Расширение encrypted_client_hello (ECH) в TLS 1.3

Автор: Савостей Валерий Витальевич

1 Проблема

В современном протоколе TLS 1.3 практически все данные, передаваемые между клиентом и сервером, защищены шифрованием. Однако начальный этап установления соединения, называемый «рукопожатием», всё ещё содержит уязвимые места. Самой серьёзной проблемой является передача расширения SNI (Server Name Indication).

SNI сообщает серверу, к какому конкретному доменному имени хочет подключиться клиент. Это необходимо, если на одном IP-адресе хостится множество сайтов. Поскольку этот запрос отправляется до того, как установлены ключи шифрования, имя сайта передаётся в открытом виде.

Это приводит к следующим негативным последствиям:

- **Слежка за пользователями:** Интернет-провайдеры и сторонние наблюдатели могут видеть список посещаемых ресурсов, формируя профиль интересов пользователя.
- **Избирательная блокировка:** Системы фильтрации трафика (цензура) могут автоматически разрывать соединение, если увидят в пакете запрещённый домен.
- **Утечка метаданных:** Даже если содержимое сайта скрыто, сам факт обращения к нему часто является конфиденциальной информацией.

Предыдущие попытки решить проблему (например, расширение ESNI) защищали только само имя сервера, но оставляли открытыми другие параметры запроса, что позволяло идентифицировать цель клиента по косвенным признакам.

2 Концепт решения

Расширение `encrypted_client_hello` (ECH) решает проблему радикально: вместо шифрования отдельных полей оно зашифровывает всё сообщение, содержащее параметры запроса клиента.

Концепция строится на разделении запроса на две части:

1. **Внешний запрос (ClientHelloOuter):** Это «оболочка», которая выглядит как обычное, ничем не примечательное приветствие. Она содержит общие параметры и ведет на какой-то нейтральный, разрешённый адрес (например, на общий домен крупного облачного провайдера).
2. **Внутренний запрос (ClientHelloInner):** Это настоящий запрос клиента, содержащий реальный адрес сайта и другие настройки. Именно эта часть зашифровывается и вкладывается внутрь внешнего запроса.

Для того чтобы клиент мог зашифровать внутренний запрос еще до связи с сервером, он должен заранее получить его **открытый ключ**. Этот ключ сервер публикует в системе доменных имен (DNS) в виде специальной записи. Только владелец целевого сервера, обладающий соответствующим секретным ключом, сможет выполнить **расшифрование** и узнать, куда на самом деле направляется пользователь.

3 Детали решения

Механизм работы ECH включает несколько ключевых этапов, обеспечивающих безопасность и устойчивость к ошибкам:

1. **Получение конфигурации:** Перед началом соединения клиент запрашивает через защищенный DNS-канал информацию о поддержке сервером ECH. В ответ он получает открытый ключ и список поддерживаемых **наборов алгоритмов шифрования**.
2. **Гибридное шифрование:** Клиент использует механизм HPKE. Это позволяет безопасно объединить асимметричное шифрование (для передачи секретного кода) и быстрое симметричное шифрование (для защиты самого текста запроса).
3. **Обработка на стороне сервера:**
   - Если сервер поддерживает ECH и имеет нужный ключ, он извлекает внутренний запрос и продолжает работу так, будто клиент сразу обратился напрямую.
   - Если сервер не может расшифровать данные (например, ключи устарели), он не обрывает соединение, а отвечает через «внешний» запрос и передаёт клиенту актуальные настройки для повторной попытки.
4. **Защита от подмены:** Вся процедура спроектирована так, что злоумышленник не может незаметно заставить клиента отказаться от шифрования, если оно поддерживается сервером.

4 Противодействие и цензура

Технология ECH делает интернет-цензуру на основе анализа имен сайтов практически невозможной. Это вызвало ответную реакцию в ряде стран с жестким государственным регулированием сети.

В частности, системы фильтрации трафика в **Китае** начали массово блокировать любой TLS-трафик, в котором обнаруживается использование расширения ECH. Поскольку цензоры не могут увидеть, куда именно идет пользователь, они предпочитают полностью запрещать такие соединения. Это создает технологическое противостояние: провайдеры стремятся внедрить ECH для защиты приватности, а регуляторы — ограничить его использование для сохранения контроля над информационным пространством.

5 Итог

ECH — это важный шаг в эволюции приватности в сети. Оно устраняет последнюю значимую утечку данных в протоколе TLS 1.3, превращая процесс установления соединения в «черный ящик» для стороннего наблюдателя. Несмотря на попытки блокировок, расширение становится стандартом де-факто для современных браузеров и облачных сервисов.