Etat de l'art :
Les techniques et méthodes d'élévation de privilèges sur Linux

Florian TURMEL, École Nationale Supérieure d'ingénieurs de Bretagne Sud

Les élévations de privilèges sont souvent une porte d'entrée lors d'attaques sur des systèmes d'informations. Ce document réalise un état de l'art des différentes méthodes de défense et de protection contre ces élévations de privilèges, et les différentes manières de s'en protéger, au niveau des correctifs, au niveau des droits des applications, ou encore au niveau du noyau. Cet article traite en deux parties des méthodes de prévention, en parlant de protection du kernel ou de gestion des mots, et de détection, en évoquant la détection de tentative d'élévation de privilèges au niveau applicatif, ou en utilisant les méthodes de séparation de privilèges. Le document traite également de la méthodologie utilisée pour obtenir un état de l'art du sujet, en partant de la recherche initiale, jusqu'au méthode raffinage des documents.

1. Introduction

La veille technologique est un aspect important du travail d'ingénieur. Chaque corps de métier évolue sans cesse, et il est important de constamment se documenter, renouveler ses sources, connaître les recherches effectuées sur différents thèmes à son métier, et s'informer des différentes avancées technologiques et scientifiques réalisées. Le milieu de l'informatique est plus particulièrement de la cybersécurité sont des domaines où avoir une veille technologique active est essentielle est nécessaire, car de nouvelles vulnérabilités et de nouvelles méthodes d'intrusions sont trouvés chaque jour. Des attaques sont également réalisées ou mises au jour constamment, et il est devenu acquis qu'aucun système n'est inviolable, et que seul des ressources comme le temps et l'argent peuvent limiter le temps pour réaliser une attaque. Avec les moyens nécessaires, tout système, même le plus protégé, peut être pénétré. La veille technologique s'inscrit donc pleinement dans cet objectif d'être en tout temps informé des évolutions, et de se rapprocher au maximum de l'état de l'art de son corps de métier. Cette veille peut être réalisé de différentes manière, en fonction du but recherché : une veille technologique sur les dernières vulnérabilités et les dernières attaques pourra se réaliser sur les réseaux sociaux, ou sur les revues régulières d'organisations certifiées comme l'ANSSI ou MITTRE ATT&CK, tandis qu'une veille plus scientifique, dénombrant en précisions et avec applications les thèmes abordés se fera dans la plupart du temps via la sortie d'articles scientifiques. Lesdits articles doivent cependant être des sources sûres en matière d'information, par exemple en possédant des DOI, en ayant une renommée dans le milieu, ou encore en étant validés par ses pairs. Ce document s'inscrit pleinement dans cette continuité, en réalisant un état de l'art d'un sujet, en prenant comme ressources des documents et des articles certifiés et validés.

De nos jours, la technologie est partout. Nous utilisons chaque jour de nombreux appareils et un nombre incalculable de logiciels. Cependant, des tentatives d'attaques sont réalisées à chaque instant, et plusieurs étapes sont nécessaires pour prendre le contrôle d'une application, d'un appareil ou d'un système d'information. Nous nous intéresserons dans ce document à une partie importante de ces attaques : les élévations de privilèges, et nous réaliserons un état de l'art de ce thème. Nous essaierons de nous répondre à la question « Quelles sont les méthodes et outils de prévention et de détection des élévations de privilèges sur Linux ? » Nous parlerons des méthodes de prévention et de détection de tentatives d'attaques par élévation de privilèges. J'évoquerai le cas de la défense du noyau Linux, la

gestion des privilèges pour garder une sécurisation maximale, la gestion du chiffrage des mots de passe, les analyses graphiques comme nouveaux moyens de protection, et la détection d'attaques sur les applications. Je présenterais pour commencer de la méthodologie que j'ai utilisée, la recherche des documents, les outils et sites internet utilisées, le filtrage des documents, le choix de thèmes à aborder et enfin le choix de la problématique visées.

2. Méthodologie

Pour commencer cet état de l'art, il me paraît nécessaire de nommer les différents outils que j'ai utilisé lors de mes recherches. Le choix de ces outils est en effet important, car ils révéleront de la pertinence, de la cohérence, de la validité et de la valeur scientifique des documents et articles utilisés. Ces outils se subdivisent en 2 catégories : les outils matériels, qui m'ont permis d'accéder aux documents, ainsi que les outils logiciels, permettant la recherche, le stockage et la création de bibliographies.

2.1. Outils Matériels

2.1.1. Ordinateurs de l'Université

L'avantage principal à travailler sur les ordinateurs de l'Université, c'est qu'ils possèdent, grâce à leur statut d'Université, un accès gratuit pour les étudiants et sans limite à la plupart des ressources intéressantes dans le cadre d'une veille technologique, et le principal de ces outils : les sites de référencements d'articles scientifiques, comme IEEE, ACM ou Springer permettant l'accès gratuit à des documents normalement très chers. Nous utiliserons le plus possible cet accès que donne l'Université.

2.1.2. Ordinateurs personnels

Cependant, travailler sur les ordinateurs de l'Université ne permet pas une portabilité des articles. Si aucune solution n'existait, le travail dans un autre lieu serait impossible, ce qui n'est pas très confortable. Heureusement, l'Université propose l'accès à un proxy, permettant de simuler l'accès sur le campus, pour avoir accès à ces outils, même chez soi ou n'importe où. De plus, en combinant cela avec l'utilisation d'outils de sauvegarde de pages HTML comme Zotero que nous verrons par la suite, nous sommes capables de lire et exploiter nos documents depuis n'importe où.

2.2. Outils logiciels

2.2.1. Outils de bibliographie

Zotero

Cet outil est l'outil le plus important de cet état de l'art. Ce petit logiciel à plusieurs buts. Le premier est la sauvegarde de documents, ou de pages HTML. En effet, Zotero permet la réalisation de « snapshots », une sauvegarde à un instant T, évitant ainsi les incohérences entre différentes versions de sites internet. En sauvegardant les pages, il est possible de retrouver la publication originale. La date de ces snapshots sera également mentionné dans la bibliographie, qui est la seconde application principale de Zotero. En effet, l'outil permet la classification des documents dans des dossiers, aidant la réalisation de bibliographie, car l'outil réalise également automatiquement une bibliographie précise et détaillée de tous les documents et pages choisies. Elle permet aussi la conversion de ces derniers dans différents formats, pour convenir aux normes des sites internet, et l'implémentation dans des templates, en utilisant Latex par exemple.

2.2.2. Outils de référencement de documents

De très nombreux outils de référencement existent, mais finalement que peu ne m'ont vraiment été très utiles. Je ne citerai donc pas la presse papier, APA de Scribbr, ou encore CORE. Au cours de mes recherches, j'ai parfois trouvé certains doublons dans les documents, lorsque j'étais confronté à ce problème, j'ai préféré choisir de conserver les documents IEEE, de manière totalement arbitraire, préférant l'aspect de ce site.

Google Scholar

Google Scholar est un moteur de recherche créé par Google, permettant la recherche de documents publiés dans des revues scientifiques. C'est un outil répertoriant la plupart des bases de données « connus » de reçues scientifiques, il m'a donc été très utile.

IEEE Xplore

IEEE Xplore a été le deuxième moteur de recherche que j'ai utilisé. Lors de recherches sur Google Scholar, j'ai remarqué que les articles provenant de la base de données de IEEE étaient très pertinents, bien qu'il y en très peu. J'ai donc décider d'utiliser directement le moteur de recherche de la plateforme pour trouver plus facilement ces documents, et en effet, avec les bons filtres, j'ai pu trouver quelques documents que je n'avais pas trouvé sur Google Scholar.

ACM

J'ai également utilisé ACM dans le but de diversifier un peu plus les sources de mes documents. Cependant, je n'ai pas vraiment trouvé de documents allant dans le sens de ma veille. J'ai pu en extraire quelques-uns, mais je les ai rapidement mis de côté pour m'attarder sur les documents vraiment utiles dans mon état de l'art.

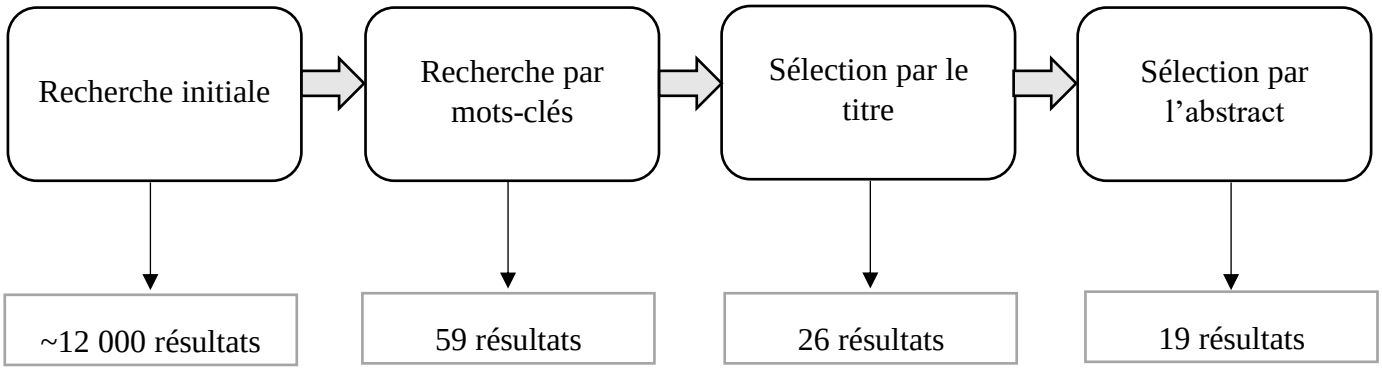
2.3. Recherche et affinage des documents

Pour rassembler les documents de mon état de l'art, il a été nécessaire de trouver de bonne méthode permettant de choisir efficacement les bons documents, pour en déterminer rapidement leur pertinence, et pouvoir facilement retirer les articles ne se concentrant pas sur le bon aspect de la veille, ou ceux hors sujet.

2.3.1. Recherche par mots clés

Une recherche sur Google Scholar ou IEEE Xplore peut s'avérer fastidieuse, tant le nombre de documents est important. C'est en effet l'un des problèmes des moteurs de recherche : il est assez difficile de trouver précisément des documents lorsque notre idée est précise. Les moteurs de recherches nous proposent cependant des outils permettant d'affiner ces recherches, et notamment le plus important : les filtres. La première étape a donc été de sélectionner nos mots clés, puis de les placer dans les filtres pour affiner le nombre de nos documents, et ainsi corriger notre surcharge d'article. L'utilisation de « Privilege Escalation », « Vulnerability », « Exploits » ou encore « Prevent » m'a permis d'afficher dans un premier temps un nombre très réduit de résultats. Ensuite, j'ai pu utiliser les exclusions, qui m'ont permis de retirer des recherches certains articles hors sujets, comme les articles parlants des systèmes d'exploitation n'étant pas Linux, car cet état de l'art se concentre uniquement sur les systèmes Linux. Enfin, j'ai pu utiliser le filtre permettant de chercher des mots clés uniquement dans le titre et/ou l'abstract, ce qui m'a permis de retirer les articles nommant les élévations de privilèges, mais sans les approfondir.

Après cette première recherche, mon nombre d'article s'est largement réduit, et j'avais désormais une soixantaine de résultats. Pour affiner encore mes recherches, j'ai dû trier une seconde fois les documents, en lisant tout d'abord le titre, ce qui enlevait les articles n'étant pas du tout orienté vers mon sujet. Enfin, j'ai supprimé les derniers documents encore hors sujets en lisant les abstracts des documents. La totalité de ces étapes m'a permis de finalement passer d'environ douze mille résultats à 19 documents, ce qui est largement moins (0,16% de la recherche initiale), et ce qui aide la prochaine étape de sélection.



2.3.2. Sélection par lecture

La deuxième étape de cette recherche de documents et la sélection des documents finaux par leur lecture. Lors de cette partie, nous allons lire les documents, avec une méthode particulière, pour en extraire les sujets principaux, et classer les documents selon leur sujet, leur structure, leur validité et leur précision vis-à-vis de l'état de l'art.

Pour la lecture des articles, j'ai choisi de me baser sur la méthode de lecture de S. Keshav, décrite dans l'article *How to Read Paper by S. Keshav* [1] décrivant une méthode rapide permettant de choisir ou non de lire un article. La méthode explique qu'il existe trois lectures d'un document : la première étant un balayage rapide du document, pour obtenir une idée générale de l'article. Ensuite, si cette le document n'est pas retiré, nous pouvons réaliser une deuxième lecture, une lecture en « diagonale » de l'article, servant à saisir le contenu du document. Cette lecture permettra de déjà voir si l'article est bien dans la ligne de conduite que nous voulons aborder. Enfin, si le document convient et qu'une troisième lecture est nécessaire, nous pouvons la réaliser, ce qui permettra de comprendre en profondeur les différents sujets abordés, et de pouvoir relater en détail le contenu de l'article.

En utilisant cette méthode, j'ai pu tout d'abord supprimer les documents non nécessaires, et dans un second temps j'ai pu ressortir de tous les documents des thèmes et enjeux principaux. Cela m'a permis de les classer en fonction de ces thèmes, et de choisir les thèmes que je vais garder, et ceux que je choisirai de ne pas traiter. Finalement, après la réalisation de plusieurs tableaux et graphiques, j'en ai venu à la conclusion que je choisirai la problématique « Quelles sont les méthodes et outils de prévention et de détection des élévations de privilèges sur Linux ? » en n'utilisant qu'au finale 11 articles sur les 19 présent avant cette étape.

Ces deux étapes terminées, j'ai donc pu finalement réaliser proprement mon état de l'art, grâce à des documents minutieusement choisis, en fonction de critères logiques et déterminés à l'avance. J'ai donc pu correctement ce document.

2.3.3. Diversité des sources

Malheureusement, mes sources ne sont que peu diversifiées, mais la recherche de documents présentant ce sujet provenant de sources plus larges a été difficile : manque de validité, peu de cohérence réelle avec le sujet, article trop porté commercialement, ou techniquement trop complexe pour la référence. De plus, le fait de trouver des doublons, et de prioriser IEEE donne comme résultat une plus petite diversité des références.

3. Définitions

Avant de commencer cet été de l'art, il est essentiel de poser les bases en revoyant quelques définitions.

Elévation de privilèges

L'élévation de privilèges est le fait d'obtenir des droits plus importants que normalement. Ce terme est assez connu, mais il faut différencier les deux types d'élévation de privilèges : l'élévation verticale et horizontale. La première est la définition d'élévation dans l'imaginaire commun, et a pour but d'augmenter ses privilèges en suivant une courbe de simple utilisateur, à administrateur ayant le plus de droits (souvent l'administrateur de domaine), en passant par administrateur locale et administrateur réseau. Cette élévation est la plus populaire et la plus utilisée. Elle est également, dans la plupart des cas, la plus efficace. Cependant, il faut la différencier de l'élévation horizontale, dont le but est de se déplacer vers un autre utilisateur. Cette méthode est bien moins utilisée, car bien plus situationnelle est précise, le but étant souvent d'usurper une identité ou d'accéder à des documents confidentiels par exemple.

Environnements d'Applications de Confiance (TEE)

Un TEE est une zone sécurisée situé dans le processeur. Son rôle est de garantir que les informations (souvent du code) chargés à l'intérieur sont protégés en matière de confidentialité et d'intégrité. Le TEE permet donc d'empêcher les entités non autorisées de modifier les données lorsqu'une entité extérieure au TEE traite les données, en ne permettant pas la modification de code du TEE sans autorisation de la part de ce dernier. Le TEE possède de nombreuses fonctionnalités comme l'exécution isolé, ou l'allocation de sections de mémoire privés à certaines applications exécuté dans le userland, le niveau applicatif de privilèges d'exécution. L'utilisation de TEE permet un niveau de sécurité plus élevé pour les applications de confiance s'exécutant sur les systèmes d'exploitation. De nombreux TEE existent, mais le plus connu est sans doute Intel Software Guard Extensions (ou Intel SGX), offrant une isolation précise et efficace de la mémoire et du code exécuté, grâce à un chiffrement de la mémoire.

Applications de Confiance (TA)

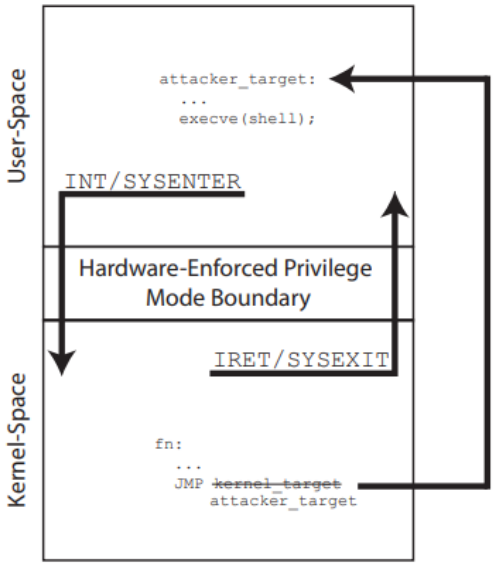
Certaines applications peuvent être décrites comme Application de Confiance, ce qui leur attribue un plus haut niveau de droits, comme l'accès à de emplacements mémoire sécurisés, ou encore l'exécution au niveau kerneland, le plus haut niveau d'exécution d'un programme. Il faut toutefois faire attention à ces applications, leurs privilèges étant très hauts.

4. Prévention

La protection contre les élévations de privilèges s'orchestre globalement comme suivant : deux grands arcs se dessinent, chacun ayant une place dans la défense contre ces attaques. Ces deux arcs sont la prévention est la détection. Ces thèmes ne se réalise pas du tout au même moment d'un processus de protection, mais sont tout de même complémentaires, l'un n'allant pas sans l'autre. Le premier, la prévention, regroupe les techniques et méthodes réalisés « avant » l'attaque. Par exemple, la gouvernance regroupe de nombreux aspects de prévention, en réalisant des procédures à suivre, pour éviter une attaque. Pour des élévations de privilèges, ceci marche globalement de la même manière : on peut contrôler l'efficacité de notre système, construire proprement l'architecture d'un système d'information, d'un système d'exploitation, ou d'une application pour empêcher un probable monté de droits. Scinder correctement notre architecture, protéger et configurer correctement notre noyau, gérer nos mots de passe ou encore mettre régulièrement à jours les programmes disponibles sur les machines font partie de cette prévention. Nous verrons dans cette partie comment protéger efficacement notre noyau, comment protéger le processus de sécurisation des mots de passe, et nous verrons comment les analyses graphiques peuvent nous aider dans le processus de prévention.

4.1. Séparer les privilèges

Containing a Confused Deputy on x86: A Survey of Privilege Escalation Mitigation Techniques [5] parle quant à lui des méthodes utilisées pour contre les attaques sur les systèmes 32 bits. Il développe l'utilisation d'anneaux de protection, un fonctionnement désormais acquis et naturel dans les systèmes d'exploitation, mais qui provient à la base de l'OS Multics, sorti en 1965. Le document réexplique le fonctionnement de ces anneaux, et le passage du *userspace* au *kernelspace*. En effet, un programme, s'exécutera dans le *userspace*, et, en fonction de ses droits, pourra accéder au *kernelspace*, où seuls les programmes autorisés peuvent exécuter. Le schéma ci-dessous présente une attaque possible sur les anneaux de privilèges du noyau.

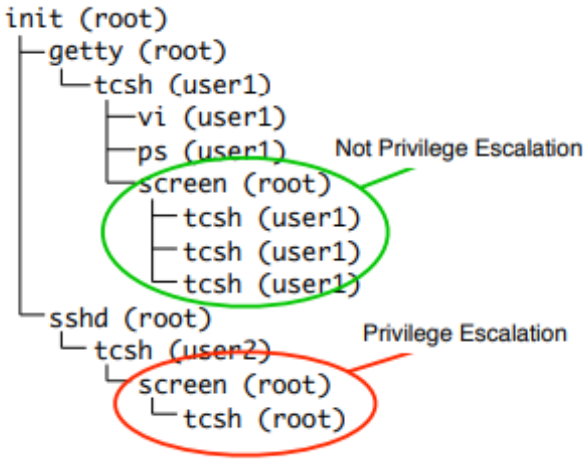


Exemple d'attaque sur un noyau [5]

L'exécution dans cet anneau permet la modification de sections de mémoires sensibles, il est donc essentiel de modérer. L'article analyse dans un second temps différentes méthodes de segmentation des données, et les compare en fonction de leur compatibilité vers les systèmes 64 bits ou encore leur contrôle de l'intégrité de la mémoire. Le document en conclue finalement qu'il n'existe pas de solution idéale, car chaque outil se porte sur un certain aspect de la sécurisation des structures 32 bits.

L'article *12th USENIX Security Symposium* [2] s'occupe dans son cas présenter l'efficacité de la séparation de privilège dans l'utilisation et l'exécution de code, en utilisant une application concrète et en évaluant même les différences de performances. L'objectif de la séparation des privilèges est de réduire la quantité de code qui s'exécute avec un privilège spécial. Pour ce faire, nous divisons une application en plusieurs parties. Une partie s'exécute avec des privilèges et les autres s'exécutent sans privilèges. Nous appelons la partie privilégiée le moniteur et les parties non privilégiées les esclaves. Bien qu'il n'y ait généralement qu'un seul esclave, ce n'est pas une obligation. Un esclave doit demander au moniteur d'effectuer toute opération nécessitant des privilèges. Avant de répondre à une demande de l'esclave, le moniteur la valide d'abord. Si la demande est autorisée, le moniteur l'exécute et communique les résultats à l'esclave. Le document se base sur ce système pour séparer les privilèges d'un bout de code. Plusieurs méthodes sont alors présentées pour contrôler les accès aux privilèges : la séparation sur OpenSSH, les différentes manières de séparer les privilèges d'une demande d'accès d'authentification (sur les phases pré-authentification et post-authentification) ou encore sur la séparation via des environnement Active Directory, et l'utilisation de GPO reliés à des groupes de sécurités spécifiques. L'article termine par réaliser de nombreux tests sur les performances, pour pouvoir en conclure finalement que ces changements n'ont que très peu d'impact sur les performances des machines, parlant d'un ralentissement d'environ 0,002 seconde pour une authentification, et 0,014 pour un transfert de données via OpenSSH. Pour faire simple, l'impact est pratiquement minime.

Enfin, l'article *Detection of Privilege Escalation for Linux Cluster Security* [6] se concentre sur l'explication et l'application de clusters de sécurité pour réaliser une séparation de privilèges, en expliquant le fonctionnement des clusters, des nœuds et le fonctionnement des architectures de droits. Ci-dessous présente une vulnérabilité possible dans l'arborescence de droits dans un environnement Linux.



Vulnérabilité dans l'arborescence [6]

Le clustering est nommé dans l'article comme solution, en permettant une hiérarchie de droits, tout en maintenant une sécurité optimale. Il explique cependant les problèmes encourus par ce système, bien qu'il soit très efficace. Deux outils de monitoring sont présentés : PCP et Clumon. Ces outils sont intéressants mais ne sont pas tournés vers de la sécurité. Le logiciel NVisionCC est présenté comme solution à ces tentatives d'élévations de privilèges, étant tourné vers la sécurité et pas uniquement sur un simple monitoring. Permettant de détecter les élévations de privilèges au sein même des cluster, NVisionCC est nommé comme l'outil idéal lors de la création de cluster.

4.2. Le noyau, élément sensible

Le noyau est un élément central dans un système d'exploitation : il permet aux applications d'accéder au matériel et de communiquer avec lui. Il est utilisé en tout temps lorsqu'un système d'exploitation est en marche. C'est également le noyau qui permet l'utilisation simultanée de plusieurs tâches et le fonctionnement multi-utilisateurs. Chez Linux, le noyau est développé en C et est partiellement libre. Il fonctionne grâce à diverses couches : la couche matérielle, la plus basse, permettant la communication avec le matériel ; la couche mémoire, permettant de communiquer avec la RAM ; la couche de l'ordonnanceur, permettant le multi-tâche ; la couche de gestion des terminaux ; et enfin la couche de gestion de fichier.

Une élévation de privilège passera très souvent par le noyau, car ce dernier, compromis, permet de réattribuer les droits des utilisateurs. Pour un attaquant, l'accès au noyau est l'assurance d'une élévation de privilèges réussie.

Protéger un noyau contre des élévations de privilèges peut se faire de différentes manières. Commençons par l'utilisation d'un AKO. L'article *Additional Kernel Observer to Prevent Privilege Escalation Attacks by Focusing on System Call Privilege Changes* [4] nous présente un système de surveillance de noyau : les AKO, des observateurs de noyaux. Le but de cet outil est d'empêcher les attaques d'escalade de privilèges qui exploitent les vulnérabilités du système d'exploitation, tout en essayant au maximum de prévenir des attaques Zero-day. Son fonctionnement se base sur l'analyse de modification de droit, comme par exemple le changement d'un uid, fsuid, ou encore un gid, mais également sur l'analyse d'appels de commandes susceptibles d'être utilisés dans ce type d'attaques. De nombreuses Proof of Work sont réalisés en utilisant des CVE connues, et chacun de ces tests a été couronné de succès. L'article évoque également le cas des faux positifs, qui est un élément à ne pas négliger, surtout dans de lourdes structures, où le temps est compté, et où un grand nombre de faux positifs peut s'avérer désastreux. Cet AKO ne produit, après des tests, que très peu de faux positifs, ce qui est un très bon point.

L'article *PrivGuard: Protecting Sensitive Kernel Data From Privilege Escalation Attacks* [8] porte son discours sur la présentation de l'outil PrivGuard, outil permettant les appels systèmes, et les analyser pour pouvoir définir s'ils sont légitimes de passer dans le *kernelspace* ou s'ils sont des tentatives d'élévation de privilège, cela grâce à la prévention d'épuisement de pile, de déplacement de pointeur dans des sections non autorisées, ou encore la protection des données en empêchant leur duplication. L'outil pourrait déterminer si un programme augmente ses privilèges de manière légitime ou non, pour empêcher les élévations de privilèges, tout en ne dégradant pas la qualité du travail des programmes. L'article démontre de manière technique une possible et attaque, et où et comment fonctionnerait PrivGuard. L'article termine par implémenter le logiciel sur un environnement Ubuntu 14.04, avec une version du noyau 3.13.0, et analyse les résultats, concluant que l'outil est efficace contre les attaques par élévations de privilèges.

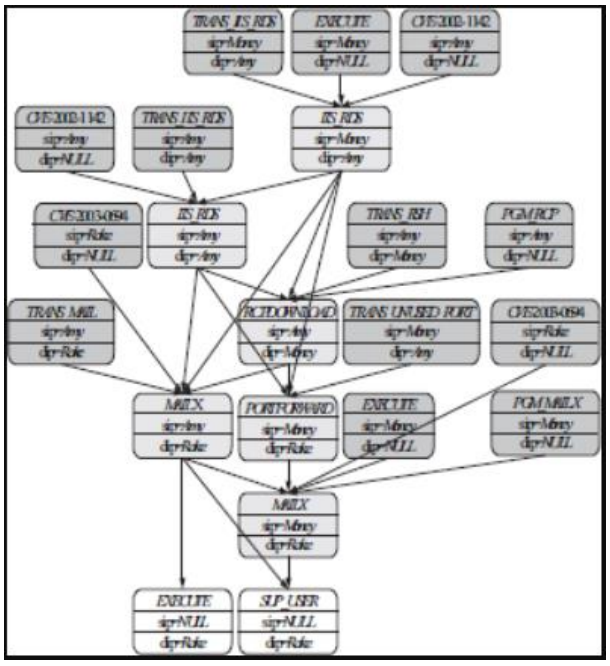
Pour finir, l'article *Security Identifier Randomization: A Method to Prevent Kernel Privilege-Escalation Attacks* [10] se porte sur une manière différente de protéger les attaques contre les noyaux : randomiser les identifiants de sécurité. Cette méthode permettrait de ralentir considérablement la réalisation d'une attaque grâce à un outil nommé SELinux. Son fonctionnement est décrit de cette manière : les privilèges d'un processus ne seraient pas donnés en fonction des identifiants de l'utilisateur, mais du contexte de sécurité. Un administrateur ne pourrait de ce ne fait pas forcément exécuter un programme en tant qu'administrateur, les droits du programme ne serait pas donnés par l'utilisateur, mais pas un programme de sécurisation annexe. L'article explique le fonctionnement technique de cet outil, et conclu par déterminer son impact sur les performances du processeur à moins de 1%, ce qui est très peu en considérant la puissance de l'outil.

4.3. La gestion des mots de passe

Privilege Escalation via Client Management Software – Part II [9], réalisé par la société SySS CmgH, présente et critiques les différentes failles dans la gestion des mots de passe du logiciel Empirum de la société Matrix42. Certaines de ces vulnérabilités sont en effet critiques et permettent une élévation simple des privilèges. Pour prendre l'exemple principal, le hashage des mots de passe s'effectue en MD5 sans sel, ce format de hashage étant très facilement cassable, principalement sans sel, cela constitue une faille majeure. Le programme réalisant ce hashage réalise également un encodage, malheureusement mineur et facilement renversable. SySS GmbH a de ce fait créé un outil nommé Empirium Password Decryptor comme Proof of Concept, démontrant la vulnérabilité de ce système. SySS GmbH notifie également que ces problèmes ont déjà été relaté à la société propriétaire du logiciel, mais qu'aucun retour ni aucune démarche n'a été réalisé pour résoudre le problème. L'article nous démontre que posséder un algorithme de hashage des mots de passe puissant et un encodage efficace protège grandement un système d'information. Nous pouvons plutôt voir ceci : l'absence d'un algorithme de hashage pertinent et un encodage faible permet l'introduction facilement d'attaquants dans le système d'information.

4.3. Analyses graphiques

Une approche assez différente des manières classiques pour prévenir des attaques par élévation de privilège est l'approche par graphique, le but étant de rassembler des différents aspects de la prévention d'attaques dans un graphique permettant de comprendre et de cerner rapidement les points faibles d'une architecture, qu'elle soit hardware ou software. L'article *Study of generating attack graph based on privilege escalation for computer networks* [11] nous présente une méthode par cette approche, en utilisant les prédicats, provenant des mathématiques logiques. Cette utilisation à pour but, en convertissant des vulnérabilités en graphique, de visualiser en graphiques et en tableaux les risques et impacts, mais également de montrer les corrélations entre les différentes vulnérabilités. L'image ci-dessous est un graphique final possible.



Exemple de graph de corrélation d'attaque [11]

L'article commence par nous présenter de manière synthétique l'approche effectuée. La méthode se base sur des triplets mathématiques (fait, prérequis, conséquence) pour représenter les différentes attaques et leur impact et une base de données relationnelle permettant les liens et corrélations entre les différentes attaques. Une base de données d'attaque est implémentée manuellement, se basant sur des attaques précédentes. Dans l'article, l'auteur se base sur des CVE. L'utilisation des prédicats pour modéliser ces attaques permet de lister la portée de l'attaque, sa dangerosité, son impact et ses conséquences, ceci en une simple ligne. Par la suite, une base de données relationnelle est utilisée pour stocker les prédicats dans les tables, qui incluent les attributs associés des attaques et des vulnérabilités, et les relations entre eux. Après de nombreux calculs et de nombreuses corrélations, la méthode est capable de ressortir plusieurs tables (ci-dessous un exemple) permettant de visualiser les différents points d'ancrage possible d'une attaque, dans le but par la suite de résoudre ces différentes vulnérabilités et corriger les potentielles porte d'entrées.

InsVulID	InsVulPredicate	Src_Host	Dst_Host
1	*TRANS_IIS_RDS*	192.168.1.10	192.168.1.19
2	*TRANS_IIS_RDS*	192.168.1.50	192.168.1.19
3	*EXECUTE*	192.168.1.10	*NULL*
4	*SERVICE_FTP*	192.168.1.19	*NULL*

Résultats des vulnérabilités [11]

L'article conclut finalement que cet outil peut être utilisé pour aider l'administration de mesures de sécurité, et aider les équipes de sécurité dans leurs changements, en apportant une approche nouvelle pour la prévention d'attaques.

5. Détection

Le deuxième grand thème est la détection. Contrairement à la prévention, la détection est utile lorsqu'un attaquant possède déjà un accès vers une machine ou un système d'exploitation, et qu'il souhaite élever ses privilèges pour prendre le contrôle. Il existe de nombreuses méthodes de détection, et de nombreux outils différents, mais nous allons nous attarder sur la détection sur des applications de confiance, et sur un outil nommé VPEC, réalisant des analyses graphiques en temps réelles, permettant la détection d'attaques.

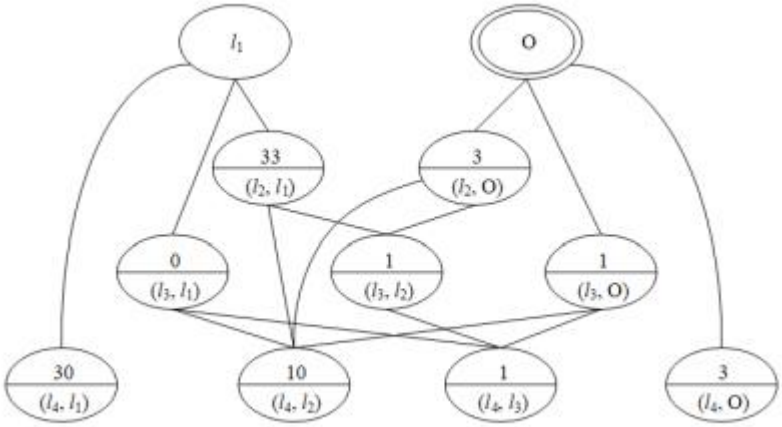
5.1. Détecter une élévation sur les applications

A Qualitative Analysis of Privilege Escalation [3] présente l'un des problèmes majeurs dans l'utilisation d'outils de détection d'élévation de privilèges dans les applications : la gestion des faux positifs. Ces alertes sont souvent dues à des erreurs dans les programmes, certains s'offrant trop de droits, ce qui fait remonter des faux positifs. Ces remontés de faux positifs sont principalement expliqués par le manque de métriques disponibles, ou par une mauvaise gestion de la métrique. Le document utilise ensuite et compare deux métriques connues et disponibles : Shatter et SeDebugPrivilege. Les attaques Shatter sont nommés comme ceci dues à la façon dont elles ressortent. ertaines applications sous Windows s'exécutent à la fois en tant que service et en tant que frontal d'interface utilisateur graphique (GUI) interactive. Et très souvent, ces services s'exécutent sous le privilège LocalSystem. Puisqu'ils sont également autorisés à interagir avec les utilisateurs, un utilisateur malveillant pourrait tirer parti de l'interface graphique, en particulier une zone de texte pour injecter et rediriger le système pour exécuter du code arbitraire sous le LocalSystem privilégié. Avec un shellcode, l'utilisateur malveillant pourrait augmenter son privilège en créant un nouveau shell avec ces privilèges élevés. Le nom Shatter vient de l'expression « Shatter the Windows » Briser Windows, Windows signifiant également fenêtre en anglais. Les attaques par SEDebugPrivilege sont différentes : SEDebugPrivilege permet à un utilisateur d'attacher un processus qui appartient à un autre utilisateur ou même un processus qui s'exécute sous le contexte de sécurité LocalSystem. L'article présente ensuite deux exemples, utilisant ces métriques, qui se révèlent être des faux positifs, ce qui montre la possibilité aux détecteurs de ressortir de nombreux faux positifs, malgré l'utilisation de métriques efficaces. L'article conclu que l'utilisation de métriques efficaces et ne révélant que peu de faux positifs est compliquée, et qu'il faudrait la lier avec des processus d'évaluation des risques, pour ensuite utiliser la meilleure métrique possible en fonction du besoin.

L'article *Horizontal Privilege Escalation in Trusted Applications* [7] développe quant à lui la notion de TA et de TEE. Le document se penche sur les TEE et leurs failles. En effet, certains systèmes d'exploitation (Kinibi, QSEE et Teegris) permettent des élévations de privilèges horizontales car ils utilisent un TEE nommé TrustZone. L'article présente différentes vulnérabilités permettant à des Applications de Confiance de récupérer les droits de la machine sur lequel elle tourne. L'article présente aussi brièvement l'outil HOOPER, permettant de détecter ces attaques. Son fonctionnement se base sur l'analyse de binaires de TA, et analysant les chemins de distribution de droits. Nous pouvons prendre l'exemple de variables étant initialisé à l'intérieur d'un TEE, mais s'exécutant en dehors. Cette vulnérabilité permettrait à un attaquant d'exécuter du code arbitraire, en dehors du TEE, et avec des privilèges élevés. Le logiciel HOOPER détecte ces tentatives d'exfiltration de données et les bloque en temps réel. Pour l'article, l'utilisation du logiciel s'est faite durant 24h, et a révélé et bloqué de nombreuses tentatives. De plus, durant ce laps de temps, aucun faux positif n'a été reporté. L'article conclut en affirmant la qualité de l'outil et son efficacité.

5.2. Solutions de détection basées sur l'analyse graphique

Dans l'article *VCPEC: Vulnerability Correlation Analysis Based on Privilege Escalation and Coritivity Theory | 2020 the 10th International Conference on Communication and Network Security* [12], nous prenons connaissance d'un autre outil d'analyse graphique que précédemment : VCPEC. Cet outil réalise en temps réels des graphiques pour corréler différentes attaques, et finalement offrir des graphiques comme présenté dans le schéma ci-dessous.



Graphique d'analyse VCPEC [12]

L'outil se base en fait sur des bases de données d'attaques, évoluant en temps réel, pour comparer et analyser le réseau en fonction des différents paramètre, portée et impact des attaques. Ces bases de données possèdent un grand nombre de CVE, et ceux-ci translaté en données mathématiques. En effet, l'outil se base sur des calculs mathématiques, en transformant des attaques en notions mathématiques, et réalise des corrélations entre ces attaques et le réseau pour comprendre le fonctionnement des différents systèmes, et empêcher la réalisation d'attaques. Une preuve de concept est réalisée sur un environnement Windows 7, relatant un total de 47 vulnérabilités trouvés. Après l'application de patchs de sécurité, ce nombre descend à 42. L'outil, placé sur des environnements plus larges et plus complexes, pourrait relater des vulnérabilités complexes et poussées.

6. Conclusion

Dans ce document nous avons vu différentes méthodes de prévention, comme l'utilisation de séparation de privilèges, la séparations des droits via des anneaux de privilèges, les vulnérabilités possibles sur les environnement x86, le fonctionnement de la séparation de privilège sous OpenSSH, l'utilisation de clusters de sécurité, la protection en amont du noyau Linux, le conditionnement des kernelspace et userspace, l'utilisation de randomisation en sécurité, la gestion des mots de passe, les outils d'analyses graphiques et enfin les outils de détection d'attaque sur des applications. Nous avons également pu comprendre le fonctionnement et les enjeux des logiciels NVisionCC, VCPEC, AKO, PrivGuard, Empirium et ses vulnérabilités, HOOPER et comprendre les métriques Shatter et SEDebugPrivilege.

Pour conclure, les élévations de privilèges peuvent avoir un impact très grand dans un système d'information, il est donc essentiel de s'en protéger le plus possible. Néanmoins, le sujet étant très vaste et les techniques d'élévations de privilèges étant présentes à chaque niveau d'un système d'information, en passant par l'hardware, les applications et leurs vulnérabilités, ou encore les

vulnérabilités dans le kernel des systèmes d'exploitation, il est indispensable de maintenir au maximum son état de l'art sur le sujet, ce qui était exactement le sujet de ce document.

Réaliser cet état de l'art m'as appris beaucoup de choses, non pas uniquement dans le sujet de la veille, le principal bénéfice à en tirer est l'apprentissage de la méthodologie qui en résulte. En effet, chaque projet commence par une étape de veille et de réalisation d'un état de l'art. Lors des projets, tant professionnels que personnels, que je réaliserais durant les prochaines années, l'application de cette méthode sera indispensable. Ce document m'a permis de me préparer à cette étape d'étude du domaine essentiel dans le travail d'un ingénieur.

7. Références

[1]

S. Keshav, « How to Read Paper by S. Keshav ». http://ccr.sigcomm.org/online/files/p83-keshavA.pdf (consulté le mars 8, 2022).

[2]

« 12th USENIX Security Symposium — Technical Paper ». https://www.usenix.org/legacy/event/sec03/tech/full_papers/provos_et_al/provos_et_al_html/ (consulté le nov. 14, 2021).

[3]

« A Qualitative Analysis of Privilege Escalation ». https://ieee.ezproxy.univ-ubs.fr/abstract/document/4018518/ (consulté le nov. 14, 2021).

[4]

T. Yamauchi, Y. Akao, R. Yoshitani, Y. Nakamura, et M. Hashimoto, « Additional kernel observer: privilege escalation attack prevention mechanism focusing on system call privilege changes », *Int. J. Inf. Secur.*, vol. 20, nº 4, p. 461-473, août 2021, doi: 10.1007/s10207-020-00514-7.

[5]

S. Brookes et S. Taylor, « Containing a Confused Deputy on x86: A Survey of Privilege Escalation Mitigation Techniques », *ijacsa*, vol. 7, nº 4, 2016, doi: 10.14569/IJACSA.2016.070463.

[6]

M. Treaster, G. A. Koenig, X. Meng, et W. Yurcik, « Detection of Privilege Escalation for Linux Cluster Security », 2005.

[7]

D. Suciu, S. McLaughlin, L. Simon, et R. Sion, « Horizontal Privilege Escalation in Trusted Applications », présenté à 29th {USENIX} Security Symposium ({USENIX} Security 20), 2020. Consulté le: nov. 14, 2021. [En ligne]. Disponible sur: https://www.usenix.org/conference/usenixsecurity20/presentation/suciu

[8]

« PrivGuard: Protecting Sensitive Kernel Data From Privilege Escalation Attacks ». https://ieee.ezproxy.univ-ubs.fr/abstract/document/8443329/ (consulté le nov. 14, 2021).

[9]
M. Deeg, « Privilege Escalation via Client Management Software – Part II », p. 4, 2015.

[10]
« Security Identifier Randomization: A Method to Prevent Kernel Privilege-Escalation Attacks ». https://ieee.ezproxy.univ-ubs.fr/abstract/document/7471307/ (consulté le nov. 14, 2021).

[11]
« Study of generating attack graph based on privilege escalation for computer networks ». https://ieee.ezproxy.univ-ubs.fr/abstract/document/4737375/ (consulté le nov. 14, 2021).

[12]
« VCPEC: Vulnerability Correlation Analysis Based on Privilege Escalation and Coritivity Theory | 2020 the 10th International Conference on Communication and Network Security ». https://dl.acm.org/doi/abs/10.1145/3442520.3442526 (consulté le nov. 14, 2021).