Engineer

HSTS

обіцянка браузера наступного разу використати HTTPS

Що це таке

HSTS, або HTTP Strict Transport Security, — це заголовок відповіді, яким сайт велить браузеру протягом заданої кількості секунд звертатися до нього лише через HTTPS і повністю обривати з'єднання, якщо з сертифікатом щось не так. Сам заголовок називається `Strict-Transport-Security`, а визначає його RFC 6797 від листопада 2012 року. Шифрування HSTS не додає; він змінює те, що браузер погоджується робити через HTTPS.

Типове значення містить дві директиви:

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

`max-age` — це час, протягом якого браузер пам'ятає правило, тут один рік. `includeSubDomains` поширює правило на всі імена під хостом, тож `api.example.com` і `old.example.com` успадковують те, що попросив `example.com`. У багатьох конфігураціях трапляється й третє слово, `preload`. Воно походить не з RFC, а зі списку preload на hstspreload.org і означає згоду сайту на те, щоб його ім'я вбудували в браузери.

Отримавши заголовок у захищеній відповіді, браузер до кінця строку робить для цього хоста три речі. Він переписує будь-яку адресу `http://` на `https://` ще до того, як запит покине комп'ютер. Він вважає будь-яку помилку сертифіката остаточною і не показує кнопки «все одно продовжити». І він починає відлік заново щоразу, коли заголовок приходить знову. Відповідь із `max-age=0` змушує його забути хост.

Найчастіше HSTS порівнюють із переспрямуванням з порту 80 на порт 443. Переспрямування теж працює, але лише після того, як браузер уже надіслав перший запит відкритим текстом, а будь-хто в тій самій мережі може сам відповісти на цей запит і втримати відвідувача на нешифрованій копії сайту. HSTS прибирає цей відкритий запит з усіх візитів після першого. Сусід з іншого боку — Upgrade Insecure Requests, директива Content Security Policy. Вона переводить на HTTPS запити, які сторінка сама надсилає по скрипти й зображення, але відвідувач, що прийшов за посиланням `http://` з чужого сайту, усе ще залежить від переспрямування.

Застосуйте поняття до рішення

Застосуйте «Що таке HSTS?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Облікові записи, паролі й права налаштовані так, щоб доступ відповідав роботі, а звільнені втрачали його того ж дня.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Написала 1 000 правил RBAC для Linux-серверів — 1,000 правил доступу написано, щоб усунути вразливість

Написала 1 000 правил RBAC для Linux-серверів Зв'язатися з нами →

Чому це важливо

Увімкнення HSTS коштує одного рядка конфігурації. Справжнє рішення — як далеко зайти, бо кожен крок захищає більше відвідувачів і його важче скасувати.

Сам заголовок лишає прогалину, яку називає сам RFC 6797. Браузер, що ще ніколи не бачив сайту або в якого збережена політика вже спливла, усе одно починає з відкритого запиту. Довгий `max-age` звужує другу частину цієї прогалини, тому рік і став звичним значенням. Першу частину закриває лише список preload: імена хостів, вкомпільовані в Chrome, на основі якого Firefox, Safari й Edge ведуть власні списки, тож щойно встановлений браузер уже знає, що треба йти через HTTPS.

Вимоги списку суворі. Базовий домен мусить мати чинний сертифікат, переспрямовувати з HTTP на HTTPS на тому самому хості, обслуговувати всі піддомени через HTTPS і надсилати заголовок із `max-age` щонайменше `31536000` секунд, `includeSubDomains` і `preload`. Супровідники списку прямо пишуть, що це стосується й внутрішніх піддоменів, яких публіка ніколи не

бачить. Вирішує все вихід: вилучення доходить до користувачів через оновлення Chrome за кілька місяців, для інших браузерів гарантій немає, а сам сайт називає цей процес «повільним і болісним».

Тож рішення — це драбина, якою підіймаються по одному щаблю:

- **Заголовок без** `includeSubDomains` годиться майже для будь-якого сайту, що вже працює на HTTPS. RFC 9325, чинна найкраща практика для TLS, каже, що вебсерверам слід його надсилати.
- `includeSubDomains` безпечний, коли хтось склав перелік усіх імен під доменом і перевірив, що кожне з них працює через HTTPS, зокрема ті, що їх веде постачальник або інший відділ.
- **Preload** — для домену, власники якого певні, що відкритий HTTP під ним більше не знадобиться ніколи: домен входу, платіжний домен, компанія, що сама керує всім своїм DNS. Маркетинговий домен, де інструмент кампаній невдовзі може попросити CNAME, — поганий кандидат, як і будь-який домен, спільний з офісною технікою.

Супровідники задають і темп: `max-age` у 300 секунд, потім тиждень, потім місяць, щоразу з `includeSubDomains`, і перед наступним щаблем треба вичекати повний строк попереднього. Браузер зберігає останнє бачене значення, тож помилка на п’ятихвилинному щаблі коштує п’ять хвилин.

Як це виглядає на практиці

На платформі з професійними та організаційними даними компанія виконувала роль інженерки з безпеки й посилила захист на кожному рівні; HSTS був одним із них. У проксі-мідлварі Next.js, `proxy.ts`, відповідь отримує HSTS і в тій самій відповіді приходять Content Security Policy зі `strict-dynamic` та nonce, що змінюється з кожним запитом, і атрибут SameSite на cookie. Проксі-мідлвар у `proxy.ts` працює перед маршрутами, які охоплює, тож правила для браузера зібрані в одному файлі, а не розкидані по сторінках.

Та сама робота показує, де HSTS закінчується. Виходили з того, що клієнт ворожий: будь-який захист, який забезпечує браузер, його власник може просто вимкнути. HSTS захищає чесного відвідувача від ворожої мережі. Проти ворожого клієнта він безсилий, тому робота пішла глибше. API отримав обмеження частоти запитів, CORS, обмеження розміру запитів, валідацію вхідних даних ще до того, як щось потрапить до бази, і журналювання подій, важливих для безпеки. До бази даних API підключався

від імені ролі, яка могла лише виконувати функції застосунку, а права доступу повторно перевірялися на сервері під час кожного запиту; обмеження на фронтенді лишилися тільки зручністю для користувача. У цьому стеку HSTS тримає сам трафік поза досяжністю, і це найдешевший його шар.

Де виникають проблеми

Заголовок прописано в конфігурації, а браузери поводяться так, ніби його немає. Його надсилають у відкритій HTTP-відповіді, зазвичай у переспрямуванні, або в HTTPS-відповіді, сертифікат якої браузер відхилив. RFC 6797 велить браузерам ігнорувати його в обох випадках, а серверам — узагалі не надсилати його відкритим HTTP. Споріднена помилка: `example.com` одразу переспрямовує на `https://www.example.com` , а заголовок є лише на `www` , тож базовий домен та інші його імена лишаються поза політикою. Додайте заголовок до всіх HTTPS-відповідей, зокрема до переспрямувань, і перевірте обидва входи:

```
curl -sI https://example.com/ | grep -i strict-transport-security
curl -sI http://example.com/ | grep -i strict-transport-security
```

Перший рядок має показати заголовок. Другий стосується переспрямування, і заголовка там бути не повинно.

Старий сервіс під доменом перестає відкриватися в усіх, хто бував на головному сайті. `includeSubDomains` на базовому домені охоплює кожне ім’я під ним: сторінку статусу на сторонньому сервісі, адмінку принтера, внутрішній інструмент, у якого ніколи не було сертифіката. Ці відвідувачі бачать помилку з’єднання без способу її обійти, а колеги, які ніколи не відкривали головний сайт, заходять на сервіс без проблем, тож скарги важко відтворити. Директиву розгорнули раніше, ніж хтось склав перелік. Складіть його спершу — із зони DNS і з журналів Certificate Transparency, де записано кожне ім’я, на яке видавали публічний сертифікат. Перенесіть кожне ім’я на HTTPS або на інший домен, а тоді додайте директиву з коротким `max-age` .

Попередження, яке співробітники звикли проклацувати, стає стіною. Staging із самопідписаним сертифікатом, внутрішній центр сертифікації, якому не довіряють ноутбуки відвідувачів, сертифікат, термін дії якого скінчився на вихідних: до HSTS кожен такий випадок давав сторінку

попередження з кнопкою «далі», а після браузер такої кнопки не показує. RFC 9325 прямо називає самопідписані сертифікати випадком, коли HSTS може погіршити безпеку. Дайте staging публічно довірений сертифікат або домен, якого політика не стосується, і ставтеся до поновлення сертифікатів як до того, що тепер у разі збою кладе сайт, не лишаючи відвідувачеві вибору.

Домен у списку preload знову потребує відкритого HTTP. Пристрій, що вміє лише HTTP, постачальник, чия функція власного домену працює без сертифіката, придбана компанія, що приносить під це ім'я свої хости. Запис уже сидить у випущених браузерах, і сервер мало що може вдіяти: запит на вилучення доходить до користувачів лише разом з оновленням їхніх браузерів. Розмістіть нове ім'я під іншим доменом, а заявку на preload подавайте лише після довгого періоду з річним `max-age` та `includeSubDomains`, за який не надійшло жодної скарги.

Що почитати далі

- RFC 6797: HTTP Strict Transport Security (HSTS) — заголовок, дві його директиви і те, коли саме браузер мусить його виконати, а коли проігнорувати.
- HSTS Preload List Submission — умови потрапляння до списку, поетапне розгортання, яке радять супровідники, і ціна виходу.
- RFC 9325, розділ 3.2 — чинна найкраща практика для TLS про HSTS: клієнти й сервери мусять його підтримувати, а серверам слід його надсилати.
- Upgrade Insecure Requests — директива W3C, що переводить на HTTPS власні запити сторінки, і те, як вона ділить роботу з HSTS.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке HSTS?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/hsts/