Engineer

Informasjonskapsel

en liten verdi et nettsted ber nettleseren ta vare på

Hva det er

En informasjonskapsel, på engelsk HTTP cookie og i daglig tale ofte bare cookie, er et kort navn og en verdi som en webserver sender til nettleseren i en header i svaret, og som nettleseren sender tilbake med de senere forespørslene sine til den samme serveren. Informasjonskapsler er måten HTTP, en protokoll som møter hver forespørsel som en fremmed, får en hukommelse på: en innlogging som overlever neste klikk, en handlekurv som fortsatt er full i kassen, et språk som velges én gang og blir husket.

Serveren setter hver informasjonskapsel med sin egen `Set-Cookie` -header, og nettleseren sender tilbake alle som gjelder, samlet i én `Cookie` -header. Her er de tre første linjene et svar, og resten er en senere forespørsel fra den samme nettleseren:

```
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 etter det første semikolonet er et attributt, en instruks om hvor lenge informasjonskapselen skal tas vare på og hvor den skal sendes. `Expires` eller `Max-Age` setter en øvre levetid. Er ingen av dem satt, varer informasjonskapselen til nettleseren bestemmer at økten er over. `Domain` utvider den til et helt domene og underdomenene, og uten dette attributtet går den bare tilbake til verten som satte den. `Path` snevrer den inn til en del av nettstedet. `Secure`

holder den unna ukrypterte forbindelser, `HttpOnly` skjuler den for skript som leser `document.cookie`, og `SameSite` avgjør om den blir med på forespørsler som et annet nettsted setter i gang.

Returreisen bærer bare navn og verdier, så serveren kan ikke se av den når en informasjonskapsel utløper, om den ble satt med `Secure`, eller hvem som skrev den sist.

Reglene står i RFC 6265, publisert i 2011. Revisjonen, kjent som rfc6265bis, legger til `SameSite`, navneprefiksene `__Secure-` og `__Host-` og et krav om at nettlesere begrenser hver levetid til høyst 400 dager. I skrivende stund er revisjonen godkjent og venter i køen hos RFC Editor på å erstatte RFC 6265.

Bruk begrepet i en beslutning

Bruk «Hva er en informasjonskapsel?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Et nettsted som er raskt, lett å finne og kan redigeres av dine egne ansatte, uten en regning hver gang en formulering endres.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Utviklet 100 webapplikasjoner med HTML/HTML5 — 100 webapplikasjoner bygd

Utviklet 100 webapplikasjoner med HTML/HTML5 Ta kontakt →

Hvorfor det er viktig

En informasjonskapsel er som regel det eneste som skiller en innlogget forespørsel fra en anonym. Etter en innlogging er øktidentifikatoren i informasjonskapselen selve personen, slik serveren ser det: den som viser den fram, får kontoen, og ingen spør etter passordet. Det meste som går galt med informasjonskapsler, handler om at noen andre leser, kopierer eller bruker en.

Nettleseren gjør det siste av disse lett. Den legger ved informasjonskapsler på hver forespørsel de passer til, uansett hvem som startet forespørselen. Et skjema på et annet nettsted som poster til en nettbutikk, kommer fram med den besøkendes informasjonskapsler, og serveren ser en forespørsel fra en innlogget person. RFC 6265 kaller dette *ambient authority*, en fullmakt som følger med av seg selv, og navngir angrepene den inviterer til: forfalskede forespørsler på tvers av nettsteder (*cross-site request forgery*) og *confused deputy*, der en mellommann lures til å bruke sin egen fullmakt for noen andre.

Informasjonskapsler er også det eldste sporingsverktøyet på nettet. Et skript som er bygd inn på tusenvis av nettsteder, kan sette en informasjonskapsel fra sitt eget domene, og nettleseren sender den tilbake fra hvert av disse nettstedene, så ett selskap kan stille opp en persons besøk på tvers av dem alle. Revisjonen beskriver denne bruken av tredjeparts informasjonskapsler, påpeker at de fleste nettlesere nå blokkerer eller partisjonerer dem, og anbefaler en policy som er så restriktiv som kompatibiliteten tillater.

Loven følger lagringen og ikke teknologien. I EU tillater artikkel 5 nr. 3 i kommunikasjonsverndirektivet, ofte kalt ePrivacy-direktivet, at et nettsted lagrer informasjon på en besøkendes enhet, eller leser det som er lagret der, bare med samtykke gitt etter klar og fullstendig informasjon. Unntakene er lagring som bare brukes til å overføre en kommunikasjon, og lagring som er strengt nødvendig for en tjeneste den besøkende uttrykkelig har bedt om. Den artikkelen er opphavet til samtykkebannerne, og den nevner aldri informasjonskapsler ved navn. Retningslinjene fra Det europeiske personvernrådet (EDPB) om det tekniske virkeområdet bruker den også på sporingspiksler og sporingslenker.

I Norge er artikkelen gjennomført i ekomloven § 3-15. Datatilsynet beskriver bestemmelsen som teknologinøytral: den gjelder lagring av og tilgang til alle slags opplysninger i kommunikasjonsutstyret, enten det er personopplysninger eller ikke. Den nye ekomloven trådte i kraft 1. januar 2025, og samtykket må nå være gyldig etter personvernforordningen (GDPR), blant annet frivillig, informert, gitt gjennom en aktiv handling og like lett å trekke tilbake som å gi. Før det var det for eksempel nok at standardinnstillingene i nettleseren tillot informasjonskapsler. Datatilsynet og Nasjonal kommunikasjonsmyndighet (Nkom) deler tilsynet med bestemmelsen.

Slik ser det ut i praksis

Dette nettstedet setter ingen. Siden om informasjonskapsler forklarer hvorfor: det finnes ingen kontoer, ingen handlekurv og ingenting å følge en besøkende med, så det finnes ikke noe samtykkebanner, fordi det ikke er noe å samtykke til. Siden lister også de andre stedene et nettsted kan legge igjen noe, fra `localStorage` til en service worker, og sier at ingen av dem brukes heller.

Å klare seg uten informasjonskapsler endrer hvordan et nettsted lærer om leserne sine. Lesertellingen ut fra tilgangsloggen alene måler lesingen uten informasjonskapsel, skript eller piksel, fra webserverens egen logg, og fjerner

`Cookie` -headeren fra hver linje idet linjen skrives, sammen med authorization-headeren. Resultatet er grovt med vilje: unike besøkende og en besøkendes vei gjennom nettstedet er utenfor rekkevidde, og siden skriver de grensene ved siden av tallene.

Der en applikasjon trenger en informasjonskapsel, er det nyttige spørsmålet hva den blir betrodd. På en plattform der én person handler på vegne av flere organisasjoner, holdes den aktive organisasjonen i nettleseren, og en informasjonskapsel på serversiden speiler den, slik at sider som renderes på serveren, viser den samme konteksten. Tilgangen avgjøres likevel av noe annet enn de to kopiene: serveren kontrollerer autorisasjonen ved hver forespørsel, så den som redigerer verdien, endrer hva vedkommendes egen skjerm viser, og ikke noe mer. Det er den trygge formen for enhver informasjonskapsel: verdien kan velge hva som skal vises, og avgjørelsen om hva noen har lov til å gjøre, blir værende på serveren.

Hvor det går galt

Skript på siden leser økten. Et skript en angriper har sprøytet inn, kan lese `document.cookie` når `HttpOnly` mangler, og sende øktidentifikatoren hvor som helst. `HttpOnly` tar informasjonskapselen ut av det API-et. Det stopper tyveri, men ikke misbruk: det innsprøytede skriptet kan fortsatt sende forespørsler, og nettleseren legger fortsatt ved informasjonskapselen, så `HttpOnly` står ved siden av en Content Security Policy og erstatter den ikke.

Informasjonskapselen krysser vanlig HTTP. En informasjonskapsel uten `Secure` går ut på enhver `http://` -forespørsel til verten sin, lesbar for alle på nettverksveien. Etter RFC 6265 vokter `Secure` bare konfidensialiteten, og en angriper på nettverket kan fortsatt overskrive en sikker informasjonskapsel fra en usikret forbindelse. Revisjonen snevrer inn gapet: den ignorerer `Secure` -informasjonskapsler som settes over vanlig HTTP, og avviser en informasjonskapsel fra vanlig HTTP som ville lagt seg over en sikker.

Et søsterunderdomene setter informasjonskapselen. `Domain=example.com` sender en informasjonskapsel til hver vert under det domenet, og hver av disse vertene kan sette en informasjonskapsel for hele domenet, som hovednettstedet da leser som sin egen. `Path` er heller ingen sikkerhetsgrense. Prefikset `__Host-` svarer på dette: en nettleser godtar en slik informasjonskapsel bare når den er `Secure` , har `Path=/` og står uten `Domain` , noe som låser den til den ene verten som satte den.

Et annet nettsted starter forespørselen. `SameSite=Strict` holder informasjonskapselen til forespørsler fra det samme nettstedet, `Lax` legger til navigasjon på toppnivå med trygge metoder, som å følge en lenke, og `None` sender den overalt og blir ignorert med mindre informasjonskapselen også er `Secure`. Revisjonen behandler en informasjonskapsel uten `SameSite` omtrent som `Lax`, og en server som er avhengig av det, bør si det eksplisitt. Et nettsted er i denne sammenhengen skjemaet i adressen, som `https`, pluss det registrerbare domenet, så `SameSite` gjør ingenting mot et fiendtlig søsterunderdomene.

Serveren tror på verdien. Alt i en informasjonskapsel kan redigeres av personen som har nettleseren den ligger i. RFC 6265 anbefaler at servere krypterer og signerer innholdet i informasjonskapsler, og advarer om at en informasjonskapsel fortsatt kan spilles av på nytt. Det mer robuste mønsteret er en tilfeldig øktidentifikator som peker på tilstand lagret på serveren og byttes ut ved innlogging. Da betyr en identifikator som en angriper plantet på forhånd, ingenting etterpå, og det er nettopp øktfikseringen (*session fixation*) RFC-en advarer mot.

Nettleseren gir slipp på den. En utløpsdato er et maksimum. Nettlesere kaster ut informasjonskapsler for å holde seg innenfor grensene sine, og en angriper kan presse informasjonskapsler ut ved å sette mange egne. Servere bør derfor ikke regne med at nettlesere tar vare på informasjonskapsler. Å slette en har også en felle: en `Set-Cookie` med en utløpsdato i fortiden fjerner informasjonskapselen bare når `Path` og `Domain` stemmer med verdiene den ble opprettet med.

Les mer

- RFC 6265: HTTP State Management Mechanism — standarden for `Set-Cookie` og `Cookie`, med reglene for attributtene og en sikkerhetsdel om ambient authority, svak integritet og øktidentifikatorer.
- Cookies: HTTP State Management Mechanism (rfc6265bis) — den godkjente revisjonen, med `SameSite`, prefiksene `__Secure-` og `__Host-`, taket på 400 dager og en del om tredjeparts informasjonskapsler.
- Directive 2002/58/EC, consolidated text — kommunikasjonsverndirektivet slik det ble endret i 2009, med artikkel 5 nr. 3 om lagring og lesing av informasjon på en enhet.
- EDPB Guidelines 2/2023 on the technical scope of Article 5(3) — hvordan regelen når sporingspiksler, sporingslenker, lokal behandling og andre teknikker ut over informasjonskapsler.

- Bruk av informasjonskapsler (cookies) — Datatilsynets oversikt over ekomloven § 3-15: samtykke etter personvernforordningen, de to unntakene og hvordan tilsynet er delt med Nkom.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er en informasjonskapsel?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/cookie/