Rapport du Projet : Hardening as Code

Automatisation du Durcissement des Systèmes Linux par le Code : Une Approche Hardening as Code

Réalisé par :
OUBDA Dimitri-Gaetan

SANOGO Cheickna

Encadré par :
Pr. Omar ACHBAROU

1. Auteurs

- **OUBDA Dimitri-Gaetan**
- **SANOGO Cheickna**
  Étudiants en 4ème année de Génie Cyber Défense et Systèmes de Télécommunications Embarqués à l'ENSA Marrakech.

2. Résumé / Abstract

Résumé

Ce projet vise à développer une solution automatisée pour durcir les systèmes d'exploitation Linux en appliquant des scripts personnalisés conformes au guide "Recommandations de configuration d'un système GNU/Linux" de l'ANSSI. En adoptant une approche Hardening as Code, nous avons créé un fichier principal structurant le durcissement en quatre couches (noyau, matériel, système et services). Cette solution se distingue par son accessibilité, son open-source et sa simplicité, offrant une alternative évolutive et reproductible pour répondre aux besoins de cybersécurité croissants. Elle cible une meilleure sécurité en minimisant les vulnérabilités et garantit une conformité accrue aux normes.

Abstract

This project aims to develop an automated solution for hardening Linux operating systems using custom scripts aligned with the "Recommandations de configuration d'un système GNU/Linux" guide by ANSSI. By adopting a Hardening as Code approach, we created a main script organizing the hardening process into four layers (kernel, hardware, system, and services). This solution distinguishes itself by being accessible, open-source, and user-friendly, offering a scalable and reproducible alternative to meet growing cybersecurity demands. It enhances security by minimizing vulnerabilities and ensures improved compliance with standards.

3. Introduction Générale

Ce projet s'inscrit dans le cadre du module **Durcissement des Systèmes Informatiques**, qui vise à enseigner et à appliquer les principes fondamentaux de sécurité des systèmes d'exploitation tout en développant une expertise technique avancée.

Le **durcissement des systèmes** consiste à réduire les surfaces d'attaque en configurant les systèmes de manière à minimiser les vulnérabilités exploitables. Cette démarche inclut des mesures telles que la gestion des droits d'accès, la sécurisation des services essentiels et l'application de politiques strictes de contrôle d'accès, visant à garantir une résilience accrue face aux menaces.

Pourquoi le durcissement des systèmes est-il essentiel ? La croissance exponentielle des cyberattaques exploitant des configurations mal sécurisées souligne l'importance d'une approche proactive et structurée. Les approches manuelles, bien qu'efficaces pour des environnements limités, présentent des inconvénients notables : elles sont chronophages, sujettes à des erreurs humaines, et difficilement reproductibles dans des environnements complexes.

C'est dans ce contexte que notre approche **Hardening as Code** intervient : elle automatise et standardise le processus de durcissement, offrant une solution pragmatique et moderne adaptée aux défis actuels de la cybersécurité.

4. Motivation et Contexte

Motivation

- La montée en puissance des cyberattaques ciblant des systèmes mal configurés, nécessitant des solutions robustes et fiables.
- Le besoin d'une approche évolutive et reproductible pour les environnements variés, allant des petites entreprises aux infrastructures critiques.

Contexte Historique

Depuis l'émergence des outils d'automatisation en sécurité IT, tels qu'Ansible, Puppet ou Chef, les pratiques de durcissement des systèmes ont considérablement évolué. Cependant, ces outils, bien que performants, requièrent souvent des connaissances approfondies et des ressources importantes. Notre projet propose une alternative basée sur des scripts personnalisés, spécialement conçus pour être accessibles, pratiques et alignés avec les recommandations de l'ANSSI, tout en éliminant les contraintes liées aux frameworks complexes.

5. Travaux Connexes

Plusieurs solutions de durcissement existent aujourd'hui :

- **Ansible** : Utilise des playbooks pour appliquer des configurations prédéfinies. Bien que très répandu, son implémentation peut être complexe pour des projets de taille modeste.
- **Puppet** et **Chef** : Ces solutions offrent des fonctionnalités avancées mais sont souvent coûteuses et nécessitent une infrastructure dédiée.

Notre solution se distingue par sa gratuité, sa simplicité et sa capacité à être facilement personnalisée pour des besoins spécifiques, tout en s'alignant strictement sur les normes de l'ANSSI.

6. Travail Proposé

6.a. Architecture du Système

Le projet repose sur une architecture modulaire organisée en quatre couches :

1. **Couche Noyau (Kernel)** : Gestion des paramètres critiques du noyau.
2. **Couche Matériel** : Configuration sécurisée des équipements et du démarrage.
3. **Couche Système** : Gestion des partitions, des comptes d'accès et des fichiers sensibles.
4. **Couche Services** : Configuration des services essentiels et des modules PAM.

6.b. Fonctionnement Détaillé

Le fichier principal (« main ») propose des menus interactifs pour chaque couche, détaillant les options suivantes :

Couche Kernel

1. Chargeur de démarrage
2. Configuration dynamique
    - Sous-couches :
        - Configuration de la mémoire
        - Configuration du noyau
        - Configuration des processus
        - Configuration du réseau
        - Configuration des systèmes de fichiers
3. Configuration statique

Couche Matériel

1. Mise à jour du système
2. Mise à jour de l'ordre de démarrage
3. Activation du Secure Boot

Couche Système

1. Mise à jour du système
2. Partitionnement

    - Configurer les points de montage et les options recommandées
    - Chiffrer une partition (VeraCrypt ou LUKS)
3. Gestion des comptes d'accès
    - Comptes utilisateurs
    - Comptes administrateurs
    - Comptes de service
4. Contrôle d'accès
    - Fichiers sensibles
    - Sockets et pipes IPC nommés
    - Droits d'accès
5. Fichiers et répertoires
    - Restriction d'accès aux fichiers sensibles
    - Restriction d'accès aux sockets et pipes
    - Vérification des droits d'accès

Couche Services

1. Paramètres généraux
2. Configuration de PAM

7. Résultats Expérimentaux

7.a. Résultats et Analyses

Une vidéo de démonstration illustrant les tests effectués sera jointe pour valider l'efficacité et la simplicité d'utilisation de la solution.

7.b. Discussion

Points positifs

- Automatisation complète du processus de durcissement.
- Conformité stricte avec les recommandations de l'ANSSI.
- Solution gratuite, open-source et hautement personnalisable.

Points négatifs

Actuellement, aucune limitation majeure identifiée. Cependant, des améliorations peuvent être envisagées pour accroître la flexibilité et l'adaptabilité de l'outil.

8. Conclusion et Perspectives

Conclusion

Ce projet démontre la pertinence et l'efficacité de l'approche Hardening as Code pour sécuriser les systèmes Linux. En intégrant des scripts bien structurés et conformes aux normes, nous avons simplifié le processus de durcissement, tout en assurant une reproductibilité et une évolutivité essentielles dans des environnements modernes.

Perspectives

- Étendre la solution à d'autres systèmes d'exploitation (Windows, macOS).
- Intégrer l'option de choix de normes spécifiques pour une flexibilité accrue.
- Ajouter une fonctionnalité de mesure de conformité permettant d'évaluer le pourcentage de sécurité atteint selon les standards appliqués.
- Enrichir l'outil avec une interface utilisateur graphique pour simplifier son adoption par des utilisateurs non techniques.