Расширение `ticket_request` (RFC 9149) в TLS 1.3

Автор: Кветко Матвей

1 Проблема

В протоколе TLS 1.3 сессионные билеты (Session Tickets) являются основным механизмом для возобновления предыдущих сессий без полного повторного рукопожатия. Сессионный билет представляет собой криптографически защищённую структуру данных, содержащую информацию, необходимую для восстановления состояния сессии. После успешного установления соединения сервер может отправить клиенту один или несколько билетов в сообщении `NewSessionTicket`. Эти билеты клиент может использовать при следующем подключении для быстрого возобновления сессии (0-RTT или 1-RTT режимы), что значительно снижает задержку установления соединения.

Однако в базовой спецификации TLS 1.3 (RFC 8446) отсутствует стандартизированный способ для клиента **явно запросить** у сервера новый сессионный билет во время рукопожатия. Это приводит к нескольким практическим проблемам:

- **Необходимость обновления ключей:** Для обеспечения защиты от будущей компрометации ключей (Forward Secrecy) клиенту может потребоваться получить новый билет, основанный на актуальных ключах, до истечения срока действия предыдущего. Без явного запроса сервер может не предоставить новый билет.
- **Планирование переподключений:** Клиенту может потребоваться заранее подготовить несколько текущих билетов для быстрого восстановления нескольких будущих соединений (например, в мобильных приложениях при ожидаемой смене сети или для параллельных запросов).
- **Отсутствие гарантий:** Клиент не может быть уверен, что сервер предоставит билет, если только тот не настроен на автоматическую выдачу билетов для всех соединений. Это создаёт неопределённость для приложений, зависящих от механизма возобновления сессии.

Таким образом, необходим механизм, позволяющий клиенту корректно запросить у сервера сессионный билет, не нарушая общих принципов безопасности и конфиденциальности TLS 1.3.

2 Концепт решения

Расширение `ticket_request`, определенное в RFC 9149, решает эту проблему, вводя минималистичный сигнальный механизм **запроса и подтверждения** возможности выдачи билета.

Концепция решения основывается на следующих принципах:

1. **Сигнал, а не данные:** Само расширение не передаёт никаких дополнительных данных. Его наличие в сообщении `ClientHello` является запросом, а в `EncryptedExtensions`

— подтверждением готовности сервера выдать билет.
2. **Согласование на этапе рукопожатия:** Запрос и подтверждение происходят во время установления соединения, что позволяет серверу принять взвешенное решение на основе текущей конфигурации и нагрузки.
3. **Отделение согласования от выдачи:** Фактическая выдача билета(ов) происходит после завершения рукопожатия в сообщениях `NewSessionTicket`. Это разделение обеспечивает безопасность: билеты выдаются только после успешной аутентификации и установления защищённого канала.
4. **Совместимость и игнорирование:** Как и любое стандартное расширение TLS, если сервер не поддерживает `ticket_request`, он просто его проигнорирует. Это обеспечивает обратную совместимость и не нарушает установление соединения.

Таким образом, расширение действует как явный сигнал от клиента и подтверждение от сервера, за которым следует действие (выдача билета) в соответствующий момент.

3 Детали решения

3.1 Поведение участников

3.1.1 Клиент (Инициатор запроса)

1. Если клиент желает получить новый сессионный билет, он включает расширение `ticket_request` с пустым полем данных в сообщение `ClientHello`.
2. Клиент **должен** быть готов принять сообщение `NewSessionTicket` в любой момент после получения `Finished` от сервера, независимо от того, было ли отправлено подтверждение в `EncryptedExtensions`. Однако, если подтверждения не было, получение билета маловероятно.
3. Клиент **может** отправить это расширение как при полном рукопожатии, так и при возобновлении сессии (используя PSK).

3.1.2 Сервер (Обработчик запроса)

1. Получив `ClientHello` с расширением `ticket_request`, сервер, поддерживающий данную функциональность, принимает решение (на основе своей политики) о том, может ли и желает ли он выдать билет.
2. Если решение положительное, сервер включает расширение `ticket_request` с пустым полем данных в сообщение `EncryptedExtensions`. Это является подтверждением клиенту.
3. **После отправки своего сообщения** `Finished` (то есть после успешного завершения рукопожатия и аутентификации), сервер отправляет клиенту одно или несколько сообщений `NewSessionTicket`. Количество билетов и момент отправки определяются внутренней политикой сервера.
4. Если сервер не поддерживает расширение или не желает выдавать билет по иным причинам, он не включает `ticket_request` в `EncryptedExtensions`. В этом случае клиенту не следует ожидать билетов. Соединение устанавливается в обычном режиме.

3.2 Взаимодействие с другими механизмами TLS 1.3

- **PSK и ранние данные (0-RTT):** Расширение `ticket_request` может использоваться совместно с `pre_shared_key` и `early_data`. Клиент, возобновляющий сессию с помощью PSK, может одновременно запросить новый билет для *последующего* подключения. Важно: билет, используемый для текущего возобновления, и новый запрашиваемый билет **должны быть криптографически независимы** (выведены из разных ключей возобновления сессии). Это предотвращает компрометацию будущих сессий.
- **Одноразовые билеты:** Сервер должен генерировать новые криптографические материалы для билетов, выдаваемых в ответ на запрос. Не допускается повторная выдача ранее выданного, ещё не истекшего билета.
- **Множественные билеты:** Сервер может отправить несколько сообщений `NewSessionTicket` в ответ на один запрос, что полезно для клиентов, планирующих несколько параллельных подключений.

3.3 Итог и значимость

Расширение `ticket_request` устраняет небольшой, но важный пробел в управлении жизненным циклом сессионных билетов TLS 1.3. Оно предоставляет:

- **Стандартизацию:** Унифицированный способ запроса билета вместо нестандартных или принудительных методов (например, разрыва соединения для получения нового билета).
- **Гибкость:** Клиенты могут активно управлять запасом билетов для обеспечения отказоустойчивости и производительности.
- **Безопасность:** Механизм не нарушает базовые принципы безопасности TLS 1.3, так как выдача билетов происходит после аутентификации и по защищённому каналу.
- **Эффективность:** Лёгкий сигнальный формат минимизирует накладные расходы.

Внедрение этого расширения особенно полезно для долгоживущих клиентов (мобильные приложения, IoT-устройства, фоновые службы), которым необходимо поддерживать возможность быстрого и безопасного переподключения к сервисам.

Источники:

1. RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3
2. RFC 9149: The `ticket_request` TLS Extension
3. IANA TLS ExtensionType Values Registry: https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml