Engineer

Subresource Integrity

ett hashvärde som en webbläsare kontrollerar en fil mot

Vad det är

Subresource Integrity låter en sida i förväg ange exakt vilka byte ett skript eller en stilmall måste innehålla, och får webbläsaren att kasta filen om det som kommer fram är något annat. Namnet förkortas oftast SRI. Sidan skriver ett kryptografiskt hashvärde av filen i ett `integrity`-attribut på det `script`- eller `link`-element som läser in den. När svaret kommer kör webbläsaren samma hashfunktion över de byte den tog emot och jämför resultatet med hashvärdet på elementet. Om de är lika körs filen, eller formar sidan, som vanligt. Om de skiljer sig behandlar webbläsaren svaret som ett nätverksfel: skriptet körs aldrig och stilmallen tillämpas aldrig.

Värdet består av namnet på en hashfunktion, ett bindestreck och hashvärdet i base64. Ett enda kommando över den färdiga filen tar fram det, och att köra det två gånger visar hur lite som krävs för att ändra svaret:

```
$ printf "alert('Hello, world.');" > hello.js
$ openssl dgst -sha384 -binary hello.js | openssl base64 -A
H8BRh8j48O9oYatfu5AZzq6A9RINhZO5H16dQZngK7T62em8MUt1FLm52t+eX6xO
$ echo >> hello.js
$ openssl dgst -sha384 -binary hello.js | openssl base64 -A
qu8P7hCLFcBpas8BOnGmaANCkfXJIpFttDEaxOj0Pq9H6Q8NYjsMb+PVqwuJygVe
```

Med `sha384-` framför är den första utskriften attributets värde för det enradiga skriptet. En enda radbrytning som läggs till på slutet ger ett hashvärde som inte har något gemensamt med det första, och en webbläsare som har det första skulle vägra den andra filen.

Specifikationen känner igen tre hashfunktioner, SHA-256, SHA-384 och SHA-512, alla från familjen SHA-2, och kallar SHA-384 en bra utgångspunkt vid tiden då den skrevs. Ett attribut får lista flera hashvärden åtskilda av mellanslag. Webbläsaren väljer den starkaste funktionen bland dem och godtar filen när den stämmer med något av de hashvärden som gjorts med den funktionen. Det låter en sida gå över till en starkare algoritm, eller godta två versioner av ett bibliotek under ett byte, utan ett fönster där sidan bara bär felaktiga hashvärden.

Attributet finns på `script` och `link`, och samma värde kan följa med i ett `Link`-svarshuvud som läser in en fil i förväg. Specifikationens nuvarande text är dess andra utgåva, ett W3C Working Draft från Web Application Security Working Group; den första utgåvan blev en W3C Recommendation 2016.

Använd begreppet i ett beslut

Använd ”Vad är Subresource Integrity?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Konton, lösenord och behörigheter ordnade så att alla har exakt den åtkomst deras arbete kräver, och de som slutar förlorar den samma dag.

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. Skrev 1 000 RBAC-regler som säkrar Linux-servrar — 1,000 åtkomstregler, skrivna för att täppa till en säkerhetslucka

Skrev 1 000 RBAC-regler som säkrar Linux-servrar Hör av er →

Varför det spelar roll

En sida som läser in ett bibliotek från ett CDN, eller från någon värd som dess ägare inte driver, ger den värden makt att köra kod på sidan. HTTPS skyddar filen på vägen över nätverket, men säger ingenting om vad värden valde att skicka från början. Om någon bryter sig in i lagringen hos ett CDN, tar över ett konto som publicerar dit eller i tysthet byter ut filen bakom en välbekant adress, kör varje sida som inkluderar den den nya koden med sidans fulla befogenheter. Sidans egen kod har inget sätt att se skillnaden.

SRI flyttar tillbaka det beslutet till sidan. Hashvärdet räknas fram över en fil som webbplatsens ägare har läst eller byggt, och det följer med inuti sidan, från webbplatsens egen server. En värd som skickar något annat får sin fil avvisad i stället för körd. Sidan förlorar en funktion tills någon tittar efter, ett synligt fel med en uppenbar orsak, i stället för att i tysthet köra någon annans kod.

Hashvärdet låser också en version. En adress som `/lib/latest.js` kan när som helst börja leverera en ny version under en sida; hashvärdet kan inte följa med. En uppgradering som ägaren aldrig valde blir ett avvisande på ett bestämt element, rapporterat genom en felhändelse.

Det lönar sig att vara exakt om vad SRI inte gör. Det kontrollerar en fil mot ett hashvärde; det bedömer inte om filen är säker. Ett hashvärde av ett bibliotek som redan innehåller en fientlig rad skyddar den fientliga raden troget. Det lämnar inbäddade skript i fred, vilket är uppgiften för en Content Security Policy. Och det litar på sidan som bär hashvärdet, så sidan själv måste fortfarande komma fram oskadd.

Så ser det ut i praktiken

Den här webbplatsen använder det på sin egen stilmall. Säkerhetssidan listar Subresource Integrity på stilmallen: paketet får ett fingeravtryck och en hash när webbplatsen byggs, så att läsarens webbläsare kan kontrollera att det kom fram oförändrat. Ingen körbar JavaScript publiceras och inga anrop går till tredje part. En läsare som räknar värdarna i webbläsarens nätverkspanel hittar en enda, webbplatsens egen.

Kolofonen beskriver samma fil från byggets sida, som ett enda CSS-paket som sammanfogas och får ett fingeravtryck när webbplatsen byggs och levereras med en SRI-hash. Fingeravtrycket lägger in en hash av filens innehåll i dess namn, så att en ändrad stilmall får en ny adress och en webbläsare aldrig tar en gammal kopia i cachen för den nuvarande, medan integritetsvärdet låter webbläsaren kontrollera de byte som finns bakom adressen. Bygget tar fram båda ur samma byte i samma steg, så varje release levererar stilmallen och dess hashvärde tillsammans, och det vanligaste felet som beskrivs nedan, en fil som ändras efter att den hashades, har mycket litet utrymme att uppstå.

Var det går fel

Något mellan bygget och webbläsaren skriver om filen. Ett CDN som minifierar skript, en proxy som injicerar något i svarsinnehållet eller ett steg i driftsättningen som skriver om en fil efter att hashvärdet togs ändrar innehållets byte, och webbläsaren avvisar resultatet fast ingen har angripit något. Omkomprimering med gzip eller Brotli som bevarar innehållet orsakar inte den avvikelsen: kontrollen använder det avkodade svarsinnehållet, inte de komprimerade byte som skickas över nätverket. Specifikationen lägger ansvaret

på mellanhänder som ändrar innehållet: de måste hålla hashvärdet i takt med det de levererar, eller leverera originalet orört, och ursprungsservern bör skicka `Cache-Control: no-transform` för att be dem lämna filen i fred. Den pålitliga ordningen är att ta hashvärdet sist, över exakt de byte i det avkodade innehållet som webbläsaren kommer att kontrollera.

En fil från ett annat ursprung kommer fram utan CORS. Specifikationen kräver att en integritetskontroll av en fil från ett annat ursprung använder CORS, och kallar ett försök utan det för ett logiskt fel. Elementet kan till exempel använda `crossorigin="anonymous"`, och värden måste svara med ett HTTP-huvud som tillåter delning av filen. `crossorigin="use-credentials"` använder också CORS, med serverhuvuden som uttryckligen tillåter förfrågans ursprung och användning av autentiseringsuppgifter. Utan en CORS-förfrågan och en motsvarande serverpolicy blockerar webbläsaren filen i stället för att läsa in den okontrollerad. Regeln skyddar andra webbplatser: utan den skulle en angripare kunna läsa in ett privat svar med ett gissat hashvärde och av utfallet lära sig t.ex. om läsaren är inloggad där.

Hashvärdet kommer fram över okrypterad HTTP. SRI skyddar en sida mot en komprometterad filvärd, aldrig mot någon på vägen genom nätverket. På en sida som levereras utan HTTPS kan den som kan skriva om filen under överföringen lika gärna skriva om eller ta bort hashvärdet i sidan, och därför rekommenderar specifikationen att metadata för integritet bara levereras till en säker kontext.

Hashfunktionen är en som webbläsaren hoppar över. Specifikationen känner bara igen SHA-256, SHA-384 och SHA-512 och rekommenderar att webbläsare vägrar kända svaga funktioner som MD5 och SHA-1. Överraskningen är vad som händer med en token utanför den listan: webbläsaren ignorerar den, och ett attribut som bara består av ignorerade token räknas som om det inte bar någon metadata alls, så filen läses in okontrollerad. Ett byggverktyg som kan ta fingeravtryck med MD5, så som verktyget i Hugo kan, gör det lätt att leverera det av misstag.

En avvisad fil drar med sig sidan. När kontrollen fallerar avfyrar webbläsaren en felhändelse på elementet, och skriptet eller stilmallen saknas helt enkelt. För ett bibliotek som sidan inte kan köras utan föreslår specifikationen att fånga händelsen och läsa in en reservkopia från en betrodd, om än långsammare, källa som webbplatsens egen server. Utan en sådan fungerar skyddet precis som det är tänkt, och läsaren får ändå en trasig sida.

Kontrollen gäller bara de element som bär ett hashvärde. Ett integritetsattribut på ett skript säger ingenting om nästa skript som någon lägger till utan det, och att granska de skript som har ett kommer aldrig att avslöja det som saknar det. Utkastet lägger till ett svarshuvud som täcker det från serversidan. `Integrity-Policy` med `blocked-destinations=(script)` blockerar varje externt skript som kommer fram utan metadata för integritet, och `Integrity-Policy-Report-Only` listar dem först, så att de kan rättas innan något blockeras.

Läs mer

- Subresource Integrity — det nuvarande utkastet, med algoritmen för verifiering, huvudena `Integrity-Policy` och noterna om proxyservrar, okrypterad HTTP och läckor mellan ursprung.
- Subresource Integrity, W3C Recommendation 2016 — den första utgåvan, den stabila rekommendation som utkastet växte fram ur.
- Fetch Standard — där en begärans metadata för integritet kontrolleras mot svarets innehåll, och CORS-protokollet som en kontroll mellan ursprung bygger på.
- Hugo: resources.Fingerprint — hur en statisk webbplatsgenerator tar ett fingeravtryck av en fil när webbplatsen byggs och lämnar tillbaka det matchande integritetsvärdet.
- FIPS 180-4, Secure Hash Standard — den federala standarden i USA som definierar SHA-256, SHA-384 och SHA-512.

Ordlista

Välj utifrån era faktiska förutsättningar

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är Subresource Integrity?” 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/sri/