Engineer

Proxy

en server, der sender en anmodning videre for dig

Hvad det er

En proxy er et program, der står mellem en klient og den server, klienten vil nå. Den tager imod klientens anmodning, sender sin egen anmodning videre til næste maskine i rækken og giver svaret tilbage. Klienten taler med proxyen, og proxyen taler med serveren, så de to ender deler aldrig en forbindelse. Der findes flere slags proxyer, og det nyttige spørgsmål at stille til hver af dem er, hvem der satte den der.

HTTP-specifikationen, RFC 9110, nævner tre almindelige mellemled (intermediaries). En proxy i snæver forstand er en, klienten selv har valgt, som regel via lokal konfiguration som en browserindstilling eller en miljøvariabel. En gateway, som RFC'en også kalder en omvendt proxy (reverse proxy), står på serverens side: over for den, der forbinder, opfører den sig som oprindelsesserveren, mens den sender hver anmodning indad til én eller flere servere bag sig. En tunnel videresender bytes blindt mellem to forbindelser og holder op med at være part i samtalen, når den først er åben. En klient beder om en tunnel med metoden CONNECT, angiver vært og port og kører så TLS igennem den fra ende til ende. RFC'en nævner også en fjerde slags, som ingen har valgt: den opsnappende proxy, ofte kaldet en transparent proxy, der omdirigerer udgående trafik på port 80 i kanten af et netværk, sådan som offentlige adgangspunkter og virksomheders firewalls gør.

Hvert hop er en separat forbindelse. RFC'ens diagram placerer tre mellemled mellem en browser og en oprindelsesserver, hvilket giver fire forbindelser, og indstillinger som keep-alive eller en timeout gælder for én af dem ad gangen. HTTP kræver også, at en proxy melder sig: den skal tilføje et Via-headerfelt til hver besked, den videresender, og det er sådan, en løkke eller et uventet mellemled viser sig i et spor.

HTTP er kun ét lag. En SOCKS-proxy, hvis version 5 er specificeret i RFC 1928 fra marts 1996, videresender TCP-forbindelser og UDP-datagrammer uden at læse, hvad der er i dem, så enhver protokol kan bruge den. Tor er et velkendt eksempel: som standard lytter den på port 9050 efter SOCKS-forbindelser fra lokale programmer.

Brug begrebet i en beslutning

Brug »Hvad er en proxy?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Wi-Fi, firewalls og fjernadgang sat op, så medarbejdere kan arbejde hvor som helst, og kun de rette kommer ind.

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 administreret 30 Ubuntu Linux VPS-instanser — 30 Linux-servere driftet, med en plan for gendannelse

Har administreret 30 Ubuntu Linux VPS-instanser Kontakt os →

Hvorfor det betyder noget

Den, der driver proxyen, kan læse trafikken. Ved almindelig HTTP gælder det hver header og hver body. Gennem en CONNECT-tunnel ser den værtsnavn og port, og resten er TLS fra klienten til serveren. En omvendt proxy, der terminerer TLS, ser alt i klartekst, og det er dens opgave, for den skal læse en anmodning for at kunne route den. At sætte en anden virksomheds omvendte proxy foran et website, hvilket er præcis, hvad et content delivery network gør, giver altså det firma adressen på hver besøgende og hver URL, før sitet selv ser dem. Det kan være den rigtige byttehandel for en webbutik under angreb. Det er stadig en beslutning om privatliv, og privatlivssiden bør sige det.

Den anden konsekvens er, hvad serveren får at vide. RFC 7239 beskriver tabet ligeud: bag en proxy ser anmodninger ud til at komme fra proxyens adresse. En log, en hastighedsgrænse, en spærreliste eller en tilladelsesliste, der bygger på klientens adresse, ser så én adresse for alle. Den sædvanlige reparation er en header, der bærer den oprindelige adresse videre: standardfeltet Forwarded fra RFC 7239 eller det ældre X-Forwarded-For. Hvilken kopi af den header man skal tro på, er der, hvor de fleste problemer begynder.

Den tredje er, at en proxy ændrer, hvad en fejl betyder. En 502 Bad Gateway eller en 504 Gateway Timeout er skrevet af et mellemled om serveren bag det: proxyen fik et ugyldigt svar eller slet intet svar i tide. En 407 betyder, at proxyen selv vil have adgangsoplysninger, før den videresender noget. At vide, hvilken maskine der skrev fejlen, er halvdelen af diagnosen.

Den fjerde er, at mange proxyer cacher. RFC 9111 kalder en cache, der gemmer svar til mere end én bruger, en delt cache, som regel en del af et mellemled. Den gør et site hurtigere og billigere at drive, og den er også det ene sted, hvor et svar ment til én person kan nå en anden.

Sådan ser det ud i praksis

Tre stykker af virksomhedens arbejde drejer sig om en proxy: én fjernet, én drevet af fremmede og én slået fra.

I august 2026 flyttede virksomheden sin DNS-zone på omkring tyve poster ind i repositoryet og afstemte den mod Cloudflares API. En del af arbejdet var at lade webtrafikken gå gennem udbyderens content-netværk som omvendt proxy, og en fungerende CDN-proxy foran sitet blev pillet ned igen for privatlivets skyld. Indvendingen gjaldt dens placering: hver læsers adresse og hver forespurgt sti ville nå en tredjepart først og blive behandlet efter dens regler, og for en virksomhed, der fortæller sine læsere, at ingen kigger med, vejede det tungere end den beskyttelse mod angreb, den gav. Beslutningen og dens begrundelse er nedfældet, så ingen genåbner den ved et uheld.

Måneden efter begyndte virksomheden at måle læsertal ud fra webserverens egen log, og så måtte den forholde sig til proxyer, den ikke selv driver. Adgangsloggen kasserer syv headere, der videregiver en klients adresse, før en linje skrives. Loggens encoder kopierer headerne uændret, så en læser bag en virksomhedsproxy ellers ville have fået en umaskeret adresse gemt under et headernavn, maskeringsreglen ikke omfattede.

Da sitet samme måned blev udgivet som Tor-onion-spejl, blev Tor-dæmonens egen proxyport til udgående trafik deaktiveret i konfigurationen. Onion-tjenesten nås gennem kredsløb, som dæmonen opretter udad, så udgivelsen af dette spejl kræver ingen særskilt indgående internetport. Når dens SOCKS-lytteport er deaktiveret, tilbyder denne Tor-dæmon ikke lokale programmer en udgående proxy.

Hvor det går galt

Hver anmodning i loggen kommer fra den samme adresse. Applikationen står bag en omvendt proxy og registrerer adressen på den forbindelse, den modtog, og det er proxyens. En hastighedsgrænse bremser så alle på én gang, og en spærring skrevet af et indtrængningsværktøj spærrer proxyen, altså hele

publikummet. Lad proxyen give klientens adresse videre i en header, og sæt applikationen op til kun at læse den header på forbindelser fra proxyens egen adresse. I nginx er det realip-modulets direktiv set_real_ip_from; de fleste webframeworks har en tilsvarende indstilling for betroede proxyer.

En klient vælger sin egen adresse ved at sende en header. Applikationen læser X-Forwarded-For på hver anmodning, også på anmodninger, der nåede den direkte, og på anmodninger, der ankom med en forfalsket værdi allerede i headeren. RFC 7239 siger klart, at feltet kan ændres af hver knude på vejen, klienten medregnet. Stol kun på de poster, din egen proxy tilføjede, talt fra højre, og bind applikationen til loopback eller et privat netværk, så proxyen er den eneste vej ind.

Loggen, der skulle anonymiseres, gemmer fulde adresser. Klientens adresse er maskeret i det felt, nogen kontrollerede, men den samme adresse rejser i Forwarded, X-Forwarded-For, X-Real-IP og lignende headere, og en logger, der skriver headerkortet, som det blev modtaget, gemmer dem i fuld længde. Skriv alle headere op, der kan bære en adresse, fjern eller masker hver af dem i det øjeblik, linjen skrives, og send så en testanmodning og læs linjen på disken.

Sider giver 502 eller 504, mens backendens egen log er stille. Fejlen er skrevet af proxyen. En 502 betyder, at den fik et ugyldigt svar: backenden var nede, lyttede på en anden port eller talte almindelig HTTP, hvor proxyen ventede TLS, eller omvendt. En 504 betyder, at backenden brugte længere tid end proxyens læse-timeout, som i nginx som standard er 60 sekunder. Læs proxyens fejllog først, for den nævner den upstream-adresse, den prøvede, og hæv kun en timeout for de ruter, man ved er langsomme.

En bruger får vist en anden brugers side. En delt cache i en proxy eller et content-netværk gemte et svar, der var personligt, for eksempel en kontoside sendt sammen med en sessionscookie, og serverede det til den næste, der bad om samme URL. Mærk hvert svar pr. bruger med Cache-Control: private, som RFC 9111 siger, en delt cache ikke må gemme, eller med no-store, hvor ingen kopi bør gemmes nogen steder, og kontrollér de headere, proxyen faktisk returnerer.

Læs mere

- RFC 9110, section 3.7: Intermediaries — definitionerne af proxy, gateway, tunnel og opsnappende proxy.

- RFC 9110, section 9.3.6: CONNECT — hvordan en klient beder en proxy om at åbne en tunnel til en vært og port.
- RFC 7239: Forwarded HTTP Extension — standardheaderen for klientens oprindelige adresse og dens forbehold.
- RFC 9111: HTTP Caching — delte og private cacher og de direktiver, der holder dem adskilt.
- RFC 1928: SOCKS Protocol Version 5 — den protokolneutrale proxy, der videresender TCP-forbindelser og UDP.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er en proxy?« 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/proxy/