Engineer

Privilège

🎟 Droit accordé à une personne ou à un processus

Ce que c’est

Un privilège est l’autorisation de faire ce qu’un système interdit normalement. Une personne ou un programme en cours d’exécution le détient parce qu’une autorité a décidé de le lui accorder : lire les fichiers d’un autre utilisateur, ouvrir un port inférieur à 1024, créer des comptes ou exécuter une fonction de base de données. On parle de privilèges au pluriel, car un sujet en possède rarement un seul, et de comptes ou processus « privilégiés » ou « non privilégiés » pour indiquer de quel côté de cette limite ils se trouvent. Le glossaire du NIST ajoute un élément facile à oublier : les privilèges décrivent le comportement autorisé d’un sujet, sont définis par une autorité et sont consignés dans une politique ou des règles.

Le mot côtoie deux autres notions. Le contrôle d’accès consiste à décider du sort de chaque demande ; un privilège est l’un des éléments concernant le demandeur sur lesquels cette décision s’appuie. Une permission correspond généralement à la même idée vue du côté de la ressource : le fichier indique qui peut le lire, et le lecteur détient le droit de le faire.

Sous Linux, le mot a un sens précis. Le manuel capabilities(7) décrit la distinction Unix traditionnelle : les processus privilégiés, qui s’exécutent avec un identifiant utilisateur effectif égal à 0 et sont dits root, contournent tous les contrôles de permission du noyau, tandis que chaque autre processus est contrôlé selon ses attributs d’identité. Depuis Linux 2.2, le noyau divise le pouvoir de root en unités distinctes, les capacités, que l’on peut activer et désactiver une par une. Un programme qui doit seulement ouvrir un port bas peut ainsi recevoir `CAP_NET_BIND_SERVICE`, et rien d’autre. La page credentials(7) explique quelle identité est contrôlée : l’identifiant utilisateur réel indique à qui appartient un processus, l’identifiant utilisateur effectif est celui que le noyau compare

aux permissions, et l'identifiant utilisateur sauvegardé permet au programme de mettre de côté une identité aux droits élevés puis de la reprendre ultérieurement.

Les bases de données et les organisations utilisent le même mot pour leurs propres autorisations. Un rôle de base de données détient des privilèges sur les schémas, les tables et les fonctions. Un compte administrateur possède des privilèges qu'un compte du personnel n'a pas. L'idée est identique à chaque niveau : un droit, accordé délibérément, de franchir une limite autrement fermée.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'un privilège ? » 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

Les privilèges d'un programme déterminent l'étendue des dégâts qu'il peut causer lorsqu'il dysfonctionne. C'est l'objectif déclaré des capacités : un programme compromis fait moins de dégâts avec des droits limités que le même programme exécuté en root. Un serveur web atteint d'une faille d'exécution de code et doté des privilèges root livre toute la machine à qui découvre la faille ; la même faille dans un processus qui peut seulement lire son propre répertoire et ouvrir un port donne accès à bien peu de choses.

Le NIST formalise cette idée sous le nom de moindre privilège. La mesure AC-6 de SP 800-53 demande à une organisation de n'autoriser que les accès dont les utilisateurs et les processus qui agissent pour eux ont besoin pour accomplir leurs tâches. Ses renforcements en font un travail courant : les administrateurs utilisent un compte non privilégié pour les tâches quotidiennes, comme la messagerie ; les comptes privilégiés sont réservés aux personnes qui en ont besoin ; l'utilisation des fonctions privilégiées est journalisée ; les privilèges sont réexaminés et retirés lorsque leur nécessité ne peut plus être démontrée.

Les réglages par défaut en sont rarement là. Dans systemd, un service système sans ligne `User=` s'exécute en root, et `NoNewPrivileges=` , le réglage qui empêche un service et ses enfants d'acquérir davantage de privilèges par `execve()` , est désactivé tant que personne ne l'active. Les outils de base de données se connectent souvent comme propriétaire de toutes les tables, parce que c'est ainsi que le schéma a été créé. Les comptes des nouveaux salariés sont souvent copiés sur celui d'un collègue, droits compris. Chaque réglage par défaut est commode le premier jour et coûteux le jour d'une intrusion.

La contrepartie est réelle. Restreindre les privilèges prend du temps, car il faut déterminer ce que le programme fait réellement, et des droits trop étroits provoquent des défaillances à l'exécution plutôt qu'à l'installation. L'équipe doit généralement décider où investir ce temps en premier : dans les composants exposés à Internet, ceux qui conservent des données dont la perte serait regrettée, et les comptes capables de modifier les autres comptes. Pour un acheteur ou un auditeur, la question utile est simple et directe : qui et quoi détient ici des droits d'administration, et pourquoi ?

À quoi cela ressemble en pratique

Sur l'hôte de l'entreprise, treize fichiers d'unité s'exécutaient avec les privilèges accordés par défaut par le superviseur, soit la plupart des privilèges. Chaque modèle d'unité a reçu un plafond de privilèges fixé à ses propres droits, un filtre d'appels système, des familles d'adresses restreintes et une mémoire accessible en écriture ou en exécution, mais jamais les deux à la fois. Le score de durcissement du mécanisme de l'homme mort est passé de 9,6 à 1,5, celui de la forge git de 8,3 à 1,5 et ceux des deux unités de sauvegarde de 9,6 à 2,3 et 2,5. Une directive était mal orthographiée dans trois modèles : le superviseur journalisait une clé inconnue, puis démarrait tout de même l'unité. Désormais, chaque unité passe donc par le vérificateur du superviseur à chaque convergence. La page précise aussi ce que le score ne montre pas : une directive qui empêche une unité de fonctionner ne le fait qu'à l'exécution.

Sur une plateforme applicative, l'API se connecte à la base avec un rôle aux privilèges minimaux qui ne détient que le droit EXECUTE sur les fonctions de l'application, exécutées en SECURITY DEFINER avec des requêtes paramétrées. Les habilitations — niveau d'offre, rôle, organisation et périmètre de navires — sont revérifiées sur le serveur à chaque requête, et les restrictions de l'interface sont considérées comme de simples éléments d'expérience utilisateur.

Dans une organisation, les identités et les accès ont été regroupés sur M365, 1Password, Red Hat SSO et OKTA SSO, le moindre privilège a été imposé et les arrivées et départs ont été standardisés, pour révoquer rapidement les accès plutôt que les laisser subsister après le départ d'une personne.

Où les problèmes surviennent

Un service s'exécute en root parce que son unité ne précise rien. Pour un service système, le superviseur utilise root par défaut : une unité sans ligne `User=` détient donc tous les privilèges de la machine. Attribuez à chaque service son propre utilisateur, ou laissez systemd en allouer un avec `DynamicUser=`, qui crée un utilisateur au démarrage de l'unité et le libère à son arrêt.

Une capacité accorde en fait l'essentiel des pouvoirs de root. `CAP_SYS_ADMIN` couvre tant d'opérations que son propre manuel la qualifie de surchargée et de « nouveau root ». Une équipe qui l'accorde pour débloquer une fonctionnalité a accordé l'essentiel des pouvoirs de root. Recherchez la capacité précise dont le programme a besoin et limitez l'ensemble maximal des capacités à cette liste :

```
[Service]
User=web
NoNewPrivileges=yes
CapabilityBoundingSet=CAP_NET_BIND_SERVICE
AmbientCapabilities=CAP_NET_BIND_SERVICE
```

Ces réglages permettent à un service exécuté comme utilisateur ordinaire d'ouvrir le port 80, et toutes les autres capacités sont retirées de son ensemble maximal de capacités.

Un processus s'élève au-dessus de ses droits de départ. Un programme exécuté comme utilisateur ordinaire peut encore lancer un binaire set-user-ID ou un fichier doté de capacités, et le noyau élève alors ses privilèges pour cette exécution. Les défenseurs reconnaissent ce cas à un identifiant utilisateur effectif différent de l'identifiant réel du processus. `NoNewPrivileges=yes` ferme cette voie à un service et à tous les enfants qu'il lance.

Les droits s'accumulent au fil des changements de poste. Une personne change d'équipe et conserve ses anciens accès en plus des nouveaux ; après quelques années, son compte possède l'union des droits de tous les rôles qu'elle

a occupés. AC-6 y répond par des revues périodiques, et AC-2 par une gestion des comptes liée aux mutations et aux départs, afin qu'un changement de poste entraîne aussi un changement d'accès.

Le compte de connexion de l'application peut tout faire. Quand une API se connecte comme propriétaire du schéma, une seule faille d'injection donne accès à toutes les tables. Un rôle de connexion limité aux fonctions utilisées par l'application rend cette même faille beaucoup moins dommageable.

Pour aller plus loin

- Privilège dans le glossaire du NIST CSRC — la définition de CNSSI 4009, avec la précision qu'une autorité définit les privilèges.
- capabilities(7) — comment Linux divise root en capacités distinctes et comment les threads les détiennent.
- credentials(7) — les identifiants utilisateur réel, effectif, sauvegardé et de système de fichiers, et ce que chacun détermine.
- systemd.exec(5) — `User=`, `DynamicUser=`, `NoNewPrivileges=` et les réglages des capacités.
- NIST SP 800-53 Rev. 5 — la mesure AC-6, le moindre privilège et les renforcements qui l'inscrivent dans les pratiques courantes.

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 privilège ? » 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/privilege/