Інформаційні технології

УДК 004.72

Шебеда Нікіта Станіславович
студент
Державного університету телекомунікацій
Шебеда Никита Станиславович
студент
Государственного университета телекоммуникаций
Shebeda Nikita
Student of the
State University of Telecommunications

Берлим Олександр Віталійович
Управління Державної служби спеціального зв'язку та захисту інформації України в Житомирській області
Берлим Александр Виталевич
Управление Государственной службы специальной связи и защиты информации Украины в Житомирской области
Berlym Oleksandr
Directorate of the State Service for Special Communication and Information Protection of Ukraine in the Zhytomyr Region

ПРИЧИНИ НИЗЬКОЇ ПОПУЛЯРНОСТІ HTTP/2. СУЧАСНИЙ СТАН ТА ПЕРЕВАГИ

Анотація. *Розглянуто фактори які спричинили низький відсоток використання протоколу HTTP/2 в Інтернет, а також наведено причини вигідності використання протоколу нового покоління в порівнянні з існуючими гіпертекстовими протоколами. З розвитком стільникових мереж проблема набирає все більшої актуальності, оскільки провайдери*

веб-ресурсів, не використовуючи розвинутий протокол гіпертекстових повідомлень, несуть втрати, як з економічної точки зору так і з точки зору безпеки. Ми здійснили порівняння практичної ефективності веб-протоколів на основі існуючих електронних ресурсів.

Ключові слова: *комп'ютерні технології, гіпертекст, протокол, обмін даними, Інтернет, сервіси.*

Аннотация. *Рассмотрены факторы, которые вызвали низкий процент использования протокола HTTP / 2 в Интернет, а также приведены причины выгодности использования протокола нового поколения по сравнению с существующими гипертекстовыми протоколами. С развитием сотовых сетей проблема приобретает все большую актуальность, поскольку провайдеры веб-ресурсов, не используя продвинутый протокол гипертекстовых сообщений, несут потери, как с экономической точки зрения, так и с точки зрения безопасности. Мы осуществили сравнение практической эффективности веб-протоколов на основе существующих электронных ресурсов.*

Ключевые слова: *компьютерные технологии, гипертекст, протокол, обмен данными, кибербезопасность, Интернет, сервисы.*

Summary. *The factors that caused a low percentage of HTTP / 2 use on the Internet are considered, as well as the reasons for the benefits of using the new generation protocol in comparison with existing hypertext protocols. With the development of cellular networks, the problem is becoming more and more urgent, as web resource providers, without using the advanced protocol of hypertext messages, are losing both economically and in terms of security. We implemented a comparison of practical effectiveness of web protocols based on existing electronic resources.*

Key words: *computer technologies, hypertext, protocol, data exchange, Internet, services.*

Актуальність роботи. Одним із головних аспектів в сучасних мережах обміну даними, що надають сервіси кінцевому користувачу, беззаперечно є швидкість. Особливо це помітно в наш час стрімкого розвитку бездротових технологій, оскільки чим більша швидкість, тим більше послуг здатна надати мережа. Але в переважній більшості, кінцевий користувач не використовує всі надані йому мережевим провайдером потужності у зв'язку з багатьма факторами, такими як відстань від станції зв'язку, недостатнє покриття, несприятливі погодні умови, щільність забудови населеного пункту, відсутність необхідності повного використання наданих можливостей, тощо. Історично склалося, що велика частина, корисної для кінцевого споживача, інформації знаходиться в текстовому та графічному вигляді і якщо користувач, надіславши запит в пошукову систему, отримав три посилання на абсолютно однакові електронні сторінки різних власників з необхідною для нього інформацією, який зробить вибір? Цілком логічно обрати перший в списку, але в реаліях сучасного світу, щоб мати більшу аудиторію не обов'язково бути першим, а необхідно швидше подати бажане. З іншого боку навіть будучи першим і швидшим, аудиторія може не довірити вам свої дані, тому що ваш продукт не забезпечує належного захисту.

Мета даної статті – мотивація розробників веб-додатків на використання протоколу HTTP/2.

1. Чим спричинена низька частка використання HTTP/2

1.1. Велика кількість існуючих веб-застосунків

HTTP – протокол передачі гіпертексту розроблений британським вченим Тімом Бернерсом-Лі в 1991 році.

HTTP/1.1 був впроваджений в 1999 році і добре служив понад 19 років, але його вік починає даватися в знаки. Завантаження веб-сторінок стає все більш ресурсоємним, ніж будь-коли і все важче стає

завантажувати елементи, оскільки цей протокол практично дозволяє виконувати лише один запит на TCP-з'єднання.

В минулому браузери використовували декілька з'єднань TCP для здійснення паралельних запитів. Однак якщо використовується занадто багато з'єднань, це становиться контрпродуктивно (функція контролю перевантаження TCP-сесій починає відхиляти з'єднання), фактично браузер відбирає більшу частину мережевих ресурсів, ніж йому необхідно в рамках сесії з деяким відсотком втрат.

Також протокол HTTP/1.1 не забезпечує необхідного захисту даним що отримує або відправляється клієнтом. Для вирішення питання безпеки в 1994 році вперше було випущене розширення HTTPS яке забезпечує шифрування SSL, усунувши проблему з безпекою та спричинивши збільшення часу завантаження сторінки. Згодом на зміну SSL прийшов TLS.

Хоча поява третього і четвертого покоління мереж стільникового зв'язку дещо згладила ситуацію, ці події не можна було залишати без уваги. В травні 2015 року на світ з'явилася специфікація HTTP/2.

Свого часу, у вересні 2014 року був оприлюднений аналітичний звіт компанії Netcraft, в якому було зазначено, що кількість персональних блогів та сайтів що працюють в Інтернет становить 1 022 954 603 вузли, що показало приріст майже на 31 мільйон веб-ресурсів всього лише за один місяць [1].

Починаючи з 1999 і по сьогодення безліч розробників для подолання затримки і проблем конвеєрної передачі, також не сиділи без діла, використовуючи наступні методи оптимізації [2]:

– Шардінг (Sharding). Розміщення файлів на різних доменах для паралельної передачі браузеру; мережі доставки контенту (CDNs) роблять це автоматично. Така оптимізація може пошкодити

продуктивності HTTP/2. Ви можете використовувати дружній з HTTP/2 шардінг для користувачів HTTP/1.1;

– Використання спрайтів (Sprite). Спрайтами називають колекції картинок, які передаються у вигляді одного файлу; після цього на стороні клієнта картинки по необхідності витягуються з колекції. Ця оптимізація менш ефективна при використанні HTTP / 2, хоча все одно може бути корисна;
– Об'єднання файлів (Concatenating). Подібно спрайтам, частина файлів, які зазвичай зберігаються окремо, об'єднуються в один. Після чого браузер знаходить і запускає код в міру необхідності в рамках «склеєного» файлу;
– Вбудовування файлів (File Inline). CSS, JavaScript і навіть зображення вставляються безпосередньо в HTML-файл, що зменшує кількість переданих файлів, за рахунок збільшення вихідного HTML-файлу.

Дані прийоми не покращували швидкість відображення контенту в браузері, а навпаки погіршували, оскільки HTTP/2 опирається на передачу більшої кількості запитів і відповідей за рахунок меншої кількості TCP-сесій.

Таким чином, піонери в розробці веб-застосунків, тестуючи новий протокол використовували вже набутий досвід, що, як зазначено вище, дало не той результат що очікувався. Це спричинило вал критики в бік нового протоколу. Також доволі холодно віднеслися й власники існуючих онлайн-ресурсів, оскільки для використання нової технології необхідно було або створювати ще один сайт з використанням HTTP/2, або повністю переробляти вже наявний. Обидва виходи були і є економічно невигідними, тому що в першому випадку необхідно збільшувати плату за хостинг, в іншому погіршувати якість надання поточних послуг частковою або повною зупинкою певних функцій на момент переходу сервера на оновлений протокол.

Станом на травень 2018 року ситуація дещо покращилася, і частка від загальної кількості сайтів в Інтернет що підтримують HTTP/2 становить 25.9% [3]. Та технологія у популярності використання все-таки поступається попереднім версіям протоколу.

1.2. Вразливості серверної реалізації в 2016 році

На конференції Black Hat USA 2016 дослідники безпеки з компанії Imperva представили звіт з результатами аналізу безпеки протоколу HTTP/2 [4]. В процесі аналізу було виявлено чотири концептуальні уразливості, які проявляються в різних реалізаціях серверної частини HTTP/2 і призводять до можливості проведення DoS-атаки (відмова в обслуговуванні). Серед виявлених виявилися дві вразливості які актуальні для версії HTTP/1.1

Одна з вразливостей (CVE-2016-1546) дозволяє зловмисникові використовувати шкідливий клієнт для дуже повільного читання відповідей, тим самим провокуючи відмову в обслуговуванні. Цей метод названий Slow Read аналогічний відомій DDoS-атаці Slowloris, при якій клієнт приймає відповідь на запит дуже повільно, що дозволяє вичерпати ліміт на число активних з'єднань. Атаці піддавалися реалізації HTTP/2 в Apache, IIS, Jetty, NGINX і nghttp2. Розробники nginx повідомили, що проблема була усунена в лютому з випуском версії NGINX 1.9.12, актуальні випуски NGINX до вразливості не схильні; Атакуючий здатний використовувати налаштування HTTP/2 для мультиплексування великої кількості каналів через одне TCP-з'єднання. Незважаючи на те, що сервер підтримує одне TCP-з'єднання, він виділяє потік для кожного каналу, що може привести до споживання всіх доступних потоків на вразливому сервері [4; 5].

Вразливість HPACK Bomb включає дві дірки (CVE-2016-1544, CVE-2016-2525), що дозволяють атакуючому створити невелике з вигляду повідомлення, яке насправді розпаковує гігабайти даних на стороні

сервера. Зловмисник може спожити всі доступні ресурси в системі і викликати відмову в обслуговуванні [4; 5].

Dependency Cycle Attack (CVE-2015-8659) впливає на механізми управління потоком, що використовуються в HTTP/2 для оптимізації мережі. За допомогою спеціально сформованих запитів атакуючий може викликати зациклення додатків. Експлуатація вразливості дозволяє здійснити DoS-атаку або виконати довільний код на системі [4; 5].

Вразливість Stream Multiplexing Abuse (CVE-2016-0150) пов'язана з мультиплексуванням потоку, коли кілька сесій проходять через одне HTTP/2-з'єднання. Так як поділ зв'язку є виключно логічним, атакуючий може скористатися цим для маніпулювання сервером [4; 5].

В реалізаціях 2018 року дані вразливості в серверній частині відповідних виробників були вирішені. Якщо версія вашої серверного додатку піддатлива цим вразливостям, нагально рекомендуємо оновитися до останньої версії.

1.3. Особливості впровадження

Перш за все, перед впровадженням будь-якої технології необхідно визначити доцільність її використання. Просто впровадити HTTP/2, при наявності необхідних знань та вмінь, не зіставляє великої праці. Однак варто розуміти, що в деяких випадках оновлений протокол не є універсальним рішенням, тому що виникає потреба впроваджувати SSL. Тобто для певних застосунків він може виявитися більш продуктивним, а для деяких, навпаки.

Наприклад, сайту, з використанням SSL/TLS, технологія пришвидшить відображення контенту. В той ж самий час ресурсу де не застосовується шифрування, необхідно буде впровадити SSL/TLS, що збільшить навантаження на обладнання і фактичного виграшу від впровадження нового гіпертекстового протоколу помітно не буде.

Можна зауважити, що навіть в гіршому випадку з'явиться безпека даних. Але чи дійсно вона потрібна на тому чи іншому ресурсі? Користувач може надавати дані які, як з його точки зору так і з точки зору розробника та закону, не мають сенсу бути захищеними. Наприклад зображення та відеоматеріали пейзажів, кошенят, цуценят тощо.

Не варто також відкидати варіант, що розробник інтернет-ресурсу вже використовує інший механізм захисту інформації. В такому випадку, впровадження SSL принесе збитковість і погіршення продуктивності роботи обладнання.

Таким чином можна виділити 5 основних недоліків з якими існує шанс зіткнутися під час впровадження HTTP/2:

– Великі витрати для одного з'єднання. Алгоритм стиснення даних HPACK вимагає підтримки таблиці перетворення на обох кінцях. Також для одного з'єднання потрібно більше пам'яті;
– Можливе використання TLS надлишкове. Якщо передана інформація не потребує захисту або вже захищена за допомогою DRM (або іншого шифрування), то в цьому випадку TLS навряд чи буде корисний;
– Пошук і видалення існуючих HTTP/1.1 оптимізацій, э необхідним для збільшення продуктивності HTTP/2, що є додатковою роботою;
– Відсутність переваг при завантаженні великих файлів. Якщо веб-додаток в основному розрахований на завантаження великих файлів або відеострімінг, то, швидше за все, використання TLS буде помилковим, а мультиплексування не принесе ніякої користі;
– Можливо відвідувачам не важливо, що дані, якими вони діляться на вашому сайті, не захищене TLS і HTTP/2.

2. Необхідність впровадження

2.1. Турбота про захист даних клієнтів

Зважаючи на всі наведені мінуси нового протоколу, здається що здійснити «правильний» перехід доволі складно, та й, здається, не завжди

потрібна безпека для даних користувача. Але в реаліях сучасності безпека інформації є одним із критичних аспектів розвитку інформаційних технологій.

Вільний доступ до спеціальних інформаційних ресурсів в купі з жагою до швидкого збагачення, а також бажання самоствердитися, призвели до зростаючої кількості так званих black-hat хакерів. В свою чергу починаючи власний шлях до вершин майстерності цифрового взлому, їм необхідний полігон для тренувань. Зазвичай обираються незахищені сайти, а також користувачі, які не мають базового розуміння про інформаційну безпеку.

Якщо надати рекомендації і поміч в захисті окремо взятого ресурсу фізично можливо, то надати необхідну технічну підтримку кожному користувачу не представляється можливим. В такому випадку, з гуманістичних зображень, відповідальність за мінімальну безпеку даних клієнта лягає на плечі власника сторінки.

Також таким чином можна приволікти до своїх джерел користувачів які турбуються про безпеку своїх даних. HTTP/2 забезпечує опортуністичний TLS, що забезпечує захист від пасивного прослуховування трафіку в мережі.

2.2. Переваги використання HTTP/2

Використання нового гіпертекстового протоколу надає наступні переваги:

– Використовується тільки одне з'єднання з сервером замість безлічі з'єднань, які передають по одному файлу. Іншими словами, зменшується кількість з'єднань, що особливо корисно при використанні TLS;
– Ефективне використання TLS. HTTP/2 робить тільки один TLS хендшейк, а мультиплексування дозволяє ефективно використовувати це з'єднання. HTTP/2 також стискає дані заголовка, а усунення HTTP/1.1

оптимізацій (таких як конкатенація файлів) дозволяє алгоритму кешування працювати більш ефективно;

- Спрощення веб-додатків. При використанні HTTP/2 можна позбутися від HTTP/1.1 оптимізацій, що спростить працю розробникам.
- Дуже добре підходить для складних веб-сторінок. HTTP/2 відмінно підходить для веб-сторінок, які одночасно використовують HTML, CSS, JavaScript, зображення і відео. Браузери можуть пріоритезувати запити до файлів, щоб найбільш необхідні частини сторінки надсилались в першу чергу;
- Безпека з'єднання. Хоча при використанні HTTP/2 може відбутися втрата продуктивності через використання TLS, але в той же час TLS зробить веб-додатки більш безпечними для користувачів.

На конференції nginx.conf 2015 [6] були опубліковані результати випробувань HTTP/2 з типовим середнім часом прийому-передачі round-trip time (RTT):

- Дуже низький RTT (0-20 мілісекунд (далі мс)): практично ніякої різниці між HTTP/1.1, HTTP/2, і HTTPS не спостерігається.
- Середній (типовий для інтернету) RTT (30-250 мс): HTTP/2 швидше ніж HTTP/1.1, і обидва швидше ніж HTTPS. Для сусідніх міст в США, RTT становить близько 30 мс, і близько 70 мс від одного берега до іншого (близько 3000 миль). По одному з найкоротших маршрутів між Токіо і Лондоном, RTT становить близько 240 мс.
- Високий RTT (300 мс і вище): HTTP/1.1 швидше ніж HTTP/2, який швидше ніж HTTPS.

Висновок. Низька популярність нового гіпертекстового протоколу була спричинена не просто так. В основі цього лягли наступні пункти:

- Велика кількість вже існуючих веб додатків на основі HTTP/1.1.

- Відсутність інформації, як «безболісно» здійснити перехід на HTTP/2 без втрати трафіку, одразу після виходу специфікації;
- Обов'язковість впровадження SSL/TLS, що в сумі з оптимізаціями HTTP/1.1 спричиняло зниження продуктивності;
- Відсутність в деяких інтернет-ресурсів необхідності здійснення переходу по тій чи іншій причині;
- Виявлені критичні вразливості серверної частини протоколу;
- Недовіра до нового;
- Хвиля негативної критики в сторону протоколу та його специфікації, у зв'язку з вищенаведеними причинами.

Наразі в Інтернет достатньо електронних ресурсів, котрі описують процедури та особливості переходу на HTTP/2 з мінімальною втратою трафіку, а розробники нових веб-додатків все частіше застосовують цю технологію і отримують більшу економічну вигоду аніж конкуренти на старіших протоколах. Впровадження обов'язкового SSL/TLS в сумі з мультиплексуванням HTTP/2 компенсують один одного, що не призвело до сильного збільшення навантаження на апаратну частину сервера, і забезпечило додаткову безпеку даним клієнта. Також були усунені критичні вразливості серверної частини HTTP/2, в той час як в HTTP/1.1 вони залишаються актуальні до сих пір. Отож, в світлі нових подій перехід на нову технологію web-додатків є нагальною потребою, як з точки зору безпеки так і економічної вигоди.

Література

1. Количество сайтов в интернете перевалило за миллиард [Електронний ресурс] / Techno.bigmir.net. – 2014. – Режим доступу до ресурсу: http://techno.bigmir.net/technology/1577735-Kolichestvo-sajtov-v-internete-perevalilo-za-milliard.

2. Usage of HTTP/2 for websites [Електронний ресурс] / W3Techs. – 2018. – Режим доступу до ресурсу: https://w3techs.com/technologies/details/ce-http2/all/all
3. Bartenev V. V. 7 Tips for Faster HTTP/2 Performance [Електронний ресурс] / Valentin Bartenev / NGINX, Inc. – 2015. – Режим доступу до ресурсу: https://www.nginx.com/blog/7-tips-for-faster-http2-performance/
4. HTTP/2: In-depth analysis of the top four flaws of the next generation web protocol [Електронний ресурс] / Imperva. – 2016. – Режим доступу до ресурсу: https://www.imperva.com/docs/Imperva_HII_HTTP2.pdf
5. В HTTP/2 обнаружены пять опасных уязвимостей.[Електронний ресурс] / SecurityLab.ru. – 2016. – Режим доступу до ресурсу: https://www.securitylab.ru/news/483279.php
6. Bartenev V. V. The HTTP/2 Module in NGINX [Електронний ресурс] / Valentin Bartenev / NGINX, Inc. – 2015. – Режим доступу до ресурсу: https://www.youtube.com/watch?v=4OiyssTW4BA