Engineer

Omvänd proxy

← en server som tar emot trafik för en webbtjänst

Vad det är

En omvänd proxy tar emot förfrågningar åt en webbtjänst. Webbläsaren ansluter till webbplatsens offentliga adress, där proxyn lyssnar. Proxyn väljer var förfrågan ska hamna, skickar vidare den och förmedlar svaret från tjänsten. Servern bakom proxyn kallas ursprungsserver. Den kan få en ny intern adress utan att besökarens bokmärken behöver ändras. HTTP-standarden använder ordet gateway för rollen. En vanlig framåtriktad proxy arbetar åt klienten och kontaktar andra tjänster för klientens räkning.

En webbplats kan till exempel hämta startsidan från en tjänst och rapporten på `/api/rapporter/42` från en annan. Båda nås genom samma offentliga domännamn. Proxyn kan läsa sökvägen i HTTP-förfrågan och använda den för sitt val. Om flera servrar kan svara på samma förfrågan kan samma produkt också fördela belastningen mellan dem. En omvänd proxy kan lika gärna betjäna en enda ursprungsserver. Håll gärna isär beslutet om vart en viss sökväg skickas och beslutet om vilken av flera likvärdiga servrar som får arbetet.

Vissa produkter kan föra vidare en krypterad TCP-anslutning till tjänsten bakom. Vägvalet kan då bygga på anslutningsuppgifter eller synliga fält i TLS-förhandlingen, när sådana fält finns tillgängliga. För att kunna välja med ledning av en krypterad HTTP-sökväg måste komponenten först få tillgång till HTTP-meddelandet. Rita därför in var TLS avslutas och om nästa förbindelse använder en ny TLS-session. Då blir det tydligt vilka uppgifter de olika delarna faktiskt kan läsa.

Använd begreppet i ett beslut

Använd ”Vad är en omvänd proxy?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Wi-Fi, brandväggar och fjärråtkomst uppsatta så att personalen kan arbeta varifrån som helst och bara rätt personer kommer in.

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. Förvaltade 30 VPS-instanser med Ubuntu Linux — 30 förvaltade Linux-servrar, med en plan för återställning

Förvaltade 30 VPS-instanser med Ubuntu Linux Hör av er →

Varför det spelar roll

En gemensam ingång gör det enklare att beskriva vilka tjänster som nås utifrån. Proxyn kan hantera certifikatet för det offentliga namnet medan applikationerna ligger på ett privat nät. Den kan också skriva åtkomstloggar, begränsa förfrågningar och styra om gamla adresser. Utgå från vad varje väg behöver. En tjänst som tar emot filer behöver kanske en högre storleksgräns, medan en strömmande tjänst behöver skicka data vidare så snart de finns.

Den omvända proxyn blir en del av webbplatsens förtroendegräns. Där TLS avslutas kan operatören läsa HTTP-meddelandet, inklusive kakor och uppgifter som användaren skickar. Ny kryptering till ursprungsservern kan skydda den anslutningen under transport. Den som sköter proxyn har fortfarande tillgång till det som proxyn behandlar. Kom därför överens om vilka uppgifter operatören behöver, vilka som får läsa loggarna och när uppgifterna raderas.

Både proxyn och ursprungsservern kan skicka ett felsvar. En gateway kan returnera `502` när den får ett ogiltigt svar, eller `504` när den har väntat för länge. En applikation kan också sätta sin egen felstatus. Anteckna tiden och adressen för en kontrollerad förfrågan, och följ samma händelse genom båda loggarna. Om ursprungsservern aldrig såg förfrågan ger dess kod för just den sökvägen få ledtrådar.

Så ser det ut i praktiken

I arbetet med att stämma av DNS-zonen som kod beskriver Engineer ApS hur en fungerande CDN-proxy togs ur bruk. När den var i drift behandlade leverantörens nät varje besökares adress och efterfrågade URL före företaget. Företaget vägde integriteten för besökarna högre än angreppsskyddet proxyfunktionen gav. DNS-zonen kunde fortfarande versionshanteras och granskas. En leverantör kan erbjuda både DNS och proxy, men uppgifterna den får tillgång till beror på vilken funktion man väljer att använda.

Prova två anrop i en testmiljö, till exempel `/health` och `/api/rapporter/42?vy=kort`. Anteckna vilken metod och sökväg som kommer fram till applikationen. Kontrollera svaret i webbläsaren och läs värdnamn och protokoll om användaren

omdirigeras. Byt sedan den interna adressen för rapporttjänsten och upprepa testen med samma offentliga URL. Då får du se att de adresser som användaren har fått fortsätter att fungera när tjänsten flyttas.

För att prova en delad cache kan du använda två separata sessioner där varje användare får en egen, ofarlig markör i svaret. Kontrollera att den ena sessionen alltid får sitt eget innehåll. Stäng sedan ursprungsservern i testmiljön och se vad webbläsaren och loggarna visar under felet. Ta med väntetiden. Det är lättare att avgöra vilket led som behöver åtgärdas när dess felsignaler är kända i förväg.

Var det går fel

Klienter kan själva skicka ett fält som heter `Forwarded` eller `X-Forwarded-For`. Bestäm vilken proxykedja applikationen får lita på och vilka värden som ska rensas bort eller ersättas när förfrågan anländer. Ursprungsservern bör bara acceptera anslutningar från avsedda motparter. Skicka ett anrop med en avsiktligt felaktig klientadress och läs både det applikationen loggar och det värde en åtkomstregel faktiskt använder.

Ett inloggningsflöde kan avslöja att applikationen har använt det interna värdnamnet eller valt HTTP trots att besökarens sida öppnades med HTTPS. Ge applikationen en verifierad uppgift om den offentliga adressen. Följ länkarna tillbaka från inloggningen och kontrollera omdirigeringar och kakor. Prova en djup sökväg med parametrar om proxyn lägger till eller tar bort ett prefix, så att testet täcker mer än startsidan.

Ett nytt försök efter ett uteblivet svar kan skapa ett nytt problem. Applikationen kan redan ha bokfört en betalning när proxyn slutar vänta. Beskriv vad som händer vid upprepade anrop innan du låter proxyn skicka en sådan förfrågan igen. En kontrollerad försening efter en genomförd åtgärd visar vem som avbryter väntan och om teamet kan avgöra vilken begäran som faktiskt utförde arbetet.

Läs mer

- RFC 9110 om mellanhänder i HTTP definierar rollerna proxy, gateway och tunnel.
- RFC 7239 om Forwarded beskriver fältet samt risker för säkerhet och integritet.
- NGINX HTTP-proxymodul tar upp vägval, fält, buffring och tidsgränser.

- Caddys reverse_proxy-direktiv förklarar serverval och vilka uppgifter som förs vidare.
- Apache HTTP Servers vägledning visar hur den offentliga servern skickar trafik vidare till tjänsterna bakom den.

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 en omvänd proxy?” 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/reverse-proxy/