Engineer

Firewall

Regeln dafür, welcher Netzwerkverkehr durchgelassen wird

Was es ist

Eine Firewall ist ein Filter auf einem Netzwerkpfad. Sie prüft jedes Paket oder jede Verbindung anhand einer Regelliste und entscheidet, ob der Verkehr durchgelassen, verworfen oder mit einer Ablehnung beantwortet wird. Sie kann als Software auf dem geschützten Rechner laufen, als separates Gerät zwischen zwei Netzwerken stehen oder als Dienst eines Cloud-Anbieters vor einer virtuellen Maschine liegen. Unter Linux wird dafür oft auch „Paketfilter“ gesagt. Cloud-Anbieter verkaufen Firewalls unter Bezeichnungen wie Security Group. Meist gibt es mehrere Firewall-Schichten: eine beim Anbieter, eine auf dem Host.

Der NIST-Leitfaden SP 800-41 Rev. 1 vom September 2009 ordnet sie danach, wie viel vom Verkehr sie lesen. Ein Paketfilter betrachtet Adressen, Protokoll und Portnummern, jeweils für ein Paket. Eine zustandsbehaftete Firewall merkt sich auch Verbindungen. Die Antwort auf eine vom Server selbst gestellte Anfrage gelangt so ohne eigene Regel wieder hinein. Ein Application-Proxy-Gateway terminiert die Verbindung und liest das darin verwendete Protokoll. Eine hostbasierte Firewall läuft auf dem Server oder Laptop, den sie schützt, und weiß, welchem Programm ein Socket gehört.

Auf einem Linux-Server filtert das netfilter-Framework des Kernels. nftables, seit Kernel 3.13 verfügbar, ersetzt die älteren iptables-Werkzeuge. Frontends wie ufw schreiben Regeln für Menschen, die lieber ihre Absicht ausdrücken als Chains angeben. Unabhängig vom schreibenden Werkzeug bleibt das Modell gleich: eine geordnete Liste, von oben gelesen, in der die erste passende Regel entscheidet.

Eine Firewall wird oft mit benachbarten Werkzeugen verwechselt. Eine Web Application Firewall liest HTTP-Anfragen und blockiert darin Angriffsmuster. Eine Netzwerkfirewall schaut nie so tief hinein. Ein Werkzeug zum Sperren von Angreifern liest Logs und ergänzt zeitweilige Regeln gegen auffällige Adressen.

Es hängt damit von einer darunterliegenden Firewall ab. Ein VPN entscheidet, wer überhaupt im Netzwerk ist. Jedes beantwortet eine andere Frage; keines ersetzt die einfache Regelliste.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist eine Firewall?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Konten, Passwörter und Berechtigungen werden so geordnet, dass alle genau den beruflich erforderlichen Zugriff haben und ihn am Tag ihres Ausscheidens verlieren.

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. 1.000 RBAC-Regeln für sichere Linux-Server — 1,000 Zugriffsregeln zum Schließen einer Sicherheitslücke

1.000 RBAC-Regeln für sichere Linux-Server Kontakt aufnehmen →

Warum es wichtig ist

Jeder Dienst, der auf einer öffentlichen Adresse lauscht, ist eine Tür, und die Firewall ist das Verzeichnis der vorhandenen Türen. Die eigentliche Teamentscheidung hat drei Teile: wo die Firewall liegt, was sie standardmäßig erlaubt und wer sie ändern darf.

Eine Anbieter-Firewall stoppt Verkehr, bevor er den Rechner erreicht, arbeitet weiter, wenn jemand Root-Zugriff auf dem Host erlangt, und kostet den Server keine CPU-Leistung. Ihre Regeln liegen in einer Konsole oder API außerhalb der Serverkonfiguration. Ohne Verwaltung als Code schleichen sich daher Abweichungen ein. Eine Host-Firewall begleitet den Rechner, kann die Rate an einem einzelnen Port begrenzen und vom Rechner selbst getestet werden. Ein kompromittiertes Root-Konto kann sie jedoch umschreiben, ebenso Software mit eigener Regelverwaltung. Beide zu betreiben ist üblich und sinnvoll. Der Preis sind zwei synchron zu haltende Listen: Jeder geöffnete oder geschlossene Port bedeutet eine Änderung an zwei Stellen.

Der zweite Teil ist das Standardverhalten. NIST empfiehlt, allen nicht ausdrücklich erlaubten Verkehr abzulehnen. Für eingehenden Verkehr sind sich fast alle einig. Beim ausgehenden Verkehr macht sich der Zielkonflikt bemerkbar: Eine Filterung begrenzt, was ein kompromittierter Dienst erreichen kann, unterbricht aber Paketaktualisierungen, Zertifikatserneuerung und Monitoring, bis jedes Ziel seine eigene Regel hat.

Der dritte Teil ist die Verantwortung. Eine Firewall mit standardmäßiger Ablehnung ist nur so korrekt wie die deklarierte Portliste. Das nützliche Arbeitsergebnis ist daher diese Liste, versioniert und gegen die aktiven Regeln geprüft. Die Software ist in jeder gängigen Linux-Distribution kostenlos. Geld kostet die Aufmerksamkeit, die die Deklaration korrekt hält. Eine Firewall ist außerdem eine zweite Verteidigungslinie. Ein an die Loopback-Adresse gebundener Dienst ist unabhängig von den Regeln von außen unerreichbar. Das ist eine stärkere Garantie als eine Verbotsregel vor einem überall lauschenden Dienst.

Wie es in der Praxis aussieht

Auf einem Host stand ein Ratenlimit für den SSH-Port ein Jahr lang in der Regelliste, ohne dass ein einziges Paket es erreichte. Eine weiter gefasste Freigabe darüber passte zuerst. Die Statuswerkzeuge führten das Limit das ganze Jahr auf, zu Recht: Es existierte. Die Prüfungen wurden auf die Position statt die bloße Anwesenheit der Regeln umgestellt. Sie vergleichen nun die Reihenfolge der nummerierten Regeln und der tatsächlich vom Kernel zum Filtern verwendeten Chain mit der beabsichtigten. Jeder Port bekommt eine Regel, jede Regel nennt ein Protokoll. SSH ist mit einer Ratenbegrenzung versehen und hat keine separate Freigabe. Die fünf öffentlichen Ports sind gemeinsam deklariert und werden als Menge geprüft. Ein offener, in der Deklaration fehlender Port lässt die Prüfung scheitern. Der Sicherheitsbericht listet die Regeln in der Reihenfolge auf, in der Pakete auf sie treffen, mit einem Hinweis zum Lesen dieser Reihenfolge.

Der erste Server wurde von Hand aufgebaut und erhielt seine Firewall erst später. Dadurch war ein frischer Rechner eine Zeit lang aus dem Internet erreichbar. Für einen zweiten Server bei einem anderen Anbieter wird die Firewall in einem eigenen Play bereits vor dem Anlegen des Rechners erstellt. Der neue Rechner trifft somit bereits mit aktiven Regeln auf das Internet. Dieser Anbieter ordnet Firewalls über Tags zu. Deshalb werden die Tags zuerst angelegt: Eine auf ein Tag gerichtete Regel braucht ein vorhandenes Tag. Jede Anbieter-Firewall wird mit einem einfachen HTTP-Task direkt über die REST-API des jeweiligen Anbieters konfiguriert. Der Task tut bei einem Dry Run nichts. Änderungen werden daher am Firewall-Play selbst erprobt, weil eine Anbieter-Firewall kostenlos angelegt und wieder gelöscht werden kann.

Container können an einer Firewall vorbeikommen, ohne sie zu berühren. Docker schreibt eigene Einträge in den Paketfilter des Kernels. Ein für einen Container veröffentlichter Port wird berücksichtigt, bevor die Regeln der Härtungsrolle des

Hosts greifen. Ein Container kann daher unabhängig von den Firewall-Vorgaben von außen erreichbar sein. Die Container dieser Plattform laufen mit Podman und Quadlet unter systemd, teilen das Netzwerk des Hosts und veröffentlichen keine Ports. Es gibt deshalb keinen Containerport, dessen Verkehr an den Regeln vorbeigeleitet werden könnte.

Wo es schiefgeht

Eine Regel steht in der Statusausgabe und greift nie. Regeln werden der Reihe nach ausgewertet. Der erste Treffer gewinnt, sodass eine breite Freigabe über einem engeren Limit oder Verbot dieses vollständig verdeckt. Die Liste zeigt weiterhin beide. Lesen Sie Regeln wie der Kernel, nummeriert und in Reihenfolge. Stellen Sie spezifische vor allgemeine Regeln und testen Sie von einem externen Rechner aus:

```
ufw status numbered
nft list ruleset
```

Ein Container antwortet auf einem von der Firewall gesperrten Port. Docker veröffentlicht Ports durch Umleiten des Verkehrs in der NAT-Tabelle des Kernels, bevor er die von ufw gefilterten Chains erreicht. Laut Dockers eigener Dokumentation ignoriert dies die Firewall-Konfiguration faktisch. Veröffentlichen Sie nur auf der Loopback-Adresse, setzen Sie eine Anbieter-Firewall davor oder betreiben Sie Container wie oben ohne veröffentlichte Ports.

IPv4 ist abgesichert, IPv6 steht weit offen. Die beiden Adressfamilien haben in iptables separate Regelsätze. ufw schreibt IPv6-Regeln nur, wenn IPv6 in seiner Voreinstellungsdatei aktiviert ist. Ein Host mit öffentlicher IPv6-Adresse und einem reinen IPv4-Regelsatz ist auf jedem lauschenden Port erreichbar. Verwenden Sie eine nftables-Tabelle der Familie `inet`, die beide sieht, oder prüfen Sie beide Regellisten. Scannen Sie die IPv6-Adresse des Hosts von außen.

Die SSH-Sitzung friert beim Aktivieren der Firewall sofort ein. Die standardmäßige Ablehnung wurde vor der SSH-Regel aktiviert, oder der SSH-Daemon wurde auf einen anderen Port verschoben, während die Regel noch 22 nennt. Fügen Sie zuerst die SSH-Regel hinzu, halten Sie die Konsole des Anbieters in einem anderen Fenster offen und planen Sie einen automatischen Rollback in einigen Minuten, den Sie nach erfolgreicher neuer Verbindung abbrechen.

Eine Regel öffnet mehr als beabsichtigt. Eine ufw-Regel, die einen Port ohne Protokoll nennt, öffnet ihn für TCP und UDP. Nennen Sie in jeder Regel das Protokoll und deklarieren Sie die erwartete Netzwerkerreichbarkeit als Ganzes. Alles Zusätzliche lässt dann eine Prüfung scheitern, statt bis zu einem Audit zu warten.

Weiterführende Lektüre

- Firewall — die Definitionen aus NIST-Veröffentlichungen, jeweils mit Link zum Ursprungsdokument.
- SP 800-41 Rev. 1: Leitlinien zu Firewalls und Firewall-Richtlinien — Firewall-Typen, standardmäßige Ablehnung und Regelverwaltung von NIST.
- ufw(8) — Ubuntus Firewall-Frontend, einschließlich Regelreihenfolge und `limit` -Regel.
- nftables-Wiki — die Dokumentation des netfilter-Projekts zum Framework, das iptables ersetzt.
- Paketfilterung und Firewalls — Dockers eigene Erklärung, wie veröffentlichte Ports die Host-Regeln umgehen.

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 Firewall?“ 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/firewall/