Engineer

Cookie

couple nom-valeur stocké et renvoyé par le navigateur

Ce que c’est

Un cookie, ou cookie HTTP, est un court couple nom-valeur qu’un serveur web envoie au navigateur dans un en-tête de réponse, et que le navigateur renvoie avec ses requêtes ultérieures vers le même serveur. C’est ainsi que le protocole HTTP, qui traite chaque requête comme une inconnue, acquiert une mémoire : une connexion qui survit au clic suivant, un panier toujours rempli au paiement, une langue choisie une fois puis conservée.

Le serveur définit chaque cookie avec son propre en-tête `Set-Cookie`, et le navigateur renvoie tous les cookies applicables dans un seul en-tête `Cookie`. Ici, les trois premières lignes sont une réponse et le reste une requête ultérieure du même navigateur :

```
HTTP/1.1 200 OK
Set-Cookie: __Host-sid=4f2c9a71e0b8; Path=/; Secure; HttpOnly; SameSite=Lax
Set-Cookie: lang=da; Max-Age=31536000; Path=/

GET /account/ HTTP/1.1
Host: shop.example
Cookie: __Host-sid=4f2c9a71e0b8; lang=da
```

Tout ce qui suit le premier point-virgule est un attribut, une instruction sur la durée de conservation du cookie et les endroits où l’envoyer. `Expires` ou `Max-Age` fixe une durée de vie maximale ; sans l’un ni l’autre, le cookie subsiste jusqu’à ce que le navigateur décide que la session est terminée. `Domain` étend sa portée à un domaine entier et à ses sous-domaines ; sans cet attribut, il ne revient qu’à l’hôte qui l’a défini. `Path` la restreint à une partie du site. `Secure` empêche son envoi sur les connexions non chiffrées, `HttpOnly` le cache aux scripts qui lisent `document.cookie`, et `SameSite` décide s’il accompagne les requêtes lancées par un autre site.

Le retour ne contient que les noms et les valeurs : le serveur ne peut donc pas en déduire quand un cookie expire, s'il a été défini avec `Secure` ni qui l'a écrit en dernier.

Les règles figurent dans la RFC 6265, publiée en 2011. Sa révision, connue sous le nom de rfc6265bis, ajoute `SameSite`, les préfixes de nom `__Secure-` et `__Host-`, et l'obligation pour les navigateurs de plafonner toute durée de vie à 400 jours. Au moment de la rédaction, elle a été approuvée et attend que le RFC Editor la publie pour remplacer la RFC 6265.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'un cookie ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Un site rapide, facile à trouver et modifiable par votre équipe, sans facture à chaque changement de texte.

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. 100 applications web avec HTML/HTML5 — 100 applications web développées

100 applications web avec HTML/HTML5 Contactez-nous →

Pourquoi c'est important

Un cookie est généralement la seule chose qui distingue une requête authentifiée d'une requête anonyme. Après la connexion, l'identifiant de session dans le cookie représente l'utilisateur aux yeux du serveur : quiconque le présente obtient l'accès au compte, sans qu'aucun mot de passe soit demandé. La plupart des problèmes de cookies viennent de la lecture, de la copie ou de l'utilisation d'un cookie par un tiers.

Le navigateur facilite ce dernier cas. Il joint les cookies à chaque requête qui leur correspond, quel qu'en soit l'initiateur. Un formulaire sur un autre site qui envoie une requête POST à une boutique arrive avec les cookies du visiteur ; le serveur voit une requête d'un utilisateur connecté. La RFC 6265 appelle cela l'autorité ambiante et nomme les attaques qu'elle favorise : la falsification de requête intersite et le problème de l'adjoint confus.

Les cookies sont aussi l'outil de suivi le plus ancien du web. Un script intégré sur des milliers de sites peut définir un cookie depuis son propre domaine, que le navigateur renvoie depuis chacun de ces sites : une seule entreprise peut ainsi relier les visites d'une personne sur tous ces sites. La révision décrit cet usage des

cookies tiers, note que la plupart des navigateurs les bloquent ou les partitionnent désormais et recommande une politique aussi restrictive que le permet la compatibilité.

Le droit s'attache au stockage plutôt qu'à la technologie. Dans l'UE, l'article 5, paragraphe 3, de la directive ePrivacy n'autorise un site à stocker des informations sur l'appareil d'un visiteur ou à lire ce qui y est stocké qu'avec un consentement donné après une information claire et complète. Les exceptions concernent le stockage utilisé uniquement pour transmettre une communication et celui strictement nécessaire à un service explicitement demandé par le visiteur. Cet article est à l'origine des bandeaux de consentement et ne mentionne jamais les cookies par leur nom. Les lignes directrices du Comité européen de la protection des données sur son champ d'application technique l'appliquent aussi aux pixels et aux liens de suivi.

À quoi cela ressemble en pratique

Ce site n'en définit aucun. Sa page sur les cookies explique pourquoi : il n'y a ni comptes, ni panier, ni moyen de suivre un visiteur ; il n'y a donc pas de bandeau de consentement, puisqu'il n'y a rien auquel consentir. Elle énumère aussi les autres endroits où un site peut laisser quelque chose, de `localStorage` à un service worker, et indique qu'aucun n'est utilisé non plus.

Se passer de cookies change la manière dont un site connaît ses lecteurs. Compter les lecteurs à partir du seul journal d'accès mesure l'audience sans cookie, script ni pixel, à partir du journal du serveur web lui-même, et retire de chaque ligne, au moment où elle est écrite, l'en-tête de requête `Cookie` ainsi que celui d'autorisation. Le résultat est volontairement peu détaillé : les visiteurs uniques et le parcours d'une personne dans le site restent hors de portée, et la page affiche ces limites à côté des chiffres.

Lorsqu'une application a besoin d'un cookie, la question utile est de savoir ce qu'on lui confie. Sur une plateforme où une même personne agit pour plusieurs organisations, l'organisation active est conservée dans le navigateur et recopiée dans un cookie côté serveur pour que les pages rendues sur le serveur affichent le même contexte. Aucune des deux copies ne décide de l'accès. Le serveur vérifie l'autorisation à chaque requête : modifier la valeur ne change donc que ce que la personne voit sur son propre écran. C'est le schéma sûr pour tout cookie : la valeur peut choisir ce qui sera affiché, mais la décision sur ce qu'une personne peut faire reste sur le serveur.

Où les problèmes surviennent

Un script de la page lit la session. Sans `HttpOnly`, un script injecté par un attaquant peut lire `document.cookie` et envoyer l'identifiant de session n'importe où. `HttpOnly` retire le cookie de cette API. Il empêche le vol, pas l'usage abusif : le script injecté peut toujours envoyer des requêtes et le navigateur y joint toujours le cookie. `HttpOnly` complète donc une Content Security Policy au lieu de la remplacer.

Le cookie passe par HTTP en clair. Un cookie dépourvu de `Secure` part avec toute requête `http://` vers son hôte et peut être lu par quiconque sur le trajet réseau. Dans la RFC 6265, `Secure` ne protège que la confidentialité ; un attaquant sur le réseau peut encore écraser un cookie portant l'attribut Secure depuis une connexion non sécurisée. La révision réduit cette faiblesse en ignorant les cookies `Secure` définis sur HTTP en clair et en refusant un cookie HTTP en clair qui recouvrirait un cookie portant l'attribut Secure.

Un sous-domaine voisin définit le cookie. `Domain=example.com` envoie un cookie à chaque hôte de ce domaine, et n'importe lequel peut définir un cookie pour le domaine entier, que le site principal lit ensuite comme le sien. `Path` n'est pas non plus une frontière de sécurité. Le préfixe `__Host-` répond à ce problème : le navigateur n'accepte un tel cookie que s'il est `Secure`, a `Path=/` et n'a pas de `Domain`, ce qui le limite au seul hôte qui l'a défini.

Un autre site lance la requête. `SameSite=Strict` réserve le cookie aux requêtes du même site ; `Lax` ajoute les navigations de premier niveau utilisant des méthodes sûres, comme suivre un lien ; `None` prévoit son envoi partout, mais le cookie est ignoré s'il n'est pas aussi `Secure`. La révision traite un cookie sans `SameSite` presque comme `Lax` ; un serveur qui en dépend devrait le dire explicitement. Un site, dans ce contexte, est le schéma plus le domaine enregistrable : `SameSite` ne protège donc pas contre un sous-domaine voisin hostile.

Le serveur croit la valeur. Tout ce qui figure dans un cookie peut être modifié par la personne dont le navigateur le conserve. La RFC 6265 recommande aux serveurs de chiffrer et signer son contenu, tout en avertissant que le cookie reste exposé au rejeu. Le schéma plus robuste est un identifiant de session aléatoire qui renvoie à un état conservé sur le serveur et qui est remplacé à la connexion. Un identifiant déposé auparavant par un attaquant n'a alors plus de valeur ; c'est contre cette fixation de session que la RFC met en garde.

Le navigateur l'élimine. Une date d'expiration est un maximum. Les navigateurs suppriment des cookies pour rester dans leurs limites ; un attaquant peut aussi en évincer en définissant un grand nombre des siens. Les serveurs ne devraient pas compter sur leur conservation par les navigateurs. La suppression recèle aussi un piège : un `Set-Cookie` avec une expiration passée ne supprime le cookie que si ses `Path` et `Domain` correspondent à ceux de sa création.

Pour aller plus loin

- RFC 6265 : mécanisme de gestion d'état HTTP — la norme de `Set-Cookie` et `Cookie`, avec les règles d'attribut et une section de sécurité sur l'autorité ambiante, les garanties d'intégrité limitées et les identifiants de session.
- Cookies : mécanisme de gestion d'état HTTP (rfc6265bis) — la révision approuvée, avec `SameSite`, les préfixes `__Secure-` et `__Host-`, le plafond de 400 jours et une section sur les cookies tiers.
- Directive 2002/58/EC, texte consolidé — la directive ePrivacy telle que modifiée en 2009, avec l'article 5, paragraphe 3, relatif au stockage et à la lecture d'informations sur l'appareil de l'utilisateur.
- Lignes directrices EDPB 2/2023 sur le champ d'application technique de l'article 5, paragraphe 3 — comment la règle s'étend aux pixels de suivi, aux liens de suivi, au traitement local et à d'autres techniques au-delà des cookies.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'un cookie ? » 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/cookie/