**Практическое занятие №33**

**SSL сертификат**

Цель работы – изучение принципов работы протокола уровня защищённых сокетов (Secure Socket Layers, SSL) на примере СУБД Microsoft SQL Server.

**Теоретический материал**

Протокол уровня защищѐнных сокетов (SSL) – криптографический протокол, предназначенный для создания конфиденциального канала связи. Протокол был создан компанией Netscape Communications и применѐн в конце девяностых годов в рамках веб-браузера Netscape Navigator для создания защищённого HTTP-соединения (HTTPS). На данный момент доказано, что этот протокол является уязвимым для атак, для решения чего был разработан протокол TLS (Transport Layer Security, защита на транспортном уровне), который сейчас находится в практическом использовании. Часто название SSL упоминается в публичных источниках информации как устоявшееся для защищённых соединений, хотя за этим названием стоит реализация TLS. Подавляющее большинство современных браузеров поддерживают протокол TLS как минимум версии 1.0. TLS в целом основан на тех же принципах, что и SSL, предназначен абсолютно для тех же целей, что и SSL, и является просто «работой над ошибками» SSL.

Протокол SSL/TLS основан на двух основных процедурах: протокол «рукопожатия» (handshake) и протокол записи. В ходе первого протокола сервер и клиент устанавливают соединение, проверяют валидность (иногда друг друга, иногда только сервера, изредка сохраняется полная анонимность) и настраивают сеансовый ключ, в ходе протокола записи идёт передача сообщений. Протокол записи достаточно прост: на основе определённого сеансового ключа (который является симметричным) идёт шифрование данных, передача их другой стороне и расшифрование данных при получении.

Протокол «рукопожатия», в общем виде, включает в себя следующие основные шаги:

1. Клиент подключается к серверу с запросом на защищённое соединение, предоставляя список поддерживаемых алгоритмов шифрования.

2. Сервер выбирает алгоритм из списка (наиболее надёжный и при этом доступный на сервере) и отправляет решение клиенту, вместе с сертификатом для аутентификации себя.

3. Клиент проверяет валидность полученного сертификата (путём обращения к указанному в сертификате центру сертификации и сверкой центра сертификации со списком доверенных центров);

4. Клиент и сервер создают общий сеансовый ключ (на основе, например, протокола Диффи-Хеллмана).

Соответственно, для создания SSL-соединения необходимы удостоверяющий сертификат сервера на сервере и наличие сертификата удостоверяющего центра сервера в списке доверенных центров сертификации у клиента.

**Ход работы**

В ходе работы будет выполнена задача соблюдения конфиденциальности данных в процессе передачи данных по открытым каналам связи с задействованием протокола SSL.

Обеспечение защищённого канала связи – непростая задача. Технически защищённый канал связи экономически оправдан только в рамках внутренней сети организации, но за её пределами проводить отдельную защищённую линию связи крайне сложно. Для защиты открытых соединений, проходящих через общую сеть, от перехвата используются криптографические протоколы, превращающие открытый текст в зашифрованный и обеспечивающие уверенность в том, что соединение происходит с доверенным сервером. Одним из таковых протоколов является SSL (TSL).

Рассматриваемая задача разделяется на следующие шаги:

1 Получение сертификата, отвечающего требованиям.

2 Введение сертификата в окружение сервера и включение режима шифрования.

3 Проверка настроенного режима.

**Шаг 1**

SQL Server реализует протокол SSL с помощью сертификатов, следовательно, асимметричного шифрования. При установке сертификата на сервере необходимо, чтобы сертификат удовлетворял следующим требованиям:

– время сертификата валидно (системное время компьютера находится между значениями времени валидности сертификата);

– сертификат выдан субъекту, название которого полностью совпадает с полным доменным именем сервера или именем компьютера-сервера;

– использование ключа сертификата включает в себя аутентификацию сервера;

– ключ допускает обмен и экспорт (создан с параметром AT\_KEYEXCHANGE);

– наличие закрытого ключа в сертификате.

Данные требования НЕ включают в себя требование вида сертификата – возможно использование любого из них. В рамках хода данной практической работы будет создан и использован самоподписанный сертификат, однако следует понимать, что такой сертификат допустимо использовать только тогда, когда нет необходимости в установлении доверенного соединения с сервером – а это только два случая: тестирование сервера и использование протокола в доверенной сети (внутренняя сеть организации, например).

Создать сертификат возможно встроенными средствами Microsoft Windows. Ранее для этого использовалось приложение MakeCert, однако начиная с Windows 10 (Windows Server 2016) функционал по генерации сертификатов входит в расширенную командную строку Windows PowerShell.

Для начала работы запустите Windows PowerShell, обязательно от имени администратора системы. Введите три следующие команды:

New-SelfSignedCertificate -DnsName WIN-8PB5C78B03Q -KeySpec KeyExchange

$Password = ConvertTo-SecureString -String “12345” -Force -AsPlainText

Export-PfxCertificate -Cert cert:\LocalMachine\My\9DAFC96BD59E10A2988A129817FEC9E430B5935F -FilePath C:\test.pfx -Password $Password

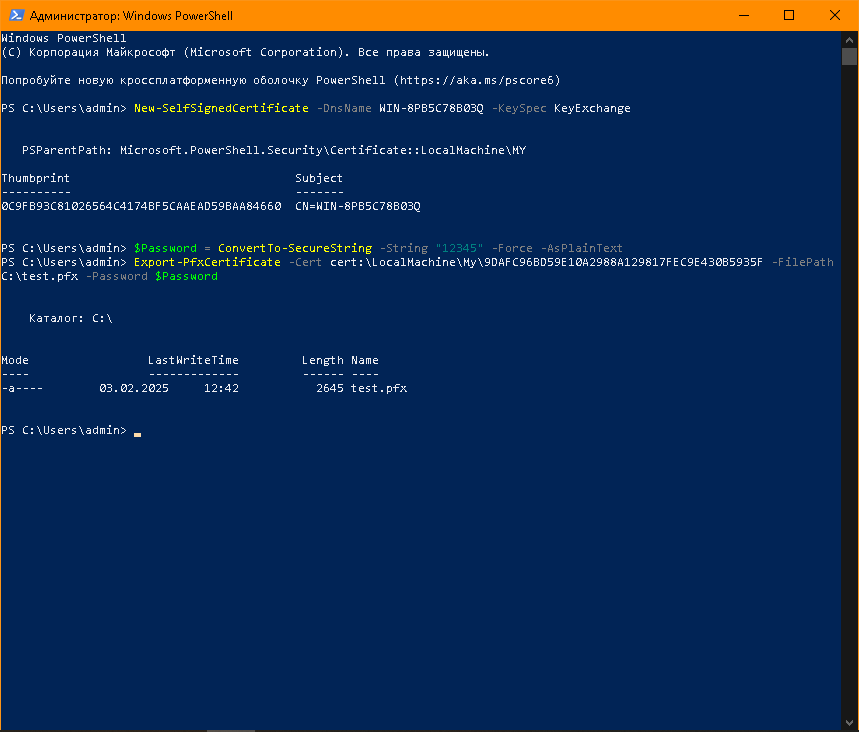


Рисунок 1 – Выполнение команд

Первая команда создаёт самоподписанный сертификат без закрытого ключа и со спецификацией ключа на обмен (остальные требования к сертификату SQL Server уже удовлетворены за счёт настроек по умолчанию) с именем виртуальной машины, вторая записывает в переменную Password пароль 12345 в формате защищённой строки (Secure String), третья экспортирует созданный сертификат по указанному пути (не забудьте указать название сертификата и расширение .pfx) с созданием закрытого ключа, защищённого заданным паролем.

Отпечаток сертификата будет выведен в консоль (под словом Thumbprint), однако, чтобы не переписывать его весь, можно открыть созданный сертификат в хранилище сертификатов и скопировать отпечаток из его состава. Чтобы открыть хранилище, с помощью окна запуска Win+R запустите mmc. Откроется окно оснасток, добавьте оснастку «Сертификаты», при определении хранилища выберите «Учётная запись компьютера», далее «Локальный компьютер». Откроется хранилище сертификатов. В хранилище найдите созданный сертификат – он будет в папке «Промежуточные корневые центры сертификации», подпапке «Сертификаты». Откройте его (двойным кликом ЛКМ) и на вкладке «Состав» найдите параметр «Отпечаток» рисунок 2), который можно скопировать. Необходимо будет удалить пробелы из отпечатка после вставки, но эта процедура гарантирует, что вы не ошибётесь при переписывании отпечатка из консоли вручную.

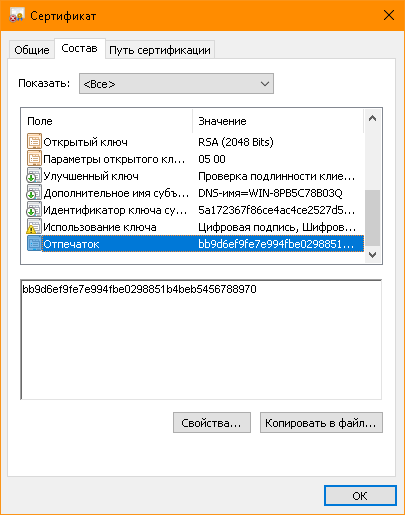


Рисунок 2 - Отпечаток сертификата

**Шаг 2**

Итак, у нас есть сертификат с закрытым ключом. Теперь необходимо импортировать его в хранилище. Для этого просто его запустите, и откроется Мастер импорта сертификатов (рисунок 3).

В первом окне Мастера укажите в качестве хранилища локальный компьютер. Второе окно оставьте без изменений. В третьем окне введите пароль, указанный при определении переменной пароля (12345 в примере) (рисунок 4).

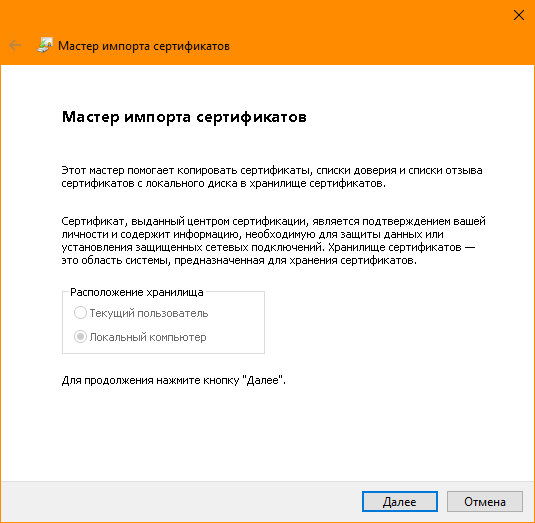


Рисунок 3 - Мастер импорта сертификатов

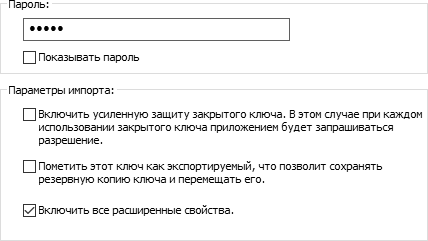


Рисунок 4 - Ввод пароля

Остальные параметры в третьем окне оставьте без изменений.

В четвёртом окне Мастер предложит указать папку (хранилище), куда будет положен сертификат. По умолчанию сертификат будет положен в папку «Личное» – именно там и должен находиться наш сертификат.

Теперь необходимо выдать доступ службе Microsoft SQL Server к этому сертификату. Для этого, находясь в окне хранилища сертификатов, в контекстном меню этого сертификата (который должен быть в папке «Личное») выберите последовательно пункты «Все задачи» -> «Управление закрытыми ключами». В открывшемся окне укажите пользователя NT Service/MSSQL$SQLSERVER и определите права на полный доступ. Если пункта «Управление закрытыми ключами» нет, значит, ключ находится в хранилище текущего пользователя, и его необходимо поместить в хранилище локального компьютера.

Чтобы сертификат мог быть определён клиентом как допустимый, необходимо положить сертификат удостоверяющего центра (в нашем случае сертификат самоподписанный, следовательно, это будет тот же самый сертификат) в хранилище «Доверенные корневые центры сертификации». Созданный нами изначально сертификат, который находится в папке «Промежуточные центры сертификации», содержащий только открытый ключ, как раз является таким сертификатом. Необходимо перенести этот сертификат в папку доверенных центров в окне хранилища сертификатов.

Теперь в хранилище есть два созданных сертификата – убедитесь, что у сертификата в папке доверенных центров нет закрытого ключа, а в личной папке – есть, что отображается на вкладке «Общие» в окне сертификата (рисунки 5 и 6).

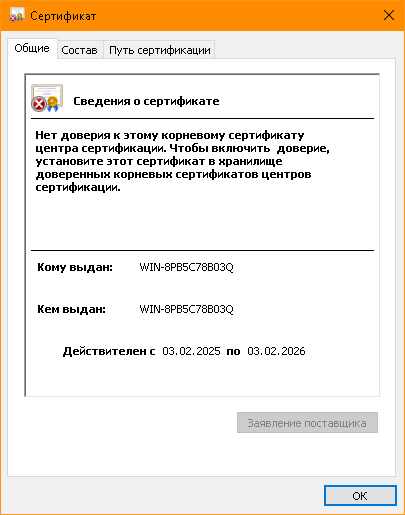


Рисунок 5 - Сертификат без закрытого ключа

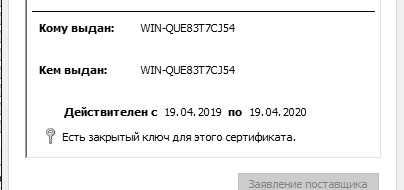
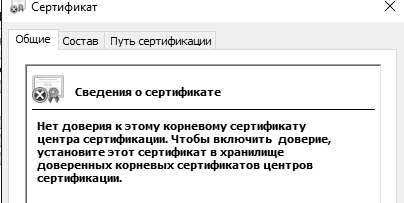


Рисунок 6 - Сертификат с закрытым ключом

Итак, нужные сертификаты добавлены в хранилище. Теперь необходимо указать SQL Server, какой именно сертификат следует использовать для шифрования соединений. Чтобы это сделать, откройте Диспетчер конфигурации SQL Server и с помощью контекстного меню откройте свойства вкладки «Протоколы для MSSQLSERVER» (рисунок 7). В окне свойств на вкладке «Флаги» включите «Принудительное шифрование», а на вкладке «Сертификат» – укажите добавленный сертификат. Если сертификат не отображается для выбора, где-то была допущена ошибка в создании и/или добавлении.

**Шаг 3**

Чтобы проверить соединение, необходимо от хоста (основной машины) или другой виртуальной машины, с настроенным соединением его с виртуальной машиной-сервером, присоединиться к серверу с помощью Microsoft SQL Server Management Console (SSMS), или иному приложению, с помощью которого такое соединение возможно.

В первую очередь необходимо добавить сертификат доверенного центра сертификации сертификата сервера (поскольку это самоподписанный сертификат, это опять тот же самый созданный нами сертификат) в хранилище сертификатов на стороне клиента. Для этого следует экспортировать сертификат с сервера на клиент и добавить его в хранилище. Для этого достаточно сертификата с открытым ключом (без закрытого), но можно взять и сертификат с закрытым ключом. Добавляется сертификат в хранилище так же, как и на стороне сервера, однако обратите внимание, что сертификат необходимо добавить в папку доверенных сертификатов (рисунок 8).

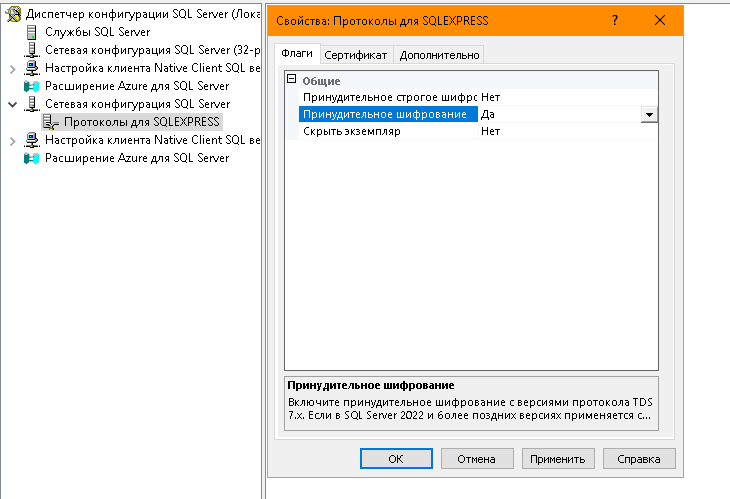


Рисунок 7 - Свойства протоколов

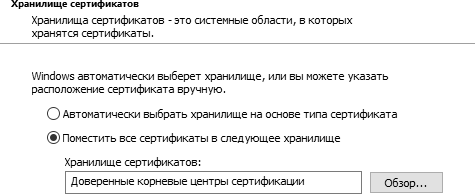


Рисунок 8 - Определение хранилища для сертификата

Теперь необходимо соединиться с сервером. Для этого, при подключении к базе данных, в качестве имени сервера используйте «tcp:WIN- QUE83T7CJ54», что означает подключение к машине с указанным именем по протоколу TCP (рисунок 9). Поскольку клиент не добавлен в список пользователей СУБД, воспользуйтесь учётной записью sa или иной другой учётной записью с входом по логину/паролю.

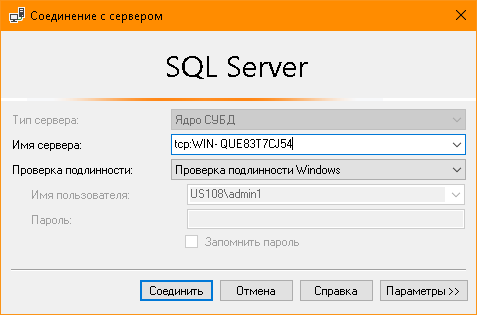


Рисунок 9 - Соединение с сервером

В зависимости от настроек на сервере, клиент может подключаться как с защищённым соединением, так и без него. Указанный ранее параметр «Принудительное шифрование» уже настроил обязательное шифрование со стороны сервера. Чтобы со стороны клиента использовать именно зашифрованное соединение, следует открыть «Параметры >>» в окне соединения и на вкладке «Свойства соединения» отметить пункт «Шифровать соединение» (рисунок 10). Убедитесь, что удаётся подключиться к серверу с использованием SSL.

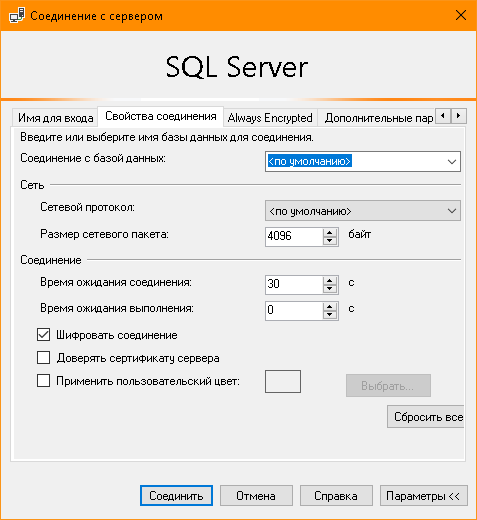


Рисунок 10 - Принудительное применение шифрования со стороны клиента

**Контрольные вопросы**

**1 Как расшифровывается SSL и TLS? Что такое SSL и в чём принципиальное отличие от TLS?**

SSL - Secure Socket Layers

TLS - Transport Layer Security

Протокол уровня защищѐнных сокетов (SSL) – криптографический протокол, предназначенный для создания конфиденциального канала связи.

На данный момент доказано, что этот протокол является уязвимым для атак, для решения чего был разработан протокол TLS (Transport Layer Security, защита на транспортном уровне), который сейчас находится в практическом использовании.

**2 Какие протоколы (фазы) используются в процессе реализации SSL? Кто участвует в SSL-соединении?**

Протокол SSL/TLS основан на двух основных процедурах: протокол «рукопожатия» (handshake) и протокол записи. В ходе первого протокола сервер и клиент устанавливают соединение, проверяют валидность (иногда друг друга, иногда только сервера, изредка сохраняется полная анонимность) и настраивают сеансовый ключ, в ходе протокола записи идёт передача сообщений. Протокол записи достаточно прост: на основе определённого сеансового ключа (который является симметричным) идёт шифрование данных, передача их другой стороне и расшифрование данных при получении.

**3 Опишите шаги первого протокола (фазы) в SSL.**

Протокол «рукопожатия», в общем виде, включает в себя следующие основные шаги:

1. Клиент подключается к серверу с запросом на защищённое соединение, предоставляя список поддерживаемых алгоритмов шифрования.

2. Сервер выбирает алгоритм из списка (наиболее надёжный и при этом доступный на сервере) и отправляет решение клиенту, вместе с сертификатом для аутентификации себя.

3. Клиент проверяет валидность полученного сертификата (путём обращения к указанному в сертификате центру сертификации и сверкой центра сертификации со списком доверенных центров);

4. Клиент и сервер создают общий сеансовый ключ (на основе, например, протокола Диффи-Хеллмана).

**4 Что необходимо для создания SSL-соединения?**

Для создания SSL-соединения необходимы удостоверяющий сертификат сервера на сервере и наличие сертификата удостоверяющего центра сервера в списке доверенных центров сертификации у клиента.

**5 Какие существуют виды сертификатов? Чем они отличаются друг от друга? Для чего используются?**

**Существуют следующие виды SSL-сертификатов**:

1. **Сертификаты с проверкой домена (DV)**. Самый простой вид, подтверждает только владение доменом. Подходит для блогов и личных сайтов.
2. **Сертификаты с проверкой организации (OV)**. Проверяют также данные о компании, что повышает доверие клиентов.
3. **Сертификаты с расширенной проверкой (EV)**. Наиболее надёжные сертификаты, которые отображают зелёную строку в браузере и подходят для крупных компаний и финансовых организаций.
4. **Wildcard SSL**. Защищает не только основной домен, но и все его поддомены, что особенно удобно для сайтов с разветвлённой структурой.
5. **Многодоменные SSL (SAN)**. Позволяют защитить несколько доменов одним сертификатом.

**Использование сертификатов** помогает обеспечить конфиденциальность информации, предотвращая её перехват злоумышленниками. Без SSL данные пользователя (например, логины, пароли, платёжные данные) могут быть скомпрометированы.

**6 Какие требования предъявляются к сертификатам в SQL Server?**

SQL Server реализует протокол SSL с помощью сертификатов, следовательно, асимметричного шифрования. При установке сертификата на сервере необходимо, чтобы сертификат удовлетворял следующим требованиям:

– время сертификата валидно (системное время компьютера находится между значениями времени валидности сертификата);

– сертификат выдан субъекту, название которого полностью совпадает с полным доменным именем сервера или именем компьютера-сервера;

– использование ключа сертификата включает в себя аутентификацию сервера;

– ключ допускает обмен и экспорт (создан с параметром AT\_KEYEXCHANGE);

– наличие закрытого ключа в сертификате.