Engineer

NAT

🔀 заміна IP-адрес у пакетах, що проходять маршрутизатор

Що це таке

NAT (англійською network address translation, трансляція мережевих адрес) — це заміна IP-адрес у пакетах, коли ті проходять через маршрутизатор, щоб хости, пронумеровані в одному адресному просторі, могли спілкуватися з хостами, пронумерованими в іншому. RFC 2663, документ 1999 року, який усталив термінологію, описує це як відображення IP-адрес з однієї адресної області (realm) в іншу з метою прозорої маршрутизації: хостам по обидва боки не потрібно знати, що це відбувається.

Форма, з якою стикаються майже всі, — це те, що RFC 3022, опублікований у січні 2001 року, називає традиційним NAT (Traditional NAT). Типовий випадок — невеликий офіс чи дім з однією IP-адресою від провайдера, тоді як машини за маршрутизатором використовують приватні адреси з блоків RFC 1918: 10.0.0.0/8, 172.16.0.0/12 і 192.168.0.0/16, які між організаціями не маршрутизуються. Маршрутизатор замінює адресу джерела в кожному вихідному пакеті своєю публічною адресою і робить зворотну заміну у відповідях. Basic NAT транслює лише адреси, тож одночасно назовні може вийти не більше внутрішніх хостів, ніж є публічних адрес у його пулі. NAPT (network address port translation) змінює ще й порт TCP чи UDP або ідентифікатор ICMP, і це дозволяє багатьом внутрішнім хостам ділити одну публічну адресу. RFC 3022 згадує, що Linux пропонував NAT під назвою IP masquerade, і інструменти фаєрвола в Linux досі вживають це слово.

Зміна адреси псує все, що з неї обчислено, тож маршрутизатор мусить виправити й це. RFC 3022 вимагає коригувати контрольну суму заголовка IP, а також контрольні суми TCP і UDP, бо вони охоплюють псевдозаголовок, що містить адреси джерела й призначення. Щоб надіслати кожну відповідь потрібному внутрішньому хостові, маршрутизатор веде таблицю трансляцій:

внутрішня адреса й порт з одного боку, зовнішня адреса й порт з іншого. Ця таблиця — стан, який зберігається посеред мережі, і більшість того, що йде не так із NAT, походить саме звідти.

Традиційний NAT працює в одному напрямку. RFC 3022 описує його сеанси як вихідні з приватної мережі, а трафік у зворотному напрямку пропускається лише через статичні трансляції для заздалегідь вибраних хостів. Тепер ця ідея працює й на рівень вище: RFC 6598 у квітні 2012 року зарезервував 100.64.0.0/10 для carrier-grade NAT, тобто NAT на рівні оператора, де цей блок нумерує ланки між NAT провайдера та пристроями клієнтів, які самі можуть транслювати адреси ще раз.

Застосуйте поняття до рішення

Застосуйте «Що таке NAT?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Wi-Fi, фаєрволи і віддалений доступ налаштовані так, щоб працювати можна було звідусіль, а заходити — лише своїм.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Керувала 30 інстансами Ubuntu Linux VPS — 30 серверів Linux під управлінням, із планом відновлення

Керувала 30 інстансами Ubuntu Linux VPS Зв'язатися з нами →

Чому це важливо

RFC 3022 відверто говорить про ціну. NAT можна встановити, не змінюючи ні хостів, ні маршрутизаторів, але він забирає, словами RFC, «наскрізне значення IP-адреси» і компенсує це більшим обсягом стану в мережі. Майже кожен практичний наслідок NAT випливає з цього речення.

Для сервера адреса джерела з’єднання часто означає не одну машину, а маршрутизатор із цілим натовпом за ним. Блокування чи обмеження частоти запитів, спрямоване на одного відвідувача, зачіпає всіх, хто ділить цю адресу, а скарга на зловживання називає маршрутизатор, а не винуватця. RFC 3022 каже те саме зсередини: приховування приватних адрес, яке він зараховує до переваг для приватності, ускладнює й пошук несправностей, бо хост, що атакує інші машини чи розсилає спам, має адресу, приховану в маршрутизаторі NAT.

Стан означає й таймери. Трансляція живе лише доти, доки маршрутизатор її пам’ятає. RFC 4787, документ категорії Best Current Practice від січня 2007 року, каже, що трансляція UDP не повинна зникати раніше ніж через дві

хвилини, і рекомендує п’ять хвилин або більше. RFC 5382, його відповідник для TCP від жовтня 2008 року, встановлює 2 години 4 хвилини для неактивного встановленого з’єднання: пакети keepalive за замовчуванням надсилаються раз на дві години, а чотири хвилини покривають пакети, які ще в дорозі. Це мінімуми для пристроїв, що дотримуються цих документів, а дотримуються не всі. З’єднання, яке мовчить довше, ніж спрацьовує власний таймер пристрою, втрачає свою трансляцію, і наступному пакетові ззовні вже нікуди йти.

Звідси випливають ще два обмеження. RFC 3022 робить обов’язковим, щоб усі пакети одного сеансу проходили через той самий маршрутизатор NAT, а там, де мережа має більше одного виходу, називає надзвичайно важливим, щоб кожен NAT мав ту саму таблицю трансляцій. А протоколи, які переносять адреси всередині корисного навантаження, потребують допомоги від пристрою NAT, і це працює лише доти, доки він може те навантаження прочитати.

Як це виглядає на практиці

На маршрутизаторі з Linux NAT виконує nftables, у ланцюжках типу `nat`. Набір правил нижче — ілюстрація цієї статті, побудована на прикладах NAT зі сторінки довідки nft(8). Тут 198.51.100.0/24 позначає внутрішню мережу, а 203.0.113.5 — публічну адресу маршрутизатора; обидві взято з документаційних діапазонів, а справжня внутрішня мережа зазвичай використовувала б один із блоків RFC 1918.

```
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
```

Перше правило — щоденний випадок: пакети з внутрішньої мережі, що виходять через eth0, отримують 203.0.113.5 як адресу джерела. Для динамічної публічної адреси довідка пропонує `masquerade`, що бере ту адресу, яку вихідний інтерфейс має в цю мить. Друге правило — статична трансляція у зворотному напрямку: TCP-з’єднання, що надходять на eth0 до порту 443, ідуть усередину, на 198.51.100.20. Так мало правил вистачає тому,

що ланцюжок nat проходить лише перший пакет з’єднання; його рішення зберігається в записі відстеження з’єднань, і кожен наступний пакет переписується так само.

Інший бік картини — хост за NAT, яким він сам не керує, і для цього у WireGuard є окремий параметр. У другій ілюстрації цієї статті внутрішній хост тримає тунель відкритим до вузла назовні:

```
[Peer]
PublicKey = <публічний ключ зовнішнього вузла>
Endpoint = 192.0.2.44:51820
AllowedIPs = 192.0.2.128/25
PersistentKeepalive = 25
```

Довідка wg(8) описує `PersistentKeepalive` як інтервал від 1 до 65 535 секунд для надсилання автентифікованих порожніх пакетів, щоб брандмауер зі станом або трансляція NAT лишалися чинними. Вона радить 25 секунд для інтерфейсу за NAT, який рідко щось надсилає сам і може отримати трафік будь-коли, а за замовчуванням параметр вимкнено. Зовнішньому вузлові нічого особливого не потрібно: його `Endpoint` для цього хоста оновлюється на найновішу адресу джерела й порт коректно автентифікованих пакетів, а це саме те, що призначив NAT.

Саме в таких тунелях проблеми з NAT зазвичай і виринають. Портфоліо описує, як упродовж 2022–2024 років керували вісьмома з’єднаннями WireGuard VPN та IPsec VPN між системами Google Cloud та Linux і усували в них несправності. Сторінка каже, що тунелі були крихкими й незручними для діагностики, коли падали, і що повторювані проблеми з маршрутизацією та обміном ключами усували в корені, а не маскували перезапуском, після чого збої зі з’єднанням, які раніше переривали роботу, припинилися. Чи відігравав тут роль NAT, вона не каже, тож обидва приклади вище належать цій статті, а не тому проєктові.

Де виникають проблеми

Тихі з’єднання, що перестають відповідати. Тунель, з’єднання з базою даних чи канал push-сповіщень, що якийсь час нічого не надсилає, втрачає свою трансляцію і рветься, хоча жоден бік його не закривав. Симптом виглядає випадковим, бо залежить від того, як довго лінія мовчала. Надсилайте пакети keepalive зсередини частіше, ніж спрацьовує

найкоротший таймер на шляху, як це робить `PersistentKeepalive`, а на NAT, який ви експлуатуєте самі, тримайте таймери не нижче мінімумів RFC 4787 і RFC 5382. RFC 4787 вимагає, щоб трансляцію оновлювали вихідні пакети, і попереджає, що оновлення вхідним трафіком дозволяє зловмисникові підтримувати її вічно, тож оновлювати її має внутрішній бік.

Очікувати, що з'єднання прийдуть ззовні. Традиційний NAT за своєю будовою вихідний, тож служба на внутрішньому хості недосяжна, доки хтось не пропише для неї статичну трансляцію. Додайте правило `dnat` рівно для тих портів, які потрібні службі, або розверніть напрямок і дайте внутрішньому хостові самому з'єднуватися назовні, як у прикладі з keepalive вище.

Відсутній hairpinning. Двом внутрішнім хостам, які звертаються один до одного через зовнішню адресу одного з них, потрібно, щоб NAT розвернув пакет назад. RFC 4787 вимагає, щоб пристрої NAT підтримували такий hairpinning і подавали розвернутий пакет із зовнішньою адресою джерела й портом. Перевірте цей шлях навмисно, а там, де пристрій його не витримує, нехай внутрішні хости звертаються за внутрішньою адресою.

Адреси всередині корисного навантаження. FTP — класичний випадок: RFC 2663 зазначає, що команда `PORT` і відповідь на `PASV` несуть IP-адресу й порт у вигляді тексту, і NAT, який переписує лише заголовки, лишає їх хибними. Пристрої NAT латають такі протоколи шлюзом прикладного рівня (Application Level Gateway, ALG), і RFC 3022 називає шлюз для FTP найпоширенішим. ALG мусить читати корисне навантаження, тож шифрування виводить його з ладу, якщо шлюз не має ключа, а RFC 3022 додає, що на наскрізний IPsec через NAT розраховувати не можна. Віддавайте перевагу протоколам, які не кладуть адрес у корисне навантаження; а там, де старий протокол мусить проходити через NAT, який ви експлуатуєте, довідка nft(8) показує помічник відстеження з'єднань для FTP.

Журнали, що називають лише маршрутизатор. За NAT журнал доступу сервера записує спільну адресу, яка сама по собі не може вказати на один внутрішній хост. Записуйте поруч з адресою порт джерела й точну позначку часу та синхронізуйте годинники, щоб скаргу на одне з'єднання можна було зіставити з трансляцією на маршрутизаторі.

Фрагменти під NAPT. Заголовок TCP чи UDP несе лише перший фрагмент, тож RFC 3022 попереджає, що трансляція вихідних фрагментів через NAPT приречена на невдачу і що два внутрішні хости, які надсилають фрагменти з тим самим ідентифікатором на ту саму адресу призначення, можуть зіпсувати обидва сеанси. Тримайте пакети досить малими, щоб вони проходили шлях цілими.

Сприймати NAT як брандмауер. Те, що NAT працює в одному напрямку, легко сплутати із захистом. RFC 4787 визнає, що NAT часто впроваджують заради цілей безпеки, але прямо каже, що спосіб, у який NAT будує трансляції, не впливає на його безпеку: її повністю визначає фільтрація, тобто те, які пакети він пропускає досередини. Пишіть явні правила фільтрації для того, що може входити, і до кожного правила `dnat` додавайте правило, яке пропускає лише той трафік, для якого його створено.

Що почитати далі

- RFC 3022: Traditional IP Network Address Translator (Traditional NAT) — опис Basic NAT і NAPT 2001 року з чесним переліком того, чого вони коштують.
- RFC 2663: IP Network Address Translator (NAT) Terminology and Considerations — термінологія NAT, від адресних областей до прозорої маршрутизації й ALG.
- RFC 4787: Network Address Translation (NAT) Behavioral Requirements for Unicast UDP — як NAT має будувати трансляції для трафіку UDP, коли їх видаляти і як розвертати пакети між внутрішніми хостами.
- RFC 5382: NAT Behavioral Requirements for TCP — відповідник для TCP, зокрема з тайм-аутами неактивності для встановлених і закриваних з'єднань.
- wg(8): set and retrieve configuration of WireGuard interfaces — довідка WireGuard з `PersistentKeepalive` і порадою для хостів за NAT.
- nft(8): nftables manual — довідник nftables, зокрема ланцюжки nat і інструкції `snat`, `dnat` та `masquerade`.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке NAT?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/nat/