Engineer

HSTS

Règle du navigateur imposant HTTPS aux visites suivantes

Ce que c’est

HTTP Strict Transport Security, presque toujours abrégé en HSTS, est un en-tête de réponse qui demande au navigateur de joindre un site uniquement en HTTPS pendant un nombre de secondes défini, et de refuser la connexion si le certificat est incorrect. Nommé `Strict-Transport-Security`, il est défini par le RFC 6797 depuis novembre 2012. Il ne chiffre rien lui-même : il change ce qu’un navigateur accepte de faire en HTTPS.

Une valeur courante comporte deux directives :

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

`max-age` indique combien de temps le navigateur conserve la règle, ici un an. Avec `includeSubDomains`, elle atteint tous les noms sous l’hôte : `api.example.com` et `old.example.com` héritent de la demande d’ `example.com`. Un troisième jeton, `preload`, figure dans de nombreuses configurations. Il vient de la liste de préchargement gérée par hstspreload.org, pas du RFC, et marque le consentement du site à inscrire son nom dans les navigateurs.

Dès qu’il a reçu l’en-tête sur une réponse sécurisée, le navigateur fait trois choses pour cet hôte jusqu’à l’expiration. Il réécrit toute adresse `http://` en `https://` avant qu’une requête quitte la machine. Il traite toute erreur de certificat comme fatale, sans bouton pour continuer malgré tout. Il relance enfin le compte à rebours à chaque nouvel en-tête reçu. Une réponse portant `max-age=0` lui fait oublier l’hôte.

La comparaison habituelle est une redirection du port 80 au port 443. Elle fonctionne, mais seulement après l'envoi de la première requête en clair ; quelqu'un sur le même réseau peut y répondre lui-même et maintenir le visiteur sur une copie non chiffrée du site. HSTS supprime cette requête en clair des visites suivant la première. Upgrade Insecure Requests, directive de la politique de sécurité du contenu, est l'autre notion voisine : elle passe en HTTPS les requêtes de la page pour ses scripts et images, mais un visiteur arrivant par le lien `http://` d'un autre site dépend encore de la redirection.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que HSTS? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des comptes, mots de passe et droits attribués selon les besoins exacts de chaque poste, puis retirés le jour du départ.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. 1 000 règles RBAC pour sécuriser Linux — 1,000 règles d'accès écrites pour combler une faille de sécurité

1 000 règles RBAC pour sécuriser Linux Contactez-nous →

Pourquoi c'est important

Activer HSTS coûte une ligne de configuration. La vraie décision est de savoir jusqu'où aller : chaque étape protège davantage de visiteurs et devient plus difficile à annuler.

L'en-tête seul laisse une faille que le RFC 6797 décrit lui-même. Un navigateur qui n'a jamais vu le site, ou dont la politique mémorisée a expiré, commence encore par une requête en clair. Un long `max-age` réduit la seconde partie de cette faille ; un an est donc courant. La première ne se ferme qu'avec la liste de préchargement : des noms compilés dans Chrome, dont Firefox, Safari et Edge maintiennent des listes dérivées, pour qu'un navigateur fraîchement installé sache déjà utiliser HTTPS.

La liste est exigeante. Le domaine de base doit avoir un certificat valide, une redirection HTTP vers HTTPS restant sur le même hôte, tous ses sous-domaines servis en HTTPS et un en-tête avec `max-age` d'au moins 31 536 000 secondes, `includeSubDomains` et `preload`. Ses responsables précisent que cela inclut les sous-domaines internes invisibles au public. La sortie est ce qui tranche la décision : un retrait met des mois à atteindre les utilisateurs par les mises à jour de Chrome, sans garantie pour les autres navigateurs ; le site le décrit comme « lent et pénible ».

Le choix se présente donc comme une échelle à gravir, un échelon à la fois :

- **L'en-tête sans** `includeSubDomains` convient à presque tout site déjà servi en HTTPS. Le RFC 9325, référence actuelle des bonnes pratiques TLS, recommande aux serveurs web de l'envoyer.
- **Ajouter** `includeSubDomains` est sûr une fois tous les noms du domaine recensés et leur réponse HTTPS vérifiée, y compris ceux exploités par un fournisseur ou un autre service.
- **Le préchargement** convient à un domaine dont les propriétaires savent qu'aucun nom sous ce domaine n'aura plus besoin de HTTP en clair : domaine de connexion, de paiement, entreprise contrôlant tout son DNS. Un domaine marketing où un outil de campagne pourrait bientôt demander un CNAME est un mauvais candidat, comme tout domaine partagé avec du matériel de bureau.

Les responsables donnent aussi le rythme de l'ascension : `max-age` à 300 secondes, puis une semaine, puis un mois, chaque fois avec `includeSubDomains`, en attendant toute la durée d'une étape avant la suivante. Le navigateur garde la dernière valeur reçue ; une erreur au palier de cinq minutes coûte donc cinq minutes.

À quoi cela ressemble en pratique

Sur une plateforme détenant des données professionnelles et organisationnelles, l'entreprise a exercé des fonctions d'ingénierie de sécurité et durci toutes les couches, dont HSTS. Dans le middleware proxy Next.js, `proxy.ts`, la réponse porte HSTS, aux côtés d'une politique de sécurité du contenu avec un nonce neuf à chaque requête et `strict-dynamic`, et de cookies SameSite. Un middleware proxy s'exécute devant les routes qu'il sélectionne : les règles destinées au navigateur restent dans un fichier, au lieu d'être réparties entre pages.

Le même travail montre où HSTS s'arrête. L'hypothèse était celle d'un client hostile : toute contrainte imposée par le navigateur peut être désactivée par la personne qui le contrôle. HSTS protège un visiteur honnête d'un réseau hostile, mais ne fait rien contre un client hostile ; le travail a donc continué en dessous. L'API a ajouté des limites de débit, CORS, des limites de taille des requêtes, une validation des entrées avant que quoi que ce soit n'atteigne la base et une journalisation des événements de sécurité. Sa connexion à la base utilisait un rôle ne pouvant qu'exécuter les fonctions de l'application ; les droits étaient revérifiés

sur le serveur à chaque requête, les barrières du frontend relevant seulement de l'expérience utilisateur. Dans cette pile, HSTS met le trafic lui-même hors d'atteinte et constitue la couche la moins coûteuse à ajouter.

Où les problèmes surviennent

L'en-tête est configuré, mais les navigateurs font comme s'il était absent. Il est envoyé sur une réponse HTTP en clair, souvent la redirection, ou sur une réponse HTTPS dont le certificat est refusé. Le RFC 6797 demande aux navigateurs de l'ignorer dans les deux cas et aux serveurs de l'omettre entièrement des réponses en clair. Une erreur voisine envoie `example.com` directement vers `https://www.example.com`, avec l'en-tête seulement sur `www`, laissant le domaine de base et ses autres noms sans protection. Ajoutez l'en-tête à toute réponse HTTPS, redirections comprises, et vérifiez les deux entrées :

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

La première ligne devrait afficher l'en-tête. La seconde correspond à la redirection et ne devrait rien afficher.

Un ancien service du domaine cesse de s'ouvrir pour tous ceux qui ont visité le site principal. `includeSubDomains` sur le domaine de base couvre chaque nom sous ce domaine : page d'état chez un fournisseur, administration d'une imprimante, outil intranet sans certificat. Ces visiteurs obtiennent une erreur sans possibilité de passer outre, tandis que les collègues n'ayant jamais ouvert le site principal y accèdent sans difficulté : les signalements sont donc difficiles à reproduire. La directive a été publiée avant l'inventaire. Faites-le d'abord, à partir de la zone DNS et des journaux de transparence des certificats, qui recensent chaque nom ayant reçu un certificat public. Passez chaque nom en HTTPS ou sur un autre domaine, puis ajoutez la directive avec un `max-age` court.

Un avertissement que le personnel franchissait d'un clic devient un mur. Une préproduction sur certificat autosigné, une autorité interne à laquelle les portables des visiteurs ne font pas confiance, un certificat expiré pendant le week-end : avant HSTS, une page d'avertissement permettait de continuer ; ensuite, le navigateur ne l'offre plus. Le RFC 9325 cite les certificats autosignés comme un cas où HSTS peut dégrader la sécurité. Donnez à la préproduction un certificat émis par une autorité de certification publique reconnue par les

navigateurs ou un domaine hors de portée de la politique. Traitez désormais le renouvellement comme une opération dont l'échec arrête le site, sans laisser de choix au visiteur.

Un domaine préchargé a de nouveau besoin de HTTP en clair. Un appareil ne parlant que HTTP, un fournisseur ne servant pas de certificat pour son option de domaine personnalisé, une acquisition apportant ses propres hôtes sous ce nom. L'entrée est déjà intégrée à des navigateurs distribués ; le serveur n'y peut pas grand-chose : le retrait n'atteint les utilisateurs qu'au fil des mises à jour de leur navigateur. Placez le nouveau nom sous un autre domaine et ne préchargez qu'après une longue période sans ticket avec `max-age` d'un an et `includeSubDomains` .

Pour aller plus loin

- RFC 6797 : HTTP Strict Transport Security (HSTS) — En-tête, deux directives et circonstances précises où le navigateur doit le respecter ou l'ignorer.
- Inscription sur la liste de préchargement HSTS — Conditions d'inscription, déploiement progressif conseillé et coût du retrait.
- RFC 9325, section 3.2 — Bonnes pratiques TLS concernant HSTS : clients et serveurs doivent le prendre en charge, et les serveurs devraient l'envoyer.
- Upgrade Insecure Requests — Directive W3C passant les requêtes de la page en HTTPS et répartition des rôles avec HSTS.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que HSTS ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/hsts/