Engineer

NAT

🔀 omskriving av IP-adresser når pakker passerer en ruter

Hva det er

NAT, kort for nettverksadresseoversettelse (network address translation), er omskriving av IP-adresser i pakker mens de passerer en ruter, slik at verter som er nummerert i ett adresserom, kan snakke med verter som er nummerert i et annet. RFC 2663, dokumentet fra 1999 som fastsatte begrepene, beskriver det som å avbilde IP-adresser fra ett område (realm) til et annet, med gjennomsiktig ruting som mål: vertene på hver side skal ikke trenge å vite at det skjer.

Den formen nesten alle møter, er det RFC 3022, publisert i januar 2001, kaller tradisjonell NAT (Traditional NAT). Typetilfellet er et lite kontor eller et hjem med én enkelt IP-adresse fra leverandøren, mens maskinene bak ruteren bruker private adresser fra RFC 1918-blokkene 10.0.0.0/8, 172.16.0.0/12 og 192.168.0.0/16, som ikke rutes mellom organisasjoner. Ruteren skriver om kildeadressen i hver utgående pakke til sin egen offentlige adresse og reverserer endringen på svarene. Basic NAT oversetter bare adresser, så bare like mange interne verter kan være ute samtidig som det finnes offentlige adresser i adressepoolen. NAPT, network address port translation, skriver også om TCP- eller UDP-porten eller ICMP-identifikatoren, og det lar mange interne verter dele én offentlig adresse. RFC 3022 nevner at Linux tilbød NAT under navnet IP masquerade, og brannmurverktøyene bruker fortsatt ordet.

Når en adresse endres, blir alt som er regnet ut fra den, feil, så ruteren må reparere det også. RFC 3022 krever at sjekksummen i IP-hodet justeres, og TCP- og UDP-sjekksummene med, fordi de dekker et pseudohode som inneholder kilde- og måladressen. For å sende hvert svar til riktig intern vert fører ruteren en tabell over oversettelser: intern adresse og port på den ene siden, ekstern adresse og port på den andre. Den tabellen er tilstand som holdes midt i nettverket, og det meste som går galt med NAT, kommer derfra.

Tradisjonell NAT virker i én retning. RFC 3022 beskriver øktene som utgående fra det private nettverket, mens trafikk den andre veien bare slippes gjennom via statiske tilordninger for verter som er valgt på forhånd. Ideen brukes nå også ett nivå høyere opp: RFC 6598 fra april 2012 reserverte 100.64.0.0/10 til carrier-grade NAT, der blokken nummererer forbindelsene mellom en leverandørs NAT og kundenes utstyr, som selv kan oversette én gang til.

Bruk begrepet i en beslutning

Bruk «Hva er NAT?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Wi-Fi, brannmurer og fjerntilgang satt opp slik at de ansatte kan jobbe hvor som helst, og bare de rette personene slipper inn.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Administrerte 30 Ubuntu Linux-VPS-instanser — 30 Linux-servere administrert, med gjenoppretting planlagt på forhånd

Administrerte 30 Ubuntu Linux-VPS-instanser Ta kontakt →

Hvorfor det er viktig

RFC 3022 er åpen om byttehandelen. NAT kan installeres uten å endre verter eller rutere, men det tar bort, med RFC-ens ord, «ende-til-ende-betydningen av en IP-adresse», og veier opp for det med mer tilstand i nettverket. Nesten alle praktiske følger av NAT springer ut av den setningen.

For en server er kildeadressen til en forbindelse ofte ikke én maskin, men en ruter med en hel flokk bak seg. En blokkering eller en ratebegrensning rettet mot én besøkende rammer alle som deler adressen, og en misbruksrapport navngir ruteren, ikke den skyldige. RFC 3022 sier det samme fra innsiden: Å skjule private adresser, noe den regner som en fordel for personvernet, gjør også feilsøking vanskeligere, fordi en vert som angriper andre maskiner eller sender søppelpost, har adressen sin skjult i NAT-ruteren.

Tilstand betyr også utløpstider. En oversettelse lever bare så lenge ruteren husker den. RFC 4787, en Best Current Practice fra januar 2007, sier at en UDP-oversettelse ikke får utløpe på under to minutter, og anbefaler fem minutter eller mer. RFC 5382, motstykket for TCP fra oktober 2008, setter 2 timer og 4 minutter for en inaktiv etablert forbindelse: keepalive-pakker sendes som standard hver andre time, og de fire minuttene dekker pakker som fortsatt er underveis. Dette er minstekrav for enheter som følger dokumentene, og ikke alle enheter gjør det. En forbindelse som er stille lenger enn enhetens egen utløpstid, mister oversettelsen sin, og neste pakke utenfra har ingen steder å gå.

To begrensninger til følger av dette. RFC 3022 gjør det obligatorisk at alle pakker i en økt passerer den samme NAT-ruteren, og der et nettverk har mer enn én utgang, kaller den det svært viktig at hver NAT har den samme oversettelsestabellen. Og protokoller som bærer adresser inne i selve innholdet, trenger hjelp fra NAT-enheten, noe som bare virker så lenge den kan lese innholdet.

Slik ser det ut i praksis

På en Linux-ruter utfører nftables NAT i kjeder av typen `nat`. Regelsettet nedenfor er denne artikkelens illustrasjon, bygd på NAT-eksemplene i manualen nft(8). Her står 198.51.100.0/24 for det interne nettverket og 203.0.113.5 for ruterens offentlige adresse; begge er hentet fra dokumentasjonsområdene, og et ekte internt nettverk ville normalt brukt en av RFC 1918-blokkene.

```
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 første regelen er hverdagstilfellet: Pakker fra det interne nettverket som går ut gjennom eth0, får 203.0.113.5 som kilde. For en dynamisk offentlig adresse tilbyr manualen `masquerade`, som bruker den adressen det utgående grensesnittet har i øyeblikket. Den andre regelen er en statisk tilordning den andre veien: TCP-forbindelser som kommer inn på eth0 til port 443, går videre til 198.51.100.20 på innsiden. Så få regler er nok fordi bare den første pakken i en forbindelse passerer en NAT-kjede; avgjørelsen lagres i oppføringen for tilkoblingssporing, og hver senere pakke skrives om på samme måte.

Den andre siden av bildet er en vert bak en NAT den ikke selv styrer, og WireGuard har en innstilling for akkurat det. I denne artikkelens andre illustrasjon holder en intern vert en tunnel åpen til en motpart utenfor:

```
[Peer]
PublicKey = <den eksterne motpartens offentlige nøkkel>
Endpoint = 192.0.2.44:51820
AllowedIPs = 192.0.2.128/25
PersistentKeepalive = 25
```

Manualen wg(8) beskriver `PersistentKeepalive` som et intervall på 1 til 65 535 sekunder for å sende en autentisert tom pakke, slik at en tilstandsfull brannmur eller en NAT-oversettelse forblir gyldig. Den foreslår 25 sekunder for et grensesnitt bak NAT som sjelden sender noe selv og kan motta trafikk når som helst, og lar innstillingen være av som standard. Motparten utenfor trenger ikke noe spesielt: `Endpoint` for denne verten oppdateres til den nyeste kildeadressen og porten fra korrekt autentiserte pakker, og det er nettopp dem NAT-en tildelte.

Tunneler som disse er der NAT-trøbbel gjerne dukker opp. Porteføljen beskriver hvordan åtte WireGuard- og IPsec-forbindelser ble forvaltet og feilsøkt på tvers av Google Cloud og Linux-systemer i årene 2022 til 2024. Siden sier at tunnelene var skjøre og vanskelige å diagnostisere når de falt ut, og at problemene med ruting og nøkkelutveksling ble løst ved roten i stedet for å bli dekket over med en omstart, hvoretter tilkoblingshendelsene som pleide å avbryte arbeidet, opphørte. Den sier ikke om NAT spilte en rolle, så de to eksemplene ovenfor hører til denne artikkelen og ikke til det prosjektet.

Hvor det går galt

Stille forbindelser som slutter å svare. En tunnel, en databasetilkobling eller en push-kanal som ikke sender noe på en stund, mister oversettelsen sin og brytes uten at noen av endene har lukket den. Symptomet ser tilfeldig ut, fordi det avhenger av hvor lenge linjen var stille. Send keepalive-pakker innenfra med kortere mellomrom enn den korteste utløpstiden på veien, slik `PersistentKeepalive` gjør, og hold utløpstidene på en NAT du selv drifter, på eller over minstekravene i RFC 4787 og RFC 5382. RFC 4787 krever at utgående pakker fornyer en oversettelse, og advarer om at fornyelse fra innkommende trafikk lar en angriper holde den i live for alltid, så det er den interne enden som bør stå for fornyelsen.

Å forvente at forbindelser kommer inn. Tradisjonell NAT er utgående av konstruksjon, så en tjeneste på en intern vert er utilgjengelig til noen skriver en statisk tilordning for den. Legg til en `dnat` -regel for nøyaktig de portene tjenesten trenger, eller snu retningen og la den interne verten koble seg ut, slik keepalive-eksemplet ovenfor gjør.

Manglende hairpinning. To interne verter som når hverandre via den eksterne adressen til en av dem, trenger at NAT-en snur pakken. RFC 4787 krever at NAT-enheter støtter denne hairpinningen og presenterer den snudde pakken med

den eksterne kildeadressen og porten. Test den veien med vilje, og der en enhet ikke klarer det, la de interne vertene bruke den interne adressen.

Adresser inne i innholdet. FTP er det klassiske tilfellet: RFC 2663 påpeker at `PORT` -kommandoen og `PASV` -svaret bærer en IP-adresse og en port som tekst, og en NAT som bare skriver om hodene, lar dem stå feil. NAT-enheter lapper slike protokoller med en Application Level Gateway (ALG), og RFC 3022 kaller den for FTP den mest utbredte. En ALG må lese innholdet, så kryptering setter den ut av spill med mindre gatewayen har nøkkelen, og RFC 3022 legger til at ende-til-ende-IPsec ikke kan forutsettes gjennom en NAT. Velg protokoller som holder adresser utenfor innholdet; der en gammel protokoll må gjennom en NAT du drifter, viser manualen nft(8) en hjelper for tilkoblingssporing av FTP.

Logger som bare navngir ruteren. Bak NAT registrerer en servers tilgangslogg den delte adressen, som alene ikke kan peke ut én intern vert. Logg kildeporten og et nøyaktig tidsstempel ved siden av adressen, og hold klokkene synkronisert, slik at en klage på én forbindelse kan sammenholdes med oversettelsen på ruteren.

Fragmenter under NAPT. Bare det første fragmentet av en pakke bærer TCP- eller UDP-hodet, så RFC 3022 advarer om at oversettelse av utgående fragmenter med NAPT er dømt til å mislykkes, og at to interne verter som sender fragmenter med samme identifikator til samme mål, kan ødelegge begge øktene. Sørg for at pakker er små nok til å komme hele fram.

Å behandle NAT som en brannmur. At NAT bare virker i én retning, er lett å forveksle med beskyttelse. RFC 4787 erkjenner at NAT ofte tas i bruk for å nå sikkerhetsmål, men sier rett ut at måten en NAT oversetter på, ikke har noen betydning for sikkerheten, som helt og holdent bestemmes av filtreringen: hvilke pakker den slipper inn. Skriv eksplisitte filterregler for hva som får komme inn, og knytt hver `dnat` -regel til en regel som bare tillater den trafikken den var ment for.

Les mer

- RFC 3022: Traditional IP Network Address Translator (Traditional NAT) — beskrivelsen fra 2001 av Basic NAT og NAPT, med en ærlig liste over hva de koster.

- RFC 2663: IP Network Address Translator (NAT) Terminology and Considerations — begrepene for NAT, fra adresseområder til gjennomsiktig ruting og ALG-er.
- RFC 4787: Network Address Translation (NAT) Behavioral Requirements for Unicast UDP — hvordan en NAT bør oversette, la oversettelser utløpe og håndtere hairpinning for UDP-trafikk.
- RFC 5382: NAT Behavioral Requirements for TCP — motstykket for TCP, med utløpstidene for etablerte forbindelser og forbindelser som lukkes.
- wg(8): set and retrieve configuration of WireGuard interfaces — WireGuard-manualen, med `PersistentKeepalive` og rådet for verter bak NAT.
- nft(8): nftables manual — referansen for nftables, med NAT-kjeder og setningene `snat`, `dnat` og `masquerade`.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er NAT?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/nat/