INSTITUT DES TECHNIQUES D'INGÉNIEUR DE L'INDUSTRIE
D'AQUITAINE

MÉMOIRE DE FIN D'ÉTUDES

PRÉSENTÉ EN VUE D'OBTENIR

LE DIPLÔME D'INGÉNIEUR

DES TECHNIQUES DE L'INDUSTRIE

Dans la Spécialité Réseaux et Télécommunications

PAR

Grégoire MOREAU

DIPLÔME DÉLIVRÉ

PAR

L'ÉCOLE NATIONALE SUPÉRIEURE D'ÉLECTRONIQUE, INFORMATIQUE
& RADIOCOMMUNICATION de BORDEAUX

CONCEPTION ET MISE EN PRODUCTION D'UN SYSTÈME AUTOMATISÉ DE SÉCURISATION DU RÉSEAU REAUMUR

SOUTENU LE 19 SEPTEMBRE 2005

REAUMUR,
351 cours de la libération
33405 TALENCE

ANNÉES : 2004 – 2005

AUTEUR : Grégoire MOREAU

TITRE : Conception et mise en production d'un système automatisé de sécurisation du réseau REAUMUR

ENTREPRISE PARTENAIRE : REAUMUR, Université Bordeaux I

MAITRE D'APPRENTISSAGE : Laurent FACQ, Directeur technique

TUTEUR PEDAGOGIQUE : Michel PALLARD

NOMBRE DE PAGES : 53 pages sans l'appareil de référence.
NOMBRE DE REFERENCES BIBLIOGRAPHIQUES : 5

RESUME : Ce projet présente la solution répondant au besoin d'automatiser la gestion de la sécurité sur le réseau REAUMUR. Elle traite différentes informations pour aboutir à la reconnaissance de problèmes de sécurité, agit pour prévenir les personnes responsables et potentiellement met en quarantaine les machines posant problème. Le travail a consisté en l'analyse du besoin précis, le choix d'une solution, son développement et sa mise en oeuvre, avec enfin une analyse des résultats.

MOTS CLES : sécurité informatique / sécurité réseau / détection d'intrusions / actions automatisées / interaction homme-machine / traitement automatisé d'informations

PARTIE À REMPLIR PAR LE MAÎTRE D'APPRENTISSAGE

ACCESSIBILITÉ DE CE RAPPORT (entourer la mention choisie) :

Classe 0 = Accès libre

Classe 1 = Confidentiel jusqu'au _ _ _ _ _ _ _ _

Classe 2 = Hautement confidentiel jusqu'au _ _ _ _ _ _ _ _ (date de fin de confidentialité)

Date : Nom du signataire : Signature

Remerciements

Je remercie tout d'abord tout le personnel de mon entreprise REAUMUR pour leur accueil et leur disponibilité. Je tiens à remercier plus particulièrement Daniel Marchand, directeur du service, pour ses efforts dans le cadre de mon recrutement en tant qu'apprenti.

Je remercie vivement mon maître d'apprentissage Laurent Facq, directeur technique, pour ses enseignements, sa disponibilité et sa confiance.

Je tiens à remercier toute l'équipe pédagogique du CFAI et de l'ENSEIRB et particulièrement Didier Thiers pour m'avoir donné l'opportunité de suivre cette formation. Je remercie tous les enseignants pour leurs cours, leur patience et la qualité de leur pédagogie. Je remercie mon tuteur pédagogique Michel Pallard de la qualité de ses conseils.

Je remercie toute la promotion RT1 pour ces 3 années enrichissantes au point de vue humain.

Je remercie ma compagne Marie-Noëlle Benassy pour sa patience et ses nombreuses relectures même si le sujet du mémoire est loin d'être sa passion !

Je remercie enfin mes parents pour leurs conseils avisés et leurs encouragements.

Table des matières

[1] Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche.
[2] REseau Aquitain des Utilisateurs des Milieux Universitaires et de Recherche.

Introduction

La sécurité des réseaux depuis quelques années, a vu son importance s'accroître au point de devenir une priorité dans de nombreuses sociétés. Des outils automatisés de plus en plus complexes, des virus à la réplication foudroyante attaquent les réseaux et menacent en permanence l'intégrité des systèmes d'information. Par exemple, une machine dotée d'un système Windows, et non protégée, est désormais contaminée en moins de 12 minutes[3].

REAUMUR, le service en charge du réseau inter-universitaire éponyme n'échappe pas à ce constat et se trouve même en première ligne de défense face à ces menaces.

Dans un tel contexte, la gestion manuelle de la sécurité devient fortement consommatrice de temps.

L'objet de mon projet de mémoire est donc d'automatiser et d'améliorer la gestion de la sécurité.

J'ai suivi une démarche commençant par l'analyse de la situation initiale pour déterminer une problématique et les besoins associés. A partir de ces besoins, j'ai recherché des solutions pouvant y répondre et réalisé le projet. J'ai ensuite évalué sa réponse aux besoins une fois celui-ci mis en oeuvre. Je termine enfin sur un bilan personnel.

[3]mesure de Sophos

Chapitre 1

Présentation de REAUMUR

1.1 Présentation générale

1.1.1 De Renater[1] à REAUMUR[2]

Le réseau Renater, depuis 1991, raccorde les différentes universités françaises entre elles et apporte à celles-ci un lien avec Internet. En Aquitaine, le besoin de créer un service d'interconnexion au niveau du campus entre les différentes universités pour fédérer en un point unique l'accès à Renater (figure 1.1) a engendré la création du réseau de campus REAUMUR.

La mission de REAUMUR est définie ainsi :

« REAUMUR a pour mission d'assurer l'exploitation, la gestion et le développement de moyens et de services réseaux communs, dans le cadre des activités scientifiques, pédagogiques, documentaires ou de gestion des différents centres, laboratoires et services des organismes partenaires. REAUMUR fixe notamment les conditions de sécurité d'utilisation, en accord avec les partenaires. »[3]. Ces partenaires sont, entre autres, les universités de Bordeaux I, III et IV, le CNRS, l'ENSEIRB ou encore l'ENSAM.[4]

1.1.2 REAUMUR, un réseau de campus, un service

Le service REAUMUR se trouve sur le campus universitaire de Talence, sur le domaine du Haut Carré. Il est rattaché administrativement à l'Université Bordeaux I et plus précisément à la DRIMM[5].

REAUMUR est donc historiquement proche des utilisateurs pour la partie Bordeaux I (dans les petits laboratoires ou certaines administrations, les usagers du réseau sont parfois

[1] Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche.
[2] REseau Aquitain des Utilisateurs des Milieux Universitaires et de Recherche.
[3] Extrait de la convention portant statuts du service interuniversitaire REAUMUR.
[4] Pour une liste exhaustive voir l'annexe A page II.
[5] Direction des Ressources Informatiques et Multimédia Mutualisées.

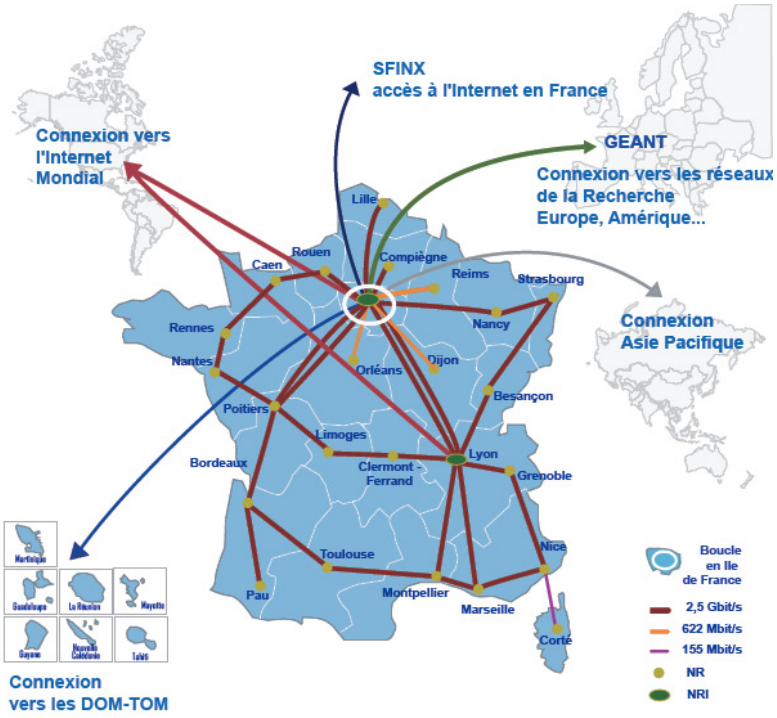

FIG. 1.1 – Réseau Renater

directement reliés à nos équipements). Cependant, de manière générale et dans le cadre de l'évolution vers les réseaux urbains et régionaux, REAUMUR a pour rôle d'interconnecter les réseaux des différents partenaires et de servir d'interface pour la connexion à Renater.

D'un point de vue technique, à sa création, REAUMUR était basé sur un réseau de fibre optique utilisant le protocole FDDI[6] (débit de 100 Mbit/s à 200 Mbit/s). En 2000, le réseau REAUMUR a migré sur le campus vers le protocole Gigabit Ethernet (1000 Mbit/s). Entre 2000 et aujourd'hui, le lien entre REAUMUR et Renater est passé de 34 Mbit/s à 1000 Mbit/s en passant par un palier à 100 Mbit/s. Cette augmentation traduit un accroissement de la demande en ressources de la part des utilisateurs.

La figure 1.2 page 8 présente les différentes manières d'être connecté à REAUMUR.

Sur cette figure, on trouve trois types d'équipements : les postes clients, les commutateurs et les routeurs.

Les commutateurs sont des éléments d'interconnexion qui permettent de constituer des réseaux locaux. Les routeurs sont des éléments d'interconnexion qui permettent la liaison de réseaux entre eux.

[6]FDDI : *Fiber Data Distribution Interface* : protocole de liaison relativement ancien, utilisant la fibre optique comme support.

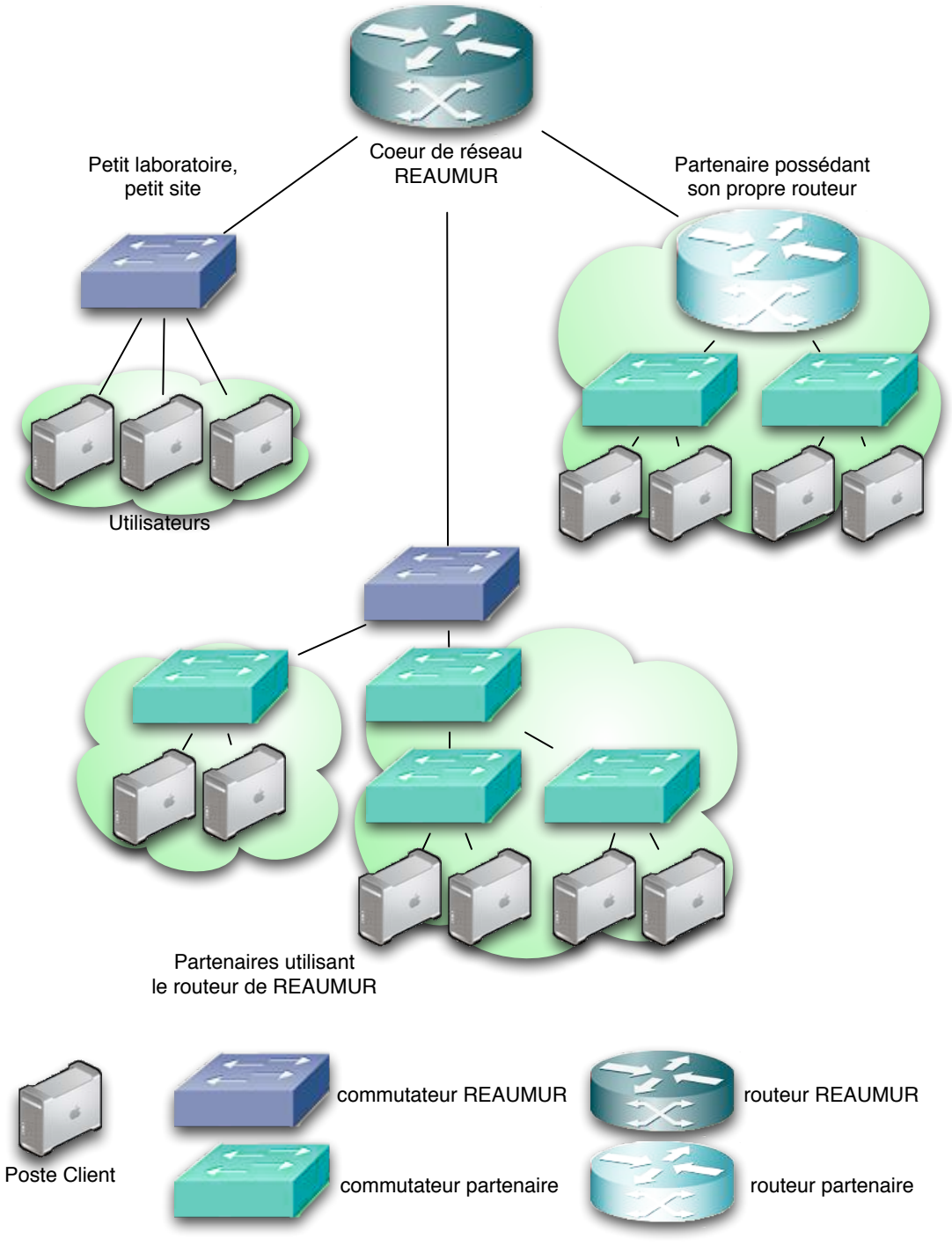

FIG. 1.2 – Diversité des connexions sur REAUMUR

On peut ainsi observer trois grands types de connexion à REAUMUR :

- la connexion type petit laboratoire ou petit site, avec un commutateur REAUMUR sur lequel des utilisateurs peuvent directement se connecter,
- la connexion d'un partenaire avec son réseau interne relié à un commutateur REAUMUR hébergé sur leur site,
- enfin, la connexion d'un partenaire qui va se faire directement sur notre routeur de cœur de réseau.

Sur le plan économique, le budget de REAUMUR est déterminé par son directeur, puis voté par le conseil d'administration rassemblant les organismes membres de REAUMUR. Il s'élève pour l'année 2004 à 570 000 €.

L'équipe de REAUMUR est composée de 7 personnes :

- 2 ingénieurs de recherche : directeur (gestion de l'administratif et des ressources humaines) et directeur technique (chef de projet, développement, sécurité du réseau),
- 1 ingénieur d'étude (développement et gestion quotidienne du réseau),
- 1 assistant ingénieur (assistance aux utilisateurs),
- 1 technicien (gestion de la partie physique du réseau),
- 1 secrétaire (assistance au directeur, gestion de la page Web du service),
- 1 apprenti (assistance aux ingénieurs, développement).

Les concurrents potentiels de REAUMUR sont les sociétés privées de gestion de réseau qui pourraient être choisies pour remplacer toute ou partie de l'équipe titulaire pour certaines fonctions.

1.1.3 REAUMUR et le réseau régional et urbain

L'ESRA[7] est le point de rassemblement des organismes d'enseignement et de recherche aquitains pour leur connexion à Renater.

Depuis son origine, ESRA est piloté par REAUMUR tandis que sa gestion était confiée à des sociétés privées. Depuis 2004, REAUMUR gère directement le réseau ESRA.

1.2 Avenir de REAUMUR

Les étudiants et personnels du campus vont disposer de moyens d'accès au réseau de plus en plus nombreux (avec le programme Aquitaine Campus Ouvert) que ce soit avec ou sans fil. La gestion de la sécurité dans ce cadre devient donc de plus en plus complexe.

Pour acquérir une dimension européenne, les universités bordelaises se rapprochent de plus en plus, avec une fusion en point de mire. Le premier pas étant la création d'un PRES[8]

[7] Réseau Enseignement Supérieur et Recherche en Aquitaine.

[8] Pole Recherche Enseignement Supérieur cf. http ://www.u-bordeaux1.fr/bx1/pres.pdf et http ://www.cpu.fr/Publications/Publication.asp ?Id=347

"Université de Bordeaux" par les 4 universités. Les structures inter-universitaires comme REAUMUR sont directement concernées.

Les débits des liaisons internes au campus et vers Renater vont encore augmenter avec la croissance des besoins des utilisateurs en termes de nouveaux services : visioconférence, grilles de calcul (réseaux à très haut débit d'ordinateurs dédiés au calcul). Ainsi, certains liens sont en cours de migration sur du 10Gigabit Ethernet.

L'augmentation du périmètre de REAUMUR, au niveau du campus de la plaque urbaine et régionale, va amener un plus grand nombre d'utilisateurs ce qui entraîne une gestion de la sécurité plus difficile.

L'augmentation du nombre d'utilisateurs amène aussi un problème. IP[9] version 4 (le protocole de base d'Internet) arrive à ses limites en terme du nombre de machines qu'il peut adresser du fait de la taille allouée à l'adresse des machines. On commence donc à manquer d'adresses IP pour les machines. Il existe des moyens pour contourner ce problème mais ils sont généralement peu satisfaisants en terme de sécurité et de compatibilité (par exemple, le NAT[10]). Pour y répondre, le nouveau protocole IP version 6 est en cours d'installation dans les différentes universités et va induire des migrations de matériels et logiciels. Cela risque d'entraîner une augmentation du besoin en ressources humaines.

En plus de la nécessité d'importants débits, les nouvelles applications ont des besoins de *qualité de service*[11] et donc d'une approche différente du réseau.

Toutes ces évolutions vont demander du temps et des moyens humains supplémentaires. Or, actuellement, la fonction publique est dans une situation où même le renouvellement des départs à la retraite pose problème. Il faut donc compter au mieux sur des ressources humaines qui vont rester constantes. Il est donc nécessaire d'économiser du temps sur les tâches qui peuvent en faire l'objet.

De ce fait, le projet va s'axer sur l'économie de temps dans une des tâches les plus consommatrices : la gestion de la sécurité.

---

[9] IP : *Internet Protocol*, protocole de niveau réseau, permettant l'adressage des machines et l'acheminent des données.

[10] NAT : *Network Address Translation* permet d'utiliser une même adresse IP sur Internet pour un ensemble de machines mais peut poser problème pour certaines applications comme la visioconférence.

[11] Qualité de service (QoS pour *Quality of Service* en anglais) : il s'agit en fait de la qualité du transport des informations dans le sens où les données arrivent dans l'ordre et avec un délai de transmission contrôlé. Dans le cadre d'un échange classique de données, il n'y a pas à s'en soucier. Par contre, par exemple pour la visioconférence, la qualité de service est nécessaire pour éviter d'avoir des mauvaises transmissions.

Chapitre 2

Situation initiale

Au commencement du projet, j'ai déterminé le contexte du projet et analysé l'existant. De cette manière, suite à des discussions avec mon maître d'apprentissage nous avons pu définir une problématique et les contraintes du projet. A partir de cette problématique nous avons dégagé des objectifs et leurs indicateurs associés pour enfin aboutir à l'expression des besoins.

2.1 Contexte du projet

REAUMUR est confronté à une grande quantité de problèmes de sécurité.

D'une part, certains de ces problèmes sont potentiellement dangereux pour le réseau car ils peuvent entraîner des dysfonctionnements des équipements réseaux et une saturation de la bande passante du campus.

D'autre part, des problèmes liés aux abus d'utilisateurs (utilisation non professionnelle du réseau) peuvent induire une surconsommation de ressources et donc un surdimensionnement du réseau. Ils peuvent entraîner une mauvaise image de REAUMUR vis à vis de Renater, voire même des plaintes en justice (dans les cas d'utilisation du réseau pour des échanges de fichiers illégaux).

De plus, ces problèmes consomment un temps de traitement manuel non négligeable aussi bien au niveau de l'équipe du service qu'au niveau des correspondants de REAUMUR. Ces correspondants (interlocuteurs privilégiés de REAUMUR) sont les personnels présents sur les sites, informaticiens ou non. Ils gèrent la communication entre utilisateurs finaux et REAUMUR.

La situation actuelle de la fonction publique, rappelons le, fait qu'il est difficile d'obtenir des ressources humaines supplémentaires. Il est donc primordial de réussir à automatiser les tâches pouvant l'être. Ainsi, du temps pourra être gagné pour l'équipe de REAUMUR. Celle-ci pourra réaliser des projets supplémentaires plutôt que de traiter des tâches quotidiennes répétitives.

2.2 Analyse de l'existant

Au début du projet, soit au début de l'année 2004, le traitement des problèmes de sécurité se faisait en 3 étapes :

- détection d'un problème, analyse puis mise en quarantaine de la machine incriminée si besoin était et envoi de courrier électronique au correspondant responsable de la machine,
- résolution du problème de la part du correspondant et envoi de courrier électronique à REAUMUR demandant le rétablissement de la connexion de la machine sur le réseau,
- prise en compte par REAUMUR, et le cas échéant, levée de la quarantaine.

Des mesures avaient été faites de manière à quantifier le temps perdu par REAUMUR pour la gestion des problèmes de sécurité (dans la portion qui peut être automatisable). Le résultat était de 1 heure et 15 minutes par jour globalement (et de temps ingénieur qui plus est !). Il faut noter que cette valeur est une moyenne et qu'il arrive sur certaines périodes (apparition d'un nouveau virus, piratage massif...) que cette durée soit largement dépassée. Il y a donc une perte de temps réelle qui, de plus, occasionne des interruptions dans les tâches d'ingénierie.

De plus, les informations concernant les problèmes de sécurité ne sont pas directement accessibles aux correspondants. Ceux-ci doivent transiter par le personnel de REAUMUR pour les obtenir. Il y a donc, en plus, un problème d'accès à l'information.

Les correspondants ont estimé qu'ils attendent en moyenne 2 heures avant que leurs demandes de levée de quarantaine ne soient traitées. La perte de temps au niveau des correspondants, et donc au niveau des utilisateurs finaux, est importante.

Enfin, le fait que la sécurité soit gérée manuellement par du personnel introduit le risque que des piratages puissent avoir lieu durant la nuit, les week-end et les vacances. Un système automatique pourrait contrer les attaques dans ces périodes.

2.3 Problématique

De cette analyse de l'existant, on peut déduire une problématique portant sur trois volets :

1. du temps est perdu pour réagir aux problèmes de sécurité dans le sens où ces réactions sont la plupart du temps systématiques et pourraient être traitées automatiquement,
2. lorsque les problèmes entraînant une mise en quarantaine ont été résolus par les correspondants, ceux-ci sont dépendants de REAUMUR pour le rétablissement de l'accès au réseau des machines posant problème,
3. un risque potentiel important existe durant les périodes d'absences de personnels dans le service.

2.4 Objectifs visés par le projet

Le projet doit aboutir à la mise en service d'un système automatisé d'assistance à la gestion de la sécurité. Il doit répondre à trois objectifs principaux qui découlent de la problématique :

1. **dégager au moins 75% de temps pour l'équipe REAUMUR dans les tâches de sécurité automatisables.**
   L'indicateur de performance associé à cet objectif est le rapport du nombre de problèmes de sécurité traités automatiquement sur le nombre de problèmes total. Ce rapport correspond au pourcentage de problèmes de sécurité n'ayant pas nécessité d'intervention, il traduit donc le temps gagné.
   Cet indicateur est relevé dans les comptes-rendus du système ;

2. **permettre une interaction instantanée avec les correspondants de REAUMUR pour 90% des problèmes de sécurité.**
   L'indicateur de performance associé à cet objectif est le rapport des problèmes de sécurité qui nécessitent une communication, et donc une perte de temps, sur le nombre total de problèmes de sécurité.
   Cet indicateur est relevé par décompte des échanges effectués par courrier électronique avec les correspondants ;

3. **traiter automatiquement au moins 50% des problèmes de sécurité qui surviennent durant les périodes non ouvrées.**
   L'indicateur de performance associé à cet objectif est le rapport du nombre de problèmes traités automatiquement sur le nombre de problèmes total durant les périodes non ouvrées.
   Cet indicateur est relevé dans les comptes rendus du système.

2.5 Contraintes

Ces objectifs doivent être remplis, mais sous certaines contraintes.

La politique de REAUMUR est d'utiliser autant que possible les logiciels libres et de mettre à disposition de la communauté les outils développés en interne. Elle s'explique par une volonté d'indépendance vis à vis des sociétés externes pour notre gestion de la sécurité du réseau. De plus, les compétences nécessaires à l'utilisation et à la conception de logiciels libres sont présentes dans l'équipe.

D'un point de vue technique, le flux réseau du campus vers l'extérieur peut potentiellement atteindre le Gigabit (il s'agit de la bande passante dont nous disposons avec Renater).

Sur le plan économique, le budget alloué est de 5 000 € pour la partie matérielle. Le coût déjà engagé pour l'apprenti est de 15 000 €(salaire et charges).

En termes de calendrier, le système devait être mis en fonctionnement d'ici Mars 2005. L'ensemble de la documentation et la présentation du système devait être fait d'ici juin - juillet 2005.

2.6 Démarche

Pour pouvoir mener à bien ce projet suivant les contraintes formulées, j'ai suivi la démarche suivante :

- expression des besoins,
- recherche de solutions,
- en fonction de la solution retenue, conception ou choix du système, réalisation et mise en oeuvre,
- validation,
- mise en beta test pour certains utilisateurs,
- réalisation d'une documentation d'utilisation,
- mise en production,
- suivi.

2.7 Expression des besoins

Suivant les objectifs et les contraintes il faut maintenant déterminer les besoins qui y sont associés et pour qui ?

2.7.1 Utilisateurs du système

On peut distinguer trois types d'utilisateurs du système :

- l'équipe REAUMUR,
- les correspondants,
- les utilisateurs finaux du réseau.

Le système doit permettre une gestion particulière des droits qui sont les suivants :

- voir les informations de sécurité,
- modifier le comportement du système,
- lever une quarantaine.

L'association droits-utilisateurs est la suivante :

- l'équipe REAUMUR a tous les droits sur tous les problèmes relevés pour tout le campus, avec le maximum de détails,
- les correspondants ont la possibilité de lever les quarantaines qui concernent le sous-réseau dont ils ont la responsabilité mais disposent d'informations partielles,
- les utilisateurs finaux n'ont que le droit de lever la quarantaine sur leur machine (avec une protection contre les abus) et n'ont que le minimum d'informations sur le problème détecté.

2.7.2 Perte de temps sur le traitement des problèmes de sécurité

Pour répondre à ce point, il faut d'abord définir les types de problèmes qui seront traités :

- actes de piratage (compromission de machine),
- propagation de virus,
- abus d'utilisateurs (utilisation du réseau non conforme à la charte REAUMUR[1]).

Le système doit donc être capable de détecter et de répondre automatiquement à ces problèmes avec un minimum d'intervention humaine.

La réponse aux problèmes sera fonction de la gravité de ceux-ci. Dans les cas les plus graves, la machine source du problème est mise en quarantaine. Dans les cas les moins graves, le système envoie uniquement un courrier électronique au correspondant responsable de la machine posant problème.

La réponse à ces problèmes doit se faire rapidement : de l'ordre de 5 minutes.

Le fait de mettre la machine en quarantaine plutôt que de faire un tri sélectif des paquets de données (pour différencier ceux qui correspondent à un problème de sécurité de ceux qui sont sains) est à la fois un choix politique et technique. C'est un choix politique car le fait de bloquer complètement une machine dans le cadre d'abus d'utilisation fait prendre conscience aux utilisateurs de leur non-respect de la charte. On espère ainsi induire des changements de comportement. C'est aussi un choix technique car, dans le cas d'une attaque pirate, si jamais un filtrage sélectif est mis en oeuvre, rien ne garantit que le pirate ne trouve finalement pas un moyen de contourner celui-ci. Une mise en quarantaine complète permet ainsi de ne pas laisser la possibilité au pirate d'avoir un quelconque accès à la machine posant problème.

2.7.3 Interaction plus rapide avec les correspondants

Pour améliorer la rapidité de l'interaction avec les correspondants, il doit leur être fourni une interface sur le système leur permettant d'agir directement sur celui-ci. Les actions principales sont :

- visualiser les événements de sécurité,
- lever les quarantaines.

2.7.4 Risques associés aux absences de personnel

Pour diminuer ces risques, le système doit au maximum pouvoir prendre des décisions automatiques. Plus les réponses sont rapides, moins les risques sont grands.

De plus, les décisions ne doivent pas être erronées pour éviter de bloquer des machines qui n'ont pas à l'être. Le système doit aussi permettre de ne pas bloquer des machines

[1] Charte REAUMUR : document signé par tout utilisateur de REAUMUR précisant que l'utilisation du réseau doit être strictement professionnelle à des fins de recherche.

considérées comme vitales pour le réseau, par exemple les serveurs de nom (DNS[2], serveurs de courrier électronique, etc. Dans le cas d'une suspicion sur une machine de ce type, la décision doit être prise par un ingénieur REAUMUR.

2.8 Planning prévisionnel

Le planning prévisionnel du projet que j'ai défini lors de la validation de mémoire est le suivant :

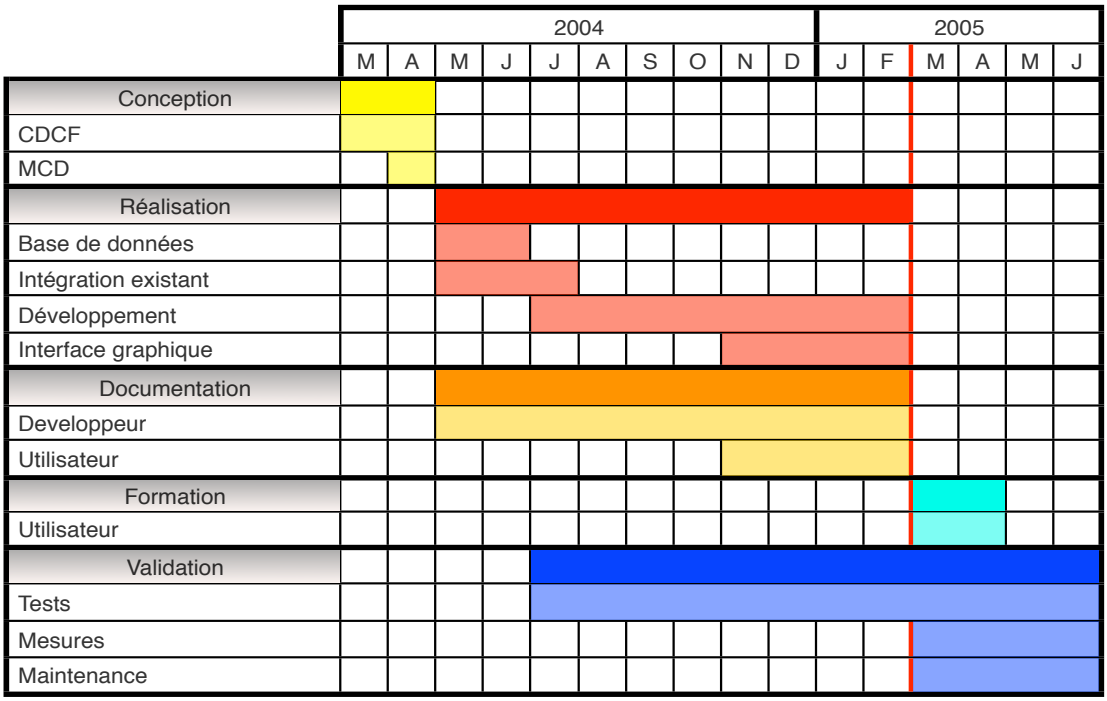

FIG. 2.1 – Planning prévisionnel du projet

CDCF : Cahier Des Charges Fonctionnel. MCD : Modèle Conceptuel des Données.

La ligne rouge correspond à la date de mise en production prévisionnelle.

L'analyse de la situation initiale m'a permis d'obtenir une problématique résumant les besoins du service et des utilisateurs. L'expression de ces besoins permet maintenant d'effectuer un choix dans les différentes solutions potentielles.

[2]DNS : *Domain Name Server*, serveur qui attribue les noms de domaines (par exemple u-bordeaux.fr), et les noms de machines sur ces domaines.

Chapitre 3

Recherche des solutions

Il existe de nombreuses solutions commerciales ou libres permettant plus ou moins de répondre aux besoins. Il est donc nécessaire d'effectuer une analyse de ces solutions afin d'effectuer le bon choix.

3.1 Solutions envisageables

Tout d'abord, examinons le principe de fonctionnement général de la solution recherchée.

3.1.1 Principe de fonctionnement général

De manière globale, les solutions pouvant répondre au problème posé sont basées sur le principe de la contre-mesure associée à une sonde de détection d'intrusion (IDS[1]). Le principe de fonctionnement d'un tel dispositif est le suivant : la sonde détecte un événement de sécurité et le système de contre-mesure réagit en filtrant la connexion réseau de la machine qui pose problème. Il existe des solutions plus puissantes permettant de vérifier sur chaque machine l'état des composants de sécurité (antivirus, pare-feu[2],...) et leur niveau de mise à jour pour ensuite autoriser ou non l'accès de la machine au réseau.

On distingue ainsi deux approches : une au niveau global, l'autre au niveau des postes clients. Dans notre situation, avec un parc informatique très hétérogène et dont la gestion ne nous appartient pas, il est impossible de mettre en place une solution nécessitant une action sur tous les postes. Du fait de notre situation d'intermédiaire, nous ne pouvons avoir qu'une approche globale vis à vis du flux réseau transitant entre le campus et l'extérieur (Renater).

Au niveau du filtrage des machines qui posent problème, il y a deux approches possibles. La première est de bloquer uniquement les communications problématiques (Peer to Peer[3],

[1]IDS : *Intrusion Detection System* : outil permettant de détecter des événements de sécurité.

[2]Pare-feu : logiciel ou matériel qui filtre les menaces les plus simples du flux réseau.

[3]Peer to Peer : protocole d'échange de données se faisant directement de poste client à poste client sans avoir besoin de serveurs. Le Peer to Peer est utilisé en grande majorité à des fins illégales.

tentative de connexion de la part d'un pirate, diffusion d'un vers[4], ...). La seconde est d'isoler totalement la machine. Notre choix est de bloquer totalement les machines posant problème (voir la section 2.7.2 page 15 pour plus de détails).

Il est possible de distinguer trois types de solutions :

- les solutions commerciales,
- les solutions libres,
- le développement en interne d'une solution.

Il faut examiner de quelle manière chacune de ces solutions répond au problème.

3.1.2 Solutions commerciales

Dans le cadre des solutions commerciales, on peut trouver de manière non exhaustive :

- Cisco PIX 525 : 9 930 €[5],
- Cisco ASA 5540 : 14 060 €,
- CheckPoint Enterprise : 17 400 €,
- Allot NetEnforcer : 24 900 €.

On peut voir que les solutions commerciales ont un principal défaut : elles coûtent très cher (largement au delà du budget maximum fixé : pour rappel 5 000 € pour la partie matérielle).

Elles apportent beaucoup de fonctionnalités inutiles dans le cadre de ce projet et sont difficilement modifiables.

Il est nécessaire de souscrire à des abonnements pour tenir les équipements à jour par rapport aux nouvelles menaces de sécurité.

Enfin, je n'ai pas détaillé davantage mon étude par rapport à ces solutions car elles ne répondent pas à une contrainte : elles nous lient à des sociétés externes pour la gestion de la sécurité.

3.1.3 Solutions libres

Dans le domaine des logiciels libres, il existe une multitude d'outils de contre-mesure associés à la sonde de détection d'intrusion que nous utilisons à REAUMUR, Snort. Ces outils sont entre autres : Hogwash, Guardian, Flexresp...

De manière générale, ces outils font le même travail. Ils traitent les paquets de données de manière transparente, laissant passer ceux qui ne déclenchent pas de signature et bloquant les autres. Cependant, ils ne permettent pas la mise en quarantaine directement comme nous le souhaitons. Ils nécessitent la déviation du flux réseau par un PC, ce qui peut induire des

[4]Un vers est un virus se propageant de manière autonome par le réseau. Contrairement à un virus classique qui se diffuse par courrier électronique, il ne nécessite pas d'action de la part de l'utilisateur.

[5]Tous les montants indiqués sont extraits des tarifs publics en vigueur au mois de Juin 2005

risques de pannes et de mauvaises performances (un PC, aussi puissant soit-il, n'est pas conçu exclusivement pour traiter un flux réseau important ; au contraire d'un routeur ou d'un pare-feu matériel).

De plus, la fiabilité de tels dispositifs est limitée. En effet, il peut y avoir des faux positifs qui entraînent des filtrages erronés.

Enfin, la configuration et « l'intelligence » de ces outils sont très limitées. Nous avons un besoin plus complexe.

Pour l'interface avec les utilisateurs, il existe plusieurs outils permettant la visualisation des événements de sécurité détectés par Snort : ACID, Snort Alert Monitor et bien d'autres. Cependant, ceux-ci ne fournissent qu'une vue unique sans offrir une gestion fine des droits nécessaire dans notre contexte multi-utilisateur.

3.1.4 Développement interne

Le développement en interne de la solution permet de répondre totalement et précisément à nos besoins. De plus, les coûts d'un tel développement sont pour leur grande part couverts par le salaire de l'apprenti.

Cependant, un tel travail comporte des risques qui peuvent empêcher d'atteindre tous les objectifs. Ces risques trouvent leur origine au niveau de la conception, de la gestion du planning ou d'une mauvaise évaluation de la faisabilité.

3.2 Choix de la solution

Des trois solutions présentées, l'une peut être éliminée directement. Il s'agit de l'utilisation de solutions commerciales. D'une part, ces solutions sont beaucoup trop coûteuses, d'autre part, elles sont en contradiction avec une contrainte : elles nous lieraient à une société pour la gestion de la sécurité.

Les deux solutions restantes divergent beaucoup dans la réponse qu'elles apportent aux besoins.

Aucune des solutions libres ne répond complètement, bien que certains besoins soient couverts par différents outils libres.

Au contraire, le développement d'une solution interne pourrait fournir une réponse complète, même si ce choix comporte des risques.

Les besoins liés à l'interface du système sont très particuliers à cause de la spécificité de notre situation à REAUMUR. En effet, il est nécessaire d'assurer une gestion très fine des droits entre les différents utilisateurs du système (voir 2.7.1 page 14). Aucune interface dans le domaine des logiciels libres ne permet actuellement de disposer d'une telle gestion des droits.

En définitive, on constate qu'il manque de nombreux éléments aux solutions libres pour pouvoir répondre à tous nos besoins. Cependant, développer des logiciels qui existent déjà

est une perte de temps et augmente les risques de non réussite.

Aussi, notre choix se porte sur l'utilisation d'outils libres avec un développement complémentaire pour répondre totalement aux besoins.

De nombreux prototypes ont déjà été réalisés par mon maître d'apprentissage suivant cette stratégie. Ils permettent de confirmer la faisabilité d'un système réalisé de cette manière. Cette faisabilité sera étudiée plus précisément par la suite.

3.3 Choix des outils logiciels et matériels pour le système

Avant de commencer le développement, il faut donc :

- rechercher quels logiciels libres peuvent être utilisés et pour s'acquitter de quelles tâches,
- déterminer les formats de stockage de l'information en fonction des contraintes techniques et des besoins,
- faire le choix des langages de programmation qui vont être utilisés,
- dimensionner le matériel supportant le système.

3.3.1 Quelle sonde de détection d'intrusion ?

Snort, une sonde de détection d'intrusion, est utilisée depuis de nombreuses années par mon maître d'apprentissage. Il s'en sert pour détecter les problèmes de sécurité avec des règles de sa création. Ce patrimoine doit être réutilisé et pouvoir évoluer. De plus, Snort demeure le détecteur d'intrusion de référence dans le monde des logiciels libres. Il est donc logique de s'en servir dans le cadre du projet.

3.3.2 Quel format de stockage utiliser pour Snort ?

Les informations que Snort récolte sont dissociées en deux parties. La première, l'alerte, contient le nom de la signature qui a été détectée avec les informations (source et destination de la communication, instant précis de détection, détails techniques sur la communication) concernant le paquet de données responsable de l'événement. La seconde, la charge utile, représente le contenu de ce paquet de données.

Snort permet de stocker ces informations de plusieurs manières et je me suis particulièrement intéressé aux suivantes :

- base de données,
- fichiers textes pour les alertes et fichiers binaires[6] pour les charges utiles,

[6]Format binaire : il s'agit du format de représentation élémentaire des données en informatique. Son alphabet est composé de 2 éléments : 0 et 1.

– socket UNIX[7],
– format binaire pour l'ensemble des informations (alertes et charges utiles), appelé *unified*.

En premier lieu, j'ai testé l'exportation dans une base de données. Mais rapidement, je me suis rendu compte qu'il y avait des problèmes. Le plus important d'entre eux est la performance. Snort, lorsqu'il écrit dans la base de données, doit attendre que celle-ci ait traité les informations transmises. Durant ce laps de temps, le flux réseau ne s'interrompt pas et des événements peuvent potentiellement ne pas être détectés. Le schéma de la base de données est pour sa part relativement complexe et rend l'exploitation difficile.

L'utilisation de fichiers textes associés à des fichiers binaires pour les charges utiles permet de disposer facilement des alertes. En contrepartie, la récupération des charges utiles correspondantes n'est pas directe. De plus, ce format est relativement peu performant.

L'utilisation de socket UNIX permet d'éviter d'utiliser de l'espace sur le disque en envoyant directement les informations (les alertes et les charges utiles associés sont alors groupées) de Snort dans un programme effectuant le traitement souhaité. Mais, ce mode de fonctionnement comporte le même inconvénient que la base de données : le fait que Snort soit obligé d'attendre un autre programme lorsqu'il retourne ses informations, entraînant de nouveau un risque de pertes.

Le format *unified* permet de stocker toutes les informations (alertes et charges utiles) dans des fichiers binaires. Ce format est celui qui demande le moins de travail à Snort et permet, d'après sa documentation, de traiter sans pertes un flux Gigabit. Pour sa lecture il nécessite l'utilisation du logiciel *barnyard*. Cela permet de désynchroniser l'écriture des informations par Snort et leur lecture. Celle-ci peut être faite indépendamment. Les informations sont alors traités sans pour autant entraver le fonctionnement de Snort.

Le principal critère de choix étant les performances, j'ai donc choisi le format le plus efficace : le format *unified*. Il permet de limiter au maximum le nombre d'événements non capturés par Snort. De plus, il offre la possibilité de rendre indépendants la capture des informations et leur traitement.

Barnyard, lorsqu'il lit les fichier *unified*, offre les mêmes formats de sortie que Snort accompagnés de variantes. J'ai choisi d'utiliser une de ces variantes : la sortie sous forme de fichier texte contenant à la fois les alertes et leurs charges utiles associées. Cependant, le stockage de telles informations sur le disque dur est complètement redondant avec les fichiers *unified* déjà stockés. C'est pourquoi j'ai utilisé un fichier FIFO[8] qui ne prend aucune place sur le disque dur. *Barnyard* va écrire dedans comme dans un fichier normal. Tant qu'un programme n'ira pas lire ce fichier, *barnyard* devra attendre (rappelons que ce n'est pas gênant du fait de la désynchronisation avec Snort).

Une fois ces données **traitées** par le système, le résultat va être stocké dans une base de données. Il ne faut cependant pas la choisir au hasard.

[7]Socket UNIX : c'est un moyen de dialogue entre programmes sur une machine ayant un système d'exploitation UNIX.

[8]FIFO : *First In First Out*, c'est une structure de données où le premier élément inséré est le premier à sortir.

3.3.3 Quelle base de données ?

Du fait de la nécessité d'utiliser un outil libre et ouvert, le choix se limite principalement à PostgreSQL et MySQL.

PostgreSQL dispose de plus de fonctions tandis que MySQL est censé être plus performant. De mon point de vue, j'ai un besoin de performance relatif tandis que certaines fonctionnalités de PostgreSQL peuvent simplifier le développement (par exemple, le type de données INET[9]). Actuellement, les différentes bases de données de REAUMUR s'appuient sur PostgreSQL. Cependant, dans un souci d'évolutivité du système, j'ai souhaité mesurer les performances de MySQL et PostgreSQL. J'ai donc installé les deux logiciels sur une machine de test et créé des conditions d'utilisations extrêmes (600 000 entrées dans la base qui représentent une année d'archives). J'ai ensuite effectué différentes requêtes :

- pour une lecture de toutes les entrées : PostgreSQL a mis 28 secondes tandis que MySQL en a mis 15,
- une lecture sous conditions, de sorte à n'obtenir qu'une centaine d'entrées correspondant à une recherche (un cas plus proche d'une utilisation réelle) fait que MySQL s'effondre, il met 240 secondes. PostgreSQL, quant à lui, ne met que 6 secondes.

D'autres mesures publiées sur Internet confirment que PostgreSQL est globalement plus performant[10]. Cependant, cela reste un sujet controversé.

Mon choix se porte sur PostgreSQL surtout du fait de notre expérience de celui-ci et des fonctionnalités avancées qu'il propose.

3.3.4 Quels langages de programmation ?

Pour la partie traitement et automatisation, il est nécessaire de disposer d'un langage pouvant dialoguer simplement avec une base de données et offrant des structures de données simples d'utilisation mais performantes.

Par ailleurs, il faut prendre en compte le langage dans lequel les différents prototypes existants ont été développés pour pouvoir potentiellement en reprendre certains. Mon choix se porte donc sur le langage PERL[11] d'autant que c'est un langage que j'ai eu plusieurs fois l'occasion de manipuler (en particulier pour mon projet technique).

Pour la partie interface graphique, il faut que celle-ci soit utilisable quel que soit le matériel de l'utilisateur (PC, station UNIX, MAC, ...). Une interface WEB répond le mieux à cette demande. Mais quel langage utiliser pour cette interface ?

[9] INET : type de données dans PostgreSQL qui permet de manipuler directement des adresses IP et sous-réseaux IP et d'effectuer directement des tests d'inclusion, de comparaison, etc.

[10] voir à ce propos http ://benchw.sourceforge.net/

[11] PERL : *Practical Extraction and Report Language*, langage de programmation permettant d'écrire des scripts puissants.

Il faut tout d'abord noter que les pages devront être générées dynamiquement. On ne choisira donc pas les pages HTML[12] simples. Dans le domaine des pages WEB dynamiques, on distingue deux possibilités : CGI[13] (en utilisant le langage C[14] ou le PERL) ou PHP[15]. Du fait des contraintes, les outils non libres tel ASP sont écartés. L'utilisation du C (avec CGI) pour créer des pages WEB peut être intéressante pour ce qui est des performances qu'elle procure. Cependant, la complexité d'un tel système risque d'être gênante. Il est aussi possible d'utiliser PERL (avec CGI) pour générer les pages mais celui-ci n'a pas été créé dans ce but en premier lieu. Il est donc logique d'utiliser l'outil le plus adapté à ce type de travail : PHP.

Pour des raisons de maintenabilité, la version de PHP que j'ai choisie est celle qui est fournie avec la distribution GNU/Linux que nous utilisons : Debian (version Sarge), PHP version 4. Ainsi, les mises à jour de PHP sont transparentes car gérées par la distribution.

3.3.5 Dimensionnement du matériel

Le matériel utilisé est un PC. Il doit être en mesure de capturer et traiter un flux réseau Gigabit et stocker les informations pertinentes sur les problèmes de sécurité dans ce flux. Le dimensionnement se fait surtout par rapport aux outils qui vont être les plus consommateurs de ressources : Snort et la base de données PostgreSQL.

Snort fonctionne depuis de nombreuses années sur le PC de mon maître d'apprentissage. Nous disposons ainsi d'informations sur les besoins en espace disque, de consommation de mémoire vive et de temps processeur pour l'utilisation de Snort. Par ailleurs, les bases de données ont besoin de beaucoup de mémoire et de grandes capacités en disque dur.

Dans le cadre du système mis en place, le flux réseau peut atteindre 1Gbit full duplex[16] au maximum. Le choix matériel s'est donc porté sur un processeur rapide accompagné d'une grande quantité de RAM, d'un disque dur de capacité relativement élevée et surtout d'une carte réseau capable de gérer un tel flux. La machine choisie dispose donc d'un processeur Pentium4 à 3,4Ghz, avec 1Go de RAM et 160Go de disque et bien évidemment une carte réseau Gigabit. D'ailleurs, il ne s'agit pas d'une carte sur bus PCI[17] mais d'un contrôleur réseau Intel directement intégré à la carte mère. Cela offre de meilleures performances (pas de limitation de bande passante à cause du bus PCI qui s'avère limité aux environs de 500Mbit/s dans le meilleur des cas).

[12]HTML : *HyperText Markup Language*, langage permettant d'écrire des pages dites hypertextes. Celles-ci permettent de relier des documents stockés sur différents serveurs par l'intermédiaire de liens.

[13]CGI : *Common Gateway Interface*, permet de faire dialoguer un navigateur WEB avec un programme exécuté côté serveur qui peut être écrit en n'importe quel langage.

[14]Langage C : langage de programmation datant des années 70, performant, mais complexe.

[15]PHP : langage de programmation spécifiquement créé pour l'écriture de pages WEB dynamiques.

[16]Full duplex : permet un échange simultané d'informations (émission et réception en même temps)

[17]PCI : *Peripheral Component Interconnect*, élément interne à un PC permettant de connecter des cartes d'extensions, comme par exemple une carte réseau...

3.4 Faisabilité du développement

Le travail à faire est en partie un redéveloppement. La faisabilité de celui-ci est donc relativement garantie du fait qu'il existe déjà un certain nombre d'éléments en état de marche même s'ils doivent être réécrits.

La partie du système qui concerne l'automatisation n'existe pas du tout. La faisabilité de cette partie est incertaine mais différents prototypes jetables[18] ont permis de valider certains concepts.

[18]Prototype jetable : c'est un programme qui sert à vérifier qu'il est possible d'effectuer une tâche. Il est écrit rapidement et n'est pas destiné à être utilisé par la suite. Il permet de se faire une première expérience.

Chapitre 4

Réalisation

Le contexte du projet est le suivant : je suis seul à développer, je re-conçois certains éléments en me basant sur les prototypes de mon maître d'apprentissage tandis que je conçois les éléments qui manquent (toute la partie automatisation).

Le but est d'obtenir rapidement un système utilisable pour ensuite l'améliorer en fonction de la réponse de celui-ci aux besoins. On s'inspire en particulier de la méthode RAD[1].

4.1 Les étapes de la réalisation

Afin de définir au mieux le besoin des utilisateurs, une enquête a été réalisée. Elle m'a permis d'élaborer le cahier des charges fonctionnel du système qui répond donc, dans la mesure du possible, aux besoins des utilisateurs et à la problématique.

Une fois ce cahier des charges validé, je me suis penché sur la création d'un modèle conceptuel des données (MCD) permettant de disposer de toutes les informations pour pouvoir réaliser les différentes fonctions dans le cadre des contraintes formulées.

Les différents modèles conceptuels des données que j'ai pu construire ont été discutés avec mon maître d'apprentissage. L'objectif de ces discussions était de disposer d'une version cohérente avec les besoins mais pas trop complexe à mettre en oeuvre tout un conservant un haut niveau de performance.

Nous avons donc choisi de nous inspirer de la méthode RAD, du fait de son fonctionnement sur la base de prototypes et la construction de l'application par modules. Nous nous en inspirons seulement car il est toujours difficile d'appliquer une méthode de manière orthodoxe (dans notre cas, par exemple, il a été impossible de ne pas dépasser 120 jours de développement et de faire valider chaque étape du développement par les utilisateurs).

[1]RAD : *Rapid Application Development*, méthode de développement rapide d'application, fondée sur des cycles courts et une forte interaction avec les utilisateurs

4.2 Enquête

L'enquête a été menée auprès de tous les correspondants de REAUMUR (au nombre de 200). Il y a eu 59 réponses individuelles soit plus de 25%. Elle a permis de déterminer plus précisément les besoins des utilisateurs et surtout définir l'ordre de priorité des différents éléments à développer.

4.3 Présentation du cahier des charges fonctionnel pour le développement

Je vais dans cette partie présenter les fonctions du système extraites du cahier des charges fonctionnel du système. La version complète de celui-ci se trouve en annexe : B page IV. Pour écrire celui-ci, j'ai utilisé la méthode de l'analyse fonctionnelle.

4.3.1 Situations de vie

Voyons tout d'abord les différentes situations dans lesquelles va se trouver le système :

- installation,
- exploitation,
- maintenance,
- évolution.

Ces différentes situations représentent les différents cas qu'il faut prévoir dans le cadre de la vie du système.

4.3.2 Définition des fonctions du système

Définissons quelques termes :

- FP est une fonction principale, celle-ci définit la réponse à un besoin,
- FC est une fonction contrainte.

FP1 Le système permet à l'équipe REAUMUR d'automatiser partiellement la gestion de la sécurité.

FP2 Le système permet aux correspondants de gérer les levées de filtrages les concernant.

FP3 Le système permet à l'équipe REAUMUR de visualiser tous les événements et informations collectés de l'ensemble du réseau.

FP4 Le système permet aux correspondants de visualiser les événements et informations collectés de la partie du réseau les concernant.

FP5 Le système permet à l'équipe REAUMUR de choisir une ou plusieurs actions pour un événement non traité automatiquement.

FP6 Le système permet aux correspondants de personnaliser le comportement de celui-ci pour leur domaine.

FP7 Le système permet aux utilisateurs finaux de lever un filtrage concernant leur machine.

FC1 Le système doit respecter les règles de sécurité informatique.

FC2 Le système doit avoir une interface homme/machine intuitive et ergonomique.

FC3 Le système doit disposer d'une interface machine/machine (option).

FC4 Le système doit utiliser des standards libres et ouverts.

FC5 Le système doit disposer d'outils de mesure sur son utilisation.

FC6 Le système doit pouvoir s'interfacer avec les outils utilisés.

FC7 Le système doit signaler les événements.

FC8 Le système doit avoir son code source et sa documentation technique en anglais.

FC11 Le système doit être disponible sous forme de paquet Debian (option).

FC21 Le système doit pouvoir être maintenu par d'autres personnes que le développeur initial.

FC22 Le système doit être modulaire.

FC31 Le système doit pouvoir s'adapter aux évolutions du réseau.

FC32 Le système doit pouvoir disposer d'une interface en plusieurs langues (option).

4.4 Analyse et principe de fonctionnement

Je vais ici détailler l'analyse qui a été faite en vue d'obtenir le modèle conceptuel des données (cf. Annexe C page XIII) et les grandes lignes à suivre pour le développement.

D'un point de vue global, le fonctionnement peut être schématisé sur la figure 4.1 page 28. Je vais par la suite le détailler élément par élément.

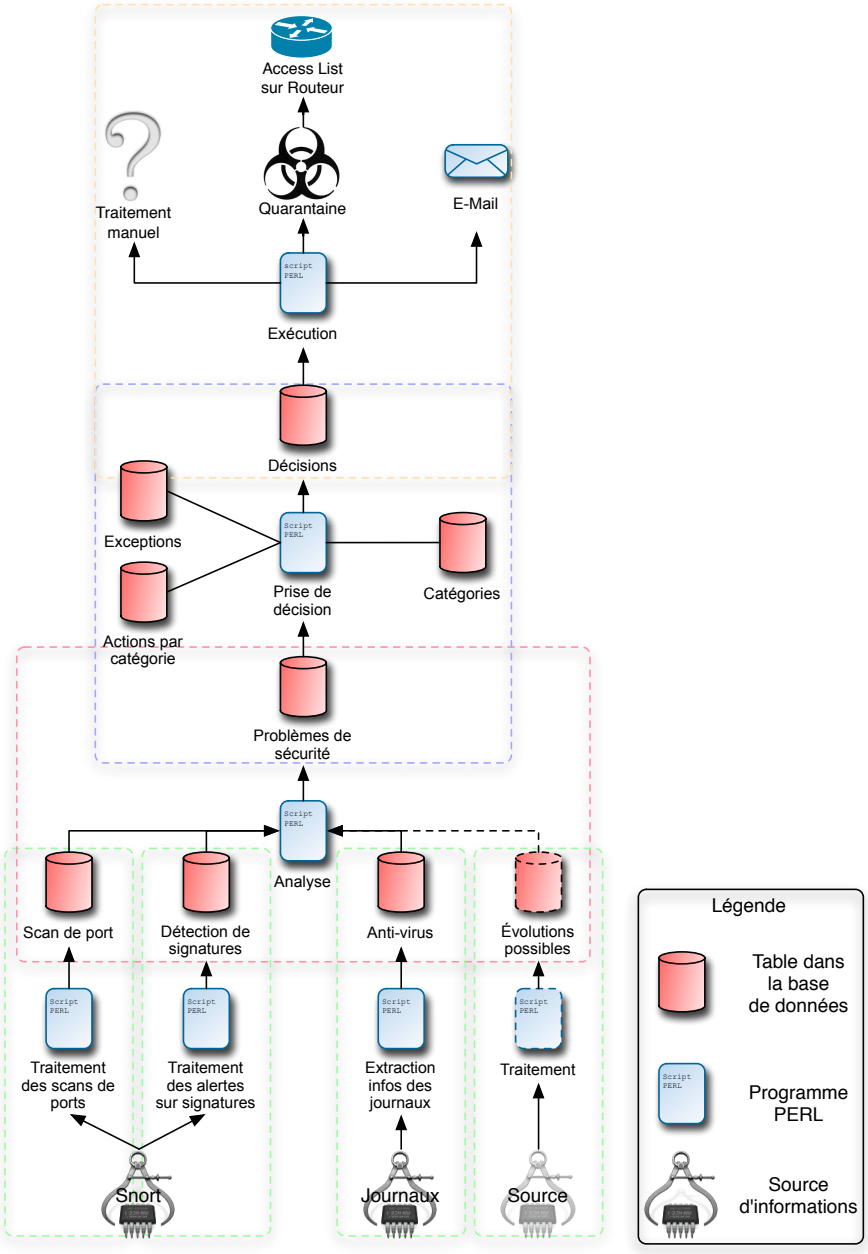

FIG. 4.1 – Principe de fonctionnement du système

4.4.1 Les sources d'informations

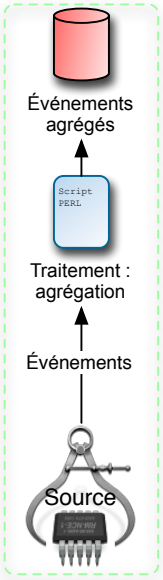

FIG. 4.2 – Traitements des *événements de sécurité* provenant des sources

La base du système est composée de différentes sources d'informations. Ces sources sont pour le moment les suivantes :

- la sonde de détection d'intrusions fournit les informations suivantes :
  - détection de scan de ports[2] ;
  - détection de signatures[3] dans le flux réseau ;
- journaux de l'antivirus du serveur de courrier électronique de REAUMUR.

Les éléments qui proviennent de ces sources sont qualifiés *d'événements de sécurité*.

Il faut noter que ces sources d'informations sont quantitativement très riches, mais les informations sont souvent récurrentes. Prenons un exemple : un virus a contaminé une machine, lorsqu'il va se dupliquer et envoyer des courriers électroniques infectés, chacun va déclencher un *événement* de l'antivirus. Par ailleurs, la sonde de détection d'intrusion lorsqu'elle détecte des communications correspondantes à ses signatures génère autant *d'événements* que de paquets de données dans la communication.

Il est donc nécessaire d'agréger ces informations de manière à ne pas remplir inutilement la base de données tout en disposant des informations nécessaires et suffisantes pour pouvoir ensuite analyser le *problème*. La figure 4.2 page 29 illustre ce mécanisme.

[2]Un scan de ports sur un réseau est une action qui vise à détecter les failles potentielles dans la sécurité des machines connectées à ce réseau : cet *événement* est souvent un signe d'acte de piratage ou de présence d'un virus.

[3]Il est possible de caractériser un flux réseau finement en analysant son contenu et en le comparant à des signatures répertoriant les différents comportements de pirates, virus ou encore logiciels utilisant le réseau.

Des sources supplémentaires peuvent être ajoutées. Il suffit de créer :

- un élément traduisant les informations rendues par la source en *événements* agrégés dans une table,
- une table contenant ces *événements* agrégés,
- une fonction dans le programme d'analyse pour gérer ce qui provient de cette nouvelle table.

4.4.2 L'analyse des données : plusieurs *événements de sécurité* pour un unique *problème de sécurité*

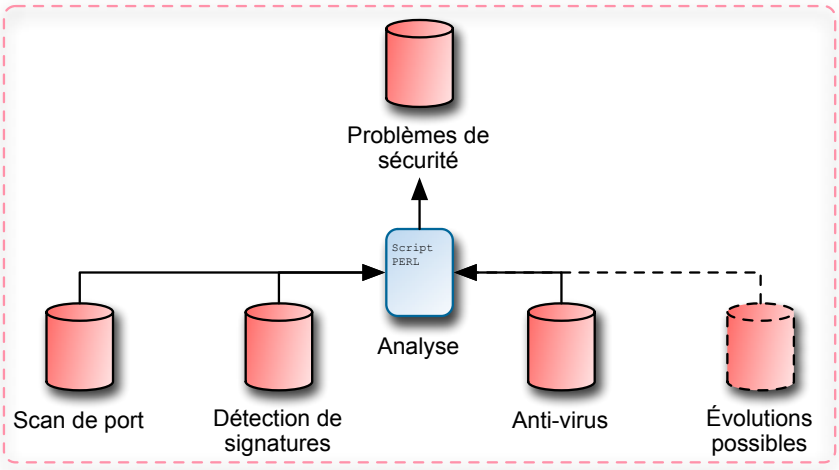

FIG. 4.3 – Analyse et seconde agrégation des *événements de sécurité* provenant des sondes

Une fois les *événements de sécurité* analysés, il en découle des *problèmes de sécurité*.

Une première agrégation est donc faite au niveau des sources d'informations mais il reste encore à pouvoir faire des recoupements entre ces différentes sources. En effet, un même *problème de sécurité* peut avoir pour origine plusieurs types *d'événements* sur les différentes sources. Par exemple, un virus n'utilise généralement pas qu'un seul vecteur de contamination. Il va à la fois envoyer des courriers électroniques et effectuer un scan du réseau. Ainsi, l'antivirus va détecter les courriers électroniques contaminés et la sonde de détection d'intrusion va détecter le scan. Comme ces deux *événements* correspondent au même *problème de sécurité*, le système doit faire le lien entre eux.

Pour pouvoir répondre simplement à ce besoin de recoupement, j'ai fait le choix d'affecter des catégories à chaque type *d'événement* précis. Par exemple, un scan sur certains ports va être catégorisé comme étant le fait d'un virus tandis qu'un *événement* provenant de l'antivirus sera lui aussi dans cette catégorie. Ces catégories sont stockées dans la base de données et leur nombre est relativement faible.

Les différents *événements* provenant des sources sont dans leurs tables respectives. Un programme va donc les lire de manière à agréger les *événements* en fonction de leur catégorie. Ainsi, la table des *problèmes de sécurité* va recevoir un seul *problème* pour un ensemble *d'événements* d'une même catégorie sur une machine donnée. La figure 4.3 page 30 représente ce mécanisme.

4.4.3 La prise de décision et l'action en découlant

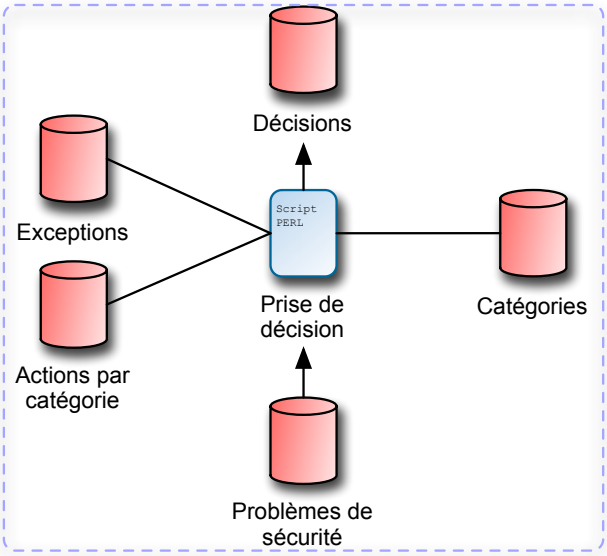

FIG. 4.4 – Prise de décision en fonction des paramètres

Les *événements* qui représentent le même *problème* sont donc regroupés dans des catégories. Par exemple, l'utilisation de clients Peer to Peer tel *Kazaa*, *Edonkey*, *Emule* va générer des *événements* distincts mais ceux-ci font partie de la même catégorie : Peer to Peer. Il suffit d'associer un comportement à chaque catégorie pour voir se profiler une action automatisée répondant à chaque *événement*. Par exemple, la catégorie Peer to Peer entraîne une mise en quarantaine de la machine source des *événements*.

Ainsi, la prise de décision se fait dès lors qu'un nombre suffisant *d'événements* se produit pour éviter les faux positifs[4]. De plus, il faut que le système puisse prendre une décision. En effet, pour certaines catégories, on sait que les informations doivent être analysées par un ingénieur. Le système pose donc une question dans ce cas.

La prise de décision est faite suivant la catégorie. La décision associe donc le *problème* à l'action qui en découle. De plus, une liste d'exceptions permet d'éviter de mettre cer-

[4]Faux positifs : les systèmes de détection d'intrusion ne sont pas exempts de défauts. Il est parfois possible que des comportements réseaux légitimes soient considérés comme anormaux. Cependant, ce genre d'erreur est très rare.

taines machines vitales (serveurs de courrier, serveurs d'application, etc.) en quarantaine. Ce processus est illustré par la figure 4.4 page 31.

Il est prévu de pouvoir effectuer des modifications du comportement du système pour les catégories de manière à ce que les utilisateurs puissent personnaliser ce comportement. Cependant, les actions choisies ne peuvent être que plus sévères que celles faites par défaut.

Les actions possibles sont :

- demander l'analyse manuelle,
- avertir le correspondant d'un site par courrier électronique lorsqu'un *problème* est détecté,
- mettre en quarantaine une machine.

Les décisions sont stockées dans une table et chacune associe donc un *problème* de sécurité à l'action correspondante.

4.4.4 Une fois la décision prise, son application

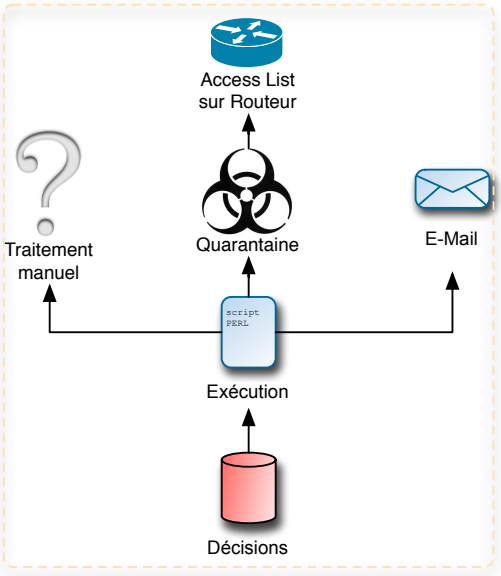

Fig. 4.5 – Exécution des décisions prises par le système

Lorsque la décision est prise par le système, elle est donc appliquée sur le réseau. Un programme va effectuer les actions en attente. Ces actions sont associées aux décisions qui ne sont pas encore traitées. La figure 4.5 page 32 illustre cette dernière étape.

Les courriers électroniques sont envoyés aux correspondants concernés et sont reçus en copie par les responsables sécurité de REAUMUR.

La mise en quarantaine s'effectue en plaçant des règles de filtrage sur un routeur de REAUMUR. Potentiellement, d'autres moyens de filtrages peuvent être mis en oeuvre.

Enfin, la demande de traitement manuel se fait par courrier électronique aux responsables sécurité de REAUMUR uniquement.

4.4.5 L'interaction des utilisateurs avec le système

Une fois les *problèmes* de sécurités traités au niveau de REAUMUR, les machines mises en quarantaine, si nécessaire, et les correspondants avertis, il faut encore que ceux-ci éliminent le *problème*. Ils peuvent ensuite directement interagir avec le système pour signaler à celui-ci que le *problème* est résolu. Il s'agit de la procédure d'acquittement.

Dans certains cas, l'utilisateur final peut avoir les connaissances nécessaires à l'élimination du *problème* (par exemple en utilisant un antivirus). Il se voit offrir la possibilité d'interagir avec le système. Ainsi, il peut signaler que le *problème* est résolu et donc acquitter celui-ci. Cependant, si jamais le *problème* est toujours présent et qu'une mise en quarantaine est de nouveau effectuée, l'utilisateur final ne pourra plus effectuer d'acquittement pour une journée et devra contacter le correspondant responsable de son site. C'est une manière de se protéger contre d'éventuels abus d'utilisateurs qui enfreignent sciemment la charte REAUMUR.

Chaque acquittement est enregistré de manière à pouvoir retracer les actions effectuées sur le système.

4.5 Développement du système

Point par point, je vais faire maintenant le lien entre la conception et l'écriture du code du système en explicitant les choix que j'ai eus à faire et les difficultés auxquelles j'ai été confronté.

4.5.1 Pour les sources d'informations

A la base du système, il y a donc les trois sources d'informations que j'ai évoquées précédemment, à savoir les résultats de scans de ports (Snort), les détections de signatures dans le flux réseau (Snort) et les comptes-rendus de l'anti-virus du serveur de courrier électronique.

Globalement, les trois sources peuvent être considérées comme de la lecture de fichier texte :

- pour les scans de port, Snort écrit directement dans un fichier ses comptes rendus,
- pour la détection de signatures, Snort écrit dans des fichiers binaires qui vont être ensuite convertis par un autre logiciel en format texte (voir la section 3.3.2 page 20 pour l'explication sur ce point) pour des questions de performance,
- les comptes-rendus de l'anti-virus sont écrits dans les journaux du système.

Le travail à effectuer dans les trois cas consiste à lire ces informations et à conserver uniquement les données utiles pour ensuite ne stocker dans la base de données que le strict nécessaire.

Le fonctionnement interne du module qui fait cette agrégation est relativement simple. Il est décrit sur la figure 4.6 page 34.

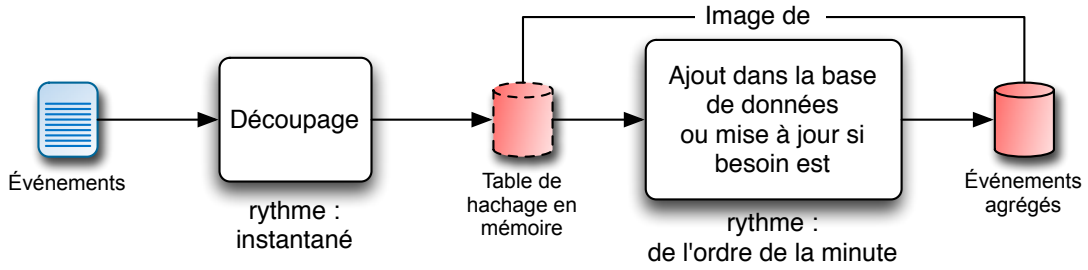

FIG. 4.6 – Fonctionnement de l'agrégation des *événements* provenant des sources

Le flux d'entrée (le texte) est découpé de manière à disposer des informations utiles pour chaque ligne lue. Pour éviter d'accéder de manière incessante à la base de données, ces informations sont écrites en mémoire vive dans une table de hachage[5]. Celle-ci est indexée suivant les éléments constituant la clef primaire[6] de la table correspondante dans la base de données.

La table de hachage et la base de données contiennent au final les mêmes informations mais chacune a un rôle précis. La table de hachage sert de mémoire de travail tandis que la base de données sert de mémoire à long terme et surtout permet l'interaction avec le reste du système.

Lorsqu'un élément se répète, il va avoir le même index que l'occurrence précédente dans la table et seules certaines informations vont être mises à jour (avec donc écrasement des anciennes valeurs). Ces informations sont le nombre d'occurrences de *l'événement* et la date de la dernière occurrence.

Pour minimiser les accès à la base de données, ceux-ci sont faits à intervalles réguliers et uniquement s'il y a une information à ajouter ou mettre à jour. Lorsqu'il y a des opérations à effectuer, il faut savoir quels sont les éléments à traiter dans la table de hachage pour les différencier de ceux déjà traités. Plutôt que de mettre cette information directement dans la table que j'ai définie auparavant, j'ai fait le choix de créer des tables contenant uniquement les index des éléments devant être ajoutés ou mis à jour dans la base de données. Ceci permet d'éviter de parcourir tous les éléments de la table de hachage pour savoir lesquels doivent faire l'objet d'une opération avec la base de données.

Cependant, cette structure a un inconvénient. Lors du redémarrage d'un module, il est nécessaire de synchroniser la table de hachage avec la base de données.

[5]Une table de hachage est un tableau dans lequel les informations ne sont pas indexées suivant des numéros mais par des clefs. Celles-ci peuvent être de n'importe quel type (mot, chiffre, etc.). Le gros avantage d'une telle structure de données est la possibilité d'avoir un accès direct à chaque élément du tableau sans avoir à le parcourir.

[6]Élément identifiant de manière unique une ligne dans une table d'une base de données.

4.5.2 L'analyse des données

Voyons maintenant comment les informations provenant des différentes sources sont traitées pour les regrouper en fonction des *problèmes* qu'elles traduisent.

La figure 4.7 page 35 illustre ce mécanisme.

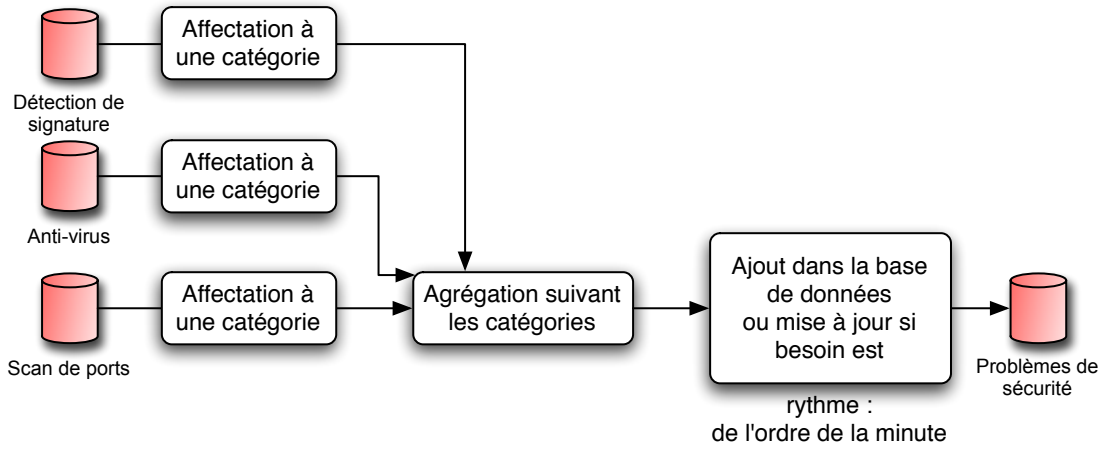

FIG. 4.7 – Fonctionnement du classement en catégories et de l'agrégation suivant celles-ci

Suivant le principe expliqué auparavant, les *problèmes* sont distingués par des catégories. Pour affecter un *événement* (d'une source d'information) à une catégorie, il faut plus ou moins de travail en fonction de l'information.

L'information la plus précise est celle qui provient de la détection de signature. En effet, chaque *événement* dispose d'un identifiant et il est possible de définir à quelle catégorie appartient chacun d'entre eux.

L'information la plus simple est celle qui vient de l'anti-virus. Tout *événement* provenant de cette source correspond tout simplement à la catégorie *virus*.

Par contre, les *événements* provenant de la détection de scan de ports sont plus difficiles à classer. En effet, il est nécessaire d'effectuer un traitement de ces informations, à savoir comparer le comportement observé par rapport à des comportements types, chacun de ceux-ci étant associé à une catégorie.

Une fois chaque *événement* affecté à une catégorie, un recoupement suivant celles-ci est effectué. De cette manière, on obtient les *problèmes de sécurité* que l'on peut stocker dans la base de données.

4.5.3 La décision

La prise de décision se fait de manière simple. Seuls les *problèmes de sécurité* qui ont été modifiés ou qui sont nouveaux sont lus. Le programme vérifie si les machines à l'origine

des *problèmes* ne sont pas dans les exceptions. Il écrit ensuite dans la table des décisions les actions correspondant à la catégorie de chaque *problème de sécurité*. Dans les faits, il s'agit de requêtes dans la base de données.

4.5.4 L'exécution de la décision

Lorsque la décision est prise par le système, celle-ci n'est pas instantanément appliquée. Par exemple, dans le cadre de la propagation d'un virus, si toute nouvelle détection de celui-ci entraîne une action directe, il y a un risque de saturer l'équipement sur lequel nous mettons en place la quarantaine.

Le programme qui exécute les décisions va donc regrouper les changements à effectuer sur les équipements et exécuter ceux-ci de manière régulière.

4.6 Interface graphique

Dans la recherche de solutions, le choix a été fait d'utiliser une interface WEB dynamique. Le langage choisi pour construire cette interface est le PHP. Cependant, l'utilisation de PHP seul ne permet pas facilement de disposer d'une interface multilingue et permettant des vues différentes suivant les utilisateurs. Des outils permettent de simplifier la réponse à ces contraintes : ce sont les *moteurs de templates*.

4.6.1 Utilisation de *moteurs de templates*

Les *moteurs de templates* permettent une séparation entre le fond (la partie programmation) et la forme (la présentation des pages affichées).

Leur utilisation est relativement simple. La partie fixe des pages s'écrit en HTML classique et contient certains marqueurs associés au *moteur de templates*. Ces marqueurs représentent les données dynamiques que l'on insère dans les pages. La partie programmation se fait en PHP classique et certaines commandes associées aux *moteurs de templates* permettent d'assigner des données aux marqueurs.

De cette manière, il est possible d'utiliser le même code PHP avec différentes pages HTML qui reprennent les marqueurs que l'on souhaite. Dans notre cas, le même code PHP va permettre avec plusieurs pages HTML de disposer de vues spécifiques à chaque utilisateur (niveau variable de détails dans les informations) et potentiellement, il est possible d'écrire les pages HTML dans d'autres langues.

4.6.2 Choix du *moteur de templates*

Il existe une multitude de *moteurs de templates* dans les logiciels libres. Dans le cadre du système, nous souhaitions disposer d'un outil pérenne. Or, un projet est actuellement soutenu par l'organisation responsable de PHP. Il s'agit de *Smarty*. De plus, celui-ci est

fourni dans la distribution GNU/Linux que nous utilisons (Debian), ce qui nous garantit des mises à jours simples.

C'est donc vers cet outil que je me suis tourné pour créer l'interface du système.

4.6.3 Sécurité

Les applications écrites en PHP peuvent faire l'objet d'attaques. Les pirates cherchent généralement à prendre le contrôle des machines sur lesquelles sont exécutées ces applications. Elles représentent une porte d'entrée potentielle si elles sont mal écrites.

L'interface du système ne concerne que les utilisateurs du campus. Le risque d'avoir un pirate sur le campus est relatif car les menaces proviennent généralement de l'étranger. Mais, il peut toujours y avoir des étudiants en informatique « doués » qui, par jeu, peuvent tenter de pirater le système.

La programmation de l'interface a donc été faite suivant l'idée qu'il y avait des menaces potentielles sur le campus. Cela se traduit par un souci constant de protection des informations contre leur modification et leur lecture non autorisées.

Il faut donc pouvoir faire la distinction entre les différents types d'utilisateurs qui accèdent au système. Cette distinction se traduit par l'authentification des utilisateurs.

4.6.4 Authentification

L'authentification a été conçue de manière solide pour qu'il n'y ait pas de possibilité d'usurper l'identité d'un utilisateur.

Pour se faire, nous utilisons les certificats X.509[7]. Ils offrent une authentification forte et quasi inviolable (quasi car aucune protection n'a le mérite d'être infaillible). Cependant, tous les utilisateurs ne disposent pas de certificats sur le campus. Il est nécessaire d'offrir un autre moyen.

Dans le cadre du déploiement du WiFi[8] sur le campus, un outil est en cours de mise en oeuvre par un stagiaire ENSEIRB (2ème année filière informatique) : Mathieu Geli. Cet outil est *Shibboleth*. Il permet l'authentification d'utilisateurs enregistrés sur les différents annuaires des universités. C'est un système qui permet le SSO[9]. Sa mise en oeuvre est très simple au niveau du système et sera utilisée pour la mise en production.

[7]Certificats X.509 : un certificat permet, comme son nom l'indique, de certifier qu'un utilisateur ou une machine sont biens ceux qu'ils prétendent. Le standard X.509 est le plus utilisé.

[8]WiFi : *Wireless Fidelity*, technologie de réseau sans fil en plein essor.

[9]SSO : *Single Sign On*, l'authentification est faite une seule fois et permet d'aller, sans avoir à ressaisir son mot de passe, sur toutes les applications utilisant cette méthode.

4.6.5 Présentation des informations et des interactions possibles

Il faut rappeler que les correspondants ont des niveaux en informatique très variables. L'interface a donc été conçue pour présenter les informations les plus simples en premier lieu. L'accès à des informations plus complètes et complexes est possible mais n'est pas présentée en premier pour ne pas submerger les correspondants non informaticiens.

Les interactions possibles (tel l'acquittement) sont directement accessibles dans la présentation simple des *problèmes*.

Chapitre 5

Mise en production

Dès lors que le développement arrive à un stade suffisamment avancé (c'est-à-dire, lorsque chaque sous-programme fait le travail qu'il a à faire dans son environnement de données), il faut faire travailler les différents sous-programmes dans un contexte réel (à savoir sur les informations transitant réellement sur le réseau).

Ces essais se font en plusieurs étapes qui correspondent à des versions du système.

La première version s'appelle la version alpha. Il s'agit d'une version qui n'a jamais subi de test dans son ensemble. C'est une version uniquement testée en interne et qui permet d'affiner le développement.

Elle est suivie par une version beta. Elle se distingue de la version alpha du fait qu'elle a déjà subi des tests et est accessible à des utilisateurs volontaires. Cette version permet d'apporter les dernières modifications avant d'obtenir une version qu'on qualifiera de stable.

5.1 Première étape : version alpha

Au mois de mars 2005, nous avons décidé de lancer la première phase de test grandeur nature du système : la version alpha.

5.1.1 Le système

Le système n'était alors que partiellement construit. Les éléments fonctionnels étaient :

- le traitement des alertes sur signatures,
- le traitement de la détection de scan de ports,
- l'analyse et l'agrégation,
- la prise de décision,
- l'exécution semi-automatique mais pas l'envoi de courriers électroniques.

Il manque le traitement des informations provenant de l'antivirus et l'exécution automatique avec envoi de courriers électroniques. En effet, pour éviter tout risque de mise en quarantaine erronée, chacune d'entre elles doit être validée avant sa mise en oeuvre effective.

5.1.2 L'interface

L'interface est alors très sommaire. Elle présente sur une seule page les différentes actions possibles et ne dispose d'aucune sécurité hormis son isolation du reste du campus. Elle est à usage strictement interne (visible uniquement depuis nos machines).

5.1.3 Résultats

Par ce test, nous avons pu constater le relatif bon fonctionnement des différents modules en conditions réelles. Mais des problèmes sont apparus quant à la fiabilité du système dans le cas d'un fonctionnement prolongé. Ces problèmes ont été analysés par mon maître d'apprentissage et moi même. A la suite des observations que nous avons pu faire j'ai donc apporté les corrections nécessaires et continué le développement des éléments manquants.

De manière à valider les modifications, chacune d'entre elles a été intégrée dans la version alpha. En parallèle l'interface a été améliorée.

Une fois le système arrivé à un niveau satisfaisant de fiabilité et de fonctionnalité nous avons donc décidé de qualifier la version alpha courante comme étant la version beta.

5.2 Deuxième étape : version beta

Le système à ce stade est jugé comme en mesure de fonctionner dans son contexte réel (avec des utilisateurs). Le 23 juin 2005, le système a donc été mis en pré-production et des accès ont été ouverts pour certains correspondants.

5.2.1 Le système

Tous les éléments du système étaient prêts. Cependant, nous avons décidé de ne pas encore laisser le système fonctionner de manière totalement automatique. Durant deux semaines, les envois de courriers électroniques et les mises en quarantaines ont été validés manuellement.

Le 11 juillet, le système a été rendu autonome.

Au fur et à mesure que d'éventuels problèmes seront observés dans le fonctionnement, les corrections associées seront apportées et intégrées à la version beta. Lorsque nous (équipe et testeurs) n'observerons plus de problèmes, le système aura atteint un état de fonctionnement satisfaisant et pourra être validé.

5.2.2 L'interface

Le *moteur de templates Smarty* est donc utilisé et permet actuellement les vues multiples sur le système. L'authentification n'est pas encore dans son état final. Celle-ci se fait pour

l'instant par certificats X.509 mais durant l'été, l'authentification par *Shibboleth* va être mise en oeuvre.

L'interface en elle-même doit être validée par les correspondants et modifiée suivant leurs retours.

5.3 Version de production stable livrée pour septembre après validation

Une fois que le système et son interface seront validés, au plus tard début septembre, il sera possible de qualifier le système comme étant dans sa version de production. A partir de celle-ci, une nouvelle version verra le jour : la version de développement. Cette version sera indépendante de la version de production, et évoluera dans le cadre de sa diffusion dans la communauté des logiciels libres.

Il est fort probable que le système et son interface soient validés avant septembre mais durant l'été une grande partie des correspondants et utilisateurs du réseau sont absents. Il est donc préférable de faire le lancement au mois de septembre.

Chapitre 6

Bilan du projet

6.1 Réponse au cahier des charges fonctionnel

La réponse au cahier des charges fonctionnel peut être mesurée sur la version actuellement en fonctionnement.

Cette réponse est la suivante pour les fonctions principales :

Fonction	FP1	FP2	FP3	FP4	FP5	FP6	FP7
Réponse	totale	totale	grande	grande	totale	faible	totale

et pour les contraintes :

Fonction	FC1	FC2	FC3	FC4	FC5	FC6	FC7	FC8
Réponse	totale	à valider	non fait	totale	faible	totale	totale	totale

Fonction	FC11	FC21	FC22	FC31	FC32
Réponse	non fait	faible	totale	grande	faible

Pour les fonctions traitées partiellement ou non traitées à ce jour, les raisons sont les suivantes :

FP3 Tous les événements détectés sont visibles, les archives aussi, mais les statistiques ne sont pas encore faites.

FP4 Idem FP3 mais du point de vue des correspondants.

FP6 Les correspondants ne peuvent pas personnaliser le comportement du système pour leur domaine actuellement. Mais la structure de la base de données a été prévue pour le permettre. Par contre, la partie de l'interface offrant ce service n'est pas écrite.

FC2 Les correspondants participant au test de la version beta vont valider ou non si l'interface est intuitive et ergonomique.

FC3 L'interface machine/machine n'a pas été développée car l'enquête a montré que celle-ci n'intéressait qu'une faible partie des correspondants. Le développement, la sécurisation, les tests auraient demandés un temps non négligeable.

FC5 Il est prévu de pouvoir effectuer des mesures, celles-ci jusqu'à maintenant ont été faites manuellement, mais il n'y a actuellement pas de programme écrit pour faire ces

mesures.

FC11 La distribution sous forme de paquet Debian se fera lors de la mise à disposition du système dans la communauté des logiciels libres.

FC21 Le code est actuellement en grande partie commenté en anglais, mais la documentation technique n'est pas encore prête.

FC31 Les évolutions de débits ne posent pas de problème par le choix des outils qui a été fait. Par contre, l'évolution à IPV6 nécessitera quelques petites modifications sur certaines portions de code (qui gèrent les adresses IP).

FC32 Actuellement, seule la version française de l'interface est prête, mais l'utilisation de *moteurs de template* permettent facilement l'utilisation d'une autre langue pour l'interface. Il reste cependant à traduire tous les textes.

Au final, nous estimons avec mon maître d'apprentissage (qui estime le projet terminé à 90%) que le projet répond en grande partie aux besoins, le manque principal est la documentation technique complète. Celle-ci devrait être terminée d'ici la mise en production finale.

6.2 Respect du planning

Le planning effectif (figure 6.1) du projet est le suivant :

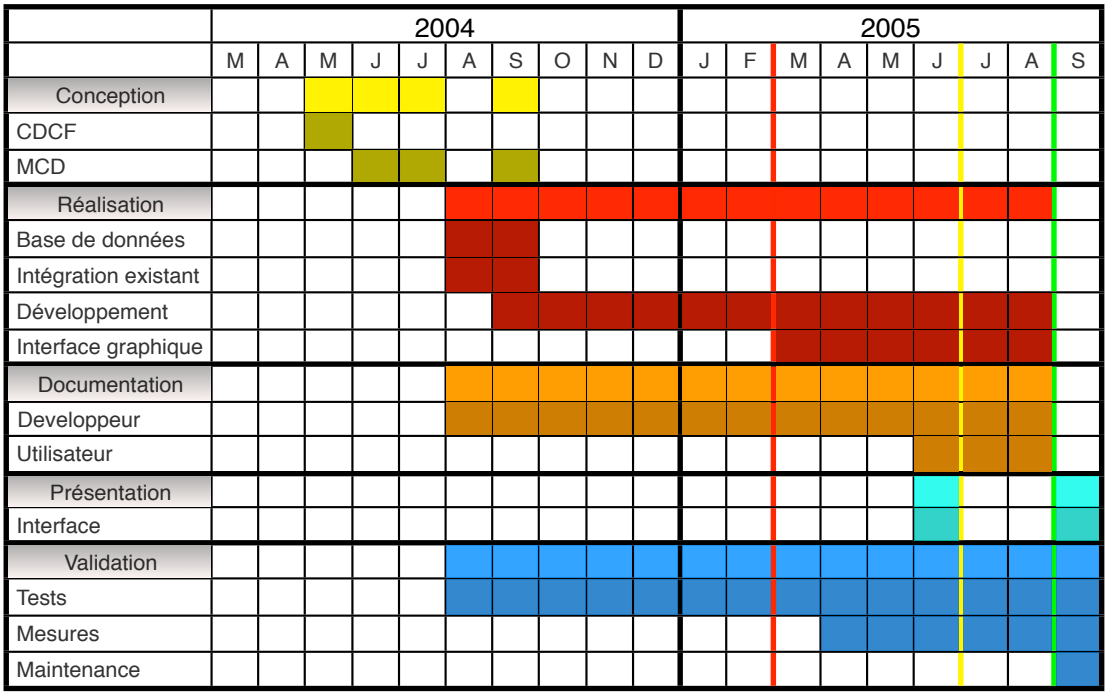

Fig. 6.1 – Planning effectif du projet

La ligne rouge correspond à la version alpha, la jaune à la version beta et la verte à la version de production.

En comparant le planning prévisionnel et le planning effectif, on peut distinguer plusieurs écarts :

- un premier retard sur le début du projet.
  En effet durant les mois de mars et d'avril des tâches qui n'étaient pas prévues m'ont été attribuées par mon maître d'apprentissage. Il s'agissait de la configuration et de l'installation d'une dizaine de routeurs dans le cadre du renouvellement des liaisons avec des sites distants ;
- un écart global entre les durées estimées et les durées effectives pour les tâches de conception et de réalisation.
  Les tâches quotidiennes ont été sous-évaluées et mon implication dans le service a été plus importante que prévue (consommation de 50% de mon temps). Enfin, j'ai sous-estimé le temps de conception et de réalisation.

6.3 Coût du projet

Le coût du projet peut être vu sous 2 angles : d'une part le coût effectif (ce qu'il a coûté) d'autre part le coût potentiel (ce qu'il aurait pu coûter si un autre choix de solution avait été fait).

6.3.1 Coût effectif

Pour la validation du thème de mémoire, un budget prévisionnel avait été fixé. Il était de 20 000 € et se répartissait ainsi : 15 000 € pour mon salaire et 5 000 € pour le matériel. Or je n'ai pas passé tout mon temps sur le projet et le matériel acheté a été moins onéreux que prévu.

Ainsi, le coût de développement du projet est simple à calculer. Il a fallu assurer ma rémunération pendant 15 mois (durée totale du projet) avec un ratio d'activité sur le projet de l'ordre de 50%, ceci nous donne 7 mois et demi de salaire apprenti (soit environ 7 500 €). À cela, il faut ajouter le prix de la machine hébergeant le système : 2 000 €. Ce qui donne un total de 9 500 €.

Si on considère le coût global, il ne faut pas se limiter au coût d'achat et de développement mais prendre aussi en compte le coût de la maintenance.

La maintenance du matériel est assurée par la société qui nous a fourni la machine (coût de l'extension de garantie 3 ans compris dans le prix donné précédemment). Par contre, au niveau logiciel, une fois le système validé en production, les bugs résiduels seront en nombre relativement faible. De plus, étant donné que le système va être distribué sous forme de logiciel libre, une grande communauté de développeurs pourra participer à sa maintenance.

Il reste maintenant la gestion des différents développeurs qui participeront à l'amélioration du logiciel en fonction de nouveaux besoins. Actuellement, rien n'est prévisible, d'autant plus que les améliorations potentielles amèneront des gains non cal-

culés.

Le coût de maintenance peut être considéré comme très inférieur au coût de développement.

Au final, le coût effectif du projet est inférieur à 10 000€, soit la moitié du budget prévisionnel.

6.3.2 Coût potentiel

Les solutions sur lesquelles nous aurions pu nous appuyer sont :

- Cisco PIX 525 : 9 930 €[1],
- Cisco ASA 5540 : 14 060 €,
- CheckPoint Enterprise : 17 400 €,
- Allot NetEnforcer : 24 900 €.

Il faut aussi compter les temps d'installation et de configuration mais ceux-ci me sont inconnus. De plus, il y a un coût supplémentaire non négligeable dans ce type de solution : les abonnements. En effet, il est nécessaire de souscrire des contrats pour que les systèmes soient maintenus à jour. De tels abonnements se chiffrent aux environs de 1 000 € par an. A ce coût il faut ajouter la maintenance du matériel (de l'ordre de 10% par an).

La seule solution comparable économiquement avec le coût effectif du développement ne répond ni à nos contraintes ni totalement à nos besoins.

D'un point de vue économique et par rapport aux fonctionnalités souhaitées, nous pouvons conclure qu'il est globalement plus avantageux d'utiliser un apprenti et les logiciels libres que de mettre en place une solution commerciale.

6.4 Gains

Ce projet apporte des gains de temps, ce qui est d'ailleurs son objectif premier. Il permet aussi une meilleure interaction avec les correspondants techniques de REAUMUR.

6.4.1 Gain de temps

Pour la validation du thème de mémoire, le temps perdu sur les tâches répétitives (et automatisables) a été estimé à 1 heure 15 minutes par jour (en temps ingénieur), ce qui fait environ 700 € par mois[2]. On obtient le graphique prévisionnel de retour sur investissement suivant :

Le premier objectif du projet est de gagner 75% de temps sur les tâches de sécurité automatisables. Lors des tests (version alpha et beta) des mesures ont été faites, le résultat

[1] Prix au mois de Juin 2005

[2] Charges comprises mais sans compter les frais de structure

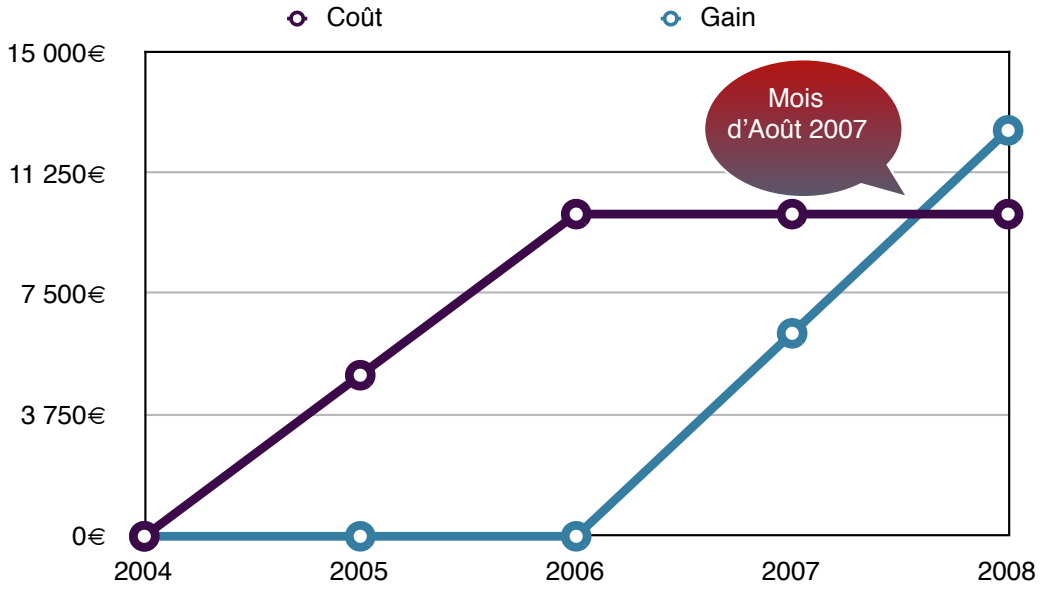

FIG. 6.2 – Retour sur investissement

est que 90% des problèmes détectés sont traités automatiquement. Les problèmes restants (10% des problèmes) ont un temps de résolution relativement court (de l'ordre de 50% du temps qu'il fallait auparavant pour traiter les problèmes). Le temps gagné est donc de 95%. Nous sommes donc très proche du retour sur investissement prévisionnel.

Le deuxième objectif du projet est d'avoir des interactions instantanées avec les correspondants pour 90% des problèmes de sécurité. Celui-ci n'a pu être mesuré pour l'instant pour cause de manque de données.

6.4.2 Gain en termes d'image et de sécurité

Le projet permet aux correspondants techniques d'avoir une meilleure visibilité sur notre gestion de la sécurité et surtout de pouvoir agir directement sans avoir à s'adresser à une personne de l'équipe. Ces fonctionnalités permettent d'améliorer l'image de REAUMUR auprès des correspondants.

Au-delà du campus et de la région, le projet va être mis à disposition de la communauté des logiciels libres (particulièrement les utilisateurs universitaires) et va potentiellement permettre ou faciliter des relations avec d'autres universités pour l'amélioration du système, voire pour le développement de nouveaux outils communs.

Le projet permet enfin et surtout d'augmenter globalement la sécurité du réseau du fait de l'activité du système 24 heures sur 24 et 7 jours sur 7. Cette amélioration de la sécurité a deux effets : un sur les conséquences liés à la non détection de problèmes de sécurité qui engendre potentiellement des fuites d'informations ou des saturations réseau ; l'autre

sur l'image que nous donnons vis à vis de Renater qui peut constater que le nombre de problèmes chez nous diminue.

6.5 Bilan humain

Au niveau humain, le projet a une incidence certaine du fait qu'il peut agir sur toute machine connectée à REAUMUR (à savoir environ 10 000 machines pour environ 50 000 utilisateurs). De plus, il touche directement les correspondants de REAUMUR (au nombre, pour rappel, de 200) qui doivent ensuite réagir par rapport aux problèmes détectés.

Etant donné que les principaux utilisateurs du système sont les correspondants, leurs besoins ont été relevés et certains volontaires ont pu participer à la finalisation du système.

Enfin, comme dans tout système interagissant avec des utilisateurs, il y a bien évidemment une documentation et une présentation formelle de celui-ci.

6.5.1 Gestion du changement et de la mise en quarantaine

Les correspondants se voient donc apporter un outil supplémentaire qui induit des changements dans leur relation à REAUMUR. Pour favoriser leur acceptation, il a donc été fait une enquête de manière à orienter le développement du système suivant leurs besoins. De plus, les remarques potentielles sont prises en compte et le système est modifié suivant celles-ci. Les correspondants sont donc, dans la mesure du possible, intégrés dans le cycle de développement.

Les utilisateurs finaux voient parfois leur machines mises en quarantaine que ce soit de leur faute (abus) ou non (virus). Ils disposent désormais d'informations précises sur le problème qu'il y a sur leur machine. Ils peuvent surtout eux-mêmes le résoudre et demander une prise en compte immédiate par le système.

6.5.2 Documentations, présentation et formation

Tout projet de développement doit être accompagné d'une documentation technique, de manière à permettre la maintenance du système par une autre personne que le développeur initial.

Du point de vue des correspondants, une documentation d'utilisation va être faite d'ici la mise en production.

De plus, une première présentation de l'interface du système a été faite lors de la réunion du Groupe des Utilisateurs de REAUMUR (GUR). Cette réunion d'une durée de 2 heures qui comprenait les correspondants des principaux sites (50 personnes) a eu lieu le 23 juin dernier. Mon intervention présenta durant 20 minutes la réalisation et l'utilisation du système.

Chapitre 7

Bilan de compétences

Sur toute la durée du projet j'ai développé de nouvelles compétences et utilisé des connaissances que j'ai acquises dans le cadre de la formation.

Ces compétences sont d'ordre scientifique, technique, organisationnel, économique et humain.

7.1 Domaine scientifique et technique

Je vais détailler ici les différentes connaissances scientifiques et techniques que j'ai pu mettre en pratique et développer.

Dans le cadre de l'analyse, la rédaction du cahier des charges et la conception du système j'ai utilisé les cours de T.Delamer en système d'information et analyse fonctionnelle. J'ai pu mettre en pratique dans le service les méthodes vues et appliquées durant ces cours. J'ai ainsi établi le cahier des charges fonctionnel et le modèle conceptuel des données du système. Cependant, j'ai eu quelques difficultés à les construire, malgré les applications faites en cours.

Durant le développement je me suis appuyé sur les cours de R.Strandh en génie logiciel pour écrire un code modulaire et lisible. Cela favorisera la mise à disposition et l'évolution du système.

Le cours de sécurité réseau de R.Devreese m'a donné une vision globale de la sécurité que j'ai pu ensuite affiner suivant les besoins du projet.

Les TP et le projet du cours de conception de site WEB (PHP) de B.Hamid m'ont permis d'avoir la base de l'interface, créée avec mon binôme Grégory Thiell. J'ai pu la réutiliser et l'adapter aux besoins du projet.

J'ai approfondi certains sujets, qui m'ont permis d'acquérir des compétences complémentaires à la formation.

J'ai ainsi complété mes connaissances en sécurité réseau dans les domaines de la compromission de machines et de la détection de comportements suspects. Je dispose désormais d'une relative maîtrise de la sonde de détection d'intrusion Snort. J'ai aussi travaillé la

sécurisation de l'interface du système et du serveur l'hébergeant.

Dans le domaine du développement, j'ai appris à optimiser et tester les performances des programmes que j'ai réalisés. J'ai conforté mes compétences dans l'utilisation du langage PERL et enrichi ma connaissance de PHP.

7.2 Domaine organisationnel

Les cours de gestion de projet de T.Delamer et D.Thiers m'ont appris à gérer un planning, à cadrer les objectifs et à ne pas m'en écarter.

Dans la pratique j'ai été amené à définir un planning prévisionnel, à tenter de le respecter et à expliquer les écarts.

Pour définir les besoins précis des correspondants j'ai effectué une enquête. J'ai ainsi acquis une expérience dans la rédaction et l'exploitation d'une enquête.

7.3 Domaine économique

Les différents cours de J.P.Guichane (économie, relation donneur d'ordre-fournisseur) m'ont permis de définir un budget prévisionnel et de calculer le retour sur investissement du projet.

Cependant, le contexte de la fonction publique est relativement spécifique par rapport aux cas que nous avons pu étudier durant les cours. Dans la mesure du possible j'ai appliqué les notions que nous avons pu apprendre.

J'ai aussi pu voir le fonctionnement des marchés publics.

7.4 Domaine humain

Suivant les cours de management de C.Couralet j'ai appliqué les connaissances acquises sur la gestion du changement auprès des correspondants. Le facteur d'acceptation étant tout aussi important que la qualité technique du projet, j'ai donc essayé de favoriser celui-ci par l'enquête et la participation des correspondants dans le cycle de développement.

Le projet touche un grand nombre de personnes comme j'ai pu l'expliquer auparavant. Tout utilisateur final du réseau peut se trouver dans une situation déplaisante : le fait d'avoir sa machine en quarantaine. Il s'agit cependant d'un choix politique nécessaire au bon fonctionnement du réseau. Nous avons fait le choix de laisser la possibilité aux utilisateurs finaux d'enlever la quarantaine en place sur leur machine. De cette manière leurs réactions d'humeur contre REAUMUR sont diminuées.

J'ai pu présenter oralement devant un public relativement important (50 personnes) la réalisation et surtout l'interface du système.

Au niveau personnel, j'ai pu enrichir ma capacité à travailler en équipe dans un contexte de pointe au niveau technologique. Je dispose désormais d'une réelle expérience du travail que je n'avais pas en entrant dans la formation.

Chapitre 8

Perspectives

Maintenant que le système est fonctionnel, il est donc prévu d'ici la fin de l'année de le distribuer en tant que logiciel libre et de l'améliorer.

8.1 Distribution du système dans la communauté des logiciels libres

Les différentes institutions universitaires de par le monde ont toujours été étroitement liées aux logiciels libres. Ainsi, il est tout naturel pour nous de distribuer le fruit de notre travail dans cette communauté de manière à ce que des personnes qui ont les mêmes besoins participent à notre projet au lieu de développer de leur côté un système similaire.

Il faut aussi noter que la participation de personnes que l'on ne connaît pas toujours peut amener des contacts qui n'auraient pas eu lieu autrement et ainsi créer des liens avec les services réseau d'autres universités.

Nous envisageons d'utiliser la licence GPL (GNU General Public License). Celle-ci permet la modification et la redistribution du système, dès lors que l'ensemble du code source du logiciel est fourni. De manière générale, sauf s'il y a de grosses divergences entre les différents programmeurs, les modifications sont réintégrées dans le logiciel d'origine.

8.2 Amélioration de l'intelligence du système

Dans son état actuel, faute de temps pour le développement, le système est encore peu intelligent. Il agit suivant des schémas prédéfinis dans la base de données. Cependant, il est tout à fait possible d'aller vers une amélioration de l'intelligence du système. On peut imaginer que le système soit capable d'apprendre les actions à mener en fonction des choix des utilisateurs sur des problèmes similaires. Mais il faut, bien évidemment, que cet apprentissage soit fiable. Ce type d'amélioration peut être risqué et long à mettre en oeuvre.

On peut espérer que la distribution, sous forme de logiciel libre, permette cette

amélioration du fait de la participation de plus nombreux développeurs. Nul doute que cette amélioration passe par l'utilisation d'outils de l'intelligence artificielle (système expert).

8.3 Élargissement du périmètre du système

Actuellement le système fonctionne uniquement à partir des flux transitants entre REAUMUR et Renater, il est tout à fait possible d'élargir le périmètre du système en ajoutant des sondes à d'autres endroits (gestion des connexions sans fils, sécurisation de site particuliers) suivant les besoins des sites. Les actions menées par le système pourront alors se faire sur d'autres types d'équipements (par exemple les bornes WiFi pour couper l'accès à certaines machines).

Conclusion

Rappelons tout d'abord l'objectif du projet : disposer d'un système automatisé de sécurisation du réseau. Ce système permet de dégager du temps pour REAUMUR et ses correspondants, d'améliorer l'image du service et d'augmenter la sécurité sur le réseau.

Mon maître d'apprentissage a évalué que le projet est actuellement opérationnel à 90%, nous faisant déjà gagner du temps. Les correspondants pourront tous utiliser le système à la rentrée universitaire 2005. Nous pouvons donc considérer que l'objectif est atteint en grande partie.

Durant chaque étape de ce projet, j'ai pu utiliser les différents acquis provenant de la formation et de mon apprentissage. Mais j'ai pu surtout enrichir ces acquis et les compléter pour, au final, développer de nouvelles compétences.

Je dispose désormais de compétences concrètes en sécurité informatique et réseau et c'est ce que je souhaitais au commencement de mon apprentissage.

Mes employeurs m'offrent la possibilité d'un contrat de travail à durée déterminée dans le service REAUMUR. Cette opportunité me permettra de développer davantage mes compétences dans les domaines de la sécurité informatique et réseau.

Bibliographie

1. Zimmermann (Jacob), Mé (Ludovic), Les systèmes de détection d'intrusion : principes algorithmiques. Magazine MISC n°3 .- Sélestat : Diamond Editions, 2002 .- pages 24-30.
2. Debar (Hervé), Morin (Benjamin), Quelques problèmes liés au déploiement de systèmes de détection d'intrusions commerciaux aujourd'hui. Magazine MISC n°3 .- Sélestat : Diamond Editions, 2002 .- pages 31-37.
3. Bidou (Renaud), Concepts et contournement des IDS. Magazine MISC n°3 .- Sélestat : Diamond Editions, 2002 .- pages 38-45.
4. Malterre (Pascal), Les nouveautés de Snort 2. Magazine MISC n°13 .- Sélestat : Diamond Editions, 2004 .- pages 56-63.
5. Bidou (Renaud), Raynal (Frédéric), Canaux cachés (ou furtifs). Magazine MISC n°18 .- Sélestat : Diamond Editions, 2005 .- pages 32-38.

Table des figures

Glossaire

Certificats X.509 : un certificat permet, comme son nom l'indique, de certifier qu'un utilisateur ou une machine sont biens ceux qu'ils prétendent. Le standard X.509 est le plus utilisé.

Clef primaire : c'est l'élément identifiant de manière unique une ligne dans une table d'une base de données.

CGI : *Common Gateway Interface*, permet de faire dialoguer un navigateur WEB avec un programme exécuté côté serveur qui peut être écrit en n'importe quel langage.

FDDI : *Fiber Data Distribution Interface*, protocole de liaison (couche 2 du modèle OSI) relativement ancien, utilisant la fibre optique comme support. Sa topologie est en double anneau, les informations sur chaque anneau circulent dans des sens opposés. L'utilisation d'un anneau (primaire) avec le deuxième (secondaire) effectuant la correction d'erreurs permet un débit de 100 Mbit/s, si le deuxième est utilisé pour transporter des informations, le débit atteint 200 Mbit/s. La distance maximale d'un réseau basé sur FDDI est de 200km.

FIFO : *First In First Out*, c'est une structure de données où le premier élément inséré est le premier à sortir.

Format binaire : il s'agit du format de représentation élémentaire des données en informatique. Son alphabet est composé de 2 éléments : 0 et 1.

Full duplex : permet un échange simultané d'informations (émission et réception en même temps).

HTML : *HyperText Markup Language*, langage permettant d'écrire des pages dites hypertextes. Celles-ci permettent de relier des documents stockés sur différents serveurs par l'intermédiaire de liens.

IP : *Internet Protocol*, protocole de niveau réseau (couche 3 du modèle OSI), permettant l'adressage des machines et l'acheminement des données.

IDS : *Intrusion Detection System*, outil permettant de détecter des événements de sécurité. Il existe deux types d'IDS :
- les HIDS (H pour Host, c'est-à-dire une machine connectée au réseau) qui travaillent au niveau des machines. Ils détectent des intrusions sur la machine en observant le comportement des utilisateurs et des logiciels ;
- les NIDS (N pour Network, réseau en français) travaillent au niveau du réseau et détectent les signes d'intrusions dans le flux réseau.

Modèle OSI : *Open Systems Interconnection Reference Model*, modèle créé par l'ISO permettant de définir une base commune à tout réseau informatique. Ce modèle attribue suivant 7 couches des tâches spécifiques à chacune dans la transmission d'informations sur un réseau. Ces couches vont du lien physique jusqu'à l'application qui souhaite transmettre des informations.

NAT : *Network Address Translation*, permet d'utiliser une même adresse IP sur Internet pour un ensemble de machines.

Pare-feu : logiciel ou matériel qui filtre les menaces les plus simples du flux réseau.

Peer to Peer : protocole d'échange de données se faisant directement de poste client à poste client sans avoir besoin de serveurs. Le Peer to Peer est utilisé en grande majorité à des fins illégales.

PCI : *Peripheral Component Interconnect*, élément interne à un PC permettant de connecter des cartes d'extensions, comme par exemple une carte réseau...

PERL : *Practical Extraction and Report Language*, langage de programmation permettant d'écrire des scripts puissants.

PHP : *PHP Hypertext Preprocessor*, langage de programmation spécifiquement créé pour l'écriture de pages WEB dynamiques.

Qualité de service (QoS pour Quality of Service en anglais) : il s'agit en fait de la qualité du transport des informations dans le sens où les données arrivent dans l'ordre et avec un délai de transmission contrôlé. Dans le cadre d'un échange classique de données, il n'y a pas à s'en soucier. Par contre, par exemple pour la visioconférence, la qualité de service est nécessaire pour éviter d'avoir des mauvaises transmissions. Cette qualité de service est spécifiée contractuellement dans les offres des fournisseurs d'accès professionnel.

RAD : *Rapid Application Development*, méthode de développement rapide d'application, fondée sur des cycles courts et une forte interaction avec les utilisateurs

Scan de ports : c'est une action qui vise à détecter les failles potentielles dans la sécurité des machines connectées à ce réseau.

Socket UNIX : c'est un moyen de dialogue entre programmes sur une machine ayant un système d'exploitation UNIX.

SSO : *Single Sign On*, procédé d'authentification, celle-ci est faite une fois et permet d'aller sans avoir à ressaisir son mot de passe sur toutes les applications utilisant cette méthode.

Table de hachage : c'est un tableau dans lequel les informations ne sont pas indexées suivant des numéros mais par des clefs. Celles-ci peuvent être de n'importe quel type (mot, chiffre, etc.).

Topologie réseau : définition de l'architecture d'un réseau du point de vue physique.

Vers : un vers est un virus se propageant de manière autonome par le réseau. Contrairement à un virus classique qui se diffuse par courrier électronique, il ne nécessite pas d'action de la part de l'utilisateur.

WiFi : *Wireless Fidelity*, technologie de réseau sans fil en plein essor.

INSTITUT DES TECHNIQUES D'INGÉNIEUR DE L'INDUSTRIE D'AQUITAINE

MÉMOIRE DE FIN D'ÉTUDES

PRÉSENTÉ EN VUE D'OBTENIR

LE DIPLÔME D'INGÉNIEUR

DES TECHNIQUES DE L'INDUSTRIE

Dans la Spécialité Réseaux et Télécommunications

PAR

Grégoire MOREAU

DIPLÔME DÉLIVRÉ

PAR

L'ÉCOLE NATIONALE SUPÉRIEURE D'ÉLECTRONIQUE, INFORMATIQUE & RADIOCOMMUNICATION de BORDEAUX

ANNEXES

CONCEPTION ET MISE EN PRODUCTION D'UN SYSTÈME AUTOMATISÉ DE SÉCURISATION DU RÉSEAU REAUMUR

SOUTENU LE 19 SEPTEMBRE 2005

REAUMUR,

351 cours de la libération

33405 TALENCE

Table des annexes

Annexe A

Partenaires de REAUMUR

Partenaires pour le réseau de campus :
- Université Bordeaux I Sciences et Technologies,
- Université Victor Segalen Bordeaux II (facultés des sciences du sport, d'œnologie et des services interuniversitaires rattachés),
- Université Michel de Montaigne Bordeaux III,
- Université Montesquieu Bordeaux IV,
- CNRS (Délégation régionale),
- ENSEIRB,
- ENSCPB,
- IEP,
- ENSAM,
- Maison des Sciences de l'Homme d'Aquitaine.

Partenaires pour la plaque urbaine :
- Rectorat (Carayon Latour, ses annexes et les services académiques),
- Bordeaux Ecole de Management,
- Ecole d'architecture et de paysage de Bordeaux,
- ENITA,
- INRA Aquitaine (Grande Ferrade),
- IUFM (Caudéran, Mérignac),
- Université Bordeaux I (Observatoire de Bordeaux, rue Lamartine, IMA)
- Université Bordeaux II (Carreire, Victoire),
- Université Bordeaux III (Renaudel)
- Université Bordeaux IV (IAE, Bibliothèque Universitaire Pluridisciplinaire, IUT GLT)
- Pôle Universitaire de Bordeaux,
- CROUS (rue du Hamel),
- DRRT,
- Ecole Nationale de la Magistrature,
- INSERM (Haut-Lévêque unité 441 et IFR cardiologie).

Partenaires pour la plaque régionale :
- INRA Aquitaine (St Pée sur Nivelle),
- CEMAGREF,
- IUFM (Agen, Mont de Marsan, Pau, Périgueux),

- Université Bordeaux I (Agen Michel Serres, Arcachon Station Marine, Périgueux EPCA),
- Université Bordeaux II (Agen, Dax),
- Université Bordeaux III (Agen Centre du Pin),
- Université Bordeaux IV (Agen Centre du Pin, Périgueux IUT et IEJE),
- Université de Pau et des Pays de l'Adour,
- Rectorat (Pau),
- Conseil Régional d'Aquitaine (RAP),
- Turboméca,
- Institut du Développement Local à Agen.

Annexe B

Cahier des charges fonctionnel

Situations de vie

Voyons tout d'abord les différentes situations dans lesquelles va se trouver le système :

- installation,
- exploitation,
- maintenance,
- évolution.

Ces différentes situations représentent les différents cas qu'il faut prévoir dans le cadre de la vie du système.

Définition des fonctions du système

Définissons quelques termes :

- FP est une fonction principale, celle-ci définit la réponse à un besoin,
- FC est une fonction contrainte.

FP1 Le système permet à l'équipe REAUMUR d'automatiser partiellement la gestion de la sécurité.

FP2 Le système permet aux correspondants de gérer les levées de filtrages les concernant.

FP3 Le système permet à l'équipe REAUMUR de visualiser tous les événements et informations collectés de l'ensemble du réseau.

FP4 Le système permet aux correspondants de visualiser les événements et informations collectés de la partie du réseau les concernant.

FP5 Le système permet à l'équipe REAUMUR de choisir une ou plusieurs actions pour un événement non traité automatiquement.

FP6 Le système permet aux correspondants de personnaliser le comportement de celui-ci pour leur domaine.

FP7 Le système permet aux utilisateurs finaux de lever un filtrage concernant leur machine.

FC1 Le système doit respecter les règles de sécurité informatique.

FC2 Le système doit avoir une interface homme/machine intuitive et ergonomique.

FC3 Le système doit disposer d'une interface machine/machine (option).

FC4 Le système doit utiliser des standards libres et ouverts.

FC5 Le système doit disposer d'outils de mesure sur son utilisation.

FC6 Le système doit pouvoir s'interfacer avec les outils utilisés.

FC7 Le système doit signaler les événements.

FC8 Le système doit avoir son code source et sa documentation technique en anglais.

FC11 Le système doit être disponible sous forme de paquet Debian (option).

FC21 Le système doit pouvoir être maintenu par d'autres personnes que le développeur initial.

FC22 Le système doit être modulaire.

FC31 Le système doit pouvoir s'adapter aux évolutions du réseau.

FC32 Le système doit pouvoir disposer d'une interface en plusieurs langues (option).

Critères de performance par fonction

FP1 Le système permet à **l'équipe de REAUMUR d'automatiser partiellement** la gestion de la **sécurité**.

- **Équipe de REAUMUR** :
  L'équipe technique est composée de deux techniciens, trois ingénieur réseaux et un apprenti.

- **Automatiser partiellement** :
  Détecter les problèmes de sécurité et les signaler à l'équipe et aux correspondants concernés.
  Si possible, les traiter (mise en place de filtrage : isolation de la machine par rapport à Internet).
  Si pas de décision automatique possible, le signaler à l'équipe et présenter le problème de manière synthétique avec un maximum d'éléments permettant la prise de décision.
  Pouvoir ajouter de nouveaux problèmes avec le comportement que doit avoir le système par rapport à ceux-ci.

- **Problème de sécurité** :
  Piratages, utilisations abusives du réseau, propagation de virus.

- L'information sur les problèmes de sécurité est strictement conservée en interne et transmise uniquement aux correspondants qui sont concernés. La durée de conservation des informations est de 1 an.

- Données nécessaires à la gestion de la sécurité :
  Informations provenant de différentes sources : netMET (débits des flux réseaux du campus et informations sur leur source et destination), Snort (détection d'intrusions et d'abus), whitelist (liste des machines ayant un traitement de faveur : serveurs, etc.), journaux des équipements (serveur de courrier, routeurs Cisco).

- Données à extraire :
  Filtrage, informations sur le problème justifiant le filtrage, courrier électronique au correspondant, durée de filtrage. Dans le cas d'un problème non traitable automatiquement, demande d'assistance auprès de l'équipe REAUMUR.

- Sécurités à prévoir :
  Ne pas filtrer les machines vitales du campus. Vérifier si la machine à filtrer n'est pas déjà filtrée. Vérifier, si possible, que le problème n'est plus présent avant de lever un filtrage.

- Mode d'accès à la gestion de la sécurité :
  Navigation ou recherche suivant un ou plusieurs critères : heure et date, adresse ip, sous-réseau, type de problème (abus, virus, piratage, etc.).

- Temps de prise de décision du système :
  Une fois que toutes les informations sont récoltées, prise de décision en moins de 5 minutes. Fréquence : aléatoire.

- Temps d'accès à l'information :
  Moins de 5 secondes pour accéder à la page de synthèse et moins d'une minute pour obtenir le détail sur un événement.

FP2 Le système permet aux **correspondants** de **gérer** les **levées de filtrages** les concernant.

- **Correspondants** :
  Sur le campus, ce sont les interlocuteurs de REAUMUR au niveau des sous-réseaux. Dans certains cas, ils sont administrateurs système ou réseau, sinon ce sont des personnels de laboratoire dont le domaine d'activité n'a pas forcément grand chose à voir avec l'informatique. Leur niveau va donc du débutant à l'expert.

- **Gérer** :
  Permettre aux correspondants la levée des filtrages de sécurité automatiques concernant leur réseau.

- **Levée de filtrage** :

Celui-ci étant mis en place automatiquement lors d'un comportement suspect (virus, piratage, abus), une fois la machine nettoyée (logiciel désinstallé, virus éradiqué, etc.) ou l'utilisateur à l'origine d'un abus averti, le correspondant peut le signaler directement au système qui lève alors directement le filtrage qui était en place.

- Données nécessaires :
  Filtrages en cours, raison du filtrage, données d'authentification du correspondant, raisons de la levée et commentaire.

- Données à extraire :
  Levée du filtrage.

- Sécurité à prévoir :
  Pas d'abus de levée de filtrage (pas plus de 3 levées de filtrages pour une même machine dans un intervalle de 30 minutes) sinon interdiction de levée temporaire. Authentification sûre.

- Mode d'accès à la levée de filtrage :
  Navigation ou recherche sur un ou plusieurs critères : date et heure, adresse ip, raison de filtrage.

- Temps d'accès :
  Inférieur à 5 secondes pour voir tous les filtrages sur le réseau du correspondant, levée du filtrage en moins de 5 minutes.

FP3 Le système permet à l'équipe REAUMUR de visualiser tous les **événements** et **informations** collectés de l'ensemble du réseau.

- **Événements** :
  Les problèmes de sécurité détectés, les filtrages effectués automatiquement en attente de levée par un correspondant, les traitements à faire manuellement et les levées de filtrage effectuées par les correspondants.

- **Informations** :
  Les statistiques et les archives d'événements. Conservation : 1 an

- Données nécessaires :
  Événements, informations.

- Données à extraire :
  Synthèse, mise en forme.

- Sécurité à prévoir :
  Authentification sûre.

- Mode d'accès aux informations :

Navigation ou recherche suivant un ou plusieurs critères : type d'événement, date et heure, adresse ip, réseau.

– Temps d'accès :
Inférieur à 5 secondes pour la page d'accueil (synthèse), inférieur à 5 minutes pour des informations précises sur un événement.

FP4 Le système permet aux correspondants de visualiser les événements et informations collectés de la partie du réseau les concernant.

Reprend FP3 mais est destiné aux correspondants donc les événements et informations ne sont que celles de leur réseau respectif.

Au niveau de la sécurité : pas de visibilité sur le réseau des autres correspondants. Authentification sûre.

FP5 Le système permet à l'équipe REAUMUR de **choisir une ou plusieurs actions** pour un problème non traité automatiquement.

– **Choisir une ou plusieurs actions** :
Déterminer le comportement du système : bloquer, demander un traitement manuel, avertir l'équipe REAUMUR, avertir le correspondant.

– Données nécessaires :
Problème non traité automatiquement, choix de comportement de la part de l'utilisateur.

– Données à extraire :
Traitement automatique choisi.

– Sécurité à prévoir :
Authentification sûre.

– Mode d'accès aux informations :
Navigation.

– Temps d'accès :
Inférieur à 5 secondes.

FP6 Le système permet aux correspondants de **personnaliser le comportement** de celui-ci pour leur domaine.

– **Personnaliser le comportement** :
Le correspondant peut choisir quelles actions associer à un problème donné pour leur

domaine.

- Données nécessaires :
  Comportement pour chaque problème, choix de comportement de la part de l'utilisateur.

- Données à extraire :
  Comportement choisi.

- Sécurité à prévoir :
  Authentification sûre. Ne pas pouvoir choisir un traitement plus faible que le traitement défini par REAUMUR (par exemple, un problème qui entraîne le blocage d'une machine ne peut pas avoir une action moins forte que le blocage). Cependant, il peut y avoir des exceptions mais cela nécessite un contact direct avec l'équipe REAUMUR qui apportera la modification.

- Mode d'accès aux informations :
  Navigation.

- Temps d'accès :
  Inférieur à 5 secondes.

FP7	Le système permet aux **utilisateurs finaux** de lever un filtrage concernant leur machine.

- **Utilisateurs finaux** :
  Les utilisateurs du réseau sur le campus, quel que soit leur organisme de rattachement.

- Données nécessaires :
  Filtrages en cours, raison du filtrage, raisons de la levée et commentaire.

- Données à extraire :
  Levée du filtrage.

- Sécurité à prévoir :
  Pas d'abus de levée de filtrage : 1 levée possible par jour uniquement. Visibilité uniquement des informations concernant la machine de l'utilisateur.

- Temps d'accès :
  Levée du filtrage en moins de 5 minutes.

Fonctions contraintes en exploitation

FC1 Le système doit respecter les règles de sécurité informatique.

- Confidentialité du système :
  Les informations et manipulations offertes par le système ne doivent être accessibles qu'aux personnes autorisées : équipe REAUMUR et correspondants. Pour chaque utilisateur, l'accès est restreint à son domaine.
- Autorisation d'accès au système :
  Une authentification sûre est mise en place de manière à distinguer sans erreur ,ni usurpation possible, l'utilisateur qui utilise le système. Cette authentification sera faite par certificats, voire SSO[1].
- Protection du système :
  Le système doit pouvoir résister à des dénis de services : ne pas s'écrouler sous une charge anormale et résister aux maximum à des piratages éventuels aussi bien externes qu'internes au réseau campus.

FC2 Le système doit avoir une interface homme/machine intuitive et ergonomique.

- Temps de formation : 1/2 journée.
- L'interface doit pouvoir être utilisée par des non informaticiens (rappelons que certains correspondants ne sont pas informaticiens) mais cependant rester complète.
- L'interface doit respecter les critères d'ergonomie.

FC3 Le système doit disposer d'une interface machine/machine (option).

- Communications entre machines pour intégration plus facile dans des systèmes tiers.
- Communication dépendant de l'utilisateur qui le demande : accès restreint de la même manière que pour l'interface homme/machine.

FC4 Le système doit utiliser des standards libres et ouverts.

- Les standards utilisés devront être ouverts de manière à ne pas dépendre d'un produit propriétaire.

FC5 Le système doit disposer d'outils de mesure sur son utilisation.

[1]SSO : Single Sign On. L'authentification est faite une fois et permet d'aller sans avoir à ressaisir son mot de passe sur toutes les applications utilisant cette méthode d'authentification.

– Des journaux d'utilisations (logs) seront mis en place et des statistiques faites à partir de ceux-ci. La présentation de ces statistiques sera faite sous forme de graphiques.

FC6 Le système doit pouvoir s'interfacer avec les outils utilisés.

– Snort : version 2.
– NetMET : toutes versions .
– Journaux d'équipements Cisco : toutes versions.
– Journaux système Linux.
– Possibilité d'autres outils pas encore connus mais qui seront libres et ouverts : on retrouve donc FC4.

FC7 Le système doit signaler les événements.

– Signal sonore audible par toute l'équipe.
– Signal visuel au niveau de l'interface graphique.

FC8 Le système doit avoir son code source et sa documentation technique en anglais.

– De manière à pouvoir être lu par tout programmeur et pas uniquement en France.

Fonctions contraintes en installation

FC11 Le système doit être disponible sous forme de paquet Debian (option).

– Debian est la seule distribution GNU/Linux utilisée à REAUMUR.
– Le système mis sous forme de paquet Debian (format de paquet libre et ouvert), permettra une installation simple et rapide.

Fonctions contraintes en maintenance

FC21 Le système doit pouvoir être maintenu par d'autres personnes que le développeur initial.

– Rédaction de documentations techniques complètes (en français et en anglais).

– Les autres personnes peuvent être les ingénieurs de l'équipe REAUMUR mais aussi des développeurs où qu'ils soient dans le monde (dans ce cas, évidement, il y a vérification des contributions externes). Ceci justifie le fait d'avoir une documentation et un code source en anglais.

FC22 Le système doit être modulaire.

– La conception et le développement du système doivent être faits de manière à pouvoir ajouter facilement des modules et permettre une maintenance aisée des modules existants.

Fonctions contraintes en évolution

FC31 Le système doit pouvoir s'adapter aux évolutions du réseau.

– Les évolutions de débits du réseau doivent pouvoir être prises en charge avec pour maximum 1000Mbit/s. La réponse en terme de performance du système doit pouvoir être mesurée.
– Les évolutions de protocole du réseau : passage de IPV4 à IPV6.

FC32 Le système doit pouvoir disposer d'une interface en plusieurs langues (option).

– En vue de proposer le système à la communauté des logiciels libres, il peut être intéressant de prévoir une interface qui puisse être traduite en plusieurs langues. Dans un premier temps ce serait l'anglais.

Annexe C

Modèle conceptuel des données

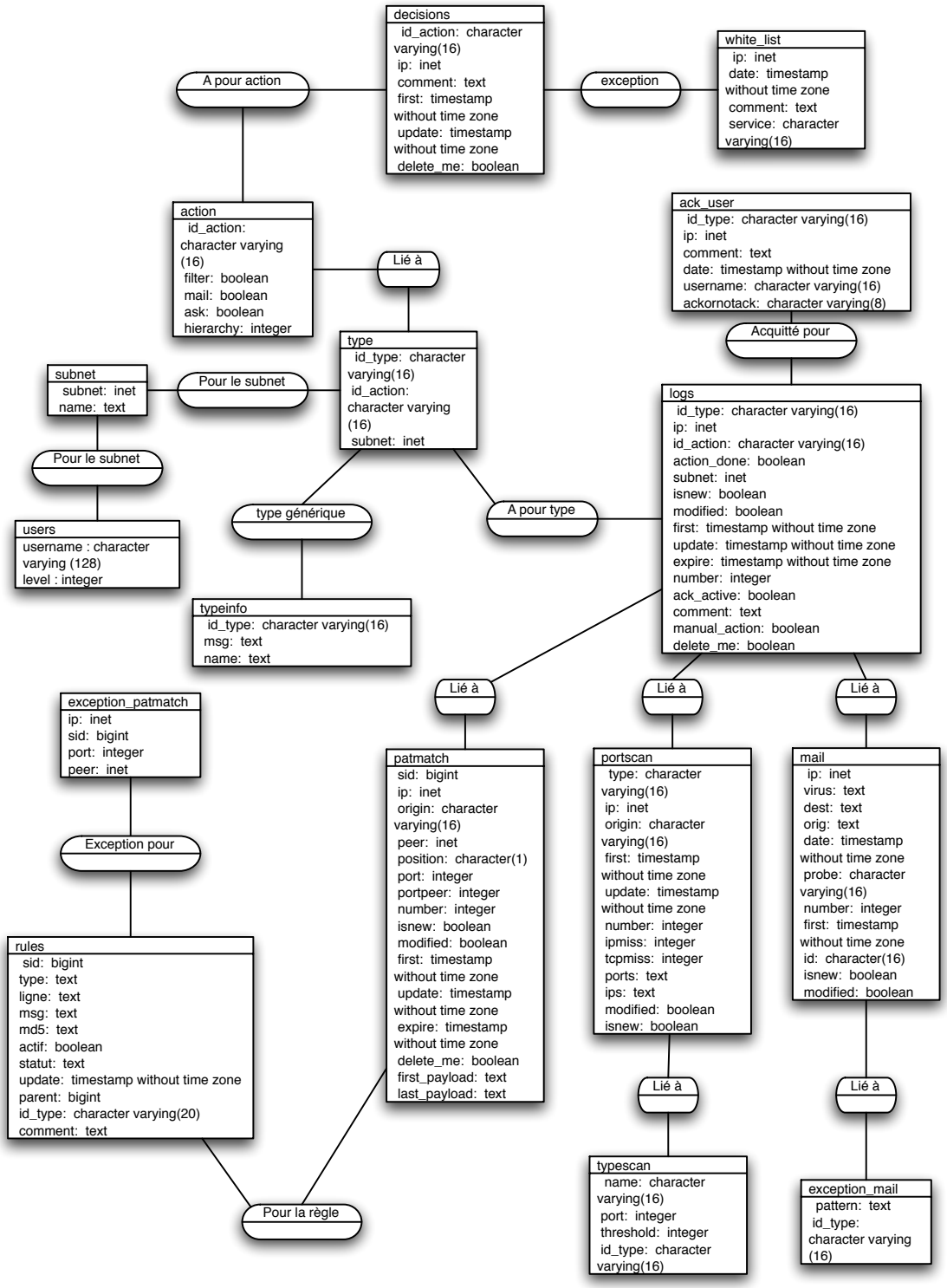

FIG. C.1 – Modèle conceptuel des données