

ХОСТИНГ, СЕРВЕРНОЕ АДМИНИСТРИРОВАНИЕ*, АДМИНИСТРИРОВАНИЕ ДОМЕННЫХ ИМЕН*, NGINX*, APACHE*

Полное руководство по переходу с HTTP на HTTPS

ПЕРЕВОД

m1rko 3 июля 2017 в 23:51  58,1k

Оригинал: [Vladislav Denishev](#)

В наше время HTTPS обязателен для каждого веб-сайта: пользователи ищут замочек в адресной строке, когда передают личные данные; [Chrome](#) и [Firefox](#) недвусмысленно помечают как небезопасные веб-сайты с формами на страницах без HTTPS; это влияет на [позиции в поисковой выдаче](#) и оказывает серьёзное [влияние на приватность](#) в целом. Кроме того, сейчас имеется несколько вариантов получить бесплатный сертификат, так что переход на HTTPS — всего лишь вопрос желания.

Установка HTTPS может немного пугать неподготовленного пользователя — она требует многих шагов с участием различных сторон, а также специфических знаний криптографии и серверных конфигураций, да и вообще в целом кажется сложной.

В этом руководстве я объясню отдельные компоненты и шаги и ясно изложу каждый этап установки. У вас должно всё пройти гладко, особенно если ваш хостер сам предоставляет сертификаты HTTPS — тогда высока вероятность, что вы быстро и просто всё сделаете не выходя из панели управления.

Сюда включены детальные инструкции для владельцев виртуального хостинга на cPanel, администраторов серверов Apache HTTP и nginx под Linux и Unix, а также Internet Information Server под Windows.

Начнём с основ.

HTTP, HTTPS, HTTP/2, SSL, TLS: где что?

Для описания процесса коммуникации между клиентом и сервером используется много акронимов. Люди, незнакомые с технической сутью, часто путают их.

Hypertext Transfer Protocol (HTTP) — основной протокол связи, который должны поддерживать клиент и сервер, чтобы установить соединение. Он описывает такие понятия как запросы и ответы, сессии, кэширование, аутентификация и др. Работу над протоколом, а также над языком гипертекстовой разметки Hypertext Markup Language (HTML) начал в 1989 году [сэр Тим Бернерс-Ли](#) и его группа в ЦЕРН. Первая официальная версия протокола (HTTP 1.0) вышла в 1996 году, а вскоре в 1997 году появилась версия HTTP 1.1, которая широко используется сегодня.

Протокол передаёт информацию между браузером и сервером в чистом тексте, позволяя видеть эту информацию в сети, через которую она проходит. Это проблема безопасности, поэтому был изобретён **HTTP Secure (HTTPS)**, позволяющий клиенту и серверу устанавливать зашифрованный канал связи, а затем передавать

сообщения чистым текстом по этому каналу, эффективно защищая их от прослушивания.

Термины SSL и TLS часто используются как взаимозаменяемые, поскольку TLS 1.0 приходит на место SSL 3.0. Сам SSL был разработан в компании Netscape, а TLS — это стандарт IETF. На момент написания этой статьи все версии SSL (1.0, 2.0, 3.0) не рекомендуются для использования из-за различных проблем с безопасностью, и современные браузеры выводят предупреждения об этом. Из стандарта TLS используются версии 1.0, 1.1 и 1.2, а версия 1.3 сейчас на стадии черновика.

Так что где-то между 1996 и 1997 годами мы получили текущую стабильную версию Интернета (HTTP 1.1 с или без SSL и TLS), которая по-прежнему поддерживается на большинстве современных веб-сайтов. Ранее HTTP использовался для несущественного трафика (например, чтения новостей), а HTTPS применяли для важного трафика (например, аутентификации и электронной коммерции): однако увеличение значения приватности привело к тому, что браузеры вроде Google Chrome сейчас помечают веб-сайты HTTP как «не конфиденциальные» и в будущем будут выводить новые предупреждения для них.

В следующем обновлении протокола HTTP — **HTTP/2** — которую поддерживает всё большее количество сайтов, реализованы новые функции (сжатие, мультиплексирование, приоритет разного трафика), чтобы уменьшить задержки и увеличить производительность и безопасность.

В HTTP версии 1.1 безопасное соединение является необязательным (у вас может быть HTTP и/или HTTPS независимо друг от друга), в то время как в HTTP/2 оно на практике обязательно — даже хотя стандарт допускает HTTP/2 без TLS, но большинство разработчиков браузеров заявили, что они [реализуют поддержку HTTP/2 только через TLS](#).

Что даёт HTTPS?

Почему в первую очередь стоит думать о HTTPS? Его внедряют по трём основным причинам:

- **Конфиденциальность**

В открытой среде, такой как Интернет, он защищает коммуникации между двумя сторонами. Например, в отсутствие HTTPS владелец точки доступа WiFi может видеть приватные данные, такие как кредитные карты, если пользователь этой точки доступа совершает покупки в онлайне.

- **Целостность**

Он гарантирует, что информация достигнет адресата в полном и нетронутом виде. Например, наш друг с точкой доступа WiFi может добавить дополнительную рекламу на наш сайт, снизить качество изображений для экономии трафика или изменить содержимое статей, которые мы читаем. HTTPS гарантирует, что веб-сайт не может быть изменён.

- **Подлинность**

Он гарантирует, что веб-сайт в реальности является тем, за кого себя выдаёт. Например, тот же самый владелец точки доступа WiFi мог бы отправлять браузеры на поддельный сайт. HTTPS

гарантирует, что веб-сайт, который представляется как `example.com`, действительно является `example.com`.

Некоторые сертификаты даже проверяют правовую идентичность владельца веб-сайта, так что вы знаете, что `yourbank.com` принадлежит YourBank, Inc.

Криптография в основе

Конфиденциальность, целостность и проверка подлинности не являются уникальными особенностями HTTPS: это ключевые концепции криптографии. Давайте посмотрим на них более внимательно.

Конфиденциальность

Конфиденциальность представляет собой приватность — так и есть, она защищает информацию от чтения посторонними лицами. Обычно этот процесс предусматривает перевод информации из читаемой формы (в том числе аудио и видео), которая называется **открытый текст**, в зашифрованную, нечитаемую форму, которая называется **шифротекст**. Этот процесс именуется **шифрованием**. Обратный процесс — превращение нечитаемого шифротекста обратно в читаемый открытый текст — называется **расшифровкой**. Для шифрования и расшифровки информации существует много методов — **шифровальных функций** (или **алгоритмов**).

Чтобы две стороны могли общаться, они должны прийти к

согласию по двум вопросам:

1. Какой алгоритм (шифровальную функцию) использовать в коммуникации.
2. Какие параметры, пароли или правила (то есть **секрет**) будут использоваться с выбранным методом.

Есть два основных метода шифрования:

- **симметричное**

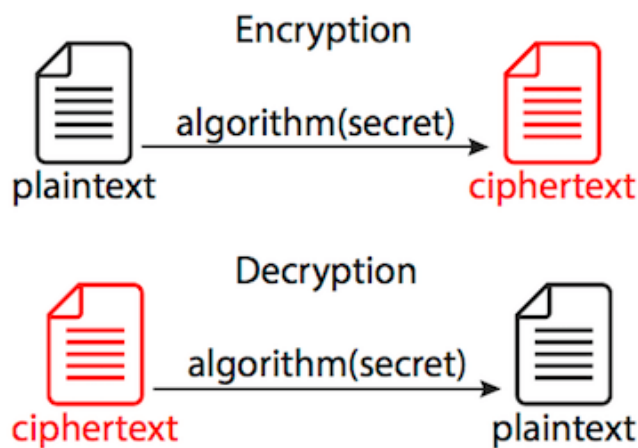
Обе стороны владеют **общим секретным ключом**.

- **асимметричное**

Одна из сторон владеет **парой из публичного и секретного ключей**, представляя основу **инфраструктуры публичных ключей** (public key infrastructure, PKI).

Симметричные методы шифрования полагаются на то, что обе стороны владеют одним и тем же секретом, который отправитель использует для шифрования, а получатель — для расшифровки тем же методом и тем же ключом (см. иллюстрацию внизу).

Проблема этих методов состоит в том, как сторонам договориться (то есть обменяться) о секретном ключе, не встречаясь физически друг с другом — им нужно установить какой-то безопасный канал связи.



Симметричное шифрование (см. [большую версию](#))

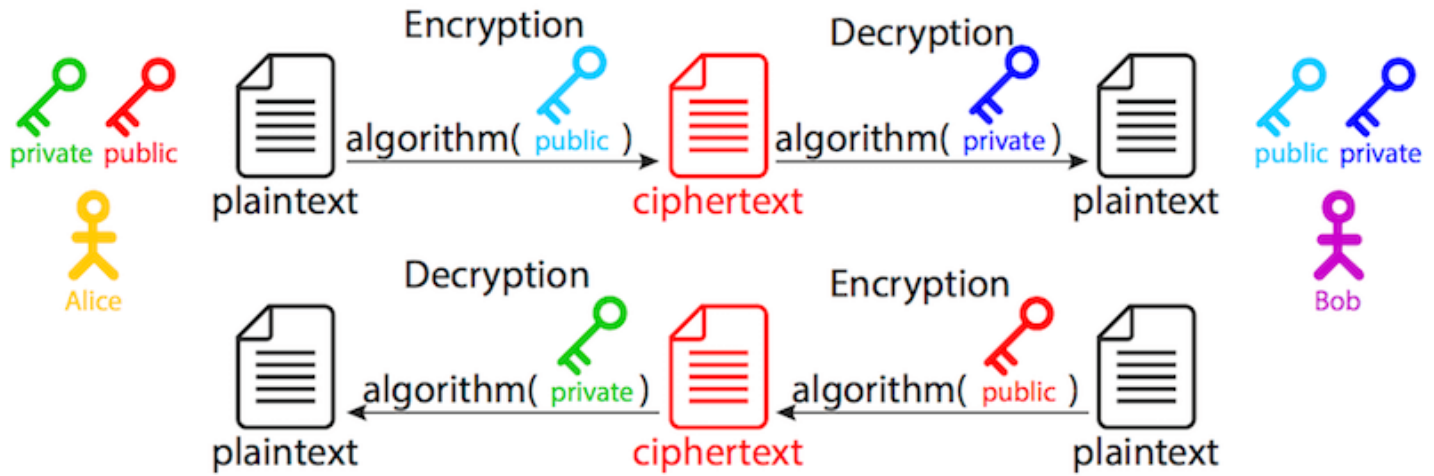
Асимметричные методы решают проблемы такого рода — они основаны на понятии публичных и секретных ключей. Открытый текст шифруется одним ключом и может быть расшифрован с использованием парного ему ключа.

Итак, как это работает? Предположим, что у нас есть две стороны, которые желают безопасно общаться друг с другом — Алиса и Боб (в каждом учебнике всегда используются такие имена вымышленных персонажей, в справочниках по безопасности и прочих, мы уважаем эту традицию). У каждого из них есть своя пара ключей: один секретный, а один публичный. Секретные ключи известны только их соответствующим владельцам; публичные ключи открыты для всех.

Если Алиса хочет отправить сообщение Бобу, она должна раздобыть его публичный ключ, зашифровать открытый текст и отправить ему шифротекст. Он затем использует свой секретный ключ для расшифровки.

Если Боб хочет отправить сообщение Алисе, он должен раздобыть её публичный ключ, зашифровать открытый текст и отправить ей

шифротекст. Она затем использует свой секретный ключ для расшифровки.



Асимметричное шифрование (см. большую версию)

Когда мы используем симметричное, а когда асимметричное шифрование?

Асимметричное шифрование используется для обмена секретом между клиентом и сервером. В реальной жизни нам обычно не нужна двусторонняя асимметричная коммуникация — достаточно, если одна из сторон (назовём её **сервер** для простоты) владеет набором ключей, так что она может **получать** зашифрованные сообщения. В реальности это защищает информацию только в **одном направлении** — от клиента к серверу, потому что информация, зашифрованная публичным ключом, может быть расшифрована только с помощью парного ему секретного ключа: значит, только сервер может её расшифровать. Другое направление не защищено — информация, зашифрованная секретным ключом сервера, может быть

расшифрована его публичным ключом, который открыт для всех. Другая сторона (также для простоты назовём её **клиент**) начинает коммуникацию с шифрования случайно сгенерированного секрета сессии публичным ключом сервера, затем отправляет шифротекст обратно на сервер, который, в свою очередь, расшифровывает его с помощью своего секретного ключа, и теперь владеет секретом.

Затем для защиты реальных данных при передаче используется симметричное шифрование, поскольку оно гораздо быстрее асимметричного. Обменявшись секретом, только две стороны (клиент и сервер) могут шифровать и расшифровывать информацию.

Вот почему первая асимметричная часть рукопожатия также известна как **обмен ключами** и вот почему реально зашифрованные коммуникации используют алгоритмы, известные как **методы шифрования**.

Целостность

Другая проблема, которую решает HTTPS — это **целостность данных**: 1) гарантия, что вся информация доставляется целиком; 2) гарантия, что никто не изменяет информацию при её передаче. Для обеспечения целостной передачи информации используются алгоритмы **дайджеста сообщения**. Вычисление **кодов аутентификации сообщений (MAC)** для каждого сообщения при обмене — это процесс **криптографического хеширования**. Например, для получения MAC (иногда он называется **тегом**) используется метод, который гарантирует практическую

невозможность (иногда используется термин **невыполнимость**) осуществить следующее:

- изменить сообщение, не затронув тег,
- сгенерировать одинаковый тег для двух различных сообщений,
- обратить процесс и получить оригинальное сообщение из тега.

Проверка подлинности

Что насчёт **аутентичности**? Проблема с реальными приложениями публичной инфраструктуры ключей состоит в том, что ни у одной стороны нет способа узнать, кем на самом деле является вторая сторона — они физически разделены друг от друга. Чтобы доказать свою аутентичность второй стороне, вовлекается **третья сторона, имеющая взаимное доверие — центр сертификации (СА)**. Этот СА выпускает **сертификат** с подтверждением того, что доменное имя `example.com` (уникальный **идентификатор**) связано с публичным ключом `xxx`. В некоторых случаях (с сертификатами EV и OV — см. ниже) СА также проверяет, что конкретная компания контролирует этот домен. Эта информация гарантирована (то есть сертифицирована) центром сертификации X, и эта гарантия действует не раньше, чем дата Y (то есть сертификат начинает действовать с этой даты), и не позже чем дата Z (то есть сертификат заканчивает своё действие в эту дату). Вся эта информация включена в один документ, который называется **сертификат HTTPS**. Чтобы привести легко понимаемую аналогию — это как ID или паспорт, который выдаёт правительство страны (то есть третья сторона, которой все доверяют) — и все, кто доверяют правительству, будут

также доверять сертификату (паспорту) его владельца и самому владельцу. Предполагается, конечно, что паспорт не поддельный, но подделка сертификатов выходит за рамки данной статьи.

Центры сертификации — это организации, которым доверяют подписать сертификаты. В операционных системах, таких как Windows, macOS, iOS и Android, а также в браузере Firefox есть список доверенных сертификатов.

Вы можете проверить, каким центрам сертификации доверяет ваш браузер:

- **Firefox**

“Options” → “Advanced” → “Certificates” → “View Certificates” → “Authorities”

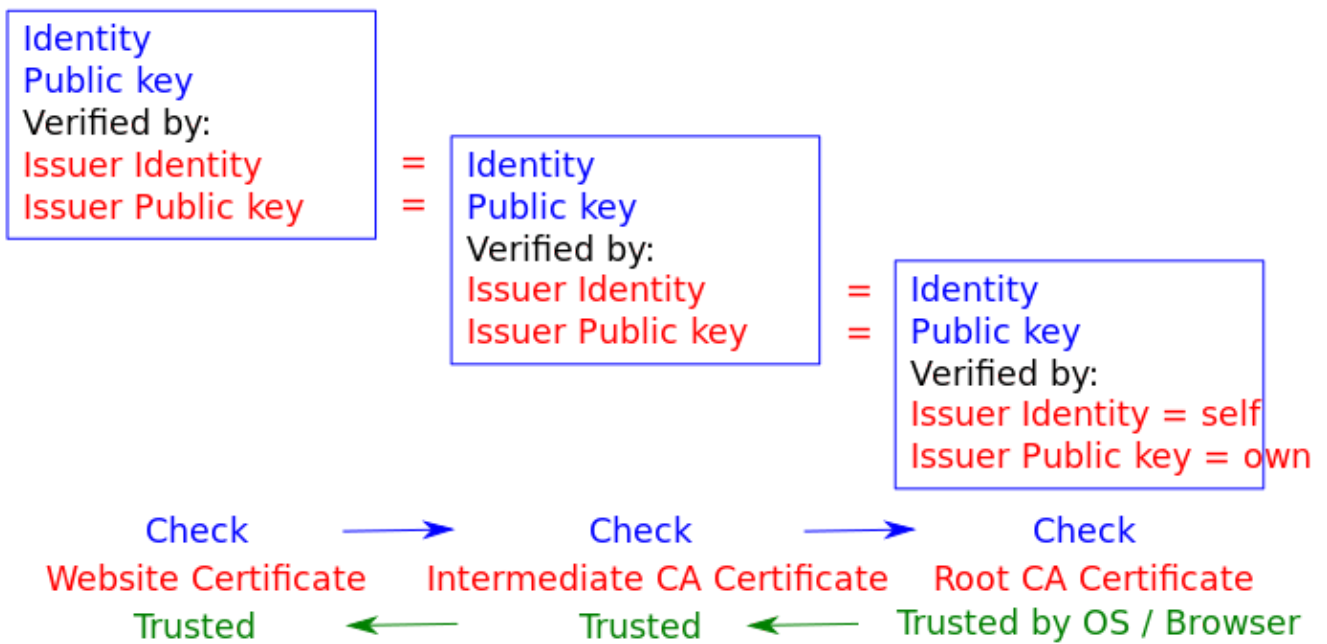
- **Windows**

“Control Panel” → “Internet Options” → “Content” — “Certificates” → “Trusted Root Certification Authorities / Intermediate Certification Authorities”

- **Mac**

“Applications” → “Utilities” → “Keychain Access.” В “Category” выберите “Certificates”

Все сертификаты затем проверяются и становятся доверенными. Проверка осуществляется либо операционной системой, либо браузером — доверие устанавливается напрямую или через проверенную доверенную сторону. Механизм передачи доверия известен как [цепочка доверия](#):



Цепочка доверия (см. большую версию)

Вы можете добавить дополнительные центры сертификации, что полезно при работе с самоподписанными сертификатами (которые мы обсудим позже).

В большинстве типичных ситуаций клиенту нужно подтвердить идентичность сервера — например, сайта электронной коммерции для покупателей — так что только этому веб-сайту нужен сертификат. В других ситуациях, таких как системы электронного правительства, одновременно и клиент, и сервер должны доказать свою идентичность. Это означает, что обеим сторонам нужно предъявить сертификаты для проверки подлинности. Такая система тоже выходит за рамки этой статьи.

Типы сертификатов HTTPS

Существует несколько типов сертификатов HTTPS. Их можно

классифицировать по следующим критериям.

1. Проверка подлинности

1. Подтверждённый домен (DV)

Самый распространённый тип сертификатов DV подтверждает, что домен соответствует определённому публичному ключу. Браузер устанавливает безопасное соединение с сервером и демонстрирует значок закрытого замка. Нажатие по значку выводит сообщение «Этот веб-сайт не предоставил информации о владельце». Для получения этого сертификата не выдвигается никаких дополнительных требований, кроме владения доменом — сертификат DV просто гарантирует, что для этого домена предъявлен правильный публичный ключ. Браузер не показывает название юридического лица. Сертификаты DV часто дешёвы (\$10 в год) или бесплатны — см. разделы о [Let's Encrypt](#) и [Cloudflare](#) ниже.

2. Расширенное подтверждение (EV)

Сертификаты EV подтверждают юридическое лицо, которому принадлежит веб-сайт. Это самый заслуживающий доверия тип сертификатов. Его выдают после того, как центр сертификации проверит юридическое лицо, которое контролирует домен. Юридическое лицо проверяется по нескольким условиям:

- управление доменом (наличие сертификата DV);
- государственный реестр для проверки, что компания зарегистрирована и действительна;
- независимые бизнес-справочники, такие как Dunn и Bradstreet, connect.data.com от Salesforce, Yellow Pages и др.;

- проверочный телефонный звонок;
- проверка всех доменных имён в сертификате (подстановочные символы явно запрещены в сертификатах EV).

Как и значок закрытого замка, сертификаты EV HTTPS показывают перед URL название проверенного юридического лица — обычно зарегистрированной компании. Некоторые устройства, такие как iOS Safari, показывают только подтверждённое юридическое лицо, полностью игнорируя URL. Нажатие на значок покажет подробности об организации, такие как полное название и юридический адрес. Стоимость этих сертификатов составляет от \$150 до \$300 в год.

3. Подтверждённая организация (OV)

Как и EV, сертификаты OV подтверждают юридическое лицо, которому принадлежит веб-сайт. Но в отличие от EV, сертификаты OV HTTPS не отображают название подтверждённого юридического лица в пользовательском интерфейсе. В результате, сертификаты OV не так популярны, поскольку у них высокие требования для проверки, но они не дают преимуществ, видимых для пользователя. Стоимость составляет от \$40 до \$100 в год.

2. Количество покрываемых доменов

В давние времена сертификаты HTTPS обычно содержали в поле CN единственный домен. Позже было добавлено «альтернативное имя субъекта» (SAN), чтобы один сертификат покрывал и дополнительные домены. В наши дни все сертификаты HTTPS

создаются одинаково: даже в сертификате на единственный домен будет поле SAN для этого единственного домена (и второе поле SAN для версии `www` этого домена). Однако многие продавцы по историческим причинам по-прежнему продают сертификаты HTTPS на один и несколько доменов.

1. Один домен

Это самый распространённый тип сертификата, действительный для доменных имён `example.com` и `www.example.com`.

2. Несколько доменов (UCC/SAN)

Этот тип сертификата, также известный как сертификат Unified Communications Certificate (UCC) или Subject Alternative Names (SAN), может покрывать список доменов (до определённого предела). Он не ограничен единственным доменом — вы можете указать различные домены и поддомены. Стоимость обычно включает в себя определённое количество доменов (от трёх до пяти) с возможностью добавить больше (до определённого предела) за дополнительную плату.

Рекомендуется использовать его только с родственными сайтами, потому что клиент при проверке сертификата на любом веб-сайте увидит основной домен, а также все дополнительные.

3. Поддомены (wildcard)

Этот тип сертификата покрывает основной домен, а также неограниченное количество поддоменов (`*.example.com`) — например, `example.com`, `www.example.com`, `mail.example.com`, `ftp.example.com` и т. д. Ограничение в том, что он покрывает только поддомены основного домена.

Разнообразие различных сертификатов показано в таблице:

| Тип сертификата | Подтверждённый домен (DV) | Подтверждённая организация (OV) | Расширенное подтверждение (EV) |
|-------------------|--|--|--|
| | HTTPS | HTTPS Проверенный правообладатель | HTTPS Проверенный правообладатель Информация о владельце отображается в браузере |
| Один домен | example.com, www.example.com | | |
| Несколько доменов | example.com, www.example.com, mail.example.com, example.net, example.org и др. Определённый заранее список, до некоторого лимита (обычно 100) | | |
| Поддомены | *.example.com Подходит для любого поддомена | Недоступно — все имена должны быть явно включены в сертификат и проверены центром сертификации | |

Конфигурация

Чтобы подвести итог, четыре компонента HTTPS требуют шифрования:

- 1. Первоначальный обмен ключами**
Используются асимметричные алгоритмы (секретный и публичный ключи).
- 2. Сертификация идентичности** (сертификат HTTPS, выданный центром сертификации)
Используются асимметричные алгоритмы (секретный и публичный ключи).

3. Реальное шифрование сообщений

Используются симметричные алгоритмы (предварительно разделённый совместный секрет).

4. Дайджест сообщений

Используются алгоритмы криптографического хеширования

В каждом из этих компонентов используется набор алгоритмов (некоторые из них уже не рекомендуются для использования) с разными размерами ключей. В процессе рукопожатия клиент и сервер договариваются, какую комбинацию методов они будут использовать — выбирают один из примерно десятка различных алгоритмов публичных ключей (обмен ключами), один из примерно десятка алгоритмов симметричных ключей (шифр) и один из трёх (два не рекомендуются для использования) алгоритмов для дайджеста сообщений, что даёт нам сотни комбинаций.

Например, выбор `ECDHE-RSA-AES256-GCM-SHA384` означает, что обмен ключами будет производиться по алгоритму **Elliptic Curve Diffie-Hellman Ephemeral (ECDHE)**; центр сертификации подписал сертификат при помощи алгоритма **Rivest-Shamir-Adleman (RSA)**; симметричное шифрование сообщений будет использовать шифр **Advanced Encryption Standard (AES)** с **256-битным** ключом и будет работать в режиме **GCM**; целостность сообщений будет обеспечивать алгоритм безопасного хеширования **SHA**, с использованием **384-битных** дайджестов. (Доступен [полный список комбинаций алгоритмов](#)).

Итак, нужно сделать выбор некоторых конфигураций.

Наборы шифров

Выбор набора шифров для использования — это компромисс между совместимостью и безопасностью:

- Для совместимости со старыми браузерами нужно, чтобы сервер поддерживал старые наборы шифров.
- Однако многие старые наборы шифров больше не считаются безопасными.

OpenSSL перечисляет поддерживаемые комбинации (см. выше) в порядке убывания их криптографической стойкости. Так сделано, чтобы во время первоначального рукопожатия между клиентом и сервером они перебирали комбинации начиная с самой сильной до тех пор, пока не найдут комбинацию, которая поддерживается обеими сторонами. Есть смысл пробовать сначала самую безопасную комбинацию, а затем постепенно ослаблять безопасность, если нет других вариантов.

Википедия содержит [исчерпывающий список алгоритмов](#) для всех компонентов TLS и указывает их поддержку в разных версиях SSL и TLS.

[Mozilla SSL Configuration Generator](#) — очень **полезный и крайне рекомендуемый справочник**, какие криптографические методы использовать на сервере. Мы позже будем использовать его в реальных серверных конфигурациях.

Типы ключей

Сертификаты Elliptic Curve Cryptography (**ECC**) быстрее обрабатываются и используют меньше CPU, чем сертификаты RSA, что особенно важно для мобильных клиентов. Однако некоторые сервисы, такие как Amazon, CloudFront и Heroku на момент написания этой статьи пока не поддерживают сертификаты ECC.

256-битная длина ключа для ECC считается достаточной.

Сертификаты Rivest Shamir Adleman (**RSA**) более медленные, но совместимы с большим разнообразием старых серверов. Ключи RSA больше по размеру, так что 2048-битный ключ RSA считается минимально допустимым. Сертификаты RSA с ключами 4096 бит и больше могут ухудшать производительность — к тому же, скорее всего, они подписаны 2048-битным ключом посредника, что по большей части подрывает дополнительную защиту!

Вы могли заметить расплывчатость заявлений, сделанных выше, и отсутствие каких-либо чисел. То, что может нагрузить один сервер, не нагрузит другой сервер. Лучший способ определить влияние на производительность — это проверить загрузку на своём собственном сервере, с реальным веб-сайтом и реальными посетителями. И даже это изменится со временем.

Процедуры

Для получения сертификаты HTTPS выполните следующие шаги:

1. Создайте пару из секретного и публичного ключей и подготовьте запрос на подпись сертификата (Certificate Signing Request, CSR), включающий информацию об организации и публичном ключе.
2. Свяжитесь с центром сертификации и запросите сертификат HTTPS на основании CSR.
3. Получите подписанный сертификат HTTPS и установите его на своём сервере.

Есть набор файлов, содержащих различные компоненты инфраструктуры публичных ключей (PKI): секретный и публичный ключи, CSR и подписанный сертификат HTTPS. Чтобы ещё больше всё усложнить, разные стороны используют разные названия (и расширения) для именования одной и той же вещи.

Для начала, есть два популярных формата хранения информации — DER и PEM. Первый из них (DER) бинарный, а второй (PEM) — это файл DER в кодировке base64 (текст). По умолчанию Windows напрямую использует формат DER, а мир свободных систем (Linux и UNIX) использует формат PEM. Существуют инструменты (OpenSSL) для конвертации файлов из одного формата в другой.

В качестве примеров мы будем использовать такие файлы:

- `example.com.key`

Файл в формате PEM с секретным ключом. Расширение `.key` не является стандартом, так что кто-то может использовать его, а кто-то нет. Файл должен быть защищён и доступен только для суперпользователя.

- `example.com.pub`

Файл в формате PEM с публичным ключом. Вам на самом деле не нужен этот файл (и он никогда не будет явно присутствовать), потому что его можно сгенерировать из секретного ключа. Он включён сюда только для примера.

- `example.com.csr`

Это запрос на подпись сертификата. Файл в формате PEM содержит информацию об организации, а также публичный ключ сервера. Его нужно отправить в центр сертификации, выдающий сертификаты HTTPS.

- `example.com.crt`

Сертификат HTTPS, выданный центром сертификации. Это файл в формате PEM, который содержит публичный ключ сервера, информацию об организации, подпись центра сертификации, даты начала и окончания срока действия и др. Расширение `.crt` не является стандартом; часто используются другие расширения, в том числе `.cert` и `.cer`.

Названия файлов (и расширения) не стандартизированы; можете использовать любые. Я выбрал такие названия, потому что они кажутся говорящими и делают очевидным, какую функцию выполняет каждый компонент. Вы можете использовать любую схему именования, которая для вас имеет смысл, главное — указать соответствующие файлы ключей и сертификата в командах и конфигурации сервера в процессе настройки.

Секретный ключ — это случайно сгенерированная строка определённой длины (мы используем 2048 бит), которая выглядит примерно так:

-----BEGIN RSA PRIVATE KEY-----

MIIEowIBAAKCAQEAm+036O2PlUQbKbSSs2ik6O6TYy6+Zsas5oAk
3GioGLl1RW9N

i8kagqdnD69Et29m1v15OIPsBoW3OWb1aBW5e3J0x9prXI1W/fpv
uP9NmrHBUN4E

S17VliRpfVH3aHfPC8rKpv3GvHYOcfOmMN+HfBZlUeKJKs6c5WmS
VdnZB0R4UAWu

Q30aHEBVqtrhgHqYDBokVe0/H4wmwZEIQTINWniCOFR5UphJf5nP
8ljGbmPxNTnf

b/iHS/chjcjF7TGMG36e7EBoQijZEUQs5IBCeVefOnFLK5jLx+BC
//X+FNzByDil

Tt+128I/3ZN1ujhak73YFbWjjLR2tjtp+LQgNQIDAQABAoIBAEAO
2KVM02wTKsWb

dZlXKEi5mrtofLhkbqvTgVE7fbOKnW8FJuqCl+2NMH31F1n03176
5p4dNF4JmRhv

/+ne4vCgOPHR/cFsH4z/0d5CpHm1C7JZQ5JjR4QDOYNOpUG51smV
amPoZjkOlyih

XGk/q72CxeU6F/gKIdLt6Dx03wBosIq9IAE8LwdMnioeuj18qaVg
1950MeIOriIn

tpWP4eFya5rTpIFfIdHdIxyXsd6hF/LrRc9BMWTY1/uOLrpYjTf7
chbdNaxhwH7k

buvKxBvCvmXmd6v/AeQQAXbUkdSnbTKDaB9B7IlUTcDJyPBjXvFS
1IzzjN6vV+06

XBwHx5ECgYEAyRZLzwnA3bw8Ep9mDw8JHDQoGuQkFEMLqRdRRoZ+
hxnBD9V9M0T6

HRiUFOizEVoXxf6zPtHm/T7cRD8AFqB+pA/Nv0ug6KpwUjA4Aihf

5ADp0gem0DNw
YlVkcA6Bu7c9IUlE0hwF7RLB7YrryJVJit9AymmUTUUHCQTWW2yB
hC8CgYEAxoHS
HGXthin5owOTNPwLwPfu2o7SybkDBKyW69uTi0KxAl3610DjyA/c
V2mxIcFlPvly
HualGd9eNoeCMBY/AUtjzI0K77yeRpjj321rj6k8c8bYWPHH539S
iBXLWTY/WQ0w
pxfT3d/Z4QMh5d6p+p5f3UIrXESYQd+fAaG5tNsCgYEAksTdTB4Y
UT9EsWr6eN9G
jPlclFQUKV3OMvq77bfYvg8EJORz32nnDDmWS7SUjoOtemwutBlM
eWbaKk25aMp3
5JNMXuV6apeMJ9Dd8GU7qBUqlIvVK31/96XPvzmnYzWZPqRVwO2H
PcRFG3YcJmkg
JmZQyexJvCQ3wFNxiYUm+y0CgYBXQSMhFnCUg4jWbbDcHlnwRT+L
njHrN2arPE3O
eKLfGL6DotmqmjxFaStaRPv2MXMWgAMUsB8sQzG/WEsSaOBQaloA
xJJlFIyhzXyE
bi1UZXhMD8BzQDu1dxLI/IN4wE6SDykumVuocEfuDxlsWDZxEgJj
WD2E/iXK9seG
yRa+9wKBgEydvz+C1ECLI/dOWb20UC9nGQ+2dMa+3dsmvFwSJJat
Qv9NGaDUdxmU
hRVzWgogZ8dZ9oH8IY3U0owNRfO65VGe0sN00sQtMoweEQi0SN0J
6FePiVCnl7pf
lvYBaemLrW2YI2B7zk5fTm6ng9BW/B1KfrH9Vm5wLQBchAN8Pjbu
-----END RSA PRIVATE KEY-----

Держите ключ в секрете! Это значит, защитите его с помощью

очень ограниченных разрешений (600) и никому не разглашайте.

Его напарник — **публичный ключ** — выглядит примерно так:

```
-----BEGIN PUBLIC KEY-----  
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAm+03602P  
lUQbKbSSs2ik  
6O6TYy6+Zsas5oAk3GioGLl1RW9Ni8kagqdnD69Et29m1v15OIPs  
BoW3OWb1aBW5  
e3J0x9prXI1W/fpvuP9NmrHBUN4ES17VliRpfVH3aHfPC8rKpv3G  
vHYOcfOmMN+H  
fBZlUeKJKs6c5WmSVdnZB0R4UAWuQ30aHEBVqtrhgHqYDBokVe0/  
H4wmwZEIQTIN  
WniCOFR5UphJf5nP8ljGbmPxNTnfb/iHS/chjcjF7TGMG36e7EBo  
QijZEUQs5IBC  
eVefOnFLK5jLx+BC//X+FNzByDilTt+l28I/3ZN1ujhak73YFbWj  
jLR2tjtp+LQg  
NQIDAQAB  
-----END PUBLIC KEY-----
```

Запрос на получение сертификата выглядит примерно так:

```
-----BEGIN CERTIFICATE REQUEST-----  
MIICzjCCABYCAQAwgYgxFDASBgNVBAMMC2V4YW1wbGUuY29tMQsw  
CQYDVQQQLDAJJ  
VDEPMA0GA1UECAwGTG9uZG9uMRIwEAYDVQQKDAlBQ01FIEluYy4x  
IDAeBgkqhkiG
```


9w0BCQEWEFkbWluQGV4YW1wbGUuY29tMQswCQYDVQQGEwJHQQjEP
MA0GA1UEBwwG
TG9uZG9uMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
m+036O2PlUQb
KbSSs2ik6O6TYy6+Zsas5oAk3GioGLl1RW9Ni8kagqdnD69Et29m
1v15OIPsBoW3
OWb1aBW5e3J0x9prXI1W/fpvuP9NmrHBUN4ES17VliRpfVH3aHfP
C8rKpv3GvHYO
cfOmMN+HfBZlUeKJKs6c5WmSVdnZB0R4UAWuQ30aHEBVqtrhgHqY
DBokVe0/H4wm
wZEIQTINWniCOFR5UphJf5nP8ljGbmPxNTnfb/iHS/chjcjF7TGM
G36e7EBoQijZ
EUQs5IBCeVefOnFLK5jLx+BC//X+FNzByDilTt+128I/3ZN1ujha
k73YFbWjjLR2
tjtp+LQgNQIDAQABoAAwDQYJKoZIhvcNAQELBQADggEBAGIQVhXf
uWdINNfceNPm
CkAGv4yzpx88L34bhO1Dw4PYWnoS2f7ItuQA5zNk9EJhjkW8gYs
pK7mPkvHDbFa
Um7lPSWsm3gjd3pU7dIaHxQ+0AW9lOw5ukiBl04t3qgt+jTVZ3Eh
MbR0jDSyjTrY
kTgfuqQrGOQSmLb5XviEtCcN0rseWib3fKI18DM69JiA2AALxyk7
DCkS1BqLNChT
pnbgvtlUhc4yFXNctwPGskXIvLsCn2LRy+qdsPM776kDLgD36hK0
Wu14Lpsoa/p+
ZRuwKqTjdaV23o2aUMULyCRuITlghEEkRdJsaXadHXtNd5I5vDJO
AAAt46PIXcyEZ
aQY=

-----END CERTIFICATE REQUEST-----

Этот конкретный CSR содержит публичный ключ сервера и информацию о компании ACME Inc., которая находится в Лондоне, Великобритания, и владеет доменом `example.com`.

Наконец, подписанный **сертификат HTTPS** выглядит примерно так:

-----BEGIN CERTIFICATE-----

MIIDjjCCAnYCCQCJdR6v1+W5RzANBgkqhkiG9w0BAQUFADCBiDEU
MBIGA1UEAwgL

ZXhhbXBsZS5jb20xCzAJBgNVBAsMAklUMQ8wDQYDVQQIDAZMb25k
b24xEjAQBGNV

BAoMCUFDTUUgSW5jLjEgMB4GCSqGSIb3DQEJARYRYWRtaW5AZXhh
bXBsZS5jb20x

CzAJBgNVBAYTAkdCMQ8wDQYDVQQHDAZMb25kb24wHhcNMTYwNDE5
MTAzMjI1WhcN

MTcwNDE5MTAzMjI1WjCBiDEU
MBIGA1UEAwgLZXhhbXBsZS5jb20x
CzAJBgNVBAsM

AklUMQ8wDQYDVQQIDAZMb25kb24xEjAQBGNVB
AoMCUFDTUUgSW5jLjEgMB4GCSqG

SIb3DQEJARYRYWRtaW5AZXhhbXBsZS5jb20xCzAJBgNVBAYTAkdC
MQ8wDQYDVQQH

DAZMb25kb24wggeiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIB
AQCb7Tfo7Y+V

RBsptJKzaKTo7pNjLr5mxqzmgCTcaKgYuXVFb02LyRqCp2cPr0S3

b2bW+Xk4g+wG
hbc5ZvVoFbl7cnTH2mtcjVb9+m+4/02ascFQ3gRLXtWWJG19Ufdo
d88Lysqm/ca8
dg5x86Yw34d8FmVR4okqzpz1aZJV2dkHRHhQBa5DfRocQFWq2uGA
epgMGiRV7T8f
jCbBkQhBMglaeII4VHlSmEl/mc/yWMZuY/E1Od9v+IdL9yGNyMXt
MYwbfp7sQGhC
KNkRRCzkgEJ5V586cUsrmMvH4EL/9f4U3MHIOKVO36Xbwj/dk3W6
OFqTvdgVtaOM
tHa2O2n4tCA1AgMBAAEwDQYJKoZIhvcNAQEFBQADggEBABwwkE7w
X5gmZMRYugSS
7peSx83Oac1ikLnUDMMOU8WmqxaLTTZQeuoq5W23xWQWgcTtfjP9
vfV50jFzXwat
5Ch3OQUS53d06hX5EiVrmTyDgybPVlfbq5147MBEC0ePGxG6uV+E
d+oUYX4OM/bB
XiFa4z7eamG+Md2d/A1cB54R3LH6vECLuyJrF0+sCGJJAGumJGhj
cOdpvUVt5gvD
FIgT9B04VJnaBatEgWbn9x50EP4j41PNFGx/A0CCLgbTs8kZCdhe
4QFMxU9T+T9t
rXgaspIi7RA4xkSE7x7B8NbvSlgP79/qUe80Z7d8Oolva6dTZduB
yr0Cejd fhLhi
mNU=
-----END CERTIFICATE-----

Все части связаны и должны соответствовать друг другу.
Последний сертификат был сгенерирован только ради примера —
это так называемый самоподписанный сертификат, потому что он

не подписан признанным центром сертификации.

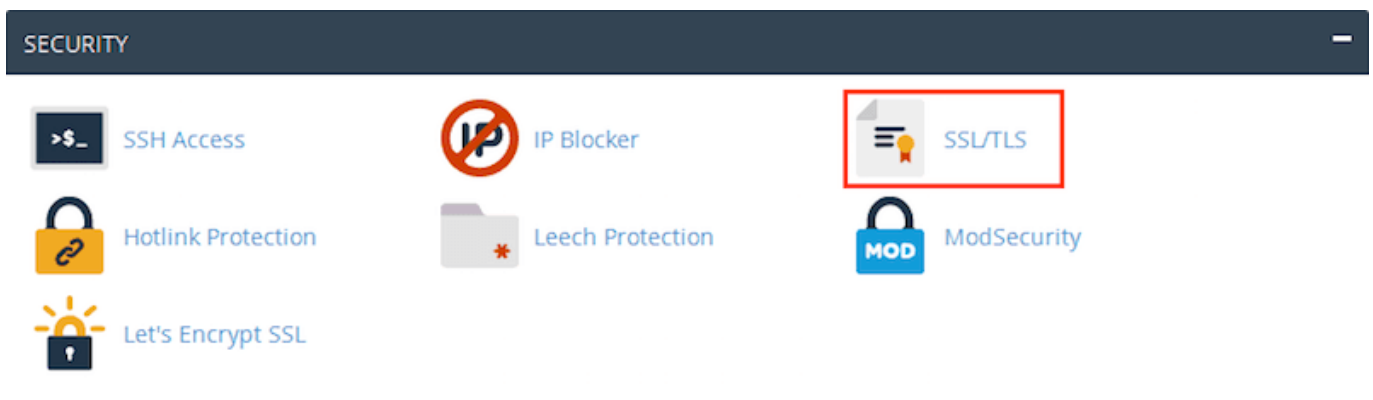
Мы проиллюстрируем этот процесс реальными шагами, которые нужно сделать в cPanel, Linux, FreeBSD и Windows. Это универсальный процесс, подходящий для всех типов сертификатов. Если вы хотите получить бесплатный сертификат DV, то следуйте другим процедурам, описанным в разделах [Let's Encrypt](#) и [Cloudflare](#).

Шаг 1. Создать секретный ключ и запрос на получение сертификата


В следующих примерах мы будем использовать 2048-битные сертификаты RSA, по причине их большей совместимости. Если ваш провайдер, у которого установлен сервер, поддерживает ECC (например, если вы не пользуетесь услугами Heroku или AWS), то можете предпочесть использовать ECC.

cPanel

1. Войдите в cPanel своего хоста.
2. Прокрутите вниз до раздела “Security” и нажмите “SSL/TLS”.



3. Теперь вы в разделе “SSL/TLS Manager”. Нажмите “Private Keys (KEY)” для создания нового секретного ключа.

 **SSL/TLS**

The SSL/TLS Manager will allow you to generate SSL certificates, certificate signing requests, and private keys. These are all parts of using SSL to secure your website. SSL allows you to secure pages on your site so that information such as logins, credit card numbers, etc are sent encrypted instead of plain text. It is important to secure your site's login areas, shopping areas, and other pages where sensitive information could be sent over the web.

Private Keys (KEY)
[Generate, view, upload, or delete your private keys.](#)


Certificate Signing Requests (CSR)
[Generate, view, or delete SSL certificate signing requests.](#)

Certificates (CRT)
[Generate, view, upload, or delete SSL certificates.](#)

Install and Manage SSL for your site (HTTPS)
[Manage SSL sites.](#)

“SSL/TLS Manager” в cPanel ([см. большую версию](#))

4. Вас перенаправят на страницу “Generate, Paste or Upload” для нового “Private Key”. Там выберете 2048 бит в выпадающем меню и нажмите “Generate”.

 **SSL/TLS**

Private Keys

A private key is used to decrypt information transmitted over SSL. When you create an SSL certificate, the first step is to generate a private key file associated with that SSL certificate. You should generate a private key for each SSL certificate you create. This private key is very important and should be kept confidential. A copy of each private key should be kept in a safe place; there is no way to recover a lost private key.

Generate a New Private Key.

You should generate a new key file for each certificate you install. A key size of 2,048 bits is recommended.

Key Size

Description:

Optional: You can use this field to provide a description for this private key.

Generate

Upload a New Private Key.

If you have an existing key, paste the key below, or upload it to the server.

Paste the key into the following text box:

Description:

Optional: You can use this field to provide a description for this private key.

Save

or

Choose a .key file.:

No file selected.

Description:


Optional: You can use this field to provide a description for this private key.

Upload

Управление секретным ключом (“Private Key”) в cPanel ([см. большую версию](#))

5. Будет сгенерирован новый секретный ключ, а вы получите подтверждение на экране:

Generate a Private Key

 The server has generated the private key as requested. To use this private key on another server, copy and paste the information from the encoded field below.

Description:
2,048 bits, created 4/19/16 9:46 AM UTC

Encoded Private Key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIIEowIBAAKCAQEAm+03602PlUQbKbSSs2ik606TYy6+Zsas5oAk3Gio
i8kagqdnD69Et29m1vl5OIPsBow30Wb1aBW5e3J0x9prXI1W/fpvuP9N
S17VliRpfVH3aHfPC8rKpv3GvHY0cfOmMN+HfBZ1UeKJKs6c5WmSVdnZ
Q30aHEBVqtrhgHqYDBokVe0/H4wmwZEIQTINWniCOFR5UpHJf5nP81jG
b/iHS/chjcjF7TGMG36e7EBoQijZEuQs5IBCeVefOnFLK5jLx+BC//X+
Tt+l28I/3ZN1ujhak73YFbwjjLR2tjtp+LQgNQIDAQABAoIBAEAO2KVM
dZlXKEi5mrtofLhkbqvTgVE7fb0KnW8FJuqCl+2NMH31F1n031765p4d
/+ne4vCgOPHR/cFsH4z/0d5CpHMLC7JZQ5JjR4QDOYN0pUG51smVamPo
XGk/q72CxeU6F/gKIdLt6Dx03wBosIq9IAE8LwdMnioeuji8qaVg1950
tpWp4eFya5rTpIFfIdHdIxyXsd6hF/LrRc9BMWY1/uOLrpYjTf7chbd
buvKxBvCvmXmd6v/AeQQAXbUkdSnbTKDaB9B7I1UTcDJyPBjXvFS1Izz
XBwHx5ECgYEAyRZLzwnA3bw8Ep9mDw8JHDQoGuQkFEMlQrRRoZ+hxNB
HRiUfOizEVoXxf6zPtHm/T7cRD8AFqB+pA/Nv0ug6KpwUjA4Aihf5ADp
YlVkcA6Bu7c9IUlE0hwF7RLB7YrryJVJit9AymmUTUuHCQTWw2yBhC8C
HGxthin5owOTNPwLwPfu2o7SybkD8KyW69uTi0KxAl3610DjyA/cV2mx
HualGd9eNoeCMBy/AutjzI0K77yeRpjj321rj6k8c8bYwPHH539SiBXL
pxfT3d/Z4QmH5d6p+p5f3UIrXESYQd+fAaG5tNsCgYEAksTdTb4YUT9E
jPlclFQUKV30Mvq77bfYvg8EJORz32nnDDmWS7SUjoOtemwutB1MeWba
5JNMXuV6apeMj9Dd8GU7qBUqlIvVK31/96XPvzmnYzWZPqRVwO2HPcRF
JmZQyexJvCQ3wFNxiYUm+y0CgYBYBXSqSMhFnCUg4jWbbDcHlnwRT+LnjHr
eKLfGL6DotmqmjxFaStaRPv2MXWgAMUsB8sQzG/wEsSaOBQa1oAxJJl
bi1UZXhMD8BzQDu1dxLI/IN4wE6SDykumVuocEfuDx1sWDZxEgJjWd2E
yRa+9wKBgEydVz+C1ECLI/d0Wb20UC9nGQ+2dMa+3dsmvFwSJJatQv9N
hRVzWgogZ8dZ9oH8IY3U0owNRfO65VGe0sN00sQtMoweEQi0SN0J6FeP
lvYBaemLrW2YI2B7zk5fTm6ng9BW/B1KfrH9Vm5wLQBchAN8Pjbu
-----END RSA PRIVATE KEY-----
```

Подтверждение секретного ключа в cPanel ([см. большую версию](#))

6. Если вернётесь назад в раздел “Private Keys”, то увидите там новый ключ:

Private Keys

A private key is used to decrypt information transmitted over SSL. When you create an SSL certificate, the first step is to generate a private key file associated with that SSL certificate. You should generate a private key for each SSL certificate you create. This private key is very important and should be kept confidential. A copy of each private key should be kept in a safe place; there is no way to recover a lost private key.

Keys on Server

| Description | ID | Size | Actions |
|---|--|------|--|
| 2,048 bits, created 4/19/16 9:46 AM UTC | 9bed3_42035_ f664455068d464437df827800e11083d | 2048 | Edit Delete |

Generate a New Private Key.

You should generate a new key file for each certificate you install. A key size of 2,048 bits is recommended.

Key Size

Description:

Optional: You can use this field to provide a description for this private key.

Раздел “Private Keys” в cPanel с новым сгенерированным ключом ([см. большую версию](#))

- Вернитесь в раздел “SSL/TLS Manager”. Нажмите “Certificate Signing Requests (CSR)” для создания нового запроса на получение сертификата.

SSL/TLS

The SSL/TLS Manager will allow you to generate SSL certificates, certificate signing requests, and private keys. These are all parts of using SSL to secure your website. SSL allows you to secure pages on your site so that information such as logins, credit card numbers, etc are sent encrypted instead of plain text. It is important to secure your site's login areas, shopping areas, and other pages where sensitive information could be sent over the web.

Private Keys (KEY)

[Generate, view, upload, or delete your private keys.](#)

Certificate Signing Requests (CSR)

[Generate, view, or delete SSL certificate signing requests.](#)

Certificates (CRT)

[Generate, view, upload, or delete SSL certificates.](#)

Install and Manage SSL for your site (HTTPS)

[Manage SSL sites.](#)

Раздел “SSL/TLS Manager” в cPanel ([см. большую версию](#))

8. Теперь вы увидите форму “Generate Service Request”. Выберите созданный ранее секретный ключ и заполните поля. Правильно ответьте на все вопросы (они будут открыты для просмотра в вашем подписанном сертификате!), особенное внимание уделите разделу “Domains”, который должен в точности совпадать с доменным именем, для которого вы запрашиваете сертификат HTTPS. Включите туда только домен верхнего уровня (`example.com`); центр сертификации обычно сам добавляет поддомен `www` (то есть `www.example.com`). По окончании нажмите кнопку “Generate”.

SSL/TLS

SSL Certificate Signing Request

If you obtain a certificate from a trusted SSL provider, you must complete the Certificate Signing Request form to provide the information needed to generate your SSL certificate.

Certificate Signing Requests on Server

| Domains | Created (UTC) | Description | Actions |
|--|---------------|-------------|---------|
| There are no certificate signing requests on the server. | | | |

Generate a New Certificate Signing Request (CSR)

Use this form to generate a new certificate signing request for your domain. Your SSL certificate authority (CA) will ask for a certificate signing request to complete the certificate purchase. Your CA may require specific information in the form below. Check with the CA's CSR requirements for the Apache web server.

Key*

2,048 bits, created 4/19/16 9:46 AM UTC

Edit

Domains *

example.com

You do not control this domain.

Provide the [FQDNs](#) that you are trying to secure, one per line. You may use a wildcard domain by adding an asterisk in a domain name in the form: *.sample.com. NOTE: Many [CAs](#) charge a higher price to issue multiple-domain certificates (sometimes called "[UCCs](#)" or "[SAN](#) certificates") and certificates that include wildcard domains.

City*

London

Provide the complete name for the city or locality. Do not use abbreviations.

State*

London

Provide the complete name for the state or province. Do not use abbreviations.

Country*

GB (United Kingdom)

Choose the country of origin for the certificate's "Company".

Company*

ACME Inc.

Provide the legally-registered name for your business. If your company name includes symbols other than a period or comma, check with your certificate authority to confirm that they are acceptable.

Company Division

IT

Provide the name of the division or group within the above company. If the division includes symbols other than a period or comma, check with your certificate authority to confirm that they are acceptable.

Email

admin@example.com

Provide a valid email address where you can be contacted for verification of domain ownership.

Passphrase

Some certificate authorities may require CSRs to have a passphrase. The certificate authority can use a CSR passphrase to confirm the identity of the person or organization with whom you wish to communicate. CSR passphrases are stored **unencrypted** in the CSR. Because of this, and also because you will share this passphrase with a third party, do not use an important password here.

Description

Generate

Форма “Create New Certificate Signing Request” в cPanel ([см. большую версию](#))

9. Будет сгенерирован новый CSR, а вы увидите окно с подтверждением:

SSL/TLS

Generated Certificate Signing Request



The Certificate Signing Request for “example.com” has been generated and saved in your user directory. To purchase a trusted certificate, you must copy the Encoded Certificate Signing Request below and send it to the Certificate Authority. Follow the instructions provided by your Certificate Authority.

Domain:

example.com

Description:

example.com

Encoded Certificate Signing Request:

```
-----BEGIN CERTIFICATE REQUEST-----
MIICZjCCAbYCAQAwYgxFDASBgNVBAMMC2V4YW1wbGUuY29tMQswCQYD
VDEPMA0GA1UECAwGTG9uZG9uMRIwEAYDVQQKDA1BQ01FIEluYy4xIDAe
9w0BCQEWEFkbWluQG9uZG9uY29tMQswCQYDVQGEwJHQPjEPMA0G
TG9uZG9uMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAm+03
KbSSs2ik606TYy6+Zsas5oAk3GioGLl1RW9Ni8kagqdnD69Et29m1v15
OWb1a8W5e3J0x9prXI1W/fpvuP9NmrHBUN4ES17V1iRpfVH3aHfPC8rK
cf0mMnLHFR711uK7Kc6r5WmSVdn7R0R411AWu030aHFRVgtcrhoHqVDRok
```

```
-----BEGIN CERTIFICATE REQUEST-----
MIIEGjCCAgIBADCCAQYJKoZIhvcNAQELBQADggEBAGIQVhXfuWdI
CkAGv4yzpx88L34bh01Dw4PYWnoS2f7ItuQA5zNk9EjhKwK8gYspK7m
Um7lPSwsm3gjd3pU7dIaHxQ+0AW9lOwSukiB104t3qgt+jTVZ3EhMbr0
kTgfuqRGOQSmLb5XviEtCcN0rseWib3fKI18DM69JiA2AALxyk7DCKS
pnbgtv1Uhc4yFXNCtwPGskXivLsCn2LRy+qdsPM776kDLgD36hK0Wu14
ZRuwKqTjdaV23o2aUMULyCRuITlghEEkRdJsaXadHXTNd5I5vDJ0AAt4
aQY=
-----END CERTIFICATE REQUEST-----
```

Decoded Certificate Signing Request:

Certificate Request:

Data:

Version: 0 (0x0)
Subject: CN = example.com, OU = IT, ST = London,
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:

```
00:9b:ed:37:e8:ed:8f:95:44:1b:29:b4:
a4:e8:ee:93:63:2e:be:66:c6:ac:e6:80:
a8:18:b9:75:45:6f:4d:8b:c9:1a:82:a7:
44:b7:6f:66:d6:f9:79:38:83:ec:06:85:
f5:68:15:b9:7b:72:74:c7:da:6b:5c:8d:
6f:b8:ff:4d:9a:b1:c1:50:de:04:4b:5e:
69:7d:51:f7:68:77:cf:0b:ca:ca:a6:fd:
0e:71:f3:a6:30:df:87:7c:16:65:51:e2:
9c:e5:69:92:55:d9:d9:07:44:78:50:05:
1a:1c:40:55:aa:da:e1:80:7a:98:0c:1a:
3f:1f:8c:26:c1:91:08:41:32:0d:5a:78:
79:52:98:49:7f:99:cf:f2:58:c6:6e:63:
df:6f:f8:87:4b:f7:21:8d:c8:c5:ed:31:
9e:ec:40:68:42:28:d9:11:44:2c:e4:80:
9f:3a:71:4b:2b:98:cb:c7:e0:42:ff:f5:
c1:c8:38:a5:4e:df:a5:db:c2:3f:dd:93:
5a:93:bd:d8:15:b5:a3:8c:b4:76:b6:3b:
20:35
```

Exponent: 65537 (0x10001)

Attributes:


a0:00

Signature Algorithm: sha256WithRSAEncryption

```
62:10:56:15:df:b9:67:48:34:d7:dc:78:d3:e6:0a:40
8c:b3:a7:1f:3c:2f:7e:1b:84:ed:43:c3:83:d8:5a:7a
fe:c8:b6:e4:00:e7:33:64:f4:42:61:8e:4c:0a:f2:06
ae:e6:3e:4b:c7:0d:b1:5a:52:6e:e5:3d:25:ac:9b:78
7a:54:ed:d2:1a:1f:14:3e:d0:05:bd:94:ec:39:ba:48
ee:2d:de:a8:2d:fa:34:d5:67:71:21:31:b4:74:8c:34
3a:d8:91:38:1f:ba:a4:2b:18:e4:12:98:b6:f9:5e:f8
27:0d:d2:bb:1e:5a:26:f7:7c:a2:25:f0:33:3a:f4:98
00:0b:c7:29:3b:0c:29:12:d4:1a:8b:34:28:53:a6:76
d9:54:85:ce:32:15:73:42:b7:03:c6:b2:45:c8:bc:bb
62:d1:cb:ea:9d:b0:f3:3b:ef:a9:03:2e:00:f7:ea:12
ed:78:2e:9b:28:6b:fa:7e:65:1b:b0:2a:a4:e3:75:a5
8d:9a:50:c5:0b:c8:24:6e:21:39:60:84:41:24:45:d2
76:9d:1d:7b:4d:77:92:39:bc:32:4e:00:0b:78:e8:f2
21:19:69:06
```

Подтверждение создания CSR в cPanel ([см. большую версию](#))

10. Если вы вернётесь назад в раздел “Certificate Signing Request”, то увидите там новый CSR:

 SSL/TLS

SSL Certificate Signing Request

If you obtain a certificate from a trusted SSL provider, you must complete the Certificate Signing Request form to provide the information needed to generate your SSL certificate.

Certificate Signing Requests on Server

| Domains | Created (UTC) | Description | Actions |
|-------------|---------------|-------------|---|
| example.com | 4/19/16 | example.com | Edit Delete |

Раздел “Certificate Signing Request” в cPanel с новым сгенерированным CSR ([см. большую версию](#))

Linux, FreeBSD

Убедитесь, что установлен OpenSSL. Вы можете проверить это:

```
openssl version
```

Если нет, то откройте консоль и установите его для своей платформы:

- **Debian, Ubuntu** и клоны

```
sudo apt-get install openssl
```

- **Red Hat, CentOS** и клоны

```
sudo yum install openssl
```

- **FreeBSD**

```
make -C /usr/ports/security/openssl install clean
```

Затем сгенерируйте секретный ключ и CSR одной командой:

```
openssl req -newkey rsa:2048 -nodes -keyout  
example.com.key -out example.com.csr
```

Секретный ключ будет сгенерирован, а вам нужно ответить на несколько вопросов для CSR:

```
Generating a 2048 bit RSA private key  
.....+++  
.....  
.....+++  
writing new private key to 'example.com.key'  
-----  
  
You are about to be asked to enter information that  
will be incorporated  
into your certificate request.  
  
What you are about to enter is what is called a  
Distinguished Name or a DN.  
  
There are quite a few fields but you can leave some  
blank  
  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.
```

Правильно ответьте на все вопросы (они будут открыты для

просмотра в вашем подписанном сертификате!), особое внимание уделите **разделу “Common Name”** (например, сервер **FQDN** или **ВАШЕ имя**), который должен в точности совпадать с доменным именем, для которого вы запрашиваете сертификат HTTPS. Включите туда только домен верхнего уровня (example.com); центр сертификации обычно сам добавляет поддомен www (то есть www.example.com):

Country Name (2 letter code) [AU]:GB

State or Province Name (full name) [Some-State]:London

Locality Name (eg, city) []:London

Organization Name (eg, company) [Internet Widgits Pty Ltd]:ACME Inc.

Organizational Unit Name (eg, section) []:IT

Common Name (e.g. server FQDN or YOUR name) []:example.com

Email Address []:admin@example.com

Please enter the following 'extra' attributes to be sent with your certificate request

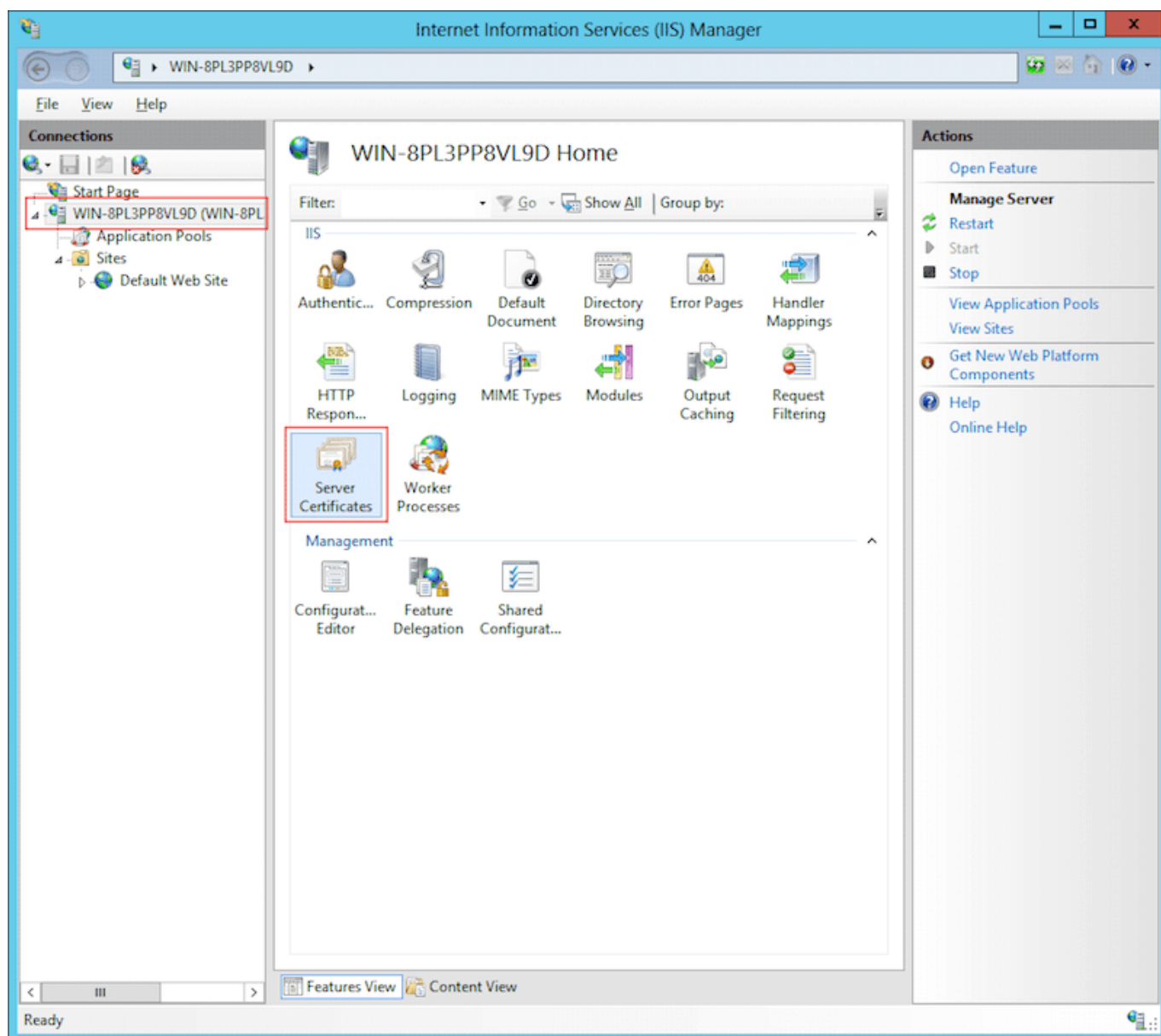
A challenge password []:

An optional company name []:

Internet Information Server под Windows

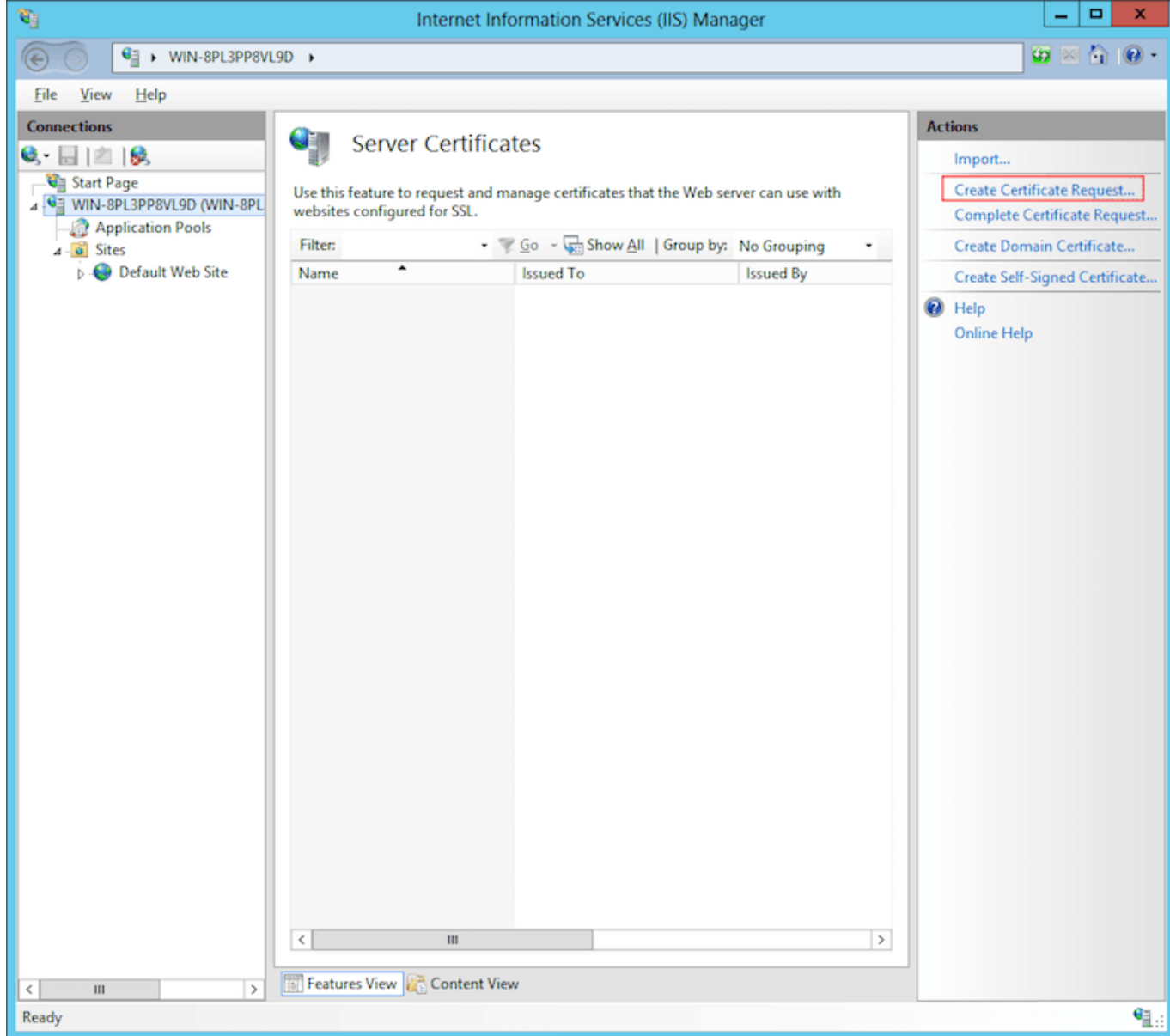
1. Откройте “Start” → “Administrative Tools” → “Internet Information Services (IIS) Manager”. Нажмите на имя сервера. Двойным

щелчком откройте “Server Certificates” в средней колонке:



Откройте “Internet Information Services (IIS) Manager”. Двойным щелчком откройте “Server Certificates”. (см. большую версию)

2. Нажмите “Create Certificate Request” в правой колонке.




Нажмите “Create Certificate Request” в правой колонке. (см. [большую версию](#))

3. Введите информацию о своей организации, особое внимание уделите графе “Common Name”, значение в которой должно соответствовать вашему доменному имени. Нажмите “Next”.

? X

Request Certificate



Distinguished Name Properties


Specify the required information for the certificate. State/province and City/locality must be specified as official names and they cannot contain abbreviations.

| | |
|----------------------|--|
| Common name: | <input type="text" value="example.com"/> |
| Organization: | <input type="text" value="ACME Inc."/> |
| Organizational unit: | <input type="text" value="IT"/> |
| City/locality | <input type="text" value="London"/> |
| State/province: | <input type="text" value="London"/> |
| Country/region: | <input type="text" value="GB"/> |

Введите информацию о своей организации. (см. большую версию)

- Оставьте значение по умолчанию в поле “Cryptographic Service Provider.” Установите “Bit length” на 2048. Нажмите “Next”.

Request Certificate

 **Cryptographic Service Provider Properties**

Select a cryptographic service provider and a bit length. The bit length of the encryption key determines the certificate's encryption strength. The greater the bit length, the stronger the security. However, a greater bit length may decrease performance.

Cryptographic service provider:

Microsoft RSA SChannel Cryptographic Provider

Bit length:


2048

Previous Next Finish Cancel

Установите “Bit length” на 2048. (см. [большую версию](#))

5. Выберите место для сохранения сгенерированного CSR и нажмите “Finish”.

Request Certificate

**File Name**

Specify the file name for the certificate request. This information can be sent to a certification authority for signing.

Specify a file name for the certificate request:

...

Previous

Next

Finish

Cancel

Выберите место для сохранения сгенерированного CSR и нажмите “Finish”. (см. [большую версию](#))

Шаг 2. Приобретение сертификата HTTPS

Чтобы получить сертификат для вашего веб-сайта, сначала купите кредит на сертификат HTTPS выбранного типа (DV, OV, EV, один сайт, несколько сайтов, поддомены — см. выше) у продавца сертификатов. По окончании процесса вам нужно будет отправить запрос на получение сертификата, который потратит купленный кредит для выбранного домена. Вас попросят предоставить (то есть вставить в поле для загрузки) весь текст CSR, включая строки
-----BEGIN CERTIFICATE REQUEST----- и -----END

CERTIFICATE REQUEST-----. Если вам нужен сертификат EV или OV, то нужно будет указать юридическое лицо, для которого вы запрашиваете сертификат. У вас также могут попросить дополнительные документы, подтверждающие тот факт, что вы представляете эту компанию. Затем регистратор сертификатов проверит ваш запрос (и все сопутствующие документы) и выдаст подписанный сертификат HTTPS.

Получение сертификата HTTPS

У вашего хостинг-провайдера или регистратора HTTPS может быть другая процедура регистрации, но общая логика одинакова.

1. Найти продавца сертификатов HTTPS.
2. Выбрать тип сертификата (DV, OV, EV, один сайт, несколько сайтов, поддомены) и добавить его в корзину. Выбрать предпочитаемый метод оплаты и совершить платёж.
3. Активировать новый сертификат HTTPS для своего домена. Вы можете или вставить в форму, или загрузить файл с запросом на подпись сертификата. Система извлечёт информацию для сертификата из CSR.
4. Вас попросят выбрать метод «утверждения контроля домена» (“Domain Control Validation”, DCV) — либо по **электронной почте**, либо загрузкой файла HTML (**на основе HTML**), либо путём добавления записи TXT к своему файлу доменной зоны (**на основе DNS**). Следуйте инструкциям по указанному методу DCV для утверждения.
5. Подождите несколько минут, пока осуществляется утверждение и готовится сертификат HTTPS. Скачайте сертификат.

Самоподписанные сертификаты

Существует возможность самому подписать сертификат, а не отдавать это право центру сертификации. Это хорошо для целей тестирования, потому что с криптографической точки зрения самоподписанный сертификат не отличается от любого другого, но браузеры не станут ему доверять, а начнут показывать предупреждение безопасности — вы можете выдавать себя за кого угодно, но это не проверено доверенной третьей стороной. Если пользователь доверяет веб-сайту, он может добавить исключение в свой браузер, который сохранит сертификат и будет доверять сайту при будущих визитах.

Например, выше опубликован самоподписанный сертификат — вы можете использовать его для домена `example.com`, и он будет работать в течение своего срока действия.

Можно создать самоподписанный сертификат на любой платформе, где есть OpenSSL.

```
openssl x509 -signkey example.com.key -in  
example.com.csr -req -days 365 -out example.com.crt
```

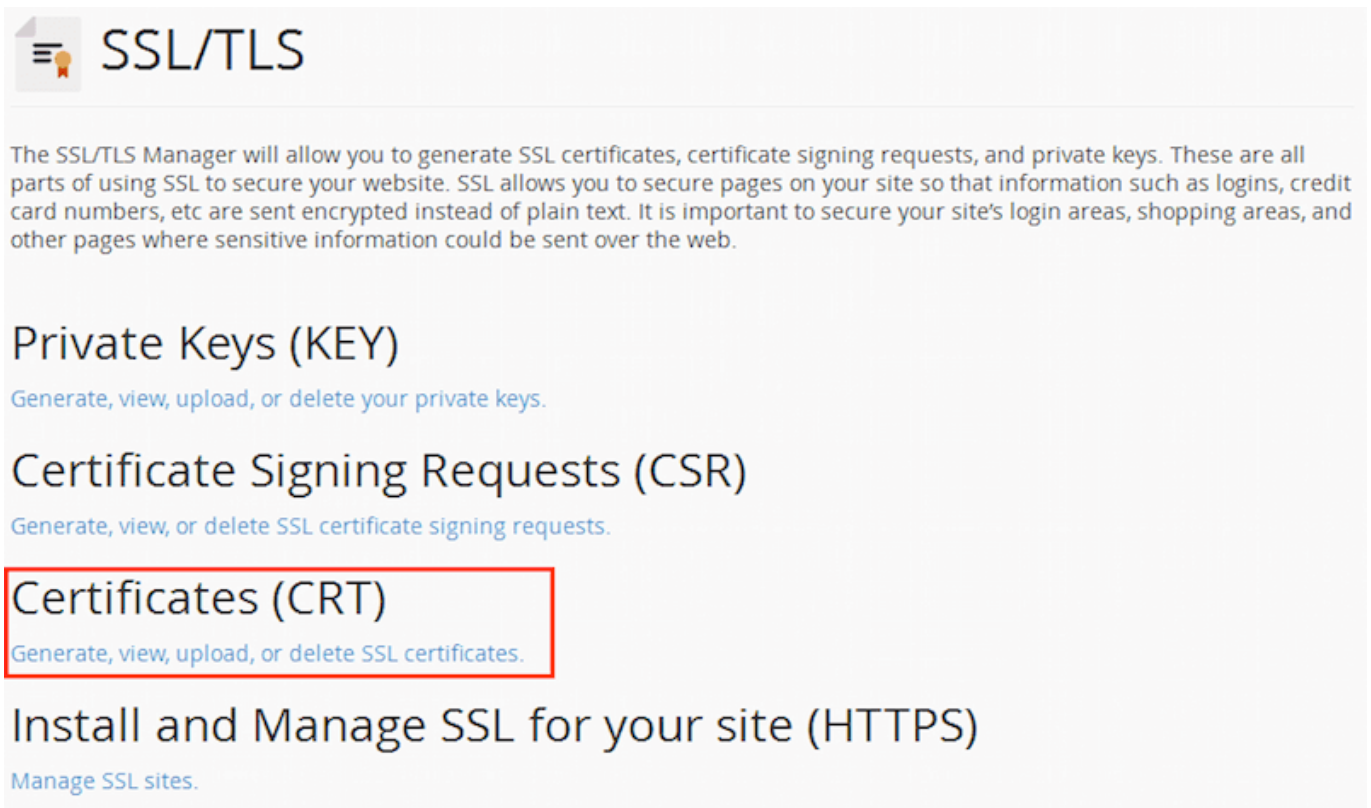
Как только сертификат сгенерирован, вам нужно установить его на свой сервер. Если у вас хостинг и сервис регистрации HTTPS от одного провайдера (многие хостинг-провайдеры также продают сертификаты HTTPS), то может действовать автоматическая процедура установки и активации вашего нового сертификата

HTTPS для веб-сайта. Если вы хоститесь где-то в другом месте, то нужно скачать сертификат и сконфигурировать сервер для его использования.

Шаг 3. Установка сертификата HTTPS для своего веб-сайта

cPanel

1. Вернитесь в “SSL/TLS Manager”. Нажмите “Certificates (CRT)” для импортирования нового сертификата.



Раздел “SSL/TLS Manager” в cPanel (см. [большую версию](#))

2. Вас перенаправят на страницу “Paste, Upload or Generate” для нового “Certificate”. Вставьте содержимое файла сертификата, полученного от регистратора HTTPS, или загрузите его с помощью кнопки “Browse”.



SSL Certificates

You can use a self-signed certificate or a trusted certificate from an SSL Certificate Authority. If you plan to use a self-signed certificate for one of your sites, you can generate it below. To use a trusted certificate, upload or provide the certificate below, after you have received the SSL certificate from your trusted provider.

Certificates on Server

| Domains | Issuer | Expiration (UTC) | Key Size | Description | Actions |
|--|--------|------------------|----------|-------------|---------|
| There are no certificates on the server. | | | | | |

Upload a New Certificate

Use this form to upload a certificate provided by a third-party Certificate Authority. You may either paste the body of the certificate or upload it from a ".crt" file.

Paste the certificate into the following text box:

Description

Save Certificate

or

Choose a certificate file (*.crt).

Browse...

No file selected.

Description

Upload Certificate

Generate a New Certificate

Use this form to generate a new, self-signed certificate for your domain. Typically, self-signed certificates are temporarily used until you receive a trusted SSL certificate from your SSL certificate authority.

Key***Domains***

Provide the [FQDNs](#) that you are trying to secure, one per line. You may also use wildcard domains by adding an asterisk in a domain name in the form: **.sample.com*. (Certificates with multiple domains are sometimes called "[UCCs](#)" or "[SAN](#) certificates".)

City*

Provide the complete name for the city or locality. Do not use abbreviations.

State*

Provide the complete name for the state or province. Do not use abbreviations.

Country*

Choose the country of origin for the certificate's "Company".

Company*

Provide the legally-registered name for your business. If your company name includes symbols other than a period or comma, check with your certificate authority to confirm that they are acceptable.

Company Division

Provide the name of the division or group within the above company. If the division includes symbols other than a period or comma, check with your certificate authority to confirm that they are acceptable.

Email

Provide a valid email address where you can be contacted for verification of domain ownership.

Description

Generate

Импорт нового сертификата HTTPS в cPanel ([см. большую версию](#))

3. После того как вы вставили содержимое нового сертификата HTTPS, оно будет проанализировано, а чисто текстовые значения покажут вам для подтверждения. Просмотрите содержимое и нажмите кнопку “Save Certificate”.

SSL Certificates

You can use a self-signed certificate or a trusted certificate from an SSL Certificate Authority. If you plan to use a self-signed certificate for one of your sites, you can generate it below. To use a trusted certificate, upload or provide the certificate below, after you have received the SSL certificate from your trusted provider.

Certificates on Server

| Domains | Issuer | Expiration (UTC) | Key Size | Description | Actions |
|--|--------|------------------|----------|-------------|---------|
| There are no certificates on the server. | | | | | |

Upload a New Certificate

Use this form to upload a certificate provided by a third-party Certificate Authority. You may either paste the body of the certificate or upload it from a ".crt" file.

Paste the certificate into the following text box:

```
tHa2O2n4tCA1AgMBAAEwDQYJKoZIhvcNAQEFBQADggEBA
BwwkE7wX5gmZMRyugSS
7peSx83Oac1ikLnUDMMOU8WmqxaLTTZQeuoq5W23
xWQWgcTtfjP9vfV50jFzXwat
5Ch3OQU553d06hX5EiVrmTyDgybPVIfbq5147MBEC0e
PGxG6uV+Ed+oUYX4OM/bB
XiFa4z7eamG+Md2d
/A1cB54R3LH6vECLuyjrF0+sCGJJAGumJGhjcOdpvUVt5g
vD
FlgT9B04VJnaBatEgWbn9x50EP4j41PNFGx
/A0CCLgbTs8kZCdhE4QFMxU9T+T9t
rXgaspli7RA4xkSE7x7B8NbvSlgP79
/qUe80Z7d8Oolva6dTZduByr0CejdfhLhi
mNU=
-----END CERTIFICATE-----
```

Domains: example.com
(self-signed) ⚠

Issuer:

Key Size: 2,048 bits (9bed37e8 ...)

Expiration: Apr 19, 2017 10:32:26 AM

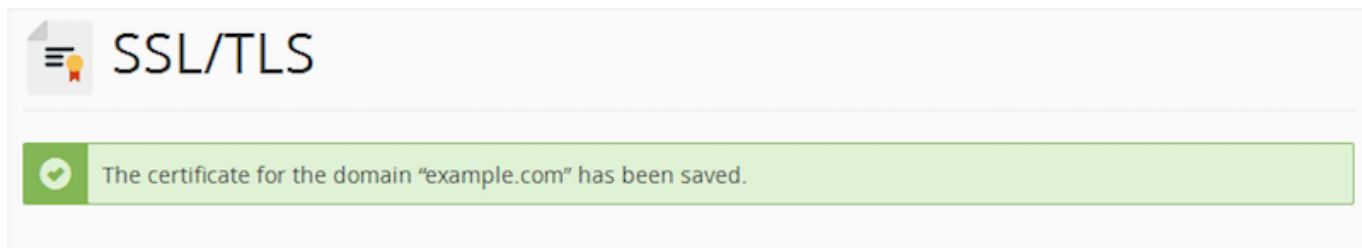
Description

Save Certificate

Self-signed
certificates will cause
browser warnings.
([More information](#))

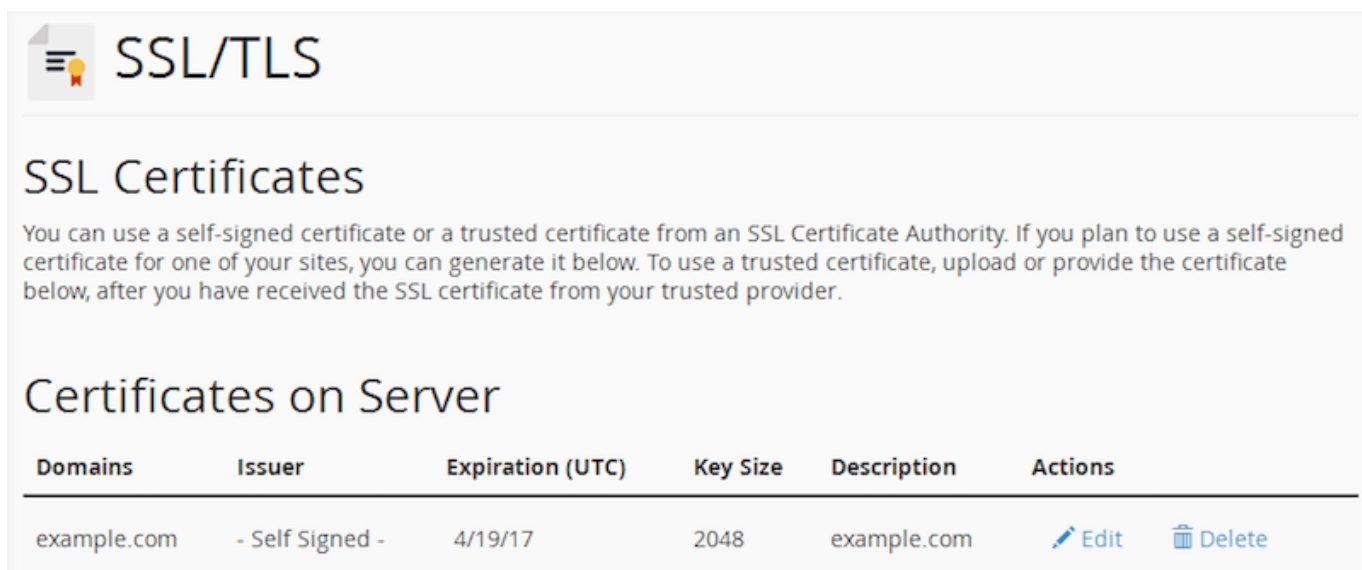
Просмотр содержимого и подтверждение сертификата
HTTPS в cPanel ([см. большую версию](#))

4. Новый сертификат будет сохранён, а вы увидите окно с подтверждением.



Подтверждение сертификата HTTPS в cPanel (см. [большую версию](#))

5. Если вернётесь в раздел “Certificates (CRT)”, то увидите там свой новый сертификат HTTPS.



Страница “Certificates” в cPanel с новым сертификатом HTTPS. (см. [большую версию](#))

6. Вернитесь в раздел “SSL/TLS Manager”. Нажмите “Install and Manage SSL for your website (HTTPS)”, чтобы присвоить существующему веб-сайту новый сертификат.

SSL/TLS

The SSL/TLS Manager will allow you to generate SSL certificates, certificate signing requests, and private keys. These are all parts of using SSL to secure your website. SSL allows you to secure pages on your site so that information such as logins, credit card numbers, etc are sent encrypted instead of plain text. It is important to secure your site's login areas, shopping areas, and other pages where sensitive information could be sent over the web.

Private Keys (KEY)

[Generate, view, upload, or delete your private keys.](#)

Certificate Signing Requests (CSR)

[Generate, view, or delete SSL certificate signing requests.](#)

Certificates (CRT)

[Generate, view, upload, or delete SSL certificates.](#)

Install and Manage SSL for your site (HTTPS)

[Manage SSL sites.](#)

Раздел “SSL/TLS Manager” в cPanel. (см. [большую версию](#))

7. Вам будет предложена новая форма “Install an SSL Website”. Нажмите кнопку “Browse Certificates” и выберите свой сертификат HTTPS. Выберите домен своего веб-сайта из выпадающего меню (если он не выбран автоматически) и проверьте значения полей “Certificate” и “Private Key”.

SSL/TLS

Manage SSL Hosts

This interface lets you configure SSL for your domains.

An SSL certificate can secure one or more domains; to create an SSL host for a domain, you must have a certificate that secures that domain. Each SSL certificate has a matching key file that must also be present to install the certificate. SSL certificates for production use usually also require a CA bundle, which this page will automatically try to obtain from the server; in the event that the server cannot find the required CA bundle, you will need to paste it here.

You may only create SSL hosts for domains that are currently attached to your account. Before you install an SSL certificate for a domain that is not listed below, you must attach the domain to your account as one of the following:

- [Subdomain](#)
- [Addon Domain](#)
- [Parked Domain](#)

When cPanel installs an SSL certificate onto one of your domains, it also installs the same certificate onto that domain's

When cPanel installs an SSL certificate onto one of your domains, it also installs the same certificate onto that domain's "www" subdomain, and vice-versa. Unless your certificate matches both domains, however, only one of the two domains will show as a secure site in a user's web browser.

Install an SSL Website

Note: You do not have a dedicated IP address. As a result, web browsers that do not support [SNI](#) will probably give false security warnings to your users when they access any of your SSL websites. Microsoft® Internet Explorer™ on Windows XP™ is the most widely used web browser that does not support SNI.

Browse Certificates

Domain

Select a Domain

IP Address

Certificate: (CRT)

The certificate may already be on your server. You can either paste the certificate here or try to retrieve it for your domain.

Private Key (KEY)

The private key may already be on your server. You can either paste the private key here or try to retrieve the matching key for your certificate.

Certificate Authority Bundle: (CABUNDLE)

In most cases, you do not need to supply the CA bundle because the server will fetch it from a public repository during installation.

☒ Enable [SNI](#) for Mail Services

Install Certificate

Reset

Форма “Install an SSL Website” в cPanel. ([см. большую версию](#))

Проверьте, что у вас есть доступ к веб-сайту по адресу `https://www.example.com`. Если всё работает нормально, то вы, вероятно, захотите поставить постоянный редирект HTTP-трафика на HTTPS. Для этого нужно добавить несколько строчек в файл `.htaccess` (если у вас веб-сервер Apache) в корневой директории на своём сервере.

```
RewriteEngine On
```

```
RewriteCond %{HTTPS} off
```

```
RewriteRule ^(.*)$ https://%{HTTP_HOST}%  
{REQUEST_URI} [L,R=301]
```

Если файл `.htaccess` уже существует, то вставьте только строчки `RewriteCond` и `RewriteRule` сразу после существующей директивы `RewriteEngine On`.

Linux, FreeBSD

Разместите в соответствующих директориях сгенерированный

секретный ключ (`example.com.key`), запрос на подпись сертификата (`example.com.csr`) и действительный сертификат HTTPS (`example.com.crt`):

- **Debian, Ubuntu** и клоны, **FreeBSD**

```
cp example.com.crt /etc/ssl/certs/  
cp example.com.key /etc/ssl/private/  
cp example.com.csr /etc/ssl/private/
```

- **Red Hat, CentOS** и клоны

```
cp example.com.crt /etc/pki/tls/certs/  
cp example.com.key /etc/pki/tls/private/  
cp example.com.csr /etc/pki/tls/private/  
restorecon -RvF /etc/pki
```

Файлы должны принадлежать руту и быть защищены настройкой разрешения 600.

- **Debian, Ubuntu** и клоны

```
chown -R root. /etc/ssl/certs /etc/ssl/private  
chmod -R 0600 /etc/ssl/certs /etc/ssl/private
```

- **Red Hat, CentOS** и клоны

```
chown -R root. /etc/pki/tls/certs  
/etc/pki/tls/private  
chmod -R 0600 /etc/pki/tls/certs  
/etc/pki/tls/private
```

- **FreeBSD**

```
chown -R root:wheel /etc/ssl/certs
```



```
/etc/ssl/private
```

```
chmod -R 0600 /etc/ssl/certs /etc/ssl/private
```

Apache

Чтобы активировать HTTPS на своём сайте, нужно сделать следующее:

- убедиться, что на сервере установлен **mod_ssl**,
- загрузить на сервер файл полученного сертификата HTTPS (.crt),
- отредактировать файлы конфигурации сервера Apache.

Начните с проверки `mod_ssl`. В зависимости от операционной системы, должен работать один из вариантов:

```
apache2 -M | grep ssl
```

или

```
httpd -M | grep ssl
```

Если `mod_ssl` установлен, то вы получите такой ответ...

```
ssl_module (shared)
```

```
Syntax OK
```

... или нечто похожее.

Если он не установлен или не работает, то попробуйте это:

- **Debian, Ubuntu** и клоны

```
sudo a2enmod ssl  
sudo service apache2 restart
```

- **Red Hat, CentOS** и клоны

```
sudo yum install mod_ssl  
sudo service httpd restart
```

- **FreeBSD**

```
make -C /usr/ports/www/apache24 config install  
clean  
apachectl restart
```

Отредактируйте файл конфигурации Apache (httpd.conf):

- **Debian, Ubuntu**

```
/etc/apache2/apache2.conf
```

- **Red Hat, CentOS**

```
/etc/httpd/conf/httpd.conf
```

- **FreeBSD**

```
/usr/local/etc/apache2x/httpd.conf
```

```
Listen          80  
Listen          443  
  
<VirtualHost *:80>  
    ServerName example.com  
    ServerAlias www.example.com  
    Redirect 301 / https://www.example.com/  
</VirtualHost>
```

```

<VirtualHost *:443>
    ServerName example.com
    Redirect 301 / https://www.example.com/
</VirtualHost>

<VirtualHost *:443>
    ServerName www.example.com
    ...
    SSLEngine on
    SSLCertificateFile/path/to/signed_certificate_followed_by_in
intermediate_certs
    SSLCertificateKeyFile /path/to/private/key

    # Uncomment the following directive when using client
certificate authentication
    #SSLCACertificateFile
/path/to/ca_certs_for_client_authentication

    # HSTS (mod_headers is required) (15768000 seconds = 6
months)
    Header always set Strict-Transport-Security "max-
age=15768000"
    ...
</VirtualHost>

# intermediate configuration, tweak to your needs
SSLProtocol                                all -SSLv3
SSLCipherSuite                            ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-
CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-
GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-
SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-
ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-
SHA:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES256-
SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES128-
SHA256:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA256:DHE-RSA-AES256-
SHA:ECDHE-ECDSA-DES-CBC3-SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-
CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-
SHA256:AES128-SHA:AES256-SHA:DES-CBC3-SHA:!DSS
SSLHonorCipherOrder                       on
SSLCompression                           off
SSLSessionTickets                         off

# OCSP Stapling, only in httpd 2.3.3 and later
SSLUseStapling                             on
SSLStaplingResponderTimeout                 5
SSLStaplingReturnResponderErrors           off

```

```
SSLStaplingCache  
shmcb:/var/run/ocsp(128000)
```

Эта конфигурация была сгенерирована с помощью упомянутого ранее [Mozilla SSL Configuration Generator](#). С его помощью проверьте актуальность конфигурации. Отредактируйте правильные пути для сертификата и секретного ключа. Показанная здесь конфигурация сгенерирована с промежуточными настройками — прочитайте об ограничениях и конфигурациях браузера для каждой настройки, прежде чем выбрать наиболее подходящую для себя.

В коде были сделаны некоторые изменения, чтобы обрабатывать редиректы с HTTP на HTTPS, а также с не-`www` на домен с `www` (полезно для задач SEO).

Nginx

Отредактируйте файл конфигурации nginx (`nginx.conf`):

- **Debian, Ubuntu, Red Hat, CentOS**

```
/etc/nginx/nginx.conf
```

- **FreeBSD**

```
/usr/local/etc/nginx/nginx.conf
```

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    # Redirect all HTTP requests to HTTPS with a 301 Moved  
    Permanently response.  
    return 301 https://$host$request_uri;
```

```
}
```

```
server {  
    listen 443 ssl http2;  
    listen [::]:443 ssl http2;  
  
    # certs sent to the client in SERVER HELLO are  
concatenated in ssl_certificate  
    ssl_certificate /path/to/signed_cert_plus_intermediates;  
    ssl_certificate_key /path/to/private_key;  
    ssl_session_timeout 1d;  
    ssl_session_cache shared:SSL:50m;  
    ssl_session_tickets off;  
  
    # Diffie-Hellman parameter for DHE ciphersuites,  
recommended 2048 bits  
    ssl_dhparam /path/to/dhparam.pem;  
  
    # intermediate configuration. tweak to your needs.  
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  
    ssl_ciphers 'ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-  
CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-  
GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-  
SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-  
ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-  
SHA:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-  
AES256-SHA384:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-  
RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA256:DHE-  
RSA-AES256-SHA:ECDHE-ECDSA-DES-CBC3-SHA:ECDHE-RSA-DES-CBC3-  
SHA:EDH-RSA-DES-CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-  
SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:DES-  
CBC3-SHA:!DSS';  
    ssl_prefer_server_ciphers on;  
  
    # HSTS (ngx_http_headers_module is required) (15768000  
seconds = 6 months)  
    add_header Strict-Transport-Security max-age=15768000;  
  
    # OCSP Stapling ---  
    # fetch OCSP records from URL in ssl_certificate and  
cache them  
    ssl_stapling on;  
    ssl_stapling_verify on;  
  
    ## verify chain of trust of OCSP response using Root CA  
and Intermediate certs  
    ssl_trusted_certificate  
/path/to/root_CA_cert_plus_intermediates;
```

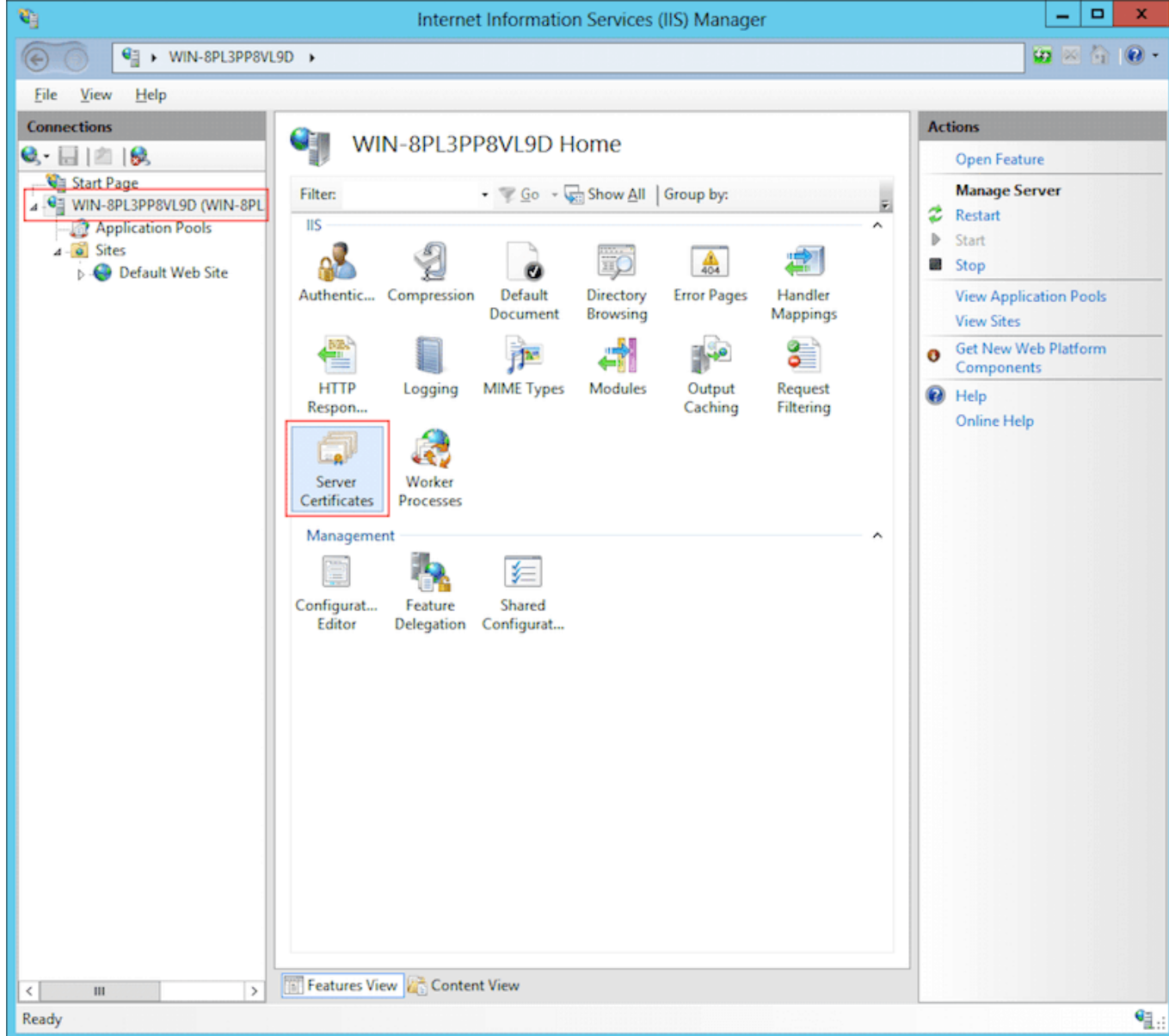
```
resolver <IP DNS resolver>;  
  
....  
}
```

Эта конфигурация была сгенерирована с помощью упомянутого ранее [Mozilla SSL Configuration Generator](#). С его помощью проверьте актуальность конфигурации. Отредактируйте правильные пути для сертификата и секретного ключа. Показанная здесь конфигурация сгенерирована с промежуточными настройками — прочитайте об ограничениях и конфигурациях браузера для каждой настройки, прежде чем выбрать наиболее подходящую для себя.

Генератор автоматически генерирует код для обработки редиректов с HTTP на HTTPS и изначально активирует поддержку HTTP/2!

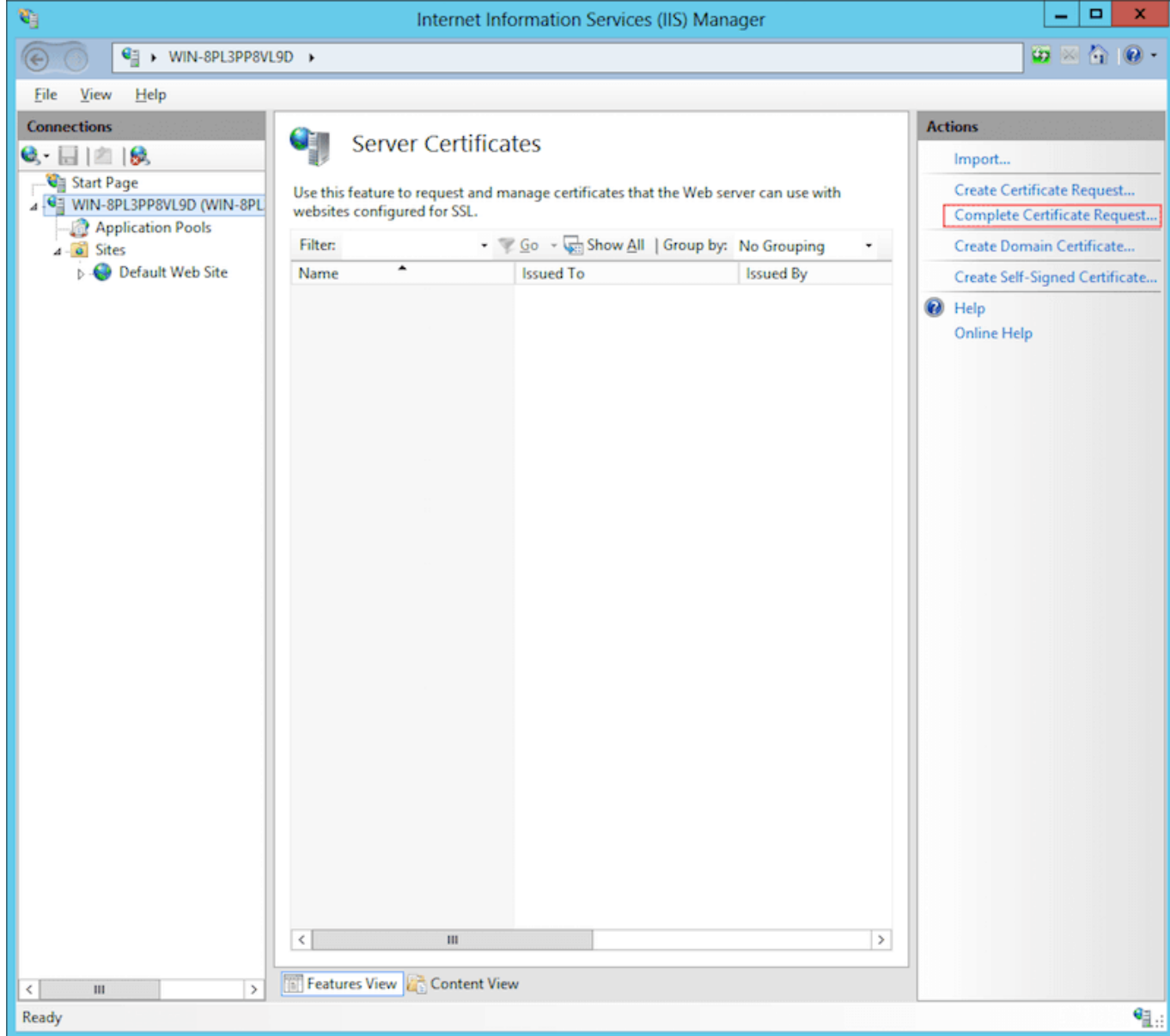
Internet Information Server под Windows

1. Откройте “Start” → “Administrative Tools” → “Internet Information Services (IIS) Manager”. Нажмите на название сервера. Двойным щелчком откройте “Server Certificates” в средней колонке.



Откройте “Internet Information Services (IIS) Manager”. Двойным щелчком откройте “Server Certificates”. (см. большую версию)


2. Нажмите “Complete Certificate Request” в правой колонке.



Нажмите “*Complete Certificate Request*” в правой колонке. (см. [большую версию](#))

3. Выберите файл подписанного сертификата (`example.com.crt`), который вы получили от центра сертификации. Введите какое-нибудь название в поле “Friendly name”, чтобы различать сертификаты впоследствии. Поместите новый сертификат в хранилище сертификатов “Personal” (IIS 8+). Нажмите “OK”.

Complete Certificate Request

 **Specify Certificate Authority Response**

Complete a previously created certificate request by retrieving the file that contains the certificate authority's response.

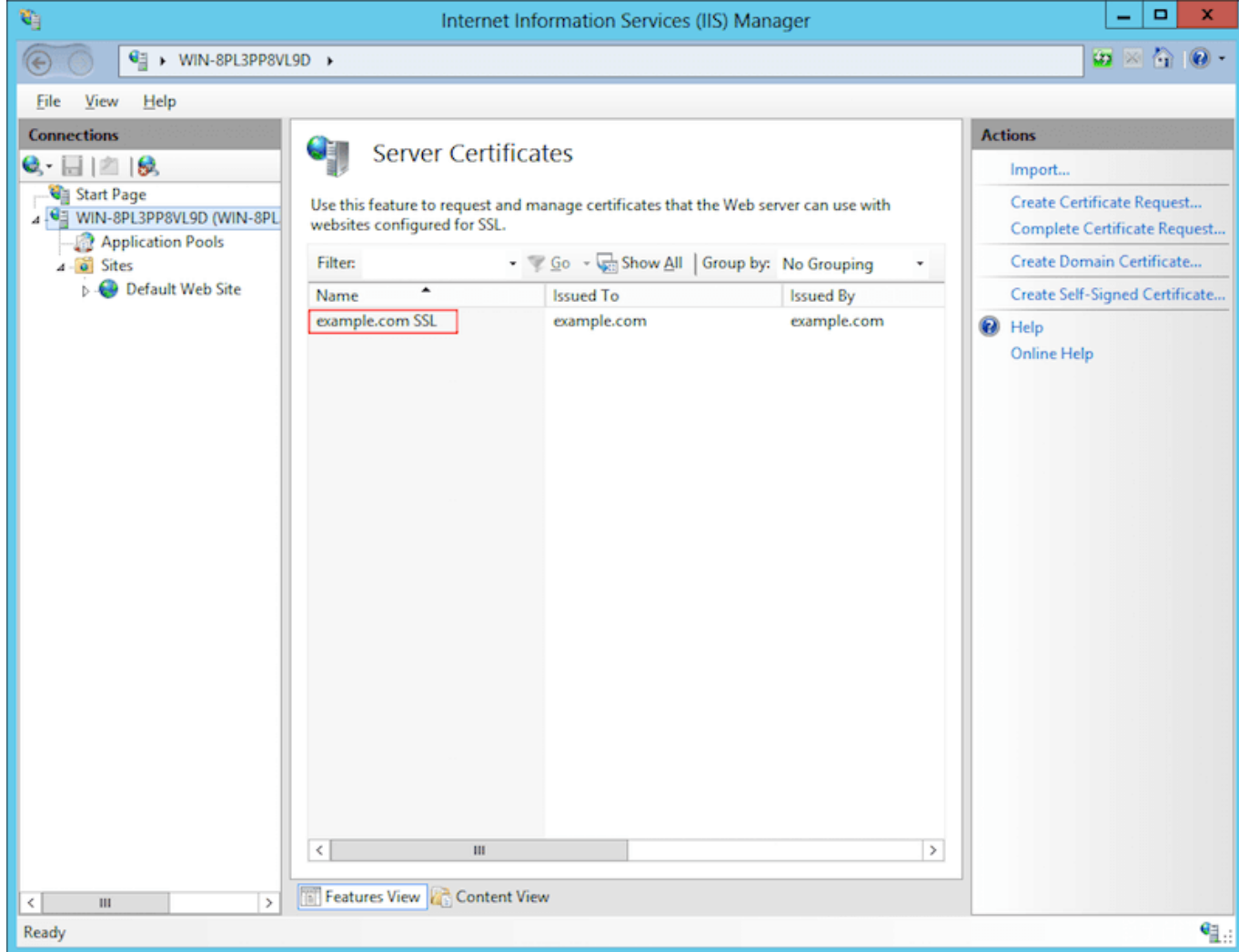
File name containing the certification authority's response:

Friendly name:

Select a certificate store for the new certificate:

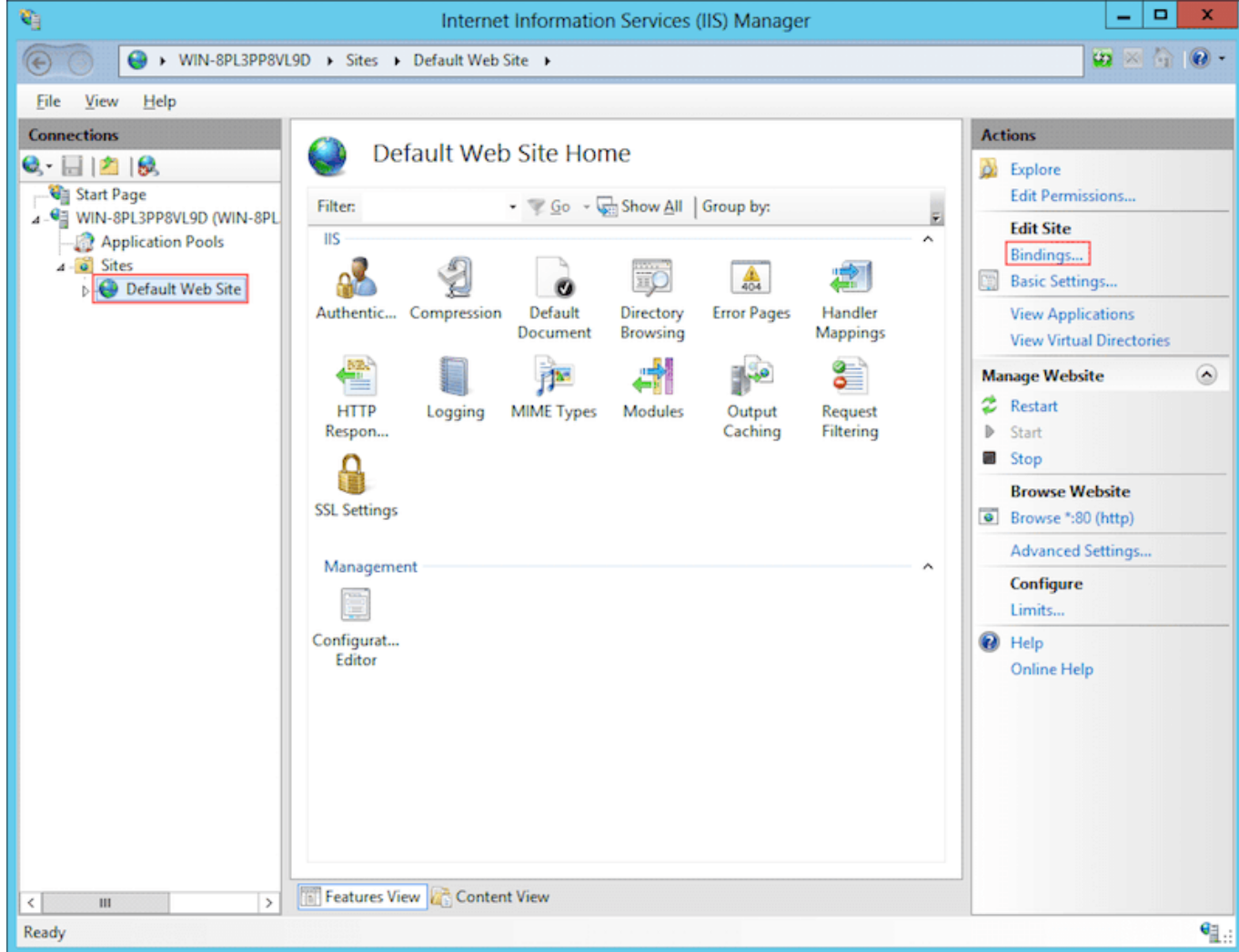
Выберите файл подписанного сертификата. (см. большую версию)

4. Если процесс прошёл нормально, вы должны увидеть сертификат на вкладке “Server Certificates”.



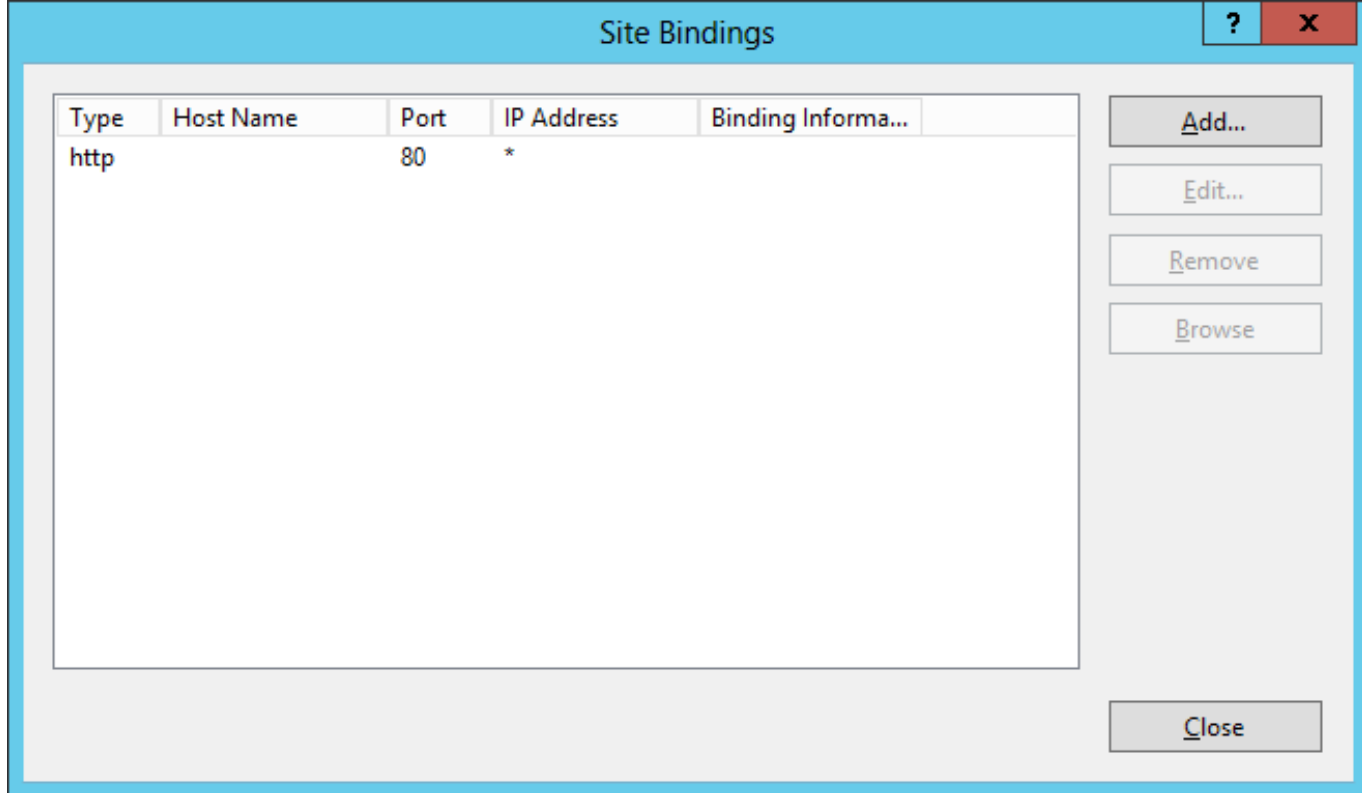
Вы должны увидеть сертификат на вкладке “Server Certificates”. (см. [большую версию](#))

5. Расширьте вкладку с названием сервера. В разделе “Sites” выберите веб-сайт, которому вы хотите присвоить сертификат HTTPS. Нажмите “Bindings” в правой колонке.



Выберите веб-сайт и нажмите “Bindings”. (см. большую версию)

6. В окне “Site Bindings” нажмите кнопку “Add”.



Нажмите кнопку “Add”. (см. большую версию)

7. В новом окне выберите:

- “Type”: “https”
- “IP address”: “All Unassigned”
- “Port”: “443”

В поле “SSL Certificate” выберите установленный сертификат HTTPS по присвоенному ему имени. Нажмите “OK”.

Add Site Binding

Type: https IP address: All Unassigned Port: 443

Host name:

☐ Require Server Name Indication

SSL certificate: example.com SSL Select... View...

OK Cancel

Выберите “HTTPS” и укажите установленный сертификат HTTPS. (см. [большую версию](#))

8. Теперь для вашего веб-сайта должны быть установлены HTTP и HTTPS.

Site Bindings

| Type | Host Name | Port | IP Address | Binding Informa... |
|-------|-----------|------|------------|--------------------|
| http | | 80 | * | |
| https | | 443 | * | |

Add... Edit... Remove Browse

Close

Теперь для вашего веб-сайта должны быть установлены HTTP и HTTPS. ([см. большую версию](#))

Предупреждения о смешанном содержимом

Возле адресной строки вы можете увидеть предупреждающий знак и сообщение вроде «Соединение небезопасно! Части страницы не защищены (такие как изображения)». Это не означает, что вы неправильно установили сертификат: просто убедитесь, что ссылки на все ресурсы (изображения, таблицы стилей, скрипты и др.), как локальные так и с удалённых серверов, не начинаются с `http://`. Все ресурсы должны указывать на адреса относительно рута (`/images/image.png`, `/styles/style.css` и т. д.) или относительно текущего документа (`../images/image.png`), или это должны быть полные URL, которые начинаются с `https://`, такие как `<script src="https://code.jquery.com/jquery-3.1.0.min.js"></script>`.

Эти советы помогут устранить предупреждения о смешанном содержимом, а ваш браузер должен показывать закрытый замок без восклицательного знака.

Тестирование сервера

После того, как ваш сервер сконфигурирован и начал работать по HTTPS, я настоятельно рекомендую проверить безопасность конфигурации при помощи [Qualys SSL Server Test](#). Он сканирует ваш веб-сайт, в том числе выполняет всестороннюю оценку

конфигурации, выявляет возможные слабости и даёт рекомендации. Следуйте его советам для ещё большего улучшения конфигурации защиты сервера.

Продление

Ваш сертификат действителен в течение определённого периода времени — обычно это год. Не дожидайтесь последнего момента для его продления — ваш регистратор начнёт присылать вам электронные письма, когда станет приближаться срок для обновления. Выпустите новый сертификат как только получите первое уведомление. Процедура примерно такая же: создать запрос на подпись сертификата, получить новый сертификат HTTPS и установить его на сервер. Срок действия сертификата начинается с момента его подписи, в то время как срок окончания его действия будет установлен через год после окончания действия предыдущего сертификата. Поэтому будет определённый промежуток времени, когда оба сертификата действительны, а затем полный год после окончания срока действия старого сертификата. В течение этого перекрытия у вас есть возможность убедиться, что новый сертификат нормально работает, прежде чем срок действия старого истечёт, что гарантирует бесперебойную работу вашего веб-сайта.

Отзыв

Если ваш сервер скомпрометирован или вы думаете, что кто-то мог получить доступ к вашему секретному ключу, вам следует немедленно аннулировать текущий сертификат HTTPS. У разных

регистраторов разные процедуры, но в целом всё сводится к пометке скомпрометированного сертификата как неактивного в специальной базе данных вашего регистратора, а затем выдачи нового сертификата HTTPS. Конечно, отзывайте текущий сертификат как можно раньше, чтобы никто не мог выдать себя за вас, и устанавливайте новый сертификат только после того, как выясните и исправите причину возникновения бреши в безопасности. Можете попросить помощи у своего регистратора.

Let's Encrypt

Процитируем сайт [Let's Encrypt](#):

Let's Encrypt — это бесплатный, автоматизированный и открытый центр сертификации (CA), который работает для общественного блага. Сервис Let's Encrypt предоставляется [Internet Security Research Group \(ISRG\)](#).

Ключевые принципы Let's Encrypt:

- **Бесплатность**

Любой владелец доменного имени может использовать Let's Encrypt для получения бесплатного доверенного сертификата.

- **Автоматизация**

Программное обеспечение на веб-сервере может взаимодействовать с Let's Encrypt для безболезненного получения сертификата, безопасной конфигурации его и автоматического продления.

- **Безопасность**

Let's Encrypt служит платформой для лучших методов продвинутой защиты TLS, как на стороне центра сертификации, так и помогая операторам веб-сайтов правильно обезопасить свои сервера.

- **Прозрачность**

Все выданные или отозванные сертификаты будут публично записаны и доступны для просмотра кем угодно.

- **Открытость**

Протокол автоматической выдачи и продления будет опубликован как открытый стандарт, который могут использовать другие.

- **Сотрудничество**

Во многом как протоколы, лежащие в основе Интернета, Let's Encrypt — это совместный проект для блага сообщества, неподконтрольный какой-либо организации.

Чтобы воспользоваться преимуществами Let's Encrypt, нужно правильно настроить свой аккаунт на хостинге или сервере. Let's Encrypt выдаёт краткосрочные сертификаты, которые следует регулярно обновлять, чтобы веб-сайт HTTPS оставался работоспособным.

Как это работает

Есть некоторые существенные отличия в работе Let's Encrypt и других центров сертификации. В соответствии с тремя первыми пунктами, перечисленными выше, вот эти отличия:

- **Бесплатность**

Сертификаты HTTPS от Let's Encrypt абсолютно бесплатны на весь срок жизни вашего сайта.

- **Автоматизация**

Сертификаты HTTPS от Let's Encrypt [действуют 90 дней](#), в отличие от обычных сертификатов HTTPS, которые действуют один год. Людей подталкивают к **автоматизации** обновления своих сертификатов; например, администратор сервера может запустить специализированный программный сервис (или периодически вызывать программу из cron) для управления первоначальной проверкой домена и последующими продлениями для всех своих доменов — в стиле «установил и забыл».

- **Безопасность**

Сертификаты HTTPS от Let's Encrypt выдаются без компромиссов относительно безопасности, что ведёт к некоторым несовместимостям со старыми и более экзотическими платформами. Посмотрите [страницу совместимости](#) для проверки, не попадаете ли вы в отсекаемые платформы.

Ограничения

Let's Encrypt выдаёт только сертификаты DV. Сертификаты OV и EV не поддерживаются, и в данный момент нет планов их поддержки. Выдаются сертификаты на один или несколько доменов, но в данный момент нет сертификатов с поддоменами (подстановочными знаками). Для получения дополнительной информации см. [Let's Encrypt FAQ](#).

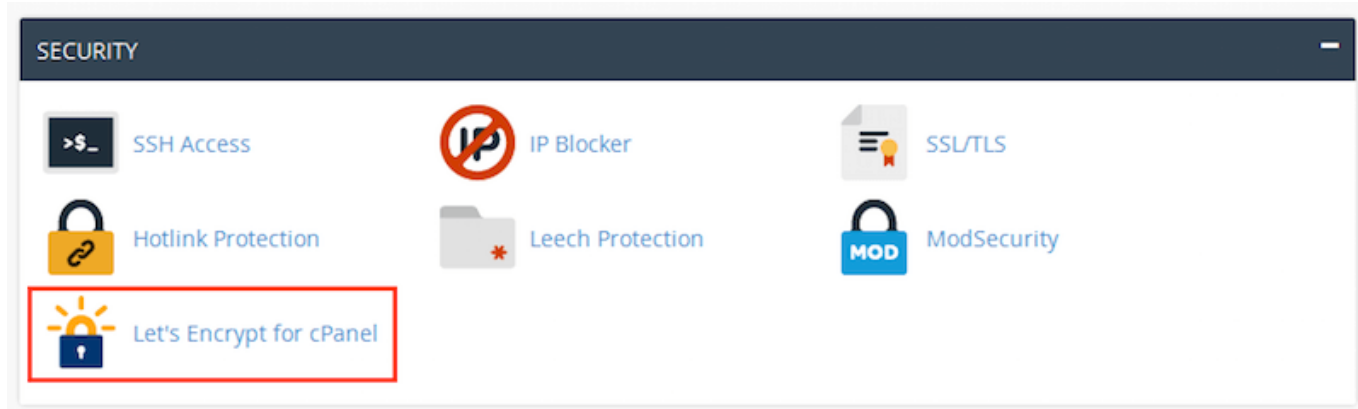
Автоматический режим работы Let's Encrypt навязывает некоторые [ограничения на использование](#), чтобы защитить инфраструктуру от умышленных и неумышленных злоупотреблений. Лимиты на интенсивность использования достаточно высоки, чтобы помешать обычным пользователям даже с сотнями доменов в распоряжении. Но если вы управляете сертификатами HTTPS в очень больших масштабах, то стоит ознакомиться с этими лимитами.

Более старые и экзотические клиенты (до Windows XP SP3) не поддерживаются. Подробности см. на [странице совместимости](#).

Использование сертификатов HTTPS от Let's Encrypt на практике

cPanel

1. Зайдите в cPanel своего хоста.
2. Прокрутите вниз до раздела “Security” и нажмите “Let's Encrypt for cPanel”.



Раздел “Security” в cPanel. ([см. большую версию](#))

3. Теперь вы в разделе “Let’s Encrypt for cPanel”. Проверьте оба доменных имени (example.com и www.example.com) и нажмите “Issue Multiple”.

Let's Encrypt SSL

Let's Encrypt is an effort to provide free domain-validated certificates in an automated fashion. This page provides a facility to issue certificates via the Let's Encrypt service. Certificates issued here will be renewed automatically.

[View settings](#)

Your domains with Let's Encrypt certificates

Show 10 entries Search:

| Domain | Alt Names | Status | Expiry | Actions |
|----------------------------|-----------|--------|--------|---------|
| No data available in table | | | | |

Showing 0 to 0 of 0 entries Previous Next

Issue a new certificate

Choose from one of your domains below. A new key and certificate will be added to the SSL/TLS manager.

[+ Issue Multiple](#)

Show 10 entries Search:

| <input type="checkbox"/> | Domain | Root | Type | Path | Actions |
|-------------------------------------|-----------------|-------------|-------|---------------------------|--------------------------------|
| <input checked="" type="checkbox"/> | example.com | example.com | Main | /home/example/public_html | + Issue Single |
| <input checked="" type="checkbox"/> | www.example.com | example.com | Alias | /home/example/public_html | + Issue Single |

Showing 1 to 2 of 2 entries Previous 1 Next

[+ Issue Multiple](#)

*Проверьте оба доменных имени и нажмите “Issue Multiple”.
(см. большую версию)*

4. Вы увидите экран подтверждения. В качестве основного будет выбран ваш домен верхнего уровня (то есть не-`www`, а в качестве алиаса указан домен `www`, он будет помещён в записи “Subject Alt Name” (SAN) сертификата HTTPS. Для продолжения нажмите “Issue”. **Пожалуйста, будьте терпеливы и не**

обновляйте эту страницу, потому что изначальная проверка может занять некоторое время — минуту или две.

Let's Encrypt SSL

Issue certificates for the following domains:

Installing certificate to: example.com

| Domain | Type | Document Root | Include? | Primary? |
|-----------------|-------|---------------------------|-------------------------------------|----------------------------------|
| example.com | Main | /home/example/public_html | <input checked="" type="checkbox"/> | <input checked="" type="radio"/> |
| www.example.com | Alias | /home/example/public_html | <input checked="" type="checkbox"/> | <input type="radio"/> |

☒ Install mail SMTPS/POP3S/IMAPS SSL certificate for example.com

[Issue](#)

[Go Back](#)

Нажмите “Issue” и подождите минуту или две. ([см. большую версию](#))

5. Если процесс завершён удачно, вы увидите сообщение с подтверждением. Нажмите «Назад», чтобы посмотреть установленный сертификат.

Let's Encrypt SSL

The SSL certificate is now installed onto the domain “example.com”.
Apache is restarting in the background.

[Go Back](#)

Если процесс завершён удачно, вы увидите сообщение с подтверждением. ([см. большую версию](#))

6. Вы увидите свой домен в списке “Your domains with Let’s Encrypt certificates”. Можете проверить детали сертификата и убедиться, что веб-сайт открывается в браузере с префиксом `https://`.

Let's Encrypt SSL

Let's Encrypt is an effort to provide free domain-validated certificates in an automated fashion. This page provides a facility to issue certificates via the Let's Encrypt service. Certificates issued here will be renewed automatically.

[View settings](#)

Your domains with Let's Encrypt certificates

Show **10** entries Search:

| Domain | Alt Names | Status | Expiry | Actions |
|-------------|-----------------|-----------|-------------|--|
| example.com | www.example.com | Installed | 28 Oct 2016 | Remove Reinstall View |

Showing 1 to 1 of 1 entries Previous **1** Next

Issue a new certificate

Choose from one of your domains below. A new key and certificate will be added to the SSL/TLS manager.

[+ Issue Multiple](#)

Show **10** entries Search:

| <input type="checkbox"/> Domain | <input type="checkbox"/> Root | <input type="checkbox"/> Type | <input type="checkbox"/> Path | <input type="checkbox"/> Actions |
|--|-------------------------------|-------------------------------|-------------------------------|----------------------------------|
| <input type="checkbox"/> example.com | example.com | Main | /home/example/public_html | + Issue Single |
| <input type="checkbox"/> www.example.com | example.com | Alias | /home/example/public_html | + Issue Single |

Showing 1 to 2 of 2 entries Previous **1** Next

[+ Issue Multiple](#)

Ваши домены с сертификатами Let’s Encrypt. (см. [большую версию](#))

Linux, FreeBSD, другие

Самый простой способ установить сертификат Let's Encrypt на своём сервере — использовать [Certbot](#). Просто укажите свой веб-сайт и операционную систему — и следуйте инструкциям.

[home](#) [about certbot](#) [faq](#) [documentation](#) [support](#) [source code](#) [donate to EFF](#)



Automatically enable HTTPS on your website with EFF's
Certbot, deploying [Let's Encrypt](#) certificates.

I'm using on

To get instructions for Certbot, choose your webserver and server operating system from the dropdown menus above. You can then pick "advanced" if you want less automation and more control.



Certbot для Let's Encrypt (см. [большую версию](#))

Internet Information Server под Windows

В данный момент нет официального клиента для IIS под Windows, но существует пара обходных путей.

Несколько проектов ставят целью создание нативного Windows-

клиента для Let's Encrypt:

- [ACMESharp](#) (PowerShell) — первая попытка написать Windows-клиент.
- [letsencrypt-win-simple](#) (для командной строки) как будто самый простой в использовании.
- [Certify](#) предоставляет GUI поверх ACMESharp, но всё ещё находится в альфа-версии.

Cloudflare

[Cloudflare](#) — сервис, который предоставляет сеть доставки контента (CDN), услуги обеспечения безопасности для веб-сайтов и защиты от DDoS-атак. Он предоставляет бесплатные сертификаты HTTPS на всех тарифных планах, включая бесплатный тариф — это коллективный сертификат DV Cloudflare Universal SSL. Чтобы получить уникальный сертификат HTTPS, нужно перейти на бизнес-тариф.

Для получения сертификата просто создайте аккаунт, поднимите веб-сайт и зайдите в раздел “Crypto”.

CertSimple

[CertSimple](#) поставляет только сертификаты EV. Он совершил такую же революцию на рынке сертификатов EV HTTPS, какую Let's Encrypt совершил на рынке сертификатов DV HTTPS, обеспечивая более быстрый и простой процесс проверки организации, который

обычно медленный и обременительный. Вот его преимущества:

- **Упрощённая процедура подачи заявлений**

Не требуется установка программного обеспечения или ответы на вопросы в командной строке. Проверка в реальном времени, а большинство деталей проверяются до оплаты.

- **Быстрая проверка**

В среднем, три часа, по сравнению со средними по отрасли 7-10 сутками.

- **Бесплатный перевыпуск на протяжении всего срока действия сертификата**

Легко можно добавить позже доменные имена или восстановить потерянный секретный ключ.

Несколько веб-сайтов HTTPS на одном IP-адресе

Из-за сути процесса рукопожатия виртуальные хосты на одном IP-адресе представляют собой проблему для TLS. Виртуальные хосты работоспособны благодаря тому, что клиент включает доменное имя в заголовок запроса HTTP, но при использовании HTTPS рукопожатие TLS происходит до того, как отправляются первые запросы HTTP — нужно установить и наладить работу безопасного канала, прежде чем отправлять какой-либо открытый текст по HTTP, в том числе заголовки. Так что перед соединением с клиентом сервер не знает, какой сертификат предъявить, поэтому он показывает первый сертификат из своего файла конфигурации. И конечно, этот сертификат действителен только для первого сайта TLS из списка.

Есть несколько способов обойти проблему: либо получить уникальные IP-адреса для каждого домена с TLS, либо зарегистрировать все домены на один сертификат. Оба способа не слишком хороши на практике — адресное пространство IPv4 уже исчерпано, а регистрация всех сайтов на один большой сертификат HTTPS означает, что при добавлении нового сайта на сервер вам придётся перевыпускать весь сертификат на многочисленные домены.

Для устранения этого ограничения было разработано расширение к протоколу TLS под названием [Server Name Indication \(SNI\)](#). Его должны поддерживать и сервер, и клиент. И хотя поддержка SNI сегодня широко распространена, она всё-таки не гарантирована на 100%, если для вас важна гарантия совместимости со всеми возможными клиентами.

Подробности о запуске SNI под [Apache](#), [nginx](#) и [IIS \(8+\)](#) см. в соответствующей документации.

Полезные ресурсы

- [Mozilla SSL Configuration Generator](#)
- [Серверный тест SSL](#), Qualys
- [Безопасность TLS на стороне сервера](#), вики Mozilla
- [Лучшие практики внедрения SSL и TLS](#), SSL Labs
- [Документация](#), Qualys SSL Labs
- [Скрипт на PHP для поиска и замены в БД](#), Interconnect IT. Для замены всех упоминаний HTTP на HTTPS (ссылки, изображения и др.) в базе данных WordPress.

Проголосовать:



+61



Поделиться:



Сохранить:



Комментарии (54)

Похожие публикации

Защита от прослушивания SIP с помощью — TLS + SRTP + шифрованный туннель и телефона Yealink T26p

MotjaX • 13 июля 2013 в 21:11

13

Поддержка протоколов TLS/SSL для сокетного соединения на AS3

FSB • 25 декабря 2012 в 14:43

6

Хакеры взломали SSL шифрование, используемое миллионами сайтов

ПЕРЕВОД

uglymeta • 21 сентября 2011 в 00:17

74

Популярное за сутки

Яндекс открывает Алису для всех разработчиков. Платформа Яндекс.Диалоги (бета)

69

BarakAdama • вчера в 10:52

Почему следует игнорировать истории основателей успешных стартапов

20

ПЕРЕВОД

m1rko • вчера в 10:44

Как получить телефон (почти) любой красоты в Москве, или интересная особенность MT_FREE

24

ИЗ ПЕСОЧНИЦЫ

sab404 • вчера в 20:27

Java и Project Reactor

10

zealot_and_frenzy • вчера в 10:56

Пользовательские агрегатные и оконные функции в PostgreSQL и Oracle

6

erogov • вчера в 12:46

Лучшее на Geektimes

Как фермеры Дикого Запада организовали телефонную сеть на колючей проволоке

NAGru • вчера в 10:10

31

Энтузиаст сделал новую материнскую плату для ThinkPad X200s

alizar • вчера в 15:32

49

Кто-то посылает секс-игрушки с Amazon незнакомцам. Amazon не знает, как их остановить

Pochtoycom • вчера в 13:06

85

Илон Маск продолжает убеждать в необходимости создания колонии людей на Марсе

marks • вчера в 14:19

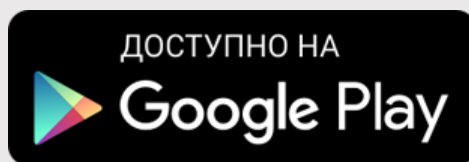
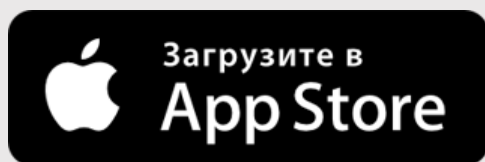
140

Дела шпионские (часть 1)

TashaFridrih • вчера в 13:16

16

Мобильное приложение



Полная версия

