Fiches pédagogiques sur la sécurité des systèmes d'exploitation

CyberEdu

CyberEdu

La sécurité par l'enseignement supérieur des NTIC

Ce document pédagogique a été rédigé par un consortium regroupant des enseignants-chercheurs et des professionnels du secteur de la cybersécurité.

Version 2.5 — Février 2017

Table des matières

Introduction

Les fiches pédagogiques présentées dans ce guide ont pour objectif de mettre en avant les éléments fondamentaux de la sécurité des systèmes d'exploitation qui peuvent être présentés à des étudiants de l'enseignement supérieur non-spécialistes du domaine. Les fiches apportent à l'enseignant des repères pédagogiques mais ne peuvent constituer à elles seules un support d'apprentissage pour l'enseignant.

Prérequis pour les étudiants

Chacune des huit fiches indique les prérequis nécessaires à sa compréhension. Les prérequis portent sur les connaissances fondamentales en systèmes d'exploitation : permissions sur les fichiers, protection mémoire, principes de sécurisation d'un OS (*operating system*, ou système d'exploitation en français), analyse de Windows et virtualisation.

Prérequis pour les formateurs

Les fiches apportent des repères pédagogiques aux enseignants, en présentant de manière structurée et concise des sujets importants de la sécurité des systèmes d'exploitation. Ces fiches ne constituent pas un cours complet sur la sécurité des systèmes d'exploitation. Il n'est pas demandé à l'enseignant de parfaitement maîtriser le domaine de la sécurité, mais il devra se renseigner sur les sujets présentés pour pleinement exploiter les fiches pédagogiques. Une maîtrise des systèmes d'exploitation étudiés est fortement conseillée.

Utilisation du guide pédagogique

Ce document contient huit fiches pédagogiques à destination des enseignants en systèmes d'exploitation informatiques dans l'enseignement supérieur. Chaque fiche permettra à l'enseignant d'illustrer son cours de systèmes d'exploitation avec des notions de sécurité. Typiquement, l'enseignant consacrera une trentaine de minutes à la sécurité à la fin de chacun de ses chapitres. Les fiches peuvent être présentées en tout ou partie, dans l'ordre approprié à l'enseignement et aux étudiants visés.

Ci-dessous figure un récapitulatif des sujets abordés, ainsi que le temps recommandé pour présenter les sujets et les prérequis correspondants.

Numéro	Sujet	Durée	Prérequis
Fiche 1	Protections mémoire	35 minutes	Notion de pile
Fiche 2	Droits spéciaux sur les fichiers Unix	35 minutes	Arborescence Unix classique
Fiche 3	Sécuriser Unix	40 minutes	Architecture globale d'un OS Unix
Fiche 4	Évolution des solutions de sécurité Windows	30 minutes	
Fiche 5	Sécurité Windows au niveau de l'AD, de la base des registres et de la base SAM	35 minutes	Notion d'annuaire
Fiche 6	Sécurité au niveau de l'architecture des systèmes d'exploitation Windows	20 minutes	Rôle du système de fichiers
Fiche 7	Gestion des comptes et des utilisateurs Windows	30 minutes	
Fiche 8	Virtualisation des OS	40 minutes	Fonctionnement d'un OS

1 Fiche 1 : Protections mémoire

1.1 Thématique

Thématique	Sécurité des systèmes d'exploitation	**Numéro de fiche**	01	**Mise à jour**	06/03/2016

1.2 Thème des cours visés

Le cours visé par cette fiche est un cours de système d'exploitation portant sur la gestion de la mémoire et la gestion des processus.

1.3 Volume horaire

35 minutes.

1.4 Prérequis / corequis

Des connaissances de base sur la structure de données pile est un prérequis.

Un cours sur l'architecture des systèmes d'exploitation est un corequis pour cette fiche.

1.5 Objectifs pédagogiques

L'objectif pédagogique de cette fiche est de sensibiliser les étudiants aux aspects de la sécurité au niveau de la gestion de la mémoire et de processus. Ainsi, nous présentons l'organisation en mémoire d'un processus, et nous décrivons des attaques classiques telles que le débordement de tampon (*buffer overflow*) exploitant des vulnérabilités dans les logiciels, ainsi que les protections mémoire existantes.

1.6 Conseils pratiques

L'enseignant peut mettre en oeuvre un scénario de l'attaque de débordement de tampon en l'introduisant dans un TP portant sur le système d'exploitation Linux.

1.7 Description

1.7.1 Introduction

Le terme de « pile » (*stack* en anglais) définit une structure de données fréquemment utilisée en informatique. Les éléments y sont empilés (opération *push*), puis dépilés (opération *pop*) dans l'ordre inverse. On parle aussi de structure LIFO (*last in, first out*).

Dans le fonctionnement d'un programme, on rencontre essentiellement trois stratégies d'allocation mémoire :

- L'allocation statique : l'espace mémoire nécessaire est spécifié dans le code source avant l'exécution du programme ; il est réservé au moment de la compilation, dans un fichier binaire, et est accessible dès le chargement du programme en mémoire, avant l'exécution ; cela concerne par exemple les variables globales dans les langages de programmation compilés.
- L'allocation dynamique sur la pile *(stack)* : lors de l'exécution d'un programme, il existe une pile contenant les cadres d'appel des fonctions (les contextes des fonctions imbriquées), qui sont propres au langage utilisé. On y trouve généralement l'adresse de retour de la fonction, ainsi que les arguments et les variables locales de la fonction.
- L'allocation dynamique sur le tas *(heap)* : le tas est utilisé pour toutes les autres variables du programme. L'avantage de cette méthode est qu'une variable allouée dans le tas peut survivre à la fin de la fonction l'ayant allouée. En C, la gestion des variables dans le tas se fait avec les fonctions `malloc`, `realloc` et `free` ; en C++, ce sont les opérateurs `new` et `delete`. Lorsque la gestion du tas incombe au développeur, elle peut mener à des fuites de mémoire ou à d'autres vulnérabilités classiques (*use after free* ou *double free*). Certains langages proposent au contraire une gestion automatique du tas, à l'aide d'un *garbage collector* (glaneur de cellules ou ramasse-miettes) : en échange d'une perte de maîtrise sur l'organisation mémoire, le développeur rend certaines classes de vulnérabilités inopérantes. La taille du tas s'adapte tout au long du programme.

1.7.2 Organisation d'un processus en mémoire

Classiquement, un processus (c'est-à-dire un programme en cours d'exécution) évolue dans un espace mémoire découpé en différentes zones. En première approche, on trouve le découpage suivant :
- une zone où sont stockées les instructions du programme en cours : le code, aussi appelé section *text* ;
- une zone où sont stockées les données allouées statiquement que manipule le programme ;
- la pile contenant le contexte des fonctions en cours d'exécution ;
- une zone d'allocation dynamique : le tas.

En réalité, le découpage en sections d'un processus peut être plus sophistiqué dans les OS récents. En particulier, la zone de données allouées statiquement peut être découpée en trois zones :
- la section *data*, qui contient les variables explicitement initialisées ;
- la section *ro-data*, qui contient d'autres variables statiques initialisées, accessibles uniquement en lecture (*ro* signifie *read-only*) ;
- la section *BSS*, utilisée par de nombreuses chaînes de compilation pour allouer des variables statiques non explicitement initialisées, qui seront mises à zéro. Le système effectue alors une mise à zéro de celles-ci. Contrairement aux deux sections précédentes, les variables de cette zone ne sont pas physiquement présentes dans l'exécutable, mais sont justes réservées.

La représentation typique d'un processus en mémoire est alors celle décrite à la figure 1.

1.7.3 Principe du débordement de tampon

L'idée de base d'un **débordement de tampon** *(*buffer overflow) est la suivante : en l'absence de contrôles adéquats de la part du développeur, un attaquant peut soumettre à un programme des données dont la taille excède le tampon qui doit les recevoir et ainsi, réécrire les données qui jouxtent le tampon. Un exemple classique est l'utilisation d'une fonction manipulant les chaînes de caractères sans vérification de la longueur, telle que `strcpy`.

Le débordement de tampon peut se faire dans les différentes zones mémoire, mais la version la plus simple est le *stack-based buffer overflow*, où l'attaquant peut écraser des éléments présents dans la

pile d'appel : d'autres variables locales ou l'adresse de retour de la fonction [1] (voir figure 2). Dans ce second cas, en écrasant l'adresse de retour par un pointeur vers les données qu'il vient d'écrire, l'attaquant peut dérouter le flux d'execution vers l'adresse de son choix.

Bien entendu, l'adresse vers laquelle l'attaquant redirige le programme doit contenir du code exécutable. Ainsi, si l'attaquant injecte un code arbitraire quelque part en mémoire et indique l'adresse de celui-ci comme adresse de retour dans la pile, il peut mener à son exécution.

1.7.4 Mécanismes de défense

Afin d'éviter que les débordements de tampon ne soient exploitables, une première ligne de défense est d'interdire qu'une zone mémoire soit à la fois inscriptible et exécutable : la zone *text* doit alors être en lecture seule, et les zones de données (*data*, *stack*, et *heap*) doivent être non exécutables. Ainsi, un code arbitraire injecté par un attaquant ne sera pas executable.

On appelle cette mesure **W xor X**.

Pour mettre en œuvre le W xor X, il est usuel d'utiliser une protection matérielle qui permet de dissocier les zones de mémoire exécutables contenant des instructions des zones contenant des données (voir figure 3) :

- La fonction est apparue dès 2003 sur les processeurs AMD sous l'appellation commerciale *Enhanced Virus Protection* ou *eXecute Never*, XN.
- En 2004, Intel a introduit, grâce à la pagination sur X86_64 ou X86 PAE (32 bits), le bit XD d'Intel, pour *eXecute Disable*, aussi appelé bit NX, pour *Never eXecute*. Dans le BIOS, ces fonctionnalités sont parfois visibles dans un onglet Sécurité (paramètre NX ou XD).
- Depuis la version Linux 2.6.20 (juin 2005), le *patch* de sécurité PaX [6] permet la mise en œuvre de W xor X ; *Exec Shield* [7] et *OpenWall* [8] font de même.
- W xor X est implémentée partiellement sous MacOS X depuis la version 10.5 Léopard (octobre 2007), et sur iOS 4.3 (mars 2011).
- Cette fonctionnalité, simplement nommée W^X, est native sous OpenBSD.

Le W xor X permet de se prémunir contre l'injection de code extérieure mais pas contre la corruption des données présentes dans la pile. Pour contrer cette menace, on ajoute des **canaris** dans la pile d'appel : à chaque fois qu'une nouvelle fonction débute, une valeur aléatoire est stockée dans la pile entre l'adresse de retour et les variables locales. Avant de retourner à l'appelant, le code vérifiera que la valeur n'a pas été modifiée. Cette mesure est parfois appelée *stack-smashing protection*. Si le canari est de taille suffisamment importante, la probabilité que l'attaquant devine la bonne valeur en aveugle est faible.

Toutefois, si l'ajout de canaris permet de protéger la pile, il n'agit en rien contre les depassements de tampons dans le tas. Pour parer ceux-ci et rendre encore plus complexe l'exploitation de débordements de tampon, on ajoute un autre mécanisme de défense : la **randomisation** de l'espace mémoire, ou ASLR *(Address Space Layout Randomization*). Cette technique consiste à placer les différentes sections d'un processus de manière aléatoire dans l'espace d'adressage du processus. Ce mécanisme rend les adresses des fonctions et de la pile difficiles à prévoir pour l'attaquant.

1.7.5 Limitations et compléments nécessaires

Face à ces contre-mesures, des attaques de plus en plus sophistiquées ont vu le jour.

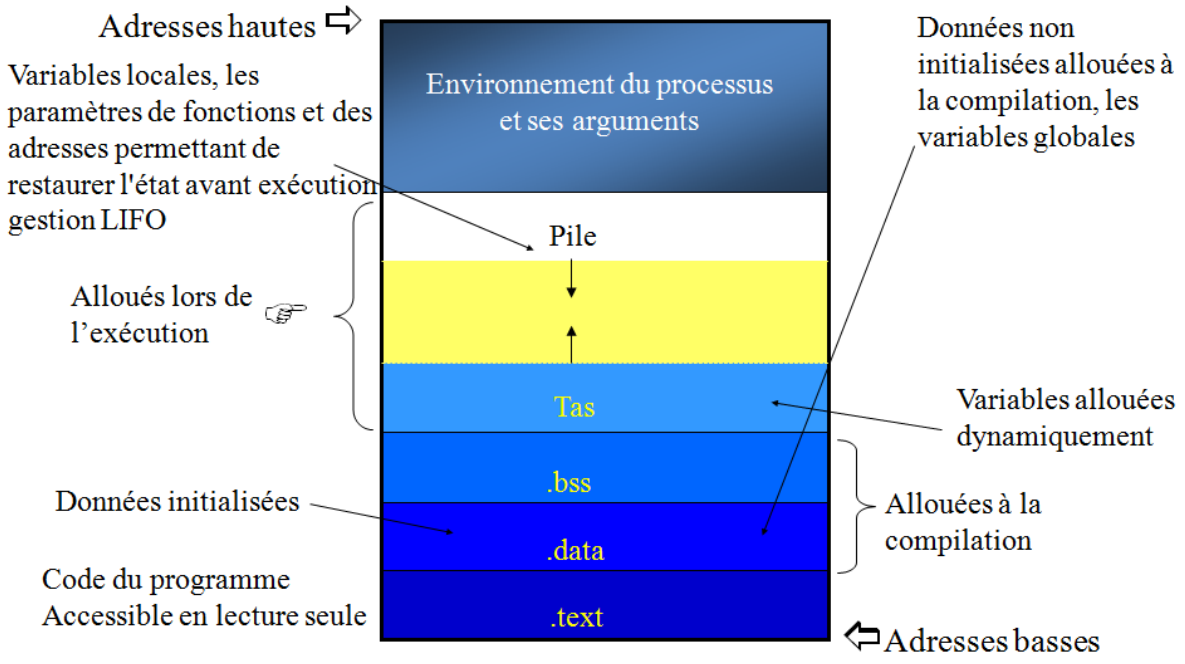

FIGURE 1 – Organisation de la mémoire.

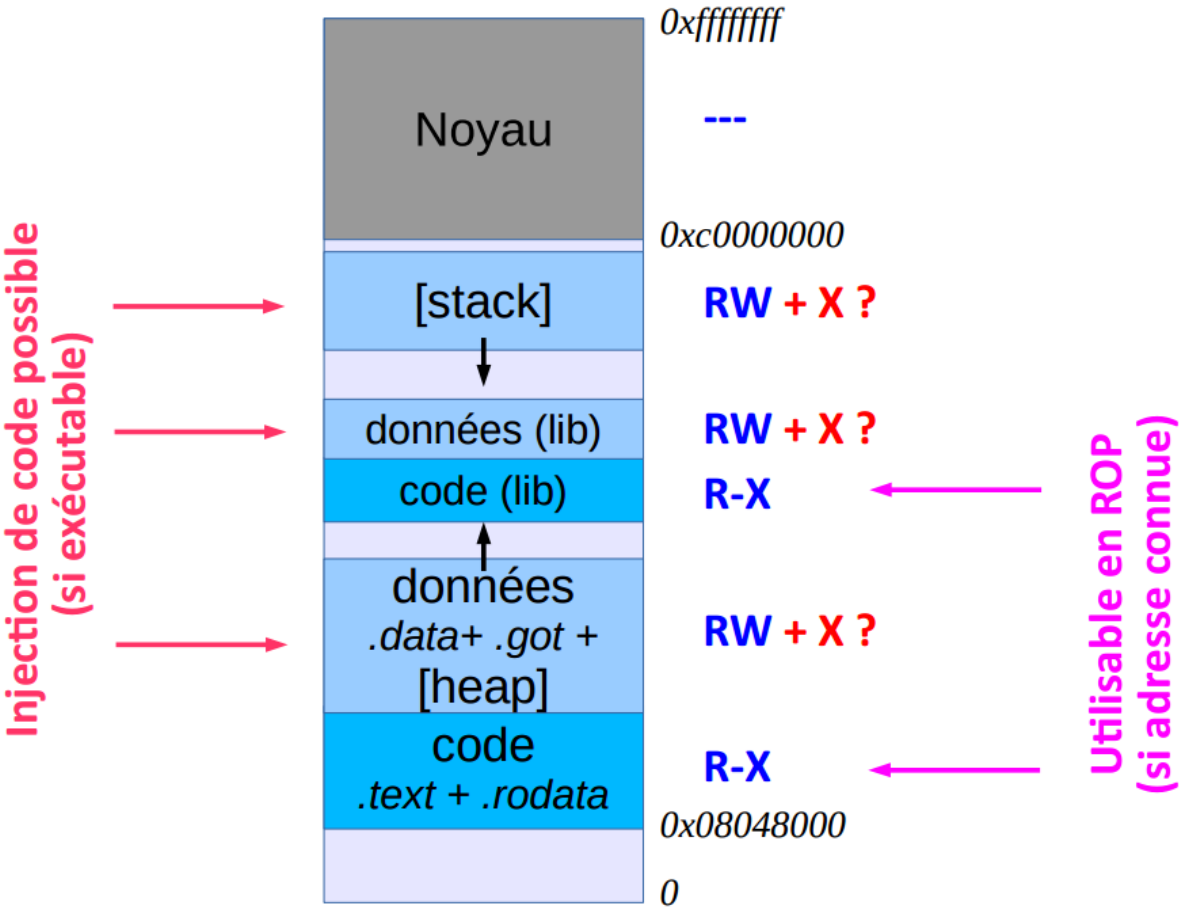

FIGURE 2 – Injections de code possibles dans la pile mémoire : ROP. Source : ANSSI.

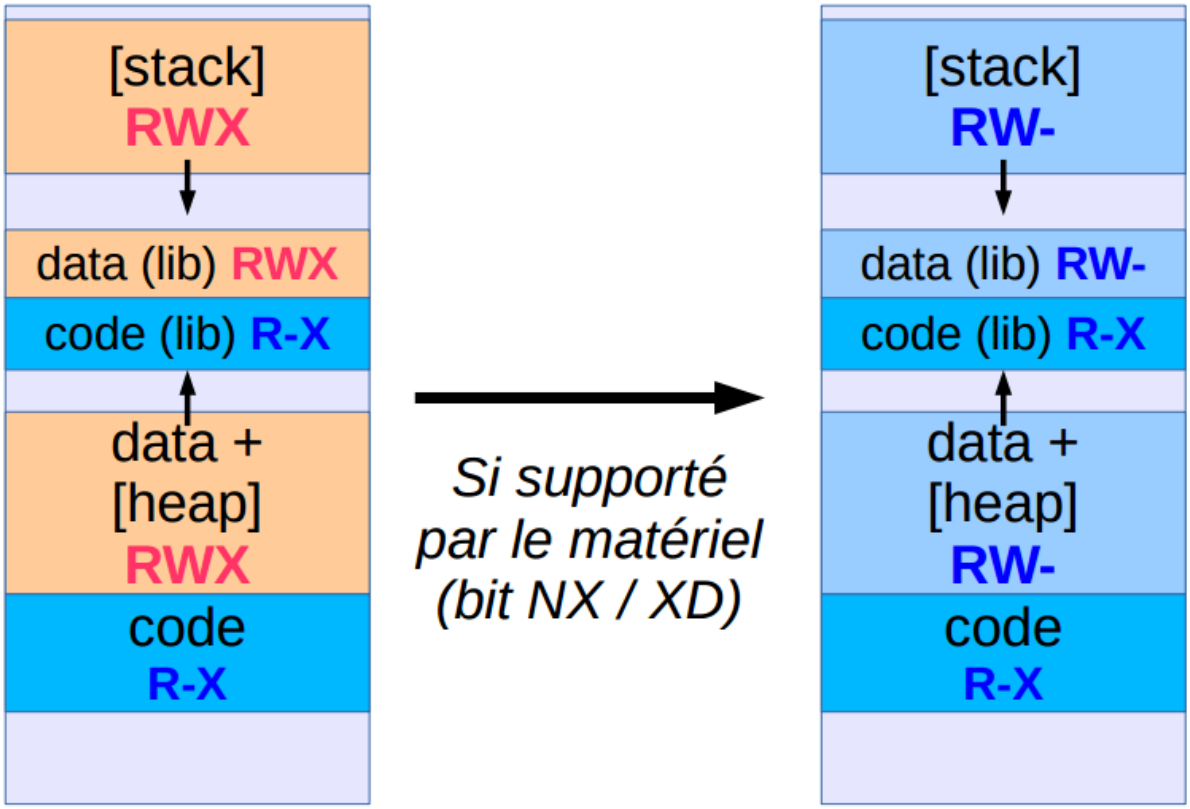

Figure 3 – Durcissement de la mémoire. Source : ANSSI.

Par exemple, en présence d'une pile non exécutable, l'attaquant ne peut plus injecter du code directement comme dans l'exemple ci-dessus (voir 2), mais il peut toujours écraser l'adresse de retour pour la faire pointer vers du code existant dans les bibliothèques ou dans le programme principal. On parle alors d'attaques de type *return-to-libc* ou encore de *return-oriented programming* [2] : l'attaquant va enchaîner des suites d'instructions ainsi identifiées à une adresse connue, nommées « gadgets », pour construire son code malveillant. C'est pourquoi les durcissements proposés dans la section précédente sont complémentaires : la randomisation rend le ROP plus difficile à mettre en œuvre.

Pour aller plus loin, on peut citer quelques éléments complémentaires :
- certains tableaux de pointeurs de fonctions sont présents à des offsets connus : la GOT *(Global Offset Table)*, les `vtables` en C++, ou encore les structures `jmp_buf` ;
- il est possible d'annuler les effets de la randomisation ou des canaris grâce à des attaques en force brute sur les applications multi-processus utilisant `fork` de manière importante (par exemple, certaines versions d'Apache forkent à chaque connexion) : chaque tentative se fait alors sur un processus ayant hérité de la même projection mémoire que son père ;
- de manière similaire au ROP, il existe le *jump-oriented programming* (JOP).

1.8 Matériels didactiques et références bibliographiques

[1] *ANNEXE_01_01 : Animation sur un Buffer Overflow*, (à ouvrir dans LibreOffice)

[2] Erik Buchanan, Ryan Roemer, Stefan Savage, Hovav Shacham, *Return-oriented Programming : Exploitation without Code Injection*, ACM CCS 2008

[3] Mickaël Deloison *Les attaques par corruption de mémoire* http://mdeloison.free.fr/downloads/memattacks.pdf/

[4] Aleph One *Smashing The Stack For Fun And Profit* http://phrack.org/issues/49/14.html

[5] *The Linux kernel patch from the Openwall project* http://www.cgsecurity.org/Articles/1-MISC/Protections-1/, http://www.cgsecurity.org/Articles/1-MISC/Protections-2/,

[6] *Home page of the PAX team* http://pax.grsecurity.net/

[7] nixCraft *Linux Disable or Enable ExecShield Buffer Overflows Protection* http://www.cyberciti.biz/faq/what-is-rhel-centos-fedora-core-execshield/

[8] *Openwall* http://www.openwall.com/

2 Fiche 2 : Droits spéciaux sur les fichiers Unix

2.1 Thématique

Thématique	Sécurité des systèmes d'exploitation Unix	**Numéro de fiche**	02	**Mise à jour**	06/03/2016

2.2 Thème des cours visés

Cette fiche vise des cours qui portent sur le système d'exploitation Unix et plus particulièrement, sur la gestion des utilisateurs, ainsi que la gestion des fichiers et des dossiers Unix. Les cours d'administration du système d'exploitation Unix sont également visés par cette fiche.

2.3 Volume horaire

35 minutes.

2.4 Prérequis / corequis

Une connaissance des commandes Unix standard, ainsi qu'une connaissance basique des différentes arborescences du système Unix sont des prérequis pour cette fiche.

2.5 Objectifs pédagogiques

Il existe des permissions spéciales sur Unix : *Sticky bit*, *SetUID* et *SetGID*. En particulier, les deux dernières consistent en une délégation de privilèges qu'un attaquant pourrait exploiter dans le cas d'une vulnérabilité dans un programme *SetUID* et *SetGID*. Cette fiche a pour objectif pédagogique de sensibiliser les étudiants à leur bon usage.

2.6 Conseils pratiques

L'enseignant peut présenter les aspects de sécurité liés à la gestion des droits lorsqu'il aborde la gestion des utilisateurs, des fichiers et des dossiers dans les arborescences du système Unix. Ainsi, il peut attirer l'attention des étudiants sur la sécurité et intégrer ces aspects principalement lors de l'administration d'un système d'exploitation Unix.

2.7 Description

2.7.1 Introduction

La commande `ls -l` nous permet d'afficher les droits d'un fichier sous Unix, nous pouvons par exemple obtenir ceci : `drwxr-xr-x`. Chaque ligne commence alors par un caractère décrivant le type de fichier, dont la signification est donnée dans le tableau ci-dessous :

Signification	Symbole
Fichier ordinaire *(regular file)*	-
Répertoire	d
Périphérique en mode caractère	c
Périphérique en mode bloc	b
Lien symbolique	l
Tube nommé	p
Socket locale	s

Tout fichier se voit attribuer des droits pour trois identités dans l'ordre suivant :
– `u` *(user)* : les droits du propriétaire ;
– `g` *(group)* : les droits des utilisateurs appartenant au groupe (en dehors du propriétaire lui-même) ;
– `o` *(others)* : les droits des autres utilisateurs.
Pour chaque identité, trois lettres décrivent les permissions associées. Ces permissions ont la signification suivante :
– `r` *(read)* : permission de lire un fichier / de lister un répertoire ;
– `w` *(write)* : permission de modifier un fichier / d'ajouter ou supprimer les fichiers d'un répertoire ;
– `x` *(execute)* : permission d'exécuter un fichier / de traverser un répertoire.
Le tableau suivant présente quelques exemples décrivant ces permissions :

Permissions lettres	Permissions octal	Description
Pour une identité donnée (u, g ou o)		
`rwx`	7	Tous les droits, lecture, écriture et exécution
`r-x`	5	Droits de lecture et d'exécution seulement, pas d'écriture
`r--`	4	Droits de lecture seulement
`---`	0	Ces permissions correspondent donc à aucun droit
Permissions globales		
`rw-r--r--`	644	110 100 100 en binaire : lecture pour tous, écriture uniquement pour le propriétaire
`rwxrwxrwx`	777	Tous les droits pour tous : combinaison à éviter ! On la retrouve cependant sur les liens symboliques, où les droits réellement considérés sont en fait ceux du fichier pointé.

La commande `chmod` (*change mode*), munie des opérateurs +, −, =, permet de définir et de changer les droits d'accès d'un fichier ou d'un ensemble de fichiers. Par exemple, la commande `chmod o-wx fichier` retire (-) aux autres utilisateurs (`o`) les droits d'écriture (`w`) et d'exécution (`x`).

Des permissions spéciales sous Unix permettent de modifier les droits sur les dossiers partagés et sur les exécutables et font l'objet des sections suivantes.

2.7.2 La substitution d'identité

On positionne un droit d'exécution spécial (**s**), qui signifie **substitution d'identité**, et qui remplace le (**x**) dans l'affichage renvoyé par `ls`. Il existe deux cas pour ces droits d'endossement, *SetUID* et *SetGID*. Dans les deux cas, il s'agit d'une délégation de privilège qui doit être maîtrisée.

Droits de propriétaire

Le bit *SetUID*, indique qu'un fichier exécutable, au lieu d'être exécuté avec les droits de l'utilisateur qui le lance, sera exécuté avec les droits du propriétaire du fichier. Il est donc recommandé de limiter autant que possible la présence de programme *SetUID* appartenant à `root`, car ils peuvent présenter des risques d'élévation de privilèges. La commande suivante liste les programmes *SetUID* `root` sur un système : `find / -user root -perm /u+s`.

Par exemple, les programmes suivants sont *SetUID* `root` sur des systèmes classiques :
– `mount`, `umount`, `pmount` et `pumount` pour le montage des systèmes de fichiers ;
– `ping`, `ping6` et `mtr` pour le diagnostic réseau ;
– `chsh` et `chfn` pour les données de comptes.
De même, la commande `passwd` permet à chacun de modifier son mot de passe, c'est-à-dire de modifier le fichier `/etc/shadow`, qui n'est accessible qu'à l'utilisateur `root`. Pour rendre cela possible, les droits du programme `/usr/bin/passwd` et du fichier `/etc/shadow` sont les suivnats :

```
-rw------- root shadow /etc/passwd
-r-s--x-x root root /usr/bin/passwd
```

Tout le monde a le droit d'exécuter la commande `passwd`, mais, lors de cette exécution, le noyau donnera au processus les droits du propriétaire de la commande, `root`, ce qui permettra alors de modifier le fichier `/etc/shadow`.

Sur certains systèmes, il est possible que les exécutables mentionnés plus haut ne soient pas *SetUID* `root`, mais possèdent une portion des droits de `root`. En effet, les distributions Linux récentes implémentent les *file capabilities*, un mécanisme de capacités de fichiers qui offre une alternative à l'utilisation d'exécutables bit *SetUID* `root`. Ce mécanisme fractionne les privilèges traditionnellement associés au super-utilisateur en unités distinctes que l'on peut activer ou inhiber individuellement.

Droits de groupe

Le droit *SetGID* concerne avant tout les programmes. Ceux-ci, au lieu d'être exécutés avec les droits de groupe de l'utilisateur qui les lance, seront exécutés avec les droits de groupe du propriétaire du fichier. Par exemple, `/usr/bin/write` ou `/usr/bin/wall`, qui ont besoin des droits effectifs du groupe `tty` pour accéder aux terminaux, utilisent ce mécanisme.

De plus, dans la convention *System V* d'Unix, un fichier créé dans un *répertoire* ayant le droit *SetGID* aura pour groupe propriétaire le groupe propriétaire du répertoire, et non le groupe de l'utilisateur qui l'a créé.

Notons que si des privilèges doivent être donnés à un utilisateur sur un exécutable spécifique, un contrôle d'accès par groupe sera préféré à un contrôle d'accès par utilisateur. La norme POSIX permet même un contôle plus fin en autorisant à nommer l'utilisateur à qui on veut donner des permissions, via les ACL *(Access Control Lists)* ; il faut pour cela que le système supporte ces ACL.

L'option de montage `suid/nosuid` autorise/interdit l'utilisation des bits *SetUID* et *SetGID* sur la partition considérée : les droits peuvent toujours être positionnés, mais seront sans effet.

2.7.3 Le *sticky bit*

Le *sticky bit* est caractérisé par un caractère (**t**) qui prend la place du caractère (**x**) concernant le groupe *other* dans l'affichage renvoyé par `ls`.

Le *sticky bit* a deux significations :

- Sur un répertoire : il protège les fichiers contenus dans le répertoire de la suppression. Il permet d'autoriser à tout le monde d'ajouter des fichiers dans un répertoire sans qu'ils puissent supprimer des fichiers existants et qui ne leur appartiennent pas. Un exemple classique est le répertoire `/tmp` :
  ```
  $ ls -l /tmp
  drwxrwxrwt 9 root root 240 Feb 17 19:06 /tmp
  ```
  Tout le monde peut ajouter des fichiers dans le répertoire `/tmp` mais chaque fichier de ce répertoire ne peut être supprimé que par son propriétaire. C'est pourquoi une bonne pratique est de limiter les répertoires en écriture pour tous sans *sticky bit*.
- Sur un exécutable : pour les anciennes versions de *Linux*, il indique au noyau que le code de l'exécutable ne doit pas être déchargé de la mémoire *ram* ou *swap*, cela pour permettre une exécution ultérieure plus rapide.

Dans la présentation en nombre octal, ces droits correspondent à un 4e chiffre octal situé à gauche :
- *SetUID* vaut 4 (représentation octale : `4XXX`) ;
- *SetGID* vaut 2 (`2XXX`) ;
- le *sticky bit* vaut 1 (`1XXX`).

Ce qui donne donc les exemples suivants :

Permissions		Description
lettres	octal	
`-rwsr-xr-x`	4755	Ajout d'un bit *SetUID* sur un exécutable
`drwxr-xr-t`	1755	Répertoire doté du *sticky bit*
`-rwsr-sr-t`	6755	On peut évidemment combiner ces droits, ici pour activer conjointement *SetUID SetGID*
`-rwSr--r--`	4644	Par convention, si on applique *SetUID* sur un fichier ne disposant pas de droits d'exécution, la lettre `s` est mise en majuscule. On obtient la même chose pour *SetGID* et le *sticky bit*.

2.8 Matériels didactiques et références bibliographiques

Pour rappel, en plus de l'option de montage `suid/nosuid`, il existe aussi une option de montage, `noexec`, interdisant de prendre en compte les exécutables lors du montage. Par défaut, certains répertoires et points de montage sont accessibles en écriture universelle et disposent de l'option `exec`, ce qui permet à un attaquant de créer un fichier exécutable pour l'aider à maintenir sa présence dans un système compromis. On peut citer en particulier `/dev/shm`, `/tmp`, `/var/tmp`.

Ainsi, il peut être utile d'ajouter l'option `noexec` à ces montages selon les modalités propres à chaque distribution : le rajout d'une ligne correspondant au montage dans `/etc/fstab` suffit généralement. Ces éléments sont décrits plus en détails dans la fiche 3 (Sécuriser Unix).

Au-delà des bits *SetUID* et *SetGID*, il existe d'autres moyens de s'octroyer des autorisations :
- La commande `sudo` *(substitute user do)* permet à certains utilisateurs ou groupes d'utilisateurs autorisés par l'administrateur, de lancer une commande en tant qu'administrateur ou comme autre utilisateur. Pour cela, `sudo` est un programme *SetUID* `root` utilisant un fichier de configuration

pour décrire sa politique. En marche normale, l'utilisation de la commande `sudo` est traçable par l'administrateur système.
– Il existe un outil en ligne de commande similaire intégré à Windows, l'option `runas`, qui permet à un utilisateur d'exécuter des outils spécifiques et des programmes avec des autorisations différentes de celles en cours d'ouverture de session de l'utilisateur [1].

[1] *Technet, Microsoft, Runas*, 2016. `https://technet.microsoft.com/fr-fr/library/cc771525%28v=ws.10%29.aspx`

3 Fiche 3 : Sécuriser Unix

3.1 Thématique

Thématique	Sécurité des systèmes d'exploitation Unix	**Numéro de fiche**	03	**Mise à jour**	06/03/2016

3.2 Thème des cours visés

Cette fiche vise un cours sur le système d'exploitation Unix (même si le cours est niveau débutant).

3.3 Volume horaire

40 minutes.

3.4 Prérequis / corequis

Des connaissances de base sur l'architecture globale d'un système d'exploitation Unix sont des prérequis pour cette fiche. De plus, comme certains exemples concernent les systèmes d'exploitation Linux, une connaissance de ces systèmes en particulier est utile pour exploiter ces exemples.

Un cours d'administration de système d'exploitation Unix est un corequis de cette fiche.

3.5 Objectifs pédagogiques

Il existe de nombreux systèmes Unix. Ils intègrent généralement des mécanismes de sécurité, et chaque système peut être personnalisé afin d'en réduire la surface d'attaque ou d'inclure des *patchs* : le choix de la solution dépendra surtout du contexte d'utilisation. L'objectif pédagogique de cette fiche est de donner les différentes recommandations afin de sécuriser un système d'exploitation Unix et de pouvoir appliquer la bonne stratégie selon le cas étudié.

3.6 Conseils pratiques

Le contenu de cette fiche peut être présenté à travers une étude de cas traitant de stratégies d'administration du système d'exploitation Unix. Dans cette étude de cas, les étudiants seront amenés à mettre en oeuvre les différents principes de sécurité présentés dans cette fiche.

Cette fiche présente des grands principes de sécurisation qui sont décrits plus en détails pour les systèmes d'exploitation Linux dans une note technique de l'ANSSI [1].

Attention : certaines mesures décrites ici nécessitent une expertise technique et leur mise en œuvre pourrait affecter la disponibilité du système.

3.7 Description

Afin de sécuriser un système, il faut privilégier les distributions connues et maintenues, celles qui ont fait leurs preuves et dont la politique des mises à jour de sécurité est réputée sérieuse. En effet, de nombreuses distributions sont aujourd'hui désuètes.

Une attaque réussie sur un OS peut être due à une faille de sécurité dans l'OS ou au niveau applicatif, à une mauvaise implémentation ou à une erreur de configuration.

Un OS sécurisé est un OS dont seules les fonctions nécessaires ont été installées, sur lequel les services applicatifs ont été isolés du reste du système. De plus, il doit être durci, et appliquer le mécanisme de défense en profondeur. Il fait également l'objet de procédures d'administration strictes, surtout en ce qui concerne les mises à jour, et d'une journalisation des événements, ce que nous allons décrire plus loin.

L'objectif de cette sécurisation est d'apporter les garanties suivantes :
- le contrôle d'accès aux données et services selon la politique définie ;
- l'authenticité, l'intégrité et la disponibilité des données ;
- le mode multi-utilisateurs assurera la protection de chaque utilisateur contre les actions potentiellement dangereuses des autres : on parle de confinement ;
- une traçabilité des actions.

Il est important de comprendre qu'une machine doit être installée et configurée au regard de son usage : serveur, poste client, passerelle, firewall, etc.

Pour information, la figure 4 présente la genèse des systèmes compatible Unix, depuis le système Unix développé par les Bell Labs dans les années 1970.

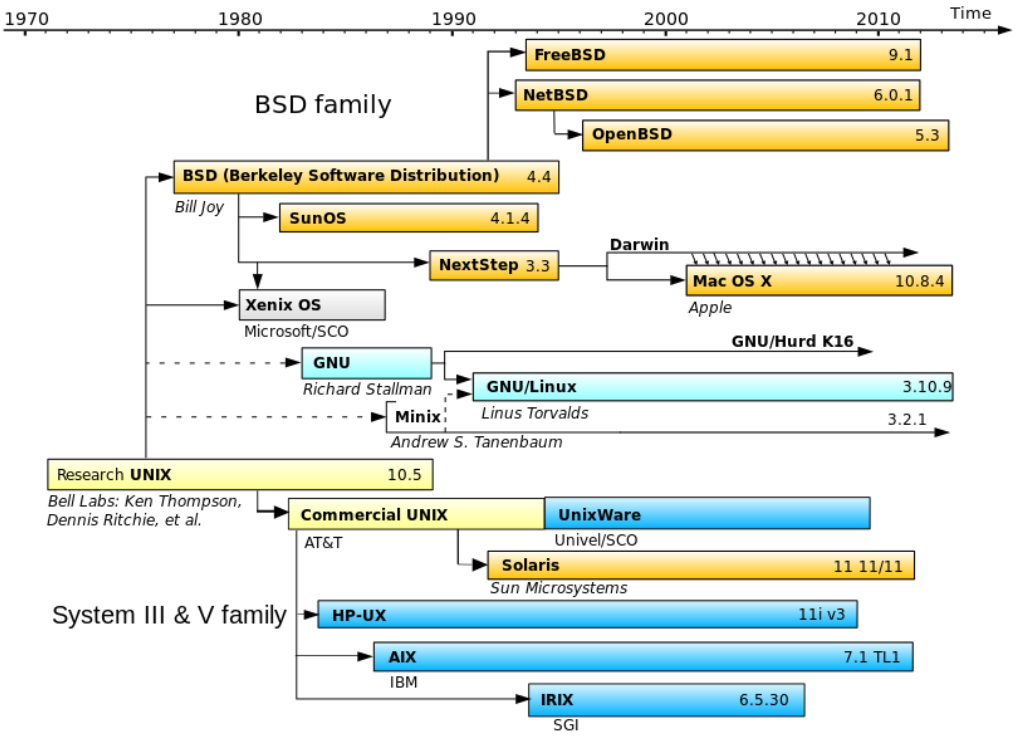

Figure 4 – « Unix genealogy tree » Guillem, Wereon, Hotmocha.

3.7.1 Principe 1 : réduire la surface d'attaque

Pour réduire le périmètre à défendre, il est utile de restreindre les périphériques matériels supportés. Pour cela, on peut désactiver le chargement dynamique de modules et se limiter au matériel présent sur la machine (il faut pour cela écrire 1 dans la variable `/proc/sys/kernel/modules_disabled` sous Linux). De même, il faut limiter les fonctions logiques dont aura besoin le système : un serveur n'aura généralement pas besoin de certaines piles protocolaires comme le Wifi ou Bluetooth, qui peuvent en revanche apporter de nombreuses failles de sécurité (voir, par exemple, CVE-2015-4001, CVE-2015-4002, CVE-2015-4003, CVE-2014-8709, CVE-2012-6544, CVE-2013-0349).

Il est important de limiter les services en écoute pour chaque interface réseau : un outil comme `netstat` inventorie les programmes qui sont en écoute sur une *socket* réseau ou locale ; il affiche aussi des informations sur les tables de routage, sur les interfaces, les connexions masquées, les membres multicast, et les messages `netlink`.

Une bonne pratique est de spécialiser les interfaces réseau d'une machine lorsque c'est possible. Par exemple, pour un serveur web, le serveur HTTP ne sera en écoute que sur l'interface de production alors que le serveur SSH ne sera en écoute que sur l'interface d'administration. La figure 5 montre une bonne configuration de ces deux services, en utilisant netstat, avec une adresse publique pour HTTP et une adresse privée pour SSH.

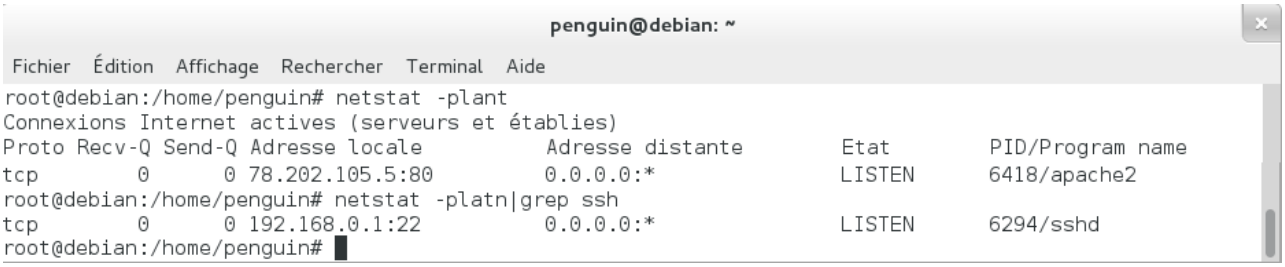

Figure 5 – Configuration des serveurs HTTP et SSH

De même, il faut supprimer ou désactiver ou restreindre les services non nécessaires au fonctionnement du système, surtout s'ils s'exécutent sous le compte `root`. Par exemple, sur un serveur, il est pertinent de retirer le serveur *X*, service multi-fenêtrage particulièrement complexe et privilégié, mais également des démons comme `pulseaudio` ou `dbus` ; en effet, de nombreux démons sont lancés par défaut, et ne sont pas nécessaires au fonctionnement du système.

Une attention devra être portée sur les programmes qui réalisent une analyse syntaxique (*parsing* en anglais) sur des données non maîtrisées correspondants à des formats complexes. Ainsi, un navigateur web, par exemple, ne doit être mis en œuvre que sur un poste client, jamais sur un serveur, cela même sous une identité non privilégiée.

Il faut également faire attention à la présence d'un compilateur ou d'un débogueur, par exemple `gdb`, sur une machine, car ils pourront faciliter le travail de l'attaquant.

Certains démons, tels que `SSH` ou `IKE`, possèdent des options de compilation ou de configuration permettant de révoquer leurs privilèges dès lors qu'ils n'en ont plus besoin ; une autre méthode classique est de séparer le démon en deux, pour déléguer les traitements complexes à un programme esclave non privilégié ;

Les mises à jour d'un système représentent un facteur important pour réduire la surface d'attaques. En effet, ces mises à jour éliminent les vulnérabilités associées aux anciennes versions du système et bloquent les attaques connues reposant sur ces vulnérabilités. Pour cela, il faudra privilégier une

mise à jour authentifiée grâce à une signature cryptographique, en passant par les mécanismes officiels mis en place par l'éditeur. À côté de ces correctifs, il est important de bien préparer les mises à jour majeures de l'OS, car certains durcissements mis en œuvre avec une version donnée peuvent nécessiter une reconfiguration avec la nouvelle version.

De manière générale, il faut appliquer le principe de moindre privilège :
– lancer chaque tâche avec les permissions nécessaires et suffisantes à son accomplissement ;
– limiter le nombre de fichiers systèmes sur lesquels on attribue des droits d'écriture à tous les utilisateurs ;
– limiter le nombre de programmes *SetUID* (voir fiche 2) ;
– positionner les droits en lecture, écriture et exécution sur les fichiers suite à une analyse fine des besoins des utilisateurs et des services locaux.

3.7.2 Principe 2 : cloisonner, isoler les différents services applicatifs

Il est possible d'enfermer les services dans des environnements d'exécution isolés du reste du système, notamment lorsqu'il s'agit d'un serveur : sous Linux, les solutions de type *LXC*, *VServers* ou *OpenVZ* sont plus robustes qu'un `chroot`, qui n'a pas été pensé pour faire de la sécurité, puisque l'isolation ne se cantonne pas au système de fichier. Sous *FreeBSD*, il est possible d'utiliser les *jails*, qui offrent plus de sécurité et de possibilités de configuration que le `chroot` [2].

En complément du mécanisme de contrôle d'accès discrétionnaire classique des systèmes Unix, il est possible d'utiliser des mécanismes de contrôle d'accès obligatoires *(mandatory access control)* qui permettent aussi de limiter les privilèges de programmes : *SELinux*, *AppArmor*, *Tomoyo*, *RBAC GRSecurity* [3]. Si ces mécanismes sont plus complexes à mettre en œuvre, les deux premiers sont souvent disponibles en standard dans les distributions Linux.

Pour les distributions qui le supportent, il est possible d'utiliser les capacités de fichiers, *file capabilities*, qui offrent une autre alternative au bit *SetUID* [4].

3.7.3 Principe 3 : durcir l'OS

Le durcissement des systèmes est le processus qui permet de corriger les faiblesses et les failles de sécurité des systèmes. Ce processus est réalisé en appliquant les derniers correctifs et mises à jour du système. Après avoir supprimé les éléments inutiles, il est utile d'appliquer des correctifs de durcissement du noyau lors de sa recompilation. Il existe ainsi des protections servant à protéger le système contre des attaques de type corruption mémoire (voir fiche 1), comme *PaX* [7] pour Linux. Il existe d'autres implantations similaires de ces mécanismes pour d'autres OS, dont *W^X* pour OpenBSD [5].

Des options de durcissement diverses sont proposées par *GRSecurity* [6] :
– le masquage des entrées de `/proc` ;
– des listes blanches d'utilisateurs autorisés à créer des sockets ;
– un durcissement des `chroot`.
La recompilation systématique des logiciels de certaines distributions, par exemple Gentoo Hardened, permet de maîtriser et de durcir la chaîne de compilation des paquetages. Il faut cependant prendre en compte le coût de maintenance d'un tel système : suivi rigoureux des avis de sécurité, besoin de personnel qualifié.

Voici quelques options de compilation qu'il est ainsi possible d'ajouter à tous les binaires du système :

- `CFLAGS=-fpie` impose que les exécutables soient relocalisables pour que la randomisation des adresses mémoire (ASLR) puisse être active ;
- `CFLAGS=-fstack-protector-all` protège le système contre le *buffer overflow* dans la pile ;
- `CFLAGS=-D_FORTIFY_SOURCE=2` détecte les erreurs liées au *buffer overflow*. La valeur 2 permet d'assurer plus de vérifications. Certaines vérifications se font lors de la phase de compilation et d'autres lors de la phase d'exécution ;
- `LDFLAGS="-Wl, -z, relro -Wl, -z, now"` permet une utilisation sécurisée des bibliothèques dynamiques.

3.7.4 Principe 4 : défendre en profondeur

La défense en profondeur, telle que définie par l'ANSSI, « *consiste à mettre en place plusieurs techniques de sécurité complémentaires afin de réduire l'impact lorsqu'un composant particulier de sécurité est compromis ou défaillant* ».

Ces différentes mesures ont pour but de :
- **Prévenir** : éviter la présence ou l'apparition de failles de sécurité ;
- **Bloquer** : empêcher les attaques de parvenir jusqu'aux composants de sécurité du système ;
- **Contenir** : limiter les conséquences de la compromission d'un composant de sécurité du système ;
- **Détecter** : pouvoir identifier, en vue d'y réagir, les incidents et les compromissions survenant sur le système d'information ;
- **Réparer** : disposer de moyens pour remettre le système en fonctionnement et en conditions de sécurité à la suite d'un incident ou d'une compromission.

Cette section présente quelques exemples illustrant ce principe.

Il faut utiliser un pare-feu local : en complément de la suppression des services inutiles, il permettra aussi de contrôler les connexions initiées depuis le serveur.

Un schéma de partitionnement adapté au contexte d'emploi permet de renforcer la sécurité du système :
- lorsque c'est possible, les partitions doivent être montées en lecture seule ;
- il est d'usage de placer sur une partition distincte tout répertoire susceptible d'être alimenté par des services extérieurs, en particulier, lorsque le service susceptible de venir saturer le répertoire tourne sous l'identité `root`. Dans cette catégorie, on peut citer les partitions suivantes :
  - journaux, *spool* d'impression,
  - files d'attente de messages,
  - base de données associée à un serveur web ;
- l'utilisation des options de montage `noexec`, `nodev` et `nosuid` permettent de limiter les droits des fichiers hébergés sur une partition. Dans le premier cas, les programmes n'y sont plus exécutables ; dans le second cas, les fichiers spéciaux de type *devices* ne sont plus pris en compte ; enfin, `nosuid` indique au noyau que les bit *SetUID* et *SetGID* (décrits dans la fiche 2) ne doivent pas être honorés. Il est classique d'utiliser de telles options pour le montage de supports amovibles.

Sur la plupart des systèmes, il est possible de lancer des tâches périodiques à l'aide de `cron`. Il est important de contrôler les scripts exécutés ainsi, puisqu'il peut s'agir d'un moyen relativement furtif pour un attaquant de maintenir sa présence sur un système. De même, si un tel script est vulnérable, il peut servir de vecteur d'infection.

Une autre bonne pratique de défense en profondeur est de chiffrer les données sur les postes clients, surtout s'ils sont nomades. On utilisera de préférence des moyens qualifiés par l'ANSSI et on veillera à ce que l'administrateur fasse une bonne gestion des clés.

3.7.5 Principe 5 : les procédures d'administration sécurisées

Il faut penser à sécuriser les modes de connexion et d'authentification à la machine, surtout pour les accès d'administration. Cela passe par exemple par l'utilisation de comptes non privilégiés et traçables, ainsi que par l'emploi correct de clés SSH, c'est-à-dire protégées par des *passphrases*. Une politique de sécurité devra également contenir des procédures de mise à jour et une gestion des sauvegardes.

Pour une administration sécurisée, il est également important de ne pas négliger la sécurité physique d'un système d'information. En effet, l'accès physique à une machine permet généralement d'accéder facilement à un compte privilégié, notamment au travers d'un accès à la console locale.

3.7.6 Principe 6 : définir et mettre en place une politique de journalisation d'événements cohérente

On pourra mettre en œuvre une supervision en temps réel, qui distingue différents niveaux de criticité selon leur impact sur la sécurité du système, et permet une remontée d'alertes en fonction de ces niveaux et de la corrélation entre différentes alertes.

Parmi les alertes pertinentes à remonter pour prévenir les dysfonctionnements, on peut citer les tentatives de connexions, l'utilisation de comptes privilégiés, la consommation excessive de ressources ou encore la terminaison inopinée de services critiques.

On pourra pour cela s'aider de tableaux de bord, en prenant garde de se retreindre à des indicateurs pertinents en nombre raisonnable. Dans tous les cas, c'est l'humain qui reste au centre de la décision et qui vérifie la pertinence des éléments ou événements mesurés.

3.8 Matériels didactiques et références bibliographiques

[1] ANSSI, *Note technique - Recommandations de sécurité relatives à un système GNU/Linux*, `http://www.ssi.gouv.fr/administration/guide/recommandations-de-securite-relatives-a-un-systeme-gnulinux/`

[2] *ANNEXE_03_01 : Jails BSD*

[3] *ANNEXE_03_02 : GRSecurity ACL documentation*

[4] *ANNEXE_03_03 : Taking Advantage of Linux Capabilities*

[5] *ANNEXE_03_04 : OpenBSD*

[6] *ANNEXE_03_05 : The case for GRSecurity*

[7] *Home page of the PAX team* `http://pax.grsecurity.net/`

4 Fiche 4 : Évolution des solutions de sécurité Windows

4.1 Thématique

Thématique	Sécurité des systèmes d'exploitation	**Numéro de fiche**	04	**Mise à jour**	05/03/2016

4.2 Thème des cours visés

Le cours visé pour cette fiche est un cours sur les systèmes d'exploitation Windows.

4.3 Volume horaire

30 minutes.

4.4 Prérequis / corequis

Un cours sur les architectures des systèmes d'exploitation est un corequis de cette fiche.

4.5 Objectifs pédagogiques

L'objectif primordial de cette fiche est de présenter aux étudiants, l'évolution des systèmes d'exploitation Windows en termes de sécurité. Installé sur plus d'un milliard d'ordinateurs dans le monde, l'OS Windows est le plus répandu sur les ordinateurs personnels, ce qui explique qu'il soit l'un des plus attaqués. De ce fait, il faut veiller à bien l'installer, à bien le configurer et à bien l'utiliser. En particulier, le nombre d'applications sous Windows est gigantesque, et beaucoup sont anciennes, et pas toujours maintenues. Cette fiche met l'accent sur les solutions de sécurité qu'intègre Windows et qui sont parfois méconnues ou mal employées.

4.6 Conseils pratiques

Le contenu de cette fiche peut être présenté par l'enseignant en introduisant les concepts théoriques des différentes solutions existantes pour les différentes versions non utilisées dans le cours. L'enseignant peut compléter, par la suite, cette partie par une manipulation pratique portant sur les solutions de sécurité qui concernent la version utilisée dans le cours du système d'exploitation présenté.

4.7 Description

Nous présentons ici les principales fonctionnalités de sécurité des OS de la famille Windows, liste non exhaustive présentée chronologiquement.

4.7.1 La famille NT

Cette famille désigne la série des OS multi-tâche préemptifs, multi-utilisateur, multi-processeur, créés par Microsoft et ne reposant pas sur le système historique MS-DOS, contrairement à Windows 1.0, 2, 3.x, 95, 98 et Me. Grâce à ce système, Microsoft et son partenaire Intel ont pu entrer sur le marché des serveurs.

Windows NT 3.5 (21 septembre 1994)
- *login* authentifié des utilisateurs ;
- contrôle d'accès discrétionnaire aux ressources ;
- journal des événements de sécurité ;
- recyclage sécurisé des zones mémoire et des disques réutilisés.

Windows NT 3.51 SP3 (14 décembre 1995)
- distinction entre comptes utilisateurs et comptes administrateurs ;
- *Trusted path* pour le *login*. En utilisant Ctrl-Alt-Suppr, qui provoque une interruption non-masquable, c'est-à-dire toujours reconnue par le microprocesseur dès que le signal électrique a été déclenché. Le système empêche alors qu'une application malveillante se fasse passer pour la fenêtre de *login*, *WinLogon* (voir [10] sur le vol de mot de passe).

4.7.2 1re génération

Windows NT 4.0 SP6a
- Microsoft DAC *(Discretionary Access Control)* : un nouveau modèle de contrôle d'accès discrétionnaire qui permet d'organiser, de gérer, de distribuer et de sécuriser les dossiers et fichiers au sein d'une infrastructure ;
- *LAN Manager* emploie **LM-Hash** : un algorithme de chiffrement pour transformer les mots de passe d'accès aux ressources partagées. Le système étant particulièrement fragile, il faut le désactiver lorsque c'est possible. En effet, le mot de passe LM-Hash a deux contraintes :
  - sa longueur est limitée à 14 caractères maximum ;
  - il ne peut contenir que des caractères ASCII imprimables.

Windows 2000 SP4
- EFS *(Encrypting File System)* : une fonctionnalité de Windows qui permet de stocker des informations sur votre disque dur dans un format chiffré ;
- *Kerberos* : un protocole d'authentification réseau reposant sur du chiffrement symétrique et un système de tickets ;
- **NT-Hash** : la nouvelle fonction de hachage par défaut pour l'authentification locale sous Windows. Les différences principales par rapport à LM-Hash sont les suivantes :
  - les mots de passe sont limités à 127 caractères ;
  - les caractères ASCII et des caractères régionaux sont acceptés ;
  - la casse est prise en compte ;
  - l'algorithme repose sur MD4.

4.7.3 2e génération

Windows XP SP3, Windows 2003 R2 SP2 :

- *DEP, Data Execution Prevention* : applique des restrictions sur les droits d'accès aux pages mémoire (les zones de données sont rendues non exécutables à l'aide du bit NX, voir fiche 1) ; de plus, si un programme tente d'exécuter du code de manière incorrecte en mémoire, la prévention de l'exécution des données le ferme ;
- ASLR (*Address Space Layout Randomization*, voir fiche 1) : place de façon aléatoire les zones de données dans la mémoire virtuelle [11] ;
- *Firewall* : le pare-feu personnel de Windows [12, 13] ;
- KPP *(Kernel Patch Protection)*, aussi connu sous le nom de *Patchguard* : une fonctionnalité qui offre une protection contre les éventuelles modifications du noyau Microsoft Windows 64 bits qui pourraient être opérées par le biais des mises à jour malveillantes.

4.7.4 3e génération

Windows Vista SP2, Windows 7 SP1, 2008

- *Windows Defender* : logiciel anti-spyware et anti-adware temps réel qui apparait sous Windows Vista et sera disponible jusqu'à Windows 7. Il est possible de l'installer sur des versions antérieures de Windows, tel que Windows XP et Windows Server 2003 ; il sera remplacé par *Microsoft Security Essentials*.
- Support du bit XD *(eXecute Disable)* : technologie équivalente au *Full NX Support* (*(Never eXecute Support)*, décrit dans la fiche 1 : c'est la continuité de DEP, une technique supportée par certains OS pour dissocier les zones mémoire contenant des instructions, donc exécutables, des zones contenant des données ; le micro-processeur ne peut alors exécuter du code dans ces zones mémoires, sur la pile ou sur le tas ;
- UAC *(User Account Control)* : en temps normal, les administrateurs et les utilisateurs ont des privilèges utilisateur ; lorsqu'une action privilégiée est sollicitée à partir d'un compte utilisateur, une fenêtre s'ouvre et demande un *login* et un mot de passe administrateur ; la fenêtre de confirmation s'affiche en mode exclusif, c'est-à-dire que le reste de l'écran est assombri et devient inutilisable, pour éviter le *spoofing* ;
- *Bitlocker* : solution de protection cryptographique des disques durs, permettant d'assurer la confidentialité et l'intégrité de l'ensemble du système de fichiers, programmes et données, sur les disques durs, ceci même en cas de vol de l'ordinateur ou d'attaque sur le disque dur machine éteinte [1].

Windows 7, Windows Server 2008 et 2008 R2

- MSE *(Microsoft Security Essentials)* : disponible fin septembre 2009, ce logiciel remplace Windows Defender ; il fournit une protection contre les virus, les vers et les chevaux de Troie et devient donc un antivirus gratuit ;
- *Application isolation* : c'est une nouvelle fonctionnalité d'impression de Windows 7 et Windows Server 2008 R2, qui isole les applications des pilotes d'impression, cela afin que celles-ci ne tombent pas en panne, si jamais un pilote d'impression se bloquait [14].
- *Discretionary Access Control List, DACL* : prolongement du DAC, c'est la liste de contrôle d'accès discrétionnaires dans le service Serveur pour NFS. Elle contient une liste d'entrées qui autorisent ou refusent certains droits à des utilisateurs ou à des groupes spécifiques. Une telle entrée est appelée ACE (*Access Control Entries*, entrée de contrôle d'accès, et est constituée d'un SID

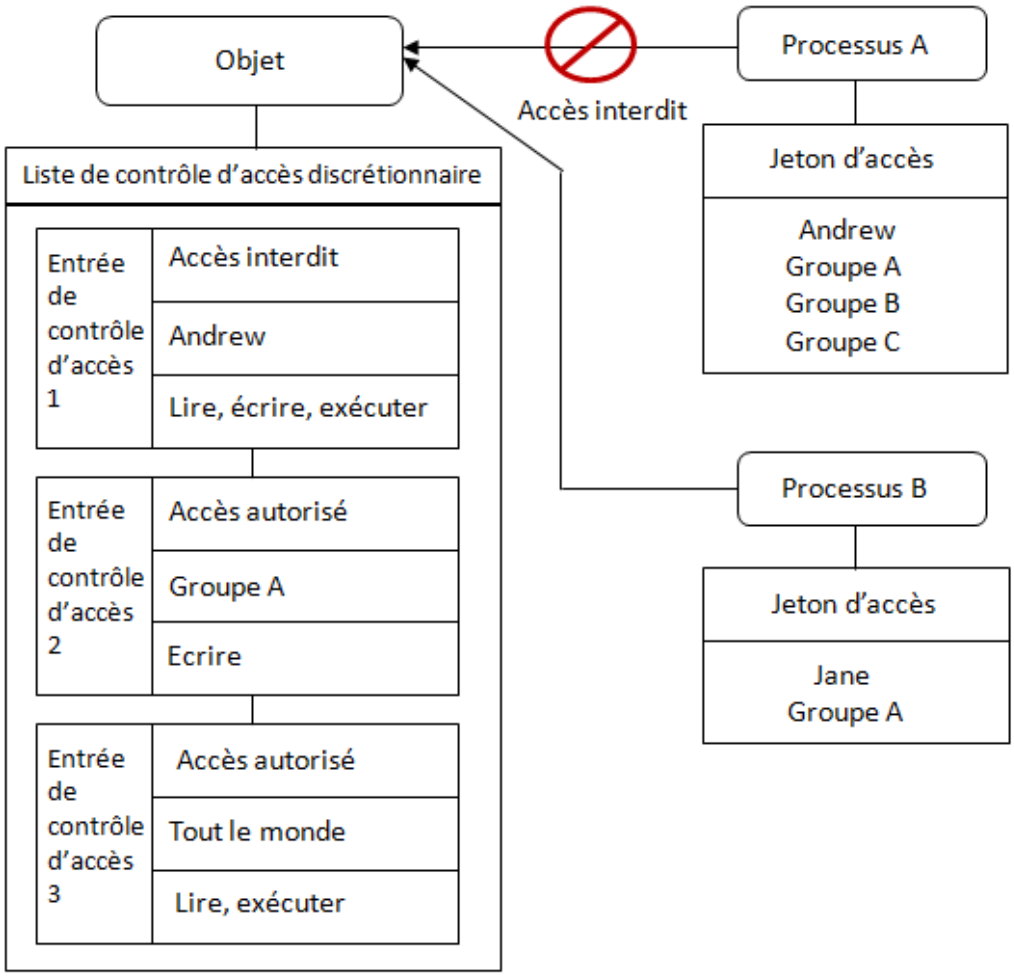

Figure 6 – Schéma : Description d'une DACL *(Discretionary Access Control List)*.

(Security Identifier), servant à identifier un utilisateur ou un groupe particulier, et d'une liste d'accès qui spécifie les autorisations accordées ou refusées à l'utilisateur ou au groupe (voir figure 6) ;

- *Browser protected mode* : mode protégé d'Internet Explorer dont l'icône s'affiche dans la barre d'état ; activé par défaut lorsque l'utilisateur navigue sur Internet, sur un intranet et sur des zones sensibles, il avertit lorsqu'une page web tente d'installer ou d'exécuter des logiciels ou qu'un logiciel s'exécute en dehors d'Internet Explorer et du mode protégé ;
- *AppLocker* : paramètres de la politique de groupe qui permettent de définir quelles sont les applications qui ont le droit de fonctionner sur le réseau de l'entreprise ; fonctionnalité qui permet donc de restreindre l'accès à certaines applications à partir du nom de l'éditeur ou du numéro de version : exécutables, scripts, fichiers Windows Installer, .dll, .exe, .com, .ps1, .bat, .cmd, .vbs, .js, .msi, .msp, .ocx et depuis Windows Server 2012, Windows 8, .mst et .appx. [15].

4.7.5 4e génération

Windows 8, 2012

- *Claims* : c'est une information unique sur un utilisateur, un appareil ou une ressource qui a été publiée par un contrôleur de domaine, ou un annuaire *Active Directory* :
  - *User claims* : les attributs AD sont associés à un utilisateur spécifique ;

  - *Device claims* : les attributs AD sont associés à un ordinateur spécifique ;
  - *Resource attributes* : les propriétés de ressource globales qui sont marquées pour l'utilisation dans les décisions d'autorisation et publiées dans l'AD.
- *Hi-ASLR* : c'est l'ASLR amélioré dans Windows 8. Le préfixe *Hi* fait allusion à l'entropie améliorée, créée par l'augmentation du nombre de bits aléatoires sur la pile et le tas ; Microsoft y a aussi ajouté la randomisation à des tas différents, des tables différentes, etc. [16] ;
- *Kerberos Armoring* : ce mécanisme repose sur le nouveau support KDC *(Key Distribution Center)*, sur les *claims* et sur l'authentification composée. Cadre de politique configuré sur le contrôleur de domaine dans l'unité organisationnelle, *Organizational Unit*, objet conteneur de la norme LDAP.

Windows 8.1, 2012 R2

- Support de l'UEFI *(Unified Extensible Firmware Interface)*, un standard qui définit un logiciel intermédiaire entre le *firmware* et l'OS de l'ordinateur. L'UEFI est le successeur du BIOS et apporte, avec ses nombresues fonctionnalités, son lot de vulnérabilités [2].
- *Protected processes* [3, 4].

Enfin, notons que depuis juillet 2008, *Windows Server Update Services* (WSUS), anciennement SUS, permet de centraliser gratuitement les mises à jour de Microsoft Windows pour un parc informatique d'entreprise. Une solution alternative pour les gros parcs informatiques, beaucoup plus puissante, existe : *Microsoft Systems Management Server*, renommé *System Center* ; toutefois cette dernière est payante.

4.8 Matériels didactiques et références bibliographiques

[1] Aurélien Bordes, *Bitlocker*, SSTIC 2011, `https://www.sstic.org/2011/presentation/bitlocker/`

[2] Pierre Chifflier, *UEFI et bootkits PCI : le danger vient d'en bas*, SSTIC 2013, `https://www.sstic.org/2013/presentation/uefi_et_bootkits_pci/`

[3] Alex Ionescou, *The Evolution of Protected Processes*, `http://www.alex-ionescu.com/?p=97`, `http://www.alex-ionescu.com/?p=116`

[4] Alex Ionescou, *Breaking Protected Processes*, NSC 2014, `http://www.nosuchcon.org/talks/2014/D3_05_Alex_ionescu_Breaking_protected_processes.pdf`

[5] *Le site des développeurs Microsoft*, `http://msdn.microsoft.com`

[6] *La segmentation de la mémoire d'un programme*, `http://www.bases-hacking.org/segmentation-memoire.html`

[7] Aurélien Bordes, Arnaud Ebalard, Raphaël Rigo, *Sécurité de RDP*, SSTIC 2012, `https://www.sstic.org/media/SSTIC2012/SSTIC-actes/securite_rdp/SSTIC2012-Article-securite_rdp-ebalard_bordes_rigo_2.pdf`

[8] Géraud de Drouas, Pierre Capillon, *Audit des permissions en environnement Activec Directory*, `http://www.ssi.gouv.fr/uploads/IMG/pdf/Audit_des_permissions_en_environnement_Active_Directory_article.pdf`

[9] Mark Russinovich, David A. Solomon, Alex Ionescu, *La bible de l'OS Windows : Sysinternals, Part.1 et Part.2*, Microsoft Press, 2012

[10] *ANNEXE_04_01 : Winlogon*

[11] *ANNEXE_04_02 : Address Space Layout Randomization*

[12] *ANNEXE_04_03 : Windows Firewall with Advanced Security Step-by-Step Guide*

[13] *ANNEXE_04_04 : Step-by-Step Guide to Deploying Windows Firewall and IPsec Policies*

[14] *ANNEXE_04_05 : Comment faire pour désactiver ou activer l'isolation d'application*

[15] *ANNEXE_04_06 : NP Applocker NoteTech-v1*

[16] *ANNEXE_04_07 : Exploit Mitigation Improvement in Windows 8 HI-ASLR*

5 Fiche 5 : Sécurité Windows au niveau de l'annuaire AD *(Active Directory)*, de la base de registre et de la base SAM *(Security Account Manager)*

5.1 Thématique

Thématique	Sécurité du système d'exploitation windows	**Numéro de fiche**	05	**Mise à jour**	01/03/2016

5.2 Thème des cours visés

Les cours visés par cette fiche sont les cours des systèmes d'exploitation Windows ainsi que les cours d'administration des systèmes d'exploitation Windows. Ces cours doivent obligatoirement aborder les notions de l'annuaire *Active Directory*, de la base de registre et de la base de gestion des comptes de sécurité (SAM, pour *Security Account Manager*).

Dans ce qui suit, nous utilisons AD pour désigner l'annuaire *Active Directory* [7] et SAM pour désigner la base de gestion des comptes de sécurité.

5.3 Volume horaire

35 minutes.

5.4 Prérequis / corequis

La connaissance et la compréhension du rôle d'un annuaire pour un système d'exploitation est un prérequis pour ce cours.

Un cours sur les architectures des systèmes d'exploitation est un corequis pour cette fiche.

5.5 Objectifs pédagogiques

Cette fiche a pour objectif de fournir les éléments nécessaires et des recommandations pour la sécurisation de l'annuaire AD, de la base de registre et de la base SAM. Cette fiche a pour objectif de sensibiliser les étudiants au besoin et à la manière d'intégrer la sécurité au niveau d'un système d'exploitation. Il s'agit ici du cas particulier du système d'exploitation Windows et de ses composants critiques vis à vis de la sécurité : l'annuaire, la base de registres et la base de gestion des comptes de sécurité.

5.6 Conseils pratiques

Il n'y a pas de conseil spécifique pour cette fiche.

5.7 Description

5.7.1 Concepts de sécurité pour l'*Active Directory*

La sensibilité des données (mots de passe des utilisateurs et des différents comptes) qu'un annuaire AD [6] contient le rend une cible priviligée des attaquants. De plus, la complexité de l'architecture et du schéma de l'AD rend la tâche des attaquants plus facile en leur permettant de s'infiltrer dans les systèmes d'information par le biais de l'AD en utilisant diverses approches qui ne sont pas forcément détectables par un simple utilisateur. On peut citer, par exemple, des attaques par élévation de privilège [2], telles que *Pass the Hash* [3] ou *Pass The Ticket* [4], des attaques reposant sur l'usurpation d'identité [5] ou encore le vol de contrôleur de domaine (la sécurité du contrôleur de domaine est compromise). De ce fait, il est crucial de bien sécuriser l'AD afin de minimiser ces risques.

L'enseignant doit mettre l'accent sur le fait que la sécurité de l'AD commence dès la phase de la définition de l'AD. Donc l'approche conseillée est de tisser ces aspects de sécurité au fur et à mesure de la présentation de l'AD.

L'architecture physique : lors de la définition de l'AD et de sa topologie, il faut faire attention à la sécurité physique des différents contrôleurs de domaines (DC, pour *Domain Controler*). Il faut s'assurer de la bonne configuration des différents DC pour assurer la sécurité de l'AD. S'il y a un DC dont la configuration (à travers la configuration des antivirus, du pare-feu, etc.) n'est pas vérifiée ou validée, il faudra minimiser au maximum les risques liés à ce DC en le configurant en mode lecture seule (RODC) et en mettant également en place un système de chiffrement des disques [9]. La mise en cache des informations d'identification des utilisateurs au niveau du RODC doit se limiter aux comptes qui n'ont aucun privilège pour limiter les risques d'éventuelles attaques qui cibleraient le RODC. Dans ce cas, il est également recommandé de configurer en écriture un autre DC qui soit sécurisé et aussi proche du RODC.

Il est également recommandé de faire attention aux ports utilisés pour la fonction de réplication. La réplication est la fonction qui permet de maintenir l'intégrité et la cohérence des données stockées dans la base d'annuaire *Active Directory*, et de mettre à jour les modifications sur l'ensemble des contrôleurs du domaine. Ainsi, il est conseillé de fixer les ports de réplication afin de maîtriser au mieux les flux réseau. Ceci simplifiera la mise en œuvre d'un filtrage des flux réseau.

L'environnement logiciel : pour assurer au mieux la sécurité de l'annuaire AD, il faut veiller à :
- l'application des mises à jour de sécurité du système ;
- la bonne configuration du système (activation du pare-feu, bonne stratégie de gestion des mots de passe, etc.) ;
- le contrôle d'accès en réduisant au minimum les privilèges pour les comptes de services qui ne nécessitent pas beaucoup de droits. Les DC ne doivent pas héberger plus de services que nécessaire pour le bon fonctionnement de l'AD ;
- la suppression des applications et des systèmes utilisant des technologies dépassées, vu le risque en termes de sécurité qu'ils présentent ;
- la mise à jour des logiciels antivirus et antimalware.

La journalisation : pour les systèmes ayant un noyau en version 5.x et 6, afin de surveiller le fonctionnement des DC, il faut journaliser un certain nombre d'événements pertinents pour détecter

les tentatives d'attaques, comme l'ajout de droits utilisateur, ou les problèmes liés à des services de sécurité (p. ex. le service pare-feu qui n'a pas pu démarrer). Une liste des événements liés à la sécurité à surveiller dans un environnement AD est donné dans la section 5.8.2. Il est aussi recommandé d'augmenter la taille des journaux d'événements.

Pour les systèmes ayant un noyau en version 5.x, il est recommandé de fixer les valeurs suivantes (tirées de la note [9]) :
– taille maximale du journal de sécurité : 179 200 Ko (175 Mo) ;
– taille maximale du journal des applications : 51 200 Ko (50 Mo) ;
– taille maximale du journal système : 51 200 Ko (50 Mo).
Pour les systèmes ayant une version du noyau supérieure ou égale à 6 :

– taille maximale du journal de sécurité : 1 024 000 Ko (1 Go) ;
– taille maximale du journal des applications : 204 800 Ko (200 Mo) ;
– taille maximale du journal système : 204 800 Ko (200 Mo).

L'outil intégré `winver.exe` [11] permet de distinguer la version du noyau.

Les niveaux fonctionnels des domaines : veillez à ce que le niveau fonctionnel des contrôleurs des domaines soit le plus élevé possible. En effet, les nouvelles versions des systèmes d'exploitation traitent en général les lacunes de sécurité présentes dans les anciennes versions.

Les groupes de sécurité : l'accès aux ressources se fait en se basant sur les droits définis pour les groupes de sécurité (voir section 5.8.1). Des mécanismes de sécurité dans l'AD sont liés à ces groupes de sécurité ayant des privilèges comme `AdminSDHolder` [7] et les groupes restreints. L'objet `AdminSDHolder` sert de paramètre pour un descripteur de sécurité restreinte afin de protéger des groupes et des comptes privilégiés. Il est recommandé de surveiller les ACL (*Access Control List*, listes de contrôle d'accès) positionnées sur cet objet `AdminSDHolder` régulièrement pour détecter tout changement. Aussi, il est nécessaire de restreindre le nombre de comptes ayant des droits d'administration sur le domaine entier. Les stratégies de groupe (GPO, *Group Policy Object*) utilisent les groupes restreints pour contrôler les membres des groupes. À chaque fois qu'une GPO ayant le paramètre groupes restreints configuré est appliquée, la liste des membres des groupes concernés est écrasée, permettant ainsi de garder son intégrité.

5.7.2 Base de registre

La base de registre [8] regroupe toutes les informations de configuration et est utilisée par le système et les applications. Pour accéder au registre, il faut lancer l'exécutable `regedit`. Mais il est recommandé de ne pas le faire pour les non experts. Toute modification non maîtrisée de la base de registre peut entrainer une instabilité du système d'exploitation.

Plusieurs attaques peuvent cibler et changer les valeurs des clés de la base de registre. Ainsi, l'accès au registre doit être sécurisé. Pour ce fait, lors de la présentation de la base de registre, il est recommandé à l'enseignant d'aborder la procédure de sécurisation de la base de registre (au niveau de l'éditeur de registre) :
– accéder à votre éditeur de registre (en utilisant `regedit`) ;
– cliquer sur « Édition » puis, « Autorisations » ;
– dans « Paramètres avancés », attribuer les droits aux utilisateurs dans l'onglet « Autorisations » ;

L'administrateur pourra alors configurer les différents privilèges à attribuer aux utilisateurs pour la manipulation et l'utilisation des différentes clés de la base de registre :
– l'onglet « Audit » permet de mettre une surveillance sur un utilisateur ;
– l'observateur d'événement fournit ensuite la liste des accès. Il se trouve dans « Panneau de configuration/Outils d'administration/Gestion de l'ordinateur/Observateurs d'événements » (Cliquer sur le journal « Sécurité »). Il permet à l'administrateur de vérifier s'il y a eu des accès non autorisés et de s'assurer que les autorisations qui ont été définies sont bien respectées.

Au niveau de la base de registre, il est également recommandé de restreindre l'accès des utilisateurs anonymes. Pour cela, il faut fixer les paramètres `RestrictAnonymous` et `RestrictAnonymousSam` (par exemple par GPO) comme suit :

– `HKLM\SYSTEM\CurrentControlSet\control\Lsa\RestrictAnonymous=1`, ce paramètre contrôle le niveau d'énumération accordé à un utilisateur anonyme. RestrictAnonymous peut être défini sur l'une des valeurs suivantes :
  – 0 : aucune. Repose sur les valeurs par défaut.
  – 1 : ne pas autoriser l'énumération des comptes et des noms des Gestionnaires de comptes de sécurité.
  – 2 : aucun accès sans permissions anonymes explicites (cette valeur n'est apparue qu'à partir de Windows 2000).
– `HKLM\SYSTEM\CurrentControlSet\control\Lsa\RestrictAnonymousSam=1`.

5.7.3 La base SAM

L'enseignant doit, également, insister sur le composant SAM *(Security Account Manager)* au niveau de la base de registre. C'est le composant le plus important lié à la sécurité :
– c'est la base de données des différents comptes locaux ;
– elle contient l'ensemble des mots de passe qui sont hachés ;
– la base SAM est stockée physiquement dans le fichier `%SystemRoot%\system32\Config\SAM` ;
– une sauvegarde de la base SAM se trouve dans `%WINDIR%\Repair` ;
– la base SAM peut aussi être dupliquée dans un fichier Excel protégé par un mot de passe.

Pour sécuriser la base de données SAM, l'enseignant peut présenter les différentes recommandations suivantes :
– il est recommandé d'utiliser l'utilitaire `syskey` [10], pour protéger la base SAM des attaquants locaux, ceux ayant potentiellement accès à la machine. `syskey` permet d'améliorer la sécurité de la clé de chiffrement des mots de passe en stockant la clé de chiffrement de la base de données SAM en dehors de l'ordinateur ;
– l'utilitaire `syskey` peut également être utilisé pour configurer un mot de passe initial qui doit être entré pour déchiffrer la clé système, cela afin que Windows puisse accéder à la base de données SAM ;
– pour des raisons de sécurité, il est conseillé aux utilisateurs anonymes d'interdire l'énumération de comptes et de partages dans la base SAM. Cette fonctionnalité est paramétrée dans la base de registre par les deux valeurs `RestrictAnonymous` et `RestrictAnonymoussam`, dans `HKEY_LOCAL_MACHINE\System\CurrentControlSet\LSA` (LSA signifie ici *Local Subsystem Authority*, et se rapporte au démon `Lsass.exe`)[1].

1. Dans Windows 2003 et XP, ces deux paramètres du registre sont modifiés via la console de management `secpol.msc`.

5.8 Matériels didactiques et références bibliographiques

5.8.1 Les groupes de sécurité de l'AD

Dans AD, des groupes prédéfinis existent. Parmi eux :
- Administrateurs du domaine : les membres de ce groupe ont des droits sur tous les objets du domaine AD. De fait, ils sont administrateurs locaux des machines.
- Administrateurs de l'entreprise : les membres de ce groupe ont des droits sur tous les objets de la forêt AD. Ils sont également administrateurs locaux des machines.
- Administrateurs du schéma : les membres de ce groupe peuvent modifier le schéma AD.
- Propriétaires créateurs de la stratégie de groupe : les membres de ce groupe peuvent ajouter, supprimer ou modifier des GPOs. Ils peuvent donc s'octroyer des droits d'administration sur toutes les machines.
- Accès compatible pré-Windows 2000 : il autorise les membres de ce groupe à lire des propriétés sur des objets AD.
- Générateurs d'approbations de forêt entrante : les membres de ce groupe peuvent créer des relations d'approbation unidirectionnelles entrantes.
- Opérateurs de compte : les membres de ce groupe peuvent créer, supprimer et modifier les comptes utilisateurs et machines (sauf dans l'OU Contrôleurs de domaine). Ils sont donc administrateurs locaux.
- Opérateurs de sauvegarde : les membres de ce groupe peuvent sauvegarder et restaurer des fichiers sur un DC, ainsi qu'ouvrir une session sur ces derniers et les arrêter. Ces privilèges sont assimilés à ceux d'un administrateur local et donc de domaine si la machine visée est un DC.
- Opérateurs d'impression : les membres de ce groupe peuvent gérer les objets de type imprimante dans l'annuaire et ouvrir une session sur les DC, ainsi que les arrêter. Ces comptes sont assimilés à des administrateurs locaux du serveur et donc du domaine si la machine ciblée est un DC.
- Opérateurs de serveur : les opérateurs de serveur peuvent ouvrir une session sur les DC, modifier l'heure du système, gérer les services, sauvegarder et restaurer des fichiers, arrêter la machine. Les membres de ce groupe sont également administrateurs locaux de la machine. Sur un DC, ils possèdent donc les mêmes privilèges que les administrateurs du domaine.

5.8.2 Listes des événements AD à surveiller

Nous donnons, ci-dessous, la liste des événements de l'AD à surveiller. Pour chaque événement, l'ID et la description sont fournis [9]. L'ID prend la forme de deux valeurs entières : (1) la première valeur entière présente l'ID de l'événement Windows en cours, (2) la deuxième valeur entière, présentée entre parenthèses, représente l'identifiant des messages d'événements renvoyés par le système.

- 4610 (514) : Un *package* d'authentification a été chargé par l'autorité de sécurité locale
- 4614 (518) : Un *package* de notification a été chargé par le gestionnaire de comptes de sécurité
- 4618 (522) : Un événement de sécurité surveillé est survenu
- 4649 (552) : Une attaque par rejeu a été détectée
- 4719 (612) : Une stratégie d'audit a été modifiée
- 4765 (669) : Un *SID History* (historique d'identifiants uniques) a été ajouté à un compte
- 4766 (670) : Une tentative d'ajout d'un *SID History* a échoué
- 4794 (698) : Une tentative d'activation du mode de restauration AD a échoué
- 4964 (868) : Un compte membre d'un groupe surveillé s'est authentifié
- 1102 (517) : Le journal d'audit a été effacé
- 4706 (610) : Une relation d'approbation a été créée

– 4713 (617) : La stratégie Kerberos a été modifiée
– 4716 (620) : Une relation d'approbation a été modifiée
– 4724 (628) : Une tentative de réinitialisation de mot de passe d'un compte a échoué
– 4739 (643) : La stratégie de domaine a été modifiée
– 4740 (644) : Un compte d'utilisateur a été verrouillé
– 4768 (672) : Un ticket d'authentification Kerberos (TGT) a été demandé
– 4769 (673) : Un ticket de service Kerberos a été demandé
– 4770 (674) : Un ticket de service Kerberos a été renouvelé
– 4771 (675) : La pré-authentification Kerberos a échoué
– 4772 (676) : Une demande de ticket d'authentification Kerberos a échoué
– 4773 (677) : Une demande de ticket de service Kerberos a échoué
– 4774 (678) : Un compte a été mappé pour l'ouverture de session
– 4775 (679) : Impossible de mapper un compte pour l'ouverture de session
– 4776 (680) : L'ordinateur a tenté de valider les informations d'identification d'un compte
– 4777 (681) : Le contrôleur de domaine n'a pas réussi à valider les informations d'identification d'un compte
– 4780 (684) : Des droits ont été modifiés sur des comptes membres du groupe Administrateurs
– 4865 (769) : Une relation d'approbation de forêt a été ajoutée
– 4867 (771) : Une relation d'approbation de forêt a été modifiée
– 4907 (811) : Des paramètres d'audit ont été modifiés
– 4908 (812) : La liste des groupes spéciaux a été modifiée
– 5030 (934) : Le service Pare-feu n'a pas pu démarrer
– 5038 (942) : L'intégrité d'un fichier n'a pas pu être vérifiée
– 6145 (2049) : Des erreurs sont survenues lors de l'application d'une stratégie de groupe
– 4608 (512) : Le système démarre
– 4609 (513) : Le système s'arrête
– 4616 (520) : L'heure du système a été modifiée
– 4698 (602) : Une tâche planifiée a été créée
– 4702 (602) : Une tâche planifiée a été modifiée
– 4704 (608) : Un droit utilisateur a été ajouté
– 4782 (686) : Un accès à l'empreinte d'un mot de passe a été effectué

5.8.3 Références bibliographiques

[1] *TechNet, Microsoft, Réduire la Surface d'attaque* Active Directory, 2016. `https://technet.microsoft.com/fr-fr/library/dn535495.aspx`

[2] *msdn, Microsoft, Élévation de privilège*, 2016. `https://msdn.microsoft.com/fr-fr/library/aa751843(v=vs.110).aspx`

[3] ANSSI, *Microsoft, Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft Techniques*, 2014. `https://download.microsoft.com/download/7/7/a/77abc5bd-8320-41af-863c-6ecfb10cb4b9/mitigating%20pass-the-hash%20(pth)%20attacks%20and%20other%20credential%20theft%20techniques_english.pdf`

[4] CERT-EU Security White Paper 2014-07, *Windows Pass-the-ticket attack*, 2014. `http://forensicmethods.com/wp-content/uploads/2014/07/PassTheGolden_Ticket_v1_0.pdf`

[5] *Microsoft, Se protéger contre l'usurpation d'identité en ligne*, 2016. http://windows.microsoft.com/fr-CA/windows-vista/Guarding-against-online-identity-theft?9c6890a0

[6] *TechNet, Microsoft, Sécurisation de* Active Directory, 2016. https://technet.microsoft.com/fr-fr/library/cc728372.aspx

[7] *TechNet, Microsoft,* Active Directory - *Concepts*, 2016. https://technet.microsoft.com/fr-fr/library/cc780336.aspx

[8] *Tout sur Windows, Tout sur la base de registre*, 2016. http://www.toutwindows.com/registre.shtml

[9] *ANSSI, Note technique No DAT-NT-17/ANSSI/SDE/NP*, 10 Septembre 2014. http://www.ssi.gouv.fr/uploads/IMG/pdf/NP_ActiveDirectory_NoteTech.pdf

[10] *Microsoft, Comment faire pour utiliser l'utilitaire SysKey pour sécuriser la base de données du gestionnaire des comptes de sécurité de Windows* , 2016. https://support.microsoft.com/fr-fr/kb/310105

[11] *Solvusoft Corporation, Qu'est-ce que Winver.exe et comment le corriger ?* , 2011-2016. http://www.solvusoft.com/fr/files/erreur-suppression-virus/exe/windows/international-geogebra-institute/betriebssystem-microsoft-windows/winver-exe/

6 Fiche 6 : Gestion des comptes et des utilisateurs Windows

6.1 Thématique

Thématique	Sécurité des systèmes d'exploitation Windows	**Numéro de fiche**	7	**Mise à jour**	03/02/2016

6.2 Thème des cours visés

Cette fiche peut être utilisée dans un cours traitant des systèmes d'exploitation Windows ou de l'administration des systèmes d'exploitation Windows.

6.3 Volume horaire

30 minutes.

6.4 Prérequis / corequis

Un cours portant sur l'annuaire *Active Directory* de Windows est un corequis de cette fiche. Il complète les connaissances des étudiants pour bien gérer et administrer un ensemble d'utilisateurs et de comptes en respectant les exigences de sécurité souhaitées.

6.5 Objectifs pédagogiques

L'objectif pédagogique principal de cette fiche est de fournir les éléments de base pour la gestion et l'administration des comptes et des utilisateurs d'un système d'exploitation Windows en assurant un bon niveau de sécurité pour le système. Le perimètre de cette fiche inclut également la gestion des comptes au niveau de l'annuaire *Active Directory*. Un deuxième objectif pédagogique de cette fiche concerne la sensibilisation des étudiants aux protocoles d'authentification Windows en termes de sécurité.

6.6 Conseils pratiques

Le contenu de cette fiche peut être traité à travers des manipulations pratiques qui permettent aux étudiants de gérer, d'une manière sécurisée, l'ensemble des comptes utilisateurs au niveau du système d'exploitation Windows.

6.7 Description

6.7.1 Gestion des comptes dans Windows

Les comptes utilisateurs représentent l'une des cibles des attaquants qui cherchent à récupérer des mots de passe afin de s'octroyer les privilèges associés à ces comptes. Ces comptes peuvent être

les comptes des utilisateurs sur le système d'exploitation local ou les comptes de l'annuaire *Active Directory*. Les mots de passe définis par les utilisateurs peuvent être faibles ou des mots de passe par défaut qui n'auraient jamais été changés. La récupération de ces mots de passe peut également être le résultat d'une attaque, comme l'attaque *Pass-the-hash* [4], qui cible la base SAM (voir fiche 5). Une fois l'attaque réussie, l'attaquant pourra alors s'authentifier à distance en utilisant les mots de passe récupérés.

Pour limiter les risques des attaques sur les comptes utilisateurs, l'enseignant doit insister sur la nécessité de mettre en place une politique de comptes. Cette politique doit s'intéresser aux comptes locaux et elle doit également interdire les connexions réseau du compte d'administration local. Aussi, cette politique doit auditer régulièrement les comptes administrateurs et renforcer l'authentification. La fiche 1 des fiches sur l'authentification donne les procédures de sécurité à respecter pour la définition et l'utilisation robustes des mots de passe.

Windows propose 3 principaux types de compte et permet de configurer chaque compte utilisateur :
- **Administrateur** : il peut accéder aux fonctions système de l'ordinateur, réaliser des opérations privilégiées comme l'installation de logiciels, la création d'autres comptes et la configuration des privilèges des autres utilisateurs ;
- **Utilisateur** : il ne peut pas réaliser d'opérations privilégiées, ni créer d'autres comptes, configurer le système ou installer certains logiciels. Ce type de compte est réservé à ceux qui ont une utilisation quotidienne de l'ordinateur. Ce type de compte protège l'ordinateur en empêchant les utilisateurs d'effectuer des modifications susceptibles d'avoir une incidence pour chacun des utilisateurs de l'ordinateur, comme la suppression de fichiers indispensables au fonctionnement de l'ordinateur ; par défaut, vous êtes avertis lorsque les programmes tentent d'apporter des modifications à votre ordinateur ; sous Windows 7, vous pouvez toutefois définir la fréquence de ces notifications, chaque paramètre ayant un impact sur la sécurité de votre ordinateur :
  - « Toujours m'avertir » : vous serez prévenu avant que les programmes puissent effectuer des modifications sur votre ordinateur ou sur des paramètres Windows nécessitant les autorisations d'un administrateur,
  - « M'avertir uniquement quand des programmes tentent d'apporter des modifications à mon ordinateur » : vous serez prévenu avant que les programmes puissent effectuer des modifications sur votre ordinateur nécessitant les autorisations d'un administrateur. La modification des paramètres Windows n'est pas considérée comme étant un programme, donc lorsque cette modification aura lieu, vous ne serez pas averti. Les notifications ont alors lieu sur le *bureau sécurisé*, ce qui signifie que la boîte de dialogue du contrôle de compte d'utilisateur s'affiche sur un fond assombri, et qu'aucun autre programme ne peut s'exécuter en même temps,
  - « M'avertir uniquement quand des programmes tentent d'apporter des modifications à mon ordinateur (ne pas estomper mon Bureau) » : identique au cas précédent, sauf que vous ne serez pas notifiés sur le *bureau sécurisé*,
  - « Ne jamais m'avertir » : Vous ne serez jamais avertis d'aucune modification qui a eu lieu ;
- **Invité** : compte générique aux droits très restreints avec l'impossibilité d'installer des logiciels ; il peut permettre l'accès à une machine placée dans un lieu public.

Un groupe « utilisateurs avec pouvoir » existe pour certaines versions de Windows (Windows 7, Windows 8 Release Preview, Windows Server 2000, Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Vista). Il a été conçu de manière à accorder aux utilisateurs des autorisations et des droits d'administrateur spécifiques pour réaliser des tâches système courantes, comme créer d'autres comptes non administrateurs ou installer des logiciels.

Un groupe « utilisateurs du bureau à distance » existe également pour ces versions. Il permet

d'octroyer aux utilisateurs et aux groupes l'autorisation de se connecter à distance à un « serveur hôte » de session « bureau à distance ».

Windows propose par défaut que tout nouveau compte soit créé avec des privilèges utilisateur, les utilisateurs sans privilèges administrateur ne pouvant pas réaliser les opérations signalées d'un petit bouclier dans le panneau de configuration, une icône qui indique qu'il s'agit d'une opération requérant les droits administrateurs. Parmi les opérations qui nécessitent des privilèges d'administrateur, on peut citer les opérations suivantes :
– écrire dans les répertoires `Program Files` et `Windows` ;
– installer des applications ou les mettre à jour ;
– créer, supprimer ou modifier un compte utilisateur ;
– configurer le pare-feu Windows ;
– accéder aux répertoires des autres utilisateurs.
Ainsi, il est recommandé de limiter l'usage des comptes administrateurs aux seules tâches qui nécessitent ces privilèges afin de limiter le danger des éventuelles attaques.

6.7.2 Fonctionnalités pour la gestion des privilèges

Le contrôle de compte d'utilisateur : c'est une fonctionnalité qui a remplacé les mécanismes UAP *(User Account Protection)* et LUP *(Least User Privilege)* présents jusqu'à la version Windows Vista. Cette fonctionnalité, appelée aussi UAC *(User Account Control)* [1], vous permet de conserver le contrôle sur l'ordinateur en prévenant l'utilisateur de chaque programme qui tente de faire des modifications nécessitant des privilèges administrateur.

Avec cette fonctionnalité, il est fortement recommandé aux utilisateurs d'utiliser les comptes utilisateurs simples et d'éviter l'utilisation d'un compte administrateur dans toutes leurs activités. Les privilèges nécesaires à l'exécution de certaintes tâches seront demandés à l'utilisateur lorsque c'est nécessaire. Ainsi, ces utilisateurs peuvent se protéger davantage de programmes malveillants qui peuvent s'installer sur l'ordinateur ou en modifier la configuration. Lorsque la fonctionnalité UAC est utilisée, quatre types de messages peuvent être affichés à l'utilisateur :

– un paramètre ou une fonctionnalité faisant partie de Windows a besoin de votre autorisation pour démarrer ;
– un programme qui ne fait pas partie de Windows a besoin de votre autorisation pour démarrer ;
– un programme dont l'éditeur est inconnu nécessite votre autorisation pour démarrer ;
– votre administrateur système a bloqué votre exécution de ce programme.

Applocker : c'est une fonctionnalité qui remplace les stratégies de restriction logicielle (SRP, *Software Restriction Policies*) pour Windows 7 et Windows Server 2008 R2. Cette fonctionnalité permet aux administrateurs de mieux contrôler les droits des utilisateurs pour accéder aux fichiers (notamment les fichiers exécutables, les scripts, les fichiers Windows Installer et les DLL) et les utiliser. Pour chaque utilisateur elle permet d'établir une liste d'applications autorisées ou interdites. Applocker permet de [2] :

– définir des règles en se basant sur les attributs dérivés de la signature numérique du fichier comme, par exemple, le nom et la version du fichier ;
– attribuer une règle à un groupe de sécurité ou à un utilisateur individuel ;
– créer des exceptions aux règles. Par exemple, vous pouvez créer une règle qui autorise l'exécution de tous les processus Windows mis à part `regedit.exe` ;

– utiliser le mode d'audit uniquement pour déployer la stratégie et comprendre son impact avant de l'appliquer ;
– importer et exporter des règles ;
– simplifier la gestion de règles AppLocker à l'aide d'applets de commande PowerShell AppLocker.

La table 1 compare AppLocker et les stratégies de restriction logicielle (SRP) (voir section 6.8).

6.7.3 Gestion des comptes au niveau de l'annuaire *Active Directory*

Les comptes de l'annuaire *Active Directory* peuvent être particulièrement une cible priviligiée pour les attaquants. La compromission de l'un de ces comptes peut parfois mener à une compromission de tout l'annuaire. De même, un compte de service ayant de hauts privilèges peut rendre vulnérable l'ensemble du système d'information. La fiche 5 détaille les différents aspects à prendre en considération pour assurer la sécurité de l'annuaire *Active Directory*. Dans cette section nous présentons les recommandations de sécurité liées à la gestion des comptes de l'*Active Directory* que l'enseignant devrait évoquer dans son cours système d'exploitation Windows ou administration des systèmes d'exploitation Windows [3] :

– mettre en place des mécanismes de restriction d'authentification distante des comptes locaux (en utilisant une GPO ainsi que le pare-feu local et l'UAC) pour filtrer les jetons d'accès privilégiés des comptes administrateurs locaux ;
– limiter la taille du cache de domaine, sur les serveurs à 0 et sur les stations de travail à 1. En effet, les serveurs n'ont pas vocation à être déconnectés du réseau ;
– mettre en place une politique de gestion des objets de l'annuaire afin de prendre en compte :
  – les objets obsolètes (comptes qui ne sont plus utilisés, reliquats d'une migration, etc.) ;
  – les objets dormants (comptes n'ayant pas accès à l'annuaire, devenant actifs pour effectuer une procédure spécifique, comme, par exemple, une restauration) ;
  – les objets inactifs (comptes non utilisés depuis une date précise).
– Concernant les comptes de services, une procédure de modification des mots de passe est à prévoir (depuis 2008 R2, les comptes de services gérés permettent de répondre à cette problématique de manière satisfaisante) ;
– De plus, il convient de ne pas utiliser de mot de passe sans date d'expiration.

6.7.4 L'authentification dans Windows

Pour assurer une bonne sécurité du système d'exploitation Windows, la bonne gestion des comptes utilisateurs doit être accompagnée d'une authentification qui soit la mieux sécurisée. L'authentification Windows est un point stratégique pour la sécurisation du système d'exploitation et en particulier pour la sécurisation de l'*Active Directory*. Dans un environnement Windows deux protocoles sont essentiellement utilisés : NTLM et Kerberos. La version la plus sécurisée de NTLM est NTLMv2. NTLM présente principalement le problème de l'absence d'authentification du serveur par le client [3]. La table 2 donne une comparaison de ces deux protocoles (voir section 6.8).

L'utilisation des protocoles est configurable avec un paramètre de stratégie de groupe nommé "Niveau d'authentification LAN Manager". Vous pouvez configurer ce paramètre de sécurité en ouvrant la stratégie appropriée et en développant l'arborescence de la console comme suit : Configuration de l'ordinateur/Paramètres Windows/Paramètres de sécurité/Stratégies locales/Options de sécurité/Sécurité réseau. Il est recommandé de définir ce paramètre au niveau le plus haut possible. Au minimum, le niveau 3 doit être implémenté dans un premier temps afin de cibler le

Fonctionnalités	SRP	Applocker
Étendue de la règle	Tous les utilisateurs	Utilisateur ou groupe spécifique
Conditions de règle fournies	Règles de hachage du fichier, de chemin d'accès, de certificat, de chemin d'accès du Registre et de zone Internet	Règles de hachage du fichier, de chemin d'accès et d'éditeur
Types de règle fournis	Autoriser et refuser	Autoriser et refuser
Action de règle par défaut	Autoriser ou refuser	Refuser
Mode audit uniquement	Non	Oui
Assistant pour créer plusieurs règles à la fois	Non	Oui
Importation ou exportation de stratégie	Non	Oui
Regroupement de règles	Non	Oui
Prise en charge de PowerShell	Non	Oui
Messages d'erreur personnalisés	Non	Oui

Table 1 – Comparaison entre Applocker et SRP.

niveau 5 dans un second temps. Cette précaution permet de se protéger contre la cryptanalyse des empreintes de mots de passe récupérées lors d'une éventuelle écoute du réseau [3]. La valeur du paramètre du Registre LMCompatibilityLevel [2], qui indique le niveau de sécurité utilisé pour le protocole d'authentification, doit être également fixée à une valeur minimale égale à 3.

6.8 Matériels didactiques et références bibliographiques

La table 1 compare AppLocker et les stratégies de restriction logicielle (SRP).

La table 2 présente une comparaison des deux familles des protocoles NTLM et Kerberos.

[1] *Microsoft, Qu'est-ce que le contrôle de compte d'utilisateur ?*, 2016. `http://windows.microsoft.com/fr-fr/windows/what-is-user-account-control#1TC=windows-7`

[2] *TechNet Microsoft, Windows AppLocker*, 2016. `https://technet.microsoft.com/fr-fr/library/dd759117.aspx`

[3] ANSSI, *Note technique, No DAT-NT-17/ANSSI/SDE/NP*, 10 septembre 2014. `http://www.ssi.gouv.fr/uploads/IMG/pdf/NP_ActiveDirectory_NoteTech.pdf`

2. `HKLM\SYSTEM\CurrentControlSet\Control\Lsa\lmcompatibilitylevel`

	NTLM	Kerberos
Méthode cryptographique	Chiffrement symétrique	Kerberos basique : cryptographie symétrique Kerberos PKINIT : cryptographie symétrique et asymétrique
Tiers de confiance	Controleur de domaine	Kerberos basique : Controleur de domaine (DC) et centre de distribution de clé (KDC) Kerberos PKINIT : DC avec service KDC et Windows Enterprise Certification Authority (CA)
Plateforme Windows	Windows 95, Windows 98, Windows ME, Windows NT 4.0, Win2K, XP, Windows server 2003/R2, Vista, et encore utilisé, dans certains cas, pour des versions plus récentes de Windows	Win2K, XP, Windows 2003/R2, Vista, Windows 7, Windows server 2008, 2008R2, Windows 8, 8.1, Windows server 2012, 2012R2
Caractéristiques	Authentification lente Authentification non mutuelle Pas de délégation de l'authentification Protocole propriétaire	Authentification rapide (SSO) Authentification mutuelle optionnelle Support de la délégation d'authentification Standard ouvert

Table 2 – Comparaison entre NTLM et Kerberos.

[4] ANSSI, *Microsoft, Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft Techniques*, 2014. https://download.microsoft.com/download/7/7/a/77abc5bd-8320-41af-863c-6ecfb10cb4b9/mitigating%20pass-the-hash%20(pth)%20attacks%20and%20other%20credential%20theft%20techniques_english.pdf

7 Fiche 7 : Virtualisation des OS

7.1 Thématique

Thématique	Sécurité des systèmes d'exploitation	**Numéro de fiche**	08	**Mise à jour**	05/03/2016

7.2 Thème des cours visés

Cette fiche vise un cours de systèmes d'exploitation.

7.3 Volume horaire

40 minutes.

7.4 Prérequis / corequis

Comprendre le fonctionnement de base d'un système d'exploitation est un prérequis pour cette fiche.

Un cours sur les architectures des systèmes d'exploitation est un corequis.

7.5 Objectifs pédagogiques

L'objectif pédagogique de cette fiche est de sensibiliser les étudiants aux risques liés à la sécurité pour la virtualisation du système d'exploitation. Virtualiser un système d'exploitation peut s'avérer bien pratique, toutefois ces technologies ont rarement été conçues pour la sécurité, et comportent elles-mêmes des failles. Des règles de base permettent toutefois de limiter les risques. Connaître la technologie employée permettra de se prémunir contre les attaques les plus courantes, d'avoir une meilleure maîtrise de son SI.

7.6 Conseils pratiques

L'enseignant peut proposer aux étudiants un TP de montage d'une machine virtuelle avec Virtual Box et aborder les différentes recommandations évoquées dans cette fiche pendant les différentes étapes de réalisation du TP.

7.7 Description

7.7.1 Introduction

Sur de nombreuses architectures matérielles, il existe au moins 2 niveaux de privilège disponibles pour séparer les droits du noyau (*kernel*) de ceux des programmes utilisateurs (*userland*). Chez Intel

ou ARM ces niveaux sont appelés *rings* et sont au nombre de 4. En pratique, ce sont surtout les *rings* 0 (noyau) et 3 (espace utilisateur) qui sont utilisés.

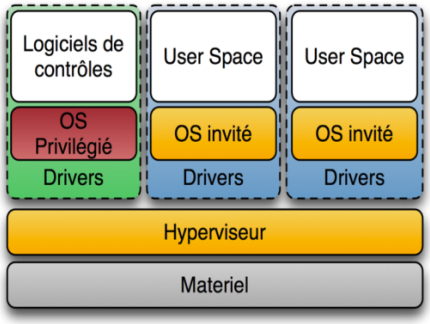

Figure 7 – Architecture reposant sur un hyperviseur. Source : Wikipédia.

La virtualisation permet d'introduire un nouveau niveau de privilège, parfois appelé, par abus de langage, *ring* -1. Ce nouveau privilège permet d'exécuter les systèmes d'exploitation (qui utilisent les niveaux de privilège classiques) de manière transparente, tout en permettant à un nouveau composant, l'hyperviseur, de garantir la séparation des privilèges entre les systèmes d'exploitation virtualisés (invités) d'une part, et entre l'hyperviseur et les systèmes d'exploitation virtualisés (invités) d'autre part (voir figure 7).

Des exemples de telles technologies sont VT-x chez Intel ou AMD-V chez AMD. On appelle alors hyperviseur le logiciel exploitant ce nouveau niveau de privilège pour permettre à plusieurs systèmes d'exploitation de cohabiter sur une même machine physique en même temps.

La virtualisation a une longue histoire, puisque le premier hyperviseur commercialisé date de 1972 : le VM/370 d'IBM. Cette technologie était à l'origine utilisée par des militaires dans un contexte de séparation et de contrôle strict des flux d'information.

En plus de la virtualisation reposant sur les extensions matérielles décrite ci-dessus, on distingue généralement d'autres grandes familles de virtualisation :

Émulation Le logiciel crée un environnement virtuel complet en émulant du matériel virtuel (voir figure 8). Cela permet une bonne isolation des OS invités mais a souvent un coût important en performance, pour la gestion des caches, des TLB, des prédictions de branchements.

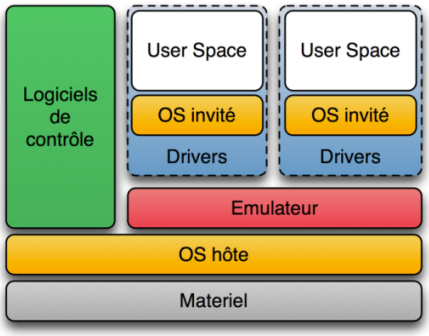

Figure 8 – Architecture utilisant l'émulation. Source : Wikipédia.

Exemples : VirtualBox, VMware Workstation/GSX, QEMU, Microsoft VirtualPC, Plex86, bochs, PearPC (plateforme PPC sur x86), Lismoresystems, Guest PC, MacOnLinux (émulateur de plateforme MacOS sur Linux PPC).

Paravirtualisation Cette solution est assez proche de la solution précédente, mais les systèmes d'exploitation invités sont modifiés et *conscients* d'être virtualisés, ce qui permet certaines optimisations : les systèmes invités peuvent accéder au matériel via des API prédéfinies (les *hypercalls*).

Exemples : XEN (supporte Linux, NetBSD et Plan9), KVM, VMware ESX, VM/CMS d'IBM.

Isolation L'idée ici est d'étendre le concept de `chroot` en isolant des groupes de processus entre eux. Cette technique permet de séparer un système en plusieurs contextes ou environnements. Pour cela, l'espace noyau n'est pas différencié, il est unique, partagé entre les différents contextes. Cela impose que les « systèmes invités » soient de même nature que le système « hôte ». Ainsi, les conteneurs Linux (*Linux Containers*) permettent de faire tourner des *userlands* Linux en parallèle sur un même noyau. La figure 9 représente cette catégorie.

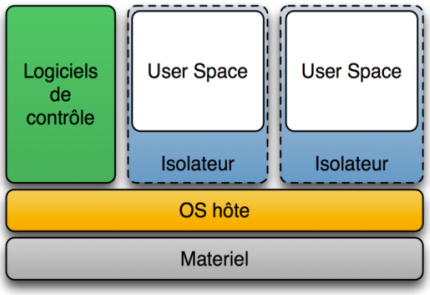

Figure 9 – Architecture reposant sur l'isolation de processus. Source : Wikipédia.

Exemples : Linux-VServer, BSD Jail, OpenVZ, Hyper-V, LXC.

7.7.2 Impact de la virtualisation sur la sécurité

Bien que la virtualisation soit généralement présentée comme une technologie permettant d'augmenter le niveau de sécurité, ce n'est pas dans ce but que les hyperviseurs récents ont été développés. En effet, leur but premier est d'optimiser l'utilisation des ressources physiques, afin de réduire les coûts.

Du point de vue de la sécurité, l'introduction de la couche de virtualisation **accroît la surface d'attaque** des systèmes hébergés. En effet, la sécurité de ces systèmes dépend maintenant également de celle de l'hyperviseur.

Parmi les nouveaux risques apportés par la virtualisation, on peut citer l'apparition de nouveaux canaux cachés liés au partage de certaines ressources (on peut penser à des attaques complexes utilisant le cache CPU par exemple, mais également à des fonctionnalités telles que le partage de fichiers entre machines virtuelles) permettent de faire fuir de l'information entre systèmes invités, ce qui remet en cause l'exigence de confinement.

De même, la disponibilité peut être remise en cause en raison du partage du matériel par les systèmes invités.

Enfin, l'administration système d'un tel système virtualisé est potentiellement plus complexe, ce qui

accroît le risque d'erreur et la difficulté de diagnostiquer un problème. Il est plus facile de comprendre une architecture réseau physique en suivant des fils que l'architecture équivalente virtualisée.

À l'inverse, la virtualisation peut avoir des bénéfices en termes de sécurité. Par exemple, elle peut permettre de décomposer des fonctionnalités qui, sans elle, seraient combinées. On peut ainsi déplacer certains composants, comme un *firewall* ou un anti-virus, dans une machine virtuelle dédiée pour conserver ces fonctionnalités de sécurité, même si un autre système invité est compromis.

La virtualisation peut également accroître la disponibilité en cas de problème matériel, puisque la décorrélation entre les systèmes invités et les socles matériels permet l'utilisation de mécanismes de migration (à froid ou à chaud). C'est d'ailleurs l'un des principes sur lequel repose le *cloud computing* [12].

7.7.3 Quelques recommandations

Ainsi, le lien entre virtualisation et sécurité est complexe. Une note technique de l'ANSSI [1] décrit plus en détails cette problématique, ainsi que des règles de sécurité élémentaires à appliquer.

Il est avant tout essentiel d'appliquer à l'hyperviseur et à l'éventuel système d'exploitation hôte les règles classiques qui s'appliquent également à un système d'exploitation de manière générique : appliquer les mises à jour, réduire la surface d'attaque, protéger les interfaces d'administration système, etc.

En complément, il faut mettre en œuvre des mesures qui prennent en compte les particularités liées à la virtualisation. On peut citer par exemple :
- les systèmes invités présents sur une même machine physique doivent appartenir à une même zone de confiance, un même réseau, et doivent manipuler des données qui ont une sensibilité similaire ;
- le chiffrement des flux permet de se protéger ; si ce n'est pas le cas, une carte réseau physique doit être utilisée par groupe de systèmes invités qui manipulent des données de même sensibilité ;
- les matériels, systèmes et couches de virtualisation sont supervisés, ce qui doit couvrir la journalisation des informations de virtualisation et la synchronisation temporelle des machines hôtes, des systèmes invités et des éléments actifs du réseau afin de pouvoir corréler les journaux ;
- l'administration et la supervision des systèmes hôtes se font sur un réseau dédié : grâce à des cartes réseau et des commutateurs distincts de ceux utilisés par les systèmes invités. Ces tâches sont réalisées par des personnes distinctes des administrateurs des systèmes invités ;
- l'utilisation par chaque machine virtuelle du processeur, de la mémoire et de l'espace disque, doit être limitée, cela afin qu'aucune machine ne puisse monopoliser le système hôte au détriment des autres.

7.7.4 Présentation de quelques vulnérabilités

Citons tout d'abord les erreurs de configuration des couches de virtualisation. Ces dernières peuvent permettre à un système invité d'accéder aux autres systèmes invités, voire au système hôte. Par exemple, la présence d'un partage de fichiers trop permissif peut donner l'accès complet au système hôte depuis le système invité et permettre à un attaquant de prendre le contrôle des machines. Il est donc essentiel que les administrateurs responsables de la couche de virtualisation soient conscients des différentes fonctionnalités proposées.

Comme tout logiciel, l'hyperviseur peut contenir des *bugs*, qui peuvent donner lieu à des vulnérabil-

ités exploitables. On peut citer par exemple l'attaque dite VENOM (CVE-2015-3456), dans laquelle un attaquant exploite une erreur dans le contrôleur de disquette virtuel de QEMU pour causer un déni de service ou pour exécuter du code arbitraire en dehors de la machine virtuelle, au niveau du système d'exploitation hôte. De même, des chercheurs ont mis au jour en 2014 (CVE-2014-8369) une erreur dans le calcul des numéros de pages mémoire dans l'hyperviseur KVM dans Linux ; là encore, la faille peut être exploitée pour causer un déni de service ou exécuter du code arbitraire dans le système hôte

7.8 Matériels didactiques et références bibliographiques

Pour aller plus loin, voici quelques ressources avancées sur les technologies de virtualisation :

- Joanna Rutkowska avait présenté son programme « *Blue Pill* » lors de la conférence *Black Hat* 2006 [2] ; celui-ci met en évidence un défaut de concept de la virtualisation d'AMD sous Windows ; *Blue Pill* qui agit comme un malware, utilise des fonctionnalités matérielles, et non des techniques logicielles. Il exploite une faille dans la conception de l'architecture de virtualisation qui lui permet de devenir hyperviseur, l'OS basculant alors dans une machine virtuelle. Cette technique marche aussi avec les processeurs de chez Intel, elle est *a priori* applicable à tout OS. Indétectable par l'OS, *Blue Pill* l'est donc aussi pour tout antivirus. La désactivation des fonctions de virtualisation est la meilleure manière de se protéger de cette attaque.
- *VT-d / I/OMMU, Input/Output Memory Management Unit* : dispositif qui permet au CPU de configurer le matériel pour qu'un périphérique ait toujours accès à la mémoire que dans la zone qui lui a été allouée. Dans le cas de la paravirtualisation : si le système invité a essayé de donner l'ordre au matériel d'exécuter un accès direct à la mémoire en utilisant des adresses physiques de l'invité, DMA, il pourrait alors corrompre la mémoire. I/OMMU permet d'éviter cela en reconstruisant la carte des adresses auxquelles le matériel a accédé.
- Loïc Duflot a aussi démontré que le matériel permettait de contourner les sécurités logicielles grâce aux contrôleurs d'entrées-sorties (E/S) : USB, FireWire, Ethernet, FPGA. Il est possible de se prémunir contre cette attaque dite DMA, pour *Direct Memory Access*, grâce à des contre mesures matérielles : I/O MMU, *Access Control Services*, ACS. Plusieurs présentations ont été faites au SSTIC sur le sujet [3, 4, 5].
- Une faille dans le *chipset* Q35 d'Intel, premier *chipset* compatible VT-d, soulevé par Joanna Rutkowska [6]. Un moyen de contourner les protections de la solution TXT, *Trusted Execution Technology*, qui permet les diagnostics et les mises à jour du poste de travail à distance. En profitant des interruptions matérielles générées par le chipset, les SMI, *System Management Interrupts* ils accèdent au SMM, *System Management Mode*. Un code malveillant de type rootkit peut prendre le contrôle du poste de travail car le système d'exploitation est incapable de se rendre compte de son exécution en mode SMM [7].
- Les travaux de Tavis Ormandy : la surconsommation des ressources par attaque DDoS ou le crach de l'hyperviseur/hôte [8].
- Xen Server : lecture possible de la mémoire d'autres machines virtuelles ou de l'hyperviseur. Xen émule 1024 *Model Specific Registers*, MSR. L'émulation APIC, *Advanced Programmable Interrupt Controller*, ne donne accès qu'à 256, alors si l'on accède au 257e... on provoque un débordement de tampon (CVE-2014-7188).
- Qubes OS, de Joanna Rutkowska, permet d'isoler des systèmes d'exploitation les uns des autres, pour scinder le poste entre plusieurs OS virtualisés [9]. Très gourmand en ressources il n'est pas de surcroit sans faille.

[1] ANSSI, *Problématiques de sécurité associées à la virtualisation des systèmes d'information*, http://www.ssi.gouv.fr/uploads/IMG/pdf/NP_Virtualisation_NoteTech_v1-1.pdf

[2] Joanna Rutkowska, *Subverting Vista Kernel For Fun And Profit*, BlackHat USA 2006, http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Rutkowska.pdf

[3] Loïc Duflot, *ACPI et routine de traitement de la SMI : des limites à l'informatique de confiance ?*, SSTIC 2009, https://www.sstic.org/2009/presentation/ACPI_et_routine_de_traitement_de_la_SMI_des_limites_a_l_informatique_de_confiance/

[4] Loïc Duflot, Yves-Alexis Perez, Guillaume Valadon, Olivier Levillain, *Quelques éléments en matière de sécurité des cartes réseau*, SSTIC 2010, https://www.sstic.org/2010/presentation/Peut_on_faire_confiance_aux_cartes_reseau/

[5] Fernand Lone Sang, Vincent Nicomette, Yves Deswarte, Loïc Duflot, *Attaques DMA peer-to-peer et contremesures*, SSTIC 2011, https://www.sstic.org/2011/presentation/attaques_dma_peer-to-peer_et_contremesures/

[6] Rafal Wojtczuk, Joanna Rutkowska, *Attacking Intel TXT*, http://invisiblethingslab.com/resources/bh09dc/Attacking%20Intel%20TXT%20-%20paper.pdf, http://invisiblethingslab.com/resources/2011/Attacking_Intel_TXT_via_SINIT_hijacking.pdf

[7] Joanna Rutkowska, Rafal Wojtczuk, *Preventing and detecting Xen hypervisor subversions*, BlackHat USA 2008, http://invisiblethingslab.com/resources/bh08/part2-full.pdf

[8] Tavis Ormandy, *An empirical study into the security exposure to hosts of hostile virtualized environments*, http://taviso.decsystem.org/virtsec.pdf

[9] *Qubes OS Project*, https://www.qubes-os.org/

[10] Nicolas RUFF, *Virtualisation ou sécurité*, http://www.ossir.org/jssi/jssi2009/3A.pdf

[11] Julien RAEIS, Nicolas COLLIGNON, *Virtualisation et sécurité*, http://www.hsc.fr/ressources/presentations/clusif-virtualisation/CLUSIF_VMware_20090204.pdf, 2009

[12] *ANNEXE_08_01 : Note de l'ANSSI - Virtualisation et sécurité du Cloud Computing*

Ce document pédagogique a été rédigé par un consortium regroupant des enseignants-chercheurs et des professionnels du secteur de la cybersécurité.

Version 2.5 — Février 2017