Engineer

IP-Adresse

Zahl, die dem Netzwerk das Ziel eines Pakets angibt

Was es ist

Eine IP-Adresse ist die Zahl, die das Internet Protocol in jedes Paket schreibt, um sein Ziel anzugeben, neben einer zweiten Adresse für seine Herkunft. RFC 791, die Spezifikation der IP-Version 4 von 1981, nennt sie Internetadresse und zieht die Grenze in einem Satz: „Eine Adresse gibt an, wo sich das Gesuchte befindet.“ Ein Name, etwa eine Domain, sagt, wonach gesucht wird, und muss zuerst in eine Adresse übersetzt werden. Das ist Aufgabe von DNS. Die Route, der Weg zwischen den beiden Adressen, bleibt den Routern dazwischen überlassen.

In IPv4 umfasst die Adresse 32 Bits, vier Oktette, geschrieben als vier durch Punkte getrennte Dezimalzahlen von 0 bis 255: 192.0.2.10. RFC 791 teilte diese Bits in eine Netzwerknummer und einen lokalen Teil auf. Drei feste Klassen bestimmten die Lage der Trennstelle. Für die meisten Organisationen passten die Klassen schlecht: Ein Klasse-C-Netz fasste höchstens 254 Hosts, ein Klasse-B-Netz bis zu 65.534. So wurden Klasse-B-Nummern knapp, während die Routingtabellen wuchsen. Im September 1993 ersetzte RFC 1519 die Klassen durch CIDR. Damit darf die Trennung hinter jedem beliebigen Bit liegen; ihre Position steht hinter einem Schrägstrich. 192.0.2.0/24 ist der Block mit festgelegten ersten 24 Bits: 256 Adressen.

IPv6 erweitert die Adresse auf 128 Bits. Sie wird als acht durch Doppelpunkte getrennte Gruppen mit bis zu vier Hexadezimalziffern geschrieben. RFC 4291 erlaubt, eine Folge von Nullgruppen zu `::` zusammenzuziehen. 2001:db8:0:0:0:0:2:1 lässt sich daher auch als 2001:db8::2:1 schreiben. Weil dieselbe Adresse damit viele Schreibweisen haben konnte, legte RFC 5952 später eine empfohlene Form fest: Kleinbuchstaben, keine führenden Nullen in einer Gruppe und `::` für die längste Folge von Nullgruppen.

Eine Adresse gehört zu einer Netzwerkschnittstelle, nicht zu einer Maschine. Schon RFC 791 sagte, dass ein einzelner Host mit mehreren Internetadressen wie mehrere Hosts auftreten können muss und manche Hosts mehrere physische Schnittstellen haben. RFC 4291 verlangt für jede IPv6-Schnittstelle mindestens eine link-lokale Adresse, die nur im eigenen Netzwerksegment gültig ist, unabhängig von weiteren Adressen. Ein Host in einem Netzwerk mit beiden Versionen hat daher mehrere Adressen gleichzeitig, zusätzlich zu seinen Loopback-Adressen, über die er sich selbst erreicht: dem Block 127.0.0.0/8 in IPv4 und `::1` in IPv6.

Nicht jede Adresse ist von überall erreichbar. RFC 1918 reserviert drei IPv4-Blöcke für private Netzwerke, 10.0.0.0/8, 172.16.0.0/12 und 192.168.0.0/16, und schreibt vor, Routinginformationen darüber nicht zwischen Organisationen weiterzugeben. Ein Gerät mit einer solchen Adresse erreicht das Internet daher nur über einen Stellvertreter, etwa NAT. RFC 3022 beschreibt NAT als Übersetzung der privaten Adresse in eine öffentliche beim Verlassen des Netzes. Drei weitere Blöcke, 192.0.2.0/24, 198.51.100.0/24 und 203.0.113.0/24, reserviert RFC 5737 für Dokumentation; RFC 3849 tut dasselbe für 2001:db8::/32 in IPv6. Jede Beispieladresse in diesem Artikel stammt aus diesen Dokumentationsbereichen.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist eine IP-Adresse?“ 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

Die Adresse ist die Angabe über einen Client, die einem Server immer vorliegt, denn ohne sie hätte die Antwort kein Ziel. Damit wird sie zum Standardschlüssel für viele Entscheidungen: Firewall-Regeln erlauben oder sperren nach Adresse, eine Ratenbegrenzung zählt Anfragen je Adresse, ein Zugriffsprotokoll schreibt sie in jede Zeile und eine Missbrauchsmeldung nennt sie. All das funktioniert nur so gut, wie die Adresse tatsächlich das identifiziert, was man ihr zuschreibt.

Sie identifiziert weniger, als es scheint. Hinter NAT können alle Geräte eines Büros oder Haushalts über dieselbe öffentliche IPv4-Adresse ins Internet gelangen. Eine Sperre oder Grenze für eines davon trifft dann alle. Hinter einem Proxy oder CDN sieht der Server bei jeder Anfrage die Adresse des Proxys. Die eigene Adresse des Clients kommt, wenn überhaupt, in einem von jemand anderem geschriebenen Header an. Eine adressbezogene Regel betrifft eine Position im Netzwerk, niemals eine Person.

Gleichzeitig verrät die Adresse mehr, als viele erwarten. RFC 7239 standardisierte den Header `Forwarded`, mit dem Proxys die Clientadresse weitergeben. Sein Datenschutzabschnitt hält fest, dass eine IP-Adresse einen Client eindeutig identifizieren, seinen Netzbetreiber verraten und seinen ungefähren Standort erkennen lassen kann. Ein Protokoll voller Adressen ist daher neben einem Werkzeug auch ein Risiko und ein guter Grund, vor dem Speichern festzulegen, was aufbewahrt werden soll.

Außerdem gibt es inzwischen zwei Arten. Alles, was Adressen einliest, speichert oder vergleicht, muss IPv4 und IPv6 gleichermaßen akzeptieren, denn ein Client in einem Netzwerk mit beiden kann über jede davon ankommen. Code aus einer Zeit, in der jede Adresse in fünfzehn Zeichen aus Ziffern und Punkten passte, scheitert am ersten IPv6-Besucher, meist weit entfernt von der Stelle, an der die Adresse eingelesen wurde.

Wie es in der Praxis aussieht

Pythons Standardbibliothek enthält das Modul `ipaddress`, das beide Versionen einliest und die Berechnungen übernimmt. Zeichenketten müssen Sie damit nicht von Hand zerlegen. Eine interaktive Sitzung:

```
>>> import ipaddress
>>> ipaddress.ip_address("203.0.113.57")
IPv4Address('203.0.113.57')
>>> ipaddress.ip_address("2001:0DB8:0000:0000:0000:0000:0002:0001")
IPv6Address('2001:db8::2:1')
>>> ipaddress.ip_network("203.0.113.57/24", strict=False)
IPv4Network('203.0.113.0/24')
>>> ipaddress.ip_network("2001:db8:42:7::5/48", strict=False)
IPv6Network('2001:db8:42::/48')
```

Der zweite Aufruf zeigt die von RFC 5952 verlangte Normalisierung: Die lange Schreibweise mit Großbuchstaben und die kurze mit Kleinbuchstaben sind dieselbe Adresse, und das Modul gibt die kanonische Form aus. Eine Zeichenkette wie 192.0.2.300 wird mit einem `ValueError` abgewiesen, weil 300 nicht in ein Oktett passt. Die letzten beiden Aufrufe machen aus einer Adresse das zugehörige Netzwerk. Ohne `strict=False` würde das Modul auch diese Eingaben ablehnen, denn in 203.0.113.57/24 sind Hostbits gesetzt. Mit dieser Option wird der Hostteil auf null gesetzt, und nur das Netzwerk bleibt übrig.

So behandelt das Webzugriffslog dieser Website die Adressen ihrer Besucher. Laut ihrer Datenschutzseite wird die Adresse beim Schreiben der Weblogzeile gekürzt: Ihr letzter Teil wird durch Nullen ersetzt, bevor die Zeile die Festplatte erreicht. Der gespeicherte Wert bezeichnet damit ein grobes Netzwerk, in dem ein ganzes Büro oder eine Straße gleich aussehen kann. Der Server erfasst die Adresse unter zwei Namen, maskiert beide und entfernt vor dem Schreiben der Zeile Header, die eine eigene Adresse enthalten können. Die Seite sagt nicht, wie viele Bits erhalten bleiben; die obigen /24 und /48 sind Beispiele dieses Artikels, keine Einstellungen der Website. Die Maskierung beim Schreiben hält vollständige Adressen aus diesem Weblog heraus, ohne von einer späteren Bereinigung abzuhängen. Dieselbe Datenschutzseite dokumentiert gesondert eine Gopher-Ausnahme: Das verbindungsbezogene Betriebssystemprotokoll nennt die Quelladresse und wird nach dreißig Tagen gelöscht.

Wo es schiefgeht

Einem Header als Quelle der Clientadresse vertrauen. `X-Forwarded-For` und der standardisierte Header `Forwarded` sind gewöhnliche Anfrageheader. RFC 7239 warnt, dass jede Zwischenstation zwischen Client und Server sowie der Client selbst den Header `Forwarded` versehentlich oder absichtlich umschreiben kann. Code, der die erste Adresse der Liste nimmt, lässt jeden Client seine eigene Adresse wählen und damit seine Ratenbegrenzung und seinen Platz auf einer Positivliste. Verwenden Sie die Adresse der TCP-Verbindung, sofern die Anfrage nicht über einen von Ihnen betriebenen Proxy kam. Lesen Sie dann von rechts aus nur die Einträge, die Ihre eigenen Proxys angehängt haben.

Eine Adresse wie eine Person behandeln. Weil viele Nutzer eine Adresse teilen können, sperrt eine Adresssperre möglicherweise ein ganzes Büro aus, und eine für eine Person bemessene Grenze drosselt eine ganze Schule. Verwenden Sie Adressen für grobe Entscheidungen, berücksichtigen Sie gemeinsam genutzte Adressen bei Grenzen und binden Sie persönliche Zuordnungen stattdessen an ein authentifiziertes Konto.

Adressen als Zeichenketten vergleichen. 2001:db8::2:1 und 2001:DB8:0:0:0:0:2:1 sind eine Adresse, aber zwei unterschiedliche Zeichenketten. Eine Positivliste mit Textabgleich akzeptiert daher die eine und lehnt die andere ab. Parsen Sie beide Seiten vor dem Vergleich oder speichern Sie ausschließlich die Form aus RFC 5952.

Nur für IPv4 parsen. Ein regulärer Ausdruck für vier durch Punkte getrennte Zahlen, eine Spalte für fünfzehn Zeichen oder Code, der Host und Port am letzten Doppelpunkt trennt, scheitern an IPv6. Verwenden Sie einen richtigen Parser und einen für beide Versionen geeigneten Spaltentyp oder eine entsprechende Länge. Steht ein Port neben einer IPv6-Adresse, empfiehlt RFC 5952 eckige Klammern wie in [2001:db8::1]:443, weil die letzte Gruppe sonst als Teil der Adresse gelesen werden könnte.

Vollständige Adressen länger als nötig aufbewahren. Protokolle, Analysetabellen und Sicherungen sammeln jahrelang Adressen, wenn nichts sie daran hindert. Bestimmen Sie die benötigte Genauigkeit, kürzen Sie die Adresse beim Schreiben statt nachträglich und geben Sie dem verbleibenden Teil ein Ablaufdatum.

Echte Adressen in Beispielen verwenden. Wer in einer Beispielkonfiguration, in Testdaten oder in einem Screenshot eine fremde reale Adresse verwendet, sorgt dafür, dass beim Übernehmen des Beispiels Datenverkehr an Unbeteiligte gesendet wird. RFC 5737 reserviert seine drei Blöcke eigens, um solche Konflikte und Verwechslungen zu verringern. Verwenden Sie diese und für IPv6 2001:db8::/32.

Weiterführende Lektüre

- RFC 791: Internet Protocol — Die IPv4-Spezifikation von 1981 mit ihrer 32-Bit-Adresse und der Unterscheidung zwischen Namen, Adressen und Routen.
- RFC 4291: Architektur der IPv6-Adressierung — IPv6-Adresstypen, ihre Textformen und die Pflicht zu link-lokalen Adressen.
- RFC 5952: Empfehlung zur Textdarstellung von IPv6-Adressen — Eine kanonische Schreibweise für IPv6-Adressen und die Probleme flexibler Schreibweisen.
- RFC 7239: HTTP-Erweiterung Forwarded — Wie Proxys eine Clientadresse weitergeben, warum man dem Ergebnis nicht einfach vertrauen kann und welche Folgen das für den Datenschutz hat.

- RFC 5737: Für Dokumentation reservierte IPv4-Adressblöcke — Die drei für Beispiele vorgesehenen Bereiche.
- ipaddress: Bibliothek zur IPv4-/IPv6-Verarbeitung — Pythons Standardmodul zum Parsen, Vergleichen und Gruppieren von Adressen.

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 eine IP-Adresse?“ 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/ip-address/