Engineer

NAT

🔀 Umschreiben von IP-Adressen beim Durchlaufen eines Routers

Was es ist

NAT steht für Network Address Translation, die Übersetzung von Netzwerkadressen. Dabei werden IP-Adressen in Paketen beim Durchlaufen eines Routers umgeschrieben, damit Hosts aus einem Adressraum mit Hosts aus einem anderen kommunizieren können. RFC 2663, das 1999 die Begriffe festlegte, beschreibt das als Zuordnung von IP-Adressen zwischen Adressbereichen mit dem Ziel transparenten Routings: Die Hosts auf beiden Seiten sollen davon nichts wissen müssen.

Die fast überall anzutreffende Form nennt RFC 3022 vom Januar 2001 Traditional NAT. Der typische Fall ist ein kleines Büro oder ein Privathaushalt mit einer einzigen IP-Adresse vom Anbieter, während die Geräte hinter dem Router private Adressen aus den RFC-1918-Blöcken 10.0.0.0/8, 172.16.0.0/12 und 192.168.0.0/16 verwenden. Diese werden nicht zwischen Organisationen geroutet. Der Router ersetzt die Quelladresse jedes ausgehenden Pakets durch seine öffentliche Adresse und kehrt die Änderung bei Antworten um. Basic NAT übersetzt nur Adressen. Es können deshalb nur so viele interne Hosts zugleich nach außen kommunizieren, wie sein Pool öffentliche Adressen enthält. NAPT, Network Address Port Translation, schreibt zusätzlich den TCP- oder UDP-Port oder die ICMP-Kennung um. Damit können viele interne Hosts eine öffentliche Adresse teilen. Laut RFC 3022 bot Linux NAT unter dem Namen IP masquerade an. Seine Firewall-Werkzeuge verwenden das Wort weiterhin.

Eine geänderte Adresse macht alles ungültig, was aus ihr berechnet wurde. Der Router muss deshalb auch das berichtigen. RFC 3022 verlangt, die Prüfsumme des IP-Headers sowie die TCP- und UDP-Prüfsummen anzupassen. Letztere decken einen Pseudo-Header mit Quell- und Zieladresse ab. Um jede Antwort an den richtigen internen Host zu senden, führt der Router eine Zuordnungstabelle:

interne Adresse und Port auf einer Seite, externe Adresse und Port auf der anderen. In dieser Tabelle wird mitten im Netzwerk Zustand gespeichert. Aus ihr entstehen die meisten NAT-Probleme.

Traditional NAT arbeitet in einer Richtung. RFC 3022 beschreibt seine Sitzungen als ausgehend aus dem privaten Netzwerk. Verkehr in die Gegenrichtung ist nur über statische Adresszuordnungen für vorher ausgewählte Hosts erlaubt. Die Idee wird inzwischen auch eine Ebene höher eingesetzt: RFC 6598 vom April 2012 reservierte 100.64.0.0/10 für Carrier-Grade-NAT. Damit werden die Verbindungen zwischen dem NAT eines Anbieters und den Geräten seiner Kunden adressiert, die möglicherweise erneut übersetzen.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist NAT?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. WLAN, Firewalls und Fernzugriff werden so eingerichtet, dass Mitarbeitende von überall arbeiten können und nur Berechtigte Zugang erhalten.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 30 Ubuntu-Linux-VPS-Instanzen verwaltet — 30 verwaltete Linux-Server mit geplantem Wiederherstellungsweg

30 Ubuntu-Linux-VPS-Instanzen verwaltet Kontakt aufnehmen →

Warum es wichtig ist

RFC 3022 beschreibt die Abwägung offen. NAT lässt sich ohne Änderungen an Hosts oder Routern einführen. Es nimmt einer IP-Adresse aber, in den Worten des RFC, ihre „Ende-zu-Ende-Bedeutung“ und gleicht das durch mehr Zustand im Netzwerk aus. Fast jede praktische Folge von NAT ergibt sich aus diesem Satz.

Für einen Server steht die Quelladresse einer Verbindung oft nicht für eine einzelne Maschine, sondern für einen Router mit vielen Geräten dahinter. Eine Sperre oder Ratenbegrenzung gegen einen Besucher trifft alle mit derselben Adresse. Eine Missbrauchsmeldung nennt den Router, nicht den Verursacher. RFC 3022 beschreibt dieselbe Folge von innen: Private Adressen zu verbergen, was er als Datenschutzvorteil zählt, erschwert auch die Fehlersuche. Ein Host, der andere angreift oder Spam sendet, verbirgt seine Adresse im NAT-Router.

Zustand bedeutet auch Zeitlimits. Eine Zuordnung lebt nur, solange der Router sie behält. RFC 4787, eine Best Current Practice vom Januar 2007, verlangt mindestens zwei Minuten Gültigkeit für eine UDP-Zuordnung und empfiehlt fünf Minuten oder mehr. Sein TCP-Gegenstück RFC 5382 vom Oktober 2008 legt für eine inaktive, bereits aufgebaute Verbindung 2 Stunden und 4 Minuten fest: Keepalives werden standardmäßig alle zwei Stunden gesendet, und die vier Minuten decken noch unterwegs befindliche Pakete ab. Das sind Mindestwerte für Geräte, die den Dokumenten folgen. Nicht jedes tut das. Bleibt eine Verbindung länger still als das eigene Zeitlimit des Geräts, verliert sie ihre Zuordnung. Das nächste Paket von außen hat dann kein Ziel.

Zwei weitere Beschränkungen folgen daraus. RFC 3022 verlangt, dass jedes Paket einer Sitzung denselben NAT-Router durchläuft. Bei mehreren Netzausgängen hält er es für unerlässlich, dass die NAT-Geräte dieselbe Übersetzungstabelle teilen. Und Protokolle, die Adressen in ihren Nutzdaten transportieren, brauchen Hilfe vom NAT-Gerät. Das funktioniert nur, solange es diese Nutzdaten lesen kann.

Wie es in der Praxis aussieht

Auf einem Linux-Router führt nftables NAT in Ketten vom Typ `nat` aus. Der folgende Regelsatz ist ein Beispiel dieses Artikels, aufgebaut auf den NAT-Beispielen im Handbuch nft(8). Hier steht 198.51.100.0/24 für das interne Netz und 203.0.113.5 für die öffentliche Router-Adresse. Beide stammen aus Dokumentationsbereichen; ein echtes internes Netz würde gewöhnlich einen der RFC-1918-Blöcke verwenden.

```
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
```

Die erste Regel ist der Alltagsfall: Pakete aus dem internen Netz, die über eth0 ausgehen, erhalten 203.0.113.5 als Quelle. Für eine dynamische öffentliche Adresse bietet das Handbuch `masquerade`, das die jeweils aktuelle Adresse der ausgehenden Schnittstelle verwendet. Die zweite Regel ist eine statische Zuordnung in Gegenrichtung: TCP-Verbindungen, die auf eth0 für Port 443 eingehen, landen intern bei 198.51.100.20. So wenige Regeln genügen, weil nur

das erste Paket einer Verbindung eine nat-Kette durchläuft. Die Entscheidung wird im Eintrag der Verbindungsverfolgung gespeichert, und jedes spätere Paket wird genauso umgeschrieben.

Die andere Seite ist ein Host hinter einem NAT, das er nicht kontrolliert. WireGuard hat dafür eine passende Einstellung. In diesem zweiten, ebenfalls für den Artikel erstellten Beispiel hält ein interner Host einen Tunnel zu einer Gegenstelle außerhalb offen:

```
[Peer]
PublicKey = <the outside peer's public key>
Endpoint = 192.0.2.44:51820
AllowedIPs = 192.0.2.128/25
PersistentKeepalive = 25
```

Das Handbuch wg(8) beschreibt `PersistentKeepalive` als Intervall von 1 bis 65.535 Sekunden zum Senden eines authentifizierten leeren Pakets. Damit bleiben der Verbindungszustand einer zustandsbehafteten Firewall oder eine NAT-Zuordnung erhalten. Es empfiehlt 25 Sekunden für eine Schnittstelle hinter NAT, die selbst selten sendet und jederzeit Verkehr empfangen könnte. Standardmäßig ist die Einstellung aus. Die externe Gegenstelle braucht nichts Besonderes: Ihr `Endpoint` für diesen Host wird auf die jüngste Quelladresse samt Port korrekt authentifizierter Pakete aktualisiert, also genau auf die vom NAT zugewiesenen Werte.

In solchen Tunneln zeigen sich NAT-Probleme häufig. Das Portfolio beschreibt acht von 2022 bis 2024 verwaltete und entstörte WireGuard- und IPsec-Verbindungen über Google Cloud und Linux-Systeme hinweg. Laut Seite waren die Tunnel störanfällig und Ausfälle schwer zu diagnostizieren. Wiederkehrende Routing- und Schlüsselaustauschprobleme wurden an der Ursache behoben, statt sie mit einem Neustart zu behandeln. Danach hörten die Unterbrechungen auf. Ob NAT dabei eine Rolle spielte, sagt die Seite nicht. Die beiden obigen Beispiele gehören deshalb zu diesem Artikel, nicht zu jenem Projekt.

Wo es schiefgeht

Stille Verbindungen antworten nicht mehr. Ein Tunnel, eine Datenbankverbindung oder ein Push-Kanal, der eine Weile nichts sendet, verliert seine Zuordnung und bricht ab, ohne dass eines der Enden die Verbindung geschlossen hätte. Das wirkt zufällig, weil es von der Dauer der Inaktivität

abhängt. Senden Sie wie bei `PersistentKeepalive` Keepalives von innen in Abständen, die kürzer sind als das kürzeste Zeitlimit auf dem Weg. Halten Sie bei einem selbst betriebenen NAT die Zeitlimits mindestens auf den Werten aus RFC 4787 und RFC 5382. RFC 4787 verlangt, dass ausgehende Pakete eine Zuordnung auffrischen, und warnt, dass eingehendes Auffrischen einem Angreifer erlaubt, sie unbegrenzt am Leben zu halten. Deshalb sollte das interne Ende auffrischen.

Eingehende Verbindungen werden erwartet. Traditional NAT ist auf ausgehende Verbindungen ausgelegt. Ein Dienst auf einem internen Host ist deshalb unerreichbar, bis jemand eine statische Zuordnung anlegt. Ergänzen Sie eine `dnat` -Regel für genau die benötigten Ports oder kehren Sie die Richtung um: Der interne Host verbindet sich nach außen, wie beim obigen Keepalive.

Hairpinning fehlt. Erreichen sich zwei interne Hosts über die externe Adresse eines von ihnen, muss das NAT das Paket umkehren. RFC 4787 verlangt dieses Hairpinning sowie die externe Quelladresse samt Port am zurückgeleiteten Paket. Prüfen Sie diesen Weg bewusst. Scheitert ein Gerät daran, lassen Sie interne Hosts die interne Adresse verwenden.

Adressen in den Nutzdaten. FTP ist der klassische Fall. RFC 2663 weist darauf hin, dass sein Befehl `PORT` und seine Antwort `PASV` eine IP-Adresse und einen Port als Text enthalten. Ein NAT, das nur Header umschreibt, lässt sie falsch zurück. NAT-Geräte helfen solchen Protokollen mit einem Application Level Gateway (ALG). RFC 3022 nennt das für FTP das verbreitetste. Ein ALG muss die Nutzdaten lesen. Verschlüsselung verhindert das deshalb, sofern das Gateway nicht den Schlüssel besitzt. RFC 3022 ergänzt, dass Ende-zu-Ende-IPsec über ein NAT hinweg nicht vorausgesetzt werden kann. Bevorzugen Sie Protokolle ohne Adressen in den Nutzdaten. Muss ein altes Protokoll ein selbst betriebenes NAT durchlaufen, zeigt das Handbuch nft(8) dafür einen Helfer der Verbindungsverfolgung für FTP.

Logs nennen nur den Router. Hinter NAT hält das Zugriffslog eines Servers die gemeinsame Adresse fest. Allein identifiziert sie keinen einzelnen internen Host. Protokollieren Sie neben der Adresse den Quellport und einen genauen Zeitstempel. Halten Sie die Uhren synchron, damit eine Beschwerde über eine Verbindung der Übersetzung auf dem Router zugeordnet werden kann.

Fragmente unter NAPT. Nur das erste Fragment eines Pakets enthält den TCP- oder UDP-Header. RFC 3022 warnt deshalb, dass das Übersetzen ausgehender Fragmente mit NAPT zwangsläufig scheitert. Senden zwei interne Hosts

Fragmente mit derselben Kennung an dasselbe Ziel, können beide Sitzungen beschädigt werden. Halten Sie Pakete klein genug, um den Weg unfragmentiert zu durchlaufen.

NAT wird als Firewall behandelt. Die einseitige Ausrichtung von NAT wird leicht mit Schutz verwechselt. RFC 4787 erkennt an, dass NAT häufig aus Sicherheitsgründen eingesetzt wird. Er sagt aber ausdrücklich, dass die Art der Zuordnung nichts an der Sicherheit ändert. Diese wird vollständig vom Filterverhalten bestimmt, also davon, welche Pakete es hineinlässt. Schreiben Sie ausdrückliche Filterregeln für zulässigen eingehenden Verkehr und ergänzen Sie jede `dnat` -Regel um eine Regel, die nur den dafür vorgesehenen Verkehr erlaubt.

Weiterführende Lektüre

- RFC 3022: Traditioneller IP-Netzwerkadressübersetzer (Traditional NAT) — die Beschreibung von Basic NAT und NAPT aus dem Jahr 2001 mit einer offenen Liste ihrer Kosten.
- RFC 2663: Begriffe und Überlegungen zur IP-Netzwerkadressübersetzung (NAT) — die NAT-Begriffe von Adressbereichen bis zu transparentem Routing und ALGs.
- RFC 4787: Verhaltensanforderungen an NAT für Unicast-UDP — wie ein NAT UDP-Verkehr zuordnen, zeitlich begrenzen und per Hairpinning zurückführen sollte.
- RFC 5382: Verhaltensanforderungen an NAT für TCP — das TCP-Gegenstück einschließlich der Inaktivitätsfristen für aufgebaute und sich schließende Verbindungen.
- wg(8): Konfiguration von WireGuard-Schnittstellen setzen und abrufen — das WireGuard-Handbuch mit `PersistentKeepalive` und Empfehlungen für Hosts hinter NAT.
- nft(8): nftables-Handbuch — die nftables-Referenz einschließlich nat-Ketten und der Anweisungen `snat` , `dnat` und `masquerade` .

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist NAT?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/nat/