Approches Multi-Agents pour la Conception de Systèmes de Détection, de Prévention et de Réponse aux Intrusions.

Clément DUHART

17 janvier 2011

Résumé

Le développement permanent des systèmes d'information a comme corolaire une augmentation significative des attaques réseaux. Le caractère vital des données qui transitent sur ces réseaux implique une nécessité de protection et de sureté.

Ce rapport s'intéresse à l'étude d'architectures basées sur des modèles multi-agents permettant la détection, la prévention et éventuellement la réponse du système protégé face à une situation d'intrusion.

A partir d'un corpus non exhaustif de publications dans le domaine, ce document présente les différentes réponses actuelles pour traiter les problèmes d'intrusions ainsi que leurs limites.

Table des matières

Introduction

La multiplication des logiciels, des services et des fonctionnalités des systèmes informatiques a rendu plus opaque les architectures des systèmes d'exploitation. Le développement des réseaux informatiques et des moyens de télécommunication complexifie considérablement les interactions au sein des systèmes d'exploitation à travers la dimension des services distants.
Les informations circulant sur les systèmes et leurs réseaux sont de plus en plus nombreuses et diverses. Elles comprennent de plus en plus de données sensibles, tant pour les industriels et les administrations que pour les particuliers. Il se développe ainsi un besoin de fiabilité et de sécurité pour ces données qui nécessite le développement d'outils permettant leur contrôle et leur protection.
A de très rares exceptions près les systèmes informatiques comportent des failles créées lors de leur conception. Le développement de réseaux informatiques complexes multiplie ces dernières, ce qui engendre une fragilisation de l'ensemble des composants de ces systèmes. Ces failles système sont considérées comme non malicieuses et non délibérées [ALRL04]: il s'agit d' anomalies laissant le sytème dans un état alternatif imprévu, produites inconsciemment par les développeurs. Les erreurs engendrées par le système lui même du fait de ses anomalies sont dites intrinséques et aléatoires. A l'inverse, certaines entités malicieuses (programmes informatiques ou êtres humains), appelées intrus, tentent d'exploiter ses failles dans le but de détourner un droit système afin de pouvoir altérer un service, l'arrêter, modifier une fonctionnalité ou une performance système, ou encore accéder à des données confidentielles [ALRL04]. Les erreurs produites par un intrus sont considérées comme logiques et malicieuses dans le sens où elles sont reproductibles et volontaires.
Dans le but de prévenir ces attaques, la communauté scientifique a élaboré une famille de systèmes de détection et de réponses des intrusions, les IDRS. A l'origine, ces systèmes étaient orientés sur la détection d'intrusion (Intrusion Detection system). Par la suite, le soucis de préserver l'intégrité des systèmes a conduit les différents acteurs du milieu à élaborer des systèmes de prévention (Intrusion Detection and Prevention System). De nos jours, ces systèmes s'orientent sur les problématiques de réponses automatiques face à une intrusion (Intrusion Detection and Reponse System). Cependant, pour plus de clarté, l'étude confondera les systèmes de prévention et de réponse dans l'acronyme IDRS.

Les premiers de ces outils étaient des systèmes statiques et centralisés. Or ces caractéristiques permettent à un intrus, après qu'il aie identifié les zones surveillées, d'éviter l'IDS. Après une intrusion au travers d'une faille du système, un intrus est ainsi en mesure d'effacer ses traces et de créer une nouvelle brèche dans une zone non ou mal surveillée par le système de détection; rendant ainsi ces futures attaques indétectables.

Pour palier à ces limitations, des nouveaux outils en cour de développement tentent d'employer une nouvelle approche à travers les systèmes multi-agents. Un Système Multi-Agent (SMA) est un paradigme d'architecture logicielle dans lequel les composants (les agents) sont considérés autonomes, parfois mobiles, à même de communiquer entre eux, et dont les fonctions et l'intelligence résident en général dans l'interaction entre chaque entité. L'un des avantages d'une telle approche réside dans le fait que l'intrus ne peut prédire la localité du système. De plus, cette dernière, de par ses capacités d'évolution et d'adaptation peut se déployer à grande échelle.

Ces propriétés ont ainsi conduit la communauté scientifique à envisager les SMA comme une approche pouvant résoudre certaines difficultées des IDRS. Celles-ci seront discutées dans ce document à travers la problématique que constitue **l'emploi d'un système multi-agents**

pour la détection d'instrusion volontaire et malicieuse dans un système supposé sans anomalie de conception.

Ce document se basera sur l'étude d'un corpus d'articles généraux et spécialisés traitant de certains types d'IDRS mais dont tous présupposent que **toute forme d'intrusion malicieuse a un impact sur le système.** Ce postulat sera repris car, en effet, toute activité malicieuse doit laisser une trace quantifiable sur le système pour que celle-ci soit détectable. En nous appuyant sur ces différents travaux, nous discuterons dans une première partie de la notion de trace, correspondant aux informations laissées par un intrus dans un système, ce qui nous conduira à analyser dans une seconde partie les différentes stratégies de détection qui peuvent être mise en oeuvre à partir de ces données. Nous aborderons alors la question des architectures de communication et de décisions permettant d'établir des réponses face à une situation d'intrusion. L'étude de ces différents aspects nous conduira à identifier certaines limites des IDRS actuels, nous reviendrons sur celles-ci en conclusion avant de proposer une piste de réflexion autour de la notion d'exo-protection.

1 Les données sur l'intrus

Toute activité sur un système consomme un certain nombre de ressources pouvant être des accès matériels (processeur, mémoire vive, etc) et des accès systèmes (planification d'exécution par le noyau, accès aux entrés et sorties, etc). A partir de ces ressources, des mesures peuvent être effectuées pour détecter l'éventuelle présence d'un intrus dans le système.

1.1 Acquisition des données

L'accès à ces ressources n'est pas à coût unique. En effet, dans un système informatique, toutes les ressources ne sont pas sur un même plan d'exécution, appelé couche. Un système se compose d'un empilement de couches. Selon le niveau de la couche à laquelle nous souhaitons accéder, le coût pour le système croît proportionnellement avec la profondeur. Dans l'idéal, pour détecter l'activité d'un intrus, il serait très efficace d'analyser le contenu de la mémoire vive en permanence car l'intrus y est forcément. Cependant cela nécessite des accès au noyau de très bas niveau qui ralentiraient considérablement le système [BFI+98].
L'acquisition des données sur les ressources s'effectue à travers une sonde logicielle. Dans le cas où la ressource n'est pas une valeur numérique directement exploitable, la sonde associe une valeur numérique ou symbolique à un état donné de la ressource. Pour un système de détection, le premier compromis à établir est de déterminer les données nécessaires à la détection en corrélation à leur coût d'acquisition.

Une alternative possible consiste à transposer l'intrus dans un autre espace que le système à protéger. Les traces ne sont alors plus dans les couches basses et peuvent être acquises à moindre coût. C'est l'approche dite du "pot de miel".

1.2 L'approche Honey Pot

L'approche Honey Pot [KG05] apporte une réponse à la problématique que représente le cout d'acquisition des données en construisant un système virtuel, vulnérable et identique au système à protéger pour y attirer l'intrus. Ainsi, la détection est plus facile puisqu'il s'agit d'un espace limité et parfaitement connu. Les ressources matérielles sont virtualisées. Il est alors possible d'associer des évènements à l'arrivée d'un intrus à la place des sondages fréquentielles sur des

ressources réelles. La procédure d'acquisition est ainsi moins consommatrice en ressource système, tout en détournant l'intrus du vrai système. Le compromis se situe entre la consommation d'un environnement virtuel complexe et le gain sur l'acquisition des données.

L'acquisition des données sur l'intrus est toujours un compromis entre les ressources nécessaires à cette acquisition et le besoin de fiabilité des données sur l'intrus.

A partir des données acquises, il s'agit alors d'être capable de proposer une solution de détection et d'identification de l'intrusion. Nous allons constater que celle-ci a également des liens étroits avec le système à protéger.

2 Stratégies de détection

Il existe deux principales approches, la détection d'anomalies reposant sur le principe qu'un système est compromis lorsqu'il change de comportement, et l'approche par signature où l'intrussion peut être identifiée car sa trace est connue [Das98].

2.1 La détection d'anomalies

Un intrus provoquant une erreur introduit un biais dans le comportement du système. Il s'agit alors d'être capable de modéliser le comportement normal d'un système pour en induire une éventuelle situation d'intrusion. Il existe un grand nombre de techniques et d'algorithmes permettant cela [KG05], tels que les réseaux bayésiens, la logique floue... Cette approche suit le raisonnement par l'absurde : s'il n'y a pas d'attaques, le système est prévisible ; sinon il est attaqué. Il n'est donc pas nécessaire de connaitre l'intrus pour le détecter, le problème réside sur le contrôle des fausses alertes qu'il faut alors minimiser lorsqu'une action ponctuelle mais légitime intervient. Le principal problème rencontré par les industriels et les chercheurs, avec les approches basés sur un modèle de comportement, est d'être capable d'obtenir un modèle du comportement correct du système. Lorsque le système est complexe, dynamique, ouvert et distribué, les informations ne sont en effet pas forcément accessibles et/ou constantes au cour du temps.
Un postulat fort est nécessaire dans cette approche, le système doit être sain lors de la construction du modèle par apprentissage. Si cela n'est pas le cas, l'intrus pourra être considéré par l'IDRS comme un composant du système. Il sera alors totalement invisible durant une procédure de détection.

2.2 La détection par signature

Tous les intrusions sont connues à priori ainsi que leurs signatures (code machine exécuté par le processeur)[ZL00]. Les fausses alertes sont alors quasi nulles. Il est nécessaire de maintenir une base de données à jour de l'ensemble des intrusions connues. Cette approche suppose deux choses : la base de donnée est invulnérable et est en permanence accessible par l'ensemble des agents décideurs. Si le coût des fausses alertes est alors minimisé, il est nécessaire d'employer le réseau pour accéder à la base de donnée. Il est alors nécessaire de disposer de suffisamment de bande passante pour transmettre les données qui éventuellement pourraient être compromises lors du transit.

Si l'approche par détection d'anomalies de comportement permet de détecter des types d'intrusions inconnues, il est souvent difficile de modéliser précisément le comportement d'un système

sans induire une rigidité de fonctionnement. En effet, la majorité des systèmes nécessitent une légère marge de manœuvre entre le modèle comportemental et leur comportement réel. La difficulté est de définir un modèle le plus prés possible du système tout en maintenant un taux de fausses alertes faible. L'approche par signatures d'intrusions, permettant un trés faible taux de fausses alertes, ne permet pas de détecter de nouvelles formes d'attaques.
Le choix de la stratégie de détection consiste à établir un compromis entre le besoin de fiabilité de détection (être confiant dans les alertes d'intrusions) et la prédictibilité du système. Si le système est totalement prévisible par sa rigidité de fonctionnement alors la stratégie de la détection par anomalie peut être trés performante. Dans le cas d'un système imprévisible, il faudra choisir une approche par signature au risque de ne pouvoir détecter de nouveaux types d'intrus. Il existe cependant des techniques basées sur un mécanisme d'apprentissage, ou de mesure de similarité pour l'identification de signatures suspectes permettant ainsi de réagir sur les nouvelles variantes d'intrusions connues.

A partir de la méthode de détection employée, il s'agit de définir à quelle échelle est opérée la détection. La détection locale peut être effectuée par une seule entité centralisée tandis qu'une approche globale nécessite une succession d'intermédiaires, pour ramener un problème de grandes échelles à la hauteur des détecteurs et décideurs responsables. Il est alors nécessaire d'établir des hiérarchies et des modèles de communication inter-agents à l'IDRS.

3 Architectures de communication et de décisions

Un intrus essaiera de contourner l'IDRS pour ne pas être détecté, ou encore de le compromettre. L'architecture détermine donc la vunérabilité de l'IDRS et la manière dont il prend des décisions ainsi que les réponses. L'architecture se compose de deux types de hiérarchie : celle de communication et celle de décision.

3.1 Architectures de communication

Les architectures de communication sont trés importantes puisqu'elles sont le système nerveux des IDRS. Selon leur type, elles favorisent et pénalisent différents aspects en terme de temps de réponse, de fiabilité des communications et d'adaptation à la mise à l'échelle.

- **Les architectures centralisées** sont caractérisées par un unique point qui centralise toutes les communications avant de les retransmettre aux agents destinataires. Cette technique est un routage centralisé et statique qui permet d'obtenir des vitesses de communication trés élevées. Cependant, cette architecture est pénalisée par la mise en échec de ce point. S'il ne fonctionne plus, aucun agent ne peut communiquer. Par ailleurs, le passage à l'échelle a nécessairement une limite de capacité, de part la nature centralisée de l'architecture.
  Il est cependant possible de découper le réseau en sous-réseau pour répartir la charge du noeud de routage.
- **Les architectures en arbre** [BFI+98] partitionnent la colonie d'agents en plusieurs clusters centralisés. Cette approche distribue l'unique point critique en plusieurs sous points critiques indépendants les un des autres, ramenant de fait le problème à une responsabilité locale à un cluster. Cependant, les agents restent fragiles au sein d'un même cluster. Un mécanisme d'élection périodique peut permettre d'alterner le point critique au sein du cluster, évitant ainsi de déterminer statiquement où se situe ce dernier [KG03].
  En supposant chaque noeud comme un point critique, il est possible de ramener le risque sur

chaque membre de la communauté.

– **Les architectures mesh** [ZL00] sont les mieux immunisées face aux points critiques puisqu'il s'agit de placer tous les agents sur un même plan et de les rendre responsable de leurs communications et du routage. Ils ont tous la même responsabilité et ne se distinguent pas les un des autres d'un point de vue structurel. Si cette approche répond aux problématiques de points critiques, elle est très coûteuse en terme de transmission réseau, notamment lors de la phase d'auto-organisation du protocole de routage. Celle-ci consiste en une procédure durant laquelle les agents définissent par vote, en fonction de leurs ressources propres, de l'attribution des rôles de routage (coordinateur, passerelle, routeur de bord, etc).

Finalement, les arbres sont fragiles mais performant en terme de vitesse de transmission et de consommation en bande passante du réseau. Inversement, les architectures mesh permettent de supprimer les points critiques. Il existe une architecture hybride qui est un compromis entre ces dernières :
Les architectures holoniques permettent à des agents d'un même niveau de communiquer entre eux. Cette approche permet de construire des compromis entre la nécessité de remonter l'information au sommet de la hiérarchie et la capacité de la colonie d'un même niveau à trouver une réponse indépendante, court-circuitant ainsi la chaine de commandement.

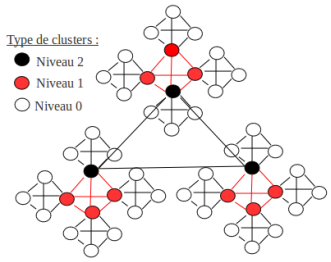

FIGURE 1 – Exemple d'architecture holonique

Le problème de dimensionnement de l'architecture vis à vis du système à protéger dépend principalement des ressources de communication disponibles. Finalement, les architectures holoniques paraissent les plus adaptées en terme de transmission, de fiabilité et de mise à l'échelle. Cependant, il est nécessaire d'établir un compromis autours du dimensionnement des clusters et du nombre de couches hiérarchiques. Ce compromis dépend de la taille du réseau initial. Dans le cadre d'un réseau dynamique (des noeuds entrent et sortent du réseau en permanence), ce compromis doit être continuellement réévalué. Si les propriétés des structures holoniques couvrent les principales problématiques, elles ne sont valides que si les clusters sont équilibrés. Il faudrait alors mettre en place des structures holoniques auto-adaptatives, capable d'auto-organiser les clusters et les couches hiérarchiques en fonction des communications réseaux. Il faudra alors justifier le choix entre une architecture mesh et une architecture holonique, qui toutes les deux nécessitent des protocoles d'auto-configuration.

Toutes les architectures de communications sont des compromis difficiles à établir lorsque le réseau est dynamique et que les performances doivent être garanties. Cette dernière contrainte est d'autant plus importante que les modèles de décisions sont dépendants de l'architecture de communication retenue.

3.2 Architectures de décisions

La décision peut être locale ou globale, individuelle ou collective.

Les décisions locales consistent à répartir la responsabilité de détection sur plusieurs agents. Dans cette configuration, l'agent doit disposer de tous les outils nécessaires à la détection. L'agent peut accéder à des outils distants à travers le réseau ou, s'il est mobile, se déplacer jusqu'à eux. Si le temps de détection est extrêmement court, les décisions strictement locales et non collectives ne peuvent pas corréler des évènements de zones différentes.
A l'inverse, les approches de décisions globales permettent de prendre en compte la possibilité d'attaques simultanées sur différents segments du réseau, un exemple d'implémentation est l'architecture Autonomous Agent For Intrusion Detection (AAFID) [BFI+98]. A travers la corrélation entre les analyses sur différents points du réseau, il est possible de mesurer l'ampleur de l'intrusion. Certaines attaques peuvent viser ponctuellement différentes zones du réseau pour éviter d'être détectées, cela est possible à travers le seuil de tolérance des agents locaux de l'IDRS. Ainsi plusieurs petites anomalies à différents endroits du réseau peuvent être considérées comme une même tentative d'intrusion. Cependant, le temps de transfert de l'information à travers la hiérarchie de communication et la quantité d'informations peuvent ne pas être acceptables vis à vis de la dimension du système.

Les stratégies de décisions individuelles consistent à nommer un agent responsable pour la prise de décision. Plusieurs agents décideurs peuvent composer la stratégie, mais ils ne prennent pas ensemble la décision. Les décideurs vérifient que les autres décideurs ne sont pas compromis. Cette stratégie est employée généralement dans les approches basées sur des architectures centralisées et certaines architectures en arbre (AAFID)[BFI+98].Les deux principales limites des décisions individuelles sont les problèmes de mise à l'échelle (lorsque le volume des données est trop important pour qu'une seule entité puisse tout analyser vis à vis des contraintes imposées) et le problème de fragilité du point critique (s'il est compromis, toute la zone sous sa responsabilité n'est plus protégée).
La prise de décision collective consiste à employer une procédure (de vote par exemple) permettant d'établir une décision par l'ensemble de la colonie d'agents. Il n'existe pas de dictateur dans la colonie dans le sens ou aucun agent n'a de droit de veto ni de valeur de vérité absolue. La fiabilité de la décision est accrue puisqu'un seul agent compromis ne devrait pas altérer dramatiquement la procédure [KG03].

3.3 Systèmes multi-agents mobiles

L'approche par agents mobiles [HWH+03, PDPP08] consiste à permettre à des agents de se mouvoir à travers le réseau pour effectuer leurs tâches. Cette approche à un certains nombres d'avantages vis à vis des problématiques soulevées précédement.

A travers le caractère mobile de cette approche, les éléments ne sont plus nécessairement rattachés à un système physique mais peuvent naviguer à travers le réseau. Cette spécificité peut permettre à des agents de se déplacer vers les ressources distantes plutôt que de faire transiter de gros volumes de données à travers le réseau [HWH+03]. Il est alors possible d'optimiser la consommation des ressources de communication à travers le rapport entre le poid des données sur le poid de l'agent lui-même mais aussi de protéger les données. En effet, si les données ne circulent plus sur le réseau, il n'y a plus de risque d'altérations malicieuses durant le transit. Cette protection des données est trés utile dans le cadre d'une détection par signature, où la base de

donnée supposée invulnérable peut devenir un bastion pour les agents. En effet, les agents ne demandent plus à la base de donnée si elle connait un intrus, ils se rendent directement sur sa plateforme pour effectuer leur recherche. Dans la perspective d'optimisation des ressources de communication, une variante est possible en permettant aux agents d'extraire et de transporter avec eux des segments de la base de donnée, réduisant la nécessité pour un agent de se rendre sur la plateforme de la base de donnée. Les agents sont alors ponctuellement spécialisés en fonction du contexte ou de leurs fonctions. Deux agents possédant un même extrait peuvent vérifier mutuellement s'ils sont compromis à travers un modèle de communication inter-agent [ZL00].
Les modèles de communication inter-agent peuvent permettre de construire des décisions collectives. Celles-ci peuvent être trés intéressantes lorsque la confiance en un agent est faible alors que la corrélation d'une communauté peut faire émerger des certitudes. Cette problématique se retrouve par exemple dans la détection par anomalie de comportement où le taux de fasses alertes est élevé. Certains algorithmes collaboratifs permettent en effet d'établir des procédures pour faire émerger des décisions et schémas collectifs, UAODV en est un exemple pour les problématiques de routage dans un graphe de type mesh [KG03].

Finalement, les systèmes multi-agents mobiles peuvent permettre de diminuer la consommation des ressources de communication ainsi que le risque pour l'IDRS d'analyser des données compromises. L'approche distribuée permet de diminuer les points critiques et les fausses alertes à travers des algorithmes collaboratifs, permettant de corréler les analyses de toute la communauté (ou d'un cluster déterminé) avant de prendre une décision.

Losqu'un intrus est détecté, il est potentiellement en train de créer des erreurs ou d'ouvrir de nouvelles failles sur le système. Il est nécessaire pour le système de déterminer une réponse pour éviter sa prolifération et la réussite de ses objectifs.

4 Méthodes de prévention et réponse du système

La réponse doit être efficace et rapide sous peine de ne plus pouvoir intervenir si l'intrus parvient à contourner l'IDRS. Une réponse a pour objectif d'interagir avec l'intrus par le biais d'une action sur le système lui-même. L'impact sur le système varie selon son type et son échelle de déploiement. Ainsi, une réponse sur-dimensionnée peut provoquer une perte de performance du système et donc de l'IDRS lui-même. Ceci pourrait servir l'intrus voir même être son but, il est donc important que la réponse soit en accord avec le danger de l'intrusion et des ressources disponibles pour y remédier.

Il existe trois principaux types de réponses allant de la protection à la résolution de la faille : l'isolation, la réinitialisation et la correction [ALRL04].

4.1 L'isolation

Pour éviter toute prolifération de l'intrus, il est nécessaire de bloquer la zone où il se situe. La granularité des données disponibles sur l'intrus (cf section 1) détermine l'échelle à laquelle l'isolation peut être appliquée. Elle peut intervenir au niveau du système en bloquant un shell d'accès, au niveau du segment réseau jusqu'à l'isolation d'un sous-réseau entier. L'isolation peut être active si l'IDRS décide de fermer différentes ressources, ou passive si les segments sains

décident de ne plus répondre à la zone compromise [ZL00]. Il s'agit d'une démarche visant à préserver l'intégrité du système.

4.2 La réinitialisation

Si la faille est ponctuelle et dépendante d'un état du système, la réinitialisation du système compromis va éjecter l'intrus et fermer la faille jusqu'à la prochaine apparition de la configuration sujette à la faille [KG05]. L'administrateur système sera alors à même de combler la faille avant que l'assaillant n'aie le temps de se réintroduire [ALRL04]. Cette réponse nécessite un temps de décision et de réponse bref pour que l'intrus n'aie pas le temps d'ouvrir une faille permanente.

4.3 La correction

La correction consiste à réparer la faille employée par l'intrus. Il s'agit d'une approche complexe puisqu'il faut une granularité d'information (cf section 1) très importante et une algorithmie permettant de corréler ces données avec une base de donnée de solutions. D'un point de vue ressource système, la correction nécessite des ressources de calcul et de communication qui doivent alors être disponibles. La procédure générale se compose d'un diagnostic permettant de construire un modèle de contexte d'erreurs, puis d'une réparation sous réserve qu'elle soit solvable [KG05].

La correction n'est pas toujours possible, par exemple si la faille est liée à la structure même du système. Dans ce cas aucune réponse ne pourra résoudre la faille, la correction ne peut pas être une solution seule. Elle doit toujours être en complément d'une isolation [ALRL04].
La correction est la solution la plus préférable si elle réussit et si sa consommation de ressources n'excède pas la réserve système. Dans le cas d'une réussite partielle ou d'un échec, les conséquences peuvent être plus importantes que l'intrusion elle-même. En effet, la correction nécessite des droits d'accès de super-utilisateurs pouvant effectuer des actions compromettantes pour le système si le diagnostic et l'évaluation ont été mal effectués. Dans le cas extrême, si l'intrus a pu compromettre l'IDRS, il a potentiellement les droits complets sur le système. L'IDRS est donc considéré comme invulnérable, cette hypothèse est formulé dans[PDPP08].

L'isolation et la réinitialisation ne résolvent pas le problème d'intrusion mais permettent de protéger l'intégrité du système. Elles sont peu consommatrices de ressources mais nécessitent que le système puisse fonctionner sans certains composants. Les approches de résolution sont idéales mais peuvent également créer la faille la plus importante de tout le système : l'accès au droit de super-utilisateur.

Conclusion et perspectives

Les différents travaux étudiés ont montré la pertinence du choix des systèmes multi-agents comme architecture générale des IDRS. Leurs propriétés d'adaptation et de répartition des composants permettent des mises à l'échelle et la tolérance aux fautes. Le caractère mobile permet d'améliorer la protection des données utilisées par l'IDRS et de rendre l'intrusion moins aisée par le déplacement aléatoire (vu de l'exterieur du système) des sondes d'acquisition.
L'acquisition des données sur un intrus s'effectue à travers des sondes systèmes, malgré la virtualisation de l'espace, l'opération d'acquisition consomme des ressources. Un compromis doit être établit entre la qualité des informations et les capacités physiques du système. La qualité des informations et le fonctionnement du système sont au coeur de la problématique de détection qui

peut être effectuée par signature d'intrusion ou par la mesure d'un biais du comportement. Si la détection par anomalie du comportement peut permettre de détecter les nouveaux types d'intrus, la possibilité de modéliser le comportement du système avec une faible marge de tolérance est trés difficile. La mise en place d'une hiérarchie basée sur des agents mobiles peut permettre une détection locale et globale. La suppression des points critiques par les systèmes multi-agents mobiles améliore la fiabilité générale des IDRS en terme de détection, de communication, de décision et de réponse.

Limites retenues

Finalement, les principales limites des IDRS sont liées à leur présence sur le système à protéger. Consommant des ressources pour l'acquisition des données et son propre fonctionnement, l'IDRS est soumis à des compromis à tous les niveaux. Ces compromis sont difficiles à établir au vue de la complexité des paramètres dont ils dépendent. La coexistence des communications de l'IDRS et du système à protéger sur un même réseau pose des problématiques difficiles pour garantir une fiabilité et protection de l'IDRS lui-même.

Une approche basée sur un IDRS possédant ces propres ressources, en terme de puissance de calcul, de capacité réseau et de sondes, résoudrait un certains nombres de limites illustrées ci-dessus.

Proposition : l'exo-protection

Le principe de l'exo-protection est que l'IDRS doit posséder son propre système matériel et son propre réseau de communication. Les coûts de ces derniers ne reposent plus sur le système à protéger et l'acquisition ne s'effectue plus à partir d'une couche du système mais depuis l'extérieur. Le développement actuel des réseaux de capteurs sans fils [SB10] (IEEE 802.15.4) permettent le développement de réseaux de communication de type mesh à trés faible consommation d'énergie. En effet, de nombreux projets liés à l'internet des objets visent à développer des réseaux de sondes basés sur des microcontrôleurs en réseau ad-hoc sans fils telque Contiki OS [DGV04] ou tinyOS.

Une architecture de communication dédiée

Si un réseau sans fils ad-hoc est dédié à l'IDRS, tous les agents peuvent effectuer des rapports à tous les autres agents. Il n'est plus nécessaire de préserver de la bande passante pour le fonctionnement du système à protéger. L'IDRS peut ainsi utiliser toute la bande passante en permanence au sein d'un même cluster (groupe d'agents à porter de communication les uns des autres). Les communications peuvent être effectuées sur un même cluster dans une fenêtre temporelle dédiée à chacun (mode beacon MAC [SB10]). Les communications broadcastées en mode beacon (totalement synchrone) peuvent être soumises à des contraintes temps-réel dures permettant d'établir des garanties sur les temps de réponses. La mise en place d'algorithmes de détection globale et de décision collective peut être effectuée sans risque pour les communications du système à protéger.

Une détection par signature protégée

Le système de protection étant sur un réseau fermé, la nécessité de considérer la base de donnée des signatures de virus comme invunérable est possible (sous réserve que l'intrus n'est aucun accès physique au réseau sans-fils ad-hoc). Les différentes techniques pour la détection par signature peuvent être développées sans risque.

Acquisition des données sans coût sur le système

Il est possible de mettre en place des agents sondes matérielles branchés directement sur les

bus processeurs des cartes réseaux pour surveiller les informations transmises. L'IDRS a ses propres ressources matérielles, l'acquisition n'a donc aucun impact sur le système à portéger. L'impact énergétique pourrait être minimal ou nul puisque la communauté de chercheurs autours de l'internet des objets s'oriente sur des objectifs de consommation inférieurs à 10µW par heure (permettant une alimentation continue sur un an par une pile AAA).
L'exo-protection pourrait être en mesure de résoudre certains des problèmes actuels liés à la coexistence du système à protéger et de l'IDRS sur un même matériel. A ce titre, une étude plus profondie pourrait être mis en place cependant, cette approche ne sera jamais une solution générique par ses contraintes de mise en place et son coût financier éventuel.

Finalement, le milieu de la recherche et les industriels ont avancé de nombreuses études ayant abouties sur différents modèles et architectures. Cependant, tous les articles rencontrés soulèvent des problématiques de compromis à établir. La proposition effectuée dans ce document n'échappe pas à la règle. L'élaboration d'une classification des différents choix d'architectures et de modèle de détection/réparation en fonction des différents contextes d'utilisation et de leurs contraintes associées pourrait s'avérer très utile. A partir de cette étude, des propositions de standardisation pourraient être envisagées afin d'aboutir à des déploiements de solutions génériques d'IDRS en fonction d'un profil système défini par le standard. L'internet des objets pourrait apporter des pistes de réflexion à travers la définition de standards basés sur des profils de contraintes dépendant d'un certains de nombre paramètres communs au IDRS.

Remerciement

Nous souhaitons remercier Cédric Herpson pour ses conseils et ses nombreuses relectures.

Références

[ALRL04] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, 1:11–33, 2004.

[BFI+98] J. S. Balasubramaniyan, Garcia J. O. Fernandez, D. Isacoff, Eugene H. Spafford, and Diego Zamboni. An architecture for intrusion detection using autonomous agents. In ACSAC, pages 13–24, 1998.

[Das98] Dipankar Dasgupta. Immunity-based intrusion detection system: a general framework, 1998.

[DGV04] Adam Dunkels, Bj? Gr?nvall, and Thiemo Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. Local Computer Networks, Annual IEEE Conference on, 0:455–462, 2004.

[HWH+03] Guy Helmer, Johnny S. K. Wong, Vasant Honavar, Les Miller, and Yanxin Wang. Lightweight agents for intrusion detection. Journal of Systems and Software, 67(2):109 – 122, 2003.

[KG03] Oleg Kachirski and Ratan Guha. Effective intrusion detection using multiple sensors in wireless ad hoc networks. Hawaii International Conference on System Sciences, 2:57a, 2003.

[KG05] Peyman Kabiri and Ali A. Ghorbani. Research on intrusion detection and response: A survey. International Journal of Network Security, 1:84–102, 2005.

[PDPP08] Nita Patil, Chhaya Das, Shreya Patankar, and Kshitija Pol. Analysis of distributed intrusion detection systems using mobile agents. In Proceedings of the 2008 First International Conference on Emerging Trends in Engineering and Technology, pages 1255–1260, Washington, DC, USA, 2008. IEEE Computer Society.

[SB10] Zach Shelby and Carsten Bormann. 6LoWPAN: The Wireless Embedded Internet. Wiley Publishing, 2010.

[ZL00] Yongguang Zhang and Wenke Lee. Intrusion detection in wireless ad-hoc networks. In Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00, pages 275–283, New York, NY, USA, 2000. ACM.