Engineer

Kaka

ett litet värde som en webbplats ber webbläsaren att spara

Vad det är

En kaka är ett kort namn med ett värde som en webbserver skickar till webbläsaren i en header i svaret, och som webbläsaren sedan skickar tillbaka med sina senare förfrågningar till samma server. På engelska heter den cookie eller HTTP cookie, och i svenskt tal säger många cookie och cookies lika gärna som kaka och kakor. Kakor är sättet som HTTP, ett protokoll som behandlar varje förfrågan som en främling, får ett minne: en inloggning som överlever nästa klick, en varukorg som fortfarande är full i kassan, ett språk som väljs en gång och sedan står kvar.

Servern sätter varje kaka med en egen `Set-Cookie` -header, och webbläsaren skickar tillbaka alla kakor som gäller i en enda `Cookie` -header. Här är de tre första raderna ett svar och resten en senare förfrågan från samma webbläsare:

```
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
```

Allt efter det första semikolonet är attribut, instruktioner om hur länge kakan ska sparas och vart den ska skickas. `Expires` eller `Max-Age` anger en längsta livstid; om inget av dem finns varar kakan tills webbläsaren bedömer att sessionen är över. `Domain` vidgar kakan till en hel domän med dess underdomäner, och utan attributet går kakan bara tillbaka till den värddator som satte den. `Path` begränsar den till en del av webbplatsen. `Secure` håller den borta från

okrypterade anslutningar, `HttpOnly` döljer den för skript som läser `document.cookie`, och `SameSite` avgör om den följer med på förfrågningar som en annan webbplats sätter i gång.

Returresan bär bara namn och värde för varje kaka, så servern kan inte se på den när en kaka går ut, om den sattes med `Secure` eller vem som senast skrev den.

Reglerna finns i RFC 6265, som publicerades 2011. Revisionen, känd som rfc6265bis, lägger till `SameSite`, namnprefixen `__Secure-` och `__Host-` och ett krav på att webbläsare begränsar varje livstid till högst 400 dagar. När detta skrivs är den godkänd och väntar hos RFC Editor på att publiceras och ersätta RFC 6265.

Använd begreppet i ett beslut

Använd ”Vad är en kaka?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. En webbplats som är snabb, lätt att hitta och går att redigera av din egen personal, utan en faktura för varje ändrad formulering.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Utvecklade 100 webbapplikationer med HTML/HTML5 — 100 byggda webbapplikationer

Utvecklade 100 webbapplikationer med HTML/HTML5 Hör av er →

Varför det spelar roll

En kaka är oftast det enda som skiljer en inloggad förfrågan från en anonym. Efter en inloggning är sessionsidentifieraren i kakan användaren, så långt servern kan se: den som visar upp den får kontot, och inget lösenord efterfrågas. Det mesta som går fel med kakor handlar om att någon annan läser, kopierar eller använder en.

Webbläsaren gör det sista av de tre enkelt. Den bifogar kakor till varje förfrågan som de matchar, oavsett vem som satte i gång förfrågan. Ett formulär på en annan webbplats som postar till en webbutik kommer fram med besökarens kakor, och servern ser en förfrågan från en inloggad användare. RFC 6265 kallar detta ”ambient authority”, en behörighet som följer med av sig själv, och namnger attackerna som den bjuder in till: cross-site request forgery, alltså förfalskade förfrågningar mellan webbplatser, och ”confused deputy”, den förvirrade ställföreträdaren.

Kakor är också webbens äldsta verktyg för spårning. Ett skript som är inbäddat på tusentals webbplatser kan sätta en kaka från sin egen domän, och webbläsaren skickar tillbaka den från var och en av dem, så att ett enda företag kan foga ihop en persons besök på dem alla. Revisionen beskriver den här användningen av tredjepartskakor, noterar att de flesta webbläsare numera blockerar eller partitionerar dem och rekommenderar en policy som är så restriktiv som kompatibiliteten tillåter.

Lagen följer lagringen snarare än tekniken. I EU tillåter artikel 5.3 i ePrivacydirektivet att en webbplats lagrar information på en besökares enhet, eller läser det som redan är lagrat där, bara med ett samtycke som har lämnats efter klar och fullständig information. Undantagen är lagring som enbart används för att överföra en kommunikation, och lagring som är absolut nödvändig för en tjänst som besökaren uttryckligen har bett om. Det är den artikeln som samtyckesrutorna kommer från, och den nämner aldrig kakor vid namn. I svensk rätt finns motsvarande regel, med samma två undantag, i 9 kap. 28 § lagen om elektronisk kommunikation. Europeiska dataskyddsstyrelsens riktlinjer om artikelns tekniska räckvidd tillämpar den även på spårningspixlar och spårningslänkar.

Så ser det ut i praktiken

Den här webbplatsen sätter inga. Dess sida om kakor förklarar varför: det finns inga konton, ingen varukorg och ingenting att följa en besökare med, så det finns ingen samtyckesruta, eftersom det inte finns något att samtycka till. Sidan räknar också upp de andra ställen där en webbplats kan lämna något efter sig, från `localStorage` till en service worker, och säger att inget av dem används heller.

Att klara sig helt utan kakor ändrar hur en webbplats lär sig något om sina läsare. Räkningen av läsare enbart ur åtkomstloggen mäter läsekretsen utan kaka, skript eller pixel, ur webbserverns egen logg, och stryker `Cookie` -headern i förfrågan från varje rad medan raden skrivs, tillsammans med authorization-headern. Resultatet är grovt med avsikt: unika besökare och en besökares väg genom webbplatsen går inte att nå, och sidan skriver ut de begränsningarna bredvid siffrorna.

Där en applikation faktiskt behöver en kaka är den användbara frågan vad kakan anförtros. På en plattform där en person agerar för flera organisationer hålls den aktiva organisationen i webbläsaren, och en kaka på serversidan speglar den, så att sidor som renderas på servern visar samma kontext. Ingen av kopiorna avgör

åtkomsten. Servern kontrollerar behörigheten vid varje förfrågan, så den som ändrar värdet ändrar vad den egna skärmen visar och ingenting mer. Det är den säkra formen för varje kaka: värdet får välja vad som ska visas, och beslutet om vad någon får göra stannar på servern.

Var det går fel

Ett skript på sidan läser sessionen. Ett skript som en angripare har lyckats injicera kan läsa `document.cookie` och skicka sessionsidentifieraren vart som helst, om kakan saknar `HttpOnly`. `HttpOnly` tar bort kakan från det gränssnittet. Det stoppar stöld men inte missbruk: det injicerade skriptet kan fortfarande skicka förfrågningar, och webbläsaren bifogar fortfarande kakan, så `HttpOnly` står bredvid en Content Security Policy snarare än att ersätta den.

Kakan färdas över okrypterad HTTP. En kaka som saknar `Secure` följer med varje `http://`-förfrågan till sin värddator, läsbar för alla längs nätverksvägen. Enligt RFC 6265 skyddar `Secure` bara konfidentialiteten, och en angripare i nätverket kan fortfarande skriva över en säker kaka från en osäker anslutning. Revisionen krymper det glappet genom att ignorera `Secure`-kakor som sätts över ren HTTP och vägra en kaka från ren HTTP som skulle lägga sig över en säker.

En underdomän bredvid sätter kakan. `Domain=example.com` skickar en kaka till varje värddator under den domänen, och var och en av dem kan sätta en kaka för hela domänen, som huvudwebbplatsen sedan läser som sin egen. `Path` är inte heller någon säkerhetsgräns. Prefixet `__Host-` är svaret: en webbläsare godtar en sådan kaka bara när den har `Secure`, har `Path=/` och saknar `Domain`, vilket låser den till den enda värddator som satte den.

En annan webbplats startar förfrågan. `SameSite=Strict` håller kakan till förfrågningar som görs från samma webbplats, `Lax` lägger till navigeringar på toppnivå med säkra metoder, som att följa en länk, och `None` skickar den överallt och ignoreras om kakan inte också har `Secure`. Revisionen behandlar en kaka utan `SameSite` ungefär som `Lax`; en server som förlitar sig på det bör säga det uttryckligen. En webbplats är i det här sammanhanget adressens protokolldel plus den registrerbara domänen, så `SameSite` gör ingenting mot en fientlig underdomän bredvid.

Servern litar på värdet. Allt i en kaka kan redigeras av den person vars webbläsare håller den. RFC 6265 rekommenderar att servrar krypterar och signerar innehållet i kakor och varnar för att kakan ändå kan spelas upp på nytt. Det stabilare mönstret är en slumpmässig sessionsidentifierare som pekar på tillstånd som hålls på servern och byts ut vid inloggning, så att en identifierare som en angripare har planterat i förväg, den session fixation som RFC:n varnar för, inte betyder någonting efteråt.

Webbläsaren släpper kakan. Ett utgångsdatum är ett maximum. Webbläsare rensar bort kakor för att hålla sig inom sina gränser, och en angripare kan tränga ut kakor genom att sätta många egna. Servrar bör inte räkna med att webbläsare behåller kakor. Även borttagningen har en fälla: en `Set-Cookie` med ett utgångsdatum i det förflutna tar bort kakan bara när dess `Path` och `Domain` stämmer med dem som kakan skapades med.

Läs mer

- RFC 6265: HTTP State Management Mechanism — standarden för `Set-Cookie` och `Cookie`, med reglerna för attributen och ett avsnitt om skydd som tar upp ambient authority, svag integritet och sessionsidentifierare.
- Cookies: HTTP State Management Mechanism (rfc6265bis) — den godkända revisionen, med `SameSite`, prefixen `__Secure-` och `__Host-`, taket på 400 dagar och ett avsnitt om tredjepartskakor.
- Directive 2002/58/EC, consolidated text — ePrivacydirektivet i dess lydelse efter ändringarna 2009, med artikel 5.3 om att lagra och läsa information på en användares enhet.
- EDPB Guidelines 2/2023 on the technical scope of Article 5(3) — hur regeln når spårningspixlar, spårningslänkar, lokal behandling och andra tekniker bortom kakor.
- Lag (2022:482) om elektronisk kommunikation — den svenska lagen, där 9 kap. 28 § är motsvarigheten till artikel 5.3, med samma krav på information och samtycke och samma två undantag.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är en kaka?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/cookie/