(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
12 août 2004 (12.08.2004)

PCT

(10) Numéro de publication internationale
WO 2004/068817 A2

(51) Classification internationale des brevets[7] : H04L 29/06

(21) Numéro de la demande internationale :
PCT/FR2004/050009

(22) Date de dépôt international : 8 janvier 2004 (08.01.2004)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
03/00719 23 janvier 2003 (23.01.2003) FR

(71) Déposant *(pour tous les États désignés sauf US)* : **EVERBEE NETWORKS S.A.** [FR/FR]; 41, boulevard des Capucines, F-75002 Paris (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants *(pour US seulement)* : **FAILLENOT, Laurent** [FR/FR]; 20, rue Claude Lorrain, F-75016 Paris (FR). **SCHOTT, Olivier** [FR/FR]; 12, quai de la Mégisserie, F-75001 Paris (FR). **STEHLE, Nicolas** [FR/FR]; 98, avenue Philippe Auguste, F-75011 Paris (FR).

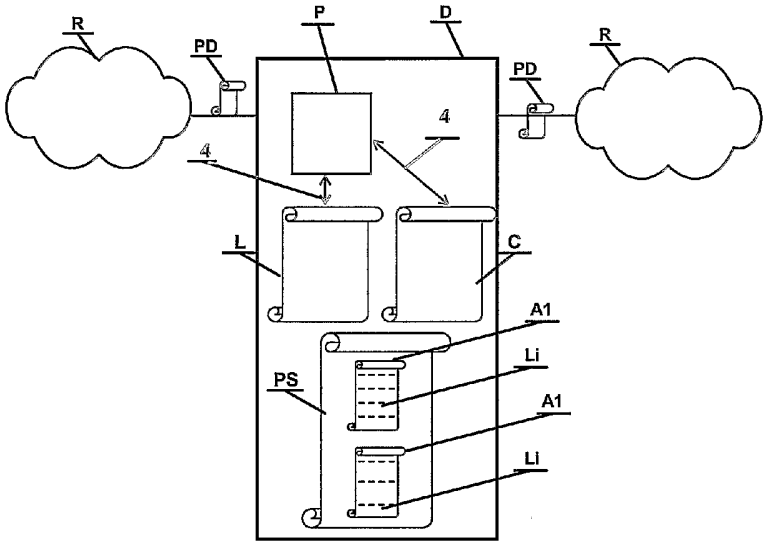
(74) Mandataires : **GRYNWALD, Albert** etc.; Cabinet Grynwald, 127, rue du Faubourg Poissonnière, F-75009 Paris (FR).

(81) États désignés *(sauf indication contraire, pour tout titre de protection nationale disponible)* : AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG,

[Suite sur la page suivante]

(54) Title: DYNAMIC SYSTEM AND METHOD FOR SECURING A COMMUNICATION NETWORK USING PORTABLE AGENTS

(54) Titre : PROCEDE ET SYSTEME DYNAMIQUE DE SECURISATION D'UN RESEAU DE COMMUNICATION AU MOYEN D'AGENTS PORTABLES



(57) Abstract: The invention relates to a device which is placed in a computer network and which is used to secure the communication flows passing therethrough. According to the invention, the communication flows are secured using portable codes, known as portable agents, which can be downloaded from a remote station. The aforementioned portable agents cannot be executed by the device until they have been compiled by a compiler (C) contained in said device, at which point they become executable agents. The compiler translates the portable agents which are written in a language independent of the processor (P) into executable agents which are written in the language of the processor of the device, while carrying out checks on the functions performed by the agent. The executable agents are then executed on the device, according to the communication flows (PD) passing therethrough and a security policy which can also be downloaded from a remote station.

[Suite sur la page suivante]

WO 2004/068817 A2

MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) États désignés *(sauf indication contraire, pour tout titre de protection régionale disponible)* **:** ARIPO (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Déclaration en vertu de la règle 4.17 :

— *relative à la qualité d'inventeur (règle 4.17.iv)) pour US seulement*

Publiée :

— *sans rapport de recherche internationale, sera republiée dès réception de ce rapport*

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(57) Abrégé : L'invention concerne un dispositif placé dans un réseau informatique et servant à sécuriser les flux de communication le traversant. La sécurisation des flux de communication est effectuée au moyen de codes portables, dénommés agents portables, qui peuvent être téléchargés depuis un poste distant. Les agents portables ne sont pas exécutables par le dispositif. Ils le deviennent une fois compilés par un compilateur (C) présent dans le dispositif : ce sont alors des agents exécutables. Le compilateur traduit les agents portables écrits dans un langage indépendant du processeur (P) en des agents exécutables écrits dans le langage du processeur du dispositif, tout en réalisant des contrôles sur les fonctions réalisées par l'agent. Les agents exécutables sont alors exécutés sur le dispositif, en fonction des flux de communication (PD) le traversant et d'une politique de sécurité qui peut également être téléchargée depuis un poste distant.

PROCEDE ET SYSTEME DYNAMIQUE DE SECURISATION D'UN RESEAU DE COMMUNICATION AU MOYEN D'AGENTS PORTABLES

La sécurité du réseau informatique est un élément critique pour une entreprise et passe par la sécurisation des communications et des accès aux éléments du réseau. Avec l'avènement d'Internet et l'opportunité commerciale qu'il représente, de plus en plus d'organisations ont ouvert leur réseau vers l'extérieur. Mais communication réseau et sécurité sont deux concepts fortement incompatibles, et les menaces qui découlent d'un assemblage malheureux de ces deux concepts ont souvent amené les entreprises vers les deux seules possibilités offertes par le marché : pas d'ouverture sur Internet ou un blindage des flux de communications entrant et sortant entraînant des surcoûts énormes. Le marché de la sécurité a donc explosé : les offres de sécurisation des réseaux d'entreprise abondent mais restent axées autour d'une protection frontalière entre deux sous-réseaux (principalement le réseau de l'entreprise et Internet). Bien qu'Internet représente une menace potentielle indéniable, la majeure partie des attaques et des menaces provient de l'intérieur. Malgré ce constat, le marché actuel de la sécurité continue de proposer des solutions qui répondent de moins en moins bien aux besoins des entreprises et, plus généralement, aux besoins des utilisateurs des réseaux.

Les procédés utilisés dans la sécurisation des réseaux informatiques reposent essentiellement sur la technologie de filtrage de paquets. Cette technologie permet d'autoriser le passage des flux de communication réseau tout en exerçant un contrôle sur ces flux. Les meilleures illustrations (et les plus répandues) sont les écluses (communément appelées « Firewall » dans le jargon informaticien : IEEE COMMUNICATIONS MAGAZINE., Vol. 32 N°9 September 1994, page 50 à 57, S.M. BELLOVIN ET AL 'NETWORK FIREWALLS') et les passerelles filtrantes au niveau application (désignées par le nom « Proxy »). Ces deux types d'entités réseau créent une barrière entre deux sous-réseaux et réalisent leur filtrage en fonction de certaines règles de sécurité définies de manière cohérente au sein d'une politique de sécurité. D'autres entités viennent compléter l'offre de sécurité en proposant des services complémentaires : citons entre autres les systèmes de détection d'intrusion (intrusion detection system ou IDS), les antivirus, les passerelles pour les réseaux privés virtuels (appelées passerelles VPN pour Virtual Private Network), les outils logiciels et matériels de chiffrement, les clients/serveurs d'authentification, les serveurs de journalisation, etc. Malgré la variété des produits, de nombreuses limitations deviennent de plus en plus contraignantes pour les entreprises. Leur demande a évolué à l'image du secteur de la sécurité informatique : la protection des réseaux ne doit plus être focalisée sur les points de contact entre plusieurs sous-réseaux mais doit être axée sur la protection de chacun des éléments constituant le réseau.

Dans cette nouvelle optique, tous les services proposés par les différentes offres de sécurité devraient pouvoir s'exercer pour chaque élément du réseau. Les technologies actuelles n'ont pas été pensées en ce sens. Ainsi, deux principaux problèmes freinent le passage vers une sécurité globale et homogène du réseau : la spécialisation des offres de sécurité et le coût d'une telle sécurité. En effet, les entités dédiées à la sécurisation du réseau sont confinées, de par la

technologie employée actuellement, à un rôle défini au préalable : une écluse ne peut être utilisée pour réaliser autre chose que du filtrage ; on ne peut ni changer sa fonction ni lui rajouter de nouvelles fonctionnalités. Dès lors, il est nécessaire de combiner un grand nombre de produits pour atteindre une vaste gamme de services (et donc une bonne sécurité) à un point donné du réseau. Ce grand nombre de produits engendre inévitablement des coûts très importants d'acquisition, de formation et de maintenance, sans atténuer les risques de défaillance dus au fait que ces produits n'ont pas été forcément développés pour travailler ensemble. Le coût de la protection d'un seul point étant déjà relativement élevé, ce coût deviendrait prohibitif dans le cas d'une protection totale du réseau.

Au-delà du coût, la multitude de produits spécialisés complique sévèrement l'administration du réseau et la mise en place d'une politique de sécurité efficace et cohérente. Chaque produit dispose de sa propre interface d'administration et cette pluralité ne permet pas de fournir une vision organisée du réseau. Ces problèmes de clarté et de cohérence entraînent non seulement la présence de failles dans la politique de sécurité mais également un ralentissement dans les temps de réaction d'une entreprise à déployer une politique de sécurité face à des menaces.

Dans la grande majorité des solutions actuelles pour la sécurisation des réseaux, l'élément central est l'écluse. La politique de sécurité de l'entreprise est centrée sur cette écluse autour de laquelle peuvent graviter d'autres entités fournissant des services de sécurité complémentaires. Un serveur d'administration permet de définir les différents éléments du réseau (ordinateurs, périphériques réseau, services réseau, utilisateurs...) et de définir des règles de filtrage entre ces différents éléments. Ces règles de filtrage constituent la politique de sécurité qui sera ensuite envoyée à l'écluse : l'autorisation ou le refus de passage des paquets des flux de communication sont alors réalisés par l'écluse conformément aux

règles de filtrage. Historiquement, l'envoi de la politique de sécurité a constitué une évolution des écluses permettant de palier à la rigidité et au manque d'évolutivité d'une configuration écrite directement dans l'écluse. De plus, les règles de filtrage ont évolué pour permettre de filtrer de nouveaux protocoles en proposant de définir ses propres éléments du réseau dans le serveur d'administration. Toutes ces nouveautés ont fait avancer la technologie des écluses jusqu'au stade dit de « firewall de troisième génération ». Néanmoins, les besoins naissant en matière de sécurité réseau nécessitent une nouvelle évolution que ne pourraient apporter des écluses, quelle que soit leur génération. Cette évolution se définit par la possibilité de réaliser n'importe quel type de traitement sur les paquets par la même entité réseau, que ce soit du filtrage de type écluse, de la détection d'intrusion, de la détection de virus, de la qualité de service réseau, etc. En effet, il devient primordial d'analyser et de contrôler les informations transitant dans les flux autorisés par l'écluse, car ces flux peuvent être utilisés à des fins de piratage informatique. Dans le modèle en couche de la norme OSI (ISO/IEC 7498-1 :1994) - dont l'une des implémentations est le protocole TCP/IP (Internet Protocol : RFC 791, Transmission Control Protocol : RFC 793), le filtrage des écluses est réalisé au niveau des couches réseau et transport. La couche la plus haute est la couche application et contient les informations transmises par les applications client/serveur. Il existe un très grand nombre de protocoles dans la couche application, ce qui représente autant de flux d'informations susceptibles de contenir une attaque. Chaque jour, de nouvelles failles sont décelées dans tel ou tel protocole et permettent à des pirates informatiques d'attaquer le système hébergeant un service utilisant ce protocole. Un produit de sécurité devrait donc être en mesure de récupérer de nouveaux services pour rester à jour face aux menaces. Autour d'une même politique de sécurité, l'administrateur devrait pouvoir définir les services qu'il veut réaliser sur chacun des

points du réseau, en fonction des utilisateurs du réseau et en fonction des menaces du moment.

Une des méthodes pour y parvenir est de disposer de codes mobiles. La théorie des codes mobiles repose sur la présence d'un module capable d'exécuter du code qui lui est fourni à distance. Ceci permet de conserver une plate-forme homogène capable d'exécuter n'importe quel type de programme. L'une des implémentations du concept de codes mobiles repose sur la présence d'une machine virtuelle. Une machine virtuelle émule un processeur (c'est-à-dire qu'elle simule, sur un autre matériel, le fonctionnement de ce processeur). C'est un processeur virtuel avec un langage qui lui est propre. A ce titre, il dispose de registres de travail et exécute les séquences d'instructions d'un code compilé dans son propre langage. Ceci est réalisé non pas de manière matérielle mais de manière logicielle.

La machine virtuelle apporte l'évolutivité à un système, en permettant l'apport de nouvelles fonctionnalités. Elle apporte également l'indépendance par rapport au système et donc la portabilité.

Certaines des écluses les plus évoluées intègrent une machine virtuelle afin de réaliser le filtrage de paquets par un code mobile généré à partir de la politique de sécurité définie dans le serveur d'administration, comme présenté dans le brevet US 5606668 ou le brevet US 5835726. Cette méthode d'utilisation de la machine virtuelle, bien que très utile, continue de confiner le rôle de l'écluse à un rôle de filtrage simpliste car la machine virtuelle est cantonnée à l'autorisation ou au refus de passage des paquets en fonction de règles de sécurité.

Bien qu'apportant quelques avantages, la machine virtuelle présente un défaut majeur : la baisse sensible de performance. En effet, la machine virtuelle émule un processeur au-dessus d'un vrai processeur, induisant par-là même une surcouche : les codes mobiles (appelés aussi applets dans le jargon informaticien) sont exécutés par la machine virtuelle qui

est elle-même exécutée par le processeur. Cette couche d'abstraction logicielle provoque une baisse de performance qui peut être critique dans des applications temps réel de type traitement de flux réseau.

Une autre implémentation possible du concept de codes mobiles consiste à envoyer directement du code natif (que nous appellerons, par convention, agent) à un dispositif. Cet agent est un code compilé dans le langage du processeur. Cette solution est optimale en terme de vitesse d'exécution. Les agents étant directement exécutés par le processeur, ils peuvent être optimisés compte tenu des caractéristiques particulières de ce processeur. Leur compilation se fait dans une phase antérieure (généralement au moment du développement de l'agent). Le compilateur traduit alors le code de l'agent : à partir d'un code développé dans un langage de haut niveau (c'est-à-dire compréhensible facilement par l'homme par ses similitudes avec un langage naturel), le compilateur génère une traduction du code dans un langage de bas niveau (compris par la machine). La compilation d'un code comprend différentes étapes au travers desquelles le code subit plusieurs transformations. « La compilation est effectuée par un compilateur. Selon une définition simplifiée, un compilateur est un programme qui lit un programme écrit dans un premier langage - le langage source - et le traduit en un programme équivalent écrit dans un autre langage - le langage cible. » (Compilateurs, principes, techniques et outils - Alfred Aho/Ravi Sethi/Jeffrey Ullman - éditions InterEditions - 1989). Un compilateur opère selon différentes phases transformant le programme source d'une représentation à une autre. La première phase est l'analyse lexicale qui groupe des caractères d'un programme source en unités lexicales (mots ou symboles). Suit l'analyse syntaxique (dite également grammaticale) qui groupe les unités lexicales en structure syntaxique qui seront utilisées par le compilateur pour synthétiser son résultat. La phase suivante, l'analyse sémantique, utilise la structure syntaxique pour contrôler si le

programme source contient des erreurs sémantiques (par exemple : un nombre réel est utilisé comme un caractère). Le compilateur construit alors une représentation intermédiaire du programme source qui est simultanément facile à produire et facile à traduire en langage cible. Une phase d'optimisation du code tente ensuite d'améliorer le code intermédiaire afin que le code résultant s'exécute plus rapidement. La phase finale du compilateur consiste en la production d'un code cible. La création d'un programme exécutable requiert en général l'utilisation de plusieurs autres programmes qui sont des cousins du compilateur. En effet, le programmeur crée en général un squelette de programme, qui est modifié par un préprocesseur afin d'obtenir un programme source. Ce dernier est compilé par le compilateur en programme cible généralement en langage assembleur. Celui-ci est transformé par un assembleur en code machine translatable qui est lui-même complété par un relieur-chargeur avec des bibliothèques ou des fichiers objets translatables afin d'obtenir un code machine absolu compréhensible par l'ordinateur. Ainsi, de manière simplifiée, différentes phases constituent la compilation: dans une première étape, les différents fichiers composant le code sont compilés individuellement en langage d'assemblage (phase de compilation regroupant les nombreuses étapes d'analyse d'un code source en langage de haut niveau), puis ils sont translatés du langage d'assemblage (langage de bas niveau) au langage machine, c'est-à-dire en langage binaire (phase d'assemblage). On dispose alors de fichiers objets qui sont la traduction en langage machine des fichiers sources. La dernière phase génère l'exécutable proprement dit : les fichiers sont liés les uns aux autres afin de former un seul fichier binaire (phase dite d'édition de liens). Le compilateur doit résoudre toutes les dépendances de chacun des fichiers objets afin de former un exécutable cohérent. L'inconvénient majeur de cette méthode est qu'elle est incompatible avec une indépendance de la plate-forme et un langage propriétaire optimisé pour les besoins de ladite plate-

forme. En effet, les codes compilés ne sont pas du tout portables car ils dépendent du processeur du dispositif. Seuls les fichiers sources sont portables. La solution de distribuer les fichiers sources pose de nombreux problèmes : le code est lisible et modifiable par tout le monde, ce qui peut être un problème pour une entreprise désireuse de conserver une expertise, un savoir-faire ou simplement des algorithmes confidentiels. En outre, les fichiers sources demandent d'être compilés pour le processeur adéquat. Il paraît peu probable qu'un client s'étant procuré différents dispositifs (avec des processeurs différents) serait prêt à effectuer les compilations des codes sources avec, à chaque fois, le bon compilateur, en vue de disposer de différentes versions binaires d'un même code source, organiser l'envoi du bon code compilé aux divers dispositifs. En outre, le fait de pouvoir envoyer au dispositif des codes compilés dans le langage de son processeur peut s'avérer très dangereux. En effet, on autorise alors n'importe quel utilisateur, y compris un utilisateur mal intentionné, à développer un code permettant de prendre entièrement le contrôle sur le dispositif. Pour limiter les possibilités des agents, il est nécessaire d'effectuer des contrôles lors de leur exécution, ce qui affecterait très sensiblement les performances.

L'invention, objet du présent brevet, permet de résoudre les problèmes évoqués précédemment sans présenter les inconvénients de l'art antérieur. L'invention permet de conserver les avantages des codes mobiles tout en accroissant les performances. Elle pallie les problèmes et les limitations des technologies existantes en proposant une solution innovante.

Description de l'invention

Le cadre général de l'invention concerne une méthode de sécurisation des réseaux informatiques par le contrôle des flux de communication entre éléments desdits réseaux. Ce contrôle s'effectue grâce à la réalisation de traitements sur les paquets des flux de communication par un procédé flexible,

dynamique, évolutif, qui peut être administré de manière simple et déployé de façon homogène sur l'ensemble du réseau.

La présente invention décrit un procédé de traitements évolutifs des flux de communication réseau, ces traitements étant effectués en temps réel.

En outre, la présente invention permet de réaliser tout type de traitements évolués des paquets à tous les niveaux du modèle OSI et en particulier au niveau de la couche application.

En outre, la présente invention rend le système évolutif en terme de nouvelles fonctionnalités pour un type de traitement donné (il est possible, par exemple, de rajouter facilement de nouvelles fonctionnalités de filtrage à une écluse (firewall) ou de nouvelles signatures virales à un antivirus).

En outre, la présente invention permet à un système de changer de type de traitement en temps réel (une écluse peut devenir un antivirus ou un système de détection d'intrusion ou encore une passerelle VPN).

En outre, la présente invention permet à un système de réaliser tous les changements évoqués précédemment, et ce, de manière évolutive et en temps réel.

En outre, la présente invention permet d'apporter une protection efficace et personnalisable en tout point du réseau, et ce de manière homogène.

En outre, la présente invention apporte des solutions en terme de performance et de rapidité d'exécution permettant à un système embarqué de traiter efficacement des flux de communication en temps réel.

En outre, le système peut se protéger, sans impacter les performances, des codes qui lui sont envoyés pour réaliser de nouveaux traitements.

La présente invention concerne un procédé permettant de réaliser l'analyse et/ou la modification sélective et/ou le filtrage sélectif de paquets de données traversant un dispositif placé en coupure dans un réseau informatique ; ledit dispositif

comprenant un processeur exécutant un compilateur et un logiciel conformément à une politique de sécurité ; ledit logiciel étant destiné au filtrage desdits paquets de données, en autorisant ou non leur passage, conformément à ladite politique de sécurité; ledit procédé étant caractérisé en ce qu'il comprend les étapes suivantes :

- l'étape de définir ladite politique de sécurité au moyen d'agents portables, écrits dans un langage informatique indépendant du langage dudit processeur et dédiés à l'analyse et/ou à la modification sélective et/ou au filtrage sélectif desdits paquets de données ;

- l'étape, pour ledit logiciel, d'appeler automatiquement ledit compilateur afin d'effectuer une compilation pour traduire lesdits agents portables en des agents exécutables écrits dans le langage dudit processeur ;

- l'étape d'exécuter ledit logiciel pour filtrer lesdits paquets de données traversant ledit dispositif, en autorisant ou non leur passage, conformément à ladite politique de sécurité ;

- l'étape d'analyser, lesdits paquets de données autorisés par ledit logiciel à traverser ledit dispositif, en exécutant lesdits agents exécutables par ledit processeur ; et/ou

- l'étape de modifier sélectivement, lesdits paquets de données autorisés par ledit logiciel à traverser ledit dispositif, en exécutant lesdits agents exécutables par ledit processeur ; et/ou

- l'étape de filtrer sélectivement, lesdits paquets de données autorisés par ledit logiciel à traverser ledit dispositif, en exécutant lesdits agents exécutables par ledit processeur.

La présente invention se caractérise donc par un dispositif se connectant au réseau. La connexion au réseau induit une coupure du réseau en deux sous-réseaux, ce qui permet

d'intercepter tous les flux de communication d'un des sous-réseaux à destination de l'autre.

Ce procédé permet à un dispositif réseau de recevoir une politique de sécurité composée de règles de filtrage classiques et également d'agents de traitement des paquets. Ces agents sont compilés—automatiquement dans le dispositif, la compilation étant lancée par le logiciel embarqué dédié au filtrage des paquets de données: ils deviennent alors directement exécutables par le processeur, ce qui est optimal en terme de vitesse d'exécution. Ainsi, le procédé permet à un dispositif de modifier son propre comportement en fonction des agents téléchargés, ce qui le rend totalement évolutif. En effet, cette modification de comportement peut être une modification globale du rôle du dispositif (une écluse devient un antivirus, par exemple) ou une simple mise à jour des fonctionnalités (un rajout de nouvelles détections de signatures par exemple). De plus, les agents sont envoyés dans un langage indépendant du processeur du dispositif. Cette indépendance assure leur portabilité sur des dispositifs utilisant des processeurs différents. De plus, cela permet éventuellement de concevoir un langage propriétaire, intermédiaire entre un langage de haut niveau et le langage natif du processeur, ce langage propriétaire pouvant avoir des fonctionnalités adaptées aux besoins en matière d'analyse, de modification et de filtrage de paquets dans des flux de communication réseau et pouvant être restreint à des fonctions sans danger pour le dispositif. Ainsi, les agents sont inintelligibles, ce qui protège la propriété intellectuelle de l'auteur. En fonctionnement, le dispositif intercepte tous les paquets le traversant et le logiciel embarqué réalise un filtrage préalable des paquets de données, conformément à une politique de sécurité. Pour les paquets autorisés par le logiciel embarqué, conformément à la politique de sécurité, des agents seront exécutés pour réaliser des traitements complémentaires. Cela permet d'optimiser les

performances du dispositif en faisant un premier filtrage des paquets avant d'exécuter les agents.

Avantageusement, la politique de sécurité comprend en outre une définition des différents objets dudit réseau informatique.

Avantageusement, la politique de sécurité comprend en outre une définition des différents services dudit réseau informatique.

Avantageusement, la politique de sécurité comprend en outre une définition des différents utilisateurs dudit réseau informatique.

Avantageusement, le procédé selon l'invention comprend l'étape de générer des paramètres de configuration permettant de configurer lesdits agents portables en fonction desdits utilisateurs dudit réseau informatique.

Avantageusement, la politique de sécurité comprend en outre une définition dudit dispositif.

Ceci permet à la politique de sécurité d'inclure de multiples paramètres représentant divers aspects du réseau. Il est donc possible de définir des règles de filtrage entre éléments du réseau ou entre des utilisateurs et des services ou encore entre le dispositif et les services réseaux. A tous ces types de règles de filtrage, il est possible de rajouter des agents qui vont réaliser des traitements supplémentaires. Le logiciel embarqué dans le dispositif réalise alors le filtrage en fonction des règles de la politique de sécurité et, pour les paquets autorisés par ces règles, lance l'exécution des agents qui ont été rajoutés pour ses règles.

Ainsi le dispositif n'est pas limité au travail d'une écluse (filtrage de paquets). En effet, il est possible, au niveau des règles de filtrage du logiciel embarqué, d'autoriser tous les flux de paquets à traverser le dispositif (ce qui a pour effet de désactiver la fonctionnalité d'écluse), tout en rajoutant des agents dédiés, par exemple, au filtrage des tentatives d'intrusion.

Avantageusement, ledit langage informatique desdits agents portables est un langage de bas niveau dédié à des traitements sur lesdits paquets de données dudit réseau informatique et permettant de contrôler et de limiter les actions possibles desdits agents portables au sein dudit dispositif.

Ainsi, les agents ne peuvent être lus car ils sont inintelligibles pour l'homme. De plus, ils peuvent préalablement être développés dans un langage de haut niveau au moment de leur conception, puis compilés et fournis par la suite dans ce langage de bas niveau. Le fournisseur des agents conserve ainsi les sources de ses agents. Le langage dans lequel sont écrits les agents est spécialement adapté aux traitements des flux de communication réseau et permet de garder un contrôle sur les possibilités de l'agent au sein du dispositif. En effet, un agent, directement compilé dans le langage du processeur du dispositif, pourrait potentiellement effectuer des dommages graves au dispositif s'il n'y a pas de contrôle lors de son exécution. Un contrôle de l'agent pendant son exécution affecterait très nettement ses performances. En limitant les possibilités de l'agent dans le langage dans lequel il est écrit et dans le compilateur de ce langage, les agents sont contrôlés lors de la compilation et non lors de l'exécution ce qui augmente les performances. En outre, il devient possible de concevoir une version améliorée de l'invention en optimisant le compilateur embarqué : le compilateur n'a besoin ici de n'effectuer qu'une translation d'un langage de bas niveau vers le langage du processeur, ce qui est beaucoup plus rapide qu'une compilation complète. Cela facilite l'implémentation du compilateur au sein de nouveaux dispositifs avec des processeurs différents, tout en conservant l'ensemble des avantages de portabilité, confidentialité et sécurité du dispositif. En effet, les phases d'analyse lexicale, syntaxique et sémantique, propres à la compilation d'un code source de haut niveau, n'ont plus besoin d'être réalisées.

Avantageusement, le procédé selon l'invention comprend l'étape de définir, sur un serveur distant dudit dispositif, ladite politique de sécurité.

Avantageusement, le procédé selon l'invention comprend l'étape de définir, sur ledit dispositif, ladite politique de sécurité.

La politique de sécurité peut être configurée à distance et envoyée au dispositif via le réseau. Elle peut également être définie directement sur le dispositif avec, par exemple, un serveur web embarqué dans le dispositif ou via un port série du dispositif.

Avantageusement, le procédé selon l'invention comprend l'étape d'authentifier le ou les utilisateurs, non authentifiés, dudit dispositif.

Avantageusement, ladite politique de sécurité comprend en outre une définition desdits utilisateurs authentifiés dudit dispositif.

Avantageusement, le procédé selon l'invention comprend l'étape d'authentifier ledit ou lesdits utilisateurs, non authentifiés, dudit dispositif à l'aide d'un moyen d'identification associé audit dispositif.

Avantageusement, le procédé selon l'invention comprend l'étape d'authentifier ledit ou lesdits utilisateurs, non authentifiés, dudit dispositif à l'aide d'une application client/serveur dont l'application serveur est contenue dans ledit dispositif.

Il devient donc possible de définir une politique de sécurité en fonction des utilisateurs du dispositif. Le procédé permet ainsi de définir une politique de sécurité et des agents propres aux utilisateurs du dispositif : sur le même dispositif, des utilisateurs différents se verront attribuer des politiques de sécurité différentes. A titre d'exemple purement illustratif et non limitatif des possibilités d'application de l'invention, on peut mettre en place une politique de sécurité dans laquelle un stagiaire, après s'être authentifié, n'aura accès qu'aux

services réseaux et aux serveurs non confidentiels, alors qu'un développeur pourra accéder aux serveurs de développement.

Les méthodes d'authentification des utilisateurs du dispositif peuvent être de plusieurs sortes : au moyen d'un élément du dispositif (à titre d'exemple purement illustratif et non limitatif des possibilités d'application de l'invention, on peut citer, entre autres, un lecteur de carte à puces ou un identificateur biométrique) ou par un mécanisme de type client / serveur dans lequel le serveur d'authentification résiderait dans le dispositif. Les informations d'authentification peuvent alors être contrôlées dans le dispositif ou sur un serveur distant dans lequel est stockée la politique de sécurité.

Avantageusement, le procédé selon l'invention comprend l'étape d'exécuter des fonctions d'une bibliothèque de fonctions contenue dans ledit logiciel et appelée par lesdits agents exécutables.

Ceci permet de mettre à disposition des agents exécutables un ensemble de fonctions répondant aux besoins et aux spécificités du dispositif.

Avantageusement, le procédé selon l'invention comprend l'étape d'exécuter des fonctions, de ladite bibliothèque de fonctions, spécialisées dans une gestion d'un cache desdits paquets de données.

Avantageusement, la gestion dudit cache desdits paquets de données comprend les étapes suivantes :

- l'étape, après exécution desdits agents exécutables, de mémoriser, dans ledit cache, des informations de paquets concernant lesdits paquets de données et en outre lesdits paquets de données eux-mêmes lorsqu'ils ont été modifiés lors de ladite exécution ;

- l'étape, lors de l'arrivée d'un paquet entrant dans ledit dispositif, de vérifier, grâce audites informations de paquets mémorisées dans ledit cache, si ledit paquet entrant est un paquet déjà reçu ;

- l'étape, lorsque ledit paquet entrant n'est pas un paquet déjà reçu, d'exécuter lesdits agents exécutables;

- l'étape, lorsque ledit paquet entrant est un paquet déjà reçu, de déterminer, grâce audites informations de paquets mémorisées dans ledit cache, si ledit paquet déjà reçu avait été modifié par lesdits agents exécutables ;

- l'étape, lorsque ledit paquet déjà reçu avait été modifié par lesdits agents exécutables, de transmettre vers ledit réseau informatique, sans exécuter lesdits agents exécutables, une version dudit paquet déjà reçu mémorisée dans ledit cache ;

- l'étape, lorsque ledit paquet déjà reçu n'avait pas été modifié par lesdits agents exécutables, de transmettre vers ledit réseau informatique ledit paquet entrant tel quel, sans exécuter lesdits agents exécutables.

Cet ensemble de fonctions permet aux agents de disposer d'une gestion adaptée de cache de paquets. Le cache de paquets permet aux agents de ne pas voir les paquets de données déjà reçus afin de conserver la vision d'un flux cohérent. De plus, le cache de paquets de données permet d'améliorer sensiblement les performances du dispositif en court-circuitant l'exécution des agents et en envoyant directement le paquet déjà reçu - dans le cas où il n'aurait pas été modifié par les agents lors de sa première réception - ou sa version modifiée, stockée dans le cache de paquets de données, - dans le cas où il aurait été modifié par les agents lors de sa première réception.

Avantageusement, le procédé selon l'invention comprend l'étape d'exécuter des fonctions, de ladite bibliothèque de fonctions, spécialisées dans une gestion des couches réseau et transport du protocole de communication utilisé.

Avantageusement, la gestion desdites couches réseau et transport comprend les étapes suivantes :

- l'étape de mémoriser des informations de protocole desdites couches réseau et transport desdits paquets de données

traversant ledit dispositif afin de réaliser un suivi des différents flux desdits paquets de données ;

- l'étape de mémoriser des modifications desdits paquets de données réalisées par lesdits agents exécutables ;

- l'étape de mettre à jour lesdites informations de protocole desdites couches réseau et transport desdits paquets de données traversant ledit dispositif, en fonction desdites informations de protocole et desdites modifications mémorisées, sur lesdits paquets de données afin de conserver une cohérence des flux desdits paquets de données.

Le procédé permet de conserver les informations importantes des flux autorisés afin de pouvoir modifier et analyser correctement les informations des paquets de données en cours de traitement. A titre d'exemple purement illustratif et non limitatif des possibilités d'application de l'invention, les informations conservées peuvent être les numéros de séquence et d'acquittement du protocole TCP (tels que définis dans la RFC 793 déjà citée) ce qui permet d'agrandir ou de réduire les paquets de données, de recalculer les sommes de contrôle des en-têtes, de conserver des informations passées dans le flux tel qu'un nom d'utilisateur, un mot clé important, l'appel d'une commande spéciale, etc.

Avantageusement, le procédé selon l'invention comprend l'étape d'exécuter des fonctions, de ladite bibliothèque de fonctions, spécialisées dans une recherche de motifs et d'expressions régulières.

Par le biais de ces fonctions, les agents peuvent alors réaliser des recherches complexes de motifs dans les paquets, ce que nécessite souvent l'analyse de paquets de données : à titre d'exemple purement illustratif et non limitatif des possibilités d'application de l'invention, ces fonctions peuvent être, entre autres, des fonctions de comparaisons de chaînes, de bloc mémoire, d'expressions régulières, des fonctions de recherche simultanée de plusieurs chaînes dans un bloc mémoire, etc.

Avantageusement, le procédé selon l'invention comprend l'étape d'exécuter des fonctions, de ladite bibliothèque de fonctions, spécialisées dans une communication entre lesdits agents exécutables.

Dans de nombreux cas, un agent aura besoin d'échanger des informations avec les autres agents afin de les avertir ou d'être prévenu d'évènements imminents : un exemple purement illustratif et non limitatif des possibilités d'application de l'invention est celui d'un agent ayant détecté la présence d'un virus et décidant d'interdire le passage du paquet. Il doit alors avertir les autres agents que le paquet a été détruit.

Avantageusement, le procédé selon l'invention comprend l'étape d'exécuter des fonctions, de ladite bibliothèque de fonctions, spécialisées dans une communication entre lesdits agents exécutables et desdits objets dudit réseau informatique.

Le procédé permet de donner la possibilité aux agents de dialoguer avec des composants réseaux dans leur protocole de communication. Ceci permet, entre autres, de reconfigurer des périphériques ou d'échanger des informations. En effet, un réseau efficacement protégé est un réseau où chaque élément a un rôle cohérent dans la politique de sécurité. Il est important que chaque composant du réseau puisse participer à la sécurité du réseau. Un exemple purement illustratif et non limitatif des possibilités d'application de l'invention est celui d'un agent utilisant les fonctions de la bibliothèque pour reconfigurer la politique de sécurité d'un routeur via le protocole SNMP (Simple Network Management Protocol : RFC 1157) ou pour envoyer des logs (messages d'information) à des serveurs de logs déjà existants (comme syslog par exemple : RFC 3164).

Avantageusement, le procédé selon l'invention comprend l'étape d'associer des composants matériels spécialisés dudit dispositif à des fonctions de ladite bibliothèque de fonctions afin d'accélérer l'exécution desdites fonctions.

Afin d'optimiser les performances du dispositif, les fonctions les plus utilisées de la bibliothèque de fonctions peuvent être directement intégrées au dispositif au niveau matériel : par exemple, des algorithmes de chiffrement ou de recherches de motifs peuvent être câblés dans un coprocesseur dédié. L'accélération matérielle permet d'obtenir un gain de performance non négligeable pour des dispositifs de traitements en temps réel.

Avantageusement, le procédé selon l'invention comprend l'étape de modifier ladite politique de sécurité en exécutant lesdits agents exécutables par ledit processeur.

Pour obtenir une sécurité globale et cohérente du dispositif et du réseau en général, les agents doivent pouvoir influer sur la politique de sécurité en cours. En effet, les agents peuvent réaliser des analyses très poussées sur les paquets, entre autres, en vue de détecter des attaques réseaux, des intrusions, des comportements anormaux, des virus, des dépassements de quota, des motifs non autorisés à transiter sur le réseau. Toutes ces analyses amènent les agents à prendre des décisions de modification de la politique de sécurité. Un exemple purement illustratif et non limitatif des possibilités d'application de l'invention est celui d'un agent chargé de détecter la négociation du port du canal de données du protocole FTP (File Transfer Protocol : RFC 959) et devant décider d'autoriser ou non les paquets du canal de données à traverser le dispositif. Un autre exemple est celui d'un agent, détectant une tentative d'attaque depuis un poste A, et rajoutant alors une règle de filtrage interdisant toute communication avec le poste A.

L'invention concerne également un système permettant de réaliser l'analyse et/ou la modification sélective et/ou le filtrage sélectif de paquets de données; ledit système comprenant :

un dispositif traversé par lesdits paquets de données et placé en coupure dans un réseau informatique, ledit

dispositif comprenant un processeur exécutant un compilateur et un logiciel conformément à une politique de sécurité ; ledit logiciel comprenant des moyens de filtrage pour filtrer lesdits paquets de données traversant ledit dispositif, en autorisant ou non leur passage, conformément à ladite politique de sécurité (PS) ; et ;

des agents portables, destinés à définir ladite politique de sécurité, écrits dans un langage informatique indépendant du langage dudit processeur et dédiés à l'analyse et/ou la modification sélective et/ou le filtrage sélectif desdits paquets de données ;

ledit compilateur étant automatiquement activé par ledit logiciel pour traduire lesdits agents portables en des agents exécutables écrits dans le langage dudit processeur ;

lesdits agents exécutables étant exécutés par ledit processeur pour :

analyser lesdits paquets de données autorisés par ledit logiciel à traverser ledit dispositif, et/ou

modifier sélectivement lesdits paquets de données autorisés par ledit logiciel à traverser ledit dispositif, et/ou

filtrer sélectivement lesdits paquets de données autorisés par ledit logiciel à traverser ledit dispositif.

Avantageusement, ladite politique de sécurité comprend en outre une définition des différents objets dudit réseau informatique.

Avantageusement, ladite politique de sécurité comprend en outre une définition des différents services dudit réseau informatique.

Avantageusement, ladite politique de sécurité comprend en outre une définition des différents utilisateurs dudit réseau informatique.

Avantageusement, ledit système comprend en outre des moyens de génération de paramètres de configuration pour configurer lesdits agents portables, en fonction desdits utilisateurs dudit réseau informatique.

Avantageusement, ladite politique de sécurité comprend en outre une définition dudit dispositif.

Avantageusement, ledit langage informatique est un langage de bas niveau dédié à des traitements sur lesdits paquets de données dudit réseau informatique et permettant de contrôler et de limiter les actions possibles desdits agents portables au sein dudit dispositif.

Avantageusement, ledit système comprend un serveur distant dudit dispositif pour définir ladite politique de sécurité.

Avantageusement, ledit dispositif comprend des moyens d'administration pour définir ladite politique de sécurité.

Avantageusement, ledit système comprend des moyens d'authentification du ou des utilisateurs, non authentifiés, dudit dispositif.

Avantageusement, ladite politique de sécurité comprend en outre une définition desdits utilisateurs authentifiés dudit dispositif.

Avantageusement, ledit dispositif comprend un moyen d'identification pour authentifier ledit ou lesdits utilisateurs, non authentifiés, dudit dispositif.

Avantageusement, ledit dispositif comprend une application serveur d'une application client/serveur destinée à authentifier ledit ou lesdits utilisateurs, non authentifiés, dudit dispositif.

Avantageusement, ledit logiciel comprend une bibliothèque de fonctions dont les fonctions sont appelées par lesdits agents exécutables.

Avantageusement, ladite bibliothèque de fonctions comprend en outre des fonctions spécialisées dans une gestion d'un cache desdits paquets de données.

Avantageusement, ledit cache desdits paquets de données comprend :

une mémoire pour stocker, après exécution desdits agents exécutables, des informations de paquets concernant

lesdits paquets de données et pour stocker lesdits paquets de données eux-mêmes ;

des moyens de contrôle pour vérifier, grâce audites informations de paquets mémorisées dans ledit cache, si un paquet entrant est un paquet déjà reçu et s'il avait été modifié par lesdits agents exécutables ;

des moyens d'activation pour activer, en fonction des vérifications opérées par les moyens de contrôles,

soit des moyens de transmission pour transmettre vers ledit réseau informatique sans modification un paquet de données stocké dans ladite mémoire ;

soit des moyens de transmission pour transmettre vers ledit réseau informatique sans modification un paquet entrant.

Avantageusement, ladite bibliothèque de fonctions comprend en outre des fonctions spécialisées dans une gestion des couches réseau et transport du protocole de communication utilisé.

Avantageusement, ledit dispositif comprend :

au moins une mémoire pour stocker des informations de protocole desdites couches réseau et transport desdits paquets de données traversant ledit dispositif afin de réaliser un suivi des différents flux desdits paquets de données, et pour stocker des modifications desdits paquets de données réalisées par lesdits agents exécutables ;

des moyens de mise à jour desdites informations de protocole desdites couches réseau et transport desdits paquets de données traversant ledit dispositif, en fonction desdites informations de protocole et desdites modifications mémorisées, sur lesdits paquets de données afin de conserver une cohérence des flux desdits paquets de données.

Avantageusement, ladite bibliothèque de fonctions comprend en outre des fonctions spécialisées dans une recherche de motifs et d'expressions régulières.

Avantageusement, ladite bibliothèque de fonctions comprend en outre des fonctions spécialisées dans une communication entre lesdits agents exécutables.

Avantageusement, ladite bibliothèque de fonctions comprend des fonctions spécialisées dans une communication entre lesdits agents exécutables et desdits objets dudit réseau informatique.

Avantageusement, ledit dispositif comprend des composants matériels spécialisés associés à des fonctions de ladite bibliothèque de fonctions afin d'accélérer l'exécution desdites fonctions.

Avantageusement, lesdits agents exécutables, exécutés par ledit processeur, modifient ladite politique de sécurité.

Le système, objet de la présente invention, permet ainsi de réaliser parfaitement toutes les fonctionnalités du procédé décrit précédemment.

Afin de mieux faire comprendre l'invention, différents exemples vont être décrits à l'aide de figures. Ces exemples donnent, à titre purement illustratif, des modes de réalisation possibles, modes auxquels ne se limite pas l'invention.

La figure 1 représente le schéma général de l'interconnexion du dispositif agissant dans l'invention avec un réseau informatique.

La figure 2 illustre l'effet de la compilation des agents au sein du dispositif.

La figure 3 représente le schéma général de l'interconnexion du dispositif agissant dans l'invention avec un réseau informatique après la compilation des agents portables en agents exécutables.

La figure 4 représente l'automate de traitement des paquets et d'exécution des agents dans le dispositif.

La figure 5 représente le schéma général du réseau informatique associé à une politique de sécurité.

La figure 6 présente l'automate d'un agent susceptible de modifier la politique de sécurité.

La figure 7 présente une procédure d'authentification d'un utilisateur du dispositif avec un serveur distant.

La figure 8 présente une procédure d'authentification d'un utilisateur du dispositif avec une application serveur dans le dispositif.

La figure 9 représente l'automate de traitement des paquets d'un agent.

La figure 10 représente un autre mode d'interconnexion du dispositif à un réseau informatique.

La figure 11 représente l'automate de cache de paquets.

La figure 12 présente un exemple de communication entre un agent et différents éléments du réseau.

La figure 13 illustre la façon dont des composants matériels spécialisés peuvent réaliser certaines fonctions de la bibliothèque de fonctions.

La figure 14 décrit une décomposition typique d'un compilateur.

Sur la figure 1, le dispositif D contient un processeur P. Le dispositif D est placé en coupure d'un réseau informatique quelconque : il peut aussi bien s'agir d'un intranet d'entreprise, du réseau Internet, de deux sous-réseaux adjacents ou bien simplement de deux postes. Il peut également s'agir d'un ordinateur connecté à un réseau. On entend par coupure la séparation physique du réseau R en deux sous-réseaux reliés entre eux à l'aide du dispositif D. Ainsi tout flux de communication composé de paquets de données PD d'un des sous-réseau à destination de l'autre sous-réseau doit traverser le dispositif D. Ceci assure le contrôle de tout flux de données et permet de fournir des services de sécurité et de filtrage au niveau du dispositif D. Le dispositif D comprend en outre un logiciel L et un compilateur C qui sont destinés à être exécutés 4 par le processeur P. Le dispositif D contient également une politique de sécurité PS. Cette politique de sécurité PS est

définie au moyen d'agents portables A1 écrits dans un langage informatique Li indépendant du langage du processeur P.

La phase de compilation des agents est présentée en figure 2. Dès que la politique de sécurité PS est présente dans le dispositif D, le logiciel L appelle automatiquement le compilateur C afin d'effectuer la compilation des agents portables A1 présents dans la politique de sécurité PS et écrits dans ledit langage informatique Li indépendant du langage du processeur P pour les traduire en agents exécutables A2 écrits dans le langage du processeur P (langage représenté par LP). Les agents portables A1 ne sont pas exécutables par le processeur P mais ils le deviennent après compilation sous leur forme d'agents exécutables A2. Les agents exécutables A2 remplacent les agents portables A1 dans la définition de la politique de sécurité PS.

La figure 3 illustre l'état du dispositif présenté dans la figure 1 après la compilation, montrée en figure 2, des agents portables A1 en agents exécutables A2. Les différences par rapport à la figure 1 sont les suivantes :

Les agents portables A1 définissant la politique de sécurité PS ont été remplacés par les agents exécutables A2 écrits dans le langage du processeur P (langage représenté par LP) et qui sont leur version compilée.

Les agents exécutables A2 sont alors exécutés 4 par le processeur P au même titre que le logiciel L et le compilateur C.

Les agents exécutables A2 sont donc au même niveau que le logiciel L et sont exécutés 4 par le processeur P. Contrairement à des codes mobiles (ou applets dans le jargon informatique), il n'y a pas de couche d'abstraction logicielle (comme une machine virtuelle). L'agent exécutable A2 apporte une nouvelle fonctionnalité au dispositif D, tout se passant comme si cette fonctionnalité était déjà présente dans le logiciel L.

Les agents portables A1 peuvent être développés dans un langage de haut niveau (comme le langage « C » défini par la

norme ISO/IEC 9899:1999) ou intermédiaire (comme de l'assembleur) puis traduit si nécessaire vers un langage de bas niveau indépendant du langage du processeur P dudit dispositif D. Le compilateur C permet de réaliser des vérifications sur les agents portables A1, pour les restreindre dans leur environnement d'exécution et protéger le dispositif D d'agents portables A1 qui seraient mal intentionnés ou mal codés. Ainsi un agent exécutable A2 ne pourra pas, par exemple, utiliser toutes les fonctions de la bibliothèque du logiciel L et/ou ne pourra pas accéder à toute la mémoire de travail et/ou de stockage du dispositif D.

Le logiciel L réalise l'ensemble des traitements au sein du dispositif D : à ce titre, il peut, selon le cas d'utilisation, authentifier les utilisateurs du dispositif D, récupérer une politique de sécurité PS, récupérer avec ladite politique de sécurité PS des agents portables A1 spécialisés dans certaines fonctions de sécurité, récupérer les paquets de données PD, filtrer les paquets de données en fonction de ladite politique de sécurité PS, etc.

Sur la figure 4, l'automate de traitement des paquets du logiciel L est présenté. Les éléments suivants constituent cette figure :

5 : Pas de paquet reçu ;

6 : Attendre l'arrivée d'un paquet ;

51 : Paquet reçu ;

7 : Filtrer le paquet ;

8 : Y a-t-il des agents exécutables A2 concernés par le paquet ?

9 : Exécuter les agents exécutables A2 ;

10 : Y a-t-il des traitements secondaires ?

11 : Effectuer les traitements secondaires sur le paquet ;

12 : Envoyer le paquet ;

13 : Paquet refusé ;

14 : Paquet autorisé ;

15 : Non ;

16 : Oui.

Le logiciel L attend l'arrivée de nouveaux paquets. Après réception, il vérifie si le paquet est conforme à la politique de sécurité PS et filtre le paquet en autorisant ou non son passage. Si le paquet est autorisé, le logiciel L vérifie si des agents exécutables A2 sont concernés par le paquet, conformément à la politique de sécurité, et le cas échéant lesdits agents exécutables A2 sont exécutés. Le paquet subit ensuite optionnellement des traitements supplémentaires (chiffrement...). Après traitement et s'il y est autorisé par les agents exécutables A2, le paquet est envoyé au destinataire, sinon il est détruit.

Afin de permettre au logiciel L de déterminer si des agents doivent être appelés pour réaliser des traitements supplémentaires sur les paquets, la politique de sécurité doit pouvoir contenir une définition des agents et des relations avec les autres éléments de la politique de sécurité.

Il est possible de concevoir une politique de sécurité classique (pour un réseau utilisant la norme TCP/IP), basée sur des actions d'autorisation et de refus de paquet en fonction des adresses IP source et destination, des ports source et destination et du protocole de transport, tout en lui rajoutant une liste d'agents à exécuter. Le tableau suivant n'est qu'un exemple de politique de sécurité et les agents donnés dans cette politique de sécurité ne sont eux-mêmes donnés qu'à titre d'exemple.

Adresse source	Adresse destination	Service	Port	Protocole	Action	Agent
IP A	IP B	FTP	1	TCP	Autorisé	Agent FTP
IP A	IP C	POP3	10	TCP	Autorisé Chiffré	Agent SSOn POP3
IP A	IP C	SMTP	25	TCP	Autorisé Chiffré	-
IP A	IP C	HTTP	80	TCP	Autorisé	Contrôle Parental
All	All	All	*	All	Refusé	-

On peut voir sur ce tableau, que tout flux de communication est interdit entre des adresses Internet autre que IP A, IP B et IP C (dernière ligne du tableau). Le flux de communication entre les adresses IP B et IP C est également interdit (il n'y a pas de règle explicite d'autorisation de communication entre B et C, c'est donc la dernière ligne qui prévaut). Entre les adresses Internet IP A et IP B, tout le flux de communication est interdit mis à part le service FTP (protocole de transfert de fichiers), service sur lequel a été rajouté un agent FTP chargé de détecter la procédure de négociation dynamique de port du protocole FTP. Et entre les adresses IP A et IP C tout le flux est interdit mis à part :

le service pop3 (réception de courriers électroniques, Post Office Protocol - Version 3 : RFC 1939) qui est autorisé, qui doit, dans cet exemple, être chiffré et sur lequel a été ajouté l'agent SSOn POP 3 chargé de détecter la procédure d'authentification et d'insérer automatiquement le mot de passe de l'utilisateur.

le service SMTP (Simple Mail Transfer Protocol, protocole de transfert de courrier électronique - RFC 821) qui est autorisé et qui, dans cet exemple, doit être chiffré.

le service HTTP (HyperText Transfer Protocol, navigation sur les pages Internet - RFC 2068) qui est autorisé et sur lequel est appliqué un contrôle parental.

La figure 5 représente un schéma de réseau qui peut être employé dans le cas d'application de la politique de sécurité décrite dans le tableau précédent. Ce réseau comprend trois hôtes représentés par les adresses Internet IP A, IP B et IP C, ces hôtes sont reliés au même réseau. Deux dispositifs D1 et D2 sont positionnés respectivement entre l'hôte d'adresse IP A et le reste du réseau et entre l'hôte d'adresse IP C et le reste du réseau. Ainsi les hôtes d'adresses IP A et IP B (ainsi que IP B et IP C) n'ont qu'un dispositif les séparant alors que les hôtes d'adresses IP A et IP C ont les deux dispositifs qui les séparent.

La figure 6 explique le fonctionnement de l'agent FTP chargé de détecter la procédure de négociation dynamique de port. Les éléments suivants constituent cette figure :

15 : Non ;

16 : Oui ;

17 : Début ;

18 : Détection d'une négociation d'ouverture dynamique de port ;

19 : Récupération de l'IP B et du port X ;

20 : Modification de la politique de sécurité par l'ajout d'une règle ;

21 : Fin.

Pour mieux comprendre l'utilité de l'exemple de l'agent FTP utilisé dans la figure 6, il faut expliquer le protocole FTP. Ce protocole est divisé en deux flux de communication distincts : le premier est le flux de contrôle permettant d'envoyer les commandes au serveur et de recevoir les réponses. Ce flux utilise habituellement le port TCP 21 ; le deuxième est le flux de données des fichiers envoyés. Le port permettant de récupérer ce deuxième flux est initialement inconnu car il est négocié dans le premier flux, ce qui rend

impossible l'autorisation préalable du flux de données FTP pendant la phase de définition de la politique de sécurité.

L'agent est appelé pour chaque paquet FTP. Il se charge de détecter la phase de négociation de port dynamique du flux de données FTP dans le flux de communication initial. Une fois qu'il l'a détectée, l'agent récupère l'adresse IP B et le port négocié, ici X. Ensuite, il modifie la politique de sécurité en rajoutant une règle temporaire autorisant ce flux à passer.

Adresse source	Adresse destination	Service	Port	PROTOCOLE	Action	Agent
IP A	IP B	FTP	21	TCP	Autorisé	Agent FTP
IP A	IP C	POP3	110	TCP	Autorisé Chiffré	Agent SSOn POP3
IP A	IP C	SMTP	25	TCP	Autorisé Chiffré	-
IP A	IP C	HTTP	80	TCP	Autorisé	Contrôle Parental
IP A	IP B	FTP Data	X	TCP	Autorisé	-
All	All	All	*	All	Refusé	-

Nous pouvons constater dans le tableau précédent que l'agent FTP, après détection de la négociation dynamique de port, a rajouté une règle à la politique de sécurité, permettant aux hôtes d'adresse IP A et IP B de s'envoyer des fichiers via le port négocié (X dans notre exemple).

Par ailleurs, la politique de sécurité du dispositif D peut être fonction du ou des utilisateurs qui se sont identifiés auprès du dispositif. Plusieurs méthodes de réalisation sont alors possibles. Deux méthodes sont présentées : une méthode liée à un serveur d'authentification distant (figure 7) et une autre méthode liée à une authentification locale (figure 8).

Les éléments suivants constituent la figure 7 :

17 : Début ;

21 : Fin ;

22 : Un utilisateur U_i s'authentifie sur le dispositif D à l'aide d'un moyen d'identification ;

23 : Envoie de l'authentification au serveur distant ;

24 : Vérification de l'authentification par le serveur distant;

25 : Le serveur distant extrait :

La politique de sécurité PS en fonction de l'utilisateur U_i,

Les agents portables A1 correspondants,

Les paramètres de configuration correspondants ;

26 : Envoie de la politique de sécurité PS, des agents portables A1 et des paramètres de configurations au dispositif ;

27 : Stockage de la politique de sécurité PS, des paramètres de configuration et des agents exécutables A2 qui ont été obtenus après compilation des agents portables A1 par le compilateur C ;

28 : Authentification refusée ;

29 : Authentification accordée.

En figure 7, un utilisateur U_i s'authentifie sur le dispositif D (ceci peut être réalisé entre autre à l'aide d'un lecteur de carte à puce ou par un système d'identification biométrique...). L'authentification est envoyée au serveur distant qui vérifie l'authentification de l'utilisateur. Si cette authentification est refusée, le serveur coupe la communication. Au contraire, si l'authentification est autorisée, le serveur construit la politique de sécurité PS en fonction de l'utilisateur U_i en y incluant les agents portables A1 et les paramètres de configuration correspondants. Le serveur envoie alors toutes ces informations au dispositif D qui les stocke (par exemple en mémoire). L'utilisateur est alors authentifié et peut utiliser le dispositif avec sa politique de sécurité.

Cette méthode permet de centraliser toutes les politiques de sécurité PS de tous les dispositifs D dans un ou plusieurs serveurs centraux sur lesquels l'administration peut

se faire globalement. Cette méthode permet en outre d'envoyer de nouveaux agents portables A1 et donc de modifier totalement le comportement de tout ou partie des dispositifs D.

Les éléments suivants constituent la figure 8 :

17 : Début ;

21 : Fin ;

27 : Stockage de la politique de sécurité PS, des paramètres de configuration et des agents A2 qui ont été obtenus après compilation des agents portables A1 par le compilateur C ;

28 : Authentification refusée ;

29 : Authentification accordée ;

30 : Un utilisateur U_i s'authentifie sur le dispositif D à l'aide d'une application client/serveur dont l'application serveur se trouve dans le dispositif D ;

31 : Vérification de l'authentification par le dispositif D ;

32 : L'application serveur extrait :

La politique de sécurité PS en fonction de l'utilisateur U_i,

Les agents portables A1 correspondants,

Les paramètres de configuration correspondants.

En figure 8, un utilisateur s'authentifie via une application serveur (par exemple un serveur HTTP) incluse dans le logiciel L du dispositif. L'application serveur vérifie l'authentification. Si celle-ci est correcte, l'application serveur récupère et active alors la politique de sécurité PS de l'utilisateur U_i (comme dans la figure 7). Les informations sont directement contenues dans le dispositif D. Il est possible de paramétrer ces fonctionnalités et, d'une façon plus générale, la politique de sécurité PS, en fonction d'un utilisateur U_i. L'administration se fait localement sur le dispositif D via l'application serveur. Cette méthode peut être utilisée dans le cadre d'un dispositif D unique pour un réseau familial accédant à Internet ou pour une petite entreprise.

Les figures 7 et 8 ne sont que des exemples d'implémentation de l'invention. Il est tout à fait possible de coupler ces deux exemples et d'avoir une authentification de l'utilisateur à l'aide d'un serveur (Web ou autre) embarqué dans le dispositif D et d'avoir un serveur central qui vérifie cette authentification, génère la politique de sécurité puis la transmet au dispositif D.

Les services autres que le filtrage classique des paquets réalisé par une écluse (firewall) classique sont réalisés par les agents. Un agent peut potentiellement réaliser n'importe quel traitement sur les paquets. L'exemple qui suit montre la facilité d'implémentation d'un agent.

La figure 9 illustre l'automate d'un agent réalisant une fonctionnalité très originale de sécurité au niveau application (et non pas au niveau TCP/IP par exemple). Les éléments suivants constituent cette figure :

15 : Non ;

16 : Oui ;

17 : Début ;

21 : Fin ;

33 : Initialisation de l'agent ;

34 : Le paquet contient-il la commande « USER » ?

35 : Le paquet contient-il la commande « PASS » ?

36 : Y a-t-il un mot de passe associé au nom d'utilisateur ?

37 : Récupération et stockage du nom d'utilisateur ;

38 : Sauvegarde des paramètres de l'agent ;

39 : Calcul de la taille des données à rajouter au paquet ;

40 : Modification de la taille du paquet ;

41 : Insertion du mot de passe dans le paquet.

Cet agent se charge de réaliser l'authentification d'un utilisateur à son serveur de messagerie électronique via le protocole POP3 (Post Office Protocol - Version 3 : RFC 1939, commandes d'authentification de POP3 : RFC 1734). L'utilisateur

n'a plus à connaître son mot de passe. L'agent se charge de placer le mot de passe en fonction de l'identifiant de l'utilisateur.

L'automate de l'agent est relativement simple. L'agent cherche un paquet contenant la commande USER et extrait l'identifiant de l'utilisateur si la commande est trouvée. Puis il cherche un paquet contenant la commande PASS. Une fois qu'il l'a trouvée, l'agent retrouve le mot de passe correspondant à l'identifiant, calcule la taille à rajouter au paquet, agrandit le paquet et insère le mot de passe valide.

Voici un exemple de ce code écrit en « C », langage informatique de haut niveau.

```
int         main()
{
  /* definition des variables */
  int packet_size;
  char      *packet;
  char *(param[6]);
  int error, login_size, offset, pass_size;

  /* Récupération du paquet et des paramètres de
     l'agent */
  if    (    !(    packet    =    agent_getPacketData(
&packet_size)))
    return OK;
  agent_getAgentParam( param);
  /* On cherche la commande USER pour
     récupérer le login */
  if ( !strncmp( packet, "USER ", 5))
  {
    login_size = size - 7;
    if (login_size > 32)
    return -1;
    /* On sauvegarde le login et sa taille */
```

```
            strncpy( param[1] , packet + 5, login_size);
            (int)(param[2]) = login_size;
          }
          /* On cherche la commande PASS pour insérer
             le mot de passe */
          if ( !strncmp( packet, "PASS ", 5))
          {
            /* On récupère le mot de passe
               correspondant au login */
            if ( (offset = agent_getMatch( param[0], param[1],
                 (int)(param[2]))) == -1)
              return OK;
            pass_size = strlen( param[0] + offset);
            /* On augmente la taille du paquet
               et on insère le passe */
            agent_modifyMemSpace( packet + 5, pass_size);
            strncpy( packet + 5, param[1], pass_size);
          }
          /* Sauvegarde des paramètres de l'agent */
          agent_saveAgentParam( param);
          return OK;
        }
```

Cet exemple montre bien la facilité apportée par la présente invention pour permettre le rajout de nouvelles fonctionnalités de sécurité et/ou de gestion du réseau, sur le dispositif D. En quelques lignes de codes, il est possible d'effectuer des opérations sur les paquets. Etant donné la facilité d'accès aux paquets des flux de communication, l'agent peut rapidement lire et modifier les données des paquets. Ainsi, tout développeur peut écrire ses propres agents et augmenter sa base de fonctionnalités. Avec l'apparition de nouvelles menaces, l'implémentation de nouveaux agents détectant ces menaces et y remédiant est rapide et efficace. La diffusion à l'ensemble des dispositifs protège de manière homogène et instantanée l'ensemble du parc informatique. Concernant des services tels

que celui présenté plus haut, une politique de sécurité globale peut être déployée de la même manière à tout un réseau informatique.

La figure 10 présente un autre mode de réalisation de l'invention. Deux utilisateurs U1 et U2, sur deux postes différents PO1 et PO2, sont identifiés auprès du dispositif D et ont leur propre politique de sécurité. Tout flux de communication provenant du réseau R à destination d'un des postes est filtré avec la politique de sécurité correspondant à l'utilisateur du poste.

Cet exemple ne limite pas la présente invention à deux utilisateurs. La présente invention est capable de protéger autant de postes et/ou d'utilisateurs que souhaité, et cela, si souhaité, avec des politiques de sécurité PS différentes pour chacun d'eux.

Afin d'optimiser l'exécution des agents exécutables A2 sur les paquets, un cache de paquets permet de n'envoyer aux agents exécutables A2 qu'une seule version d'un même paquet et donc de leur présenter un flux cohérent. Le cache de paquets permet de prendre en charge les paquets déjà reçus afin de ne pas perturber les algorithmes des agents qui ne s'attendent pas à recevoir une nouvelle fois un paquet déjà traité. Ces phénomènes sont connus sous le nom de ré-émission de paquets et sont présents au niveau du protocole TCP.

La figure 11 donne l'automate général d'un cache de paquets. Les éléments suivants constituent cette figure :

15 : Non ;

16 : Oui ;

42 : Arrivée d'un paquet dans le dispositif ;

43 : Le paquet a-t-il déjà été reçu ?

44 : Le paquet a-t-il été modifié précédemment par les agents ?

45 : Envoi du paquet modifié sauvegardé ;

46 : Envoi du paquet ;

47 : Traitement par les agents ;

48 : Le paquet a-t-il été modifié par les agents ?

49 : Mémorise le paquet et les informations qui l'identifient ;

50 : Mémorise les informations identifiant le paquet.

Lorsqu'un paquet est reçu (42), le cache de paquet vérifie si le paquet a déjà été reçu précédemment (43). Dans la négative, les agents concernés par le paquet sont appelés (47). Une fois traitées, les informations permettant d'identifier le paquet (par exemple, son numéro de séquence TCP) sont sauvegardées ((49) ou (50)). Si le paquet a été modifié par les agents, le paquet modifié est sauvegardé avec les informations l'identifiant (49) puis il est envoyé sur le réseau (46). Sinon, il est simplement envoyé sur le réseau (46) après sauvegarde des informations permettant de l'identifier (50). Si le paquet a déjà été reçu (c'est-à-dire si on retrouve les informations l'identifiant dans le cache de paquet), le cache de paquet vérifie si le paquet modifié est sauvegardé (44), auquel cas le paquet modifié est envoyé sur le réseau sans exécution des agents (45). Sinon, le paquet déjà reçu est directement envoyé sur le réseau sans exécution des agents (46). On garantit ainsi aux agents qu'ils ne recevront pas une nouvelle fois un paquet qu'ils ont déjà traité.

Illustrons un cas particulier possible : un agent chargé de détecter un virus suspecte la présence d'un virus dans un paquet 1, mais nécessite de réaliser une analyse du paquet 2 pour en être convaincu. Si le paquet 1 est reçu pour la deuxième fois (paquet 1 bis), l'agent réalisera le traitement du paquet 2 sur ce paquet 1 bis, ce qui faussera l'analyse. Le cache de paquet permet de renvoyer directement la bonne version du paquet 1 sans exécuter les agents. Deux cas se présentent : le paquet 1 a ou n'a pas été modifié par un agent lors de sa première réception. Dans le premier cas, le paquet 1 modifié a été sauvegardé la première fois. C'est la version sauvegardée qui est envoyée sans exécuter les agents. Dans le second cas, le paquet 1 bis est renvoyé directement sans exécution des agents.

Les agents disposent de nombreuses fonctions de traitement des paquets. Mais ils disposent également de fonctions leur permettant de communiquer avec l'ensemble des éléments constitutifs du réseau. Ces fonctions sont indispensables pour mettre en place une sécurité globale du réseau.

La figure 12 en démontre l'intérêt. Considérons un réseau R peu sûr, dans lequel évolue un pirate informatique depuis un poste H. Ce pirate élabore une attaque (1) en destination d'un serveur web SW d'une entreprise, ce serveur web étant accessible via un routeur RO. Le serveur web SW est protégé par le dispositif D (mettant en œuvre l'invention). L'agent A est chargé de sécuriser le serveur web. A la détection de l'attaque (1), l'agent A bloque l'attaque (1) et envoie un ordre (2) de reconfiguration du routeur RO afin de bloquer les communications venant du poste H (par exemple via le protocole SNMP). Il envoie alors un message d'avertissement (3) au serveur de log SL centralisant un journal des évènements (par exemple via le protocole syslog). Le dispositif D est ainsi réactif aux attaques et il peut communiquer aux autres périphériques réseau des informations oeuvrant pour la sécurité du réseau.

La figure 13 présente l'utilisation de fonctions F d'une bibliothèque de fonctions BF contenue le logiciel L, certaines de ces fonctions F pouvant être associées à des composants matériels spécialisés CM présents dans le dispositif D. Le processeur P contenu dans le dispositif D exécute 4 le logiciel L. Le logiciel L fait appel à des fonctions F contenues dans la bibliothèque de fonctions BF. Ces fonctions F peuvent être codées sous la forme d'un logiciel exécuté par le processeur P. Elles peuvent aussi utiliser (52) des composants matériels spécialisés CM qui leur sont associés.

La figure 14 décrit les différentes phases d'un compilateur. Elle est constituée des éléments suivants :

53 : programme source ;

54 : analyseur lexical ;

55 : analyseur syntaxique ;
56 : analyseur sémantique ;
57 : générateur de code intermédiaire ;
58 : optimiseur de code ;
59 : générateur de code ;
60 : programme cible ;
61 : gestionnaire de la table de symboles ;
62 : gestionnaire d'erreurs.

Un programme source (53) écrit dans un langage est transformé par le compilateur en un programme cible (60) écrit dans un autre langage de plus bas niveau (plus proche du langage machine). Le programme source passe par les différentes phases suivantes :

analyse : trois analyseurs constituent cette phase d'analyse, ce sont les analyseurs lexical (54), syntaxique (55) et sémantique (56) qui décomposent le code en unités lexicales, les classent hiérarchiquement et contrôlent s'il y a, ou non, des erreurs sémantiques.

génération de code intermédiaire : le programme source est transformé par le générateur de code intermédiaire (57) en code intermédiaire facile à produire et facile à traduire en langage cible.

optimisation : un optimiseur de code (58) tente d'améliorer le code intermédiaire de façon que le code résultant s'exécute plus rapidement.

génération de code : un générateur de code (59) produit alors le programme cible (60).

Toutes les phases décrites ci-dessus utilisent le gestionnaire de la table des symboles (61) et le gestionnaire d'erreurs (62). Le premier enregistre les identificateurs utilisés dans le programme source et collecte de l'information sur divers attributs de chaque identificateur. Le second gère les erreurs provenant des différentes phases qui les traitent de façon à pouvoir continuer la compilation afin de détecter d'autres éventuelles erreurs.

REVENDICATIONS

1. Procédé permettant de réaliser l'analyse et/ou la modification sélective et/ou le filtrage sélectif de paquets de données (PD) traversant un dispositif (D) placé en coupure dans un réseau informatique (R) ; ledit dispositif (D) comprenant un processeur (P) exécutant un compilateur (C) et un logiciel (L) conformément à une politique de sécurité (PS) ; ledit logiciel (L) étant destiné au filtrage desdits paquets de données (PD), en autorisant ou non leur passage, conformément à ladite politique de sécurité (PS); ledit procédé étant caractérisé en ce qu'il comprend les étapes suivantes :

- l'étape de définir ladite politique de sécurité (PS) au moyen d'agents portables (A1), écrits dans un langage informatique (Li) indépendant du langage dudit processeur (P) et dédiés à l'analyse et/ou à la modification sélective et/ou au filtrage sélectif desdits paquets de données (PD) ;
- l'étape, pour ledit logiciel (L), d'appeler automatiquement ledit compilateur (C) afin d'effectuer une compilation pour traduire lesdits agents portables (A1) en des agents exécutables (A2) écrits dans le langage dudit processeur (P) ;
- l'étape d'exécuter ledit logiciel (L) pour filtrer lesdits paquets de données (PD) traversant ledit dispositif (D), en autorisant ou non leur passage, conformément à ladite politique de sécurité (PS) ;
- l'étape d'analyser, lesdits paquets de données (PD) autorisés par ledit logiciel (L) à traverser ledit dispositif (D), en exécutant lesdits agents exécutables (A2) par ledit processeur (P) ; et/ou
- l'étape de modifier sélectivement, lesdits paquets de données (PD) autorisés par ledit logiciel (L) à traverser ledit dispositif (D), en exécutant lesdits agents exécutables (A2) par ledit processeur (P) ; et/ou
- l'étape de filtrer sélectivement, lesdits paquets de données (PD) autorisés par ledit logiciel (L) à traverser ledit

dispositif (D), en exécutant lesdits agents exécutables (A2) par ledit processeur (P).

2. Procédé selon la revendication 1, caractérisé en ce que ladite politique de sécurité (PS) comprend en outre une définition des différents objets dudit réseau informatique (R).

3. Procédé selon l'une quelconque des revendications 1 ou 2, caractérisé en ce que ladite politique de sécurité (PS) comprend en outre une définition des différents services dudit réseau informatique (R).

4. Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce que ladite politique de sécurité (PS) comprend en outre une définition des différents utilisateurs (U_i) dudit réseau informatique (R).

5. Procédé selon la revendication 4, caractérisé en ce qu'il comprend en outre l'étape de générer des paramètres de configuration permettant de configurer lesdits agents portables (A1) en fonction desdits utilisateurs (U_i) dudit réseau informatique (R).

6. Procédé selon l'une quelconque des revendications 1 à 5, caractérisé en ce que ladite politique de sécurité (PS) comprend en outre une définition dudit dispositif (D).

7. Procédé selon l'une quelconque des revendications 1 à 6, caractérisé en ce que ledit langage informatique (Li) est un langage de bas niveau dédié à des traitements sur lesdits paquets de données (PD) dudit réseau informatique (R) et permettant de contrôler et de limiter les actions possibles desdits agents portables (A1) au sein dudit dispositif (D).

8. Procédé selon l'une quelconque des revendications 1 à 7, caractérisé en ce qu'il comprend en outre l'étape de définir, sur un serveur distant dudit dispositif (D), ladite politique de sécurité (PS).

9. Procédé selon l'une quelconque des revendications 1 à 7, caractérisé en ce qu'il comprend en outre l'étape de définir, sur ledit dispositif (D), ladite politique de sécurité (PS).

10. Procédé selon l'une quelconque des revendications 1 à 9, caractérisé en ce qu'il comprend en outre l'étape d'authentifier le ou les utilisateurs (U_i), non authentifiés, dudit dispositif (D).

11. Procédé selon la revendication 10, caractérisé en ce que ladite politique de sécurité (PS) comprend en outre une définition desdits utilisateurs (U_i) authentifiés dudit dispositif (D).

12. Procédé selon la revendication 11, caractérisé en ce qu'il comprend en outre l'étape d'authentifier ledit ou lesdits utilisateurs (U_i), non authentifiés, dudit dispositif (D) à l'aide d'un moyen d'identification associé audit dispositif (D).

13. Procédé selon la revendication 11, caractérisé en ce qu'il comprend en outre l'étape d'authentifier ledit ou lesdits utilisateurs (U_i), non authentifiés, dudit dispositif (D) à l'aide d'une application client/serveur dont l'application serveur est contenue dans ledit dispositif (D).

14. Procédé selon l'une quelconque des revendications 1 à 13, caractérisé en ce qu'il comprend en outre l'étape d'exécuter des fonctions (F) d'une bibliothèque de fonctions (BF) contenue dans ledit logiciel (L) et appelée par lesdits agents exécutables (A2).

15. Procédé selon la revendication 14 caractérisé, en ce qu'il comprend en outre l'étape d'exécuter des fonctions (F), de ladite bibliothèque de fonctions (BF), spécialisées dans une gestion d'un cache desdits paquets de données (PD).

16. Procédé selon la revendication 15, caractérisé en ce que la gestion dudit cache desdits paquets de données (PD) comprend les étapes suivantes :

- l'étape, après exécution desdits agents exécutables (A2), de mémoriser, dans ledit cache, des informations de paquets concernant lesdits paquets de données (PD) et en outre lesdits paquets de données (PD) eux-mêmes lorsqu'ils ont été modifiés lors de ladite exécution ;

- l'étape, lors de l'arrivée d'un paquet entrant dans ledit dispositif (D), de vérifier, grâce audites informations de paquets enregistrées dans ledit cache, si ledit paquet entrant est un paquet déjà reçu ;

- l'étape, lorsque ledit paquet entrant n'est pas un paquet déjà reçu, d'exécuter lesdits agents exécutables (A2) ;

- l'étape, lorsque ledit paquet entrant est un paquet déjà reçu, de déterminer, grâce audites informations de paquets mémorisées dans ledit cache, si ledit paquet déjà reçu avait été modifié par lesdits agents exécutables (A2) ;

- l'étape, lorsque ledit paquet déjà reçu avait été modifié par lesdits agents exécutables (A2), de transmettre vers ledit réseau informatique (R), sans exécuter lesdits agents exécutables (A2), une version dudit paquet déjà reçu mémorisée dans ledit cache ;

- l'étape, lorsque ledit paquet déjà reçu n'avait pas été modifié par lesdits agents exécutables (A2), de transmettre vers ledit réseau informatique (R) ledit paquet entrant tel quel, sans exécuter lesdits agents exécutables (A2).

17. Procédé selon l'une quelconque des revendications 14 à 16, caractérisé en ce qu'il comprend en outre l'étape d'exécuter des fonctions (F), de ladite bibliothèque de fonctions (BF), spécialisées dans une gestion des couches réseau et transport du protocole de communication utilisé.

18. Procédé selon la revendication 17 caractérisé en ce que la gestion desdites couches réseau et transport comprend en outre les étapes suivantes :

- l'étape de mémoriser des informations de protocole desdites couches réseau et transport desdits paquets de données (PD) traversant ledit dispositif (D) afin de réaliser un suivi des différents flux desdits paquets de données (PD) ;

- l'étape de mémoriser des modifications desdits paquets de données (PD) réalisées par lesdits agents exécutables (A2) ;

- l'étape de mettre à jour lesdites informations de protocole desdites couches réseau et transport desdits paquets de données (PD) traversant ledit dispositif (D), en fonction desdites informations de protocole et desdites modifications mémorisées, sur lesdits paquets de données (PD) afin de conserver une cohérence des flux desdits paquets de données (PD).

19. Procédé selon l'une quelconque des revendications 14 à 18 caractérisé en ce qu'il comprend en outre l'étape d'exécuter des fonctions (F), de ladite bibliothèque de fonctions (BF), spécialisées dans une recherche de motifs et d'expressions régulières.

20. Procédé selon l'une quelconque des revendications 14 à 19 caractérisé en ce qu'il comprend en outre l'étape d'exécuter des fonctions (F), de ladite bibliothèque de fonctions (BF), spécialisées dans une communication entre lesdits agents exécutables (A2).

21. Procédé selon l'une quelconque des revendications 14 à 20 caractérisé en ce qu'il comprend en outre l'étape d'exécuter des fonctions (F), de ladite bibliothèque de fonctions (BF), spécialisées dans une communication entre lesdits agents exécutables (A2) et desdits objets dudit réseau informatique (R).

22. Procédé selon l'une quelconque des revendications 14 à 21 caractérisé en ce qu'il comprend en outre l'étape d'associer des composants matériels spécialisés (CM) dudit dispositif (D) à des fonctions (F) de ladite bibliothèque de fonctions (BF) afin d'accélérer l'exécution desdites fonctions (F).

23. Procédé selon l'une quelconque des revendications 1 à 22 caractérisé en ce qu'il comprend en outre l'étape de modifier ladite politique de sécurité (PS) en exécutant lesdits agents exécutables (A2) par ledit processeur (P).

24. Système permettant de réaliser l'analyse et/ou la modification sélective et/ou le filtrage sélectif de paquets de données (PD) ; ledit système comprenant :

un dispositif (D) traversé par lesdits paquets de données (PD) et placé en coupure dans un réseau informatique (R), ledit dispositif (D) comprenant un processeur (P) exécutant un compilateur (C) et un logiciel (L) conformément à une politique de sécurité (PS) ; ledit logiciel (L) comprenant des moyens de filtrage pour filtrer lesdits paquets de données (PD) traversant ledit dispositif (D), en autorisant ou non leur passage, conformément à ladite politique de sécurité (PS) ; et ;

des agents portables (A1), destinés à définir ladite politique de sécurité (PS), écrits dans un langage informatique (Li) indépendant du langage dudit processeur (P) et dédiés à l'analyse et/ou la modification sélective et/ou le filtrage sélectif desdits paquets de données (PD) ;

ledit compilateur (C) étant activé automatiquement par ledit logiciel (L) pour traduire lesdits agents portables (A1) en des agents exécutables (A2) écrits dans le langage dudit processeur (P) ; lesdits agents exécutables (A2) étant exécutés par ledit processeur (P) pour :

analyser lesdits paquets de données (PD) autorisés par ledit logiciel (L) à traverser ledit dispositif (D), et/ou

modifier sélectivement lesdits paquets de données (PD) autorisés par ledit logiciel (L) à traverser ledit dispositif (D), et/ou

filtrer sélectivement lesdits paquets de données (PD) autorisés par ledit logiciel (L) à traverser ledit dispositif (D).

25. Système selon la revendication 24 ; ledit système étant tel que ladite politique de sécurité (PS) comprend en outre une définition des différents objets dudit réseau informatique (R).

26. Système selon l'une quelconque des revendications 24 ou 25 ; ledit système étant tel que ladite politique de

sécurité (PS) comprend en outre une définition des différents services dudit réseau informatique (R).

27. Système selon l'une quelconque des revendications 24 à 26 ; ledit système étant tel que ladite politique de sécurité (PS) comprend en outre une définition des différents utilisateurs (U_i) dudit réseau informatique (R).

28. Système selon la revendication 27 caractérisé en ce qu'il comprend en outre des moyens de génération de paramètres de configuration pour configurer lesdits agents portables (A1), en fonction desdits utilisateurs (U_i) dudit réseau informatique (R).

29. Système selon l'une quelconque des revendications 24 à 28 ; ledit système étant tel que ladite politique de sécurité (PS) comprend en outre une définition dudit dispositif (D).

30. Système selon l'une quelconque des revendications 24 à 29 ; ledit système étant tel que ledit langage informatique (Li) est un langage de bas niveau dédié à des traitements sur lesdits paquets de données (PD) dudit réseau informatique (R) et permettant de contrôler et de limiter les actions possibles desdits agents portables (A1) au sein dudit dispositif (D).

31. Système selon l'une quelconque des revendications 24 à 30 caractérisé en ce qu'il comprend un serveur distant dudit dispositif (D) pour définir ladite politique de sécurité (PS).

32. Système selon l'une quelconque des revendications 24 à 30 ; ledit système étant tel que ledit dispositif comprend des moyens d'administration pour définir ladite politique de sécurité (PS).

33. Système selon l'une quelconque des revendications 24 à 32 caractérisé en ce qu'il comprend des moyens d'authentification du ou des utilisateurs (U_i), non authentifiés, dudit dispositif (D).

34. Système selon la revendication 33 ; ledit système étant tel que ladite politique de sécurité (PS) comprend en

outre une définition desdits utilisateurs (U_i) authentifiés dudit dispositif (D).

35. Système selon la revendication 34 caractérisé en ce que ledit dispositif (D) comprend un moyen d'identification pour authentifier ledit ou lesdits utilisateurs (U_i), non authentifiés, dudit dispositif (D).

36. Système selon la revendication 34 caractérisé en ce que ledit dispositif (D) comprend une application serveur d'une application client/serveur destinée à authentifier ledit ou lesdits utilisateurs (U_i), non authentifiés, dudit dispositif (D).

37. Système selon l'une quelconque des revendications 24 à 36 caractérisé en ce que ledit logiciel comprend une bibliothèque de fonctions (BF) dont les fonctions (F) sont appelées par lesdits agents exécutables (A2).

38. Système selon la revendication 37 ; ledit système étant tel que ladite bibliothèque de fonctions (BF) comprend en outre des fonctions (F) spécialisées dans une gestion d'un cache desdits paquets de données (PD).

39. Système selon la revendication 38 caractérisé en ce que ledit cache desdits paquets de données (PD) comprend :

une mémoire pour stocker, après exécution desdits agents exécutables (A2), des informations de paquets concernant lesdits paquets de données (PD) et pour stocker lesdits paquets de données (PD) eux-mêmes ;

des moyens de contrôle pour vérifier, grâce audites informations de paquets mémorisées dans ledit cache, si un paquet entrant est un paquet déjà reçu et s'il avait été modifié par lesdits agents exécutables (A2) ;

des moyens d'activation pour activer, en fonction des vérifications opérées par les moyens de contrôles,

soit des moyens de transmission pour transmettre vers ledit réseau informatique (R) sans modification un paquet de données (PD) stocké dans ladite mémoire ;

soit des moyens de transmission pour transmettre vers ledit réseau informatique (R) sans modification un paquet entrant.

40. Système selon l'une quelconque des revendications 37 à 39 ; ledit système étant tel que ladite bibliothèque de fonctions (BF) comprend en outre des fonctions (F) spécialisées dans une gestion des couches réseau et transport du protocole de communication utilisé.

41. Système selon la revendication 40 ; ledit système étant tel que ledit dispositif (D) comprend :

au moins une mémoire

pour stocker des informations de protocole desdites couches réseau et transport desdits paquets de données (PD) traversant ledit dispositif (D) afin de réaliser un suivi des différents flux desdits paquets de données (PD),

pour stocker des modifications desdits paquets de données (PD) réalisées par lesdits agents exécutables (A2) ;

des moyens de mise à jour desdites informations de protocole desdites couches réseau et transport desdits paquets de données (PD) traversant ledit dispositif (D), en fonction desdites informations de protocole et desdites modifications mémorisées, sur lesdits paquets de données (PD) afin de conserver une cohérence des flux desdits paquets de données (PD).

42. Système selon l'une quelconque des revendications 37 à 41 ; ledit système étant tel que ladite bibliothèque de fonctions (BF) comprend en outre des fonctions (F) spécialisées dans une recherche de motifs et d'expressions régulières.

43. Système selon l'une quelconque des revendications 37 à 42 ; ledit système étant tel que ladite bibliothèque de fonctions (BF) comprend en outre des fonctions (F) spécialisées dans une communication entre lesdits agents exécutables (A2).

44. Système selon l'une quelconque des revendications 37 à 43 ; ledit système étant tel que ladite bibliothèque de fonctions (BF) comprend des fonctions (F) spécialisées dans une

communication entre lesdits agents exécutables (A2) et desdits objets dudit réseau informatique (R).

45. Système selon l'une quelconque des revendications 37 à 44 caractérisé en ce que ledit dispositif (D) comprend des composants matériels spécialisés (CM) associés à des fonctions (F) de ladite bibliothèque de fonctions (BF) afin d'accélérer l'exécution desdites fonctions (F).

46. Système selon l'une quelconque des revendications 24 à 45 ; ledit système étant tel que lesdits agents exécutables (A2), exécutés par ledit processeur (P), modifient ladite politique de sécurité (PS).

FIGURE 1

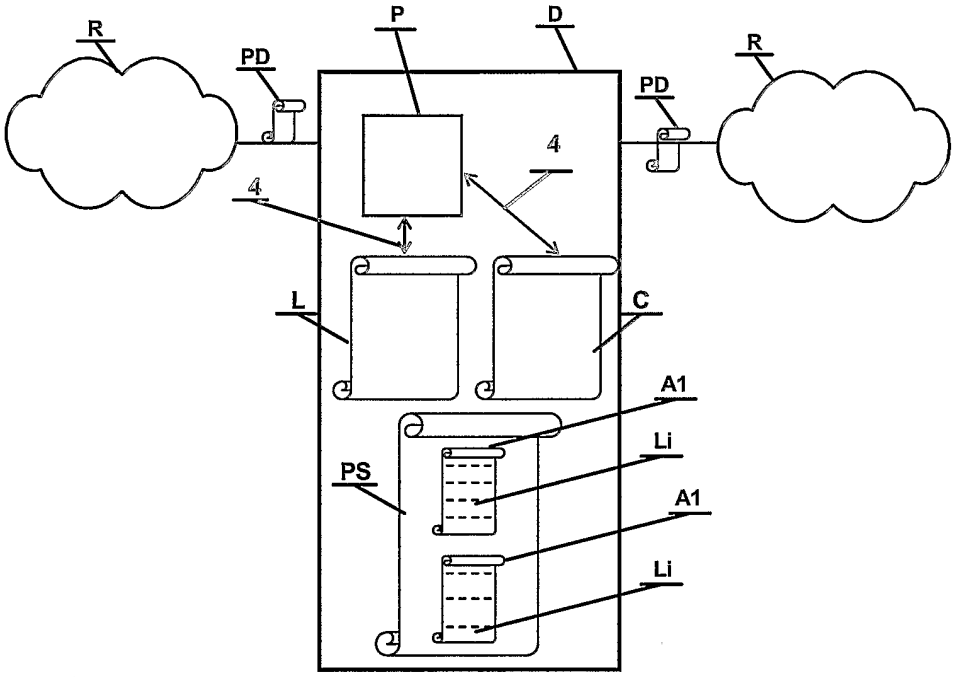


FIGURE 2

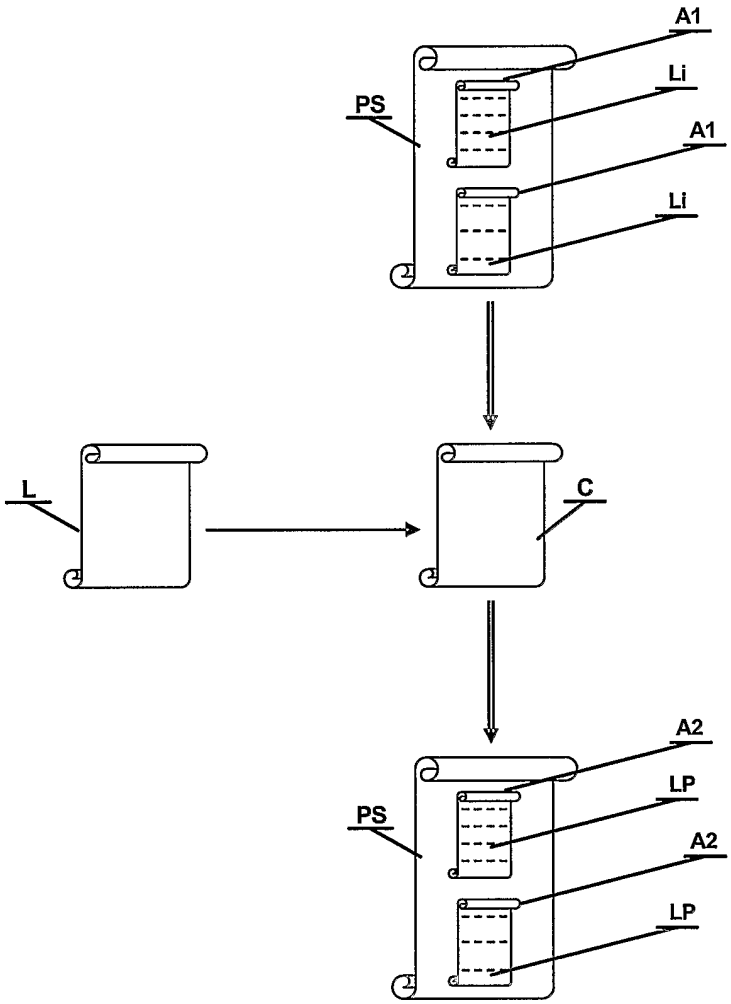


FIGURE 3

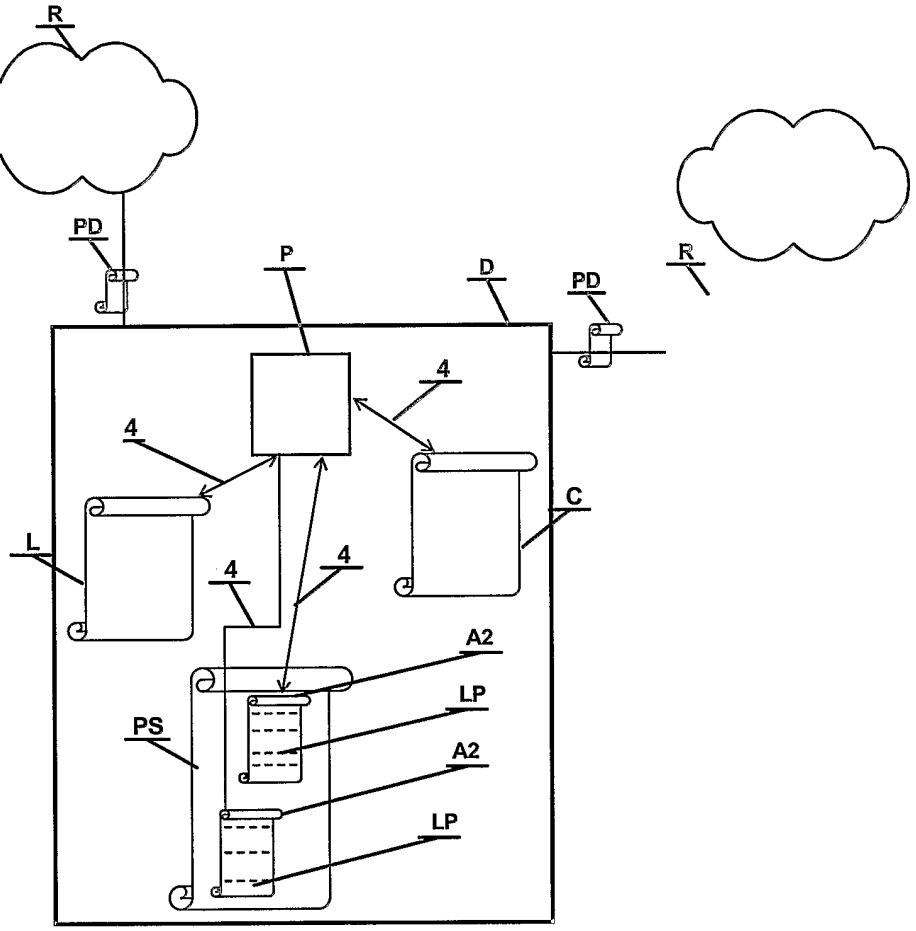


FIGURE 4

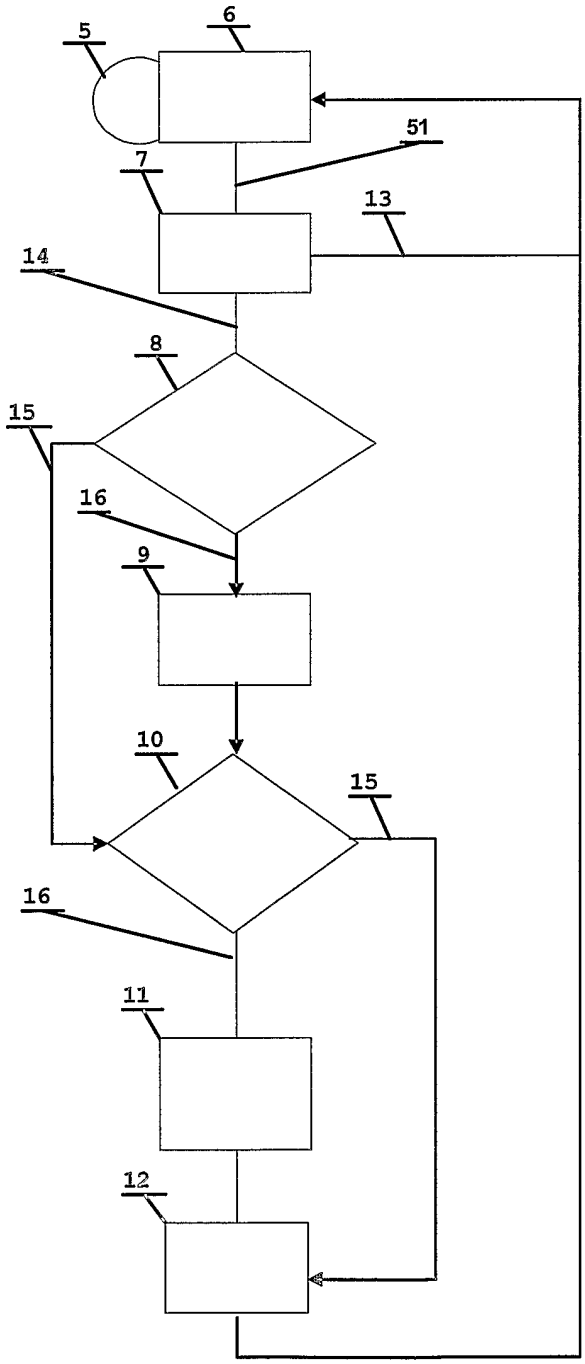


FIGURE 5

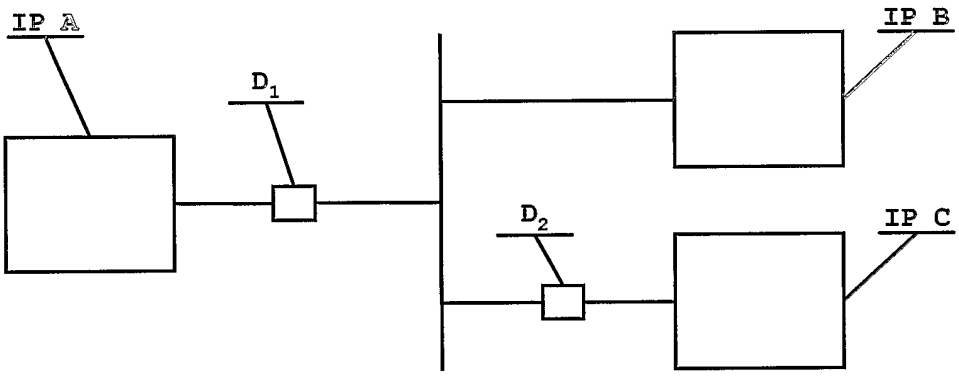


FIGURE 6

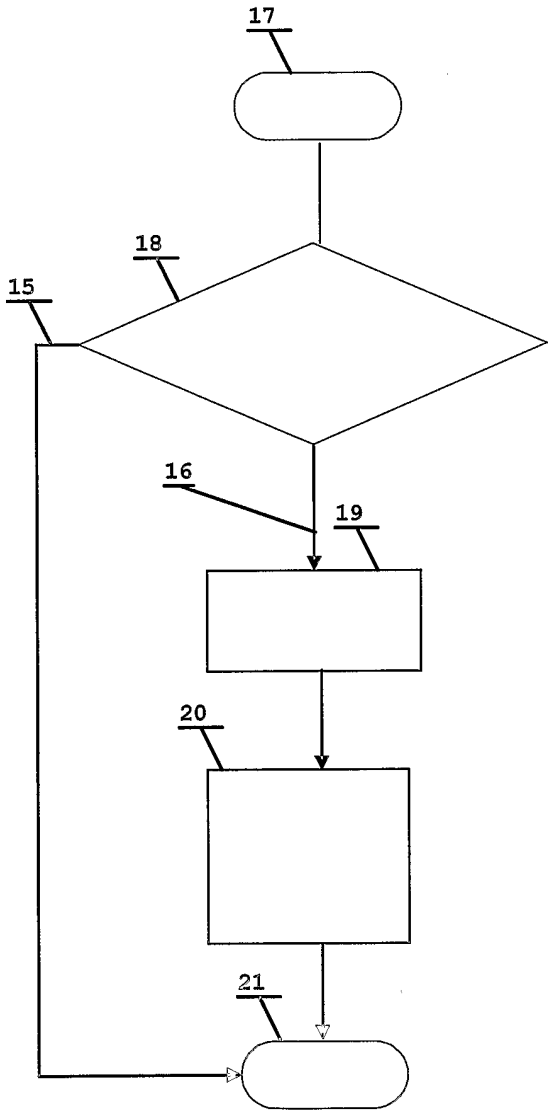


FIGURE 7

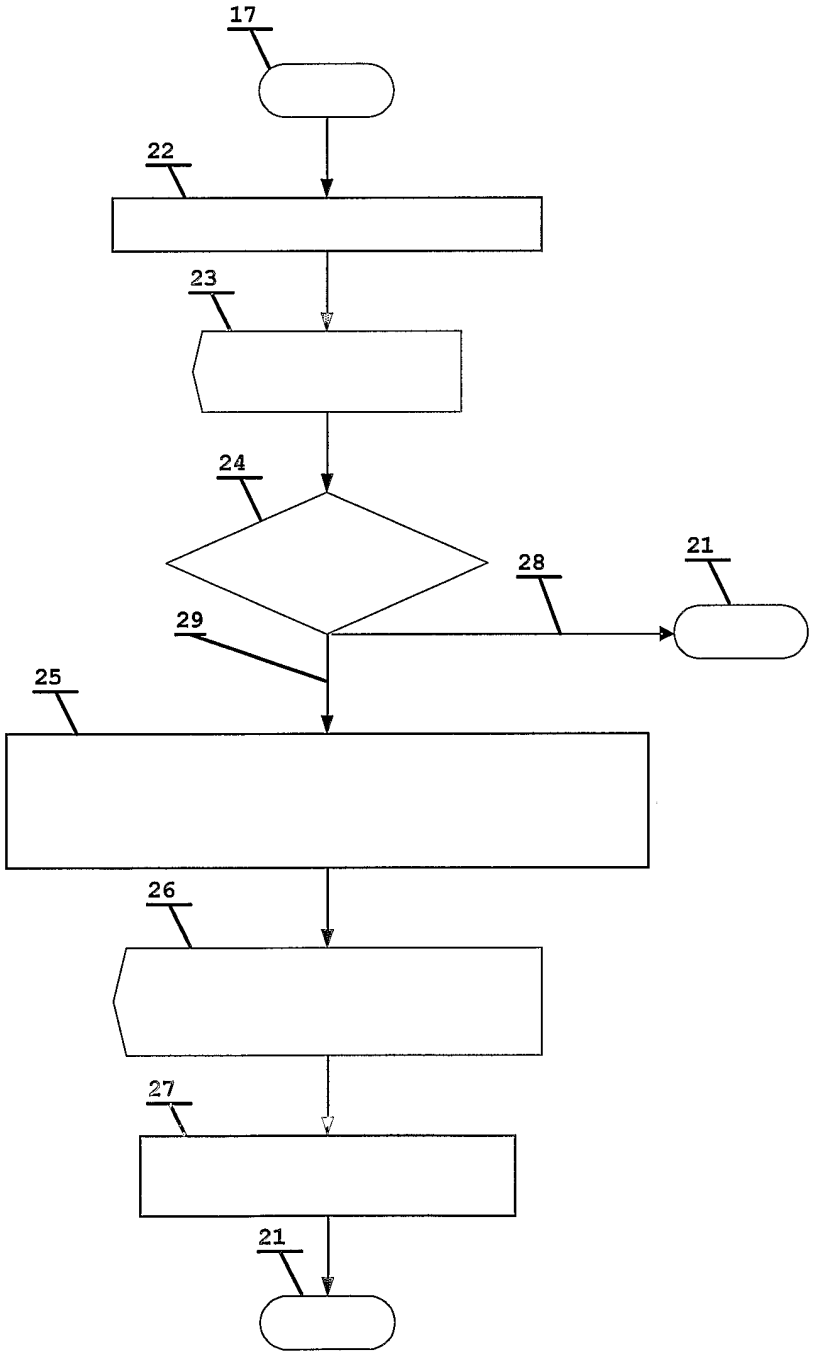


FIGURE 8

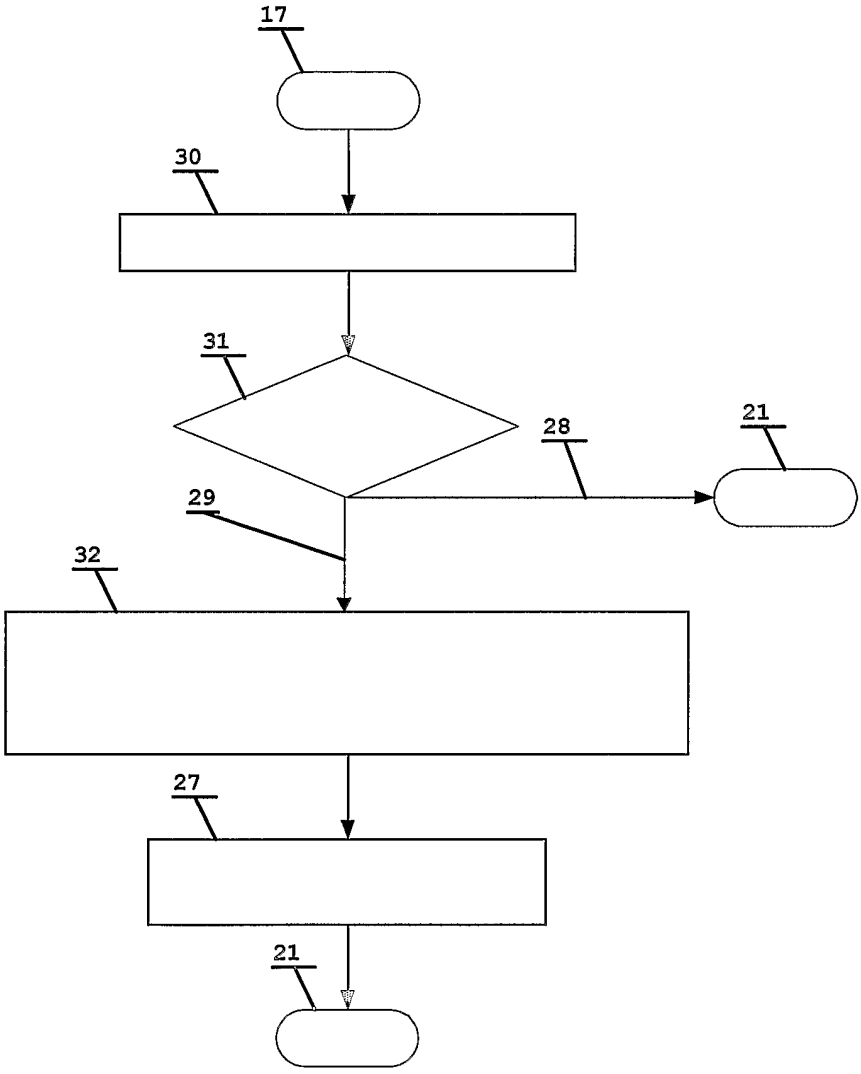


FIGURE 9

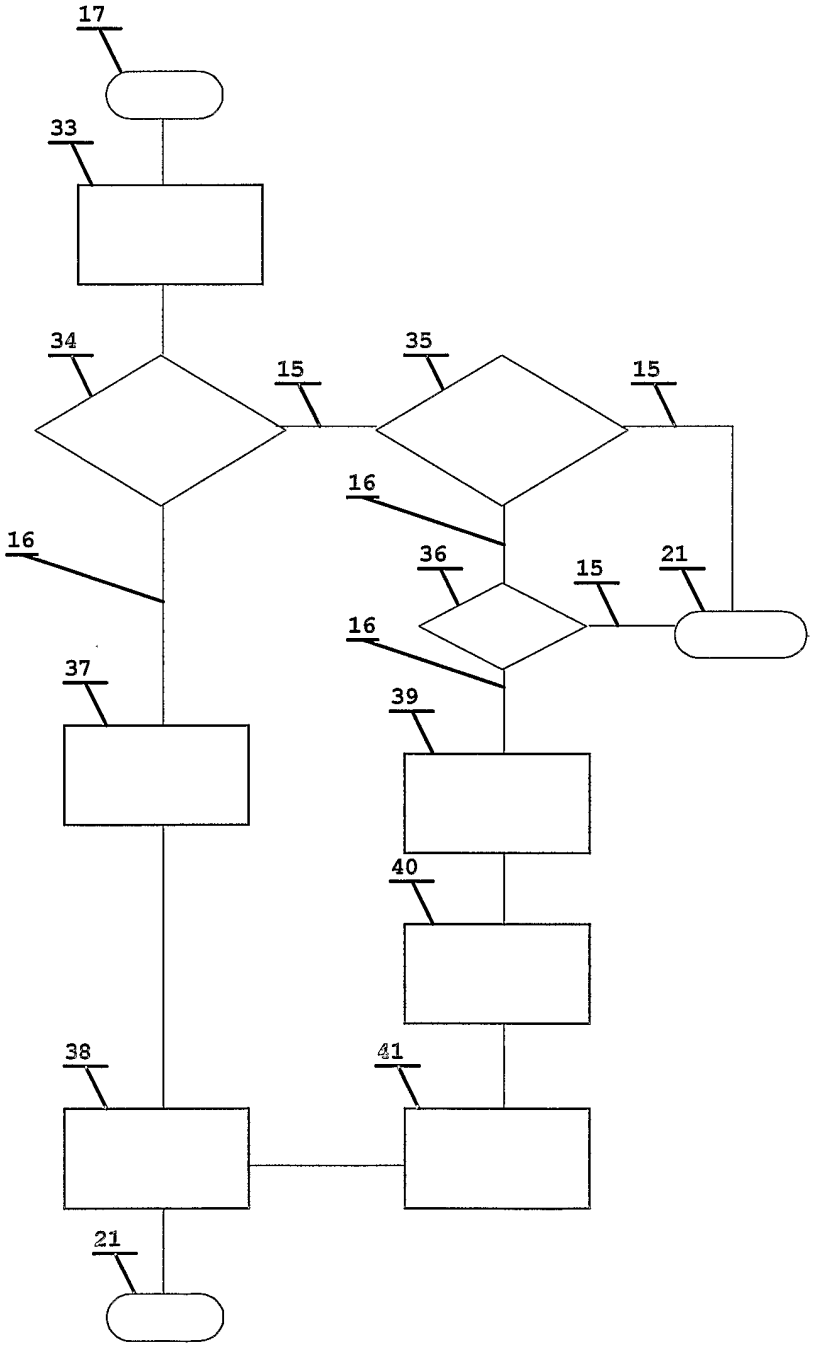


FIGURE 10

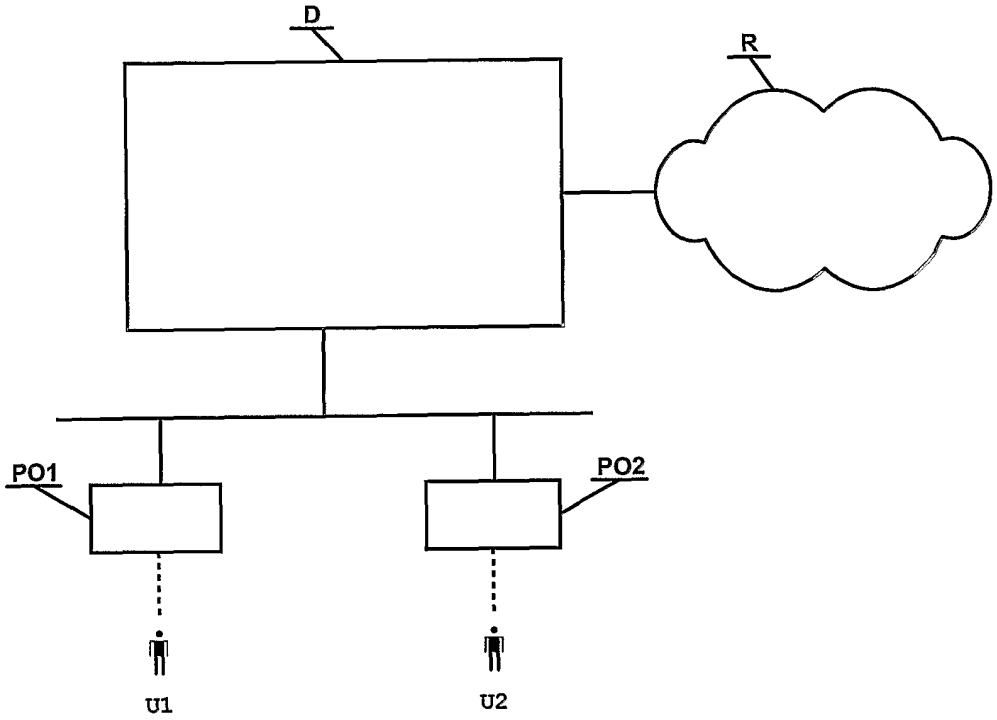


FIGURE 11

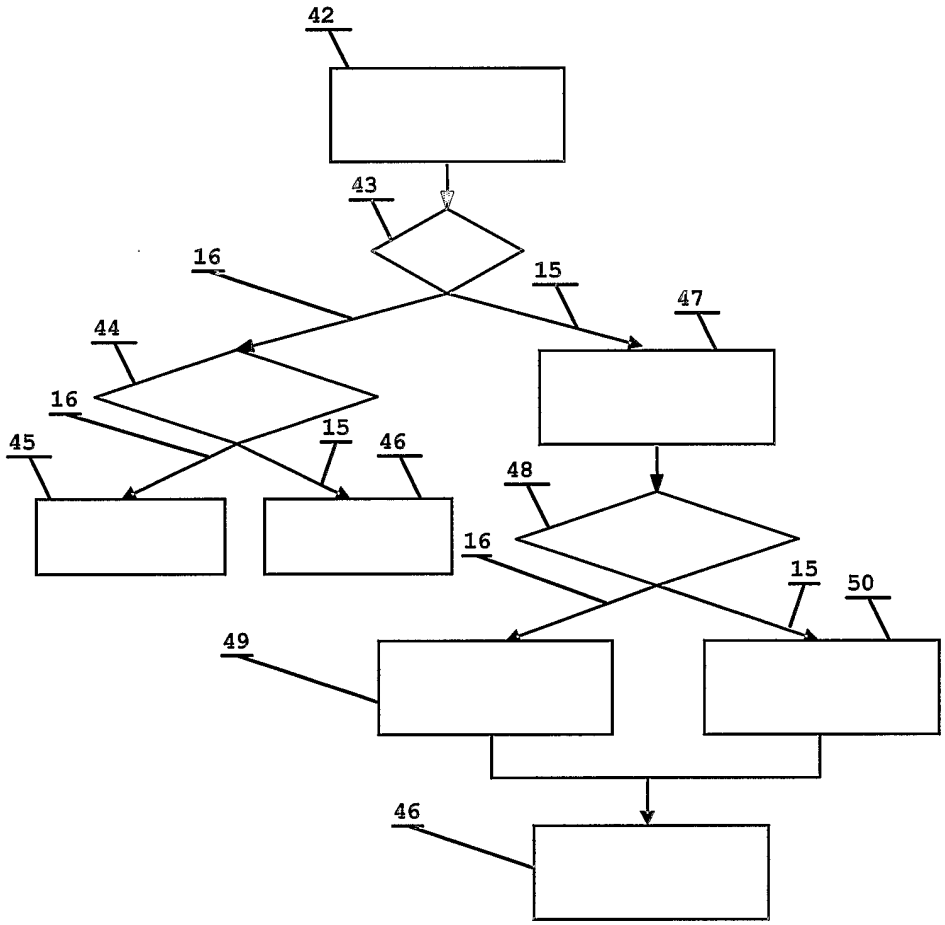


FIGURE 12

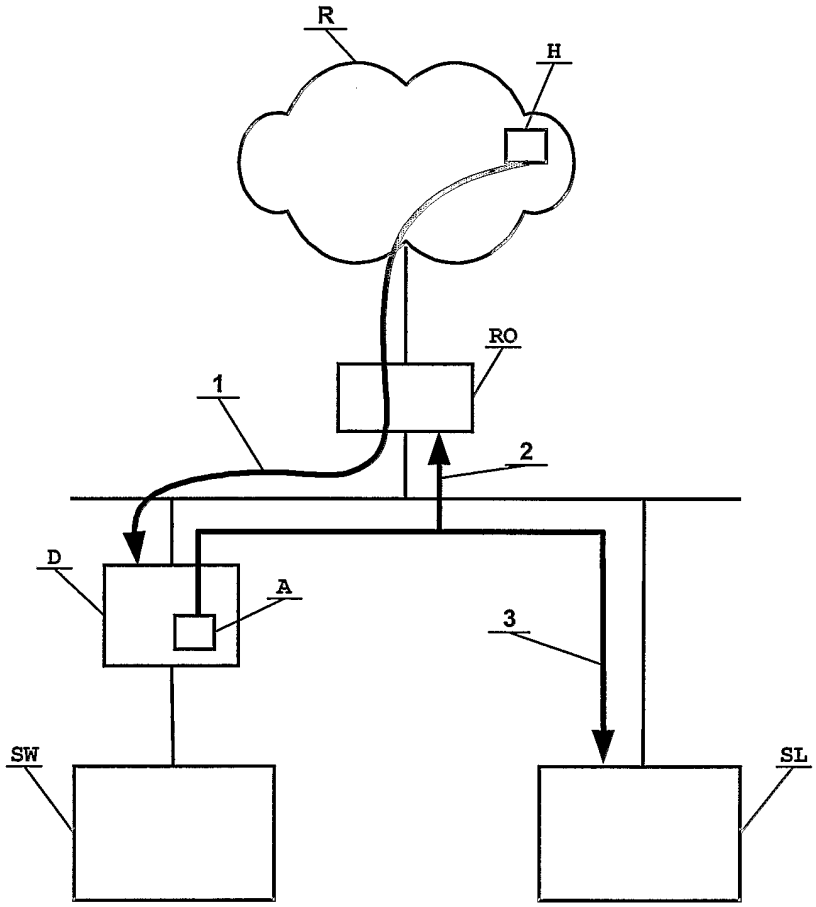


FIGURE 13

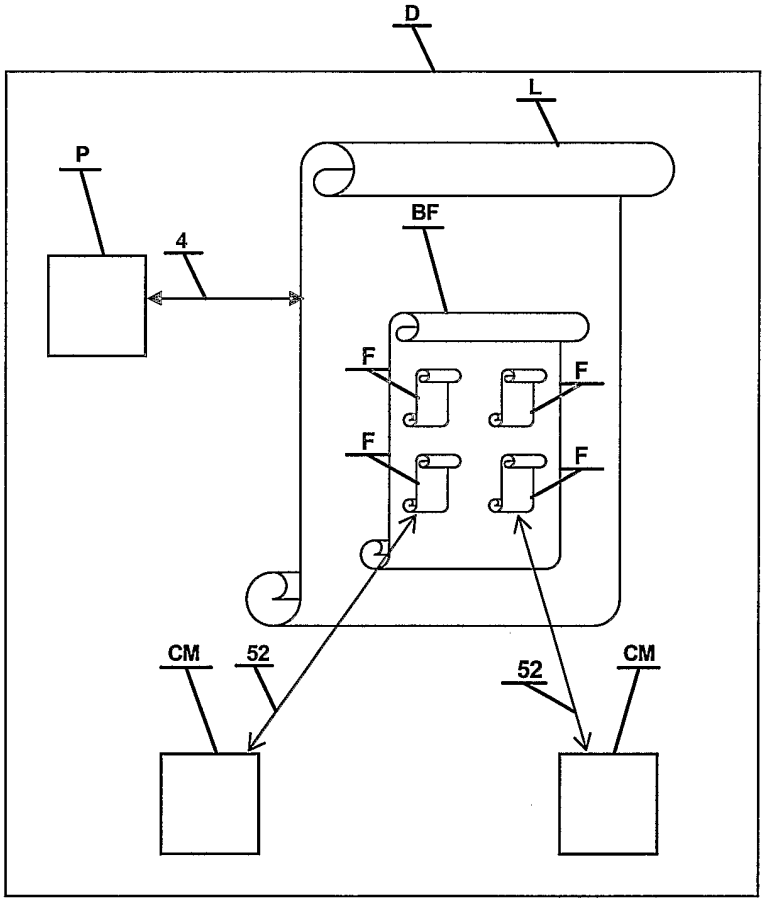


FIGURE 14