Chapitre 5
Le contrôle de comptes utilisateurs

1. Introduction

Le contrôle de comptes utilisateurs UAC (*User Account Control*) est un changement important apparu lors de la sortie de Windows Vista et qui a fait couler beaucoup d'encre depuis les premières versions de test de Windows Vista. Ce chapitre a pour but d'expliquer les problèmes que UAC tente de résoudre, comment il fonctionne en détail, comment il est implémenté dans le système d'exploitation et comment il peut être paramétré. Enfin, la dernière section tente d'apporter une réponse aux questions les plus fréquemment posées.

Le contrôle de compte d'utilisateur (UAC) a été créé pour rendre le système d'exploitation moins vulnérable aux logiciels malveillants en obligeant les utilisateurs, même les administrateurs, à exécuter la plupart des applications avec des privilèges utilisateur standard. Bien entendu, la protection ne sera pas efficace à 100 % et avec le temps, des failles ou des moyens de contournement seront probablement découverts. Avant d'aller plus loin dans la présentation d'UAC, il est important d'alerter le lecteur que UAC n'est pas un périmètre de sécurité en soi, qu'il n'a pas été développé en ce sens, mais qu'il est censé apporter une aide pour diminuer l'exposition du système aux travers des utilisateurs logués et exposés par exemple aux dangers d'Internet. Le dernier paragraphe de ce chapitre introduira les risques et menaces résiduels malgré l'utilisation d'UAC.

Par le passé, les systèmes d'exploitation de Microsoft ont régulièrement été infestés par des logiciels malveillants comme les virus, les vers, chevaux de Troie et autres rootkits. En soi, le système d'exploitation n'était pas moins sécurisé que ses concurrents, mais les auteurs de codes malveillants ont profité d'une part de la large base installée pour prendre le contrôle d'un nombre conséquent de postes, et d'autre part du fait que la plupart des utilisateurs étaient logués en tant qu'administrateur du système. De plus, bon nombre de ces utilisateurs offraient une cible facilement piégeable du fait de leur manque d'expertise en informatique : il était alors aisé de leur faire exécuter une application reçue en pièce jointe d'un message envoyé par un inconnu depuis Internet : cette application exécutée avec toutes les permissions nécessaires, puisque l'utilisateur est administrateur de son poste, peut alors prendre le contrôle de l'ordinateur.

Le contrôle de comptes utilisateurs a donc été développé dans l'optique de permettre aux utilisateurs d'utiliser leurs ordinateurs sans avoir besoin d'être administrateur afin qu'ils soient mieux protégés lors de la navigation sur Internet, la lecture de messages électroniques, la messagerie instantanée, la lecture de forums de discussion, les jeux, et plus généralement toute utilisation pouvant nécessiter l'accès à des ressources disponibles sur Internet. Lors de ces connexions, tout code téléchargé depuis Internet est potentiellement dangereux ; exécuté dans le contexte utilisateur, il peut corrompre ou modifier le paramétrage du système d'exploitation si l'utilisateur dispose de droits d'administration (comme c'est le cas sur les plates-formes Microsoft avant Windows Vista).

La finalité d'UAC, est de permettre d'utiliser le système d'exploitation et les applications installées sans avoir besoin d'être administrateur. Pour atteindre ce but, plusieurs changements importants ont été nécessaires au fil des années, et notamment après Windows XP. Tout d'abord, pour que l'utilisateur puisse se loguer sans être administrateur et cependant administrer correctement son système, il a fallu revoir la plupart des tâches d'administration et les réimplémenter de façon différente afin que l'utilisateur n'ait pas nécessairement besoin d'être administrateur pour les réaliser (par exemple, changer de fuseau horaire pour un utilisateur itinérant, ou encore se connecter à un réseau sans fil, ou installer un pilote d'impression).

Un certain nombre de tâches nécessitant toujours des privilèges d'administration, il a fallu ensuite concevoir un mécanisme d'élévation de privilèges afin que l'utilisateur puisse exécuter une tâche en tant qu'administrateur sans avoir à se loguer avec un autre compte ou utiliser un outil en ligne de commande comme runas.exe.

Le dernier objectif de UAC est d'isoler les applications s'exécutant en tant qu'administrateur des autres applications s'exécutant dans le même bureau afin qu'ils ne puissent pas élever leurs privilèges en prenant le contrôle d'une application s'exécutant dans un contexte administrateur.

Historiquement, l'utilisateur Windows a toujours été administrateur de son poste. C'est dû en partie au fait que les premiers systèmes d'exploitation qui ont démocratisé la plate-forme de Microsoft ne possédaient pas de notion de sécurité (Windows 3.1, Windows 95, Windows 98) et les tâches d'administration devaient être simplifiées et être accessibles à un utilisateur final. Lors de la convergence de la plate-forme Windows 95 et Windows NT Server qui a abouti à Windows 2000 Server et Workstation, un certain nombre de tâches sont restées en l'état, afin d'assurer la compatibilité avec une importante base installée d'ordinateur. Résultat, un grand nombre de tâches qui pouvaient avoir besoin d'accéder en écriture à des parties protégées du système d'exploitation (des clés de registre sous HKEY_LOCAL_MACHINE), écrire dans un fichier de configuration dans le répertoire SYSTEM32 ou sous %PROGRAM FILES% n'ont pas été redéveloppées. Ainsi, un grand nombre de tâches ont nécessité de disposer de permissions réservées aux administrateurs pour s'exécuter et ceci a perduré jusqu'à Windows 2003.

Dans le même temps, de nombreux éditeurs de logiciels ont développé des applications, prenant pour postulat que l'utilisateur était administrateur de sa machine, ceci aussi bien chez Microsoft que chez les autres éditeurs. De même bon nombre de jeux nécessitent d'être administrateur de la machine.

Cet état des lieux est le résultat d'une double responsabilité. Responsabilité des éditeurs de logiciels qui ne concevaient pas leur applications pour s'exécuter dans un contexte de privilèges restreints et responsabilité de Microsoft qui n'a pas poussé en ce sens (ou alors trop tard et trop timidement) et qui n'a pas toujours fourni des interfaces de programmation qui ne nécessitent pas d'être administrateur à l'exécution.

UAC, qui tente d'apporter une solution à ces problèmes, n'est qu'une pièce importante de la stratégie de Microsoft qui consiste à mettre en avant un nouveau modèle dans lequel personne n'est administrateur de la machine, et les tâches réservées aux administrateurs vont pouvoir s'exécuter à l'aide d'un mécanisme d'élévation de privilèges déclenchés à la demande par les applications nécessitant d'être administrateur.

L'utilité et l'efficacité de l'UAC ont été démontrées à maintes reprises sous Windows Vista. Cependant la technologie n'avait pas forcément été bien accueillie par les utilisateurs finaux car les sollicitations engendrées par l'UAC étaient trop nombreuses.

Windows a beaucoup amélioré l'expérience utilisateur liée à l'UAC en réduisant considérablement le nombre de prompts affichés. Certains exécutables et certaines tâches ont également été réécrits afin de ne plus nécessiter des privilèges administrateur. L'UAC s'est également adapté aux nouvelles possibilités d'authentification et la fenêtre d'élévation de privilèges supporte ainsi de s'authentifier via un compte Windows Hello, un code PIN, un certificat ou le mot de passe plus classique.

2. Définitions

2.1 SID

Dans les systèmes Windows, tout compte de sécurité possède un identifiant unique appelé SID pour Secure Identifier. Un compte de sécurité peut être un compte utilisateur, un groupe d'utilisateurs ou un ordinateur, mais il en existe d'autres (comme par exemple une relation d'approbation). Depuis Windows Vista, ça peut également être un service.

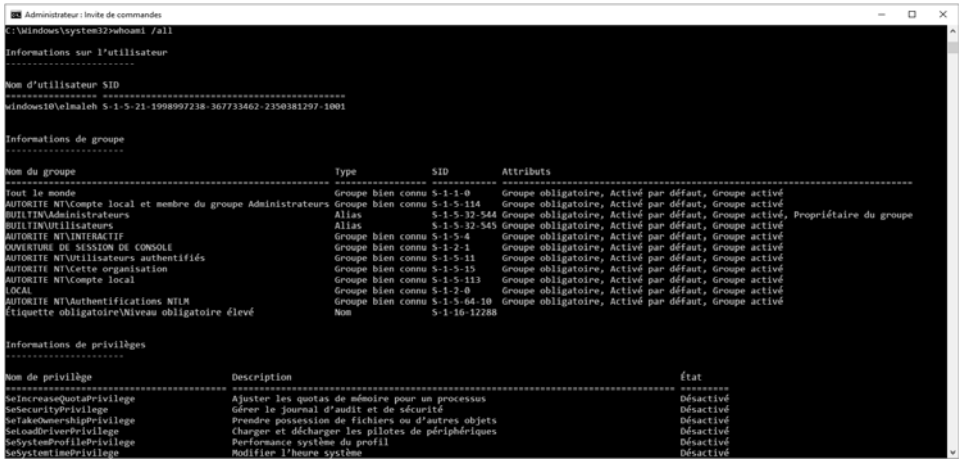
En interne, le système manipule des SID. Les noms d'utilisateurs, de groupes ou d'ordinateurs ne sont que des représentations sous forme de chaînes de caractères de ces comptes référencés par un SID. Cela permet par exemple de renommer un utilisateur ou un groupe tout en conservant les permissions pour cet utilisateur ou groupe puisque cela pointe toujours sur le même compte de sécurité (SID).

Par contre, supprimer un utilisateur puis le recréer immédiatement à l'identique crée deux comptes de sécurité différents : la gestion des SID est assurée par le système d'exploitation qui incrémente par 1 le SID à chaque création de compte et il n'est pas possible de créer un compte en spécifiant un SID donné. Si vous créez un compte et lui assignez des permissions sur une ressource, en interne, c'est le SID qui est référencé dans la liste des permissions. Si vous supprimez alors le compte et en recréez un nouveau avec le même nom, il se voit assigner un SID différent et n'aura donc pas les permissions du compte précédent.

Un SID se présente sous la forme suivante : S-1-5-21-2577476284-2273008180-4043047528-1000. Sans décortiquer bit à bit le SID en question, sachez que les premiers octets du SID identifient le domaine auquel appartient l'utilisateur si celui-ci est membre du domaine, ou l'ordinateur si c'est un utilisateur local. Pour un domaine ou un ordinateur donné, le préfixe sera donc toujours le même. Le dernier nombre (ici 1000) identifie l'utilisateur. Ce nombre est incrémenté de 1 à chaque création de compte par l'ordinateur local si un utilisateur ou un groupe local est créé, ou par un contrôleur de domaine si c'est un compte de sécurité du domaine qui est créé.

Les systèmes d'exploitation Microsoft sont installés avec un certain nombre de comptes génériques, qui ont toujours le même nombre relatif d'un système à l'autre. Par exemple, tout compte générique administrateur sera de la forme S-1-5-domaine-500 (c'est d'ailleurs la raison pour laquelle le renommage du compte Administrateur ne sert pas à grand chose car l'attaquant l'identifiera rapidement grâce à son SID). De même, le groupe générique administrateurs aura toujours le SID S-1-5-32-544 d'un système à l'autre. Microsoft maintient une liste des numéros de SID génériques. Il est utile de l'avoir sous la main lorsqu'il s'agit par exemple d'analyser des journaux d'événements pour déterminer quel compte a réalisé telle action. L'annexe de ce livre fournit un lien sur la fiche technique qui référence les SID génériques.

L'outil système en ligne de commande **whoami** utilisé avec le paramètre **/all** permet de déterminer le SID de l'utilisateur logué ainsi que les noms et SIDS des groupes auxquels il appartient.



L'outil **psgetsid**, disponible en téléchargement sur le site de Microsoft, permet quant à lui d'afficher le SID de l'ordinateur ou de l'utilisateur de son choix.

2.2 Jeton d'accès (Access Token)

Lors du processus de Logon, un jeton d'accès (en anglais *Access Token*) est créé par le sous-système de sécurité LSA (*Local Security Authentication*) appelé LSASS en référence au nom du processus système en charge de réaliser cette opération. Le jeton d'accès contient le SID unique de l'utilisateur, la liste de tous les SID de groupe local et du domaine auquel le compte de sécurité appartient et également la liste des privilèges dont dispose l'utilisateur. Dans Windows 7/10, il contient également le niveau d'intégrité du compte qui sera abordé un petit peu plus loin. La notion de privilèges sera présentée au chapitre suivant, pour l'instant il suffit de savoir qu'un privilège est un droit de réaliser une opération d'administration particulière comme arrêter le système ou changer de fuseau horaire, tâches qui ne sont pas contrôlées par des permissions sur un objet.

Le jeton d'accès de l'utilisateur étant créé par le système au moment du logon de l'utilisateur, le rajout d'un compte dans un groupe n'est pas pris en compte dynamiquement. Celui-ci doit se déloguer et se reloguer pour obtenir un jeton contenant les nouveaux groupes auxquels il appartient.