ЗАЩИТА ТРАФИКА РАСПРЕДЕЛЕННЫХ СИСТЕМ НА ОСНОВЕ ПРОТОКОЛА MUTUAL TLS

Крагель В. А., Ярмош А. Д., Скиба И. Г.
Отдел информационных технологий, Центр информатизации и инновационных разработок,
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: {v.kragel, a.iarmosh, i.skiba}@bsuir.by

В работе рассмотрен протокол Mutual TLS (mTLS) как средство обеспечения аутентификации и конфиденциальности трафика в распределенных системах. Проведен анализ принципов работы mTLS, его преимущества перед односторонним TLS, приведены ключевые аспекты практического внедрения в контексте парадигмы Zero Trust.

Введение

Современные информационные системы эволюционировали в сторону распределенных архитектур, таких как микросервисы и облачные сервисы. В таких системах взаимодействие между множеством независимых компонентов составляет основу их функционирования. Однако эта же особенность создает проблемы безопасности. Традиционный периметровый подход, основанный на защите границ сети, становится неэффективным, так как злоумышленник, получив доступ к одному компоненту, может беспрепятственно перемещаться по внутренней сети системы.

В связи с этим на первый план выходит парадигма Zero Trust («Никому не доверяй, всегда проверяй»), которая предполагает отсутствие неявного доверия к любым сетевым субъектам. Для ее реализации необходимы механизмы, обеспечивающие строгую аутентификацию и шифрование для каждого сетевого взаимодействия. Стандартный протокол TLS решает задачу конфиденциальности и аутентификации сервера для клиента, но не проверяет подлинность клиента перед сервером. Именно этот пробел восполняет Mutual TLS (mTLS).

В данной работе рассматривается протокол mTLS как механизм защиты трафика распределенных систем, его архитектурные преимущества и практические аспекты внедрения.

I. Принцип работы и отличия от стандартного TLS

В стандартном TLS-рукопожатии только сервер аутентифицируется перед клиентом, предъявляя свой цифровой сертификат [1]. Клиент может оставаться анонимным, например, при посещении веб-сайта, или аутентифицироваться на уровне приложения, используя логин и пароль. Это создает уязвимость: установленное TLS-соединение может быть использовано любым клиентом, в том числе и злоумышленником, если он получил доступ к сети.

mTLS добавляет в процесс рукопожатия дополнительный шаг – аутентификацию клиента [2–3]. Для этого клиент также должен обладать своим цифровым сертификатом, подписанным доверенным удостоверяющим центром (Certificate Authority, CA).

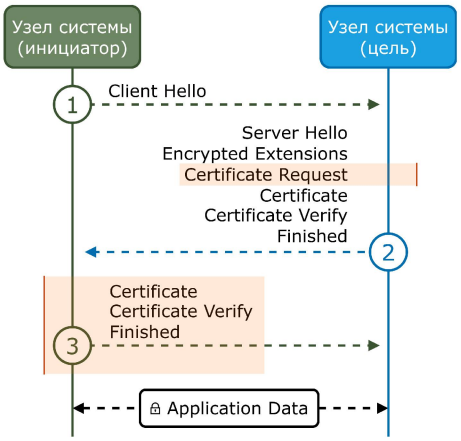

Рис. 1 – Схема процесса рукопожатия с использованием Mutual TLS

Процесс рукопожатия в TLS 1.3 с использованием mTLS осуществляется следующим образом:

1. Client Hello. Стандартное начало TLS-рукопожатия.
2. Server Hello. Наряду с базовыми сообщениями сервер отправляет Certificate Request. Это ключевое отличие от одностороннего TLS, которое указывает на требование аутентификации клиента и приводит к добавлению еще одного этапа рукопожатия.
3. В ответ на запрос сервера, клиент отправляет свой сертификат (Certificate) и сообщение Certificate Verify, которое служит криптографическим доказательством владения приватным ключом. Это обеспечивает аутентификацию клиента перед сервером.

Таким образом, mTLS обеспечивает взаимную аутентификацию (mutual authentication), гарантируя, что каждая сторона соединения знает, с кем она взаимодействует. После успешного завершения рукопожатия между аутентифицированными сторонами устанавливается защищенное соединение для обмена данными.

II. Преимущества использования mTLS в

распределенных системах

Внедрение mTLS предоставляет следующие преимущества для безопасности распределенных систем [4–5]:

- Сквозная аутентификация: каждый узел уверен в подлинности своего собеседника. Это предотвращает атаки подмены (spoofing), когда злоумышленник выдает себя за легитимный сервис, например, за сервис аутентификации или базу данных.
- Конфиденциальность трафика: как и в стандартном TLS, весь обмен данными шифруется, что защищает его от перехвата (sniffing) при передаче между узлами.
- Независимость от сетевого уровня: безопасность реализуется на транспортном уровне модели OSI. Примером преимущества такого подхода служит то, что политики безопасности остаются в силе при изменении IP-адресов сервисов, что является обычной практикой в контейнерных средах, таких как Kubernetes.
- Основа для Zero Trust Architecture: mTLS является практической реализацией принципа «проверяй каждый запрос», ликвидируя концепцию доверенной внутренней сети.

III. Ключевые аспекты внедрения

Основная сложность использования mTLS заключается не в самом протоколе, а в управлении жизненным циклом цифровых сертификатов:

- Масштабируемость: в системе из сотен узлов необходимо генерировать, распределять и обновлять сотни сертификатов. Ручное управление невозможно.
- Автоматизация: в условиях динамически изменяющейся инфраструктуры автоматическая выдача (provisioning) и ротация (rotation) сертификатов становится необходимостью.
- Безопасность приватных ключей: приватные ключи сертификатов должны надежно храниться на каждом сервисе, поскольку обладание таким ключом эквивалентно получению полного доверия со стороны зависимых сервисов, что приводит к нарушению конфиденциальности системы.

Для решения этих задач в распределенных системах применяются специализированные инструменты, которые абстрагируют сложность управления mTLS.

Инструменты, такие как Istio и Linkerd, используют архитектурный паттерн sidecar-прокси – это дополнительный контейнер, который развертывается рядом с каждым экземпляром сервиса и перехватывает весь его сетевой трафик. В сочетании с другими сетевыми функциями образуется так называемая сервисная сетка (Service Mesh). Она использует собственный корневой CA, на основе которого выдает уникальные сертификаты для каждого сервиса через sidecar-прокси. Получая входящий зашифрованный трафик, sidecar-прокси выполняет всю работу по mTLS: проверяет сертификат клиента, расшифровывает данные и передает их основному сервису в открытом виде, используя loopback-интерфейс. Аналогично, исходящий трафик от сервиса перехватывается прокси, который устанавливает mTLS-соединение с удаленным узлом. Для прокси автоматически выполняется ротация сертификатов, жизненный цикл которых поддерживается кратким. Помимо этого, интеграция сервисной сетки осуществляется без вмешательства в код приложения.

Инструменты управления секретами, наподобие HashiCorp Vault, действуют как полнофункциональный доверенный корневой CA для инфраструктуры. Приложения могут программно обращаться к API Vault для динамического запроса короткоживущих сертификатов непосредственно перед установлением соединения. Vault подписывает их, проверяя права приложения на получение сертификата, например, через роль в Kubernetes.

Заключение

Протокол Mutual TLS представляет собой стандартизированный механизм, обеспечивающий взаимную аутентификацию сторон и конфиденциальность передаваемых данных, что делает его инструментом защиты трафика в распределённых системах. Применение mTLS позволяет предотвратить несанкционированный доступ, подмену или перехват данных, формируя основу для реализации архитектур, соответствующих парадигме Zero Trust.

Несмотря на дополнительные затраты на администрирование сертификатов, mTLS обеспечивает высокий уровень доверия между сервисами. Процессы внедрения и управления решаются средствами оркестрации контейнеров и инструментами автоматизации сетевого взаимодействия, что делает использование mTLS масштабируемым и управляемым на практике.

1. Ristić, I. Bulletproof SSL and TLS / I. Ristić – London : Feisty Duck Limited, 2014. – 516 p.
2. RFC 8446 – The Transport Layer Security (TLS) Protocol Version 1.3 [Electronic resource] – Mode of access: `https://datatracker.ietf.org/doc/html/rfc8446`. – Date of access: 26.09.2025.
3. RFC 5246 – The Transport Layer Security (TLS) Protocol Version 1.2 [Electronic resource] – Mode of access: `https://datatracker.ietf.org/doc/html/rfc5246`. – Date of access: 29.09.2025.
4. Whitman, M. E. Principles of Information Security, Fifth Edition / M. E. Whitman, H. J. Mattord – Boston : Cengage Learning, 2014. – 722 p.
5. Горлов А. В., Ноженко К. Э. Анализ и методика проведения перехвата сетевого трафика // Мировая наука. – 2025. – №1. – С. 46–56.