HTTPS HyperText Transfer Protocol Secure

29 апреля 2023 года

HTTPS (HyperText Transfer Protocol Secure) - это протокол, который шифрует данные, передаваемые между сервером и клиентом во время обмена информацией в Интернете. HTTPS использует SSL/TLS-шифрование для защиты данных от перехвата злоумышленниками. HTTPS был разработан в 1994 году компанией Netscape Communications Corporation для обеспечения безопасности передачи данных в Интернете.

Использование HTTPS на сайте очень важно для обеспечения безопасности пользователей при передаче конфиденциальной информации, такой как пароли, номера кредитных карт, персональные данные и т.д. Если сайт не использует HTTPS, злоумышленники могут перехватывать данные, передаваемые через этот сайт, или подделывать страницы сайта для получения личной информации у пользователей.

Для установления безопасного соединения по протоколу HTTPS используется протокол SSL/TLS (Secure Sockets Layer/Transport Layer Security), который обеспечивает проверку подлинности сервера и клиента, а также целостность и конфиденциальность передаваемых данных. Каждый раз, когда пользователь обращается к серверу по протоколу HTTPS, происходит рукопожатие SSL/TLS. В рамках этого процесса клиент и сервер обмениваются информацией о своих ключах и параметрах шифрования, чтобы установить безопасное соединение.

HTTPS использует цифровые сертификаты для проверки подлинности сервера. Цифровые сертификаты выдаются независимыми удостоверяющими центрами (CA), которые производят предварительную проверку идентичности владельца домена. Таким образом, сертификаты гарантируют, что сервер принадлежит конкретному лицу или организации.

HTTPS не только шифрует данные, но и подписывает их цифровой подписью, чтобы проверить их подлинность. Это обеспечивает целостность передаваемых данных и защищает от подмены данных злоумышленниками. HTTPS также может защититься от ряда атак, например от Man-in-the-Middle (MitM), когда злоумышленник может перехватывать и изменять данные, передаваемые между сервером и клиентом, а также от атак типа "cookie theft" и "session hijacking", когда злоумышленник может получить доступ к сессии пользователя и выполнить действия от его имени.

Однако, следует отметить, что использование HTTPS само по себе не гарантирует полную безопасность сайта. Важно также следить за обновлением программного обеспечения сайта, использованием сильных паролей, регулярной проверкой на наличие уязвимостей и другими мерами безопасности.

HTTPS может замедлить скорость загрузки сайта из-за дополнительной нагрузки на сервер, связанной с шифрованием и расшифровкой данных. Однако, благодаря использованию более современных алгоритмов шифрования и ускорению аппаратных возможностей серверов, этот процесс становится все более быстрым. Некоторые браузеры начали помечать в адресной строке страницы, которые не используют HTTPS, как "небезопасные". Это может отразиться на доверии пользователей к сайту и повлиять на его посещаемость. Поэтому сейчас многие сайты переходят на использование HTTPS вместо HTTP.

Кроме того, первоначально HTTPS использовал порт 443 вместо стандартного порта 80 для HTTP. Это было сделано для того, чтобы различать протоколы при передаче данных между серверами.

Для асимметричного шифрования HTTPS использует два ключа: открытый и закрытый. Открытый ключ используется для шифрования сообщений, а закрытый ключ используется для их расшифровки. При этом данные остаются защищенными даже в случае, если злоумышленник узнает открытый ключ.

Существует несколько версий SSL/TLS, каждая из которых имеет свои сильные и слабые стороны в плане безопасности. Например, SSL 2.0 считается устаревшим и небезопасным, а SSL 3.0 и TLS 1.0 также имеют некоторые уязвимости. Поэтому рекомендуется использовать более новые версии протоколов SSL/TLS, такие как TLS 1.2 и 1.3, которые обладают более высоким уровнем безопасности.

В целом, использование HTTPS является одной из важнейших мер для защиты данных пользователей в Интернете и повышения доверия к сайту. Для обеспечения максимальной безопасности рекомендуется не только использовать HTTPS, но и соблюдать другие меры безопасности, такие как регулярную проверку на наличие уязвимостей и использование сильных паролей.