La Sécurité des Réseaux : une approche de détection de malveillances

Jacky AKOKA[1], Dominique BRIOLAT[2], Isabelle COMYN-WATTIAU[3]

[1]Conservatoire National des Arts et Métiers (CNAM)
[2]Ecole Supérieure des Sciences Economiques et Commerciales (ESSEC)
[3]ESSEC et Laboratoire PRiSM, Université de Versailles

RÉSUMÉ

L'objet de cet article est la présentation d'un modèle de comportement d'un utilisateur sur un réseau d'ordinateurs. Ce modèle est utilisé pour la détection avancée de malveillances sur un réseau. Le système, appelé DAMaR (Détection Avancée de Malveillances sur un Réseau), étudie le comportement ou profil habituel de l'utilisateur défini notamment par des composantes quantitatives (temps CPU, nombre moyen d'erreurs de connexion, nombre moyen d'utilisations des primitives du système et des logiciels, nombre d'accès en lecture/écriture à chaque base de données, etc.) et des composantes qualitatives (jour et heure habituels de connexion, poste de travail, etc.). Un changement de comportement peut dans certains cas traduire une intention de malveillance. Dès qu'un profil subit des modifications sensibles, une enquête est déclenchée. Ce profil n'est pas déterminé a priori, mais il se crée à l'issue d'une période d'apprentissage et d'observation des utilisateurs. Pour éviter de saturer le réseau, les utilisateurs sont répartis dans différentes classes de risque. Au départ, tous les individus sont dans la même classe. Au fur et à mesure des variations de leur comportement, les individus changent de classe. Le passage d'une classe à l'autre se fait soit de façon automatique soit par l'administrateur sur la suggestion de l'automate. Ce système, fonctionnant suivant un mode centralisé, a été expérimenté avec succès sur un réseau UNIX. Il a mis en évidence la dynamique du modèle en matérialisant les changements de classes des utilisateurs.

Mots-clés : Sécurité informatique, Réseau, Malveillance, Détection, Profil utilisateur, Classe de risque, Modèle de comportement, Automate centralisé.

Remerciements *: Nous tenons à remercier Sylvain Faurie, à l'origine de ce projet, Vincent Furnon qui a développé le prototype et SID Partners pour le financement.*

ABSTRACT

This paper describes a system allowing us to detect malevolences in a computer network. This system, called DAMaR (a French acronym for Advanced Detection of Malevolences in a Computer Network), analyzes users behavior characterized by quantitative components such as CPU time, average number of erroneous connections, average number of system and software primitives usage, etc. and qualitative components such as day and time of regular connections, workstation number, etc. Any change in the behavior may be interpreted as a malevolent intention. Whenever user profile changes in a significant manner, an enquiry is triggered. The profile is not predefined but initialized during a learning period where the system observes users. To avoid network saturation, users are assigned to risk classes. At the beginning, all the users are located in the same class. Users behavior variations induce class changes. The system has been implemented on a UNIX network on a centralized mode. It allowed us to illustrate the dynamics of the model by exhibiting user class changes.

Key-words : Computer security, Network, Malevolence, Detection, User profile, Risk class, Behavior model, Centralized system.

I. INTRODUCTION

Dire des entreprises qu'elles dépendent de leurs systèmes d'information pour acquérir et maintenir un avantage concurrentiel est devenu depuis un certain temps un lieu commun. Il est moins aisé d'en tirer les conséquences au niveau de la sécurité des systèmes d'information. Ces derniers sont vulnérables notamment depuis l'extension du rôle des réseaux, des autoroutes de l'information, et d'Internet. Pour préserver les systèmes d'information, toute entreprise doit assurer un niveau de qualité et de sécurité adéquat. Même si elle ne représente pas la panacée face à un fléau aux conséquences dramatiques, la sécurité informatique reste incontournable. Cette dernière se décline suivant deux composantes : physique et logique. La sécurité physique préserve les biens matériels contre les malveillances et les accidents matériels. La sécurité logique préserve les ressources immatérielles contre les erreurs et les malveillances immatérielles. Le niveau de sécurité à assurer est dépendant de la nature des risques. Ces derniers s'analysent en termes de :

- disponibilité des ressources matérielles et logicielles,
- intégrité des données et des programmes,
- confidentialité selon les autorisations d'accès aux ressources.

On considère généralement trois types de risques :

- *les accidents* menant à des destructions totales ou partielles de matériels, par incendie, dégât des eaux, explosions, chocs, etc. On y inclut aussi le vol, et le sabotage matériel. Enfin, les pannes et les dysfonctionnements font partie de ces risques accidentels ;
- *les erreurs* liées aux opérations de saisie, de procédures de transmission, et d'utilisation des données. A ces erreurs primaires s'ajoutent les erreurs logicielles, notamment lors de la conception et du développement. Enfin, on y inclut les erreurs d'exploitation ;
- *les malveillances* dues à des fraudes (détournement de fonds et de biens) et à des sabotages immatériels (sabotage de logiciels et de données). A ces malveillances s'ajoutent celles relatives à des copies illégales de logiciels et à des vols d'informations.

Ces risques peuvent être de nature volontaire ou involontaire. Ils peuvent provenir de personnes externes ou internes à l'entreprise. La répartition de ces risques et leurs conséquences en termes de disponibilité, intégrité et confidentialité sont données ci-dessous.

Durant plusieurs décennies, ces risques étaient concentrés dans un environnement restreint, celui des systèmes centralisés. Les systèmes informatiques actuels sont marqués par des architectures comportant des données et traitements répartis sur plusieurs systèmes localisés dans des sites géographiquement distincts. De ce fait, la sécurité informatique est devenue un maillon faible de la sécurité dans les entreprises. Les systèmes d'information et de communication des entreprises sont devenus extrêmement vulnérables et les pertes sont notables. En France, le montant des pertes est estimé par le CLUSIF (Club de la Sécurité Informatique Français) à 12,5 milliards de francs en 1996. Il s'agit d'une augmentation considérable, puisque ce montant était

estimé à 6 milliards en 1984. On considère généralement que les fraudes et les sabotages représentent 62 % de ce montant. Plus préoccupant est le fait que 80 % des fraudes soient imputés à du personnel interne aux entreprises. Aux USA, 84 % des entreprises ont connu des actes de malveillance, selon une étude d'Ernst & Young effectuée en 1995. Là aussi, 70 % de ces malveillances sont d'origine interne. Le coût total de la criminalité informatique est estimé à 555 milliards de dollars par an (étude du National Center Computer Crime Data en 1995). Le montant moyen d'une fraude informatique est estimé à 109 000 dollars. Près de 90 % des compagnies américaines ont perdu de l'argent suite à une fraude informatique. L'enquête d'Ernst & Young effectuée en 1995 aux USA montre que 50 % des entreprises ont perdu des informations (ou données) critiques durant les deux dernières années précédant l'enquête.

Ces statistiques relatives aux risques informatiques attestent de la gravité de la situation. Le plus préoccupant est certainement l'augmentation des pertes dues à la malveillance : fraude, sabotage, détournement de logiciel, indiscrétion et détournement d'informations. Les données publiées par le CLUSIF donnent la répartition suivante des pertes : 25 % par accident, 17 % par erreur et 58 % pour les actes de malveillance, avec un accroissement de 11 % en 10 ans. La malveillance connaît une expansion vertigineuse, particulièrement inquiétante. En 1984, le montant des pertes dues à la malveillance en France était de deux milliards de francs. Ce montant a doublé en 1987. En 1992, il atteint le chiffre de 5,9 milliards de francs (SAC, 1993). L'explosion du phénomène "réseaux" et micro accroît ce risque avec des prévisions de près de seize milliards de francs en 1998. Pour se prémunir contre de tels risques, les entreprises doivent se doter de moyens de prévention. L'accentuation des pertes est en effet susceptible de porter atteinte à l'existence même des entreprises.

L'objet de cet article est la présentation d'un système de Détection Avancée de Malveillances sur un Réseau d'ordinateurs (DAMaR). Il est fondé sur un modèle de comportement de l'utilisateur face à la machine. Il intègre les quatre dimensions caractéristiques de l'interface homme-machine : conceptuelle, communication, physique et contextuelle. La *deuxième section* propose un état de l'art sur la sécurité dans les réseaux informatiques. La *troisième section* décrit les concepts fondamentaux du modèle et du système. La *section suivante* décrit en détail la mise en œuvre du modèle de détection avancée de malveillances sur un réseau d'ordinateurs et son application sur un réseau UNIX. La *cinquième section* conclut et propose des voies nouvelles de recherche.

II. SÉCURITÉ DES RÉSEAUX INFORMATIQUES : ÉTAT DE L'ART

Il est généralement admis que la grande majorité des malveillances informatiques ont pour origine l'attitude des personnes "autorisées" à accéder aux ressources informatiques et aux données. L'enquête du Michigan State University effectuée en 1995 évalue à 70 % des malveillances,

celles perpétrées par du personnel interne. Les auteurs de ces malveillances peuvent être les employés, les cadres ou même des sous-traitants. La menace représentée par le personnel interne est considérée la plus dangereuse. En effet, ce personnel, à la différence des acteurs externes, connaît bien les systèmes. Il est familier avec les procédures d'accès. De plus, ce personnel ne fait pas *a priori* l'objet d'une suspicion de la part des spécialistes chargés de la sécurité informatique. L'attitude de ce personnel interne entraîne donc des menaces qui pèsent considérablement sur les données, notamment pour leur intégrité, confidentialité et disponibilité. L'enquête menée par le Michigan State University en 1995 est conforme à celle réalisée par le SAC en 1991. Elle montre que, pour 60 % des personnes interrogées, les accès ou modifications non autorisés aux données constituent l'un des principaux risques. Les moyens jugés les plus efficaces pour faire face à ces risques sont :

- la sécurité et les contrôles d'accès (73 %),
- les politiques, normes et procédures (7 %),
- la séparation des tâches (6 %).

II.1. Les outils de la sécurité

On considère qu'il existe généralement deux modèles de sécurité des systèmes d'information répartis :

- la sécurité par le réseau,
- la sécurité par l'hôte.

Dans le cas de la sécurité par le réseau, celle-ci est pensée par rapport à l'accès depuis le réseau aux serveurs et services qu'il offre. Les "firewalls" (pare-feux) sont un exemple d'outil conforme à ce modèle. Le principe consiste à contrôler l'accès selon le contenu des paquets d'une connexion. Le "firewall" fournit donc un point unique de contrôle pour la protection d'un réseau. Un "firewall" a une double vocation :

- protéger le réseau des attaques extérieures ;
- contrôler les accès afin d'interdire ou de limiter l'usage de certains services.

Parmi les produits les plus répandus, citons :

- Checkpoint Firewall-1 3.0 (de Checkpoint Software Technologies),
- IBM Firewall 3.1,
- M>Wall 2.0 (de Matra),
- Solsoft Net Security Master 3.0 (de Solsoft).

Dans le cas de la sécurité par l'hôte, la sécurité est pensée machine par machine. Il est aisé de comprendre que le nombre de machines et leur diversité au sein d'un réseau sont un frein à ce schéma de sécurité. Ce modèle n'est rentable que pour les sites les plus petits et les plus simples. Les outils qui s'inscrivent dans ce modèle sont les logiciels de cryptographie, les outils de contrôle d'accès, de vérification de l'identité, etc. Parmi ces outils, citons :

- SecuRemote 3.0 (de Checkpoint Software Technologies), logiciel de chiffrement client qui multiplie les opérations d'authentification des utilisateurs distants.
- SessionWall-3 (de Abirnet) qui conserve les traces de toutes les transactions réalisées sur tout ou partie du réseau Intranet.
- Cybercop (de Network General) qui permet une surveillance du

réseau Intranet en temps réel. Il se place en aval d'un "firewall" et permet de détecter les visiteurs qui auraient contourné ce dernier. Enfin, il transmet les informations au nœud central du système qui déclenche les alarmes.

- D-Control (Calyx Data Control) se charge de chiffrer les fichiers et de n'autoriser l'accès aux données protégées qu'après identification.
- SmartPass/56 assure une communication privée confidentielle en chiffrant les communications entrantes et sortantes.
- Satan 1.1.1 (Dan Farmer) permet de façon interactive de chercher la présence de failles de sécurité sur le réseau. Il teste aussi les paramétrages des "firewalls" et des routeurs.
- Le produit Access Master de la société Bull permet de :
  - gérer les personnes et leurs privilèges ;
  - assurer une authentification stricte ;
  - affecter aux utilisateurs des attributs de sécurité et une inscription implicite à un service donné ;
  - assurer des fonctions de chiffrement et de déchiffrement ;
  - mettre en œuvre un système d'audit de manière à déclencher une politique de localisation et de prévention sur certains types d'événements (connexion, déconnexion, etc.) de manière à localiser ou prévenir des fraudes. Malgré quelques similitudes, notre système DAMaR, décrit dans cet article, se différencie d'Access Master par :
    - un modèle global de la sécurité,
    - le concept de profil d'utilisateur,
    - une fonction de surveillance,
    - la définition de classes de risques.

II.2. Les concepts de la sécurité logique

Le contrôle de la sécurité logique comprend :

- *l'identification* des utilisateurs des données et des ressources,
- *l'authentification* de ces utilisateurs,
- *le contrôle d'accès* aux données et aux ressources,
- *l'audit du comportement* des utilisateurs.

L'identification des utilisateurs est une phase importante de la sécurité. Il s'agit de connaître précisément l'identité de tout utilisateur. Il existe plusieurs modèles d'identification (Krauss 1980, Parker 1981, Smith et Lim 1984, Parker 1986, Weber 1988, Wood 1990, Clay 1995).

L'authentification est le processus de vérification des personnes. Elle permet la reconnaissance d'un individu à l'aide de ses caractéristiques propres et de ses différences par rapport à un groupe ou une organisation. Ce processus est généralement utilisé à des fins de contrôle pour l'accès aux réseaux et aux ressources informatiques. Cela nécessite généralement l'utilisation d'un mot de passe. Certaines techniques sont utilisées pour protéger ce mot de passe (Baskerville 1989, 1992, 1993). L'authentification par mot de passe comporte des lacunes et peut être utilement complétée par l'exhibition d'objet (Fisher 1984)

- le nombre d'utilisations de chaque ressource matérielle,
- le nombre d'utilisations des primitives du système,
- le nombre d'utilisations de chaque logiciel,
- le nombre d'utilisations de chaque base de données ou fichier,
- le temps CPU utilisé,

qui constituent la dimension *conceptuelle*.

La dimension *communication* caractérise la nature du dialogue homme-machine. Elle traduit le comportement cognitif de l'utilisateur. Parmi les paramètres les plus caractéristiques, nous avons retenu le nombre d'erreurs de connexion qui peut traduire la difficulté pour l'utilisateur malveillant de se conformer au mode cognitif original.

Parmi les composantes *physiques*, nous avons inclu les jours et les heures de connexion et de déconnexion, le poste de travail (identification et nature).

Enfin, pour caractériser la dimension *contextuelle* qui traduit l'insertion de l'utilisateur dans le cadre de procédures organisationnelles, nous utilisons :

- la séquence d'instructions après la connexion,
- la séquence d'instructions précédant la déconnexion,
- le nombre de violations d'autorisation.

Cette liste ne prétend pas à l'exhaustivité. Elle présente cependant l'avantage de caractériser, à l'aide d'un nombre fini de paramètres mesurables, les quatre dimensions de l'interface homme-machine.

La prise en compte de ces différents paramètres dans l'élaboration et le suivi du profil d'activité d'un utilisateur nous a conduits à distinguer les variables "quantitatives" et les variables "qualitatives". Les composantes quantitatives résultent de statistiques sur les actions (comptage, somme, etc.). Elles peuvent subir des opérations mathématiques. Il s'agit des caractéristiques suivantes :

- le temps CPU utilisé,
- le nombre d'erreurs de connexion,
- le nombre de violations d'autorisation,
- le nombre d'utilisations de chaque ressource matérielle,
- le nombre d'utilisations des primitives du système,
- le nombre d'utilisations de chaque logiciel,
- le nombre d'utilisations de chaque base de données ou fichier.

Les composantes qualitatives reflètent plutôt un état et ne se prêtent généralement pas aux opérations mathématiques. A titre d'exemple, mentionnons les composantes suivantes :

- les heures de connexion et déconnexion,
- les dates de connexion et déconnexion,
- la séquence d'instructions après la connexion,
- la séquence d'instructions précédant la déconnexion,
- le poste de travail (identification et nature).

Chacun des deux types de composantes devra être analysé de façon différente, comme décrit plus loin. Toutefois, le profil est défini par l'ensemble des composantes à la fois quantitatives et qualita-

tives. Nous donnons ci-dessous un exemple de profil d'activité, caractérisant un développeur d'applications :

Caractéristique	Valeur	Quantitatif	Qualitatif	Conceptuel	Communication	Physique	Contextuel
Identification utilisateur	XXXX						
Jours et heures de connexion	Lundi-Samedi/7h-19h		X			X	
Poste	Eventuellement plusieurs, mais prédéterminés		X			X	
Temps CPU consommé	30 minutes	X		X			
Nombre d'erreurs de connexion	0	X			X		
Séquence à la connexion	Editeur, mail, compilateur (C, ...), sessions de tests, ...		X				X
Séquence à la déconnexion	Mail, logout		X				X
Nombre d'utilisations de chaque primitive système	Important (à déterminer plus précisément)	X		X			
Nombre d'utilisations de chaque logiciel ou progiciel installé	10 à 30 pour compilateur, 1 à 10 pour éditeur, 10 à 30 pour débogage	X		X			
Nombre d'accès en lecture (et écriture) à chaque base de données	10 à 30 pour bibliothèque de languages, variable pour les autres bases de données	X		X			
Nombre de violations d'autorisations	0 à 30 à cause d'éventuelles erreurs de programmation	X					X
Nombre d'utilisations de chaque ressource matérielle	1 à 30 pour imprimante matricielle, 0 à 20 pour imprimante laser, 0 à 5 pour lecteur de disquette, ...	X		X			

III.3. Classe de risque

La surveillance du réseau ne doit pas pénaliser son fonctionnement. De plus, il est inutile de saturer le réseau en surveillant tous les utilisateurs alors que les contrevenants sont largement minoritaires. De façon pratique, on constitue cinq classes de risque (la classe 0 étant la classe la moins suspecte, la classe 4 la plus suspecte). Au départ, tous les individus sont rangés dans la même classe (classe 0). Au fur et à mesure des variations de comportement, on transfère les individus dans des classes de plus haut niveau. Bien sûr, on permet à un individu d'être déplacé dans des classes à risque moindre une fois que son "innocence" est prouvée. Cela peut être fait manuellement par le responsable de la surveillance du réseau ou automatiquement par le système.

III.4. Fonction de surveillance

La nature spécifique des composantes qualitatives et quantitatives du profil utilisateur nécessite une analyse différenciée des données qui les constituent.

a) Traitement des composantes quantitatives

On considère ici les composantes quantitatives décrites plus haut. Soit $X_{i,k}$ le vecteur profil à l'instant k pour un individu i et X_i le vecteur profil de référence du même individu. Les vecteurs sont de dimension n, n étant le nombre de composantes. Soient $X_{i,k,j}$ et $X_{i,j}$, les jèmes composantes de ces vecteurs.

La variation du comportement de l'utilisateur par rapport à son profil n'est considérée comme préoccupante qu'au-delà d'un certain seuil appelé seuil de tolérance. Soit T_i le vecteur de tolérance associé à un individu. Chaque composante de ce vecteur est un pourcentage représentant la variation de profil au-delà de laquelle on suppose qu'il y a eu une violation de sécurité.

On appelle violation tout écart du comportement au-delà du seuil de tolérance. Soit $V_{i,k}$ le nombre de violations perpétrées par un individu i à l'instant k :

$$V_{i,k} = \sum_j E \left[\frac{X_{i,k,j} - X_{i,j}}{X_{i,j} \times \frac{T_{i,j}}{100}} \right]$$

où E représente la fonction partie entière.

Remarquons qu'une variation de profil qui dépasse deux fois la tolérance est comptabilisée comme deux violations. De même, une variation de n fois la tolérance aboutit à n violations.

On peut aussi pondérer cette somme suivant les individus et les composantes du profil. Soit $P_{i,j}$ le coefficient de pondération de la composante j de l'individu i.

Le nombre pondéré de violations $V_{i,k}$ d'un individu i à l'instant k est alors égal à :

$$V_{i,k} = \sum_j E \left[P_{i,j} \frac{X_{i,k,j} - X_{i,j}}{X_{i,j} \times \frac{T_{i,j}}{100}} \right]$$

où $T_{i,j}$ représente la jème composante de la tolérance associée à un individu i.

La fonction de surveillance f définit le nombre de violations au-delà duquel le changement de classe est effectué :

$$\Delta\, Classes_{i,k} = f(V_{i,k})$$

Cette fonction est définie intuitivement au départ, puis ajustée de façon empirique. Elle dépend de la taille des vecteurs profils et de la sensibilité voulue. La définition initiale de f est la suivante :

- si $0 \le V_{i,k} < 3$ alors $f(V_{i,k}) = 0$
- si $3 \le V_{i,k} < 7$ alors $f(V_{i,k}) = 1$
- si $7 \le V_{i,k} < 10$ alors $f(V_{i,k}) = 2$
- si $10 \le V_{i,k}$ alors $f(V_{i,k}) = 3$

A ce stade, deux remarques s'imposent :

- dans une utilisation réelle, il conviendra de crypter les profils de façon à éviter toute modification non autorisée ;
- à la fin de chaque période de référence, les profils sont réini-

tialisés afin que la comparaison avec le profil de référence ait un sens.

b) Traitement des composantes qualitatives

On rappelle que les composantes qualitatives du profil sont :

- les heures de connexion et déconnexion,
- les dates de connexion et déconnexion,
- la séquence d'activités à la connexion,
- la séquence d'activités à la déconnexion,
- le poste de travail (identification et nature).

A chaque fois que cela s'avère nécessaire, DAMaR compare une de ces composantes avec la situation courante. S'il y a une différence, on ajoute à $V_{l,k}$ les valeurs suivantes :

- d_1 pour une heure de connexion qui n'est pas dans la plage horaire prévue ;
- d_2 pour une date de connexion qui n'est pas prévue ;
- d_3 pour chaque différence dans la séquence d'activités à la connexion (on prendra ensuite la valeur entière) ;
- d_4 pour chaque différence dans la séquence d'activités à la déconnexion ;
- d_5 pour une modification liée au poste de travail.

Les valeurs des d_i sont déterminées par l'administrateur du réseau. Notons que le changement de classe s'effectue suivant la procédure décrite plus haut.

Finalement, la variation du profil d'un individu peut être décrite par la somme en valeurs absolues des variations quantitatives et qualitatives :

$$\left|V_{lk}\right| + \sum_i \left|d_i\right|$$

IV. LE SYSTÈME DE DÉTECTION AVANCÉE DE MALVEILLANCES : ARCHITECTURE ET FONCTIONNALITÉS

Dans cette partie, après avoir justifié la localisation du module de surveillance, nous décrivons successivement la structure de l'automate et ses fonctionnalités.

IV.1. Localisation du Module de Surveillance

La surveillance d'un réseau peut être effectuée de manière centralisée ou décentralisée. La solution décentralisée consiste à installer un "mouchard" sur chaque machine. Ce dernier est alors activé à chaque connexion. Il envoie périodiquement (en fonction de la classe de risque de l'utilisateur) des messages à une console centrale sur laquelle s'exécute en permanence un programme de contrôle. Les "mouchards" ont pour but d'enregistrer les opérations réalisées par l'utilisateur. Ils mettent à jour son profil, calculent périodiquement les violations et transmettent le résultat au contrôleur. A chaque réception de ces informations, le programme contrôleur vérifie s'il y a un changement de classe. Dans l'affirmative, il modifie la fréquence de surveillance de l'utilisateur concerné. Le programme contrôleur a donc ici un rôle plus passif que dans la première solution. C'est lui néanmoins qui va générer les différents

types d'alarmes et servir d'interface avec la personne chargée de la sécurité du réseau. L'avantage d'une telle solution réside dans sa simplicité de mise en œuvre. Il est plus facile, en effet, de créer des processus, sur différentes machines, qui communiquent entre eux que d'en gérer un seul chargé de décoder les trames qui passent sur le réseau. Une telle solution peut néanmoins complexifier le contrôle de l'intégrité des données, les profils étant manipulés au niveau de chaque machine. Cependant on peut remédier à ce problème en effectuant un cryptage des fichiers critiques.

La surveillance d'un réseau peut aussi s'effectuer de manière centralisée à l'aide d'un programme contrôleur localisé sur une machine unique (la console du super-utilisateur, par exemple). Ce programme doit alors filtrer les messages circulant sur le réseau en déterminant leurs expéditeurs et destinataires. Ainsi il peut mettre à jour les profils des utilisateurs en temps réel et procéder à l'analyse de ces données sur la console. L'inconvénient d'une telle solution réside dans la nécessité de filtrer et analyser tous les messages qui circulent sur le réseau. Cela peut être contraignant s'il y a un grand nombre d'utilisateurs et/ou si le système d'exploitation ne permet pas de surveiller directement les trames qui transitent sur le réseau. Cette méthode présente toutefois deux avantages majeurs :

- elle assure un contrôle très précis de l'activité du réseau ;
- elle facilite la protection des données de surveillance telles que les profils des utilisateurs.

Pour ces raisons, nous avons préféré cette solution dans laquelle l'automate de surveillance fonctionne sur un mode centralisé. Il est installé sur une machine unique (la console super-utilisateur ou serveur d'un réseau UNIX par exemple) et filtre les messages circulant dans le réseau. Il met à jour en temps réel le profil des utilisateurs. Il procède à l'analyse des données recueillies sur chaque utilisateur.

IV.2. Fonctionnement et Structure de l'Automate

Le fonctionnement de l'automate, schématisé à la figure 1, peut être brièvement décrit comme suit. A chaque connexion de l'utilisateur, on déclenche un processus "mouchard" depuis le serveur. Ce processus communique avec le programme de contrôle via des fichiers cryptés. Il se positionne entre l'interpréteur de commandes (le "shell") et l'utilisateur. Il enregistre en temps réel toutes les actions de l'utilisateur. Aucune analyse n'est effectuée à ce stade.

Le programme de contrôle central comporte un processus de récupération des informations qui effectue une surveillance plus globale du réseau (liste des utilisateurs connectés, temps CPU, date et heure de connexion / déconnexion et toutes les composantes non quantitatives du profil). Il lui incombe aussi d'effectuer les analyses des données et d'assurer l'interface avec la personne chargée de la surveillance. Cette sécurisation du système interdit à toute personne, même bien renseignée et compétente, de le contourner.

Dès la saisie d'une commande utilisateur, le processus "mouchard" informe le programme de contrôle central par le biais d'un

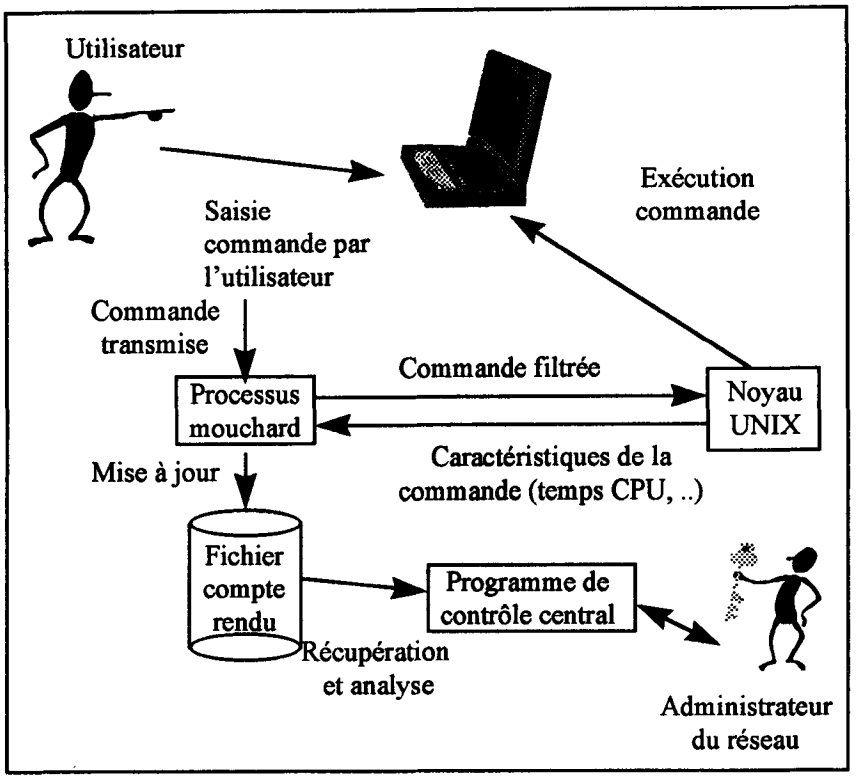

Figure 1 : Fonctionnement de DAMaR

fichier compte rendu. La commande ainsi filtrée est alors envoyée au noyau UNIX qui l'exécute et rend compte à l'utilisateur. Il rend compte aussi au "mouchard" en expédiant un résumé des caractéristiques de l'exécution de la commande (temps CPU, etc.). Le programme de contrôle central analyse le contenu des fichiers comptes-rendus et effectue l'ensemble des tâches de surveillance :

- comparaison des profils,
- calcul des violations,
- changement de classes,
- etc.

Il permet aussi tous les paramétrages possibles et assure l'interface avec la personne chargée de la sécurité.

Pour être certain qu'un utilisateur est bien sous surveillance, il faut sécuriser le "mouchard". Pour cela, il suffit de consulter régulièrement la liste des utilisateurs connectés (sous UNIX, l'instruction "who" est suffisante par exemple) et de la comparer à la liste des personnes sous surveillance. Si un utilisateur est connecté mais n'est pas surveillé, on supposera qu'il essaie de frauder et on déclenchera l'action appropriée (par exemple sa déconnexion). On empêche aussi l'envoi de terminaison (via "kill", sous

UNIX) vers le processus "mouchard".

Sous UNIX, les deux processus suivants tournent donc en parallèle tout en communiquant à l'aide de tubes *(pipe)* ou de messages *(kill et signal)* :

- le *processus de surveillance et de recueil des informations* : modifie et met à jour périodiquement les profils courants des utilisateurs. Il émet également les alarmes après avoir analysé ces nouveaux profils ;
- le *processus père* crée le processus précédent. Il permet au responsable de la sécurité de paramétrer la surveillance (périodes, mode de fonctionnement), d'éditer et de consulter les profils, les filtres, les tolérances, les fonctions de changement de classes et de calcul du niveau de violation ou de danger des utilisateurs (figure 2).

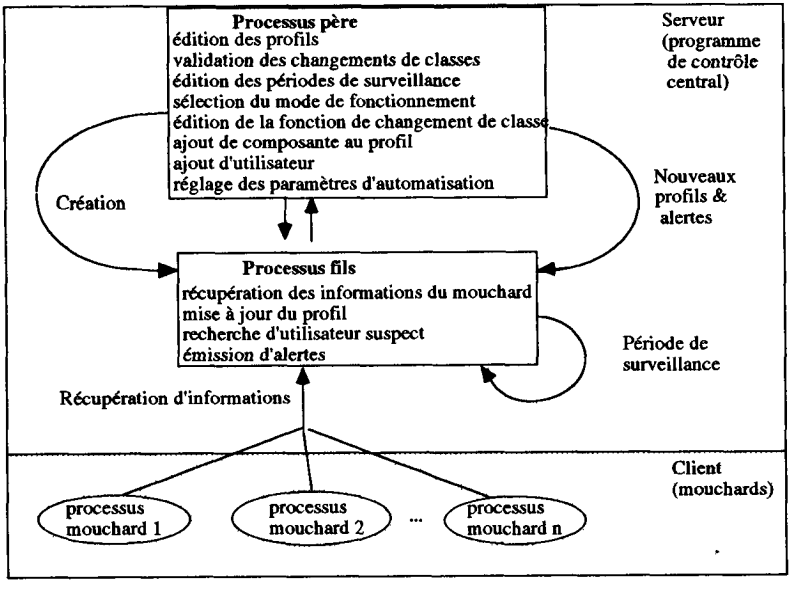

Figure 2 : Structure de DAMaR

IV.3. Les fonctionnalités de DAMaR

Le prototype de surveillance réalise les fonctions suivantes :

- édition et mise à jour des profils,
- édition et mise à jour des périodicités de surveillance,
- paramétrage de l'autonomie de décision relative au changement de classe,
- paramétrage de la fonction de changement de classe,
- ajout d'utilisateur,
- ajout/modification de composantes au profil des utilisateurs,
- validation du changement de classe.

DAMaR a été expérimenté sur un réseau UNIX. Le logiciel écrit en C propose dans un menu les fonctionnalités suivantes (figure 3).

MENU PRINCIPAL DE SURVEILLANCE

0) Validation des changements de classes - alerte !
1) Edition / mise à jour des profils
2) Edition / mise à jour des périodicités de surveillance
3) Sélection du mode opératoire (surveillance / mise en route)
4) Paramétrage de l'autonomie de décision
5) Paramétrage de la fonction de changement de classes
6) Changer un utilisateur de classe
7) Ajout d'un utilisateur
8) Ajout d'une composante
9) Fin

Tapez la touche correspondant à votre choix :
?

Figure 3 : Menu de lancement de DAMaR

La figure 4 visualise l'écran de consultation et de modification des profils de tous les utilisateurs contenus dans la base de données :

EDITION / MISE À JOUR DE PROFILS

User ID : FURNON — Classe de risque : 2
Info : Furnon Vincent Stagiaire

	Profil de référence	Profil courant	Pondération	Tolérance
Nb util. kill	4	5	2	50 %
Nb util. ps	10	4	1	20 %
Nb util. who	19	15	1	50 %
Nb util. prog (X)	12	5	1	30 %
Nb util. rm	20	34	1	100 %

(X : composante bloquée par le filtre)

F1 : AIDE M : MODIFICATION L : LISTER A : ALLER D : DÉTRUIRE S : SPÉCIALE F10 : FIN

Figure 4 : Edition / mise à jour de profils

On peut alors modifier respectivement le filtre, le nombre d'utilisations de la commande (courant et de référence), le coefficient de pondération et la tolérance de la ligne choisie.

Le filtre active ou désactive la prise en compte des données relatives à une commande lors de l'évaluation des changements de classes. Le coefficient de pondération (entier et supérieur à 1) est

un coefficient multiplicatif du nombre de violations produites par la commande en question. La tolérance est le pourcentage de variation du nombre d'utilisations de la commande au-dessous duquel on considère qu'il n'y a pas de violations.

On peut ainsi lister tous les utilisateurs triés par classe ou consulter directement la fiche d'un utilisateur. On peut aussi détruire les données concernant l'utilisateur à l'écran. Enfin, on peut avoir accès au compte rendu sur les composantes spéciales du profil. On peut encore avoir accès aux informations qualitatives telles que :

- la séquence de commandes utilisées juste après la connexion de l'utilisateur (et aux violations par rapport à la séquence prévue) ;
- la plage horaire où l'utilisateur devrait se connecter (et aux violations qui en découlent) ;
- la violation des jours autorisés pour la connexion ;
- le temps CPU courant et de référence de l'utilisateur (et aux violations qui en découlent).

Ainsi, les fonctionnalités de DAMaR décrites dans ce paragraphe concrétisent le concept de surveillance d'un réseau d'ordinateurs et celui de détection avancée de malveillances. Les fonctionnalités peuvent être étendues à d'autres environnements que celui d'Unix.

IV.4. Expérimentation de DAMaR et limites

Le prototype a été expérimenté sur un réseau UNIX comprenant vingt machines à la disposition d'étudiants dans une université. Les étudiants concernés suivaient une formation spécialisée impliquant le développement de projets informatiques utilisant des compilateurs, des systèmes de gestion de bases de données, des outils bureautiques, etc. Les étudiants n'étaient pas informés de la présence du "mouchard". La période de rodage pour l'élaboration des profils a été de trois jours. La période de test a duré deux jours. Elle a permis de constater les changements de classe se produisant à chaque fois qu'un étudiant passait d'un projet à un autre. L'opérateur pouvait alors se rendre auprès du poste de travail et constater la modification effective du profil. Notons cependant qu'aucune action de malveillance n'a été détectée pendant cette période, ce qui nous rassure quant au sérieux de nos étudiants. Bien sûr, nous avons aussi testé nous-mêmes la réactivité du prototype en répétant des commandes erronées, en changeant de poste de travail, etc. La présence du "mouchard" sur le réseau n'a pas entraîné de dégradation des performances du réseau. Finalement, l'expérimentation nous a paru prometteuse.

Il convient toutefois de relativiser ces conclusions en soulignant les limites inhérentes à cette expérimentation :

- Le contexte universitaire est très spécifique. Il serait souhaitable d'effectuer une expérimentation comparable dans un environnement professionnel.
- La non-dégradation des performances du réseau doit être vérifiée sur un réseau de plus grande taille.
- Le réseau ayant servi à l'expérimentation est composé de machines homogènes. Cette caractéristique peut peut-être faciliter la tâche du prototype.

V. CONCLUSION ET NOUVELLES VOIES DE RECHERCHE

Dans cet article, nous avons présenté les caractéristiques principales d'un modèle de comportement d'utilisateur, sous-jacent à un système chargé de détecter les malveillances pouvant survenir dans un réseau d'ordinateurs. Il permet de détecter les comportements malveillants des utilisateurs du réseau. Ces derniers sont décrits par un profil ayant des composantes tant quantitatives que qualitatives, décrivant les quatre dimensions conceptuelle, communication, physique et contextuelle de l'interface homme-machine. Ce profil n'est pas nécessairement déterminé *a priori* mais peut être construit de manière dynamique. Les utilisateurs sont au départ mis dans une même classe de risque. Le passage à une autre classe de risque se fait suivant le changement de comportement des utilisateurs du réseau, laissant présager une attitude malveillante.

L'automate fonctionne sous un mode centralisé, permettant de réaliser une analyse des données recueillies sur chaque utilisateur. Un prototype a été développé en C sous UNIX. Il donne des résultats très satisfaisants.

Le prototype a été comparé au progiciel PC_AUDIT de sécurité de Staff&Line. Ce dernier effectue uniquement un historique précis des événements relatifs à un poste de travail observé sans établir de profils utilisateurs. Contrairement à DAMaR, il n'analyse pas les données relevées, ne "juge" pas l'attitude des utilisateurs et n'établit pas de classes de risque. Il paraît acceptable pour une station unique, fonctionnant sous DOS. Il ne peut être utilisé sur un réseau fonctionnant sous UNIX. Une comparaison plus complète sera réalisée avec les produits phares du marché.

L'approche utilisée dans DAMaR n'est pas spécifique à UNIX et nous envisageons le portage de l'automate sur d'autres systèmes d'exploitation (VMS/DEC, MVS/IBM, etc.). Dans cette optique, l'algorithme principal d'étude des profils et de détermination des changements de classe reste totalement applicable. En revanche, le "mouchard", utilisant des paramètres et des commandes spécifiques à UNIX, nécessitera une adaptation au nouvel environnement. Le prototype peut être utilement complété dans un développement futur par le cryptage des données et l'identification des usurpateurs d'identité par la méthode dite "du chasseur d'informations" (Delobel et Adiba 1982). D'autres voies de recherche et d'expérimentation concernent notamment l'interfaçage du prototype avec d'autres outils, par exemple les "firewalls". Sur un plan plus théorique, nous souhaitons aussi étendre l'analyse de la raison des malveillances, de façon à ce que le système soit en mesure de distinguer les usurpateurs d'identité des utilisateurs autorisés mais mal intentionnés.

BIBLIOGRAPHIE

Akoka, J., Briolat, D. et Comyn-Wattiau, I. (1995), « DAMaR - Automate de détection avancée de malveillances sur un réseau », *Actes XIIIe Congrès de l'AFAI - Réussir avec les nouvelles technologies de l'information*, Paris, 1995.

Akoka, J. et Comyn-Wattiau, I. (1996), « A Knowledge Based System For Auditing Computer And Management Information Systems », *Expert*

Systems with Applications, Vol. 11, n° 3.

Baskerville, R. (1989), « Logical Controls Specification : An Approach to Information Systems Security », In *Systems Development for Human Progress*, North-Holland, Amsterdam.

Baskerville, R. (1992), « The Developmental Duality of Information Systems Security », *Journal of Management Systems*, Vol. 4, n° 1.

Baskerville, R. (1993), « The Threat in Security for the Adaptive Organization », *Information Systems*, Vol. 2, n° 1.

Beck, L.L. (1980), « A Security Mechanism for Statistical Databases », *ACM Transactions on Database Systems*, Vol. 5, n° 3.

Bonczek, R.H. et al. (1977), « A Transformational Grammar-based Query Processor for Access Control in a Planning System », *ACM Transactions on Database Systems*, Vol. 2, n° 4.

Buss, M.D.J. et Salerno, L.M. (1984), « Common Sense and Computer Security », *Harvard Business Review*, March-April.

Clay, B.M. (1995), « MC Security Criteria A 70Z », *IS Audit & Control Journal*, Vol. 5.

Delobel, C., Adiba, M. (1982), *Bases de Données et Systèmes Relationnels*, Dunod, Paris.

Denning, D.E. et Denning, P.J. (1979), « Data Security », *ACM Computing Surveys*, Vol. 11, n° 3.

Denning, D.E. (1982), *Cryptography and Data Security*, Addison-Wesley, Reading, Mass.

Ducloux, G. (1990), « Sécurité des Réseaux et Cryptographie », *Marketing Réseaux*, IBM France.

Ducloux, G. (1990), « Sécurité Informatique : Concepts, Stratégie et Produits », *Marketing Réseaux*, IBM France.

Faivre, C. et Loreau, Y.-M. (1992), *Audit de la Micro-Informatique*, Publi-Union.

Faurie, S. et Sarret, P. (1991), *L'Audit Informatique*, Masson & Nouvelles Editions Fiduciaires, Paris.

Fidji, P. (1993), « Superviser en Temps Réel pour Prévoir les Défaillances », 01 Informatique, n° 22.

Fisher, R. (1984), *Information Systems Security*, Prentice-Hall, Englewood Cliffs, N.J.

Fitzgerald, J. (1978), *Internal Controls for Computerized Systems*, Underwood Press, San Ceandro, California.

Fitzgerald, J. (1990), *Designing Controls into Computerized Systems*, 2d ed., Jerry FitzGerald & Associates, Redwood City, California.

Gallegos, F., Richardson, D. et Borthick, A. (1987), *Audit and Control of Information Systems South-Western*, Cincinnati, Ohio.

Guinier, D. (1990), *Sécurité et Qualité des Systèmes d'Information - Approche Systémique*, Masson, Paris.

Jameson, J.S. (1996), « Deterring Information Technology Crime », *Information System Audit and Control Association*, 24th International Conference, Calgary.

Jouas, J.-P., Harari, A., Lamère, J. M. et Tourby, J. (1992), *Le Risque Informatique*, Dunod, Paris.

Krauss, L. (1980), *SAFE : Security Audit and Field Evaluation for Computer Facilities and Information*, Revised ed., Amacon, New York.

Lamère, J.-M. (1987), *La Sécurité des Réseaux : Méthodes et Techniques*, Dunod, Paris.

Landwehr, C.E. (1981), « Formal Models for Computer Security », *ACM Computing Surveys*, Vol. 13, n° 3.

Landwehr, C.E. (1983), « The Best Available Technologies for Computer Security », *IEEE Computer*, Vol. 16, n° 7.

Madnick, S.E. (1978), « Management Policies and Procedures Needed for Effective Computer Security », *Sloan Management Review*, Fall.

Marro, P.E. (1995), « Overview of Computer Crime and Security », *IS Audit & Control Journal*, Vol. V.

McLean, J. (1990), « The Specification and Modeling of Computer Security », *Computer*, Vol. 23, n° 1.

Moran, T.P. (1981), « Command Language Grammar », *International Journal Man-Machine Studies*, n° 15.

Parker, D. (1981), *Computer Security Management*, Reston, Mass.

Parker, D. (1986), *Computer Crime : Computer Security Techniques*, U.S. Department of Justice, Bureau of Justice Statistics, Document J29.2: C86, Washington D.C.

SAC (1991), *IIARF, Audit et Contrôle des Systèmes d'Information*, Sac Report 1991 (Traduction IFACI).

SAC (1993), *Audit et Contrôle des Systèmes d'Information - Module 8*, Sac Report (Traduction IFACI).

Smith, S. et Lim, J. (1984), « An Automated Method for Assessing the Effectiveness of Computer Security Safeguards », In *Computer Security : A Global Challenge*, North-Holland, Amsterdam.

Von Solms, R., Van de Haar, H., Von Solms, S.H. et Caelli, W.J. (1994), « A Framework for Information Security Evaluation », *Information & Management*, Vol. 26.

Thuraisingham, B. (1995), « Multilevel Security for Information Retrieval Systems », *Information & Management*, Vol. 28.

Stonebraker, M. et Wong, E. (1974), « Access Control in a Relational Database Management System by Query Modification », *Proceedings of ACM Annual Conference.*

Weber, R. (1988), *EDP Auditing : Conceptual Foundations and Practice*, 2nd ed., McGraw-Hill, New York.

Wood, C. (1990), « Principles of Secure Information Systems », *Design. Comput.*, Vol. 9, n° 1.

Zemor, G. (1990), *Sécurité des transmissions et des réseaux*, Polycopié de cours, ENST Paris.