INFORMATION À LA SÉCURITÉ LINUX :

CADRES SELINUX ET SMACK

KATHY TUFTO, CHEF DE PRODUIT

Mentor®
A Siemens Business

BIVARNEC

SYSTÈMES EMBARQUÉS

INTRODUCTION

Avec la prolifération des appareils intelligents et de l'Internet des objets (IoT), la sécurisation des appareils connectés à Internet n'a jamais été aussi importante. Les développeurs de petites et grandes entreprises sont de plus en plus soucieux de protéger leurs appareils contre les attaques malveillantes. Le cabinet d'études Gartner prédit que plus de 8,4 milliards d'« objets » connectés à Internet seront utilisés dans le monde en 2017, soit une hausse de 31 % par rapport à l'année dernière. D'ici 2020, Gartner prévoit 20,8 milliards d'appareils connectés.

En outre, des attaques très médiatisées contre les principaux systèmes informatiques des entreprises et des gouvernements ont sensibilisé davantage le public à la sécurité des systèmes informatiques. Bon nombre de ces cyberattaques très médiatisées ont compromis les informations personnelles de millions de consommateurs, entraînant la réémission de millions de cartes de crédit dont les numéros avaient été volés par les cybercriminels à l'origine de ces attaques. Les vulnérabilités logicielles portant des noms tels que Rex, Mirai, Heartbleed et Shellshock sont devenues des termes familiers même en dehors des cercles informatiques. Pour ces raisons, entre autres, la création d'appareils sécurisés connectés à Internet n'a jamais été aussi importante.

Construire un système sécurisé implique de nombreux composants et couches de sécurité. De la sécurité physique à la protection contre les menaces basées sur le cloud, chaque aspect d'un système doit être évalué et protégé. Linux® propose plusieurs frameworks pour protéger le système d'exploitation et les composants associés. Ici, nous examinerons deux frameworks Linux populaires, SELinux et SMACK.

Security-Enhanced Linux (SELinux) a été intégré pour la première fois au noyau Linux open source (version 2.6) en 2003.
Simple Mandatory Access Control Kernel (SMACK) est le nouveau venu dans les cadres de sécurité Linux et a trouvé du terrain dans les appareils embarqués car il est plus compact et plus facile à administrer et à configurer. SELinux et SMACK sont tous deux des mécanismes permettant de protéger les ressources du système d'exploitation contre tout accès non autorisé à l'aide d'un mécanisme appelé Mandatory Access Control (MAC).

CONTRÔLE D'ACCÈS DISCRÉTIONNAIRE

Pour comprendre la différence entre les deux modes de contrôle d'accès, examinons d'abord le comportement du contrôle d'accès discrétionnaire (DAC). Les systèmes Linux standard utilisent un ensemble d'attributs d'accès qui font partie de chaque ressource du système de fichiers. Ces attributs régissent les autorisations d'accès pour une ressource de système de fichiers donnée. Ces autorisations incluent « Lire », « Écrire » et « eXecute » ou (RWX).

Les attributs existent dans trois catégories d'utilisateurs du système : l'utilisateur, les groupes et autres. La catégorie Utilisateur fait référence à un seul compte de connexion individuel sur un système. Les systèmes Linux et UNIX contiennent de nombreux groupes pour faciliter l'accès au partitionnement des ressources système. Par exemple, l'utilisateur Judy peut appartenir à deux groupes « Marketing » et « Administrateurs ».
La catégorie Autre fait référence à tout autre utilisateur du système au-delà de l'utilisateur actuel ou de tout groupe défini.

DAC accorde au propriétaire de la ressource le pouvoir de décider qui a accès à ces ressources. Il convient à la protection contre les violations d'accès accidentelles. La politique du CAD est centrée sur les utilisateurs. La politique de sécurité du DAC répond à ces questions :

1) Quelles ressources du système de fichiers cet utilisateur peut-il lire, écrire ou exécuter ? (Utilisateur)

2) Quelles ressources du système de fichiers ce groupe d'utilisateurs peut-il lire, écrire ou exécuter ? (Groupe)

3) Quelles ressources du système de fichiers sur le système tout le monde peut-il lire, écrire ou exécuter ? (Autre)

4) Qui peut exécuter des programmes (également des ressources du système de fichiers) sur ce système ? (Exécuter les autorisations)

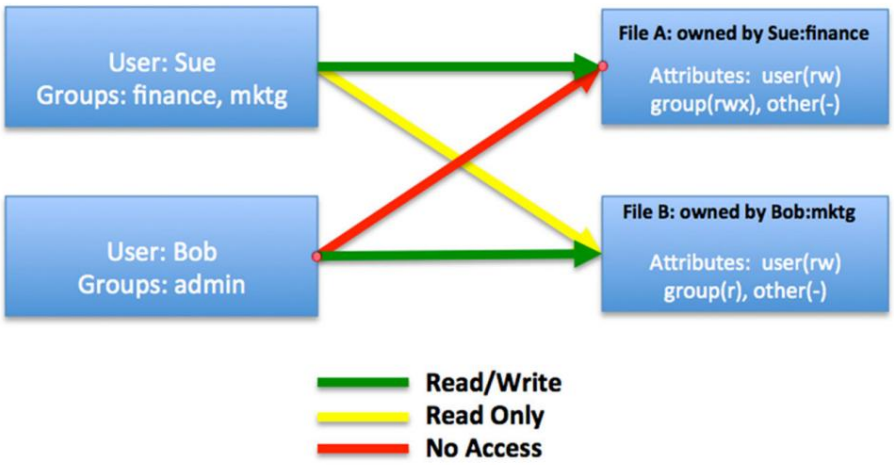

Figure 1 : Un scénario simple utilisant DAC.

La figure 1 illustre deux utilisateurs du système, Sue et Bob, qui disposent de différentes autorisations d'accès à une paire de ressources du système de fichiers. Le fichier A appartient à Sue et appartient au groupe « Finance » et possède les attributs Read (R) et Write (W) pour l'utilisateur Sue. Par conséquent, Sue dispose d'un accès en lecture/écriture à ce fichier. Étant donné que le fichier A n'a pas d'accès (-) pour « Autre » et que l'utilisateur Bob ne fait pas partie du groupe « Finance », il n'a pas accès à ce fichier.

De même pour le fichier B appartenant à Bob, Sue peut lire (R), mais pas écrire (W) dans le fichier B, car Sue appartient au « Marketing » et l'attribut de groupe du fichier B est défini sur Lecture (R). Bien entendu, chacun peut lire (R) et écrire (W) dans ses propres fichiers respectifs car les attributs utilisateur sont définis sur Lecture/écriture dans chaque cas.

La discussion autour de la figure 1 est un exemple classique de DAC et est utilisée dans pratiquement tous les systèmes Linux. La politique DAC permet aux utilisateurs de modifier la politique système – c'est de là que vient le terme « discrétionnaire ».

CONTRÔLE D'ACCÈS OBLIGATOIRE

Les règles de politique pour les systèmes Linux basées sur le contrôle d'accès obligatoire (MAC) sont contrôlées de manière centralisée par le système d'exploitation et ne peuvent pas être modifiées par les utilisateurs ordinaires du système. Un développeur de politiques contrôle quels programmes ou processus peuvent effectuer des opérations spécifiques sur les ressources système. Les utilisateurs ne peuvent pas modifier les autorisations d'accès accidentellement ou intentionnellement. La politique MAC se concentre sur les programmes plutôt que sur les utilisateurs, répondant à la question « Que peut faire ce programme ou ne pas faire ? » Les utilisateurs exécutent des programmes en leur nom au nom du système.

En revanche, les systèmes DAC régissent principalement l'accès aux ressources du système de fichiers, en attachant des attributs d'autorisation à un fichier, tels que Lire, Écrire et Exécuter. SELinux dispose d'un contrôle beaucoup plus fin régissant les droits d'accès aux ressources au-delà du système de fichiers pour inclure la mémoire, les E/S, les sockets, les éléments IPC, etc.

LINUX À SÉCURITÉ AMÉLIORÉE (SELinux)

Security-Enhanced Linux (SELinux) est un cadre et un ensemble d'outils développés à l'origine par la National Security Agency (NSA) des États-Unis et utilisés pour renforcer les systèmes Linux contre les menaces potentielles. Ces menaces peuvent inclure des attaques délibérées, une utilisation abusive ou des vulnérabilités logicielles, notamment des virus et des logiciels malveillants. SELinux a été initialement intégré au noyau Linux principal il y a plus de dix ans, au début du noyau Linux 2.6.
Bien qu'aucun framework ne puisse protéger contre certains bugs logiciels, SELinux a le potentiel de créer un système

beaucoup plus robuste et beaucoup moins vulnérable aux menaces externes, notamment les virus et les logiciels malveillants. SELinux est un outil important dans l'arsenal des analystes de sécurité et est utilisé comme élément clé d'une stratégie globale de sécurité du système.

SELinux par défaut n'autorise aucun accès. Des règles doivent être créées et chargées dans le système d'exploitation pour spécifier les droits d'accès autorisés. Avec SELinux, tous les accès doivent être explicitement accordés. L'ensemble de règles est appelé la politique SELinux.

Contrairement aux attributs d'autorisation DAC, SELinux utilise des attributs de contrôle d'accès attachés à chaque système de fichiers appelés contexte de sécurité. Les champs contextuels les plus couramment utilisés incluent l'utilisateur, le rôle et le type, généralement écrits sous la forme user:role:type. Chaque objet (processus, fichiers, etc.) possède un contexte de sécurité, notamment et plus particulièrement les processus et fichiers. Les règles de stratégie permettent à un processus dans un contexte d'effectuer des opérations sur un objet dans un autre contexte.

Les contextes SELinux pour les programmes pourraient ressembler à ceci :

/sbin/dhclient -> système _ u:objet _ r:dhcpc _ exécutif _ t:s0

/sbin/fdisque -> système _ u:objet _ r:mille _ t:s0

/etc/mot de passe -> système _ u:objet _ r:etc _ t:s0

Le s0 est appelé niveau et est utilisé pour les politiques de sécurité à plusieurs niveaux et autres, et est souvent ignoré dans les règles de politique. Ces trois programmes binaires sont étiquetés comme user:system_u et role:object_r. Chacun a son propre type basé sur son utilisation dans le système afin que les règles puissent être construites séparément pour chaque programme.

Les règles de politique doivent être écrites pour permettre l'accès à chaque objet du système. Les règles d'autorisation spécifient un sujet et un objet ainsi que les autorisations accordées pour un accès spécifique. Les règles d'autorisation comportent quatre éléments : source_type(s), target_type(s), object_class(es) et autorisation(s). Une règle d'autorisation de stratégie peut ressembler à ceci :

autoriser l'utilisateur _ poubelle _ t : fichier {lire exécuter getattr} ;

Cette règle spécifie qu'un processus avec le type de domaine user_t dispose de l'autorisation de lecture, d'exécution et « obtenir des attributs » (stat) sur un objet fichier de type bin_t. (Linux a un appel système getattr() qui permet à un processus d'obtenir les métadonnées d'un fichier sans le lire.)

SELinux est livré avec des utilitaires qui aident à gérer sa configuration. Par exemple, un outil d'audit enregistre chaque accès au système refusé. Un autre outil analyse le journal d'audit et crée une règle d'accès basée sur l'entrée du journal du refus. Un autre service public peut ajouter cette règle à la politique pour la rendre permanente. Cela facilite la détection et la résolution des problèmes d'accès dans les systèmes SELinux. De plus, un développeur n'a pas besoin de créer des règles à partir de zéro.
Il existe plusieurs politiques de référence, allant de minimales à complètes, qui constituent une bonne base à partir de laquelle créer votre propre politique SELinux personnalisée.

La puissance de SELinux vient de sa granularité. SELinux contrôle de nombreuses ressources du noyau, au-delà du modèle DAC consistant uniquement à contrôler les autorisations de fichiers. En raison de cette granularité, les politiques SELinux, même pour les systèmes simples, contiennent des centaines, voire des milliers de règles. Il est bien entendu que SELinux comporte à la fois une courbe d'apprentissage et une charge administrative importante.

NOYAU MAC SIMPLIFIÉ (SMACK)

Les complexités de SELinux ont donné naissance au noyau MAC simplifié (SMACK). SMACK utilise la même infrastructure de noyau sous-jacente que SELinux, mais réduit la granularité pour faciliter le développement, la configuration et l'administration du système. Il s'agit toujours d'un système MAC, ce qui signifie qu'il est régi par une politique centrale et non par les utilisateurs du système. Selon ses concepteurs, la simplicité était leur principal objectif de conception.

SMACK se compose de trois composants : un noyau compatible SMACK, des utilitaires SMACK et la politique de données de configuration. SMACK est basé sur des étiquettes attachées aux objets. La seule opération jamais effectuée sur une étiquette est de tester l'égalité. Chaque tâche sur un système basé sur SMACK se voit attribuer une étiquette. Des étiquettes spéciales sont attribuées aux tâches système telles que init. Plusieurs étiquettes spéciales ont une signification spécifique, comme l'astérisque en tant que caractère générique.

SMACK utilise les autorisations d'accès UNIX/Linux traditionnelles telles que la lecture, l'écriture et l'exécution. Les règles du SMACK sont très simples :

- Tout accès demandé par une tâche étiquetée « * » est refusé
- Un accès en lecture ou en exécution demandé par une tâche étiquetée « ^ » est autorisé
- Un accès en lecture ou en exécution demandé sur un objet étiqueté « _ » est autorisé
- Tout accès demandé sur un objet marqué « * » est autorisé
- Tout accès demandé par une tâche sur un objet de même label est autorisé
- Tout accès demandé et explicitement défini dans l'ensemble de règles chargé est autorisé.
- Tout autre accès est refusé

Les règles SMACK ont le format :

[étiquette-sujet] [étiquette-objet] [accès]

La chaîne d'accès utilise des caractères familiers aux administrateurs système Linux : r, w, x, pour lire, écrire, exécuter, etc.

Subject	Object	Access
TopSecret	Secret	rx
Secret	Unclass	r
Manager	Game	x
User	HR	rw
New	Old	r
Close	Off	-

Figure 2 : Règles sur un système SMACK typique.

En examinant la figure 2, nous voyons que tout sujet intitulé « TopSecret » souhaitant effectuer une opération sur un objet intitulé « Secret » peut le faire avec des autorisations de lecture et d'exécution. De même, un sujet intitulé « Manager » peut exécuter son objet préféré, à condition que cet objet porte l'étiquette « Jeu ». Les sujets étiquetés « Fermer » n'ont pas accès aux objets étiquetés « Désactivé ». SMACK est beaucoup plus facile à configurer, mais contrairement au SELinux plus puissant, il ne peut que modérer l'accès aux objets du système de fichiers.

RÉSUMÉ

SELinux est complexe et conçu pour une sécurité au niveau de l'entreprise. Même un système simple nécessite des milliers de règles individuelles pour fonctionner. Les autorisations SELinux sont beaucoup plus granulaires et peuvent donc protéger les ressources système telles que la mémoire, les E/S, les sockets, etc. SELinux est mature et dispose de bons outils pour la surveillance et le contrôle, le dépannage, l'examen des politiques et la maintenance.

SMACK a été conçu en pensant aux systèmes embarqués et pour sécuriser les appareils connectés à Internet. En conséquence, il est plus facile à configurer et à maintenir. Veuillez noter que les outils SMACK ne sont pas aussi riches que ceux disponibles pour une utilisation avec SELinux. Les autorisations ne sont pas aussi granulaires.

Quel que soit le cadre que vous choisissez, n'oubliez pas que le contrôle d'accès n'est qu'un élément d'un système sécurisé. La sécurité des appareils Linux embarqués doit être prise en compte dès la conception et peut être difficile à mettre en œuvre. Une chose est sûre : l'importance de protéger vos appareils contre les attaques malveillantes ne peut plus être secondaire.

La fonctionnalité SMACK abordée dans cet article est proposée dans le cadre de Mentor® Embedded Linux®.

Module complémentaire de sécurité.

Veuillez visiter le Mentor Embedded Linux site Web pour plus d'informations sur la sécurité ou pour des informations plus générales sur l'offre Linux de Mentor.

Biographie de l'auteur
Kathy Tufto est chef de produit à la division des systèmes embarqués de Mentor, responsable de Mentor Embedded Linux et de Mentor Embedded Sourcery CodeBench. Avant de rejoindre Mentor, Kathy a travaillé chez The MathWorks en tant qu'ingénieur de formation senior et développeur de cours senior où elle a enseigné et développé des cours dans le domaine de la simulation multidomaine, de la conception basée sur des modèles et de la génération de code embarqué pour les systèmes dynamiques et embarqués. Kathy est titulaire d'un diplôme en EE de l'Université de Boston.

Corporate Headquarters
Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070-7777
Phone: 503.685.7000
Fax: 503.685.1204

Sales and Product Information
Phone: 800.547.3000
sales_info@mentor.com

Silicon Valley
Mentor Graphics Corporation
46871 Bayside Parkway
Fremont, CA 94538 USA
Phone: 510.354.7400
Fax: 510.354.7467

North American Support Center
Phone: 800.547.4303

Europe
Mentor Graphics
Deutschland GmbH
Arnulfstrasse 201
80634 Munich
Germany
Phone: +49.89.57096.0
Fax: +49.89.57096.400

Pacific Rim
Mentor Graphics (Taiwan)
11F, No. 120, Section 2,
Gongdao 5th Road
HsinChu City 300,
Taiwan, ROC
Phone: 886.3.513.1000
Fax: 886.3.573.4734

Japan
Mentor Graphics Japan Co., Ltd.
Gotenyama Trust Tower
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo 140-0001
Japan
Phone: +81.3.5488.3033
Fax: +81.3.5488.3004

MGCVM17 TECH15490-W