Q&A - Hardening Linux

Question 1 :

Quel est le but principal de l’utilisation des contrôles d’accès obligatoires (MAC) comme SELinux ou AppArmor dans un environnement Linux ?

Réponse correcte :
Pour ajouter une couche de sécurité qui impose des politiques de contrôle d'accès basées sur des règles de sécurité strictes, réduisant la probabilité qu'un processus compromis puisse accéder à des ressources critiques du système en contournant les permissions utilisateur classiques.

Explication des réponses incorrectes :

- **Pour permettre une segmentation avancée des processus en fonction de leurs niveaux de sensibilité, tout en offrant des mécanismes de confinement pour limiter l'impact des attaques ciblant les services réseau :** Bien que la segmentation soit une conséquence possible de l’utilisation des MAC, ce n’est pas leur objectif principal. Leur but est de renforcer la sécurité en imposant des politiques strictes sur les actions des processus.
- **Pour renforcer les capacités de logging en associant chaque événement à un contexte de sécurité détaillé, permettant une traçabilité complète des actions effectuées par les processus système :** Les MAC ne sont pas principalement conçus pour améliorer la journalisation, bien que certains systèmes puissent inclure des éléments de traçabilité.
- **Pour optimiser les performances du noyau en répartissant dynamiquement les processus entre différents espaces d'adressage mémoire, tout en maintenant un contrôle strict sur les accès inter-processus :** Les MAC sont liés à la sécurité et non à l'optimisation des performances ou à la gestion de la mémoire.

Question 2 :

Quelle stratégie de gestion des journaux est la plus appropriée pour garantir l'intégrité des logs dans un environnement critique ?

Réponse correcte :
Mettre en place un système de journalisation centralisé où les logs sont transférés de manière sécurisée en utilisant des tunnels chiffrés, tout en appliquant des signatures numériques pour garantir que les logs n'ont pas été altérés lors de leur transit.

Explication des réponses incorrectes :

- **Utiliser une infrastructure de gestion des journaux basée sur la blockchain pour enregistrer chaque entrée de log de manière immuable, rendant toute modification non autorisée immédiatement détectable :** Bien que la blockchain puisse offrir une immutabilité, elle est souvent complexe à implémenter pour la journalisation standard, et cette approche n'est pas couramment utilisée dans les environnements critiques.
- **Configurer des politiques de rétention rigoureuses combinées à des sauvegardes régulières des logs avec une vérification des sommes de contrôle pour détecter toute tentative d'altération ou de suppression des journaux historiques :** Cette méthode aide à protéger les logs à long terme mais ne garantit pas l'intégrité des logs en temps réel, ce qui est crucial dans un environnement critique.
- **Déployer un serveur de logs résilient avec des disques en RAID et une redondance géographique pour garantir la disponibilité et la protection des logs contre les défaillances matérielles et les attaques malveillantes :** Bien que cela améliore la disponibilité des logs, cela ne se concentre pas spécifiquement sur l'intégrité ou la protection contre les altérations.

Question 3 :

Lors de la mise en œuvre d'un mécanisme de surveillance comme auditd, quel est l'impact de l'activation du mode audit d'enforcement (audit=1) dans le noyau Linux ?

Réponse correcte :
Il impose au noyau de journaliser toutes les tentatives d'accès et les modifications effectuées sur les fichiers critiques du système, tout en déclenchant des alertes pour les événements qui violent les politiques de sécurité prédéfinies, ce qui peut entraîner une augmentation significative de la charge CPU et de l'utilisation de l'espace disque.

Explication des réponses incorrectes :

- **Il active un mode de conformité stricte où chaque action effectuée par un utilisateur ou un processus est enregistrée avec un niveau de détail élevé, permettant une reconstruction complète des événements lors d'un incident de sécurité, mais avec un impact potentiel sur la performance globale du système :** Bien que cela soit en partie vrai, cette réponse exagère l'impact sur la performance et ne décrit pas entièrement le fonctionnement d'auditd.
- **Il transforme le système en un état où seules les opérations explicitement autorisées par les politiques d'audit sont permises, limitant ainsi les possibilités pour un attaquant de dissimuler ses traces, mais augmentant le risque de faux positifs qui peuvent interrompre les opérations normales :** Cette description correspond plus à un système de contrôle d'accès très strict, pas spécifiquement à l’audit d’enforcement.
- **Il configure le noyau pour effectuer une surveillance proactive, détectant et réagissant aux comportements anormaux des processus en temps réel, tout en enregistrant les événements dans une mémoire sécurisée pour éviter toute**

altération des logs en cas de compromission du système : Ceci décrit un système de détection d'intrusion plus avancé, et non la fonction spécifique du mode d'enforcement d'auditd.

Question 4 :

Dans un contexte de durcissement, pourquoi la configuration de points de montage avec l'option noexec est-elle recommandée pour certaines partitions (par exemple /tmp) ?

Réponse correcte :
Pour empêcher l'exécution de binaires et de scripts non autorisés sur des partitions temporaires, réduisant ainsi la surface d'attaque en limitant la capacité d'un utilisateur malveillant ou d'un malware à exploiter des vulnérabilités via des fichiers temporaires ou des scripts téléversés.

Explication des réponses incorrectes :

- **Pour renforcer l'intégrité des systèmes de fichiers en empêchant les modifications non intentionnelles des structures de données critiques, tout en garantissant que seuls les processus légitimes peuvent interagir avec les fichiers systèmes sensibles :** `noexec` n'empêche pas les modifications des fichiers, mais uniquement leur exécution.
- **Pour assurer que les données sensibles stockées dans les partitions temporaires ne soient jamais exécutées par le système, tout en forçant les processus à utiliser des chemins d'exécution sécurisés prédéfinis qui sont soumis à des politiques de contrôle d'accès plus strictes :** Cette réponse mélange des concepts, car `noexec` empêche seulement l'exécution, sans forcer l'utilisation de chemins sécurisés.
- **Pour permettre la segmentation des espaces de stockage entre les différents utilisateurs du système, en appliquant des politiques de restriction qui empêchent l'utilisation abusive des ressources partagées par des processus non autorisés :** `noexec` ne segmente pas le stockage ni ne gère l'accès aux ressources partagées.

Question 5 :

Quelle méthode serait la plus efficace pour détecter les changements non autorisés dans les fichiers systèmes critiques sur un serveur Linux ?

Réponse correcte :
Déployer un système de détection d'intrusion basé sur l'hôte (HIDS) qui surveille en temps réel les modifications des fichiers systèmes critiques, en utilisant des algorithmes de hachage sécurisés pour comparer les sommes de contrôle des fichiers avec des bases de données de références intègres, tout en générant des alertes

instantanées en cas de détection de divergence.

Explication des réponses incorrectes :

- **Configurer un mécanisme de journaux d'audit renforcé qui enregistre non seulement les changements des fichiers mais aussi les métadonnées associées (propriétaire, permissions, horodatages) :** Bien que cela puisse être utile, il ne surveille pas directement les modifications des fichiers en temps réel comme le ferait un HIDS.
- **Mettre en place un système de versionnement des fichiers systèmes critiques qui conserve des copies immuables des versions antérieures :** Le versionnement permet de restaurer des fichiers mais ne détecte pas activement les changements non autorisés en temps réel.
- **Utiliser une combinaison de signatures numériques et de cryptographie asymétrique pour valider l'intégrité des fichiers critiques lors de chaque accès :** Bien que les signatures numériques puissent garantir l'intégrité lors de l'accès, elles ne surveillent pas activement les modifications non autorisées en temps réel.

Question 6 :

Comment peut-on renforcer la sécurité de l'authentification SSH sur un serveur Linux utilisé pour l'administration critique ?

Réponse correcte :
Implémenter une authentification multi-facteurs (MFA) combinant une clé publique protégée par une phrase de passe complexe avec un jeton matériel ou une application mobile pour renforcer la vérification de l'identité de l'utilisateur avant l'établissement de la connexion SSH.

Explication des réponses incorrectes :

- **Configurer le démon SSH pour limiter les tentatives de connexion simultanées et activer un délai de backoff progressif pour chaque tentative échouée :** Bien que cela aide à prévenir les attaques par force brute, cela ne renforce pas autant la sécurité que l'ajout d'une authentification multi-facteurs.
- **Activer le logging détaillé des connexions SSH, incluant la capture des tentatives de login échouées avec des informations sur l'adresse IP source :** Le logging est utile pour la surveillance, mais ne renforce pas directement l'authentification SSH.
- **Déployer une politique de rotation des clés SSH régulièrement :** La rotation des clés est une bonne pratique, mais elle n'offre pas le même niveau de sécurité que l'authentification multi-facteurs.

Question 7 :

**Quelle stratégie peut être mise en place pour éviter la persistance de malware après un redém

arrage du système ?**

Réponse correcte :
Mettre en œuvre un schéma de partitions multiples avec des points de montage en lecture seule pour les répertoires critiques (comme /boot et /usr), combiné à un mécanisme de détection précoce des anomalies pendant le processus de démarrage, empêchant ainsi tout code malveillant de s'exécuter ou de s'injecter dans les services essentiels du système.

Explication des réponses incorrectes :

- **Configurer un processus de démarrage sécurisé (Secure Boot) avec des mesures de validation cryptographique :** Secure Boot garantit que seuls les éléments approuvés peuvent être chargés, mais il ne s'agit pas d'une stratégie directe pour gérer la persistance des malwares après le démarrage.
- **Déployer une stratégie de mise en quarantaine automatisée des services suspects lors du redémarrage :** Bien que cela puisse être utile, elle ne prévient pas la persistance des malwares mais plutôt la propagation de services suspects.
- **Utiliser un environnement de récupération immuable stocké hors ligne :** Cette méthode aide à restaurer l'état du système mais ne prévient pas la persistance du malware entre les redémarrages.

Question 8 :

Dans un système Linux durci, pourquoi est-il essentiel de désactiver les services non utilisés ?

Réponse correcte :
Pour réduire la surface d'attaque en minimisant les processus en écoute sur les ports réseau, limitant ainsi les vecteurs d'accès exploitables par des attaquants externes ou internes, et pour éliminer les risques de vulnérabilités associées à des services obsolètes ou non maintenus.

Explication des réponses incorrectes :

- **Pour renforcer la sécurité des processus critiques en réduisant la concurrence pour les ressources du système :** Bien que cela puisse être un effet secondaire, l'objectif principal de désactiver les services non utilisés est de réduire la surface d'attaque, et non de gérer les ressources.
- **Pour simplifier la gestion des mises à jour de sécurité :** Cela peut aider, mais l'objectif principal est de minimiser les points d'entrée pour les attaques potentielles, plutôt que de gérer les mises à jour.
- **Pour garantir que les politiques de contrôle d'accès (ACL) restent efficaces :** La désactivation des services n'a pas d'impact direct sur l'efficacité des ACL, mais elle réduit

les vecteurs d'attaque.

Question 9 :

Pourquoi est-il important de sécuriser le chargeur de démarrage GRUB sur un système Linux ?

Réponse correcte :
Pour empêcher les utilisateurs non autorisés de modifier les options de démarrage ou d'accéder à un shell root sans authentification.

Explication des réponses incorrectes :

- **Pour améliorer la vitesse de démarrage du système :** Sécuriser GRUB n'a aucun effet sur la vitesse de démarrage du système.
- **Pour empêcher la fragmentation des partitions :** GRUB n'est pas impliqué dans la gestion de la fragmentation des partitions.
- **Pour permettre une gestion plus facile des modules du noyau :** La sécurisation de GRUB ne concerne pas directement la gestion des modules du noyau.

Question 10 :

Quel est l'avantage principal de l'utilisation de LUKS (Linux Unified Key Setup) pour le chiffrement des disques sous Linux ?

Réponse correcte :
Il fournit un mécanisme standard pour le chiffrement complet des disques, garantissant la confidentialité des données même en cas de vol physique du disque.

Explication des réponses incorrectes :

- **Il permet de compresser les données sur le disque pour économiser de l'espace :** LUKS est conçu pour le chiffrement, pas pour la compression des données.
- **Il améliore les performances d'accès aux disques en réduisant le temps de lecture/écriture :** Le chiffrement via LUKS n'améliore pas les performances d'accès, et peut même les réduire légèrement.
- **Il simplifie la gestion des utilisateurs et des groupes sur le système :** LUKS gère le chiffrement des disques, pas la gestion des utilisateurs et des groupes.

Revision #2
Created 3 September 2024 20:58:17 by Alek
Updated 6 September 2024 08:19:49 by Alek