УДК 004.728.5

О. В. Салієва
Д. О. Немировська

ПОРІВНЯЛЬНИЙ АНАЛІЗ ТРАНСПОРТНИХ ПРОТОКОЛІВ ЗАХИСТУ TLS ТА QUIC

Вінницький національний технічний університет

Анотація

Дослідження присвячено аналізу архітектурних і функціональних особливостей транспортних протоколів TLS та QUIC, а також визначенню продуктивності їхньої роботи в різних умовах застосування.

Ключові слова: TLS, TCP, QUIC, UDP, протоколи транспортного рівня, захист інформації на транспортному рівні

Abstract

The research analyzes the architectural and functional features of the TLS and QUIC transport protocols, as well as determines their performance in various application conditions.

Keywords: TLS, TCP, QUIC, UDP, transport layer protocols, information security at the transport layer.

Вступ

Безпечний обмін даними в мережі є одним із головних пріоритетів розвитку цифрових технологій сьогодення. Протоколи, які працюють на транспортному рівні, повинні гарантувати конфіденційність, цілісність і достовірність інформації, що передається через незахищені канали зв'язку. Серед найбільш популярних і широко застосовуваних рішень виділяються протоколи TLS, DTLS, а також IPsec, кожен з яких має свої унікальні особливості та переваги. Відносно новим протоколом є QUIC, що був розроблений компанією Google у 2013 році на заміну традиційному TLS. Виходячи з тенденцій розвитку мережевих технологій, було вирішено провести порівняльний аналіз QUIC та TLS.

Основна частина

Протоколи QUIC та TLS мають спільну мету – забезпечити конфіденційність, цілісність та автентифікацію даних, проте вони реалізують її різними шляхами. На рисунку 1 наведено порівняння архітектури HTTP/2 (з використанням TLS і TCP) та HTTP/3 (на базі QUIC), який демонструє інтеграцію функцій і рівнів захисту в обох підходах.

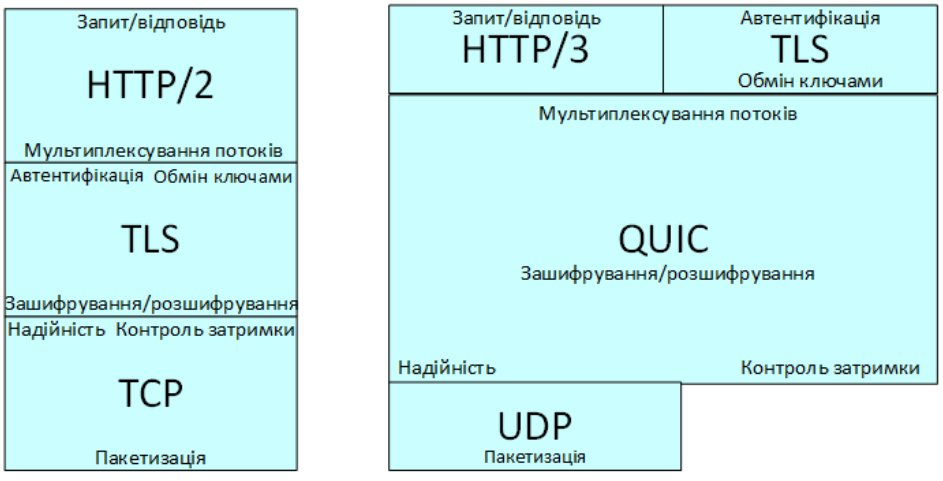

Рисунок 1 – Порівняння технологій TLS та QUIC

TLS є стандартом захисту передавання даних через мережу Інтернет, забезпечуючи конфіденційність, цілісність і автентифікацію з'єднань між клієнтом і сервером. Цей протокол

еволюціонував як продовження SSL і широко використовується для HTTPS-з'єднань. Коли клієнт встановлює з'єднання з сервером через TLS, відбувається процес рукостискання, під час якого сервер надає клієнту свій цифровий сертифікат. Цей сертифікат виданий довіреним сертифікаційним центром (CA) і містить публічний ключ сервера. Клієнт перевіряє дійсність сертифіката, переконуючись у тому, що він підписаний довіреним CA, не протермінований і відповідає доменному імені сервера. Це гарантує, що клієнт спілкується саме з автентичним сервером, а не зі зловмисником, який намагається перехопити або підробити з'єднання. Після успішної перевірки сертифіката клієнт і сервер узгоджують набір криптографічних алгоритмів і генерують спільний секретний ключ за допомогою протоколу обміну ключами Diffie-Hellman (або його варіації). Цей ключ використовується для симетричного шифрування подальшого обміну даними, забезпечуючи конфіденційність і цілісність інформації. Сильні алгоритми шифрування AES з великим розміром ключа і надійні хеш-функції гарантують, що навіть якщо зловмисник перехопить зашифрований трафік, він не зможе його розшифрувати без відповідного ключа. Проте багатоступеневий підхід TLS додає затримок, особливо в умовах нестабільного з'єднання [1].

QUIC, навпаки, розроблений для зниження затримок і забезпечення гнучкості. Цей протокол інтегрує функції TLS 1.3 у свою архітектуру, поєднуючи встановлення з'єднання з процесом шифрування [2]. Завдяки цьому QUIC дозволяє уникати затримок, характерних для TCP. Важливою особливістю є функція 0-RTT (Zero Round Trip Time), яка дає змогу надсилати дані одразу при повторних підключеннях, оскільки під час першого з'єднання між клієнтом і сервером відбувається повне рукостискання, вони обмінюються необхідними криптографічними параметрами та встановлюють спільний секретний ключ для шифрування даних. Сервер може надати клієнту спеціальний "сесійний квиток" або попередні параметри, які клієнт зберігає для майбутніх підключень. Коли клієнт знову підключається до того ж сервера, він використовує збережені попередні параметри для негайного встановлення захищеного з'єднання. Завдяки цьому клієнт може одразу почати надсилати зашифровані дані, не очікуючи на відповідь від сервера для завершення рукостискання. Такий механізм значно зменшує затримки, пов'язані з встановленням з'єднання. Однак при впровадженні QUIC можуть виникнути проблеми сумісності з існуючою мережею та інфраструктурою безпеки. Оскільки QUIC працює поверх протоколу UDP, він може зіткнутися з проблемами через фаєрволи, маршрутизатори та пристрої NAT, які зазвичай налаштовані на блокування або обмеження UDP-трафіку. Не зважаючи на це, Google, YouTube та інші великі сервіси активно впроваджують QUIC, що свідчить про його зростаючу популярність.

QUIC також використовує унікальні ідентифікатори з'єднань – Connection ID, які забезпечують збереження сеансу навіть у разі зміни IP-адреси або порту клієнта, що є особливо корисним для мобільних пристроїв [3]. Мультиплексування потоків у QUIC усуває проблему блокування на початковій стадії обробки, що покращує продуктивність при передаванні кількох потоків даних [4]. У традиційному протоколі TCP, який використовується разом з TLS, усі дані передаються через одне послідовне з'єднання. Якщо пакет втрачається або затримується, весь потік даних зупиняється, доки втрачені пакети не будуть повторно передані та отримані. Це означає, що один втраченій або затриманий пакет може затримати всю передачу даних, навіть якщо інші пакети готові до відправки. Натомість QUIC реалізує мультиплексування кількох незалежних потоків даних всередині одного з'єднання. Кожен потік має свій власний ідентифікатор і управляється окремо від інших. Якщо пакет в одному потоці втрачається, це не впливає на інші потоки: вони можуть продовжувати передаватися без перешкод. Втрачені пакети повторно передаються лише для відповідного потоку, що мінімізує затримки.

Зазначимо, що протокол тLS забезпечує захист від атак "людина посередині" завдяки перевірці сертифікатів та сильним алгоритмам шифрування. Проте його TCP-архітектура може бути вразливою до атак типу флуду та на встановлення з'єднання. У TCP процес встановлення з'єднання відбувається через тристороннє рукостискання: клієнт надсилає серверу SYN-пакет, сервер відповідає SYN-ACK, і клієнт завершує встановлення з'єднання, надсилаючи ACK. Зловмисник може скористатися цим механізмом для проведення SYN-флуд атаки, надсилаючи велику кількість SYN-пакетів з підроблених або неіснуючих IP-адрес. Сервер, у свою чергу, відповідає на кожен запит і виділяє ресурси для очікування підтвердження від клієнта, яке ніколи не надходить. Це може призвести до виснаження ресурсів сервера, оскільки він зберігає стан для кожного напіввідкритого з'єднання, і, як наслідок, до відмови в обслуговуванні легітимних користувачів. Базуючись на UDP, QUIC знижує ризик таких атак,

оскільки встановлення з'єднання відбувається швидше і менш витратно для серверів. Крім того, інтеграція TLS 1.3 у QUIC забезпечує сучасний рівень захисту даних. Водночас функція 0-RTT може створювати потенційні ризики, зокрема атаки повторного відтворення, якщо попередні параметри з'єднання не були належним чином захищені.

Для більш наочного розуміння відмінностей між протоколами TLS і QUIC представлено таблицю 1, в якій підсумовуються ключові аспекти кожного протоколу, включаючи їхні технічні характеристики, переваги, недоліки та сфери застосування.

Таблиця 1 – Порівняння протоколів транспортного рівня TLS і QUIC

Аспект	TLS	QUIC
Рівень протоколу	Працює на транспортному рівні поверх TCP.	Інтегрує транспортний і протокол безпеки поверх UDP.
Транспортний протокол	TCP	UDP
Рукостискання	Багатокроковий процес, який вимагає кількох обмінів повідомленнями.	Поєднує встановлення з'єднання та шифрування, що зменшує затримки.
0-RTT (Zero Round Trip Time)	Підтримується в TLS 1.3, але з обмеженнями.	Дозволяє надсилати дані одразу при повторних підключеннях, значно прискорюючи взаємодію.
Мультиплексування потоків	Не підтримується на рівні TCP; може виникати блокування на рівні голови черги.	Підтримує мультиплексування кількох потоків даних, усуваючи блокування та покращуючи продуктивність.
Захист від атак	Забезпечується через перевірку сертифікатів та сильні алгоритми шифрування.	Інтегрує захист TLS 1.3
Вразливості	TCP-архітектура вразлива до атак типу флуду та на встановлення з'єднання (SYN-флуд).	Менш вразливий завдяки використанню UDP і швидкому встановленню з'єднання.
Сумісність з інфраструктурою	Широко підтримується існуючими мережевими пристроями та фаєрволами.	Може стикатися з проблемами через фаєрволи або NAT, які блокують UDP-трафік; вимагає оновлення мережевих політик.
Балансування навантаження	Підтримується традиційними методами на основі відкритих заголовків TCP.	Ускладнене через шифрування заголовків і використання Connection ID. Потребує спеціалізованих балансувальників.
Мобільність з'єднань	Зміна IP-адреси призводить до розриву з'єднання.	Підтримує зміну IP-адреси або порту без втрати з'єднання завдяки використанню Connection ID.
Продуктивність	Може мати затримки через багатоступеневий процес рукостискання та блокування голови черги.	Вища продуктивність завдяки швидкому встановленню з'єднання, 0-RTT та усуненню блокування початкової стадії обробки.
Адаптивність до мережевих умов	Менш ефективний при високій затримці або втраті пакетів.	Краще адаптується до умов мережі завдяки вбудованим механізмам корекції помилок та контролю перевантаження.
Впровадження та прийняття	Стандарт для захисту більшості веб-сайтів; широко використовується для HTTPS-з'єднань.	Активно впроваджується великими сервісами (Google, YouTube, Facebook); стандартизований IETF у 2021 році.

Таким чином, QUIC значно перевершує TLS у продуктивності завдяки використанню UDP і підтримці функцій швидкого підключення. Його гнучкість робить його ідеальним вибором для ситуацій, де важливі низька затримка та висока швидкість, наприклад, потокове відео чи онлайн-ігри [5]. TLS, зі свого боку, залишається ключовим стандартом для захисту більшості веб-сайтів, забезпечуючи стабільність і надійність.

Висновок

У підсумку, TLS і QUIC вирішують схожі завдання, але кожен має свої унікальні особливості. TLS є перевіреним і надійним вибором для традиційних веб-додатків, тоді як QUIC пропонує новаторські рішення для мобільних та швидкісних мережевих сценаріїв. Обидва протоколи

використовують сучасні методи шифрування для забезпечення конфіденційності та захисту даних, але QUIC відзначається більшою швидкістю і адаптивністю. З огляду на зростаючі вимоги до швидкості та ефективності мережевих з'єднань, QUIC має потенціал стати новим стандартом для безпечної передачі даних через мережу Інтернет. Проте, його широке впровадження залежить від сумісності з існуючою інфраструктурою та подальшої стандартизації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Учасники проектів Вікімедіа. Transport Layer Security – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Transport_Layer_Security (дата звернення: 21.11.2024).
2. Учасники проектів Вікімедіа. QUIC – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/QUIC (дата звернення: 21.11.2024).
3. What Is QUIC and Why Threat Hunters Care - NetQuest. *NetQuest*. URL: https://netquestcorp.com/what-is-quic/ (дата звернення: 21.11.2024).
4. Codavel. QUIC vs TCP+TLS – and why QUIC is not the next big thing. *Medium*. URL: https://medium.com/codavel-blog/quic-vs-tcp-tls-and-why-quic-is-not-the-next-big-thing-d4ef59143efd (дата звернення: 22.11.2024).
5. Lekkala S. TLS vs DTLS vs QUIC: Navigating the Protocols of Secure Data Transmission. *Medium*. URL: https://medium.com/@seshalekkala227/tls-vs-dtls-vs-quic-navigating-the-protocols-of-secure-data-transmission-1081eb80da2f (дата звернення: 22.11.2024).

Салієва Ольга Володимирівна – доктор філософії (PhD) за спеціальністю 125 «Кібербезпека», доцент кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет, Вінниця, e-mail: salieva8257@gmail.com

Saliieva Olha Volodymyrivna – Doctor of Philosophy (PhD) in specialty 125 «Cyber Security», Associate Professor of the Department of Information Systems Management and Security, Vinnytsia National Technical University, Vinnytsia, email: salieva8257@gmail.com

Немировська Дар'я Олександрівна – студентка групи 1БКС-22б, Факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, Україна, e-mail: nemyrovskadaria@gmail.com

Nemyrovska Daria Oleksandrivna - student of group 1BKS-22b, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, Ukraine, e-mail: nemyrovskadaria@gmail.com