Engineer

NAT

🔀 omskrivning av IP-adresser när paket passerar en router

Vad det är

NAT, kort för nätverksadressöversättning (network address translation), är omskrivning av IP-adresser i paket medan de passerar en router, så att värdar som är numrerade i en adressrymd kan tala med värdar som är numrerade i en annan. RFC 2663, dokumentet från 1999 som slog fast begreppen, beskriver det som att avbilda IP-adresser från ett adressområde (realm) till ett annat, med transparent routning som mål: värdarna på vardera sidan ska inte behöva veta att det sker.

Den form som nästan alla möter är det som RFC 3022, publicerad i januari 2001, kallar traditionell NAT (Traditional NAT). Typfallet är ett litet kontor eller ett hem med en enda IP-adress från leverantören, medan maskinerna bakom routern använder privata adresser från RFC 1918-blocken 10.0.0.0/8, 172.16.0.0/12 och 192.168.0.0/16, som inte routas mellan organisationer. Routern skriver om källadressen i varje utgående paket till sin egen publika adress och vänder tillbaka ändringen på svaren. Basic NAT översätter bara adresser, så bara lika många interna värdar kan vara ute samtidigt som det finns publika adresser i adresspoolen. NAPT, network address port translation, skriver också om TCP- eller UDP-porten eller ICMP-identifieraren, och det låter många interna värdar dela en publik adress. RFC 3022 nämner att Linux erbjöd NAT under namnet IP masquerade, och brandväggsverktygen använder fortfarande ordet.

När en adress ändras blir allt som har räknats fram ur den fel, så routern måste laga det också. RFC 3022 kräver att kontrollsumman i IP-huvudet justeras, och TCP- och UDP-kontrollsummorna med, eftersom de täcker ett pseudohuvud som innehåller käll- och måladressen. För att skicka varje svar till rätt intern värd för routern en tabell över översättningar: intern adress och port på ena sidan, extern adress och port på den andra. Den tabellen är tillstånd som hålls mitt i nätet, och det mesta som går fel med NAT kommer därifrån.

Traditionell NAT fungerar i en riktning. RFC 3022 beskriver sessionerna som utgående från det privata nätet, medan trafik åt andra hållet bara släpps igenom via statiska avbildningar för värdar som valts i förväg. Idén används nu också en nivå högre upp: RFC 6598 från april 2012 reserverade 100.64.0.0/10 för carrier-grade NAT, där blocket numrerar förbindelserna mellan en leverantörs NAT och kundernas utrustning, som själv kan översätta en gång till.

Använd begreppet i ett beslut

Använd ”Vad är NAT?” 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

RFC 3022 är öppen med avvägningen. NAT kan installeras utan att värdar eller routrar ändras, men det tar bort, med RFC:ns ord, ”IP-adressens betydelse från ände till ände”, och väger upp det med mer tillstånd i nätet. Nästan varje praktisk följd av NAT går tillbaka på den meningen.

För en server är källadressen för en anslutning ofta inte en maskin utan en router med en hel skara bakom sig. En spärr eller en hastighetsbegränsning riktad mot en besökare träffar alla som delar adressen, och en missbruksanmälan pekar ut routern, inte den skyldige. RFC 3022 säger samma sak inifrån: att dölja privata adresser, vilket den räknar som en fördel för integriteten, gör också felsökning svårare, eftersom en värd som angriper andra maskiner eller skickar skräppost har sin adress dold i NAT-routern.

Tillstånd betyder också tidsgränser. En översättning lever bara så länge routern minns den. RFC 4787, en Best Current Practice från januari 2007, säger att en UDP-översättning inte får löpa ut på under två minuter och rekommenderar fem minuter eller mer. RFC 5382, motsvarigheten för TCP från oktober 2008, sätter 2 timmar och 4 minuter för en inaktiv etablerad anslutning: keepalive-paket skickas som standard varannan timme, och de fyra minuterna täcker paket som fortfarande är på väg. Det här är minimikrav för enheter som följer dokumenten,

och alla enheter gör inte det. En anslutning som är tyst längre än enhetens egen tidsgräns förlorar sin översättning, och nästa paket utifrån har ingenstans att ta vägen.

Två ytterligare begränsningar följer av detta. RFC 3022 gör det obligatoriskt att alla paket i en session passerar samma NAT-router, och där ett nät har mer än en utgång kallar den det mycket viktigt att varje NAT har samma översättningstabell. Och protokoll som bär adresser inne i själva innehållet behöver hjälp av NAT-enheten, något som bara fungerar så länge den kan läsa innehållet.

Så ser det ut i praktiken

På en Linux-router utför nftables NAT i kedjor av typen `nat`. Regeluppsättningen nedan är den här artikelns illustration, byggd på NAT-exemplen i manualen nft(8). Här står 198.51.100.0/24 för det interna nätet och 203.0.113.5 för routerns publika adress; båda kommer från dokumentationsblocken, och ett verkligt internt nät skulle normalt använda ett av RFC 1918-blocken.

```
add table nat
add chain nat prerouting { type nat hook prerouting priority dstnat; }
add chain nat postrouting { type nat hook postrouting priority srcnat; }
add rule nat postrouting oif eth0 ip saddr 198.51.100.0/24 snat to 203.0.113.5
add rule nat prerouting iif eth0 tcp dport 443 dnat to 198.51.100.20
```

Den första regeln är vardagsfallet: paket från det interna nätet som går ut genom eth0 får 203.0.113.5 som källa. För en dynamisk publik adress erbjuder manualen `masquerade`, som använder den adress det utgående gränssnittet har för tillfället. Den andra regeln är en statisk avbildning åt andra hållet: TCP-anslutningar som kommer in på eth0 till port 443 går vidare till 198.51.100.20 på insidan. Så få regler räcker eftersom bara det första paketet i en anslutning passerar en NAT-kedja; beslutet sparas i posten för anslutningsspårning, och varje senare paket skrivs om på samma sätt.

Den andra sidan av saken är en värd bakom en NAT som den inte själv styr, och WireGuard har en inställning för just det. I den här artikelns andra illustration håller en intern värd en tunnel öppen till en motpart utanför:

```
[Peer]
PublicKey = <den externa motpartens publika nyckel>
Endpoint = 192.0.2.44:51820
AllowedIPs = 192.0.2.128/25
PersistentKeepalive = 25
```

Manualen wg(8) beskriver `PersistentKeepalive` som ett intervall på 1 till 65 535 sekunder för att skicka ett autentiserat tomt paket, så att en tillståndsbaserad brandvägg eller en NAT-översättning förblir giltig. Den föreslår 25 sekunder för ett gränssnitt bakom NAT som sällan skickar något själv och kan ta emot trafik när som helst, och låter inställningen vara avstängd som standard. Motparten utanför behöver inget särskilt: dess `Endpoint` för den här värden uppdateras till den senaste källadressen och porten från korrekt autentiserade paket, och det är just dem som NAT:en tilldelade.

Tunnlar som de här är där NAT-problem ofta visar sig. Portföljen beskriver hur åtta VPN-anslutningar över WireGuard och IPsec förvaltades och felsöktes i Google Cloud och på Linux-system under åren 2022 till 2024. Sidan säger att tunnlarna var bräckliga och besvärliga att diagnostisera när de gick ned, och att problemen med routning och nyckelutbyte åtgärdades vid roten i stället för att slätas över med en omstart, varefter de återkommande anslutningsproblemen som brukade avbryta arbetet upphörde. Den säger inte om NAT spelade någon roll, så de två exemplen ovan hör till den här artikeln och inte till det projektet.

Var det går fel

Tysta anslutningar som slutar svara. En tunnel, en databasanslutning eller en push-kanal som inte skickar något på ett tag förlorar sin översättning och bryts utan att någon av ändarna har stängt den. Symptomet ser slumpmässigt ut, eftersom det beror på hur länge linjen var tyst. Skicka keepalive-paket inifrån med kortare mellanrum än den kortaste tidsgränsen på vägen, som `PersistentKeepalive` gör, och håll tidsgränserna på en NAT du själv driver på eller över minimikraven i RFC 4787 och RFC 5382. RFC 4787 kräver att utgående paket förnyar en översättning och varnar för att förnyelse från inkommande trafik låter en angripare hålla den vid liv för evigt, så det är den interna änden som bör stå för förnyelsen.

Att vänta sig att anslutningar kommer in. Traditionell NAT är utgående till sin konstruktion, så en tjänst på en intern värd går inte att nå förrän någon skriver en statisk avbildning för den. Lägg till en `dnat` -regel för exakt de portar tjänsten

behöver, eller vänd på riktningen och låt den interna värden ansluta utåt, som keepalive-exemplet ovan gör.

Saknad hairpinning. Två interna värdar som når varandra via den externa adressen till en av dem behöver att NAT:en vänder paketet. RFC 4787 kräver att NAT-enheter stöder sådan hairpinning och presenterar det vända paketet med den externa källadressen och porten. Testa den vägen medvetet, och där en enhet inte klarar det, låt de interna värdarna använda den interna adressen.

Adresser inne i innehållet. FTP är det klassiska fallet: RFC 2663 påpekar att kommandot `PORT` och svaret på `PASV` bär en IP-adress och en port som text, och en NAT som bara skriver om huvudena lämnar dem felaktiga. NAT-enheter lappar sådana protokoll med en Application Level Gateway (ALG), och RFC 3022 kallar den för FTP den mest spridda. En ALG måste läsa innehållet, så kryptering sätter den ur spel om inte gatewayen har nyckeln, och RFC 3022 tillägger att IPsec från ände till ände inte kan förutsättas genom en NAT. Välj protokoll som håller adresser utanför innehållet; där ett gammalt protokoll måste passera en NAT du driver visar manualen nft(8) en hjälpare för anslutningsspårning av FTP.

Loggar som bara pekar ut routern. Bakom NAT registrerar en servers åtkomstlogg den delade adressen, som ensam inte kan peka ut en enskild intern värd. Logga källporten och en exakt tidsstämpel bredvid adressen och håll klockorna synkroniserade, så att ett klagomål på en anslutning kan matchas mot översättningen i routern.

Fragment under NAPT. Bara det första fragmentet av ett paket bär TCP- eller UDP-huvudet, så RFC 3022 varnar för att översättning av utgående fragment med NAPT är dömd att misslyckas, och att två interna värdar som skickar fragment med samma identifierare till samma mål kan förstöra båda sessionerna. Se till att paket är små nog att ta sig hela vägen i ett stycke.

Att behandla NAT som en brandvägg. Att NAT bara fungerar i en riktning är lätt att ta för skydd. RFC 4787 medger att NAT ofta införs för att nå säkerhetsmål, men säger rakt ut att det sätt en NAT avbildar på inte gör någon skillnad för säkerheten, som helt bestäms av filtreringen: vilka paket den släpper in. Skriv uttryckliga filterregler för vad som får komma in, och para ihop varje `dnat` -regel med en regel som bara tillåter den trafik den var avsedd för.

Läs mer

- RFC 3022: Traditional IP Network Address Translator (Traditional NAT) — beskrivningen från 2001 av Basic NAT och NAPT, med en ärlig lista över vad de kostar.
- RFC 2663: IP Network Address Translator (NAT) Terminology and Considerations — begreppen kring NAT, från adressområden till transparent routning och ALG:er.
- RFC 4787: Network Address Translation (NAT) Behavioral Requirements for Unicast UDP — hur en NAT bör avbilda, låta översättningar löpa ut och hantera hairpinning för UDP-trafik.
- RFC 5382: NAT Behavioral Requirements for TCP — motsvarigheten för TCP, med tidsgränserna för etablerade anslutningar och anslutningar som stängs.
- wg(8): set and retrieve configuration of WireGuard interfaces — WireGuard-manualen, med `PersistentKeepalive` och rådet för värdar bakom NAT.
- nft(8): nftables manual — referensen för nftables, med NAT-kedjor och satserna `snat`, `dnat` och `masquerade`.

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 NAT?” 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/nat/