Engineer

Pare-feu

ensemble de règles décidant quel trafic peut passer

Ce que c’est

Un pare-feu est un filtre sur un trajet réseau qui compare chaque paquet ou connexion à une liste de règles, puis le laisse passer, l’abandonne ou répond par un refus. Ce peut être un logiciel sur la machine qu’il protège, un équipement distinct entre deux réseaux ou un service qu’un fournisseur de cloud place devant une machine virtuelle. Les administrateurs Linux parlent souvent de filtre de paquets pour la même chose, et les fournisseurs de cloud vendent des pare-feu sous des noms tels que groupe de sécurité. Les pare-feu sont généralement disposés en couches : l’un chez le fournisseur, l’autre sur l’hôte.

Les recommandations du NIST sur le sujet, SP 800-41 rév. 1 de septembre 2009, les classent selon ce qu’ils lisent du trafic. Un filtre de paquets examine adresses, protocole et numéros de port, un paquet à la fois. Un pare-feu avec suivi d’état mémorise aussi les connexions ; la réponse à une requête émise par le serveur lui-même rentre donc sans règle qui lui soit propre. Une passerelle mandataire applicative sert de point de terminaison à la connexion et lit le protocole qu’elle transporte. Un pare-feu sur l’hôte s’exécute sur le serveur ou le portable qu’il protège et sait à quel programme appartient une socket.

Sur un serveur Linux, le filtrage se fait dans le framework netfilter du noyau. nftables, disponible depuis le noyau 3.13, remplace les anciens outils iptables. Des interfaces telles qu’ufw écrivent les règles pour ceux qui préfèrent exprimer une intention plutôt que des chaînes. Quel que soit l’outil qui les écrit, le modèle reste le même : une liste ordonnée, lue depuis le début, où la première règle correspondante décide.

On confond souvent un pare-feu avec des outils voisins. Un pare-feu applicatif web lit les requêtes HTTP et bloque les motifs d’attaque qu’il y trouve ; un pare-feu réseau ne les examine jamais aussi profondément. Un outil de bannissement

des intrusions lit les journaux et ajoute des règles temporaires contre les adresses au comportement anormal ; il dépend donc d'un pare-feu sous-jacent. Un VPN décide qui appartient au réseau. Chacun répond à une question différente, et aucun ne remplace la simple liste de règles.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'un pare-feu ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des comptes, mots de passe et droits attribués selon les besoins exacts de chaque poste, puis retirés le jour du départ.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. 1 000 règles RBAC pour sécuriser Linux — 1,000 règles d'accès écrites pour combler une faille de sécurité

1 000 règles RBAC pour sécuriser Linux Contactez-nous →

Pourquoi c'est important

Chaque service qui écoute sur une adresse publique est une porte, et le pare-feu inventorie celles qui existent. La décision réelle d'une équipe a trois volets : l'endroit où se trouve le pare-feu, ce qu'il autorise sans règle explicite et les personnes qui peuvent le modifier.

Un pare-feu chez le fournisseur arrête le trafic avant qu'il atteigne la machine, continue de fonctionner si quelqu'un obtient les droits root sur l'hôte et ne consomme aucune ressource processeur du serveur. Ses règles vivent dans une console ou une API, hors de la configuration du serveur ; elles dérivent donc si elles ne sont pas écrites sous forme de code. Un pare-feu sur l'hôte suit la machine, peut appliquer une limitation de fréquence à un port donné et peut être testé depuis la machine elle-même. Un compte root compromis peut cependant le réécrire, tout comme un logiciel qui gère ses propres règles. Faire fonctionner les deux est courant et raisonnable, au prix du maintien de deux listes cohérentes : chaque ouverture ou fermeture de port exige deux modifications.

La politique appliquée sans règle explicite est le deuxième volet. Le NIST recommande de refuser tout trafic qui n'est pas expressément autorisé, et presque tout le monde s'accorde sur le trafic entrant. Le compromis devient sensible sur le filtrage sortant : il limite ce qu'un service compromis peut atteindre, mais casse les mises à jour de paquets, le renouvellement des certificats et la surveillance tant que chaque destination n'a pas sa règle.

Le troisième volet est la responsabilité. Un pare-feu qui refuse tout trafic non expressément autorisé ne peut être correct que si la liste des ports déclarés l'est aussi. L'élément utile est donc cette liste, conservée dans le gestionnaire de versions et comparée aux règles actives. Le logiciel est gratuit dans toutes les grandes distributions Linux ; ce qui coûte de l'argent est l'attention nécessaire pour que la déclaration reste fidèle à la réalité. Un pare-feu est par ailleurs une seconde ligne de défense. Un service lié à l'adresse de bouclage est inaccessible depuis l'extérieur quelles que soient les règles : la garantie est plus forte qu'une règle de refus devant un service qui écoute partout.

À quoi cela ressemble en pratique

Sur un hôte, une règle limitant la fréquence des connexions au port SSH est restée un an dans la liste sans recevoir un seul paquet, parce qu'une autorisation plus large, placée au-dessus, correspondait en premier. Les outils d'état l'ont affichée toute l'année, à juste titre : elle existait. Les vérifications ont été réécrites pour lire la position des règles plutôt que leur seule présence : elles comparent désormais l'ordre des règles numérotées, et celui de la chaîne réellement utilisée par le noyau pour filtrer, à l'ordre voulu. Chaque port a une règle et chaque règle nomme un protocole ; les connexions SSH sont limitées en fréquence, sans règle d'autorisation distincte. Les cinq ports publics sont déclarés ensemble et vérifiés comme un ensemble : un port ouvert absent de la déclaration fait donc échouer la vérification. Le rapport de sécurité énumère les règles dans l'ordre où les paquets les rencontrent, avec une note pour lire cet ordre.

Le premier serveur a été construit à la main et n'a reçu son pare-feu qu'ensuite, laissant un intervalle où une machine neuve était accessible depuis Internet. Pour un deuxième serveur chez un autre fournisseur, le pare-feu est créé par son propre play avant la machine, de sorte que la nouvelle machine est exposée à Internet avec ses règles déjà en place. Ce fournisseur attache les pare-feu par étiquette ; les étiquettes sont donc créées en premier, car une règle qui en cible une exige qu'elle existe. Le pare-feu de chaque fournisseur est configuré directement par son API REST, avec une simple tâche HTTP. La tâche ne fait rien lors d'une simulation ; les modifications sont donc testées en exécutant le play du pare-feu lui-même, puisqu'un pare-feu chez ce fournisseur se crée gratuitement et peut être supprimé ensuite.

Des conteneurs peuvent contourner un pare-feu sans le modifier. Docker écrit ses propres entrées dans le filtre de paquets du noyau, et un port publié pour un conteneur est pris en compte avant tout ce qu'a placé le rôle de durcissement de

l'hôte : un conteneur peut donc être accessible de l'extérieur, indépendamment des consignes données au pare-feu. Les conteneurs de cette plateforme fonctionnent sous systemd avec Podman et Quadlet, partagent le réseau de l'hôte et ne publient aucun port ; il n'y a donc aucun port de conteneur publié qui permette de contourner les règles.

Où les problèmes surviennent

Une règle apparaît dans l'affichage d'état et ne s'applique jamais. Les règles sont évaluées dans l'ordre et la première correspondance l'emporte ; une règle d'autorisation large placée avant une règle plus précise de limitation ou de refus la masque donc complètement. La liste affiche toujours les deux. Lisez les règles comme le noyau, numérotées et dans l'ordre, placez les règles précises avant les générales et testez depuis une machine extérieure :

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

Un conteneur répond sur un port refusé par le pare-feu. Docker publie des ports en redirigeant le trafic dans la table NAT du noyau, avant qu'il n'atteigne les chaînes utilisées par ufw pour filtrer. Sa documentation indique que cela revient à ignorer la configuration du pare-feu. Ne publiez que sur l'adresse de bouclage, placez un pare-feu du fournisseur devant ou exécutez des conteneurs sans ports publiés, comme décrit plus haut.

IPv4 est verrouillé et IPv6 grand ouvert. Les deux familles d'adresses ont des jeux de règles distincts dans iptables, et ufw n'écrit des règles IPv6 que si IPv6 est activé dans son fichier de réglages prédéfinis. Un hôte ayant une adresse IPv6 publique et des règles uniquement IPv4 expose chaque port sur lequel il écoute. Utilisez une table nftables de la famille `inet`, qui voit les deux, ou auditez les deux listes, puis analysez les ports de l'adresse IPv6 de l'hôte depuis l'extérieur.

La session SSH se fige dès l'activation du pare-feu. Le refus du trafic non expressément autorisé a pris effet avant la règle SSH, ou le démon SSH a été déplacé sur un autre port alors que la règle nomme encore 22. Ajoutez d'abord la règle SSH, gardez la console du fournisseur ouverte dans une autre fenêtre et programmez un retour arrière automatique dans quelques minutes, à annuler une fois une nouvelle session établie.

Une règle ouvre plus que son auteur ne le voulait. Dans ufw, nommer un port sans protocole l'ouvre pour TCP et UDP. Nommez le protocole dans chaque règle et déclarez l'ensemble de la surface d'exposition attendue, pour que tout élément supplémentaire fasse échouer une vérification au lieu d'attendre un audit.

Pour aller plus loin

- Pare-feu — les définitions des publications du NIST, chacune reliée à son document d'origine.
- SP 800-41 rév. 1 : recommandations sur les pare-feu et leur politique — les types de pare-feu, le refus du trafic non expressément autorisé et la gestion des règles, selon le NIST.
- ufw(8) — l'interface de pare-feu d'Ubuntu, avec l'ordre des règles et la règle `limit` .
- Wiki nftables — la documentation du projet netfilter sur le framework qui remplace iptables.
- Filtrage des paquets et pare-feu — l'explication de Docker sur la manière dont les ports publiés contournent les règles de l'hôte.



Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'un pare-feu ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/firewall/