ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE DES TECHNOLOGIES DE L'INFORMATION
M. Ing.

PAR
Maxime DUMAS

ALERTWHEEL : VISUALISATION RADIALE DE GRAPHES BIPARTIS APPLIQUÉE AUX SYSTÈMES DE DÉTECTION D'INTRUSIONS SUR DES RÉSEAUX INFORMATIQUES

MONTRÉAL, LE 16 DÉCEMBRE 2011

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Jean-Marc Robert, directeur de mémoire
Génie logiciel et TI à l'École de technologie supérieure

M. Michael J. McGuffin, codirecteur de mémoire
Génie logiciel et TI à l'École de technologie supérieure

M. Éric Paquette, président du jury
Génie logiciel et TI à l'École de technologie supérieure

M. Chamseddine Talhi, membre du jury
Génie logiciel et TI à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 25 NOVEMBRE 2011

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

J’aimerais tout d’abord remercier mon directeur de maîtrise, Jean-Marc Robert, ainsi que mon codirecteur, Michael J. McGuffin, pour leur appui tout au long de ce projet. Je me considère extrêmement privilégié d’avoir rencontré ces deux personnes extraordinaires et d’avoir pu collaborer aussi étroitement avec eux pendant ces deux années. Merci pour tout!

Je tiens également à remercier Marie-Claire Willig, ma partenaire dans le cadre de ce projet, avec qui j’ai eu beaucoup de plaisir à travailler. Je remercie également les membres du groupe HIFIV, et tout particulièrement Christophe Viau, pour leurs idées et pour leurs conseils.

J’aimerais également remercier mes parents et ma conjointe Véronique pour leur soutien, leurs conseils et leur écoute tout au long de mon cheminement académique et professionnel.

Finalement, j’aimerais remercier le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) ainsi que l’École de Technologie Supérieure pour le financement de ce projet.

ALERTWHEEL : VISUALISATION RADIALE DE GRAPHES BIPARTIS APPLIQUÉE AUX SYSTÈMES DE DÉTECTION D'INTRUSIONS SUR DES RÉSEAUX INFORMATIQUES

Maxime DUMAS

RÉSUMÉ

Les systèmes de détection d'intrusions (IDS) sont couramment employés pour détecter des attaques sur des réseaux informatiques. Ces appareils analysent le trafic entrant et sortant à la recherche d'anomalies ou d'activités suspectes. Malheureusement, ces appareils génèrent une quantité importante de bruit (ex. : faux positifs, alertes redondantes, etc.), complexifiant grandement l'analyse des données.

Ce mémoire présente AlertWheel, une nouvelle application logicielle visant à faciliter l'analyse des alertes sur des grands réseaux. L'application intègre une visualisation radiale affichant simultanément plusieurs milliers d'alertes et permettant de percevoir rapidement les patrons d'attaques importants. AlertWheel propose, entre autres, une nouvelle façon de représenter un graphe biparti. Les liens sont conçus et positionnés de façon à réduire l'occlusion sur le graphique. Contrairement aux travaux antérieurs, AlertWheel combine l'utilisation simultanée de trois techniques de regroupement des liens afin d'améliorer la lisibilité sur la représentation. L'application intègre également des fonctionnalités de filtrage, d'annotation, de journalisation et de « détails sur demande », de façon à supporter les processus d'analyse des spécialistes en sécurité informatique.

L'application se décompose essentiellement en trois niveaux : vue globale (roue), vue intermédiaire (matrice d'alertes) et vue détails (une seule alerte). L'application supporte plusieurs combinaisons et dispositions de vues, de façon à s'adapter facilement à la plupart des types d'analyse.

AlertWheel a été développé principalement dans le but d'étudier le trafic sur des pots de miel (*honeypot*s). Dans la mesure où tout le trafic sur un *honeypot* est nécessairement malveillant, ces derniers permettent d'isoler plus facilement les attaques.

AlertWheel a été évalué à partir de données provenant du réseau international de *honeypots* WOMBAT. Grâce à l'application, il a été possible d'isoler rapidement des attaques concrètes et de cibler des patrons d'attaques globaux.

Mots clés : visualisation, IDS, honeypot, intrusion, sécurité

ALERTWHEEL : VISUALISATION RADIALE DE GRAPHES BIPARTIS APPLIQUÉE AUX SYSTÈMES DE DÉTECTION D'INTRUSIONS SUR DES RÉSEAUX INFORMATIQUES

Maxime DUMAS

ABSTRACT

Intrusion detection systems (IDS) are widely used to detect attacks on computer networks. These tools scan incoming and outgoing traffic, searching for anomalies or suspicious activities. Unfortunately, they also generate much noise (i.e. false positives, redundant alerts, etc.), greatly complicating data analysis.

This thesis presents AlertWheel, a new software application easing network analysis on large-scale networks. It is based on a novel radial visualization capable of simultaneously displaying several thousand alerts, emphasizing the most important alerts or patterns in the dataset. Among other things, AlertWheel offers a new technique for representing bipartite graphs (where links exist between two distinct node groups). Using this approach, links are positioned in a way to reduce occlusion in the visualization. AlertWheel simultaneously combines three link bundling techniques in a novel way to reduce cluttering on the interface. Our solution also incorporates filtering options, annotation, logging and details-on-demand, to support analysis processes as described by specialists in this field.

AlertWheel enables three different levels of analysis: high level analysis (the alert wheel), intermediate analysis (alert matrix) and a detailed analysis (single alert). Our prototype supports different combinations and layouts of views, to adapt to many kinds of analysis.

The application was mainly developed to support honeypot analysis (virtually vulnerable computers used as a trap to analyze malicious traffic). AlertWheel could also be used on large computer networks where traditional techniques could not be adapted.

AlertWheel was evaluated using network traffic captured on the international honeypot network WOMBAT. Using our solution, it was possible to rapidly isolate actual attacks and identify high level attack patterns.

Keywords: visualization, IDS, honeypot, intrusion, security

TABLE DES MATIÈRES

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

APIDS	Application-based Intrusion Detection System (IDS applicatif)
AS	Autonomous System (Système autonome)
DOD	U.S. Department of Defense (défense nationale américaine)
HIDS	Host Intrusion Detection System (IDS hôte)
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System (système de détection des intrusions)
IP	Internet Protocol
IPS	Intrusion Prevention System (système de prévention des intrusions)
NIDS	Network Intrusion Detection System (IDS réseau)
TI	Technologies de l'Information
TNV	The Network Visualizer ou Time-based Network Visualizer
WOMBAT	Worldwide Observatory of Malicious Behaviors and Attack Threats

INTRODUCTION

> "It is easy to run a secure computer system. You merely have to disconnect all dial-up connections and permit only direct-wired terminals, put the machine and its terminal in a shielded room, and post a guard at the door" (F.T. Gramp et R.H. Morris).

À une certaine époque, sécuriser un ordinateur aurait pu se résumer ainsi. Malheureusement, de nos jours, la réalité est plus complexe. Sites web, serveurs DNS, serveurs de courriels sont quelques exemples de services nécessitant la connexion à un réseau informatique (pour être utiles, du moins!). À partir du moment où un système doit interagir avec l'extérieur, il est beaucoup plus difficile de conserver le contrôle sur la sécurité du système et des données. Chaque système relié au réseau peut devenir la source d'une activité malicieuse. Il existe plusieurs solutions classiques pour réduire les risques liés à la connexion à un réseau. Par exemple, il est possible d'installer un pare-feu afin de contrôler les services accessibles à l'extérieur de la zone contrôlée (ex : Internet), ou encore d'utiliser des techniques cryptographiques pour s'assurer que les données ne seront pas lues ou modifiées durant le transport. Malheureusement, ces techniques ne permettent pas de régler tous les problèmes.

S'il est impossible de prévenir complètement les attaques, il est primordial de pouvoir les détecter le plus rapidement possible et de prendre les mesures appropriées pour en limiter les conséquences. Une technique consiste à déployer des systèmes de détection d'intrusions (IDS), de façon à détecter tout comportement anormal sur le trafic réseau de l'organisation. Ces appareils informent rapidement les responsables du réseau dès qu'une anomalie est détectée. Ces derniers sont alors en mesure de réagir rapidement en fonction de la gravité de la situation.

Malheureusement, les IDS apportent également leur lot d'inconvénients. Par exemple, un IDS aura beaucoup de difficulté à identifier une attaque répartie sur une longue période de temps. Certains types d'IDS peuvent également être déjoués en utilisant une attaque déviant légèrement de la signature connue de l'IDS. De plus, il n'est pas rare de se retrouver avec

plusieurs milliers d'alertes chaque jour, alors que la majorité ne présente aucun problème de sécurité. Pire encore, chaque trace laissée par un attaquant sera enregistrée individuellement et sera perdue à travers cette mer de fausses alarmes, rendant difficile la détection des véritables attaques.

Heureusement, il est possible de limiter l'ampleur de la plupart des problèmes mentionnés ci-dessous à l'aide de certaines techniques de visualisation. Ces dernières consistent à représenter de façon graphique des ensembles d'informations complexes afin d'en faire ressortir les éléments importants. Plusieurs groupes de chercheurs ont présenté, au cours des dernières années, des approches innovatrices permettant de visualiser les journaux des systèmes de détection d'intrusions. Quelques-unes de ces approches seront d'ailleurs présentées dans le chapitre suivant. Malheureusement, on relève habituellement pour ces approches deux problèmes importants. Tout d'abord, la plupart d'entre elles fonctionnent efficacement uniquement si le nombre d'alertes ou de systèmes à surveiller est limité. En effet, lorsque le nombre d'alertes ou de systèmes augmente significativement, les visualisations deviennent rapidement illisibles ou trop chargées pour transmettre efficacement l'information désirée. À l'inverse, lorsque les visualisations gèrent efficacement un grand nombre d'alertes, le nombre d'informations représentées sur les visualisations devient extrêmement limité (ex. : un seul pixel par adresse / alerte). Aucune solution proposée jusqu'à présent ne permet de visualiser efficacement les alertes provenant d'un très grand réseau avec un nombre suffisant d'informations pour faire ressortir efficacement les attaques importantes. Plusieurs analystes affirment également que les solutions proposées jusqu'à présent sont déconnectées de la réalité, et qu'ils ne répondent pas convenablement à leurs besoins, à leur façon de procéder.

La solution proposée dans le cadre de ce projet, AlertWheel, permet de visualiser efficacement un réseau de très grande envergure, tout en offrant une quantité suffisante d'informations pour analyser et comprendre des attaques très pointues. AlertWheel a été développé principalement dans le but d'analyser le trafic capturé sur un *honeynet*, un réseau international de systèmes volontairement vulnérables visant à piéger des pirates

informatiques. AlertWheel doit donc visualiser des attaques provenant de partout dans le monde. La solution proposée est également basée entièrement sur le processus d'analyse tel que défini par les spécialistes dans le domaine. Contrairement à plusieurs approches concurrentes, AlertWheel répond donc aux besoins réels des analystes dans le domaine, et met l'emphase sur leur façon de travailler.

D'un point de vue visualisation, la solution propose plusieurs caractéristiques innovatrices. AlertWheel propose, entre autres, une nouvelle façon de représenter un graphe biparti (graphe où des liens sont tracés entre deux ensembles de nœuds distincts) de façon à limiter l'occlusion et la confusion sur la visualisation. L'application propose également trois techniques de regroupement des données utilisées simultanément afin de limiter le nombre de liens à afficher. L'application repose sur une combinaison de trois vues : une visualisation radiale (vue globale), une matrice d'alertes (vue intermédiaire) et une vue détaillée où chaque alerte peut être étudiée en détail individuellement. L'application offre également plusieurs autres caractéristiques essentielles pour respecter le processus d'analyse des spécialistes dans le domaine : techniques de filtrage, journalisation, annotation, scénarios d'attaques, etc.

Le premier chapitre effectuera un survol de la littérature dans le domaine, présentant les principaux concepts et solutions existantes. Une présentation plus approfondie de la problématique sera effectuée au chapitre 2, et la solution retenue sera présentée en détail au chapitre 3. Le chapitre 4 présentera enfin une étude de cas basée sur des données réelles provenant du *honeynet* international WOMBAT, où il sera possible d'évaluer les capacités de la solution proposée pour reconnaître et analyser des attaques concrètes. Le chapitre 5 discutera enfin de l'apport de la solution face à la problématique initiale.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

1.1 La détection des intrusions

Selon Kruegel, Valeur et Vigna (2005), il existe trois grandes classes de mécanismes de sécurité : la prévention, le « contournement » (*avoidance*) et la détection des attaques. La première classe consiste à mettre en place des mesures permettant de se défendre contre certaines attaques avant même qu'elles ne se produisent. On retrouve entre autres dans cette catégorie les listes de contrôle d'accès et les pare-feu. Le contournement des attaques tient ensuite pour acquis que les mesures de prévention peuvent être déjouées, et tente de transformer le contenu de façon à le rendre inutilisable par un utilisateur malveillant. On retrouve dans cette catégorie les mécanismes de cryptographie, de certification, etc. Malheureusement, il arrive parfois que les mécanismes de prévention et de contournement ne soient pas suffisants, et qu'un attaquant parvienne malgré tout à altérer ou à accéder au contenu protégé. La dernière classe présente donc des mécanismes de derniers recours pour détecter des attaques ou des tentatives d'intrusions. Les systèmes de détection d'intrusions (IDS) figurent habituellement parmi les meilleures solutions pour effectuer ce travail.

Le Code criminel américain définit une intrusion informatique comme un accès intentionnel à un ordinateur sans autorisation ou en abusant des accès autorisés (United States (2010)). De façon plus générale, une intrusion correspond à la violation d'une politique de sécurité de l'organisme (Bejtlich (2004)). Selon Cole (2009), le rôle d'un système de détection d'intrusions consiste à capturer la présence d'un utilisateur malveillant sur un réseau compromis, à éliminer toute malversation causée par la présence de l'intrus et à cataloguer les activités de façon à pouvoir éviter des attaques similaires à l'avenir.

Selon Kruegel, Valeur et Vigna (2005), les systèmes de détection d'intrusions peuvent être classifiés selon plusieurs caractéristiques, dont les plus importantes sont présentées ci-après.

Méthode de détection : Les IDS sont habituellement répartis selon deux types de stratégies de détection : par base de connaissances (*misuse-based*) ou par anomalie (*anomaly-based*). La première stratégie consiste à conserver un dictionnaire des modèles d'attaques (signatures) et à déterminer si ces éléments se retrouvent dans le trafic réseau analysé. Cette technique possède l'avantage d'être très performante puisqu'elle génère moins de faux positifs (fausses alertes générées par un IDS) et qu'elle nécessite moins de ressources pour l'analyse (il s'agit essentiellement d'une simple comparaison). Dans la mesure où les informations brutes sont comparées à des signatures d'attaques connues préalablement, seules les données correspondant exactement à la signature seront retenues. En contrepartie, le fait de nécessiter une signature pour reconnaître une attaque réduit les capacités du système à détecter les nouvelles attaques ou les attaques polymorphes (attaques changeant leur signature à chaque fois). Le nombre de faux négatifs (attaques non détectées par le système) est également généralement plus important que par anomalie, puisqu'il est nécessaire de connaître les signatures de chaque attaque pour qu'elle puisse être détectée. De plus, cette approche implique des ressources humaines dédiées à la création et à l'ajustement des signatures afin d'être constamment à jour face aux nouvelles attaques. À l'inverse, les approches par anomalie recherchent plutôt des déviations de la « normalité » sur le réseau. Un IDS par anomalies apprendra, à l'aide de techniques d'apprentissage-machine, ce qui est normal sur le réseau. Tout ce qui dévie suffisamment de cette normalité sera considéré comme suspect et sera rapporté aux responsables. Cette stratégie s'avère beaucoup plus flexible que l'approche par signatures pour détecter des attaques *zero-day*. Une attaque « jour 0 » exploite des vulnérabilités inconnues jusqu'à présent. Il n'existe donc habituellement pas de signature pour détecter ces attaques qui n'ont jamais été répertoriées. Puisque les IDS par anomalies n'ont pas à connaître les caractéristiques des attaques pour les détecter, ces derniers peuvent détecter des attaques *zero-day* beaucoup plus facilement. Malheureusement, les IDS par anomalies s'avèrent habituellement moins précis et génèrent souvent beaucoup plus de faux positifs.

État : Les IDS se distinguent également par leur gestion d'état (*stateless* / *stateful*). Un IDS « sans état » traite chaque événement indépendamment, et toutes les informations concernant cet événement seront détruites une fois le traitement terminé. À l'inverse, un IDS «avec état»

conserve des informations sur les événements passés. Cette caractéristique permet, par exemple, de détecter plus facilement des attaques composées de plusieurs étapes, où plusieurs signatures doivent être présentes dans un ordre précis pour qu'une attaque soit signalée. Dans la mesure où il est plus difficile de reproduire de fausses attaques impliquant plusieurs étapes, les IDS « avec état » peuvent généralement résister plus facilement aux attaques de type *alert storm*, où un utilisateur surcharge un IDS de trafic générant un grand nombre de fausses alertes afin de dissimuler des actions malveillantes dans cette mer d'alertes. Par contre, un IDS sans état sera beaucoup plus performant au niveau de la rapidité d'exécution, n'ayant pas à effectuer les analyses complexes rattachées à l'état et à la corrélation des alertes. Certains IDS utilisent également l'approche « par transition d'état », qui consiste à modéliser les différents états du système à chaque phase de l'attaque. Cette approche présente plusieurs avantages, dont celui de permettre une réponse avant l'étape finale de l'attaque (en détectant une série d'états intermédiaires). Cette approche est toutefois plus difficile à maintenir en cas d'attaques multiples simultanées.

Type de réponse : La plupart des IDS sont passifs (ne font que rapporter les incidents). Toutefois, certains peuvent être actifs et réagir automatiquement face à des intrusions (ex : tuer des processus, etc.). Ces derniers peuvent malheureusement causer encore plus de mal dans certains cas (ex : causer des dénis de service, etc.). Lorsqu'un IDS est actif, il prend généralement le nom d'IPS (*Intrusion Prevention System*).

Fréquence d'utilisation : On retrouve deux types de fréquence : en temps réel et en différé. L'analyse en temps réel ou dynamique possède évidemment l'avantage de détecter plus rapidement les attaques et permet de réagir avant qu'il ne soit trop tard. À l'inverse, l'analyse différée ou statique réduit les ressources nécessaires (ex. : bande passante) en effectuant des analyses au besoin ou à intervalle régulier sur des extraits des données. Cette approche peut permettre, par exemple, des analyses plus approfondies sur les données, sans réduire les performances des installations.

Coopération et corrélation : Certains IDS de nouvelle génération permettent des analyses collaboratives en échangeant, par exemple, des résultats d'analyse. Deux IDS utilisant des techniques d'analyse différentes peuvent compléter leur zone de couverture et réduire le nombre de faux positifs. On entend par corrélation l'ajout d'informations provenant de sources externes permettant de compléter l'analyse. Il est possible de corréler les résultats d'analyse de plusieurs IDS situés à différents endroits sur le réseau ou encore de considérer les alertes des autres systèmes sur le réseau.

Source des données d'audit : Il existe quatre grandes familles d'IDS à ce niveau : *host-based* (HIDS), *application-based* (APIDS), *network-based* (NIDS), et les systèmes de corrélation. Les systèmes basés sur des hôtes détectent les attaques perpétrées contre un système informatique hôte en particulier. Ces IDS utilisent, entre autres, les journaux du système et les informations du système (ex. : processus actifs) pour détecter les intrusions. Les IDS basés sur des applications sont particulièrement adaptés pour protéger des processus précis (ex. : serveur web). Ils peuvent être intégrés directement dans le code source de l'application, ou encore être intégrés via des interfaces publiques ou des extensions (*hooks*). De leur côté, les IDS réseau analysent habituellement le trafic échangé entre les nœuds d'un réseau. Ce type d'IDS est particulièrement répandu, principalement grâce à sa simplicité de déploiement (simplement insérer entre deux nœuds du réseau). Ils apportent malheureusement leur lot de difficultés : capacité limitée de traitement réduisant la rapidité du réseau, incapacité à analyser les données cryptées, etc. Enfin, les systèmes de corrélation combinent habituellement les informations de plusieurs solutions (ex. : un HIDS et un NIDS) pour obtenir une vue plus générale de la situation, laquelle serait impossible à récupérer à l'aide d'un seul outil.

Malheureusement, l'utilisation d'un IDS apporte également quelques désavantages. La plupart des analystes déplorent tout d'abord la quantité de faux positifs qu'un IDS peut générer. Plusieurs rapportent devoir gérer plusieurs milliers d'alertes par jour, alors que moins d'une dizaine d'événements intéressants doivent être rapportés. La plupart des IDS deviennent également impuissants lorsque la communication est cryptée. Enfin, selon

Axelsson et Sands (2005), ces appareils peuvent facilement être déjoués. Les auteurs présentent en effet plusieurs techniques simples permettant de contourner les engins de détection ou de causer des dénis de service. D'autres inconvénients s'ajoutent également à cette liste, selon le type d'IDS utilisé. Malgré tout, ces systèmes demeurent des outils de premier choix pour assister les analystes dans la détection des intrusions. Parcourir quelques milliers d'alertes s'avère malgré tout plus efficace qu'explorer l'ensemble des journaux de tous les équipements de l'organisation...

1.2 Les pots de miel (*honeypots*)

Habituellement, l'objectif principal en sécurité consiste à protéger toutes les installations contre de possibles attaques informatiques. Il existe toutefois un cas particulier où les outils visent exactement le cas inverse. Selon Lance Spitzner, fondateur du projet Honeynet, un *honeypot* est « un système d'information dont la valeur réside dans l'utilisation non autorisée ou illicite de cette ressource » (Spitzner (2003)). Si personne n'attaque cette ressource, elle ne sert strictement à rien. Pour être utile, un *honeypot* doit être sondé, attaqué et compromis (Feng et al. (2003)).

Par définition, le *honeypot* est une ressource sans aucune activité autorisée. Puisqu'elle n'est rattachée à aucune activité légitime, cette ressource ne devrait en théorie observer aucun trafic. Il est donc possible d'affirmer que toute interaction avec un *honeypot* est nécessairement non autorisée ou malicieuse. Selon Spitzner (2003), un *honeypot* a donc les avantages suivants :

- **Petite quantité de données** : Les *honeypots* récoltent uniquement une petite quantité d'information. Le niveau de bruit est beaucoup plus faible qu'avec un système normal où les activités malicieuses se mélangent au trafic légitime. Par conséquent, il est beaucoup plus facile d'analyser les données provenant un *honeypot* pour découvrir des attaques.
- **Nouveaux outils et tactiques** : Les *honeypots* sont conçus pour capturer tout ce qui leur est transmis, incluant des outils et des tactiques jamais vus jusqu'à présent.

- **Ressources minimales** : Un *honeypot* demande très peu de ressources. Un vieux système fait habituellement amplement le travail.
- **Supporte le cryptage et IPv6 :** Contrairement aux IDS réseau, par exemple, un *honeypot* gère le cryptage et les environnements IPv6.
- **Information :** Permet d'obtenir des informations de bas niveau qu'il serait habituellement difficile de collecter à cause de la quantité de bruit (trop grande quantité d'information).
- **Simplicité :** L'architecture d'un *honeypot* est très simple. Il n'y a pas de signature à maintenir, de tables de références, etc.

Malheureusement, les *honeypots* ont également quelques inconvénients. Par exemple :

- **Vue limitée :** Les *honeypots* ne peuvent capturer que les activités les impliquant directement. Ils ne pourront donc pas capturer des attaques destinées à d'autres systèmes du réseau.
- **Risque** : Les *honeypots* peuvent également être compromis par les utilisateurs malveillants et utilisés pour atteindre d'autres installations du réseau. Le niveau de risque dépend essentiellement du type de *honeypot* utilisé. Isoler le *honeypot* du reste du réseau peut généralement limiter ce type d'ennui.
- **Détection** : Un *honeypot* peut ne pas répondre de la même façon qu'un véritable système en production. La plupart des *honeypots* simuleront des services vulnérables pour tenter de leurrer les attaquants. Toutefois, ces derniers pourraient détecter que la réponse n'est pas « normale », et ainsi déterminer qu'ils font affaire avec un *honeypot*.

Spitzner divise essentiellement les *honeypots* en deux catégories : bas niveau d'interaction et haut niveau d'interaction. L'interaction définit le niveau d'activité qu'un *honeypot* autorise à l'attaquant. Par exemple, un système à bas niveau peut simuler uniquement l'authentification sur un service FTP, alors qu'un système à haut niveau supportera l'ensemble des commandes, simulera des vulnérabilités, etc.

Les *honeypots* peuvent être utilisés de deux façons : en production ou en recherche. En production, les *honeypots* visent à protéger une organisation. Les analystes en sécurité utiliseront les informations recueillies par les *honeypots* pour comprendre des attaques détectées ailleurs sur le réseau, ou encore pour prévenir des intrusions avant qu'elles ne surviennent sur les systèmes en production. Dans le cadre de la recherche, les *honeypots* servent habituellement à recueillir des informations sur les attaques courantes durant une période donnée. Ces informations permettent d'étudier les activités globales sur Internet, d'obtenir des informations rapidement sur les tendances à venir, effectuer des prédictions, etc.

Dans le cadre de ce projet de recherche, les *honeypots* seront utilisés en collaboration avec des IDS pour étudier les patrons d'attaques et leur provenance. Nous utiliserons l'IDS Snort (Sourcefire Inc (2011)) pour analyser le trafic reçu par les *honeypots* du réseau WOMBAT installés un peu partout sur la planète. Toutefois, obtenir les données n'est pas une finalité en soi. Encore faut-il être en mesure de les analyser. Plusieurs articles présentés au cours des dernières années décrivent les grandes lignes des principaux processus d'analyse des données tels que définis par les analystes dans le domaine. La section suivante permettra de survoler ces processus, de façon à mieux comprendre les façons de faire et les principaux besoins dans le domaine.

1.3 Le processus d'analyse en sécurité informatique

Mitch Kabay affirmait en 1998 que la « sécurité est un processus, et non pas un état final » (Kabay (1998)). Dans le même ordre d'idée, selon Bejtlich (2004), la sécurité est un processus permettant de maintenir un niveau acceptable du risque perçu. La sécurité ne se résume donc pas à installer des appareils de sécurité sur un réseau. Plusieurs études réalisées au cours des dernières années auprès d'analystes en sécurité dans différents milieux ont démontré qu'il existe en effet un processus continu de sécurité, peu importe le champ d'activité de l'organisme. Ce processus peut être plus ou moins formel, selon le milieu et les

données à protéger. Les principales contributions de ces études seront présentées dans le cadre de cette section.

Selon Bejtlich (2004), le processus de sécurité est habituellement représenté par un cycle divisé en quatre étapes :

1) **Évaluation** : La première étape consiste à comprendre les obligations, les lois, les politiques, les procédures, etc. Elle vise également à évaluer les risques auxquels l'organisme est exposé. Il est primordial à cette étape de bien comprendre ce qu'il faut protéger, contre qui et à quel coût.
2) **Protection** : La deuxième étape consiste à établir et à appliquer des contre-mesures pour réduire le risque d'être compromis. Cette étape peut impliquer, par exemple, d'installer des outils de contrôle ou de surveillance sur le réseau.
3) **Détection** : La troisième étape consiste à identifier des intrusions sur le réseau. Elle prend habituellement la forme d'une action non autorisée ou non acceptable envers les installations du réseau.
4) **Réponse** : La dernière étape consiste enfin à prendre des mesures pour remédier à la situation. La réponse peut prendre la forme de nouvelles règles sur un pare-feu, de l'ajout de nouveaux équipements, etc. Elle peut également impliquer la collecte de preuves pour soutenir des sanctions ou des poursuites envers les contrevenants.

L'étape de détection s'avère particulièrement intéressante, principalement à cause de son niveau de complexité. La détection des intrusions demande habituellement des connaissances techniques très importantes, en plus de bien maîtriser l'environnement de l'organisation (Goodall, Lutters et Komlodi (2009)). Par exemple, bien qu'il existe plusieurs outils pour assister les analystes dans ce travail (ex. : systèmes de détection d'intrusions (IDS)), il est habituellement nécessaire de corréler une multitude d'informations de toutes sortes pour obtenir une image claire de la situation (Goodall, Lutters et Komlodi (2004)). Nous discuterons plus en détail des défis de la détection des intrusions un peu plus loin dans ce chapitre.

Les études présentées ci-dessous démontrent qu'il existe également des processus plus ou moins formels pour structurer le travail d'analyse et de détection des intrusions. Ces études ont été réalisées auprès d'une multitude d'analystes expérimentés œuvrant dans différents milieux (éducation, finance, militaire, etc.). Elles prennent habituellement la forme

d'entrevues semi-structurées, d'examens d'artefacts d'analyse, d'évaluations dans l'environnement de travail, etc. Ces différentes études permettent de mieux comprendre le processus d'analyse en général, le rôle des analystes dans le processus et, surtout, d'en apprendre davantage sur leurs façons de travailler. Ces informations s'avèrent critiques pour comprendre les enjeux du processus d'analyse et combler efficacement leurs besoins.

Goodall, Lutters et Komlodi (2009) présentent un processus d'analyse composé de quatre étapes : **surveillance, triage, analyse et réponse**. La première étape consiste à examiner les différents systèmes à la recherche de traces d'activités anormales ou malicieuses. Règle générale, cette étape est centrée sur l'analyse des alertes produites par les IDS. Les auteurs précisent toutefois que d'autres outils de surveillance et de balayage de vulnérabilités peuvent également être utiles pour obtenir une vue plus complète de la situation. Les analystes ont également recours à plusieurs bases de connaissances (sites web, des listes de distribution, réseaux de contacts, etc.) pour connaître les nouveautés et les tendances au niveau des attaques et des vulnérabilités découvertes. Les auteurs rapportent que la plupart des responsables en sécurité interrogés au cours de l'étude doivent partager leur temps entre leurs activités liées à la sécurité et d'autres tâches de gestion. Conséquemment, le temps attribué à la surveillance et à l'analyse d'alertes provenant des IDS est très limité. Seule une fraction des alertes rapportées seront habituellement analysées plus en profondeur, même si cela implique d'échapper quelques attaques au passage. À l'inverse, les quelques spécialistes dédiés à temps plein à l'analyse de sécurité investiront une grande partie de leur temps à analyser les journaux bruts générés par les différents systèmes du réseau à la recherche d'intrusions non détectées par les IDS, de façon à détecter le plus d'attaques possible. Dans tous les cas, l'étape de surveillance nécessite la plus grande partie du temps investi par les spécialistes interrogés.

L'étape de triage consiste essentiellement à effectuer une première analyse à haut niveau pour extraire les événements anormaux de l'ensemble des données récupérées. Toutes les alertes générées par des IDS ne doivent pas nécessairement être prises au sérieux. Ces systèmes sont en effet réputés pour générer une quantité importante de bruit et de faux

positifs. Les analystes tentent donc de retrouver à travers l'ensemble des alertes générées les événements importants pour la sécurité de l'organisation.

L'étape d'analyse est habituellement déclenchée par la découverte d'une anomalie. Il peut s'agir, par exemple, d'une alerte générée par un IDS ou d'un événement anormal détecté sur le réseau (ex. : lenteur anormale rapportée par les usagers). L'analyse peut également être déclenchée par l'annonce d'une nouvelle vulnérabilité. Dans ce cas, il s'agit d'évaluer si les systèmes de l'organisation sont à risque, quel serait l'impact si elle était exploitée et quelles sont les solutions pour la contrer. Dans le cas d'une intrusion, l'analyse demande habituellement la mise en relation de plusieurs informations provenant de différentes sources. Ces informations, permettant de définir le contexte d'une alerte, sont primordiales pour déterminer s'il s'agit bel et bien d'une intrusion et pour en évaluer le degré de sévérité. Cette étape repose en grande partie sur l'expertise de l'analyste. L'expérience et les connaissances de l'environnement permettent fréquemment d'écarter rapidement des alertes « impossibles » sur les installations ou peu importantes dans le contexte de l'organisation. Par exemple, une alerte concernant un service Apache sous Linux a de fortes chances d'être écartée rapidement si elle cible un système fonctionnant sous Windows. La connaissance du réseau et des systèmes s'avère donc cruciale pour bien comprendre le contexte d'une attaque. La plupart des analystes interrogés se fient essentiellement sur leur mémoire personnelle pour obtenir ces informations, sans aucun support externe. Dans plusieurs cas, les analystes connaissent ces informations puisqu'ils ont eux-mêmes mis en place les serveurs en question. Ils jouent également fréquemment le rôle d'administrateur du réseau en plus de celui d'analyste en sécurité. Les auteurs rapportent que la plupart des analystes ont développé leurs propres outils maison pour les assister dans ce processus, n'ayant pas de solution disponible sur le marché pour répondre convenablement à leurs besoins. En ce qui concerne les réseaux de plus grande envergure, la responsabilité de l'administration des systèmes est habituellement davantage distribuée, rendant nécessaire l'usage d'outils externes pour documenter l'environnement et pour déterminer le contexte des attaques. Malgré tout, l'étape d'analyse repose essentiellement sur l'expertise et sur les connaissances des analystes.

Enfin, l'étape de réponse se rapporte essentiellement à son homonyme au niveau du processus global de sécurité. Il s'agit essentiellement d'intervenir sur le réseau pour mettre fin à une attaque ou pour éviter qu'elle ne se reproduise. Les analystes doivent fréquemment réajuster les signatures des IDS, soit pour réduire le nombre de faux positifs générés ou pour leur permettre à l'avenir de détecter plus facilement des intrusions détectées dans l'environnement. Les analystes doivent finalement documenter l'ensemble du processus pour des fins légales ou de gestion. Ces documents peuvent également faciliter des analyses similaires ultérieures.

Werlinger et al. (2010) présente un processus similaire, mais beaucoup plus détaillé. Leur approche met davantage l'accent sur l'approche collaborative et itérative du processus de détection. La figure 1.1 illustre le processus décrit par les auteurs. On remarque rapidement que le processus ne se limite plus uniquement à l'analyste. L'évaluation d'incident implique nécessairement une myriade d'intervenants, autant au niveau des gestionnaires, des utilisateurs finaux, des spécialistes en TI, etc.

Outre l'aspect collaboratif, les auteurs ajoutent une étape **préparation** au processus énoncé par Goodall, Lutters et Komlodi (2004). Cette étape consiste essentiellement à mettre en place des mesures pour prévenir les incidents. Elle implique également l'utilisation d'outils spécialisés pour évaluer les vulnérabilités sur les installations (ex. : l'application Nessus). Il suffit ensuite de mettre en place des mesures pour contrer les vulnérabilités posant un risque envers les installations.

De leur côté, D'Amico et Whitley (2008) présentent une vue un peu différente du processus de détection. Cette distinction s'explique essentiellement par le fait que les spécialistes interrogés proviennent exclusivement du U.S. Department of Defense (DoD). Le processus est donc évidemment beaucoup plus structuré et discipliné. Un schéma du processus proposé est présenté à l'annexe I. L'étape de perception / détection consiste essentiellement à surveiller les équipements et à effectuer le triage jusqu'à ce qu'un élément anormal soit détecté. L'étape de compréhension / évaluation de la situation vise ensuite à analyser

l'anomalie et à appliquer au besoin des mesures correctives. La troisième étape n'apparaît pas dans les processus présentés précédemment. Elle consiste à mettre en corrélation plusieurs incidents distincts à la recherche de patrons d'attaques. Elle permet également d'ajouter des « sources d'intelligence » pour identifier les responsables des attaques et en comprendre les motifs. À partir de ces informations, il est ensuite possible d'effectuer des prédictions sur les attaques à venir.

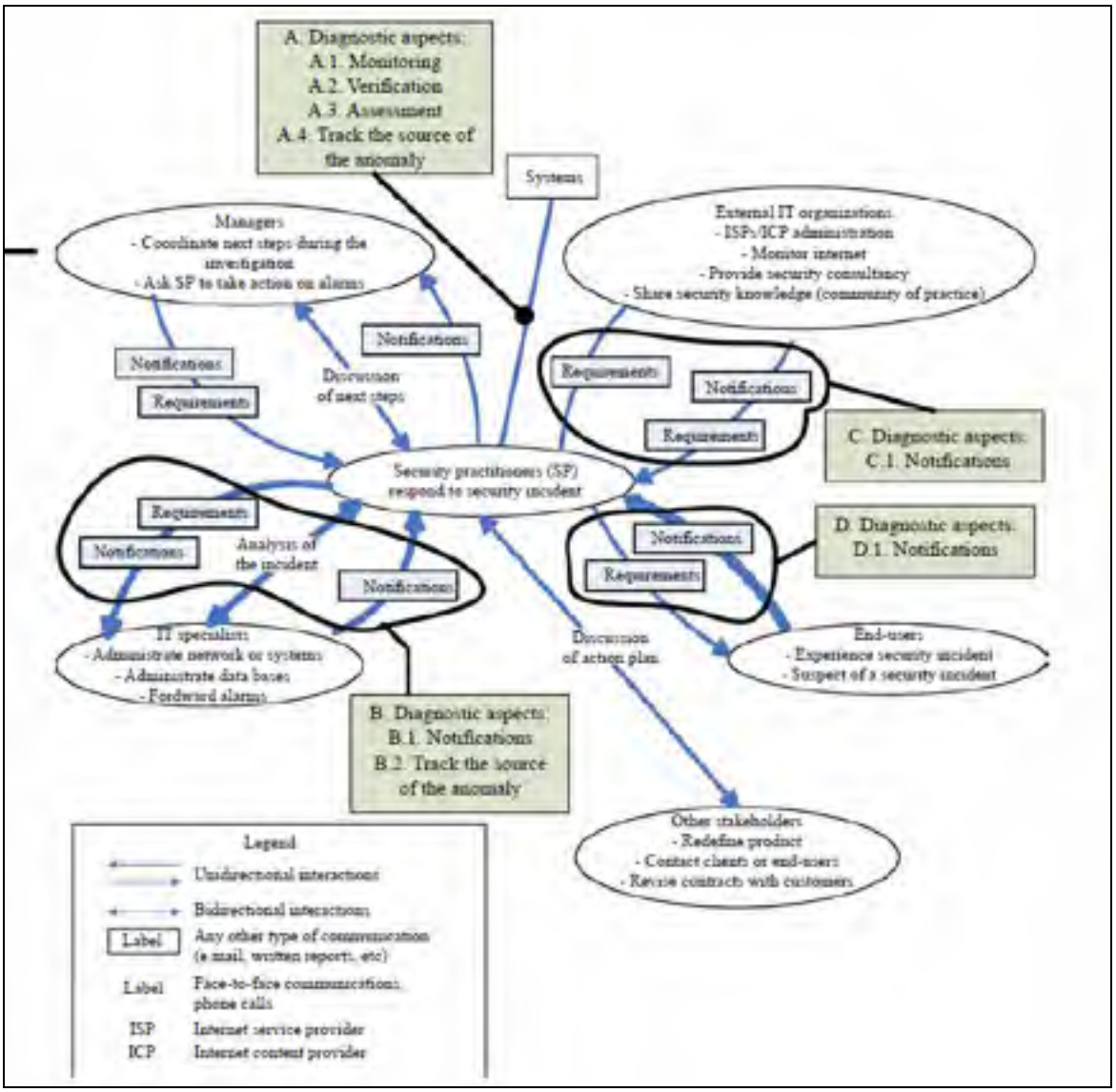

Figure 1.1 Schéma représentant le processus de détection des intrusions.
Tirée de Werlinger et al. (2010, p. 30)

Dans le même ordre d'idée, les auteurs identifient également deux catégories d'intervention : les analyses tactiques et stratégiques. Les interventions tactiques visent à se défendre contre une attaque immédiate et à maintenir l'état opérationnel des installations. Elle englobe essentiellement les étapes 1 et 2 du processus. L'analyse stratégique débute habituellement après l'analyse tactique, et vise plutôt à comprendre les implications à haut niveau des attaques et la recherche de patrons. Elle est ainsi effectuée surtout au cours de l'étape 3 du processus.

Les auteurs présentent également un processus de transformation des données. Ce processus décrit l'évolution des données au cours du processus de détection. Le tableau 1.1 présente les différentes étapes décrites par les auteurs.

Tableau 1.1 Hiérarchie des données dans le processus de détection des intrusions selon D'Amico et Whitley (2008)

Données brutes	Données provenant des IDS (non filtrées et non traitées)
Activités intéressantes	Premier filtrage pour retirer les faux positifs évidents
Activités suspectes	Éléments jugés anormaux, nécessitant une analyse plus poussée
Événements	Événements à rapporter, allant à l'encontre des politiques de sécurité de l'organisation.
Incident	Incident confirmé et documenté (intrusion)
Ensemble d'incidents	Regroupement d'incidents partageant des propriétés communes

Cette classification hiérarchique permet de mettre en évidence l'aspect itératif du processus d'analyse. Ces itérations s'avèrent nécessaires dans le cadre du DoD, dans la mesure où chaque étape est effectuée par des analystes différents. Le tableau 1.2 résume les principales tâches des analystes en fonction de leur rôle dans le processus. Règle générale, les premières étapes sont accomplies par des analystes moins expérimentés, alors que les étapes finales demandent un niveau d'expertise beaucoup plus important. Les auteurs découpent le travail des analystes selon six rôles distincts. Il est toutefois important de noter qu'un analyste peut jouer plusieurs rôles au cours du processus.

Tableau 1.2 Rôles des analystes selon D'Amico et Whitley (2008)

Triage	Première analyse des données brutes et des activités intéressantes. Vise essentiellement à retirer les faux positifs et à déterminer quels éléments doivent être analysés plus en profondeur. Les analystes ne passent en moyenne que quelques minutes sur les données.
Escalade	Enquête des activités suspectes détectées lors du triage. Elle implique habituellement de rassembler des informations provenant de plusieurs sources, de façon à obtenir une image détaillée de la situation. Le but ultime de l'escalade consiste à produire un rapport détaillé de l'incident. Ce processus peut nécessiter plusieurs heures, voire plusieurs jours.
Corrélation	Recherche de patrons et de tendances dans les données courantes et historiques. La corrélation vise à regrouper des incidents en fonction de leurs caractéristiques communes. Ce processus nécessite normalement plusieurs semaines, voire plusieurs mois.
Intelligence	Utilisation de l'intelligence pour compléter l'analyse des incidents. On entend par intelligence des sources de données externes permettant de comprendre l'identité, les motifs et les commanditaires des attaques. Il s'agit de l'une des seules analyses proactives du processus, où les résultats d'une analyse peuvent permettre de découvrir des faits avant qu'une attaque ne se produise.
Réponse	Recommande ou met en place des mesures en réponse à un incident confirmé.
Légal	Regroupe des informations pour des procédures légales suite à un incident.

L'ensemble des ouvrages cités précédemment permettent d'obtenir des informations cruciales sur la façon de travailler des analystes, sur leurs rôles et leurs besoins. Ces informations sont particulièrement difficiles à obtenir normalement, dans la mesure où l'accès à ces spécialistes est très limité. Dans la mesure où il n'aura pas été possible de rencontrer d'analyste dans le cadre du projet, les informations présentées ci-dessus serviront de base pour la définition des requis du projet.

1.4 La visualisation

Installer un système de détection d'intrusions ou tout autre outil de sécurité permet sans aucun doute d'améliorer les capacités de l'organisme pour assurer sa sécurité. Les processus présentés précédemment illustrent très clairement qu'il ne suffit pas uniquement d'implanter les outils pour être en sécurité. Encore faut-il être en mesure d'extraire les informations importantes et de réagir face aux intrusions avant qu'il ne soit trop tard, ou encore mettre en place des contre-mesures pour éviter que ces situations ne se reproduisent à l'avenir.

Conti et al. (2006) présente une étude réalisée auprès de 39 analystes professionnels et gradués en sécurité informatique. L'objectif de l'étude consistait à déterminer le nombre maximal d'alertes que les intervenants étaient en mesure de traiter manuellement (à partir des données générées par un IDS) avant d'être dépassés par les événements. Les résultats démontrent que les spécialistes étaient en mesure de traiter en moyenne un maximum de 230 alertes par heure. Malheureusement, le nombre d'alertes à traiter dépasse habituellement largement cette quantité. Par exemple, sur le campus de l'université Georgia Tech, où l'étude a été réalisée, les systèmes de détection produisent en moyenne 50 000 alarmes par jour. Comment parvenir alors à traiter autant d'information sans avoir recours à une armée d'analystes? La solution : visualiser les données.

La visualisation, d'un point de vue sécurité, consiste à générer une image basée sur les données contenues dans les entrées de journaux des appareils. Elle définit de quelle façon les entrées du journal sont transformées sur la représentation visuelle (Marty (2008)). L'intérêt de la visualisation repose sur les capacités cognitives de l'humain. En effet, selon Ware (2004), les yeux et le cortex visuel du cerveau forment un processeur massivement parallèle, offrant une bande passante très élevée pour le centre cognitif de l'humain. Selon l'auteur, en déchiffrant le fonctionnement de la perception humaine, il est possible d'utiliser ces connaissances pour former des représentations graphiques permettant aux éléments importants des données de ressortir par elles-mêmes. Il existe une multitude d'ouvrages sur les bases de la visualisation et de la perception. Outre Ware (2004), on retrouve également

Card, Mackinlay et Shneiderman (1999), Tufte (1986), Spence (2007), et plusieurs autres. Ces ouvrages démontrent très clairement qu'il est possible de communiquer une quantité très importante d'informations à l'utilisateur, et d'ainsi déjouer les limites des approches textuelles conventionnelles. En plus de permettre de traiter une quantité beaucoup plus importante de données, la visualisation apporte plusieurs autres avantages. Par exemple, selon Marty (2008), la visualisation permet :

- **De répondre à des questions :** La visualisation permet de mettre en relation les différentes entrées afin de répondre à des interrogations auxquelles il serait très difficile de répondre autrement.
- **De poser de nouvelles questions :** Il arrive fréquemment que les patrons présentés par une visualisation ne correspondent pas à l'idée que l'on se faisait des données. Ces nouvelles découvertes peuvent jeter une vision différente sur un problème, et suggérer de nouveaux questionnements.
- **Explorer et découvrir :** Différentes représentations ou configurations permettent souvent d'identifier des informations ou des relations inconnues jusqu'à présent.
- **Supporter les décisions :** La visualisation permet de transformer de gros ensembles de données en information logique et utilisable. Elle permet donc de prendre des décisions à l'aide d'informations qui n'auraient possiblement pas été accessibles autrement. Bien comprendre une situation permet de prendre une décision plus éclairée.
- **Communiquer l'information :** La visualisation possède également un important pouvoir de communication que des journaux textuels ne possèdent pas, par exemple. Il est beaucoup plus facile pour un intervenant externe de comprendre rapidement la situation avec quelques images qu'avec des piles de documents textes.
- **Augmenter la productivité :** Il est beaucoup plus facile et rapide de se référer à une image pour détecter des erreurs ou des situations anormales que d'avoir à explorer des milliers de lignes de données brutes.

Chabot (2009) propose quatre cas classiques où la visualisation est particulièrement intéressante :

- Pour analyser et comprendre des ensembles de données volumineux;
- Pour analyser et comprendre des ensembles de données complexes;
- Pour découvrir et comprendre de nouveaux paradigmes visuels;
- Pour découvrir et comprendre des éléments cachés dans les données.

On remarque rapidement que les données traitées par les analystes en sécurité répondent à l'ensemble de ces situations. En effet, tel que démontré précédemment, les analystes doivent naviguer à travers des dizaines de milliers d'alertes de toutes sortes. Ils doivent mettre en corrélation une multitude de données brutes complexes provenant de différentes sources afin de détecter les attaques et d'en comprendre les détails.

1.5 La visualisation des alarmes d'intrusions

L'utilisation des techniques de visualisation en sécurité n'est pas un concept nouveau. Tamassia, Palazzi et Papamanthou (2009) présente un survol intéressant des principales utilisations de la visualisation dans le domaine. La visualisation est particulièrement intéressante lorsque le nombre d'éléments à analyser est très important. Elle est donc très appropriée pour l'analyse des alarmes générées par des IDS. Plusieurs approches ont été présentées au cours des dernières années sur ce sujet. Nous diviserons ces approches selon quatre grandes tendances en visualisation d'alarmes :

- Les approches textuelles et tabulaires;
- Les approches « classiques » et coordonnées;
- Les approches par pixel;
- Les approches hybrides ou non conventionnelles.

1.5.1 Les approches textuelles et tabulaires

La première catégorie regroupe essentiellement l'ensemble des approches tabulaires, où certains éléments graphiques ont été insérés pour faciliter la compréhension de l'usager. Parmi les meilleures approches de cette catégorie, citons SnortView Koike et Ohno (2004) (figure 1.2) et TNV Goodall et al. (2005) (figure 1.3).

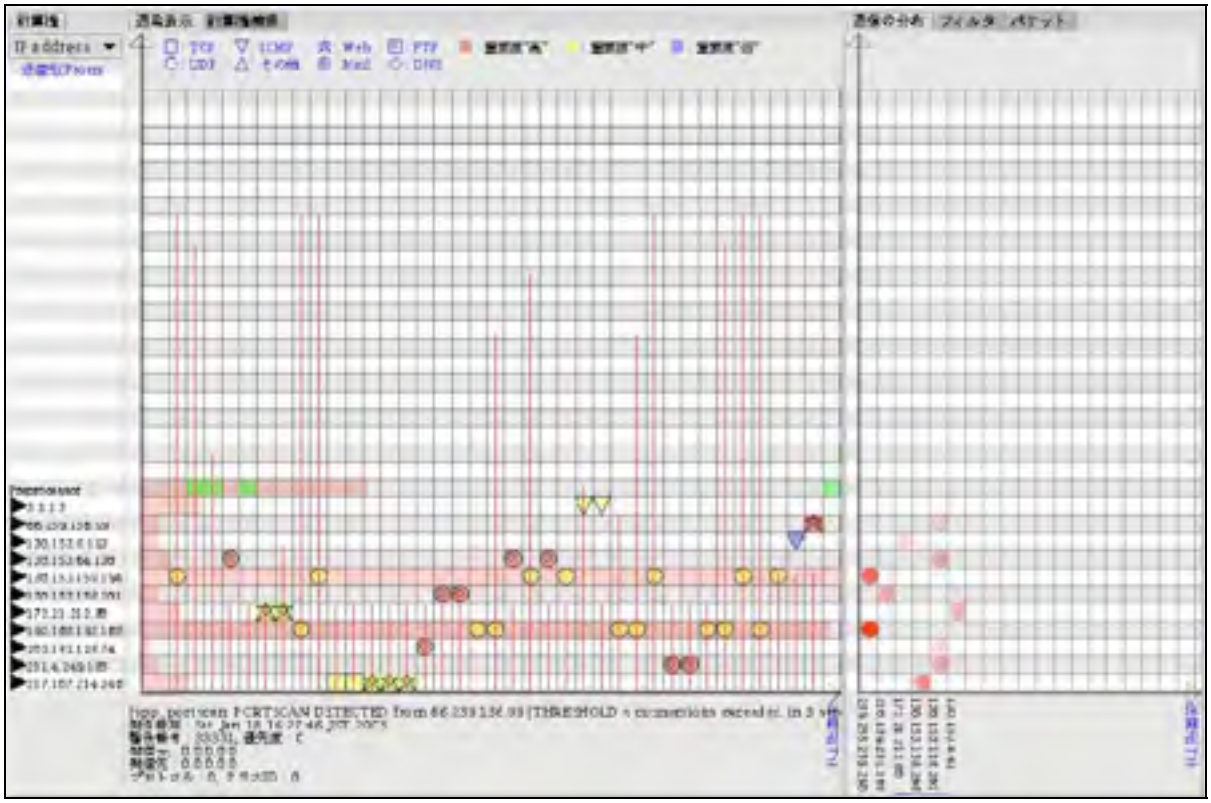

Figure 1.2 Capture d'écran de SnortView utilisant une approche tabulaire
Tirée de Koike et Ohno (2004, p. 144)

Ces approches se rapprochent souvent du niveau de détail obtenu avec les approches manuelles, tout en intégrant certains avantages de la visualisation. Par exemple, elles permettent d'obtenir en détail l'ensemble des entrées des journaux générés par les appareils de façon plus structurée et organisée qu'uniquement avec les entrées textuelles. L'intégration de couleurs ou de glyphes (symboles ayant des significations explicites) permet de faciliter l'analyse des données en dirigeant le centre d'attention de l'usager sur les éléments

importants des données. Ces approches ont également habituellement l'avantage d'être le plus près possible des données brutes. Chaque entrée est en effet disponible visuellement sur l'interface, avec l'ensemble de ses paramètres facilement accessible (par exemple, avec une infobulle). Malheureusement, ces approches deviennent beaucoup moins intéressantes lorsque le nombre d'alertes augmente significativement. En effet, ces approches ne permettent pas d'isoler les alertes importantes dans un grand ensemble de données. Elles s'avèrent donc plus intéressantes pour obtenir une vue détaillée qu'une vue globale de la situation (*situational awareness*) sur un réseau.

Figure 1.3 Interface de TNV, une autre approche tabulaire
Tirée de Goodall et al. (2005, p. 1404)

1.5.2 Les approches « classiques » et coordonnées

La deuxième catégorie regroupe l'ensemble des approches classiques de visualisation. Ces approches intègrent des techniques de visualisation bien connues et déjà utilisées dans la plupart des domaines de la visualisation. Par exemple, on retrouve dans cette catégorie les graphiques statistiques, telles que les approches basées sur les histogrammes (ex : Musa et Parish (2008)), les *treemaps* (ex : Mansmann et al. (2009)), les réseaux (ex : D'Amico et al. (2007)), les glyphes (ex : Komlodi et al. (2005)), les coordonnées parallèles (ex : Lee et Copeland (2006)), les *heat maps* (ex : Günter et al. (2003)), etc.

La figure 1.4 illustre un exemple de l'utilisation des treemaps pour représenter des attaques sur un réseau. Chaque rectangle du *treemap* correspond à une adresse d'un système à l'interne (la victime), et les points situés à l'extérieur représentent les attaquants. Les figures deviennent rapidement illisibles lorsque le nombre d'attaques augmente, tel qu'illustré par la figure de droite.

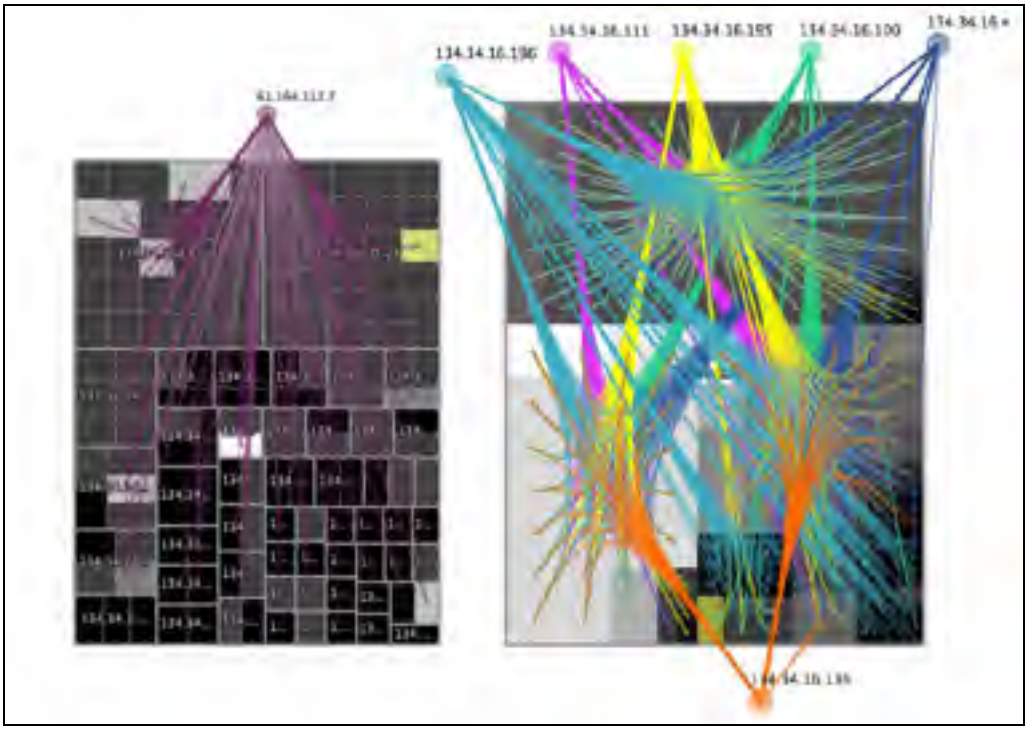

Figure 1.4 Détection des intrusions à l'aide des *treemaps*.
Adaptée de Mansmann et al. (2009)

De façon générale, ces approches permettent plus facilement d'obtenir une vue à haut niveau de la situation sur le réseau. Elles permettent en effet habituellement d'offrir une synthèse de l'information à l'utilisateur, basée sur les caractéristiques les plus importantes des données à analyser. Ces techniques visent essentiellement à mettre en évidence toute situation déviant de la norme et à en évaluer l'impact. Par exemple, représenter les intrusions sur un treemap (figure 1.4) permet d'évaluer rapidement l'ampleur et le niveau de sévérité d'une attaque (par exemple, le degré de criticité des systèmes sous attaque).

La figure 1.5 illustre une autre utilisation classique de la visualisation pour représenter des relations. L'axe de gauche illustre le port du service exploité, alors que l'axe de droite illustre l'adresse de l'attaquant. L'application propose également une vue tabulaire plus classique et des informations plus détaillées dans la section de gauche. On retrouve également des options de filtrage afin de limiter la quantité d'information sur la visualisation.

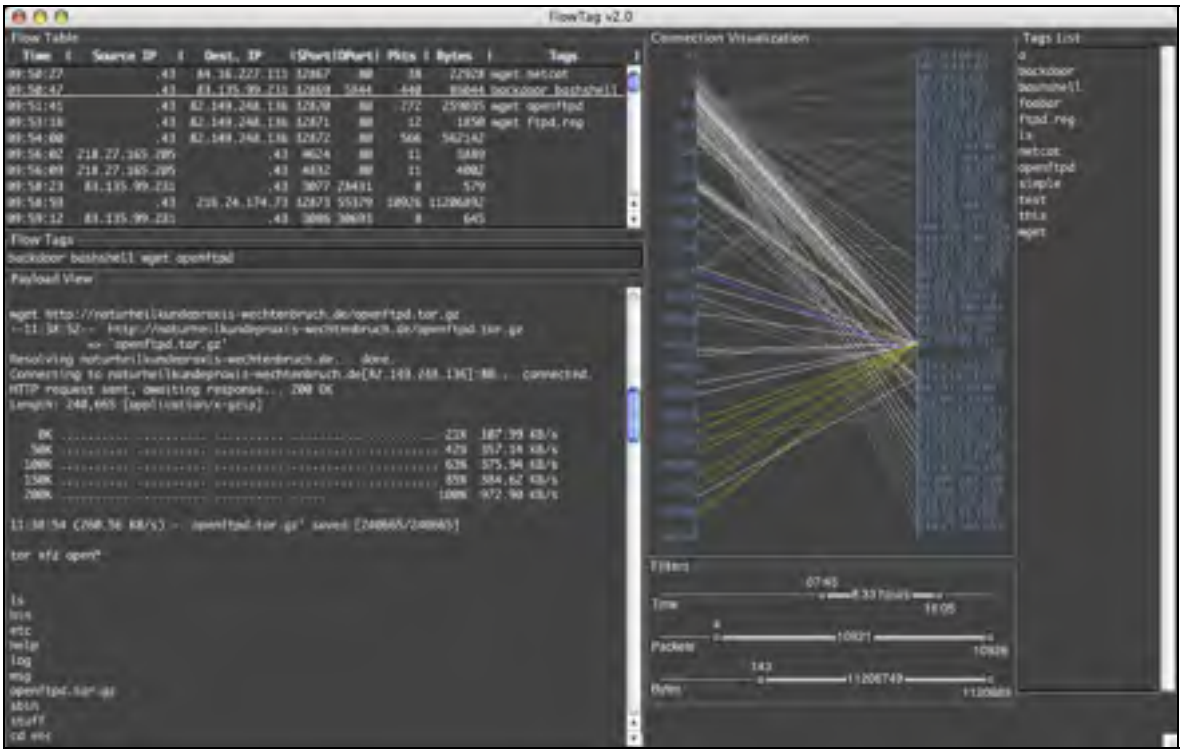

Figure 1.5 Détection des intrusions à l'aide des coordonnées parallèles.
Tirée de Lee et Copeland (2006, p. 3)

Malheureusement, bien que ces solutions s'adaptent un peu mieux que les solutions tabulaires pour des grands ensembles, on observe fréquemment de l'occlusion lorsque le nombre de données augmente significativement. Il devient très difficile de déterminer vers quelles adresses sont orientées les attaques, surtout pour les courbes roses par exemple, qui se trouvent en dessous de la pile. La figure 1.4 droite illustre le problème d'occlusion causé par un trop grand nombre de données affichées simultanément. Il est très difficile de déterminer avec précision la destination des courbes aux points où les courbes se superposent. Certaines solutions classiques n'ont heureusement pas ce problème. Par exemple, même avec plusieurs centaines de milliers d'alertes, des histogrammes ne causeront aucun problème d'occlusion sur l'interface. Malheureusement, la plupart de ces solutions ne permettent de présenter qu'un résumé ou une petite sélection des caractéristiques des données. Il faut donc les compléter avec d'autres visualisations ou sources d'informations (ex. : journaux bruts) pour obtenir une vue détaillée de la situation. Pour contrer ce problème, plusieurs chercheurs ont proposé des approches de type « tableau de bord » combinant une multitude de visualisations simples pour obtenir une vue complète de la situation. Par exemple, Goodall et Tesone (2009) et D'Amico et al. (2007) proposent une série de vues coordonnées permettant d'analyser des ensembles complexes de données, tel qu'illustré sur la figure 1.6.

L'interface proposée par l'équipe de John Goodall intègre plusieurs visualisations classiques, incluant des coordonnées parallèles et des réseaux pour représenter les communications entre les systèmes, des histogrammes pour illustrer les cumulatifs d'alertes pour chaque catégorie, des nuages de points pour illustrer des corrélations entre les différentes variables du système, des données tabulaires pour présenter les données brutes, etc. L'ensemble de ces vues sont coordonnées de façon à afficher les mêmes données, les mêmes filtres, mettre en surbrillance les mêmes éléments, etc. Ce type de solution demande malheureusement à l'utilisateur d'analyser et de corréler lui-même les informations provenant d'une multitude de graphiques pour comprendre la situation sur le réseau. Il s'agit donc d'une solution beaucoup plus complexe, demandant un certain niveau d'expertise pour parvenir à faire ressortir les événements importants.

Figure 1.6 Interface de VIAssist, une approche proposant des vues coordonnées
Tirée de Goodall et Tesone (2009, p. 200)

1.5.3 Les approches par pixel

La troisième catégorie regroupe les visualisations où un ensemble de données est représenté par un seul pixel sur l'interface. Le nuage de points représente un exemple classique de visualisation par pixels. Ce type de visualisation pourrait également être considéré comme une approche classique, dans la mesure où une multitude de visualisations utilisent cette approche dans tous les domaines de la visualisation. Nous avons toutefois préféré isoler cette technique des autres approches classiques puisque la forme et le sens de la représentation varient souvent en fonction du domaine étudié. Une grande partie des techniques de visualisation d'intrusions / d'alarmes présentées dans la littérature au cours des dernières années pourraient se retrouver dans cette catégorie. Ce phénomène s'explique par la flexibilité qu'offre ce type de visualisation. En comparaison avec la plupart des autres approches, il est en effet possible de regrouper sur une même interface beaucoup plus d'information, dans la mesure où un ensemble de données ne requiert en général qu'un seul pixel. Parmi les approches les plus populaires, citons IDS Rainstorm de Abdullah et al. (2005) (figure 1.7) et IP Matrix de Koike, Ohno et Koizumi (2005) (figure 1.8).

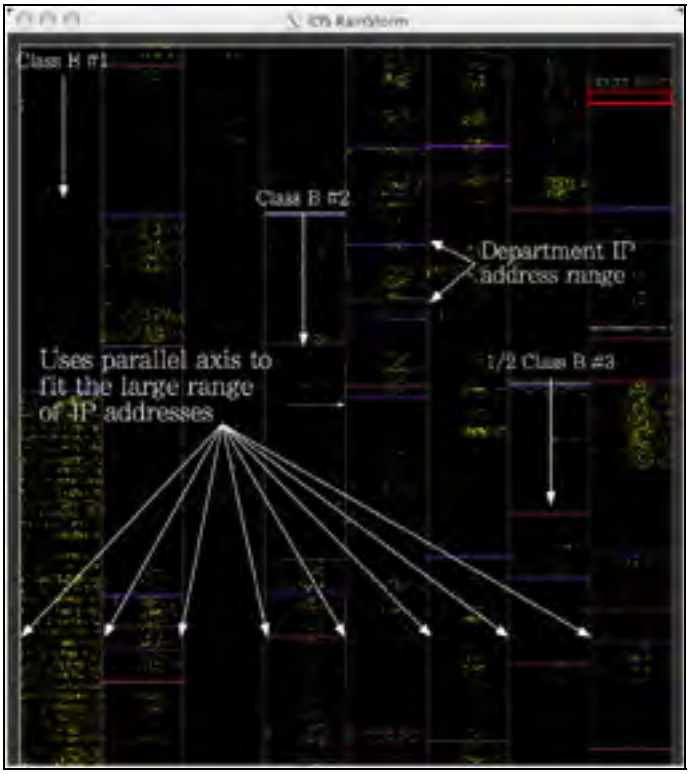

Figure 1.7 Interface principale d'IDS Rainstorm
Tirée de Abdullah et al. (2005, p. 3)

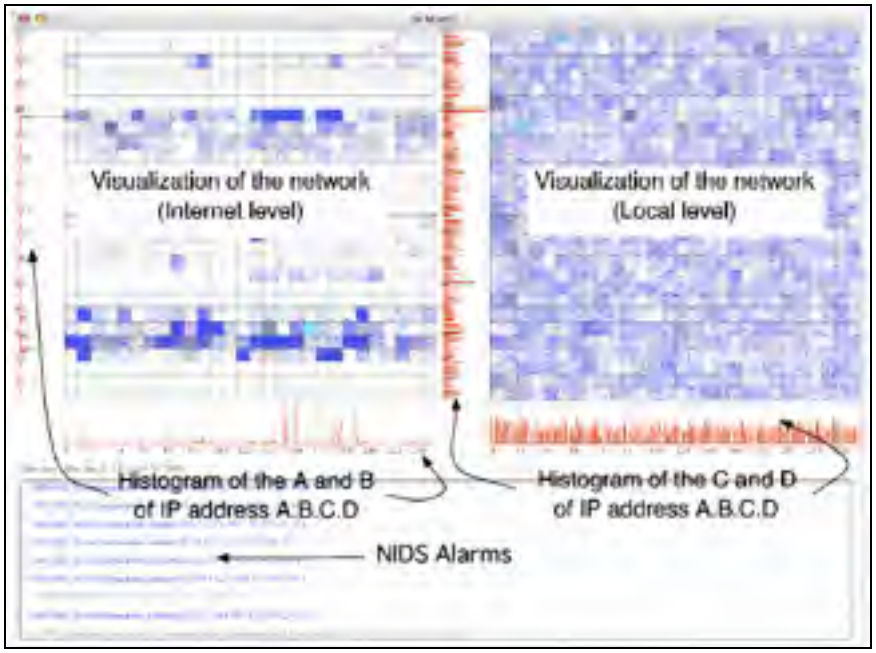

Figure 1.8 Interface de IP Matrix
Tirée de Koike, Ohno et Koizumi (2005, p. 93)

IDS Rainstorm (figure 1.7) permet d'afficher un résumé des alertes générées durant une période de 24 heures sur deux réseaux et demi de classes B, soit 163 840 adresses IP distinctes. Pour contrer la limite de pixels disponibles sur l'écran, les auteurs ont proposé d'utiliser plusieurs axes parallèles (colonnes). Chaque pixel de l'écran représente donc ici les alertes générées au cours d'un intervalle de 20 minutes pour environ 20 adresses IP. La couleur du pixel représente ici le niveau de sévérité.

IP Matrix (figure 1.8) propose une interface similaire permettant de représenter les alertes répertoriées sur un réseau international de *honeypots*. Il s'agit d'ailleurs de l'une des rares approches proposées pour visualiser ce genre de données. Les auteurs décomposent ici l'adresse en deux segments. En considérant une adresse A.B.C.D, les auteurs affichent respectivement les valeurs A et B sur les axes X et Y du premier graphique, et les valeurs C et D sur les axes X et Y du deuxième graphique. Cette technique permet d'afficher l'ensemble des adresses IPv4 possibles, soit 2^{32} adresses, sur deux graphiques de 256x256 pixels. Chaque pixel représente une fois de plus le degré de sévérité de l'alerte. Cette approche permet facilement par exemple d'identifier les plages d'adresses touchées par la propagation d'un vers sur la toile.

Les approches par pixels offrent l'avantage de s'adapter plus facilement à de larges ensembles de données sans nécessairement souffrir des problèmes d'occlusion. Il est possible de représenter beaucoup d'entrées individuelles (l'écran possède une quantité très importante de pixels), mais le nombre d'informations pour chaque entrée est nécessairement limité (intensité ou couleur du pixel). Par exemple, pour la figure 1.7, la couleur du pixel représente uniquement le degré de sévérité maximal des alertes générées pendant l'intervalle de 30 minutes représenté par le pixel. Il n'est pas possible de déterminer, à un instant précis, le nombre d'alertes générées, l'identité de l'attaquant, le type d'alerte, etc. Il est uniquement possible de déterminer que pour cet intervalle, au moins une alerte de sévérité X a été répertoriée. Il devient donc nécessaire de naviguer entre plusieurs vues distinctes pour obtenir une image complète de ce qui se passe sur le réseau.

1.5.4 Les approches hybrides ou non conventionnelles

La dernière catégorie regroupe les approches hybrides combinant différentes techniques présentées jusqu'à présent, ainsi que les autres approches non conventionnelles. L'approche la plus intéressante de cette catégorie dans le cadre du projet a été proposée par Foresti et al. (2006). VisAlert propose une visualisation radiale permettant de corréler de façon visuelle les informations provenant d'un IDS, de journaux systèmes, etc. L'application permet, sur un même graphique, d'illustrer la source, la destination, le type, la fréquence des attaques et même la topologie du réseau. La solution repose sur le concept du w^3 (ou la règle des trois "w"), soit les *What* (Quoi), *When* (Quand) et *Where* (Où). Les auteurs illustrent clairement le lien entre une alerte précise et une adresse sur le réseau. La notion de fréquence est également considérée par la taille des cercles au centre, représentant les adresses sur le réseau, et par l'épaisseur des liens alerte → adresse IP. Malheureusement, la solution n'est pas parfaite non plus. En effet, le fait d'utiliser la taille des cercles des adresses IP (au centre) pour représenter la fréquence des alertes cause nécessairement de l'occlusion, dissimulant les cercles situés à proximité (le phénomène est clairement visible sur la figure 1.9). Le graphique devient également rapidement illisible lorsque le nombre d'adresses augmente significativement, les cercles devant nécessairement se superposer, ou le nombre de liens devenant beaucoup trop important pour permettre une lecture convenable.

Jusqu'à présent, nous n'avons pas été en mesure de trouver une visualisation (unique) capable de supporter à la fois un grand nombre d'adresses IP et un nombre suffisant d'informations pour obtenir une idée claire des événements en cours sur le réseau, et ce, sans causer d'occlusion sur la visualisation.

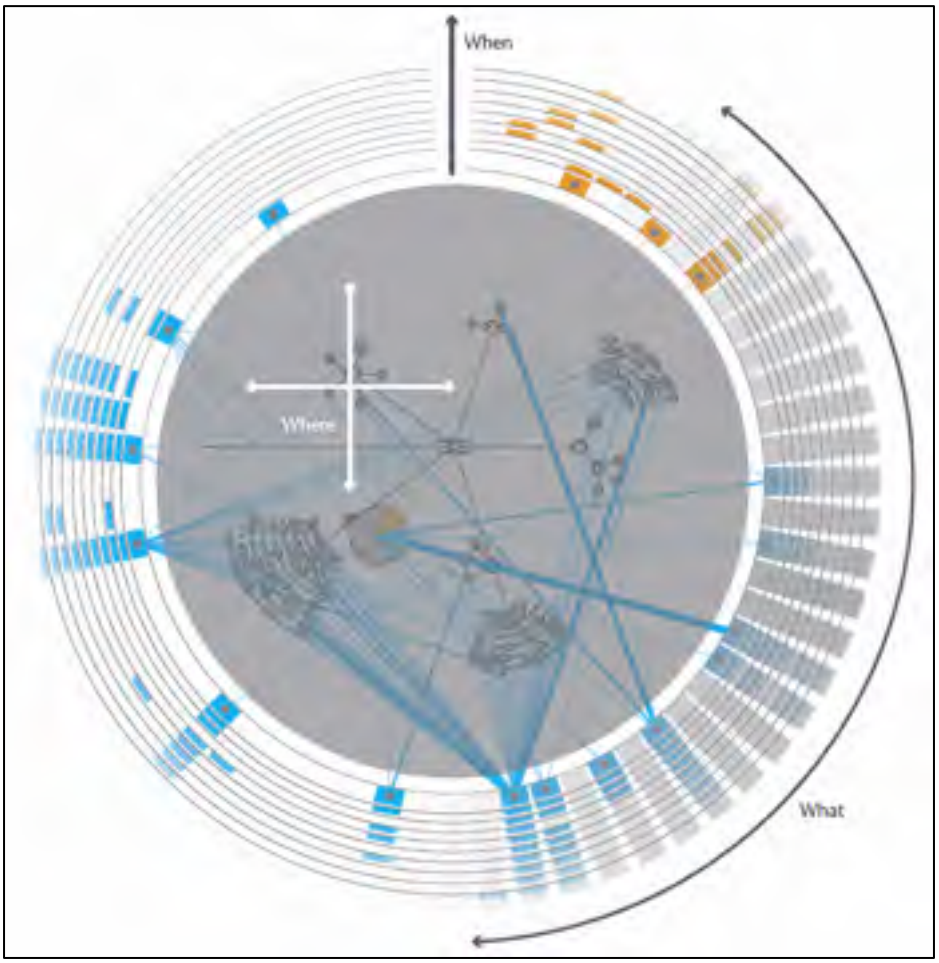

Figure 1.9 Vue principale de VisAlert
Tirée de Foresti et al. (2006, p. 52)

1.6 Le regroupement des liens

VisAlert représente une alerte par un lien entre une signature (définition de l'attaque) et une adresse IP (machine victime à l'interne du réseau). Ce type de représentation correspond à un graphe biparti, c'est-à-dire un graphe composé de deux ensembles distincts de nœuds où chaque lien relie un nœud d'un ensemble à un nœud de l'autre ensemble. Malheureusement, la représentation utilisée par VisAlert cause un niveau important d'occlusion sous les liens et sous les nœuds internes. Par contre, ce problème n'est pas nouveau, et plusieurs recherches ont été effectuées au cours des dernières années pour limiter ce genre d'ennui lors de la visualisation de graphes bipartis.

Une technique couramment employée consiste à regrouper les liens afin de réduire le désordre sur la représentation (connu sous le nom de *edge bundling*). Les regroupements peuvent être au niveau de la tige (aux extrémités) ou sur une section intermédiaire (au centre), tel qu'illustré par la figure 1.10.

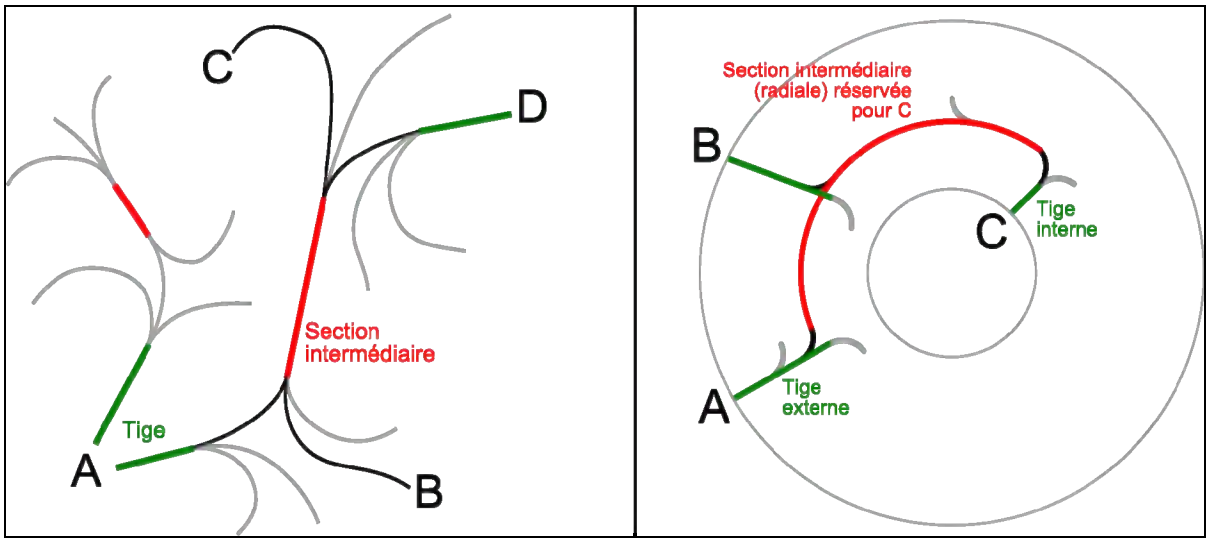

Figure 1.10 Utilisation habituelle du regroupement des liens (gauche) et approche utilisée par AlertWheel (droite).

La plupart des approches observées dans la littérature n'utilisent qu'une seule des deux techniques de regroupement. Par exemple, Holten et Van Wijk (2009) présente une approche où le regroupement est effectué sur la section intermédiaire, tel qu'illustré sur la figure 1.11. Bien que cette approche améliore grandement la lisibilité du graphique, elle crée malheureusement de l'ambiguïté au niveau des liens. En effet, il est impossible de déterminer la source et la destination des liens à l'aide d'une représentation réduite de cette façon, à moins de considérer le graphe comme biparti complet (où tous les nœuds d'un ensemble sont connectés à tous les nœuds de l'autre ensemble). Par exemple, sur la figure 1.10 gauche, il est impossible de déterminer si le nœud A est connecté uniquement au nœud C, D, ou à tous les deux.

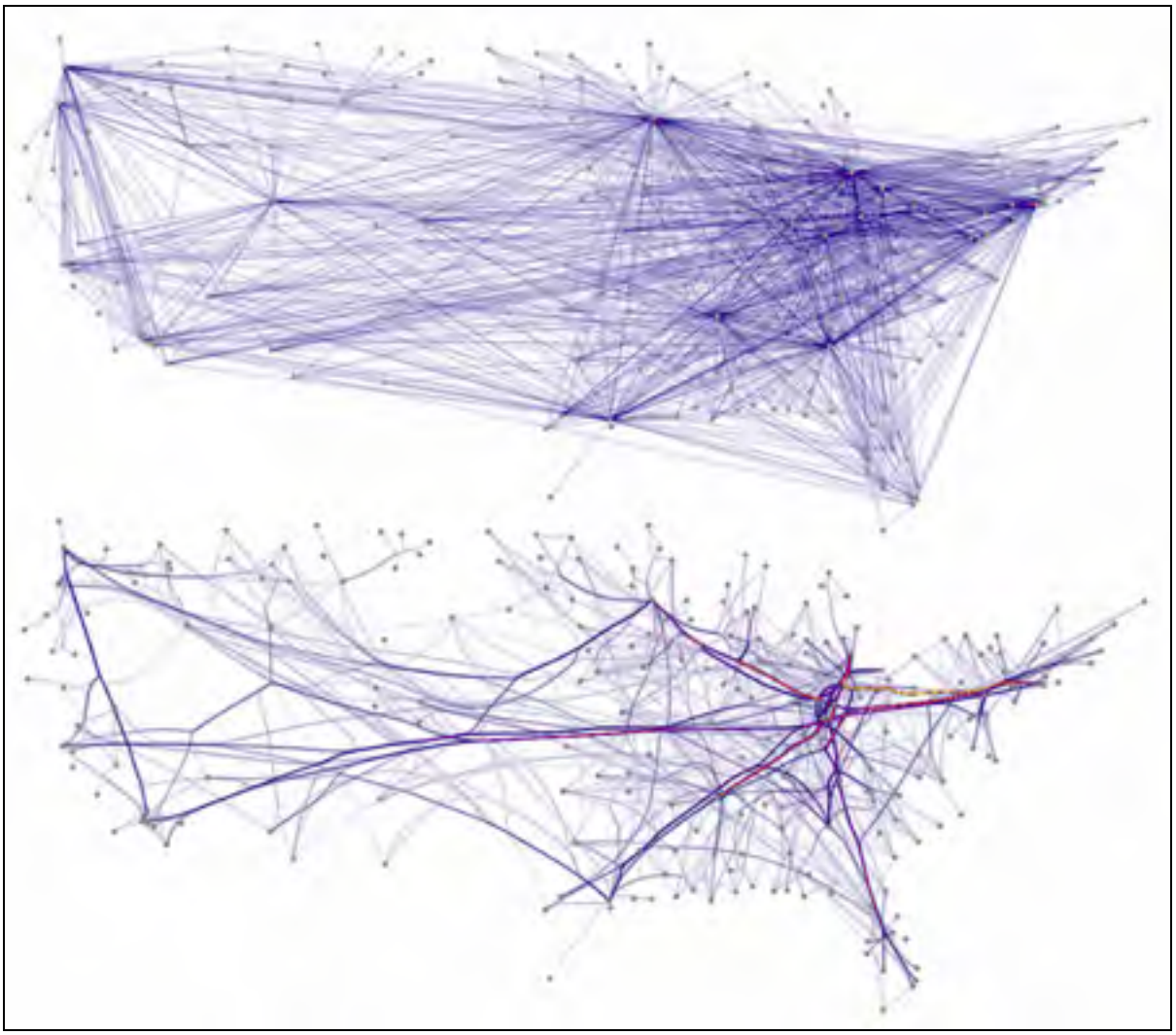

Figure 1.11 Représentation sans regroupement (haut) et avec regroupement (bas) par section intermédiaire (*edge bundling*) appliqué à un réseau de nœuds classique (non biparti) Adaptée de Holten et Van Wijk (2009)

D'autres ouvrages ont toutefois présenté des approches permettant d'utiliser le regroupement par section intermédiaire de façon non ambiguë. Par exemple, Dickerson et al. (2005) utilise ces regroupements pour relier des sous-graphes bipartis complets. Par contre, pour que cette approche puisse être employée sur la figure 1.10, il faudrait que A et B soient tous deux connectés à C et à D. Il s'agit d'une contrainte plutôt gênante pour représenter des alertes sur un réseau, puisque les liens des adresses sources ou destinations vers les signatures, par exemple, ne forment généralement pas un graphe biparti complet.

Tel que démontré dans Lex et al. (2010), il est également possible d'effectuer le regroupement par tige (*stem bundling*). Cette approche consiste à regrouper l'ensemble des liens provenant d'une source ou se dirigeant vers une destination. Ce type de regroupement possède l'avantage de ne pas créer d'ambiguïté, dans la mesure où la source et la destination sont clairement représentées (voir figure 1.10 droite). Selon nos recherches, Pupyrev, Nachmanson et Kaufmann (2011) semble être la seule approche intégrant les deux types de regroupements (tige et section intermédiaire) sur un même graphique (figure 1.12). Cette approche évite l'ambiguïté en ne juxtaposant pas complètement les segments regroupés. Il est donc possible de visualiser chaque lien individuellement en se rapprochant suffisamment de la figure. En contrepartie, puisque tous les liens sont dessinés individuellement, il est nécessaire de réserver des espaces importants pour chaque regroupement lorsque le nombre de liens augmente significativement. van Ham, Wattenberg et Viegas (2009) proposent une troisième technique de regroupement : la compression des liens (edge compression). Ce type de regroupement consiste à regrouper l'ensemble des nœuds possédant uniquement des liens vers des destinations communes. La figure 1.13 illustre ce concept.

En observant le graphique de gauche, on remarque rapidement que les nœuds B et D possèdent exactement les mêmes voisins. Même constat pour les nœuds C, E et F. Il est donc possible de réduire la représentation en regroupant tous les nœuds possédant les mêmes voisins. La figure de droite représente le graphique compressé à l'aide de cette technique.

Jusqu'à présent, il a été impossible de trouver une approche combinant les trois techniques de regroupement. Également, aucune solution présentée jusqu'à présent n'applique ces concepts à une visualisation radiale, telle que proposée dans le cadre de ce projet.

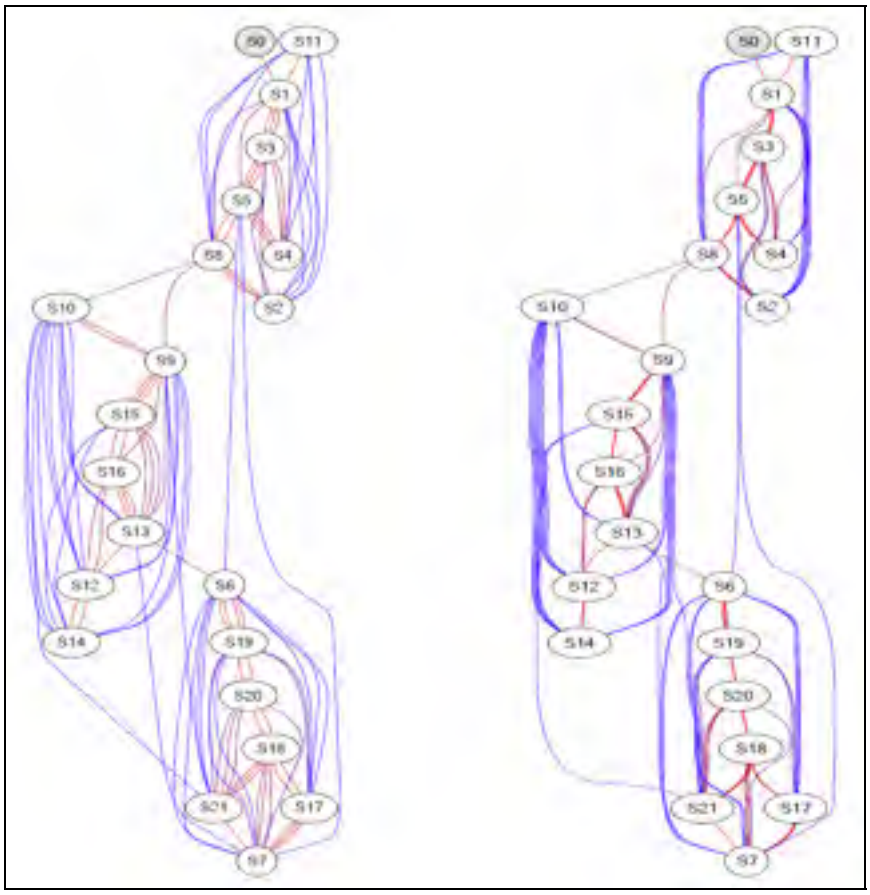

Figure 1.12 Regroupement par tige et par section intermédiaire appliqué à un réseau classique (non biparti)
Tirée de Pupyrev, Nachmanson et Kaufmann (2011, p. 330)

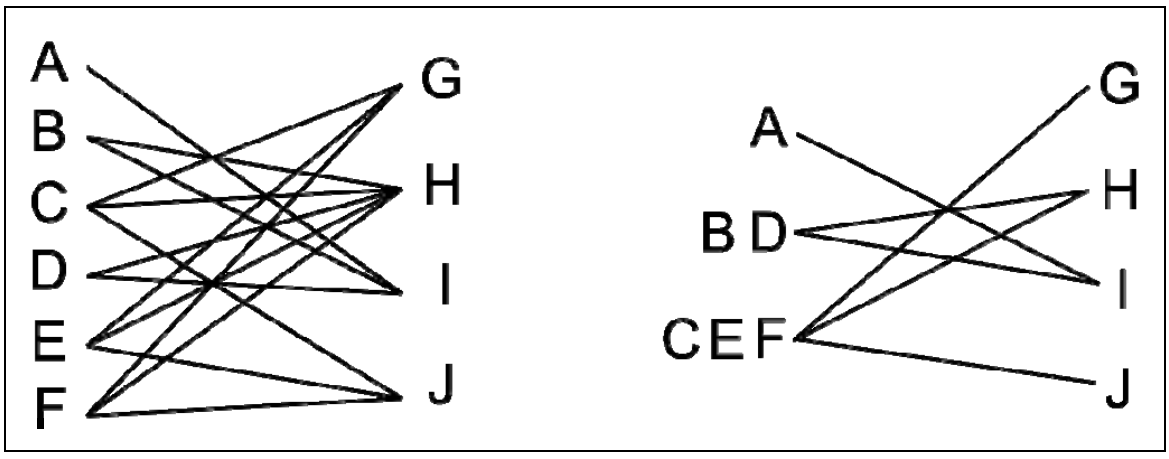

Figure 1.13 À gauche, représentation conventionnelle des liens dans un graphe biparti. À droite, réduction du graphique à l'aide de la compression des liens

CHAPITRE 2

PROBLÉMATIQUE

2.1 Complexité de l'analyse

Le principal défi des analystes en sécurité consiste à donner un sens à la pléthore de données générées par les différents systèmes sur le réseau. Nous avons mentionné au chapitre précédent qu'un analyste professionnel expérimenté peut gérer en moyenne 230 alertes par heure, mais qu'en réalité, le nombre est significativement plus élevé. Plusieurs affirment alors devoir choisir arbitrairement les alertes qu'ils évalueront plus en détail, quitte à en échapper quelques-unes au passage. Sans support approprié, il est donc très difficile pour les analystes de prioriser efficacement les événements.

Les systèmes de détection facilitent évidemment un peu le travail en guidant les analystes sur des événements suspects détectés sur le réseau. Malheureusement, ces appareils sont réputés pour produire une quantité importante de faux positifs et de bruit (ex. : doublons, événements anodins, etc.). Les véritables alertes importantes pour la sécurité des installations se retrouvent perdues à travers la multitude d'événements rapportés par les IDS. Le temps de réponse joue également souvent un rôle majeur dans le succès d'une opération. Détecter une intrusion quelques jours après qu'elle ait débuté peut être fatal pour une organisation. Il est primordial que la réponse à une attaque survienne le plus rapidement possible pour en réduire les impacts, et éviter dans le meilleur des mondes que les ressources critiques soient atteintes.

Plusieurs spécialistes dans le domaine voient la solution à travers les systèmes de prévention des intrusions (IPS). Contrairement aux IDS, ces derniers jouent un rôle actif au niveau de la sécurité du réseau, et peuvent réagir dès qu'une intrusion est détectée, soit en changeant les règles d'un pare-feu, en fermant des ports, etc. Bien que cette approche semble intéressante de prime abord, elle ne fait pas l'unanimité à travers les spécialistes dans le domaine. Par exemple, plusieurs analystes interrogés au cours des différentes études présentées au chapitre 1 montrent un regard plutôt sceptique face à cette approche entièrement automatisée.

> « The central breakdown that motivates our analysis is an industry-wide attempt to automate all network security activities, without regard for the critical role of human expertise, from network monitoring through automated response via intrusion prevention systems (IPS). […] All of our participants agreed that fully automated IDSs are never a completely effetive solution, and despite attempts at automated solutions, there is no substitute for the intuition that human analysts bring to bear on the process. » Goodall, Lutters et Komlodi (2009, pp. 92-93)

Bien que les IPS aient l'avantage de la rapidité d'exécution, ils ne possèdent pas les capacités d'analyste, d'interprétation et de corrélation des analystes. Puisque les IPS possèdent la même faiblesse que les IDS au niveau des faux positifs, il n'est pas rare que ceux-ci interdisent du trafic légitime en croyant mettre fin à une intrusion. Les IPS ne possèdent pas non plus l'expérience et les connaissances d'un analyste sur l'environnement et les différents systèmes de l'organisation. Pire encore, plusieurs attaques sont difficilement détectables à partir d'une seule source d'informations. Il est donc nécessaire de corréler des informations provenant de plusieurs sources distinctes pour obtenir l'heure juste sur la situation en cours. L'ensemble de ces éléments sont très difficiles à intégrer dans une solution automatisée, d'où la nécessité de maintenir l'analyste dans le processus de sécurité.

La figure suivante illustre le contenu brut d'une alerte Snort dans un format compréhensible par un spécialiste :

```
[**] [1:477:3] ICMP Packet [**]
[Priority: 0]
07/27-20:41:57.230345 > l/l len: 0 l/l type: 0x200 0:0:0:0:0:0
pkt type:0x4 proto: 0x800 len:0x64
209.85.231.102 -> 209.85.231.104 ICMP TTL:64 TOS:0x0 ID:0 IpLen:20
DgmLen:84 DF
Type:8  Code:0  ID:24905   Seq:1  ECHO
```

Figure 2.1 Exemple d'une alerte générée par Snort pour un paquet ICMP

Cet exemple illustre la complexité des données à analyser. Analyser manuellement des milliers d'alertes de ce genre par jour manuellement peut s'avérer une tâche impossible à

réaliser, même pour un spécialiste dans le domaine. Certes, toutes les valeurs de cet exemple ne présentent pas le même intérêt pour un analyste. Par exemple, tel que discuté au chapitre 2, lors de l'analyse initiale, les champs suivants seront particulièrement intéressants : l'heure, les adresses IP source et destination, les ports, la signature et la priorité. Même avec une liste réduite de champs à vérifier, il est très difficile d'obtenir une image globale de la situation lorsque le nombre d'alertes dépasse quelques dizaines.

Une technique intéressante consiste donc à utiliser la visualisation pour explorer efficacement les ensembles de données. Parmi les principales critiques observées chez les analystes par rapport aux solutions existantes, notons la déconnexion des concepteurs de logiciels avec la réalité observée sur le terrain. Cette déconnexion s'explique en partie par le manque de connaissance du processus d'analyse. Par exemple, la plupart des outils fonctionnent de manière indépendante, et ne permettent pas le travail collaboratif. Tel que nous l'avons démontré dans la section 1.3, le processus d'analyse est itératif, et implique généralement plusieurs intervenants internes et externes. Par exemple, le simple fait de pouvoir conserver les traces des événements découverts dans les données et de pouvoir les annoter permettrait de simplifier le passage d'un analyste à un autre au cours des différentes phases du processus. Il est donc primordial de bâtir une solution sur les besoins véritables des analystes, en fonction de leur façon de travailler et de communiquer l'information.

2.2 L'analyse des pots de miel (Honeypots)

A première vue, il n'existe pas de différence majeure entre le format des alertes générées par un IDS sur un réseau d'entreprise et un IDS sur un *honeypot*. La différence repose essentiellement sur les informations que l'analyste cherche à explorer. Certes, le contexte de l'alerte joue possiblement un peu moins un rôle critique dans le processus d'analyse. En effet, on s'intéresse davantage à la source qu'à la destination (on la connaît déjà, il s'agit du *honeypot*). Les patrons d'attaques recherchés seront donc nécessairement différents que ceux observés sur un réseau conventionnel. Le type de *honeypot* joue également un rôle important à ce niveau. S'il s'agit d'un système destiné à détecter des attaques sur un réseau d'une

entreprise, l'objectif consistera principalement à appuyer l'analyse effectuée par les IDS réseau, de façon à détecter plus facilement les intrus et à mieux comprendre leurs motifs. Par contre, un *honeypot* utilisé à des fins de recherche visera plutôt à découvrir des patrons et des tendances à la grandeur de la toile, dans le but, par exemple, de détecter et d'évaluer la propagation de vers. La plupart des outils d'analyse des alertes présentés jusqu'ici mettent l'emphase sur les machines ciblées ou sur la communication entre la source et la destination plutôt que sur la corrélation entre les éléments communs entre les différentes sources, par exemple. Il est donc plutôt difficile d'extraire l'information pertinente pour l'analyse d'un *honeypot* à l'aide des approches conventionnelles. Les approches existantes ne peuvent pas non plus supporter suffisamment de sources pour représenter le réseau Internet sur la visualisation.

Il existe très peu de solutions développées spécifiquement pour analyser les données générées par des *honeypots*. Parmi l'ensemble des techniques évaluées, seule l'approche de Koike, Ohno et Koizumi (2005) semble intéressante pour l'analyse de ces systèmes. Cette approche met en effet l'emphase sur la source de l'attaque, et permet de consulter sur une matrice la propagation d'une attaque à travers l'ensemble d'Internet (voir figure 1.8). Malheureusement, l'interface limite grandement la quantité d'information qu'il est possible d'afficher simultanément, rendant une analyse plus approfondie plutôt difficile. Cette approche tient également pour acquis que le ver est connu et qu'un IDS peut le détecter. Il est en effet impossible d'utiliser les informations présentées par cette solution pour détecter des tendances ou de nouvelles attaques inconnues jusqu'à présent.

2.3 Techniques de visualisation

Plusieurs approches liées à la visualisation en sécurité ont été présentées à la section 1.5. De façon générale, les approches existantes présentent des signes de faiblesse lorsque le nombre d'événements augmente considérablement. Malheureusement, aussi bien pour l'analyse réseau que pour l'analyse *d'honeypots*, le nombre d'informations à visualiser est pratiquement toujours très élevé. Il n'est donc pas rare de se retrouver avec des surcharges

d'information et de l'occlusion sur les représentations. De leur côté, les quelques approches supportant beaucoup de données y parviennent habituellement en sacrifiant des éléments très importants pour la compréhension de la situation.

Selon l'ensemble des articles évalués à la section 1.2, une variable semble particulièrement guider l'analyse : le temps. Connaître l'ordre des événements, et à quel moment ils se sont produits semble particulièrement critique pour clarifier un incident. Des alertes concernant, par exemple, un balayage de ports, la détection d'un balayage de vulnérabilités ou de trafic sur un port suspect peuvent être jugés peu critiques lorsqu'analysées individuellement. Toutefois, lorsqu'ils se produisent séquentiellement dans un court laps de temps, ces événements peuvent prendre un tout autre sens. Il doit donc être possible de mettre en évidence des séquences d'événements et de pouvoir naviguer facilement dans l'espace temporel. La plupart des approches visuelles négligent complètement cette variable, et regroupent l'ensemble des données sur une même échelle. La plupart des attaques sont donc invisibles pour l'analyste, qui doit se fier ultimement sur les journaux bruts pour les détecter.

Plusieurs approches visuelles se fient sur le degré de sévérité des attaques selon ce que rapporte l'IDS, ou encore sur la fréquence des alertes pour déterminer les éléments importants pour l'usager. Ces hypothèses négligent toutefois deux caractéristiques importantes des IDS. Tout d'abord, les IDS sont particulièrement réputées pour la quantité de faux positifs qu'ils génèrent. Lors de l'analyse du trafic sur un réseau, une partie significative des alertes rapportées sur le graphique risque donc de présenter de faux événements et ne devrait pas être considérée. Également, un IDS peut générer une quantité faramineuse d'alertes en doublons pour rapporter un même problème. Il est donc très clair que de mettre l'emphase uniquement sur la fréquence ne fait aucun sens. En ce qui concerne le degré de sévérité, il faut se rappeler que chaque signature d'un IDS prend nécessairement un sens différent selon le type de données qu'il faut protéger. Un événement critique peut être banal pour un autre organisme. Ce type de classement peut évidemment faciliter le tri, mais ne devrait pas être la seule variable en jeu pour déterminer la criticité des attaques.

2.4 Liste des caractéristiques recherchées

En résumé, voici les principales caractéristiques recherchées dans le cadre du projet :

1. L'outil doit pouvoir traiter une quantité importante de données sans être surchargé, de façon à mettre en évidence les événements importants sur le réseau et ainsi assister les analystes dans leur travail.
2. L'outil doit permettre d'analyser simultanément plusieurs milliers d'alertes, tout en offrant suffisamment de détails pour analyser une alerte en profondeur.
3. L'outil doit être orienté sur la façon de travailler des analystes, et être axé sur les bonnes caractéristiques selon le type de données à analyser.
4. L'outil doit permettre de gérer le bruit et les faux positifs produits par les IDS, et faciliter l'interaction entre les différents intervenants au cours du processus.
5. L'outil doit pouvoir supporter l'analyse sur des réseaux conventionnels ainsi que sur des *honeypots*.
6. L'outil doit permettre la découverte de patrons d'attaques sur les données globales.
7. L'outil doit pouvoir servir de base de connaissances pour les analyses ultérieures.

Jusqu'à présent, il semble que la solution n'existe pas pour répondre à l'ensemble de ces critères.

CHAPITRE 3

PRÉSENTATION DE LA SOLUTION

3.1 Principes fondamentaux

La solution proposée dans le cadre de ce projet se veut donc une réponse à l'ensemble des critiques énoncées au chapitre précédent. Inspiré de l'approche proposée par Foresti et al. (2006), AlertWheel propose une interface radiale permettant de visualiser les alertes générées par des IDS sur un réseau de grande envergure. Contrairement aux approches présentées au chapitre 1, la solution s'adapte aussi bien aux réseaux commerciaux qu'aux *honeypots*.

Figure 3.1 Interface principale d'AlertWheel

AlertWheel repose en grande partie sur le processus d'analyse discuté à la section 1.2. La solution se veut un outil de premier plan pour les phases de surveillance et d'analyse du processus, basé sur les besoins réels des analystes. Pour répondre à ces besoins, l'application implémente le « Visual Information-Seeking Mantra » proposé par Shneiderman (1996). Ce principe définit trois étapes lors de l'analyse de l'information visuelle : *Overview first, zoom and filter, details-on-demand.* Lorsque l'utilisateur démarre AlertWheel, il obtient initialement une vue à haut niveau qui présente globalement l'état du réseau (voir figure 3.1). L'utilisateur peut ensuite interagir avec l'interface, par exemple en appliquant différents filtres afin de réduire la quantité d'information sur la représentation et cibler précisément certains événements jugés anormaux. Finalement, l'interface propose deux niveaux de visualisation supplémentaires permettant d'obtenir davantage de détails sur une sélection d'alertes à examiner. Nous discuterons plus en détail de chacune de ces interfaces dans les sections suivantes de ce chapitre.

L'interface principale d'AlertWheel intègre également le concept W³ proposé par Foresti et al. (2006). W³ fait référence aux trois attributs majeurs pour représenter une alerte : *When, Where* et *What*.

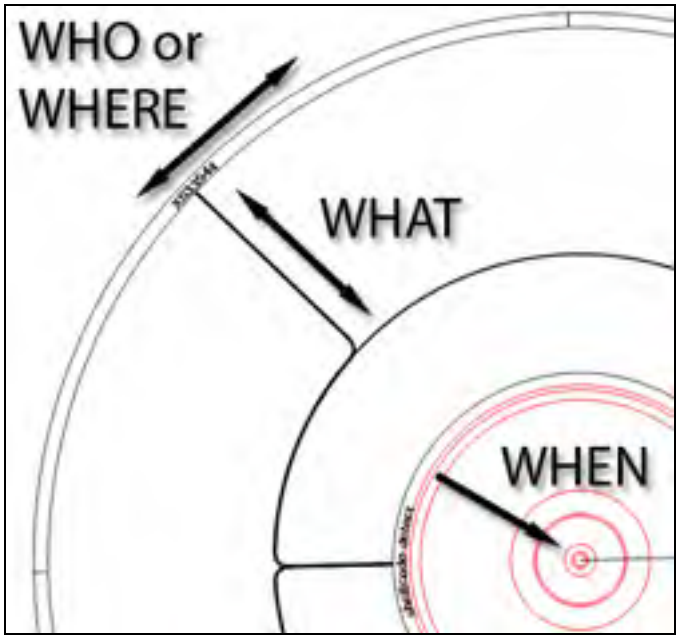

Figure 3.2 Concept étendu W^4 tel que proposé par AlertWheel

AlertWheel étend ce concept en ajoutant *Who*, dans la mesure où la source de l'attaque est primordiale pour analyser les alertes sur des *honeypots*. La figure 3.2 illustre le concept W^4 tel que proposé sur la solution. On retrouve tout d'abord les concepts *Who* et *Where* sur l'anneau extérieur. L'application permet en effet d'afficher au choix la source (*Who*) ou la destination (*Where*) de l'alerte. La courbe reliant l'anneau extérieur au cercle intérieur représente une ou plusieurs alertes (*What*), et chaque arc de couleur à l'intérieur du cercle intérieur représente le moment de la détection (*When*). Chacun de ces éléments sera présenté un peu plus en détail dans la section 3.3.

3.2 Environnement de l'application

L'application AlertWheel a été développée en C# 4.0 sous Visual Studio 2010. Les données utilisées par l'application sont conservées dans une base de données Microsoft SQL Compact Database, bien qu'elles puissent être déplacées facilement dans un autre SGBD au besoin. L'application supporte le format d'alertes *Unified2* proposé par l'IDS Snort (The Snort Project (2011, p. 138)), et peut importer directement l'ensemble des configurations de cet IDS. L'architecture d'AlertWheel permettrait toutefois de supporter facilement d'autres solutions en implémentant simplement les adaptateurs appropriés.

Les données d'évaluation proviennent entièrement des différents *honeypots* du réseau international Wombat (The Wombat Project (2011)). Ces données prennent la forme de traces réseau observées par les différents serveurs répartis un peu partout sur la planète, lesquelles sont ensuite analysées avec Snort pour obtenir des journaux d'alertes. Les adresses IP sources (attaquants) ont également été mises en relation avec leur système autonome (AS) grâce au module GeoIP de l'entreprise MaxMind (MaxMind Inc. (2011)). Vu la nature sensible des données, il n'a malheureusement pas été possible d'obtenir de traces provenant d'un réseau commercial ou institutionnel. Les évaluations ont donc uniquement été effectuées à l'aide des données provenant des *honeypots*. Il est également important de noter que les données ont été anonymisées à l'aide d'une technique de hachage pour les différentes captures de ce document afin de préserver la confidentialité des données.

3.3 Vue principale – Roue d'alertes

La figure 3.1 présente l'interface principale de premier niveau de l'application. Celle-ci se décompose essentiellement en trois sections : les roues d'alertes (visualisations principales), les filtres et les scénarios.

Chaque roue est essentiellement composée d'un anneau et d'un cercle superposés, reliés par une série de courbes disposées de façon particulière.

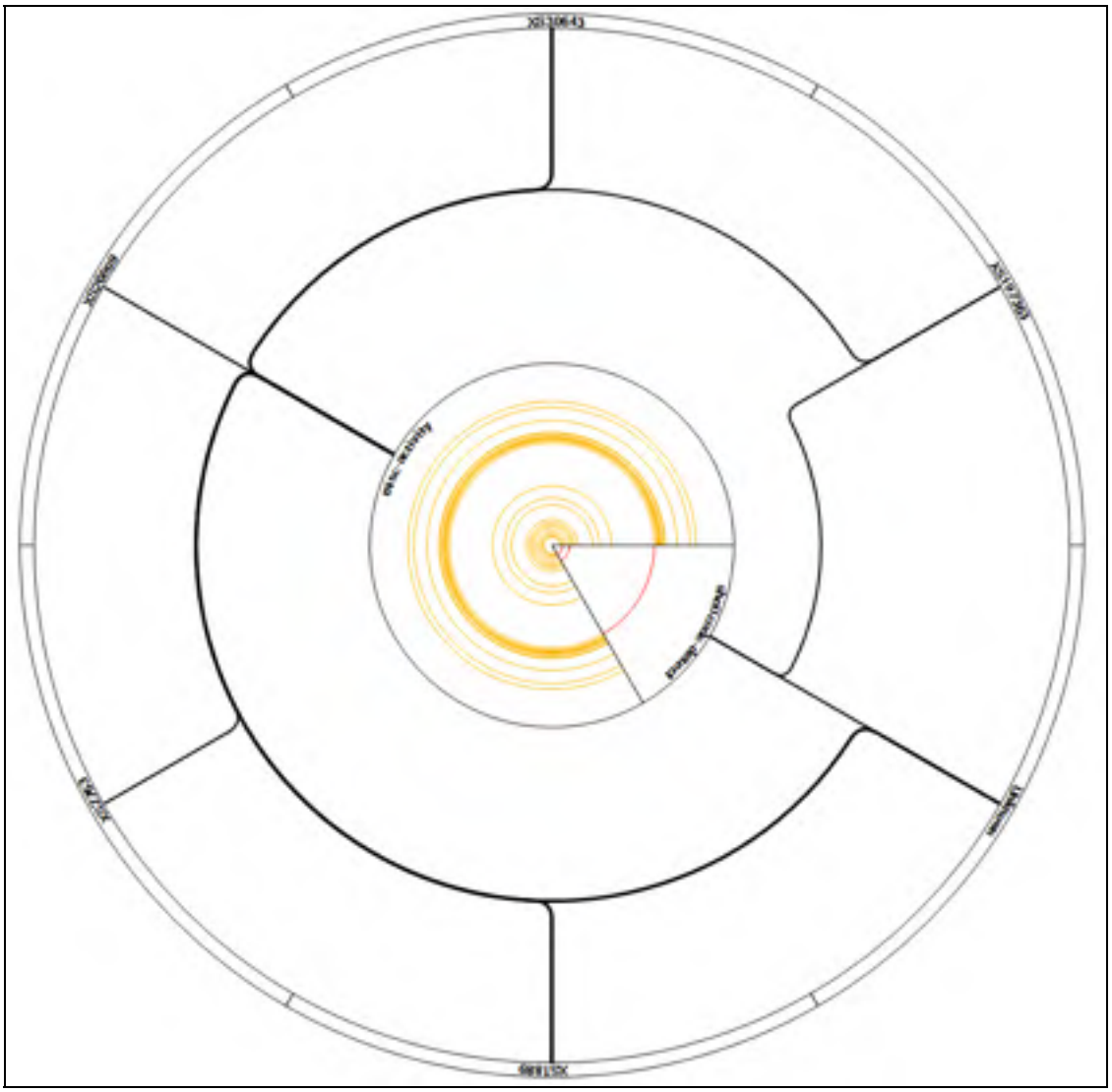

Figure 3.3 Roue d'alertes

3.3.1 Anneau extérieur (source / destination)

L'anneau extérieur permet d'afficher les informations concernant la source ou la destination de l'attaque. L'utilisateur peut en effet sélectionner le type de données qu'il désire afficher sur l'anneau extérieur à l'aide d'un menu contextuel, tel qu'illustré par la figure 3.4.

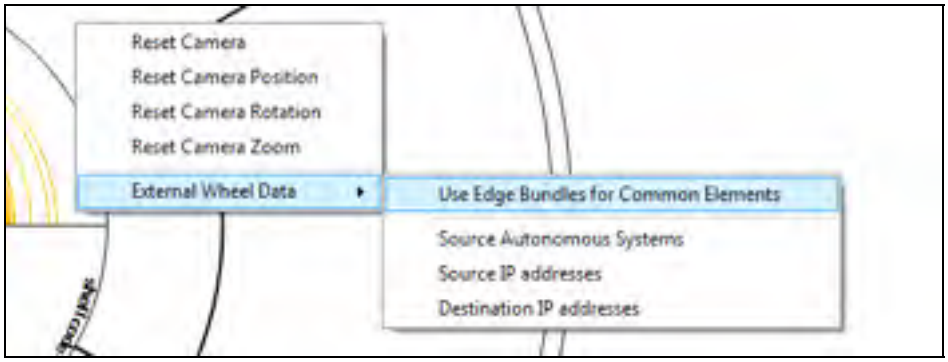

Figure 3.4 Menu contextuel permettant de sélectionner les données affichées sur l'anneau extérieur

Il est tout d'abord important de préciser les concepts de source et de destination. On entend ici par la source l'attaquant ayant déclenché l'alerte. La destination sera donc la victime à l'intérieur du réseau interne de l'organisme dans le cas d'une analyse réseau, ou encore l'adresse du *honeypot* où l'alerte a été déclenchée. Cette distinction est particulièrement importante, dans la mesure où un IDS peut inverser les deux concepts selon la direction du paquet intercepté. En effet, il est possible qu'une alerte soit déclenchée lorsque l'attaquant envoie du contenu malveillant, mais également lorsque la victime répond, par exemple, avec une invite de commande. Dans ce cas, l'adresse source serait celle de la victime, et la destination serait l'adresse de l'attaquant. AlertWheel peut toutefois régler cette situation en considérant simplement les adresses IP internes comme la destination de l'attaque. Malheureusement, cette distinction est impossible si la source et la destination de l'attaque se trouvent à l'interne du réseau (par exemple, dans le cas d'un ver qui se propage à l'interne à partir d'une autre machine du réseau). Dans ce cas, aucune modification ne sera apportée aux adresses, et la communication complète sera nécessairement divisée entre les deux adresses

sur la représentation (on retrouvera les deux adresses à la fois comme source et destination). Il s'agit de l'une des principales limites de la solution.

Par défaut, l'application affiche l'adresse IP source sur le cercle extérieur, dans la mesure où il s'agit de l'information la plus pertinente pour l'analyse initiale des données provenant d'un *honeypot*. Cette adresse n'est malheureusement pas toujours valide. Il est en effet possible que l'attaquant masque son identité en usurpant son adresse IP (*IP spoofing*) ou en utilisant une passerelle réseau (ex. : proxy, machine d'une autre victime, etc.). Dans le premier cas, l'utilisateur falsifie son adresse IP dans tous les paquets réseau qu'il envoie à sa victime. Cette technique est toutefois très peu utilisée, dans la mesure où la victime ne peut retourner aucune information à la source. Ce type d'attaque est habituellement utilisé lorsqu'un attaquant désire infecter une machine sans établir de communication avec sa victime (ex. : l'attaque du ver Slammer (Moore et al. (2003)). Dans la plupart des cas, l'attaquant dissimulera plutôt son identité à l'aide d'une passerelle quelconque. Bien que cette technique empêche d'identifier l'attaquant directement, il sera malgré tout possible de suivre plus facilement la séquence de l'attaque en rattachant l'ensemble des étapes à une même adresse. Dans le cas de l'analyse des *honeypots*, connaître la source permet également d'évaluer la propagation d'un ver à travers le monde, par exemple.

L'application permet d'afficher la source de deux façons : selon son adresse IP ou son système autonome (AS). Un système autonome est un regroupement de réseaux IP sous le contrôle d'une autorité unique, par exemple un fournisseur d'accès à Internet. Connaître le système autonome permet d'évaluer l'emplacement géographique (le pays, entre autres) de la source de l'attaque. Afficher les sources en fonction du système autonome permet de regrouper efficacement les adresses IP et d'ainsi réduire significativement l'espace nécessaire pour afficher l'ensemble des sources. La figure 3.5 illustre l'intérêt de cette réduction. Ce type de regroupement permet de faciliter une première analyse des données en réduisant le nombre de courbes présentées sur le graphique. Une fois l'analyse préliminaire effectuée, l'analyste peut appliquer certains filtres pour retirer les AS jugés non pertinents. Une fois le

nombre de données réduites, il peut alors changer l'affichage par adresse, et ainsi obtenir une vue plus détaillée sur les données.

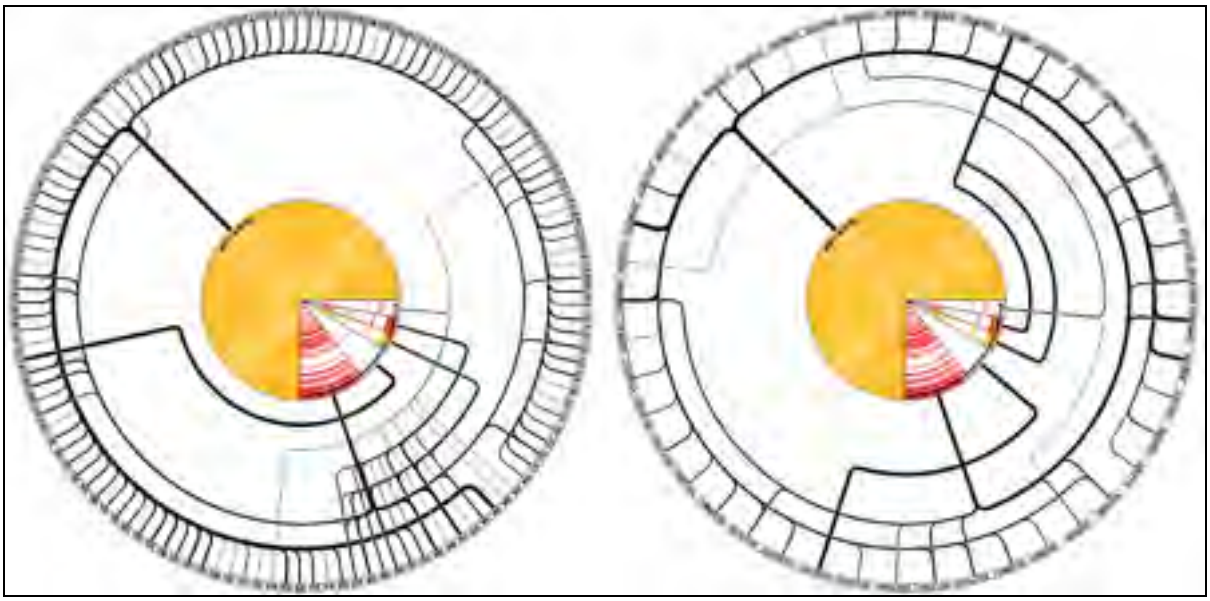

Figure 3.5 Regroupement des adresses sources par IP (gauche) et par AS (droite)

L'application permet de réduire encore davantage le nombre de courbes en tirant profit de la compression des liens, tel que discuté à la section 1.6. Rappelons que la compression des liens consiste à regrouper l'ensemble des adresses IP ou AS dont tous les liens sont identiques. On remarque par exemple sur la figure 3.5 droite que 11 systèmes autonomes ont à la fois des alertes de type « Shellcode detect » et « misc-activity ». On remarque également 27 AS ne possédant qu'un seul lien vers « misc-activity ». Il est donc possible d'épargner 36 courbes sur le graphique en regroupant uniquement ces AS en deux classes, tel qu'illustré sur la figure 3.6. Cette approche permet de rapidement identifier des patrons d'attaques sur un ensemble de 2629 alertes sans avoir à en retirer une seule de la représentation (aucun filtrage), et sans aucun problème d'occlusion ou de surcharge sur le graphique.

Il est également possible d'afficher les adresses de destination (les victimes) au lieu des adresses IP source (attaquants) sur le cercle extérieur. Pour l'analyse des *honeypots*, cette technique permet de déterminer les adresses des systèmes où ont été enregistrées les alertes, dans le cas où il y aurait plusieurs adresses IP actives sur le système, ou encore sur quel

honeypot les alertes ont été enregistrées. Pour l'analyse des réseaux, cette technique permettrait d'identifier les systèmes touchés sur le réseau de l'organisation.

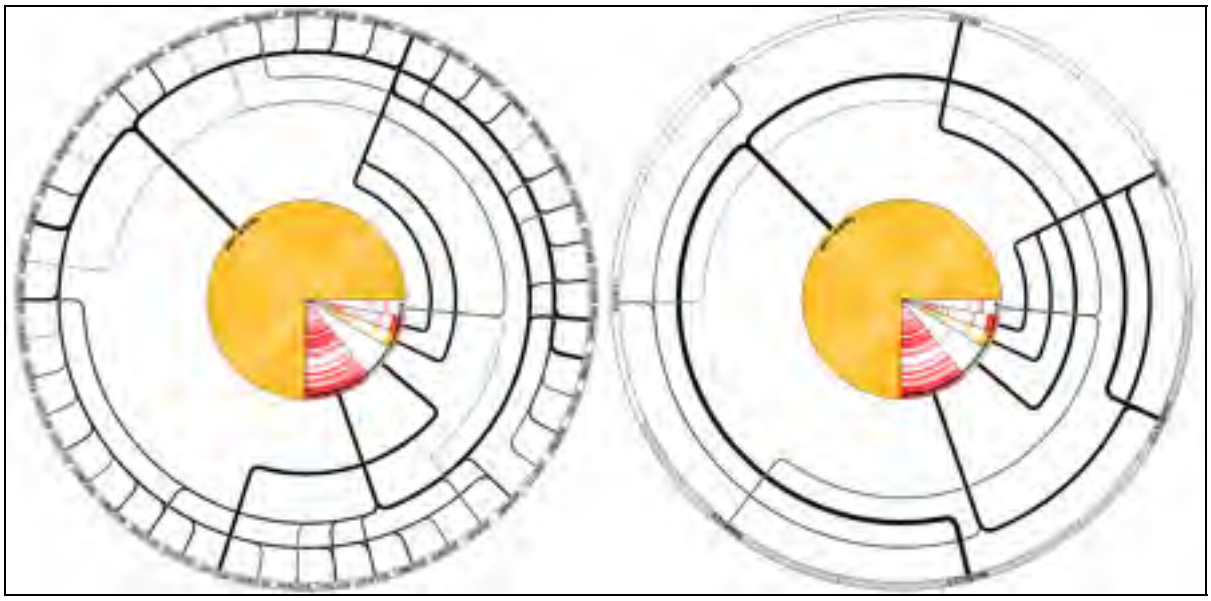

Figure 3.6 Sources affichées par AS (gauche) et regroupées par compression des liens (droite)

Finalement, lorsque l'utilisateur positionne son curseur sur une division de l'anneau, par exemple sur un AS, il obtient des informations complémentaires, telles que le nom complet de l'AS, le nombre d'alertes associées, la plage d'adresses IP de l'AS, etc.

Les adresses IP et les AS présentés sur le cercle extérieur sont triés par ordre alphabétique. Cette technique facilite grandement la recherche des éléments sur le cercle, surtout lorsque l'espace est insuffisant pour afficher les étiquettes complètes et que seul le préfixe des adresses est affiché, par exemple. Il est très fréquent, en effet, que l'on désire retrouver un AS ou une adresse sur le cercle extérieur pour observer en surbrillance ses associations. Il est toutefois évident que cette technique n'offre pas la disposition optimale des secteurs pour réduire la longueur des courbes. Il est possible qu'un AS ou une adresse se trouve d'un côté du cercle à cause de sa position dans l'ordre alphabétique et que la classification associée se trouve complètement à l'opposé, forçant des courbes très longues inutilement. Toutefois, vu

la nature des données, il a été convenu que l'ordre alphabétique était préférable dans cette situation, quitte à allonger quelques courbes pour les besoins de la cause.

3.3.2 Cercle intérieur (classifications)

On retrouve ensuite au centre du graphique un cercle plein divisé en pointes de tartes (secteurs). Chaque secteur représente une classification d'alertes. Une classification est un regroupement de signatures d'un IDS. Par exemple, la version 2.9.0 de Snort proposait 34 classifications regroupant 9403 signatures distinctes (voir annexe II pour la liste des classifications). Vu la quantité de signatures, il est impossible de présenter l'ensemble sur le cercle extérieur. Le nombre de classifications est évidemment beaucoup plus raisonnable. Par exemple, avec les ensembles de données évalués, seules 6 classifications d'alertes ont été observées. Le nombre de classifications dépend évidemment des signatures utilisées, de la sensibilité de l'IDS, du type de données traitées, du nombre d'attaques observées, de leur complexité, etc.

Les secteurs sont divisés au prorata du nombre d'alertes qu'ils contiennent. Par exemple, sur la figure 3.7, on observe que *misc-activity* occupe une très grande partie des alertes, avec 2045 des 2629 alertes. Cette répartition de l'espace est toutefois ajustée de façon à ce que les secteurs comprenant un nombre très faible d'alertes soient malgré tout visibles sur le graphique. Par exemple, la classification *attempted-admin*, avec seulement 7 alertes, n'occuperait environ que 0.26% de l'espace. Pour éviter ce problème, une dimension minimale est tout d'abord attribuée et l'espace restant est ensuite distribué aux autres secteurs. Cette technique offre à l'analyste une première impression de la situation sur le réseau, bien que la fréquence ne permette pas à elle seule de déterminer le degré de criticité d'une attaque. Malgré tout, une fréquence élevée d'alertes critiques pourrait soulever des doutes quant à la situation sur le réseau.

Les secteurs sont également triés en fonction du nombre d'alertes qu'ils contiennent pour faciliter la comparaison. Comparer des secteurs sur un cercle s'avère en effet plutôt

complexe. Il est plutôt difficile de déterminer si une pointe couvre une plus grande surface qu'une autre si les deux pointes ne se trouvent pas à proximité. En triant par fréquence, cette comparaison n'est donc plus nécessaire.

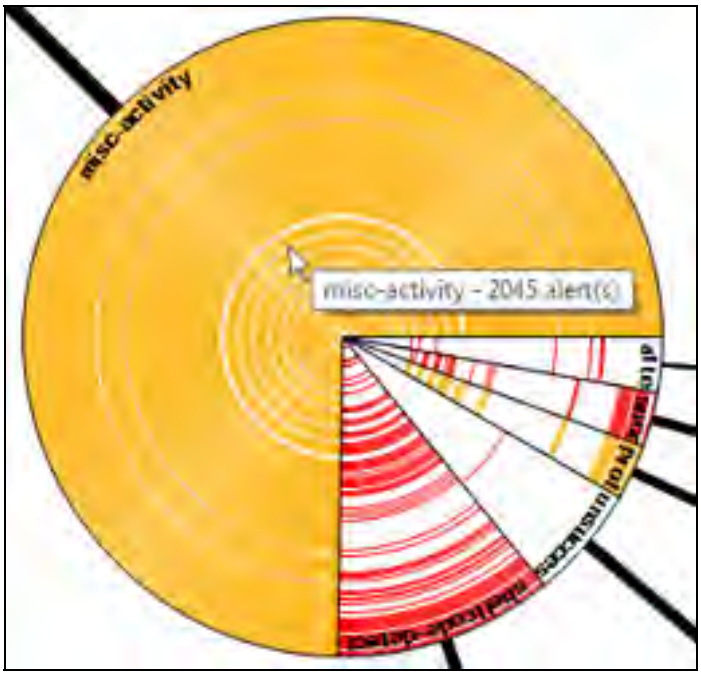

Figure 3.7 Cercle intérieur

Le cercle intérieur permet également de représenter la séquence des événements telle qu'ils ont été détectés par l'IDS. On retrouve en effet une série d'arcs de couleurs représentant le moment où chaque alerte a été détectée. Les alertes les plus récentes sont représentées par un arc situé à l'extrémité du cercle, alors que les alertes les plus anciennes se trouvent près du centre du cercle. Par exemple, sur la figure 3.8, la période d'analyse varie entre 05:10 et 6:10. Le cercle a été tourné de 180 degrés et les heures ont été ajoutées manuellement pour illustrer le propos.

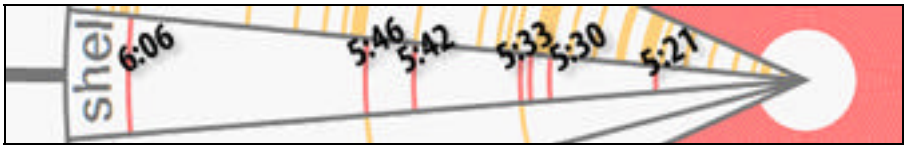

Figure 3.8 Alertes représentées sur un secteur du cercle intérieur.

La durée associée à chaque arc est adaptée de façon dynamique en fonction de la période d'analyse. Cette dernière est déterminée par l'analyste à l'aide d'un filtre sur les données. Le rayon représente donc une échelle de temps variable couvrant automatiquement la période choisie. Autrement dit, chaque arc peut représenter quelques secondes, quelques minutes, voire plusieurs heures ou journées selon la période d'analyse choisie. Il est donc fréquent qu'un arc représente plus d'une alerte. Dans certains cas, il serait intéressant de mettre en évidence les alertes en fonction de la fréquence à laquelle elles ont été enregistrées durant une période donnée. Une option permet donc à l'analyste d'utiliser l'opacité des arcs pour représenter la fréquence des alertes, tel qu'illustré sur la figure 3.9. Dans cet exemple, on remarque que les alertes *unsuccessful-user* ont été enregistrées avec une fréquence très importante au début de la période d'analyse (apparaissent avec un rouge très intense) comparativement aux autres alertes enregistrées.

La couleur des arcs représente le degré de sévérité rapporté par l'IDS (jaune pour faible, orange pour moyen, rouge pour critique). Dans la mesure où le degré de sévérité est associé à une classification (voir annexe II), la couleur demeure constante sur un secteur donné. Les versions initiales de l'application utilisaient la plage plus conventionnelle vert-jaune-rouge pour représenter la sévérité. Certaines critiques ont toutefois été adressées quant à l'utilisation de la couleur verte, habituellement associée à quelque chose de positif dans la culture populaire. Jaune-orange-rouge est fréquemment utilisé pour représenter des degrés de sévérité, jaune étant naturellement associé à un avertissement dans la culture populaire, et rouge à un événement critique. Malheureusement, il est possible que certaines personnes démontrent des difficultés à différencier ces trois tintes plutôt rapprochées. Une autre option consisterait à utiliser une couleur neutre pour les alertes mineures. Par exemple, la plage gris-jaune-rouge serait possiblement plus facile à distinguer, mais serait possiblement moins évidente de prime à bord que jaune-orange-rouge. Ultimement, le choix de couleur pourrait être laissé à la discrétion de l'utilisateur.

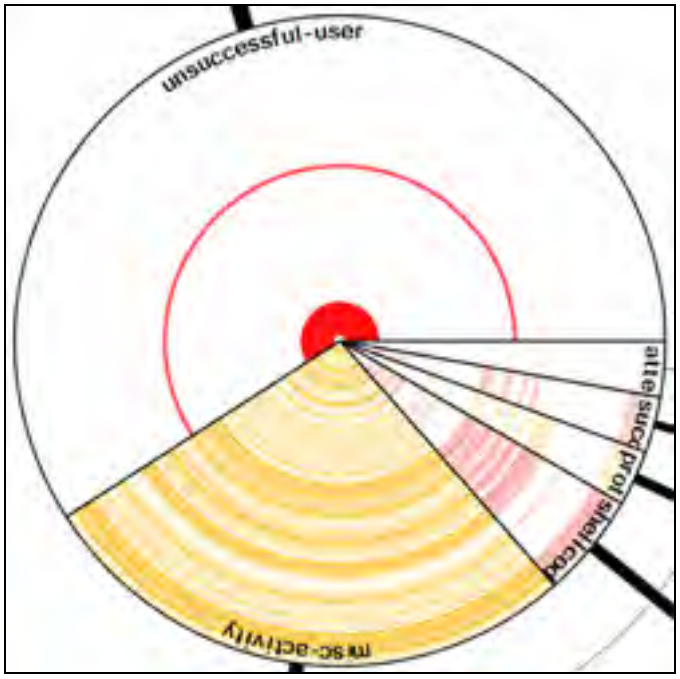

Figure 3.9 Utilisation de l'opacité des arcs
pour représenter la fréquence

Dans certains cas, baser l'analyse uniquement sur les classifications n'est pas suffisant. L'application permet alors également d'exploser une classification afin de diviser un secteur en fonction des signatures actives qu'il regroupe. Afin de faciliter la lecture, les secteurs couvrant moins de la moitié du cercle sont également ajustés de façon à offrir un minimalement 180 degrés, peu importe le nombre d'alertes que la classification peut contenir. Dans ce cas, chaque sous-secteur contient ses propres arcs afin d'illustrer la séquence des événements pour chaque signature distincte.

La version actuelle de l'application n'affiche qu'un seul lien vers le secteur, peu importe qu'il soit explosé ou non. Par exemple, sur la figure 3.10, on remarque un seul point d'entrée, bien qu'il y ait quatre sous-secteurs. Cette technique permet de réduire le nombre de liens sur la représentation, surtout lorsque le nombre de signatures d'une classification devient important.

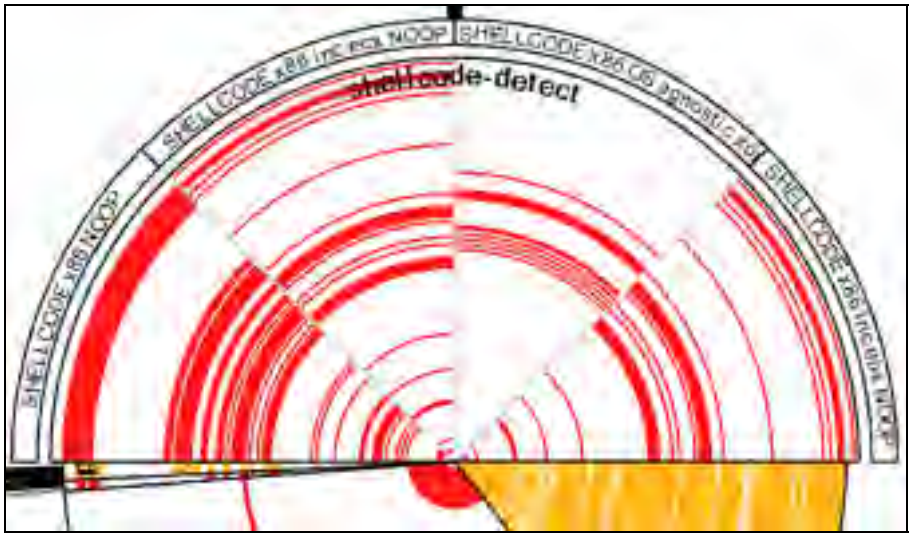

Figure 3.10 L'explosion de la classification *shellcode-detect* présente quatre signatures distinctes

3.3.3 Courbes (alertes)

Les courbes reliant l'anneau extérieur au cercle intérieur représentent sans contredit l'information la plus importante sur la visualisation. Elles permettent de représenter les relations entre les systèmes (source / destination) et des attaques précises. Sans ces liens, il est impossible de déterminer qui a fait quoi.

Les courbes sont dessinées d'une façon particulière, de manière à éviter toute forme d'occlusion causée par des superpositions sur la représentation. La figure 3.11 illustre le concept des courbes tel que présenté par la solution, appliqué à l'approche radiale de la solution. La représentation 1 utilise la représentation traditionnelle, alors que la deuxième illustre le même ensemble de données simplifié avec la technique des rayons réservés proposée par AlertWheel.

L'approche proposée consiste à réserver un rayon (ou une droite dans l'approche linéaire) pour chaque élément du cercle intérieur (éléments situés à droite dans l'approche linéaire). Chaque nœud est ensuite relié par une droite orthogonale aux rayons / droites associés. De façon à éviter toute forme d'ambiguïté, une courbe est tracée pour représenter chaque liaison.

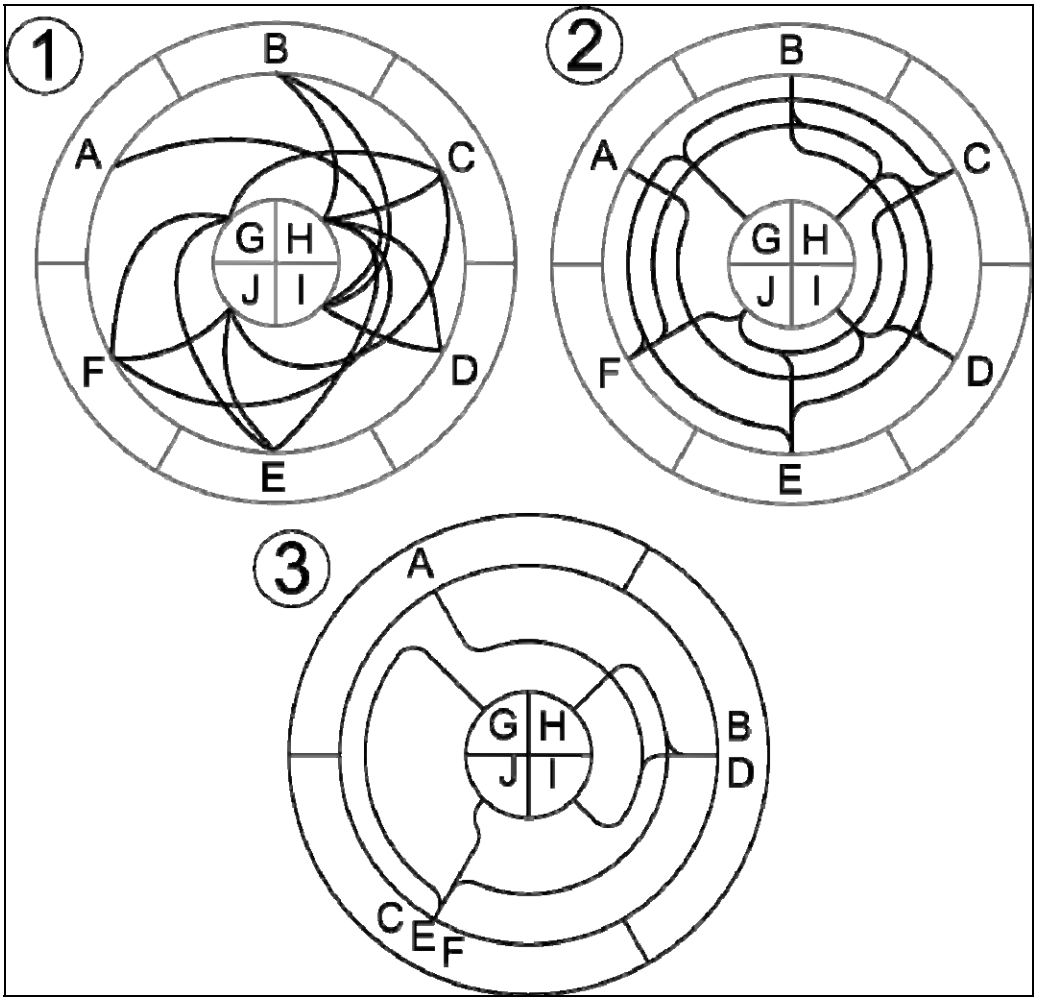

Figure 3.11 Représentations des courbes sur une visualisation radiale pour un même ensemble de données.

Quatre rayons (représentés par des cercles) ont été réservés entre l'anneau extérieur et le cercle intérieur pour les nœuds intérieurs G, H, I et J. On remarque que le nœud B, par exemple, est rattaché à H et à I. Par contre, A n'est pas rattaché à G puisqu'il n'y a pas de courbe entre la droite perpendiculaire de A et le cercle réservé pour G.

Cette approche permet d'éviter toute forme d'ambiguïté ou de superposition sur le graphique. Par exemple, sur la représentation #1, on remarque que les courbes se croisent et se superposent à plusieurs endroits, rendant plus difficile le suivi des courbes. Lorsque le

nombre de courbes augmente significativement, il devient pratiquement impossible de suivre une courbe du début à la fin. À l'inverse, les représentations 2 peut représenter facilement un graphe biparti complet (tous les nœuds d'un ensemble reliés à tous les nœuds de l'autre ensemble) sans superposition ou confusion, à condition qu'un espace suffisant soit assuré entre les cercles réservés (environ l'équivalent du rayon des courbes utilisées pour relier les droites au cercle). La représentation 3 illustre l'application du concept de compression des liens énoncé précédemment.

L'épaisseur des courbes offre une indication du nombre d'alertes représentées par ce lien. L'épaisseur est calculée en fonction du logarithme du nombre d'alertes représentées par le lien. Cette technique permet d'éviter de se retrouver avec des traits trop épais lorsque le nombre augmente significativement, ce qui pourrait causer de l'occlusion si l'épaisseur est suffisante pour dissimuler les courbes voisines. Le logarithme croit malgré tout suffisamment rapidement pour dissocier les liens avec une seules alertes des liens avec quelques alertes.

Bachmaier (2007) démontre clairement que l'ordre d'assignation des cercles intérieurs peut avoir une importance significative. Par exemple, considérons la figure 3.12. La longueur totale des parties radiales dépend du nombre de sources e_i devant être connectées. L'objectif consiste alors à minimiser la somme totale de la longueur des liens. Il existe deux approches : une approche heuristique et une approche optimale. La première assigne les cercles intérieurs à partir du centre selon la fréquence des alertes. La catégorie la moins fréquente se retrouve ainsi associée au cercle ayant le plus petit rayon. La catégorie la plus fréquente se retrouve associée au cercle ayant le plus grand rayon. Cet ordonnancement reflète donc l'ordonnancement des secteurs du cercle intérieur, facilitant ainsi la lecture.

La deuxième approche assigne les cercles intérieurs à partir du centre selon le nombre de sources e_i devant être connectées. Cette approche est différente de l'heuristique précédente. Une source peut générer en principe un très grand nombre d'alertes d'un type donné. De ce fait, la fréquence de cette alerte pourrait être très grande tout en étant connectée à une seule

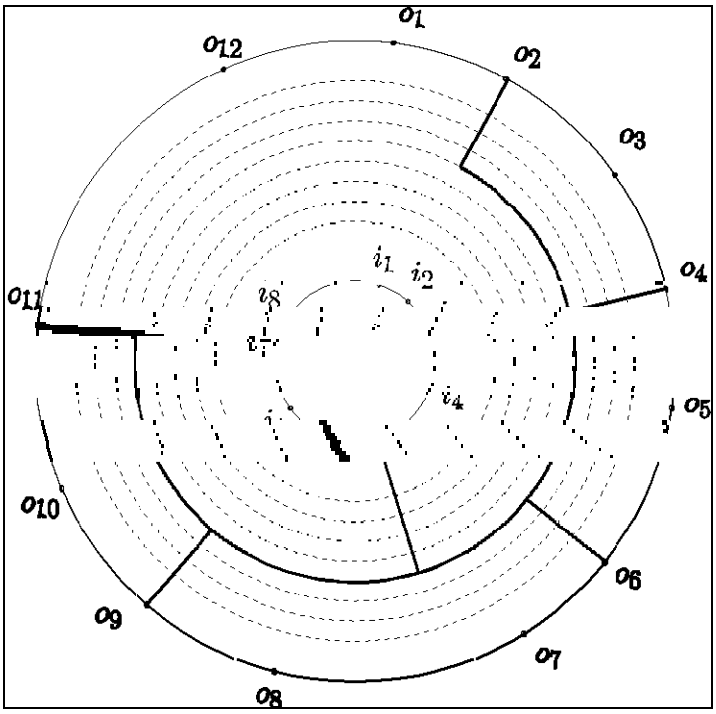

Figure 3.12 Cas simple où un nœud intérieur est relié
à plusieurs nœuds extérieurs

source. Cette deuxième approche donne une solution optimale minimisant la somme totale de la longueur des parties radiales des liens, tel que démontré par le lemme suivant :

Lemme 1 . *Supposons que le nombre de courbes* e_i *soit établi de façon à ce que* $a_1 \leq a_2 \leq \cdots \leq a_m$. *Si les rayons R sont établis de façon à ce que* $r_1 \leq r_2 \leq \cdots \leq r_m$, *alors* $S^R = \sum_{k=1}^{m} l_k^R$ *est minimal, où l représente la longueur de la courbe.*

Preuve. Supposons qu'il existe une meilleure disposition offrant une solution optimale $S' < S^R$. Sans perte de généralité, supposons que $a_k < a_{k+1}$ mais que $r'_k > r'_{k+1}$. Alors,

$$a_k(r_{k+1} - r'_k) + (r'_k - r_k) + a_{k+1}(r_{k+1} - r'_{k+1}) + (r'_{k+1} - r_k) > \qquad (3.1)$$

$$a_k(r_{k+1} - r'_{k+1}) + (r'_{k+1} - r_k) + a_{k+1}(r_{k+1} - r'_k) + (r'_k - r_k)$$

Ainsi, une meilleure solution S' peut être trouvée en permutant les points associés aux cercles avec des rayons r_k et r_{k+1}. Ceci contredit donc l'optimalité de S^*.

Q.E.D.

Pour que la solution soit optimale, il faudrait que les courbes avec le plus de sources (le plus d'embranchements) soient situées près du cercle extérieur pour réduire leur longueur pour atteindre le rayon réservé, plutôt que de considérer le nombre total d'alertes par catégorie. Malgré tout, l'approche heuristique basée sur les fréquences a été choisie afin de faciliter la lecture. Toutefois, il est bon de remarquer que cette heuristique peut donner des résultats proches de la solution optimale. En effet, dans de nombreux cas, les classifications avec le plus grand nombre d'alertes auront le plus grand nombre de sources. Il ne s'agit toutefois pas d'une règle, et il arrive parfois qu'une classification possède un très grand nombre d'alertes provenant de seulement une ou deux adresses IP source. Au final, le choix final entre les deux approches pourrait être laissé à la discrétion de l'utilisateur sous la forme d'une option de l'application, les deux approches étant justifiées et pratiques dans certains cas particuliers.

Dans la mesure où l'objectif de la représentation consiste à réduire le bruit inutile sur la représentation, les cercles réservés sont dessinés seulement lorsqu'ils sont utilisés, et uniquement entre les nœuds à rattacher. Les cercles prennent donc habituellement plutôt la forme de deux arcs situés des deux côtés du nœud concerné sur le cercle intérieur. Cette approche permet de relier deux points avec un arc d'un maximum de 180 degrés. En effet, l'orientation de l'arc sera déterminée de façon à minimiser sa longueur, facilitant ainsi la lecture (voir figure 3.12).

De façon à faciliter encore davantage la lecture et pour réduire encore davantage les risques d'ambiguïté, des traits blancs sont dessinés en bordure des droites. Cette technique permet de distinguer plus facilement les droites reliées à des arcs de celles les chevauchant. La figure 3.13 illustre ce concept. On remarque facilement sur cette figure que le premier et le troisième cercle ne sont pas reliés à l'AS XS40066, étant coupés par deux bordures blanches. Sans ces bordures, le deuxième lien porterait possiblement à confusion. La relation est

également mise en évidence par des courbes reliant la droite à l'arc. Le fait d'utiliser une courbe permet d'élargir la connexion et d'ainsi faciliter la lecture.

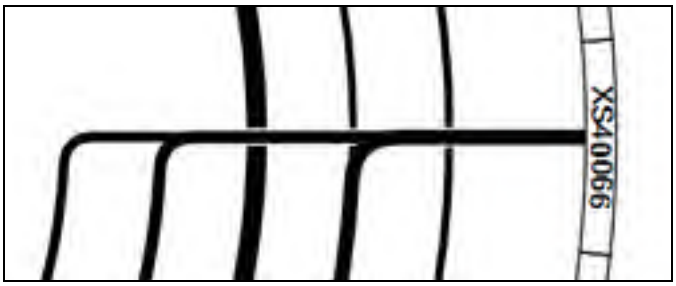

Figure 3.13 Utilisation de traits blancs autour de la droite pour délimiter les croisements

3.4 Vue intermédiaire – Matrice des données

Bien que la roue offre davantage d'informations que les approches similaires présentées au chapitre 1, le niveau de détails n'est pas suffisant pour effectuer une analyse complète. AlertWheel propose alors une vue secondaire complémentaire, la matrice de données (Matrix View), permettant de naviguer à travers chaque alerte individuellement sur une grille aménagée spécialement pour cette tâche. La figure 3.14 illustre la vue proposée par l'application.

La matrice propose six colonnes présentant les principales informations des alertes : l'adresse IP source, la signature, le temps, l'état, la sévérité et la destination. On entend par état de l'alerte le fait qu'elle soit associée à un scénario ou non. Ce concept sera présenté un peu plus loin à la section 3.7. L'utilisateur peut trier les valeurs du tableau en cliquant sur l'en-tête de la colonne désirée. Initialement, les valeurs sont triées en fonction du temps, de façon à faire ressortir les séquences d'événements.

Les colonnes *Source IP* et *Destination IP* proposent une nouvelle variation sur le thème des courbes présenté à la section 3.3.3. Chaque valeur de l'une des deux colonnes est positionnée immédiatement au croisement de la première valeur et de la droite réservée (précédemment « rayon réservé » pour le cercle) pour cette valeur. Cette approche permet de

simplifier la lecture et réduit la longueur des droites (dans le cas où toutes les étiquettes étaient alignées à l'autre extrémité de la colonne). Les droites réservées sont distribuées de droite à gauche (source) ou de gauche à droite (destination), de façon à éviter toute superposition avec les étiquettes. Le fait de placer les étiquettes à côté de la première valeur permet également d'éviter que d'autres liens soient superposés par-dessus celles-ci. La distance entre les rayons dépend du nombre de valeurs à afficher. Cette distance est recalculée à chaque fois que l'utilisateur défile dans la vue ou que les données sont rafraîchies, de façon à conserver la même structure sans occlusion.

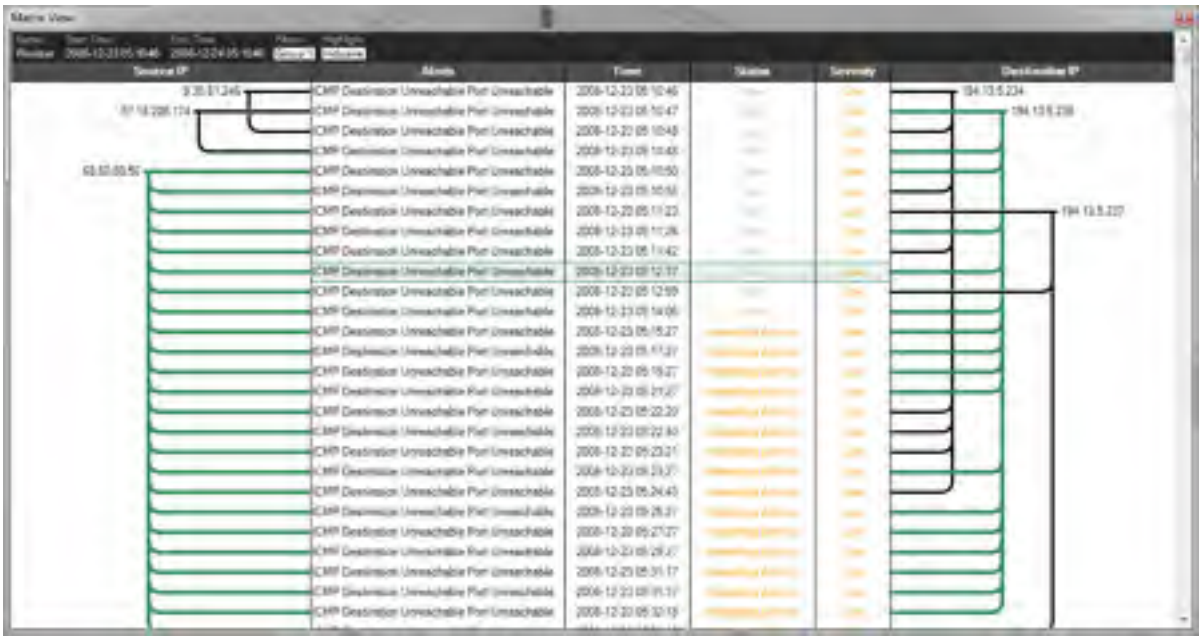

Figure 3.14 Matrice des données

Lorsque l'utilisateur sélectionne une alerte de la matrice, l'ensemble des liens rattachés à la même source ou à la même destination sont placés en évidence (les liens sont dessinés en vert). Cette technique permet d'étudier plus facilement des séquences d'événements ou des patrons d'attaques.

Dans le même ordre d'idée que sur la figure 3.13, chaque ligne horizontale est également délimitée par un trait blanc et une courbe est utilisée pour représenter chaque connexion, tel

qu'illustré sur la figure ci-dessous. Cette technique permet, une fois de plus, de faciliter la lecture et de dissocier facilement les croisements des connexions.

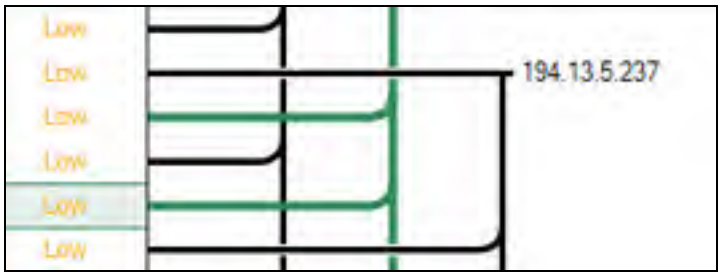

Figure 3.15 Utilisation de traits blancs autour de la droite pour délimiter les croisements

3.5 Vue détails

3.5.1 Propriétés d'une alerte

L'application propose enfin une troisième vue permettant d'accéder aux données brutes liées à chaque alerte individuellement. Cette interface est accessible en double-cliquant sur n'importe quelle ligne de la matrice. Elle s'avère essentielle lors de l'étape d'analyse du processus, dans la mesure où elle permet à l'analyste de mieux comprendre les particularités d'une attaque. Cette interface est présentée ici comme base de référence. L'interface finale devrait être développée en collaboration avec les analystes, afin d'obtenir une meilleure compréhension de leurs besoins à ce niveau. Toutefois, ce travail dépasse le cadre du projet dans sa définition actuelle. Malgré tout, l'interface proposée est suffisante pour permettre aux analystes d'effectuer leur travail, et n'aurait qu'à être raffinée en fonction des commentaires obtenus lors de la mise en production.

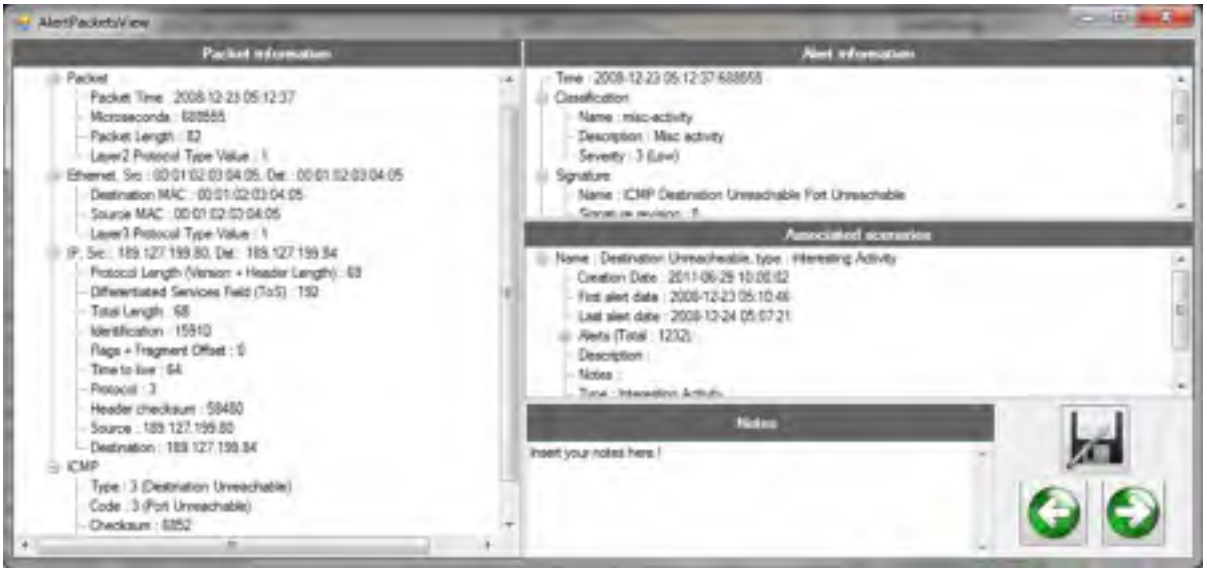

Figure 3.16 Vue détaillée d'une alerte

3.5.2 Affichage du paquet

En plus des différentes données contenues dans l'alerte générée par l'IDS, la vue détaillée présente toutes les informations récupérées sur le paquet ayant déclenché l'alerte. Ces informations sont décomposées par couche du modèle OSI. Par exemple, la figure 3.15 illustre les différentes valeurs des champs d'un paquet ICMP et de toutes les couches inférieures. Ces informations permettent à l'analyste d'obtenir des informations brutes sur les éléments de la communication et d'ainsi, possiblement, obtenir une meilleure compréhension d'une alerte précise ou de confirmer qu'il s'agit d'un faux positif, par exemple.

3.5.3 Autres informations

Chaque alerte peut également être associée à un ou plusieurs scénarios (section 3.7). Ces scénarios sont présentés sur la vue détail afin de fournir le contexte de l'alerte lorsqu'il est disponible. Enfin, les analystes peuvent insérer des notes à chaque alerte afin de documenter leur processus.

3.6 Filtres

Il a été question au début du chapitre du Visual Information-Seeking Mantra de Ben Shneiderman. Rappelons que ce principe définit trois étapes au niveau du processus d'analyse de l'information visuelle : *Overview first*, *zoom and filter* and *details-on-demand*. Jusqu'à présent, l'ensemble des techniques proposées permettent de réduire le nombre d'éléments sur le graphique sans avoir à retirer la moindre donnée. Les techniques de filtrage permettront d'alléger encore davantage la visualisation en retirant les données ne respectant pas certains critères déterminés par l'analyste. Au fur et à mesure que l'analyse chemine dans le processus présenté au chapitre 1, des filtres de plus en plus raffinés seront mis en place afin d'isoler les quelques alertes rattachées à l'incident en cours.

La figure 3.17 illustre l'interface de filtrage proposée par AlertWheel. La solution de filtrage a été développée de façon à accélérer le travail des analystes. Les différents filtres sont regroupés en sept grandes catégories.

- **Analysis Period** : Permet de délimiter la période d'analyse
- **Classification** : Filtrage par classifications de signatures d'alertes (Annexe II)
- **Signature** : Filtrage par signatures d'alertes
- **Source Autonomous Systems** : Filtrage par AS (source)
- **Source IP**: Filtrage par adresse IP (source)
- **Destination IP** : Filtrage par adresse IP (destination)
- **Scenarios** : Afficher / retirer les données associées à des scénarios (voir section 3.7)

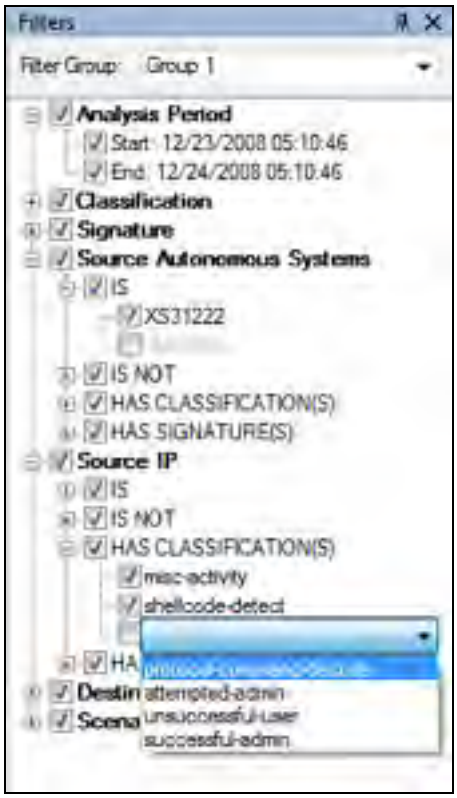

Figure 3.17 Gestionnaire des filtres

Sous chacune de ces catégories, on retrouve certains « opérateurs » permettant de traiter les données d'une façon particulière. La première catégorie (*Analysis Period*) possède deux opérateurs particuliers permettant de définir l'heure et la date de début / de fin de l'analyse. Seules les données comprises durant cette période seront affichées sur les différentes interfaces de l'application. Les six autres catégories possèdent les opérateurs « IS » et « IS NOT ». Ces deux opérateurs couvrent les cas classiques de filtrage. Par exemple, sous *Source IP*, « IS » permet de n'afficher que les alertes dont l'adresse IP source figure dans les valeurs définies par l'utilisateur. À l'inverse, « IS NOT » permettra de retirer certaines alertes de la représentation. Les catégories *Source Autonomous Systems, Source IP* et *Destination IP* offrent également deux autres opérateurs particuliers : « HAS CLASSIFICATION(S) » et « HAS SIGNATURE(S) ». Ces opérateurs permettent de couvrir deux cas un peu plus complexes où l'on voudrait afficher, par exemple, uniquement les AS ayant généré à la fois des alertes d'une classification X et Y. Cette technique est primordiale pour explorer les

patrons d'attaque sur le graphique. Elle permet, par exemple, de vérifier si une attaque détectée précédemment a été répétée à un autre moment dans les données.

Les requêtes sont bâties de façon à ce que les conditions à un niveau supérieur ou égal à l'opérateur soient des conjonctions, alors que la plupart de conditions ce qui se trouve sous l'opérateur sont considérés comme des disjonctions. Autrement dit, si un analyste ajoute un filtre « *Source IP IS 65.21.14.23* » et un filtre « *Destination IP IS NOT 25.37.123.2* », seules les alertes répondant aux deux filtres simultanément seront affichées (conjonction des clauses principales). Par contre, si l'utilisateur applique les filtres « *Source IP IS 65.21.14.23* » et « *Source IP IS 125.18.154.24* », l'application affichera les alertes dont la source est l'une des deux valeurs (disjonction des clauses secondaires). Effectuer une conjonction des clauses secondaires retournerait toujours un ensemble vide puisqu'une adresse ne peut pas prendre deux valeurs simultanément. Les opérateurs « HAS CLASSIFICATION(S) » et « HAS SIGNATURE(S) » ajoutent toutefois une exception à la règle, puisque ces valeurs sont traitées comme des conjonctions, de façon à pouvoir traiter les cas présentés au paragraphe précédent.

La structure en arbre permet d'activer / désactiver des filtres rapidement grâce aux cases à cocher situées à gauche des éléments de la liste. Il est donc possible de modifier l'état des filtres à tous les niveaux de la structure à l'aide d'un seul clic. La modification des filtres s'effectue également très rapidement à l'aide de quelques clics de souris, directement dans la structure en arborescence. Cette approche possède également l'avantage de ne pas nécessiter d'interface secondaire pour la saisie des filtres. Les analystes peuvent consulter et modifier les filtres directement sur l'interface de l'application.

La structure en arbre pose toutefois certaines limites quant à la combinaison des filtres. Il est en effet impossible de jumeler des combinaisons de filtres par l'approche conventionnelle présentée ci-dessus. Par exemple, il serait impossible de traiter le cas $(A \cap B) \cup (C \cap D)$ ou toute autre variante plus complexe. Pour traiter ce genre de combinaisons, il serait toutefois possible d'utiliser des scénarios, bien que cette approche possède malgré tout des limites.

3.7 Scénarios

On entend par scénario une collection d'alertes qu'un analyste veut conserver ou annoter pour utilisation / consultation ultérieure. Les scénarios permettent les transitions entre les différentes étapes du processus d'analyse. Ils proposent en effet à l'analyste d'annoter les données recueillies pour servir de base à l'étape suivante. Ces données peuvent ensuite être utilisées comme filtre de base lors de l'analyse suivante.

Figure 3.18 Gestionnaire des scénarios

La figure 3.18 illustre l'interface des scénarios telle que proposée par AlertWheel. Chaque scénario est tout d'abord identifié par son type. Le type du scénario correspond au niveau dans la hiérarchie des données présentée précédemment dans le tableau 1.1. Il représente en quelque sorte l'étape de l'analyse. On retrouve également le nom du scénario, tel que défini

par l'utilisateur, la date de création du scénario, les dates et heures des premières et dernières alertes contenues dans le scénario, ainsi que le nombre d'alertes pour chaque classification.

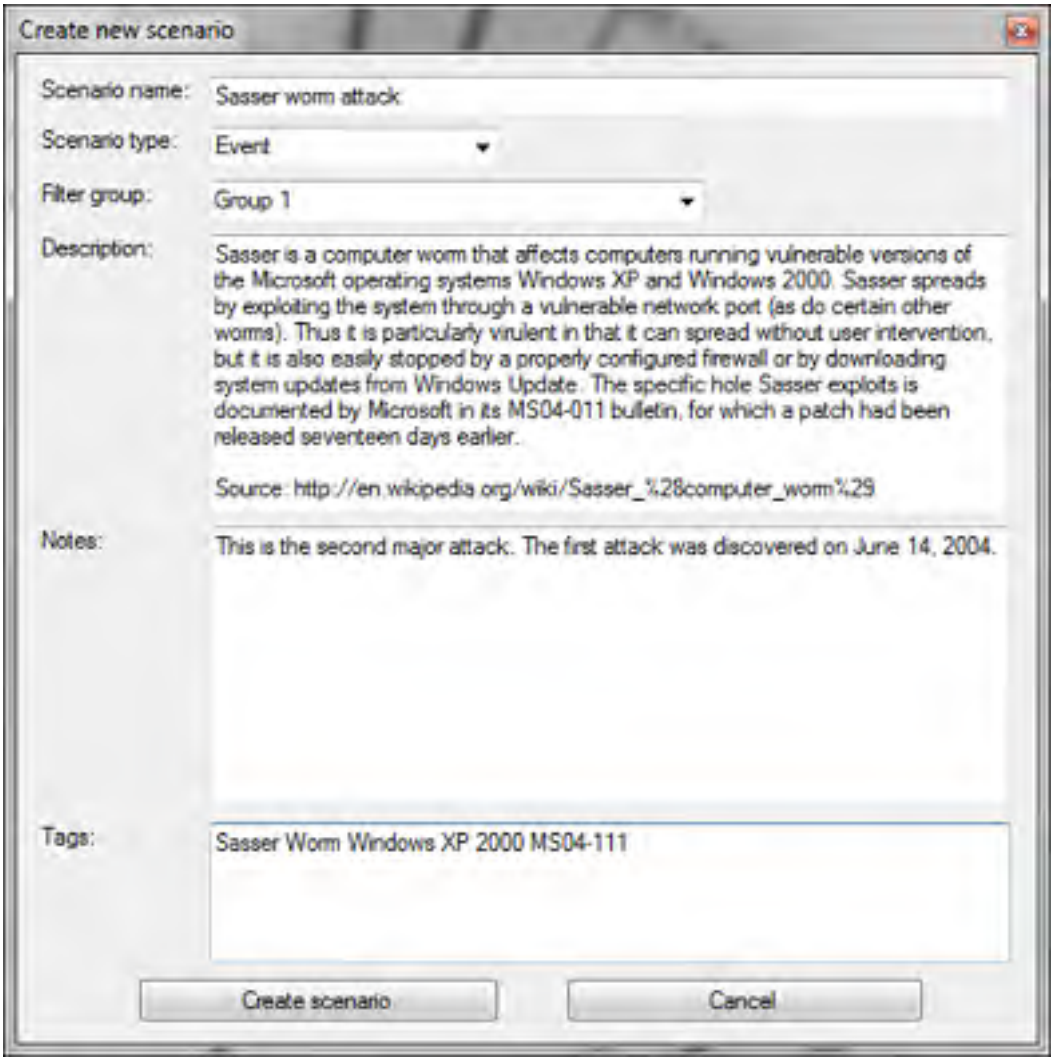

Figure 3.19 Interface de création de scénarios

Dans la version actuelle, AlertWheel propose uniquement une version limitée de la gestion des scénarios. Jusqu'à présent, les scénarios sont principalement utilisés pour conserver et simplifier les opérations de filtrage. Il est en effet possible d'utiliser des scénarios comme filtres, soit en affichant uniquement les alertes d'un scénario, ou encore en les retirant de la vue pour ne conserver que les alertes non traitées.

À terme, chaque scénario servirait de journal complet des événements, permettant aux analystes de conserver l'ensemble des informations à un seul endroit. L'application permettrait, par exemple, de rechercher à travers les différents scénarios, d'utiliser d'autres visualisations pour cibler les tendances par mots clés (par exemple, un *tag cloud*), de comparer des scénarios, de supporter le travail collaboratif, de générer des rapports imprimables, etc. Dans la mesure où la gestion des scénarios pourrait former à elle seule un projet complet, il a été convenu de limiter le développement à une base fonctionnelle du concept pour cette version de l'application. Malgré tout, les différentes fonctionnalités offertes jusqu'à présent rendent l'utilisation des scénarios indispensable pour tout projet d'analyse, et devraient être suffisantes pour satisfaire les analystes initialement.

Il a été question dans la section précédente qu'il était possible d'utiliser les scénarios pour effectuer des filtrages plus complexes. Par exemple. il serait en effet possible de tirer profit des scénarios pour traiter le cas $(A \cap B) \cup (C \cap D)$. Il suffirait alors de créer deux scénarios pour les deux intersections $ScenarioA \leftarrow (A \cap B)$ et $ScenarioB \leftarrow (C \cap D)$, et d'appliquer des filtres « Scenario IS ScenarioA » et « Scenario IS ScenarioB ». Puisque la clause secondaire scenario est définie par la disjonction de ses termes (les identifiants des scénarios), le résultat serait celui escompté.

Il est donc possible d'affirmer que l'application permet de traiter les requêtes conventionnelles qu'un analyste pourrait désirer effectuer, sans nécessiter la saisie de requêtes manuelles. L'analyste n'a alors qu'à utiliser la forme canonique disjonctive (disjonction de conjonctions) de sa requête (Cormen et al. (2001, p. 399)).

3.8 Surbrillance

L'application permet également de mettre des éléments en évidence sans nécessairement appliquer un filtre pour retirer les autres items de la vue. La figure 3.20 illustre l'utilisation de la surbrillance, telle que proposée par AlertWheel. Pour mettre un item en surbrillance, il suffit de le sélectionner avec le curseur. L'item est alors dessiné avec un contour vert, et

seules les alertes de cet item sont présentées en évidence. Les éléments non concernés par la surbrillance sont dessinés avec un niveau d'opacité très faible, simplement pour préserver la structure globale. On remarque sur la figure 3.20 que seules les courbes reliées à l'AS XS40066 sont placées en évidence. Le même traitement est également appliqué sur les arcs du cercle intérieur, de façon à cibler à quel moment durant la période d'analyse les alertes sélectionnées ont été générées. La surbrillance peut être appliquée aussi bien sur un élément de l'anneau extérieur, du cercle intérieur que sur une courbe. La surbrillance s'avère un outil très intéressant, par exemple, pour évaluer quels filtres appliquer sur les données sans avoir à les essayer un par un.

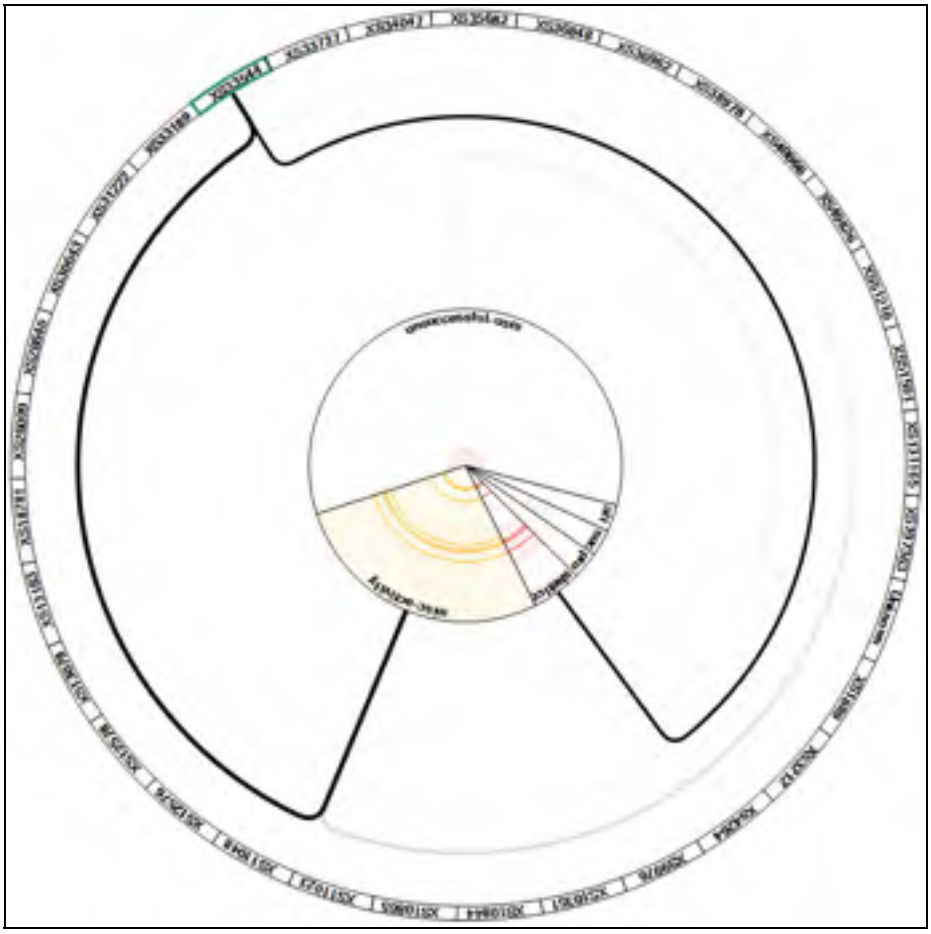

Figure 3.20 Surbrillance après avoir sélectionné l'AS XS33544

La surbrillance est également disponible sur la matrice. Lorsqu'un utilisateur sélectionne un AS sur l'anneau extérieur de la roue, par exemple, l'ensemble des alertes seront présentées dans un encadré vert, tel qu'illustré sur la figure 3.21. Par défaut, l'application affiche malgré tout l'ensemble des alertes sur la grille en présentant celles sélectionnées dans un encadré vert. Vu le nombre d'alertes, il peut toutefois être difficile de retrouver celles qui ont été sélectionnées dans la liste. Pour cette raison, l'application offre deux modes de surbrillance: exclusif et inclusif. Le mode exclusif n'affiche que les alertes sélectionnées sur la roue, alors que le mode inclusif présente l'ensemble des alertes avec celles sélectionnées en surbrillance. Le mode inclusif peut malgré tout être utile pour observer les événements qui se sont produits durant la même période, par exemple. La figure 3.21 illustre le mode exclusif. Le mode inclusif a été utilisé précédemment sur la figure 3.14.

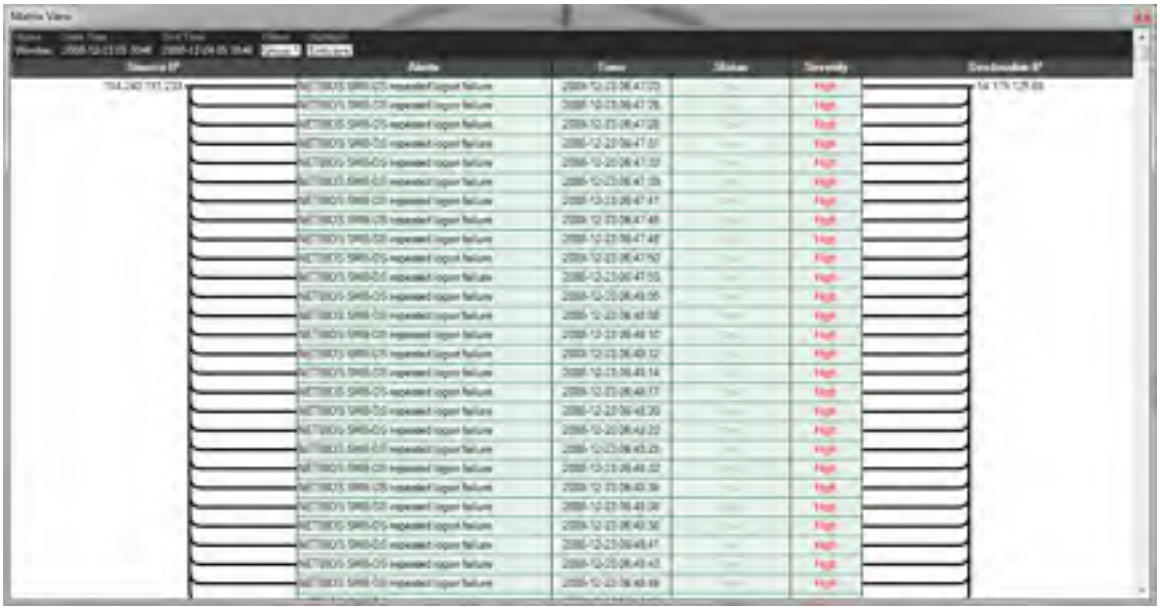

Figure 3.21 Surbrillance exclusive sur la matrice des alertes provenant de la source 184.240.191.233

3.9 Approche à plusieurs vues coordonnées

Parfois, analyser les données sur une seule roue d'alertes (vue principale) à la fois peut être insuffisant pour bien comprendre une situation. AlertWheel permet alors de combiner plusieurs roues (et même plusieurs matrices), de façon à compléter la vue offerte à l'analyste.

La figure 3.22 illustre une combinaison possible des différentes vues sur l'interface. Cette combinaison permet, par exemple, d'afficher à la fois une roue présentant les adresses sources et l'autre les adresses de la destination. Les vues sont également coordonnées afin d'afficher en surbrillance les mêmes alertes lorsque la situation le permet.

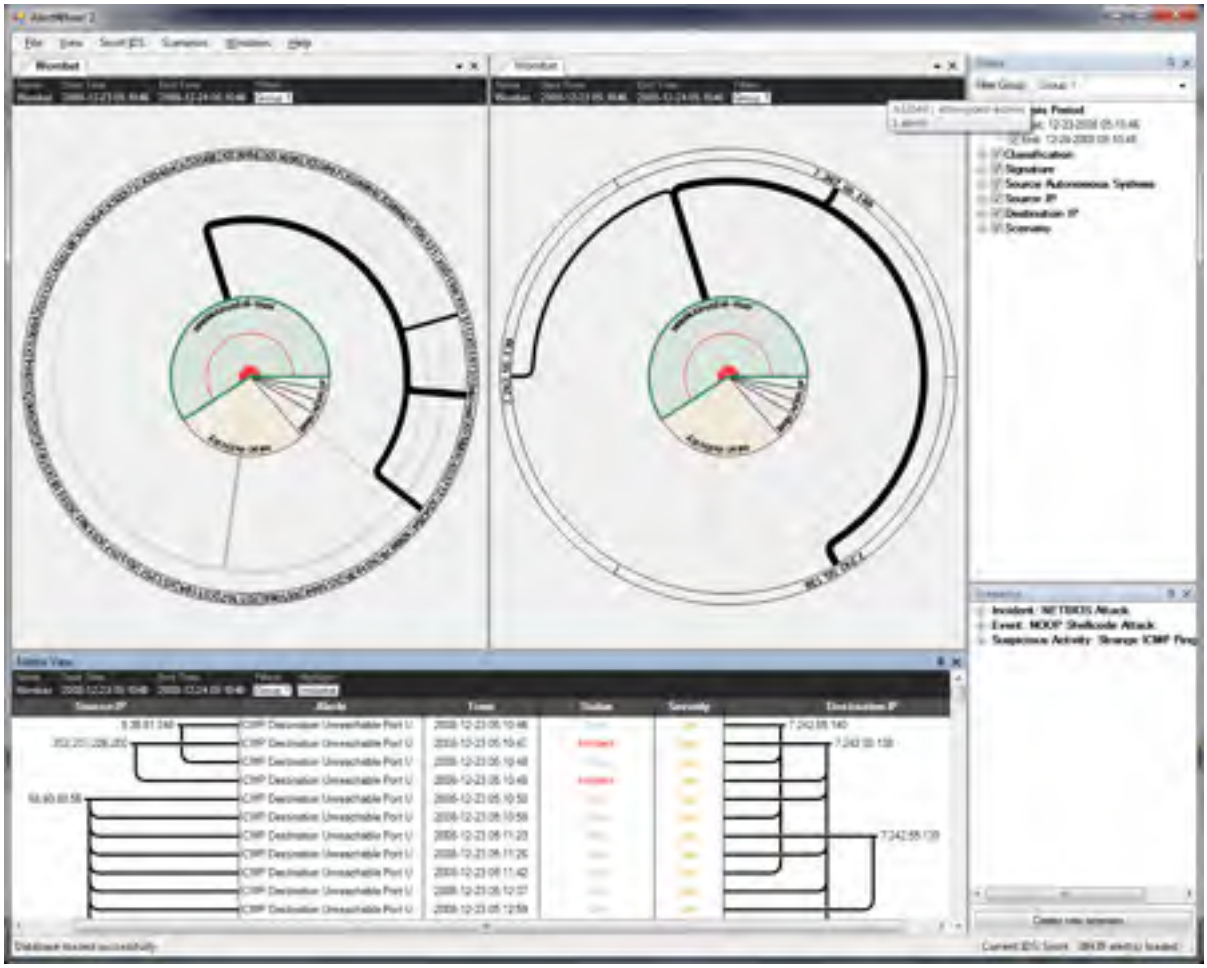

Figure 3.22 Interface combinant deux roues et une matrice. La roue de gauche affiche les sources, alors que la vue de droite affiche les destinations

AlertWheel n'impose aucune contrainte quant au nombre et à la disposition des vues sur l'interface. Les vues peuvent même être retirées de la fenêtre de l'application pour être disposées sur plusieurs écrans.

Il peut être intéressant, dans certaines circonstances, d'appliquer des filtres différents sur les vues de l'interface. Par exemple, cette approche permet de comparer deux périodes d'analyses différences, ou encore deux scénarios similaires. L'application offre ici le concept de groupe de filtres. Le terme apparaît d'ailleurs dans le haut de la figure 3.17 (gestionnaire des filtres), sur la figure 3.18 (scénarios) et dans la barre noire située dans le haut des différentes vues de l'application. Plutôt que d'associer des filtres différents à chaque vue, l'application propose de créer des groupes de filtres autonomes associés à une ou plusieurs vues de l'application. L'ensemble des sept catégories présentées à la figure 3.17 forme un groupe de filtres. Chaque vue est ensuite associée à un groupe de filtres (dans la barre noire située dans le haut de la fenêtre) pour déterminer quel ensemble de filtres doit être appliqué.

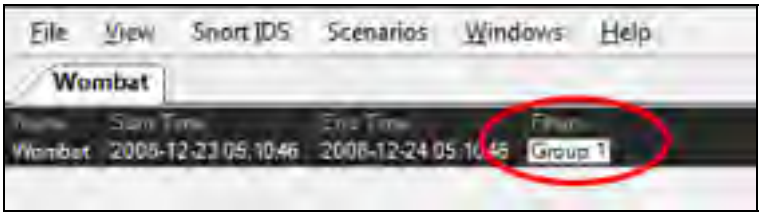

Figure 3.23 Sélection du groupe de filtres dans la barre d'état d'une vue

Cette technique s'avère la plus flexible et la plus simple d'utilisation. Elle permet aussi bien de n'avoir qu'un seul ensemble de filtres pour l'ensemble des vues (par exemple, si une roue représente les sources et l'autre les destinations) qu'un ensemble de filtres pour chaque vue (pour comparer des scénarios, etc.). Elle permet également de copier des groupes de filtres existants pour les raffiner sans nécessairement modifier le groupe original (par exemple, pour comparer une vue filtrée avec une vue plus globale).

Le concept de groupe de filtres est également important lors de la création d'un scénario, de façon à sélectionner le bon ensemble d'alertes à conserver dans le scénario, dans le cas où il y aurait plusieurs vues avec des filtres différents.

3.10 Les faux positifs

Le problème des faux positifs générés par les IDS a été mentionné à plusieurs reprises déjà, et figure parmi les principaux défis des approches en visualisation. Malheureusement, il peut être très difficile d'isoler automatiquement à coup sûr tous les faux positifs générés par des IDS sans introduire de faux négatifs (véritables attaques considérées comme des erreurs de détection). Tel que nous l'avons mentionné au chapitre 1, l'évaluation des attaques doit nécessairement être basée sur l'environnement particulier du réseau, sur les expériences antérieures, sur les attaques passées, sur les tendances du moment, etc. À ce niveau, seul un analyste expérimenté peut déterminer à coup sûr s'il s'agit d'un faux positif ou d'une véritable alerte à investiguer. Il est également important de comprendre que ce problème touche surtout l'analyse réseau. L'analyse des *honeypots* est beaucoup moins sujette à ce genre de problèmes puisque tout le trafic est nécessairement malveillant.

Plutôt que tenter d'isoler automatiquement les faux positifs, AlertWheel propose plutôt des solutions permettant à l'analyste de retirer lui-même les faux positifs, basé sur ses connaissances et sur les différentes techniques de visualisation de l'application. Par exemple, il est possible d'obtenir rapidement les signatures déclenchées et les destinations rattachées, permettant d'évaluer la plausibilité des alertes rapportées. De plus, une fois un faux positif détecté, il est possible d'utiliser les techniques de filtrage présentées précédemment pour récupérer l'ensemble des cas semblables et d'ainsi les retirer très rapidement.

CHAPITRE 4

ÉTUDES DE CAS

4.1 Contexte

Le projet WOMBAT (The Wombat Project (2011)) prône la mise en place de nouvelles techniques permettant d'analyser et de comprendre les menaces émergentes envers les services et l'infrastructure globale d'Internet. Le projet propose la création d'une structure internationale d'acquisition de données brutes grâce à une multitude d'intervenants de tous les milieux. L'acquisition est effectuée en partie grâce à des réseaux *d'honeypots* répartis de façon stratégique un peu partout sur la planète. Ces données permettent de mieux comprendre les menaces et les tendances au niveau international.

Afin d'évaluer les techniques proposées par AlertWheel, il a été possible d'obtenir un échantillon de trafic IP capturé par un des *honeypots* de Wombat (possédant trois adresses IP distinctes, mais pointant sur un même serveur) entre le 23 décembre 2008 et le 14 janvier 2009. Le résultat prend la forme d'un fichier PCAP de 150 Mo contenant l'ensemble du trafic capturé sur les trois interfaces réseau de *l'honeypot*. Les données ont ensuite été analysées par Snort (version 2.9.0), où 38439 alertes ont été générées. Ces alertes ont ensuite été importées dans AlertWheel pour servir de base d'analyse. Les informations ont toutefois été anonymisées pour protéger les entités concernées.

4.2 Cas #1 : Analyse des données d'une journée

De façon générale, les analystes préconisent des analyses quotidiennes des données enregistrées. Cette contrainte sur la période d'analyse permet de limiter le nombre d'alertes à visualiser et de représenter plus facilement la séquence des événements sur le cercle intérieur (les courbes de couleur au centre représentent le moment des attaques). Elle permet également de mettre plus facilement en évidence les patrons d'attaques sur la représentation, Afin d'illustrer ce concept, nous utiliserons ici la journée du 24 décembre 2008. Malgré le

fait qu'il s'agissait de la veille de Noël, 2114 alertes ont été générées par Snort pour cette seule journée.

Une technique intéressante pour repérer initialement des comportements suspects consiste à activer la compression des liens. La figure ci-dessous illustre les données de la journée après le regroupement.

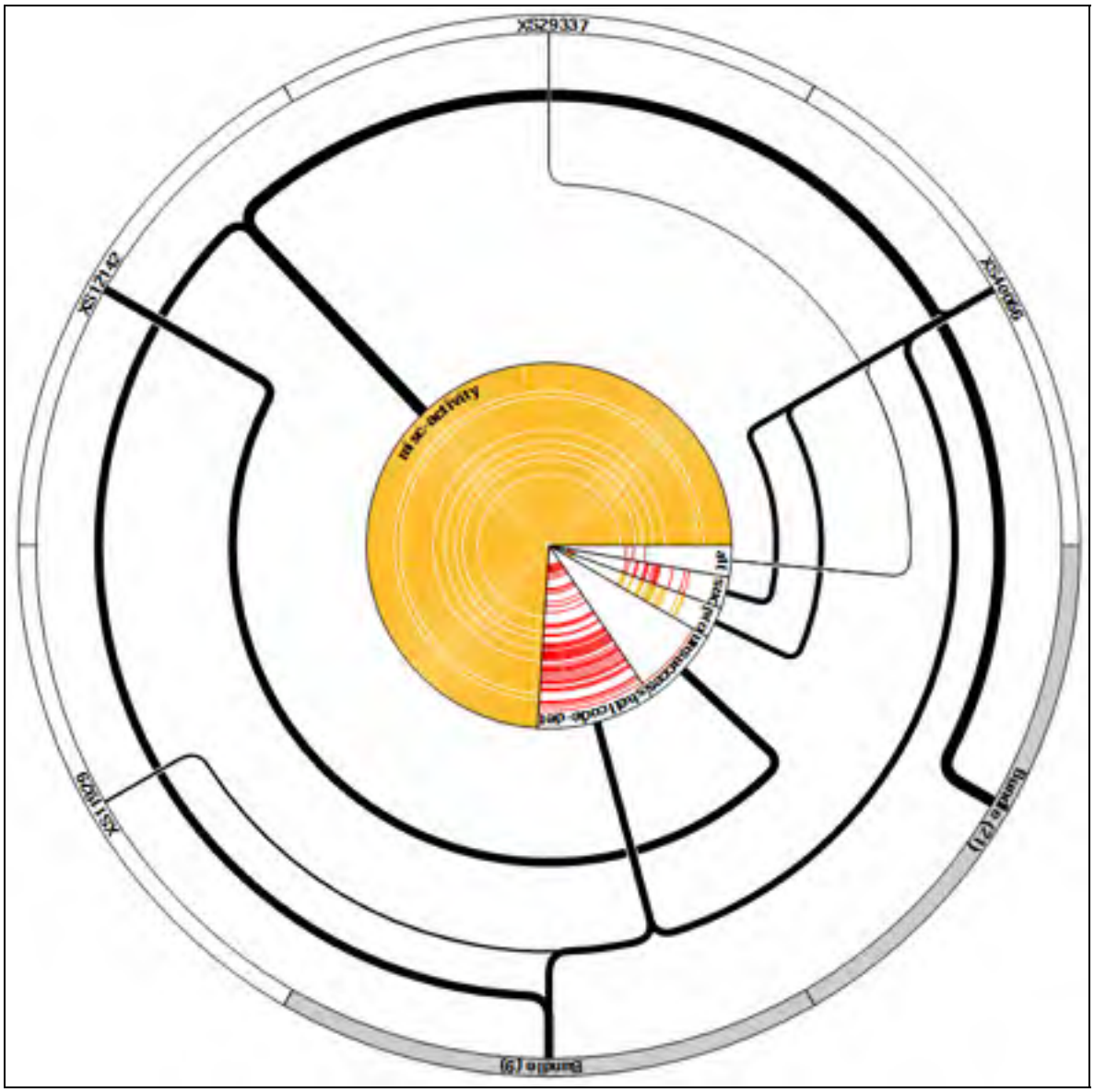

Figure 4.1 Ensemble des alertes du 24 décembre 2008 après le regroupement par compression des liens

On remarque rapidement que la plupart des AS présentent des attaques similaires, regroupées sous Bundle (21) et Bundle (9). Seuls quatre AS présentent des patrons différents. L'AS XS40066 semble particulièrement intéressant dans les circonstances, dans la mesure où il présente des liens vers toutes sauf une classification. En sélectionnant le secteur, on applique la surbrillance, tel qu'illustré par la figure ci-dessous.

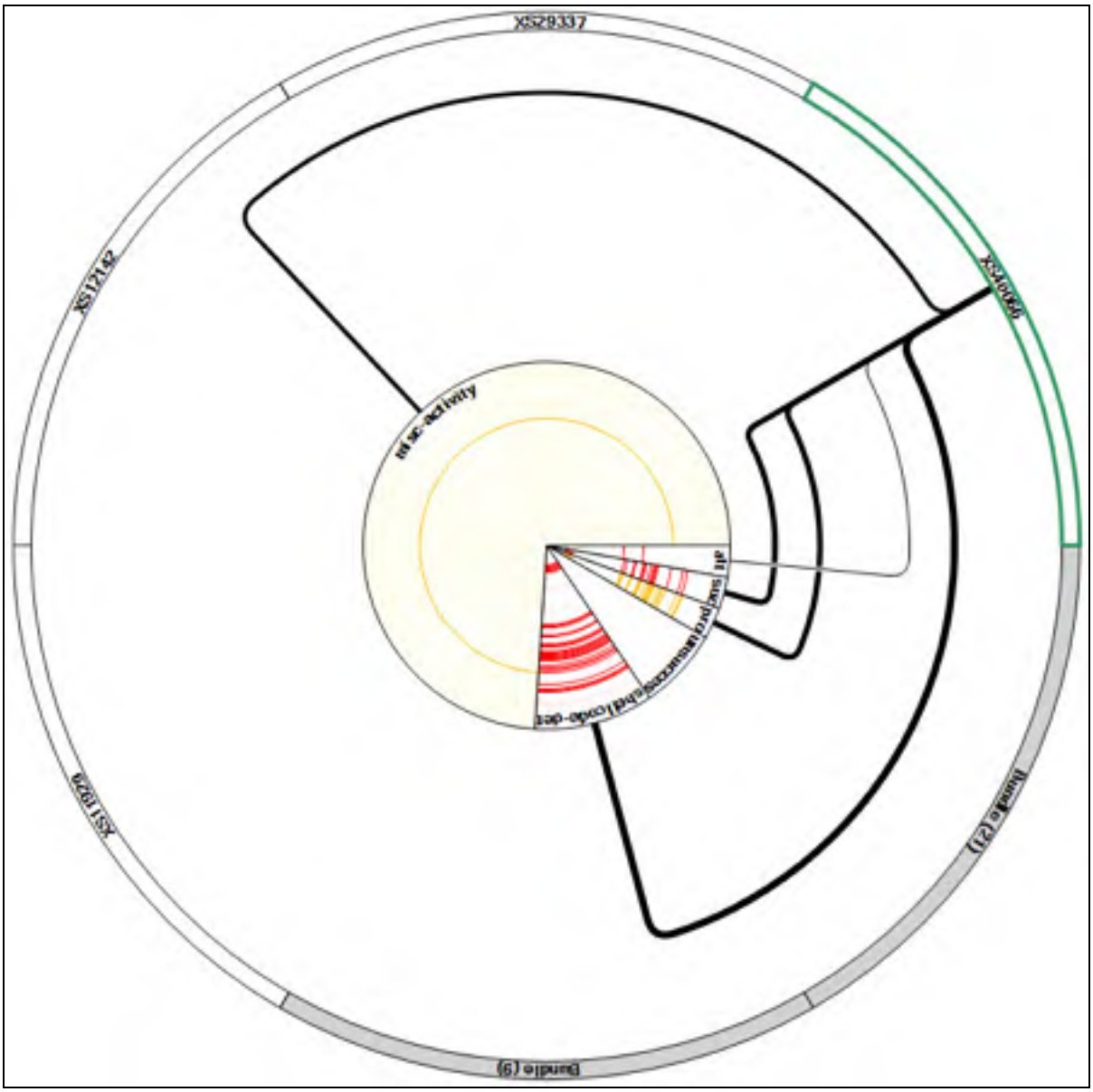

Figure 4.2 Surbrillance appliquée sur l'AS XS40066

On remarque rapidement qu'une grande partie des alertes critiques (rouges) semblent également provenir de cet AS, ce qui confirme par le fait même l'intérêt pour celui-ci. On applique alors une règle de filtrage « *Source Autonomous Systems* IS *XS40066* » pour ne retenir que ce système sur la visualisation.

Une fois la règle appliquée, il ne reste qu'un seul élément sur le cercle extérieur. À ce moment, il devient intéressant d'exploser les AS et d'afficher plutôt directement les adresses IP source sur le cercle extérieur. Le résultat est illustré sur la figure ci-dessous.

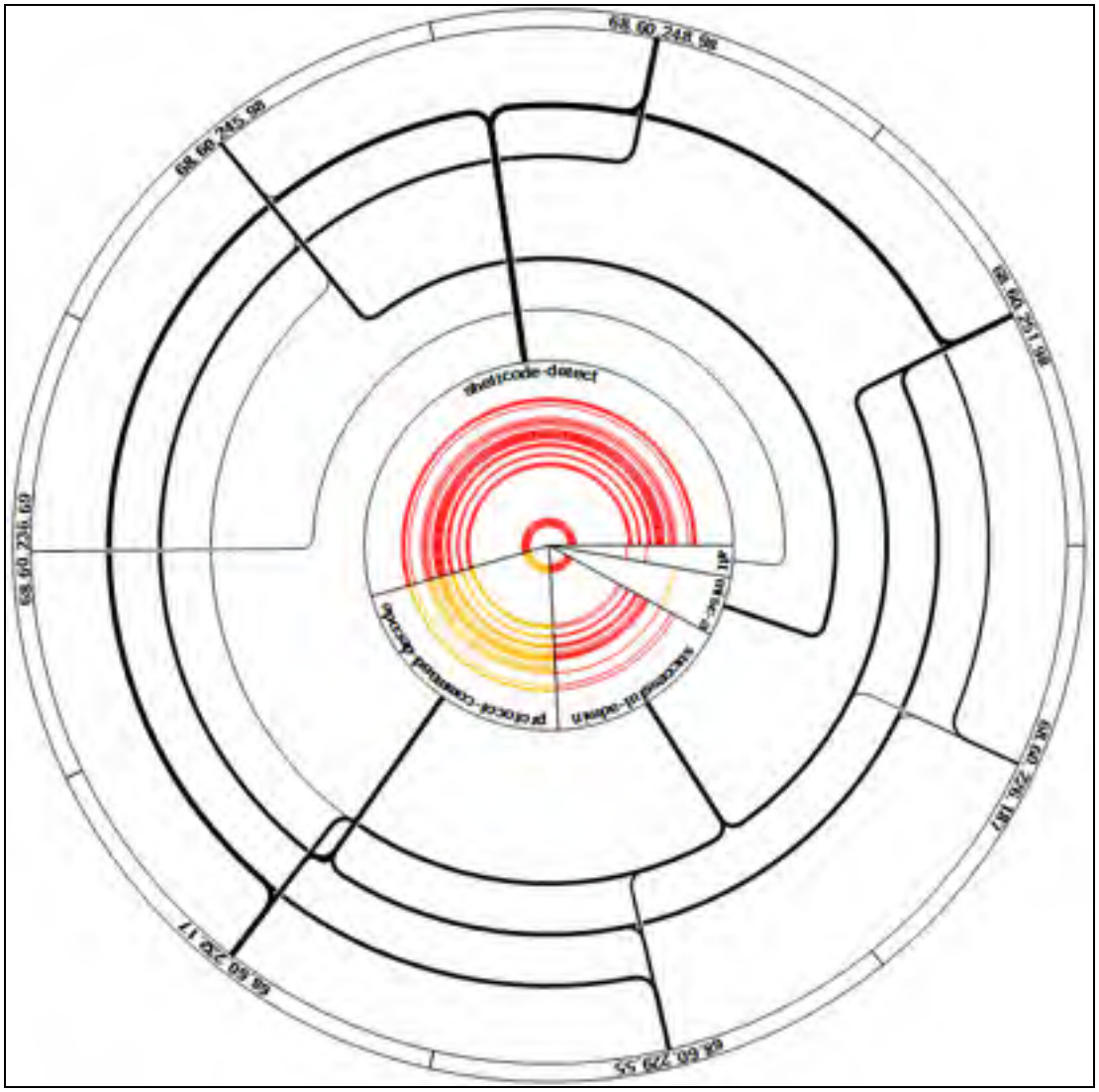

Figure 4.3 Alertes regroupées par sources sur le cercle extérieur pour l'AS XS40066

En se fiant sur l'épaisseur des courbes, on remarque qu'une grande partie des alertes proviennent des adresses 68.60.232.17, 68.60.248.96 et 68.60.251.98. On peut d'ailleurs confirmer cette hypothèse en utilisant la surbrillance sur ces adresses. Notre attention sera portée dans ce cas-ci sur l'adresse 68.60.232.17, laquelle semble avoir effectué des attaques à plusieurs reprises sur les *honeypots* (plusieurs traits distincts au centre). On applique alors un filtre pour isoler cette adresse « S*ource IP* IS *68.60.232.17* ».

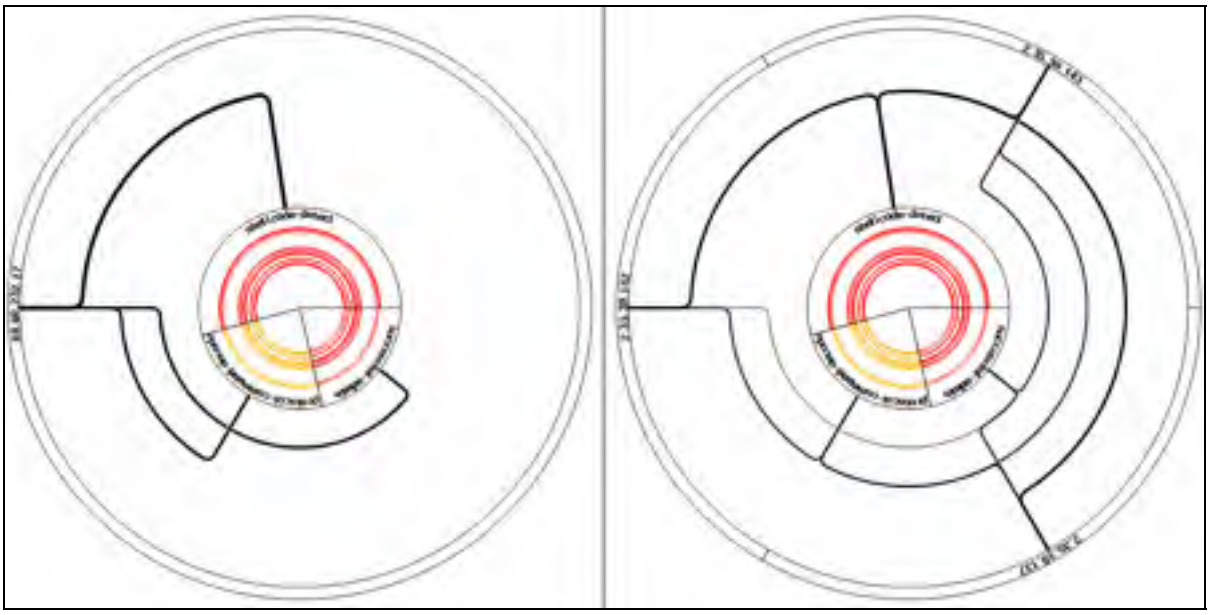

Figure 4.4 Alertes pour l'adresse 68.60.232.17 regroupées par source (gauche) et par destination (droite)

En affichant les destinations sur le cercle extérieur, on remarque que cette source semble avoir effectué des attaques sur les trois adresses du système. Cette remarque n'a rien de surprenant en soi, dans la mesure où la plupart des attaques répertoriées par des *honeypots* ont été initiées par un balayage d'adresses. Avec trois adresses de destination séquentielles (2.35.39.142, 2.35.39.143, 2.35.39.144), cette constatation est donc tout à fait normale.

Une fois une adresse source isolée, il est intéressant d'utiliser la vue matrice pour obtenir plus de détails sur les alertes individuelles rapportées pour cette source. La matrice permet d'afficher les alertes selon la séquence d'enregistrement des événements. Cette technique permettra donc de mieux comprendre ce que l'attaquant a effectué comme attaque, et de

quelle façon la victime a répondu. La figure ci-dessous illustre la matrice obtenue à partir des données affichées sur la roue.

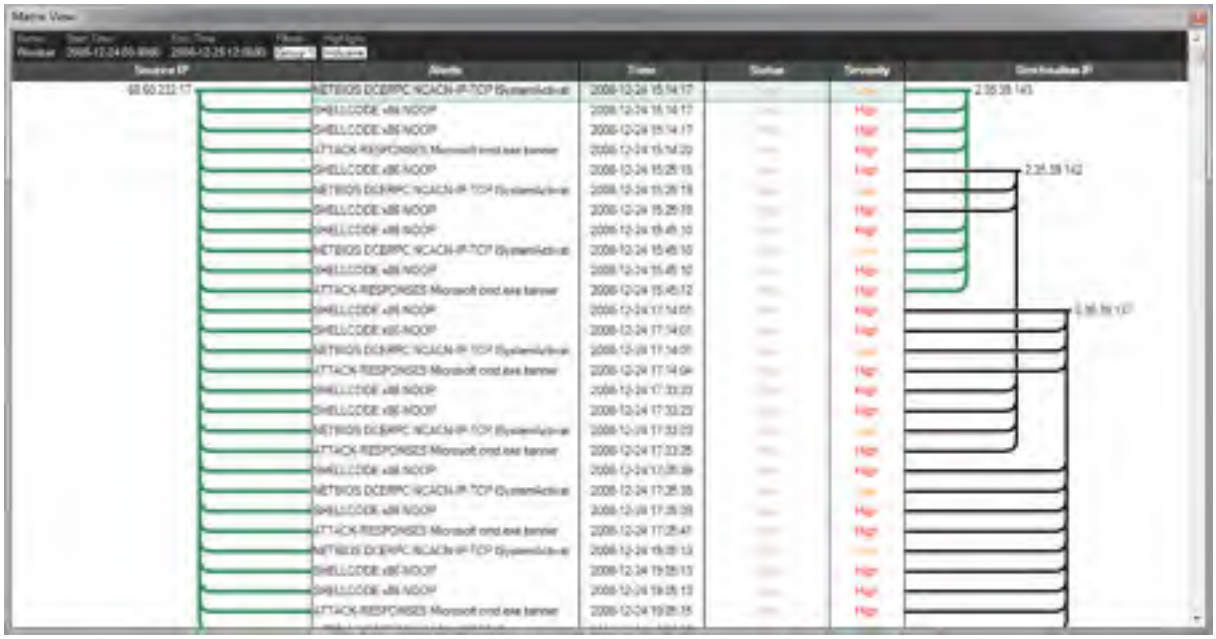

Figure 4.5 Vue matrice des données

En analysant les données de la matrice, un patron d'attaque récurrent sur l'ensemble des adresses du *honeypot* ressort facilement. L'attaquant exploite une vulnérabilité connue du service RPC DCOM de Windows XP/NT/2000/2003, détectée par Snort comme « NETBIOS DCERPC NCACN-IP-TCP ISystemActivator RemoteCreateInstance attempt ». L'attaque se déroule de la façon suivante :

1. L'attaquant envoie simplement une requête malformée sur un port RPC du serveur, causant un dépassement de tampon (buffer overflow).
2. L'attaquant envoie ensuite des commandes shellcodes, lesquelles peuvent exécuter des opérations en tant qu'administrateur.
3. L'alerte « ATTACK-RESPONSES Microsoft cmd.exe banner » démontre que l'attaque a été réussie, et que l'utilisateur est parvenu à accéder à l'invite de commande Windows (simulé par le *honeypot*).

Il est probable que l'attaquant ait tenté cette manoeuvre à plusieurs reprises avec quelques variantes, possiblement insatisfait des réponses du serveur. Il est possible de confirmer cette hypothèse en accédant aux informations du paquet en double-cliquant sur une alerte de la matrice. Il suffit maintenant de créer un scénario pour conserver cette attaque dans la base de données pour consultation ultérieure ou pour approfondir l'analyse au besoin. Il est possible d'imaginer, par exemple, qu'un analyste de premier niveau pourrait maintenant transférer le dossier à un analyste de deuxième niveau afin de comprendre les spécificités de l'attaque. Ce dernier n'aurait alors qu'à charger le scénario à l'aide d'un filtre et à explorer les différentes alertes individuellement pour mieux en comprendre les détails.

4.3 Cas #2 : Généralisation d'une attaque

Une fois une attaque isolée et bien évaluée, il peut être intéressant de déterminer si cette attaque est un cas isolé ou si elle se retrouve à plusieurs endroits dans les journaux. L'application permet de rechercher des patrons de ce genre grâce à l'opérateur « HAS SIGNATURE(S) ». On recherche ici toutes les sources ayant rapporté à la fois des alertes de type « NETBIOS DCERPC NCACN-IP-TCP ISystemActivator RemoteCreateInstance attempt », « SHELLCODE x86 NOOP » et « ATTACK-RESPONSES Microsoft cmd.exe banner ». On utilise donc le filtre « *Source IP* HAS SIGNATURE(S) » pour l'ensemble des trois signatures. Le résultat est présenté sur la figure 4.6.

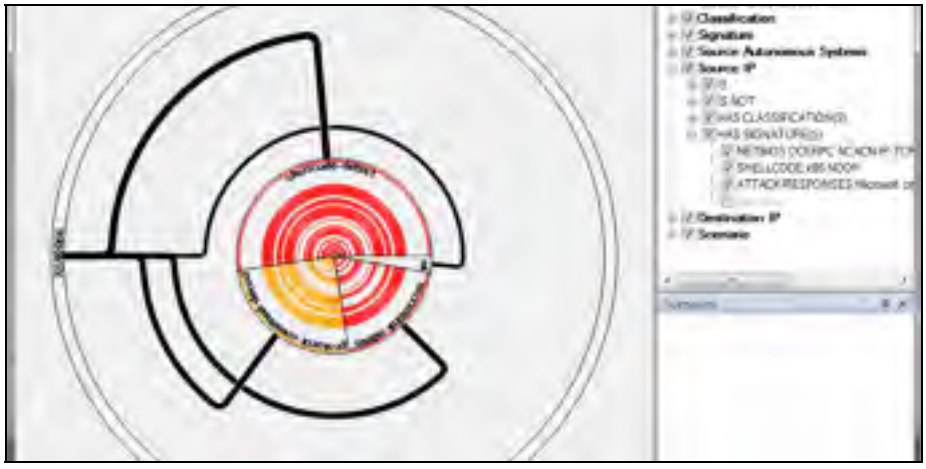

Figure 4.6 Filtrage à l'aide de l'opérateur « HAS SIGNATURE »

Afin d'explorer davantage les données, la période d'analyse a été étendue à la semaine du 23 au 30 décembre 2008. On remarque ici que ce patron d'attaque n'est présent que pour les sources regroupées sous l'AS XS40066. Lorsque l'on explose la visualisation pour afficher les adresses sources, on remarque une quantité un peu plus importante d'adresses que précédemment.

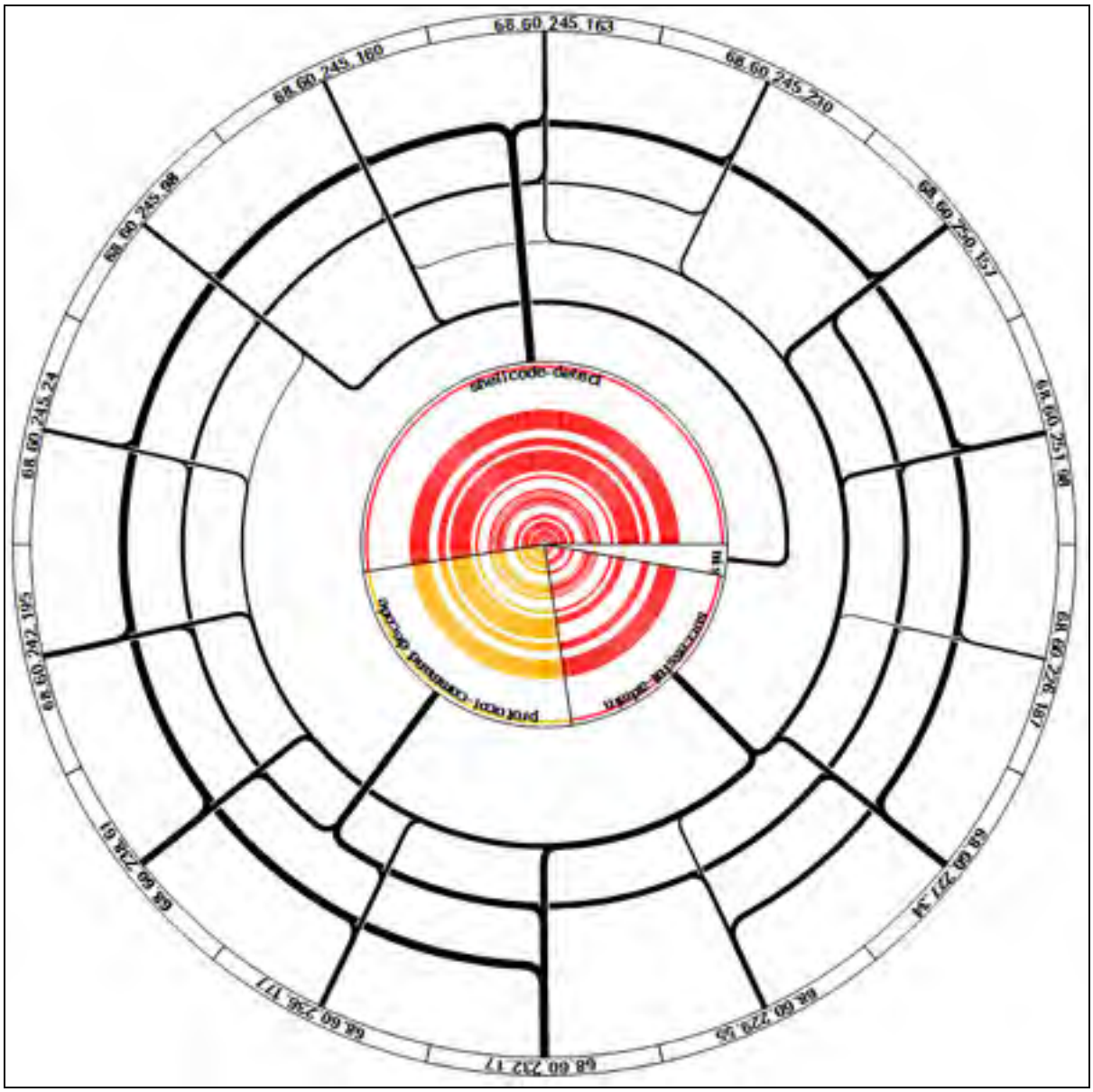

Figure 4.7 Adresses IP des systèmes présentant le même patron d'attaque

La vue matrice confirme que l'ensemble des sources présentent le même patron d'attaque. Le fait que l'ensemble des sources sont regroupées dans un même système autonome et que les attaques soient effectuées de façon aussi similaire laisse présager que l'ensemble des systèmes puissent être contrôlés par une même entité (ex. : *botnet*). Un attaquant pourrait avoir pris le contrôle de ces machines au cours d'une attaque précédente et les utiliserait à l'insu de leur propriétaire pour attaquer d'autres systèmes de façon anonyme. Il faudrait toutefois investiguer un peu plus en profondeur pour confirmer cette théorie, par exemple en recherchant une signature commune dans les paquets envoyés par les différents attaquants.

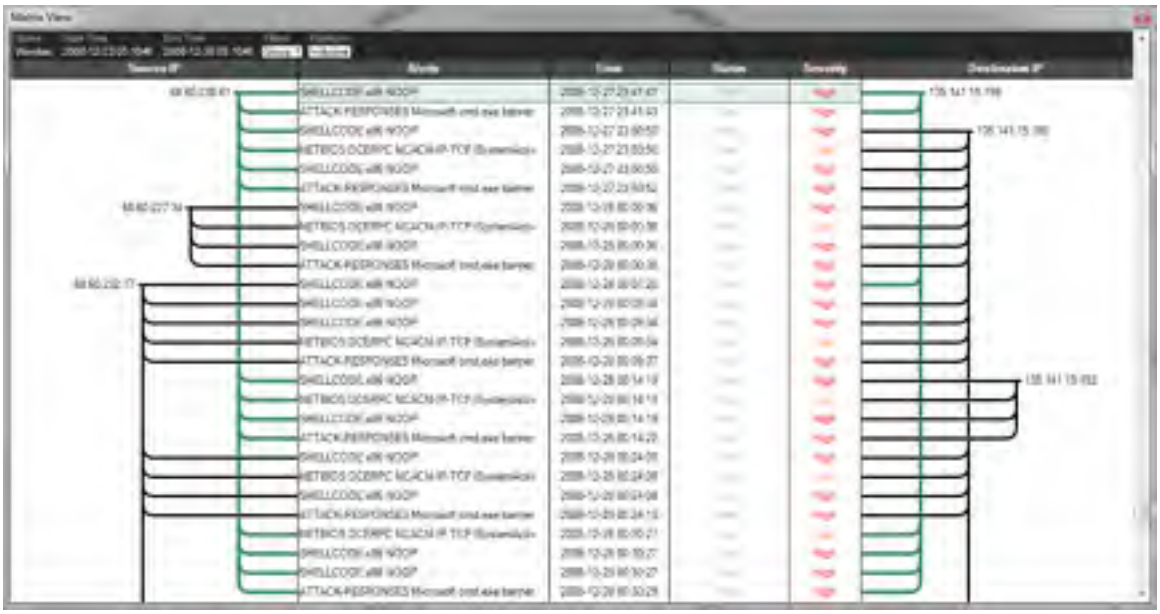

Figure 4.8 La vue matrice confirme qu'il s'agit bien du même patron d'attaque

Ces deux cas auront permis de démontrer la facilité avec laquelle il est possible d'utiliser l'interface proposée pour repérer et investiguer des attaques dans un ensemble très important d'alertes. En combinant une ou plusieurs roues, la matrice, les filtres et les scénarios, il est possible de couvrir la plupart des cas classiques d'analyse, peu importe le nombre d'alertes à afficher. L'application peut en effet afficher plusieurs dizaines de milliers d'alertes grâce aux différentes techniques de regroupement et à la disposition proposée. L'interface d'AlertWheel se distingue également par ses capacités à généraliser les patrons d'attaques, tel qu'illustré par le deuxième cas.

CHAPITRE 5

DISCUSSION

5.1 Nombre d'alertes et de sources

L'étude de cas présentée au chapitre 4 illustre très clairement qu'un nombre d'alertes élevé ne ralentit en rien l'analyse des données avec AlertWheel. Les différentes techniques de regroupement utilisées par l'application permettent de limiter le nombre d'éléments sur le graphique sans nécessairement perdre d'information critique pour l'analyse. Les différentes techniques de filtrage permettent ensuite de retirer les éléments non pertinents de l'analyse, jusqu'à ce qu'un cas précis soit isolé sur la représentation. L'utilisation de scénarios permet enfin de documenter le cas découvert et de s'en resservir un peu plus tard pour des analyses plus approfondies.

Une des principales critiques exprimées envers VisAlert, la solution la plus raffinée jusqu'à présent dans le domaine, concernait le nombre peu élevé d'adresses source / destination et d'alertes qu'il était possible d'afficher sur la visualisation avant que la représentation ne devienne complètement illisible à cause des superpositions des éléments. Contrairement à cette approche, il est possible d'afficher plusieurs centaines d'adresses sur le cercle extérieur d'AlertWheel avant de se retrouver avec de l'occlusion. Mieux encore, l'application permet de regrouper les adresses en fonction de leur système autonome, ce qui permet de réduire considérablement le nombre d'éléments à afficher sur le cercle extérieur, et surtout le nombre de liens sur la visualisation. L'application est donc beaucoup moins limitée quant au nombre de sources qu'il est possible d'afficher. De plus, les éléments de la visualisation ne sont jamais placés de façon superposée, ce qui limite grandement les risques d'occlusion. En fait, l'occlusion est possible uniquement dans le cas où l'espace nécessaire pour afficher une courbe n'est pas suffisant à cause du nombre extrêmement important de sources sur le cercle extérieur ou de catégories au centre. Malgré tout, avec 38 439 alertes provenant de 868 sources différentes, aucune occlusion n'est présente sur la visualisation lorsque les adresses sources sont regroupées par AS (400 secteurs).

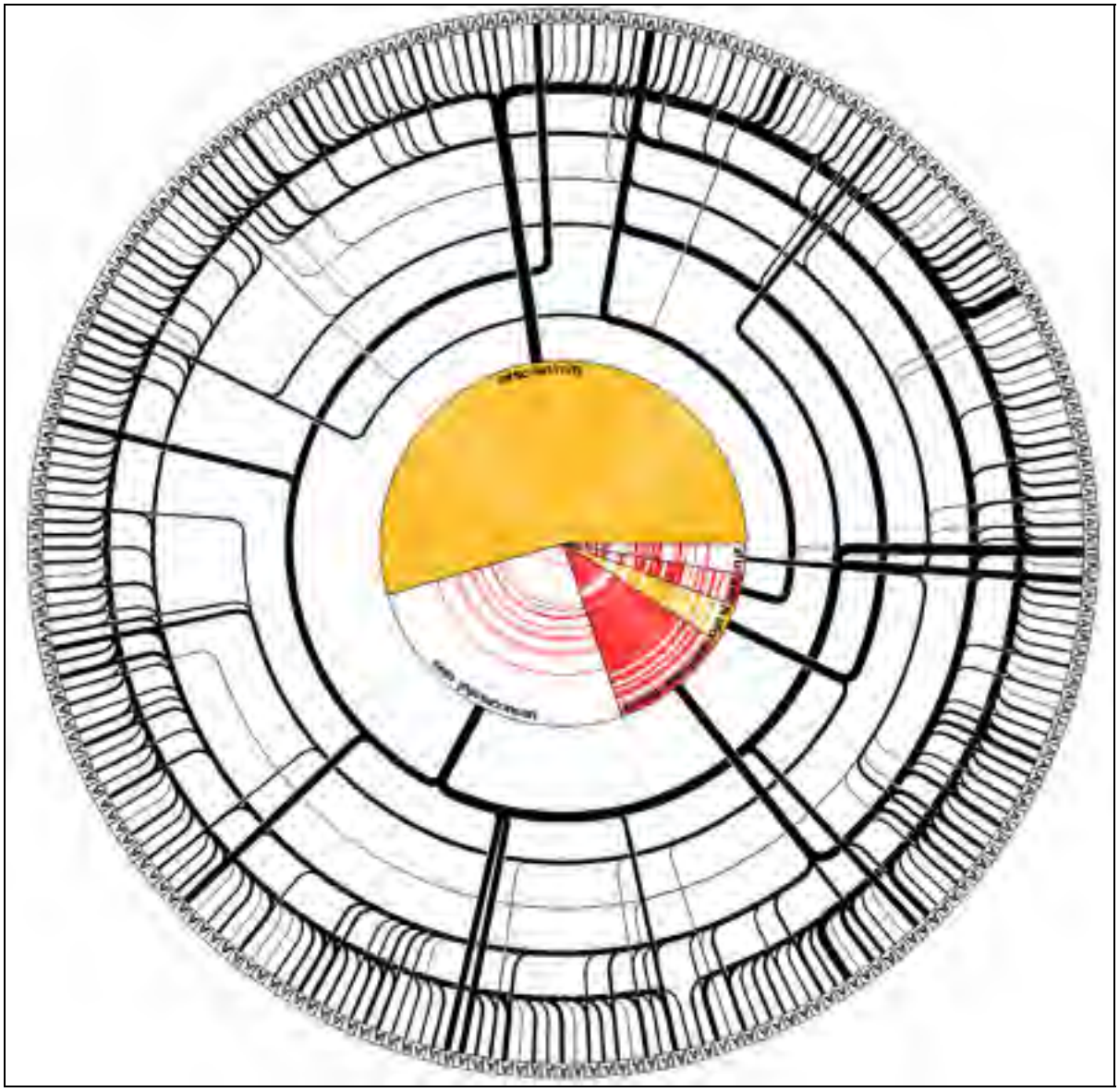

Figure 5.1 Bien que la figure soit très chargée, les 400 AS peuvent malgré tout être distingués individuellement sans occlusion

Cette caractéristique est particulièrement importante pour visualiser les données provenant des *honeypots*. Puisque l'analyse de ces données est principalement orientée sur l'adresse IP de l'attaquant (source), l'application doit être en mesure de représenter n'importe quelle adresse IP accessible sur Internet. En regroupant par AS, on limite malgré tout le nombre d'éléments potentiels à afficher sur le cercle extérieur à un peu plus de 36 000 (d'après le

fichier de configurations de Snort utilisé au moment de l'écriture du mémoire). Évidemment, il serait impossible d'afficher sans occlusion 36 000 éléments sur le cercle extérieur. Il serait toutefois possible d'appliquer des filtres élémentaires (ex : retirer initialement les attaques mineures comme les misc-activity) pour réduire le nombre de données et malgré tout obtenir une vue significative sans perdre trop d'information. En pratique, il est très peu probable que des attaques provenant d'autant d'AS soient observées dans un intervalle de temps raisonnable. Il est donc possible d'affirmer que la solution présentée est suffisamment flexible pour couvrir la plupart des cas d'analyse possibles, même lorsque le nombre d'alertes augmente significativement.

La solution repose toutefois sur un principe simple : le nombre d'éléments au centre (les classifications) est significativement inférieur au nombre d'éléments sur le cercle extérieur. Dans la mesure où un rayon est réservé pour chaque secteur du cercle intérieur, il est primordial que le nombre d'éléments au centre soit limité, sans quoi les cercles risqueraient de se chevaucher, causant par le fait même de l'occlusion et rendant la visualisation inutilisable. Dans le contexte, puisque le nombre de catégories est limité, cette hypothèse peut s'avérer réaliste. Toutefois, certains ajustements seraient nécessaires si cette quantité venait qu'à augmenter.

5.2 Aspect temporel

Une autre critique exercée à l'endroit des approches existantes concernait l'absence de la notion de séquence des événements. Il est en effet souvent impossible de déterminer l'ordre dans lequel les événements sont survenus, un élément souvent crucial pour l'analyse des patrons d'attaques. Tel qu'illustré au chapitre 3, AlertWheel permet de placer en évidence l'aspect temporel aussi bien au niveau de l'analyse globale qu'au niveau de la liste des alertes. Sur la vue principale, il est par exemple possible de visualiser à quel moment un AS a effectué son / ses attaque(s) au cours de la période d'analyse grâce aux courbes du cercle intérieur (figure 3.7). Tel qu'illustré dans l'étude de cas #1, cette technique permet d'évaluer rapidement l'ampleur d'une attaque (durée, fréquence, etc.). Elle permet également d'évaluer

au premier coup d'œil le lien entre les différentes classifications (par exemple, est-ce que les alertes de différentes catégories se retrouvent au même moment?), et d'ainsi établir une première image d'un patron d'attaque. Il est ensuite possible d'évaluer de façon plus précise la séquence exacte des événements grâce à la vue matrice, qui permet de trier les alertes en fonction du moment où elles ont été générées.

Toutefois, il est possible, voire probable, que les alertes ne soient pas générées exactement dans l'ordre où les événements se sont produits, principalement à cause des techniques de détection des IDS. Il est également possible que les heures ne soient pas parfaitement synchronisées entre les différents serveurs ayant capturé le trafic réseau. Il est donc important de considérer cet aspect lors de l'analyse et d'éviter de tirer des conclusions quant à la séquence d'événements très rapprochés temporellement. Malgré tout, le fait qu'une série d'alertes provenant d'une même source vers une même destination dans un espace-temps très rapproché peut être suffisant pour comprendre le patron d'attaque, et l'application met tout en œuvre pour que ces patrons soient placés en évidence, peu importe leur séquence exacte.

5.3 La réalité du processus

AlertWheel a été développé de façon à respecter le mieux possible les différents processus tels qu'énoncés par les analystes dans le domaine. Évidemment, il est difficile d'évaluer à quel point cet objectif a été réalisé puisqu'aucun professionnel n'a pu être impliqué dans le cadre de ce projet. Malgré tout, les principales exigences ont été clairement identifiées grâce aux articles présentant cette réalité, et des solutions concrètes ont été élaborées pour répondre à ces attentes.

La principale amélioration apportée par AlertWheel comparativement aux approches concurrentes concerne l'utilisation des scénarios et de la journalisation. Le fait de pouvoir concentrer à un seul endroit l'ensemble du processus et de la documentation facilitera à coup sûr grandement le travail des analystes et des intervenants dans le domaine. L'utilisation des

scénarios permet également de documenter chaque étape du processus d'analyse, ce qui n'était pas possible non plus avec les approches existantes.

De plus, dans la mesure où aucun spécialiste du milieu n'a été impliqué dans le projet, il n'a pas été possible d'évaluer la solution dans un milieu pratique. Les différentes analyses effectuées avec l'application ont été réalisées essentiellement en fonction des observations recueillies par les différents travaux de recherches antérieurs présentés au chapitre 1.3.

CONCLUSION

AlertWheel propose une solution novatrice permettant de visualiser une quantité importante d'alarmes provenant d'IDS pour des réseaux de grande envergure, sans introduire d'occlusion ou de confusion dans la visualisation, ni perdre d'information importante pour l'analyse des données. Tel qu'illustré au chapitre 4, l'application permet de détecter des patrons d'attaques de façon rapide et efficace dans de grands ensembles d'alertes, et de les documenter et de les conserver pour des analyses ultérieures. Contrairement à la plupart des approches, AlertWheel respecte également les techniques de travail et les processus généraux établis par les analystes dans le domaine.

La solution présentée dans ce document exploite plusieurs techniques novatrices de visualisation permettant d'alléger les différentes vues, d'éviter toute forme de confusion, tout en assurant un niveau de détails suffisant sur la représentation. L'approche propose plusieurs contributions au domaine de recherche. Parmi celles-ci, citons la particularité des courbes pour la visualisation radiale d'un graphe biparti, les différentes dispositions des liens proposées (linéaire et radiale), la combinaison de la compression et du regroupement des liens et l'utilisation des courbes jumelées à une matrice pour représenter plus efficacement les données. L'interface de filtrage propose également une structure originale en arborescence très flexible dont il n'a pas été possible de trouver d'équivalent jusqu'à présent.

Un concept important du processus d'analyse a toutefois été laissé de côté dans le cadre de ce projet : la corrélation. Il a été question de ce concept dans le chapitre 1 lorsqu'il était question du processus d'analyse. En effet, à un certain point, l'analyste doit vérifier les informations provenant de sources autres que les IDS pour obtenir une image complète de la situation. VisAlert propose, entre autres, une solution intéressante pour corréler les journaux des IDS aux journaux systèmes (par exemple, événements Windows) et aux alertes des différents services sur le réseau. En inversant les axes par rapport à VisAlert, AlertWheel limite la quantité d'information qu'il est possible d'afficher au centre du graphique. Il serait alors intéressant d'évaluer de quelle façon il serait possible d'afficher ces informations sur la

visualisation. Une technique consisterait à regrouper initialement les événements par source de détection (IDS, Antivirus, FTP, Windows, etc.). De la même manière que pour les AS / adresses sources, il serait alors possible d'exploser en catégories d'alertes une fois le nombre d'éléments limité par les différentes techniques de filtrage, par exemple.

Un autre problème concerne la corrélation entre les différents IDS. Jusqu'à présent, les alertes présentées dans l'application proviennent exclusivement d'un seul IDS (Snort). S'il est nécessaire d'afficher les résultats de différents IDS, il est évident que les signatures, les classifications, les degrés de sévérité et autres éléments particuliers de ce genre seront différents et qu'il sera impossible de les corréler. Une solution consisterait à prétraiter les alertes à partir d'une solution intermédiaire, de façon à ce qu'une seule source cohérente d'information soit utilisée par l'application. Il serait toutefois intéressant de trouver une façon efficace d'effectuer ce travail directement dans l'application sans avoir à tenir de table d'équivalences.

Il serait également intéressant de trouver une façon d'identifier clairement les faux positifs et de les traiter d'une façon particulière dans le processus d'analyse. Il est possible, en ce moment, de créer un scénario « Faux positifs » et d'y insérer toutes les alertes. Il serait toutefois plus intéressant d'adresser ce problème par un concept différent, de façon à les représenter possiblement différemment sur la représentation, par exemple. Dans un cas idéal, certains algorithmes d'intelligence artificielle pourraient également automatiquement identifier des faux positifs en fonction des données historiques recueillies par l'application et permettre à l'analyste de confirmer la nature de ces alertes.

Malgré ces limites, il est possible d'affirmer qu'AlertWheel offre plusieurs techniques pertinentes pour l'analyse des données réseau, et qu'il s'agit déjà d'un outil intéressant pour les analystes dans le domaine. À terme, AlertWheel offrira aux analystes une solution intégrée d'exploration, d'analyse, et de documentation pour les alarmes provenant des IDS.

ANNEXE I

Processus de détection des intrusions selon D'Amico et Whitley (2008)

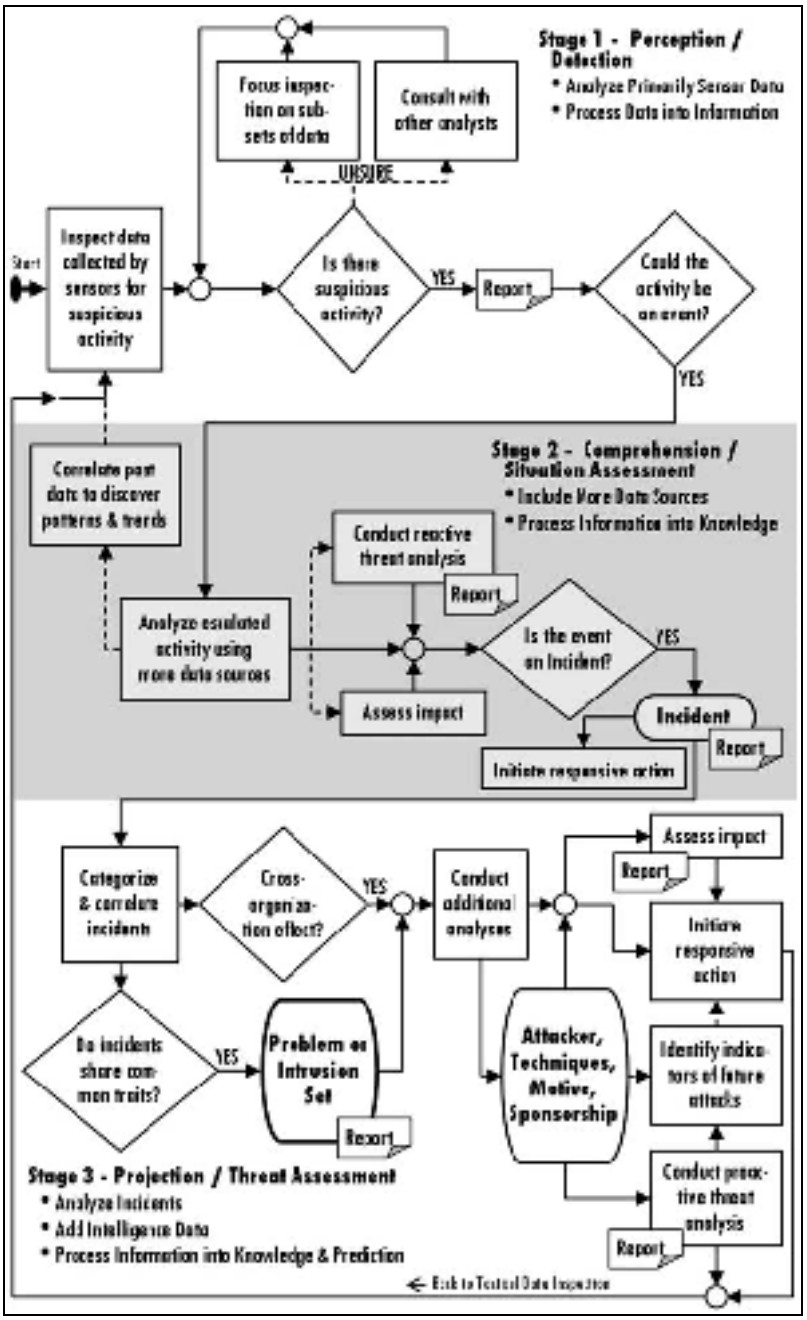

Figure-A I-1 Processus de détection des intrusions
Tirée de D'Amico et Whitley (2008, p. 31)

ANNEXE II

Liste des classifications des alertes

Tableau-A II-1 Classifications des alertes pour Snort 2.9.0

Tirée de The Snort Project (2011, p. 142)

Classtype	Description	Priority
attempted-admin	Attempted Administrator Privilege Gain	high
attempted-user	Attempted User Privilege Gain	high
kickass-porn	SCORE! Get the lotion!	high
policy-violation	Potential Corporate Privacy Violation	high
shellcode-detect	Executable code was detected	high
successful-admin	Successful Administrator Privilege Gain	high
successful-user	Successful User Privilege Gain	high
trojan-activity	A Network Trojan was detected	high
unsuccessful-user	Unsuccessful User Privilege Gain	high
web-application-attack	Web Application Attack	high
attempted-dos	Attempted Denial of Service	medium
attempted-recon	Attempted Information Leak	medium
bad-unknown	Potentially Bad Traffic	medium
default-login-attempt	Attempt to login by a default username and password	medium
denial-of-service	Detection of a Denial of Service Attack	medium
misc-attack	Misc Attack	medium
non-standard-protocol	Detection of a non-standard protocol or event	medium
rpc-portmap-decode	Decode of an RPC Query	medium
successful-dos	Denial of Service	medium
successful-recon-largescale	Large Scale Information Leak	medium
successful-recon-limited	Information Leak	medium
suspicious-filename-detect	A suspicious filename was detected	medium
suspicious-login	An attempted login using a suspicious user-name was detected	medium
system-call-detect	A system call was detected	medium
unusual-client-port-connection	A client was using an unusual port	medium
web-application-activity	Access to a potentially vulnerable web application	medium
icmp-event	Generic ICMP event	low
misc-activity	Misc activity	low
network-scan	Detection of a Network Scan	low
not-suspicious	Not Suspicious Traffic	low
protocol-command-decode	Generic Protocol Command Decode	low
string-detect	A suspicious string was detected	low
unknown	Unknown Traffic	low
tcp-connection	A TCP connection was detected	very low

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Abdullah, Kulsoom, Chris Lee, Gregory Conti, John A. Copeland et John Stasko. 2005. « IDS RainStorm: Visualizing IDS Alarms ». In *Proceedings of the IEEE Workshops on Visualization for Computer Security* (Oct. 26 2005). p. 1-10. Minneapolis: IEEE Computer Society.

Axelsson, Stefan, et David Sands. 2005. *Understanding Intrusion Detection Through Visualization*. Coll. « Advances in Information Security ». New York: Springer-Verlag Inc., 165 p.

Bachmaier, Christian. 2007. « A Radial Adaptation of the Sugiyama Framework for Visualizing Hierarchical Information ». *IEEE Transactions on Visualization and Computer Graphics,* vol. 13, n^{o} 3, p. 583-594.

Bejtlich, Richard. 2004. *The Tao Of Network Security Monitoring: Beyond Intrusion Detection*. Boston: Addison-Wesley Professional, 838 p.

Card, Stuart K., Jock D. Mackinlay et Ben Shneiderman. 1999. *Readings in Information Visualization: Using Vision to Think*. San Francisco: Morgan Kaufmann, 712 p.

Chabot, Christian. 2009. « Demystifying Visual Analytics ». *IEEE Computer Graphics and Applications,* vol. 29, n^{o} 2, p. 84-87.

Cole, Eric. 2009. *Network Security Bible*. Indianapolis: Wiley Publishing, 936 p.

Conti, Greg, Kulsoom Abdullah, Julian Grizzard, John Stasko, John A. Copeland, Mustaque Ahamad, Henry L. Owen et Chris Lee. 2006. « Countering Security Information Overload Through Alert and Packet Visualization ». *IEEE Computer Graphics and Applications,* vol. 26, n^{o} 2, p. 60-70.

Cormen, Tomas H., Charles E. Leiserson, Ronald L. Rivers et Clifford Stein (1180). 2001. *Introduction to Algorithms*, 2nd Edition. Cambridge: The MIT Press.

D'Amico, Anita, John R. Goodall, Daniel R. Tesone et Jason K. Kopylec. 2007. « Visual Discovery in Computer Network Defense ». *Computer Graphics and Applications, IEEE,* vol. 27, n^{o} 5, p. 20-27.

D'Amico, Anita, et Kristen Whitley. 2008. « The Real Work of Computer Network Defense Analysts : The Analysis Roles and Processes that Transform Network Data into Security Situation Awareness ». In *Proceedings of the 4th Workshop on Visualization for Computer Security* (Oct. 28 - Nov. 1 2007). p. 19-37. Sacramento: Springer Berlin Heidelberg.

Dickerson, Matthew, David Eppstein, Michael T. Goodrich et Jeremy Y. Meng. 2005. « Confluent Drawings: Visualizing Non-Planar Diagrams in a Planar Way ». *Graph Algorithms and Applications,* vol. 9, n° 1, p. 31-52.

Feng, Zhang, Zhou Shijie, Qin Zhiguang et Liu Jinde. 2003. « Honeypot: a Supplemented Active Defense System for Network Security ». In *Proceedings of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies* (Aug. 27-29 2003). p. 231-235. Chengdu, China: Institute of Electrical & Electronics Engineering.

Foresti, Stefano, James Agutter, Yarden Livnat, Shaun Moon et Robert Erbacher. 2006. « Visual Correlation of Network Alerts ». *Computer Graphics and Applications, IEEE,* vol. 26, n° 2, p. 48-59.

Goodall, John R., Wayne G. Lutters et Anita Komlodi. 2004. « The Work of Intrusion Detection: Rethinking the Role of Security Analysts ». In *Proceedings of the 13th Americas Conference on Information Systems* (Aug. 6-8 2004). p. 1421–1427. New York: AIS Press.

Goodall, John R., Wayne G. Lutters et Anita Komlodi. 2009. « Developing Expertise for Network Intrusion Detection ». *Information Technology & People,* vol. 22, n° 2, p. 92-108.

Goodall, John R., A. Ant Ozok, Wayne G. Lutters, Penny Rheingans et Anita Komlodi. 2005. « A User-Centered Approach to Visualizing Network Traffic for Intrusion Detection ». In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems* (Apr. 2-7 2005). p. 1403-1406. Portland: ACM.

Goodall, John R., et Daniel R. Tesone. 2009. « Visual Analytics for Network Flow Analysis ». In *Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security* (March 3-4 2009). p. 199-204. Washington: IEEE Computer Society.

Günter, Andreas, Rudolf Kruse, Bernd Neumann, Daniel Keim, Christian Panse, Jörn Schneidewind, Mike Sips, Ming Hao et Umeshwar Dayal. 2003. « Pushing the Limit in Visual Data Exploration: Techniques and Applications ». In *Proceedings of the 26th Annual German Conference on AI - KI 2003: Advances in Artificial Intelligence* (Sept. 15-18 2003). p. 37-51. Hamburg, Germany: Springer Berlin / Heidelberg.

Holten, Danny, et Jarke J. Van Wijk. 2009. « Force-Directed Edge Bundling for Graph Visualization ». *Computer Graphics Forum,* vol. 28, n° 3, p. 983-990.

Kabay, Mitch. 1998. « Perils of Rushing to Market ». *The Risks Digest - Forum on Risks to the Public in Computers and Related Systems*. ACM Committee on Computers and Public Policy. Vol. 19, n° 91.

Koike, Hideki, et Kazuhiro Ohno. 2004. « SnortView: Visualization System of Snort Logs ». In *Proceedings of the 2004 ACM workshop on Visualization and Data Mining for Computer Security* (Oct. 29, 2004). p. 143-147. Washington DC: ACM.

Koike, Hideki, Kazuhiro Ohno et Kanba Koizumi. 2005. « Visualizing Cyber Attacks using IP Matrix ». In *Proceedings of the IEEE Workshops on Visualization for Computer Security* (Oct. 26 2005). p. 91-98. Minneapolis: IEEE Computer Society.

Komlodi, Anita, Penny Rheingans, Ayachit Utkarsha, John R. Goodall et Joshi Amit. 2005. « A User-Centered Look at Glyph-Based Security Visualization ». In *Proceedings of the 2nd International Workshop on Visualization for Computer Security* (Oct. 26 2005). p. 21-28. Minneapolis: IEEE Computer Society.

Kruegel, Christopher, Fredrik Valeur et Giovanni Vigna. 2005. *Intrusion Detection and Correlation : Challenges and Solutions*. Coll. « Advances in Information Security », Vol. 14. Boston: Springer Science + Business Media, Inc., 122 p.

Lee, Christopher P., et John A. Copeland. 2006. « Flowtag: a Collaborative Attack-Analysis, Reporting, and Sharing Tool for Security Researchers ». In *Proceedings of the 3rd international workshop on Visualization for computer security* (Nov. 3 2006). p. 103-108. Alexandria: ACM.

Lex, Alexander, Marc Streit, Christian Partl et Dieter Schmalstieg. 2010. « Comparative Analysis of Multidimensional, Quantitative Data ». *IEEE Transactions on Visualization and Computer Graphics,* vol. 16, n^{o} 6, p. 1027-1035.

Mansmann, Florian, Fabian Fischer, Daniel A. Keim et Stephen C. North. 2009. « Visual Support for Analyzing Network Traffic and Intrusion Detection Events Using TreeMap and Graph Representations ». In *Proceedings of the 3rd Symposium on Computer Human Interaction for the Management of Information Technology* (Nov. 7–8, 2009). p. 19-28. Baltimore: ACM.

Marty, Raffael. 2008. *Applied Security Visualization*. Boston: Addison-Wesley, 552 p.

MaxMind Inc. 2011. « GeoIP MaxMind C# API ». En ligne. <http://www.maxmind.com/app/csharp>.

Moore, D., V. Paxson, S. Savage, C. Shannon, S. Staniford et N. Weaver. 2003. « The spread of the sapphire/slammer worm ». En ligne. <http://www.caida.org/publications/papers/2003/sapphire/sapphire.html>.

Musa, Shahrulniza, et David J. Parish. 2008. « Using Time Series 3D AlertGraph and False Alert Classification to Analyse Snort Alerts ». In *Proceedings of the 5th international workshop on Visualization for Computer Security* (Sept. 15 2008). p. 169-180. Cambridge: Springer-Verlag.

Pupyrev, Sergey, Lev Nachmanson et Michael Kaufmann. 2011. « Improving Layered Graph Layouts with Edge Bundling ». In *Proceedings of the 18th International Conference on Graph Drawing* (Sept. 21-24 2010). p. 329-340. Konstanz, Germany: Springer-Verlag.

Shneiderman, Ben. 1996. « The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations ». In *Proceedings of the 1996 IEEE Symposium on Visual Languages* (Sept. 3-6 1996). p. 336 - 343. Boulder: IEEE Computer Society.

Sourcefire Inc. 2011. « Snort Home Page ». En ligne. <http://www.snort.org/>.

Spence, Robert. 2007. *Information Visualization: Design for Interaction*, 2e éd. Coll. « Pearson Education ». Harlow: Prentice-Hall, 304 p.

Spitzner, Lance. 2003. « Honeypots, Definitions and Value of Honeypots ». En ligne. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1236295>.

Tamassia, Roberto, Bernardo Palazzi et Charalampos Papamanthou. 2009. « Graph Drawing for Security Visualization ». In *Proceedings of the 16th International Symposium on Graph Drawing* (Sept. 21 - 24 2008). p. 2-13. Heraklion, Grèce: Springer-Verlag.

The Snort Project. 2011. « SNORT Users Manual ». (12 juillet 2011), p. 12. <http://www.snort.org/assets/166/snort_manual.pdf>.

The Wombat Project. 2011. « Worldwide Observatory of Malicious Behaviors and Attack Threats ». En ligne. <http://www.wombat-project.eu/>.

Tufte, Edward R. 1986. *The Visual Display of Quantitative Information*. Cheshire: Graphics Press, 197 p.

United States. 2010. « Crimes and Criminal Procedure : Title 18, Part I, Chapter 47, § 1030 ». En ligne. <http://www.law.cornell.edu/uscode/18>.

van Ham, Frank, Martin Wattenberg et Fernanda B. Viegas. 2009. « Mapping Text with Phrase Nets ». *IEEE Transactions on Visualization and Computer Graphics,* vol. 15, n° 6, p. 1169-1176.

Ware, Colin. 2004. *Information Visualization: Perception for Design*. San Francisco: Morgan Kaufmann Publishers Inc., 464 p.

Werlinger, Rodrigo, Kasia Muldner, Kirstie Hawkey et Konstantin Beznosov. 2010. « Preparation, Detection, and Analysis: The Diagnostic Work of IT Security Incident Response ». *Information Management and Computer Security,* vol. 18, n^{o} 1, p. 26-42.