Engineer

Cookie

サーバーがブラウザーに送り、後の該当リクエストで返される名前と値の組

どのようなものか

Cookie、またはHTTP Cookieは、ウェブサーバーが応答ヘッダーでブラウザーへ送り、ブラウザーが後で同じサーバーへ送るリクエストに付けて返す、短い名前と値の組です。リクエストを毎回初対面として扱うHTTPというプロトコルに、記憶を与えます。次にクリックしてもログインが続くこと、購入手続きに進んでも買い物かごの中身が残ること、一度選んだ言語を維持することに使います。

サーバーはCookieごとに独立した `Set-Cookie` ヘッダーで設定し、ブラウザーは該当するすべてのCookieを一つの `Cookie` ヘッダーにまとめて返します。次の最初の三行は応答で、残りは同じブラウザーが後から送るリクエストです。

```
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
```

最初のセミコロンより後はすべて属性で、保存期間と送信先についての指示です。 `Expires` または `Max-Age` は有効期間の上限を設定します。どちらもなければ、ブラウザーがセッションの終了を判断するまで残ります。 `Domain` は対象をドメイン全体とそのサブドメインへ広げ、指定しなければ、設定したホストだけに返します。 `Path` は対象をサイトの一部に絞ります。 `Secure` は暗号化されていない接続へ送らないようにし、 `HttpOnly` は `document.cookie` を読むスクリプトから隠します。 `SameSite` は、別のサイトが開始したリクエストに付けるかどうかを決めます。

返す際に含まれるのは名前と値だけです。そのため、サーバーは、それだけでは有効期限、設定時に `Secure` を指定したか、最後に誰が書き込んだかを判断できません。

規則は、2011年に公開されたRFC 6265にあります。rfc6265bisと呼ぶ改訂版は、`SameSite`、名前の接頭辞 `__Secure-` と `__Host-`、そしてすべてのCookieの有効期間を400日以下に制限するブラウザーへの要件を追加しています。執筆時点では承認済みで、RFC 6265の後継としてRFC Editorが公開するのを待っています。

概念を判断に生かす

「Cookieとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 表示が速く、検索で見つけやすく、文章を変えるたびに料金が発生せず、自社の担当者が編集できるサイトをつくります。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 HTML/HTML5でWebアプリケーション100個を開発 — 100 構築したWebアプリケーション

HTML/HTML5でWebアプリケーション100個を開発 お問い合わせ →

なぜ重要なのか

通常、ログイン済みのリクエストと匿名のリクエストの違いは、Cookieだけです。ログイン後、サーバーにとって、Cookieのセッション識別子は利用者そのものになります。それを提示する人は誰でもそのアカウントを使え、パスワードは尋ねられません。Cookieで起きる問題の大半は、誰か別の人が読み取り、コピーし、利用することです。

ブラウザーは、最後の「利用すること」を容易にします。誰が開始したリクエストであっても、条件に合えばCookieを添付します。別のサイトにあるフォームから店へ送ったPOSTも、訪問者のCookie付きで届き、サーバーにはログイン済み利用者のリクエストに見えます。RFC 6265はこれを暗黙の権限（ambient authority）と呼び、そこから生じる攻撃として、クロスサイトリクエストフォージェリと、混乱した代理人（confused deputy）の問題を挙げています。

Cookieは、ウェブで最も古くからある追跡手段でもあります。何千ものサイトに埋め込まれたスクリプトが、自分のドメインからCookieを設定すると、ブラウザーはどのサイトからもそのCookieを送り返すため、一つの企業がそれらすべてにまたがる訪問履歴を結び付けられます。改訂版は、こうしたサードパーティーCookieの利用を説明し、現在は大半のブラウザーが遮断するか、分割して管理していると述べ、互換性が許す限り制限の強い方針を勧めています。

法律が対象にするのは、技術名ではなく保存という行為です。EUでは、ePrivacy指令の第5条第3項により、訪問者の端末に情報を保存したり、そこにある情報を読んだりするには、明確で包括的な説明を行った後に得た同意が必要です。例外は、通信の伝送だけを目的とする保存と、訪問者が明示的に求めたサービスの提供に不可欠な保存です。同意バナーの根拠はこの条文であり、Cookieという名前は一度も出てきません。欧州データ保護会議の技術的な適用範囲に関するガイドラインは、追跡ピクセルと追跡リンクにも適用しています。

実際の活用例

このサイトはCookieを一つも設定しません。Cookieのページで、その理由を説明しています。アカウントも買い物かごもなく、訪問者を追跡するものもありません。同意する対象がないため、同意バナーもありません。このページは、`localStorage` からサービスワーカーまで、サイトが端末に何かを残せるほかの場所も挙げ、それらも一切使っていないと説明しています。

Cookieを使わないと、読者について知る方法も変わります。アクセスログだけで読者を数えるでは、ウェブサーバー自身のログから、Cookie、スクリプト、ピクセルを使わずに閲覧状況を測定しています。各行を書き込む際に、認証ヘッダーとともに `Cookie` リクエストヘッダーを除外します。結果が粗いのは意図どおりです。ユニーク訪問者や、訪問者がサイト内をどう移動したかは分かりません。事例ページは、その限界を数値のそばに明記しています。

アプリケーションでCookieが必要な場合は、何を任せてよいかを考えると役立ちます。一人が複数の組織を代表して操作するプラットフォームでは、現在の組織をブラウザーで保持し、サーバー側のCookieにも反映します。サーバーで描画するページも、同じ組織を対象に表示するためです。どちらのコピーも、アクセス権を決めません。サーバーがリクエストごとに認可を確認するため、値を編集しても、自分の画面に表示する内容が変わるだけです。この役割分担が、どのCookieにも当てはまる安全な設計です。値で描画内容を選ぶことはできても、誰が何をできるかの判断はサーバーに残します。

問題が起きるところ

ページ上のスクリプトがセッションを読み取ります。 `HttpOnly` がなければ、攻撃者が挿入したスクリプトは `document.cookie` を読み、セッション識別子をどこへでも送れます。 `HttpOnly` は、そのAPIからCookieを除きます。盗み取りは防いでも、不正利用は防ぎません。挿入されたスクリプトは引き続きリクエストを送れ、ブラウザーもCookieを添付します。そのため、 `HttpOnly` はContent Security Policyと併用するものであり、その代わりにはなりません。

Cookieが平文のHTTPを通ります。 `Secure` のないCookieは、そのホストへのあらゆる `http://` リクエストに付いて送られ、経路上の誰にでも読めます。RFC 6265では、 `Secure` が守るのは機密性だけです。ネットワーク上の攻撃者は、暗号化されていない接続から、Secure属性付きのCookieを依然として上書きできます。改訂版は、平文のHTTP経由で設定しようとする `Secure` 付きのCookieを無視し、Secure属性付きのCookieに重なる平文HTTPのCookieを拒否することで、この隙間を狭めています。

同じ親ドメインの別のサブドメインがCookieを設定します。 `Domain=example.com` は、そのドメイン配下のすべてのホストへCookieを送り、それらのホストはどれでもドメイン全体のCookieを設定できます。主サイトは、それを自分のCookieとして読み取ります。 `Path` もセキュリティ上の境界にはなりません。 `__Host-` 接頭辞は、この問題に対応します。ブラウザーがそのCookieを受け入れるのは、 `Secure` があり、 `Path=/` で、 `Domain` がない場合だけです。これによって、設定した一つのホストに限定されます。

別のサイトがリクエストを開始します。 `SameSite=Strict` は同じサイトから行うリクエストだけにCookieを付け、 `Lax` は、リンクをたどるなどの安全なメソッドによるトップレベルの画面遷移も加えます。 `None` はどこへでも送る指定ですが、Cookieに `Secure` もなければ、そのCookieは無視されます。改訂版は、 `SameSite` のないCookieをほぼ `Lax` と同じように扱います。その動作に依存するサーバーは、明示的に指定すべきです。この目的でいうサイトは、スキームと登録可能なドメインの組み合わせなので、 `SameSite` は、同じ親ドメインの悪意あるサブドメインには効果がありません。

サーバーが値を信用します。 Cookie内のものはすべて、それを持つブラウザーの利用者が編集できます。RFC 6265は、Cookieの内容を暗号化して署名するようサーバーに勧めていますが、それでもリプレイ攻撃は可能だと注意しています。より堅牢な方法は、サーバーに保持した状態を指すランダムなセッション識別子を使い、ログイン時に置き換えることです。こうすれば、攻撃者が事前に仕込んだ識別子は、その後は意味を持ちません。RFCが注意するセッション固定攻撃への対策です。

ブラウザーがCookieを手放します。 有効期限は上限です。ブラウザーは、自らの制限内に収めるためにCookieを追い出します。攻撃者も、自分のCookieを大量に設定して、ほかのCookieを追い出せます。サーバーは、ブラウザーがCookieを保持し続けることを当てにすべきではありません。削除にも落とし穴があります。有効期限を過去にした `Set-Cookie` で削除できるのは、 `Path` と `Domain` が作成時のものと一致する場合だけです。

参考資料

- RFC 6265：HTTPの状態管理機構 — `Set-Cookie` と `Cookie` の標準です。属性の規則と、暗黙の権限（ambient authority）、完全性の弱さ、セッション識別子についてのセキュリティの節があります。
- Cookie：HTTPの状態管理機構（rfc6265bis） — 承認済みの改訂版です。`SameSite`、接頭辞 `__Secure-` と `__Host-`、400日という上限、サードパーティーCookieの節を含みます。
- 指令2002/58/EC、統合版 — 2009年に改正されたePrivacy指令です。利用者の端末での情報の保存と読み取りに関する第5条第3項があります。
- 第5条第3項の技術的な適用範囲に関するEDPBガイドライン2/2023 — Cookie以外の追跡ピクセル、追跡リンク、ローカル処理、ほかの手法に規則がどう及ぶかを説明します。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「Cookieとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/cookie/