|  |  |  |
| --- | --- | --- |
| 1. Генерация приватного ключа CА | |  |
| Картинка №1 | * используем библиотеку opensll * генерация с помощью алгоритма rsa * дополнительно используется алгоритм des3 * выходной файл – CA.key * количество бит, которые будут использоваться   + ключевая фраза  Для шифрования по алгоритму des3 (без этогоо алгоритма ключ не требуется – покажем позже) | |
| Visual Code | Открываем СА.key | |
|  |  | |
| 1. генерация сертификата CА | | |
| Картинка №2 | * применяется стандарт х509 * ключ который используется – CA.key * время которое он будет действителен – 700 дней * шифрование по алгоритму sha256 * выходной файл – CA.crt   + ключевая фраза  Для шифрования согласно с приватным ключом для того чтобы можно было использовать des3  + что пишем дополнительно   * страна * название провинции * город * имя организации * подразделение организации * имя * электронный адрес | |
| VS | CA.crt => получили шифрованный сертификат по которому будем получить шифровать запросы от ресурса на сертификат | |
|  | | |
| 1. генерация приватного ключа ресурса | | |
| Картинка №3 | С помощью библиотеки openssl   * генерируем приватный ключ с помощью алгоритма rsa * указываем выходной файл * 2048 бит будут использоваться   Т.к.не указан дополнительный алгоритм шифрования как вверху, то никаких подтверждений и фраз не надо | |
| VS | LAB.key | |
|  | | |
| 1. Генерация запроса на сертификат для ресурса | | |
| VS  Файл  LAB.cfg | * Конфигурационный файл для ресурса * Версия откуда пишем * Страна, город, имя организации, подразделение и имя * Alt\_names – какие адреса поддерживаем (зайдем на другие – предупреждение что такие не упомянуты)   Генерируем запрос на сертификат | |
| Картинка №4 | Указываем   * Ключ * Выходной файл - наш запрос * Хэширование ша256 * Конфигурационный файл   Нету фразы, т.к.используем ключ без шифрования des3 | |
| VS | LAB.csr – передаем нашему CA и он должен это подписать | |
|  | | |
| 1. Генерация сертификата для ресурса | | |
| Картинка №5 | * Используем бибилотеку openssl * Применяется стандарт X509 * Входной файл – наш запрос * Сертификат CA.crt * Ключ CA.key * Создаем серийный номер * Выходной файл * Срок действия * Хэширование * Берет данные из v3\_req * Конфигурационный файл   Получаем наш сертификат для ресурса | |
| VS | LAB.crt | |
|  | | |
| 1. Работаем с хранилищем сертификатов | | |
| Картинка №6 | В cmd вбиваем mmc  Получаем консоль с оснасткой для работы с хранилищем сертификатов | |
| Картинка №7 | Посмотрим как настроить импорт сертификата в хранилище доверенных корневых центров сертификации | |
| Картинка №8 | Указываем файл для импорта | |
| Картинка №9 | Проверяем что наш сертификат появился  POIT | |
| Картинка №10-11 | Нажимаем на наш сертификат и смотрим некоторые параметры  ОБЩИЕ: кому и кем выдан, срок годности  СОСТАВ: содержит алгоритм хэширования, подпись, серийный номер и версию, указан издатель (нажимаем) и смотрим что выводится в окошко внизу. | |
| Картинка №12-15 | Находим сертификат для нашего ресурса  СОСТАВ:  Содержит как издателя так и субъекта  Т.е.видим кто подписал и для кого  смотрим «допоплнительные имена субъекта» - это наши разрешенные домены  и можно посмотреть путь сертификации | |
| Картинка №16 | Разрешенные dns можно посмотреть и изменить в файле hosts (C:\Windows\System32\drivers\etc)  Писали соответсвия ip адреса доменному имени  Т.к.они нигде не зареганы – прописываем сами | |

Пробуем ввести localhost – предупреждение – дополнительно

Написать knv, lab22-knv – все нормально

Клиент посылает запрос на сервер. Сервер подписывает сертификат и обратно отсылает клиенту. Клиент проверяет в хранилище сертификатов данный сертификат на соответствие и отсылает шифры которые могут использоваться для передачи данных. Дальше сервер шифрует с помощью открытого ключа и передает инфу с помощью tls.

TLS-аутентификация – аутентификация, которая осуществляется на основе сертификата х509.

X509 – стандартный формат хранения и транспортироваки отрибутов безс-ти.

- эл.док, кот. выдается Центром Сертификации.

Содержит: имя держателя, адрес, серийный номер сертификата, даты проверки, открытый ключ держателя.

Вместе с сертификатом выдается секретный и публичный ключ.

Публичный находится на самом сертификате.

Секретный ключ выдается отдельно.

*Кто выдает сертификаты* – есть утвержденные сертификационные центры, которые выдают сертификаты. Можно самому – но нужно регистрировать свой сертификационный центр, чтобы он был для всех и добавлять в хранилище сертификатов.

Аутентификация – процедура, направленная на подтверждение id пользователя (логин и пароль)

Авторизация – проверка есть ли у пользователя права на выполнение тех или иных действий

Идентификация – заявление пользователя о себе.

Хэширование – преобразование входных данных в уникальную последовательность символов, из которой невозможно получить исходное сообщение. – для проверки целостности.

Шифрование – исходное сообщение становится нечитаемым для любого, кто не владеет ключом.