OSAE : Automatisation de la sécurité sur REAUMUR

Grégoire Moreau, Laurent Facq

OSAE : Automatisation de la sécurité sur REAUMUR

Grégoire Moreau
Cellule SSR – Département TIC du PRES Université de Bordeaux
351 cours de la libération 33405 TALENCE
g.moreau@pres.u-bordeaux.fr

Laurent Facq
Cellule SSR – Département TIC du PRES Université de Bordeaux
351 cours de la libération 33405 TALENCE
l.facq@pres.u-bordeaux.fr

Résumé

Le système, OSAE (Outil de Sécurisation Automatique sur Événements), traite depuis 2005, pour les besoins du réseau de collecte régional inter-universitaire REAUMUR, la plupart des problèmes courants de sécurité : FTP Warez, botnets, vers, redirecteurs de spam, abus d'utilisation (P2P, ...).

Le traitement est automatique dans la plupart des cas, ce qui accroît globalement la sécurité du réseau, limite les entraves éventuelles à son bon fonctionnement et nous fait gagner un temps précieux. Ce traitement automatique n'engendre qu'un faible taux de faux positifs et conduit des actions :

- *mise en quarantaine de machines ;*
- *avertissement des correspondants par e-mail et suivi d'incident (interactions avec plus de 200 contacts).*

Lorsque le système ne peut prendre de décision, un administrateur sécurité intervient, son action se limitant à désigner l'action à effectuer par le système pour un problème donné.

Mots clefs

Sécurité, Réseau de collecte, Automatisation, IDS, Quarantaine, Contres mesures, Détection d'intrusion.

1 Introduction

La cellule SSR (Services et Sécurité Réseau) du département TIC du PRES de Bordeaux gère les différents services réseau associés au réseau de collecte d'Aquitaine Nord REAUMUR (Réseau Aquitaine des Utilisateurs des Milieux Universitaires et de Recherche). Cette cellule gère en particulier les questions de sécurité sur le réseau REAUMUR.

Les menaces pour un réseau de collecte régional sont concrètes, des abus massifs, des piratages et autres virus peuvent potentiellement diminuer les performances des cœurs de réseaux ou saturer les liaisons opérateurs. Il y a quelques années, ces menaces pouvaient même écrouler un routeur haut de gamme de l'époque.

Ces menaces sont de plus en plus automatisées, faites pour se répandre à grande échelle le plus rapidement possible. Face à ces dernières, il existe heureusement différentes solutions plus ou moins onéreuses et efficaces. Du simple pare-feu *stateful*, à l'IPS (Intrusion Prevention System) dernier cri, différentes technologies permettent de réduire les risques inhérents au fait d'être raccordé directement à Internet.

2 Historique

Dès 1994[1], dans la préhistoire des systèmes de détection d'intrusion, nous avions commencé le développement et l'exploitation d'un système primitif de détection d'intrusion basé sur le code source (modifié par nos soins) d'un analyseur réseau (type tcpdump) et fonctionnant par analyse protocolaire jusqu'au niveau applicatif. L'accès Internet de REAUMUR était alors de 2Mb/s, les scans étaient très rares et presque à chaque fois signe d'un piratage effectif. Ils donnaient donc lieu à une analyse manuelle presque systématiquement. Les machines compromises ou ayant un comportement abusif étaient mises en quarantaine manuellement.

En complément de ce système, nous effectuions des audits de sécurité automatiques de l'ensemble des machines du réseau à l'aide du logiciel Satan. Le fruit de ces analyses était diffusé aux correspondants techniques des sites.

2.1 Évolution des outils internes

Entre 1995 et 2003, en parallèle[2] à l'augmentation du rythme des scans, l'évolution des vers Internet et la montée du P2P, nous avons développé quelques scripts Perl afin de faciliter l'analyse et le traitement des informations collectées. La gestion des incidents à la main, pour quelques cas par semaine, restait raisonnable, sachant que pour chaque cas il était nécessaire de :

- mettre en place la quarantaine (modification de la configuration des routeurs pour bloquer le trafic de la machine posant problème) ;
- prévenir le correspondant technique concerné par email et lui fournir une description du problème ;
- procéder à la réouverture sur demande de sa part par email.

Afin de prévenir également l'utilisateur final que sa machine était en quarantaine et qu'il devait contacter au plus vite son administrateur local, un mécanisme de *policy-routing* (« routage politique ») et de tunnel GRE fut mis en place pour rerouter les requêtes web des machines en

quarantaines sur une page donnant une information statique sur les causes probables (virus ou abus). Des exceptions furent mises en place pour permettre aux machines compromises d'effectuer néanmoins leurs mises à jour (système, anti-virus) et de récupérer les principaux outils de désinfection via le réseau.

D'un point de vue logiciel, pour gagner du temps et conserver des outils à jour, nous avons remplacé le couple [IDS maison + Satan] par [Snort + Nessus].

2.2 Limites face aux menaces modernes

L'évolution des menaces et du périmètre du réseau ont rendu, au fil des années, de plus en plus complexe et consommatrice de temps la gestion manuelle des problèmes de sécurité que nous détections.

En effet, les menaces modernes sont désormais fortement automatisées pour exploiter les failles se trouvant sur de nombreuses machines.

Il est devenu clair que dans des situations de crise (les différents vers des années 2003-2004 par exemple) nous passions beaucoup trop de temps à bloquer et débloquer des machines suivant les désinfections effectuées par nos correspondants sur sites.

3 Projet

En 2004, le service disposait depuis un an d'un élève ingénieur ENSEIRB[1] en apprentissage en la personne de Grégoire MOREAU. Ce dernier devait réaliser un projet de mémoire d'une durée de 15 à 18 mois. Il a donc été décidé de lui confier la responsabilité du projet visant à améliorer la gestion de la sécurité. Une démarche projet classique a été utilisée, nous ne la détaillerons pas ici[3].

3.1 Objectifs

Les objectifs du projet sont multiples :

- faire gagner du temps au service sur les traitements courants de sécurité ;
- améliorer la sécurité durant les périodes non ouvrées ;
- déléguer la gestion des levées de quarantaine, et offrir une visibilité aux correspondants de sites.

3.2 Contexte et contraintes

REAUMUR étant un réseau de collecte inter-établissements, nous n'avons généralement pas la maîtrise des réseaux locaux au plus près des machines, ce qui interdit, par exemple, la mise en quarantaine sur un vlan particulier.

Nous ne disposons pas non plus de bases d'informations sur les rôles des machines connectées et sur leur comportement réseau habituel.

Ce contexte nous oblige donc à utiliser des mécanismes robustes et génériques.

[1]ENSEIRB: École Nationale Supérieure d' Électronique, Informatique et Radio-télécommunications de Bordeaux

3.3 Choix de solution

Les différents choix qui s'offraient à nous pour répondre aux objectifs étaient les suivants : l'achat d'un outil du commerce (IDS ou IPS), l'utilisation de logiciels libres, le développement maison.

La particularité principale de la solution que nous cherchions était le fait de pouvoir disposer d'une part d'un outil d'automatisation, d'autre part d'un système permettant d'interagir efficacement avec nos correspondants.

Pour des raisons économiques et politiques notre choix s'est porté sur des logiciels libres que nous avons complétés par des développements internes afin d'en faire une solution sur mesure.

Les logiciels libres en questions sont : Snort et Clamav. Pour les besoins du développement les langages Perl et PHP (associé à Smarty) ainsi que le SGBD PostgreSQL sont utilisés.

3.4 Comparaison des approches [IDS + Quarantaine + Scanner] et IPS

A plusieurs reprises au cours de ce projet, nous nous sommes interrogés sur notre approche (un IDS, analysant surtout les flux sortants, permettant de mettre en quarantaine les machines problématiques complété d'un scanner de vulnérabilités), pour la comparer à d'autres approches possibles et en particulier aux systèmes, apparus plus récemment, de prévention d'intrusions (IPS) destinés à intercepter au vol les paquets dangereux. Voici nos principaux arguments en faveur de notre approche.

- Coût pour fiabilité & performance : un IPS intercepte les paquets jugés dangereux ou anormaux selon la politique définie. Il doit donc être placé en coupure sur l'arrivée réseau ce qui implique obligatoirement : un mécanisme de haute disponibilité, un niveau de performance en rapport avec le débit du réseau même en cas de déni de service, d'où un coût d'investissement important.
  A contrario, un IDS n'impose pas ces contraintes (pas besoin d'être en coupure) d'où un coût réduit. D'autant plus que dans le cas d'un gros réseau, ne pouvant travailler très finement, nous pouvons tolérer quelques pertes de paquets sans dégradation notable - les problèmes que nous sommes en mesure de détecter générant un nombre non négligeable de paquets (problèmes « bruyants »).
- Problème de grain et d'échelle : pour tirer au mieux parti d'un IPS, une connaissance fine et à jour des caractéristiques des équipements protégés est nécessaire. Dans notre contexte de réseau de collecte, cette contrainte est difficile à respecter car on ne peut se substituer à une gestion fine, sur mesure, qui ne peut-être pratiquée qu'au plus près, à l'échelle d'un unique site.
- Bruit & faux positifs : s'intéressant aux tentatives d'intrusion et non aux intrusions réussies, un IPS s'inquiète pour les innombrables attaques potentiellement dangereuses (éventuellement diminué par recoupement avec un inventaire des versions de

services vulnérables) alors que nous ne recherchons que les intrusions effectives qui ne se détectent essentiellement (pour être fiable) que sur les flux sortants en utilisant un jeu de règles relativement restreint : il y a donc beaucoup moins de faux positifs.

- Obligation de corriger : un IPS peut fonctionner sans obliger à corriger les failles de sécurité ni mettre les machines en quarantaine, ce qui peut être vu comme un confort. Toutefois comme un IPS ne protège qu'en un ou quelques points du réseau, un portable infecté placé en aval pourra profiter des failles existantes. Les audits (Nessus), même si nous ne les jugeons pas suffisamment fiables pour déclencher des blocages automatiques, permettent de corriger ces failles et sont complémentaires de notre approche IDS+quarantaine.
- Éducation des usagers : la mise en quarantaine, en cas d'abus ou de compromission, a des vertus éducatrices auprès des utilisateurs, ce que les IPS ne vont pas forcément apporter en filtrant spécifiquement les sessions et paquets problématiques.
- Prévention d'intrusion par scanner : la partie prévention est assurée par les audits de sécurité (Nessus) réguliers permettant de détecter les vulnérabilités connues qui servent également de base aux signatures IPS.

4 Architecture et fonctionnement

On peut distinguer 4 niveaux principaux dans le système : l'*événement de sécurité*, le *problème de sécurité*, *les prises de décision et l'exécution de ces dernières* (voir Figure 2).

Pour les traitements automatiques nous avons choisi le langage Perl qui offre un niveau de performance raisonnablement élevé, et permet un développement rapide et souple.

4.1 Événement de sécurité

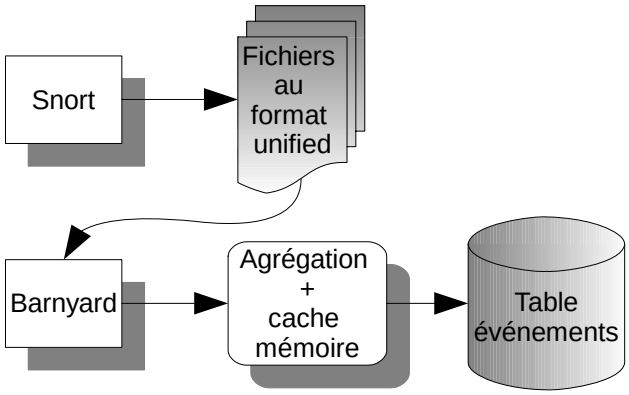

Figure 1: Traitements en vue d'obtenir les événements de sécurité agrégés

L'*événement de sécurité* constitue la brique de base du système, il correspond actuellement principalement aux alertes de Snort (*pattern matching* et scans de ports).

Quelques exemples d'événements : recherche de fichier sur edonkey, connexion IRC, scan de port 135, connexion ftp sur port non standard, connexion ssl sur port non standard, connexion à un tracker bittorrent, etc.

Une première agrégation est effectuée pour qu'une entrée dans la base de données corresponde à un événement distinct. Les nouvelles occurrences de ce dernier incrémentent uniquement un compteur, et seules les informations sur la première et la dernière occurrence sont conservées dans la base.

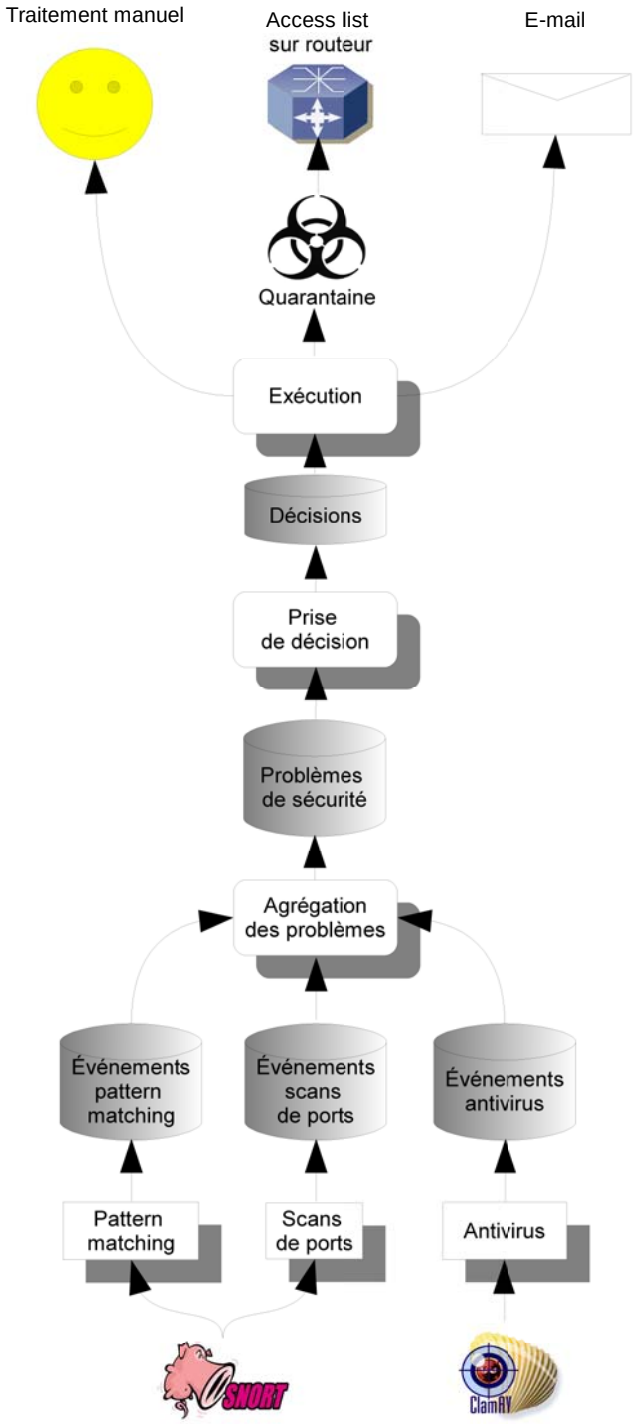

Figure 2: Schéma d'ensemble du système

4.1.1 Format de données pour Snort

Au niveau de Snort, nous avions testé le format de sortie « base de données ». Ce dernier était relativement inadapté car bloquant pour Snort lorsque la base ne suivait pas. Il y avait une quantité gigantesque d'entrées dans la base de données, ce qui la rendait très lourde à exploiter ; nous avons donc opté pour le format de données binaires « unified ». Il s'agit du format le plus efficace. De cette manière on minimise les risques de perte d'information au niveau de Snort qui est le seul élément du système où une contrainte « temps réel » s'impose.

Un petit outil, barnyard, permet de reconvertir ces fichiers « unified » sous l'un des formats classiques de Snort, pour ensuite pouvoir l'exploiter avec des outils courants. A noter ici que cela autorise un traitement asynchrone par rapport au flux réseau. Le format de sortie choisi est le format « log_dump », il donne la description de l'alerte, les détails sur les entêtes du paquet posant problème, et le contenu sous forme hexadécimale. Ces données sont envoyées dans un fichier FIFO qui est lu par le premier script d'analyse du système, ce dernier insérant les événements dans la base de données.

Pour éviter de surcharger la base de données de requêtes, des caches mémoires sont utilisés dans tous les scripts et les mises à jour de la base se font à intervalle raisonnable (de l'ordre de la minute).

4.2 Problème de sécurité

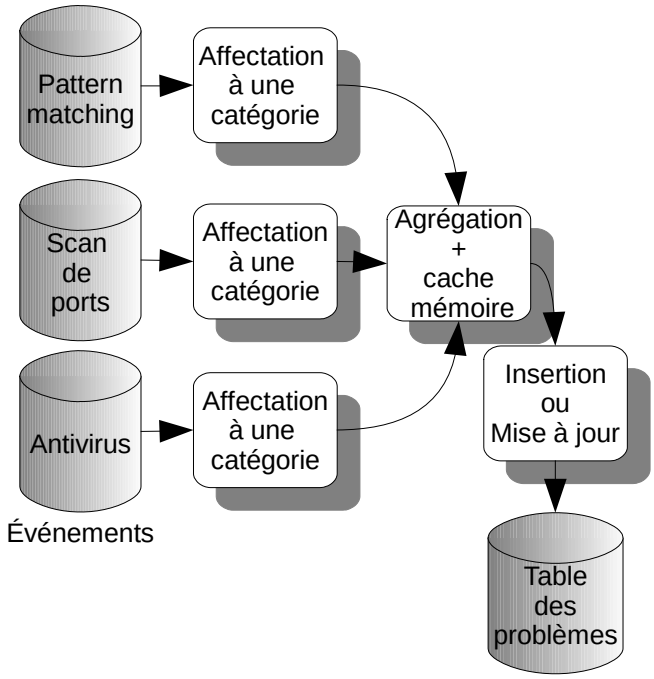

Figure 3: Traitements en vue d'obtenir les problèmes de sécurité agrégés

Les *problèmes de sécurité* sont issus d'un à plusieurs *événements*, ils sont plus génériques que ces derniers. En effet pour un événement « recherche de fichier sur edonkey » on va associer le problème de type « P2P », ou encore pour un « scan de port 135 » on associera un problème de type « WORM ». S'il y a un cumul d'événements pour une même IP, disons l'utilisation de bittorrent, edonkey, kazaa et gnutella, un seul problème de type P2P sera créé.

Les types de problèmes correspondent à différentes catégories (P2P, WORM, FTP WAREZ, ROBOT IRC, etc.).

4.3 Choix des actions

Aux catégories sus-mentionnées, nous avons associé des actions, comme le fait d'envoyer un mail aux correspondants concernés, de mettre en quarantaine la machine posant problème ou encore de se mettre en attente d'une validation, voir d'une analyse, par un administrateur sécurité de notre service. De manière à éviter les mises en

quarantaine de serveurs pouvant avoir un comportement assimilable à un problème (serveur de mail ou redirecteur de SPAM ?) une liste d'adresses IP (whitelist) a été définie. Pour ces adresses le système n'effectue qu'un envoi de mail d'avertissement.

Nous avons parlé du fait que le système dans certains cas attendait la validation ou l'analyse d'un administrateur. En effet, nous savons de par notre expérience des règles Snort que nous utilisons, que certaines génèrent des faux-positifs ; il faut alors valider si le problème est effectif ou non. De plus, nous avons fait le choix de laisser remonter des alertes pour des scans de ports non déterminés, qu'il reste alors à analyser manuellement. Nous pouvons par la suite ajouter des comportements adaptés aux nouveaux cas que nous trouvons. Enfin il faut garder en tête que seule l'analyse est faite manuellement, tout le reste (mail, quarantaine) reste traité automatiquement.

4.4 Structure de la base de données

Chaque table est rattachée à une fonction : les événements sont dans des tables indépendantes suivant leur type (Snort, antivirus, portscan).

Les *problèmes* sont stockés dans une table unique et rattachés aux événements par des tables de jointure.

Différentes tables régissent le comportement du système, l'état des règles Snort, et contiennent les données concernant des traitements particuliers (machines en « whitelist », exceptions diverses - nous y revenons dans la partie 7 Évolutions).

L'intérêt de disposer de la configuration du comportement du système dans la base de données est de pouvoir intervenir très simplement, en cours de fonctionnement, pour modifier les traitements.

4.5 Interface utilisateur

L'interface est relativement basique, les informations affichées sont différentes en fonction de l'authentification. Les correspondants ne disposent que d'une visibilité sur leur réseau. Les administrateurs sécurité du service ont une visibilité sur tous les sous-réseaux. Les utilisateurs finaux n'ont qu'une information synthétique concernant les problèmes associés à leur IP.

Pour ouvrir notre système au maximum d'établissements, nous authentifions les correspondants, soit grâce à la fédération d'identités Shibboleth des établissements Aquitains, soit à l'aide de certificats X.509 (CNRS et CRU).

4.6 Mise en œuvre du filtrage

Selon les actions déclenchées par les problèmes de sécurité, le système rafraîchit autant que nécessaire la liste des adresses IP à mettre en quarantaine. Cette liste est lue à intervalles réguliers (quelques minutes) pour mettre à jour les filtres et redirections nécessaires sur nos routeurs.

Ces filtres sont placés de façon à ce que l'analyse du trafic réseau des machines mises en quarantaine soit toujours effectuée. Ceci permet souvent de continuer à détecter les problèmes indiquant qu'une machine est toujours compromise sans avoir à la débloquer.

5 Utilisation du système

Nous allons maintenant passer en revue les 3 niveaux d'utilisation du système en décrivant les opérations courantes que sont amenés à effectuer les différents intervenants.

5.1 Point de vue des administrateurs d'OSAE

Les administrateurs d'OSAE (actuellement, les 2 auteurs) sont en copie de tous les courriers envoyés par le système aux correspondants, ce qui construit simplement un journal des problèmes majeurs permettant de suivre individuellement les activités importantes du système.

Des visites régulières dans l'interface (en pratique entre 5 et 10 par jours) sont nécessaires pour re-qualifier, si besoin, les problèmes ambigus.

Un mécanisme de marquage facilite ce suivi à plusieurs intervenants :

- les problèmes qui n'ont jamais fait l'objet d'un contrôle manuel apparaissent sur fond orange pour attirer l'attention ;
- après contrôle manuel, un bouton permet de cocher le problème comme ayant été vu (le système retient l'identité du dernier « contrôleur ») ;
- lorsque le problème a évolué après le dernier contrôle manuel, il passe sur fond jaune pour signaler qu'un nouveau contrôle serait utile.

Des alertes sonores se déclenchent régulièrement pour nous rappeler que des problèmes nouveaux et potentiellement sérieux sont en attente de validation.

L'envoi d'emails associés à un problème détecté est grandement facilité : il suffit de remplir un champ de saisie. Toutes les autres informations liées aux problèmes sont automatiquement communiquées aux bons correspondants, l'éventuelle suite du dialogue se faisant par messagerie classique.

5.2 Point de vue des correspondants techniques des sites

Les correspondants des sites se rendent dans l'interface OSAE, soit suite à la réception d'un courrier en provenance du système, soit à la demande d'un de leurs utilisateurs se trouvant en quarantaine.
Ils ont une vue limitée aux informations suivantes relatives aux problèmes : type/description, IP/ports source/destination, date premier/dernier, historique des acquittements/détections (avec commentaires éventuels).
Une fois les problèmes réglés (machines désinfectées, utilisateurs châtiés, ...) ils peuvent acquitter les problèmes en y adjoignant un commentaire sur leur action :

- dans le cas d'une quarantaine, l'acquittement va libérer la machine. Si le problème se renouvelle par la suite, la quarantaine se remettra en place automatiquement avec un envoi de courrier ;

- dans le cas d'un simple envoi de mail d'avertissement, l'acquittement va réarmer l'envoi de mail en cas de re-détection du problème. L'administrateur de site peut donc à juste titre laisser certains faux problèmes non acquittés pour éviter de nouvelles alertes inutiles.

5.3 Point de vue de l'utilisateur final

L'utilisateur final dont la machine est mise en quarantaine est « immédiatement » averti du blocage via la redirection de ses requêtes web sur une page spéciale d'information (« votre accès internet est bloqué ... »). Cette page lui permet de se rendre sur l'interface OSAE et lui donne la possibilité de débloquer sa machine. Cette possibilité, limitée à une fois par 24h pour éviter les abus, permet d'éviter les situations de blocage trop longues en cas d'absences des correspondants compétents et si les utilisateurs finaux sont en mesure de résoudre eux-mêmes le problème.

6 Retour sur expérience

6.1 Difficultés rencontrées

Bien que les plus grosses difficultés aient été surmontées de façon pragmatique pour aboutir à un système fonctionnant de manière raisonnablement fiable, il demeure que certains points délicats restent à affiner dans le futur. Certains aspects nous paraissent néanmoins intrinsèquement insolubles de façon parfaitement rigoureuse et définitive. Ils se résument presque tous à condenser *l'information de façon fidèle*. Il s'agit de réduire la quantité d'éléments d'information (par agrégation, filtrage, expiration, ...) pour que le système soit exploitable, ceci devant se faire sans perte de qualité d'information.

- Limiter les faux positifs et faux négatifs : pour limiter les mauvaises décisions du système, nous avons choisi de classer les problèmes en 3 catégories : 2 pour les cas pratiquement certains (le problème est avéré, le problème n'en est pas un) et une pour les problèmes ambigus nécessitant une analyse manuelle. Cette dernière classe est très importante et résulte de notre volonté de ne pas chercher coûte que coûte à tout traiter de manière automatique : c'est impossible et ce serait faire fausse route. En effet cela rendrait passifs les gestionnaires du système alors que la veille sécurité est toujours nécessaire.
- Délimitation et durée de vie des problèmes : il n'y a pas de moyen fiable et général permettant de déterminer en un temps borné qu'un problème (machine compromise par exemple) est réglé, ce qui pousse à garder le plus longtemps possible les informations sur les problèmes. Mais on ne peut pas non plus garder indéfiniment tous les problèmes dans l'interface car cela deviendrait illisible et inutilisable. Il est donc nécessaire de purger les problèmes de façon quelque peu arbitraire quand ceux si n'ont pas été rafraîchis par des événements récents.
- Gestion des exceptions : de manière générale, toute exception se comporte comme un filtre permettant de diminuer les problèmes à prendre en compte. Deux problèmes se posent :
  - la durée de vie de ces exceptions : elles ne doivent pas être considérées comme éternelles (les machines évoluent, leurs rôles changent) ;
  - leur finesse : suffisamment fines pour ne pas cacher des problèmes mais assez larges pour ne pas passer son temps à analyser un même faux problème récurrent.
- Classifier les problèmes : comme on peut le voir sur la Figure 4, les moyens de détection que nous utilisons ne travaillent pas tous sur le même plan : certains vont détecter le but de la compromission, d'autres les moyens de conserver un accès sur la machine pour le pirate (contrôle). Il est donc difficile de déterminer de façon automatique toutes les caractéristiques des problèmes, ce qui est gênant pour établir des statistiques et nécessite l'expérience des administrateurs sécurité pour retrouver, si nécessaire, les éléments manquants.

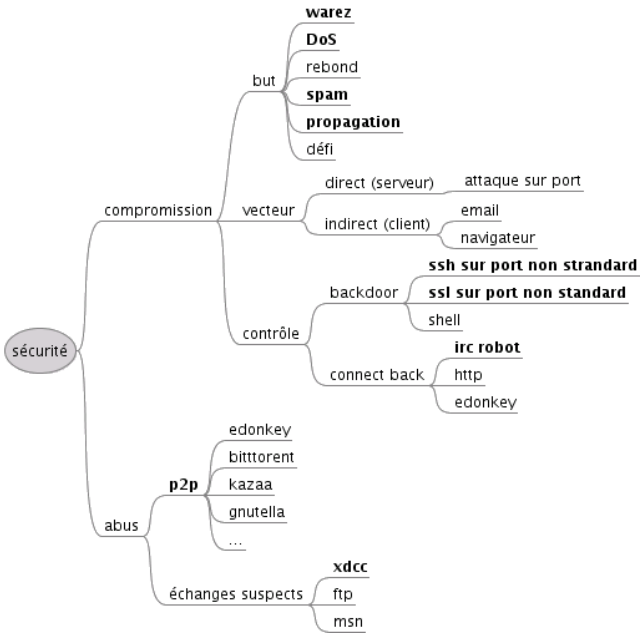

Figure 4: Essai de cartographie des problèmes courants (en gras ceux traités automatiquement)

- Classification des scans : les scans détectés par Snort se présentent de façon brute sous la forme de quadruplet (IP/port source, IP/port destination). Il est nécessaire de classifier ces scans et de choisir des seuils permettant de reconnaître certains problèmes. Comme on ne peut pas attendre la fin d'un scan pour réagir (durée inconnue *a priori* et nécessité de réagir dès que possible), il faut classifier et reclassifier en permanence avec les dernières informations reçues.

6.2 Mise en production progressive

Du fait que le système allait effectuer des blocages automatiques, il était important préalablement d'en valider le bon fonctionnement.

Une première étape de test a validé le fait que le système ne ratait pas de problèmes, ni n'en inventait...

La deuxième étape a été de faire valider l'interface par des testeurs.

La troisième a été de valider la montée en charge, durant la période de rentrée, avec les débits en augmentation. Nous avons ainsi pu constater que les scripts Perl ne consommaient que très peu de ressources.

6.3 Bilan / Résultats

Depuis sa mise en service en 2005, OSAE a effectué près de 850 mises en quarantaine, et envoyé plus de 3200 emails.

Concernant les mises en quarantaine, 86% ont été effectuées de façon totalement automatique contre 14% après analyse manuelle.

Dans des périodes d'attaques relativement fortes, plusieurs dizaines de machines ont pu être mises en quarantaine automatiquement, sans que le moindre traitement manuel ne soit nécessaire.

Il est clair que nos objectifs sont remplis : nous gagnons du temps, les périodes non ouvrées sont couvertes et les correspondants sont désormais autonomes.

7 Évolutions

Nous décrivons maintenant les évolutions consécutives à la mise en production initiale.

7.1 Évolutions déjà mises en oeuvre

Entre la mise en production et aujourd'hui, les principales améliorations sont les suivantes.

- Post traitement : afin d'éviter la multiplication des règles Snort tout en diminuant le nombre de cas à traiter manuellement, un post-traitement est effectué sur les contenus (payload) des alertes, de manière à pouvoir, avec une règle Snort générique (détection du protocole Bittorrent par exemple), déterminer s'il y a un problème effectif ou pas (pour Bittorrent en fonction du paramètre « Host » dans les connexions au tracker).
- Affinage de la « whitelist » : pour éviter de mettre en quarantaine des serveurs ayant naturellement un comportement pouvant ressembler à un problème (serveur de mail par exemple qui peut sembler faire des scans du port 25 ...) ou des équipements de translation d'adresses protégeant un réseau, nous avions mis en place une liste d'IP dite « whitelist ». Pour plus de précision, nous avons créé un système plus fin, qui permet d'ignorer uniquement les comportements normaux (scan du port 25 pour un serveur de mail, scan du port 80 pour un serveur proxy, etc.) de sorte que si la machine a un comportement différent de ce qui a été défini, une alerte soit émise et éventuellement accompagnée d'une mise en quarantaine.
- Accélération de l'interface : étant donné la forte réactivité nécessaire pour des tâches de routine dans l'interface, nous avons optimisé certaines interactions en utilisant la technologie AJAX pour :
  - pouvoir cocher très rapidement les problèmes vu sans que la page courante ne se recharge à chaque fois ;
  - réaliser la résolution de nom de manière asynchrone (les pages donnant les détails sur les problèmes liés à une machine étaient auparavant très longues à charger à cause de la résolution des noms sur les adresses IP des machines impliquées. Elles s'affichent désormais directement sans résolution de noms, puis, dès que les informations sont disponibles, les adresses résolues s'affichent).
- Prise en compte des politiques de sites : ces dernières peuvent parfois diverger, particulièrement sur la question de mettre en quarantaine ou pas. Nous avons donc permis la redéfinition pour chaque établissement (via leurs plages d'adresses IP) des comportements du système suivant les différents types de problèmes.

7.2 Évolutions en cours

Nous arrivons aujourd'hui aux limites de ce que peut traiter notre serveur (mono-processeur) en termes de débit (nous avons régulièrement un trafic de l'ordre de 300Mbit/s). Il faut donc faire évoluer notre architecture, mais Snort a le défaut de ne fonctionner que sur un seul *thread*.

Différentes techniques existent, matérielles (répartiteur de charge pour IDS, carte réseau spécialisée) ou logicielles (l2cc ou ebtables[4], filtres BPF pour Snort) pour répartir le flux réseau entre différentes machines ou processeurs.

Nous testons actuellement le fonctionnement parallèle de 4 instances Snort sur une machine bi-processeurs double coeurs (soit 4 coeurs à 2.66GHz), chaque instance étant capable de soutenir un débit de l'ordre de 400Mbit/s.

Dans la pratique, chaque instance de Snort a un filtre BPF (Berkeley Packet Filter) au niveau noyau qui effectue une répartition suivant un hachage très basique des adresses IP observées.

7.3 Évolutions futures

Pour augmenter le degré d'automatisation tout en gardant une bonne fiabilité du système, nous envisageons d'intégrer de nouvelles sources d'informations et des mécanismes de corrélation, soit en utilisant des outils de collecte brute (type Nessus, nmap, p0f) soit directement via des plates-formes (tel que ossim ou prelude) intégrant déjà de tels outils ainsi que des mécanismes de corrélation.

Dans le cas des problèmes nécessitant une investigation manuelle, certains pré-diagnostics (ex: lancer nmap sur un port suspect) pourraient être déclenchés automatiquement de façon à disposer directement dans l'interface des informations pertinentes.

7.4 Diffusion éventuelle

Jusqu'à présent, nous n'avons pas eu le temps de rendre possible la diffusion d'OSAE. Ce dernier est lié à notre système d'information et à notre infrastructure réseau. Il

serait nécessaire d'effectuer des adaptations pour pouvoir être utilisé dans un autre contexte.

De plus, il s'agit d'un outil qui a évolué au fil du temps suivant nos besoins et nos expérimentations. Il faudrait certainement le reconstruire en tirant parti de l'expérience acquise.

Nous sommes ouverts à la collaboration avec des personnes qui se trouvent dans une situation similaire et qui sont prêts à investir un temps non négligeable aux travaux nécessaires.

8 Conclusion

Plus que les modules utilisés ou utilisables (Snort, Nessus, IPS ou IDS), c'est dans leur intégration en un système unique, automatique, disposant d'une interface de suivi sur mesure avec délégation et dans lequel nous avons confiance, que se joue ici le gain en temps et en sécurité.

OSAE nous permet de gagner du temps sur le suivi et l'analyse des problèmes de sécurité mais la veille sécurité reste plus que jamais nécessaire pour faire vivre le système et entretenir son efficacité.

Bibliographie

[1] L. Facq, B. Lemaire, D. Marchand, La sécurité sur REAUMUR, *JRES 1997*
http://1997.jres.org/articles/admin1/lemaire/index.htm

[2] L. Facq, *Expérience de blocage de machines sur un campus*, Journée RSSI Décembre 2004
http://facq.reaumur.net/Documents/LF-expblc.pdf

[3] G. Moreau, *Conception et mise en production d'un système automatisé de sécurisation du réseau REAUMUR*, Mémoire d'ingénieur.
http://facq.reaumur.net/Documents/2005-Apprentissage-Moreau-OSAE.pdf

[4] Jeremy M. Guthrie, *IDS Load Balancing HOWTO*, http://lwn.net/Articles/145406/