. Используется команда **sudo apachectl start**



Утилита stunnel входит в дистрибутив КриптоПро CSP для MAC OS X и должна быть выбрана для установки вместе с остальными пакетами.

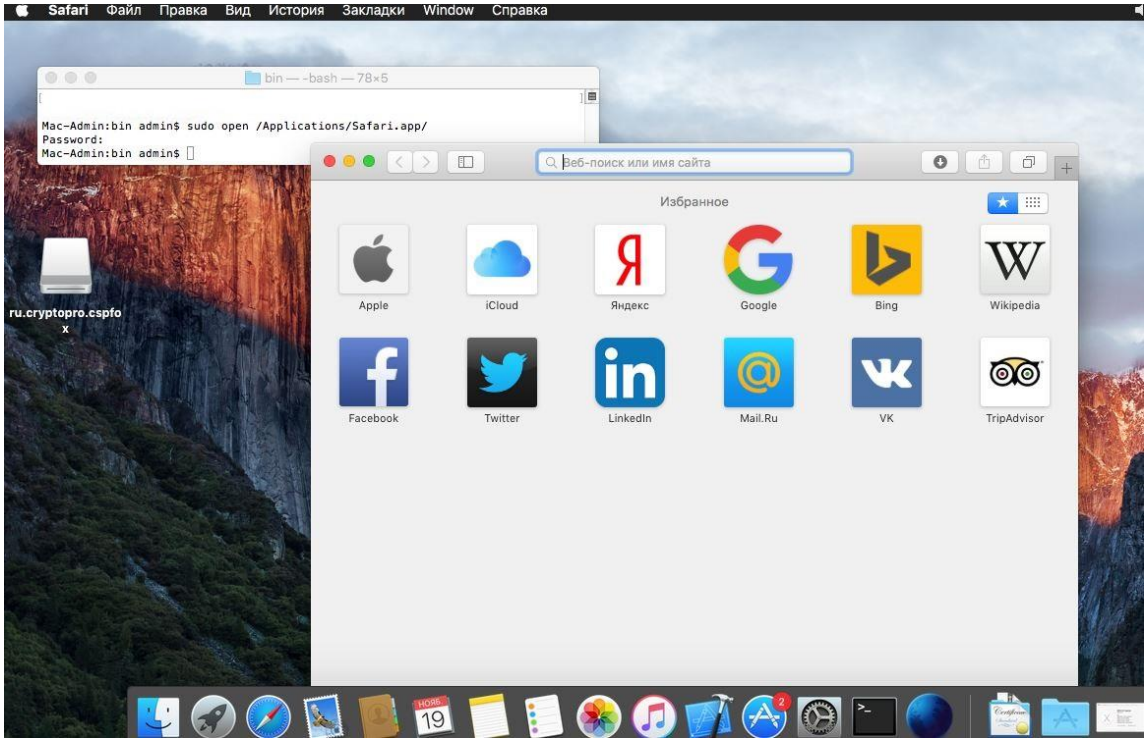


Завершив установку, можно проверить содержимое папки /opt/cprocsp/sbin

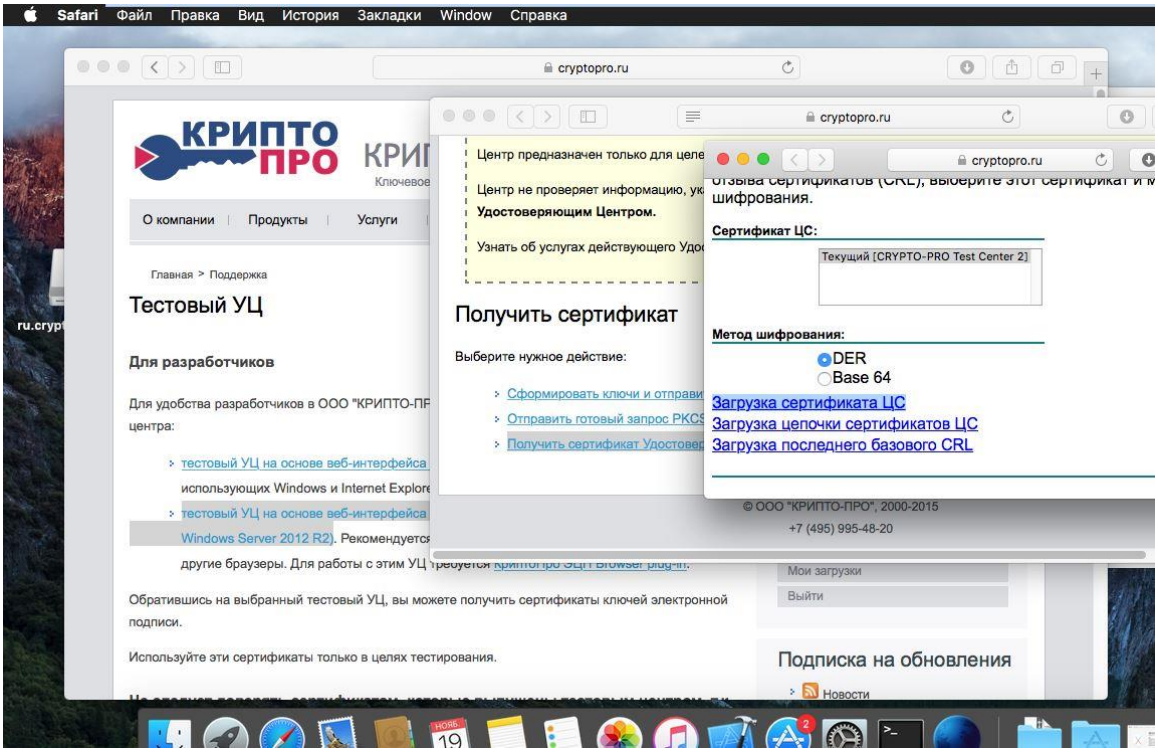


Далее будет установлен корневой сертификат удостоверяющего центра, а также произведён выпуск сертификата сервера. Для работы с интерфейсом тестового удостоверяющего центра (т.к. генерация контейнера будет производиться в хранилище компьютера, а не пользователя) понадобится запустить браузер Safari с правами администратора. Для этого используется команда:

sudo open /Applications/Safari.app

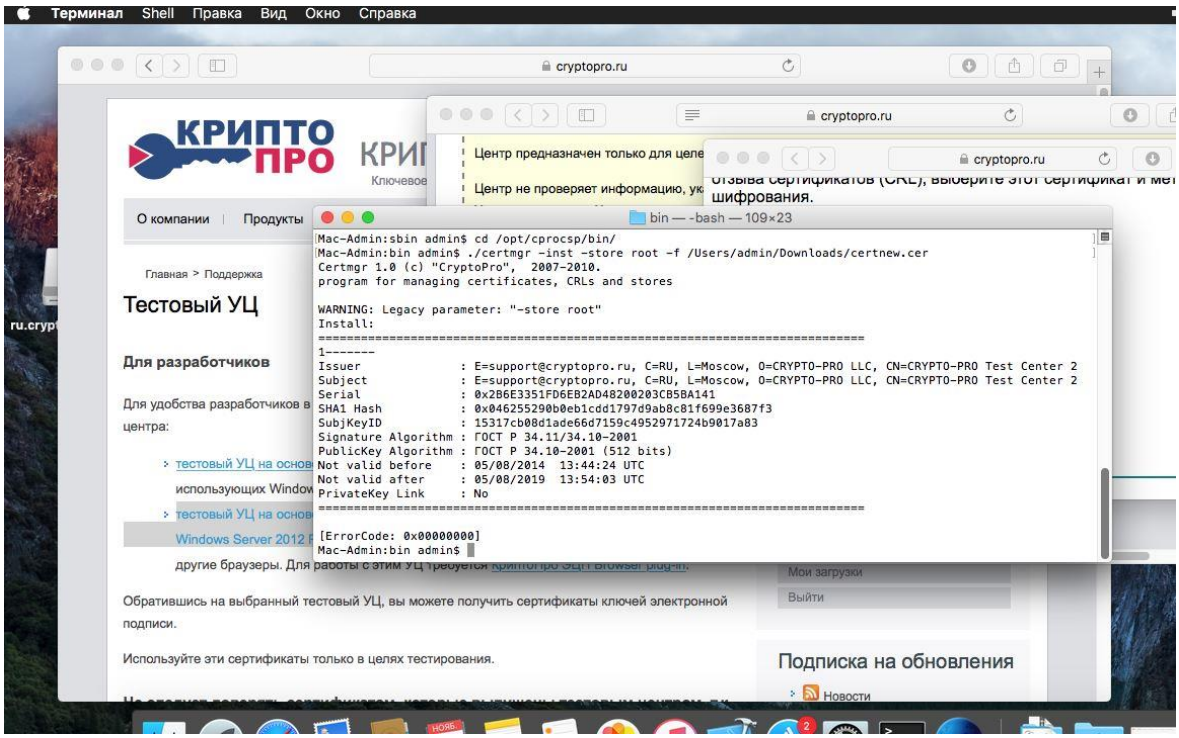


Необходимо скачать и установить корневой сертификат тестового УЦ ООО "КРИПТО-ПРО" на котором будет произведён выпуск сертификата сервера.

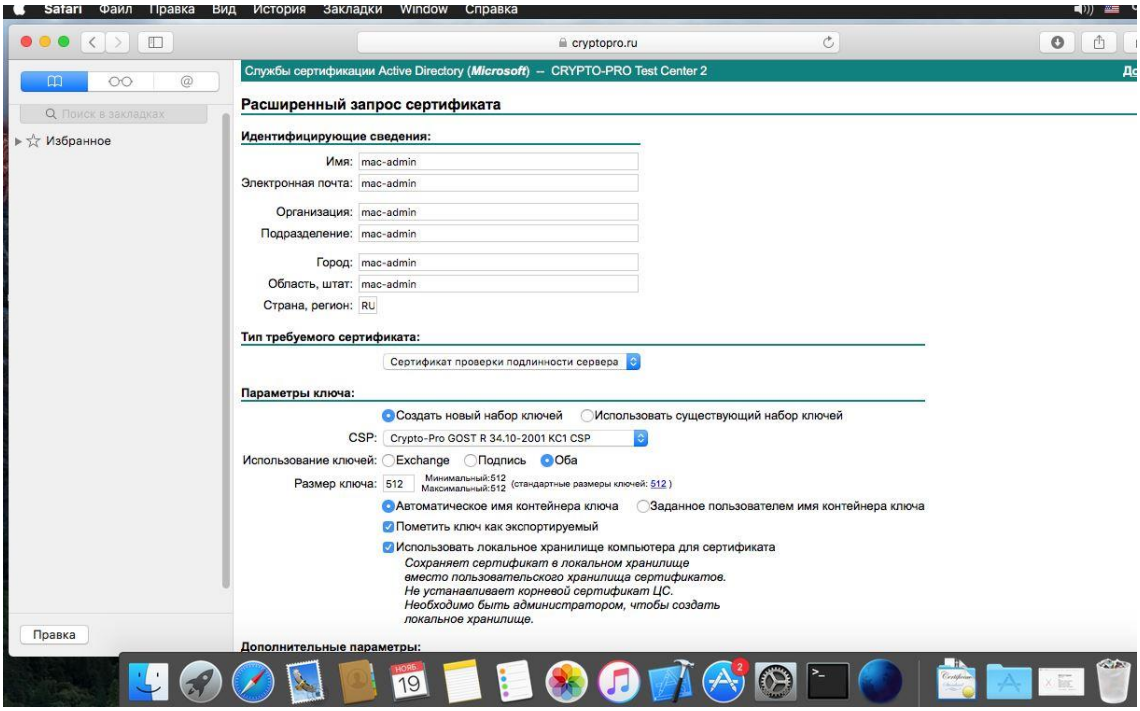


Установка производится командой

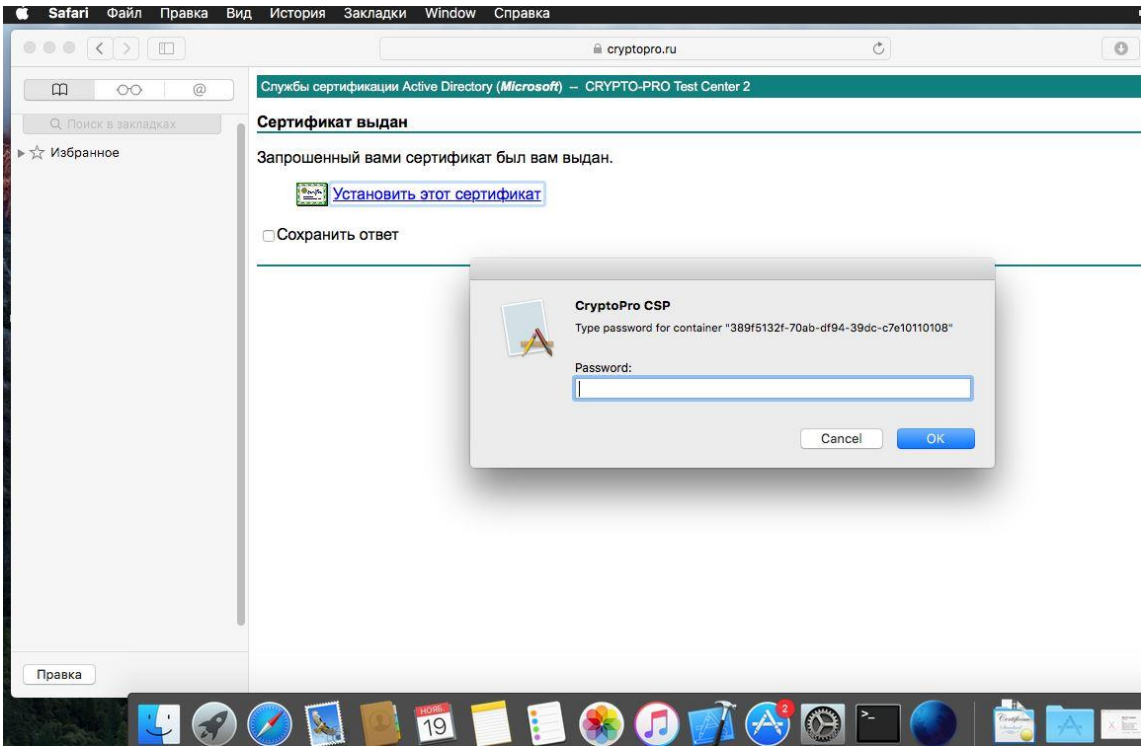
./certmgr –inst –store –root –f /Users/admin/Downloads/certnew.cer



Далее средствами тестового УЦ ООО "КРИПТО-ПРО" производится выпуск сертификата с типом “проверка подлинности сервера”. Ключевым моментном здесь является соответствие значения поля “Имя” имени самого сервера(hostname).

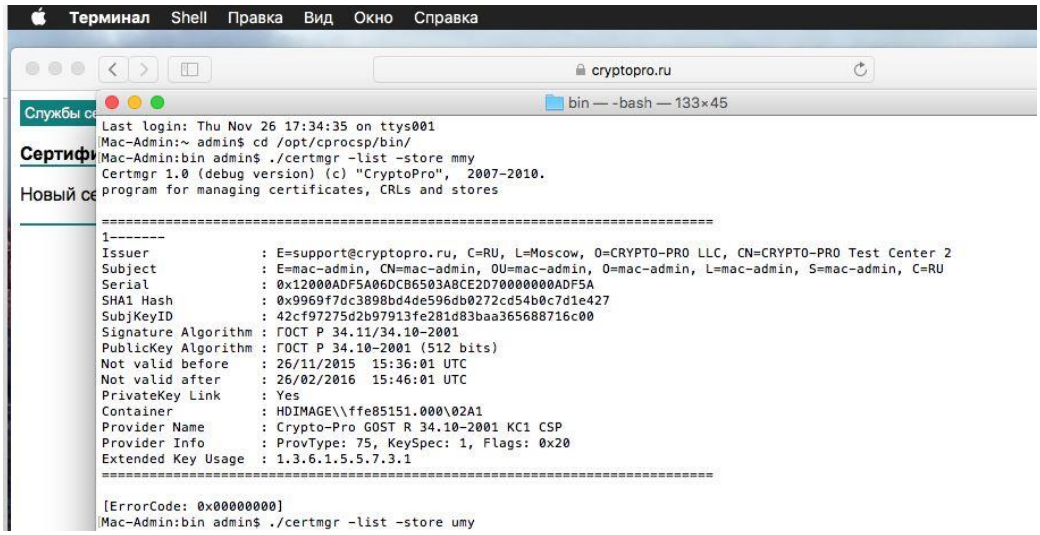


Производится генерация контейнера и задается PIN.



Проверить наличие сертификата в хранилище “Личное” можно, используя команду:

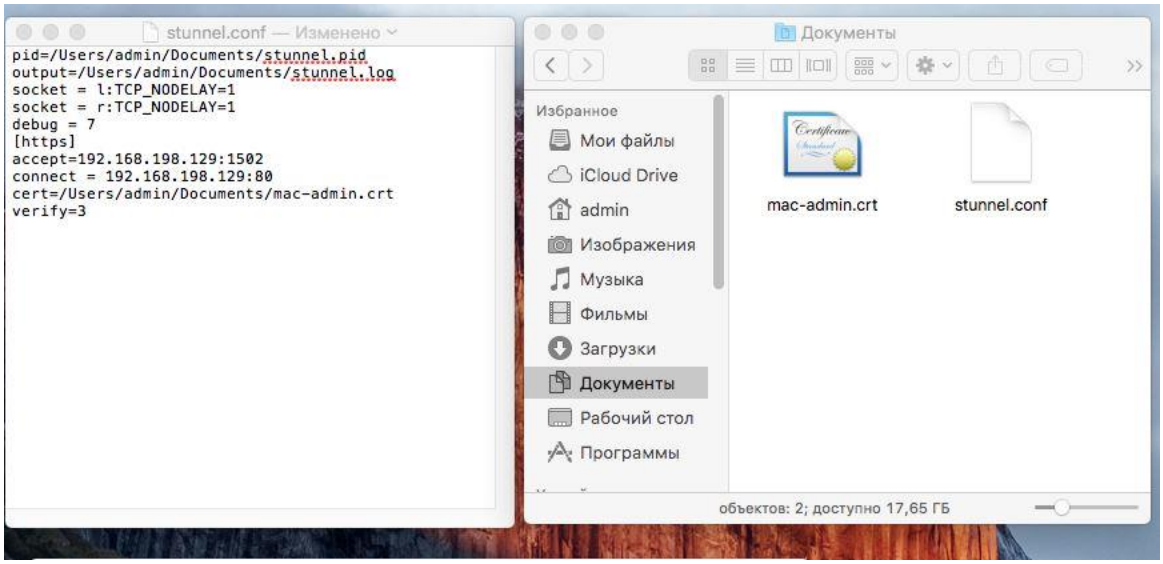
./certmgr –list –store mmy



Далее следует определить рабочую папку утилиты и произвести в неё экспорт сертификата.

* В случае, если использование тестового УЦ не предполагается, а контейнер с сертификатом сервера получен ранее и присутствует на внешнем носителе – производится копирование контейнера в папку **/var/opt/cprocsp/keys**, после чего осуществляется установка сертификата. В зависимости от типа ключевого носителя механизм копирования контейнера может отличаться. В случае работы с флеш-картой осуществляется копирование папки **[имя_контейнера].000** в указанную выше директорию с помощью команды **cp** . Если же используется к примеру защищённый ключевой носитель Рутокен S, то сначала командой **./csptestf –keyset –enum_cont –verifyc -fqcn** проверяется наличие на нём контейнеров, а затем производится копирование нужного контейнера в хранилище локального компьютера. Для этого используется команда **./ csptest -keycopy -src '\\.\Aktiv Rutoken S 00 00\имя_контейнера' -dest '\\.\HDIMAGE\имя_контейнера' –machinekeyset**
Установка сертификата производится командой **./csptestf -absorb -certs -machine**

Следующим шагом будет формирование конфигурационного файла stunnel.conf в рабочей папке программы (файл может быть сформирован в произвольном месте).



Содержимое конфигурационного файла stunnel.conf серверной части:

pid = /Users/admin/Documents/stunnel.pid

output=/Users/admin/Documents/stunnel.log

socket=l:TCP_NODELAY=1

socket=r:TCP_NODELAY=1

debug=7

[https]

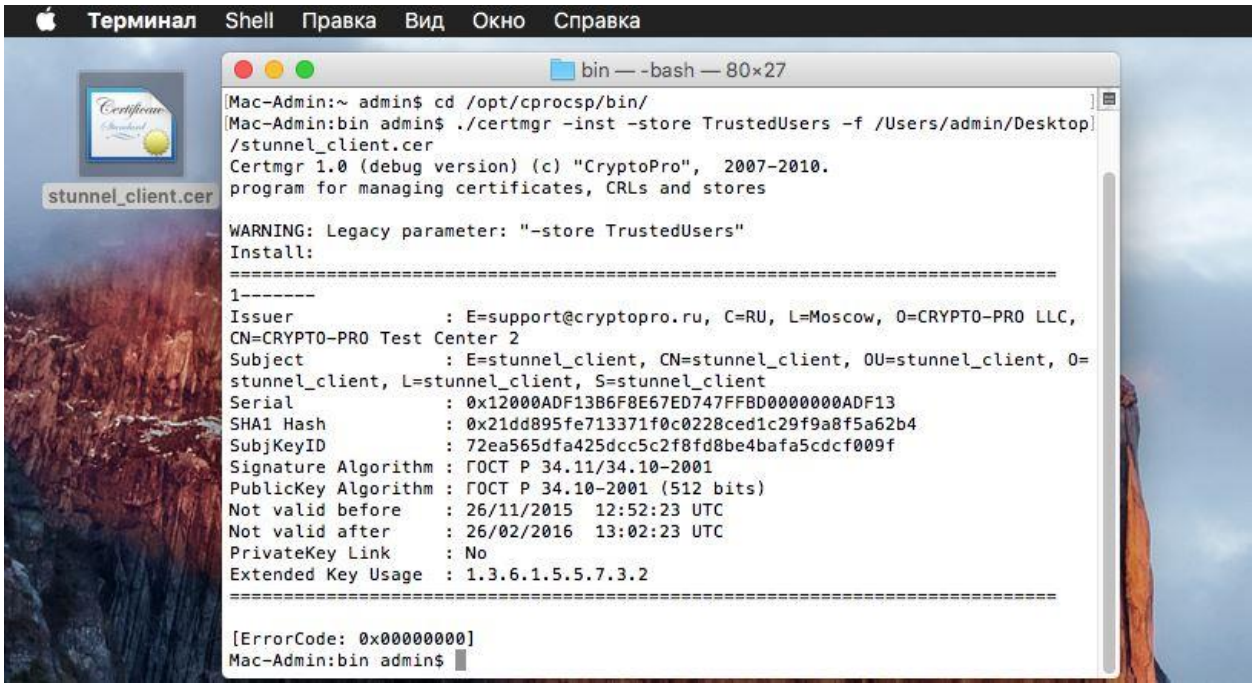
Accept=192.168.198.129:1502

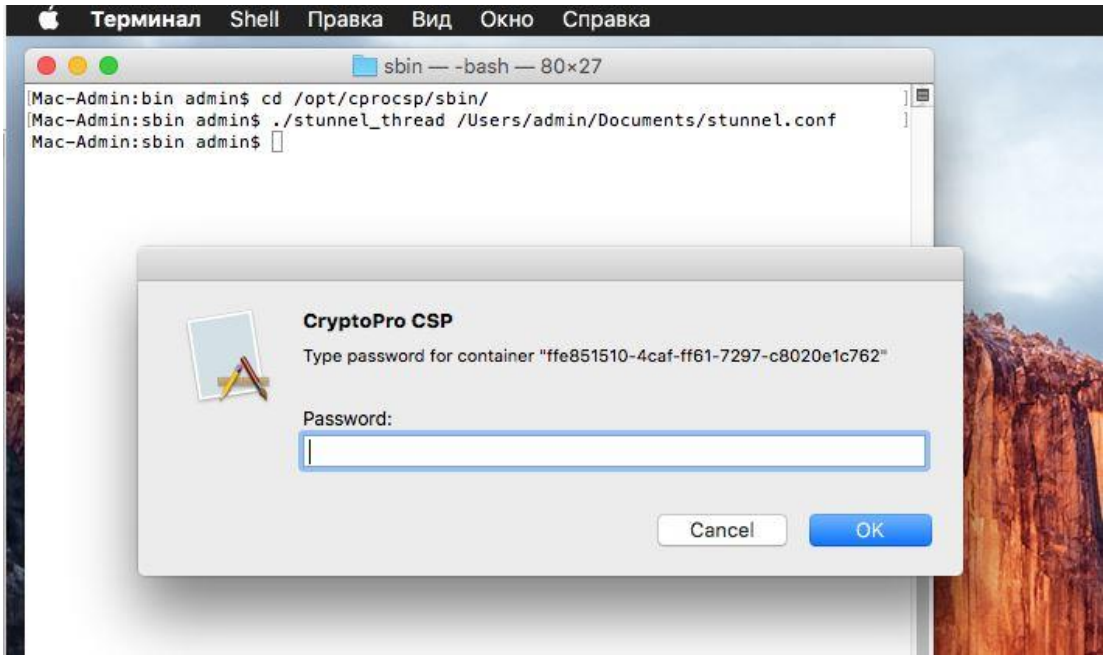
Connect=192.168.198.129:80

Cert=/Users/asdmin/Documents/mac-admin.crt

Verify=3

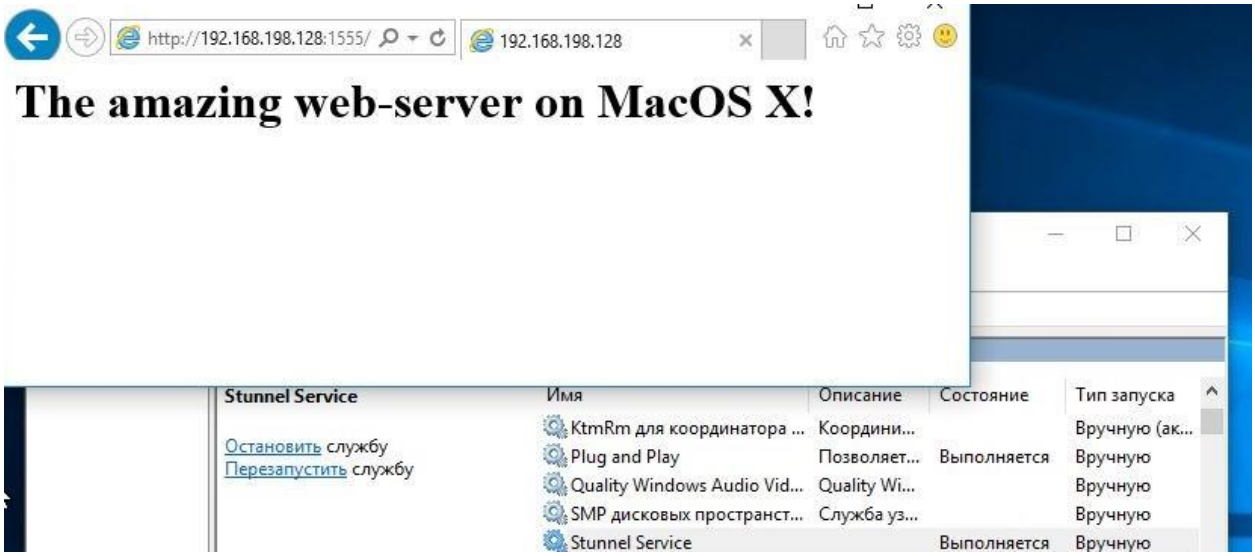
В данном примере показан случай, когда проверка клиента производится по конкретному сертификату (опция verify=3), который должен присутствовать в хранилище TrustedUsers сервера. Производится копирование сертификата на сервер, а затем установка в указанное хранилище. Сертификат устанавливается командой
./certmgr -inst -store TrustedUsers -f /Users/admin/Desktop/stunnel_client.cer





Производится запуск утилиты stunnel_thread с указанием конфигурационного файла. Если присутствующий на контейнере PIN не указан в явном виде в файле конфигурации, отобразится окно с запросом ввода.

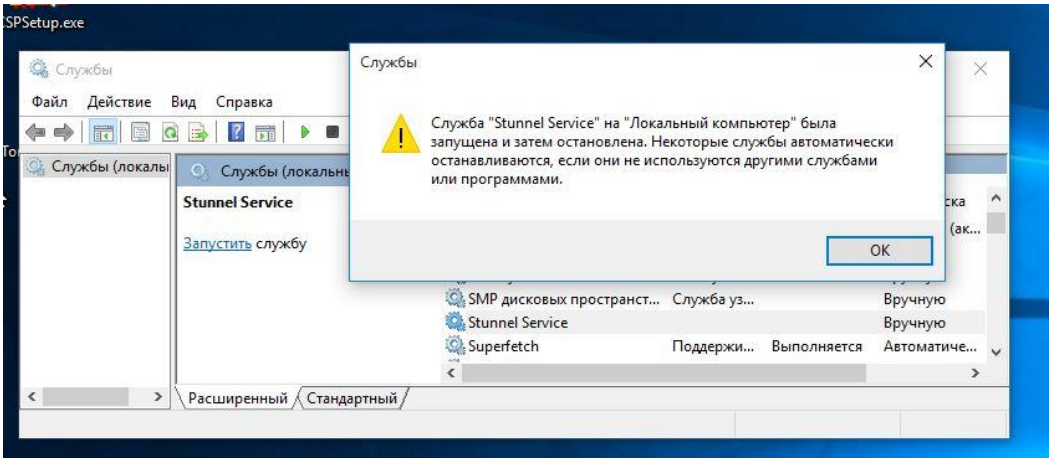
Далее необходимо проверить корректность работы. Для этого производится попытка доступа по соответствующему адресу http://192.168.198.128:1555 с клиентской машины.



Возможные ошибки в работе

В случае возникновения проблем с подключением следует изучить содержимое файла журнала событий, который создаётся автоматически при первом запуске службы по указанному в конфигурационном файле пути. При некорректно сформированном конфигурационном файле утилита не запустится, при этом файлы журнала и идентификатора процесса могут не создаваться.

1. Если в файле конфигурации PIN на контейнер указан с ошибкой - служба запустится, но при попытке соединения клиента PIN будет запрошен вновь.
2. При настройке клиента на ОС следует удостовериться в корректности значения опции **accept**. При наличии ошибок в файле конфигурации запустить службу не удастся.



3. При отсутствии в хранилище **root** корневого сертификата сервера(в случае, если сертификаты сервера и клиента выпущены разными УЦ) к которому производится подключение, в журнале событий отобразится сообщение следующего плана:

```
Error 0x20 ((unknown)) returned by CertVerifyCertificateChainPolicy!
Error 0x20 when validate certificate

Error 0x80092004 returned by VerifyCertChain
**** Error 0x80092004 authenticating server credentials!
Connection reset: 0 byte(s) sent to SSL, 0 byte(s) sent to socket
free Buffers
```

4. Если при использовании опции verify=3 в хранилище сертификатов “TrustedUsers” будет отсутствовать сертификат клиента – подключение не удастся, а в логах отобразится ошибка.

```
:56:29 LOG7[32181:123145306087424]:  Recieve 96 bytes from client on
Loop
:56:29 LOG7[32181:123145306087424]: AcceptSecurityContext finish, scRet

:56:29 LOG5[32181:123145306087424]: Send 1139 handshake bytes to client
:56:29 LOG7[32181:123145306087424]: reading in SSPINeg err = 890
:56:29 LOG7[32181:123145306087424]:  Recieve 890 bytes from client on
Loop
:56:29 LOG7[32181:123145306087424]: AcceptSecurityContext finish, scRet

:56:29 LOG3[32181:123145306087424]: Error 0x80092004 returned by
ficateInStore "TrustedUsers"

:56:29 LOG5[32181:123145306087424]: User not authorized for connect
:56:29 LOG5[32181:123145306087424]: 11 bytes of close_notify data sent
:56:29 LOG5[32181:123145306087424]: Connection reset: 0 byte(s) sent to
) sent to socket
:56:29 LOG7[32181:123145306087424]: free Buffers
:56:29 LOG7[32181:123145306087424]: delete c->hContext
:56:29 LOG5[32181:123145306087424]: incomp_mess = 0, extra_data = 0
:56:29 LOG7[32181:123145306087424]: Local socket (FD=4) closed
:56:29 LOG7[32181:123145306087424]: Service [https] finished (0 left)
:56:29 LOG7[32181:123145306087424]: str_stats: 7 block(s), 824 data
control byte(s)
:56:29 LOG7[32181:123145306087424]: str_stats: 128 byte(s) at /
anches/CSP_4_0/CSPbuild/CSP/samples/stunnel/src/network.c:430
:56:29 LOG7[32181:123145306087424]: str_stats: 128 byte(s) at /
anches/CSP_4_0/CSPbuild/CSP/samples/stunnel/src/network.c:429
```

Никитков Александр Александрович

Подписано цифровой подписью: Никитков Александр Александрович
Дата: 2016.10.26 12:37:42 +03'00'