Engineer

Cookie

en lille værdi, et site beder browseren gemme

Hvad det er

En cookie, eller HTTP-cookie, er et kort navn med en værdi, som en webserver sender til browseren i en svarheader, og som browseren sender med igen i sine senere forespørgsler til den samme server. Det er cookies, der giver HTTP en hukommelse, selv om protokollen møder hver forespørgsel som en fremmed: et login, der holder til næste klik, en kurv, der stadig er fuld ved kassen, et sprog, man vælger én gang. Så længe cookien gælder, genkender serveren browseren.

Serveren sætter hver cookie med sin egen `Set-Cookie` -header, og browseren sender alle de cookies, der passer, tilbage i én `Cookie` -header. Her er de første tre linjer et svar, og resten er en senere forespørgsel fra den samme browser:

```
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
```

Alt efter det første semikolon er attributter, instrukser om, hvor længe cookien skal gemmes, og hvor den skal sendes hen. `Expires` eller `Max-Age` sætter en maksimal levetid; uden nogen af dem lever cookien, til browseren afgør, at sessionen er slut. `Domain` udvider cookien til et helt domæne med underdomæner, og uden den går cookien kun tilbage til den vært, der satte den. `Path` indsnævrer den til en del af sitet. `Secure` holder den væk fra ukrypterede forbindelser, `HttpOnly` skjuler den for scripts, der læser `document.cookie` , og `SameSite` afgør, om den følger med på forespørgsler, som et andet site sætter i gang.

Vejen tilbage bærer kun navne og værdier. Serveren får intet at vide om, hvornår en cookie udløber, om den blev sat med `Secure`, eller hvem der sidst skrev den.

Reglerne står i RFC 6265 fra 2011. Revisionen, kendt som rfc6265bis, tilføjer `SameSite`, navnepræfikserne `__Secure-` og `__Host-` og et krav om, at browsere begrænser enhver levetid til højst 400 dage. I skrivende stund er den godkendt og venter hos RFC Editoren på at afløse RFC 6265.

Brug begrebet i en beslutning

Brug »Hvad er en cookie?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Et site, der er hurtigt, kan findes og kan redigeres af jeres egne folk, uden en regning for hver ændret sætning.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har udviklet 100 webapplikationer med HTML/HTML5 — 100 webapplikationer bygget

Har udviklet 100 webapplikationer med HTML/HTML5 Kontakt os →

Hvorfor det betyder noget

En cookie er som regel det eneste, der skiller en forespørgsel fra en indlogget bruger fra en anonym. Efter et login er sessionsidentifikatoren i cookien brugeren, set fra serveren: den, der viser den frem, får kontoen, og ingen beder om en adgangskode. Det meste, der går galt med cookies, handler om, at en anden læser, kopierer eller bruger en.

Det sidste gør browseren let. Den hænger cookies på hver forespørgsel, de passer til, uanset hvem der satte forespørgslen i gang. En formular på et andet site, der sender til en webshop, kommer frem med den besøgendes cookies, og serveren ser en forespørgsel fra en indlogget bruger. RFC 6265 kalder det ambient authority og nævner de angreb, det åbner for: cross-site request forgery og confused deputy.

Cookies er også nettets ældste sporingsværktøj. Et script, der er indlejret på tusindvis af sites, kan sætte en cookie fra sit eget domæne, som browseren sender tilbage fra hvert af dem, så én virksomhed kan lægge en persons besøg på dem alle ved siden af hinanden. Revisionen beskriver denne brug af tredjepartscookies, nævner, at de fleste browsere nu blokerer eller partitionerer dem, og anbefaler en politik, der er så restriktiv, som kompatibiliteten tillader.

Loven følger lagringen frem for teknologien. I EU må et site efter artikel 5, stk. 3, i ePrivacy-direktivet kun gemme oplysninger i en besøgendes terminaludstyr, eller læse det, der allerede ligger der, når den besøgende har givet samtykke efter at have fået klare og fyldestgørende oplysninger. Undtagelserne er lagring, der alene tjener til at overføre kommunikation, og lagring, der er absolut påkrævet for en tjeneste, den besøgende udtrykkeligt har bedt om. Det er den artikel, samtykkebannere kommer fra, og den nævner slet ikke cookies ved navn. Det Europæiske Databeskyttelsesråds retningslinjer om dens tekniske rækkevidde anvender den også på sporingspixels og sporingslinks.

Sådan ser det ud i praksis

Dette site sætter ingen. Cookiesiden forklarer hvorfor: der er ingen konti, ingen kurv og intet at følge en besøgende med, så der er intet samtykkebanner, fordi der ikke er noget at samtykke til. Den gennemgår også de andre steder, et site kan efterlade noget, fra `localStorage` til en service worker, og siger, at de heller ikke bruges.

At undvære cookies ændrer, hvordan et site lærer sine læsere at kende. Optælling af læsere alene ud fra adgangsloggen måler læsertal uden cookie, script eller pixel, ud fra webserverens egen log, og fjerner `Cookie`-headeren fra hver linje i samme øjeblik, den skrives, sammen med autorisationsheaderen. Resultatet er groft med vilje: unikke besøgende og en besøgendes vej gennem sitet ligger uden for rækkevidde, og siden skriver de grænser ved siden af tallene.

Hvor en applikation har brug for en cookie, er det nyttige spørgsmål, hvad cookien bliver betroet. På en platform, hvor én person handler på vegne af flere organisationer, gemmes den aktive organisation i browseren, og en cookie spejler den til serveren, så sider, der renderes på serveren, viser den samme kontekst. Ingen af kopierne afgør adgang. Serveren tjekker autorisationen ved hver forespørgsel, så den, der redigerer værdien, ændrer, hvad deres egen skærm viser, og intet mere. Det er den sikre form for enhver cookie: værdien må vælge, hvad der vises, og beslutningen om, hvad nogen må, bliver på serveren.

Hvor det går galt

Et script på siden læser sessionen. Uden `HttpOnly` kan et script, som en angriber sprøjter ind, læse `document.cookie` og sende sessionsidentifikatoren hvor som helst hen. `HttpOnly` tager cookien ud af det API. Det stopper tyveri,

ikke misbrug: det indsprøjtede script kan stadig sende forespørgsler, og browseren hænger stadig cookien på, så `HttpOnly` står ved siden af en Content Security Policy i stedet for at erstatte den.

Cookien krydser almindelig HTTP. En cookie uden `Secure` sendes med på enhver `http://` -forespørgsel til sin vært og kan læses af alle på netværksvejen. Efter RFC 6265 beskytter `Secure` kun fortroligheden, og en angriber på netværket kan stadig overskrive en sikker cookie fra en usikker forbindelse. Revisionen mindsker hullet ved at ignorere `Secure` -cookies, der sættes over almindelig HTTP, og ved at afvise en cookie fra almindelig HTTP, der ville lægge sig over en sikker.

Et søsterunderdomæne sætter cookien. `Domain=example.com` sender en cookie til hver vært under domænet, og enhver af de værter kan sætte en cookie for hele domænet, som hovedsitet så læser som sin egen. `Path` er heller ingen sikkerhedsgrænse. Præfikset `__Host-` svarer på det: en browser godtager kun sådan en cookie, når den er `Secure` , har `Path=/` og ingen `Domain` , og det låser den til den ene vært, der satte den.

Et andet site starter forespørgslen. `SameSite=Strict` holder cookien til forespørgsler fra det samme site, `Lax` tilføjer navigation på øverste niveau med sikre metoder, som når man følger et link, og `None` sender den overalt og ignoreres, medmindre cookien også er `Secure` . Revisionen behandler en cookie uden `SameSite` stort set som `Lax` ; en server, der regner med det, bør skrive værdien ud. Et site er i denne sammenhæng URL'ens skema plus det registrerbare domæne, så `SameSite` gør intet mod et fjendtligt søsterunderdomæne.

Serveren tror på værdien. Alt i en cookie kan redigeres af den person, hvis browser holder den. RFC 6265 anbefaler, at servere krypterer og signerer indholdet, og advarer om, at cookien så stadig kan afspilles igen. Det mere robuste mønster er en tilfældig sessionsidentifikator, der peger på tilstand gemt på serveren og udskiftes ved login, så en identifikator, som en angriber plantede på forhånd, den session fixation RFC'en advarer om, intet betyder bagefter.

Browseren giver slip. En udløbsdato er et maksimum. Browsere smider cookies ud for at holde sig inden for deres grænser, og en angriber kan skubbe cookies ud ved at sætte mange af sine egne. Servere bør ikke regne med, at browsere

beholder cookies. Også sletning har en fælde: en `Set-Cookie` med en udløbsdato i fortiden fjerner kun cookien, når dens `Path` og `Domain` stemmer med dem, den blev oprettet med.

Læs mere

- RFC 6265: HTTP State Management Mechanism — standarden for `Set-Cookie` og `Cookie`, med reglerne for attributter og et sikkerhedsafsnit om ambient authority, svag integritet og sessionsidentifikatorer.
- Cookies: HTTP State Management Mechanism (rfc6265bis) — den godkendte revision med `SameSite`, præfikserne `__Secure-` og `__Host-`, loftet på 400 dage og et afsnit om tredjepartscookies.
- Direktiv 2002/58/EF, konsolideret tekst — ePrivacy-direktivet på dansk, som ændret i 2009, med artikel 5, stk. 3, om at gemme og læse oplysninger i brugerens terminaludstyr.
- EDPB-retningslinjer 2/2023 om den tekniske rækkevidde af artikel 5, stk. 3 — på engelsk: hvordan reglen når sporingspixels, sporingslinks, lokal behandling og andre teknikker ud over cookies.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er en cookie?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/cookie/