UNIVERSITE de CAEN/BASSE-NORMANDIE

U.F.R. : SCIENCES
ECOLE DOCTORALE SIMEM

THÈSE

présentée par

Jouni Viinikka

et soutenu

le 24 novembre 2006

en vue de l'obtention du

Doctorat de l'Université de Caen

spécialité : Informatique

(Arrêté du 25 avril 2002)

Traitement de Flux d'Alertes en Détection d'Intrusions avec des Méthodes d'Analyse de Séries Temporelles

---

Membres du jury

Mr Felix Wu, Professeur, University of Califoria Davis (rapporteur)
Mr Marc Dacier, Professeur à Eurecom, Institut Eurecom (rapporteur)
Mr Ludovic Mé, Professeur à Supélec, HDR, Supélec (directeur)
Mr Hervé Debar, HDR, France Telecom (co-directeur)
Mr Philippe Owezarski, Chargé des Recherches, HDR, LAAS-CNRS
Mr François Bourdon, Professeur des Universités, Université de Caen

Remerciements

Cette thèse et ces trois ans et quelques mois m'ont donné beaucoup et je tiens à remercier beaucoup de personnes.

Tout d'abord, je veux remercier mon co-directeur Hervé Debar. C'est lui qui m'a accueilli à France Telecom, d'abord pour mon stage, et après pour cette thèse. Il m'a fait confiance, donné du temps et de la liberté à chercher. En même temps, il m'a guidé pendant toute cette initiation au recherche.

Ludovic Mé, mon directeur, est la deuxième personne envers laquelle je suis très reconnaissant. Toujours disponible, pour m'aider avec les problèmes scientifiques, ainsi qu'administratifs.

Les réunions de thèse, soit à Rennes, soit à Caen, étaient toujours très utiles. Les deux m'ont posé des questions que je n'avais pas attendues, auxquelles je n'avais même pas pensé et auxquelles je n'avais pas de réponses. J'en sortais toujours avec plein de travail, mais aussi encouragé et confiant pour la suite. Je suis très content d'avoir eu le privilège de travailler avec vous.

Je tiens à remercier Marc Dacier et Felix Wu pour avoir accepté de rapporter cette thèse. Marc aussi pour tout ce qu'il a fait pour moi déjà depuis Eurecom, plus particulièrement m'avoir présenté à Hervé.

Je souhaite remercier Philippe Owezarski et François Bourdon de m'avoir fait l'honneur d'être dans mon jury.

L'aide d'Anssi Lehikoinen a été de grande valeur, et j'ai trouvé les discussions sur les principes et les détails de traitement du signal extrêmement utiles. De plus, mes séjours à Kuopio et aux alentours m'ont permis de recharger les batteries. En parlant de traitement du signal, je tiens à remercier aussi Mika Tarvainen et Renaud Séguier.

Les bureaux de MAPS/NSS/SPR à France Telecom ont été un bon environnement pour travailler, grâce à les gens qui les occupaient, et je vous remercie tous pour cela. Je voudrais plus particulièrement remercier Vincent, Yohann et Diala. Vincent était toujours disponible, que ce soit pour le prêt d'un serveur, comme pour les déménagements (du bureau ou de l'appartement). On peut toujours compter sur lui. Avec Yohann et Diala j'ai passé beaucoup des bon moments, au travail, ainsi qu'en dehors.

Ce sont ces bons moments hors du travail qui m'amènent aux amis qui ne travaillent pas à France Telecom. Benoît, merci pour m'avoir fait découvrir tous ces chemins pour courir. Vanessa et Ali, merci pour tous les dîners chez vous et ailleurs. Et Claire, quand même. Chaque bon moment et chaque rire était essentiel pour ce travail, au moins indirectement.

Je veux aussi remercier ma mère, ma soeur et la reste de ma famille qui sont loin. La base sur laquelle tout cela s'est construit, vient de chez vous.

La dernière mais la plus importante, Elodie, je te remercie de tout mon coeur. Je tiens déjà à te remercier pour le grand travail effectué pour que la version française de cette thèse voit le jour. Mais beaucoup plus important, je suis très reconnaissant pour

ton soutien et ton encouragement inconditionnel. Je crois que c'est toi qui m'a fait le plus confiance pendant cette thèse, et c'est aussi toi qui a subi les horaires flexibles (bien souvent seulement dans un sens) d'un thésard.

Pour finir, je souhaite remercier les organisations ayant soutenu ce travail financièrement : France Telecom, Supélec, l'Association Nationale de la Recherche Technique, la Fondation Finlandaise pour la Culture et Nokia Foundation.

Table des matières

Table des figures

Liste des tableaux

Abbreviations

AI	artificial intelligence
apm	alerts per minute
aph	alerts per hour
SNMP	Simple Network Management Protocol
ICMP	Internet Control Message Protocol
HTTP	Hyper Text Transfer Protocol
WWW	World Wide Web
IDS	intrusion detection system
NIDS	network intrusion detection system
HIDS	host intrusion detection system
iid	independently and identically distributed
OS	operating system
OD flow	origin-destination flow
TCP/IP	Transport Control Protocol / Internet Protocol
IM	instant messaging
IRC	Internet Relay Chat
P2P	peer-to-peer

Notations

ν	the degrees of freedom
μ_y	the average value of y
$\Phi(x)$	the cumulative distribution function (cdf)
$\Phi^{-1}(x)$	the inverse cumulative distribution function
σ_y	the standard deviation of y
σ_y^2	the variance of y
θ_t	state vector
θ_t	state at the discrete time instant t (state space notation)
$\hat{\theta}_t$	the estimate of θ_t
$\tilde{\theta}_t$	the estimation error of the parameter θ_t
$(1-\lambda)$	EWMA smoothing factor
AR (p)	autoregressive model of order p
ARMA (p,q)	autoregressive moving average model of orders p and q
a_j	stationary AR coefficient $j = 1, \ldots, p$
a_t^j	time-dependent AR coefficient $j = 1, \ldots, p$, t is discrete time instant
b_k	stationary MA coefficient $k = 1, \ldots, q$
b_t^k	time-dependent MA coefficient $k = 1, \ldots, q$, t is discrete time instant
C_y	the covariance matrix of y
C_{xy}	the cross covariance matrix of y
f	frequency
f_s	sampling frequency
e_t	observation noise
E $[x]$	expected value of x
E $[x\|y]$	expected value of x given y
H	observation matrice
I	identity matrice
p	AR model order
P $[x]$	probability density function of x
P $[x,y]$	joint probability density of x and y
P $[x\|y]$	conditional probability density of x given y
q	MA model order
w_t	state noise
y	observation
$\bar{y}$	sample mean
y_t	observation at discrete time instant t
t	time instant
t_s	sampling interval

Chapitre 1

Introduction : Détection d'Intrusion et Gestion d'Alertes

Un des principes de base dans la sécurité des systèmes d'information est que les mesures de prévention s'échoueront. La sécurité parfaite n'existe pas, ou au moins impliquerait un coût trop élevé [DDW99] et tout logiciel a des failles [LBMC94]. Même si les failles n'existaient pas, les humains qui utilisent les systèmes, feraient des erreurs. Les logiciels de plus en plus complèxes et l'interaction croissante entre les composants logiciels rendent le problème encore plus difficile. La connectivité ajoutée par l'Internet expose ses nombreuses failles à des adveraires beaucoup plus nombreux qu'avant.

En conséquence, les systèmes subissent des intrusions. Les systèmes de détection d'intrusions *(intrusion detection system, IDS)* ont été concus pour surveiller les systèmes d'informations (SI) et découvrir les violations de la politique de sécurité automatiquement, et en informer l'*opérateur*.

Dans ce chapitre, nous présenterons brièvement la détection d'intrusions, les méthodes, les techniques utilisées, les problèmes et quelques solutions proposées. À la fin du chapitre nous définirons le but de cette thèse et son organisation.

1.1 Détection d'Intrusions

Le but de détection d'intrusions est de repérer les violations de la politique de sécurité du système surveillé. Au début, ceci était un processus manuel, consistant à inspecter des audits système. La quantité des données à analyser a augmenté rapidement, et les IDS ont été concus pour automatiser ce processus. Anderson [And80] a été le premier à proposer l'automatisation de la détection. Depuis son rapport, plusieurs IDS ont été conçus dans les universités, l'industrie et par les militaires, pricipalements aux États-Unis. Pour une discussion plus détaillée, le lecteur est invité à se reporter au livre de Bace [Bac00], à la taxinomie de Debar et al. [DDW99], ou à la taxinomie d'Axelsson [Axe00a].

Anderson a divisé les attaquants en *externe* et *interne*, selon leurs privilèges d'accès. Un intrus externe est quelqu'un ne disposant d'aucun accès, ni physique, ni logique, au système, ou éventuellement disposant d'un accès réseau, mais sans accès au système ciblé. Pour lui, le premier pas est d'acquérir un accès à la cible. Une fois réussi, l'intrus externe est considéré comme un intru interne par Anderson. McHugh a remarqué [McH01, Sect. 2] qu'une certaine classe d'attaques, comme les attaques par *déni de service(Denial of Service, DoS)*, ne nécessitent pas que l'attaquant de devienne un utilisateur légitime du

système.

Les intrus internes étaient divisés en trois classes, les *usurpateurs*, les *utilisateurs légitimes*, et les *utilisateurs clandestins*. Les usurpateurs ont pris l'identité d'un utilisateur légitime. Ils sont soit des utilisateurs internes et légitimes, ou des intrus externes qui ont pénétré les contrôles d'accès. Les utilisateurs légitimes comme les intrus abusent des privilèges qui leur ont été accordés dans le cadre de l'accomplissement normal de leurs tâches. Un utilisateur clandestin a souvent de bonnes connaissances en informatique, des informations concernant du système ciblé, et eventuellement même informations concernant l'IDS. Ils peuvent utiliser leurs connaissances pour éluder la détection.

1.1.1 Architecture du Système

Nous utilisons le modèle du groupe "Intrusion Detection Working Group" (IDWG)[1] de l'Internet Engineering Task Force (IETF) pour décrire un IDS. La Figure 1.1 montre les différentes composantes logicielles ou matérielles sous forme de rectangles, les rôles des humains sous forme d'ellipses et les flux de données sous forme de flèches.

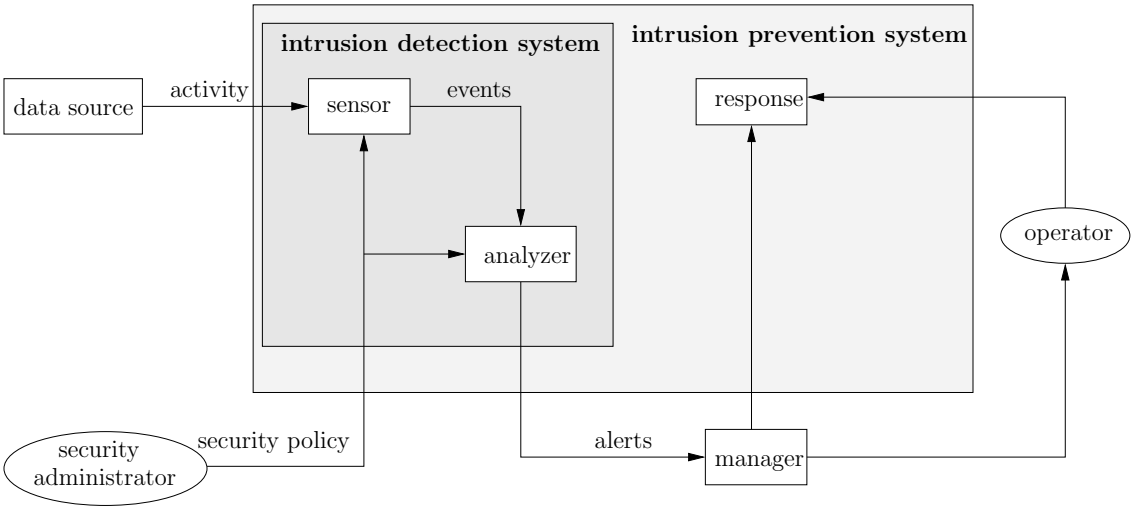


FIG. 1.1 – L'architecture d'un IDS/IPS

Le système de détection d'intrusion obtient les observations de l'activité du système surveillé par une *source de données (data source)*. La *sonde (sensor)* capture les données, les transforme en *événements* qui sont passés à l'*analyseur (analyzer)*. Un IDS pourrait utiliser plusieurs sources de données, potentiellement hétérogènes. En pratique, ceci est rare.

L'analyseur sépare les événements entre sains et malveillants et génère une alerte dans ce dernier cas. Il existe des *méthodes de détection* différentes et plusieurs algorithmes pour chacune des méthodes. En plus, l'analyse est influencée par la politique de sécurité, par exemple en définissant les valeurs des composantes différentes d'un SI et des seuils des plusieurs sortes. Pour faciliter l'intéropérabilité, les alertes sont encodées en eXtensible Markup Language (XML) dans le format Intrusion Detection Message Exchange Format

[1] `http://www.ietf.org/html.charters/OLD/idwg-charter.html`, visité 2006-07-18

(IDMEF) [DCF06]. La politique de sécurité est définie par un humain, l'*administrateur de sécurité*.

Les alertes sont envoyées vers le *manager*, ce qui permet à un humain de les visualiser. L'humain qui surveille la sécurité du système s'appelle l'*opérateur (operator)*. Le manager peut aussi réagir automatiquement aux alertes avec une composante de *réponse (response)*. Si les contre-mesures sont appliquées automatiquement, le système s'appelle *système de prévention d'intrusions (intrusion prevention system, IPS)*[2] au lieu de système de détection d'intrusions.

Dans les sous-chapitres suivants, nous regarderons quelques composantes plus en détail. Nous commencerons par la présentation des sources des données, et continuerons avec les méthodes d'analyse. Nous discuterons ensuite des effetsde la source de données et de la méthode d'analyse sur les propriétés des alertes et introduirons la division classique des alertes selon l'exactitude des alertes. Nous finirons cette introduction par le traitement d'alertes. C'est une fonction qui peut être placée au niveau du manager ou comme un IDS de deuxième niveau.

1.1.2 Sources de Données

Les IDS peuvent être divisés selon le type de la source de données, en quatre classes suivantes : les IDS système *(host-based)*, les IDS réseau *(network-based)*, les IDS applicatifs *(application-based)* et les IDS de deuxième niveau *(alert-based)* qui utilisent les alertes comme une source de données. Les deux premières sont les classes les plus utilisées, par exemple dans [Bac00, McH01] et dans la partie introduction de plusieurs articles sur la détection d'intrusions. Axelsson a groupé les logs (ou audit) applicatifs dans les audits système [Axe00a, p.10]. Les quatre classes ont toutes été utilisées au moins dans [DDW99, ACF$^+$00]. D'un côté Axelsson a remarqué que le nombre d'IDS système était élévé par rapport au nombre d'IDS réseau dans sa taxinomie [Axe00a, p.13]. De l'autre côté, en 1999 Debar et al. [DDW99, p.13] ont écrit que la tendance était vers les IDS réseau et ont expliqué le grand nombre des IDS système par le fait que historiquement la recherche était centrée autour des IDS système. Nous décrivons brièvement chaque classe et ses atouts et faiblesses.

Audit Système

Les sondes système utilisent des données d'audit dont l'origine est la machine surveillée. La plupart des premiers IDS utilisent l'audit système comme leur source de données. Ces sources iront sur une machine UNIX du système de fichiers `proc`, les outils système commme `ps`, `lsof` et `syslog` jusqu'à l'audit de sécurité C2 défini par le US Department of Defense [DoD85]. Même si ces sources sont différentes entre elles, nous pouvons faire les généralisations grossières suivantes :

Atouts L'audit système est souvent riche en informations. Les outils peuvent accéder directement à la mémoire noyau et récupérer les informations comme par exemple le propriétaire du processus, l'utilisation des ressources, les noms des applications et l'information concernant les appels système. En plus, les audits système sont

[2] Premièrement, nous considérons le terme commercial IPS mal adapté. La signification traditionelle de *prévention* est les mésures passives de protection, en sécurité informatique et en sécurité plus générale. En plus, la sûreté de détection devrait être un prérequis des contre-mesures automatiques. Jusqu'à aujourd'hui nous ne sommes pas au courant de l'évaluation rigoureuse des IDS montrant une telle sûreté de détection.

disponibles sur la quasi-totalité des systèmes, allant des mainframes à l'équipement réseau. Par exemple, le gouvernement américain exige que toutes les acquisitions d'ordinateurs supportent l'audit de sécurité C2 qui a été utilisée par de nombreux IDS. Un avantage important est la possibilité de détecter les intrus avec un accès physique à la machine. Ceci peut être difficile, voire impossible, à partir des autres sources de données. Un deuxième avantage est le fait que les sondes système ne sont pas aveuglées par le chiffrement du trafic réseau. Même les programmes malveillants et polymorphiques, difficile à détecter en observant le trafic réseau, interagit avec le système laissant des traces dans les audits système[3].

Faiblesses Les logs de système peuvent être volumineux et leur création peut réduire la performance du système d'une façon significative en termes d'utilisation du CPU *(Central Processing Unit)* et d'espace de stockage. Les différentes implémentations de l'audit de sécurité C2 utilisent des formats et des interfaces hétérogènes. Ceci implique que les IDS sont dépendants de la plateforme sur laquelle ils fonctionnent. Les sources de données système sont, par définition, limitées dans la visibilité à une seule machine. Cette limite peut être contournée avec un traitement central des alertes issues de plusieurs sondes. Par contre, le traitement central est un problème difficile et non-résolu, et le volume des logs rende le problème encore plus difficile.

Audit Réseau

Les sondes réseau utilisent des données d'audit obtenus à partir du réseau surveillé. À ce jour la plupart des IDS commerciaux sont des IDS réseau, et qui utilisent éventuellement d'autres sources en plus. Le trafic réseau peut être observé directement avec des sniffers réseau ou indirectement via des équipements réseau en utilisant par exemple SNMP *(Simple Network Management Protocl)* MIB *(Management Information Base)* ou netflow de Cisco. Bro [Pax99], Snort [Roe99] et PAYL [WS04] serviront d'exemples pour les IDS réseau.

Atouts Les IDS réseau peuvent surveiller plusieurs hôtes simultanément. Ceci résout quelques problèmes associés aux IDS système. Le nombre d'IDS à déployer est réduit significativement, et cette approche n'encombre pas les hôtes surveillés car les données sont capturées et analysées sur une machine dédiée.

L'utilisation répandue du protocole TCP/IP fournit un degré d'homogénéité bienvenu mais limité[4]. De plus, et en fonction de son placement, l'IDS peut détecter des attaques ayant un impact étendu sur un réseau. Il existe aussi des attaque spécifiques aux réseaux, comme les attaques par déni de service et la propagation des vers informatiques. La détection rapide de ces attaques est difficile à partir des sources de données système. La rapidité de détection est très important pour arrêter ou réduire l'impact de ce type d'attaques.

Étant donné qu'une machine dédiée surveille le réseau, aucun logiciel n'a besoin d'être installé sur les hôtes surveillés. Dans un contexte où le contrôle sur les hôtes surveillés est limité, ceci est un avantage important. Les fournisseurs d'accès internet (FAI) ou les fournisseurs d'infogérance de sécurité *(Managed Security Service Provider, MSSP)* sont un exemple, et les configurations certifiées selon les critères

[3]Sauf si cachés par rootkit.

[4]Les piles protocoles sont implémentées différemment et ceci ouvre des posssibilités aux attaques d'évasion et insertion [PN98]

communs *(Common Criteria, CC)*[5], en sont un deuxième. Les modifications autorisées dans les cibles d'évaluation sont limitées pour ne pas perdre le certificat. Dans un tel environnement, une sonde réseau est mieux adaptée que les sondes système pour la surveillance.

Faiblesses Le placement d'une sonde réseau peut être difficile dans un réseau commuté pour garantir la visibilité requise de la sonde. Avec les vitesses de réseaux actuels, la capture des paquets est déjà un défi sans parler de l'analyse. Même si la sonde est bien placée, le chiffrement du trafic empêche la plupart de sondes réseaux de marcher. Le chiffrement rend au moins le contenu des paquets *(payload)* illisible, éventuellement aussi les en-têtes.

À partir de l'audit réseau il est difficile de tracer l'attaquant, alors que l'audit système contient souvent au moins l'identité sous laquelle l'attaque a été effectuée.

Les IDS réseau peuvent détecter certaines attaques par déni de service, mais en même temps elles sont elles mêmes vulnérables à ces attaques. Les vecteurs d'attaque sont le système d'exploitation de la machine sur laquelle la sonde est installée, et la sonde elle même. Les sondes système ne sont pas inattaquables, mais la cible est plus petite.

Audit Applicatif

Les IDS applicatifs surveillent une application au lieu d'un hôte ou l'ensemble d'un réseau. Néanmois, cette application peut être distribuée sur un cluster de machines. Les sources de données sont souvent des logs d'applications comme les serveurs web, mail ou base de données. Dans cette categorie on trouve, par exemple, les IDS comme DIDAFIT [LLT02], WebWatcher [ADD00] et WebAnalyzer [TDMD04] pour les serveurs web. Dans [TMM05] Totel et al. présentent l'idée d'utiliser la diversité de COTS *(components-off-the-self)* pour détecter les intrusions aux applications. Ils décrivent une réalisation de cette idée avec les serveurs web.

Atouts Les logs applicatifs sont souvent de sources de données plus correctes, completes et concises que les audits système et les audits réseau. Ces deux derniers sont traités avant que l'IDS comprenne l'interaction avec la cible. Le traitement est basé sur les spécifications des protocoles ou des APIs *(Application Programming Interface)*. Ces spécifications peuvent être interprêtées d'une façon différente par les développeurs d'un OS ou d'une application et les developpeurs d'IDS. Les différences impliquent des incohérences entre l'état réel de système surveillé et l'état perçu par l'IDS. Les logs applicatifs sont une source plus correcte, parce que l'IDS perçoit l'état de système comme rapporté par le système lui-même. Les logs sont plus complets car souvent ils contiennent aussi les codes de retour et des messages d'erreur, qui fournissent un diagnostic de l'impact de l'action observée. Ils sont plus concis car une ligne de log peut résumer l'effet de plusieurs paquets réseau ou appels système.

Les attaques telles que le cross-site scripting (XSS) ou l'injection SQL sont des problèmes au niveau applicatif, difficile à détecter au niveau réseau ou hôte. En plus, les logs applicatifs ne sont pas sensibles au chiffrement du trafic.

Faiblesses Les attaques de bas niveau ne sont pas visibles dans les logs applicatifs et une attaque réussie peut empêcher l'application d'écrire la ligne de log nécessaire pour la détection. En plus, les logs ont souvent été conçus pour les raisons autres

[5] `http://niap.nist.gov/cc-scheme/index.html`, visité 2006-07-18

que la détection d'intrusions. Pour cette raison, il n'y a pas toujours suffisamment d'information dans les logs pour la détection.

Audit Sonde

Particulièrement dans les déploiements à grande échelle, les IDS de bas niveau peuvent générer des alertes en grandes quantités. Pour pouvoir tolérer le volume important d'alertes les IDS de plus haut niveau peuvent être utilisés pour aggréger et corréler des alertes. Les données d'entrée pour les systèmes de corrélation sont les alertes générées par les autres IDS, et éventuellement des informations contextuelles comme la topologie et les vulnérabilités du système surveillé. Dans ce cas, les IDS de bas niveau ont un rôle de sonde (cf. Fig. 1.1), et le moteur de corrélation fonctionne comme l'analyseur. Dans cette thèse nous travaillons dans un tel contexte et utilisons la terme sonde pour les IDS de bas niveau.

Atouts En utilisant l'information additionnelle, indisponible dans la source de données de l'IDS de bas niveau, les moteurs de corrélation peuvent améliorer la qualité des alertes, par exemple en identifiant des *faux positifs*, les alertes générées sans une cause liée à la sécurité. Il est possible de complémenter les capacités de détection des sondes hétérogènes, comme une sonde système et une sonde réseau, en combinant ensemble les alertes générées par ces sondes. Les sondes distribuées permettent la détection d'attaques à grande échelle, éventuellement touchant plusieurs réseaux.

Faiblesses Les IDS de bas niveau peuvent générer une quantité significative d'alertes, et les moteurs de corrélation utilisant des algorithmes complèxes ont besoin beaucoup de ressources. En plus, la plupart d'alertes peuvent être fausses ou non-pertinentes, ce qui rend le problème de corrélation difficile. Les données d'entrée hétérogènes c'est à dire les formats d'alertes et les noms d'attaques non-standardisés ajoutent à la difficulté de corrélation. Il existe des efforts de standardisation pour les formats d'alertes et pour la dénomination des vulnérabilités. Le format d'alerte IDMEF est assez mûr et commence être accepté et utilisé par la communauté sécurité et par les entreprises. CVE *(Common Vulnerabilities and Exposures)*[6] est un schema de dénomination pour l'identification des vulnérabilités. Pour les logiciels malveillants *(malware)* et les virus, CME *(Common Malware Enumeration)*[7] est un effort plus récent pour l'identification des virus.

1.1.3 Méthodes de Détection

Une autre façon de classer les systèmes de détection d'intrusions est basé sur la méthode de détection. La division la plus utilisée est entre détection de mauvaise utilisation *(misuse)* (aussi détection par signatures, par scénario ou détection morphosciente) et détection comportementale (aussi détection d'anomalies). Les autres divisions existent, par exemple dans [Axe99] Axelsson a utilisé un troisième type, détection par spécification *(specification-based)*. Il a regroupé la détection de mauvaise utilisation et la détection par spécification ensemble dans la catégorie détection par politique *(policy-based)*. Cependant, dans son taxinomie un an plus tard [Axe00a], il utilise les deux classes habituelles. Pour nos besoins, la division entre détection comportementale et de mauvaise utilisation suffit.

[6] `http://cve.mitre.org`, visité 2006-07-18
[7] `http://cme.mitre.org`, visité 2006-07-18

Détection comportementale

La détection comportementale utilise un modèle du comportement normal. Tout comportement qui ne fait pas partie du modèle est considéré suspect. Le modèle est construit à partir des observations de comportement normal d'un système, une application ou un utilisateur. Typiquement le modèle est basé sur les statistiques, les systèmes expert, les systèms d'immunologie ou les différentes techniques d'intelligence artificielle (IA) : réseaux de neurones (RN), algorithmes génetiques.

suspect

Parmi les approches statistiques, les travaux de SRI International sur IDES, NIDES et EMERALD [JV91, JV93, NP99] sont peut-être les mieux connus. Les réseaux de neurones sont souvent mentionnés comme une méthode prometteuse, mais il existe peu de recherche concernant leur utilisation dans la détection d'intrusions. Debar et al. [DBS92] ont utilisé RN pour characteriser le comportement d'utilisateur. Dans [TC97] les réseaux de neurones ont été utilisé détecter les services réseau anormaux en observant les connections TCP. Plus tard, les cartes auto organisatrice *(Self-Organizing Map, SOM)* ont été utilisé dans la détection d'intrusions, cf. par exemple [ROT03]. Forrest et al. ont utilisé l'approche immunologique [FHSL96], et Mé a analysé l'audit système avec des algorithmes génetiques. Lee et al. ont utilisé une approche systématique pour construire des classificateurs avec les méthodes de fouille de données *(data mining)*. Ils ont créé des classificateurs pour détection comportementale et détection de mauvaise utilisation [LS98, LSM98].

Atouts Étant donné que l'analyse comportementale détecte ce qui n'est pas normal, elle a le potentiel de détecter les attaques nouvelles et l'abus par les utilisateurs légitimes. Certains algorithmes peuvent s'adapter au comportement observé, et au moins en principe ces approches peuvent être plus facile à maintenir que les approches utilisant la détection de mauvaise utilisation.

Faiblesses Très souvent, il est difficile de définir ce qui est normal et les systèmes de détection des anomalies peuvent avoir un taux de faux positifs important. Les alertes issues par la les sondes de détection d'anomalies contient typiquement moins de diagnostique sur la cause d'alerte que les alertes issues par l'analyse par signatures. Ceci parce que les alertes ne peuvent pas être associées aux attaques spécifiques.

Détection morphosciente

Les sondes *morphoscientes* ont un modèle de comportement malveillant. C'est à dire que tout ce qui n'est pas reconnu comme malveillant, est considéré normal. On pourrait dire que ceci est une sorte de politique autorisé-par-défaut *(default allow)*. Le modèle contient des *descriptions d'attaques* qui encodent la connaissance de la façon dont une attaque se manifeste dans la source de données. Les techniques utilisées sont, par exemple, l'analyse par signatures, les systèmes expert et l'analyse par transition d'état *(state transition)*.

L'analyse par signatures encode la connaissance en signatures, un format qui peut être utilisé facilement, souvent avec les méthodes de reconnaissance de motifs *(pattern matching)*. L'analyse par signature est efficace et beaucoup d'IDS commerciaux utilisent cette technique. Par exemple, Snort [Roe99] est une sonde qui utilise l'analyse par signatures.

Les systèmes experts encodent la connaissance des attaques avec des règles si-alors *(if-then rules)*. La partie «si» contient les prérequis d'une attaque, et la partie «alors» contient les actions menés si la partie «si» est vraie. La détection est plus compliquée parce que les événements sont transformés en *faits*. Après, les faits sont evalués selon la

connaissance encodée dans le système. P-BEST [LP99] est un exemple de système expert et utilisé pour la partie morphosciente dans IDES et NIDES.

Les techniques de l'analyse de transition d'état contient des approches basées sur les réseaux de Petri *(Petri-net)* et des techniques de l'analyse de transition d'état *(state-transition analysis)*. IDIOT [KS94] utilise une adaptation des réseaux de Petri colorés pour représenter la connaissance d'attaques avec des graphes spécialisés. Comme avec les systèmes expert, l'encodage est plus abstrait que dans l'analyse par signatures. L'analyse des transitions d'état utilise des machines à état fini *(finite state machine)* pour modéliser les intrusions. Dans ce formalisme, une intrusion est une séquence des actions menées par l'attaquant. Cette séquence amène la cible de l'état initiale à l'état compromis. L'IDS vérifie pour chaque action si un des graphes peut changer l'état. Si un des graphes est amené à l'état compromis, une alerte est issue. La famille STAT des IDS [VEK00] est un exemple de ces systèmes.

Il existe aussi des systèmes hybrides qui utilisent les approches comportementales et morphoscientes pour améliorer la précision et la performance [TDMD04].

Atouts Quand une sonde morphosciente détecte une attaque, elle peut l'identifier. Ceci n'est pas le cas pour la plupart de sondes comportementales. Les alertes dénommées sont plus utiles pour l'operateur car elles l'aident à comprendre la situation. En plus, il est plus façile d'agir sur les alertes dénommées. Les taux de faux positifs sont rapportés inférieurs à ceux de sondes comportementales. Les approches morphoscientes et spécialement l'analyse par signatures sont efficaces en termes de calcul.

Faiblesses Seules les attaques qui sont décrites dans la base de connaissances peuvent être détectées. Pour cette raison les sondes morphoscientes ont besoin des mises-à-jour régulières des descriptions d'attaques. Ceci est une tâche non-triviale et laborieuse. Pour la même raison, l'analyse morphosciente ne peut pas détecter des attaques nouvelles. Les descriptions d'attaques sont souvent spécifiques à l'environnement : le système d'exploitation, les applications et leurs versions par exemple. Les descriptions d'attaques abstraites sont plus flexible dans ce sens. En plus, l'abus de privilèges par les utilisateurs légitimes est difficile à détecter car ils n'ont pas nécessairement besoin d'exploiter une vulnérabilité.

1.1.4 Alertes

Les alertes issues de sondes de détection d'intrusions sont différentes en termes de contenu. Les différences sont crées par les méthodes d'analyse (comportementale vs morphosciente) ainsi que par les sources de données (système vs réseau par exemple). Les sondes systèmes peuvent fournir des informations concernant les utilisateurs, les proceccus, et les commandes quant les sondes réseau peuvent fournir les addresses et ports source et destination et le contenu des paquets. En plus les addresses IP source sont faciles à forger *(spoof)* donc elles ne sont pas toujours fiables. Les sondes morphoscientes peuvent souvent indiquer les attaques précisement et les nommer. Les sondes comportementales fournissent des alertes plus ambiguës selon les modèles de comportement normale utilisés. De plus, le profil d'un réseau produit souvent des alertes moins spécifiques que le profil d'une machine individuelle.

En plus de détails variés, les formats des alertes peuvent être hétérogènes. Ceci rend la traitement d'alertes plus difficile, particulièrement dans les installations à grande échelle avec les sondes hétérogènes. Comme mentionné auparavant, le format IDMEF pour les alertes vise à résoudre ce problème.

Les alertes générées par un IDS sont traditionellement divisées entre les *vrais positifs* et les *faux positifs*. Un vrai positif est une alerte émise correctement, en presence d'une attaque. Un faux positif est une alerte émise quand il n'y avait pas d'attaque. Le *vrai négatif* est une décision de la part d'un IDS de ne pas alerter quand en absence d'une attaque, et le *faux négatif* est une attaque non-détecté.

En pratique, pourtant, cette division n'est pas toujours si claire. Par exemple, Snort, une sonde utilisant l'analyse par signatures d'audit réseau, dispose de signatures pour détecter l'utilisation normale du réseau. Les alertes peuvent être provoqués par les messages SNMP *(Simple Network Management Protocol)*. Dans ce cas, quand Snort émet une alerte sur un tel événement, qui n'est pas une attaque, est-ce que l'alerte est un faux positif ? Nous développerons cette question dans Sect. 2.3.

1.2 La correlation et gestion d'alertes

Les systèmes de détection d'intrusions génèrent un grand nombre d'alertes et l'augmentation du volume de trafic réseau empire la situation. Le problème n'est pas récent : Manganaris et al. ont rapporté en 1999 la génération des milliers d'alertes par jour par les sondes réseau morphoscientes [MCZH99]. D'autres chercheurs ont mentionné des observations similaires [Jul01, JD02]. Nous avons vu des chiffres encore plus grands, jusqu'à quelques cents milles alertes journalières pendant des expérimentations recentes. Ces expérimentations sont décrites dans le Chap. 3.

Le nombre des alertes est un problème en soi, mais en plus, une grande partie de ces alertes sont soit fausses ou non-pertinentes. Selon Julisch jusqu'à 99% des alertes peuvent être des faux positifs [Jul01]. Ceci fait que l'analyse et l'utilisation sont plus difficiles.

Morin a trouvé des taux de faux positifs significativement plus petits [Mor03, p.25]. Il explique la différence par le placement des sondes, et il suppose que les sondes étudiées par Julisch surveillaient des réseaux internes. Les sondes au plus près des points d'accès Internet sont plus exposées aux attaques, et pour cette raison pourraient voir un taux d'attaques plus important.

L'inondation par les alertes est à l'origine de beaucoup de recherches sous le terme *corrélation d'alertes (alert correlation)* ou plus tard *gestion d'alertes (alert management)*. La corrélation d'alertes a trois objectifs principaux par rapport à l'information montrée à l'opérateur.

La réduction du volume : Le groupement où la suppression des alertes, selon les propriétés en commun. Par exemple, plusieurs alertes liées à un scan devraient être groupées dans une alerte.

L'amélioration du contenu : Un ajout dans l'information contenu dans une alerte individuelle. Par exemple, l'utilisation de la topologie et l'information sur les vulnérabilités dans le système surveillé pour vérifier ou évaluer la sévérité d'une attaque.

Le suivi d'activité : Le traçage des intrusions à l'origine de plusieurs alertes au cours du temps. Par exemple, si l'attaquant commence par une scan, continue avec une attaque *remote-to-local*, pour finir avec un accès niveau administrateur, les alertes individuelles devraient être groupées dans une alerte.

Le terme *gestion d'information liée à la sécurité (Security Information Management, SIM)* contient tous les aspects de la gestion d'alertes et des autres informations liés à la

sécurité des systèmes d'information. Une plate-forme SIM a quatre fonctions [DV06], l'acquisition des événements, la gestion d'informations contextuelles, la corrélation d'alertes, ainsi que le reporting et l'échange d'information.

L'acquisition des événements concentre les différents logs sur un point central de traitement. Ceci peut contenir des opérations de filtrage

La gestion d'information contextuelle forme le lien entre l'information additionnelle, comme les données d'inventaire et de vulnérabilité aux hôtes et utilisateurs associés aux alertes.

La corrélation d'alertes a les trois objectifs décrits ci-dessus. Ce composant contient l'intelligence de la plate-forme de traitement.

Le reporting et l'exchange d'information concerne la création des liens entre plusieurs consoles et autres composants, qui peuvent être organisés dans une hiérarchie des sondes et managers. Pour les utilisateurs, typiquement trois types d'interfaces sont requis. Une interface en temps réel est utilisée par l'opérateur pour traiter les alertes à l'apparition. Une console d'investigation *(forensics)* est utilisé par les analystes pour examiner les incidents et analyser l'état du système sans des besoins de temps réel. Le troisème est une console de reporting des incidents pour suivre les effets des contre-mesures ou le fonctionnement normal du système. Une autre possibilité est que la plate-forme SIM envoie l'information concernant la sécurité du système vers la console de gestion de système *(system management console)*. Dans [SS05, p.2] une structure similaire semble dans un produit de Sourcefire est mentionné, sans une description explicite de la console.

Dans la littérature, les approches de corrélation sont souvent divisées entre *implicites* et *explicites*.

1.2.1 Corrélation explicite

Les méthodes de corrélation explicites utilisent des *scénarios d'attaque* définis préalablement pour corréler les alertes entre elles. En plus, les alertes peuvent être vérifiés et/ou assignées une priorité en utilisant l'information sur la configuration, sur la topologie du système surveillé et sur les vulnérabilités exploitées par les attaques. Le but de la plupart des méthodes explicites est l'amélioration du contenu et le suivi d'activité.

On peut faire une analogie entre les signatures d'attaque au niveau sonde, et les scénarios d'attaque. Les scénarios encodent la connaissance requise pour trouver les rapports entre les alertes elles mêmes, ainsi que entre les alertes et les informations supplémentaires et les événements plus généraux du système. Les rapports peuvent être causaux, basés sur la connaissance des étapes dont l'attaquant a besoin pour atteindre son but, ou être liés à la topologie du système.

Il y a plusieurs façons de modéliser les rapports de corrélation. Morin et Debar utilisent des formes temporelles appellés *chroniques (chronicle)* pour décrire des scénarios d'attaque et des scénarios des faux positifs [MD03]. Debar et Wespi ont utilisés des techniques de *backward reasoning* et *forward reasoning* pour trouver, respectivement, des *doublons (duplicate)* et des *conséquences (consequence)*. Cette approche est utilisé dans IBM Tivoli Risk Manager [DW01].

JIGSAW [TK00] utilise un modèle *nécessite/fournit (requires/provides)* pour les scénarios d'attaque. L'assomption est que les attaques sophistiquées ont besoin de plusieurs étapes pour que l'attaquant arrive à son but. Pour réussir, une étape a certaines prérequis et

une fois réussie, l'étape fournit certaines capabilités. Ces capabilités peuvent remplir les prérequis d'une des étapes suivantes. L'idée est de modéliser ce que chaque étape nécessite et fournit. Après les variations avec le même resultat peuvent être détectés sans explicitement encoder tous les variations possibles. Cette approche n'est pas capable d'utiliser des étapes non-modélisées dans le processus de corrélation : il s'agit d'une approche explicite.

Un concept similaire est utilisé par autres chercheurs. Dans le module de corrélation CRIM, Cuppens et Miège utilisent des *pre-conditions* and *post-conditions* [CM02] spécifiées en langage LAMBDA [CO00]. Cette langage consiste en des prédicats logiques. Ning et al. [NCR02a, NCR02b] utilisent également des *prérequis (prerequisite)* et *conséquences (consequence)* dans la construction des graphes de corrélation. Les graphes sont une façon visuelle de résumer les attaques et les stratégies d'attaque très volumineuses en termes d'alertes. Cheung et al. [CLF03] modélisent les attaques avec une langage appelé CAML visant à faciliter le développement des modèles d'attaques et des scénarios.

Les méthodes explicites peuvent être divisées entre *explicites* et *semi-explicites (semi-explicit)*. Les premières encodent les scénarios directement, et les dernières encodent les éléments de scénarios avec les règles pour les combiner. Les méthodes semi-explicites peuvent ainsi construire des combinaisons différentes à partir de ces éléments. Avec cette division, les chroniques et le raisonnement forward-backward sont des méthodes explicites, et les différentes variations de pré-condition/post-condition sont des méthodes semi-explicites.

L'information additionnelle, comme la topologie et la configuration du système surveillé, peut être utilisée dans le processus de corrélation. Par exemple, si une vulnérabilité est un prérequis d'une attaque reussie, le moteur de corrélation pourrait et devrait utiliser cette information. Morin et al. [MMDD02] ont proposé le modèle M2D2 pour stocker et utiliser ce type d'information. Cette information peut être acquise, partiellement, de plusieurs façons, comme l'écoute passive [TDM06] ou en inférant des empreintes d'OS à partir de leur comportement DHCP [HWI05]. Le M-Correlator de SRI International développé par Porras et al. [PFV02] utilise les bases de connaissance de traitement des incidents pour stocker l'information sur les vulnérabilités et la configuration système requise par les attaques. Celle-ci est combinée avec la topologie interne obtenu par Nmap[8]. Porras et al. évaluent la priorité d'une alerte par rapport à l'impact sur la mission de système en utilisant un réseau Bayésien. Un des atouts de ce dernier est sa tolérance aux valeurs inexistantes.

Dans [KR04] Kruegel et Robinson présentent le résumé des méthodes de vérification d'alertes dont le but est de vérifier la réussite de l'attaque détectée. En plus, ils décrivent une implementation utilisant Snort et Nessus. Dreger et al. utilisent l'information fournie par un agent système pour améliorer la précision d'une sonde réseau [DKPS05].

Atouts Comparée aux alertes simples, la corrélation en général peut donner une vue synthètique sur les alertes. Ceci aide l'utilisateur à trouver les ensembles liés et pertinents d'alertes parmi le bruit de fond. Comparée à la corrélation implicite, les scénarios d'attaque contiennent une raison précise pour le groupement d'alertes. Cela rend les résultats de corrélation plus rapidement utilisables que les résultats de corrélation implicite. Ces derniers ont souvent besoin de l'interprétation.

Faiblesses Ces méthodes cherchent les scénarios connus d'attaque, mais elles ont une capacité très limitée de détection des scénarios d'attaque nouveaux et inconnus. Les méthodes semi-explicites peuvent détecter des variations qui combinent les éléments

[8] `http://www.insecure.org/nmap/`, visité 2006-07-18

d'attaque d'une nouvelle façon. La définition des scénarios d'attaque ou les pré- et post-conditions de toutes les attaques est une tâche non-triviale et laborieuse. De plus, la plupart de ces méthodes ont une capacité limitée ou inexistante de gérer les étape d'attaque manquantes. Les fausses alertes émises par les sondes, ou le grand volume des alertes en général peuvent créer des problèmes pour les méthodes de corrélation explicites.

1.2.2 Corrélation implicite

Les méthodes de corrélation implicite cherchent des rapports entre les alertes, par exemple utilisant des statistiques, la fouille de données ou les techniques de visualization. Les scénarios prédéfinis ne sont pas utilisés, et les méthodes implicites peuvent donc eventuellement trouver des liens nouveaux et inconnus parmi les alertes. Dans ceci, l'hypothèse est que les alertes similaires sont, d'une façon ou l'autre, liées.

Valdes et Skinner utilisent une méthode de corrélation basée sur la similarité probabilistique des attribus des alertes [VS01]. La méthode permet d'agréger les alertes similaires, de trouver les alertes liées à une étape d'attaque, et de trouver les liens entre les étapes d'une attaque.

Le composant de corrélation concu par Debar et Wespi [DW01] groupe les alertes en *situations (situation)*. Les alertes dans une situation partagent une ou plusieurs attributs parmi la source, la destination et la classe d'alerte. En changeant le nombre des *wildcards* acceptés, on arrive à sept types de situation différents. Ces différents types de situation construisent des scénarios différents.

Les techniques de fouille de données ont été largement utilisées dans la corrélation. Manganaris et al. [MCZH99] analysent des rafales d'alertes utilisant l'*analyse des associations (association analysis)*. Ils construisent un modèle de comportement normal d'une sonde en termes d'*ensembles des items fréquents (frequent itemset)* et les *règles d'association (association rule)*.

Dain et Cunningham comparent trois approches de construction de scénarios. La première utilise des régles simples écrits à la main, la deuxième utilise une heuristique avec trois mesures, et la troisième utilise une approche de fouille de données [DC01, DC02]. En fait, ils ont examiné trois différentes techniques de fouilles de données, et trouvé que l'arbre de décision *(decision tree)* avait la meilleure performance avec des données particulières.

Julisch analyse des *causes racines (root cause)* en utilisant des taxinomies dans le clustering des alertes *(alert clustering)* [Jul01]. Une fois les causes racines identifiées, il est possible d'écrire des filtres pour enlever les alertes générées par les causes racines. Julisch et Dacier ont aussi utilisé les *règles d'épisode (episode rule)* dans [JD02].

Qin et Lee utilisent les modèles de séries temporelles et le test de causalité de Granger *(Granger Causality Test, GCT)*. Le test permet de trouver des rapports statistiques, et eventuellement inconnus parmi les alertes.

Les techniques de visualisation sont basées sur la capacité évoluée de reconnaissance de formes chez les humains. Ces techniques ont été utilisé pour la détection des anomalies [TMWZ02a, TMWZ02b, TMW03, TM03, Axe03]. Pour la corrélation d'alertes, Nyarko et al. visualisent les données fournies par des sondes réseau. En plus, ils utilisent une interface exotique appelé une interface *tactile*[9] *(haptic)*, qui permet à l'utilisateur de tou-

[9]La technologie tactile est une technologie qui repose l'interaction avec l'utilisateur avec le toucher. Les premières utilisations étaient dans les contrôles des avions, aujourd'hui les interfaces tactiles sont considérées comme une partie importante de la réalité virtuelle. C.f. par exemple `http://en.wikipedia.`

cher et manipuler les données avec leur mains [NCSLO02]. Erbacher et al. visualisent des logs et des données fournies par la sonde Hummer [FTM+98] deployée sur un hôte. Leur but est de faciliter l'analyse des données d'audit [EWF02, ET03]. Nous considerons l'approche d'Erbacher et al. comme une technique de corrélation car ils combinent plusieurs sources de logs. Selon [FTM+98], la sonde Hummer génére aussi des rapports de mauvaise utilisation.

Atouts Les techniques de corrélation implicites peuvent trouver des rapports nouveaux et inconnus dans les données d'alerte. La partie de modélisation d'attaques et de construction des scénarios de méthodes explicites est éliminée. Cette partie est à la fois difficile, et consommatrice de temps.

Faiblesses Les rapports trouvés par les méthodes implicites ne sont pas expliqués. Par consequence, leur analyse peut demander beaucoup d'effort et de temps. De plus, ces techniques peuvent trouver des rapports statistiques dans les données ayant aucune signification dans le monde réelle. Même si nous pensons que les resultats devraient toujours être analysés par un humain, les techniques de visualisation ne laissent même pas la possibilité de traitement automatique de résultats. C'est-à-dire que l'échelle d'utilisation des méthodes de visualisation est limité.

1.3 Organisation et la cible de la thèse

Dans cette thèse[10], nous nous concentrerons sur la corrélation d'alertes et plus particulièrement sur la réduction du volume des alertes, émises par des sondes réseau morphoscientes.

Notre travail est inspiré par des données d'alerte existantes générées dans un environnement réel. Nous considerons ceci un aspect important de ce travail : nous avons cherché des méthodes qui marchent avec les données, et non les données qui s'appliquent aux méthodes. A ce jour, les données simulées ont des problèmes importants de ressemblance avec les données réelles. Peut-être l'effort le plus significative de simualation est le DARPA Intrusion Detection Evaluation [LFG+00, LHF+00]. Celui-ci a été critiqué par McHugh [McH00], et est considéré inconvenable pour développer et évaluer des IDS par plusieurs personnes de communauté de détection d'intrusions.

Comme nous verrons plus tard, une grande partie des alertes sont issues de signatures qui déclenchent sur l'utilisation normale du système surveillé. La plupart des méthodes de corrélation actuelles ont été concues pour d'autres types d'alertes. Nous examinerons des méthodes de traitement d'alertes complémentaires, dédiées à ce type d'alertes.

Notre but est de filtrer les manifestations de ce comportement normal, et de détecter les anomalies en analysant des flux d'alertes au lieu des alertes individuelles. Ces méthodes de filtrage sont basées sur des techniques d'analyse de séries temporelles. La plus simple utilise des moyennes glissantes pour modéliser le comportement normal d'un flux. La plus sophistiquée utilise des modéles autoregressifs non-stationnaires *(non-stationary autoregressive (AR) models)* et filtrage Bayésien. Les méthodes présentées dans cette thèse peuvent être utilisées pour filtrer l'information remontée à l'interface temps réel de l'opérateur, ainsi tant que l'outil d'analyse dans le console d'investigation. La méthode la plus simple a déjà été implémentée dans la plate-forme SIM de France Telecom.

`org/wiki/Haptic`, visité 2006-08-03.

[10] La version originale de cette thèse est en anglais, et le lecteur est invité à se reporter à la version anglaise

Plus précisement, cette thèse soutient l'argument suivant en quatre parts.

1) Dans les environnements réels, les alertes liées à la *conscience* de l'état du système, peuvent constituer une part significative des alertes issues par les sondes.
2) Ce type d'alertes ne contienne pas d'information, qui permet l'analyse alerte par alerte. En même temps, l'analyse des flux d'alertes rend possible l'extraction d'informations utiles pour l'opérateur.
3) Les flux d'alertes contiennent des régularités importantes, qui sont générées par l'utilisation et le comportement normal du système surveillé.
4) Au lieu d'ignorer les alertes générées par des signatures prolifiques ou des hôtes bavards, il est possible de modéliser le comportement normal. En utilisant ces modéles on peut filtrer les alertes liées au comportement normal, et souligner les évènements intéressants.

La suite de cette thèse est organisée de la façon suivante. Le Chapitre 2 décrit les origines des flux d'alertes volumineux. De plus, l'état de l'art dans la corrélation d'alertes est présenté dans ce chapitre. Dans le Chap. 3 nous analysons des jeux d'alertes pour, 1) justifier le besoin pour ce type d'approche, et 2) caractériser le comportement des flux à filtrer et les anomalies à détecter

Les trois chapitres suivants présenteront trois approches différentes pour le traitement d'alertes, le Chap. 4 utilises des modèles de tendance *(trend modeling)*, le Chap. 5 des modèles stationnaires *(stationary)* de séries temporelles, et le Chap. 6 des modèles non-stationnaires *(non-stationary)* de séries temporelles. Nous discuterons le travail existant du point de vue des méthodes et idées dans chacun de ces chapitres. Le Chap. 8 fera une comparison entre ces trois approches, et finalement, le Chap. 8 conclura la thèse.

Concernant les arguments mentionnés ci-dessus, les Chapitres 2 et 3 portent sur les 1, 2 et 3. Les Chapitres 4, 5, et 6 sont liés à l'argument 4.

Chapitre 2

Abondance d'alertes : Description du problème et l'état de l'art

La grande quantité d'alertes est un problème pour les utilisateurs et les chercheurs en détection d'intrusion depuis longtemps déjà. Les systèmes de détection d'intrusion sont apparus pour automatiser l'analyse des vérifications rétrospectives volumineuses [And80, p. 3 and 28]. Aujourd'hui on peut dire que l'histoire est revenue à son point de départ. Depuis quelques temps déjà, des méthodes de traitement d'alertes sont apparues pour automatiser l'analyse de gros volumes d'alertes. Comme mentionné auparavant, de grandes quantités d'alertes générées par des systèmes de détection d'intrusion sont rapportées être des faux positifs, ce qui rend le problème encore plus difficile.

Nous regarderons les raisons de ces abondances d'alertes dans la section 2.1. Comme nous ne sommes pas tout à fait d'accord avec la définition des faux positifs utilisée dans la littérature, nous souhaitons utiliser une division plus fine de cette définition. Cela sera discuté dans la section 2.3. Nous analyserons avec plus de détail des travaux effectués en corrélation d'alertes dans la section 2.2.

Même si ce chapitre considère des problèmes généraux, le chapitre 3 contient des exemples détaillés ainsi que des analyses de données que nous avons recontrés.

2.1 Raisons des abondances d'alertes

Il y a une multitude de raisons pour lesquelles il y a un grand nombre d'alertes générées par les sondes de détection d'intrusion. Trois causes majeures sont les limites des sondes, l'insuffisance et la maintenance des sondes, et la nouvelle utilisation des sondes. Le grand volume de données analysées peut être vu comme une quatrième cause. Depuis que cela augmente le nombre d'alertes générées par les trois autres causes, on peut le considérer comme un problème orthogonal. De plus, la nature des données analysées, et en particulier les nouvelles utilisations des sondes, affectent le type d'alerte constituant l'abondance d'alertes.

2.1.1 Limites des sondes

En 2002, Debar et Morin ont testé des systèmes de détection d'intrusion commerciaux et ont signalé des déficiences dans la précision, l'exactitude et les capacités de diagnostic [DM02]. Des observations similaires ont été reportées, par exemple par Kruegel et Robertson dans [KR04]. Nous pouvons de plus diviser les limites des sondes en

quatre problèmes distincts : la non connaissance de l'environnement de fonctionnement, les limites de la méthode de détection, les contraintes de performance et l'analyse mono-événementielle.

Ignorance de l'environnement de fonctionnement La plupart des sondes de détection d'intrusion n'ont pas ou peu de connaissances sur le système d'information qu'elles protègent. Considérons une sonde réseau qui n'est pas au courant du type de réseau. Elle voit une attaque passer, la détecte et donne une alerte. Cependant, le champ TTL (Time To Live) du paquet est trop petit pour atteindre sa destination. Même si l'attaque a été correctement identifiée, l'alerte est de plus basse priorité parce que la structure cause l'échec de l'attaque. En fonction de l'intérêt de l'opérateur, l'alerte peut même être considérée comme non pertinente. Un exemple classique est une attaque pour le serveur Web IIS qui est utilisée contre le serveur Web Apache. L'attaque n'a aucune chance de réussir, et de plus, comme elle dépend de la politique de sécurité du site, la priorité de l'alerte peut être ajustée ou ne pas aboutir. A moins que l'IDS soit au courant de la configuration du logiciel sur la cible, cela ne peut pas être pris en compte.

De plus, différentes implementations du Transport Control Protocol (TCP) et Internet Protocol (IP) coexistent dans différents systèmes, ce qui rend difficile de dire comment quelques paquets sont arrivés à destination. Ptacek and Newsham ont étudié les problèmes TCP/IP en profondeur et indiqué [PN98, p.5] *en regardant les paquets en transit, il n'y a pas assez d'information pour savoir correctement ce qui s'est produit dans des protocoles de transactions complexes.* Par exemple, même le fait de connaitre la destination ne garantit pas que nous pouvons prédire si la machine va accepter le paquet ou pas. Un système qui manque de mémoire jettera des paquets qu'il accepterait en temps normal.

Les limites des méthodes de détection Les approches de détection ont leurs limites ce qui peut résulter aussi bien en faux négatifs qu'en faux positifs. Des systèmes par scénarios ne peuvent pas détecter des intrusions pour lesquelles ils n'ont pas de description d'attaques. Si la description de l'attaque est trop spécifique, ils vont manquer les variations de la même attaque. D'autre part, des descriptions d'attaques trop générales augmentent le risque de faux positifs, étant donné que la description atteint aussi des événements rétrospectifs bénins. Ce problème a été mentionné par exemple par McHugh [McH01, Sect. 4.5]. Ces systèmes qui utilisent des descriptions d'attaques résumées sont moins exposées à ce problème spécifique que les sondes utilisant les analyses de signatures. Cependant, la majorité des sondes couramment utilisées se fondent sur des signatures de pattern matching.

Snort est un bon exemple de ce type de problème. Une signature avec Snort ID (SID) 469 détecte `NMAP` scans ping en recherchant des messages echo ICMP de longueur 0. Aucune autre application ne peut utiliser les mêmes messages ICMP en même temps. Enfin, `Kontiki` et l'antivirus `avast !` causent des faux positifs[1].

L'hypothèse sous-jacente sur les approches d'analyse comportementale est qu'une intrusion se manifeste d'une manière assez anormale, de telle sorte qu'elle est détectée. Cependant, selon Anderson [And80, Sect. 2.3.3] et ensuite Axelsson [Axe00b, Sect.3.4], et McHugh [McH01, Sect. 4.4], ce n'est pas nécessairement le cas. Par exemple, des utilisateurs clandestins qui connaissent le système, voire même l'IDS, peuvent passer hors du radar. Un autre exemple est la détection d'anomalie en observant les taux

[1] `http://www.snort.org/pub-bin/sigs.cgi?sid=469`,visited 2006-07-18

de paquets et d'octets à certains points centraux du réseau. Ce type d'approche est typiquement utilisé par les ISPs pour détecter les vers et les attaques DoS [LCD04, Sect. 1]. Borgnat et al., ont cependant reporté de telles métriques sous optimales pour détecter des attaques *de déni de services distribués* (DDoS) [BLAO05]. Ils ont montré que les statistiques de premier et de second ordre n'étaient pas suffisantes pour détecter une attaque DDoS. Cette attaque était réelle mais modeste et créée par les chercheurs eux-mêmes dans un trafic normal. Même si l'attaque était vraiment modeste, cela sert d'exemple de problèmes qui surviennent quand l'hypothèse sous-jacente ne tient pas.

Constraintes de performance La quantité de données rétrospectives dont un IDS a besoin pour faire des analyses est en train d'augmenter étant donné que les performances des systèmes et les vitesses des réseaux augmentent, tout comme l'utilisation des systèmes et des réseaux. Le problème est d'autant plus important pour les systèmes réseaux faisant face à la taille totale des systèmes dans le réseau.

Le temps disponible pour le traitement d'un audit est petit, et de plus, la profondeur des analyses qui peuvent être effectuées est limitée. En utilisant encore Snort comme exemple, l'analyse est faite seulement sur un nombre limité de octets par défaut. De plus, les attaques qui sont loin dans le contenu sont manquées. Même si la sonde peut être configurée pour examiner le paquet entier, ce n'est pas une solution faisable, étant donné que la sonde ne correspond pas avec la vitesse du réseau. [2]. PAYL [WS04] peut utiliser des paquets entiers ou des connexions TCP, mais des auteurs ont montré des économies considérables de calculs de temps et d'espace en utilisant non seulement les premiers N octets mais aussi les dernier N octets du contenu. Une sonde de détection d'anomalie appelée ALAD [MC02] inspecte des sessions de données TCP. Elle utilise uniquement le premier mot du contenu supposant que ce dernier est en format texte et que les espaces délimitent les mots. La sonde inspecte les 1000 premiers octets du contenu.

Des algorithmes de détection d'anomalies peuvent être relativement exigeants en terme de CPU et de mémoire. Par exemple, Lakhina et al. classifient les méthodes d'analyse de signaux comme les analyses d'ondelettes utilisées dans [BKPR02] comme méthodes *batch-mode* contrairement aux méthodes *en ligne* [LCD04].

Dans les deux cas, une bonne conception du système est essentielle, en commençant tout d'abord par le niveau OS. En 2001, McHugh a analysé le nombre de cycles de mémoire disponible pour faire des analyses par paquet, et a reporté que, avec le matériel actuellement disponible, *un traitement réaliste du contenu des paquets n'est vraisemblablement pas possible pour les taux de données en dessous de 50 mégabits/s, étant donné que même une simple reconnaissance de la forme contre une seule forme requiert vraisemblablement plus de cycles de mémoire par byte qu'il y en a de disponible* [McH01, Sect. 4.2].

Pour les sondes réseaux, les captures de paquets peuvent déjà être un défi. Par exemple, une capture standard `libpcap` sous les noyaux de Linux 2.4 a besoin de passer les paquets via le noyau. La carte réseau utilise des interruptions pour avoir l'attention du noyau, et, avec des taux de paquets élevés, le noyau passe la plupart de son temps à traiter des interruptions causant des pertes significatives de paquets. Pour les systèmes d'exploitation non modifiés, Deri a reporté dans [Der03] des pertes

[2] Regarder par exemple la discussion sur l'exploitation WMF `snort-users` mailing list, thread *flow_depth and WMF exploit* commencée par Jason Haar at 2006-01-04.

de paquets respectivement de 99.8% pour le noyau Linux de série 2.4, de 66% pour FreeBSD 4.8, et de 32% pour Windows 2K.

Analyse Mono-événementielle Dans certains cas, la plupart des sondes réseau analysent les paquets un par un après un assemblage de quelques paquets, par exemple pour surveiller une requête HTTP entière. Cela empêche la sonde d'associer les étapes d'une attaque à une autre alerte synthétique. Par exemple, Morin et Debar ont trouvé que sur une infection Nimda , Snort générait 20 alertes et Dragon [3] en générait 30 [MD03]. Un autre exemple est un scan sur lequel toutes les alertes relatives doivent être regroupées en une seule. Ce type de problème peut être vu comme un manque de *contexte de l'alerte* dans la sonde - la sonde ne prend pas en compte d'autres événements dans son analyse.

2.1.2 Configuration Insuffisante de la sonde

Les sondes de détection d'intrusion sont loin du type de solutions fire-and-forget. Elles ont besoin d'être configurées dans un environnement opérationnel. Après le travail initial effectué, l'environnement change, et les sondes ont besoin d'être maintenues et contrôlées en plus des alertes qui ont besoin d'être analysées.

Par exemple, Snort autorise la définition de réseaux interne et externe, et les adresses des serveurs Web, pour détecter les attaques de l'extérieur vers l'intérieur uniquement. Comme la partie du trafic qui est analysé est limitée, cela aide à éviter les faux positifs.

D'un autre côté, le risque de faux négatifs augmente également. Utiliser une liste de serveurs Web implique que n'importe quelle attaque contre les serveurs Web n'y figurant pas restera non détectée. Il peut y avoir plusieurs raisons pour qu'un serveur ad hoc apparaisse. Un serveur peut être déployé par des développeurs faisant des tests sur leur projet etc. Typiquement, dans les organisations de recherche, les utilisateurs ont plus de liberté en respectant cela. Même une liste approximative peut être difficile à tenir.

Plus généralement, quand on regarde seulement les attaques qui proviennent des réseaux externes vers les réseaux internes, on peut manquer 1) l'activité d'un pénétrateur externe, humain ou programme, allant au-delà de la protection et utilisant la première machine compromise comme appui, et 2) les attaques venant directement des réseaux internes. Un ver cherchant les vulnérabilités distantes est un attaquant externe seulement jusqu'à la première infection à l'intérieur du périmètre - la propagation à l'intérieur manquera complètement si on regarde uniquement les attaques venant de l'extérieur. Dans le cas d'un portable infecté, en revenant de vacances ou d'une conférence, même la première étape externe est inexistante - le ver attaque directement de l'intérieur. Les attaques internes intentionnelles peuvent être lancées par des personnels d'organisation, des consultants, des internes, des visiteurs, etc. Un autre exemple d'attaque de ver par l'intérieur est reportée dans [Hal03] où l'infection SQL Slammer prend place via une connexion relais au siège, même si les défenses du périmètre bloquent les attaques externes. Hally a mentionné aussi le problème de l'attaque interne manquée quand on restreint l'analyse à des attaques venant directement du réseau externe au réseau interne.

De plus, la séparation entre les réseaux internes et les réseaux externes n'est pas toujours simple. Plus particulièrement dans les grandes organisations, les structures administratives et les réseaux "compartimentés" peuvent segmenter le périmètre. Considérons une organisation avec plusieurs divisions. Le réseau d'une division peut être considéré confidentielm et dans ce cas, le trafic réseau des autres divisions, comme le SNMP, peut

[3] `https://dragon.enterasys.com`, visited 2006-07-18

être considéré comme un trafic externe. Dans un même temps, l'activité Web des autres divisions peut être considérée comme interne. Nous devons noter cependant que si ces différents périmètres sont clairs pour l'administrateur de sécurité, ils peuvent être pris en compte, par exemple avec Snort. L'infection mentionnée ci-dessus à travers un lien frame relay au siège sert d'exemple ici aussi.

Le résultat est que des problèmes de configuration sont possibles, ce qui n'est pas toujours simple, ou bien même voulu pour renforcer la division entre réseaux internes et externes en terme de sources et destinations des attaques, ou pour énumérer tous les différents types de serveurs.

2.1.3 Nouvelles utilisations : Utilisation et politique du système de surveillance

Aujourd'hui, les sondes de détections d'intrusions, plus particulièrement les sondes réseau basées sur les signatures, sont aussi utilisées pour d'autres buts qu'une simple détection d'attaque. La supervision de la conformité des politiques d'utilisation et de sécurité en est un exemple, et cela est une des utilisations prévues de Bro[4]. Une autre nouvelle utilisation est de se servir des sondes de détection d'intrusion pour surveiller le fonctionnement et l'utilisation normale du système en terme de différents contrôles et de gestion du trafic.

Une organisation peut avoir des politiques restreignant ou interdisant les utilisations de messagerie instantanée (IM) et d'application peer-to-peer (P2P). Par conséquent, le besoin de surveiller l'utilisation d'un tel logiciel peut exister. David Goodrum de NFR[5] a mentionné dans sa mailing list `focus-ids` que des gens utilisent les IDS pour surveiller les versions de logiciels de navigation et l'utilisation de IPv6[6]. Scott Hazel a également écrit sur la politique de conformité et la surveillance sur cette même liste[7].

Le trafic de gestion et contrôle de réseaux peut être surveillé, par exemple pour les raisons suivantes : 1) il fournit un vecteur pour rassembler les informations et influencer le fonctionnement du réseau, et 2) il est utile pour résoudre tout type de problèmes. Par exemple, Morin a reporté en 2002 que seulement 33% des signatures Snort ont référencé une vulnérabilité dans la base de données [Mor03, p.19] Common Vulnerabilities and Exposures [8] (CVE). Abi Haidar a présenté une classification détaillée des signatures Snort selon les références CVE, Bugtraq, arachnids, Nessus, et McAfee. En 2005, elle a analysé 2505 signatures Snort, parmi lesquelles 749 ou 30% n'avaient pas de référence, et 1008 ou 40% avaient une référence de vulnérabilité CVE.

Dans notre expérience [VD04, VDMS06], ces types d'alertes sont responsables d'une grande partie d'abondance d'alertes. Par exemple, dans une base de donnée d'alertes générées par les sondes Snort, seulement cinq signatures, toutes venant de cette catégorie, étaient responsables de 68% des alertes.

---

[4] `http://bro-ids.org/Overview.html`, visited 2006-07-18

[5] `http://www.nfr.com`, visited 2006-07-31

[6] David Goodrum, *On the definition of false positive - was : Re : location of an IPS* on the `focus-ids` mailing list, 2005-10-28

[7] Scott Hazel, *Re : Tuning false positives*, on the `focus-ids` mailing list, 2005-12-28

[8] `http://cve.mitre.org`, visited 2006-07-18

2.1.4 Volumes analysés et nature des données

Comme mentionné ci-dessus dans les problèmes de limites et de performances des sondes, les volumes d'audit de données sont en train d'augmenter. Les trois causes majeures de l'abondance d'alertes mentionnées auparavant créent des alertes à un certain volume. Quand la quantité des données inspectées augmente, le nombre absolu d'alertes générées par chacune des trois causes a tendance à grimper. De plus, ce quatrième problème peut être vu comme orthogonal aux trois autres.

Le taux de base (*base rate*), la proportion des manifestations d'intrusions et les données normales dans les sources d'audit sont des problèmes liés. Axelsson parle des erreurs de taux de base *(base-rate fallacy)* dans [Axe99]. Cela veut dire que, souvent, les gens ne prennent pas en compte le taux de base d'incidence quand ils considèrent intuitivement des problèmes probabilistes. Prenons I, respectivement ¬I, les comportement intrusifs, respectivement les comportements bénins. De même, prenons A, respectivement ¬A pour les sondes alertées et les sondes non alertées. Le taux des vrais positifs est exprimé par $\mathrm{P}(A|I)$, probabilité d'avoir une alerte en présence de comportements intrusifs. De la même manière, le taux des faux positifs est exprimé par $\mathrm{P}(A|\neg I)$, probabilité d'avoir une alerte dans les comportements bénins. Le taux des faux négatifs est $\mathrm{P}(\neg A|I)$, et le taux des vrais négatifs est $\mathrm{P}(\neg A|\neg I)$. Maintenant, considérons le taux de détection bayésien $\mathrm{P}(I|A)$, probabilité qu'une alerte indique correctement une intrusion. Dans [Axe99, Sect. 4.2] on a alors

$$P(I|A) = \frac{P(I) \cdot P(A|I)}{P(I) \cdot P(A|I) + P(\neg I) \cdot P(A|\neg I)} \quad . \tag{2.1}$$

De la même manière, le taux négatif de détection bayésienne $\mathrm{P}(\neg I|\neg A)$, probabilité qu'en cas d'absence d'alerte il n'y a pas d'intrusion. On a alors

$$P(\neg I|\neg A) = \frac{P(\neg I) \cdot P(\neg A|\neg I)}{P(\neg I) \cdot P(\neg A|\neg I) + P(I) \cdot P(\neg A|I)} \quad . \tag{2.2}$$

Certains voudront avoir les deux taux de détections aussi élevés que possible. Le taux de détection bayésien est important du point de vue de 1) l'opérateur tenant les alertes, et 2) méthodes d'alertes post-traitement. Dans le premier cas, un taux de détection bayésien bas est problématique, étant donné que l'opérateur deviendra surchargé de faux positifs et manquera les vrais positifs occasionnels. Dans certains cas, la corrélation et les processus de réponse automatiques auront besoin de prendre en compte l'incertitude. Un taux de détection bayésien négatif bas est moins évident pour l'opérateur étant donné que cela ne rempli pas la console d'alertes. Dans un même temps, le système ne détecte pas les intrusions comme c'était attendu. Cela cause des problèmes aussi pour les méthodes post-traitement automatisées qui auraient besoin d'être capable de faire face aux étapes d'intrusions manquantes.

Souvent, le taux des vrais positifs $\mathrm{P}(A|I)$ et celui des faux positifs $\mathrm{P}(A|\neg I)$ sont approximativement au moins du même ordre de grandeur. Maintenant, si la quantité de comportements intrusifs est significativement plus petite que la quantité des comportements bénins, $\mathrm{P}(I) << \mathrm{P}(\neg I)$, le taux de détection bayésien et le nombre de faux positifs sont contrôlés par le taux des faux positifs de la sonde. Cela peut être vu dans (2.1), étant donné que dans le dénominateur, le terme $\mathrm{P}(\neg I) \cdot \mathrm{P}(A|\neg I)$ est dominant par rapport à $\mathrm{P}(I) \cdot \mathrm{P}(A|I)$.

Axelsson a obtenu des taux de détection bayésiens bas et alarmants avec ses calculs, en utilisant un taux de base de $2 \cdot 10^{-5}$. Il a estimé le taux de base avec un ensemble fictif

de petits réseaux, générant des événements d'audit sécurité C2 et avec deux tentatives d'intrusion par jour. Même avec un taux de détection irréaliste de 1.0, on aura besoin d'un taux bas de faux positifs, de l'ordre de $1 \cdot 10^{-5}$, pour les deux tiers des alertes pour indiquer les activités vraiment intrusives.

Comme note latérale, nous aimerions faire quelques remarques en respectant ces nombres. Le taux de base pourrait être plus haut, étant donné que de nos jours, Internet n'est plus un endroit sûr. En dehors du firewall c'est une zone neutre pleine de vers, ou encore de logiciels malveillants et des gens organisés malintentionnés.

En 2004, Pang et al. [PYB+04] ont signalé que la quantité de trafics malveillants sur Internet est en progression. Selon eux, des études récentes sur les trafics Internet de 1992 à 1997 ne mentionnent pas d'attaques de trafic en cours, mais de nos jours, les opérateurs réseaux sont familiers avec *la présence incessante de trafic, ce qui ne prédit rien de bon.*

En juillet 2005, Sophos a dit dans leur communiqué de presse que les machines Windows non protégées seraient infectées avec 50% de probabilité en 12 minutes [9]. La presse en question ne spécifie pas la configuration du système avec plus de détail. Plus récemment, Symantec a obtenu un temps plus long à compromettre, qui est d'une heure et 12 secondes pour un Windows XP Professionnel qui ne serait pas à jour [Sym06, p.10].

Depuis que des mesures de détection existent pour détecter les failles de sécurité, un IDS devrait être placé derrière le firewall[10]. Même dans les réseaux internes, différents logiciels malveillants peuvent faire incliner la balance vers un taux de base plus élevé. Un exemple de comment peuvent être répartis les logiciels malveillants est une entrée de blog de l'équipe d'ingénieurs contre les logiciels malveillants de Microsoft, le 03 avril 2006[11]. L'équipe a trouvé qu'un outil automatique a enlevé 250 000 infections d'un ver appelé Alcan.B[12] à l'origine découvert en juin 2005. Auparavant, ils avaient enlevé 40 000 infections de Nyxem aka Win32/MyWife.E[13]. Et c'est pour cela que l'équipe contre les logiciels malveillants a considéré le ver avec plus *d'exagération que la réalité*[14].

Alcan.B a écarté des fichiers partagés à travers des réseaux P2P, et a caché une nouvelle copie des siens dans un programme craqué. Cela contacte une multitude de sites Web pour trouver un ver. Nyxem se propage via des emails et des réseaux partagés. Même si ce type de ver génère des comportements moins intrusifs que les vers à balayage par exemple, s'ajoutent aux taux de base. Par exemple, Snort a une signature au moins pour Nyxem (Community rules, sid 100000226).

En 2004, America Online et la National Cyber Security Alliance ont investigué la situation des spyware dans 329 foyers. 57% étaient au courant d'un spyware dans leur ordinateur même si 80% étaient vraiment infectés. En moyenne, 93 pièces de spyware ont été trouvés par machine infectée [OA04]. En mai 2005, Moshchuk et al. ont reporté avoir examiné 18 millions d'URL avec une 'chenille'. Ils ont trouvé par hasard 21 200 exécutables parmi lesquels 13% étaient infectés par un spyware. La proportion après un nouveau scan en octobre 2005 était cependant significativement plus petite, 5.5% [MBGL06]. Même si on s'attend à ce que les foyers soient moins attentifs, moins protégés, et moins renseignés que les utilisateurs professionnels, les gens sont tentés d'ouvrir les emails suspects, tous les

[9] `http://www.sophos.com/pressoffice/news/articles/2005/07/pr_uk_midyearroundup2005.html`, visited 2006-07-18

[10] Seulement un est intéressé à observer le comportement malicieux sur Internet. Mais il y a des façons adaptées pour apprendre à connaître son ennemi, voir e.g. [PDP05].

[11] `http://blogs.technet.com/antimalware/archive/2006/04/03/424113.aspx`, visited 2006-07-18

[12] `http://www3.ca.com/securityadvisor/virusinfo/virus.aspx?id=43230`, visited 2006-07-18

[13] `http://www.microsoft.com/technet/security/advisory/904420.mspx`, visited 2006-07-18

[14] `http://blogs.technet.com/antimalware/archive/2006/02/06/418876.aspx`, visited 2006-07-18

types de pièces joints et cliquent sur les liens. Il y a aussi des indications selon lesquelles des réseaux professionnels seraient infectés d'un spyware[15].

Dans un article, Websense a examiné des fichiers exécutables indexés par Google, et a trouvé des milliers de pièces de logiciels malveillants, dont le nom est inofensif[16]. H.D. Moore a relevé un outil similaire[17] pour les utilisations publiques. Moore a signalé à IDG qu'il avait examiné 2 400 ensembles d'exécutables et avait trouvé 125 logiciels malveillants. Plus de 90 étaient des parties d'emails malveillants dans les archives des listes d'emails en ligne[18]. Le nombre d'exécutables malveillants est petit. Selon Moore, Google indexe de petits nombres d'exécutables et, en général, ces types de recherches sont utiles pour trouver les sites qui distribuent des logiciels malveillants, mais pas pour étudier les logiciels malveillants en tant que tel.

Même si la propagation rapide des scans génère une grande quantité de trafics intrusifs, la propagation active est souvent détectée et des mesures sont prises en compte contre elles. Les spyware sont répandus par les activités légitimes comme des emails et des recherches sur le web, et peuvent rester non détectées et garder le trafic réseau contaminé pendant une longue période.

Vu que nous n'avons pas de vrais nombres pour supporter la spéculation à propos du taux de base, tout cela n'est qu'une hypothèse appuyée sur quelques exemples. Nous voulons principalement montrer que la situation n'est pas sans espoir comme elle l'est décrite par Axelsson. Dans tous les cas, la nature bénigne de la plupart des données surveillées augmente le nombre de faux positifs et d'alertes non pertinentes et l'opérateur ne devrait pas recevoir des alertes d'intrusion qui ne sont pas vraiment des indications d'intrusions. Si des positifs non pertinents sont générés, ils devraient être masqués, ou du moins, clairement marqués.

2.2 Etat de l'art en corrélation d'alertes

Dans la première partie de ce chapitre, nous avons vu une analyse sur les raisons de ces abondances d'alertes. Nous avons identifié quatre raisons principales : 1) limites des sondes 2) configuration insuffisante des sondes 3) nouvelles utilisations des sondes, et 4) gros volume de données surveillées. Comment pouvons nous traiter ces problèmes ? Le *volume global* est un fait auquel nous devons faire face. Des solutions possibles seraient d'améliorer la performance des sondes ou de choisir des données sources plus denses comme les données sources des applications de base. La performance des sondes peut être améliorée par exemple en filtrant la donnée qui doit être analysée. Dans [TDMD04], les auteurs emploient un composant d'anomalie simple et rapide pour se dégager des événements considérés comme sécurisés. Après le reste est donné au composant misuse qui est plus demandeur de ressources. Une idée similaire est utilisée dans [GYB06], dans lequel les auteurs proposent d'utiliser une simple sonde de machine de base pour fournir les informations autant directement de l'Apache qu'indirectement du log Apache. Cette approche pourrait autoriser à décharger les décodeurs HTTP par exemple, et à réutiliser le travail effectué par Apache dans tous les cas. Des exemples de données sources plus

[15] The Register reports a likely-biased survey by Trend Micro with 47% of corporate users have noticed spyware at work `http://www.theregister.co.uk/2005/10/11/spyware_survey/print.html`, visited 2006-07-18

[16] `http://www.websense.com/securitylabs/alerts/alert.php?AlertID=547`, visited 2006-08-10

[17] `http://metasploit.com/research/misc/mwsearch/mwsearch.html`, visited 2006-08-10

[18] `http://www.cio.com/blog_view.html?CID=23075`, visited 2006-08-10

denses sont par exemple les logs deeserveurs Web utilisés par les WIDS dans [TDMD04] et les sondes présentées dans [ADD00]. Les applications des sondes dédiées pourraient également être une solution. Par exemple, Sengar et al. ont proposé une sonde pour les moniteurs VoIP [SWWJ06]. En résumé, la solution est pratiquement la même au niveau de la sonde.

Les *problèmes de configuration* que nous avons considérés ici sont de définir les réseaux internes et externes pour les sondes. Par exemple, si quelqu'un désire détecter des anomalies, des attaques ou des problèmes venant de l'intérieur du système, il est difficile d'utiliser la division entre les réseaux internes et externes. Par conséquent, quelques signatures se déclenchent à cause du fonctionnement normal du système, générant probablement une grand nombre d'alertes. Les *nouvelles utilisations des sondes* ont tendance à causer un effet similaire. Des méthodes de corrélation dans le but de réduire le volume peuvent être utilisées pour filtrer ces types d'alertes. Dans cette partie, nous passerons en revue l'état de l'art qui peut être utilisé pour répondre à ce problème. Nous verrons qu'ils sont principalement conçus pour filtrer les alertes basées sur les attributs d'une alerte individuelle. Comme nous le verrons plus tard dans le chapitre 3, plus particulièrement dans le cas des nouvelles utilisations des sondes, la filtrage a besoin d'utiliser plus d'informations pour prendre une décision sur la signification des alertes.

Pour répondre aux *limites des sondes*, une solution naturelle serait d'améliorer les sondes. Cependant, cette approche a ses limites. Comme l'ont mentionné Ptacek et Newsham [PN98], quelques fois, il n'y a pas assez d'information sur le fil pour prendre les bonnes décisions. Une autre solution au problème est proposée, c'est la corrélation d'alerte. Les limites des sondes causent un éventail de problèmes, mais comme l'a remarqué Julisch [Jul03a], la plupart des méthodes de corrélation sont centrées sur les attaques, leur détection, leur vérification et les scénarios construits.

Par exemple, une partie des problèmes est liée à l'ignorance des sondes de son environnement opérationnel. Une partie d'entre eux peut être résolue en utilisant les modèles de données adéquats comme celui présenté dans [MMDD02] ou par de techniques de vérifications d'alertes comme décrites dans [KR04]. Dreger et al. ont également proposé d'obtenir des connaissances additionnelles et redondantes directement à partir de l'Apache ou alternativement à partir de logs Apache pour améliorer l'exactitude de Bro [DKPS05]. Les configurations des hôtes peuvent être, ou du moins partiellement, découvertes par des mesures passives [TDM06] ou actives comme Nmap ou des produits commerciaux de scan (e.g. Qualys[19] and Foundstone products from McAfee[20]).

Un autre exemple est le manque d'analyses multi-événementielles. En réponse, un grand corps de recherche de corrélation est centré sur les détections de scénarios d'attaques, incluant [NCR02b, NCR02a, DC01, DC02, CO00, CM02, MD03]. Aucun d'entre eux n'est le but de la thèse, d'ailleurs nous ne les détaillerons pas. Les lecteurs intéressés peuvent lire [Mor03] pour plus de détails.

Notre but principal repose sur la réduction du volume d'alertes. Nous verrons ici la plupart des approches de corrélation implicites dressant le même problème. Nous considèrerons cinq approches : agrégation de situation, corrélation probabiliste, analyses de causes statistiques, et deux approches de data mining. Les trois premières sont uniquement utilisées pour des reconstitutions de scénarios, mais étant donné qu'il y a des méthodes implicites qui fournissent des moyens d'agréger des alertes, elles seront présentées. Les deux approches de data mining sont centrées sur les faux positifs au lieu des attaques. Elles ont

[19] `http://www.qualys.com/`, visited 2006-07-18
[20] `http://www.mcafee.com`, visited 2006-07-18

pour objectif la réduction du volume par filtrage des alertes à partir des connaissances découvertes par les alertes data mining.

2.2.1 Situations et axes de projection

Debar et Wespi emploient des composants implicites d'agrégation dans [DW01]. Les alertes sont projetées sur trois axes : l'adresse de la source, l'adresse cible et la classe d'alerte. Ensuite, les alertes qui sont égales dans cette projection sont agrégées ensembles étant donné les situations.

Pour contrôler les types d'agrégations qui sont faits, les *wildcards* sont autorisés en assortissant le long des trois axes. Pas plus de deux *wildcards* sont autorisés à la fois. Il y a donc sept types de situations pour lesquelles les différentes projections correspondent. Nous notons les types de situation comme (`src`, `dst`, `class`), et pour indiquer sur quels axes les valeurs sont égales nous utilisons `*` pour un *wildcard* . Les différents types de situation sont :

Type 1 (`src`, `dst`, `class`) capture les alertes pour lesquelles les trois attributs sont égaux. Ce type de situation capture l'activité récurrente d'un attaquant seul.

Type 2-1 (`src`, `dst`, `*`) capture les alertes pour lesquelles la source et les adresses de destination sont égales. Ce type capture les différents types d'activité faites par un attaquant seul contre une machine seule. Par exemple, les alertes causées par un attaquant qui attaque plusieurs services d'un serveur à partir d'une machine sera capturé dans ce type de situation.

Type 2-2 (`*`, `dst`, `class`) capture les alertes qui ont la même classe d'attaque et la même destination. Ce type de situation agrège par exemple les alertes à partir d'une attaque distribuée.

Type 2-3 (`src`, `*`, `class`) capture les alertes avec la même source et la même classe d'attaque. Le type de situation agrège par exemple des scans horizontaux, une attaque contre plusieurs machines, ou plus d'attaques ciblées contre les serveurs Web en utilisant la même exploitation.

Type 3-1 (`src`, `*`, `*`) capture les alertes avec la même adresse source. Par exemple, un attaquant qui utilise une machine pour reconnaissance et ensuite les attaques actuelles dépendant des cibles OS et des services, est capturé dans ce type de situation.

Type 3-2 (`*`, `dst`, `*`) capture les alertes avec la même adresse de destination. Un exemple typique est les attaques distribuées.

Type 3-3 (`*`, `*`, `class`) capture les alertes partageant la même classe d'attaques. Cela regroupe par exemple les alertes causées par propagation de vers.

Les types de situation se chevauchent et dépendent des volumes relatifs aux situations. Une ou deux situations peuvent être déclenchées. Par exemple, si des alertes sont générées pour les paquets d'alertes DDoS, elles correspondront aux types 2-2 et 3-2, (`*`, `dst`, `class`), (`*`, `dst`, `*`). Si le type de situation 2-2 contribue à plus de 50% au type de situation 3-2, seul le type de situation 2-2 sera déclenché. Si le type de situation 2-2 correspond à moins de 10% à la situation 3-2, alors seul le type de situation 3-2 sera déclenché. Si la proportion est entre 10% et 50%, les deux situations sont relevées.

Les situations agrègent les alertes, et de plus aident à faire face au volume, mais ne fournissent pas un moyen de filtrer les alertes. L'agrégation des situations peut être vue comme une partie de reconnaissance de scénario. Le composant de corrélation basé sur

les reproductions/conséquences ajoute des raisonnements explicites sur des liens logiques entre les alertes.

2.2.2 Approche probabiliste et similitudes attendues

Valdes and Skinner ont introduit dans [VS01] une approche probabiliste à la corrélation d'alerte. Ils ont définit pour chaque attribut d'alerte comparable (source IP, destination IP, numéros de port, classe d'attaque, sonde, temps, etc.) une valeur similaire entre 0 et 1, 1 signifie une correspondance parfaite. La similitude d'adresse est basée sur la probabilité que les deux adresses soient du même sous-réseau. La similitude de classe d'attaque est une valeur définie comme experte codant comment la connaissance de certains types d'étapes d'attaque peuvent en précéder d'autres, et ainsi de suite. Ils nomment les alertes groupées des méta-alertes, et les considérations similaires sont données typiquement entre les attributs des méta-alertes et une nouvelle alerte. L'article définit comment les valeurs attribuées sont fusionnées quand les nouvelles alertes sont ajoutées aux méta-alertes.

Valdes and Skinner utilisent une *similitude minimum* par attribut défini comme seuil qui doit être rencontré pour tous les attributs, pour que deux alertes soient fusionnées. La similitude attendue (*Expected similarity*) exprime l'attente que deux attributs soient similaires si les alertes sont liées.

Ils ont défini une métrique similaire entre les alertes de cette façon :

$$sim\,(x, y) = \frac{\sum_j E_j sim(x_j, y_j)}{\sum_j E_j} \quad , \tag{2.3}$$

où x est la méta-alerte candidate pour faire correspondre la nouvelle alerte y, j est un index au-dessus de toutes les alertes attribuées, E_j est la similitude attendue pour l'attribut j et x_j, y_j sont les valeurs pour l'attribut j dans les alertes x et y. Différentes sondes peuvent fournir des attributs différents, e.g. considérons une hôte contre des sondes réseaux, et seulement les attributs chevauchants sont utilisés pour les calculs.

La similitude peut être vue comme une somme pondérée de similitudes de caractéristiques, les pondérations sont fournies par les espérances de similitude. Elles sont utilisées pour guider le type d'agrégation ou de corrélation exécutés par la méthode. Il est possible de créer *des fils d'alertes*, *des rapports d'incident*, et *des rapports d'attaques corrélées*, ce qui correspond respectivement à l'aggrégation du niveau de l'alerte, l'agrégation du niveau de l'attaque, et à la détection de scénario.

Comme pour les situations, l'approche fournit une façon d'agréger les alertes, et de plus, apporte une vue plus synthétique. Ici, les connexions logiques entre les étapes d'attaques peuvent être données avec la même méthode en ajustant les similitudes espérées. Cette approche fournit les significations actuelles pour filtrer les faux positifs par exemple, même s'ils peuvent être regroupés comme méta-alertes.

2.2.3 Analyse statistique de causalité

Qin et Lee [QL03, QL04] utilisent des méthodes d'analyse de séries temporelles pour trouver les relations implicites dans les données d'alertes. Ils regroupent les alertes partageant tous les attributs ensemble autorisant un petit laps de temps de l'ordre de quelques secondes. Cela regroupe les alertes issues de la même attaque. Dans l'étape suivante, ils regroupent les alertes ayant les mêmes valeurs d'attributs indépendamment de la sonde. Cette étape agrège ensemble les alertes reliées par la même attaque issue de sondes

hétérogènes. Là encore, une petite différence dans les heures de génération des alertes est autorisée. Les alertes agrégées résultantes sont appelées méta-alertes. Une série de temps $\{y_t\}, t = 0, \ldots, N-1$ est formée comme le nombre de méta-alertes dans une unité de temps. Qin et Lee regardent pour des relations statistiques entre deux séries de temps avec le Granger Causality Test (GCT) [Gra69]. De plus, l'approche utilise une méthode d'alertes prioritaires à partir de [PFV02].

le test GCT indique si une série temporelle $\{x_t\}$ est la cause d'autres séries temporelles $\{y_t\}$. Cela fait appel à un test statistique pour décider si la série retardée $\{x_t\}$ fournit des informations statistiquement significatives sur les modèles de série

$$y_t = \sum_{k=1}^{p} a_k y_{t-k} + e_t \ ,$$

et

$$y_t = \sum_{k=1}^{p} b_k y_{t-k} + \sum_{k=1}^{p} c_k x_{t-k} + e'_t \ ,$$

où p est la longueur de retard, et les paramètres modèles $a_k, b_k,$ et $c_k, k = 1, \ldots, p$ sont estimés avec la méthode des moindres carrés. Les sommes des carrées des résidus des modèles AR et ARMA R_0 et R_1 (resp.) sont $R_0 = \sum_{t=1}^{T} e_t^2$ et $R_1 = \sum_{t=1}^{T} e'^2_t$, où $T = N - p$. Le modèle AR utilise la somme pondérée des p valeurs anciennes de y pour prédire la valeur actuelle de y. Le modèle ARMA utilise en plus une somme pondérée de p valeurs anciennes de x pour prédire la valeur de y. Si on avait $R_1 < R_0$, cela indiquerait que la valeur des séries $\{y_t\}$ est aussi bien exprimée en utilisant les valeurs des séries $\{x_t\}$ qu'avec les valeurs de $\{y_t\}$ seules i.e. les deux séries sont corrélées.

Un Index de Causalité de Granger (GCI) g est définit par

$$g = \frac{(R_0 - R_1)/p}{R_1/(T - 2p - 1)} \sim F(p, T - 3p - 1)$$

c'est-à-dire que g est conforme à la distribution$F(p, T - 3p - 1)$[21]. L'hypothèse nulle est que les deux séries sont non corrélées i.e. $c_k = 0$, $k = 1, \ldots, p$. Si la valeur de g est plus grande que le seuil acceptable dans le F-test, l'hypothèse nulle est rejetée, et on dit que la série $\{x_t\}$ *Granger-causes* la serie $\{y_t\}$.

Chaque hyper-alerte est utilisée à son tour comme une alerte cible, et toutes les autres hyper-alertes sont testées deux-à-deux contre cette cible pour trouver une corrélation statistique, et celles qui passent le F-test sont enregistrées. Finalement, m hyper-alertes avec les plus grandes valeurs GCI sont considérées comme corrélées statistiquement à l'hyper-alerte cible, et passent à l'opérateur pour validation.

Cette approche de corrélation existe pour les scenarios de reconnaissance implicites et peuvent trouver une relation inconnue entre les alertes. Cependant, cela demande qu'il y ait une relation statistique entre les séries d'alertes. Le test a tendance à signaler des faux positifs à cause des alertes qui sont présentes pratiquement tout le temps, mais selon les auteurs, elles sont facilement identifiables via un manuel d'inspection. Le test était donné en utilisant la version DARPA 3.1 du Grand Challenge Problem [22]. De plus, avec les données réelles, le problème peut être plus important. L'approche est faite pour un traitement hors-ligne, même si [QL04, Sect. 3.2] mentionne aussi un mode opératoire en ligne.

[21] In [QL03] la distribution est $F(p, T - 2p - 1)$, mais nous assumerons que la référence [QL04] est précise

[22] `http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html`, visited 2006-07-18

2.2.4 Fouille de règles d'association

Une des approches récentes appliquant le data mining pour la corrélation d'alerte est le travail effectué par Manganaris et al. [MCZH99]. Ils extraient des règles d'association et *itemsets fréquents* en utilisant des données réelles des Services de Réponses d'Urgence d'IBM. Ils ont reporté des centaines de milliers d'alertes innocentes par jour et ont sondé dans un environnement réel. Le but est de modéliser les comportements innocents et de filtrer ces alertes en prenant en compte le contexte d'alerte en terme d'autres alertes.

Les règles d'association sont définies comme dans [Man96,Man97]. Soit $R = \{A_1, A_2, \ldots, A_p\}$ un ensemble d'attributs binaires. Une relation $r = \{t_1, \ldots, t_n\}$ sur le schéma R est une $p \times n$ matrice avec des colonnes correspondant à R. Maintenant, la règle d'association sur la relation r est exprimée par $X \Rightarrow B$, où $X \subseteq R$ et $B \in R \setminus X$. L'interprétation est que si une rangée dans r a 1 dans les colonnes correspondant à X, alors c'est comme si la rangée avait 1 aussi dans les colonnes correspondant à B.

Pour $X \subseteq R$ le *support* (alternativement *fréquence*) de X dans r, $s(X, r)$ est une fraction de lignes dans r ayant des 1 dans les colonnes correspondant à X. La *confiance* de la règle $X \Rightarrow B$ dans r est définit par $s(X \cup \{B\}, r) / s(X, r)$. X est appelé un itemset fréquent dans r si $s(X, r) \geq \sigma$, où σ est appelé seuil support minimum.

Manganaris et al. partitionnent le flux d'alertes en rafales. Ces rafales sont limitées par des alertes de temps entre arrivées plus grandes que la moyenne. Chaque rafale est considérée comme une *transaction*. Les itemsets fréquents sont des combinaisons d'alertes qui se produisent souvent en rafales, et les règles d'association relient les occurrences d'un itemset fréquent à un autre. L'ordre des alertes n'est pas pris en compte et il n'y a pas de notion de temps quand on utilise les règles d'association, on analyse juste les transactions. Un comportement normal d'une sonde est modelé comme une collection d'itemsets fréquents et de règles d'association de haute confiance.

Les déviations à partir d'un comportement normal dans une rafale sont détectées par 1) les alertes n'appartenant à aucun itemset fréquent, ou 2) les alertes manquées qui cassent les règles d'association de haute confiance. D'une autre façon, une rafale associée aux itemsets fréquents et ne cassant aucune règles d'association correspondant aux itemsets associés est considérée comme une partie d'un comportement normal d'une sonde, i.e. des fausses alertes.

Cette approche est centrée sur la réduction des faux positifs au lieu des attaques. Elle propose également des significations pour modeler et filtrer des comportements normaux de sondes, en prenant en compte le contexte d'alerte. Cependant, les auteurs considèrent que le comportement normal d'une sonde consiste uniquement aux fausses alertes, et qu'il n'y a pas d'intérêt dans ce comportement. Tous les itemsets fréquents dans une rafale sont considérés comme normaux, qu'il y ait un ou une centaine d'entre eux dans la rafale. S'il y a un intérêt dans ce comportement normal, cette approche est limitée.

2.2.5 Analyse des causes racines avec Data Mining

Julisch utilise le data mining dans l'analyse des causes racines dans [Jul01, JD02, Jul03a]. La cause racine d'une alerte est le phénomène qui cause l'alerte. Des résultats empiriques montrent qu'un petit nombre de causes racines crée de grandes quantités d'alertes, soit plus de 90%. Des exemples de causes racines sont les mauvaises configurations, les piles TCP/IP défectueuses, et les proxies dont le comportement ressemble à des activités de balayage. Typiquement, ces phénomènes bénins ou normaux sont persistants, systématiques et répétitifs. Ils ne disparaissent pas d'eux-mêmes et génèrent un grand nombre d'alertes.

Julisch montre que beaucoup d'approches de corrélation existantes sont centrées sur les attaques, et par conséquent, ne sont pas faites pour supporter un grand nombre de faux positifs. Il propose de découvrir les causes racines en utilisant le clustering. Avec Dacier, il a également expérimenté cela avec des règles d'épisodes. Les deux approches visent à fournir des moyens de filtrer les alertes causées par des causes racines normales ou bénines.

Clustering Conceptuel

Le clustering conceptuel est une méthode de clustering priorisant la représentation des clusters. Il fournit des clusters qui sont faciles à interpréter, et il supporte assez bien des attributs catégoriques. Par exemple, les adresses IP et les types d'alertes sont des attributs catégoriques. Selon les observations de Julisch, les alertes causées par une cause racine sont souvent similaires, et de plus, peuvent être regroupées ensembles en clusters. Les causes racines peuvent être découvertes en interprétant les clusters d'alertes produits par la méthode de clustering proposée. La prochaine étape est soit d'enlever les causes racines, soit de construire des filtre pour enlever les alertes correspondantes.

Les alertes sont définies comme des n-uplés du produit cartésien $dom(A_1) \times \cdots \times dom(A_n)$, où $\{A_1, \ldots, A_n\}$ est l'ensemble des attributs des alertes et $dom(A_i)$ est le domaine de l'attribut A_i. L'attribut A_1 d'une alerte a est notée $a.A_1$.

Un *valeur d'attribut généralisée* est le nom d'un concept représentant un sous-ensemble de valeurs d'un domaine d'attribut $dom(A_i), i \in 1, \ldots, n$. Par exemple, pour les attributs d'adresses IP, une valeur généralisée pourrait être `firewall`, `DMZ`, ou `any-IP`, où le `firewall` pourraient déjà contenir plusieurs adresses IP. Le *domaine étendu* $Dom(A_i)$ est le $dom(A_i)$ étendu avec des valeurs d'attributs généralisées pour A_i. Une *alerte généralisée* g est un n-uplé $(Dom(A_1) \times \cdots \times Dom(A_n)) \setminus (dom(A_1) \times \cdots \times dom(A_n))$ i.e. toutes ses valeurs d'attributs sont généralisées.

La *hiérarchie généralisée* G_i pour un attribut est une taxonomie ou une hiérarchie définissant comment les valeurs d'attribut peuvent être généralisées. Dans d'anciens articles [Jul01, JD02], les hiérarchies étaient des arbres, mais dans [Jul03a] des graphes acycliques (DAG) sont utilisés. Supposons que nous avons deux éléments, $x, y \in Dom(A_i)$. y est appelé le *père* de x si la hiérarchie généralisée G_i contient un chemin direct de y vers x. On peut l'écrire $x \trianglelefteq y$. Etant donné deux alertes a, g, l'alerte g est le père de l'alerte a si $a.A_i \trianglelefteq g.A_i$, $i = 0, \ldots, n$ et on dit qu'une alerte généralisée modèle toutes les alertes non généralisées dont elle est le père.

La *dissimilitude* de deux éléments $x_1, x_2 \in Dom(A_i)$, $d(x_1, x_2)$, est la longueur du plus petit "pas" entre ces éléments dans la hiérarchie généralisée G_i. La dissimilitude de deux alertes g et a, est la somme de ses dissimilitudes d'attributs

$$d(g, a) = \sum_{i=1}^{n} d(g.A_i, a.A_i).$$

Nous avons toujours besoin des concepts *dissimilitude moyenne* et *hétérogénéité*, et *couverture* d'un cluster pour définir le problème de clustering de l'alerte. Prenons un cluster C d'alertes et une alerte généralisée g qui est père de toutes les alertes dans le cluster, $a \trianglelefteq g, \forall a \in C$. La taille du cluster est notée $|C|$. Maintenant, la dissimilitude moyenne est définie par

$$\bar{d}(g, C) = \frac{1}{|C|} \sum_{a \in C} d(g, a) \quad ,$$

et l'hétérogénéité par

$$H(C) = \min\left\{\bar{d}(g, C)\right\} \quad .$$

De plus, l'alerte généralisée g pour laquelle la dissimilitude moyenne du cluster est minimale, $\bar{d}(g, C) = H(C)$ est appelé la couverture de C.

Etant donné une alerte log L, un seuil $\sigma \in \mathbb{N}$, et des hiérarchies généralisées G_i, $i = 1, \ldots, n$, le problème de clustering d'alerte est de trouver un cluster $C \subseteq L$ avec

- une taille ne devant pas être plus petite que le seuil définit, $|C| \geq \sigma$, et
- une hétérogénéité minimale $H(C)$.

Pour traiter plus loin l'alerte log, une fois que l'hétérogénéité minimale du cluster parmi tout $C \subseteq L$ avec $|C| \geq \sigma$ a été atteinte, d'autres clusters peuvent être recherchés à partir des alertes restantes $L \setminus C$. Pour une discussion plus détaillée, le lecteur est invité à lire [Jul03a]. L'article fournit plus d'information sur la construction des attributs généralisés pour différents types d'attributs, comme numériques, temps et chaînes de caractères. Définir des hiérarchies généralisées est une tâche non triviale, similaire pour écrire des descriptions de scénarios.

Le problème de clusterisation d'alertes est un problème NP-complet, et Julisch propose d'étendre la condition sur l'hétérogénéité minimale pour atteindre une solution praticable. Il propose une approche heuristique, qui est une variante de *attribute-oriented induction* (AOI) pour les clusterisations d'alertes.

Les clusters d'alertes aident à identifier la cause racine des alertes dans le cluster. Julisch propose alors d'enlever les causes racines, et dans le cas où ce n'est pas possible, de les filtrer. Par exemple, les causes racines peuvent être hors du contrôle de l'opérateur ou alors trop chères à réparer. Julisch a reporté que les filtres sur les données réelles capturent 82% des alertes. L'approche est efficace pour filtrer les faux positifs, mais selon l'approche de Manganaris et al., elle est censée filtrer toutes les alertes liées aux causes racines. S'il existe un intérêt dans le comportement de la cause racine et qu'elle a besoin d'un filtrage plus sélectif, la partie *filtrage* de l'approche aura besoin d'être élaborée.

Règles d'épisodes

Dans [JD02] Julisch et Dacier ont également utilisé des règles d'épisodes dans l'analyse d'alertes. Les règles d'association utilisées dans [MCZH99] traitent des ensembles de données désordonnés. Cependant, la détection d'intrusion d'alertes se produit en séquence, et leur ordre et le temps d'occurence sont des aspects importants à prendre en compte dans l'analyse.

Un épisode est une combinaison d'événements avec un ordre partiellement spécifié. Un épisode *se produit* dans une séquence si les événements de l'épisode apparaît dans un ordre définit par l'épisode et dans un temps donné [MTV97].

En utilisant les notations de [MT96], un épisode P sur des variables $\{x_1, \ldots, x_k\}$ notées $P(x_1, \ldots, x_n)$ est une conjonction

$$\bigwedge_{i=1}^{k} \varphi_i(y_i, z_i),$$

où $y_i, z_i \in \{x_1, \ldots, x_k\}$ sont des variables événementielles, et $\varphi_i(x, y)$ est de la forme $\alpha(x.A_1)$, $\beta(x.A_1, y.A_2)$, ou $x.T \leq y.T$. Ici, α est un prédicat unaire dans $dom(A_1)$, β

est un prédicat binaire dans $dom(A_1) \times dom(A_2)$, et $x.T$ est le temps que l'événement x apparaisse.

Un épisode apparaît dans une séquence $S = (e_1, \ldots, e_n)$ dans l'intervalle $[t, t']$ s'il y a des événements disjoints $e_{j_1}, \ldots, e_{j_k}$ tels que $P(e_{j_1}, \ldots, e_{j_k})$ soit vrai et que tout les e_{j_i} soient dans l'intervalle $[t, t']$. L'occurence de l'épisode P dans l'intervalle $[t, t']$ est *minimale* si P n'apparaît dans aucun sous-intervalle propre $[u, u'] \subset [t, t']$.

Les épisodes sont appelés *séries* d'épisodes si x_i a besoin d'être d'ordre total. Les épisodes parallèles n'ont pas de conditions respectant l'ordre relatif des événements. La fréquence d'un épisode P d'une séquence S est le nombre d'occurrences minimales de P dans S.

Une *règle d'épisode* concernant deux épisodes P et Q est de la forme $P[W_1] \Rightarrow Q[W_2]$, où W_1, W_2 sont des nombres réels indiquant la taille de la fenêtre. La règle d'épisode indique que si l'épisode P a une occurrence minimale dans l'intervalle $[t, t']$ avec $t' - t \leq W_1$, alors l'épisode Q apparaît dans l'intervalle $[t, t''], t'' - t \leq W_2$. La probabilité que l'épisode Q apparaisse si P est apparue est appelée *confiance*.

Julisch et Dacier utilisent les alertes comme des événements $\varphi_i(x, y)$ de la forme $\alpha(x.A_1)$, et limitent les règles dans la forme où P est un sous-episode de Q et $W_1 = W_2$. Ils ont reporté avoir trouvé quelques modèles d'alertes intéressants, comme un outil d'attaque avec un temps d'envergure de plus de 16 heures. Cependant, ils considèrent des règles d'épisodes peu convenables pour le traitement d'alertes étant donné que 1) le degré possible d'automatisation est bas, et 2) l'épisode mining a tendance à produire un grand nombre de modèles insignifiants.

2.3 Classification d'alerte par degré de vérité et d'importance

Le problème des fausses alertes n'est pas aussi noir et blanc qu'on le croit. En grande partie parce que les gens inventent des nouvelles utilisations pour les systèmes de détection d'intrusion, nous devons aussi considérer les alertes qui sont principalement issues des événements bénins par défaut. Il convient de noter que cela s'applique plus spécifiquement aux sondes par scénario étant donné qu'elles sont capables de labelliser les alertes avec des noms précis. Les sondes de détection d'anomalies, par définition, recherchent les comportements anormaux qu'elles pouront utiliser pour détecter des violations non intrusives de politique, souvent elles ne labellisent pas les anomalies détectées[23] Fournir des diagnostics est moins précis, et notre classification n'aboutit pas nécessairement.

2.3.1 Trois classes d'alertes

Dans [KR04] Kruegel et Robertson utilisent trois classes pour des alertes issues par des sondes : les vrais positifs, les positifs non pertinents, et les faux positifs. Le vrai positif est une attaque réussie correctement identifiée par la sonde. Le positif non pertinent est également correctement identifié, mais l'attaque n'est pas réussie. Le faux positif est un événement bénin incorrectement identifié comme attaque. Ils ont ajouté aux définitions classiques la notion de succès. A cause des nouvelles utilisations des sondes, nous pensons

[23] Un contre-exemple : eBayes-TCP [VS00] a fourni une classification plus précise que normal/anormal, pour les sessions TCP analysées. Même avec 13 classes de connexions dans sa forme initiale, avant une génération d'hypothèses dynamiques possibles, c'est toujours à granularité plus large que des sondes misuse-base.

qu'il est nécessaire d'ajouter aussi les notions de type d'alerte et d'importance d'événement en respectant la politique du système et la politique de sécurité. De plus, nous étendons la classification comme suit :

Vrai positif (t_+) est une identification correcte. Refouler à partir d'une utilisation classique d'une sonde, cela peut être une attaque réussie, correctement identifiée par la sonde. Alternativement, cela peut être un événement de non-attaque qui viole une politique du système et requiert une action immédiate. Des exemples de ce dernier pourraient être l'utilisation de telnet pour se connecter à un routeur, dans un environnement de sécurité, ou une tentative de connexion à une hôte en dehors de l'organisation à partir d'un serveur contenant des données hautement sensibles. En fait, ce ne sont pas des attaques, mais elles pourraient être des violations de politiques requiérant une action immédiate.

Les moteurs de corrélation détectant les scénarios d'attaques sont conçus pour traiter ce type d'alertes, des approches de reconnaissance de différents scénarios visent à les grouper ensemble pour fournir une vue synthétique à l'opérateur. Ces méthodes incluent les articles mentionnés précédemment [MD03,CM02,CO00,NCR02a,NCR02b, CLF03,TK00,VS01,DC01,DC02]. Les processus de priorité d'alerte peuvent également accentuer les attaques qui sont considérées comme plus critiquables pour le système d'information surveillé et ses missions, comme présenté dans [PFV02].

Faux positif (f_+) est un événement incorrectement identifié. Là encore, par les définitions traditionnelles, c'est un événement bénin pris pour une attaque. Dans le cas actuel, cela pourrait être un événement lié au fonctionnement du système qui était incorrectement identifié.

Les approches de data mining [MCZH99, Jul01, JD02, Jul03a] vu dans l'état de l'art Sect. 2.2 dressent le problème des faux positifs. Ces méthodes aident à construire des filtres qui enlèvent les faux positifs de manière efficace. Les chroniques [MD03] peuvent également être utilisées pour détecter des scénarios de faux positifs pour un effet similaire de filtrage. En effet, les approches de data mining peuvent aider un analyste écrivant ces types de chroniques pour découvrir des scénarios fréquents de faux positifs.

Positif non pertinent (irr_+) est une identification correcte qui ne requiert pas une action immédiate. Cela peut être une attaque correctement identifiée mais qui échoue. Alternativement, nous pouvons avoir un événement bénin correctement identifié mais qui est surveillé pour être au courant de tout l'état de sécurité du système, du dépannage, pour soutenir les investigations, ou vérifier le respect des politiques d'utilisation. Un exemple de positifs non pertinents est une organisation avec des politiques restreignant l'utilisation des applications de fichiers partagés P2P. Là, la détection ne requièrerait pas une action immédiate de l'opérateur, mais par exemple une discussion face-à-face avec le responsable de l'utilisateur. De plus, il n'y a aucun besoin d'envoyer ce type de positifs à une console en temps réel, mais peut-être de résumer en un rapport journalier/hebdomadaire/mensuel au responsable. Un exemple de surveillance de l'état général de sécurité peut être de suivre l'utilisation SNMP dans le réseau. Snort a plusieurs signatures qui déclenchent les événements liés au SNMP qui, par définition, n'ont pas besoin d'être malveillants. La plupart du temps, ils sont correctement détectés. L'alerte identifie un message SNMP qui passe dans le réseau. Cependant, des alertes individuelles sont souvent non pertinentes pour l'opérateur.

Les approches de vérification et de prioritisation d'alertes comme dans [KR04, PFV02] aident à enlever le premier type de positifs non pertinents, des attaques échouées. Le second type, causé par les nouvelles utilisations des sondes, est un problème plus difficile.

- La vérification d'alerte n'est pas adaptée étant donné que ces alertes sont typiquement précises
- Les approches de prioritisation d'alertes peuvent être utilisées pour affecter une priorité basse à de telles alertes. Cependant, cela peut être, et ça l'est déjà, donné dans les définitions de signature.
- Les méthodes de data mining aidant dans les deux analyses d'alertes et de développement de filtre, aideraient probablement à construire des filtres efficaces aussi pour ces alertes. Cependant, étant donné que les signatures générant ces alertes sont activées, il y a un certain intérêt pour ces alertes. Si les alertes n'ont aucune valeur, une solution plus facile serait d'enlever ces signatures.

Nous concluons que des mécanismes de corrélation existants ne sont pas conçus pour ce type d'alertes. Nous argumentons que nous avons besoin de techniques de filtres plus sophistiquées que celles basées sur des attributs d'alertes. Nous développerons plus loin l'analyse de donnée d'alertes dans Chap. 3.

Nous pensons qu'une proportion significative des alertes générées par des sondes IDS peuvent être des positifs non pertinents. Dans la littérature de corrélation d'alerte, la division a souvent été binaire entre faux ou vrais positifs , mais avec notre division, une large quantité d'alertes peuvent également tomber dans la catégorie des positifs non pertinents.

Manganaris et al. [MCZH99] mentionnent des alertes innocentes et non intéressantes, sans vraiment les détailler. Le taux de 99% de faux positifs reporté par Julisch [Jul01, Sect. 1] est basé sur ses propres observations et celles de Manganaris. Julisch utilise trois exemples comme causes racines pour les fausses alertes : une pile TCP/IP mal implémentée qui est à l'origine des alertes "IP fragmentées", un proxy apparaissant comme l'origine des scans vers un réseau externe, et un serveur DNS mal configuré qui fait des transferts de zone et déclenche les alertes associées.

Avec notre classification, le premier et le troisième exemple seraient des positifs non pertinents. Si une alerte réagit uniquement au trafic IP fragmenté, les alertes issues du trafic à partir du noeud avec la pile TCP/IP mal implémentée sont correctement identifiées. Si l'alerte faisait appel quelque chose d'autre, comme une attaque par évasion, ce serait un faux positif. Comme nous n'avons pas accès aux signatures NetRanger, nous ne pouvons pas vérifier le message actuel d'alerte.

Dans le troisième exemple, si les alertes de la zone de transfert DNS "DNS Zone Transfer" réagissent uniquement au trafic DNS contenant la requête pour la zone de transfert, la signature détecte correctement la requête, même si dans ce cas les alertes sont non pertinentes. Là encore, nous ne pouvons pas vérifier la signature actuelle. Pour finir, la signature Snort `DNS zone transfer TCP`[24] et `DNS zone transfer UDP`[25] réagit aux valeurs numériques correspondant au opcode AFXR à un certain offset dans le paquet du port 53.

Le deuxième exemple est également un faux positif selon notre classification, étant donné que les alertes signalent un événement qui n'a pas lieu.

[24] `http://www.snort.org/pub-bin/sigs.cgi?sid=255`, visited 2006-07-18
[25] `http://www.snort.org/pub-bin/sigs.cgi?sid=1948`, visited 2006-07-18

2.3.2 Considérations de précision par signature

Les taux de détection bayésiens indiquent la confiance que nous pouvons avoir en les alertes issues des sondes. Cette classification en trois classes complique l'analyse du taux de base décrit dans Sect. 2.1.4. Souvent, les taux de faux positifs sont affectés aux sondes. Dans cette section, nous argumentons que des taux extrêmement élevés de faux positifs peuvent être considérés signature par signature à cause des grandes différences entre les signatures. Par conséquent, les taux de détection bayésiens devraient également être considérés par signature comme indicateur de confiance que l'on peut avoir en cette signature particulière.

Les mêmes applications principales : nous aimerions avoir un taux de détection bayésien et un taux négatif de détection bayésien aussi large que possible. Cependant, étant donné que les descriptions d'alertes peuvent décrire des événements bénins, le taux de base $\mathrm{P}(I)$ devrait être interprété comme le taux d'événements indiqué par les signatures au lieu d'uniquement le taux d'intrusions.

Il peut y avoir de grandes différences entres les signatures et les environnements dans le taux de base, tout comme dans les taux des faux positifs et des faux négatifs. Pour illustrer cela, considérons deux signatures, `ICMP PING NMAP`[26] et `SNMP request udp`[27]. Le premier signale une récolte d'informations définitive et le deuxième une récolte d'informations éventuelle. Identifier correctement des pings NMAP peuvent aussi bien être des vrais positifs que des positifs non pertinents, étant donné la politique de sécurité de l'organisation et la tâche de l'opérateur. Ceci est également vrai pour les requêtes SNMP sur UDP, même si les alertes ont plus tendance à être des positifs non pertinents. Les faux positifs pour `ICMP NMAP PING` ont été reportés, en raison de la signature plutôt générale, alors que nous ne sommes au courant d'aucun rapport de faux positifs pour `SNMP request udp` [28].

Maintenant, tous les termes de (2.1), i.e. le taux de vrai positif, $\mathrm{P}(A|I)$, le taux de faux positif, $\mathrm{P}(A|\neg I)$, et le taux de base $\mathrm{P}(I)$ (et par conséquent également $\mathrm{P}(\neg I)$) sont différents pour chaque signature, et dépendent de l'environnement où la sonde est déployée. Dans les environnements où SNMP est très largement utilisé, le taux de base pour `SNMP request udp` peut être élevé en proportion des événements de balayage, sans parler des activités malveillantes. Par exemple, nous avons vu des taux constants de trafic SNMP à des taux extrêmement stables variant de 84 alertes par heure à des taux variant à plus de 100 par minute.

Nous n'avons pas de comptes de paquets correspondant aux périodes d'observation pour estimer les taux de base actuels, mais nous pouvons dire qu'il y a des différences entre les signatures. Par exemple, dans `set-3` il y a 8 065 524 alertes `SNMP request udp`, et 494 517 alertes `ICMP PING NMAP`. En supposant que l'alerte reflète approximativement les vrais taux de paquets SNMP et NMAP, et en faisant la moyenne sur tout l'intervalle d'observation, le taux de paquets SNMP est 16 fois plus grand. Cependant, comme l'a indiqué Axelsson, le terme $\mathrm{P}(\neg I) \cdot \mathrm{P}(A|\neg I)$ domine dans le dénominateur de (2.1). Il se pourrait que même à des centaines de paquets par minute, $\mathrm{P}(\neg I)$ ne soit pas affecté. Si pour `SNMP request udp` nous avons un taux de zéro faux positif $\mathrm{P}(A|\neg I) = 0$, le taux de

[26] `http://www.snort.org/pub-bin/sigs.cgi?sid=469`, visited 2006-07-18

[27] `http://www.snort.org/pub-bin/sigs.cgi?sid=1417`, visited 2006-07-18

[28] Puisque la signature se déclenche sur tous les paquets udp envoyés au port 161, il n'y a rien qui empêche les faux positifs d'apparaître. Mais jusqu'ici, nous n'avons pas rencontré un tel trafic. Nous pensons également qu'il est moins probable que des requêtes d'écho ICMP se produisent sans contenu causant des faux positifs avec `ICMP PING NMAP`. De nombreuses applications utilisent les pings ICMP pour tester la connectivité, alors que l'utilisation légitime du port 161 pour autre chose que SNMP est improbable

détection bayésien est affecté.

Puisque le taux de détection bayésien peut varier beaucoup entre les différentes signatures, l'analyse est plus utile qu'une évaluation de la sonde dans l'ensemble. Considérons un process de corrélation prenant en compte la confiance en l'analyse fournie par les différentes sondes. Une estimation de confiance globale pour une sonde anormale serait raisonnable, mais pour une sonde de base, cela pourrait être trompeur.

Même si quelques signatures peuvent être très précises, le problème persiste. Si la machine de l'opérateur est inondée d'alertes correctes mais non pertinentes, il est susceptible d'apprendre à ignorer toutes les alertes dans le cas des fausses alertes [Axe99, Sect. 4.2]. De plus, de nombreuses alertes non pertinentes soumettent également une contrainte aux moteurs de corrélation qui traitent les alertes en ligne, par exemple, les scénarios de reconnaissance.

2.4 Conclusion

Dans la section 2.1 nous avons vu les raisons causant les flux d'alertes. La plupart des problèmes sont les faits par lesquels nous avons besoin de nous adapter. Concernant le grand volume global que les sondes doivent analyser, nous ne pouvons pas le changer au niveau de la corrélation. Au niveau de la sonde, la situation peut être améliorée avec des données sources plus denses telles les applications de logs. Une autre possibilité est d'essayer d'utiliser des méthodes plus efficaces pour filtrer les événements normaux avant d'utiliser des analyses de composants plus coûteuses en terme de ressources.

Les problèmes liés aux limites des sondes et des nouvelles utilisations peuvent cependant être traités par la corrélation d'alertes dans une certaine mesure, et nous avons présenté l'état de l'art en corrélation d'alertes dans Sect. 2.2. Le travail était centré sur des méthodes implicites ayant pour but de réduire le volume, comme dans cette thèse en général. La prioritisation d'alerte et les approches de scénario de détection ont pour but de résoudre un problème différent. Les méthodes de réduction de volume existantes sont conçues pour supprimer efficacement les faux positifs. Les règles d'association utilisées par Manganaris et al. sont capables de signaler également les alertes manquantes qui auraient dû apparaître normalement. Cependant, ces mécanismes de corrélation sont conçus pour les alertes produites par les nouvelles utilisations des sondes, et le type de positifs non pertinents produits par ces utilisations. Nous avons défini ce que nous voulons dire par différents types d'alertes, vrais positifs, faux positifs et positifs non pertinents dans Sect. 2.3.

Dans le chapitre suivant, nous analyserons avec plus de détail les alertes liées à la politique d'application et au fonctionnement d'utilisation normale du système. Nous fournirons des exemples de bases de données d'alertes réelles pour montrer que ces alertes ont des caractéristiques spécifiques, ce qui les rend encore plus difficiles à traiter par les approches de corrélation existantes. Nous verrons que nous aurons besoin d'approches de filtrage efficaces pour ce type d'alertes et que le filtrage aura besoin d'être basé sur plus d'attributs qu'une alerte individuelle.

Chapitre 3

Caractéristiques du bruit des alertes

Dans le chapitre précédent, nous avons étudié les causes de l'inondation d'alertes. Nous avons également établi que l'inondation contenait différents types d'alertes, des vrais positifs et des faux positifs, mais aussi des alertes que nous appelons des positifs non pertinents. Les études réalisées sur la corrélation des alertes se concentrent soit sur les vrais positifs, soit sur les faux positifs, mais aucune approche n'est conçue pour les positifs non pertinents liés à l'utilisation et au fonctionnement normaux du système. Nous appelons ces alertes *bruit des alertes* et nous développons des approches pour traiter le bruit.

Dans ce chapitre, nous aborderons les proportions dans lesquelles différentes causes contribuent à différents types d'alertes, en donnant des exemples de journaux d'alertes réelles à la section 3.1. Nous analyserons ensuite les caractéristiques du bruit des alertes à la section 3.2.

Il est essentiel de connaître les origines de ces alertes pour comprendre les raisons de cette surveillance. Les caractéristiques de ces alertes permettent de comprendre pourquoi les approches de filtrage des alertes existantes ne conviennent pas pour la tâche et quel type de traitement est nécessaire. Les raisons principales sont 1) que ces alertes doivent être traitées comme des flux d'alertes et non comme des alertes individuelles et que 2) même si les alertes ne nécessitent pas une réaction immédiate, elles véhiculent des informations utiles.

3.1 Décomposition du flux d'alertes

Dans cette section, nous établissons une correspondance entre les origines du flux et les types d'alertes. Nous analysons trois ensembles d'alertes sur lesquels nous basons notre analyse. Enfin, nous définissons le sous-ensemble d'alertes que nous souhaitons traiter. Ce sous-ensemble se compose principalement de positifs non pertinents, mais dans certains cas également de faux positifs, tous deux provoqués par l'utilisation et le fonctionnement normaux du système surveillé.

3.1.1 Données utilisées : Trois bases de données d'alertes

Nous présentons trois ensembles de données que nous appelons simplement `ensemble-1`, `ensemble-2` et `ensemble-3`. L'`ensemble-1` comprend des alertes générées sur une période de 43 jours au début de l'année 2003 par un site utilisant des systèmes de production.

Il contient environ 580K d'alertes. Le tableau 3.1 présente cinq des signatures les plus prolifiques et le nombre d'alertes générées par elles.

L'`ensemble-2` contient environ 3,3M d'alertes. Elles ont été collectées à la fin de l'année 2003, pendant une période de 100 jours, depuis le même site que l'`ensemble-1`. Les configurations soit du site, soit des sondes avaient changé entre les deux périodes de collecte de données ; en effet, certaines des signatures les plus prolifiques de l'`ensemble-1` n'existaient plus et de nouvelles signatures prolifiques avaient été introduites. Le tableau 3.2 montre les signatures qui ont généré plus de 10K d'alertes. Les deux lignes avec `WEB-PHP content-disposition` proviennent de deux sondes différentes que nous considérons séparément.

Un ensemble de données plus récent, l'`ensemble-3`, a été collecté au dernier semestre de l'année 2005, sur une période de 43 jours. Il totalise environ 37M d'alertes et provient d'un site de recherche comportant quelques serveurs opérationnels, fournissant principalement des services pour l'intranet de l'organisation. Le tableau 3.3 montre les signatures responsables de plus de 1M d'alertes.

L'ensemble des signatures par défaut des sondes était amplifié de quelques règles supplémentaires dans les trois ensembles. Dans le cas de l'`ensemble-1` et de l'`ensemble-2`, les signatures supplémentaires étaient des règles locales, tandis que dans l'`ensemble-3`, quelques règles de Bleeding Snort[1] avaient été activées en plus de l'ensemble par défaut. Dans l'`ensemble-1`, 315 signatures ont généré des alertes. Seules cinq des signatures les plus prolifiques étaient responsables de 68 % des alertes. Dans l'`ensemble-2`, 420 signatures ont généré des alertes, les cinq plus prolifiques étant à l'origine de 84 % et les signatures du tableau 3.2 étant la cause de 95 % des alertes. Dans l'`ensemble-3`, 217 signatures ont généré des alertes, les cinq les plus prolifiques étant la cause de 69 % et les signatures du tableau 3.3 étant la cause de 78 % des alertes.

L'ensemble d'alertes, l'`ensemble-1`, a été le premier ensemble à notre disposition. Il a déjà été utilisé lors des premiers tests. C'est pour cette raison qu'il sert d'ensemble de référence tout au long de la présente thèse. Les cinq signatures seront décrites plus en détail à la section 3.1.3, ainsi que dans les caractéristiques des alertes générées par ces signatures à la section 3.2. En ce qui concerne les ensembles de données, l'`ensemble-2` et l'`ensemble-3`, nous nous concentrons principalement sur les caractéristiques de génération des alertes.

TAB. 3.1 – Cinq des signatures les plus prolifiques de l'`ensemble-1` en 2003. Elles représentent 68 % de toutes les alertes. Les signatures concernent principalement les fonctions du système de surveillance.

nom de signature	**alertes**
SNMP Request udp	176 009
ICMP PING WhatsupGold Windows	72 427
ICMP Destination Unreachable (Comm Adm Proh)	57 420
LOCAL-POLICY External connexion from HTTP server	51 674
ICMP PING Speedera	32 961
total	**390 491**
toutes les alerts	**578 301**

[1] `http://www.bleedingsnort.com/`, consulté le 18.07.2006.

TAB. 3.2 – Signatures ayant généré plus de 10K d'alertes dans l'`ensemble-2` de 2003. Les cinq premières sont responsables de 84 % et l'ensemble du tableau de 95 % des alertes. Les signatures concernent principalement le fonctionnement du système de surveillance (ICMP, SNMP) et l'application des politiques (IRC, MSN). Les messages IRC pourraient également être liés au contrôle du réseau des zombies. La proportion d'alertes liées aux vers et à l'outil DDoS est plutôt modeste.

nom de signature	**alertes**
ICMP PING CyberKit 2.2 Windows	1 280 647
CHAT IRC message	1 063 237
ICMP Destination Unreachable (Comm Adm Proh)	250 523
ICMP PING speedera	126 058
(frag2) TTL Limit Exceeded (reassemble) detection	103 655
ICMP Large ICMP Packet	76 236
(stream4) TTL LIMIT Exceeded	50 227
ICMP L3retriever Ping	40 268
CHAT MSN message	29 406
WEB-IIS view source via translate header	26 715
CHAT IRC nick change	24 860
BAD-TRAFFIC loopback traffic	23 903
WEB-PHP content-disposition	23 766
WEB-PHP content-disposition	19 487
(stream4) STEALTH ACTIVITY (SYN FIN scan) detect	18 829
DDOS Stacheldraht agent->handler (skillz)	14 797
LOCAL-WEB-IIS Nimda.A attempt	13 700
ICMP Dest Unr (Comm w/ Dest Host Adm Proh)	11 983
total	**3 198 297**
toutes les alertes	**3 354 860**

Nom de la signature Alertes

SNMP request udp 8 065 524 SNMP public access udp 6 895 768 ICMP L3retriever Ping 6 270 314 (http inspect) BARE BYTE UNICODE ENCODING 2 310 954 NETBIOS SMB-DS DCERPC NTLMSSP asn1 oflow att 1 936 139 NETBIOS SMB-DS IPC$ share unicode access 1 901 695 NETBIOS SMB IPC$ share unicode access 1 362 219

Total 28 742 613 Toutes les alertes 36 862 451

3.1.2 Décomposition

La figure 3.1 illustre la manière dont différentes causes contribuent aux différents types d'alertes du flux. Il est à noter que cette répartition n'est pas exclusive, il peut y avoir plusieurs causes à une alerte. Nous examinerons ensuite la cause de chaque flux d'alertes, les types de nuisances provoquées et leur signification dans les ensembles de données.

Types causés par les limitations des sondes

Les limitations des sondes contribuent à la fois aux faux positifs et aux positifs non pertinents. Les faux positifs peuvent être générés par les limitations des méthodes de détection, comme la signature `ICMP PING NMAP` de Snort susmentionnée, se déclenchant sur des messages d'écho ICMP d'une longueur de contenu 0. Le manque de connaissance

TAB. 3.3 – Signatures ayant généré plus de 1M d'alertes dans l'`ensemble-3` de 2005. Les cinq premières sont responsables de 69 % et tout le tableau, de 78 % des alertes. Les sources de bruit les plus significatives sont les signatures donnant l'alerte sur le trafic administratif (SNMP, ICMP), les alertes liées à l'utilisation normale du système étant la deuxième source par ordre d'importance (NETBIOS)

signature name	**alerts**
SNMP request udp	8 065 524
SNMP public access udp	6 895 768
ICMP L3retriever Ping	6 270 314
(http_inspect) BARE BYTE UNICODE ENCODING	2 310 954
NETBIOS SMB-DS DCERPC NTLMSSP asn1 oflow att	1 936 139
NETBIOS SMB-DS IPC$ share unicode access	1 901 695
NETBIOS SMB IPC$ share unicode access	1 362 219
total	**28 742 613**
toutes les alertes	**36 862 451**

de la configuration du système surveillé provoque des positifs non pertinents, comme les attaques IIS ciblant les serveurs qui exécutent Apache.

Types causés par des problèmes de configuration

Les problèmes de configuration peuvent contribuer aux faux positifs en combinaison avec le manque de connaissance de l'environnement. Par exemple, la signature `ICMP L3retriever Ping` de Snort est connue pour créer de faux positifs avec les contrôleurs de domaines Windows dans les réseaux Windows[2]. Ces problèmes pourraient être évités, soit en ignorant le trafic ICMP de et vers les contrôleurs de domaines dans la configuration des sondes, soit en analysant la situation plus en détail et en utilisant le seuillage avec des limites fixes fournies par Snort. Ce type de seuillage ne peut toutefois pas dépasser cette limite et se heurte aux limitations des sondes.

Si quelqu'un ou quelque chose, un ver par exemple, attaque des adresses IP aléatoires avec un exploit de serveur Web, les alertes émises sur les paquets transmis vers des machines qui n'exploitent pas de serveur Web seraient non pertinentes. Si les serveurs Web étaient connus, ils pourraient être énumérés dans la configuration des sondes et, de ce fait, les signatures correspondantes pourraient être utilisées uniquement contre le trafic destiné à ces hôtes. Cela empêcherait toutefois la détection de compromission de n'importe quel serveur qui n'est pas entré explicitement dans le fichier de configuration. Pour cette raison, l'administrateur sécurité peut choisir de ne pas utiliser ce type de configuration statique et nous aurions des positifs non pertinents liés à la configuration. Ces derniers pourraient être évités si le sonde procédait à une analyse plus détaillée, y compris les messages protocolaires TCP et port inaccessible ICMP.

[2]Par exemple, Javier Fernandez-Sanguino, *L3Retriever false positives* dans la liste de diffusion `snort-sigs`, 25.01.2005.

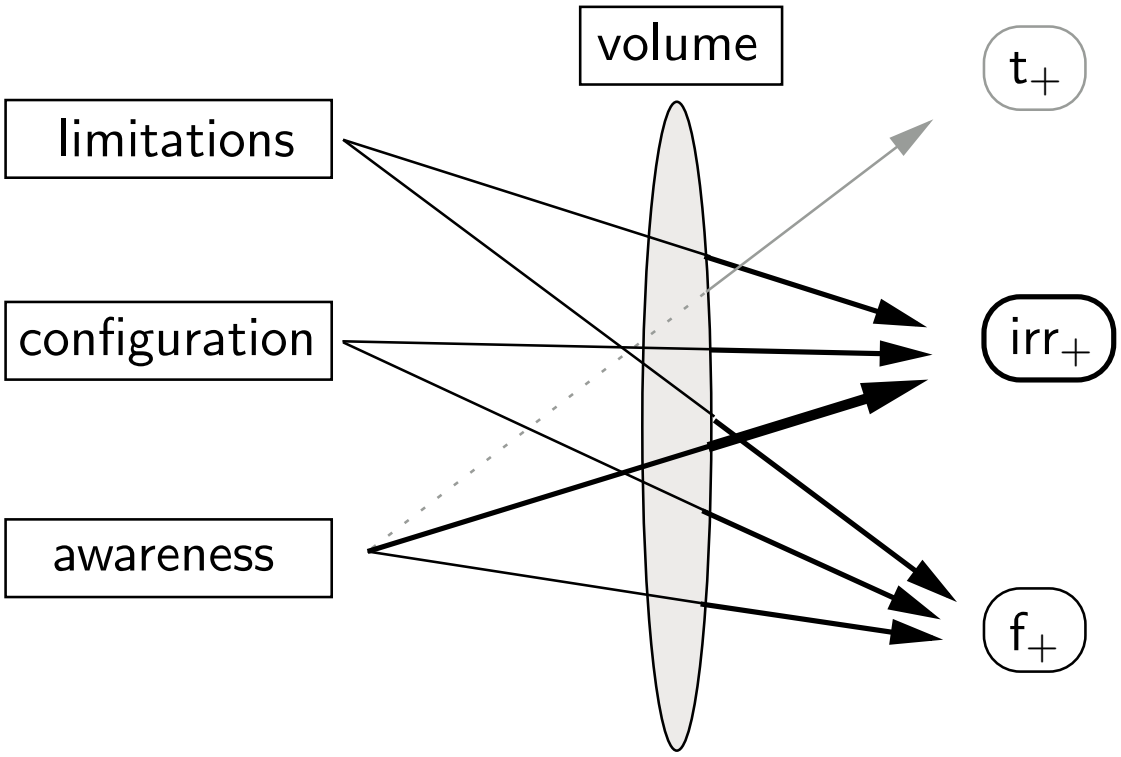


FIG. 3.1 – Différentes causes du flux d'alertes. Il peut s'agir de limitations des sondes, de problèmes de configuration, de nouvelles utilisations de sondes et d'un problème général d'importants volumes de données qui exigent une analyse. Les causes de la génération des alertes peuvent se chevaucher et l'inondation d'alertes contient différents types d'alertes, des vrais positifs t_+, des positifs non pertinents irr_+ et des faux positifs f_+. Les flèches indiquent quelles causes provoquent quels types d'alertes, tandis que les flèches plus larges et plus sombres indiquent des volumes plus importants dans nos observations

Types provoqués par de nouvelles utilisations

Les nouvelles utilisations, appelées *informatives* à la figure 3.1 génèrent, de manière typique, d'importants volumes d'alertes non pertinentes. Le taux de base des événements liés à l'utilisation normale du réseau et à l'administration est typiquement élevé par rapport au comportement intrusif. La même chose peut également s'appliquer à des événements liés aux politiques de sécurité ou d'utilisation. Par conséquent, les signatures informatives ont tendance à créer de grands nombres d'alertes. Les alertes individuelles ne sont souvent pas pertinentes, la signification doit être déterminée dans le contexte d'autres alertes. Dès lors, combinées à des sondes limitées qui analysent les événements un par un, une grande proportion d'alertes informatives n'est pas pertinente.

Les mêmes limitations de sondes et les mêmes difficultés d'élaborer de bonnes signatures s'appliquent tant avec les signatures informatives qu'avec les signatures intrusives. Par conséquent, ce type de signatures peut également créer de faux positifs. Etant donné la politique de sécurité de l'organisation, de simples scannages peuvent être considérés comme alertes informatives. Dans ce cas, les faux positifs `ICMP PING NMAP` seraient considérés comme faux positifs informatifs.

Les signatures informatives peuvent également être utilisées pour répondre aux événements liés à la politique ou au comportement du système et qui nécessitent une action immédiate de la part de l'opérateur. Dans ce cas, l'événement identifié correctement serait considéré comme vrai positif. Dans des conditions normales, des événements inoffensifs sont plus fréquents que les événements problématiques, si bien que le nombre de vrais positifs devrait en principe être nettement inférieur au nombre de positifs non pertinents.

Proportions des différents types d'alertes

La proportion de t_+, irr_+ et f_- varie entre les différents réseaux, tout comme la signification des effets des limitations des sondes, des problèmes de configuration et des utilisations informatives. Sur la base de nos observations, nous soulignons la relation informative – positif non pertinent comme la responsable majeure du flux d'alertes. D'après les tableaux 3.1, 3.2, et 3.3, nous pouvons constater que :

- Une petite proportion des signatures est responsable de la majorité des alertes.
- Ces signatures prolifiques sont souvent informatives. La surveillance du trafic SNMP est la cause d'alertes la plus significative de l'`ensemble-1` et de l'`ensemble-3`. Les signatures ICMP sont également fortement présentes dans tous les ensembles de données. Le comportement intrusif réel est détecté par seulement quelques signatures prolifiques et il s'agit de comportements volumineux de nature, à savoir des vers et des outils DDoS. Dans l'`ensemble-2`, nous avons une signature locale se déclenchant sur les tentatives de propagation du ver Nimda, LOCAL-WEB-IIS Nimda.A attempt et un zombie DDoS signalant à son maître qu'il est vivant et attend des instructions, DD0S Stacheldraht agent-¿handler (skillz)[3]. L'`ensemble-2` contient également des règles d'application de la politique liées à l'utilisation IRC et IM. Il est à noter que l'utilisation IRC peut également être considérée comme une surveillance informative, étant donné que plusieurs zombies se connectent aux serveurs IRC pour obtenir des ordres[4].
- Etant donné le nombre de faux positifs généralement signalés dans la littérature, la proportion d'alertes que nous souhaiterions éliminer par filtrage sans autre considération est étonnamment petit.

3.1.3 Bruit des alertes

A la section précédente, nous avons vu comment les différentes origines des inondations créaient différents types d'alertes dans l'inondation. Nous nous concentrons sur un sous-ensemble de ces alertes consistant en majorité de irr_+ informatifs, mais aussi dans une certaine mesure de f_+ provoqués par les limitations des sondes.

La figure 3.2 montre différents types d'alertes classés en fonction soit de la nature de l'événement qui déclenche l'alerte, soit de l'axe vérité-pertinence. La nature est soit intrusive soit informative et, sur l'axe vérité-pertinence, nous avons les vrais positifs, les faux positifs et les positifs non pertinents. La zone marquée et les flèches plus sombres et plus épaisses montrent les composantes du bruit des alertes. Les feuilles ne sont pas exclusives, étant donné que chaque alerte peut être classée selon deux critères. Les tirets montrent la relation entre les classifications liées qui seront discutées ci-après.

[3] `http://www.snort.org/pub-bin/sigs.cgi?sid=1855`, consulté le 18.07.2006.

[4] Know Your Enemy : Tracking Botnets, `http://www.honeynet.org/papers/bots/`, 13.03.2005, consulté le 18.07.2006.

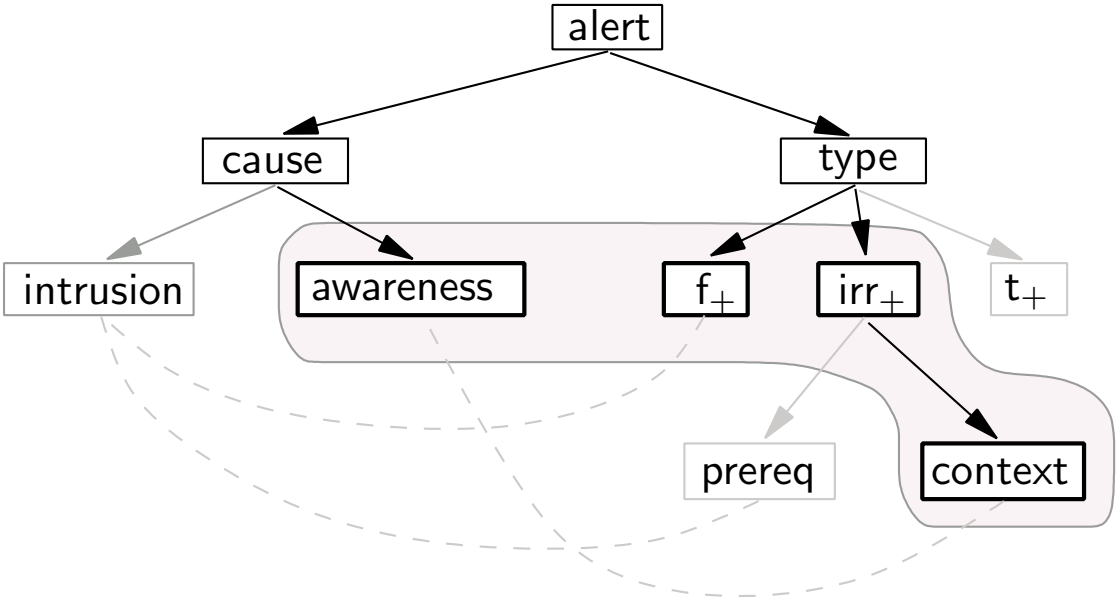


FIG. 3.2 – Différents types d'alertes. La zone marquée et les lignes plus sombres et plus épaisses montrent les catégories d'alertes que nous visons. Les lignes en pointillés montrent la relation entre la cause et la classification des types

Pour commencer, un cas très précis est l'exclusion des vrais positifs pertinents. Notre intention n'est pas de traiter des vrais positifs et positifs pertinents, c'est-à-dire des alertes nécessitant une action immédiate de l'opérateur. Cette catégorie d'alertes est l'entrée adéquate pour de nombreux moteurs de corrélation existants.

Lorsque nous pensons aux alertes du point de vue de la cause, nous avons deux classes, les intrusives et les informatives. Les alertes intrusives ont tendance à provoquer également des faux positifs et des positifs non pertinents en plus des vrais positifs. Si nous avons un positif non pertinent intrusif, l'exemple type est une attaque correctement identifiée n'ayant aucune chance de réussite en raison du manque de prérequis. Souvenons-nous de l'exploit IIS utilisé contre les serveurs Web Apache. Notre intention n'est pas de traiter ce type d'alertes. Différentes techniques de vérification d'alertes abordent ce type de problème.

Faux positifs intrusifs

Les faux positifs intrusifs sont la première catégorie d'alertes que nous estimons intéressantes. Par exemple, dans l'`ensemble-3`, nous avons `NETBIOS SMB-DS DCERPC NTLMSSP asn1 overflow attempt` qui est très probablement un flux de faux positifs. En raison des limitations des sondes, il n'est pas possible de faire la différence entre les paquets NETBIOS inoffensifs et les exploits réels. Si le fonctionnement ou l'utilisation normal du système déclenche ces faux positifs, ils sont susceptibles 1) d'intervenir en grands nombres et 2) de refléter une certaine forme de régularités associées au fonctionnement ou à l'utilisation normal(e) du système provoquant ces alertes. Si les alertes de cette catégorie doivent être surveillées, cela nécessiterait des méthodes de traitement automatisées pour éliminer par filtrage un grand nombre de faux positifs. Autre exemple : les faux positifs déclenchés par

la signature `ICMP PING NMAP` - si une application légitime déclenche ces alertes et que, en même temps, il n'est pas possible de désactiver la signature, nous devrions être capables d'éliminer par filtrage les alertes liées au comportement normal.

Si l'on devait prendre un exemple quelque peu paranoïaque, supposons que votre réseau utilise le gestionnaire de livraisons `Kontiki`[5] pour fournir des contenus vidéos aux utilisateurs et qu'il déclenche de fausses alertes `ICMP PING NMAP`. Les approches de corrélation, dont l'objectif est de reconnaître le scénario, sont susceptibles de laisser ce type d'alerte seul, ou alors des approches implicites telles que [QL03] peuvent mettre en corrélation ces alertes et n'importe quelles autres alertes. L'approche de regroupement conceptuelle de Julisch désignerait probablement le gestionnaire de livraisons `Kontiki` comme étant le responsable des alertes et, dans ce cadre, la cause primaire devrait être supprimée ou les alertes devraient être filtrées en tant que telles. Si, supposons, toutes les alertes `ICMP PING NMAP` survenant pendant les heures ouvrables avec l'hôte `Kontiki` comme source et des hôtes intérieurs comme cibles, doivent être filtrées, un attaquant compromettant l'hôte `Kontiki` peut scanner librement le réseau surveillé. Cet exemple est peu vraisemblable et doit être considéré comme tel si un tel risque vaut la peine d'essayer d'utiliser un filtrage à granularité plus fine pour ces alertes. Nous considérons certains faux positifs intrusifs comme des cibles possibles de méthodes de traitement explorées dans cette thèse. L'exemple décrit étant quelque peu paranoïaque, nous considérons les faux positifs de ce type uniquement comme une composante mineure du bruit des alertes.

Positifs non pertinents informatifs

Les alertes informatives, en revanche, sont un problème nettement plus réaliste. Il faut se rappeler que les vrais positifs pertinents, c'est-à-dire les alertes liées à la politique ou au fonctionnement du système et nécessitant une action immédiate de l'opérateur, ont déjà été éliminés de notre champ d'étude. Dans nos données, nous n'avons pas rencontré de grands nombres de faux positifs informatifs. En revanche, le nombre de positifs non pertinents est élevé. Par exemple, les alertes sur différents messages SNMP et sur la plupart de l'activité ICMP des trois ensembles font partie de cette catégorie, tout comme les alertes de chat de l'`ensemble-2`. Les positifs non pertinents informatifs font souvent partie de cette deuxième sous-catégorie de positifs non pertinents de la figure 3.2, où la signification de l'alerte dépend de son contexte. Par contexte, nous entendons le nombre d'alertes similaires dans une période de temps proche de l'alerte actuelle.

Pertinence de la signature

Nous avons parlé de la pertinence des alertes individuelles, mais la pertinence de la signature est tout aussi importante et ces deux aspects peuvent être différents. La pertinence de la signature dépend d'une organisation, de l'environnement, des politiques et de la mission de l'opérateur. Nous n'avons besoin d'un traitement automatisé que si l'information véhiculée potentiellement par la signature est, d'une quelconque manière, pertinente en rapport avec les politiques de sécurité et d'utilisation de l'organisation. Si l'information n'a aucune pertinence, la signature ne devrait absolument pas être active.

Une signature peut être pertinente, même si les alertes individuelles ne le sont pas - l'information pertinente peut être, par exemple, l'intensité des alertes (c'est-à-dire l'intensité des événements surveillés), les alertes par unité de temps, par rapport aux observations

[5] `http://www.kontiki.com/products/deliverymanager/`, consulté le 18.07.2006.

antérieures. C'est exactement le cas des signatures informatives. Dans la plupart des cas, l'information selon laquelle le noeud A a transmis au noeud B un message de requête SNMP n'a aucune pertinence pour l'opérateur. En revanche, le fait que le taux de ces messages soit passé de 50 messages par heure à 150 messages par heure, alors que le taux ne varie normalement que peu autour des 50 messages par heure, est un fait pertinent. La section suivante examine cette information contextuelle du bruit des alertes.

3.2 Caractéristiques du bruit des alertes

Dans la section précédente, nous avons passé en revue quelques origines possibles et quelques types d'alertes qui composent le sous-ensemble de l'inondation des alertes que nous avons baptisé bruit des alertes. Nous avons déjà décrit le concept de contexte des alertes comme l'occurrence d'alertes similaires sur une période proche de l'alerte actuelle. Plus important encore, nous avons évoqué l'idée que la signification de l'alerte peut dépendre de son contexte par rapport à son historique. Dans cette section, nous analyserons plus en profondeur les caractéristiques du bruit des alertes que nous avons trouvées dans les données des alertes.

3.2.1 Analyse du flux d'alertes

De nombreux positifs non pertinents n'ont pas les informations pour décider de leur signification. Le contexte peut être pris en compte pour analyser les alertes comme un agrégat. Pour analyser le contexte, nous devons mesurer l'*intensité des alertes*, le nombre d'alertes similaires dans une unité de temps. Nous appelons la séquence des observations de l'intensité des alertes, un *flux d'alertes*. Les alertes similaires comptabilisées pour obtenir l'intensité sont définies par les *critères d'agrégation*. Nous nous intéressons au *comportement du flux* actuel par rapport à leur comportement passé. En d'autres termes :

Definition Les *critères d'agrégation* sont n'importe quel(s) attribut(s) d'alertes selon lequel (lesquels) un groupe d'alertes peut se former.

Definition Un *flux d'alertes* est la séquence d'alertes qui répond aux critères d'agrégation du flux.

Definition L'*intensité des alertes* est le nombre d'alertes dans le flux d'alertes au cours d'un *intervalle d'échantillonnage*.

Definition Supposons que yt soit l'intensité des alertes observée à l'instant discret t. Maintenant, l'*intensité du flux* est la série des observations de l'intensité des alertes y_t pour $t = 0, 1, 2, \ldots$.

Definition Le *comportement du flux* signifie l'évolution de l'intensité du flux en fonction du temps.

Analyser les flux d'alertes au lieu des alertes individuelles présente deux avantages, à savoir les possibilités de 1. voir la manifestation de l'utilisation et/ou du comportement normal du système et 2. détecter les anomalies invisibles au niveau des alertes. Ces deux avantages sont essentiels à notre travail. En une phrase, notre but est d'éliminer par filtrage les alertes liées au comportement normal du système et de signaler les anomalies à examiner plus en profondeur.

Cinq flux de l'ensemble-1

Nous décrirons ensuite cinq flux d'alertes provenant de l'ensemble-1[6], obtenus en agrégeant les alertes par signature génératrice. L'idée est de fournir des exemples concrets de positifs non pertinents informatifs, de flux d'alertes et des avantages de l'analyse du flux en comparaison avec l'analyse d'alertes individuelles. Comme mentionné ci-avant, ces flux d'alertes serviront également d'ensemble de référence pour différentes méthodes de traitement présentées dans les derniers chapitres.

Nous décrirons brièvement la signature génératrice pour chaque flux, nous énumérerons les phénomènes intéressants identifiés et nous expliquerons ces phénomènes dans la mesure du possible. La documentation sur la signature se trouve sur le site Web de Snort[7] et dans le package Snort. La figure 3.3 montre l'intensité des alertes en fonction du temps sur la période de mesure. Les lignes pointillées montrent la division en données d'estimation et données de validation. La division sera expliquée et utilisée au chapitre 5, nous pouvons l'ignorer pour le moment.

SNMP request udp réagit aux requêtes du protocole SNMP (Simple Network Management Protocol) qui sont identifiées simplement par le port de destination. Etant donné le grand nombre d'alertes, elles sont probablement émises sur le trafic de gestion du réseau normal ou sur des messages provenant d'équipements réseau mal configurés. Dans les deux cas, la cause primaire peut être hors de portée du contrôle de l'opérateur.

Le flux d'alertes est extrêmement régulier, comme on peut le voir à la figure 3.3(a), avec quelques pics et quelques creux, ainsi que quelques anomalies plus petites. Dans ce cas, le nombre total de paires (source, destination) n'était que de neuf et seulement trois étaient responsables de la grande majorité des alertes. En d'autres termes, les causes primaires ont été identifiées et les alertes générées par ces noeuds auraient pu être éliminées par filtrage, même en échappant au contrôle de l'opérateur. Dans ce cas, toutefois, l'opérateur aurait également très probablement manqué le changement abrupt d'intensité des alertes, identifié par p_1.

Nous avons identifié cinq phénomènes intéressants dans le flux. Le premier, p_1, est un pic énorme, probablement lié à un changement dans la configuration de l'interaction, puisqu'il a également entraîné une augmentation de la composante constante du flux. Ensuite, p_2 et p_4 sont des pics plus petits, tandis que p_3 et p_5 sont des chutes, probablement dues à des problèmes de connectivité, dans un flux d'alertes par ailleurs extrêmement constant.

Une activité de faible intensité, quasiment invisible sur la photo, a également été constatée. Ce type d'activité de bas niveau se perdrait facilement dans la masse des alertes si les alertes étaient traitées manuellement. Dans le cas d'un évitement complet de la signature, cette activité aurait tout simplement été impossible à détecter.

ICMP PING WhatsupGold Windows est déclenchée par les messages d'écho

[6] Il est à noter que, même si les signatures définissent la direction du paquet en termes de réseau externe, réseau interne, serveurs Web, etc., nous ne savons pas comment elles sont configurées. Nous ne disposions que d'une base de données d'alertes avec les fichiers de configuration réelle des sondes. Comme décrit à la section 2.1 sur le thème des problèmes de configuration, ce type de configuration des sondes n'est pas toujours facile, ni même souhaitable. Certaines indications suggèrent que c'était également le cas avec ces sondes. La signature `SNMP request udp` définit la direction du paquet des réseaux externes vers les réseaux internes, et trois composantes majeures du flux provenaient de plages d'adresses privées pour se diriger vers d'autres plages d'adresses privées. Par conséquent, la sonde n'a fait aucune différence entre les adresses internes et externes pour ces adresses, ou certaines parties de la même organisation ont été considérées comme entités externes du point de vue sécurité.

[7] `http:/www.snort.org`, consulté le 18.07.2006.

ICMP avec un message dans le contenu du paquet réclamant leur création par un outil de mesure des performances[8].

Dans ce flux, visible à la figure 3.3(b), nous observons une composante constante au-dessus de laquelle nous avons une composante périodique créant des pics sur cinq jours de travail, mais pas d'alertes supplémentaires pendant les week-ends. Les alertes générées par ces deux composantes sont régulières et considérées comme légitimes, provenant en fait de l'utilisation de l'outil de mesure des performances. Le nombre total de paires (source, destination) générant ces alertes était de 30, chiffre conforme à l'utilisation normale de l'application WhatsUp : surveillance de serveurs comme les serveurs de messagerie électronique, les serveurs Web, etc. L'outil se compose d'un serveur qui interroge les noeuds de réseau pour vérifier leur disponibilité. Par conséquent, le comportement normal du flux est indépendant du volume du trafic général sur le réseau, contrairement au cas de `ICMP PING speedera`. Cette origine automatisée explique également la grande régularité des variations quotidiennes et hebdomadaires et l'existence de la composante constante.

Les phénomènes intéressants sont les changements dans le niveau constant. Les phénomènes p_1, p_4, p_6 et p_7 sont des chutes, tandis que p_2, p_3, p_5 et p_8 sont des augmentations de l'intensité de la composante constante. Comme avec le flux précédent, il s'agit de changements dans le taux de génération des alertes, et le filtrage basé uniquement sur les attributs des alertes empêcherait l'opérateur de voir ces changements. Par exemple, une attaque DoS[9] contre une application WhatsUp pourrait provoquer des changements anormaux dans l'intensité des alertes - une attaque inconnue pourrait être détectée par les échos qu'elle crée dans le fonctionnement normal du système.

ICMP Destination Unreachable Communication Administratively Prohibited réagit à des messages ICMP normalement générés lorsqu'un noeud de réseau, par exemple un routeur, élimine un paquet pour des raisons administratives.

Pour ce flux, illustré à la figure 3.3(c), il existe plusieurs causes possibles, comme une rétrodiffusion provenant des attaques DoS [MVS01, MCB+06], des interruptions de réseau ou des problèmes de routage. Etant donné le comportement irrégulier et les 2176 paires (source, destination) distinctes, il s'agit probablement d'une combinaison de toutes ces causes et d'autres. En d'autres termes, nous ne pourrions pas définir les causes primaires. Par conséquent, le filtrage basé sur ces alertes individuelles serait difficile. Ces messages sont comme le rayonnement de fond ; presque tout réseau est susceptible de les avoir dans le trafic entrant lorsque les hôtes à l'intérieur essaient de se connecter à des adresses invalides/non accessibles. Le flux ne contient pas autant de structure que les quatre autres. Le lissage de l'intensité à l'aide d'une moyenne glissante a permis de dégager une certaine structure, indiquant plutôt un rythme hebdomadaire faible. Une corrélation entre l'utilisation du réseau et les messages entrants ICMP Destination Unreachable semble assez logique et expliquerait le rythme hebdomadaire.

LOCAL-POLICY External connexion from http server est une signature personnalisée dont le nom est suffisamment explicite. Le fondement de cette signature est que, comme tous les serveurs principaux sont internes, un serveur Web ne devrait pas se trouver sur le côté SYN d'un protocole de transfert TCP avec des hôtes *externes*, à moins d'être par exemple infecté par un ver. Quoi qu'il en soit, c'est ce qu'il s'est passé et nous n'avons pas été en mesure d'expliquer ce comportement sur la base des traces d'alertes.

L'intensité des alertes est décrite à la figure 3.3(d). Même si le flux se compose d'im-

[8] `http:/www.ipswitch.com/Products/WhatsUp/Professional`, consulté le 18.07.2006.

[9] Josh Zlatin-Amishav, *DoS vulnerability in Ipswitch WhatsUp Pro* on `Bugtraq`, 22.02.2006, `http:///www.securityfocus.com/archive/1/425780/30/0/threaded`, consulté le 18.07.2006.

pulsions, une certaine périodicité peut être constatée. Il existe des pics de milliers d'alertes suivant un rythme hebdomadaire et des pics plus petits qui suivent des cycles quotidiens, voire même plus petits. Il existe en outre des anomalies : p_1 manque et p_3 correspond à un changement dans l'activité bas niveau. Le phénomène p_2 est un pic extrêmement élevé par rapport aux observations antérieures et p_4 et p_5 sont de gros pics qui brisent le rythme normal. Seulement trois paires (source, destination) ont généré ces alertes.

ICMP PING speedera se déclenche sur des messages d'écho ICMP qui véhiculent un message spécifique. Elle prétend que le paquet provient d'un serveur appartenant à la société de distribution de contenu Speedera[10]. Speedera utilise ces messages pour déterminer la mémoire cache la plus proche de la machine demandant le contenu hébergé par eux. Ce n'est qu'un exemple de la difficulté d'analyser les machines dont le contrôle nous échappe.

Le nombre de ces messages est proportionnel aux accès aux sites hébergés par la société. Par conséquent, le flux, illustré à la figure 3.3(e), possède une composante périodique forte, avec des pics pendant les heures de travail et des creux pendant la nuit et les week-ends. Comparé à `ICMP PING WhatsupGold Windows`, l'autre flux généré par les échos ICMP, le comportement comporte plus de variations haute fréquence autour des composantes périodiques. Les alertes ont été générées par 159 paires (source, destination) contrairement au 30 de WhatsUp, ce qui peut expliquer en partie la différence. Nous considérons la composante périodique comme un comportement normal du système. Deux anomalies ont été identifiées, p_1 est un pic durant les heures de grande activité et p_2 est un pic durant les périodes de faible activité. La cause probable est une augmentation du trafic légitime lié à quelque chose comme la diffusion du calendrier Pirelli sur un serveur hébergé par Speedera.

Phénomènes intéressants visibles uniquement dans les flux

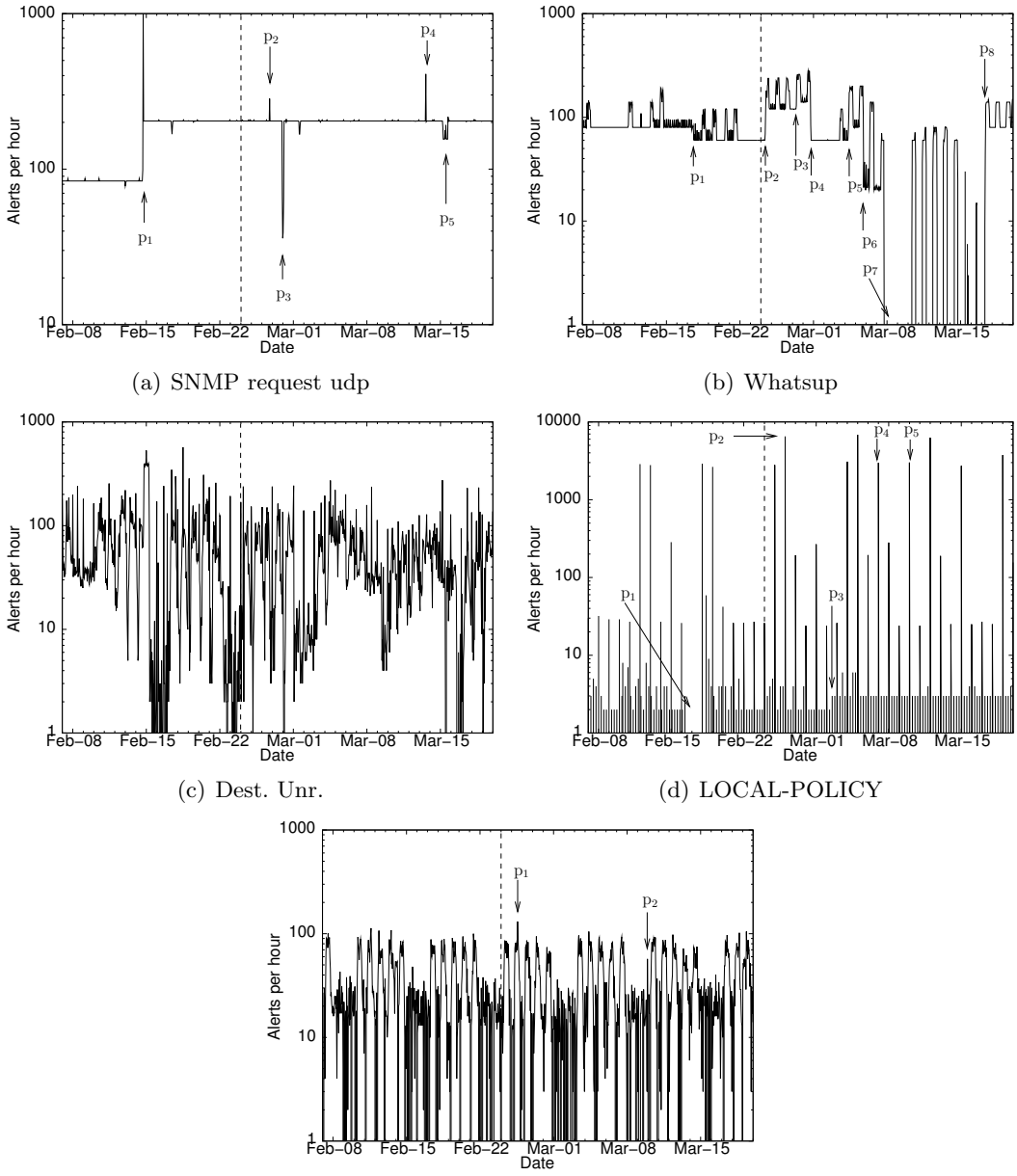
Les phénomènes intéressants dans ces flux sont souvent intermittents et sont soit 1) des changements dans l'intensité, généré par le comportement normal du système ou par ce que l'on appelle les causes primaires et/ou 2) l'apparition de sources anormales, c'est-à-dire autre chose que les causes primaires. Nous avons besoin d'un filtrage du niveau du flux pour ces deux raisons. Etant donné que toutes les alertes du flux peuvent avoir les mêmes valeurs d'attribut en dehors des estampilles temporelles, le filtrage alerte par alerte perd généralement les phénomènes antérieurs et, sans aucun filtrage, ces derniers phénomènes risquent de se perdre dans le bruit.

Nous aimerions également souligner que quatre des cinq signatures déclenchent des alertes dans un flot quasi-continu. Sans automatisation supplémentaire du traitement des alertes, l'opérateur devrait analyser ces alertes en permanence. Ce serait une tâche laborieuse, exigeant à tout le moins du temps et la mise à l'épreuve des ressources mentales des analystes.

Critères d'agrégation

Dans la suite de cette thèse, sauf spécifications contraires, nous utiliserons la signature et le sonde génératrices comme critères d'agrégation. Nous parlerons par conséquent de flux d'alertes avec les noms des signatures qui génèrent les flux. Par flux d'alertes `SNMP request udp`, nous entendons le flux d'alertes obtenu en agrégeant toutes les alertes générées par

[10] `http:/www.speedera.com`, consulté le 18.07.2006, Speedera fait maintenant partie de Akamai.



(a) SNMP request udp

(b) Whatsup

(c) Dest. Unr.

(d) LOCAL-POLICY

(e) Speedera

FIG. 3.3 – Intensité horaire des alertes pour cinq des flux les plus prolifiques agrégés par signature de l'`ensemble-1`. L'axe horizontal est celui du temps, tandis que l'axe vertical montre le nombre d'alertes par heure dans une échelle logarithmique. Les flèches désignent les phénomènes qui nous intéressent

la signature `SNMP request udp` et des alertes séparées de différentes sondes. En pratique, dans la plupart des cas, il n'y a eu qu'un seul sonde par ensemble d'alertes sur laquelle les signatures prolifiques ont généré des alertes.

Nous avons également essayé des critères à granularité plus ou moins fine, comme des classes entières de signatures et des alertes générées par une seule signature, source et destination. L'un des inconvénients des méthodes de détection des anomalies est le manque de diagnostic. De manière typique, une anomalie est signalée, éventuellement avec un type de mesure du degré d'anomalie de l'événement. Considérons les trois rapports d'anomalies suivants, classés par ordre décroissant d'agrégation, mais par ordre croissant de spécificité :

1. L'intensité des alertes non critiques est nettement inférieure à la normale.
2. L'intensité des alertes liées à ICMP est nettement inférieure à la normale.
3. L'intensité `ICMP L3retriever Ping` est nettement inférieure à la normale.

La première pourrait être émise lors de l'agrégation de tous les bruits d'alertes ensemble, la deuxième lors de l'agrégation selon une classe d'alerte et la troisième lors de l'agrégation selon une signature. Pour l'opérateur, la troisième est la plus facile à analyser. Typiquement, chaque signature a ses spécificités et dès que le flux d'alertes contient plus d'une signature, analyser l'anomalie signalée prendrait plus de temps. On pourrait poursuivre la liste comme suit :

1. L'intensité ICMP L3retriever Ping vers 1.2.3.4 est nettement inférieure à la normale.
2. L'intensité ICMP L3retriever Ping depuis IP 1.2.3.4 est nettement inférieure à la normale.
3. L'intensité ICMP L3retriever Ping depuis 1.2.3.5 vers 1.2.3.4 est nettement inférieure à la normale.

Il est à noter que les cas 1 et 2 se situent sur le même niveau d'agrégation. De nouveau, plus on descend dans la liste, plus le message d'anomalie est spécifique et offre un point de départ plus facile pour l'opérateur, et probablement quelque chose de plus utilisable pour d'éventuels autres moteurs de corrélation.

Toutefois, deux défauts se dégagent lorsque l'on descend jusqu'au niveau source, destination ou source-destination.

Perte de généralité En descendant de toutes les alertes jusqu'au niveau de la signature, il est facile d'appliquer les critères d'agrégation à toutes les signatures prolifiques, si on le souhaite. Si l'on passe à des flux plus spécifiques, il faudrait vérifier au cas pas cas s'il est ou non raisonnable d'utiliser une agrégation à granularité si fine. Si le nombre de sources et de destinations est élevé, il se pourrait que les flux détaillés ne contiennent pas suffisamment d'informations pour une analyse significative. Pour certaines signatures, il pourrait être utile d'énumérer quelques sources, destinations ou paires (source, destination) tout en ne faisant qu'un du reste, mais cela n'a pas été le cas général dans nos ensembles de données.

Explosion éventuelle du nombre de flux De nombreuses signatures ont généré des alertes pour un très grand nombre de sources et de destinations. Si l'on utilise des critères d'agrégation à granularité plus fine que les signatures, le nombre de flux à traiter pourrait exploser. La section 3.2.4 s'étendra un peu plus sur le sujet.

L'une des possibilités serait d'exclure la signature des critères d'agrégation et d'examiner, par exemple, des paires (source, destination), mais dans ce cas nous serions confrontés à des alertes d'anomalies encore moins significatives. Nous reconnaissons que les méthodes

de regroupement d'alertes telles que celles proposées par Julisch [Jul03a] pourrait offrir un moyen de trouver des critères d'agrégation à granularité encore plus fine.

Dans le travail précédent, nous avons également procédé à des agrégations selon la signature et des critères de temps, comme le jour de la semaine et l'heure du jour [Vii03]. Cette approche crée deux sous-flux ou plus selon la séparation temporelle.

Dans la catégorie jour de la semaine, nous avons examiné la séparation des mesures entre jour de la semaine et week-end, tandis que dans la catégorie heure du jour, nous avons analysé séparément les mesures d'intensité 1. des jours et des nuits et 2. de chaque heure. Premièrement, il est difficile de définir les limites des différentes plages horaires, par exemple le week-end commence-t-il le vendredi soir à 17.00 ou à 20.00 heures, quelle taille de plages doit-on utiliser pour l'analyse de l'heure du jour, etc. En outre, la taille et le placement des plages horaires peuvent varier d'un flux à l'autre. Deuxièmement, des changements abrupts de niveau d'intensité, comme celui visible à la figure 3.3(a) aux alentours du 15 février, apparaissent dans chaque sous-flux et pourraient générer une alerte d'anomalie pour chaque sous-flux. En conclusion, le traitement du flux continu est plus facile à comprendre pour l'opérateur et, dans la plupart des cas analysés, a fourni au moins d'aussi bons résultats que le traitement de flux séparés selon le jour ou l'heure.

3.2.2 Régularités et anomalies

Dans le premier ensemble de données, l'`ensemble-1`, nous avons trouvé quatre types de flux que nous appelons profils des alertes. Les profils sont définis selon leur régularité et notre capacité et assurance à expliquer le comportement normal et anormal.

KC (connu, constant) La génération d'alertes est constante, avec éventuellement de petites variations. Le flux comporte relativement peu d'anomalies que nous pouvons expliquer et attribuer à 1) un changement dans la configuration de l'interaction, 2) un problème.

KP (connu, périodique) Le flux d'alertes contient une périodicité clairement visible et/ou une composante constante avec une origine bénigne. Le flux comporte quelques anomalies, dont nous pouvons expliquer la majorité.

UP (inconnu, périodique) Le flux d'alertes est moins stable que dans la classe KP, il comporte plus d'anomalies visibles et nous ne savons pas nécessairement comment les expliquer.

UR (inconnu, aléatoire) Le flux est plus ou moins aléatoire. Seule une petite structure est visible à l'oeil nu, ou pas de structure du tout. Nous n'avons que des explications limitées à proposer pour l'origine de ces alertes.

La régularité et l'explicabilité s'étendent sur deux axes et les quatre profils sont placés dans ce plan à la figure 3.4. Aucun de ces quatre profils ne tombe dans le deuxième quadrant. Ces alertes sont bien caractérisées : elles se présentent seulement occasionnellement, elles sont souvent associées à une vulnérabilité et sont soit des manifestations réelles d'attaques ou des faux positifs. Les alertes du deuxième quadrant sont traitées typiquement par d'autres techniques de corrélation d'alertes.

Dans le premier quadrant, nous avons deux classes différentes, KC et KP. La raison pour laquelle nous avons deux classes séparées est le type différent de régularité, constante ou périodique, des classes KC et KP. Les paquets qui créent des flux d'alertes constants ont vraisemblablement peu de causes liées à la machine, étant donné leur comportement prédéterminé, de type horloge. La cause du comportement périodique est très

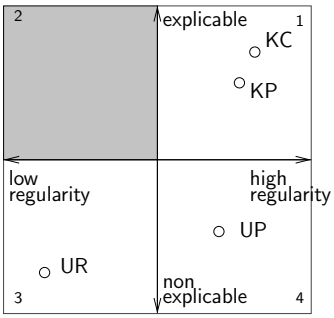


FIG. 3.4 – Classes d'alertes par rapport aux deux axes, régularité et explicabilité

probablement l'activité humaine ou, éventuellement, un nombre important de machines. Par exemple, si le trafic sur le réseau déclenchant les alertes est créé par des actions d'un grand nombre de personnes, des rythmes naturels avec des périodes d'un jour ou d'une semaine peuvent exister dans les flux d'alertes.

En établissant le profil des flux de l'`ensemble-1` (Figure 3.3, Section 3.2.1), nous plaçons `SNMP request udp` dans KC, en raison de l'intensité extrêmement constante des alertes et du petit nombre d'hôtes qui génère les alertes. `ICMP PING WhatsupGold Windows` est classé dans UP, étant donné que l'intensité augmente les jours ouvrables et que nous ne pouvons pas expliquer les raisons des anomalies. `ICMP PING Communication Administratively Prohibited` est attribué à UR, étant donné que nous ne pouvons pas associer la variabilité importante ni les anomalies à certains phénomènes ou à un petit ensemble de causes primaires. Nous classons `LOCAL-POLICY external connexion from http server` dans UP, étant donné que nous ne pouvons pas expliquer les pics d'alertes, mais il y a un comportement périodique. Enfin, `ICMP PING speedera` fait partie de KP, en raison de sa nature hautement périodique et nous sommes assez confiants dans l'explication à la fois du comportement normal et des anomalies.

La question que nous avons posée et qui a été posée après avoir analysé l'`ensemble-1`, était de savoir si ces profils d'alertes sont plus généraux. L'analyse de flux supplémentaires de l'`ensemble-2` et de l'`ensemble-3` nous a permis de trouver des comportements similaires. Nous présentons ici quelques exemples de ces flux. Nous avons observé les caractéristiques communes suivantes :

- Fortes dépendances par rapport au jour de la semaine et à l'heure du jour
- Flux constants
- Stabilité des profils
- Anomalies intermittentes, soit des changements brusques de niveaux, soit des augmentations et diminutions similaires à des impulsions

Nous décrirons les observations à partir d'abord de l'`ensemble-2`, puis de l'`ensemble-3`.

Dans le tableau 3.4, nous avons seulement deux profils d'alertes dans la classe KR (connu, aléatoire), à savoir `ICMP CyberKit 2.2 Windows` et `LOCAL-WEB-IIS Nimda.A attempt`. Nous avons mentionné auparavant que les alertes faisant partie de cette catégorie ont souvent été associées à une vulnérabilité et pouvaient être identifiées par d'autres moyens. Le premier flux se compose en fait de faux positifs, étant donné que l'alerte `ICMP CyberKit`

TAB. 3.4 – Flux ayant généré plus de 10K d'alertes dans l'`ensemble-2` et profils attribués. Le statut inconnu pourrait être modifié suite à des analyses ultérieures. L'âge de l'ensemble d'alertes et le manque d'accès au système rendent une analyse approfondie difficile

nom du flux	**profil**
ICMP PING CyberKit 2.2 Windows	KR
CHAT IRC message	UP
ICMP Destination Unreachable (Comm Adm Proh)	UR
ICMP PING speedera	KP
(frag2) TTL Limit Exceeded (reassemble) detection	UR
ICMP Large ICMP Packet	UP
(stream4) TTL LIMIT Exceeded	UR
ICMP L3retriever Ping	KC
CHAT MSN message	KP
WEB-IIS view source via translate header	UP
CHAT IRC nick change	KC − > KP
BAD-TRAFFIC loopback traffic	UR
WEB-PHP content-disposition	UP
WEB-PHP content-disposition	UP
(stream4) STEALTH ACTIVITY (SYN FIN scan) detect	UR
DDOS Stacheldraht agent->handler (skillz)	KC
LOCAL-WEB-IIS Nimda.A attempt	KR
ICMP Dest Unr (Comm w/ Dest Host Adm Proh)	UP

`2.2 Windows` est également déclenchée par des scannages du ver[11] Welchi/Nachi[12]. Le flux comporte 611 420 adresses sources qui, selon les vérifications aléatoires, sont seulement des adresses non privées. Pour ces raisons et étant donné l'intervalle de temps de l'`ensemble-2`, nous sommes convaincus que les alertes sont déclenchées sur l'activité Nachi, c'est-à-dire des faux positifs. Les alertes liées à Nimda peuvent, du moins en théorie, être vérifiées et classées par ordre de priorité à l'aide des informations de configuration du système surveillé. Ces types d'alertes ne sont pas la cible principale des techniques de traitement des alertes que nous développons.

Le troisième flux d'alertes à strictement parler non informatif est `DDOS Stacheldraht->handler (skillz)`. Un hôte a communiqué avec les deux adresses de destination selon un schéma horaire précis pendant quelques jours. Il s'agissait clairement d'une machine compromise.

Outre ces trois flux, nous pouvons dire que les 15 flux restants sont informatifs. Un grand nombre de flux présente un comportement régulier. Neuf flux comportent des composantes périodiques évidentes. Deux peuvent être caractérisées de constantes, même si elles présentent des fluctuations haute fréquence autour des niveaux constants et ne sont pas aussi constantes que `SNMP request udp` dans l'`ensemble-1` (Figure 3.3(a)).

Les profils de flux étaient stationnaires, dans le sens où le comportement est resté constant tout au long de la période d'observation, hors mis `CHAT IRC nick change`, qui avait au départ deux composantes, un comportement de type horloge et une composante périodique d'un nettement moins gros volume avec un rythme hebdomadaire. La composante constante de type horloge a été exclue. Nous pensons que la composante constante

[11] G. Larrat, *Re : [Snort-users] ICMP CyberKit 2.2 Windows* on snort-user, 19.08.2003 `http://archives.neohapsis.com/archives/snort/2003-08/0604.htlm`, consulté le 18.07.2006.

[12] `http://www.f-secure.com/v-descs/welchi.shtml`, consulté le 18.07.2006.

aurait pu être causée par une sorte de trafic de contrôle du réseau des zombies, alors que le rythme hebdomadaire proviendrait de l'utilisation normale de IRC.

Dans la plupart des cas, les alertes ont été générées pendant les 100 jours. Le flux `CHAT IRC nick change` n'est apparu qu'au troisième tiers de la période d'observation. Des alertes `BAD-TRAFFIC loopback traffic` ont été observées du jour 59 au jour 64, tout comme les alertes `DDOS Stacheldraht->handler (skillz)`[13].

Des anomalies sont souvent des pics ou creux très courts dans l'intensité du flux. C'est le cas typique des flux périodiques. Dans les flux constants, nous pouvons également observer des changements de niveaux. Seuls les flux `ICMP Large ICMP Packet` et `ICMP Destination Unreachable Communication Administratively Prohibited` ont présenté des anomalies basse fréquence visibles, même au niveau hebdomadaire, probablement en rapport avec la dynamique du réseau. Il pourrait également s'agir de rétrodiffusion d'attaques DoS de longue durée utilisant le protocole ICMP et une adresse IP source usurpée à la plage d'adressage du réseau surveillé [MVS01].

Les profils des flux de l'`ensemble-3` sont présentés au tableau 3.5.

Les flux liés à SNMP contiennent des composantes périodiques haute fréquence, mais la ligne de base de l'activité est plutôt plate. Les alertes sont provoquées par l'activité SNMP normale dans le réseau surveillé.

Un comportement périodique important est visible dans les flux `ICMP L3retriever Ping`, `(http_inspect) BARE BYTE UNICODE ENCODING` et dans les trois flux NETBIOS. Le flux `ICMP L3retriever Ping` présente un profil totalement différent de celui du même flux dans l'`ensemble-2`. La figure 3.5 illustre les différences entre les deux. Etant donné la précision des utilitaires Ping de l'`ensemble-2`, l'origine probable est réellement un scanner Retriever des réseaux L3, désormais possédé par Symantec. Des hôtes Windows communiquant avec le contrôleur de domaine ont été signalés à Arachnids[14] et dans la liste de diffusion `snort-users`[15] comme l'origine de faux positifs. C'est ce que nous voyons dans l'`ensemble-3`, étant donné que les adresses source et destination sont privées et étant donné la forte corrélation entre l'intensité des alertes et l'heure du jour, ainsi que le jour de la semaine.

Le flux `(http_inspect) BARE BYTE UNICODE ENCODING` est généré par le préprocesseur de Snort. Selon la documentation de la règle fournie avec Snort, il se déclenche sur des encodages Unicode spécifiques qui sont compris par les serveurs IIS, mais qui ne doivent être utilisés par aucun client. Il a été signalé que des serveurs SMS Windows causaient des faux positifs[16]. Cela semblerait exact, étant donné que les hôtes de destination les plus significatifs sont les serveurs proxy www et un serveur SMS.

Le flux `NETBIOS SMB-DS DCERPC NTLMSSP asn1 overflow` attempt se compose de faux positifs. Les raisons de le croire sont le grand nombre d'alertes, la nature fortement périodique du flux et la forte corrélation croisée statistique avec d'autres alertes NETBIOS.

---

[13] Ces alertes sont probablement liés, étant donné que bad traffic n'a été transmis qu'à deux adresses, l'une recevant plus de 23K de paquets et l'autre, seulement 21. En outre le rapport du taux par minute entre `Stacheldraht->handler (skillz)` et `BAD-TRAFFIC loopback traffic` est de 1 :50. `Bad traffic` provenait de 105 adresses qui auraient pu être usurpées par l'agent Stacheldraht via le gestionnaire.

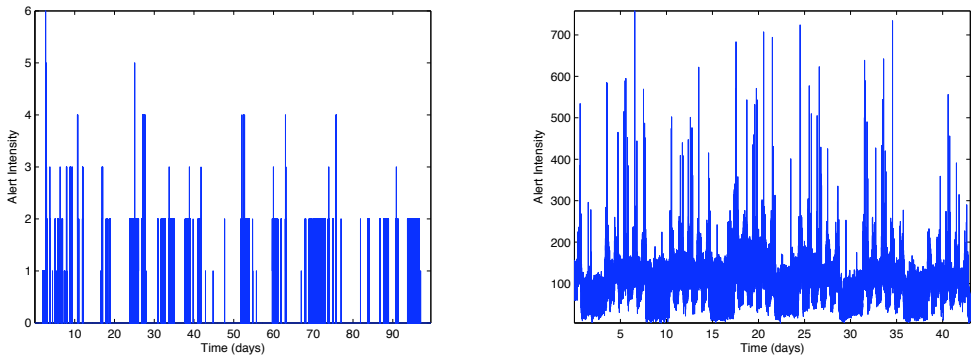
[14] `http://www.digitaltrust.it/arachnids/IDS311/event.html` , consulté le 18.07.2006.

[15] J. Jordan, *Re : [Snort-users] ICMP L3retriever Ping*, 22.12.2003, `http://archives.neohapsis.com/archives/snort/2003-12/0430.html`, consulté le 18.07.2006.

[16] Snort Forums Archives, transmis par MikeDaGeek le 20.03.2005, `http://www.snort.org/archive-3-146.html`, consulté le 18.07.2006.

TAB. 3.5 – Flux ayant généré plus de 1M d'alertes dans l'`ensemble-3` et profils attribués. Le statut inconnu du flux lié à NETBIOS pourrait être modifié suite à une analyse plus détaillée. Pour le moment, nous ne pouvons que suspecter les origines

nom de la signature	**alertes**
SNMP request udp	KC
SNMP public access udp	KC
ICMP L3retriever Ping	KP
(http_inspect) BARE BYTE UNICODE ENCODING	UP
NETBIOS SMB-DS DCERPC NTLMSSP asn1 oflow att	UP
NETBIOS SMB-DS IPC$ share unicode access	KP
NETBIOS SMB IPC$ share unicode access	KP



(a) `L3retriever` from `set-2` (b) `L3retriever` from `set-3`

FIG. 3.5 – Différence entre les profils de flux ICMP L3retriever Ping de l'`ensemble-2` et de l'`ensemble-3`. Dans le premier cas, les alertes sont probablement générées par l'outil de scan Retriever 1.5 du réseau L3. Dans le deuxième cas, le flux est probablement des faux positifs provoqués par les hôtes Windows qui communiquent avec le contrôleur de domaine

Les deux flux NETBIOS restants sont émis à l'accès normal aux parts de réseau et présentent un comportement périodique important, avec des rythmes hebdomadaires et quotidiens.

Résumons les observations concernant le comportement normal et anormal. Comme nous l'avons vu, les alertes informatives présentent fréquemment des régularités importantes. Ces régularités sont souvent périodiques, constantes ou une combinaison des deux. Les cycles de comportement périodiques sont naturels, des rythmes soit hebdomadaires soit quotidiens. Les composantes constantes sont pour la plupart créées par seulement quelques sources ou destinations, mais le comportement périodique est associé à un grand nombre d'hôtes participants. Lorsque l'on analyse les flux avec les régularités, nous pouvons souvent expliquer le comportement régulier par l'utilisation ou le fonctionnement normal du système. Pour la plupart des flux, les schémas de comportement restent les mêmes dans l'horizon temporel des ensembles de données.

En général, nous supposons que la régularité et la structure stationnaire présentes dans ces flux d'alertes ont des origines bénignes. Par définition, la structure stationnaire existait par le passé et est restée inchangée. Cela devrait également être normal selon l'hypothèse sous-jacente de la détection des anomalies. Si ce n'est pas le cas, cela signifie que des problèmes fondamentaux existent aussi bien au niveau fonctionnel qu'au niveau sécurité.

3.2.3 Intervalle d'échantillonnage

Nous avons précédemment étudié les différents types de régularités et d'anomalies observés dans les flux d'alertes. L'intervalle d'échantillonnage t_s est un problème lié aux profils des flux, mais aussi un choix important dans l'ensemble. Il affecte :

1. la visibilité et la clarté des phénomènes
2. la rapidité de détection
3. la quantité de données à traiter
4. l'atteinte en temps réel du nombre fixé d'observations

La visibilité signifie que les phénomènes dont l'échelle de temps est nettement plus petite ou plus grande que l'intervalle d'échantillonnage risquent de se perdre. Les changements rapides à l'intérieur d'un intervalle d'échantillonnage peuvent être lissés, tandis que les changements lents peuvent être noyés parmi des variations à plus court terme.

La détection des anomalies sera différée au moins jusqu'à la fin de l'intervalle d'échantillonnage et de la durée de traitement de l'observation. Si la méthode de traitement utilise des informations depuis les instants $n+1$ ou plus pour analyser l'observation depuis l'instant n, le retard sera encore plus long. Etant donné que nous traitons principalement les alertes de faible priorité, la rapidité est de moindre importance que dans les sondes de bas niveau. Les alertes politiques et informatives n'ont en aucun cas été soumises à une surveillance constante 24 h/24, 7j/7 sur les sites d'où proviennent nos ensembles de données.

La quantité de données à traiter augmente bien entendu au fur et à mesure que t_s diminue. Même si les méthodes de traitement n'ont pas nécessité de capacité de stockage supplémentaire, l'utilisation du processeur et éventuellement la charge du réseau augmentent avec un t_s plus petit. La charge du réseau augmenterait en cas d'architecture distribuée exécutant des méthodes de traitement sur une machine autre que celle faisant fonction de stockage d'alertes.

La taille de la fenêtre d'observation en temps qui affecte l'analyse diminue avec t_s, étant donné que l'historique observé par les méthodes de traitement explorées est mesuré

en nombre d'observations, c'est-à-dire en mesures d'intensité des alertes. Si les 24 dernières observations affectent l'analyse, avec des intervalles d'échantillonnage d'une heure et d'une minute, le temps réel est atteint respectivement en un jour et 24 minutes.

Chronologiquement parlant, l'`ensemble-1` a été le premier à être utilisé. La technique d'analyse basée sur l'analyse de tendances développée pour ces données est décrite au chapitre 4. A cette époque, nous avons exploré de longs intervalles d'échantillonnage variant entre 30 et 240 minutes. Etant donné que nous analysons des alertes de faible priorité, cette tranche de temps était et est toujours considérée comme suffisante. L'intervalle d'échantillonnage d'une heure était considéré comme le plus approprié, car il offrait un bon compromis entre le lissage de quelques petites fluctuations rapides sans intérêt et l'empêchement du lissage de phénomènes intéressants, comme décrit ci-dessus. De plus, même si la rapidité n'était pas la priorité absolue, nous ne souhaitions pas différer la détection plus que nécessaire. Par la suite, nous avons également utilisé une autre technique basée sur les modèles de séries temporelles non stationnaires, décrits au chapitre 4, pour analyser les mêmes données.

L'`ensemble-2` et l'`ensemble-3` ont été analysés par la suite avec la technique d'analyse basée sur les modèles de séries temporelles non stationnaires décrits au chapitre 6. A cette époque, nous souhaitions améliorer la rapidité en général et en particulier, étant donné que l'algorithme d'estimation utilisé introduisait un délai supplémentaire en termes d'observations.

Avec ces ensembles, nous avons commencé par le niveau d'une minute et augmenté l'intervalle d'échantillonnage pour en examiner l'effet. Un intervalle d'échantillonnage plus long fait mieux ressortir différentes régularités et lisse les fluctuations haute fréquence qui dominent le flux à un intervalle d'échantillonnage d'une minute.

Comportement normal visible à toutes les échelles

Les figures 3.6 et 3.7 présentent des exemples de problèmes de visibilité. Les deux figures sont des flux d'alertes provenant de l'`ensemble-3`, la figure 3.6 est le flux `SNMP request udp` et la figure 3.7 est le flux `NETBIOS SMB IPC$ share unicode access`. Dans les deux figures, le flux supérieur est obtenu avec un intervalle d'échantillonnage d'une minute et, pour le suivant, l'intervalle d'échantillonnage est de 5, 10, 15 et 20 minutes. A la figure 3.6, le rectangle rouge montre comment une augmentation des alertes est plus visible sur des échelles de temps plus larges. La figure 3.7 donne des exemples tant de l'augmentation que de la diminution de la visibilité avec un intervalle d'échantillonnage plus long. Le rectangle rouge de gauche montre comment une augmentation du nombre des alertes est plus évidente sur des échelles de temps plus larges. Puis, en guise d'exemple contraire, le rectangle rouge de droite met en évidence un changement rapide clairement visible à une échelle de temps plus petite, mais qui est lissé à des échelles de temps plus larges.

Dans certains cas et en plus de fluctuations toujours présentes, à des intervalles d'échantillonnage courts, le flux comporte un grand nombre de pics similaires à des impulsions. La figure 3.8 illustre le flux `ICMP L3retriever Ping` de l'`ensemble-3`, où nous pouvons constater :

- le nombre élevé de pics similaires à des impulsions présents dans tout le flux avec t_s = 1 min et qui sont lissés lorsque t_s augmente,
- une fois encore, les fluctuations haute fréquence toujours présentes avec t_s = 1 min,
- une fois encore, les rythmes quotidiens et hebdomadaires de renforcement lorsque t_s augmente.

Bien que le flux devienne plus lisse et les régularités plus évidentes au fur et à mesure

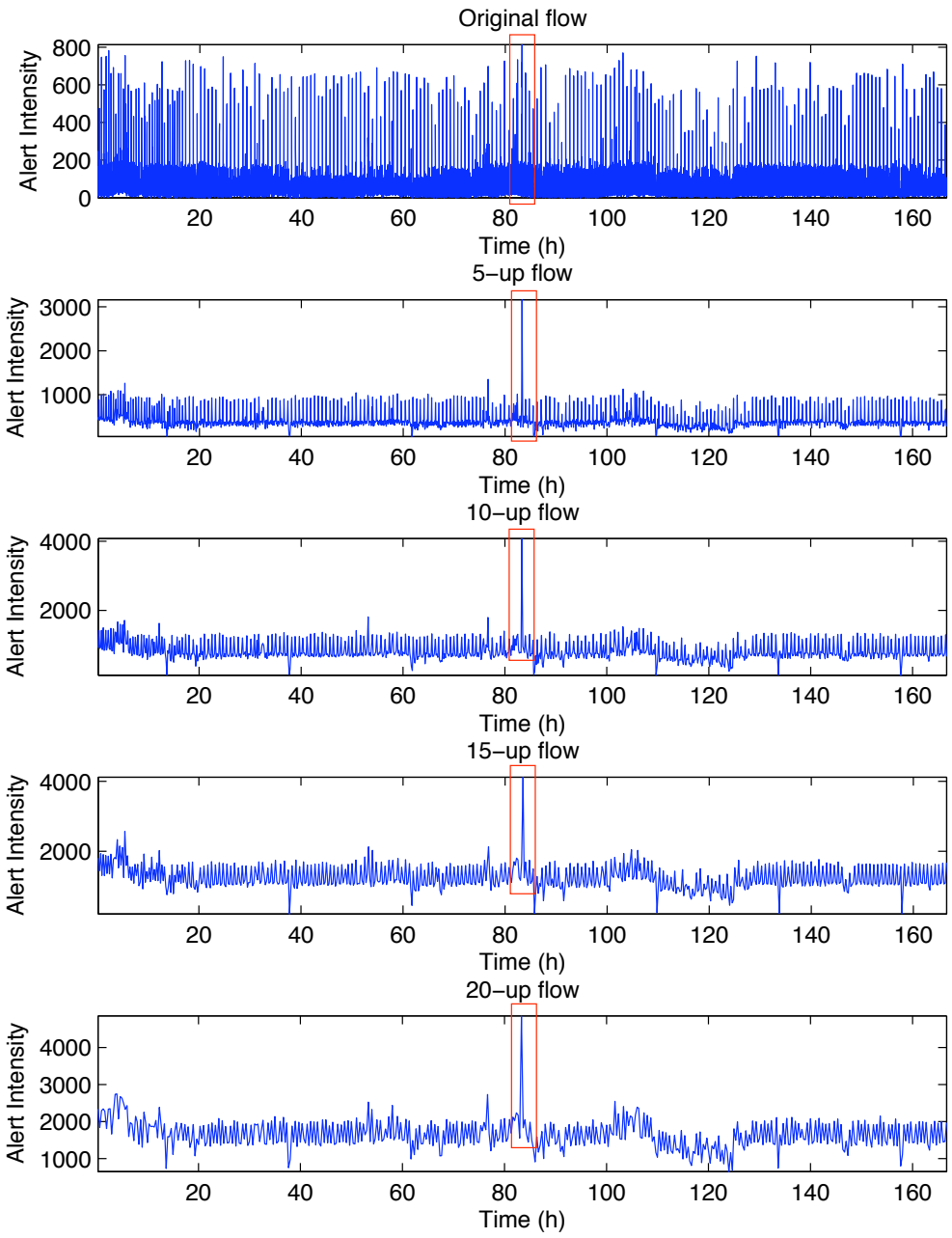


FIG. 3.6 – Effet de l'intervalle d'échantillonnage sur la visibilité des phénomènes dans le flux `SNMP request udp` de l'`ensemble-3`. L'intensité du flux original a été mesurée une fois par minute. Les flux suivants ont été obtenus en totalisant 5, 10, 15 et 20 échantillons. Le rectangle rouge montre comment une augmentation des messages SNMP est plus visible à des échelles de temps plus larges

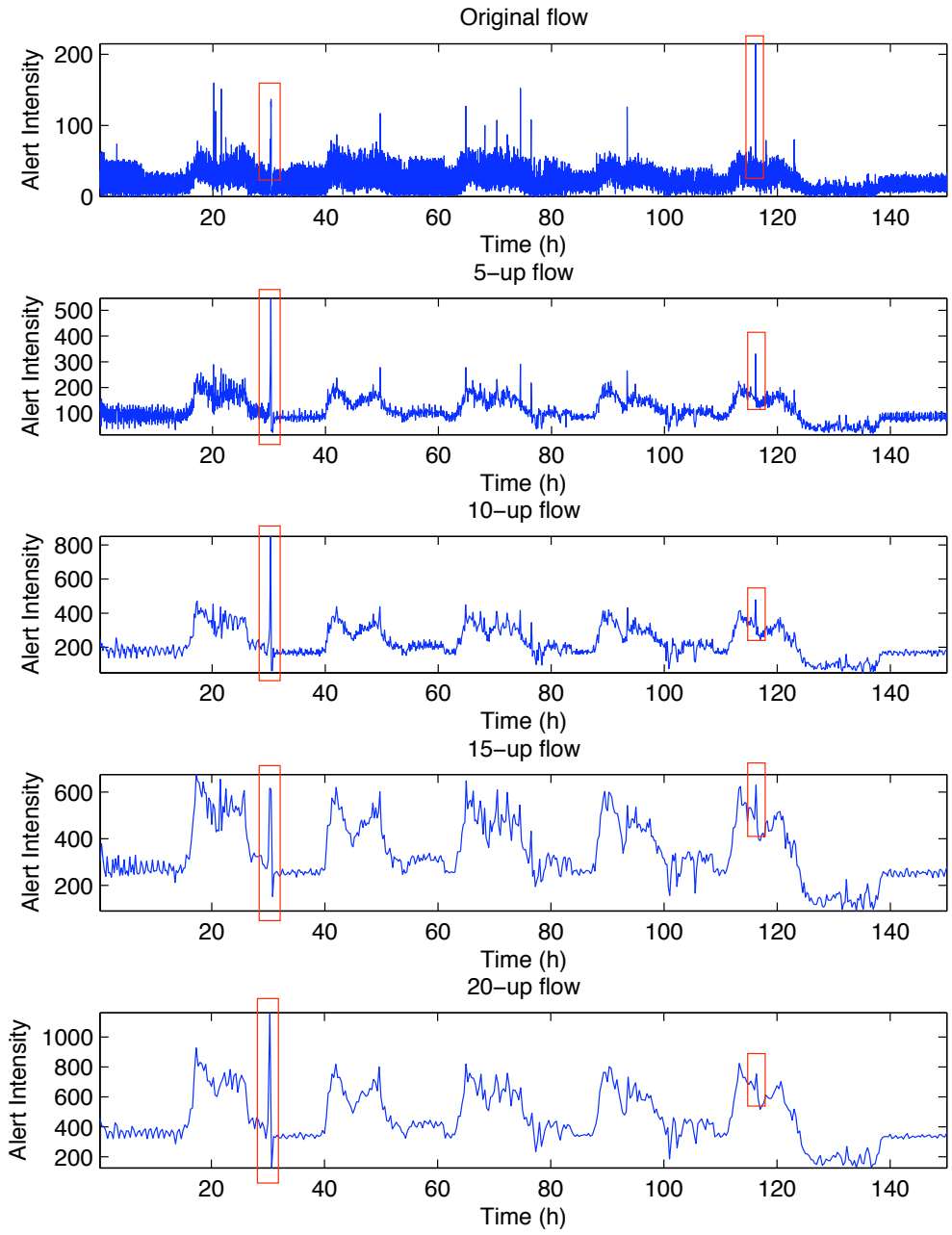


FIG. 3.7 – Effet de l'intervalle d'échantillonnage sur la visibilité des phénomènes dans le flux `NETBIOS SMB IPC$ share unicode access` de l'`ensemble-3`. L'intensité du flux original a été mesurée une fois par minute. Les flux suivants ont été obtenus en totalisant 5, 10, 15 et 20 échantillons. Le rectangle rouge de gauche montre la manière dont une échelle de temps plus large aide à détecter une augmentation de l'intensité des alertes. Le rectangle rouge de droite souligne la manière dont un changement rapide est lissé à des échelles de temps plus larges.

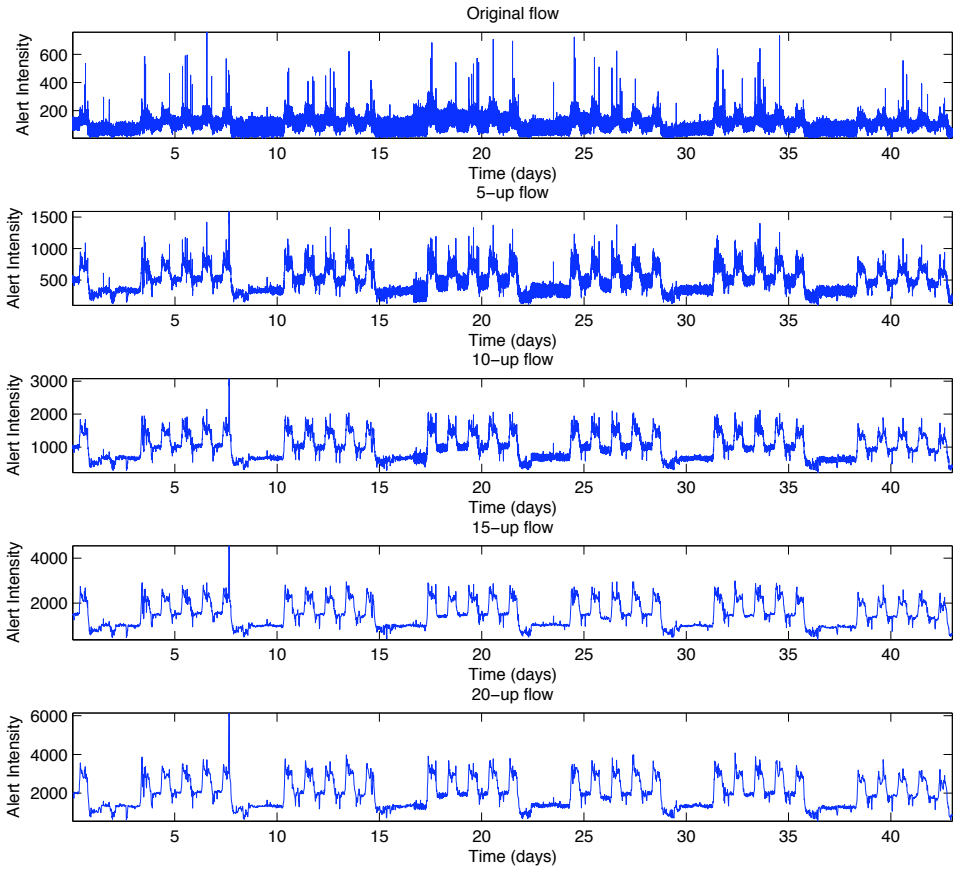


FIG. 3.8 – Exemple d'anomalies avec différentes échelles de temps. Les impulsions sont visibles dans le flux original et sont lissées sur des échelles de temps plus larges. Parallèlement, à des échelles de temps plus larges, le pic d'intensité aux alentours du jour 7 est plus clair

que l'intervalle d'échantillonnage augmente, nous aimerions souligner que les régularités sont visibles sur toute la plage des intervalles d'échantillonnage explorés. Ceci est très important, car cela signifie que, au moins en principe, nous pouvons modéliser le comportement du flux normal, quel que soit le t_s.

Typiquement, les phénomènes rencontrés sont intermittents et présentent des changements rapides comparativement au comportement normal. Le *type* d'anomalies est identique sur toutes les échelles de temps. Toutefois, le choix de t_s affecte l'échelle de temps des anomalies soulignées. Des t_s plus petits rendent les changements rapides visibles et des t_s plus larges font ressortir des anomalies de plus longue durée. Par ailleurs, avec un petit t_s, les fluctuations ont tendance à cacher des anomalies à plus long terme et, inversement, avec un grand t_s, les changements rapides sont lissés.

Un comportement normal plus lisse avec un t_s plus large est visible dans toutes les figures de flux présentées dans cette section, ce que fait le mieux ressortir la figure 3.7, en comparant le flux original avec 20 autres flux. Les rectangles rouges montrent comment les anomalies ressortent mieux à certains intervalles d'échantillonnage. En même temps, les anomalies se présentent sous forme de pics similaires à des impulsions à certains intervalles d'échantillonnage.

Traitement du flux le plus difficile

La détection visuelle des anomalies est plus facile lorsque le comportement normal est plus lisse. Ceci s'applique également aux méthodes automatisées que nous décrirons dans cette thèse. La logique veut que lorsque le comportement du flux normal fluctue largement ou contient des impulsions, il faut accepter que des écarts proportionnellement plus larges fassent partie du comportement normal, plutôt que de signaler une anomalie et vice versa. Dès lors, les flux obtenus avec des intervalles d'échantillonnage courts sont plus difficiles à traiter. Si nous pouvons éliminer par filtrage le comportement normal d'un flux avec $t_s = 1$ min, nous devrions être capables de l'éliminer par filtrage des flux obtenus en augmentant le t_s. C'est pour cette raison que nous nous concentrons sur les flux avec $t_s = 1$ min. Le choix offre également la plus grande rapidité de détection possible. Même si nous n'avons pas besoin d'une détection dans la minute pour ce type d'alertes, nous voulions savoir si nous pouvions modéliser et éliminer par filtrage le comportement normal de ces flux. Si

- il faut détecter des anomalies d'une échelle de temps plus large,
- il faut réduire la quantité d'informations à traiter,
- il n'y a pas besoin d'une surveillance minute par minute des alertes de faible priorité,

l'intervalle d'échantillonnage peut être augmenté[17]. Par exemple, au chapitre 6, nous présenterons quelques exemples d'utilisation d'intervalles d'échantillonnage plus larges pour saisir les anomalies sur une échelle de temps plus longue.

3.2.4 Variation des nombres de sources et destinations

Le tableau 3.6 montre le nombre de paires (source, destination), sources et destinations distinctes pour les flux les plus prolifiques de l'`ensemble-3` dans les colonnes (s,d), (s,*) et (*,d). Pour chaque critère, le nombre d'éléments ayant généré plus de 1000 alertes est illustré dans les colonnes > 1K.

[17] Du point de vue de l'implémentation, la même série temporelle des alertes peut être utilisée en totalisant deux observations ou plus du flux original.

TAB. 3.6 – Nombre de paires (source, destination) (s,d), sources (s,*) et destinations (*,d) pour les flux les plus prolifiques de l'`ensemble-3`. Pour chaque critère, le nombre d'éléments ayant généré plus de 1000 alertes est illustré à côté des nombres bruts

nom de la signature	**(s,d)**	**> 1K**	**(s,*)**	**> 1K**	**(*,d)**	**> 1K**
SNMP request udp	545	207	217	46	136	41
SNMP public access udp	417	80	215	45	127	41
ICMP L3retriever Ping	10 303	5949	3143	711	44	11
(http_inspect) BARE BYTE	6985	4974	4877	446	155	13
NETBIOS SMB-DS DCERPC	1962	1110	1635	525	16	4
NETBIOS SMB-DS IPC$	6992	3547	2241	544	37	10
NETBIOS SMB IPC$	6942	3530	2173	90	44	14

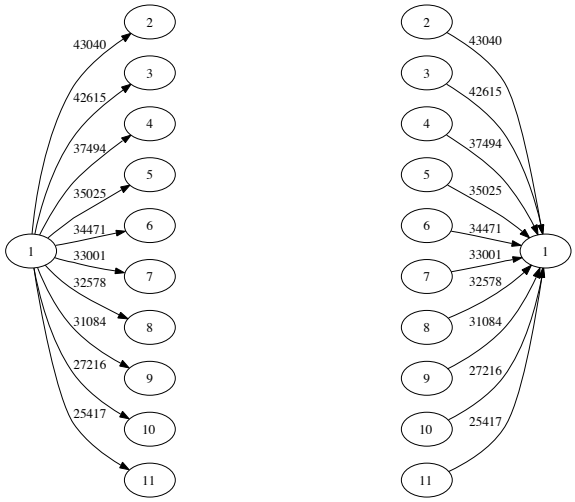
Il y a plus de sources que de destinations et, en particulier pour les alertes liées à NETBIOS, le nombre de destinations recevant plus de 1000 paquets déclenchant des alertes est petit. Pour les alertes liées à SNMP, la proportion de sources et de destinations est mieux équilibrée, mais on observe toujours une tendance à un nombre inférieur de destinations.

Il est possible que des approches telles que celles de Julisch [Jul03a] désigneraient certaines des dix destinations du flux `NETBIOS SMB-DS IPC$` comme causes primaires de ces types d'alertes, peut-être même pour toutes les alertes de type `NETBIOS`, à condition qu'une hiérarchie de généralisation adéquate soit définie pour les types d'alertes. Nous pensons que de telles méthodes de regroupement des alertes pourraient fournir des moyens de trouver de meilleurs critères d'agrégation que les nôtres. Toutefois, nous ne souhaitons pas filtrer directement tous ces types de positifs non pertinents, car ils véhiculent des informations utiles.

Même si l'on souhaite filtrer des alertes provoquées par le trafic de ou vers un certain hôte, cela pourrait se révéler difficile en raison des schémas de communication compliqués. La figure 3.9 illustre deux situations artificielles où il serait facile de filtrer les alertes générées par le trafic de l'hôte 1 (Figure 3.9(a)) ou vers l'hôte 1 (Figure 3.9(b)). Les hôtes sont représentés comme les noeuds du graphique. La direction de la communication, c'est-à-dire les bords, va de gauche à droite et l'annotation au bord indique le nombre d'alertes générées par la parie source-destination en question. Toutefois, en réalité, les schémas de communication peuvent être plus compliqués. Se servant de la même représentation, la figure 3.11 donne un exemple des liens de communication dans le flux `SNMP request udp`, un des flux avec un nombre plus modeste de sources et destinations. La figure a été générée avec Graphviz [GN00]. Les noeuds sont classés principalement dans trois colonnes, la gauche, la centrale et la droite. La version électronique du document permet un zoom rapproché.

Comme nous le voyons d'après les nombres, l'utilisation de critères d'agrégation à granularité plus fine que la signature génératrice pourrait augmenter significativement le nombre de flux. Etant donné le nombre élevé de paires (source, destination), l'énumération et la sélection manuelles prendraient du temps.

C'est l'une des raisons d'utiliser des critères d'agrégation à granularité aussi large que la signature génératrice. Nous reconnaissons que nous pourrions sélectionner l'agrégation automatiquement, par exemple en vérifiant périodiquement le nombre d'alertes dans les flux définis par différents critères d'agrégation comme `(sig, *,*)`, `(sig,src,*)` ou `(sig,*,dst)`, où la notation est la même qu'à la section 2.2.1. En cas de dépassement de seuils donnés de volume et persistance des alertes, un nouveau flux sera généré. La persistance se me-



(a) From host 1 (b) To host 1

FIG. 3.9 – Exemples de schémas de communication faciles à filtrer par des méthodes traditionnelles

sure sous forme du nombre de valeurs d'intensité non nulles y_t pour le flux. De manière similaire, un flux pourrait être éliminé si $y_t = 0$ pendant trop longtemps. Il est à noter que, dans une telle situation, les flux avec des critères d'agrégation à granularité plus large pourraient chevaucher des flux à granularité plus fine.

3.3 Alternatives et inconvénients

Dans cette section, nous regroupons et discutons brièvement des approches alternatives de traitement de ces alertes informatives. Nous mentionnons également certains inconvénients de ce type de surveillance.

3.3.1 Alternatives

Les signatures informatives ont tendance à générer de grosses quantités d'alertes. Les possibilités actuelles pour les traiter sont :

Désactivation de la signature Les alertes sans aucune signification par rapport aux politiques du site devraient être désactivées. Toutefois, si les alertes ne véhiculent pas des informations utiles, ce choix est de toute évidence radical.

Seuillage Snort, par exemple, permet de configurer des seuils et limites fixes pour les signatures. Un seuil signifie qu'une alerte est effectivement générée à chaque fois m que la signature est déclenchée, et limite signifie que les alertes sont générées seulement les premières fois m que la signature est déclenchée. Ces compteurs peuvent être suivis par la source ou la destination et la période peut être définie. Les seuils pour les alertes devinant les mots de passe sont même nécessaires, leur utilité est limitée par un flux d'une intensité d'alertes qui évolue avec le temps.

Filtrage basé sur les attributs. Les approches de filtrage comme celles de Julish se basent sur des attributs d'alertes individuels et elles sont conçues pour éliminer les

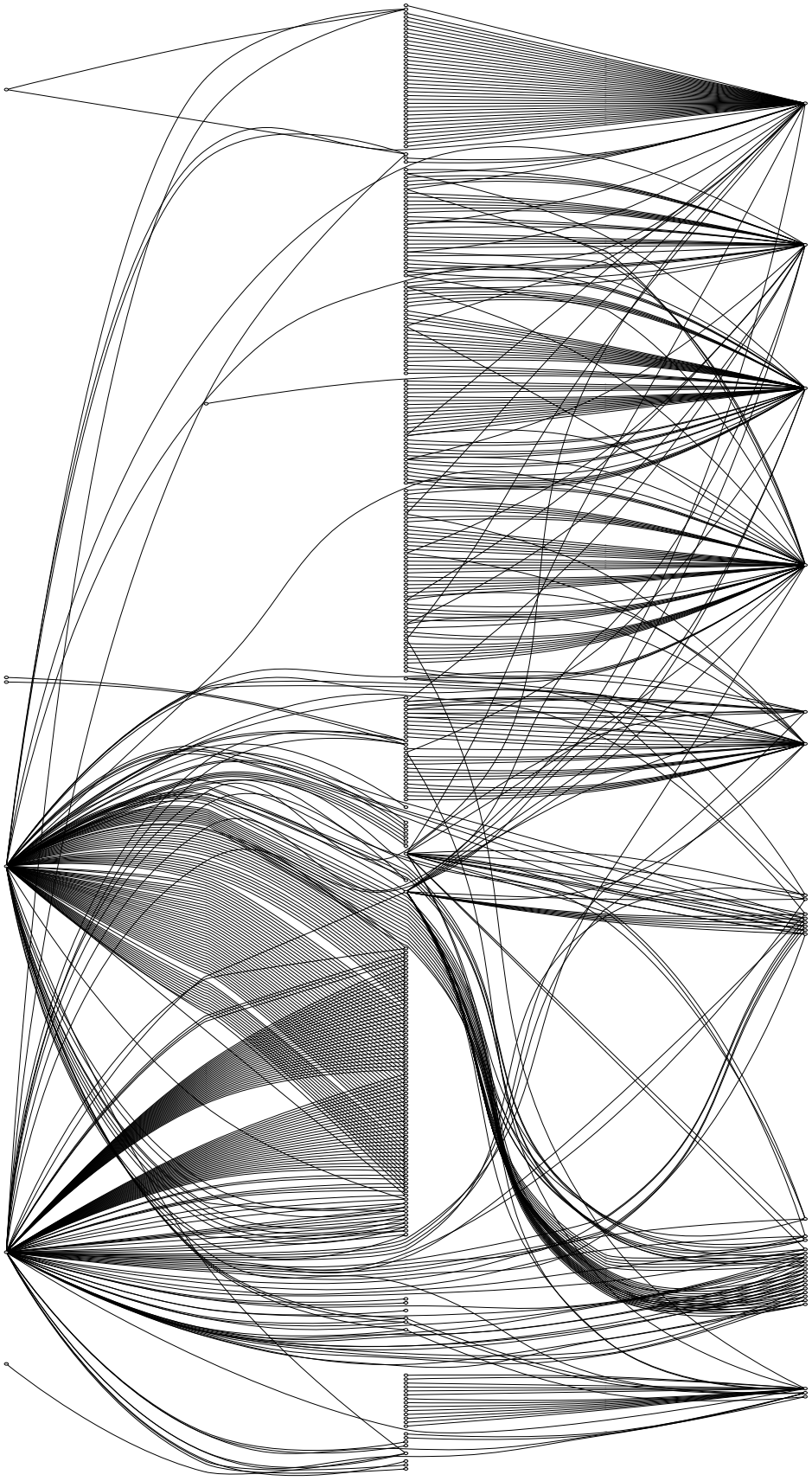


FIG. 3.10 – The communication pattern of SNMP messages from `set-3`. The direction of communication is from left to right, the nodes are mainly in three columns. The electronic version of the dissertation allows zooming : the labels on the edges indicate the number of alerts caused by the source-destination pair

FIG. 3.11 – Schéma de communication des messages SNMP de l'`ensemble-3`. La direction

faux positifs. Pour les positifs informatifs, nous devons tenir compte du contexte des alertes si nous souhaitons détecter les changements dans le flux d'intensité, tels qu'ils apparaissent à la figure 3.3(a).

Les sondes de réseau ordinaires, par exemple les informations fournies par *netflow*[18], ou les comptages de paquets et octets provenant des équipements réseau sensibles à SNMP pourraient être utilisés pour surveiller certains de ces paquets au niveau du réseau. Cette approche est utilisée, par exemple, dans [BKPR02]. Toutefois, nous utilisons Snort pour les raisons suivantes :

- Même si une sonde de réseau ordinaire pouvait être utilisée pour surveiller les messages ICMP, par exemple, et que différents types de messages pouvaient être séparés avec les informations d'en-tête, cela ne suffirait pas pour nos besoins. Nous avons besoin d'agrégation d'alertes à granularité plus fine utilisant un schéma correspondant au message dans le contenu du paquet. Par exemple, les messages d'écho ICMP peuvent contenir des informations sur l'origine supposée du paquet Ping dans le contenu, ce qui nous intéresse. De plus, [SLB+06,SLO+06,BLAO05] signalent que les comptages généraux des octets et paquets peuvent constituer une source de données difficile, exigeant des techniques d'analyse sophistiquées pour identifier les attaques.
- Par son agrégation de paquets, Netflow crée une couche d'abstraction qui n'est pas présente lorsque nous utilisons des alertes individuelles de Snort. Nous pourrions veiller à exécuter nous-mêmes une abstraction du niveau du flux qui répond mieux à nos besoins que celle fournie par Netflow.
- Snort est l'interface avec le réseau surveillé à notre disposition. Il est un fait que nous devons l'accepter dans notre environnement de travail et nous essayons de faire le meilleur usage possible des outils dont nous disposons.

3.3.2 Inconvénients du bruit des alertes

Maintenir les signatures informatives activées présente certains désavantages. Le volume important des alertes augmente la charge des sondes et, dans le cas d'un enregistrement centralisé, les transferts d'alertes consomment de la largeur de bande réseau et augmentent également la charge du stockage des alertes.

De plus, chaque alerte passant à travers les composantes de haut niveau, comme les moteurs de corrélation, consomme des ressources. Le dernier maillon de la chaîne, l'opérateur humain, est également vite dépassé par les alertes. Même si la console d'alertes permet à l'opérateur de limiter le type d'alertes affichées, il peut y avoir des effets néfastes indirects. Si la console accède à une base de données relationnelle pour afficher les alertes, les demandes peuvent être considérablement ralenties en raison du nombre élevé d'alertes dans la base de données.

Pour donner une indication des volumes de données, la sonde utilisée pour collecter des données dans l'`ensemble-3` a généré 300K-2M d'alertes par jour. En format `syslog` de Snort, un lot d'environ 300K d'alertes correspond à 100 Mbytes non comprimés et à 2-4 Mbytes comprimés. Les alertes ont été introduites dans une base de données qui, avec 36M d'alertes, utilise environ 10 Gbytes sur le disque. Nous n'avons pas enregistré les messages dans le contenu des paquets, car cela aurait augmenté considérablement la taille. Nous considérons toujours ces charges comme raisonnables, mais l'archivage des anciennes alertes devrait être pris en compte pour améliorer le temps de réponse aux demandes provenant, par exemple, de la console d'alertes.

[18] `http://www.cisco.com/go/netflow/`, consulté le 18.07.2006.

3.4 Conclusion

Dans ce chapitre, nous avons vu quelles causes des flux d'alertes créaient quels types d'alertes. Ceci englobe des positifs non pertinents et des faux positifs ainsi que, dans une moindre mesure, également des vrais positifs. L'utilisation informative de sondes IDS est la plus grande responsable des ensembles de données examinés.

Nous avons également vu que ces alertes informatives ne peuvent pas être analysées une par une. Les informations pertinentes se trouvent dans le contexte des alertes, dans les variations de l'intensité du flux d'alertes.

Nous avons analysé les caractéristiques des signatures les plus prolifiques dans trois ensembles de données et constaté que les flux d'alertes informatives présentent une régularité importante, en termes de composantes constantes du flux et/ou de dépendances par rapport au jour de la semaine et de l'heure du jour dans l'intensité du flux. Le comportement du flux d'alertes est resté stable sur des périodes d'observation de plus d'un mois et entre deux ensembles de données collectées à plusieurs mois d'intervalle depuis le système surveillé. Le comportement normal du flux est visible à tous les intervalles d'échantillonnage utilisés dans la plage d'une minute à plusieurs heures.

Nous avons constaté que les anomalies présentes dans ces flux sont pour la plupart des impulsions, des chutes d'intensité brusques, des changements abrupts et autres phénomènes intermittents. Des anomalies ont été constatées à différentes échelles de temps, mais elles ont la même forme à des intervalles d'échantillonnage proches de l'échelle de temps de l'anomalie.

La nature des anomalies est similaire tout au long des échelles de temps examinées, mais à des échelles de temps plus larges, le comportement normal devient plus lisse. Dès lors, l'analyse des flux avec de petits intervalles d'échantillonnage est plus difficile et donne par conséquent une meilleure idée des capacités et limitations des méthodes de traitement.

Nous avons également discuté des inconvénients provoqués par la surveillance informative. Le problème principal est la charge pour le système et l'utilisateur. Il conviendrait d'examiner au cas pas cas si oui ou non les alertes informatives sont intéressantes et si oui ou non les ressources sont suffisantes en termes de personnel et de matériel pour traiter le volume et réaliser l'analyse.

Les trois chapitres suivants présenteront trois différentes méthodes qui aident l'utilisateur à traiter le volume et extraire les informations utiles sur l'état du système surveillé. Sous la forme présentée dans la présente thèse, ces techniques ont pour objectif de faciliter la partie analyse du problème. Il serait toutefois faisable d'intégrer ces techniques de traitement aux sondes et, par conséquent, nous pourrions réduire également la charge sur l'infrastructure de gestion des alertes. Le traitement au niveau des sondes pourrait réduire le nombre d'échanges d'alertes sur le réseau et le nombre d'alertes enregistrées et traitées par les installations de stockage et corrélation des alertes.

Chapitre 4

Modélisation des tendances

Dans les chapitres précédents, nous avons constaté que le filtrage basé sur les attributs d'alertes individuelles ne convient pas parfaitement aux alertes informatives. Leur importance dépend du contexte des alertes, du nombre d'alertes similaires proches dans le temps et des risques que le filtrage des alertes basé sur les attributs élimine des alertes qui véhiculent des informations intéressantes lorsqu'elles sont agrégées avec d'autres alertes similaires. Nous devons donc adopter une vision plus abstraite et traiter ces types d'alertes comme des flux d'alertes.

Ce chapitre présente une approche simple de la modélisation du comportement normal du flux. L'objectif est d'éliminer par filtrage des alertes provoquées par l'utilisation normale du système et de mettre en évidence des phénomènes intéressants du flux à examiner plus en profondeur. Le modèle se base sur les tendances à court terme dans le flux d'alertes. Ces tendances sont saisies à l'aide d'une méthode appelée moyenne glissante pondérée exponentiellement (EWMA).

La section 4.1 présente les méthodes que nous utilisons dans ce chapitre, tandis que la section 4.2 aborde le travail réalisé du point de vue de la méthode. Les expérimentations réalisées avec deux ensembles de données sont décrites à la section 4.3. Nous discuterons ensuite des points faibles et des points forts de l'approche à la section 4.4, pour conclure le chapitre à la section 4.5.

4.1 Méthodes

Cette section présente, à la sous-section 4.1.1, les cartes de contrôle EWMA utilisées comme système de maîtrise statistique des processus (SPC), puis ses adaptations à la détection d'intrusions à la sous-section 4.1.1. La principale différence entre les applications dans le cadre du SPC et la détection d'intrusions est l'environnement dynamique dans lequel la détection d'intrusions a lieu. De plus, notre besoin en matière de détection est quelque peu différent du domaine d'origine de la carte de contrôle EWMA. Ceci s'applique également, mais dans une moindre mesure, au travail connexe dans le domaine de la détection d'intrusions. Nous discuterons de ces points ci-après.

4.1.1 Cartes de contrôle EWMA

Dans cette section, nous présentons brièvement les contextes mathématiques d'EWMA, son utilisation pour la procédure des cartes de contrôle, puis notre variation pour la surveillance du bruit.

Contexte de la maîtrise statistique des processus

Les cartes de contrôle EWMA ont été mises au point initialement pour la maîtrise statistique des processus par Roberts [Rob59], qui utilisait le terme "moyennes glissantes géométriques" plutôt que EWMA, et depuis lors, la carte et en particulier la moyenne glissante pondérée exponentiellement, sont utilisées dans divers contextes, comme les applications économiques et la détection d'intrusions [YVC03, YBC02, MWR02]. De plus amples informations à ce sujet figurent à la section 4.2.

Dans le cadre du SPC, un processus de fabrication est considéré comme une entité mesurable avec une distribution. L'on considère que la qualité générale du produit qui résulte du processus dépend de la moyenne de processus, qui doit être maintenue au niveau fixé, avec des variations aussi minimes que possible. Une carte de contrôle EWMA peut être utilisée pour surveiller la moyenne de processus en mettant à jour une *moyenne glissante exponentiellement* de la valeur de processus. La moyenne est comparée aux *limites de contrôle* prédéterminées, définissant la plage acceptable de valeurs. Nous allons maintenant décrire cette procédure plus en détail.

La moyenne glissante pondérée exponentiellement z_t des observations y_t est définie de la manière suivante :

$$z_t = (1-\lambda)z_{t-1} + \lambda y_t \ , \tag{4.1}$$

où $0 < \lambda < 1$. Ici, z_t est la valeur actuelle de la moyenne lissée exponentiellement, z_{t-1} est la valeur lissée précédente et y_t, l'observation actuelle. La valeur EWMA mise à jour est une moyenne pondérée de la valeur EWMA précédente et de l'observation actuelle avec des pondérations $(1-\lambda)$ et λ, respectivement. Le terme "lissage exponentiel" est également utilisé pour ce type de lissage. Une troisième façon de considérer la moyenne glissante exponentielle est le filtrage passe-bas des observations.

Cette formulation récursive distingue EWMA des moyennes glissantes de base, comme les moyennes glissantes simples ou les moyennes glissantes pondérées. Le lissage exponentiel tient compte de toutes les données passées, dont la signification décroît exponentiellement en fonction du temps. Toutefois, en même temps, seules la valeur lissée précédente et la mesure actuelle sont nécessaires pour calculer la nouvelle valeur lissée. La décroissance est contrôlée par le facteur λ et $(1-\lambda)$ est appelé le *facteur de lissage*. Le terme devient plus clair en reformulant (4.1) de la manière suivante :

$$\begin{aligned} z_t &= \lambda y_t + \lambda(1-\lambda)^1 y_{t-1} + \lambda(1-\lambda)^2 y_{t-2} + \dots \\ &\quad \dots + \lambda(1-\lambda)^{t-2} y_2 + \lambda(1-\lambda)^{t-1} y_1 + (1-\lambda)^t y_0 \ , \end{aligned} \tag{4.2}$$

où $t \geq 0$. Nous pouvons maintenant constaté que l'observation actuelle y_t a la pondération λ et que les anciennes données de l'instant $t-j$ ont la pondération $\lambda(1-\lambda)^j$, ce qui signifie que les anciennes observations sont lissées avec les pondérations à décroissance exponentielle. Si l'on s'intéresse à la tendance à long terme, il convient d'utiliser des facteurs de lissage importants et vice versa.

Dans les applications SPC, il arrive souvent que la statistique surveillée dans (4.1) soit la moyenne d'un moyenne de processus industriels et que y_t soit la moyenne de sous-groupe de n échantillons prélevés à l'instant t. L'écart-type σ_z de z peut être obtenu de la manière suivante :

$$\sigma_z = \sqrt{\frac{\lambda}{2-\lambda}}\sigma_{\bar{y}} \ . \tag{4.3}$$

Etant donné que y_t est une moyenne de n échantillons,$\sigma_{\bar{y}}$ est $\sigma_y/\sqrt{n}$, où σ_y est l'écart-type de y, supposé être connu a priori.

Les limites de contrôle inférieures et supérieures (UCL, LCL) définies de la manière suivante :

$$y_0 \pm 3\sigma_z \tag{4.4}$$

définissent l'intervalle pour z, où le processus est considéré être sous contrôle. Ici, y_0 est la moyenne nominale de processus, également supposée être connue a priori. Pour chaque nouvelle mesure, la valeur actuelle de la statistique z est calculée à l'aide de (4.1) et, si les limites de contrôle sont dépassées, une instabilité est signalée.

Le lissage exponentiel est à peu près équivalent à une moyenne glissante standard (ou simple), avec une taille de fenêtre n. Selon Roberts [Rob59], pour un λ donné, une taille de fenêtre n approximativement équivalente est déterminée par :

$$n = \frac{2}{\lambda} - 1 \ . \tag{4.5}$$

Avec cette équation, la vitesse de décroissance devient plus intuitive, connaissant la longueur d'intervalle de mesure. Nous appelons le produit de n et la longueur de l'intervalle d'échantillonnage pour les observations y_t, la *mémoire* de la statistique, les événements plus anciens n'ayant que peu de signification pour le z actuel. Par exemple, le facteur de lissage 0.92 se traduirait par une taille de fenêtre 24, tandis que pour 0.80, le n correspondant est 9. Cela montre également qu'un facteur de lissage plus large correspond au lissage sur un nombre d'échantillons plus important.

Carte de contrôle pour les flux d'alertes

Nos besoins diffèrent nettement de ceux de Robert et également, dans une moindre mesure, de ceux du travail connexe réalisé dans la détection d'intrusions (section 4.2). Nous décrirons ci-après notre variation de la technique, largement inspirée de [MWR02] et nous fournirons les raisons des changements et des choix.

La mesure surveillée est l'intensité du flux y_t, le nombre d'alertes par intervalle de temps. Souvenons-nous que le flux d'alertes se compose typiquement d'alertes générées par une signature, mais d'autres flux, comme des alertes générées par toute une classe de signatures, ont également été utilisées. L'intensité y_t est utilisée pour former la statistique EWMA de (4.5) que nous appelons la *tendance* au temps t. Il est à noter que, comparées aux tendances généralement utilisées dans l'analyse de séries temporelles, les nôtres sont des tendances à terme nettement plus court. Par exemple, dans [BD02, p.10] la *fonction à changement lent* est appelée tendance et les exemples incluent des tendances visibles dans des ensembles de données s'étalant sur plusieurs années. Dans notre cas, nous examinons des tendances à très court terme, de l'ordre de quelques heures et jours, dans des données qui changent rapidement.

Il est quasi impossible de définir une moyenne nominale comme la ligne de base de test y_0 pour (4.4), étant donné que ces flux évoluent significativement au cours du temps. Comme le spécifiaient Mahadik et al. [MWR02], pour intégrer la nature dynamique non stationnaire des flux, la ligne de base de test peut s'adapter aux changements constatés dans le flux d'alertes et les limites de contrôle pour l'intensité des alertes y_t à l'instant t sont les suivantes :

$$z_{t-1} \pm n \cdot \sigma_{z_{t-1}} \ . \tag{4.6}$$

Ici, n n est un facteur exprimant l'acceptabilité de l'ampleur d'un écart par rapport à la tendance et σ_{z-1} est l'écart-type de z à l'intervalle $t-1$. Les limites de contrôle pour l'intervalle t sont donc calculées à l'aide des statistiques de tendance depuis l'intervalle $t-1$.

Pour obtenir l'écart-type σ_z, une autre statistique EWMA :

$$z_t^2 = (1-\lambda)z_{t-1}^2 + \lambda y_t^2 \tag{4.7}$$

est mise à jour, où y_t est l'intensité actuelle comme dans le calcul de la tendance. L'écart-type s'obtient par l'équation suivante :

$$\sigma_{z_t} = \sqrt{z_t^2 - (z_t)^2} \ . \tag{4.8}$$

Maintenant, pour chaque nouvel intervalle et pour chaque flux d'alertes, 1) l'intensité des alertes est mesurée, 2) les limites de contrôle sont calculées, et 3) la décision de savoir si l'intervalle est anormal ou non est prise sur la base de l'intensité des alertes et des limites de contrôle actuelles. Dans [YVC03] et [MWR02] les auteurs testent l'intensité lissée des événements par rapport aux limites de contrôle. Ils utilisent un facteur de lissage plus grand avec (4.1) pour obtenir la ligne de base z_t de (4.6) et appliquent (4.1) avec un facteur de lissage plus petit pour obtenir une intensité lissée des événements, testée par rapport aux limites de contrôle. Ceci a pour objectif de réduire l'effet de valeurs sauvages dans les observations. Toutefois, dans le cas des alertes, ces observations aberrantes sont généralement intéressantes pour l'opérateur et pour tester l'intensité brute ou, en d'autres termes, en utilisant $(1-\lambda) = 0$ pour obtenir la valeur qui est testée par rapport aux limites de contrôle, nous avons pu mieux saisir les petites variations survenant dans des flux d'alertes extrêmement stables.

4.2 Travail connexe

Le travail connexe dans la corrélation des alertes a été examiné à la section 2.2. Dans cette section, nous allons nous concentrer sur le travail connexe du point de vue de la méthode. En plus de la pléthore d'autres applications, le modèle EWMA a également été exploité pour la détection d'intrusions. Deux approches, l'une sans nom de Ye et al. [YVC03,YBC02] et une autre, ArQoS, développée par Mahadik et al. [WMR03,MWR02], utilise toutes deux EWMA. Un algorithme Holt-Winters plus sophistiqué est utilisé par Brutlag pour la détection des anomalies dans les données de gestion de réseau [Bru00]. Nous passerons chaque approche en revue et nous soulignerons les différences avec notre méthode, avant de poursuivre par les expérimentations à la section 4.3.

4.2.1 Surveillance de l'intensité des événements d'audit BSM

Ye et al. testent différentes cartes de contrôle dans [YVC03], une pour les données autocorrélées et l'autre pour les données non corrélées. Dans [YBC02] également une carte de contrôle pour l'écart-type est utilisé. Ye et al. examinent l'adaptation de ces contrôles pour la détection DoS en surveillant l'intensité des événements d'audit BSM sur des ordinateurs avec le système d'exploitation Solaris. Ils utilisent donc une source de données à commande

système, tandis que nous surveillons une source à commande réseau. Ils agrègent tous les événements d'audit en un seul flot d'événements et comptent l'intensité des événements pendant l'intervalle d'échantillonnage. La carte des données non corrélées est proche de celle que nous utilisons, avec les différences suivantes :

- L'intensité des événements est lissée deux fois. Cela équivaudrait aux observations du filtrage passe-bas y_t avant de les utiliser dans (4.1).
- Ils utilisent une variation d'EWMA, qui permet la mise à jour de l'intensité lissée des événements lorsque l'événement j survient pendant l'intervalle d'échantillonnage t au lieu de survenir à la fin de l'intervalle d'échantillonnage t. Désignant l'intervalle d'échantillonnage pendant lequel le j^{eme} survient par t_j, nous pouvons transformer (4.1) en l'équation suivante :

$$z_{t_j} = (1-\lambda)^{t_j - t_{j-1}} z_{t_{j-1}} + \lambda \cdot 1 \ . \tag{4.9}$$

  Le désavantage de cette formule réside dans le fait que lorsqu'il n'y a pas d'événements, z_{t_j} n'est pas mis à jour. Comme la carte est utilisée pour la détection des attaques DoS, l'hypothèse est que l'intensité des événements augmente sous l'effet de l'attaque et les événements manquants sont moins intéressants. Dans notre cas, les augmentations et les diminutions sont tout aussi intéressantes. Nous préférons donc utiliser le test par intervalle d'échantillonnage pour saisir également les événements manquants avec la même carte, sans tests supplémentaires.
- Les limites de contrôle se basent sur la variance de l'erreur de prédiction "one-step-ahead" (une étape d'avance). Nous servant de notre notation, la prédiction one-step-ahead pour y_t (filtré passe-bas) est la suivante

$$\hat{y}_t = z_{t-1}$$

  et l'erreur de prédiction ont-step-ahead e_t est

$$e_t = y_t - \hat{y}_t = y_t - z_{t-1} \ ,$$

  où et sont indépendants et distribués de manière identique (iid) avec un écart moyen nul et un écart type σ_e ; σ_e est estimé de la manière suivante :

$$\hat{\sigma}^2_{e_t} = (1-\gamma)\,\hat{\sigma}^2_{e_{t-1}} + \gamma e_t^2 \ ,$$

  où $0 < \gamma < 1$ et $\hat{\sigma}_{e_{t-1}}$ est utilisé dans (4.6) au lieu de $\sigma_{z_{t-1}}$ pour obtenir les limites de contrôle pour y_t.

La carte pour les données corrélées utilise des limites statiques et teste l'intensité lissée des événements deux fois par rapport à ces limites. La carte EWMS pour l'écart-type utilise l'écart-type glissant pondéré exponentiellement S défini de la manière suivante :

$$S_t^2 = (1-\beta)\,S_{t-1}^2 + \beta\,(y_t - \mu_y) \ , \tag{4.10}$$

où μ_y est la moyenne pour y_t, supposé connu a priori ou déduit des données d'entraînement. L'estimation de l'écart-type est d'environ χ^2 distribué avec $\nu = (2-\beta)\beta$ degrés de liberté. Si σ_0 dénote l'écart-type pendant le contrôle, les limites de contrôle sont obtenues de la manière suivante :

$$\mathrm{UCL} = \hat{\sigma}_0 \sqrt{\frac{\chi^2_{\nu,\alpha/2}}{\nu}} \ ,$$
$$\mathrm{LCL} = \hat{\sigma}_0 \sqrt{\frac{\chi^2_{\nu,1-\alpha/2}}{\nu}} \ ,$$

où $\chi^2_{\nu,\alpha/2}$ est le point de la distribution χ^2 avec v degrés de liberté pour laquelle la probabilité de queue est $\alpha/2$.

Ils en ont conclu que toutes les cartes différentes pouvaient être utilisées pour détecter des attaques provoquant des changements significatifs statistiquement parlants au niveau de l'intensité des événements.

En plus des cartes de contrôle EWMA, dans [EY01], Ye et Emram testent l'efficacité de la métrique de Canberra dans la détection d'intrusions et utilisent également EWMA comme technique de lissage. Ils suivent les intensités des événements de tous les types d'événement d'audit BSM avec EWMA dans un vecteur de 284 valeurs. Le profil normal est obtenu sous forme de l'écart moyen et type des distances entre les vecteurs d'intensité des événements en conditions d'utilisation normale du système. Ye et al. utilisent une version modifiée de la métrique de Canberra comme mesure de la distance. La métrique de Canberra a été identifiée comme tout à fait inadaptée à la détection d'intrusions.

4.2.2 ArQoS

ArQoS est un IDS qui surveille les paramètres de Qualité de Service (QoS) dans un réseau DiffServ pour détecter les attaques contre QoS. Les paramètres surveillés sont, par exemple, le taux de bits, la gigue et la perte de paquets. ArQoS utilise une source de données à commande réseau.

Les comptages d'octets et de paquets ainsi que les mesures de gigue sont analysés à l'aide de deux techniques, un test chi^2 pour la statistique Q basé sur l'algorithme de NIDES [JV93] et une carte de contrôle EWMA adaptée. La statistique Q est utilisée pour les valeurs non stationnaires, tandis que la carte de contrôle EWMA est utilisée pour les valeurs stationnaires. Si la situation de la métrique surveillée évolue, le système bascule en conséquence entre la statistique Q et la carte de contrôle EWMA.

La statistique Q mesure le degré d'anomalie entre les distributions à long terme et à court terme pour les observations avec les valeurs possibles $E_1, \ldots, E_k$ et les probabilités attendues $p_1, \ldots, p_k$. En désignant les nombres de résultats réels dans une expérimentation de N essais comme $Y_1, \ldots, Y_k$, la variable aléatoire Q est définie de la manière suivante :

$$Q = \sum_{i=1}^{k} \frac{(Y_i - Np_i)^2}{Np_i} \ . \tag{4.11}$$

Pour grand N, Q est environ chi^2 distribué avec $k-1$ degrés de liberté. Pour normaliser les valeurs Q à l'intervalle $[0, \infty[$, une nouvelle variable S est utilisée. Elle est définie de la manière suivante :

$$S = \Phi^{-1}\left(1 - \frac{p\,(Q > q)}{2}\right) \ , \tag{4.12}$$

où Φ^{-1} est la fonction de distribution cumulative inverse d'une variable $N(0,1)$.

Pendant la période d'entraînement, la plage de valeurs observées pour chaque paramètre est divisée en 32 plages de taille égale, qui constituent les valeurs possibles $E_1, \ldots, E_k$. Les distributions à long et court termes sont actualisées à l'aide de moyennes glissantes pondérées exponentiellement pour mettre à jour les comptages des événements dans chaque plage. Les valeurs des distributions à long et court termes correspondent (respectivement) aux termes Np_i et Y_i dans (4.11). La distribution à court terme est comparée à la distribution à long terme pour obtenir les valeurs S. Si les valeurs S correspondent à une probabilité de queue Q suffisamment petite, une anomalie est signalée.

Selon Mahadik et al., cette approche ne fonctionne pas bien avec des variables stationnaires. Par conséquent, si les observations commencent à tomber dans seulement trois plages ou moins, la méthode de détection passe à la carte de contrôle EWMA. La carte ArQoS surveille une statistique lissée, tandis que nous filtrons des observations non filtrées. Dans notre cas, cela équivaudrait à procéder à un filtrage passe-bas des observations y_t avant de les tester par rapport aux limites de contrôle. Il est à noter que cette méthode est différente des deux lissages utilisés dans [YBC02, YVC03], où le filtrage passe-bas est effectué en premier lieu et affecte, par exemple, les calculs des limites de contrôle. Notre carte de contrôle s'inspire largement de la carte utilisée dans ArQoS, mais les applications sont quelque peu différentes :

- Plutôt que des variables hautement stationnaires, nous surveillons l'intensité de flux non stationnaires. De plus, nous ajustons notre carte pour détecter les variations à court terme plutôt que les changements de moyenne.
- Les sources de données sont différentes, ArQoS utilisant des données provenant des routeurs, tandis que nous inspectons les alertes IDSu
- Mahadik et al. utilisent la carte de contrôle pour détecter les dégradations dans QoS, tandis que nous l'utilisons pour détecter les changements dans le comportement normal du système.

4.2.3 Détection des comportements aberrants

Brutlag utilise l'algorithme de prévision Holt-Winters plus sophistiqué dans [Bru00] pour détecter les anomalies à court terme dans diverses séries temporelles de service réseau. Il considère EWMA comme un algorithme pour prédire la valeur y_{t+1} à l'aide des observations jusqu'à y_t. L'estimation de $\hat{y}_{t+1}$ s'obtient de la manière suivante :

$$\hat{y}_{t+1} = (1 - \lambda)\, \hat{y}_t + \lambda y_t \ ,$$

où $0 < \lambda < 1$. La prévision de Holt-Winters se base sur EWMA et suppose que la série temporelle consiste en trois composantes, la composante constante, la composante tendancielle et la composante saisonnière. La prévision de la valeur de la série suivante est la somme de ces composantes :

$$\hat{y}_{t+1} = a_t + b_t + c_{(t+1)-m} \ ,$$

où a_t et la composante constante, b_t la composante tendancielle et c_t la composante saisonnière avec la période m. Pour percevoir le lien avec EWMA, rappelons la notion de $(1 - \lambda)$ comme la *pondération de la valeur lissée précédente*, et λ comme la pondération de *l'observation actuelle* dans les équations suivantes. Nous verrons ensuite comment l'on obtient les différentes composantes. Premièrement, la composante constante :

$$a_t = (1 - \alpha)\,(a_{t-1} + b_{t-1}) + \alpha\,(y_t - c_{t-m}) \ ,$$

où $0 < \alpha < 1$. La nouvelle composante constante est la somme pondérée de 1) la valeur précédente de la composante constante ajustée avec l'effet de la tendance, et 2) l'observation actuelle avec la composante saisonnière supprimée. La composante tendancielle s'obtient comme suit :

$$b_t = (1 - \beta)\, b_{t-1} + \beta\, (a_t - a_{t-1}) \quad ,$$

où $0 < \beta < 1$. La nouvelle composante tendancielle est la somme pondérée de 1) la composante tendancielle précédente et 2) la différente entre les composantes constantes actuelle et précédente. La composante saisonnière c_t s'obtient comme suit :

$$c_t = (1 - \gamma)\, c_{t-m} + \gamma\, (y_t - a_t)$$

où $0 < \gamma < 1$. La nouvelle composante saisonnière est la somme pondérée de 1) la valeur saisonnière précédente correspondant à cette phase du cycle saisonnier, c'est-à-dire m étapes auparavant et 2) la différence entre l'observation actuelle et la composante constante.

Les facteurs de lissage sont $(1 - \alpha), (1 - \beta)$ et $(1 - \gamma)$, des facteurs de lissage plus grands ralentissant l'adaptation aux observations actuelles et signifiant une plus grande pondération par rapport aux données historiques.

Les limites de contrôle ou les limites de confiance se basent sur une mesure de la variabilité saisonnière de la modélisation de l'écart et sont définies comme suit :

$$d_t = (1 - \gamma)\, d_{t-m} + \gamma |y_t - \hat{y}_t| \quad , \tag{4.13}$$

où d_t est l'écart prévu au temps t et $|y_t - \hat{y}_t|$ est la valeur absolue de l'erreur de prévision de l'algorithme de Holt-Winters. Le paramètre γ est le même que celui utilisé pour les mises à jour de la composante saisonnière. Les limites de contrôle pour les observations γ_t sont maintenant définies comme l'intervalle $[\hat{y}_t - nd_t, \hat{y}_t + nd_t]$ où n est un facteur qui détermine la largeur des limites de contrôle. Pour éviter les faux positifs, une fenêtre glissante de longueur w est utilisée et une anomalie est signalée si plus de k observations de la fenêtre tombent en dehors des limites de contrôle. Pour plus de détails sur l'algorithme de Holt-Winters, voir par exemple [BD02, section 9.3].

Brutlag aborde également le lissage temporel dans un cycle saisonnier pour c_t et d_t, le choix des paramètres et présente les modifications apportées à RRDtool[1] pour appliquer la prévision de Holt-Winters et la détection des anomalies, mais ne fait pas état des résultats des expérimentations, en dehors de quelques exemples.

L'algorithme est plus sophistiqué que notre modèle EWMA. Il a pour but de modéliser différentes composantes de la série temporelle au prix de l'utilisation de trois moyennes glissantes différentes pour une série. Ce qui peut également s'appliquer au traitement du flux d'alertes. Une cause possible du problème est la modélisation de la composante saisonnière. Holt-Winters utilise une composante saisonnière, tandis que nous avons identifié deux d'entre elles dans de nombreux flux d'alertes, l'une avec une période d'une semaine et l'autre avec une période d'un jour.

Pour obtenir un effet similaire à celui obtenu avec la composante saisonnière, nous divisons les flux en sous-flux en fonction du jour de la semaine et de l'heure du jour. Cette approche a été discutée à la section 3.2.1. Toutefois, avec le modèle EWMA, nous avons eu des problèmes de changements de niveau abrupts. Dans l'`ensemble-1`, l'approche a très bien fonctionné avec ICMP Ping speedera, mais pas avec SNMP request udp, comme

[1] `http://oss.oetiker.ch/rrdtool/`, consulté le 18.07.2006.

illustré à la figure 3,3(a). Considérons le cas d'un sous-flux séparé pour chaque heure du jour. L'augmentation importante de la composante aux alentours du 15 février apparaît tour à tour dans chaque sous-flux et plusieurs anomalies doubles ont été signalées. Avec l'algorithme de Holt-Winters, la composante saisonnière c_t serait probablement proche de zéro dans le cas de SNMP request udp, étant donné que la valeur observée pour la composante saisonnière utilisée pour mettre à jour c_t est $y_t - a_t$, la différence entre la valeur de la série observée et la valeur constante. Par conséquent Holt-Winters pourrait permettre, en même temps, de mieux tenir compte des variations quotidiennes et éviter les problèmes provoqués par les flux avec des composantes constantes et des changements abrupts.

Au moment de développer l'approche EWMA, nous n'étions pas conscients de ce travail. David Plonka l'a mis en évidence lors de RAID 2004, au cours duquel l'article correspondant a été présenté. Néanmois nous étions en train de travailler avec les modèles AR stationnaires. Cette approche est similaire à Holt-Winters algorithme et nous ferons la comparison de ces deux au chapitre suivant. En bref, les techniques que nous présenterons permettent de éliminer les composantes constant, tendance et saissonnière au lieu de les modéliser.

Si modèles de type moyennes glissantes doivent être utilisés, il pourrait être utile d'examiner l'applicabilité de l'algorithme de Holt-Winters au traitement des flux d'alertes. Même si, selon ses propriétés, un algorithme ou modèle semblerait meilleur qu'un autre, ce n'est pas nécessairement le cas dans la pratique. Par exemple, dans [YECV02], les auteurs ont comparé les tests T^2 et χ^2 de Hotelling pour les données multivariables pour la détection d'intrusions à commande système. Le test T^2 de Hotelling est capable de saisir la corrélation entre les variables et détecter des contre-relations et des changements de moyenne. Le test χ^2 détecte uniquement les changements de moyenne et donne toujours de meilleurs résultats, selon les auteurs.

4.3 Expérimentations

Comme nous l'avons vu à la section 4.2, un travail connexe utilise différentes cartes de contrôle EWMA et un autre algorithme basé sur EWMA. Ces méthodes sont utilisées dans des domaines connexes, pour analyser le trafic sur le réseau et, au niveau du capteur, dans la détection d'intrusions.

Dans cette section, nous présenterons deux groupes d'expérimentations pour démontrer comment notre version des cartes de contrôle EWMA peut être utilisée pour modéliser et filtrer les flux d'alertes.

Nous avons utilisé l'`ensemble-1` dans la première phase pour trouver des paramètres adaptés pour le modèle, puis nous les avons validés à l'aide d'un autre ensemble de données, baptisé `ensemble-4`. Nous commencerons par la phase d'apprentissage et décrirons la validation à la section 4.3.1.

4.3.1 Ensemble-1 : Déploiement de la carte de contrôle

Les cinq flux d'alertes de l'`ensemble-1` ont été utilisés pour explorer l'effet des paramètres du modèle défini à la section 4.1.1. Nous avons cherché une combinaison qui pourrait 1) saisir les artéfacts souhaités dans le flux d'alertes et 2) créer une quantité de nouvelles alertes aussi petite que possible. Ne disposant pas d'une définition exacte

d'artéfacts intéressants de la part d'utilisateurs réels, nous avons dû chercher un comportement qui nous semblait valoir la peine d'être examiné plus en profondeur. En plus des paramètres réels, différents critères d'agrégation et le prétraitement des entrées ont été utilisés.

Définition des paramètres de la carte

La largeur des limites de contrôle dans (4.6) a été définie sur trois écarts-types, comme déjà proposé par Roberts [Rob59]. Les valeurs {1, 2, 3, 6} ont été utilisées avant de faire le choix. La mémoire de la carte dépend du facteur de lissage et de la longueur de l'intervalle d'échantillonnage. Les figures 4.1 et 4.2 décrivent l'effet de la longueur de la mémoire sur la tendance et les limites de contrôle, avec un intervalle d'échantillonnage d'une heure et $(1 - \lambda)$ avec des valeurs 0.8 et 0.99407[2], respectivement. Un facteur de lissage plus petit a pour résultat que la tendance et les limites de contrôle suivent la valeur actuelle de près. Le décalage entre la tendance et la réalité est faible dans la figure 4.1 et les limites de contrôle se resserrent relativement vite après un changement abrupt dans l'intensité du flux. Le comportement du modèle avec un facteur de lissage nettement plus important à la figure 4.2 montre que les valeurs récentes ont relativement peu d'effet sur la tendance. L'écart-type atteint des valeurs tellement importantes que les limites de contrôle absorbent toutes les variations du flux. Pour $(1 - \lambda)$ dans la plage $[0.2, 0.8]$, le taux de balisage a augmenté vers des facteurs de lissage plus petits, la raideur de l'augmentation variant d'un flux à l'autre.

Toutefois, de manière surprenante, la longueur de l'intervalle d'échantillonnage a eu peu d'effet sur la proportion des intervalles et alertes considérés comme anormaux. Ceci s'applique également à l'intervalle de lissage, exception faite des valeurs très extrêmes. Les figures 4.3 et 4.4 montrent la proportion des alertes anormales pour deux signatures déclenchées par des messages d'écho ICMP en fonction du facteur de lissage, où $(1 - \lambda) \in [0.8, 0.99407]$ et des intervalles d'échantillonnage {0.5, 1, 2, 4} heures. Pour les deux, la proportion des alertes balisées comme anormales est dans la plage de quatre pour cent, sauf avec les facteurs de lissage les plus importants. A la figure 4.3, le balisage d'alertes augmente avec les facteurs de lissage les plus importants, un phénomène qui a été causé par la grande différence dans la tendance et la valeur actuelle en raison de la tendance au décalage. A la figure 4.4, le balisage chute abruptement lorsque le facteur de lissage augmente. L'effet inverse était généralement lié aux limites de contrôle larges, avec pour conséquence que le comportement du flux était intégralement considéré comme normal. Un exemple de ce type de situation est visible dans la moitié droite de la figure 4.2.

Définir l'intervalle d'échantillonnage sur une heure et utiliser les facteurs de lissage 0.8 et 0.92 ont permis de baliser ces types d'anomalies constatées dans les flux d'alertes et considérées comme intéressantes également par l'exploration visuelle et comme illustré à la figure 3.3. Comme susmentionné, la longueur de l'intervalle d'échantillonnage a semblé n'avoir qu'un effet mineur sur le taux de balisage. De plus, selon (4.5), cela donne au modèle une mémoire de 9 et 24 heures, respectivement. Pour l'utilisateur, cela offre une association intuitive avec le jour ouvrable et le jour, ce qui est également un aspect important.

Une heure entre l'événement et la notification est une longue période en termes de détection d'intrusions. Mais n'oublions pas que ce dont nous besoin, c'est de récapituler fortement le bruit de fond plutôt que la détection temps réel d'un compromis.

[2]Avec (4.5) le valeur 0.99407 correspond à 336 observations effectifs. Avec l'intervalle d'observation d'une heure, les observations effectifs les plus vieux sont deux semaines antérieurs à l'instant courant t.

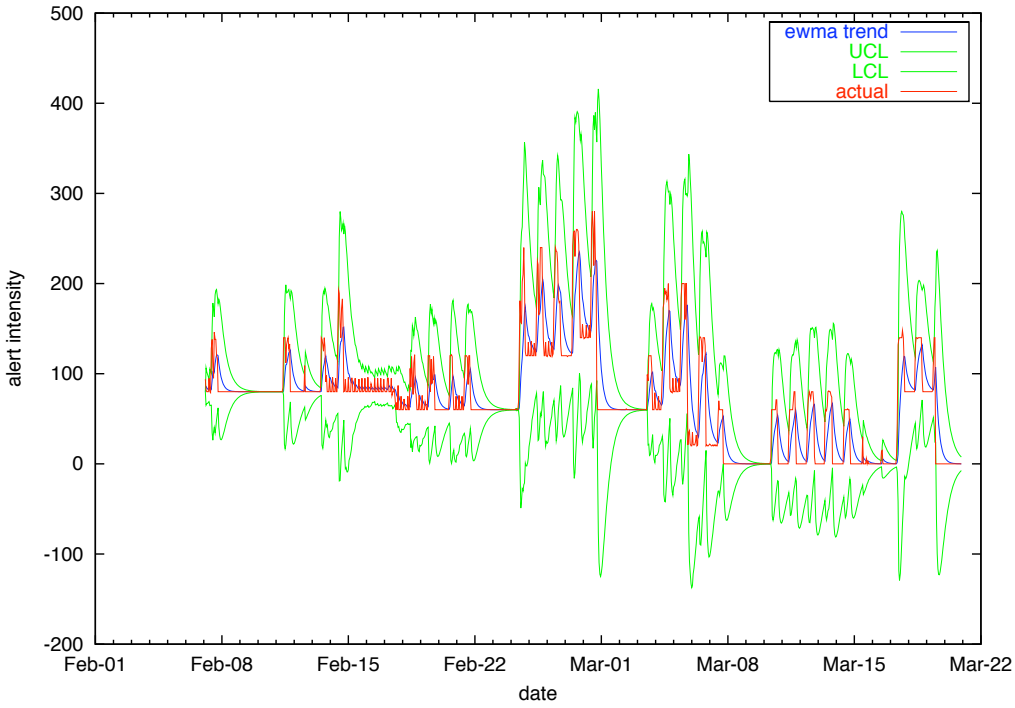


FIG. 4.1 – Effet d'un petit facteur de lissage sur la tendance et les limites de contrôle

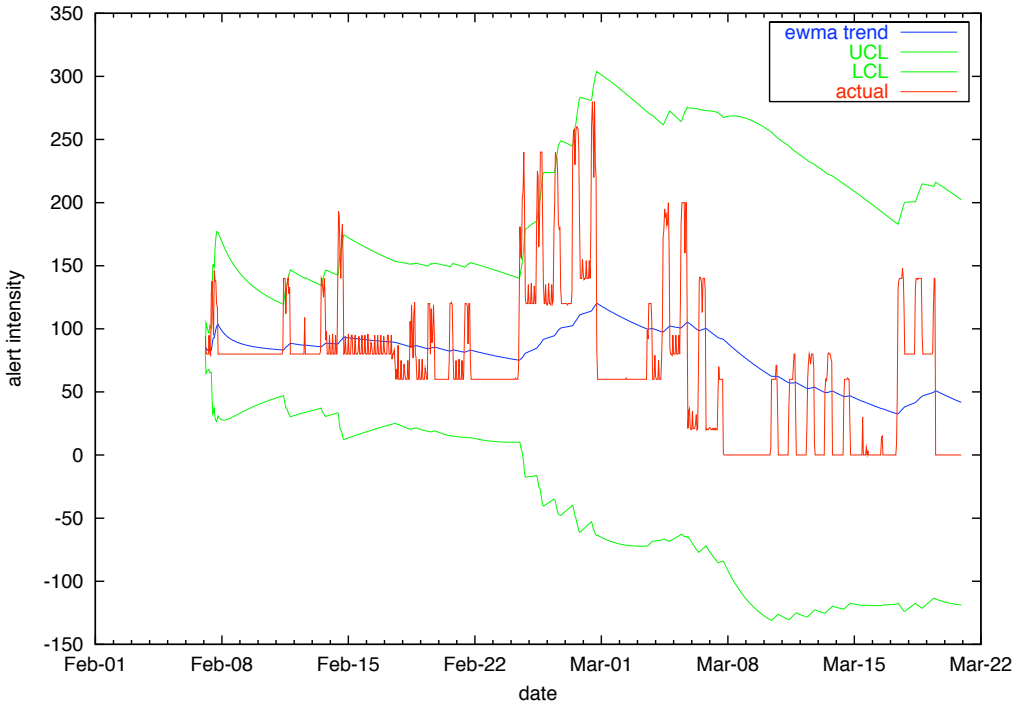


FIG. 4.2 – Effet d'un gros facteur de lissage sur la tendance et les limites de contrôle

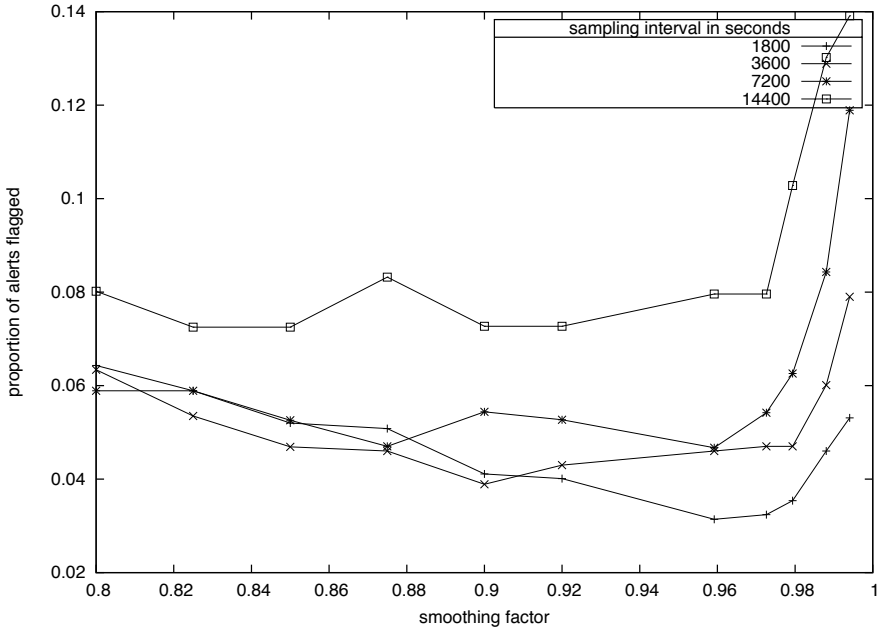


FIG. 4.3 – Effet de la longueur de l'intervalle d'échantillonnage et du facteur de lissage sur la réduction des alertes dans le flux `ICMP PING WhatsupGold Windows`

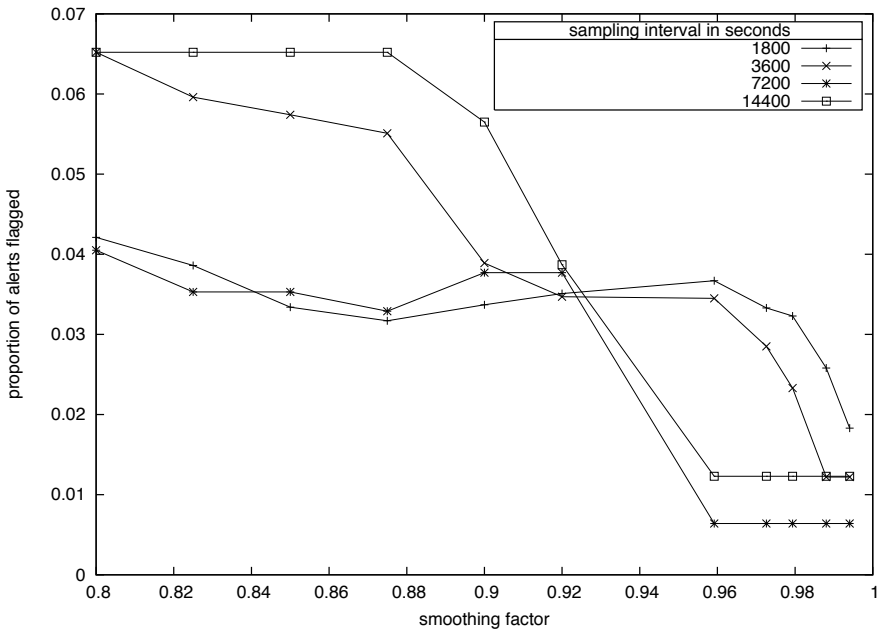


FIG. 4.4 – Effet de la longueur de l'intervalle d'échantillonnage et du facteur de lissage sur la réduction des alertes dans le flux `ICMP PING speedera`

Pour saisir le comportement lié au temps visible pour quelques signatures, deux autres modèles ont été déployés en plus de celui utilisé pour surveiller le flux d'alertes de manière continue. Le deuxième modèle, désigné ci-après modèle *quotidien*, utilise une statistique séparée pour l'intensité de chaque heure du jour. Le troisième modèle, que nous appelons modèle *hebdomadaire*, tient des statistiques séparées pour les intensités de la semaine et du week-end.

Critères d'agrégation

Nous avons également combiné différentes signatures comme un flux. Par exemple, Snort a plusieurs signatures pour différents messages ICMP Destination Unreachable et pour le trafic Web, utilisés pour former respectivement deux agrégats. Dans l'`ensemble-1`, 232 signatures liées au Web ont généré des alertes et seules les signatures Destination Unreachable réagissant aux événements Communication Administratively (SID Snort 485, 486, 487) ont déclenché des alertes. Ces combinaisons n'avaient aucun sens, étant donné qu'il s'agissait d'alertes provenant de quelques signatures dominant les agrégats, qui ont par la suite reflété uniquement le comportement de ces alertes. Seules des signatures séparées et des classes de signatures ont été choisies pour subir un examen plus approfondi. Toutefois, les flux d'agrégats ont pu être formés de nombreuses manières, comme discuté à la section 3.2.1.

Traitement des entrées

Plutôt que d'utiliser la valeur mesurée de l'intensité d'événement comme x_i dans (4.1), une opération de lissage supplémentaire avec un petit facteur de lissage est effectuée dans [YVC03, YBC02]. Nous avons réalisé l'expérimentation avec des facteurs de lissage 0.2 et 0.3 pour l'entrée des tendances, mais l'effet sur la réduction des alertes s'est révélé négligeable. En introduisant l'intensité brute dans les calculs de la tendance, σ_z atteint des valeurs plus élevées plus rapidement. Cela signifie également que les limites de contrôle s'élargissent rapidement, ce qui aide à éviter de baliser plusieurs intervalles après un changement abrupt dans le niveau de l'intensité. Par exemple, à la figure 3.3(a), lorsque l'intensité augmente aux alentours du 15 février, la ligne de tendance se décale sous la valeur réelle pour un moment. Si les limites de contrôle ne sont pas suffisamment larges, plusieurs intervalles sont balisés au lieu de seulement celui contenant le changement et le pic.

Les valeurs de coupure pour y_t basées sur $z_{t-1} \pm m\sigma_{z_{t-1}}$ pour limiter les mises à jour de la tendance ont également été utilisées. Cela signifie que même si l'observation réelle était en dehors des limites de coupure, y_t utilisé pour mettre à jour la valeur EWMA z_t dans (4.1) était limité à $z_{t-1} \pm m\sigma_{z_{t-1}}$. Cela n'a pas bien fonctionné, et a provoqué des problèmes, en particulier avec les flux stables. Lorsque σ_t a approché de zéro, la tendance est devenue trop lente pour s'adapter à des changements drastiques. De nouveau, l'exemple susmentionné avec SNMP request udp s'applique.

Afin de valider le choix des paramètres et constater l'adaptabilité à notre approche, nous avons procédé à une expérimentation avec un ensemble de données plus large.

4.3.2 Validation avec l'`ensemble-4`

Pour valider les paramètres fixés avec les flux de l'`ensemble-1`, nous avons utilisé une base de données d'alertes plus large provenant du même système que nous appelons

`ensemble-4`. Elle contient environ 2M d'alertes générées par 1836 signatures pendant 112 jours.

Dans cette section, nous décrirons la métrique de test utilisée et les résultats des expérimentations de validation. Nous analyserons la manière dont le volume du flux affecte la récapitulation et identifierons les raisons de sa mauvaise efficacité. Nous examinerons ensuite les types d'alertes qui provoquent de grands nombres d'alertes, l'impact du choix de la tranche de temps et d'agrégats plus larges sur le comportement du flux, ainsi que la stabilité des profils du flux. Enfin, nous évaluerons l'utilité générale de notre méthode de récapitulation des alertes.

Métrique de test

Comme mentionné par Mell et al. [MHL+03], le test des IDS n'est pas une tâche simple et manque de méthodologie rigoureuse. Dans notre situation, la difficulté provient du fait que notre intention est de signaler à l'utilisateur que quelque chose d'anormal se passe au niveau du bruit de fond et l'aider à traiter les flux d'alertes en réduisant le nombre d'alertes signalées. En l'état, nous ne sommes pas en mesure d'effectuer une séparation stricte entre les alarmes correctes et les fausses alarmes et il est difficile d'utiliser des métriques comme la précision ou l'exhaustivité [DDW99]. Il est également essentiel de connaître la vraie nature des données de test, par exemple pour la métrique proposée dans [GFD+06]. En fin de compte, c'est l'utilisateur qui décide si oui ou non les informations extraites sont utiles.

Nous pouvons toutefois décrire les capacités de récapitulation à l'aide des deux métriques, 1) la proportion d'alertes balisées comme anormales et 2) la proportion de tranches de temps occupées. Etant donné que la carte de contrôle ne signale que l'anormalité d'un intervalle, nous comptons toutes les alertes à partir d'un intervalle anormal à baliser, ce qui donne une approximation grossière de la réduction des alertes individuelles. Etant donné que notre intention est de récapituler le bruit de fond, il est peu probable que l'opérateur passe en revue toutes les alertes individuelles, sauf peut-être à partir d'un intervalle anormal. Pour les besoins de la discussion, supposons qu'il utilise une unité de temps $t_{controle}$ pour passer en revue le bruit de fond brut généré par un flux pendant un intervalle de longueur T pour détecter qu'il semble normal avec T égal à l'intervalle d'échantillonnage t_s de la surveillance EWMA. Dans notre cas, T est une heure et il est probable que $t_{controle} << T$. Si l'opérateur utilise la surveillance EWMA pour le flux, il ne sera averti que lorsque le flux se comportera de manière anormale. Maintenant, les unités de temps $t_{controle}$ qui seraient utilisées pour détecter manuellement le comportement normal peuvent être utilisées pour des tâches plus utiles, comme l'investigation et la réaction aux alertes plus graves. Par conséquent, il est plus intéressant d'examiner le nombre d'intervalles anormaux après la récapitulation par rapport aux intervalles présentant une activité dans le flux brut plutôt que juste la réduction des alertes.

La proportion de tranches de temps occupées s'obtient en divisant le nombre d'intervalles anormaux par le nombre d'intervalles présentant une activité non nulle pour le flux. La proportion indique la constance avec laquelle l'utilisateur doit se préoccuper du flux avec la surveillance EWMA en comparaison avec le contrôle manuel du bruit accumulé tous les T. Les petites valeurs de flux correspondent à une nuisance moins importante pour l'utilisateur, tandis que des grosses valeurs indiquent que la surveillance EWMA n'est pas capable de récapituler l'activité de ce flux.

Etant donné qu'une anomalie peut être provoquée par un intervalle comportant zéro

alerte, la proportion pourrait en théorie être supérieure à l'unité. Par exemple, imaginons un flux avec un profil de type train d'impulsions, tel que LOCAL-POLICY à la figure 3.3(d) pour lequel tous les intervalles actifs plus quelques intervalles à intensité nulle pourraient être balisés. Toutefois, dans la pratique, nous n'en avons jamais vu. Pour les modèles quotidiens et hebdomadaires, nous combinons les résultats des statistiques de tranches de temps individuelles pour obtenir la performance globale. Un désavantage de ces métriques est qu'il n'y a aucun coût associé à eux, même si en ne s'intéressant qu'aux alertes d'anormalité du flux au lieu des alertes individuelles, des informations peuvent se perdre.

Pour chaque flux, ces mesures s'effectuent avec les trois modèles, continu, quotidien et hebdomadaire, avec deux facteurs de lissage différents, donnant six statistiques par flux.

Effet du volume du flux

Comme nous nous intéressons à la surveillance des agrégats de grands volumes, nous n'avons considéré que les signatures qui ont créé plus de 100 alertes dans l'`ensemble-4`. Après cette présélection, il nous restait 85 signatures.

Les tableaux 4.1 et 4.2 décrivent respectivement la réduction sous forme de pourcentage par rapport aux intervalles non nuls et les anomalies balisées comme anormales pour les flux de plus de 10K d'alertes. La réduction est illustrée avec des facteurs de lissage 0.80 et 0.92 pour chacun des trois modèles, continu, horaire et hebdomadaire. Le tableau 4.1 indique également le nombre total d'intervalles actifs et le tableau 4.2, le nombre total d'alertes pour chaque flux.

A en juger par la réduction de l'intervalle occupé, la méthode est particulièrement utile pour des flux d'alertes qui ont créé plus de 10K d'alertes, l'efficacité augmentant avec le volume du flux. La réduction de l'intervalle occupé pour les flux de moins de 10K d'alertes est déjà plus modeste, et en dessous de 1K d'alertes, la réduction est relativement négligeable.

Le tableau 4.3 résume les résultats de la réduction des alertes avec un modèle continu et un facteur de lissage 0.92. Les 85 flux sont groupés en quatre classes, en fonction de leur volume de sorties (plus de 100, 1K, 10 ou 100K d'alertes) et de la réduction réalisée dans des intervalles occupés et les alertes (moins de 5 %, 10 %, 50 %, 100 % de la valeur d'origine), respectivement. Ces résultats montrent également la performance plus médiocre pour les flux inférieurs à la limite de 10K. Les intervalles occupés présentent une relation plus cohérente entre le volume et la réduction. Le côté droit du tableau 4.3 dans la classe de plus de 100K d'alertes, `ICMP Destination Unreachable (Comm Administratively Prohibited)` ressort, avec une réduction significativement inférieure aux autres de la même classe. Nous avons trouvé deux explications à ce comportement. Premièrement, une grande impulsion d'environ 17K d'alertes a été balisée dans les données de test ; ce qui constitue environ 10 % des alertes balisées. Deuxièmement, la nature du flux est plus aléatoire comparée aux autres, ce qui est visible à la figure 3.3(c) pour les données d'apprentissage et s'applique également à l'ensemble de données plus large. Ce caractère aléatoire entraîne une augmentation du balisage des alertes, mais la réduction dans les intervalles occupés reste comparable aux autres flux de cette classe de volume.

Motifs de la mauvaise récapitulation

Il semble y avoir deux raisons à ces moins bonnes performances. 1) De nombreux flux ont présenté quelques pics d'alertes énormes augmentant significativement le balisage des

TAB. 4.1 – Pourcentage d'intervalles actifs balisés avec différents modèles et facteurs de lissage

		cont.		quotidien		hebdo.	
flux	**int.**	.80	.92	.80	.92	.80	.92
Known DDOS Stacheldraht infection	563	1.6	1.8	8.9	8.5	2.0	2.5
SNMP request udp	2311	4.3	2.9	5.8	4.6	4.2	3.0
ICMP PING WhatsupGold Windows	2069	5.1	3.3	5.8	2.6	5.1	3.2
DDOS Stacheldraht agent–>handler (skillz)	512	1.2	1.6	12	16	1.8	2.1
ICMP Dst Unr (Comm Adm Proh)	2578	5.4	3.5	6.7	5.8	5.4	3.4
ICMP PING speedera	2456	3.3	1.7	4.2	2.9	3.3	0.9
WEB-IIS view source via translate header	2548	5.2	3.8	6.4	5.7	5.1	4.0
WEB-PHP content-disposition	2287	6.8	4.3	7.7	5.2	6.7	4.0
SQL Sapphire Worm (incoming)	1721	2.2	1.2	4.9	3.5	2.4	1.6
(spp_rpc_decode) Frag RPC Records	421	13	7.8	20	20	12	9.0
(spp_rpc_decode) Incompl RPC segment	276	21	13	27	27	22	13
BAD TRAFFIC bad frag bits	432	34	23	37	33	35	22
LOCAL-WEB-IIS Nimda.A attempt	537	24	16	30	25	24	16
LOCAL-WEB-IIS CodeRed II attempt	1229	6.3	4.6	14	14	6.9	5.3
DNS zone transfer	855	9.7	6.7	13	10	9.8	6.5
ICMP L3retriever Ping	107	29	26	71	70	28	23
WEB-MISC http directory traversal	708	12	9.3	15	13	12	9.5
(spp_stream4)STLTH ACT(SYN FIN scan)	29	65	58	82	79	62	62

TAB. 4.2 – Pourcentage d'alertes balisées avec différents modèles et facteurs de lissage

		cont.		quotidien		hebdo.	
flux	**alerts**	.80	.92	.80	.92	.80	.92
Known DDOS Stacheldraht infection	308548	1.2	1.2	4.4	8.4	1.4	1.5
SNMP request udp	303201	4.4	3.0	4.9	4.4	4.2	3.2
ICMP PING WhatsupGold Windows	297437	5.4	4.0	4.5	2.9	5.2	3.1
DDOS Stacheldraht agent–>handler (skillz)	280685	0.8	1.0	7.3	7.0	1.2	1.2
ICMP Dst Unr (Comm Adm Proh)	183020	32	28	39	37	32	28
ICMP PING speedera	95850	5.5	3.1	2.5	2.3	5.3	1.4
WEB-IIS view source via translate header	58600	25	21	12	11	24	22
WEB-PHP content-disposition	48423	18	14	15	13	18	14
SQL Sapphire Worm (incoming)	38905	3.0	1.9	11	9.1	3.1	2.5
(spp_rpc_decode) Frag RPC Records	38804	63	62	94	93	63	62
(spp_rpc_decode) Incompl RPC segment	28715	64	62	93	93	64	62
BAD TRAFFIC bad frag bits	27203	51	42	57	54	53	42
LOCAL-WEB-IIS Nimda.A attempt	25038	65	61	69	64	64	62
LOCAL-WEB-IIS CodeRed II attempt	20418	11	7.5	17	22	11	7.1
DNS zone transfer	15575	32	35	55	55	32	36
ICMP L3retriever Ping	12908	11	12	90	90	11	12
WEB-MISC http directory traversal	10620	41	38	46	45	41	38
(spp_stream4)STLTH ACT(SYN FIN scan)	10182	96	90	93	93	96	96

TAB. 4.3 – Les 85 flux groupés par nombre d'alertes créées et le pourcentage en dessous duquel les intervalles occupés ou les alertes ont été balisés. Résultats pour modèle continu et $(1 - \lambda) = 0.92$

réduction de l'interval occupé

alertes	**5 %**	**10 %**	**50 %**	**100 %**
> 100 K	5	0	0	0
> 10 K	5	3	4	1
> 1 K	0	4	19	7
> 100	0	1	12	24
somme	10	8	35	32

réduction des alertes

alertes	**5 %**	**10 %**	**50 %**	**100 %**
> 100 K	4	0	1	0
> 10 K	2	1	6	4
> 1 K	0	1	15	14
> 100	0	0	8	29
somme	6	2	30	47

alertes. 2) Le profil d'intensité a la forme d'un train d'impulsions qui a un impact négatif à la fois sur la réduction des alertes et sur les intervalles occupés. Etant donné que la première cause n'augmente pas de manière significative le nombre d'intervalles anormaux signalés, c'est-à-dire le nombre de fois que l'utilisateur est dérangé, il s'agit d'un problème moins important. Toutefois, la deuxième cause rend notre approche plutôt impraticable pour la surveillance d'un flux, étant donné que l'opérateur est averti à la plupart des intervalles qui présentent une activité. Le flux `(spp_stream4) STEALTH ACTIVITY (SYN FIN scan)` detection à la dernière ligne des tableaux 4.1 et 4.2 est un exemple typique, son profil d'alerte se composant uniquement d'impulsions. Dans une telle situation, une grande majorité d'intervalles actifs sont balisés comme anormaux. Une analyse plus approfondie des impulsions d'alertes a révélé qu'elles étaient généralement générées dans un intervalle de temps tellement court que l'augmentation de la fréquence d'échantillonnage ne serait pas d'une grande aide. D'autres moyens devraient au contraire être envisagés pour les traiter.

Types d'alertes et omniprésence

Dans l'`ensemble-4`, la proportion d'alertes informatives est importante. Des signatures liées à ICMP et SNMP reflètent l'utilisation normale du système, `DNZ zone transfer` qui se déclenche éventuellement à la collecte d'informations ou au fonctionnement normal du système. Des alertes provoquées par une activité intrusive sont liées à l'utilisation d'un outil DDoS et l'activité d'un ver, représentés par cinq signatures. Les deux signatures DDoS sont en fait les mêmes, différents noms ont été utilisés par l'opérateur pour des raisons de gestion des alertes.

Des alertes se sont également déclenchées suite à des anomalies de protocole. Les flux `(spp_rcp_decode) Incomplete RPC segment`, `(spp_rpc_decode) Fragmented RPC Records`, `(spp_stream4) STEALTH ACTIVITY (SYN FIN scan)` detection sont générés par les préprocesseurs de Snort, et `BAD-TRAFFIC bad traffic bits` est généré par une signature. Nous les examinerons tous, à l'exception de `(spp_stream4) STEALTH ACTIVITY (SYN FIN scan)` detection, constitué de positifs non pertinents et provoqué par une surveillance informative, étant donné que les alertes sont déclenchées suite à des anomalies de protocole et non un comportement intrusif en soi.

Le tableau 4.4 illustre les flux, classés par ordre d'omniprésence dans l'ensemble d'alertes. Le nombre d'intervalles actifs est illustré dans la colonne *actif* avec le pourcentage de tous les intervalles dans l'ensemble de données dans la colonne *présent*. Nous avons be-

TAB. 4.4 – Omniprésence des flux et leurs types. Présence mesurée pendant les intervalles actifs. Les types sont *aw* pour l'activité informative et *intr* pour l'activité hostile

signature	**type**	$< 5\,\%$	**actif**	**présent**
ICMP Dst Unr (Comm Adm Proh)	aw	ok	2578	95 %
WEB-IIS view source via translate header	aw	ok	2548	93 %
ICMP PING speedera	aw	ok	2456	90 %
SNMP request udp	aw	ok	2311	85 %
WEB-PHP content-disposition	aw	ok	2287	84 %
ICMP PING WhatsupGold Windows	aw	ok	2069	76 %
SQL Sapphire Worm (incoming)	intr	ok	1721	63 %
LOCAL-WEB-IIS CodeRed II attempt	intr	ok	1229	45 %
DNS zone transfer	aw	no	855	31 %
WEB-MISC http directory traversal	aw	no	708	26 %
Known DDOS Stacheldraht infection	intr	ok	563	20 %
LOCAL-WEB-IIS Nimda.A attempt	intr	no	537	19 %
DDOS Stacheldraht agent–>handler (skillz)	intr	ok	512	18 %
BAD TRAFFIC bad frag bits	aw	no	432	15 %
(spp_rpc_decode) Frag RPC Records	aw	no	421	15 %
(spp_rpc_decode) Incompl RPC segment	aw	no	276	10 %
ICMP L3retriever Ping	aw	no	107	3 %
(spp_stream4)STLTH ACT(SYN FIN scan)	aw	no	29	1 %

soin d'une réduction d'intervalles actifs à moins de 5 % pour considérer la surveillance EWMA applicable pour un flux. Les résultats sont affichés à la colonne $< 5\,\%$ pour le modèle continu et $(1 - \lambda) = 0.92$. La colonne *type* montre la division entre alertes intrusives (int.) et alertes informatives (aw). La division dépend de la mission de l'opérateur. Par exemple, `ICMP L3retriever Ping` et `(spp_stream4) STEALTH ACTIVITY (SYN FIN scan) detection` ne sont présents que très occasionnellement et correspondent à une reconnaissance probablement réelle, et pourraient par exemple être classés comme étant provoqués par une activité intuitive. `WEB-IIS view source via translate header` réagit à un mot clé dans une adresse URL que IIS ne peut pas traiter en retournant le code source du script demandé. Toutefois, certaines applications Microsoft, comme Outlook Web Access, sont connues pour déclencher de telles alertes[3]. Etant donné le nombre important et la nature omniprésente du flux, nous avons considéré ces alertes comme informatives, déclenchées par une application légitime. Dans un autre contexte, les alertes pourraient être provoquées clairement par un comportement intrusif. Dans l'ensemble, ces observations correspondent à celles présentées au chapitre 3 : un grand nombre d'alerte sont déclenchées par l'utilisation normale du système et ces alertes sont générées de manière continue.

Nous pouvons constater que, pour toutes les signatures présentant une activité à plus de 45 % des intervalles, le nombre d'alertes émises par l'opérateur peut être significativement réduit dans ce système. Il semble que l'omniprésence des signatures soit un critère possible pour déterminer si oui ou non la surveillance EWMA serait utile. Par exemple, la cause des alertes est un indicateur moins cohérent, étant donné que des flux non adaptés à la surveillance EWMA sont provoqués à la fois par des signatures intrusives et des signatures informatives.

[3] `http://www.snort.org/pub-bin/sigs.cgi?sid=1425`, consulté le 18.07.2006.

Comparaison des trois différents modèles

Selon ces matrices, l'utilité des modèles quotidiens et hebdomadaires s'est limitée à quelques exceptions, le modèle continu donnant d'aussi bons résultats que les autres. Le flux `ICMP PING speedera`, avec des variations hebdomadaires et quotidiennes stables, est une de ces exceptions. Toutefois, les métriques sont limitées pour ce type de comparaisons. Il est particulièrement difficile de dire si l'approche horaire balise simplement plus d'intervalles comme anormaux ou si la raison d'artéfacts intéressants est différente. A de nombreuses occasions, la réduction plus petite était due – du moins en partie – à des changements abrupts d'intensité. Plusieurs statistiques différentes constituant le modèle horaire signalent une anomalie, tandis que la statistique mise à jour en continu ne le fait qu'une seule fois. Les deux flux DDoS présentaient des profils d'intensité similaires à une fonction échelon, avec pour conséquence que le modèle horaire balisait significativement plus d'alertes que le modèle continu. Un autre facteur gêne les comparaisons, à savoir les différences de longueurs efficaces de mémoires de modèles. Alors que les statistiques de tranches horaires des modèles horaires et hebdomadaires ne sont mises à jour qu'avec les mesures d'intensité correspondantes, les valeurs moyennes ont une plus longue envergure en temps réel. Par exemple, les statistiques du modèle horaire sont affectées par des mesures datant de 8 ou 24 *jours*.

Classes de flux

L'agrégation de classes de signatures a augmenté le pourcentage de balisage. Le tableau 4.5 montre les réductions obtenues avec le modèle continu et $1 - \lambda = 0.92$ pour les agrégats de classes de plus de 1000 alertes. En fait, presque chaque classe contient une ou plusieurs signatures volumineuses posant déjà en soi des problèmes en matière de statistiques, et cela affecte également l'agrégat de classes. L'augmentation du balisage pourrait également indiquer que des anomalies de flux basées sur la signature de plus petit volume sont détectées jusqu'à un certain degré. Les niveaux d'intervalles occupés sont relativement bien réduits et, de nouveau, le balisage augmente de manière générale au fur et à mesure que le volume des alertes diminue. L'agrégation par classes peut être utilisée pour obtenir une abstraction encore plus grande et des récapitulatifs de niveaux plus hauts dans des situations saturées d'alertes. Toutefois, il existe probablement de meilleurs critères d'agrégation que les classes d'alertes.

Stabilité des flux

Pour donner une idée de la stabilité des profils de flux, le tableau 4.6 compare la réduction des alertes et des intervalles occupés obtenue pour quatre signatures utilisées lors de la phase d'apprentissage, par rapport à la réduction des données de test. En général, le balisage est légèrement plus élevé dans l'ensemble des données d'entraînement, mais pour ICMP Destination Unreachable (Communication Administratively Prohibited), un nombre significativement plus élevé d'alertes a été balisé comme anormales dans l'ensemble de test. L'importante impulsion d'alertes dans ce flux, mentionné ci-avant, compte pour environ 14 % de cette augmentation dans les données de test. Même si ces alertes ont été supprimées, l'augmentation serait importante. Quoiqu'il en soit, la réduction dans les intervalles occupés est fortement similaire, suggérant que la cause serait des pics plus élevés dans l'ensemble de test. La cinquième signature appliquant une politique locale, présente dans la phase d'apprentissage, n'existait plus dans l'ensemble de données de validation.

TAB. 4.5 – Réduction des alertes et des intervalles occupés lors de l'agrégation selon les classes de signatures. Résultats pour le modèle continu avec $1 - \lambda = 0.92$

	bruts		anormaux	
flux	**int.**	**alertes**	**int.**	**alerts**
misc-activity	2678	618273	1.9 %	8.9 %
class_none	1429	380568	4.8 %	18.3 %
attempted-recon	2635	360613	3.7 %	7.0 %
known-issue	563	308548	1.7 %	1.1 %
web-application-activity	2569	88554	3.3 %	16.3 %
bad-unknown	2559	65883	3.7 %	20.9 %
known-trojan	1511	46014	5.4 %	34.9 %
misc-attack	1727	39070	1.3 %	2.1 %
web-application-attack	1017	9587	9.1 %	40.5 %
attempted-user	272	3694	19.4 %	40.6 %
attempted-dos	361	2782	24.3 %	67.8 %
attempted-admin	444	1760	20.2 %	33.1 %

TAB. 4.6 – Comparaison des résultats obtenus lors des phases d'apprentissage et de test. $(1 - \lambda) = 0.92$

	alertes		intervalles	
flow	**appr.**	**test**	**appr.**	**test**
SNMP request udp	2.7	3.5	2.2	3.5
ICMP PING WhatsupGold Windows	4.6	3.6	2.9	3.6
ICMP Dst Unr (Comm Adm Proh)	12	36	3.2	3.7
ICMP PING speedera	2.8	3.2	1.3	2.0

Cette signature a créé des impulsions d'alertes (voir LOCAL-POLICY à la figure 3.3(d)) et la réduction des alertes s'est révélée marginale dans les données d'apprentissage.

Il semble que, avec les paramètres utilisés, la performance de la réduction reste quasi constante. Cela suggère que, après avoir défini les paramètres en accord avec les besoins de l'opérateur, notre approche est capable de s'adapter à de moindres changements dans le comportement du flux d'alertes, sans ajustement supplémentaire. Au cours au moins de cette période de test, aucun des flux se comportant suffisamment bien à l'origine n'est devenu problématique comme une impulsion, ni vice versa. Les signatures ayant un flux d'alertes constant ou un comportement de type à traitement plus aléatoire, toutes deux possibles pour la méthode, ont conservé leur profil original. Ces observations correspondent à celles signalées à chapitre 3 par rapport à la stabilité du profil.

Applicabilité et efficacité

Pour en conclure avec les résultats, il semble possible d'utiliser cette approche pour récapituler et surveiller les niveaux de bruit de fond de haut volume constatés par IDS. Jusqu'à 95 % des tranches de temps d'une heure affichant une activité dans un tel flux d'alertes peuvent être déchargés de la distraction. Pour les intervalles restants, plutôt qu'un barrage d'alertes, seule une alerte serait émise à la fin de l'intervalle. Etant donné

que les deux ensembles de données proviennent du même système, la généralité de ces observations est plutôt limitée et des tests plus complets seraient nécessaires pour poursuivre la validation.

Si l'utilisateur s'inquiète que l'agrégation au niveau de la signature perd trop de données, il est possible d'utiliser des critères supplémentaires, comme les adresses et/ou ports source et destination pour obtenir des flots d'alertes plus ciblés. La réduction de l'agrégation est susceptible de créer davantage d'intervalles balisés, et il s'agit ici d'un compromis que l'utilisateur doit prendre en compte en fonction de ses besoins et de l'environnement d'exploitation. Il n'a pas été possible de déterminer si la récapitulation masquait des événements importants dans l'ensemble de test, étant donné que nous ne possédons pas des registres des intrusions réellement détectées et des problèmes dans le système surveillé avec lesquels nous pourrions comparer nos résultats.

4.4 Discussion

Les problèmes de l'approche résultent des limitations du modèle EWMA. En termes de techniques d'analyse de séries temporelles, la moyenne glissante, c'est-à-dire le filtrage passe-bas, est un outil très basique. Le modèle n'est en fait pas capable de saisir des comportements complexes, mais avec un facteur de lissage adapté, il s'adapte parfaitement aux changements provoqués par l'utilisation normale du système. Le comportement normal du système, tel qu'il est reflété par les flux d'alertes informatives avec $t_s = 1$ h est en général suffisamment lisse pour notre modèle. Nous utilisons un facteur de lissage relativement petit comparé aux applications classiques dans la maîtrise statistique des procédés. Ceci, conformément à (4.7), nous donne également une estimation de la variance, qui augmente suffisamment rapidement suite à des changements d'intensité importants. Suffisamment rapidement, dans le sens que les limites de contrôle s'élargissent afin de ne pas signaler de trop nombreuses alertes autour d'anomalies significatives ou de changements de ligne de base. Les limites de contrôle, telles qu'elles sont définies par (4.6), sont placées à un multiple de l'écart-type de la moyenne glissante. Par conséquent, lorsque la variance du flux augmente, de plus grands changements d'intensité de flux sont acceptés comme étant normaux. Par exemple, avec le flux ICMP Ping speedera (figure 3.3(e)), le modèle EWMA ne saisit en fait pas les comportements hebdomadaires et quotidiens clairement visibles, il s'adapte juste suffisamment bien à ces rythmes. Dans l'`ensemble-1`, cela a été constaté par le fait que le modèle a signalé des anomalies après chaque week-end de faible intensité, lorsque l'augmentation normale du lundi matin arrivait.

Côté positif, le modèle fonctionne bien avec la plupart des flux volumineux et omniprésents présentant divers comportements. De plus, les mêmes paramètres, λ, n et t_s fonctionnent pour différents flux. Comme nous l'avons vu à la section précédente, le modèle EWMA peut être utilisé pour récapituler très efficacement des flux volumineux et réguliers en termes de réduction des intervalles occupés. Un autre point fort de ce modèle réside dans sa simplicité, laquelle résulte des trois avantages notables suivants : l'algorithme est 1) très facile à appliquer, 2) compréhensible pour les utilisateurs, 3) bon marché en termes de mémoire et de processeur.

L'implémentation ne doit entretenir que deux statistiques données par les équations, une pour la moyenne selon (4.1) et l'autre pour la variance telle que définie dans (4.7). Pour les calculer, nous n'avons besoin que des valeurs lissées précédentes et l'observation actuelle y_t. Ces valeurs permettent de calculer l'écart-type et des limites de contrôle supplémentaires selon (4.8) et (4.6). Dans l'ensemble, l'implémentation se compose de

quelques équations simples et utilise très peu d'états par flux.

Il est facile de décrire le modèle du comportement du flux normal en anglais comme étant la *"smoothed i.e. averaged value of the flow intensity during past 9 or 24 hours"* (la valeur lissée, c'est-à-dire moyenne, de l'intensité du flux des 9 ou 24 dernières heures), et les anomalies comme les *"intensity values"* (valeurs d'intensité) qui sont trop éloignées de cette moyenne. En tant que tel, le modèle, et la signification des graphes tels que ceux représentés aux figures 4.2 et 4.1, est compréhensible.

4.5 Conclusion

Nous avons présenté une méthode de traitement des alertes basée sur les cartes de contrôle EWMA pour récapituler des flux d'alertes volumineux se composant d'alertes informatives. Nous utilisons un modèle de comportement normal du flux basé sur les tendances à court terme dans le flux. Les mesures de faible intensité qui s'écartent trop du modèle sont considérées comme anormales et dignes d'être soumises à un examen plus approfondi et signalées à l'utilisateur.

Nous avons décrit les expérimentations à l'aide d'un ensemble de données pour définir les paramètres du modèle et un autre ensemble de données pour la validation. Selon les résultats obtenus, la méthode d'EWMA peut être utilisée pour mettre en valeur des anomalies dans des flux d'alertes de grand volume présentant un degré suffisant de régularité.

Avec cette approche, nous pouvons rendre les niveaux élevés d'alertes associés à ces flux plus durables, sans désactiver complètement les signatures correspondantes. De plus, la méthode permet le filtrage sur la base du contexte des alertes, du nombre d'alertes similaires proches dans le temps, plutôt que des attributs d'alertes individuelles. Ce type de traitement nous permet de trouver des informations intéressantes dans des alertes qui n'ont aucune pertinence individuellement.

Nous pensons que la méthode pourrait être utilisée telle quelle ou en complément d'autres moyens de corrélation, pour surveiller les alertes considérées comme bruit de fond d'un système opérationnel. Les capacités de diagnostic supplémentaires fournies peuvent être modestes, mais ce qui est plus important, c'est que grâce à la récapitulation, l'opérateur peut gagner du temps pour des tâches plus pertinentes, puisqu'il est informé uniquement des changements significatifs du niveau du bruit. Nous avons proposé une métrique basée sur la proportion des unités de temps libérées du traitement manuel, lors de la surveillance d'un agrégat plutôt que d'un flux d'alertes brut.

Il est préférable de traiter les flux d'alertes qui créent moins d'alertes ou qui ont des exigences strictes au niveau de la rapidité de détection avec d'autres moyens, étant donné que l'intervalle d'échantillonnage est rare et que la méthode n'est pas capable de trouver des tendances utiles à partir d'une petite quantité d'alertes.

Nous avons utilisé les signatures et les classes de signatures comme critères d'agrégation. L'utilisation d'hôtes ou réseaux source et destination pourraient au besoin être une étape vers des flux plus spécifiques. Les approches de regroupement d'alertes, telles que celles proposées par Julisch [Jul03a] pourraient fournir d'autres critères d'agrégation.

Ce travail a été présenté dans [VD04] au RAID 2004[4] et la méthode d'EWMA a été mise en oeuvre et intégrée à une plate-forme de gestion des informations de sécurité à la fois utilisée en usage interne et vendue comme service de sécurité gérée à des clients externes. L'auteur n'a pas procédé à l'implémentation et l'intégration d'une version opérationnelle,

[4] `http://raid04.eurecom.fr/`, consulté le 19.07.2006.

mais il montre que la méthode proposée aborde un problème réel et qu'elle est d'une utilité pratique.

Les points forts et les points faibles du modèle résultent de sa simplicité. Même si le modèle s'adapte à des rythmes hebdomadaires et des rythmes quotidiens dans certains flux, dans les faits, il ne saisit pas ce type de comportement. Pour traiter ce problème, nous avons besoin de modèles plus complexes et le chapitre suivant examinera les techniques d'analyse de séries temporelles classiques, qui nous permettent d'améliorer cet aspect au prix d'algorithmes plus complexes.

Chapitre 5

Modélisation des séries temporelles stationnaires

Nous avons vu précédemment l'utilisation de modèles de tendances pour appréhender le comportement normal dans les flux d'alertes. Les modèles de tendances, même s'ils ont fonctionné dans certains cas, étaient une projection très brute de la réalité. L'idée de base consistait en ce que les tendances à court terme dans le comportement du flux devraient enregistrer le comportement normal et exclure les changements abrupts comme des anomalies. Certains flux ont des rythmes quotidiens et hebdomadaires. Considérons le flux `ICMP PING speedera` de l'`ensemble-1` : pendant les jours ouvrables, l'intensité des alertes augmente le matin, atteint un pic le jour et diminue à nouveau le soir. Un comportement similaire intervient dans les comptages des octets dans des réseaux, par exemple Brutlag [Bru00]. De plus, `ICMP PING speedera` a un rythme hebdomadaire, étant donné que l'intensité des alertes est plus petit les week-ends que les jours de la semaine. Ces types de flux ont également été constatés dans l'`ensemble-3`. Le modèle EWMA ne saisit pas un tel comportement périodique et, de plus, avec les paramètres que nous utilisons, la mémoire du modèle est limitée à 24 heures. Dans ce chapitre, nous examinerons la possibilité d'utiliser certaines méthodes d'analyse de séries temporelles classiques pour surmonter le problème. En d'autres termes, l'objectif est d'améliorer la précision du modèle du comportement normal.

L'idée sous-jacente est venue en analysant le comportement normal et anormal que nous avions constaté dans les flux lors d'expérimentations réalisées au chapitre précédent. Comme le comportement normal est constant, régulier et/ou périodique, nous pourrions peut-être le modéliser à l'aide de méthodes pour des modèles de séries temporelles stationnaires. Même si les composantes périodiques ne sont pas stationnaires, il existe des méthodes pour éliminer ces composantes de la série temporelle analysée, etc. Les anomalies étaient intermittentes, abruptes, c'est-à-dire des phénomènes non stationnaires. L'hypothèse que nous examinerons dans ce chapitre est que les modèles de séries temporelles stationnaires ne saisissent pas les non-stationnarités et que ces modèles pourraient par conséquent être utilisés pour modéliser le comportement normal du flux, le filtrer et mettre en valeur les anomalies non stationnaires aux fins de les analyser plus en détail.

Le reste du chapitre est organisé comme suit. A la section 5.2, nous présentons le travail connexe et articulons notre travail en rapport avec celui-ci. La section 5.1 décrit les méthodes utilisées pour traiter les alertes générées par des signatures prolifiques. Les résultats pratiques sont décrits à la section 5.3 et, enfin, nous proposons nos conclusions

à la section 5.6.

5.1 Méthodes de séries temporelles stationnaires

Dans cette section, nous proposons une méthode pour modéliser et filtrer les flux d'alertes. Pour présenter nos observations de manière plus exacte, nous utiliserons certaines notations et certains concepts du domaine de l'analyse des séries temporelles, principalement selon [BD91, BD02, Tar04]. Il convient de faire remarquer que la théorie et les méthodes sur lesquelles nous nous basons sont classiques et simples, notre intention étant d'établir une correspondance avec le traitement des alertes et avec notre problème, puis de proposer la manière d'appliquer ces méthodes. Etant donné que ces méthodes sont simples, elles ne sont probablement pas les mieux adaptées à tous les flux d'alertes. Toutefois, nous souhaitions commencer par le simple pour voir si cela suffisait. Nous commencerons par une vue d'ensemble de la méthode de traitement des alertes, après quoi nous décrirons chaque étape plus en détail.

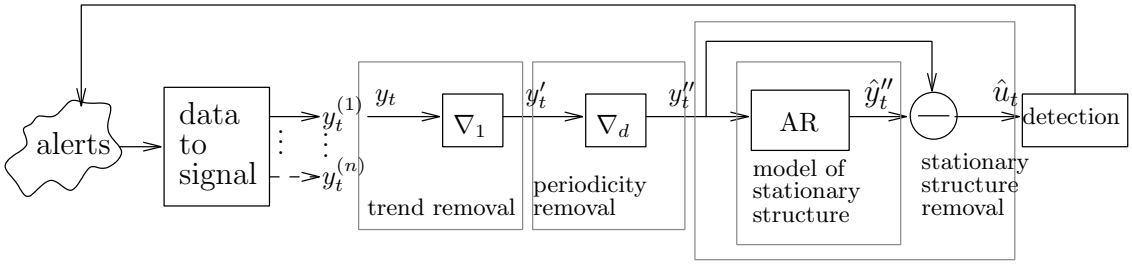


FIG. 5.1 – Schéma du processus de détection

5.1.1 Vue d'ensemble

Un flux d'alertes est un flot d'alertes successives qui répondent aux critères d'agrégation. Nous agrégeons les alertes selon la signature qui a généré l'alerte et examinons les mesures de l'intensité du flux prises à des intervalles fixes comme une série temporelle y_t. L'objectif est d'éliminer par filtrage de ce flux ou cette série temporelle les composantes qui correspondent à l'activité normale du système d'information.

Un modèle de *série temporelle discrète* pour un ensemble d'observations $\{y_t\}$ est une représentation de distributions à plusieurs variables d'une séquence de variables aléatoires $\{Y_t\}$ dont on postule que $\{y_t\}$ est une réalisation. Chaque y_t est enregistré au temps t, où le T_0 défini des durées d'observations est un ensemble discret. Nous ne considérons ici que les observations effectuées avec des intervalles de temps fixes. En pratique, la spécification de distributions à plusieurs variables est rarement disponible et impossible à estimer, seuls les moments de premier et deuxièmes ordres des distributions à plusieurs variables sont utilisés.

Nous utilisons le modèle suivant pour la série d'alertes y_t

$$y_t = x_t + u_t \ , \tag{5.1}$$

où x_t représente la partie du flux d'alertes provoquée par l'activité normale et u_t, le reste, c'est-à-dire la partie anormale du flux d'alertes. Comme mentionné auparavant,

suite à l'analyse du flux d'alertes, nous considérons que les phénomènes de régularité et de stationnarité dans les séries d'alertes font partie du comportement normal du système. Tandis que nous considérons que les phénomènes intermittents non stationnaires dans la série d'alertes sont provoqués par le comportement anormal du système. Ce comportement anormal peut être des attaques ou des problèmes plus généraux. La distinction entre les alertes liées à l'activité normale et celles liées à l'activité anormale est invisible au niveau de l'alerte, mais peut être constatée au niveau du flux.

Selon le modèle de décomposition classique [BD91], une série temporelle x_t peut être considérée comme contenant trois composantes, *tendancielle*, *périodique*, c'est-à-dire *saisonnière*, et *aléatoire*

$$x_t = l_t + s_t + r_t \ , \tag{5.2}$$

où l_t est une tendance à changement lent, s_t est une composante périodique avec une période connue d, et r_t est une composante stationnaire aléatoire. Il est à noter que r_t n'est pas nécessairement complètement aléatoire au sens commun du terme, mais peut contenir une structure qui ne s'intègre ni dans l_t ni dans s_t. Dans la littérature, r_t est également appelé la composante du bruit, mais pour éviter la confusion avec notre terme *bruit des alertes*, nous préférons utiliser le terme composante aléatoire pour r_t.

Par exemple, l_t correspond à des changements lents dans l'intensité des alertes, provoqués par des changements généraux du trafic surveillé dans le temps. s_t correspond à la périodicité créée dans le flux d'alertes par l'activité périodique du système liée aux heures de travail et aux jours ouvrables. Prenons l'exemple plus spécifique de `ICMP PING speedera` qui possède une forte composante périodique, voir figure 3.3(e). Le comportement moins banal qui est toujours présent et invariable sur une plus longue période correspond à la partie structurée de r_t.

Pour que l'opérateur puisse se concentrer sur les alertes qui permettent un examen plus approfondi, nous éliminons par filtrage les alertes qui se conforment à la description du comportement normal du système, c'est-à-dire les composantes du flux d'alertes qui s'intègrent dans le x_t du (5.2). La composante anormale d'un flux d'alertes est $u_t = y_t - x_t$, dont on obtient une estimation $\hat{u}_t$ après filtrage. Seuls les phénomènes les plus significatifs de $\hat{u}_t$ sont signalés à l'opérateur comme des anomalies.

La figure 5.1 décrit les étapes de ce processus. La transformation de données en séries y_t est dans ce cas banale, étant donné que nous comptons seulement le nombre d'alertes correspondant aux critères d'agrégation dans une unité de temps, une heure. Nous reconnaissons que le choix de l'unité de temps affecte en particulier la rapidité de détection et la visibilité de certains phénomènes dans le flux d'alertes. Etant donné la ressemblance de la nature des alertes surveillées avec le bruit, nous constatons que les mesures horaires sont suffisantes, comme dans le chapitre précédent. Nous considérons également les séries à une variable, même si nous pouvons envisager d'autres transformées créant une série à variables multiples, de données d'alertes en une série.

A la section 5.1.2 et à la section 5.1.3 , nous décrirons comment supprimer l_t et s_t, respectivement, de y_t pour obtenir la première série dont on a retiré la tendance y'_t, puis y''_t duquel la composante périodique a également été supprimée. Maintenant, la série y''_t contient le r_t aléatoire et les composantes u_t anormales. La structure dans r_t est saisie dans un modèle de série temporelle, décrit à la section 5.1.4 et une estimation de $\hat{u}_t$ est obtenue comme la différence entre la sortie du modèle $\hat{y}''_t$ et les observations y''_t. Enfin, la détection des anomalies les plus significatives est détaillée à la section 5.1.5.

5.1.2 Suppression de la tendance

Premièrement, supposons que la tendance l_t éventuellement présente dans le flux d'alertes soit constante. Le raisonnement qui sous-tend cela s'appuie sur le sens commun et sur nos observations. Si la tendance est d'un degré plus élevé, cela signifie qu'il existe un problème grave avec les capteurs et la capacité de stockage des capteurs s'épuisera bientôt. Cela ne peut pas être la situation normale et, si cela arrive, nous ne souhaitons même pas supprimer ces informations de la série.

Il existe différents moyens de supprimer les composantes tendancielles et périodiques de la série temporelle [BD02]. Nous avons choisi d'utiliser l'opérateur de *différenciation décalage-d*, ∇_d, pour deux raisons. Premièrement, il ne nécessite pas que la tendance reste constante au fil du temps et, deuxièmement, il ne nécessite pas d'estimation de plusieurs paramètres. Un avantage supplémentaire est que le même opérateur peut être appliqué aussi bien aux composantes tendancielles que périodiques. Il est défini comme suit :

$$\nabla_d y_t = y_t - y_{t-d} \ . \tag{5.3}$$

En d'autres termes, la série qui en résulte est la différence entre deux observations de la série originale, exception faite des unités de temps d. Avec $d = 1$, il s'agit d'une analogie de l'opérateur d'écart pour les fonctions continues.

Lorsque l'on applique ∇_1 à une tendance constante l_t de forme $l_t = \mathrm{b}t + c$, nous obtenons $\nabla_1 l_t = l_t - l_{t-1} = \mathrm{b}t + \mathrm{c} - (\mathrm{b}\,(t-1)) + \mathrm{c} = \mathrm{b}$, la pente de la fonction tendancielle. Si on la considère comme l'écart, cette étape nous donne une série représentant le taux de changement dans le flux d'alertes. Par exemple, avec `SNMP request udp`, cette étape supprime la composante constante, visible à la figure 3.3(a) et effectue les changements de niveau. Les phénomènes intéressants sont donc plus apparents. La série transformée, c'est-à-dire celle se composant des valeurs y'_t de la figure 5.1 est illustrée à la figure 5.2.

Nous appliquons ∇_1 à toutes les séries. Cela entraînera une perte d'informations contenues dans la série, mais pour la détection des anomalies, la tendance linéaire et la valeur absolue de l'intensité des alertes ne sont pas nécessaires.

5.1.3 Suppression de la périodicité

Comme susmentionné, l'opérateur ∇_d peut également être utilisé pour supprimer la composante périodique de la série temporelle. L'application de ∇_d au modèle x_t de (5.2) sans la composante tendancielle, qui peut être éliminée comme illustré ci-avant, et où s_t a la période d, donne pour résultat

$$\nabla_d x_t = s_t - s_{t-d} + r_t - r_{t-d} = r_t - r_{t-d} \ . \tag{5.4}$$

Après cette opération, il nous reste une composante aléatoire $(r_t - r_{t-d})$. L'application de ∇_d nécessite de connaître la période d. Nous développerons ce point à la section 5.3. Pour le moment, nous supposerons qu'elle est connue.

Etant donné que chaque opération de différenciation supprime l'information de la série temporelle, nous n'appliquons pas ∇_d à toutes les séries. En général, si une série temporelle se compose de iid observations, aucune modélisation ne doit être effectuée. Il existe plusieurs moyens de tester le caractère aléatoire de la série [BD02, LB78] et nous examinerons la fonction d'autocorrélation d'échantillons. Pour définir la fonction d'autocorrélation, nous avons besoin de la fonction d'autocovariance de $\{y_t\}$, définie comme

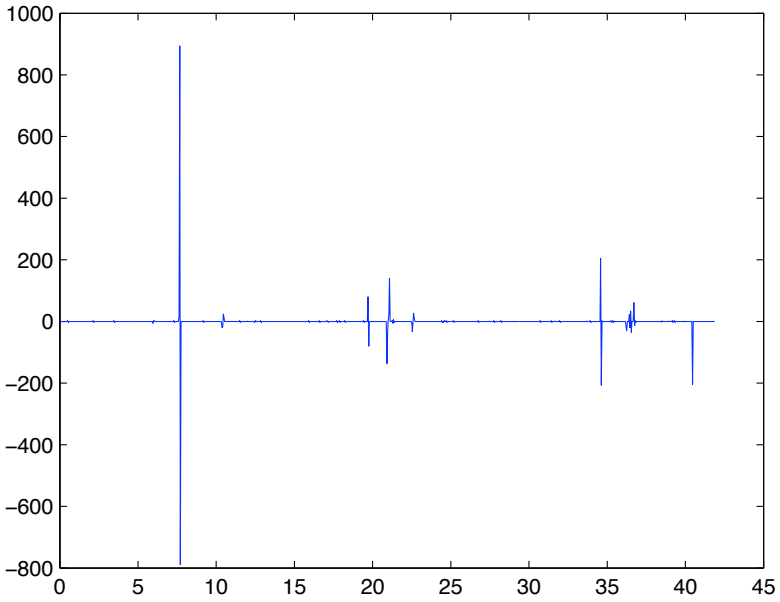


FIG. 5.2 – Flux `SNMP request udp` de l'`ensemble-1` après retrait de tendance. La composante constante a été supprimée, mais les changements de niveau constant restent visibles.

$$\gamma_y(r,s) = Cov(y_r, y_s) = E[(y_r - \mu_y(r))(y_s - \mu_y(s))] \quad , \tag{5.5}$$

où $\mu_y(t) = E(y_t)$. La fonction d'autocorrélation (ACF) de $\{y_t\}$ comme décalage de h est

$$\rho_y(h) = \frac{\gamma_y(h)}{\gamma_y(0)} \quad . \tag{5.6}$$

Dans la pratique, nous commencerons par les données observées plutôt que par la définition du processus et pour évaluer les dépendances dans les données, nous pouvons utiliser la fonction d'autocorrélation des échantillons

$$\hat{\rho}_y(h) = \frac{\hat{\gamma}_y(h)}{\hat{\gamma}_y(0)} \quad . \tag{5.7}$$

La fonction d'autocovariance des échantillons se définit comme suit :

$$\hat{\gamma}(h) = \frac{1}{n}\sum_{t=1}^{n-|h|}\left(y_{t+|h|} - \bar{y}\right)(y_t - \bar{y}) \quad , \tag{5.8}$$

où n est la taille de les échantillons et $\bar{y} = 1/n\sum_{t=1}^{n} y_t$ est la moyenne des échantillons.

Pour une séquence de n idd variables aléatoires dont $y_1, \ldots, y_n$ est une réalisation, environ 95 % des valeurs d'autocorrélation tombent dans les limites de $\pm 1.96/\sqrt{n}$ [BD91, p.222]. En d'autres termes, $\pm 1.96/\sqrt{n}$ sont les 95 % de limites de confiance pour les valeurs d'autocorrélation de iid variables aléatoires. Si la valeur ACF des échantillons pour le décalage d se situe en dehors des 95 % des limites de confiance, nous appliquons ∇_d à la série y'_t pour obtenir y''_t.

5.1.4 Suppression de la structure stationnaire

Pour supprimer la structure stationnaire restante r_t [1] de la série, nous élaborons un modèle de r_t. Une fois le modèle élaboré, nous utilisons la série de laquelle nous avons supprimé la tendance et la périodicité, y_t'', comme entrée dans le modèle. La différence entre la sortie du modèle et $\hat{y_t}''$ et y_t'' est une estimation de la composante anormale $\hat{u}_t$. La structure r_t est saisie dans un modèle de série temporelle AR (p) autorégressif.

La série temporelle r_t, par exemple l'intensité des alertes observée et éventuellement transformée, est appelée *processus* autorégressif d'ordre p, si elle satisfait à l'équation de différenciation :

$$r_t = -\sum_{k=1}^{p} a_k r_{t-k} + e_t \ , \tag{5.9}$$

où $\{et\}$ $\{e_t\} \sim \text{WN}\left(0, \sigma^2\right)$, c'est-à-dire le bruit blanc et a_k, $k = 1, \ldots, p$ sont des constantes. En anglais, cela signifie que la valeur de r_t à l'instant actuel t est une somme de deux termes, la somme pondérée de p observations précédentes de $r_{t-1}, \ldots, r_{t-p}$, et le terme du bruit e_t.

Dans la modélisation AR, les observations r_t sont supposées être un processus AR (p). On estime que les paramètres du modèle $a_1, \ldots, a_p$ minimisent la variance de l'*erreur de prévision* ou le ϵ_t *résiduel*.

$$\epsilon_t = r_t + \sum_{k=1}^{p} a_k r_{t-k} \tag{5.10}$$

dépendant, en quelque sorte, de la méthode d'estimation.

Nous avons également utilisé les modèles ARMA (p, q) de moyenne glissante autorégressive. La série temporelle r_t est réputée être un processus ARMA d'ordre p et q, si elle satisfait à l'équation de différenciation :

$$r_t = -\sum_{k=1}^{p} a_k r_{t-k} + \sum_{j=1}^{q} b_j e_{t-j} + e_t \ , \tag{5.11}$$

où $\{e_t\} \sim \text{WN}\left(0, \sigma^2\right)$, et a_k, $k = 1, \ldots, p$ et $b_j, j = 1, \ldots, q$ sont des constantes. Dans le cas spécial de $q = 0$, le modèle ARMA $(p, 0)$ est réduit à un modèle AR (p). Ici, en plus de la somme pondérée des observations précédentes, une somme pondérée de bruit blanc définit la valeur actuelle de r_t. Toutefois, étant donné que les résultats rapportés à la section 6.4 étaient aussi bons, voire meilleurs que les modèles AR, nous ne tiendrons compte par la suite que des modèles AR.

Le modèle AR dans (5.9) est un modèle paramétrique et nous devons donc 1) choisir le degré de modèle p et 2) estimer les paramètres a_k for $k = 1, \ldots, p$ avant de pouvoir utiliser le modèle. Généralement, le choix de p s'effectue en élaborant différents modèles et en choisissant le « meilleur » en fonction de certains critères. Dans notre cas, il s'agit d'un compromis décent entre les phénomènes intéressants et les phénomènes non intéressants signalés par le modèle à l'aide des données d'estimation. Des métriques comme le critère d'*erreur de prévision finale* (FPE) et le *critère d'information d'Akaike* (AIC) qui tentent

[1] En fait, comme vu au point (5.4), elle est également passée par les transformées et nous avons r_t'', mais pour la simplicité de notation, nous utilisons r_t comme dans [BD02]

d'établir un compromis entre l'adaptation et la complexité du modèle [Tar04, p.44] pourraient être utilisées pour choisir l'ordre des modèles. Elles assument toutefois que la série observée doit être modélisée avec précision. Comme, dans notre cas, la série observée contient également des anomalies et que notre but est de modéliser uniquement le comportement normal, nous ne souhaitons même pas un type parfait. Par conséquent, ces critères ne sont pas applicables directement à notre situation.

L'estimation des paramètres s'effectue à l'aide de l'algorithme des moindres carrés. En résumé, cela signifie que les paramètres choisis minimisent le carré de la différence entre les observations et la prévision réalisée par le modèle[2]. L'estimation des paramètres du modèle AR est un problème linéaire, tandis que pour le modèle ARMA, le problème est non linéaire. C'est une raison de plus pour utiliser les modèles AR plutôt que les modèles ARMA.

Supposons qu'il n'y ait aucune anomalie dans y_t et que le modèle de décomposition ((5.2)) décrive suffisamment le comportement normal de la série temporelle y_t. Dans ce cas, après avoir supprimé la tendance et la périodicité, on pourrait définir la nécessité de poursuivre la modélisation avec les modèles AR en testant la blancheur de la série restante, comme susmentionné.

Si y'' ressemble à un bruit blanc, il n'y a plus de structure à modéliser et cette étape est inutile. En revanche, les données d'alertes réelles contiennent des anomalies et le modèle de l'équation (5.2) n'est pas parfait, ce raisonnement est donc moins faisable. Par conséquent, nous élaborons le modèle AR pour toutes les séries, provoquant éventuellement un surdébit informatique inutile.

Cette étape diffère des précédentes, étant donné que nous utilisons des données d'entraînement pour l'estimation des paramètres. Une fois l'entraînement réalisé et les paramètres estimés, nous pouvons utiliser chaque nouvelle observation y''_t, comme entrée dans le modèle et obtenir $\hat{u}_t$ comme la différence entre l'observation et la sortie du modèle $\hat{y_t}''$.

5.1.5 Détection des anomalies

Après ces étapes, nous avons isolé l'estimation de la composante anormale $\hat{u}_t$ de la série d'alertes. Si la série originale y_t ne contenait aucune anomalie et que notre modèle de comportement normal x_t était exact, la composante anormale estimée serait un bruit blanc, $\hat{u}_t \sim \mathrm{WN}\left(0, \sigma^2\right)$.

Toutefois, dans la réalité, ce n'est jamais le cas. Les anomalies de la série d'alertes et les insuffisances du modèle, à la fois au niveau conceptuel et dans l'estimation des paramètres, signifient que $\hat{u}_t$ n'est pas un bruit blanc. Afin d'éviter de signaler des artéfacts provoqués par les déficiences du modèle et les variations aléatoires, nous choisissons uniquement les changements les plus significatifs de $\hat{u}_t$ avec la même carte de contrôle EWMA que nous avons utilisée. Une anomalie est signalée si la valeur $\hat{u}_t$ actuelle diffère plus de n écarts-types de la moyenne des valeurs passées. La valeur par défaut pour n est trois, mais elle peut être ajustée pour augmenter ou réduire le nombre d'anomalies signalées. La moyenne des valeurs antérieures et l'écart-type sont estimés à l'aide de moyennes glissantes pondérées exponentiellement. En d'autres termes, les limites de contrôle pour $\hat{u}_t$ sont définies selon ((4.6)), c'est-à-dire

[2]Pour une documentation du programme d'estimation, voir `http://www.mathworks.com/access/helpdesk/help/toolbox/ident/arx.html`, consulté le 20.07.2006

$$z_{t-1} \pm n \cdot \sigma_{z_{t-1}} \ , \tag{5.12}$$

où z_t est le EWMA de $\hat{u}_t$ selon ((4.1)) avec $(1 - \lambda) = 0.92$.

Il serait intéressant de connaître l'activité normale en tant que telle, car cela peut aider l'opérateur à mieux comprendre le système surveillé. Toutefois, ce type d'analyse nécessite un certain niveau de compréhension de la théorie sous-jacente de l'opérateur. Pour simplifier les choses, nous ne lui présentons que les phénomènes les plus significatifs de la partie anormale du flux d'alertes.

5.2 Travail connexe

Dans cette section, nous présenterons le travail connexe du point de vue de la méthode avant de passer aux expérimentations à la section 5.3.

A la section 2.2.3, nous avons vu comment Qin et Lee utilisent les modèles AR et ARMA dans le test de causalité de Granger pour trouver les relations entre les séries d'alertes. La différence clé est que Qin et Lee recherchent des relations *entre* deux séries d'alertes, alors que nous modélisons le comportement normal dans les séries d'alertes. De plus, nous nous concentrons sur les alertes informatives plutôt que sur les alertes intrusives.

L'algorithme de Holt-Winters utilisé par Brutlag [Bru00] à la section 4.2.3 utilise le modèle de décomposition classique augmenté de la notion de *niveau*, mais au lieu de supprimer les composantes constantes, tendancielles et périodiques, l'algorithme de Holt-Winters essaie de les estimer. Dans notre cas, le signal les transforme pour supprimer toutes ces composantes, et nous modélisons la structure restante avec des modèles autorégressifs plus sophistiqués.

Barford et al. [BKPR02] analysent les comptages des paquets et octets de flux de réseau. Ils utilisent un algorithme de détection basé sur la variance locale des composantes moyenne et haute fréquence du signal. Nous discuterons de l'intégralité de l'approche au chapitre 6, puisque les méthodes sont plus proches l'une de l'autre. Pour le moment, nous nous concentrerons seulement sur la partie détection. En résumé, leur plate-forme d'analyse fournit des parties basse (L), moyenne (M) et haute (H) fréquence des signaux d'octets, paquets et débit dans le domaine temps.

Les signaux présentent un comportement similaire à celui de certains flux d'alertes et Barford et al. utilisent l'algorithme de détection des anomalies appelé *score des écarts*. Il s'agit d'une procédure en trois étapes :

1. Ils utilisent uniquement les parties H et M des signaux et les normalisent séparément à la variance unitaire. Pour une taille de fenêtre donnée t_1, ils calculent la variabilité locale dans une fenêtre glissante.
2. Les variabilités locales des parties H et M sont combinées à l'aide de la somme pondérée pour obtenir la partie (V) variable du signal, c'est-à-dire le score des écarts.
3. Les scores des écarts 2.0 ou supérieurs sont considérés comme des anomalies à haut degré de confiance et les scores inférieurs à 1.25 sont considérés comme des anomalies à faible degré de confiance.

Nous analysons la série résiduelle $\hat{u}_t$ pour détecter les anomalies plutôt que la série proprement dite. Notre carte de contrôle utilise également un type de variabilité locale via l'estimation d'écart-type fourni par EWMA. La procédure de normalisation signifie que la

TAB. 5.1 – Périodes les plus fortes trouvées par l'algorithme

Flux	Décalages (heures)			
SNMP	-	-	-	-
Whatsup	168	24	23	15
Dest Unr	144	72	48	24
LOCAL-POLICY	168	167	24	24
Speedera	24	12	10	2

détection s'effectue post-mortem, une fois que toute la série est obtenue, tandis que nous testons les observations lorsqu'elles arrivent, en ligne.

Zou et al. [ZGTG05] utilisent le modèle AR (1) stationnaire pour modéliser le comportement du ver. Leur travail est discuté plus en détail à la section 6.2.1. En résumé, le modèle est très simple et son objectif est la détection rapide de la tendance exponentielle, pas la modélisation d'un comportement complexe.

Dans l'ensemble, les modèles AR sont peu utilisés dans la détection d'intrusions et les utilisations existantes sont très différentes de la méthode présentée à la section 5.1. Pour démontrer comment notre méthode fonctionne dans la pratique, la section suivante présente quelques expérimentations.

5.3 Expérimentations

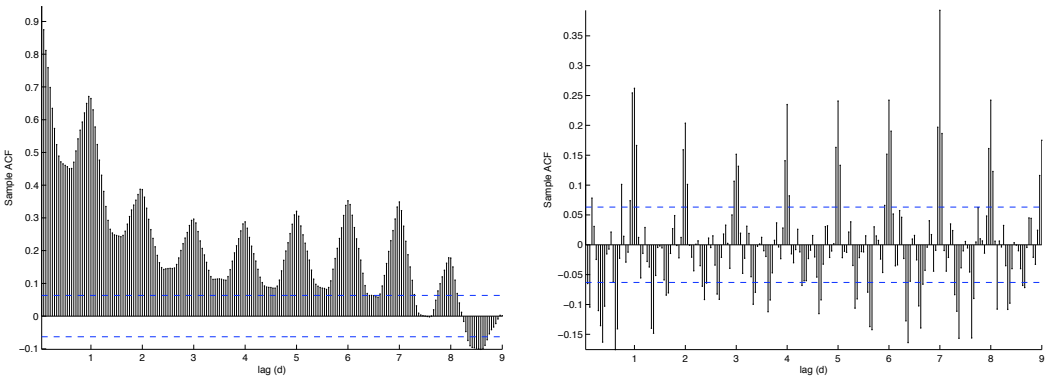
Dans cette section, nous présentons les résultats obtenus lorsque nous appliquons cette méthode aux alertes de l'`ensemble-1` décrites à la section 3.2.1. La première partie du corps des alertes, 406 observations, a été utilisée comme données d'entraînement pour estimer les paramètres de modèles AR. La dernière partie, 600 observations, a été soumise à la validation. L'outil a été appliqué à l'aide de Matlab.

5.4 Périodicité dans le flux d'alertes

Pour supprimer la composante périodique s_t avec une période d à l'aide de ∇_d, nous devons connaître d. L'inspection visuelle des séries d'alertes et des valeurs ACF des échantillons, ainsi que l'intuition ont suggéré des périodes proches d'un jour ou une semaine. La figure 5.3 montre les valeurs ACF des échantillons pour les observations de ICMP PING WhatsupGold Windows jusqu'à un décalage de neuf jours à l'aide du flux original (Figure 5.3), après la suppression de la tendance avec ∇_1 (Figure 5.3(b)), et après la suppression de la périodicité avec ∇_{168} (Figure 5.3(c)). Les traits indiquent un intervalle de confiance de 95 % pour les valeurs ACF des échantillons du bruit blanc.

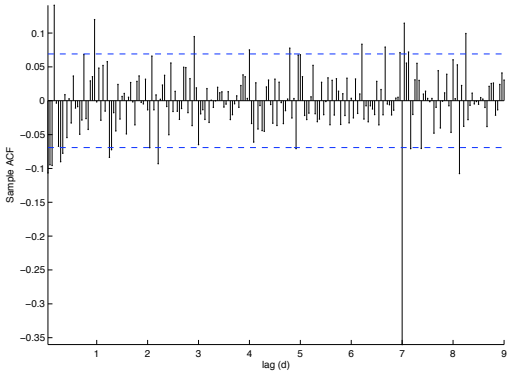
On peut voir qu'il existe plutôt de fortes corrélations positives pour des décalages de un et sept jours, puisque les valeurs ACF des échantillons les plus élevées sont à des décalages correspondant à une semaine et après suppression de la tendance à la figure 5.3(b). On peut également voir que les corrélations aux décalages correspondant à des multiples d'un jour disparaissent après l'application de ∇_{168}.

Nous avons également utilisé un algorithme qui supprime la composante périodique correspondant au décalage de la valeur absolue ACF des échantillons la plus grande. Le tableau 5.1 montre les quatre premiers décalages utilisés par l'algorithme pour la sup-



(a) Original flow y_t

(b) After trend removal y'_t



(c) After periodicity removal y''_t

FIG. 5.3 – Valeurs d'autocorrélation des échantillons pour `ICMP PING WhatsupGold Windows` à différentes étapes du traitement et jusqu'au décalage correspondant à neuf jours

pression de la périodicité. La première, c'est-à-dire la composante périodique la plus forte supprimée, avait une période d'un multiple de 24 heures pour tout sauf `SNMP request udp`, qui n'a aucune composante périodique présente, comme on peut le voir à la figure 3.3(a). Si la série d'alertes a présenté une autocorrélation importante, elle incluait de fortes composantes hebdomadaires et quotidiennes. Cette observation a confirmé l'intuition que nous avons eue concernant la périodicité dans les flux d'alertes.

Plutôt que l'algorithme, nous avons choisi d'utiliser ∇_d uniquement avec d correspondant à une semaine, au moins pour les raisons suivantes : 1) chaque application de ∇_d supprime les informations de la série, 2) provoque la perte des d premières observations, étant donné qu'il n'y a pas suffisamment de données historiques pour appliquer l'opérateur, 3) l'application de ∇_d avec d correspondant à un jour ou une semaine a supprimé la majorité des autocorrélations importantes et 4) les autocorrélations avec le décalage de 168 heures (une semaine) se sont révélées être les plus importantes. Comme expliqué à la section 5.1.3, la valeur ACF des échantillons correspondant à un décalage d'une semaine est utilisée pour déterminer si oui ou non ∇_d est appliqué à un flux.

5.4.1 Choix des degrés du modèle

Nous avons utilisé plusieurs degrés de modèle p, à savoir 4, 10, 16, 26 avec les données d'estimation pour trouver le plus approprié pour chaque flux. En plus des modèles AR, une gamme de modèles ARMA (p, q) a été estimée. Du moins avec les degrés de modèle et les méthodes d'estimation utilisés, ces modèles n'ont pas présenté d'amélioration significative, voire aucune, par rapport aux modèles AR, mais leur estimation consomme plus de ressources.

Le p a été choisi de manière à ce que la méthode détecte autant de phénomènes intéressants que possible, comme décrit à la section 3.2.1, et signale aussi peu de phénomènes inintéressants que possible. Selon les anomalies signalées à partir des données d'estimation, nous avons choisi les degrés de modèle comme suit : `SNMP request udp` AR(4), `ICMP PING WhatsupGold Windows` AR(26), `ICMP Destination Unreachable Communication Administratively Prohibited` AR(26), LOCAL-POLICY External Connexion from http server AR(26) et `ICMP PING speedera` AR(26).

5.4.2 Anomalies détectées

Après avoir fixé les degrés du modèle et estimé les paramètres à partir des données d'estimation, nous avons appliqué la méthode complète de traitement des alertes aux données de validation. Nous avons ensuite examiné en détail les anomalies signalées pour chaque flux par les modèles choisis. Les résultats généraux figurent au tableau 5.2. Pour chaque flux, le nombre d'anomalies signalées est repris dans la colonne An. La détection des phénomènes connus et intéressants p_i est couverte dans les colonnes K+ et K-, montrant les phénomènes signalés et manqués, respectivement. Les p_i ont été identifiés à la section 3.2.1. Il est à noter que les p_i ne sont pas tous dans les données de validation et que K- ne contribue pas à An. De plus, l'outil a également signalé des anomalies. Il peut s'agir de :

1) nouveaux phénomènes intéressants, non identifiés dans l'inspection manuelle,
2) phénomènes inintéressants qui, à première vue, pourraient sembler quelque peu significatifs, mais qui font en fait partie du comportement normal,
3) artéfacts créés par les transformées que nous avons réalisées dans la chaîne de traitement, comme l'utilisation de l'opérateur ∇_{semaine}.

Les phénomènes du premier cas sont utiles à l'opérateur et le nombre d'occurrences est repris à la colonne N+. Les anomalies du deuxième cas sont plutôt inoffensives et généralement très facilement identifiées comme telles. Toutefois, elles font perdre du temps à l'opérateur. Le troisième cas est le pire, puisque l'outil signale des anomalies artificielles là où elles n'existent pas. Selon le flux d'alertes, leur identification correcte peut être soit banale, soit très difficile. Le nombre d'occurrences pour les deux derniers est repris à la colonne N-. Pour `ICMP PING speedera`, nous décrivons N+ et N- à la figure 5.4.2. Nous comparons également le modèle AR stationnaire au modèle EWMA. Les résultats avec $(1 - \lambda) = 0.92$ et $n = 3$ pour les données de validation sont présentés au tableau 5.3.

SNMP request udp Les quatre phénomènes connus dans les données de validation ont été signalés et toutes les anomalies supplémentaires sont des anomalies réelles, de petites vibrations en plus du flux d'alertes constant. Dans ce sens, ils ne sont pas du tout néfastes, nous avons juste considéré les onze autres moins intéressants que les plus importants. Les nouveaux phénomènes signalés peuvent être la manifestation

TAB. 5.2 – Phénomènes signalés et manqués, données de validation provenant de l'`ensemble-1` avec la modélisation AR stationnaire

Flux	An	K+	K-	N+	N-
SNMP	15	p_2, p_3, p_4, p_5	-	11	0
Whatsup	12	p_3, p_6, p_8	p_4, p_5, p_7	3	6
Dest unr	12	-	-	3	9
Local policy	12	p_2, p_4, p_5	p_3	4	5
Speedera	5	p_1, p_2	-	1	2
Total	56	12	4	22	22

TAB. 5.3 – Phénomènes signalés et manqués, données de validation provenant de l'`ensemble-1` avec la modélisation EWMA

Flux	An	K+	K-	N+	N-
SNMP	15	p_2, p_3, p_4, p_5	-	11	0
Whatsup	8	p_3, p_5, p_8	p_4, p_6, p_7	3	2
Dest unr	16	-	-	2	14
Local policy	9	p_2, p_4, p_5	p_3	2	4
Speedera	4	p_1	p_2	0	3
Total	55	11	5	18	23

de l'injection du trafic SNMP ou un comportement inoffensif mais intermittent dans le flux par ailleurs extrêmement constant.

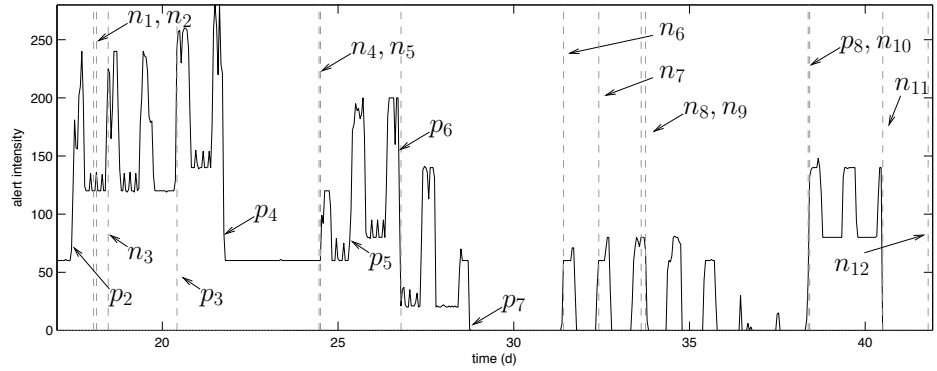
Pour ce flux, l'approche EWMA a signalé quelques anomalies, ce qui n'est pas étonnant, étant donné la nature constante du flux d'alertes.

ICMP PING WhatsupGold Windows Le flux et les anomalies émises sont illustrés à la figure 5.4(a). Les observations apparaissent en noir et les anomalies sont identifiées par des traits verticaux et des lignes grises. La série résiduelle est illustrée à la figure 5.4(b), où les valeurs résiduelles apparaissent en noir, le EWMA de l'erreur en bleu foncé, les limites de contrôle d'erreur en bleu clair et les anomalies, comme ci-dessus.

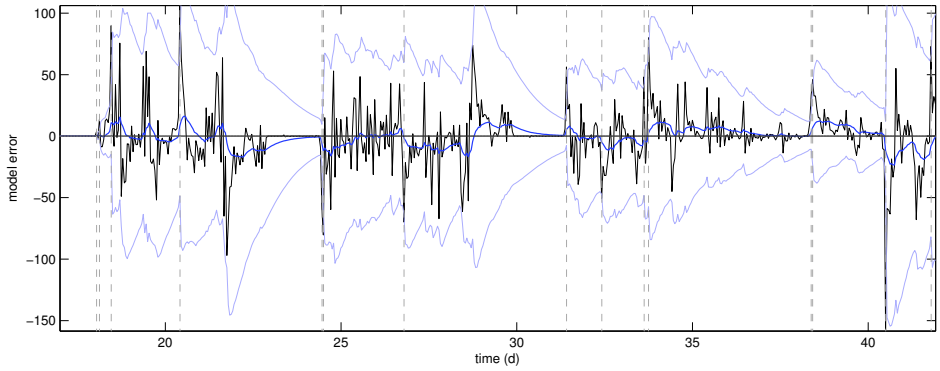
Les phénomènes connus p_3, p_6 et p_8 ont été détectés, mais p_4, p_5, et p_7 ont été manqués par l'outil. Le p_2 se trouve dans les premières observations des données de validation et nous l'avons exclu des considérations, de même que n_1 et n_2, étant donné que 1) un modèle AR(p) commence à fournir de bonnes prévisions à partir seulement de la $(\mathrm{p}+1)^{\mathrm{th}}$ observation intégrée dans l'équation (5.9), et 2) la composante détection doit recevoir plus de données pour un seuillage correct.

Deux nouvelles anomalies intéressantes, n_4 et n_7, sont émises aux changements de hauteurs des pics hebdomadaires, par rapport à la situation de la semaine précédente. La troisième, n_{11}, est émise lorsque l'intensité chute à zéro à la fin de la série. Tous les phénomènes sont des changements dans l'activité constante ou une augmentation durant les jours de la semaine. Les facteurs communs pour K- sont la petitesse du changement et le fait que le changement est placé après d'autres anomalies. La raison en sera discutée plus en détail à la section 5.5.

Des six occurrences N, n_5, n_9 et n_{10} sont des anomalies supplémentaires ou doubles,



(a) Observations and anomalies



(b) Residuals, baseline, control limits and signaled anomalies

FIG. 5.4 – Flux `ICMP PING WhatsupGold Windows`

signalées juste après un phénomène connu ou nouveau. Les trois anomalies restantes, n_3, n_6 et n_8 ont été émises aux pics hebdomadaires que nous considérons comme sans intérêt. En tant que telles, ces trois anomalies sont une distraction mineure pour l'opérateur, mais en réalité inoffensives. L'anomalie n_{12} est signalée à la fin du flux et exclue.

En comparaison avec l'approche EWMA, les anomalies signalées sont quasiment les mêmes. Le modèle AR a détecté p_6 et manqué p_5 et EWMA et vice versa. De plus, le modèle AR réagit à certaines variations quotidiennes qui ont été tolérées par les limites de contrôle plus importantes du modèle EWMA. Le modèle AR du comportement normal est plus précis et les anomalies connues manquées peuvent être constatées dans la série résiduelle $\hat{u}_t$, même si la méthode de détection ne les a pas relevées. Cette meilleure précision débouche sur des limites de contrôle plus étroites et, par conséquent, quelques anomalies N- inoffensives en plus.

ICMP Destination Unreachable Communication Administratively Prohibited

Ce flux ne contenait aucun phénomène intéressant connu, seulement un rythme hebdomadaire plutôt faible. L'outil a signalé au total douze anomalies.

Nous en avons classé trois dans la catégorie intéressante, étant donné qu'elles indiquaient des changements d'intensité généraux aussi bien pendant les périodes de

haute activité (semaine) que de faible activité (week-end), par rapport à la situation de la semaine précédente. Des neuf anomalies N-, deux étaient de toute évidence des artéfacts créés par ∇_{week}. Elles ont été relativement faciles à identifier comme phénomènes artificiels.

Nous n'avons pas trouvé d'explication aux sept autres. Dans un premier temps, cela peut sembler un résultat décevant. Toutefois, il est également intéressant de noter que l'étape de suppression de la périodicité explique de nombreux pics du flux original, qui auraient pu paraître suspects lors d'une inspection visuelle. Même si nous ne signalons que trois phénomènes intéressants, nous avons été capables d'en expliquer bien plus dans le cadre du rythme hebdomadaire.

Le modèle EWMA a signalé d'avantage d'anomalies, en majorité des pics faisant partie du rythme hebdomadaire. Nous avons attribué deux anomalies à N+, étant donné qu'elles étaient les mêmes que celles indiquées par le modèle AR. Toutefois, en analysant le flux pour la première fois au chapitre 4, elles n'ont pas pu être identifiées comme ayant une relation avec le changement dans le rythme hebdomadaire. Le modèle n'a pas été en mesure d'indiquer l'un des changements de niveau que nous avons constatés avec la modélisation AR. De plus, le modèle EWMA n'a pas été capable d'écarter les pics faisant partie du rythme hebdomadaire.

LOCAL-POLICY External connexion from http server Les phénomènes intéressants connus p_2, p_4 et p_5 ont été détectés, mais p_3 a été manqué. Etant donné que le p_3 manqué est un changement dans l'intensité des pics haute fréquence, passant des niveaux 1 à 2, parmi les pics atteignant jusqu'à 10000 alertes, nous ne considérons pas cela comme un défaut grave.

L'outil a indiqué quatre nouveaux phénomènes intéressants. Il s'agissait de pics supplémentaires ou manquants brisant le rythme hebdomadaire, ou des changements dans l'intensité des pics périodiques. En examinant la situation a posteriori, ils semblaient tout à fait évidents, mais pas si faciles à relever avant une analyse plus approfondie de la structure du flux à l'aide de la modélisation AR.

Les cinq N- contiennent deux artéfacts générés par ∇_{week}, deux anomalies signalées à des pics extrêmement élevés (3000 et 8000 alertes) qui, toutefois, font partie du rythme hebdomadaire, et une anomalie qui pourrait être causée par des variations dans les composantes bas niveau. Ces variations sont intéressantes, mais comme nous ne pouvons pas être certains que l'anomalie est réellement signalée à cause de ces variations, elle a été attribuée à N-. Etant donné que les deux anomalies artificielles étaient faciles à identifier, la catégorie N- pour ce flux contient des anomalies plutôt inoffensives.

Même si le flux d'alertes est très difficile à traiter, se composant principalement d'impulsions d'alertes énormes, nous avons été capables de relever les phénomènes intéressants connus. De plus, l'analyse avec la modélisation AR nous a aidé à mieux comprendre la structure du flux d'alertes en signalant de nouvelles anomalies et en ne réagissant pas à certains pics.

L'approche EWMA a été confrontée à des difficultés, en particulier avec ce flux. Dans la pratique, elle a signalé chaque pic et a été incapable d'exclure les pics suivant le rythme hebdomadaire. Toutefois, étant donné que la modélisation EWMA ne crée pas d'artéfacts, le nombre d'anomalies N- est plus petit qu'avec la modélisation AR. Nous avons considéré deux anomalies comme N+, étant donné qu'elles étaient les mêmes que celles signalées par la modélisation AR. Toutefois, de manière similaire

qu'avec `ICMP Destination Unreachable Communication Adminsitratively Prohibited`, nous ne les avons pas identifiées comme des anomalies intéressantes avec la modélisation EWMA seule.

ICMP PING speedera Le flux est décrit avec les anomalies signalées à la figure 5.4.2. Il contenait seulement deux phénomènes intéressants connus, p_1 et p_2, tous deux ayant été détectés.

N+ ne contenait qu'une seule alerte n_3, indiquant un vendredi d'une intensité inférieure à la normale.

Dans N-, deux anomalies signalées étaient liées à des artéfacts créés par ∇_{semaine}, Les deux anomalies connues sont renvoyées en écho dans la série transformée (y_1'', voir figure 5.1) une semaine plus tard, p_1 causant n_4 et p_2 causant n_5. Connaissant le comportement de ∇_{week}, ces artéfacts étaient faciles à identifier dans ce flux.

Les phénomènes n_1 et n_2 au début de la série sont des artéfacts créés pour les mêmes raisons que celles expliquées ci-avant pour le flux `ICMP PING WhatsupGold Windows`. Ces types d'artéfacts sont créés à l'aide aussi bien des approches EWMA que AR pour chaque flux. Nous pouvons ignorer ces anomalies systématiquement et nous les avons donc exclues des résultats.

Les résultats pour `ICMP PING speedera` contiennent de bons exemples d'artéfacts non désirés qui peuvent être créés par ∇_{semaine} et, en même temps, ils montrent à quel point nous sommes capables de filtrer les alertes faisant partie du comportement normal du flux.

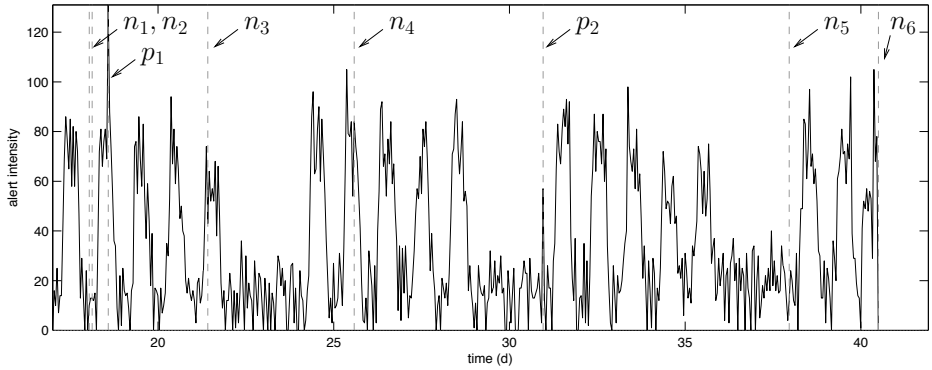
L'approche EWMA signale une anomalie uniquement au début de chaque lundi, lorsque l'intensité périodique augmente. En tant que telle, elle est incapable à la fois de traiter la forte composante périodique et de détecter tout changement non drastique dans le comportement du flux.

5.5 Discussion

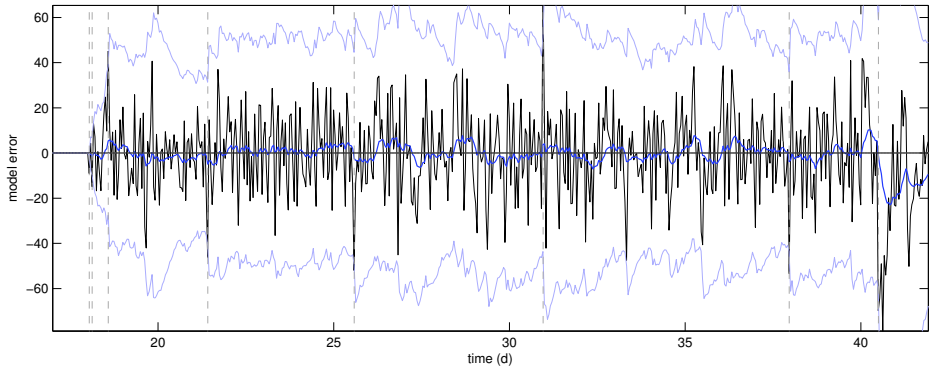
Dans l'ensemble, nous pouvons modéliser le comportement régulier et périodique dans les flux d'alertes et nous pouvons détecter des pics et creux brusques et similaires à des impulsions (les cinq flux) en dehors des rythmes quotidiens et hebdomadaires du flux d'alertes. Le comportement normal des flux d'alertes, se composant uniquement de pics similaires à des impulsions, par exemple `LOCAL-POLICY external connexion from http server`, est difficile à saisir avec le modèle proposé. Nous pouvons également détecter, jusqu'à un certain degré, les changements dans les composantes constantes (`SNMP request udp` et `ICMP PING WhatsupGold Windows`) et exclure certaines impulsions faisant partie du rythme normal (`ICMP Destination Unreachable Communication Adminsitratively Prohibited`). Toutefois, le retrait de tendance des séries rend la détection de ces changements plus difficile, puisque seule la transition demeure visible par la suite.

Les variations précédentes dans la série résiduelle (composante anormale) peuvent masquer l'anomalie actuelle, parce que le seuil de détection se base sur l'écart-type. De ce fait, les petits changements de niveaux, comme ceux dans `ICMP PING WhatsupGold Windows`, sont confrontés au risque de perte lorsque le flux d'alertes n'est pas très régulier ou constant. Toutefois, les changements sont présents dans les valeurs résiduelles et pourraient être relevés par d'autres moyens.

En d'autres termes, nous pourrions éventuellement saisir plus d'anomalies en développant la composante détection de la chaîne de traitement (Figure 5.1) ou simplement en person-



(a) Observations and anomalies



(b) Residuals, baseline, control limits and signaled anomalies

FIG. 5.5 – Anomalies détectées pour `ICMP PING speedera`. La série est le flux d'alertes original, correspondant aux valeurs s_1 de la figure 5.1 et les anomalies signalées sont indiquées en traits verticaux gris

nalisant le facteur de lissage et les seuils d'alertes. Toutefois, l'approche actuelle fonctionne suffisamment bien, bien qu'elle soit plutôt générique et les outils génériques sont plus faciles à déployer. Pour ces raisons, nous avons jugé que l'étape supplémentaire n'en valait pas la peine.

Pour en revenir à notre objectif principal : permettre à l'opérateur de se concentrer sur des tâches plus pertinentes en le soulageant de l'inspection manuelle de nombreuses alertes bénignes. Comme discuté au chapitre 4, nous utilisons la réduction dans des intervalles occupés pour mesurer l'efficacité de la récapitulation. Le tableau 5.4 montre le nombre d'anomalies signalées par l'outil dans la colonne *Anomalies* et le nombre d'intervalles non nuls nécessitant une inspection manuelle sans traitement automatisé dans la colonne *Manuel*. La colonne *Gain* montre le gain de temps sous forme de la proportion des tranches de temps d'une heure libérées de l'inspection manuelle lorsque le traitement des alertes est automatisé. L'approche proposée peut soulager l'opérateur de 90 % ou plus des contrôles d'état et le temps gagné peut être consacré à des tâches plus pertinentes. De plus, l'analyse réalisée en appliquant la méthodologie peut indiquer des phénomènes qu'il serait difficile de voir via une inspection visuelle, même en examinant les alertes au niveau du flux.

Tab. 5.4 – Gain de tranches de temps

Flux	Anomalies	Manuel	Gain
SNMP	15	564	0.97
Whatsup	11	390	0.97
Dest Unr	12	556	0.98
LOCAL-POLICY	12	118	0.90
Speedera	5	518	0.99

L'un des aspects négatifs réside dans le fait que la méthodologie signale quelques phénomènes artificiels, mais ils semblent généralement faciles à identifier. L'approche manque également certains phénomènes intéressants, typiquement de petits changements proches de perturbations plus larges. Ces coûts doivent être pondérés par rapport au gain. D'après les résultats, nous pouvons voir que le nombre de faux positifs est relativement petit. Toutefois, nous ne pouvons pas fournir des statistiques complètes sur la qualité de la détection et il nous manque des ensembles d'alertes analysées et réelles.

5.6 Conclusion

A partir des observations et de l'expérimentation de la modélisation EWMA, nous avons proposé une deuxième approche pour modéliser le comportement normal du flux. Nous avons supposé que les modèles de séries temporelles stationnaires pourraient être utilisés pour modéliser la partie normale du flux d'alertes. L'approche vise à filtrer le comportement normal du flux en trois étapes, en supprimant :

1. la tendance linéaire
2. la composante périodique
3. la structure stationnaire

à l'aide des techniques d'analyse de séries temporelles classiques.

Nous avons ensuite supposé que, dans des conditions normales, il n'existe que des tendances linéaires dans les flux d'alertes. La première étape utilise l'opérateur de différenciation pour supprimer ces tendances.

La deuxième étape utilise l'opération de différenciation également, mais nous devons connaître la période de la composante périodique. L'intuition a suggéré, et les résultats empiriques ont confirmé que les périodicités les plus importantes correspondent aux rythmes quotidiens et hebdomadaires. Nous avons constaté que, en éliminant la composante périodique avec la période d'une semaine, nous pouvions supprimer également les rythmes quotidiens.

La troisième étape, la modélisation autorégressive, avait pour objectif de saisir puis supprimer la structure stationnaire du flux. Nous avons défini le degré du modèle par itération manuelle, étant donné que les critères classiques de sélection des degrés ne conviennent pas dans notre cas en raison des données d'entraînement impures. Le besoin d'estimation des paramètres du modèle divise l'approche en phases d'entraînement et opérationnelle. Dans la phase d'entraînement, le modèle est estimé comme le problème des moindres carrés puis, dans la phase opérationnelle, la sortie du modèle est utilisée pour éliminer par filtrage la structure stationnaire restante du flux.

Après ces trois étapes de filtrage, nous avons une estimation de la composante anormale du flux d'alertes. Nous détectons les parties les plus significatives en utilisant une carte de

contrôle EWMA modifiée.

Nous avons présenté les résultats avec un outil utilisant cette approche. Ils indiquent que nous pouvons libérer une grande quantité de temps consacré à analyser ces alertes, en comparaison avec le traitement manuel. Comparativement à la méthode EWMA, les rythmes hebdomadaires ont été mieux pris en compte et la méthode a contribué à expliquer certains schémas de comportement non détectés par l'analyse EWMA.

Etant donné que nous utilisons des données réelles pour l'estimation du modèle, nous intégrons également des anomalies existantes en plus du comportement normal dans nos paramètres de modèle. Les résultats montrent que les anomalies dans les données d'estimation n'ont pas eu d'effet néfaste sur les capacités de détection.

Il faut toutefois conserver ce fait à l'esprit et éviter d'adapter trop bien ce modèle aux données.

Comme avec toutes les méthodes de traitement automatisées, il existe un risque de filtrer des alertes intéressantes et de manquer des phénomènes intéressants. Les exemples présentés ont montré que les phénomènes qui sont précédés d'autres anomalies peuvent être masqués. De plus, le traitement proposé a pour effet secondaire possible la création d'anomalies artificielles dans les données des alertes. Ces risques semblent toutefois relativement petits par rapport aux gains.

Les anomalies artificielles peuvent, dans certains cas, être difficiles à détecter et sont de toutes façons source d'ennui pour l'utilisateur. De plus, l'hypothèse stationnaire du comportement normal et le besoin de données d'entraînement qui s'étalent sur plus d'une semaine peuvent s'avérer de trop dans les environnements dynamiques. Pour traiter ces problèmes et améliorer la précision générale du modèle, nous explorerons l'utilisation de modèles non stationnaires et d'algorithmes d'estimation adaptative au chapitre suivant.

Chapitre 6

Modélisation des séries temporelles non stationnaires

Au chapitre 4, nous avons étudié l'utilisation d'un modèle de tendance pour appréhender le comportement normal dans des flux d'alertes. Même s'il a fonctionné dans certains cas, le modèle de tendance était une projection très brute de la réalité. L'idée de base consistait en le fait que les tendances à court terme dans le comportement du flux enregistreraient le comportement normal et que des changements abrupts seraient signalés comme des anomalies, puisqu'ils ne seraient pas prédits par le modèle.

L'étape suivante, décrite au chapitre 5, était basée sur l'hypothèse que, en plus des tendances linéaires, des composantes de flux périodiques et stationnaires sont provoquées par le comportement normal. En comparaison avec le modèle EWMA, nous avions une vue plus détaillée du comportement normal plutôt que de nous fier à une moyenne glissante à court terme pour saisir l'ensemble du comportement régulier. Les principaux inconvénients résidaient dans le fait que 1) nous avions besoin de supprimer les composantes tendancielles et périodiques du flux, 2) le modèle du comportement normal restant était stationnaire, c'est-à-dire estimé une seule fois et 3) la phase d'estimation nécessitait des données d'entraînement avant d'entrer dans la phase opérationnelle.

Le problème, avec la suppression de la composante, réside dans le risque d'introduire des artéfacts dans le flux. De plus, cela rend l'interprétation des résultats plus difficile, étant donné que le signal analysé est très différent de l'original. Par ailleurs, nous avons estimé la période de la composante périodique en utilisant une heuristique simple basée sur l'analyse des données d'alerte. L'heuristique permet de décider automatiquement si la transformation permettant de supprimer la composante périodique est appliquée, mais elle n'est pas parfaite et, de nouveau, elle augmente le risque d'artéfacts.

Un modèle stationnaire ne peut pas s'adapter aux changements dans le comportement normal du flux et doit être réestimé si le comportement normal du système évolue. Cela introduit une étape de gestion supplémentaire dans le système de traitement. De plus, des propriétés indésirables dans les données d'entraînement affecteraient le traitement jusqu'à la prochaine réestimation.

Si les paramètres du modèle tiennent compte des rythmes de comportement hebdomadaires, il faut au moins la valeur des données d'une semaine dans la phase d'estimation. De plus, si l'on utilise la transformation par suppression de la périodicité, il faudra encore plus de données. Tout cela retarde l'entrée dans la phase opérationnelle.

Enfin, les erreurs de modèle se sont révélées toujours assez importantes, indiquant

des problèmes dans les hypothèses sous-jacentes, dans les modèles ou dans les méthodes d'estimation du modèle.

Dans ce chapitre, l'idée de base est toujours la même : nous analysons la différence entre le comportement du flux observé et la sortie du modèle pour détecter les écarts par rapport au profil normal. Pour surmonter les problèmes susmentionnés, dans ce chapitre, nous modéliserons le comportement normal du flux avec des modèles de séries temporelles non stationnaires. En combinaison avec l'algorithme d'estimation adaptative, nous pouvons modéliser le flux d'alertes directement, sans supprimer aucune composante du flux. Dans le même temps, des données d'entraînement ou des phases de réestimation de modèle explicite ne sont pas nécessaires. Les revers de l'approche sont des algorithmes plus complexes et le risque d'incorporer un comportement anormal dans les modèles avec l'algorithme d'estimation adaptative.

La figure 6.1 est un exemple d'anomalies signalées lorsque les alertes sont traitées à l'aide de méthodes que nous présenterons dans ce chapitre. Le flux `ICMP L3retriever PING` a été échantillonné à un intervalle d'une minute. Dans le chiffre supérieur, nous reconnaissons le comportement du flux du jeudi matin au lundi soir, avec un week-end calme entre les jours ouvrables. Le graphique inférieur zoome sur les données du jeudi au vendredi. L'intensité observée des alertes est illustrée en noir, tandis que les traits verticaux et les lignes grises indiquent les anomalies signalées.

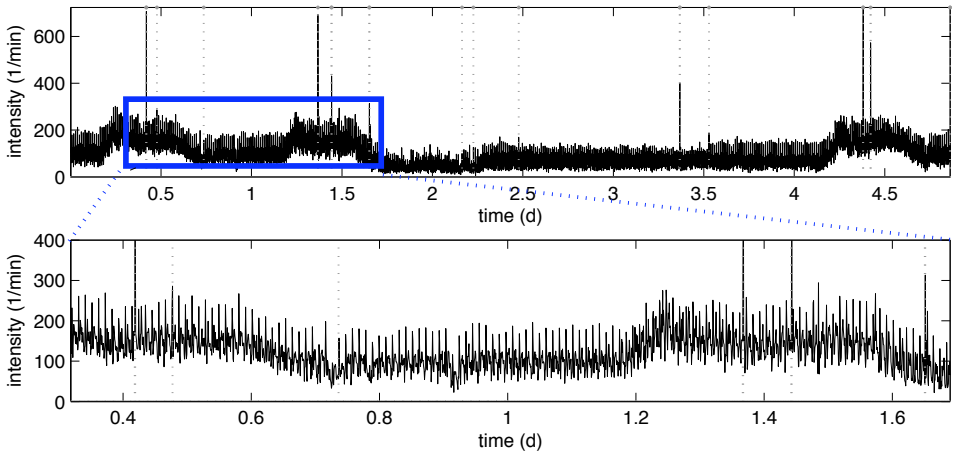


FIG. 6.1 – Exemple d'anomalies signalées lorsque les alertes sont traitées à l'aide de méthodes présentées dans ce chapitre. Le flux `ICMP L3retriever PING` a été échantillonné à un intervalle d'une minute. Dans le chiffre supérieur, nous reconnaissons le flux du jeudi matin au lundi soir, avec un week-end calme entre les jours ouvrables. Le graphique inférieur zoome sur les données du jeudi au vendredi

Le reste du chapitre est organisé comme suit. La section 6.1 décrit les modèles AR non stationnaires et un algorithme récursif appelé le *filtre de Kalman*, avec la variation que nous utilisons pour estimer les coefficients AR en fonction du temps. Nous montrons également comment les estimations spectrales instationnaires sont obtenues à partir des paramètres de modèles AR, puisqu'elles sont utilisées pour développer le futur travail. La section 6.2 décrit le travail connexe dans la détection d'intrusions et l'analyse du trafic sur le réseau. Les expérimentations faites avec les données de référence de l'`ensemble-1` sont décrites à la section 6.3 avec les expérimentations faites avec les flux obtenus par

échantillonnage à une fréquence supérieure dans l'`ensemble-3`. La section 6.4 discute des résultats et nous concluons le chapitre à la section 6.5.

6.1 Modèles et algorithmes

Un lecteur qui ne s'intéresse pas aux méthodes et équations peut sauter cette section et passer directement aux expérimentations à la section 6.3, en ne retenant que les éléments clés suivants :

- Le comportement normal du flux est modélisé comme une somme pondérée des observations précédentes.
- Les pondérations sont réestimées ou mises à jour à chaque nouvelle observation.
- L'estimation s'effectue à l'aide d'algorithmes, rendant l'estimation plus facile qu'il n'y paraît à première vue au niveau calcul.
- Grâce à l'étape de mise à jour et aux algorithmes utilisés, nous pouvons analyser les flux d'alertes sans supprimer aucune composante du flux.
- Les anomalies signalées sont les différences les plus significatives entre les prévisions fournies par le modèle et les observations.

Pour le reste, nous décrivons ces éléments clés ainsi que quelques autres éléments ci-dessous.

Un flux d'alertes est un flot d'alertes successives répondant aux critères d'agrégation à savoir la signature et le capteur générateurs, comme nous l'avons déjà vu. La mesure de l'intensité du flux prise à des intervalles de temps discrets fixes forme une série temporelle $\{y_t\}$. A la section 6.1.1, nous décrivons le modèle autorégressif non stationnaire que nous utilisons pour saisir le comportement normal du flux. Etant donné que nous utilisons un modèle paramétrique, les paramètres doivent être estimés.

Nous introduirons tout d'abord le *filtrage bayésien* à la section 6.1.2 et le cas spécial du filtrage de Kalman à la section 6.1.3. Le filtre de Kalman est un traitement de données adaptatif et récursif adapté à une estimation en ligne. Nous utilisons en fait une variation du filtre de Kalman appelé le lisseur de Kalman, décrit à la section 6.1.4. Il fournit de meilleures estimations au prix d'un petit retard supplémentaire. Le modèle non stationnaire et l'algorithme d'estimation adaptative nous permettent d'éviter les différentes étapes de transformation requises dans le chapitre précédent.

L'algorithme de détection des anomalies est défini à la section 6.1.5. et est essentiellement identique à celui utilisé avec le modèle AR stationnaire au chapitre précédent.

6.1.1 Modèle AR non stationnaire

Nous avons défini le modèle AR (p) stationnaire dans ((5.9)). Le modèle AR (p) non stationnaire est le même, avec les coefficients en fonction du temps :

$$y_t = \sum_{k=1}^{p} a_t^k y_{t-k} + e_t \ . \tag{6.1}$$

Ici, y_t sont les observations, e_t est le bruit blanc et a_t^k sont les paramètres du modèle instationnaire. En anglais, le modèle utilise une somme pondérée de p valeurs pour estimer la valeur actuelle. Les pondérations sont $a_t^k, \ \ k = 1 \ldots p$.

La non-stationnarité signifie que les pondérations a_t^k sont fonction du temps. L'idée est que l'utilisation normale du système provoque un flux suffisamment régulier et lisse,

que la valeur actuelle peut être prévue comme une combinaison linéaire des p valeurs passées. La partie du comportement du flux qui ne peut pas être prévue de cette manière est suffisamment anormale pour être signalée à l'utilisateur. En utilisant des modèles AR instationnaires, nous permettons au modèle du comportement normal de s'adapter aux changements du système surveillé.

6.1.2 Filtrage bayésien : traitement de données adaptatif en temps réel

Les paramètres du modèle dans ((6.1)) doivent être estimés. Contrairement aux modèles stationnaires, comme ceux utilisés au chapitre 5, les paramètres sont réestimés pour chaque instant t. Au chapitre précédent, l'estimation des paramètres a été réalisée à l'aide de l'estimation des moindres carrés avec l'algorithme utilisant toutes les données d'entraînement pour trouver les pondérations a_k, $k = 1, \ldots, p$ minimisant l'erreur entre la sortie du modèle et les observations. Dans le cas non stationnaire, toutefois, les paramètres a_t^k sont estimés à chaque instant t. L'approche naïve est de procéder de manière similaire pour chaque instant. Il existe heureusement des approches plus efficaces. Dans la théorie de l'estimation, le terme *filtrage* est utilisé pour la méthode de traitement des données en temps réel, fournissant des estimations des paramètres à partir des observations. Les estimations sont obtenues en utilisant 1) toutes les observations antérieures jusqu'à l'instant actuel et 2) un modèle d'évolution qui décrit les caractéristiques instationnaires de la cible. Le filtrage peut être considéré comme un processus où nous mettons à jour les connaissances du système de manière incrémentielle, au fur et à mesure de l'arrivée de nouvelles observations. En d'autres termes, plutôt que de commencer l'estimation depuis le tout début pour tous les t, nous mettons simplement à jour l'estimation précédente chaque fois qu'une nouvelle observation est effectuée. Du point de vue calcul, l'étape de la mise à jour est nettement plus facile étant donné que les algorithmes bénéficient de certaines propriétés des modèles et des données. Grâce au filtrage, il est faisable - point de vue calcul - d'utiliser des modèles non stationnaires.

Pour décrire le processus, nous utilisons un formalisme d'états [DK01]. Ce formalisme est également requis pour le filtrage et le lissage de Kalman. Dans notre cas, nous estimons les paramètres du modèle AR à partir des séries d'alertes *observées* $\{y_t\}$. Les vrais paramètres ne peuvent pas être observés directement et, dans la terminologie de l'espace des états, on les appelle l'*état* θ. Supposons maintenant que nous avons un *modèle d'observations* donnant la relation entre l'état non observable et les observations et un *modèle d'évolution* décrivant la nature instationnaire de l'état :

$$y_t = M_t\left(\theta_t, e_t\right) \quad , \tag{6.2}$$

$$\theta_{t+1} = N_t\left(\theta_t, w_t\right) \quad , \tag{6.3}$$

où les fonctions M_t et N_t sont supposées être des fonctions connues, (6.3) est markovien, t est un index temps discret, e_t et w_t sont des bruits d'observation et d'état. Les équations (6.2) et (6.3) sont appelées, respectivement, équations d'observation et d'état.

Dans le filtrage bayésien, les paramètres θ sont considérés comme des variables aléatoires et non comme des constantes déterministes. Deux densités de probabilité $p\left(y_t|\theta_t\right)$ et $p\left(\theta_t|\theta_{t-1}\right)$ sont liées aux modèles d'observation et d'évolution, et elles sont appelées respectivement, densité de certitude et d'évolution. La densité de certitude donne la probabilité d'avoir un certain état à l'instant actuel, étant donné l'état à l'instant précédent.

L'objectif du filtrage est de déterminer la distribution postérieure de l'état θ_t, conditionnée par les observations faites jusqu'à présent, $p(\theta_t|D_t)$, où $D_t = \{y_1, \ldots, y_t\}$. Si le premier état connu, c'est-à-dire $p(\theta_0|D_0) = p(\theta_0)$, les densités postérieures sont obtenues de manière récursive avec deux équations de mise à jour

$$p(\theta_t|D_{t-1}) = \int p(\theta_t|\theta_{t-1})\, p(\theta_{t-1}|D_{t-1})\, d\theta_{t-1} \ , \tag{6.4}$$

$$p(\theta_t|D_t) = \frac{p(y_t|\theta_t)\, p(\theta_t|D_{t-1})}{p(y_t|D_{t-1})} \ , \tag{6.5}$$

où

$$p(y_t|D_{t-1}) = \int p(y_t|\theta_t)\, p(\theta_t|D_{t-1})\, d\theta_t \ . \tag{6.6}$$

L'équation (6.4) est appelée l'équation de mise à jour de l'évolution et (6.5) l'équation de mise à jour de l'observation.

La densité de probabilité $p(\theta_t|D_{t-1})$ obtenue avec la mise à jour de l'évolution (6.4) est la *densité antérieure* pour l'état θ_t, lorsqu'une nouvelle observation y_t est effectuée. L'application alternée des étapes de mises à jour de l'évolution et de l'observation peut être interprétée comme une application séquentielle de la formule de Bayes (posterieur = certitude × anterieur). Le dénominateur $p(y_t|D_{t-1})$ dans (6.4) est juste un facteur d'échelle lorsque y_t est donné [Tar04, p.29].

La densité postérieure caractérise la solution du problème d'estimation, puisqu'elle décrit les probabilités des différents états donnés, étant donné les observations. Dans notre cas, cela donnerait les probabilités d'avoir certains coefficients AR étant donné les séries d'alertes observées. Dans la pratique, toutefois, différentes estimations de points sont utilisées, étant donné que la densité proprement dite n'est très pas illustrative. Lorsque le modèle d'évolution (6.3) est linéaire et que les densités de probabilité sont gaussiennes, l'estimation optimale de l'état peut être obtenue à l'aide d'un filtre de Kalman.

6.1.3 Cas normal linéaire : Estimation des paramètres AR avec le filtre de Kalman

L'algorithme du filtre de Kalman fournit des estimations optimales pour l'état θ_t dans les conditions suivantes : 1) les densités de probabilité sont gaussiennes, et 2) les modèles d'évolution et d'observation sont constantes. Même si les densités de probabilité ne sont pas gaussiennes, le filtre de Kalman offre l'estimateur linéaire optimal [Tar04, p.31] et, dans le cas de modèles non linéaires, il est possible d'utiliser le filtre de Kalman étendu [KS05].

Lorsque les conditions susmentionnées sont réunies, les modèles d'observation et dévolution (6.2) et (6.3) sont simplifiées comme suit

$$\theta_{t+1} = F_t\theta_t + G_t w_t \ , \tag{6.7}$$

$$y_t = H_t\theta_t + e_t \ , \tag{6.8}$$

où les fonctions F_t, G_t, et H_t sont supposées être connues. $e_t \sim N(0, W_t)$, $w_t \sim N(0, R_t)$ sont toutes deux indépendantes de θ_t et de l'une de l'autre. En d'autres termes, les fonctions $M(\theta_t, e_t)$ et $N(\theta_t, w_t)$ ont été simplifiées en des équations linéaires et des distributions arbitraires ont été limitées aux distributions gaussiennes de moyenne nulle.

Le modèle AR non stationnaire de (6.1) peut être mis sous forme de vecteur. Avec des coefficients instationnaires tels que

$$\theta_t = \left(-a_t^1 \ldots - a_t^p\right)^T \quad , \tag{6.9}$$

et les observations passées sous la forme

$$H_t = (y_{t-1} \ldots y_{t-p}) \quad , \tag{6.10}$$

nous avons le modèle AR non stationnaire de (6.1) sous la forme

$$y_t = H_t\theta_t + e_t \quad . \tag{6.11}$$

Sans information préalable, l'évolution de l'état (6.7) est souvent décrite avec un modèle de cheminement aléatoire [Tar04, p.54]. Cela signifie que dans (6.7) $F_t = F = I$ et $G_t = G = I$, c'est-à-dire que ce sont des matrices d'identité qui ne varient pas dans le temps. L'équation d'état devient :

$$\theta_{t+1} = \theta_t + w_t \quad . \tag{6.12}$$

Même s'ils sont simples, les modèles d'évolution de cheminement aléatoire sont utilisés dans de nombreuses applications [DK01, p.38]. Avec (6.12) et (6.11), nous avons le modèle AR non stationnaire sous forme d'espace d'états.

Pour représenter les équations du filtre de Kalman, nous utilisons $\hat{\theta}$ pour noter l'estimation de θ, $\tilde{\theta}$ pour l'erreur d'estimation et C_y pour la matrice de covariance de y. Les équations du filtre de Kalman pour l'estimation des paramètres AR en fonction du temps sont

$$C_{\tilde{\theta}_{t|t-1}} = C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}} \quad , \tag{6.13}$$

$$K_t = C_{\tilde{\theta}_{t|t-1}} H_t^T \left(H_t C_{\tilde{\theta}_{t|t-1}} H_t^T + C_{e_t}\right)^{-1} \quad , \tag{6.14}$$

$$\epsilon_t = y_t - H_t\hat{\theta}_{t-1} \quad , \tag{6.15}$$

$$\hat{\theta}_t = \hat{\theta}_{t-1} K_t \epsilon_t \quad , \tag{6.16}$$

$$C_{\tilde{\theta}_t} = (I - K_t H_t)\, C_{\tilde{\theta}_{t|t-1}} \quad . \tag{6.17}$$

Il est à noter que ce ne sont pas les équations générales du filtre de Kalman, étant donné que l'utilisation du modèle d'acheminement aléatoire simplifie la situation. De plus, l'erreur de prévision à une étape ϵ_t de l'observation y_t est utilisée pour estimer le bruit d'observation inconnu et non mesurable e_t. Voir [Tar04] pour plus de détails.

Dans la pratique, l'utilisation du filtre de Kalman nécessite des valeurs initiales pour l'état θ_0, la covariance d'erreur $C_{\tilde{\theta}_0}$, la covariance de bruit d'état C_w et la covariance du bruit de l'observation C_e. Une approche courante est de définir $\theta_0 = 0$ et $C_{\tilde{\theta}_0} = I$ et d'exécuter l'algorithme sur un court segment depuis les données d'observation en sens inverse. Les valeurs obtenues de cette manière pour θ et $C_{\tilde{\theta}_0}$ sont ensuite utilisées pour initialiser ces valeurs dans le traitement réel. Même si nous avons utilisé cette étape d'initialisation, elle n'est pas nécessaire à moins qu'il ne s'agisse d'événements intéressants précoces dans les données. Pour le rapport, nous avons utilisé cette approche avec $200 + p$ points de données lors de l'estimation des modèles AR non stationnaires.

La vitesse d'adaptation du filtre de Kalman est déterminée par la covariance du bruit d'état C_{w_t}. En d'autres termes, elle contrôle la rapidité avec laquelle l'état s'adapte aux

changements dans les observations. Dans [Tar04, p.55], $C_{w_t} = \sigma_w^2 I$ et $C_{e_t} = \sigma_e^2 = 1$ (σ_w^2 est le coefficient de la covariance du bruit d'état et σ_e^2 le coefficient du bruit d'observation) ont été jugés favorables et ont également été utilisés dans cette thèse. Cela signifie qu'il y a un seul coefficient σ_w^2 pour ajuster l'adaptation.
La vitesse d'adaptation augmente avec σ_w^2 et la variance des estimations d'état est inversement proportionnelle à la valeur de σ_w^2 . Elle devrait dès lors être choisie pour obtenir l'équilibre souhaité entre la variance d'estimation d'état et la vitesse d'adaptation du filtre.

6.1.4 Estimations améliorées mais retardées avec le lisseur à décalage fixe de Kalman

L'utilisation de l'estimation d'état des observations futures est appelée *lissage*. Lorsque nous utilisons les observations $y_1, \ldots, y_{t+L}$ pour estimer l'état θ_t with $L > 0$ avec $L > 0$, la qualité de l'estimation peut probablement s'améliorer par rapport à l'estimation obtenue avec le filtrage de Kalman. Il existe trois algorithmes de lissage classique, *lisseur à point fixe*, *lisseur à intervalle fixe* et *lisseur à décalage fixe*. Chacun d'entre eux a une utilisation particulière.

Supposons que N soit le nombre total d'observations. Un lisseur à point fixe pourrait être utilisé pour estimer l'état du processus initial lorsque le processus progresse et que de nouvelles observations sont disponibles. En d'autres termes, le lisseur à point fixe fournit des estimations $\hat{\theta}_{t|L}$ pour le point fixe $t = j$ pour tous les $L = 1, \ldots, N$, au fur et à mesure que de nouvelles observations deviennent disponibles.

Un lissage à intervalle fixe convient pour l'utilisation hors ligne permettant de traiter les données par lot. Il utilise toutes les observations du lot pour estimer l'état à chaque instant. En d'autres termes, le lisseur à intervalle fixe fournit des estimations $\hat{\theta}_{t|L}$ pour tous les instants $t = 1, \ldots, N$ avec $L = N$ fixe utilisant toutes les données $(y_1, \ldots, y_N)$.

Un lissage à décalage fixe convient pour le traitement en ligne lorsqu'un petit délai fixe de L est autorisé. En d'autres termes, le lisseur à décalage fixe fournit des estimations $\hat{\theta}_{t|t+L}$ pour tous les instants $t = 1, \ldots, N$; avec un décalage fixe de L nous devons attendre des points de données futurs.

Dans cette thèse, nous utilisons le lissage à décalage fixe, car il convient au traitement en ligne. Pour plus d'informations sur d'autres méthodes de lissage, voir [DK01, Tar04, KS05].

Pour estimer l'état θ_t à l'instant t avec un lisseur à décalage fixe, nous attendrons d'avoir des observations jusqu'à l'instant $t + L$ où $N \geq 0$. A l'aide de la notation probabiliste, l distribution postérieure $p(\theta_t|D_t)$ devient $p(\theta_t|D_{t+L})$ et ce que l'on appelle les *variables étendues* sont utilisées dans les équations d'état et d'observation. Les équations du filtre de Kalman restent les mêmes, mais les variables étendues d'état et d'observation sont :

$$\theta_t^* = \begin{bmatrix} \theta_t \\ \theta_{t-1} \\ \vdots \\ \theta_{t-L} \end{bmatrix}, \tag{6.18}$$

$$H_t^* = \begin{bmatrix} H_t & 0 & \ldots & 0 \end{bmatrix}, \tag{6.19}$$

et l'équation d'état simplifiée (6.12) devient :

$$\begin{bmatrix} \theta_{t+1} \\ \theta_t \\ \theta_{t-1} \\ \vdots \\ \theta_{t-(L-1)} \end{bmatrix} = \begin{bmatrix} \theta_t \\ \theta_{t-1} \\ \theta_{t-2} \\ \vdots \\ \theta_{t-L} \end{bmatrix} + \begin{bmatrix} w_t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6.20}$$

$$\theta^*_{t+1} = \theta^*_t + w^*_t \ .$$

L'équation d'observation (6.8) peut être formulée comme suit :

$$y_t = \begin{bmatrix} H_t & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \theta_t \\ \theta_{t-1} \\ \vdots \\ \theta_{t-L} \end{bmatrix} + \epsilon_t \tag{6.21}$$

$$y_t = H^*_t \theta^*_t + \epsilon_t \ .$$

Il existe différentes solutions au problème du lissage à décalage fixe de Kalman. Pour donner une idée du coût du calcul, il a été rapporté que l'algorithme de lissage proposé dans [CST94] est d'environ $(M+2)/2$ fois aussi cher que le filtrage de Kalman. Ici, $M \leq L$ est « le nombre d'analyses rétrospectives calculées à chaque temps d'observation ». Dans notre cas, $M = L$ et les valeurs L sont 1 ou 5, comme nous le décrirons à la section 6.3. Cela signifie que l'utilisation d'un lisseur ou lieu d'un filtre augmente les coûts de calcul d'environ 1,5 à 3,5 fois, respectivement.

6.1.5 Détection des anomalies

A notre avis, le modèle devrait saisir le comportement normal du flux. De ce fait, la différence entre les prévisions à une étape à l'aide de (6.1) et les observations représente la composante anormale du flux. En d'autres termes, nous utilisons la sortie du modèle AR $\hat{y}_t$ obtenue à l'aide des observations y_{t-1} comme entrée de (6.1) pour former une série résiduelle $\tilde{y}_t = y_t - \hat{y}_t$. La série résiduelle représente la composante anormale du flux d'alertes.

Nous détections les anomalies en analysant la composante anormale. L'algorithme de détection est le même que celui utilisé avec les modèles AR stationnaires au chapitre 5 et défini dans (5.12). Nous le répétons ici pour plus de facilité. Nous estimons la moyenne résiduelle $\mu_{\tilde{y}_t}$ avec EWMA et l'écart-type $\sigma_{\tilde{y}_t}$ à l'aide de (4.8). Nous signalons une anomalie lorsque la valeur résiduelle actuelle $\tilde{y}_t$ dépasse les limites de contrôle $\hat{\mu}_{\tilde{y}_{t-1}} \pm n\hat{\sigma}_{\tilde{y}_{t-1}}$, où $n \in \mathbb{R}_+$ contrôle la largeur des limites de contrôle.

6.2 Travail connexe

Dans cette section, nous examinerons le travail précédent à l'aide de méthodes similaires dans les domaines de la détection d'intrusions et de la surveillance du réseau. Après cet examen, nous poursuivrons avec les résultats pratiques à la section 6.3. La vue d'ensemble des différentes approches de corrélation a été présentée au chapitre 2. Nous nous concentrerons essentiellement ici sur des approches similaires du point de vue de la méthode. Chaque sous-section présente un champ d'application pour des méthodes similaires et aborde les différences avec notre approche.

6.2.1 Modélisation et détection des vers

Zou et al. détectent les vers en surveillant les scannages quittant et pénétrant un réseau et en examinant la croissance exponentielle du nombre de scannages [ZGTG05]. L'idée est que le taux de scannage reste sous un certain seuil dans des conditions normales et augmente exponentiellement uniquement en présence d'un ver. Le document présente trois différents modèles pour la propagation du ver, dont l'un basé sur le modèle épidémique classique $\frac{\mathrm{d}I_t}{\mathrm{d}t} = \beta I_t \left[N - I_t\right]$, où I_t est le nombre d'hôtes infectés au moment t, N est le nombre d'hôtes vulnérables, $\beta = \alpha N$ le taux d'infection par paires et α, le taux d'infection par hôte infecté. Le modèle a été utilisé sous différentes formes, par exemple par Staniford et al. [SPW02]. Le deuxième modèle est une version logarithmique du premier. Le troisième modèle est le *modèle exponentiel AR*, $I_t = (1 + \alpha\Delta)\, I_{t-1}$ où Δ est la longueur de l'intervalle d'échantillonnage.

Le document compare les propriétés des différents modèles. Pour nous, toutefois, les points d'intérêt sont 1) le modèle autorégressif et 2) l'utilisation de l'algorithme du filtre de Kalman pour estimer le taux de croissance exponentielle α dans les trois modèles.

Dans les trois modèles, Zou et al. s'intéressent à α, qui est considéré comme ne variant pas dans le temps et utilisent un filtre de Kalman pour obtenir l'estimation $\hat{\alpha}$. Lorsque l'estimation oscille autour de zéro, les scannages entrants sont considérés comme bruit de fond et l'oscillation autour d'une constante positive, comme un signe de propagation du ver. Côté modèle, Zou et al. utilisent un modèle stationnaire et à faible degré pour modéliser le comportement malicieux. Dans notre cas, le degré du modèle est nettement plus élevé, $p = 20$, pour saisir un comportement plus complexe que la croissance exponentielle. En raison également de la modélisation d'un comportement plus complexe, nous utilisons un modèle non stationnaire. Côté estimation, impliquée par la stationnarité du modèle de Zou et al., le paramètre estimé est stationnaire et leur objectif est d'obtenir une bonne estimation de la valeur du paramètre aussi tôt que possible. Dans notre cas, les paramètres estimés varient dans le temps.

6.2.2 Surveillance du réseau

Dans la surveillance du réseau, une vue du trafic au niveau du réseau dans un domaine administratif s'appelle une *matrice de trafic* (TM). Il s'agit d'une représentation des volumes de trafic dans les flux *origine-destination* (OD) dans le réseau et son estimation et sa prévision constituent un des problèmes actuels dans le domaine de la surveillance du réseau. La matrice de trafic pourrait être observée directement, par exemple à l'aide de NetFlow dans tous les routeurs du réseau, mais l'observation directe à l'aide des méthodes actuelles est considérée comme trop onéreuse [PTL04].

Soule et al. [SSNT05] utilisent le filtrage de Kalman et la modélisation d'états pour estimer les flux origine-destination à partir des mesures niveau liaison, comme les comptages d'octets fournis par SNMP. Les flux OD modélisés sont des agrégats de haut niveau à la granularité d'un point de présence (PoP) à un point de présence dans le réseau de base européen de Sprint.

Ils utilisent un système qui est linéaire et gaussien, de sorte que les équations d'évolution et d'observation se présentent respectivement sous la forme (6.7) et (6.8). Le vecteur d'état θ_t contient des comptages d'octets du flux OD qui doivent être estimés. L'observation y_t est un vecteur de comptage d'octets niveau liaison obtenu via SNMP. H_t est la matrice de routage où l'élément h_{ij} est 1 si la paire OD i est présente sur la liaison j. La matrice F_t

« saisit le comportement dynamique du système » et est estimée à part avec l'algorithme *attente-maximisation* (EM) (pour plus de détails, voir [BD02]).

Au niveau de l'idée, on peut constater qu'il y a une certaine équivalence entre l'estimation des flux OD à partir des métriques de liaison et l'estimation du comportement normal du flux à partir des intensités du flux d'alertes. De plus, les flux d'alertes et les flux OD illustrés dans le document semblent partager certains comportements.

Il existe une différence évidente dans les modèles. Soule et al. considèrent H_t, θ_t et F_t comme stationnaires. Nos H_t et θ_t sont instationnaires, contenant des observations passées y_{t-j} et des coefficients de modèle AR a_t^j pour $j = 1, \ldots, p$. Notre $F_t = I$, est également statique. La matrice de routage (H_t) et la dynamique du système (F_t) peuvent toutefois évoluer au fil du temps. Dans leur approche, cette évolution est détectée comme une augmentation durable dans l'erreur de modèle mesurée avec une métrique d'erreur suivant la trace de l'erreur d'estimation générale normalisée sur toute la matrice de trafic. Dans ce cas, les capteurs NetFlow sont activés temporairement et F_t est recalibré à l'aide de l'algorithme EM.

Une autre différence provient du fait qu'ils *peuvent* obtenir de bonnes données connues sur les flux OD. Dans notre cas, il est quasiment impossible d'obtenir des flux d'alertes vraiment propres et normaux. Par conséquent, nous ne connaissons pas non plus les bons coefficients AR connus. Il s'agit en quelque sorte d'un problème fondamental[1], discuté plus en détail dans 6.4.

Il existe plusieurs autres approches de l'estimation TM. Dans [SLT+05], les méthodes existantes sont divisées en trois générations et cinq méthodes d'estimation différentes des générations deux et trois sont comparées. Elles s'appellent *tomogravité, changement de route, sortance, PCA* (analyse en composantes principales), les méthodes de *Kalman*. La méthode de Kalman est la même que dans [SSNT05], discutée ci-dessus. Selon les auteurs, c'est la première fois que les techniques de filtrage de Kalman sont utilisées pour le problème. De la comparaison, il résulte que les approches de la troisième génération sont les mieux adaptées à l'estimation TM, mais ce résultat ne s'applique pas à l'estimation du comportement normal du flux, en raison des différences de modèles et de contexte.

6.2.3 Détection d'intrusions

Pour autant que nos connaissances soient exactes, le filtrage de Kalman n'a pas été utilisé largement dans la détection d'intrusions. Hall et al. [HBE04] utilisent le filtrage bayésien pour améliorer la détection de l'usurpation d'adresses MAC dans les réseaux sans fil. La détection se base sur les empreintes digitales de la fréquence radio et l'association d'une empreinte digitale avec une adresse MAC. Le filtrage bayésien est utilisé pour réduire l'incertitude de détection en utilisant plusieurs observations avant la décision finale quant à savoir si l'adresse MAC doit ou non être considérée comme usurpée.

L'interférence bayésienne ou les réseaux bayésiens ont été utilisés dans la corrélation des alertes, par exemple par Vlades et Skinner dans [VS00], Goldman et al. dans [GHH+01], et Qin et Lee dans [QL04]. A part le nom et certains principes bayésiens sous-jacents, les méthodes sont différentes.

[1] Les spécialistes du traitement des signaux et de la physique avec qui nous avons discuté sont toujours étonnés par le manque d'exemples propres ou de définitions de comportement « bon » ou « normal » que nous souhaitons modéliser, ainsi que par le manque de bonnes données de test. Le domaine de détection d'intrusions semble en convenir et, depuis l'apparition d'ensembles de données DARPA largement disputés [LFG+00, LHF+00], il semble que des efforts doivent encore être faits pour améliorer la situation [SYB06]

6.2.4 Analyse spectrale pour la détection DDoS

L'analyse spectrale a été utilisée dans la détection, l'identification et la classification des attaques DDoS. Dans [CKT02], les auteurs observent le nombre d'arrivées de paquets sur une liaison pendant un intervalle de temps fixe pour former un signal. Ils proposent de mesurer la puissance de la fréquence harmonique fondamentale du signal pour distinguer les flux TCP normaux des flux d'attaques. L'idée de base est que, étant donné la procédure d'*accusé de réception* (ACK) dans TCP et les *round trip times* (RTT) du réseau, les flux TCP présentent une périodicité importante à des fréquences correspondant à des multiples de RTT. Une inondation d'attaques DoS n'aurait pas ce comportement périodique. Selon les auteurs, une attaque DoS par inondation de paquets pourrait être détectée en mesurant la puissance de la fréquence fondamentale. Ils présentent des résultats de simulations, indiquant que au moins un flux d'attaques UDP pourrait être distingué du flux TCP près de son origine. Plus on s'éloigne de l'attaquant, plus le nombre de flux agrégés augmente et le signal d'attaque s'atténue. Le document présente également l'analyse des traces du trafic TCP normal pour confirmer que les flux TCP normaux n'ont pas une composante harmonique fondamentale importante. La PSD est estimée à l'aide de la méthode dite *périodogramme de Welch* [Wel67].

Hussain et al. [HHP03a] utilisent la fréquence des composantes fondamentales et harmoniques des attaques DoS pour les classer comme attaques à source unique ou à sources multiples. En analysant le trafic réel provenant d'un ISP de taille modérée, les auteurs signalent que, pour un seul attaquant, la fréquence fondamentale est supérieure que pour plusieurs attaquants. Pour un attaquant unique, des facteurs comme un processeur hôte et l'interface de réseau, ainsi que la vitesse de connexion, limitent le taux d'attaque et créent une harmonique fondamentale à haute fréquence. Dans le cas de plusieurs attaquants, la fréquence fondamentale est inférieure, étant donné que les flux d'attaques « s'effacent ensemble ». La PSD est estimée comme une transformée de Fourier de la séquence d'autocorrélation des observations. Seuls les segments de données déterminés comme stationnaires sont utilisés pour l'estimation du spectre. La tendance du signal est estimée à l'aide de la régression linéaire des moindres carrés et la pente doit être de zéro avec 95 % de confiance pour le segment de données à considérer comme stationnaire. Le document propose également de mesurer le comportement d'accélération pour faire la distinction avec des attaques à source unique et à sources multiples. Dans [HHP03b], Hussain et al. proposent une empreinte digitale spectrale pour identifier les attaques DoS exécutées à partir du même groupe d'hôtes avec le même outil.

En comparaison avec notre travail, la philosophie de détection est différente dans ces approches. Ces méthodes se basent sur la représentation de domaine fréquentiel du signal. De plus, au moins des parties du signal sont supposées stationnaires, tandis que dans notre cas, les séries sont non stationnaires. Les méthodes se concentrent également sur la détection et l'analyse des attaques DDoS, alors que nous nous intéressons aux anomalies plus en général.

6.2.5 Analyse en ondelettes pour la détection d'anomalies de réseau

Comme susmentionné, l'analyse en ondelettes fait partie de la même famille de méthodes que l'analyse de la PSD instationnaire. Les ondelettes ont été utilisées par Barford et al. [BKPR02] pour analyser le trafic sur le réseau. Ils se servent des intensités de flux, paquets et octets des données NSMP et NetFlow pour former le signal analysé. Leur idée est de disposer d'un système générique qui serait 1) portable dans les environnements d'exploi-

tation et 2) utilisable dans différentes parties du système, comme les routeurs marginaux et de sous-réseau. Même s'ils utilisent la plate-forme d'analyse pour visualiser les signaux, leur but est la détection automatisée des anomalies. Leur système est fortement orienté sur l'absence d'artéfacts et est relativement précis tant en temps qu'en fréquence grâce aux propriétés de la transformée en ondelettes.

Ils analysent un signal enregistré sur deux mois. Le signal transformé en ondelettes est divisé en composantes basse, moyenne et haute fréquence. Ces composantes séparées sont utilisées pour synthétiser trois signaux du domaine temporel correspondant aux trois niveaux de détail. Les composantes moyenne et haute fréquence sont normalisées en variance unitaire, puis mises en fenêtres et combinées pour former la partie variable du signal. Une mesure de la variabilité locale appelée *score des écarts* détecte les anomalies avec un degré de confiance bas et haut. L'approche est comparée à la prévision de Holt-Winters en exécutant des signaux correspondant à un journal des anomalies du réseau fournissant des anomalies identifiées manuellement dans les données. Le score des écarts donne des résultats légèrement meilleurs et, selon les auteurs, présente de meilleures propriétés dans l'ensemble.

Lakhina et al. ont classé l'analyse en ondelettes comme une méthode d'analyse en mode par lots [LCD04] et c'est également la manière dont elle a été utilisée dans [BKPR02]. Même si l'application de l'algorithme du filtre de Kalman que nous utilisons fonctionne en mode par lots, le filtre et le lisseur à décalage fixe de Kalman sont spécialement prévus pour le traitement en ligne. La récursivité nous permet de mettre à jour nos estimations de manière incrémentielle au fur et à mesure que de nouvelles observations arrivent et c'est une différence clé. Même le filtrage en mode par lots traite les observations une à la fois et chaque nouvelle observation entraîne une mise à jour des estimations. Il pourrait donc facilement être appliqué comme algorithme en ligne. La puissance des approches basées sur les ondelettes est la nature multi-échelle de la transformée. En analysant les signaux dans le domaine temporel avec un ensemble donné de paramètres, nous ne détectons probablement les anomalies que d'une certaine échelle. Pour la détection multi-échelle, nous devons analyser plusieurs cas parallèles de flux à différents intervalles d'échantillonnage.

6.3 Expérimentations : Augmentation de la précision du modèle

Après avoir présenté la méthode et l'algorithme à la section 6.1 et les avoir comparés au travail existant à la section 6.2, cette section démontre comment notre méthode fonctionne dans la pratique.

Nous présenterons les résultats des deux expérimentations suivantes. Nous commencerons par appliquer les algorithmes du lisseur de Kalman sur les données de l'`ensemble-1` pour comparer l'approche aux deux méthodes précédentes. La deuxième partie montre les résultats obtenus avec le filtre de Kalman et les données de l'`ensemble-3`. L'outil utilisé pour le test a été élaboré en Matlab et nous avons utilisé les programmes du filtre et du lisseur de Kalman développés par Mika Tarvainen à l'Université de Kuopio, Finlande.

6.3.1 Expérimentations faites avec l'`ensemble-1`

Les flux d'alertes générés par les cinq signatures les plus prolifiques de l'`ensemble-1` ont été traités avec une méthode AR non stationnaire. Un lisseur à décalage fixe de Kalman,

tel (6.21),(6.20) et(6.13)-(6.17) avec un décalage $L = 5$, a été utilise pour estimer le modèle AR (p) instationnaire de (6.1).

L'adaptabilité du lisseur de Kalman est contrôlée par le coefficient de covariance du bruit d'état σ_w^2 , comme expliqué à la section 6.1.3. Le choix est similaire à l'équilibrage des pondérations des valeurs historiques et des nouvelles valeurs avec le facteur de lissage $(1 - \lambda)$ pour EWMA (voir section 4.1). Après quelques expérimentations initiales, nous avons choisi $\sigma_w^2 = 0.000025$. Des valeurs plus importantes rendent l'algorithme plus adaptatif et diminuent l'erreur de modélisation. Toutefois, le problème est que nous ne souhaitons pas un modèle trop exact, étant donné que les observations contiennent des anomalies de courte durée que nous ne souhaitons pas modéliser.

Pour comparer le modèle non stationnaire et le lissage à décalage fixe de Kalman avec les modèles stationnaires utilisés au chapitre 5, nous avons utilisé $p = 26$ pour tous les flux sauf `SNMP request udp`. Nous avons constaté que, en raison de la nature stable des flux, $p = 4$ était suffisant dans le cas stationnaire. L'application du lisseur de Kalman nécessite toutefois d'utiliser $p > L$ et donc, pour SNMP request udp, nous avons utilisé $p = 6$. Avec les modèles AR stationnaires, nous avons utilisé les 400 premiers échantillons comme données d'entraînement pour estimer les paramètres du modèle. Des algorithmes adaptatifs nous permettent d'entrer presque immédiatement dans la phase opérationnelle. Nous attendons $1.2p$ observations, les tronquant à une valeur entière, avant de commencer à signaler les anomalies, étant donné que les toutes premières estimations se sont quelque peu égarées. Le signal résiduel $\tilde{y}_t$ a été analysé pour les anomalies, comme décrit à la section 5.12, en utilisant $n = 3$ comme avec le modèle stationnaire.

Le tableau 6.1 montre les phénomènes connus détectés et non détectés identifiés à la section 3.2.1. Contrairement au cas stationnaire, aucune anomalie connue ne se trouvait en dehors des données d'entraînement. En plus des anomalies connues, la méthode de traitement a signalé de nouvelles anomalies. Il peut s'agir de

1. nouveaux phénomènes intéressants, non identifiés dans l'inspection manuelle,
2. phénomènes inintéressants qui, à première vue, pourraient sembler quelque peu significatifs, mais qui font en fait partie du comportement normal.

Les phénomènes du premier cas peuvent être utiles à l'opérateur et le nombre d'occurrences est repris à la colonne N+. Les anomalies du deuxième cas sont reprises à la colonne N- . Elles sont plutôt inoffensives et généralement très facilement identifiées comme telles. Toutefois, elles font perdre du temps à l'opérateur. La plupart du temps, nous avons classé dans N- des phénomènes trop insignifiants ou des anomalies supplémentaires signalées à proximité d'une anomalie intéressante déjà signalée. La différence la plus notable est l'absence du cas trois, les artéfacts générés par les transformées de signal, en comparaison avec la méthode AR stationnaire. En comparant la méthode stationnaire à la méthode non stationnaire, on pourrait croire que, en dehors du fait que les méthodes AR non stationnaires signalent plus d'anomalies, les deux donnent des résultats similaires. Pour voir les différences, il faut analyser chaque flux plus en détail.

Nous présenterons ensuite deux figures pour chacun des cinq flux. La première illustre les observations en noir, la sortie du modèle en rouge et les limites de contrôle en bleu clair. Les anomalies signalées sont indiquées par des traits verticaux gris.

SNMP request udp est illustré à la figure 6.2. Les pics les plus élevés aussi bien dans les observations que dans les erreurs de modèle sont coupés pour mieux montrer les détails de l'intensité plus petite. La méthode de traitement signale toutes les anomalies connues. La catégorie N+ contient de petites vibrations en plus de la composante

TAB. 6.1 – Phénomènes signalés et manqués

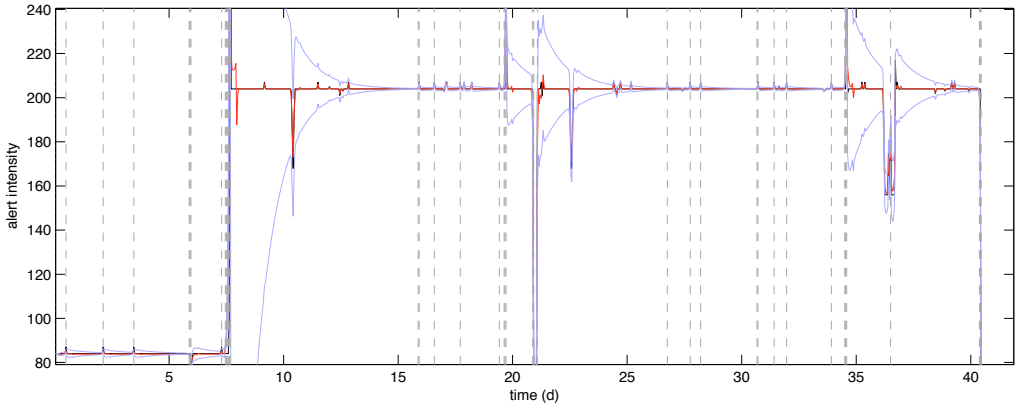
Flux	An	K+	K-	N+	N-
SNMP	39	p_1, p_2, p_3, p_4, p_5	-	17	17
Whatsup	26	$p_1, p_2, p_3, p_5, p_7, p_8$	p_4, p_6	4	16
Dest Unr	30	-	-	3	27
LOCAL-POLICY	19	p_4, p_5	p_1, p_2, p_3	0	17
Speedera	10	p_2	p_1	7	2
Total	124	14	6	56	54

de flux constante. Les anomalies N- sont toutes des doubles anomalies émises peu après les anomalies connues. En comparaison avec la méthode AR stationnaire, le nombre d'anomalies N+ est plus grand et les anomalies N- sont totalement nouvelles. La méthode AR non stationnaire est plus sensible, signalant une plus grande proportion de variations plus petites dans le flux. En même temps, les anomalies précédentes masquent moins que leurs successeurs. Par exemple, p_3 indique une chute de courte durée dans l'intensité des alertes et la méthode de traitement émet trois anomalies, deux à la chute de l'intensité et une troisième lorsque l'intensité remonte au niveau normal. Toutes les anomalies N- sont faciles à identifier comme doubles.

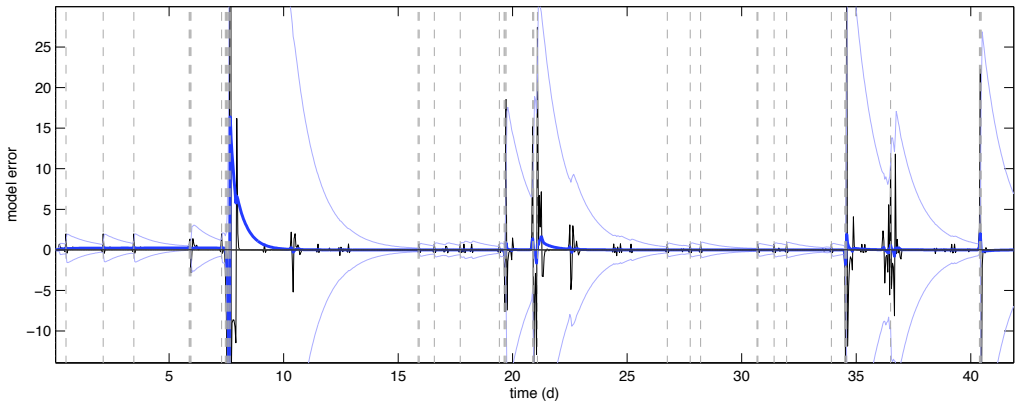
ICMP PING WhatsupGold Windows est illustré à la figure 6.3. Des anomalies connues, à part p_4 et p_6 ont été détectées. Les deux anomalies non détectées sont des chutes dans la composante constante, combinées à la fin d'un pic quotidien. La méthode AR non stationnaire est capable d'indiquer des changements de niveau constants, de même que des variations en milieu de semaine, comme p_3 et p_5 et la disparition de la composante constante marquée par p_7. Les anomalies N+ sont de réels petits pics dans une période normalement calme. Les deux pics de courte durée d'environ 20 et 30 alertes par heure pendant le week-end des jours 36 et 37 sont deux exemples de N+. Les alertes N- sont pour la plupart des doubles et certaines indiquent des changements qui pourraient être considérés comme anormaux dans un horizon à court terme, par exemple un jour, mais qui s'intègrent en fait dans le rythme hebdomadaire.

ICMP Destination Unreachable Communication Administratively Prohibited est illustré à la figure 6.4 Le flux ne contient aucune anomalie connue et a moins de structure que les autres flux. La méthode AR non stationnaire indique effectivement la plupart des pics isolés durant le temps d'une observation. De plus, l'augmentation d'intensité plus longue au cours des jours 7 et 8 est balisée. Nous avons classé trois anomalies dans N+. La première était une augmentation d'intensité de plus longue durée, la deuxième était le pic du jour 11 qui monte même plus que l'anomalie de longue durée et la troisième était un petit pic de petite intensité en dehors du rythme hebdomadaire normal au cours du jour 29. Deux des anomalies N- sont des doubles, et le reste est émis aux pics du flux. Il s'agit d'anomalies claires et isolées, mais que nous avons classé dans N- en raison de leur grand nombre.

Même dans un tel flux variable, la méthode est capable d'isoler la plupart des pics de petite durée et rien d'autre, sans ajuster spécifiquement les paramètres de la méthode pour ce type de flux. Cela indique de bonne capacités générales de détection et filtrage de la méthode. L'intérêt de ces anomalies dépend du contexte général, mais



(a) Observations, model output, control limits and anomalies



(b) Model error, control limits and anomalies

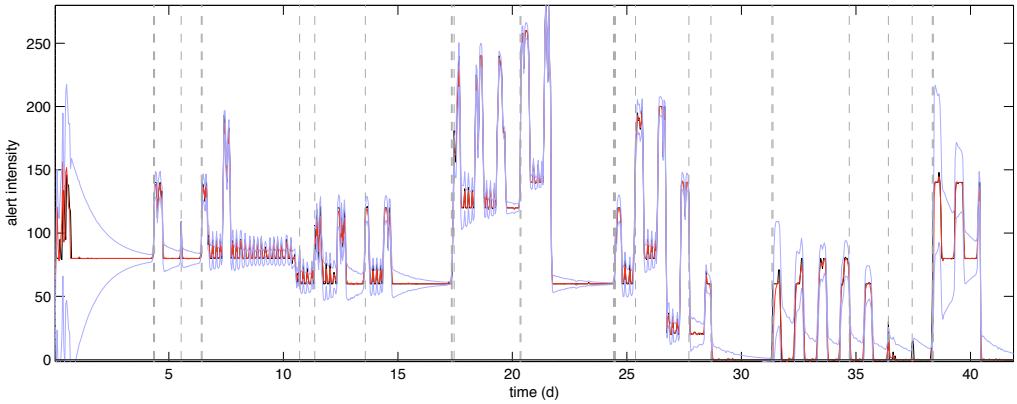
FIG. 6.2 – Flux `SNMP request udp`

la méthode est au moins capable de les mettre en valeur.

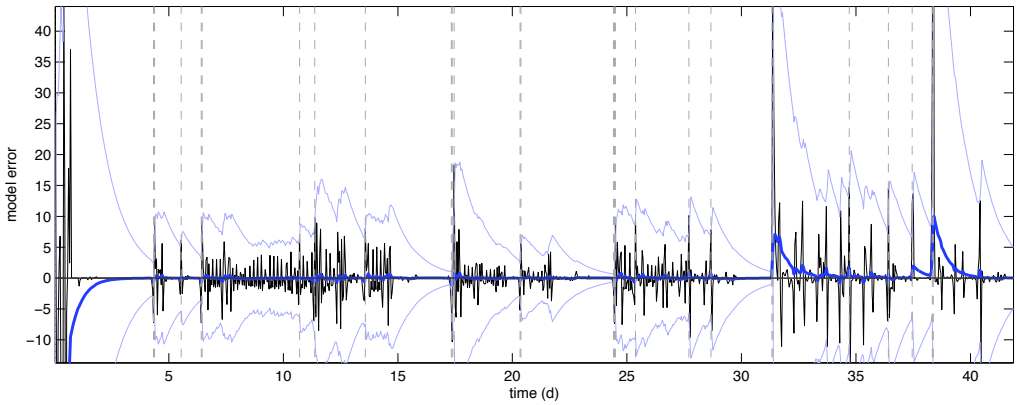
D'une part, en comparaison avec la méthode AR stationnaire, la méthode AR non stationnaire n'est pas capable de supprimer la partie pics du rythme hebdomadaire. D'autre part, cette méthode n'introduit pas d'artéfacts qui pourraient être difficiles à identifier dans ce type de flux. Même si certaines impulsions ne sont pas signalées par l'algorithme de détection, elles sont visibles dans la série résiduelle.

En comparaison avec les approches EWMA et AR stationnaire, l'approche AR non stationnaire relève significativement plus de pics. Le comportement normal du flux est modélisé de manière plus précise et, par conséquent, les pics ressortent plus clairement dans la série résiduelle.

LOCAL-POLICY External connexion from http server est illustré à la figure 6.5. Les anomalies détectées connues, p_4 et p_5 sont des pics de grande intensité. Le grand pic p_2 manqué était masqué par le pic précédent. Dans cette perspective, il n'est pas surprenant que les changements de l'activité d'extrêmement bas niveau, les anomalies connues p_1 et p_3, n'aient pas été détectées. Les performances sont les mêmes qu'avec la méthode AR stationnaire. Toutefois, comme avec le flux précédent, la méthode AR stationnaire saisit mieux le rythme horaire de très grosses impulsions.



(a) Observations, model output, control limits and anomalies

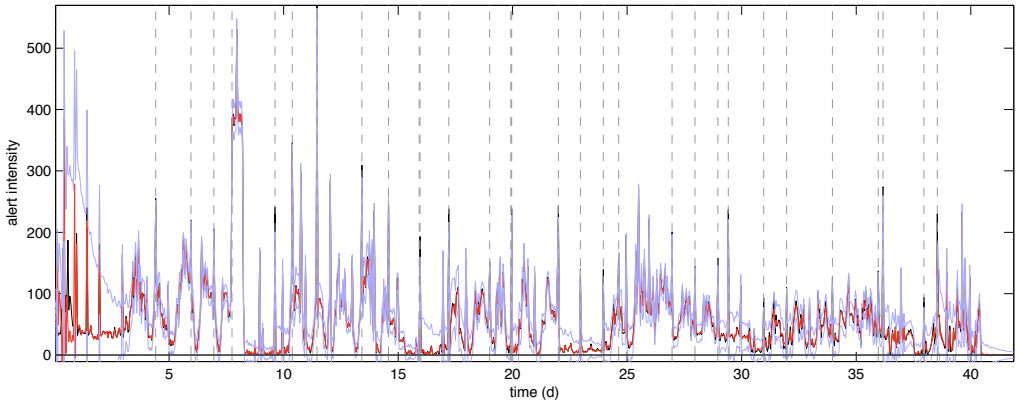


(b) Model error, control limits and anomalies

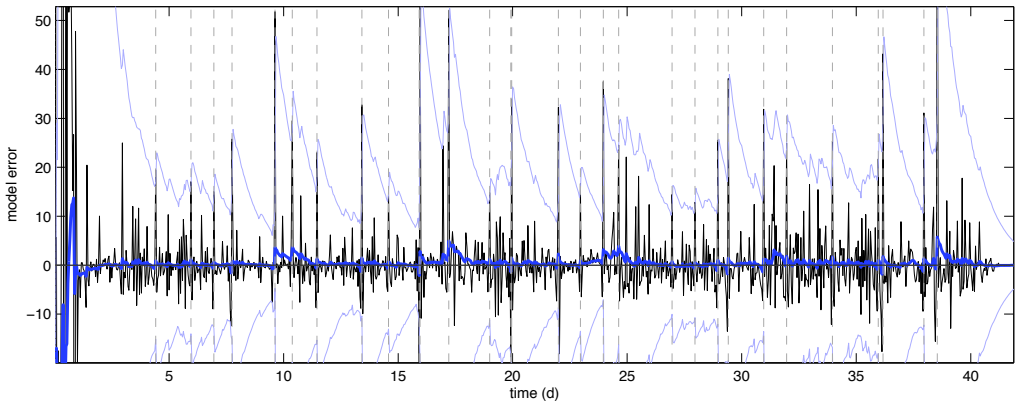
FIG. 6.3 – Flux `ICMP PING WhatsupGold Windows`

ICMP PING speedera est illustré à la figure 6.6. De ce flux, la méthode AR non stationnaire indique des pics isolés, à la fois pendant les jours de grande intensité que pendant les week-ends de faible intensité. Sur les anomalies connues, p_1 a été manqué et p_2 a été détecté. L'absence de détection est probablement due au fait qu'elle s'étale sur plusieurs observations. Par conséquent, elle est moins abrupte que quelques petits pics signalés pendant la semaine.

A en juger par la série d'erreur du modèle des cinq flux, le modèle non stationnaire est plus précis que le modèle stationnaire. De plus, lorsque nous utilisons des modèles non stationnaires et l'algorithme d'estimation adaptative, nous n'avons pas besoin de transformées de signaux telles que ∇_{semaine} utilisées avec les modèles AR stationnaires. C'est à la fois un point fort et un point faible. Même si l'utilisation de ∇_{semaine} peut introduire des anomalies artificielles, elle empêche certains phénomènes suivant un rythme hebdomadaire d'être signalés comme des anomalies. Ces phénomènes sont des pics isolés abrupts, tandis que les rythmes hebdomadaires plus lisses sont éliminés par filtrage. La visibilité du modèle AR par le passé est limitée par le degré de modèle p. Avec un intervalle d'échantillonnage d'une heure et $p = 26$, le modèle utilise uniquement les observations faites jusqu'à il y a 26 heures pour prédire les prochaines observations. Pour vérifier ce raisonnement, nous exécutons la méthode AR non stationnaire à l'aide de $p = 170$, ce qui



(a) Observations, model output, control limits and anomalies



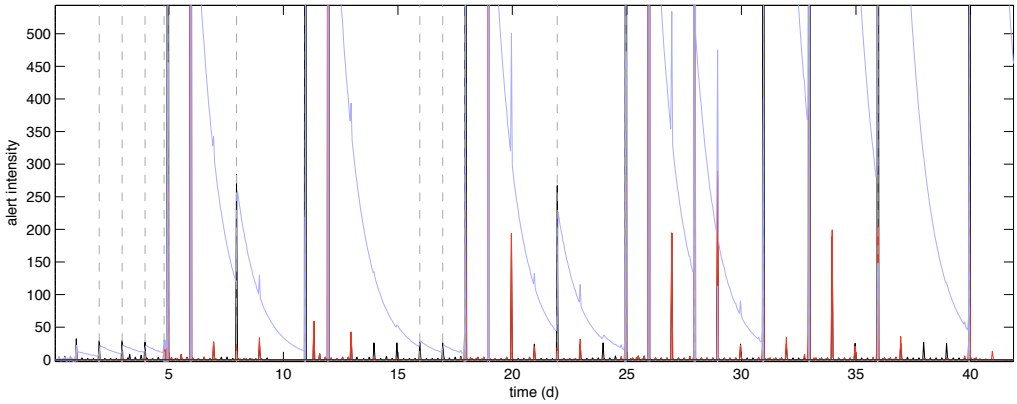
(b) Model error, control limits and anomalies

FIG. 6.4 – Flux `ICMP Destination Unreachable`

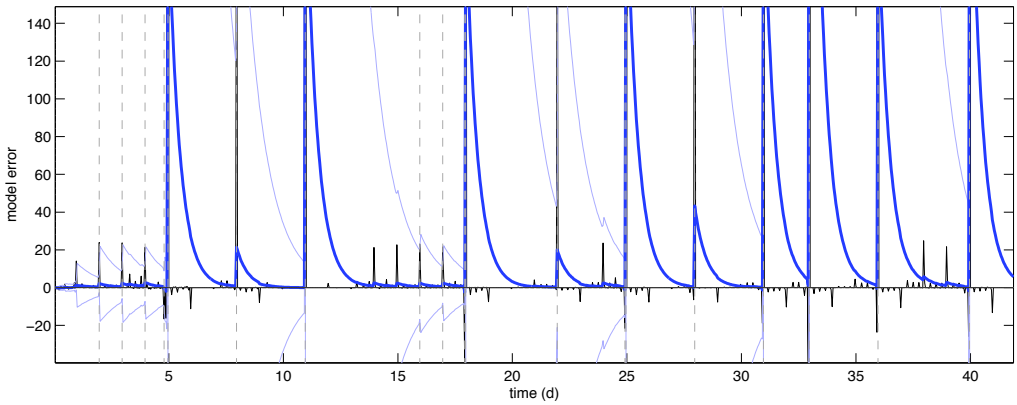
est un peu plus de 168 heures dans une semaine.

La figure 6.7 montre les résultats pour le flux `ICMP PING speedera` et la figure 6.8, les résultats pour le flux `ICMP Destination Unreachable`. Dans le cas de `ICMP PING speedera`, on peut voir d'après les graphes des erreurs du modèle que le modèle saisit réellement les comportements hebdomadaires. A la figure 6.6(b) avec $p = 26$, un pic d'erreurs intervient les lundis, lorsque l'intensité des alertes augmente selon le rythme hebdomadaire. A la figure 6.7(b), le comportement des lundis matins peuvent être prévus par le modèle. Par conséquent, il n'y a pas de pics d'erreur les lundis matins pouvant 1) provoquer des anomalies de type N- ou 2) masquer les anomalies suivantes. Le premier lundi matin est une exception, étant donné que le modèle n'a pas de comportement passé qu'il peut utiliser comme référence. Les pics d'alertes suivants peuvent sembler intervenir au même moment que l'augmentation des alertes les lundis, mais ce n'est pas le cas. A la figure 6.7(b), le pic d'erreurs et l'anomalie signalée aux alentours du jour 18 est p_2. Le pic d'erreurs au jour 26 est en fait provoqué par l'activité plus élevée que la normale le mardi matin. Le pic d'erreurs et l'anomalie émise à la fin du jour 31 est p_2, une anomalie pendant le week-end. Enfin, pour la semaine, seulement partiellement dans les données, il n'y a pas de pic d'erreurs le matin du jour 38.

Dans le cas de `ICMP Destination Unreachable`, la méthode AR non stationnaire



(a) Observations, model output, control limits and anomalies

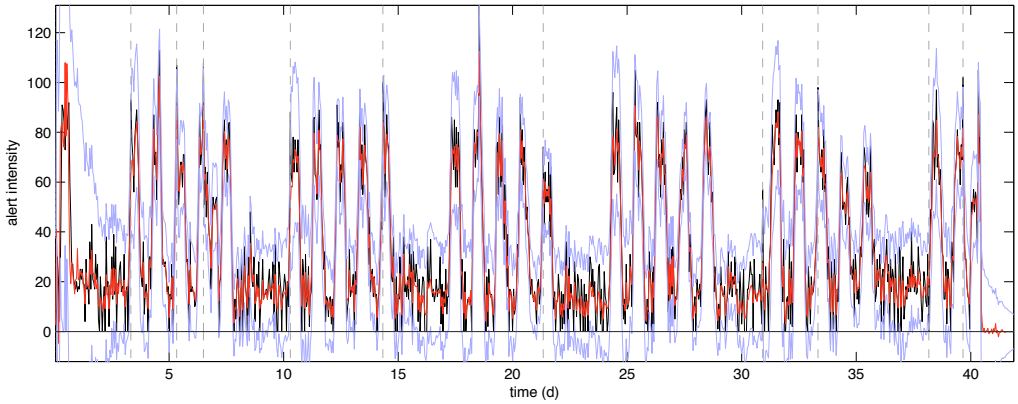


(b) Model error, control limits and anomalies

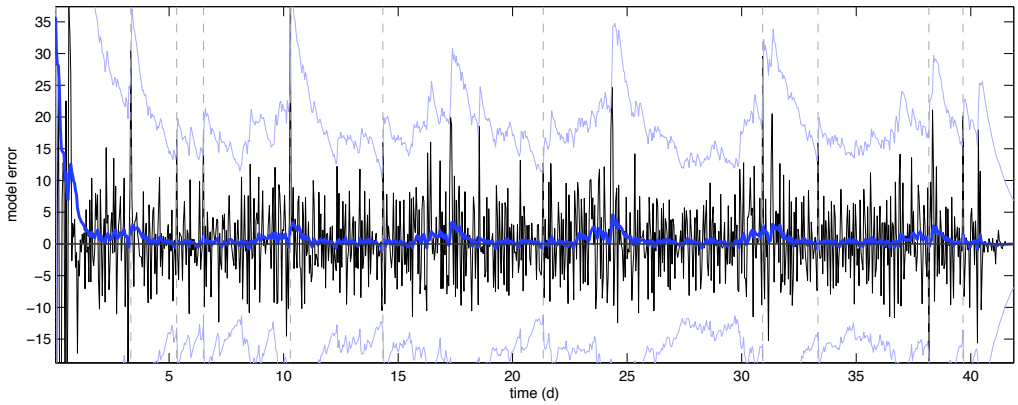
FIG. 6.5 – Flux `LOCAL-POLICY external connexion from HTTP server`

signale effectivement les mêmes anomalies avec $p = 26$ et $p = 170$. Avec $p = 170$, il y a cinq anomalies de moins qu'avec $p = 26$. Un p plus grand a pour effet que la méthode entre dans la phase opérationnelle plus tard, mais elle signale ensuite quelques pics de plus une fois opérationnelle. Toutefois, les erreurs de modèle sont significativement plus petites, montrant qu'un p plus grand signifie souvent un modèle plus précis (ainsi que des coûts de calcul plus élevés). Si l'on examine les erreurs de plus près, on constate que certains pics qui font partie du rythme hebdomadaire sont *proportionnellement* plus petits avec $p = 170$ qu'avec $p = 26$. Par exemple, des pics font partie du rythme hebdomadaire au cours des week-ends de basse intensité des jours 15-16 et 22-23. Les erreurs à ces pics sont *proportionnellement* plus petits avec $p = 170$ qu'avec $p = 26$. Par proportionnellement, nous entendons l'erreur par rapport à l'erreur lors des pics dans les observations tombant en dehors du rythme hebdomadaire. Cela signifie que avec $p = 170$, le modèle explique mieux ces pics qu'avec $p = 26$, c'est-à-dire que le comportement hebdomadaire est pris en compte jusqu'à un certain degré.

Dans les deux flux, les pics d'intensité abrupts ressortent également dans les valeurs résiduelles. Ce qui indiquerait que, même avec un degré de modèle aussi élevé et malgré l'estimation dynamique, les anomalies ne sont pas totalement incluses dans le modèle de comportement normal.



(a) Observations, model output, control limits and anomalies



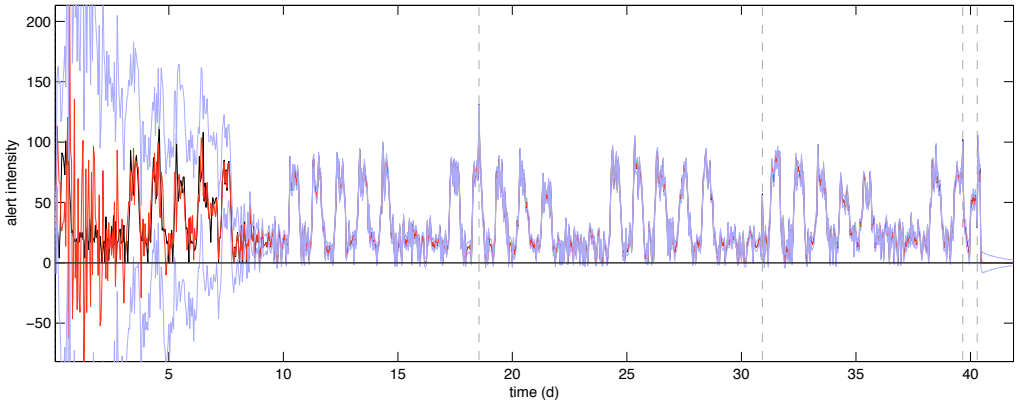
(b) Model error, control limits and anomalies

FIG. 6.6 – Flux `ICMP PING speedera`

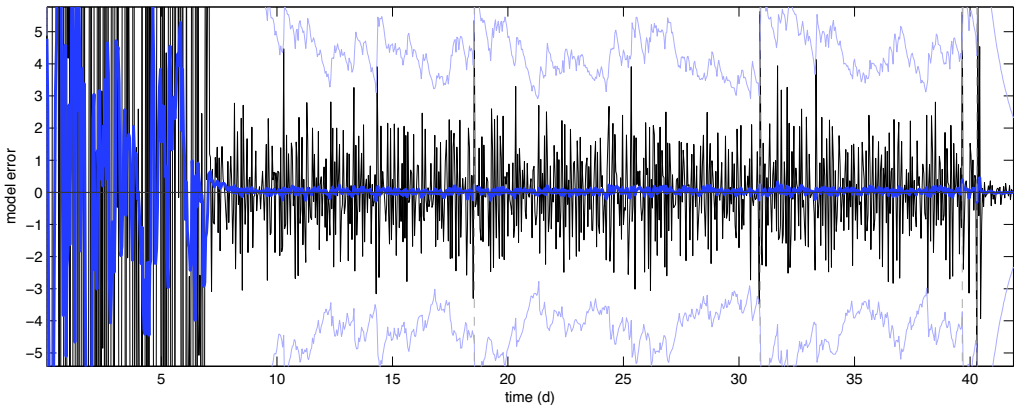
6.3.2 Expérimentations avec l'ensemble-3

Même si l'intervalle d'échantillonnage d'une heure a été jugé suffisant pour notre utilisation avec le décalage supplémentaire de L observations provoquées par le lissage de Kalman, le retard de détection devient long. Pour améliorer la rapidité de détection, nous utilisons des intervalles d'échantillonnage plus petits, d'une minute à vingt minutes pour les flux d'alertes dans l'`ensemble-3`. Les flux ont été présentés et analysés au chapitre 3. Comme mentionné dans ce chapitre, l'augmentation de t_s entraîne des flux plus lisses, étant donné que les fluctuations haute fréquence sont lissées. Le comportement normal s'est révélé visible à tous les intervalles d'échantillonnage utilisés, mais seules les anomalies dont l'échelle de temps est proche de l'intervalle d'échantillonnage sont clairement visibles dans le flux. La forme d'anomalie est toutefois restée similaire à travers toutes les échelles de temps.

Etant donné que les fluctuations haute fréquence rendent le traitement du flux plus difficile, l'utilisation de $t_s = 1$ min constitue le plus grand défi. Nous souhaitons signaler le même type d'anomalie que dans le flux avec un intervalle d'échantillonnage plus grand, tout en évitant de baliser les fluctuations aléatoires les plus visibles avec $t_s = 1$ min. Nous pouvons diminuer ces fluctuations basse intensité, haute fréquence en lissant le flux avec



(a) Observations, model output, control limits and anomalies

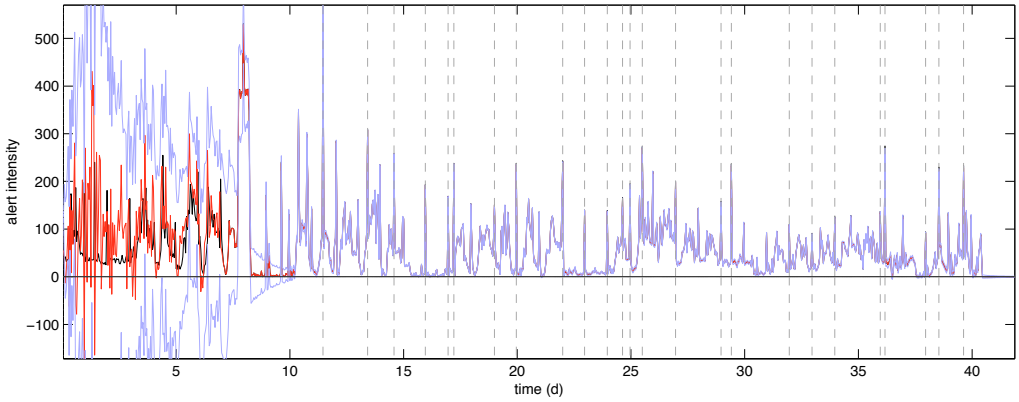


(b) Model error, control limits and anomalies

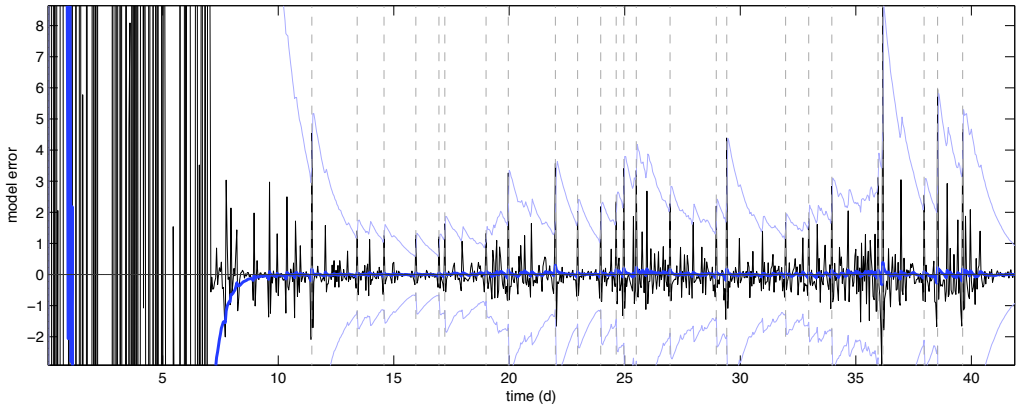
FIG. 6.7 – Flux `ICMP PING speedera` avec $p = 170$

EWMA avant analyse. Nous avons utilisé le facteur de lissage 0.6 qui équivaut à utiliser la moyenne glissante simple avec une fenêtre de longueur quatre. L'opération peut également être considérée comme filtrage passe-bas des observations. Nous montrerons également comment, par exemple, la méthode se comporte à des intervalles d'échantillonnage plus grands pour détecter des anomalies d'échelle de temps plus grande.

Nous souhaitions explorer la possibilité d'avoir un ensemble général de paramètres (p, L, n, σ_w^2) qui pourrait fonctionner pour la plupart des flux. Cela a été fait en trouvant un ensemble efficace de paramètres à l'aide d'un échantillon de certains flux, puis en les appliquant à tous les flux. Nous avons commencé l'expérimentation par le flux `SNMP request udp`, mais, comme nous l'expliquerons ultérieurement, ce flux a été particulièrement difficile à traiter en raison de la présence de fluctuations similaires à un train d'impulsions dans tout le flux. Nous avons donc utilisé à la place une portion d'environ les neuf premiers jours du flux `SNMP public access udp` pour trouver des valeurs adaptées aux paramètres. Ces paramètres ont ensuite été utilisés pour tous les flux, pour voir comment ils fonctionnaient. La méthode de traitement a été jugée impropre pour deux flux, en raison des composantes importantes similaires à des impulsions. Pour le reste, l'ensemble de paramètres donne de bons résultats. Même pour les flux les plus difficiles, il est possible de trouver un meilleur ensemble de paramètres, en abandonnant l'approche générale et en travaillant flux par



(a) Observations, model output, control limits and anomalies



(b) Model error, control limits and anomalies

FIG. 6.8 – Flux `ICMP Dest Unr` avec $p = 170$

flux. Passons maintenant aux détails.

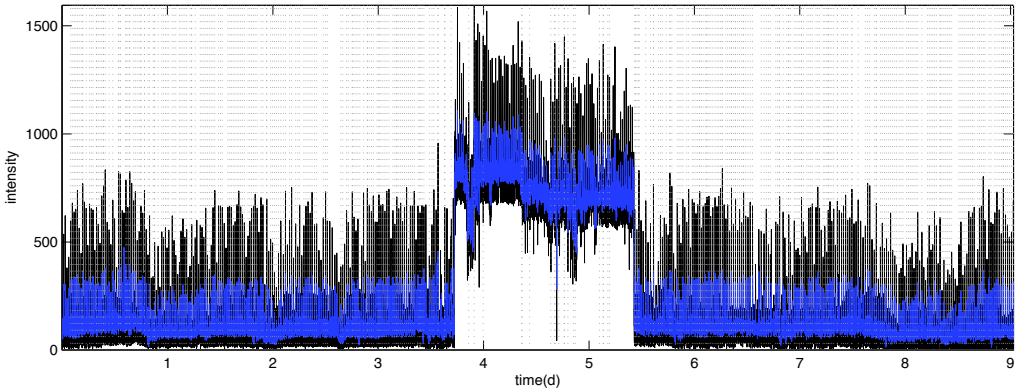
Avec l'`ensemble-1` et t_s = 1h, la raison d'avoir $p > 24$ était de permettre aux modèles de saisir les rythmes quotidiens. Etant donné que nous utilisons un intervalle d'échantillonnage plus court, ce point n'est plus valide. Pour réduire les coûts de calcul, nous avons choisi d'utiliser $p = 20$. En pratique, cela signifie que le modèle estime la densité du flux à l'aide des observations des 20 dernières minutes. Les paramètres du modèle sont estimés, toutefois en utilisant toutes les données passées via les récursivités de Kalman. Par ailleurs, nous avons décidé d'utiliser les mêmes paramètres d'algorithme qu'avec l'`ensemble-1` : $L = 5$ et $\sigma_w^2 = 0.000025$.

Les premières expérimentations ont montré que nous avons besoin de $n = 4$ dans l'algorithme de détection basé sur EWMA de (5.12) pour maintenir des niveaux d'alertes raisonnables. En comparaison avec la méthode AR stationnaire, nous avons déjà amélioré le retard de détection de la pire éventualité, d'une heure à six minutes. Pour réduire encore le retard, nous avons utilisé $L = 1$ et nous sommes également passés au filtre de Kalman, c'est-à-dire $L = 0$. Une inspection visuelle de la série résiduelle nous a permis de constater une nette amélioration dans la précision de modèle en passant du filtre de Kalman au lisseur de Kalman avec $L = 1$. La précision a alors augmenté plus lentement avec L. Ce qui est plus important encore, du point de vue des anomalies signalées, c'est qu'il n'y a eu

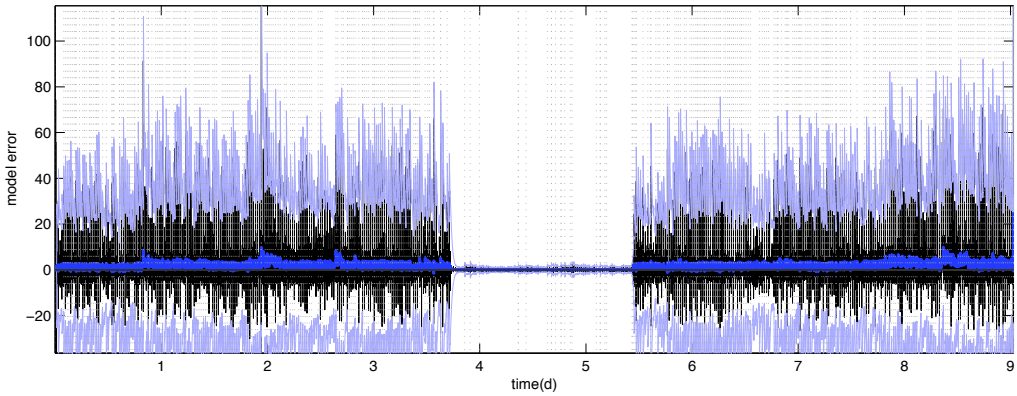
que des différences mineures entre $L = 5$ et $L = 1$.

Nous avons donc opté pour $L = 1$ pour réduire le retard de détection de la pire éventualité à deux minutes. Cela ne comprend ni les retards de calcul, ni les retards provoqués par d'autres composantes dans l'architecture de détection d'intrusions. Une explication possible est l'inadaptabilité du modèle AR pour saisir un tel comportement.

Nous présentons ensuite les expérimentations faites en appliquant la méthode AR non stationnaire aux flux d'alertes contenant plus de 1M d'alertes dans l'`ensemble-3`. Les flux ont été énumérés au tableau 3.3. Etant donné que chaque flux d'alertes contient plus de 60K d'observations obtenues sur 43 jours, ils sont plutôt volumineux. De plus, les profils des flux sont plutôt stables au fil du temps. C'est pour cette raison que nous présentons ici les détails sur les parties sélectionnées du flux seulement qui contiennent quelques anomalies intéressantes. Les exemples couvrent également un intervalle de temps suffisamment long pour montrer que le comportement normal du flux n'est pas balisé. Nous proposons deux figures pour chaque flux. Dans les figures du haut, nous avons les observations en noir, des observations lissées en bleu clair et les anomalies signées apparaissent sous forme de traits verticaux gris. Les figures du bas illustrent l'erreur de modèle en noir, la ligne de base de détection en bleu foncé, les limites de contrôle en bleu clair et les anomalies signalées en traits verticaux gris.



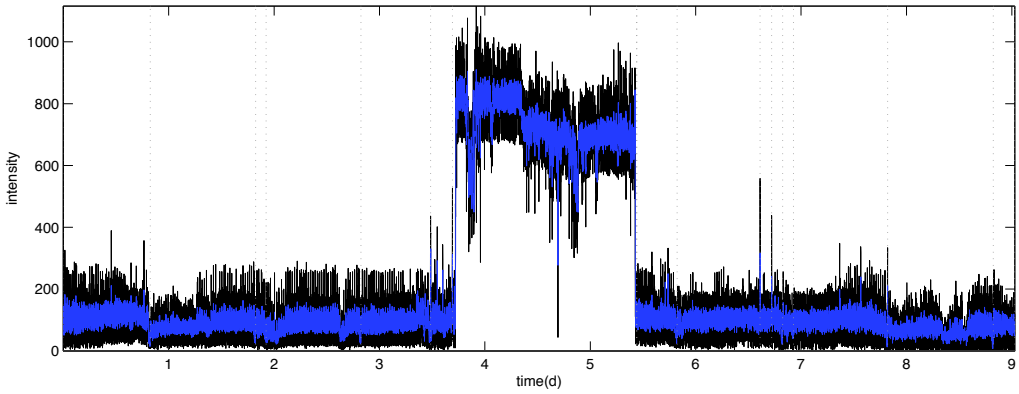
(a) Observations, smoothed observations and anomalies



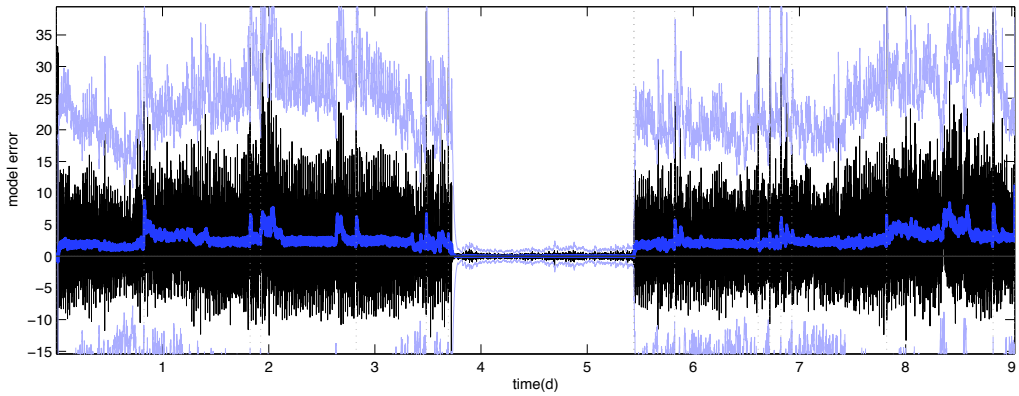
(b) Model error, control limits and anomalies

FIG. 6.9 – Flux `SNMP request udp` de l' `ensemble-3`

Le flux `SNMP request udp` était également présent dans l'ensemble-3 et les observations pour $t = 1, \ldots, 13\text{K}$ sont illustrées à la figure 6.9. Cette partie contient la seule anomalie majeure présente dans ce flux, visible du jour trois au jour cinq sous forme d'une impulsion d'alerte rectangulaire. A part cela, le flux était très similaire aux parties restantes visibles à la figure 6.9. Cette partie du flux se composait de 3 027 649 alertes et nous avons signalé 362 anomalies. Même si les anomalies signalées sont des pics dans l'intensité des alertes et que la proportion des observations anormales de toutes les observations est relativement petite, 2,8 %, la figure 6.9 montre clairement que la méthode de traitement ne convient pas pour ce flux. Le flux est rempli d'impulsions haute intensité à des intervalles très réguliers. Nombre d'entre elles sont supérieures à 500 alertes par minute (apm). La méthode de traitement balise la majorité de ces impulsions et l'opérateur serait noyé par des anomalies non intéressantes. La série d'erreurs du modèle présente une augmentation importante de la précision du modèle pendant l'impulsion d'alertes. La cause probable est une distribution plus homogène des mesures d'intensité autour de leur moyenne au cours de l'impulsion d'alertes.



(a) Observations, smoothed observations and anomalies

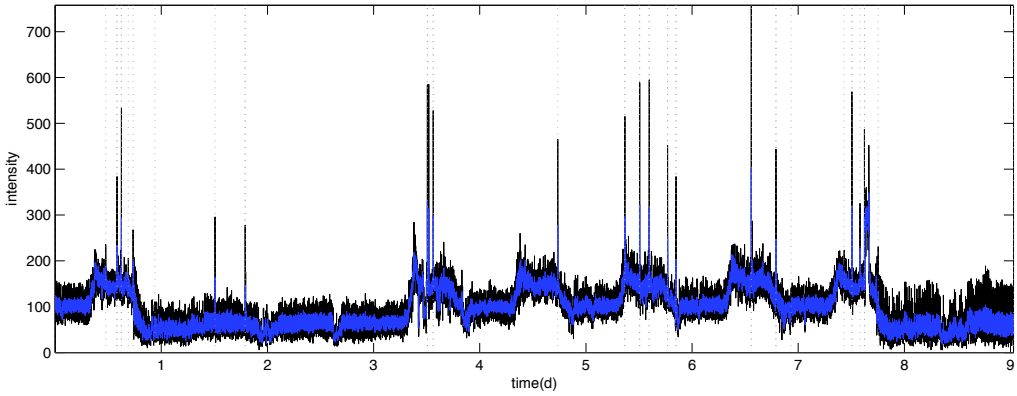


(b) Model error, control limits and anomalies

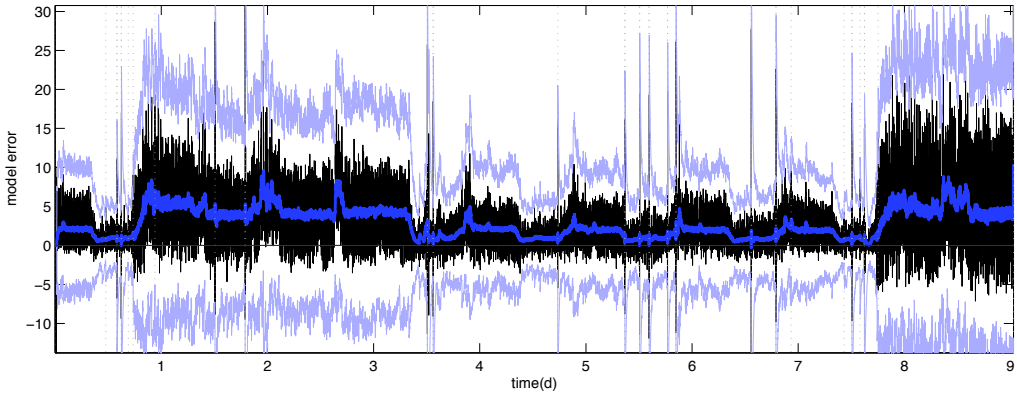
FIG. 6.10 – Flux `SNMP public access udp` de l' `ensemble-3`

Les observations du flux `SNMP public access udp` pour $t = 1, \ldots, 13\text{K}$ sont illustrées à la figure 6.10. Le flux est très similaire à `SNMP request udp`, mais l'amplitude de la composante haute fréquence est nettement plus petite. La hauteur moyenne de l'impulsion est

800 apm dans la première moitié et 700 apm dans la deuxième moitié. L'exemple contient 2 757 028 alertes et la méthode signale 16 anomalies. Il n'y a que deux doubles mais, pour le reste, les anomalies signalées sont des pics, manquant des alertes ou changements dans l'intensité de la ligne de base. La première anomalie, par exemple, est une petite chute dans l'activité de la ligne de base. La deuxième signale une chute de courte durée dans l'intensité des alertes, à peine visible à l'impression. Le bord montant de l'impulsion d'alertes n'est pas détecté, même s'il provoque l'une des erreurs négatives les plus importantes de l'exemple. Il est précédé d'un pic de 400 apm qui est balisé et, juste avant le bord, d'un pic de 550 apm. Ces pics masquent le bord montant en augmentant l'écart-type de l'erreur $\sigma_{\tilde{y}_t}$, qui est utilisé pour les limites de contrôle d'erreur. La chute de la ligne de base est signalée. L'erreur de modèle nous permet de constater une augmentation similaire de la précision au cours de l'impulsion d'alerte, comme avec `SNMP request udp`.



(a) Observations, smoothed observations and anomalies

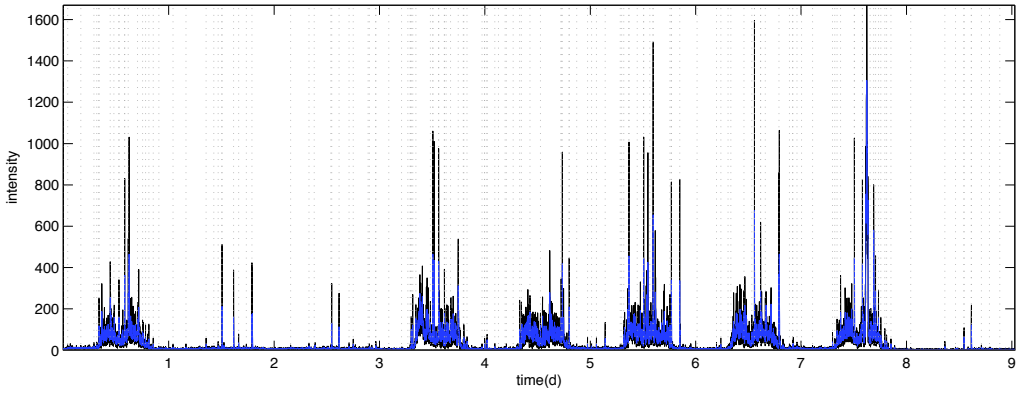


(b) Model error, baseline, control limits and anomalies

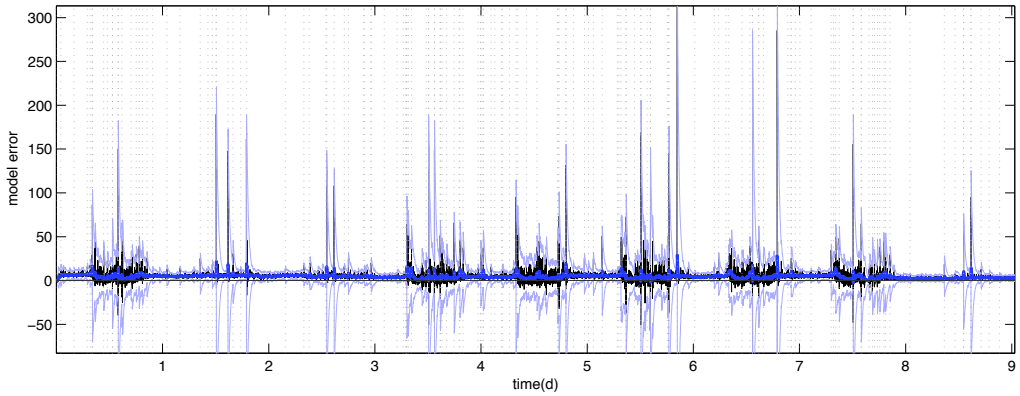
FIG. 6.11 – Flux `ICMP L3retriever Ping` de l'`ensemble-3`

Les observations du flux `ICMP L3retriever Ping` pour $t = 1, \ldots, 13\text{K}$ sont illustrées à la figure 6.11. L'exemple contient 1 269 529 alertes et la méthode signale 39 anomalies. Il s'agit clairement d'un rythme hebdomadaire dans le flux. Un week-end de faible activité est visible les jours deux et trois. Les jours ouvrables, il y a plus d'alertes en journée que pendant les nuits, qui sont toujours plus occupées que les week-ends. Les anomalies signalées sont principalement des pics similaires à des impulsions, atteignant plus de deux

fois l'activité de la ligne de base. Aucune anomalie n'est émise suite aux variations quotidiennes et hebdomadaires normales, mais certains des phénomènes anormaux sont signalés plus d'une fois.



(a) Observations, smoothed observations and anomalies

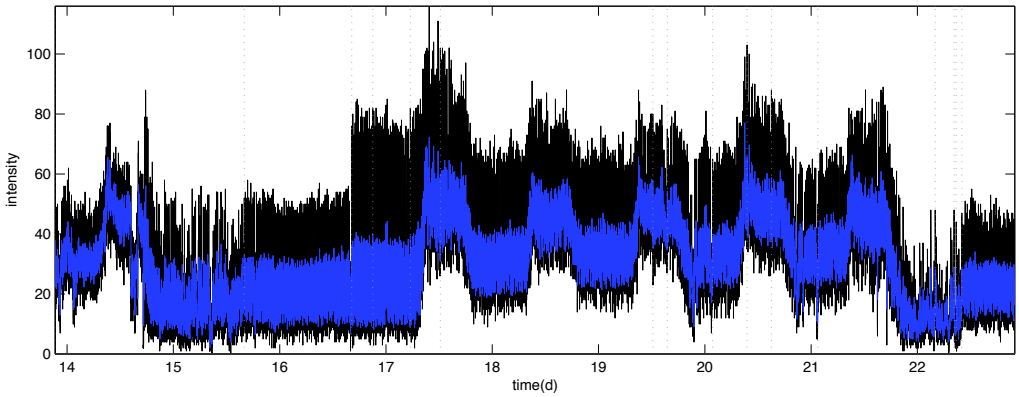


(b) Model error, baseline, control limits and anomalies

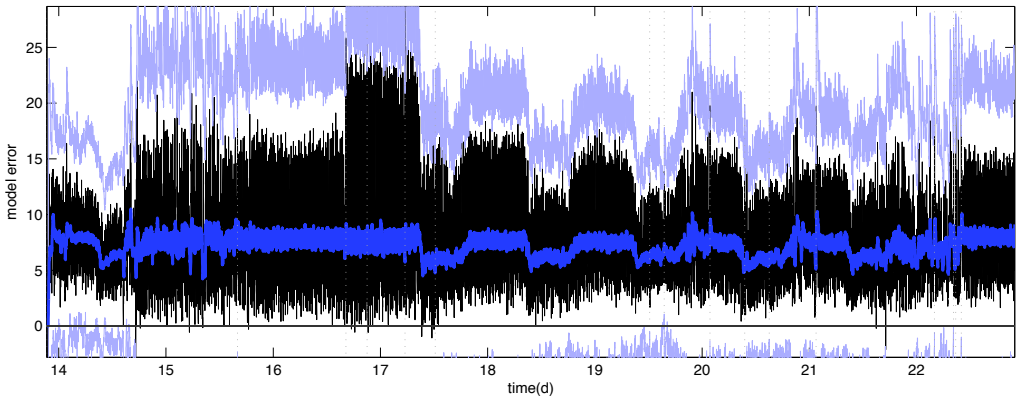
FIG. 6.12 – Flux `(http_inspect) BARE BYTE UNICODE ENCODING` de l'`ensemble-3`

Les observations du flux `(http_inspect) BARE BYTE UNICODE ENCODING` pour $t = 1, \ldots, 13\text{K}$ sont illustrées à la figure 6.12. L'exemple contient 403 508 alertes et la méthode signale 147 anomalies. Des rythmes quotidiens et hebdomadaires ressortent clairement, mais ils sont couverts par un comportement similaire à une impulsion tout au long du flux. En comparaison avec `SNMP request udp`, les impulsions sont nettement plus irrégulières aussi bien dans le temps que dans l'amplitude. Alors que la ligne de base hebdomadaire passée au filtrage passe-bas reste typiquement inférieure à 200 apm, bon nombre des impulsions sont supérieures à 600 apm, les plus élevées atteignant 1600 apm et sont souvent balisées. Les week-ends, un schéma similaire est répété à une échelle plus petite et avec une ligne de base plate. Il s'agit du deuxième flux difficile à traiter avec notre méthode, étant donné que la méthode AR non stationnaire balise une majeure partie des impulsions.

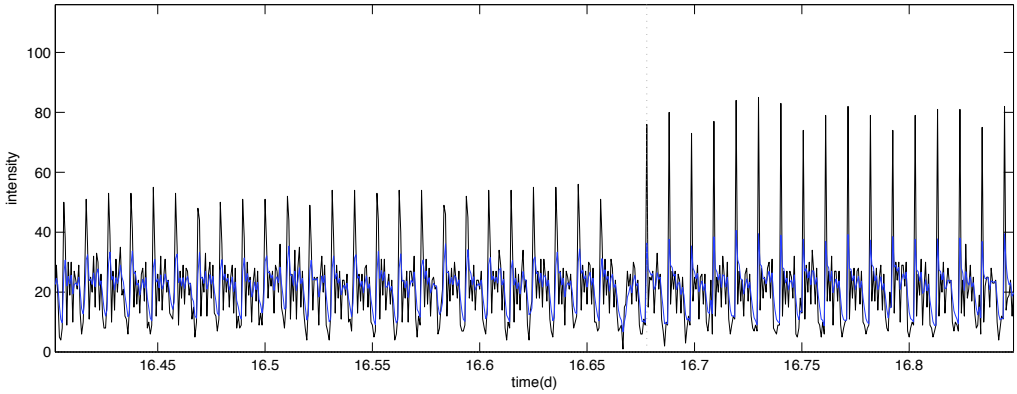
Les observations du flux `NETBIOS SMB-DS DCERPC NTLMSSP asn1 overflow attempt` pour $t = 20\text{K}, \ldots, 33\text{K}$ sont illustrées à la figure 6.13 et pour $t = 50\text{K}, \ldots, 55\text{K}$ à la figure 6.14. De nouveau, les rythmes hebdomadaires et quotidiens sont clairement visibles et suivent le même schéma que celui déjà vu. Les anomalies émises signalent principalement



(a) Observations, smoothed observations and anomalies



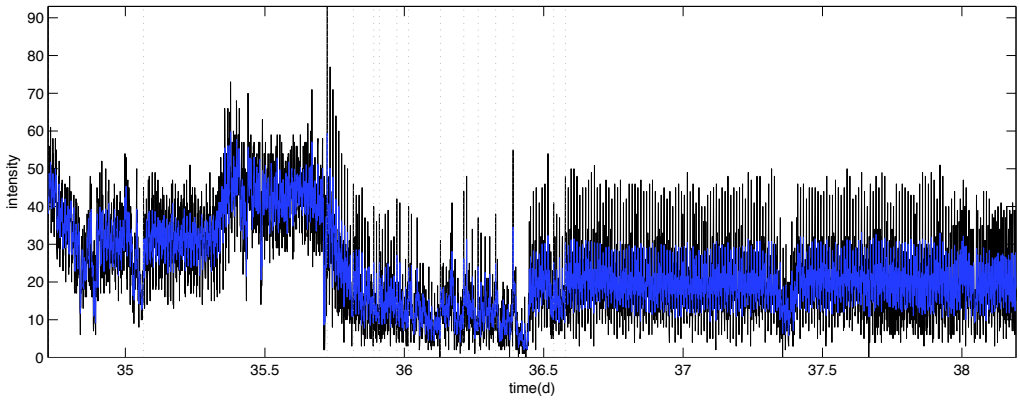
(b) Model error, baseline, control limits and anomalies



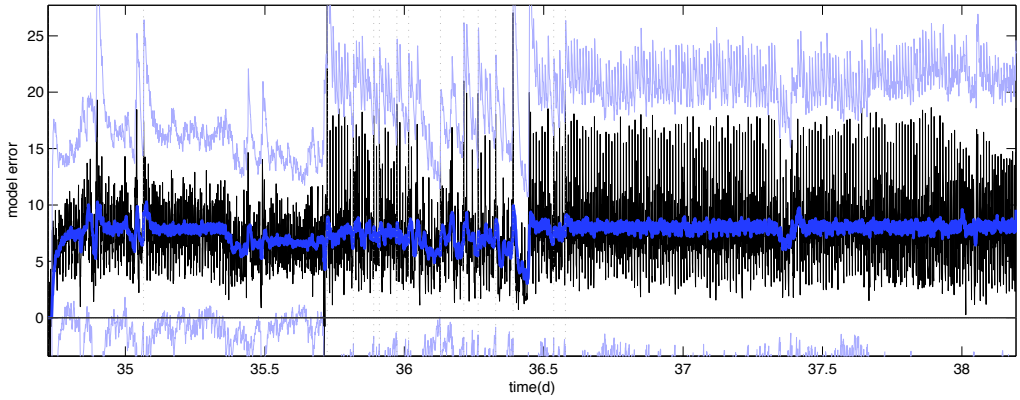
(c) Zoomed observations, smoothed observations and anomalies

FIG. 6.13 – Flux `NETBIOS SMB-DS DCERPC`, exemple 1 de l'`ensemble-3`

des changements rapides. Un bon exemple est le changement graduel qui n'est pas signalé et qui peut être constaté le soir du jour 19. L'intensité des alertes diminue très rapidement, mais s'étale toujours sur plusieurs observations. En fait, le phénomène s'étend sur plus de 70 observations. Même si la chute crée un pic dans l'erreur du modèle, elle est masquée par



(a) Observations, smoothed observations and anomalies



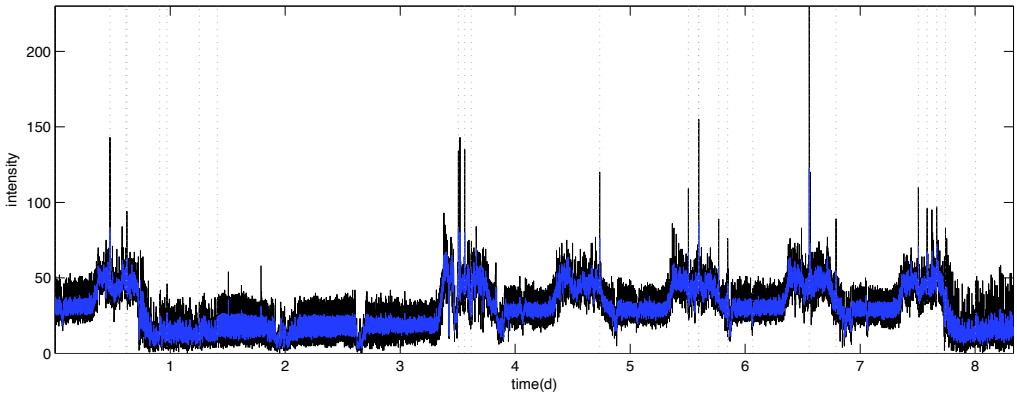
(b) Model error, baseline, control limits and anomalies

FIG. 6.14 – Flux `NETBIOS SMB-DS DCERPC`, exemple 2 de l'`ensemble-3`

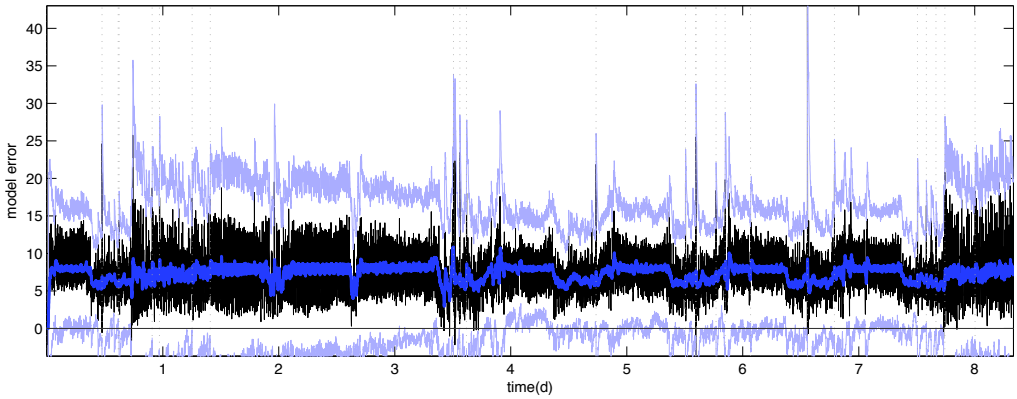
l'augmentation de la variation dans le signal d'erreur. Juste un peu plus tard, la première anomalie du jour 20 est émise suite à une chute nette de l'intensité des alertes. Cette fois, la chute est suffisamment abrupte pour être signalée. Nous avons également trois exemples de détection de changements dans le comportement de la ligne de base. La première tombe dans la première moitié du jour 15, où le flux quelque peu inégal prend une forme plus uniforme. Ensuite, environ un jour plus tard, on constate une nette augmentation de l'amplitude de la composante haute fréquence balisée par l'anomalie signalée suivante. La figure 6.13(c) zoome sur cette partie du signal, montrant la manière dont l'intensité de la composante haute fréquence toujours présente est amplifiée.

La figure 6.14 montre comment les instabilités du flux du vendredi soir (jour 35) au samedi matin (jour 36) sont balisées. Dès que le flux se stabilise le samedi après-midi, plus aucune anomalie n'est signalée. Un peu avant dimanche midi, une chute graduelle intervient dans l'intensité des alertes, chute non détectée. D'après la série résiduelle, nous pouvons voir qu'une anomalie était quasiment signalée sur le bord montant. Le changement s'étalait cependant sur trop d'observations pour être détecté, étant donné le nombre de fluctuations dans le flux d'alertes.

Les observations du flux `NETBIOS SMB-DS IPC$ share unicode access` pour $t =$



(a) Observations, smoothed observations and anomalies



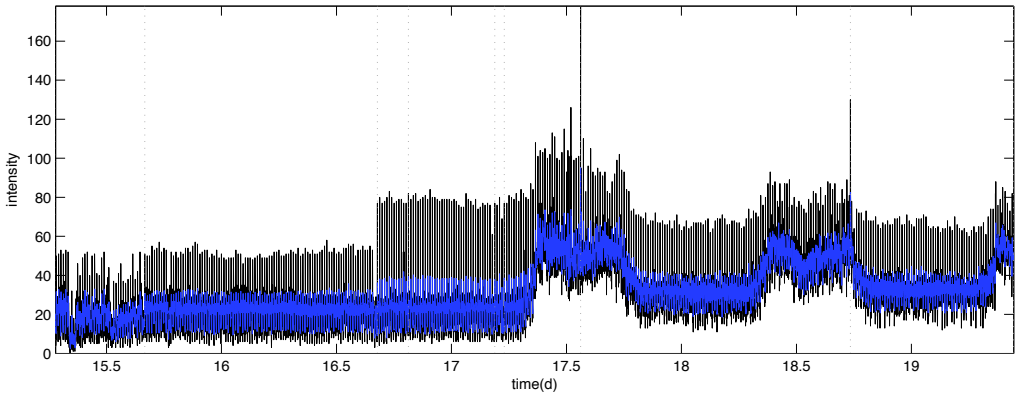
(b) Model error, baseline, control limits and anomalies

FIG. 6.15 – Flux `NETBIOS SMB-DS IPC$`, exemple 1 de l'`ensemble-3`

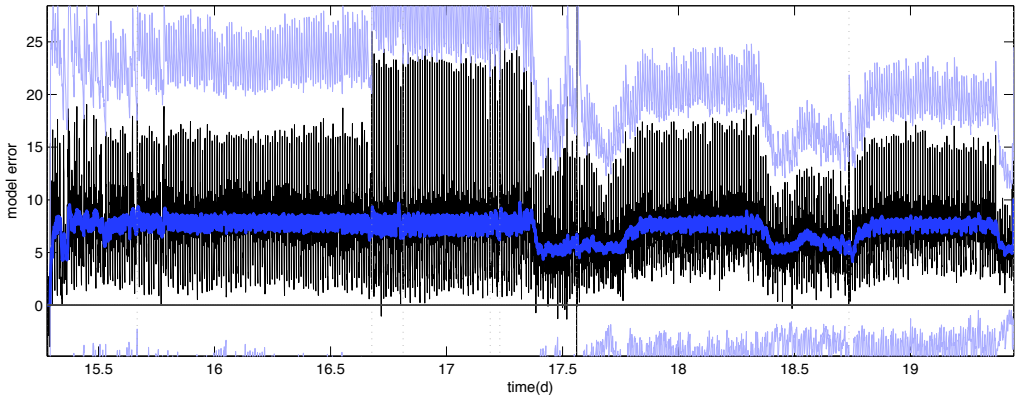
$1, \ldots, 13$K sont illustrées à la figure 6.15 et pour $t = 22\text{K}, \ldots, 28\text{K}$ à la figure 6.16. Le premier exemple contient 353 377 alertes et la méthode signale 24 anomalies. Les deux premières anomalies sont émises suite à des impulsions. La troisième est due à l'apparition d'une composante haute fréquence inégale, qui est très instable jusqu'à la sixième anomalie signalée. Cela montre qu'il est possible de détecter l'apparition et les changements dans certaines composantes du flux. Les anomalies signalées restantes sont similaires aux anomalies trouvées dans les flux précédents en termes de types d'anomalies que nous pouvons détecter.

Le deuxième exemple contient 183 354 alertes et 9 anomalies sont signalées. Nous pouvons constater l'augmentation dans l'amplitude d'une composante haute fréquence balisée dans l'après-midi du jour 15. Dans l'après-midi du jour 16, une augmentation d'amplitude similaire à celle dans le flux `NETBIOS SMB-DS DCERPC NTLMSSP asn1 overflow attempt` est balisée. Au cours de la première moitié du jour 17, deux chutes relativement petites de l'amplitude de la composante haute fréquence sont balisées. Les anomalies restantes sont des impulsions d'alertes tout à fait standard.

Les observations du flux `NETBIOS SMB IPC$ share unicode access` pour $t = 20\text{K}, \ldots, 33\text{K}$ sont illustrées à la figure 6.17. L'exemple contient 278 333 alertes et la méthode signale



(a) Observations, smoothed observations and anomalies



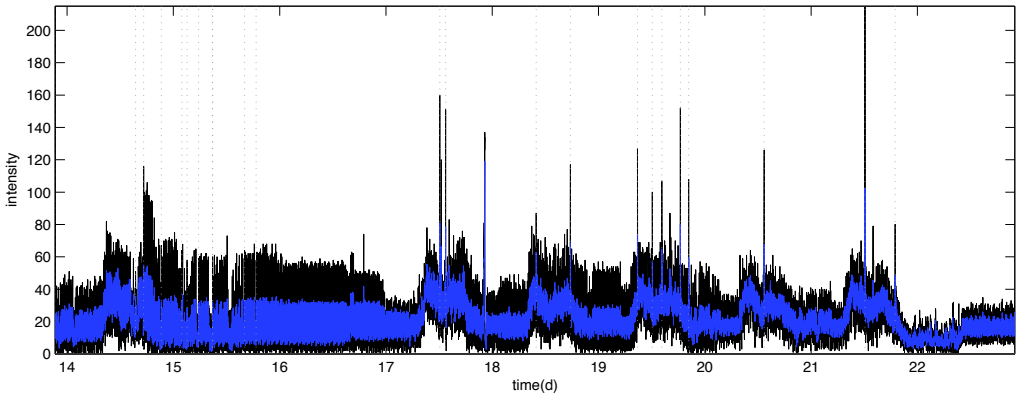
(b) Model error, baseline, control limits and anomalies

FIG. 6.16 – Flux `NETBIOS SMB-DS IPC$`, exemple 2 de l'`ensemble-3`

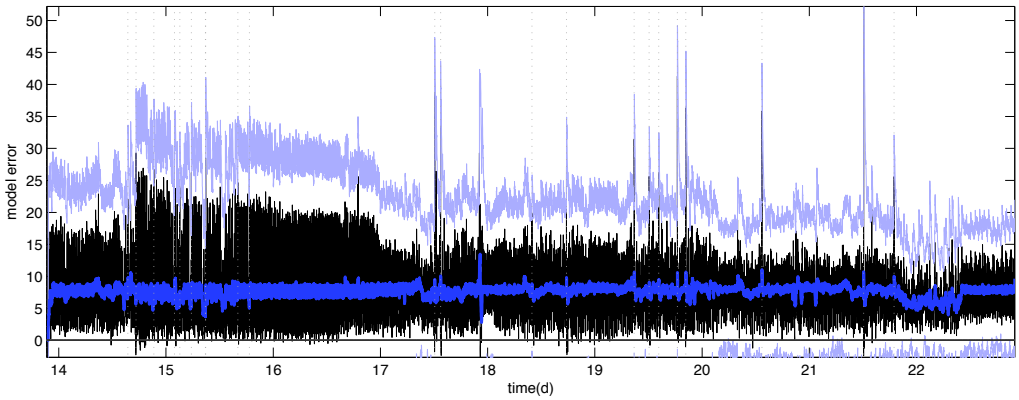
22 anomalies. Au début de l'après-midi du jour 14 jusqu'à la fin du jour 15, une composante haute fréquence instable est balisée plusieurs fois. Dès que la composante haute fréquence s'est stabilisée au cours du jour suivant, plus aucune anomalie n'est signalée. La situation est similaire à celle dans le flux `NETBIOS SMB-DS DCERPC NTLMSSP asn1 overflow attempt` aux alentours des mêmes dates. La figure 6.18 zoome sur cette partie de l'exemple. D'après la figure, nous pouvons voir que les anomalies sont en fait signalées suite à la réapparition de la composante haute fréquence et le retard réel de détection de l'anomalie est dans ce cas plus long que juste L échantillons.

Le tableau 6.2 résume les nombres à partir de ces exemples. Dans la colonne *Flux*, nous avons les noms des flux par ordre d'apparition. Les nombres d'observations et d'alertes dans les flux apparaissent respectivement dans les colonnes *Observ* et *Alertes*. La méthode signale les *An* anomalies et la proportion des intervalles anormaux est donné dans la colonne *Prop*. La dernière colonne, *OK*, indique si oui ou non la méthode AR non stationnaire avec l'ensemble de paramètres utilisé convient au traitement du flux.

Pour tous les flux, le nombre d'intervalles avec zéro alerte est petit. Le nombre le plus élevé étant pour le flux `(http_inspect) BARE BYTE UNICODE ENCODING`, 190 fois. La plupart du temps, l'intensité des alertes est de zéro moins de dix fois. L'intensité nulle



(a) Observations, smoothed observations and anomalies



(b) Model error, baseline, control limits and anomalies

FIG. 6.17 – Flux`NETBIOS SMB IPC$` de l'`ensemble-3`

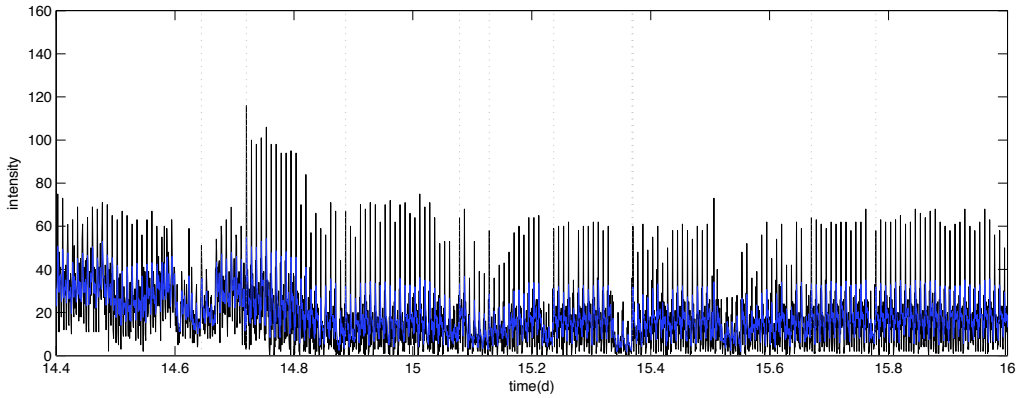
étant si rare, nous ne faisons pas de différence entre les intervalles actifs et nuls, comme dans les chapitres précédents.

Comme discuté à la section 3.2.3, lors de l'examen des intervalles d'échantillonnage dans la plage de une à vingt minutes, le comportement normal du flux est visible à tous les intervalles d'échantillonnage. Les anomalies à différents intervalles d'échantillonnage étaient de forme similaire, mais les anomalies d'échelle de temps différente n'étaient pas visibles à tous les t_s.

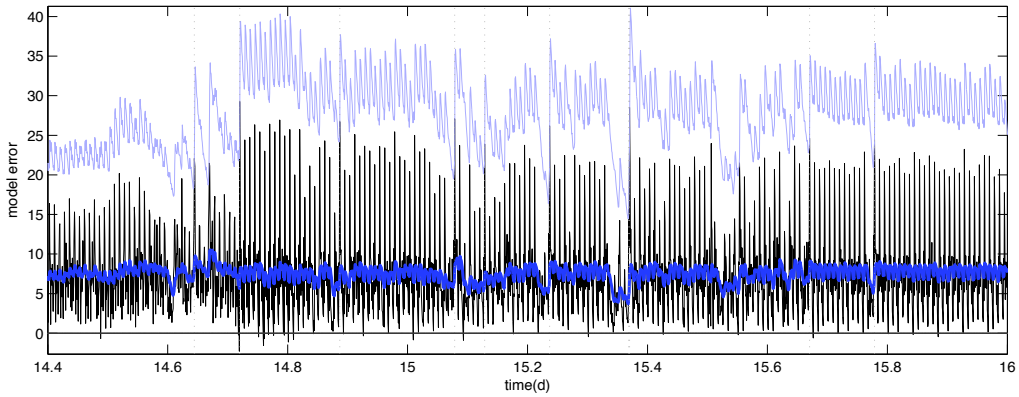
Nous n'avons utilisé que l'intervalle d'échantillonnage le plus court, étant donné que les flux avec $t_s = 1$ min ont les fluctuations haute fréquence visibles et perturbent le processus de détection des anomalies.

Pour démontrer comment les anomalies de différentes échelles de temps peuvent être détectées, la figure 6.19 montre le flux `NETBIOS SMB IPC$ share unicode access` à des intervalles d'échantillonnage plus importants, 20 minutes à la figure 6.19(a) et 60 minutes à la figure 6.19(c).

La figure 6.19(b) zoome sur le flux d'alertes à un intervalle d'échantillonnage de 20 minutes. La même partie du flux est montrée à un intervalle d'échantillonnage d'une minute à la figure 6.17(a). Nous ne montrons pas entièrement le flux d'alertes à un intervalle d'échantillonnage d'une minute, en raison de la grande taille de l'image. Les observations



(a) Observations, smoothed observations and anomalies

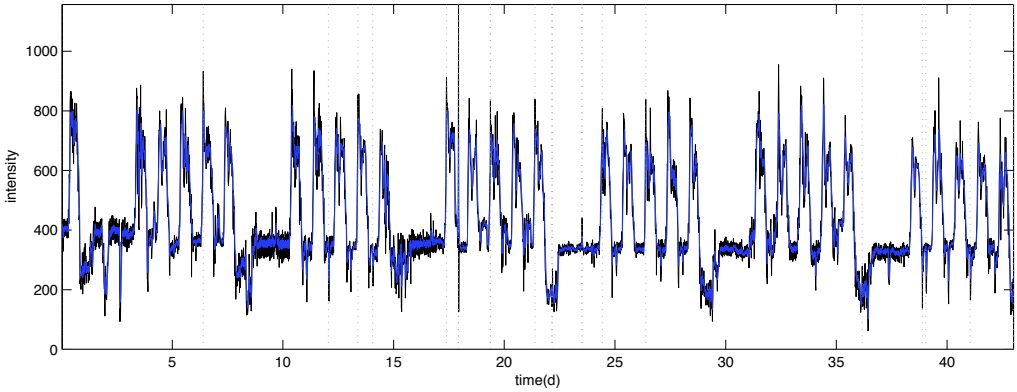


(b) Model error, baseline, control limits and anomalies

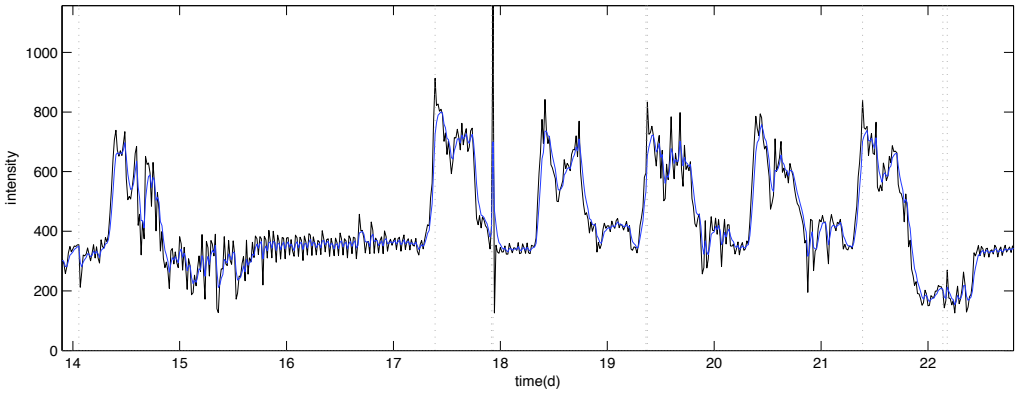
FIG. 6.18 – Zoom sur le flux `NETBIOS SMB IPC$` de l'`ensemble-3`

sont illustrées en noir, les observations passées au filtrage passe-bas sont en bleu et les anomalies signalées sont représentées par des traits verticaux gris. Tous les autres paramètres de la méthode de traitement sont restées les mêmes qu'avant, nous avons seulement augmenté t_s.

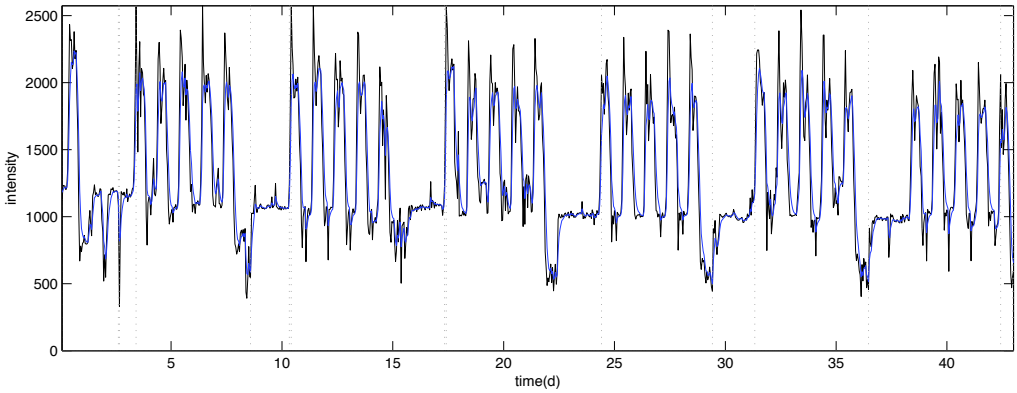
En examinant les jours 14-22, on peut voir que le comportement normal est visible aux trois intervalles d'échantillonnage, mais les anomalies ne sont pas les mêmes, elles sont seulement de forme similaire. A la figure 6.19(b), la première anomalie est émise suite à une chute dans l'intensité des alertes, ce qui n'est pas visible avec $t_s = 1$ min à la figure 6.17(a). De manière similaire, des anomalies d'échelle plus longue interviennent à la fin du jour 17 et au début du jour 22, non visible avec $t_s = 1$ min. Des anomalies sont également signalées à $t_s = 60$ min. Par exemple, au cours du jour trois, une chute de l'intensité des alertes est balisée et, au cours du jour 37, une nette augmentation est balisée après une période de faible intensité. D'un autre côté, ce sont des anomalies de courte durée au cours des jours 19 et 20 qui sont lissées à des intervalles d'échantillonnage plus longs. Les anomalies sont des changements similaires à des impulsions dans l'intensité des alertes ; même si elles peuvent se manifester comme de petites ondulations à d'autres intervalles d'échantillonnage, elle tendent à former un changement abrupt à t_s proche de leur échelle de temps.



(a) Observations, smoothed observations and anomalies with $t_s = 20$min



(b) A zoom in to the flow with $t_s = 20$min



(c) Observations, smoothed observations and anomalies with $t_s = 60$min

FIG. 6.19 – Flux `SNMP request udp`

TAB. 6.2 – Volumes de flux et anomalies signalées pour les flux de l'`ensemble-3`

Flux	Observ (10^3)	Alertes (10^6)	An	Prop (10^{-3})	OK
SNMP request	13	3.0	362	28	NOK
SNMP public	13	2.8	16	1.2	OK
L3retriever	13	1.3	39	3.0	OK
(http_inspect)	13	0.4	147	11	NOK
DCERPC 1	13	0.4	15	1.2	OK
DCERPC 2	5	0.1	14	2.8	OK
SMB-DS IPC$ 1	13	0.4	24	1.8	OK
SMB-DS IPC$ 2	4	0.2	9	2.3	OK
SMB IPC$	13	0.3	22	1.7	OK

6.4 Discussion

Même si nous perdons la visibilité des événements de plus de 20 minutes avec une méthode AR non stationnaire, $p = 20$ et $t_s = 1$ min, nous sommes capables de saisir remarquablement bien les comportements quotidiens et hebdomadaires. La non-stationnarité du modèle en est une des raisons, l'autre étant l'adaptabilité de l'algorithme d'estimation. Le court intervalle d'échantillonnage rend également la situation plus facile, étant donné que les changements sont dispersés sur plusieurs observations et que le modèle est capable de s'adapter à des changements suffisamment lisses. D'un autre côté, cela signifie qu'il se peut que nous manquions des anomalies qui provoquent des changements d'intensité à un taux de changement similaire.

La méthode est capable de baliser des pics et creux relativement petits mais abrupts dans les flux d'alertes. Il est toutefois important de noter que de nombreux phénomènes anormaux passent sans être détectés. A un taux d'échantillonnage d'une minute, les flux fluctuent beaucoup. Pour éviter le balisage de toutes ces variations normales, il faut relâcher le seuil de détection en augmentant n et, par exemple, en soumettant les observations au filtrage passe-bas avant de les analyser. Ces deux contre-mesures en matière de bruit augmentent le risque de perdre quelques phénomènes anormaux. La détection n'est pas déterministe, nous ne pouvons pas garantir la détection de certains types d'anomalies. La détection dépend beaucoup de la structure du flux et en particulier des observations précédentes. En général, toutefois, la plupart des anomalies sont des changements similaires à des impulsions et la méthode de détection est très efficace pour les relever.

Des sept flux analysés ici, les paramètres d'algorithme utilisés ont fonctionné dans cinq cas, et également à différentes échelles de temps. Des deux restantes, dans le cas de `(http_inspect) BARE BYTE UNICODE ENCODING`, le problème est la nature du flux d'alertes. Le flux est rempli de pics similaires à des alertes. Pour l'autre flux, `SNMP request udp`, il a été facile de trouver un ensemble de paramètres d'algorithme de manière à ce que la composante haute fréquence problématique ne soit plus signalée comme une anomalie.

Pour les autres flux également, il est possible de mieux adapter les paramètres de la méthode aux spécificités de chaque flux. Ces spécificités découlent 1) du type et de l'amplitude des fluctuations haute fréquence et 2) de la puissance et de la stabilité des rythmes hebdomadaires et quotidiens. Il y a toutefois toujours un compromis à consentir. Si nous souhaitons baliser moins de bizarreries du comportement normal dans un flux inégal et instable, il se peut que nous manquions en même temps quelques phénomènes

intéressants. De manière typique, des variations puissantes antérieures - normales ou anormales - peuvent masquer des comportements intéressants du flux. Le risque augmente lorsque nous devons intégrer des variations plus nombreuses et plus puissantes dans le comportement normal.

Dans tous les cas, les ajustements ont été réalisés à l'aide d'un échantillon de seulement quelques jours couvrant le comportement aussi bien du week-end que de la semaine. Ce serait donc possible dans la pratique en collectant des données d'entraînement dans le système surveillé. Nous accordons toutefois plus d'importance à la possibilité d'utiliser des paramètres généraux qui fonctionnent relativement bien avec la plupart des flux. Cette généralité signifie un déploiement plus direct dans la pratique.

Lorsque l'on compare les observations et les valeurs résiduelles, nous pouvons voir que dans le cas de `ICMP L3retriever Ping` et `NETBIOS SMB-DS DCERPC NTLMSSP asn1 overflow attempt`, la série résiduelle ressemble à la série d'alertes. On pourrait se demander pourquoi appliquer le lisseur de Kalman si, en fin de compte, nous analysons encore des signaux similaires. Premièrement, ce n'est pas un cas général. Deuxièmement, avec ces flux aussi, même si des variations hebdomadaires sont présentes dans les erreurs de modèles, elles sont significativement atténuées et, de ce fait, la série résiduelle est plus facile à analyser. Un autre exemple de séries d'alertes et résiduelles similaires est repris à la figure 6.18 pour le flux `NETBIOS SMB IPC$ share unicode access`. Dans cet exemple, l'application directe de la détection EWMA aux observations pourrait provoquer des anomalies quasiment ou tout à fait identiques. Ce n'est toutefois vrai que pour cette partie de la série. Si l'on examine l'intégralité de l'exemple (figure 6.17) et en particulier la série résiduelle, nous voyons la différence entre la série d'alertes et la série résiduelle pendant la semaine. Comme nous l'avons vu aux chapitres précédents, le modèle de tendance n'est pas capable de traiter ce type de variation et la méthode AR stationnaire a ses propres revers.

Rien que par leur définition, les deux flux `IPC$ access` sont corrélés. Les graphes d'intensité permettent de voir qu'il existe également de fortes similitudes entre les autres flux, comme `NETBIOS SMB-DS DCERPC NTLMSSP asn1 overflow attempt` et `ICMP L3retriever Ping`. Par exemple, les jours 15 et 16 se ressemblent dans les flux `NETBIOS SMB IPC$ share unicode access` et `SMB-DS DCERPC NTLMSSP asn1 overflow attempt`. Ce dernier et le flux `NETBIOS SMB IPC$ share unicode access` présentent tous deux une augmentation similaire dans l'amplitude de la composante haute fréquence l'après-midi du jour 16. Les séquences de corrélation croisée entre les flux avec des rythmes hebdomadaires présentent des pics locaux aux alentours des pics correspondant à des jours entiers et à une semaine. La corrélation est importante, en particulier entre les flux `NETBIOS et le flux ICMP L3retriever Ping`.

Ces types de corrélation pourraient aider à examiner les anomalies signalées. Si l'anomalie est présente dans plus d'un flux corrélé, le problème est plus large, et s'il n'est présent que dans un seul flux, il s'agit plus probablement d'une anomalie isolée dans le type d'activité surveillée par le flux. L'utilisation de ces types de corrélations en analyse dépend toutefois fortement du système et du flux. Nous n'avons pas examiné ce problème en détail. Les corrélations *entre* les flux font partie du domaine d'autres méthodes de corrélation, comme le travail de Qin et Lee avec le test de causalité de Granger [QL03] qui analyse les relations statistiques entre les séries d'alertes.

6.4.1 Limitations des algorithmes des modèles et des estimations

On pourrait avancer que le modèle AR est trop limité pour la modélisation du type de flux que nous avons dans nos données. Les erreurs faisant suite à des changements rapides dans nos observations sont une manifestation des limitations du modèle AR et des modèles non linéaires, par exemple, pourrait assurer une meilleure réactivité aux changements rapides[2]. Nous avons deux raisons de nous en tenir aux modèles linéaires : 1) Etant donné notre idée sous-jacente, nous ne souhaitons même pas modéliser le flux d'alertes de manière exacte ; les changements rapides en particulier devraient être exclus du modèle. 2) Les modèles simples signifient des algorithmes plus simples, ce qui se traduit par des applications plus rapides et plus faciles, ce qui est un aspect important pour le déploiement de la méthode.

Si nous avions attribué des données pour le test et l'entraînement ou nettoyé des données normales, nous pourrions procéder à une analyse plus approfondie de l'effet des différents paramètres de l'algorithme :

Degré du modèle AR p pourrait être sélectionné à l'aide, par exemple, des critères d'informations AIC ou BIC [BJR94, pp.200-202]. Ces critères essaient d'équilibrer le compromis entre l'augmentation de la précision et la complexité du modèle lorsque p augmente. Selon Tarvainen [Tar04] , dans la pratique, le degré du modèle est souvent fixé en fonction de certaines connaissances ou directives préalables. Dans notre cas, avec de longs intervalles d'échantillonnage, nous souhaitions un modèle utilisant des observations datant de plus d'un jour. D'un autre côté, nous devons limiter le degré du modèle, étant donné que le nombre de flux à traiter peut être important et, en particulier à des intervalles d'échantillonnage petits, le lisseur de Kalman doit être exécuté plus souvent pour mettre à jour les estimations.

Largeur de la limite de contrôle n pourrait être défini sur un taux acceptable de faux positifs. La méthode de traitement pourrait être exécutée sur un ensemble de données normal connu pour trouver la valeur n, de manière à ce que les variations normales ne provoquent pas plus de faux positifs que ce qui est acceptable.

Facteur de covariance du bruit d'état σ_w^2 pourrait être défini de manière à assurer un équilibre adéquat dans l'adaptation au comportement normal et à éviter d'intégrer le comportement anormal dans le modèle.

En plus du modèle choisi, l'adaptabilité et l'adéquation sont également largement affectées par la méthode d'estimation des paramètres. D'autres algorithmes adaptatifs pourraient être utilisés à la place du filtre et du lisseur à décalage fixe de Kalman. Par exemple, dans les applications de biosignal, les estimations par les moindres carrés moyens (LSM) et les moindres carrés récursifs (RLS) sont populaires [Tar04]. Nous ne prétendons pas que le filtre/lisseur de Kalman présente le meilleur rapport qualité/prix, juste qu'il est suffisamment bon pour prouver que, dans l'ensemble, il est possible de modéliser le comportement normal du flux d'alertes avec des modèles de séries temporelles linéaires non stationnaires et des algorithmes adaptatifs. Nous avançons également que, comme nous manquons de bonnes données de test, il est 1) très difficile et 2) plutôt inutile de passer du temps à optimiser de tels facteurs. Les valeurs optimales peuvent varier d'un environnement à l'autre et en fonction des besoins de l'utilisateur. Les paramètres devraient par conséquent être fixés dans la phase de déploiement.

[2]Communication personnelle, David Mercier, Laboratoire Electronique et Traitement du Signal, Commissariat à l'Energie Atomique, 14.02.2006.

Alternativement, si des ensembles de données de bonne qualité deviennent disponibles, on pourrait évaluer les capacités du système actuel ou essayer de trouver des algorithmes et modèles d'estimation optimum, et les comparer à d'autres méthodes de corrélation. En plus des critères théoriques d'estimation et d'information classiques, il existe également des métriques spécialement conçues pour les systèmes de détection d'intrusions. La *capacité de détection d'intrusions* [GFD+06] pourrait être utilisée pour choisir le point de fonctionnement. Pour le moment, toutefois, le réglage fin du système ne semble pas avoir beaucoup de sens.

Un des problèmes de la méthode est sa nature à échelle unique. Le type d'anomalies que la méthode détecte est identique à différents intervalles d'échantillonnage. On peut le voir en comparant les résultats de l'`ensemble-1` et de l'`ensemble-3`, qui ont été obtenus avec différents intervalles d'échantillonnage. Dans les deux cas, la méthode met en évidence des changements rapides et des changements typiques de courte durée ressortent généralement mieux que des anomalies de longue durée. Au chapitre 3, nous avons mentionné que des anomalies de différentes échelles de temps avaient tendance à être plutôt abruptes à certains intervalles d'échantillonnage. Par conséquent, en utilisant des intervalles d'échantillonnage différents, nous pouvons détecter les anomalies dans les échelles de temps qui nous intéressent. Nous avons donc utilisé trois intervalles d'échantillonnage différents, $t_s = 1, 20, 60$ minutes. A des petits intervalles d'échantillonnage, le modèle ne saisit en fait pas les rythmes normaux avec de longues périodes, comme une semaine ou même un seul jour, à moins d'augmenter significativement le degré du modèle p, de manière à ce que $p \cdot t_s$ couvre la fenêtre de temps suffisamment longue. Pour des raisons de calcul, cela n'est pas faisable. En même temps, la méthode s'adapte suffisamment bien à ces rythmes pour ne pas déclencher trop d'anomalies non pertinentes.

Avec ces trois intervalles d'échantillonnage, nous ne détectons pas des changements très lents, comme les changements graduels dans les comportements hebdomadaires. Par exemple, pour des raisons d'ingénierie de réseau, la détection de tels changements dans le trafic général peut être cruciale pour anticiper la croissance de la charge du réseau. Nous affirmons toutefois que ces variations dans le type d'activité couverte par les signatures informatives sont moins intéressantes pour la surveillance de la sécurité du réseau. Si les comportements à long terme doivent être modélisés ou les variations à long terme, détectées, la nature multi-échelle de l'analyse en ondelettes pourrait se révéler utile.

6.4.2 Détection d'anomalies

Comme les modèles AR sont relativement bien adaptés au lisseur à décalage fixe de Kalman, nous devons reconsidérer notre idée sous-jacente de modélisation du comportement normal. L'estimation adaptative des paramètres avec des données souillées ajoute des caractéristiques anormales dans le modèle AR. C'est l'un des revers de l'utilisation du filtrage et du lissage de Kalman. Le problème existe également avec les modèles stationnaires, mais dans une moindre mesure. La vitesse d'adaptation des algorithmes contrôle l'équilibre entre la certitude des estimations précédentes et les observations actuelles. Avec un filtre ou lisseur hautement adaptatif en particulier, nous risquons de modéliser des anomalies de courte durée dans le modèle AR. Les limitations du modèle AR nous protègent toutefois dans une certaine mesure, étant donné que la plupart des anomalies sont des changements intermittents et abrupts, de sorte que le modèle AR ne les saisit pas correctement.

Etant donné que nous pouvons également modéliser le flux, comme les expérimentations nous l'ont montré, nous pourrions aussi baser la détection directement sur les caractéristiques

du signal et non sur les valeurs résiduelles du modèle. Comme l'analyse du domaine temporel est difficile et que les comportements réguliers semblent avoir des périodes constantes dans la plage de quelques minutes à une semaine, l'analyse du domaine fréquentiel pourrait être utile. Le comportement normal du flux, comme les rythmes quotidiens et hebdomadaires, sont probablement visibles aux basses et très basses fréquences correspondantes. Les petites fluctuations visibles dans tous les flux de l'`ensemble-3` devraient intervenir dans la bande haute fréquence. Enfin, des changements intermittents rapides pourraient être visibles sur tout le spectre. Ces différentes propriétés d'apparition pourraient être utilisées pour classer les anomalies signalées.

6.5 Conclusion

Dans ce chapitre, nous avons présenté une troisième approche pour la modélisation et le filtrage d'alertes non pertinentes. L'idée de base est la même que dans les chapitres précédents. Les régularités et les changements lisses dans l'intensité des alertes des flux d'alertes sont considérés comme des échos de l'utilisation normale du système. Ce comportement normal n'est pas observable au niveau de l'alerte, raison pour laquelle nous surveillons les flux d'alertes et modélisons le flux normal pour l'éliminer par filtrage.

Nous utilisons un modèle AR non stationnaire et estimons ses paramètres avec un algorithme de lisseur à décalage fixe de Kalman. Cette combinaison offre un modèle adaptatif et une augmentation significative de la précision du modèle en comparaison avec les approches précédentes.

Les anomalies sont détectées comme auparavant, à partir de séries résiduelles obtenues comme la différence entre les observations et les prévisions one-step-ahead du modèle. Nous signalons des erreurs qui sont importantes par rapport à la moyenne et la variance récentes dans la série d'erreurs en utilisant une carte de contrôle EWMA modifiée.

La méthode élimine efficacement par filtrage les alertes liées au comportement normal du flux. Même si avec de plus petits intervalles d'échantillonnage, le modèle perd de sa visibilité au cours des rythmes quotidiens et hebdomadaires, il peut s'adapter à ces changements étant donné qu'ils sont suffisamment lisses du point de vue des observations. Par exemple, à un intervalle d'échantillonnage d'une minute, les changements provoqués par l'utilisation normale se répartissent sur plusieurs observations et le modèle s'adapte à ces changements graduels, tandis que les mêmes changements à un intervalle d'échantillonnage d'une heure apparaissent plus abrupts.

Etant donné que les flux contiennent des variations significatives, dont une partie est le comportement normal du flux, les anomalies signalées par la méthode doivent être considérées comme des phénomènes intéressants qui valent la peine d'être soumis à un *examen plus approfondi*, autrement dit une composante humaine est nécessaire avant toute action supplémentaire. Dans ce sens, la méthode est capable de trier la majorité des intervalles de temps, laissant 0,2-0,3 % à l'analyste, ce que démontrent les exemples illustrés. Il s'agit d'une amélioration importante, étant donné que tous les flux contiennent des alertes pendant presque tous les intervalles et qui peuvent être classées comme faisant partie du comportement normal du flux.

L'adaptabilité du lisseur de Kalman augmente le risque de modélisation du comportement anormal dans le modèle. Sur la base des données observées, nous affirmons que les anomalies dans les flux d'alertes sont souvent des changements similaires à des impulsions. Le modèle AR linéaire n'est pas bien adapté à la saisie d'un tel comportement et cela limite le risque de modéliser un comportement indésirable.

Si le risque est considéré comme trop important, nous proposons d'explorer la possibilité d'analyser la densité spectrale instationnaire plutôt que le signal résiduel. Nous pensons que les rythmes naturels que nous avons vus dans le flux d'alertes pourraient fournir des indications pour choisir les bandes de fréquence à surveiller pour détecter différents types d'alertes. Pour poursuivre notre recherche, nous aimerions faire des expérimentations avec les méthodes de détection basées sur les représentations temps-fréquence, en commençant par la densité spectrale instationnaire et en continuant, au besoin, par l'analyse en ondelettes. Alternativement, des signatures statistiques plus sophistiquées telles que celles présentées dans [SLO+06] et basées sur les modèles Gamma FARIMA pourraient être une deuxième option. L'analyse en ondelettes et les modèles Gamma FARIMA pourraient offrir de meilleures capacités de détection mais, en même temps, ils consomment plus de ressources et/ou exigent plus de données pour être efficaces. Etant donné que les comportements périodiques générant une grande majorité d'alertes dans ces flux suivent des périodes naturelles, comme des jours et des semaines, nous pensons que l'analyse sur quelques échelles offre un bon compromis entre la complexité, les coûts et la capacité de détection en comparaison avec une réelle analyse multi-échelle.

Chapitre 7

Comparaison

Etant donné que les trois méthodes présentées, EWMA, stationnaires et non stationnaires, constituent un certain continuum, nous avons attendu de les avoir décrites toutes les trois avant de procéder à leur comparaison. Dans ce chapitre, nous résumerons ce que nous avons vu jusqu'à présent, puis comparerons les trois méthodes entre elles.

Nous discuterons également de problèmes plus généraux liés à l'estimation et à la précision du modèle.

7.1 Résumé

Premièrement, un bref résumé de l'analyse du flux d'alertes et des méthodes utilisées.

Alertes informatives Au chapitre 2, nous avons vu comment les différents types d'alertes contribuent à l'inondation d'alertes. Il peut s'agir d'alertes aussi bien intrusives qu'informatives, et chaque alerte peut être un vrai positif, un positif non pertinent ou un faux positif. Au départ, les systèmes de détection d'intrusions ont été développés pour détecter uniquement les intrusions, mais ils sont désormais utilisés à d'autres fins. La surveillance de la conformité à la politique de sécurité et de l'utilisation générale du système sont quelques exemples d'utilisation générant des alertes informatives.

Traditionnellement, les alertes ont été classées comme vrais ou faux positifs, mais en particulier avec les nouvelles utilisations des capteurs, nous avons le sentiment qu'il est nécessaire d'introduire une division à granularité plus fine. Nous considérons comme vrais positifs, les alertes qui sont émises avec précision suite à une violation de la politique de sécurité, et nécessitant une réaction immédiate de la part de l'opérateur. En comparaison avec la définition classique du vrai positif, nous ajoutons l'exigence d'un comportement intrusif et la nécessité d'une réaction immédiate. Nous considérons les alertes qui sont émises de manière imprécise comme faux positifs.

Nous utilisons une troisième classe, des positifs non pertinents, pour les alertes qui sont émises de manière précise mais qui ne requièrent pas l'attention immédiate de l'opérateur ou qui, prises individuellement, sont insignifiantes. Cette classe contient principalement deux types d'alertes. Appartiennent à celle classe premièrement, les alertes émises avec précision suite à une activité intrusive qui a toutefois échoué pour une raison quelconque. L'échec peut être causé, par exemple, par une topologie de réseau qui entraîne l'élimination du paquet intrusif ou la vulnérabilité inexistante sur la cible. L'autre type est constitué d'alertes émises avec précision suite à une activité non intrusive, trafic SNMP ou IM. Ces alertes sont liées à la surveillance de

la politique et de l'utilisation du système et les événements sous-jacents sont très probablement détectés correctement, mais les alertes individuelles n'ont aucune pertinence. Une interdiction stricte de ce trafic pourrait rendre ces alertes pertinentes, mais dans de nombreux cas il est attendu et/ou toléré jusqu'à une certaine mesure et entre certains hôtes.

Des classifications similaires ont été évoquées dans les discussions de la liste de diffusion (voir chapitre 2) et proposées par Kruegel et Robertson [KR04].

Caractéristiques du flux d'alertes. Après la classification générale au chapitre 2, nous avons analysé les flux d'alertes réels du chapitre 3. Nous avons identifié les types et les classes de signatures les plus prolifiques, et démontré la nécessité de surveiller de telles alertes comme des flux plutôt que comme des alertes individuelles. Nous avons vu que les positifs non pertinents informatifs sont des responsables majeurs de l'inondation d'alertes et, étant donné la difficulté de les traiter avec des approches de corrélation existantes, ils sont notre principale cible pour le traitement. Nous appelons ces alertes le *bruit des alertes*.

Nous avons démontré que, dans les données analysées, le comportement normal du flux est souvent régulier et stationnaire. Le comportement normal est également visible dans les flux obtenus avec tous les intervalles d'échantillonnage utilisés, d'une minute à quelques heures. Les fluctuations aléatoires et haute fréquence deviennent plus visibles à des intervalles d'échantillonnage plus courts et font partie du comportement normal. Par contre, les anomalies de différentes échelles sont devenues plus clairement visibles à des intervalles d'échantillonnage proches de l'échelle de l'anomalie.

A des intervalles d'échantillonnage plus importants, par exemple une heure et plus, des anomalies à court terme peuvent être lissées, en fonction du comportement général du flux. Des anomalies à court terme apparaissent également à des intervalles d'échantillonnage plus importants dans les flux constants ou si la densité de l'anomalie est forte. De nouveau, l'intervalle d'échantillonnage utilisé devrait refléter les besoins de l'opérateur. Des intervalles d'échantillonnage plus petits permettent de détecter des anomalies d'intensité plus petite et de plus courte durée au prix d'une augmentation du nombre d'anomalies signalées. Si l'on s'intéresse uniquement aux changements majeurs et aux pics de l'intensité du flux, des intervalles d'échantillonnage plus larges suffisent.

Les anomalies intéressantes sont souvent des pics et chutes intermittents et abrupts dans le signal. Des anomalies de différente durée deviennent visibles à différentes échelles de temps, mais dans des formes similaires.

En conséquence, le problème de la détection est similaire à différentes échelles de temps, la différence la plus significative étant l'amplitude des fluctuations haute fréquence. En d'autres termes, la modélisation et la détection deviennent plus difficiles à des petits intervalles d'échantillonnage, étant donné que les variations aléatoires sont plus significatives.

Modèle des tendances. Au chapitre 4, nous avons vu l'utilisation de la modélisation des tendances avec des moyennes glissantes pondérées exponentiellement. Le modèle était de toute évidence le plus simple de ceux que nous avons utilisé et ses capacités de modélisation sont quelque peu limitées. Toutefois, il a fonctionné relativement bien avec certains flux, il est facile à appliquer et compréhensible pour l'utilisateur. En utilisant des facteurs de lissage relativement petits par rapport à l'utilisation

traditionnelle de la carte de contrôle, notre modèle suit seulement des tendances à court terme dans les données. De plus, les limites de contrôle dépendent de la variance à court terme des données. C'est pour ces deux raisons que le modèle EWMA est capable de s'adapter très bien à différents types de flux, sans générer une quantité excessive d'anomalies lorsque la variance du flux est élevée.

Modèle AR stationnaire. Au cours de l'étape suivante, au chapitre 5, nous avons utilisé les modèles autorégressifs stationnaires accompagnés de transformées pour supprimer les composantes tendancielles et périodiques. D'une part, les transformées permettent de supprimer les rythmes hebdomadaires et quotidiens des flux mais, d'autre part, ils rendent le signal analysé plus difficile à interpréter et créent des artéfacts dans les flux. Le modèle AR stationnaire a fonctionné avec le même type de flux que la modélisation EWMA et surtout nettement mieux que le modèle EWMA. Le modèle présente deux défauts majeurs, le premier étant la génération d'artéfacts et le deuxième la nécessité de séparer les données d'entraînement pour l'estimation des paramètres du modèle.

Modèles AR non stationnaire. Enfin, au chapitre 6, nous avons utilisé les modèles AR non stationnaires dont les paramètres ont été estimés avec un algorithme adaptatif, appelé lisseur à décalage fixe de Kalman. Cette approche a apporté deux avantages significatifs par rapport au modèle AR stationnaire. Premièrement, nous n'avons pas besoin d'utiliser les transformées de signaux et, deuxièmement, la phase d'entraînement est pratiquement inexistante.

Un troisième avantage est l'augmentation significative de la précision du modèle. Etant donné la précision du modèle et l'estimation dynamique des paramètres, le problème d'intégration des anomalies dans le modèle est devenu plus pertinent. Toutefois, le modèle AR est par nature mal adapté à la saisie des changements rapides et des phénomènes intermittents, ce qui aide à éviter le problème. De plus, les résultats ont montré que des anomalies ont été détectées malgré l'estimation dynamique.

Nous avons utilisé un intervalle d'échantillonnage d'une minute pour la première fois dans ce chapitre et proposé d'utiliser en plus des intervalles d'échantillonnage de 20 et 60 minutes pour saisir les anomalies présentes à échelle de temps plus large.

7.2 Comparaison des trois approches

Nous allons ensuite comparer les trois approches selon les quatre axes suivants :
- Généralité des paramètres de la méthode
- Précision du modèle
- Possibilité d'interprétation et cohérence des résultats
- Manière dont les flux peuvent être attribués automatiquement en cours de surveillance

La généralité des paramètres de la méthode est un aspect important du point de vue du déploiement. Les paramètres généraux permettent d'installer plus facilement la composante surveillance sur un nouveau système. Dans le cas idéal, le système pourrait être installé sans intervention manuelle d'un expert qui comprend les rouages internes et le comportement des modèles et des algorithmes.

La précision des modèles nous donne une idée de l'adéquation du modèle pour la surveillance du flux d'alertes en général. La composante détection est la même dans les trois approches et se base sur la carte de contrôle EWMA. Avec la modélisation des

tendances, la détection s'effectue directement à partir des observations. Avec les modèles AR stationnaires et non stationnaires, la détection s'effectue à partir des séries résiduelles. Malgré cette différence, le principe est le même pour les trois méthodes. Par conséquent, les capacités de détection des approches dépendent fortement de l'adaptabilité et de la précision du modèle. C'est pour cette raison que nous examinons les méthodes de ce point de vue.

La possibilité d'interprétation et la cohérence ont une signification similaire aux deux points précédents, mais du point de vue de l'utilisateur. Une grande possibilité d'interprétation et une grande cohérence pourraient signifier que les anomalies émises par le système peuvent être confirmées et analysées par un utilisateur qui ne sait pas comment les algorithmes sous-jacents fonctionnent et se comportent.

L'attribution automatique des flux pour la surveillance et le filtrage est de toute évidence un problème lié à la facilité de déploiement. Nos méthodes de modélisation et de filtrage sont clairement inadaptées à certains flux d'alertes. A moins que les flux surveillés ne soient choisis automatiquement, la sélection doit s'effectuer manuellement dans la phase de déploiement. De plus, le comportement du système surveillé peut évoluer au fil du temps. Dans ce cas, la sélection doit être vérifiée et éventuellement mise à jour. Si les mises à jour de la sélection peuvent également être automatisées, le travail manuel requis pour la maintenance du système est réduit.

7.2.1 Généralité des paramètres de la méthode

Si nous pouvons utiliser le même ensemble de paramètres partout, le déploiement du système est plus facile. Dans [MT00], Maxion et Tan ont découvert que la différence dans la régularité des données d'un utilisateur à l'autre rend plus difficile l'utilisation du même détecteur pour détecter le comportement anormal sur tout l'ensemble des utilisateurs. Cette question est tout aussi pertinente dans notre contexte et sous deux angles. Etant donné une méthode de traitement, peut-on utiliser le même ensemble de paramètres

1. pour tous les flux, et
2. pour différents types d'anomalies dans un flux ?

Dans cette section, nous discuterons ces points. Le premier est davantage lié aux différents comportements normaux, tandis que le deuxième dépend plus des types d'anomalies que nous avons rencontré.

Paramètres identiques pour tous les flux

Même si nous pouvons identifier des caractéristiques communes et certains types de comportement, la gamme des comportements normaux observés est très vaste. Nous avons rencontré des flux constants, différentes régularités, allant du sinus lisse à des fonctions graduelles abruptes.

Les modèles EWMA et AR non stationnaires sont mieux adaptés du fait qu'ils utilisent le même ensemble de paramètres pour la plupart des flux. Les deux modèles sont adaptatifs et le comportement normal et les anomalies sont suffisamment similaires à travers les flux. Les modèles AR stationnaires doivent au moins être estimés séparément pour chaque flux. Même si nous avons utilisé deux degrés de modèle différents, AR (4) et AR (26) au chapitre 5, nous aurions pu utiliser AR (26) pour tous les flux. Dans ce sens également, l'approche stationnaire pourrait être utilisée pour une vaste gamme de flux avec un seul

ensemble de paramètres. Toutefois, la situation dépend toujours de l'environnement d'exploitation.

Le choix d'utiliser le même ensemble de paramètres pour tous les flux implique également certaines contraintes. Certains flux pour lesquels les approches de surveillance ne sont pas applicables pourraient tirer profit d'un ajustement manuel, comme nous l'avons par exemple vu au chapitre 6. En général, avec une certaine compréhension de base des méthodes, l'ajustement par flux donne de meilleurs résultats au prix d'un plus grand travail manuel.

Dans l'ensemble, on peut dire que si la régularité dans le flux d'alertes prend principalement la forme d'impulsions, ni nos modèles ni les détecteurs ne sont adaptés à leur traitement. Le flux `LOCAL-POLICY external connexion from http server` de l'`ensemble-1` en est un exemple. Avec l'approche AR stationnaire et en particulier avec l'utilisation d'un opérateur de différenciation hebdomadaire, nous avons pu ignorer certains des pics réguliers. Il s'agissait toutefois plus d'une exception que de la norme et nous n'avons toujours pas pu saisir suffisamment bien le comportement normal. Nous n'avons même pas cherché à modéliser ce comportement similaire à une impulsion et une approche différente de la modélisation et de la détection devrait être développée pour ce type de flux.

Paramètres identiques pour différents types d'anomalies

Les anomalies intéressantes sont souvent soit des pics positifs ou négatifs dans l'intensité du flux, soit des changements de niveaux abrupts. En règle générale, avec les paramètres utilisés, les trois approches ont pu détecter ces types d'anomalies. A en juger par les résultats obtenus avec l'`ensemble-1`, les différences de détection des anomalies connues sont plutôt petites.

A ce point, nous devrions également songer aux anomalies de flux éventuellement non identifiées. Il existe deux raisons principales au fait de ne pas avoir identifié une anomalie à l'aide de l'inspection manuelle des flux. Premièrement, des anomalies d'échelle de temps courte pourraient avoir été manquées en raison du choix de l'intervalle d'échantillonnage. De manière similaire, des anomalies d'échelle de temps longue pourraient nécessiter des intervalles d'échantillonnage plus importants pour ressortir du flux. Deuxièmement, certaines anomalies peuvent ne pas être visibles en observant les comptages d'alertes. Bien entendu, nous ne pouvons pas être certains que ces anomalies existent ou non. Même si l'analyse du trafic du réseau est un domaine différent, il existe des similitudes entre les flux de réseau et les flux d'alertes, similitudes que nous analysons. Des études plus approfondies ont été menées concernant les caractéristiques et anomalies des flux de réseau et nous pouvons les utiliser comme point de départ dans cette recherche.

Barford et Plonka ont caractérisé les anomalies des flux du trafic du réseau à l'Université de Wisconsin - Madison dans [BP01]. Les gestionnaires de réseau ont analysé le trafic recueilli au cours de deux années. Ils ont utilisé différentes méthodes ad hoc pour détecter les anomalies. Toutes les anomalies ont été confirmées, diagnostiquées et enregistrées en détail. Barford et Plonka ont classé les anomalies en trois grandes catégories, anomalies de fonctionnement du réseau, anomalies « flash crowd » et anomalies d'utilisation abusive du réseau. Les anomalies des deux premières catégories ressemblent aux anomalies que nous avons rencontrées dans les flux d'alertes : changements de niveaux rapides, quasiment instantanés dans les comptages d'octets et de paquets. Les auteurs rapportent que les anomalies d'utilisation abusive du réseau, comme les attaques DoS, peuvent être difficiles à détecter à partir des comptages d'octets et de paquets, mais que les mesures de

comptages de flux sont une meilleure source de données. Borgnat et al. confirment la difficulté de la détection des attaques DoS à partir des comptages d'octets dans [BLAO05], en particulier à des fréquences d'échantillonnage élevées. Ils signalent que la moyenne et la variance sont des statistiques trop simples pour distinguer le trafic des attaques du trafic normal dans leurs données. Même si l'architecture de surveillance de l'Université de Wisconsin - Madison permet un échantillonnage par seconde, dans les exemples fournis, les intervalles d'échantillonnage varient de cinq minutes à une heure.

Les observations faites par Barford et Plonka étayent le raisonnement selon lequel nous pouvons couvrir une large gamme d'anomalies intéressantes en utilisant des intervalles d'échantillonnage d'une minute à une heure. De plus, leur catégorisation indiquerait que notre définition d'anomalies intéressantes pourrait en fait couvrir une grande partie des anomalies. Nos méthodes utilisent des observations de l'intensité du flux et de la variance pour la modélisation et la détection. De manière analogue aux difficultés de la détection des attaques DoS à partir des statistiques de premier ordre de Borgnat et al., certaines anomalies du flux d'alertes peuvent même ne pas être détectables par ces méthodes. Nous devons reconnaître les limitations introduites par l'intervalle d'échantillonnage et les métriques utilisées pour la modélisation et la détection.

Pour détecter les anomalies d'une échelle de temps plus large, nous avons suggéré trois intervalles d'échantillonnage différents, 1, 20 et 60 minutes pour couvrir une échelle plus large d'anomalies. Cette suggestion se base sur l'analyse du flux d'alertes au chapitre 3 et les résultats au chapitre 6. Nous avons utilisé différents intervalles d'échantillonnage uniquement avec la méthode non stationnaire, de sorte que nous ne savons pas comment les modèles EWMA et AR stationnaires se comportent avec des intervalles d'échantillonnage inférieurs à une heure. Nous avons pu utiliser les mêmes paramètres à différents intervalles d'échantillonnage. En d'autres termes, nous pouvons adapter simplement l'intervalle d'échantillonnage pour détecter des anomalies d'échelles de temps différentes. Il a également été mentionné au chapitre 3 que la forme des anomalies restait la même à différents intervalles d'échantillonnage. La forme similaire des anomalies explique en partie pourquoi nous pouvons conserver les mêmes paramètres à différents intervalles d'échantillonnage. De nouveau, l'ajustement des paramètres pour chaque intervalle d'échantillonnage est susceptible de donner de meilleurs résultats au prix d'une augmentation du travail manuel. Le point clé est que la méthode non stationnaire fonctionne suffisamment bien avec le même ensemble de paramètres.

Il semble qu'il n'y ait pas de différences significatives entre les trois modèles en termes de détection des différentes anomalies avec un seul ensemble de paramètres. Les trois approches peuvent détecter les types d'anomalies que nous considérons intéressantes avec le même ensemble de paramètres. Pour les méthodes EWMA et AR stationnaire, c'est ce que nous avons vu en termes de différentes anomalies à un seul intervalle d'échantillonnage. Pour la méthode non stationnaire, le même ensemble de paramètres a fonctionné également à différents intervalles d'échantillonnage.

Il existe toutefois des différences significatives dans les capacités de modélisation effectives et la manière dont les modèles relèvent ces anomalies intéressantes. C'est ce dont nous allons discuter ensuite.

7.2.2 Modèles et précision

Les nombres d'anomalies détectées connues dans l'`ensemble-1` ne nous permettent pas de différencier clairement les trois approches. Toutefois, en examinant le comporte-

ment modélisé du flux, c'est-à-dire la sortie du modèle et les valeurs résiduelles au lieu des anomalies détectées connues, la situation change. La comparaison de la précision des différentes approches est acceptable, étant donné que les différences sont plutôt grandes. Parallèlement, une comparaison détaillée, par exemple des différentes méthodes d'estimation pour l'un des modèles paramétriques n'est pas judicieuse en raison du manque de données d'entraînement propres.

Le modèle EWMA est réellement une projection brute de la réalité et, de ce fait, les erreurs de modèle sont grandes. Nous pourrions dire que l'approche est du type « cela fonctionne tout simplement » pour certains flux, mais la modélisation réelle du comportement normal est plutôt modeste.

L'approche AR stationnaire constitue une étape de plus vers un modèle réel de comportement normal. Le travail présenté au chapitre 5 nous a permis de mieux comprendre la nature des flux d'alertes, et la suppression des périodicités avec un opérateur de différenciation décalage-semaine nous a permis de traiter les problèmes générés par les rythmes hebdomadaires et quotidiens, jusqu'à un certain point. En examinant les valeurs résiduelles de l'`ensemble-1` du flux `ICMP PING speedera`, nous pouvons voir qu'il n'y a pas de pic d'erreurs significatif les lundis, ce qui indique que le modèle est capable de prédire les changements hebdomadaires. Toutefois, les erreurs générales sont toujours importantes par rapport aux valeurs d'intensité du flux. Pour `ICMP PING speedera` (figure 5.5(b)), les erreurs se situent pour la plupart dans la plage $[-30, 30]$, tandis que l'intensité du flux se situe dans la plage $[0, 100]$. Pour `ICMP PING WhatsupGold Windows` (figure 5.4(b)), les erreurs se situent principalement dans la plage $[-50, 50]$ et les valeurs d'intensité du flux se situent pour la plupart dans la plage $[0, 200]$. En même temps, on peut voir que la série résiduelle présente un certain type de rythmes hebdomadaires et quotidiens, indiquant des insuffisances dans le modèle.

Les modèles AR non stationnaires améliorent encore la précision. Lorsqu'elles sont estimées avec le lisseur à décalage fixe de Kalman avec un décalage de cinq observations, les erreurs pour `ICMP PING speedera` (figure 6.6(b)) se situent pour la plupart dans la plage $[-10, 15]$, et pour `ICMP PING Whatsupgold Windows` (figure 6.3(b)), dans la plage $[-10, 15]$. Comme on peut le voir dans les figures du chapitre 6, la sortie du modèle suit de près les observations réelles. Les rythmes hebdomadaires et quotidiens sont également moins prononcés dans les valeurs résiduelles du modèle AR non stationnaire que dans le modèle AR stationnaire, indiquant que le modèle AR non stationnaire s'adapte mieux aux rythmes des flux. En même temps, les changements abrupts dans les flux ressortent dans les valeurs résiduelles, ce qui signifie que malgré l'estimation adaptative, le comportement anormal n'est pas totalement inclus dans le modèle.

Une discussion plus détaillée sur l'adéquation des modèles et la qualité des algorithmes d'estimation serait quelque peu vaine, étant donné le manque de données de grande qualité. Grande qualité dans le sens que nous aurions des données normales connues qui, en même temps, seraient suffisamment riches pour contenir les bizarreries d'un réseau réel. Dans le cas idéal, le modèle du comportement normal serait interprété à partir de données propres. Cela permettrait à la fois l'utilisation de métrique classique d'adéquation de modèle pour une comparaison plus détaillée et l'établissement de taux de faux positifs. Toutefois, dans la pratique, c'est très difficile, voire impossible. Les environnements simulés ne donnent pas de résultats satisfaisants, voir par exemple [McH00], et les données réelles contiennent toujours une vaste gamme de bizarreries et d'anomalies. Nous avons choisi d'utiliser des données réelles pour voir comment les méthodes fonctionnent dans la pratique. Par conséquent, nos données ne sont pas propres et nous devons nous abste-

nir d'utiliser les métriques de performances classiques utilisées dans l'analyse des séries temporelles.

7.2.3 Possibilité d'interprétation et cohérence

La facilité d'utilisation de l'opérateur est affectée à la fois par la possibilité d'interprétation des anomalies et la cohérence dans la manière dont la méthode balise les anomalies.

Par possibilité d'interprétation, nous entendons la manière dont la méthode soutient le processus de confirmation et l'analyse d'une anomalie. Pour une meilleure possibilité d'interprétation, la sortie du modèle, les valeurs résiduelles et les anomalies doivent être visualisées avec l'intensité du flux. Ce qui fournit le contexte, c'est-à-dire le comportement passé du flux pour étayer l'analyse. Cela est nécessaire pour l'approche AR stationnaire, étant donné que les transformées du signal peuvent créer des anomalies artificielles. Pour diagnostiquer correctement ces artéfacts, le comportement passé du flux doit être visualisé. Le modèle EWMA est le plus facile à interpréter, car la tendance à court terme peut être visualisée à l'oeil nu d'après les flux. La sortie et les anomalies du modèle AR non stationnaire sont un peu plus difficiles à interpréter en raison de la plus grande complexité du modèle et des mises à jour dynamiques des coefficients du modèle. Toutefois, sa bonne précision et son adaptabilité facilitent l'analyse des anomalies, étant donné que le modèle suit le comportement observé suffisamment de près. En termes de possibilité d'interprétation, l'utilisation des modèles EWMA ou non stationnaires est clairement plus avantageuse que le modèle stationnaire.

Par cohérence, nous entendons la manière dont la méthode indique des types similaires d'anomalies dans différents contextes. Les trois modèles sont affectés par le comportement du flux précédant l'anomalie. Une anomalie antérieure ou une variabilité importante dans l'intensité du flux peut empêcher la détection de l'anomalie actuelle. L'adaptabilité et la précision du modèle sont étroitement liées à la cohérence du comportement. Par exemple, la méthode non stationnaire s'adapte rapidement, même à des changements majeurs dans le comportement du flux, et les anomalies antérieures sont moins susceptibles de masquer l'anomalie actuelle. De nouveau, la méthode AR stationnaire est visiblement la pire sous cet aspect, principalement en raison des transformées du signal. La suppression des périodicités en particulier peut introduire un comportement incohérent provoqué par le comportement passé du flux. Le modèle EWMA se comporte plus comme le modèle AR stationnaire, mais il est moins adaptatif et précis. En termes de cohérence, le modèle AR non stationnaire est clairement le meilleur des trois. Les résultats avec `ICMP Destination Unreachable Communication Administratively Prohibited` de l'`ensemble-1` en sont un bon exemple. Le modèle AR non stationnaire indique des pics d'alertes plus cohérents que les deux autres modèles (voir section 6.3.1).

7.2.4 Attribution des flux à la surveillance

La modélisation EWMA peut être utilisée sans données d'entraînement et les paramètres, une fois ajustés pour un système, semblent convenir pour une large gamme de flux d'alertes. Cela permet d'attribuer à la volée des flux pour la surveillance EWMA. Par exemple, un contrôle périodique pourrait analyser les données d'alertes pour les nouveaux flux potentiels en fonction de certains critères d'agrégation et certaines conditions de seuil. Si un flux dépasse les critères de seuil, les alertes appartenant à ce flux pourraient être

attribuées à la surveillance EWMA. Vice versa, si un flux existant tombe sous le seuil trop longtemps, la surveillance EWMA pour ce flux pourrait être exclue.

Les critères d'agrégation potentiels qui pourraient être examinés automatiquement sont les suivants :

- (capteur, signature)
- (capteur, signature, IP source)
- (capteur, signature, IP destination)
- (capteur, signature, IP source, IP destination)
- (capteur, IP source, IP destination)

Les seuils à utiliser pourraient être les suivants :

- nombre d'alertes dans le flux plus élevé que c
- au cours de n intervalles d'observations passées, le flux a des alertes sur des intervalles de plus de $0,45n$

La modélisation AR stationnaire nécessite une intervention manuelle et une telle attribution des flux surveillés n'est pas possible, tandis que la modélisation AR non stationnaire pourrait être utilisée de manière similaire.

7.3 Conclusion

Dans ce chapitre, nous avons dans un premier temps présenté un résumé des méthodes d'analyse et de traitement des alertes. Nous avons ensuite comparé les trois méthodes selon les quatre axes suivants : 1) généralité des paramètres de la méthode, 2) précision du modèle, 3) possibilité d'interprétation et cohérence des résultats, et 4) manière dont les flux peuvent être attribués à la surveillance.

Le premier, le troisième et le quatrième aspects sont liés à la facilité de déploiement et à la convivialité des méthodes. Le deuxième aspect décrit l'adaptabilité générale de la méthode pour la modélisation des flux d'alertes.

Dans l'ensemble, selon les axes liés à la facilité de déploiement et d'utilisation, les méthodes EWMA et AR non stationnaires sont très similaires. La méthode EWMA présente un léger avantage en termes de possibilité d'interprétation et la méthode AR non stationnaire un avantage certain en termes de cohérence. La méthode AR stationnaire nécessite d'avantage de travail manuel aussi bien dans la phase de déploiement que dans la phase de maintenance. De plus, les transformées du signal créent des problèmes au niveau de la possibilité d'interprétation et de la cohérence des résultats de la méthode AR stationnaire.

En termes de cohérence du modèle, la méthode AR non stationnaire présente un avantage évident par rapport aux deux autres, au prix d'algorithmes plus gourmands en ressources. Par rapport au modèle AR stationnaire, le modèle AR non stationnaire prend l'avantage avec l'estimation dynamique des paramètres. En comparaison avec le modèle EWMA, le modèle AR non stationnaire est capable de saisir un comportement plus complexe. De plus, l'algorithme du lisseur de Kalman permet au modèle de s'adapter plus rapidement aux changements du flux que le modèle EWMA. En résumé, la méthode AR non stationnaire est la plus adaptée des trois, aidant l'opérateur à faire face aux grands nombres d'alertes informatives.

Chapitre 8

Conclusion

Ce dernier chapitre conclura cette thèse[1] et nous discuterons brièvement quelques perspectives pour continuer ces travaux.

8.1 Conclusion

Dans cette thèse, notre but était de proposer des méthodes de traitement pour des alertes très volumineuses et non-pertinentes. La première contribution de la thèse est l'analyse portant sur les alertes, sur les flux d'alertes et leurs caractéristiques. Par l'analyse, nous avons identifié des régularités importantes dans les flux d'alertes générés par les signatures prolifiques. De plus, elle nous a permis d'expliquer et de justifier le besoin du type de méthodes de traitement proposées dans cette thèse. Par exemple, un des rapporteurs de l'article [VD04] a mentionné l'importance de ce problème pratique. Selon lui, ce problème est souvent négligé et en attente d'être résolu par quelqu'un autre. L'analyse a validé les trois premières parts de l'argument de cette thèse (Sect. 1.3).

Les méthodes elles mêmes constituent la deuxième contribution de cette thèse. Nous avons proposé de modéliser le comportement normal des flux d'alertes, et d'utiliser ces modèles pour réaliser un filtrage plus approprié que celui effectué par les outils de corrélation actuels.

Nous avons utilisé trois approches différentes pour cette tâche. En général, on peut dire que les complexités cognitive et de calcul de ces méthodes augmentent avec leur capacité de modélisation. Le plus simple, le modèle EWMA, est relativement efficace pour réduire la volumétrie des alertes. En même temps, sa capacité de modélisation est modeste.

Le modèle AR stationnaire est le premier pas vers les modèles de séries temporelles plus sophistiqués. Ce modèle fonctionne assez bien avec les mêmes types de flux que le modèle EWMA, et produit des résultats légèrement meilleurs que ce dernier. Par contre, il apporte aussi quelques désagréments au niveau de l'homogénéité des résultats et de leur interprétation. Cette approche est clairement la moins testée des trois.

Le modèle AR non-stationnaire n'apporte pas ces désagréments, et en plus, sa précision est significativement meilleure que celle de deux autres. Néanmoins, l'estimation dynamique utilisée avec ce modèle demande plus de ressources de calcul. Même si la consommation de ressources est plus élevée, nous voudrions rappeler que les algorithmes *Kalman*

[1] Nous souhaitons rappeler le lecteur que la version originale de cette thèse est en anglais, et que le lecteur est invité à se reporter à la version anglaise.

filter et *Kalman fixed-lag smoother* ont été conçus pour l'utilisation en ligne, en temps réel.

Ces méthodes, notamment l'approche non-stationnaire, sont capables de filtrer une quantité importante du bruit contenu dans le flux d'alertes. De plus, ils arrivent à souligner des phénomènes intéressants pour l'*investigation*. Les flux d'alertes reflètent la nature complexe du trafic réseau. Même si la projection faite par les sondes IDS diminue la complexité apparente, le comportement normal contient des phénomènes suffisamment significatifs pour qu'ils soient signalés par l'algorithme de détection. Pour cette raison, une investigation des anomalies est recommandée pour vérifier leur origine.

La réduction du bruit dans le flux d'alertes peut atteindre l'ordre 10^{-3} dans nos expérimentations, et peut être ajusté selon les besoins. Nous souhaitons rappeler, néanmoins, que la réduction est toujours un compromis entre le nombres d'anomalies non-pertinentes et le nombre de phénomènes intéressants manqués. Les résultats des expérimentations valident la partie quatre de notre argument de thèse.

8.2 Perspectives

Avec les méthodes proposés nous détectons des changements abrupts de l'intensité des flux, ceci même parmi un bruit important. Les changements progressifs et les anomalies de longue durée ne sont pas détectés. Par l'inspection visuelle des flux d'alertes, nous avons constaté que ces derniers sont rares. Actuellement, et avec les paramètres utilisés, le composant de détection ne souligne que les changements abrupts, et les anomalies précédentes peuvent masquer l'anomalie actuelle. Nous avons mentionné la possibilité d'utiliser de la détection spectrale dans le Chap. 6 et considérons que ceci est la première direction pour poursuivre ces travaux.

La seconde perspective est liée à l'analyse mono-échelle, discuté dans le Chap. 6. Nous pourrions dire que l'analyse actuelle est fait sur trois échelles de temps en utilisant les trois intervalles d'échantillonnage proposés. L'analyse multi-échelle, par exemple avec les ondelettes, pourrait améliorer cet aspect. Nous voudrions examiner la différence entre l'analyse multi-échelle et l'analyse sur trois échelles, en termes de résultats et ressources utilisées.

La troisième direction serait de chercher d'autres critères d'agrégation. Dans le Chap. 3 nous avons discuté la possibilité d'utiliser des méthodes comme celles de Julisch [Jul03b] pour utiliser une niveau d'agrégation plus détaillé. Ceci pourrait aider notamment l'interprétation des anomalies. En plus, il pourrait être utile d'examiner l'utilisation des autres mesures que l'intensité d'alertes. Ceci pourrait permettre de trouver des invariants, par exemple en étudiant les rapports des intensités de certain types d'alertes, et faciliter la détection. Nous avons vu aussi des corrélations entre des flux volumineux, ce qui pourrait être utilisé pour améliorer le filtrage, détection ou interprétation des anomalies.

Le quatrième possibilité est d'essayer de pousser ces méthodes de traitement vers les sondes. Si les alertes étaient traitées au niveau sonde, la volumétrie des alertes émises par la sonde pourrait être diminué. Cela permettrait de réduire la charge induite sur les composants de stockage et de corrélation, et sur les équipements de réseau.

Bibliographie

[ACF+00] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner. State of the Practice of Intrusion Detection Technologies. Technical Report CMU/SEI-99-TR-028, Carnegie Mellon Software Engineering Institute, Pittsburgh, PA, January 2000.

[ADD00] Magnus Almgren, Hervé Debar, and Marc Dacier. A Lightweight Tool for Detecting Web Server Attacks. In *Proceedings of the 2000 ISOC Symposium on Network and Distributed Systems Security*, pages 157–170, 2000.

[And80] James P. Anderson. Computer Security Threat Monitoring and Surveillance. Technical report, James P. Anderson Co., Fort Washington, Pa 19034, April 1980.

[Axe99] Stefan Axelsson. The Base-Rate Fallacy and Its Implications for the Difficulty of Intrusion Detection. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 1–7, Singapore, November 1999. Kent Ridge Digital Labs, ACM.

[Axe00a] Stefan Axelsson. Intrusion Detection Systems : A Survey and Taxonomy. Technical Report No 99-15, Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden, March 2000.

[Axe00b] Stefan Axelsson. A preliminary attempt to apply detection and estimation theory to intrusion detection. Technical Report 00-4, Department of Computer Engineering, Chalmers University of Technology, Gothenburg, Sweden, March 2000.

[Axe03] Stefan Axelsson. Visualization for intrusion detection : Hooking the worm. In *Proceedings of the 8th European Symposium on Research in Computer Security (ESORICS 2003)*, volume 2808 of *SER :LNCS*, Gjövik, Norway, October 2003. Springer–Verlag. URL : `http://www.cs.chalmers.se/~sax/pub/worm-web-vis.pdf`.

[Bac00] Rebecca Gurley Bace. *Intrusion Detection*. Macmillan Technical Publishing, 2000.

[BD91] Peter J. Brockwell and Richard A. Davis. *Time series : theory and methods*. Springer Texts in Statistics. Springer–Verlag, Heidelberg, Germany, 2nd edition, 1991.

[BD02] Peter J. Brockwell and Richard A. Davis. *Introduction to time series and forecasting*. Springer Texts in Statistics. Springer–Verlag, Heidelberg, Germany, 2nd edition, 2002.

[BJR94] George E. P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis : Forecasting and Control*. Prentice-Hall International, Inc., Upper Saddle River, New Jersey, 3rd edition, 1994.

[BKPR02] Paul Barford, Jeffery Kline, David Plonka, and Amos Ron. A Signal Analysis of Network Traffic Anomalies. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, November 2002. Available online : `http://www.icir.org/vern/imw-2002/imw2002-papers/173.pdf`.

[BLAO05] Pierre Borgnat, Nicolas Larrieu, Patrice Abry, and Philippe Owezarski. Détections d'attaques de déni de services : ruptures dans les statistiques du trafic. In *GRETSI'05*, Louvain-la-Neuve, Belgique, September 2005. URL : `http://www.laas.fr/METROSEC/gretsi_metrosec.pdf`.

[BP01] Paul Barford and David Plonka. Characteristics of network traffic flow anomalies. In *Proceedings of ACM SIGGCOMM Internet Measurement Workshop*, San Francisco, CA, USA, November 2001. URL : `http://www.aciri.org/vern/imw-2001/imw2001-papers/47.pdf`.

[Bru00] Jake D. Brutlag. Aberrant behavior detection in time series for network monitoring. In *Proceedings of the 14th Systems Administration Conference (LISA 2000)*, pages 139–146, New Orleans, Louisiana, USA, December 2000. USENIX, USENIX Association. URL : `http://www.usenix.org/events/lisa2000/full_papers/brutlag/`.

[CKT02] Chen-Mou Cheng, H. T. Kung, and Koan-Sin Tan. Use of Spectral Analysis in Defense Against DoS Attacks. In *Proceedings of IEEE Globecom 2002*, October 2002. Available online : `http://www.eecs.harvard.edu/~htk/publication/2002-globecom-cheng-kung-tan.pdf`.

[CLF03] S. Cheung, U. Lindqvist, and M. W. Fong. Modeling Multistep Cyber Attacks for Scenario Recognition. In *Proceedings of the third DARPA Information Survivability Conference and Exposition(DISCEX III)*, Washington D.C., USA, April 2003.

[CM02] Frédéric Cuppens and Alexandre Miège. Alert Correlation in a Cooperative Intrusion Detection Framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, 2002.

[CO00] Frédéric Cuppens and Rodolphe Ortalo. LAMBDA : A Language to Model a Database for Detection of Attacks. In Debar et al. [DMW00], pages 197–216.

[CST94] Stephen E. Cohn, N.S. Sivakumaran, and Ricardo Todling. A fixed-lag kalman smoother for retrospective data assimilation. *Monthly Weather Review*, 122(12) :2838–2867, 1994. URL : `http://citeseer.ist.psu.edu/cohn94fixedlag.html`.

[DBS92] Hervé Debar, Monique Becker, and Didier Siboni. A neural network component for an intrusion detection system. In *Proceedings of the IEEE Symposium on Research in Computer Security and Privacy*, pages 240–250, Oakland, CA, USA, May 1992.

[DC01] Oliver Dain and Robert K. Cunningham. Fusing a heterogeneous alert stream into scenarios. In *Proceedings of the Eighth ACM Conference on Computer and Communications Security*, 2001. URL : `http://www.ll.mit.edu/IST/pubs/acm_02_omd_rkc.pdf`.

[DC02] Oliver Dain and Robert K. Cunningham. Building scenarios from a heterogeneous alert stream. *IEEE Transactions on Systems, Man and Cybernetics*, 2002. URL : `http://www.ll.mit.edu/IST/pubs/ieee_02_omd_rkc.pdf`.

[DCF06] H. Debar, D. Curry, and B. Feinstein. The intrusion detection message exchange format. IETF Draft, March 2006. URL : `http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-16.txt`.

[DDW99] Hervé Debar, Marc Dacier, and Andreas Wespi. A Revised Taxonomy for Intrusion-Detection Systems. Technical Report RZ 3176 (#93222), IBM Research, Zurich, October 1999.

[Der03] Luca Deri. Improving Passive Packet Capture :Beyond Device Polling. URL : `http://luca.ntop.org/Ring.pdf`, November 2003.

[DK01] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, Ofxord, Great Britain, 2001.

[DKPS05] Holger Dreger, Christian Kreibich, Vern Paxson, and Robin Sommer. Enchancing the accuracy of network-based intrusion detection with host-based context. In *Detection of Intrusions and Malware, and Vulnerability Assessment, Second International Conference, (DIMVA 2005)*, Lecture Notes in Computer Science, pages 206–221, Heidelberg, Germany, July 2005. Springer–Verlag. URL : `http://www.cl.cam.ac.uk/~cpk25/publications/dimva05.pdf`.

[DM02] Hervé Debar and Benjamin Morin. Evaluation of the Diagnostic Capabilities of Commercial Intrusion Detection Systems. In Wespi et al. [WVD02].

[DMW00] Hervé Debar, Ludovic Mé, and Shyhtsun Felix Wu, editors. *Proceedings of the 3rd International Symposium on Recent Advances in Intrusion Detection (RAID 2000)*, volume 1907 of *Lecture Notes in Computer Science*, Heidelberg, Germany, 2000. Springer–Verlag.

[DoD85] Department of Defense Trusted Computer System Evaluation Criteria. Technical Report 5200.28-STD Orange Book, Department of Defense, December 1985.

[DV06] Hervé Debar and Jouni Viinikka. Security information management as an outsourced service. *Information Management & Computer Security*, 13(2) :416–434, 2006. URL : `http://www.emeraldinsight.com/Insight/viewContentItem.do?contentType=Article&contentId=1575972`.

[DW01] Hervé Debar and Andreas Wespi. Aggregation and Correlation of Intrusion-Detection Alerts. In Lee et al. [LMW01], pages 85–103.

[ET03] Robert F. Erbacher and Zhouxuan Teng. Analysis and Application of Node Layout Algorithms for Intrusion Detection. In *Proceedings of the SPIE'2003 Conference on Visualization and Data Analysis*, Santa Clara, CA, USA, January 2003.

[EWF02] Robert F. Erbacher, Kenneth L. Walker, and Deborah A. Frincke. Intrusion and Misuse Detection in Large-Scale Systems. *Computer Graphics and Applications*, 22(1) :38–48, January 2002.

[EY01] Syed Masum Emran and Nong Ye. Robustness of canberra metric in computer intrusion detection. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, pages 80–84, United States Military Academy, West Point, NY, USA, June 2001. URL `http://www.itoc.usma.edu/Workshop/2001/Authors/Submitted_Abstracts/paperT2A1(02).pdf`.

[FHSL96] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A Sense of Self for Unix Processes. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 120–128, 1996.

[FTM+98] D. A. Frincke, D. Tobin, J. C. McConnell, J. Marconi, and D. Polla. A Framework for Cooperative Intrusion Detection. In *Proceedings of the 21st NIST-NCSC National Information Systems Security Conference*, pages 361–373, 1998. URL : `http://csrc.nist.gov/nissc/1998/proceedings/paperF6.pdf`.

[GFD+06] Guofei Gu, Prahlad Fogla, David Dagon, Wenke Lee, and Boris Škorić. Measuring intrusion detection capability : An information-theoretic approach. In Shiuhpyng Shieh and Sushil Jajodia, editors, *Proceedings of the ACM Symposium on Information, Computer and Communications Security (AsiaCCS'06)*, Taipei, Taiwan, March 2006. ACM.

[GHH+01] R. P. Goldman, W. Heimerdinger, S. A. Harp, C. W. Geib, V. Thomas, and R. L. Carter. Information modeling for intrusion report aggregation. In *Proceedings of the DARPA Information Survivability Conference and Exposition*, pages 329–342, June 2001.

[GN00] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *Software – Practice and Experience*, 30(11) :1203–1233, 2000. URL : `http://www.graphviz.org/Documentation/GN99.pdf`.

[Gra69] C. W. J. Granger. Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3) :424–438, August 1969.

[GYB06] Vikas Garg, Vinod Yegneswaran, and Paul Barford. Improving nids performance through hardware-based connection filtering. In *Proceedings of IEEE International Conference on Communications*, June 2006. To appear, URL : `http://www.cs.wisc.edu/~pb/filters_final.pdf`.

[Hal03] John Hally. Back-door'ed by the slammer. SANS GIAC report, September 2003. URL : `http://www.giac.org/certified_professionals/practicals/gcih/0477.php`.

[HBE04] J. Hall, M. Barbeau, and Kranakis E. Enhancing intrusion detection in wireless network using radio frequency fingerprinting. Extended Abstract in Communicatons, Internet and Information Technology, November 2004. URL : `http://www.scs.carleton.ca/~jhall2/Publications/CIIT04.pdf`.

[HHP03a] Alefiya Hussain, John Heidemann, and Christos Papadopoulos. A Framework for Classifying Denial of Service Attacks. Technical Report ISI-TR-2003-569b, USC/Information Sciences Institute, August 2003. Extended version of SIGCOMM 2003 paper. Available online : `http://www.isi.edu/~hussain/pubs/Hussain03a.html`.

[HHP03b] Alefiya Hussain, John Heidemann, and Christos Papadopoulos. Identification of Repeated DoS Attacks using Network Traffic Forensics. Technical Report ISI-TR-2003-577, USC/Information Sciences Institute, August 2003. Available online : `http://www.isi.edu/~hussain/pubs/Hussain03c.html`.

[HWI05] Dave Hull and George F. Willard III. Next generation dhcp deployments. *Sys Admin*, 14(2), February 2005. URL : `http://www.insipid.com/NGDHCP.pdf`.

[JD02] Klaus Julisch and Marc Dacier. Mining Intrusion Detection Alarms for Actionable Knowledge. In *Proceedings of Knowledge Discovery in Data and Data Mining (SIGKDD)*, 2002.

[Jul01] Klaus Julisch. Mining Alarm Clusters to Improve Alarm Handling Efficiency. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC2001)*, December 2001.

[Jul03a] Klaus Julisch. Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security*, 6(4), nov 2003. URL : `http://www.zurich.ibm.com/~kju/tissec.pdf`.

[Jul03b] Klaus Julisch. *Using Root Cause Analysis to Handle Intrusion Detection Alarms*. Phd thesis, University of Dortmund, Germany, 2003. URL :`http://eldorado.uni-dortmund.de:8088/FB4/ls6/forschung/2003/Julisch`.

[JV91] Harold S. Javitz and Alfonso Valdes. The SRI IDES Statistical Anomaly Detector. In *IEEE Symposium on Research in Security and Privacy*, May 1991.

[JV93] Harold S. Javitz and Alfonso Valdes. The NIDES Statistical Component : Description and Justification. Technical report, SRI International, Menlo Park, California 94025, March 1993. Broken ps, reversed pdf.

[KR04] C. Kruegel and W. Robertson. Alert verification : Determining the success of intrusion attempts. In *Proceedings of the 1st Workshop on the Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, Dortmund, Germany, July 2004. URL : `http://www.cs.ucsb.edu/~wkr/publications/dimva04verification.pdf`.

[KS94] Sandeep Kumar and Eugene H. Spafford. A Pattern Matching Model for Misuse Intrusion Detection. In *Proceedings of the Seventeenth National Computer Security Conference*, pages 11–21, October 1994.

[KS05] Jari P. Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*, volume 160 of *Applied Mathematical Sciences*. Springer–Verlag, Heidelberg, Germany, 2005.

[LB78] Greta M. Ljung and G. E. P. Box. On a Measure of Lack of Fit in Time Series Models. *Biometrica*, 65(2) :297–303, August 1978.

[LBMC94] Carl E. Landwher, Alan R. Bull, John P. McDermott, and William S. Choi. A Taxonomy of Computer Program Security Flaws. *ACM Computing Surveys*, 26(3) :211–254, September 1994.

[LCD04] Anukool Lakhina, Mark Crovella, and Christophe Diot. Diagnosing network-wide traffic anomalies. In *Proceedings of the ACM SIGCOMM'04*, pages 219–230, Portland, Oregon, USA, August 2004. URL : `http://www.cs.bu.edu/faculty/crovella/paper-archive/sigc04-network-wide-anomalies.pdf`.

[LFG+00] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman. Evaluating Intrusion Detection Systems : the 1998 DARPA

Off-Line Intrusion Detection Evaluation. In *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, volume 2, 2000.

[LHF+00] Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, and Kumar Das. Analysis and results of the 1999 darpa off-line intrusion detection evaluation. In Debar et al. [DMW00], pages 162–182. URL : `http://www.scs.carleton.ca/~soma/id-2006w/readings/lippmann-raid00.pdf`.

[LLT02] Wai Lup Low, Joseph Lee, and Peter Teoh. Didafit : Detecting intrusions in databases through fingerprinting transactions. In *Proceedings of the 4th International Conference on Enterprise Information Systems*, pages 121–128, Ciudad Real, Spain, April 2002. URL : `http://security.dso.org.sg/files/p0/30/a1_1f_016_220_low.pdf`.

[LMW01] Wenke Lee, Ludovic Mé, and Andreas Wespi, editors. *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID 2001)*, volume 2212 of *Lecture Notes in Computer Science*, Heidelberg, Germany, 2001. Springer–Verlag.

[LP99] Ulf Lindqvist and Phillip A. Porras. Detecting Computer and Network Misuse Through the Production-Based Expert System Toolset (P-BEST). In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, 1999.

[LS98] Wenke Lee and Sal Stolfo. Data Mining Approaches for Intrusion Detection. In *Proceedings of the 7th USENIX Security Symposium*, January 1998. Available online : `http://www.cc.gatech.edu/~wenke/papers/usenix/usenix.html`.

[LSM98] Wenke Lee, Sal Stolfo, and Kui W. Mok. Mining Audit Data to Build Intrusion Detection Models. In Rakesh Agrawal and Paul Stolorz, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, 445 Burgess Drive, Menlo Park, CA 94025-3496, USA, 1998. AAAI Press. Available online : `http://www.cc.gatech.edu/~wenke/papers/kdd98.ps`.

[Man96] Hannu Mannila. Data mining : machine learning, statistics, and databases. In *Proceedings of the Eight International Conference on Scientific and Statistical Database Management*, pages 1–8, 1996. Available online : `http://www.cs.helsinki.fi/~mannila/postscripts/ssdbm.ps`.

[Man97] Hannu Mannila. Methods and Problems in Data Mining. In F. Afrati and P. Kolaitis, editors, *Proceedings of International Conference on Database Theory (ICDT'97)*, pages 41–55. pub :springer, 1997. Available online : `http://www.cs.helsinki.fi/~mannila/postscripts/icdt-tutorial.ps`.

[MBGL06] Alexander Moshchuk, Tanya Bragin, Steven D. Gribble, and Henry M. Levy. A crawler-based study of spyware on the web. In *Proceedings of the 13th Annual Networked and Distributed System Security Symposium(NDSS 2006)*, San Diego, CA, USA, February 2006. URL : `http://www.cs.washington.edu/homes/gribble/rw/papers/index_assets/spycrawler.pdf`.

[MC02] Matthew V. Mahoney and Philip K. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, pages

376–385, Edmonton, Alberta, Canada, July 2002. URL : `http://www.cs.fit.edu/~mmahoney/paper4.pdf`.

[MCB+06] David Moore, Shannon Colleen, Doug Brown, Geoffrey M. Voelker, and Stefan Savage. Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems*, 24(2) :115–139, May 2006. URL : `http://www.caida.org/publications/papers/2006/backscatter_dos/backscatter_dos.pdf`.

[McH00] John McHugh. The 1998 Lincoln Laboratory IDS Evaluation. In Debar et al. [DMW00], pages 145–161.

[McH01] John McHugh. Intrusion and intrusion detection. *International Journal of Information Security*, July 2001.

[MCZH99] Stefanos Manganaris, Marvin Christensen, Dan Zerkle, and Keith Hermiz. A Data Mining Analysis of RTID Alarms. 2nd International Symposium on Recent Advances in Intrusion Detection (RAID 1999), 1999. Available online : `http://www.raid-symposium.org/raid99/PAPERS/Manganaris.pdf`.

[MD03] Benjamin Morin and Hervé Debar. Correlation of Intrusion Symptoms : an Application of Chronicles. In Vigna et al. [VJK03], pages 94–112.

[MHL+03] Peter Mell, Vincent Hu, Richard Lippman, Joss Haines, and Marc Zissman. An Overview of Issues in Testing Intrusion Detection Systems. NIST IR 7007, NIST CSRC - National Institute of Standards and Technology, Computer Security Resource Center, June 2003.

[MMDD02] Benjamin Morin, Ludovic Mé, Hervé Debar, and Mireille Ducassé. M2D2 : A Formal Data Model for IDS Alert Correlation. In Wespi et al. [WVD02], pages 115–137.

[Mor03] Benjamin Morin. *Corrélation d'alertes issues d'outils de détection d'intrusions avec prise en compte d'informations sur le système surveillé*. PhD thesis, INSA, Rennes, France, February 2003.

[MT96] Heikki Mannila and Hannu Toivonen. Discovering generalized episodes using minimal occurrences. In Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, pages 146–151, 445 Burgess Drive, Menlo Park, CA 94025-3496, USA, 1996. AAAI Press. Available online : `http://www.cs.helsinki.fi/research/fdk/datamining/pubs/kdd96b.ps.gz`.

[MT00] Roy A. Maxion and Kymie M.C. Tan. Benchmarking anomaly-based detection systems. In *Proceedings of the International Conference on Dependable Systems and Networks*, pages 623–630, New York, New York, USA, June 2000. IEEE Computer Society Press. URL : `http://www.cs.cmu.edu/afs/cs.cmu.edu/user/maxion/www/pubs/maxiontan00.pdf`.

[MTV97] Heikki Mannila, Hannu Toivonen, and A. Inkeri Virkamo. Discovery of frequent episodes in event sequences. Technical Report C-1997-15, University of Helsinki, Department of Computer Science, Helsinki, Finland, February 1997. Available online : `http://www.cs.helsinki.fi/research/fdk/datamining/pubs/C-1997-15.ps.gz`.

[MVS01] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring internet denial-of-service activity. In *Proceedings of the 10th USENIX Security Symposium*, Washington, D.C., USA, August 2001. URL : `http://www.caida.org/publications/papers/2001/BackScatter/usenixsecurity01.pdf`.

[MWR02] Vinay A. Mahadik, Xiaoyong Wu, and Douglas S. Reeves. Detection of Denial of QoS Attacks Based on χ^2 Statistic and EWMA Control Chart. URL : `http://arqos.csc.ncsu.edu/papers.htm`, February 2002.

[NCR02a] Peng Ning, Yun Cui, and Douglas Reeves. Constructing attack scenarios through correlation of intrusion alerts. In *Proceedings of the 9th ACM Conference on Computer & Communications Security*, pages 245–254, Washington D.C., USA, November 2002. URL : `http://discovery.csc.ncsu.edu/~pning/pubs/ccs02.pdf`.

[NCR02b] Peng Ning, Yun Cui, and Douglas Reeves. Constructing attack scenarios through correlation of intrusion alerts. Full version of paper in CCS'03, 2002. URL : `http://discovery.csc.ncsu.edu/~pning/pubs/AttackScenarios.ps`.

[NCSLO02] Kofi Nyarko, Tanya Carpers, Craig Scott, and Kemi Ladeji-Osias. Network Intrusion Visualization with NIVA, an Intrusion Detection Visual Analyzer with Haptic Integration. In *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 277–284. IEEE, 2002.

[NP99] Peter G. Neumann and Phillip A. Porras. Experience with EMERALD to Date. In *Proceedings of 1st USENIX Workshop on Intrusion Detection and Network Monitoring*, Santa Clara, California, April 1999.

[OA04] America Online and National Cyber Security Alliance. Aol/ncsa safety study. URL : `http://www.staysafeonline.info/pdf/safety_study_v04.pdf`, October 2004.

[Pax99] Vern Paxson. Bro : A System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23–24) :2435–2463, December 1999.

[PDP05] Fabien Pouget, Marc Dacier, and Van Hau Pham. Leurre.com : on the advantages of deploying a large scale distributed honeypot platform. In *Proceedings of the E-Crime and Computer Conference 2005 (ECCE'05)*, Monaco, March 2005. URL : `http://www.honeynet.org/papers/individual/ECCE_pouget_dacier_pham.pdf`.

[PFV02] P. A. Porras, M. W. Fong, and A. Valdes. A Mission-Impact-Based Approach to INFOSEC Alarm Correlation. In Wespi et al. [WVD02], pages 95–114.

[PN98] Thomas H. Ptacek and Timothy N. Newsham. Insertion, Evasion, and Denial of Service : Eluding Network Intrusion Detection. Technical report, Secure Networks, Inc., January 1998.

[PTL04] Konstantina Papagiannaki, Nina Taft, and Anukool Lakhina. A distributed approach to measure ip traffic matrices. In *Proceedings of the ACM Internet Measurement Conference*, Taormina, Sicily, Italy, October 2004. URL : `http://berkeley.intel-research.net/nina/Publications/Taft-IMC04.pdf`.

[PYB+04] Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson. Characteristics of internet backroung radiation. In *Proceedings of the ACM Internet Measurement Conference*, October 2004. URL : `http://www.cs.wisc.edu/~pb/background_final.pdf`.

[QL03] Xinzhou Qin and Wenke Lee. Statistical Causality Analysis of INFOSEC Alert Data. In Vigna et al. [VJK03], pages 73–93.

[QL04] Xinzhou Qin and Wenke Lee. Discovering novel attack strategies from infosec alerts. In *Proceedings of The 9th European Symposium on Research in Computer Security (ESORICS 2004)*, Sophia–Antipolis, France, September 2004. URL : `http://www.cc.gatech.edu/~wenke/papers/esorics_paper_2004.pdf`.

[Rob59] S. W. Roberts. Control Chart Tests Based On Geometric Moving Averages. *Technometrics*, 1(3) :230–250, 1959.

[Roe99] Martin Roesch. Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of LISA'99*, Seattle, Washington, USA, November 1999.

[ROT03] Manikantan Ramadas, Shawn Ostermann, and Brett Tjaden. Detecting anomalous network traffic with self-organizing maps. In Vigna et al. [VJK03]. URL : `http://www.cs.fit.edu/~pkc/id/related/ramadas03raid.ps.gz`.

[SLB+06] A. Scherrer, N. Larrieu, P. Borgnat, P. Owezarski, and P. Abry. Non gaussian and long memory statistical modeling of internet traffic. In *Proceedings of the Internet Performance, Simulation, Monitoring and Measurements*, Salzburg, Austria, February 2006.

[SLO+06] A. Scherrer, N. Larrieu, P. Owezarski, P. Borgnat, and P. Abry. Une caractérisation non gaussienne et à longue mémoire du trafic internet et de ses anomalies. In *Proceedings of the 5th Conference on Security and Network Architectures (SAR 2006)*, pages 29–50, Seignosse, France, June 2006.

[SLT+05] Augustin Soule, Anukool Lakhina, Nina Taft, Konstantina Papagiannaki, Kavé Salamatian, Antonio Nucci, Mark Crovella, and Christophe Diot. Traffic matrices : Balancing measurements, inference and modeling. In *Proceedings of the ACM SIGMETRICS'05*, Banff, Alberta, Canada, June 2005. URL : `http://berkeley.intel-research.net/nina/Publications/TMComparison-Sigmetrics05.pdf`.

[SPW02] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to 0wn the Internet in Your Spare Time. In *Proceedings of USENIX Security Symposium*, 2002.

[SS05] Jerry Shenck and Dave Shackleford. Sourcefire real-time network awareness. SANS Analyst Program, 2005. URL : `http://www.sourcefire.com/products/wp_request.html`.

[SSNT05] Augustin Soule, Kavé Salamatian, Antonio Nucci, and Nina Taft. Traffic matrix tracking using kalman filters. *ACM Sigmetrics Performance Evaluation Review*, 33(3) :24–31, December 2005. URL : `http://berkeley.intel-research.net/nina/Publications/Taft-LSNI2005.pdf`.

[SWWJ06] Hemant Sengar, Duminda Wijesekera, Haining Wang, and Sushil Jajodia. Voip intrusion detection through interacting protocol state machines. In

Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN'06), 2006. URL :`http://www.cs.wm.edu/~hnw/paper/dsn06.pdf`.

[SYB06] Joel Sommers, Vinod Yegeneswaran, and Paul Barford. Recent advances in netowork intrusion detection system tuning. In *Proceedings of the 40th IEEE Conference on Information Sciences and Systesm*, March 2006. URL : `http://www.cs.wisc.edu/~pb/tuning_final.pdf`.

[Sym06] Symantec. Symantex internet security threat report. URL : `http://www.symantec.com/enterprise/threatreport/index.jsp`, March 2006. Volume IX.

[Tar04] Mika Tarvainen. *Estimation Methods for Nonstationary Biosignals*. PhD thesis, Department of Applied Physics, University of Kuopio, Kuopio, Finland, 2004. URL : `http://it.uku.fi/biosignal/pdf/MT_PhD_ThesisCOL.pdf`.

[TC97] K. M. C. Tan and B. S. Collie. Detection and classification of tcp/ip network services. In *Proceedings of the 13th Annual Computer Security Applications Conference (ACSAC 1997)*, pages 99–107, San Diego, CA, USA, December 1997.

[TDM06] Yohann Thomas, Hervé Debar, and Benjamin Morin. Improving security management through passive network observation. In *Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)*, pages 382–389, Vienna, Austria, April 2006.

[TDMD04] Elvis Tombini, Hervé Debar, Ludovic Mé, and Mireille Ducassé. A serial combination of anomaly and misuse idses applied to http traffic. In *20th Annual Computer Security Applications Conference (ACSAC 2004)*, pages 428–437, Tucson, AZ, USA, December 2004.

[TK00] Steven J. Templeton and Levitt Karl. A requires/provides model for computer attaacks. In *Proceedings of the ACM New Security Paradigms Workshop*, pages 31–38, Cork Ireland, September 2000. URL : `http://seclab.cs.ucdavis.edu/papers/NP2000-rev.pdf`.

[TM03] Soon Tee Teoh and Kwan-Liu Ma. StarClass : Interactive Visual Classification Using Star Coordinates. In *Proceedings of the SIAM International Conference on Data Mining*, 2003.

[TMM05] Eric Totel, Frédéric Majorczyk, and Ludovic Mé. Diversity based intrusion detection and application to web servers. In Alfonso Valdes and Diego Zamboni, editors, *Proceedings of the Eighth International Symposium on Recent Advances in Intrusion Detection (RAID2005)*, Lecture Notes in Computer Science, Heidelberg, Germany, September 2005. Springer–Verlag. URL : `http://www.rennes.supelec.fr/ren/rd/ssir/publis/tmm05.pdf`.

[TMW03] Soon Tee Teoh, Kwan-Liu Ma, and S. Felix Wu. A Visual Exploration Process for the Analysis of Internet Routing Data. In *Proceedings of IEEE Visualization*, 2003.

[TMWZ02a] Soon Tee Teoh, Kwan-Liu Ma, S. Felix Wu, and Xiaoliang Zhao. A Visual Technique for Internet Anomaly Detection. In *Proceedings of IASTED Computer Graphics and Imaging*. ACTA Press, 2002.

[TMWZ02b] Soon Tee Teoh, Kwan-Liu Ma, S. Felix Wu, and Xiaoliang Zhao. Case Study : Interactive Visualization for Internet Security. In *Proceedings of IEEE Visualization*, 2002.

[VD04] Jouni Viinikka and Hervé Debar. Monitoring IDS Background Noise Using EWMA Control Charts and Alert Information. In *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID)*, volume 3224 of *Lecture Notes in Computer Science*. Springer–Verlag, 2004.

[VDMS06] Jouni Viinikka, Hervé Debar, Ludovic Mé, and Renaud Séguier. Time series modeling for ids alert management. In *Proceedings of the ACM Symposium on InformAtion, Computer and Communications Security(AsiaCCS'06)*, pages 102–113, Taipei, Taiwan, March 2006.

[VEK00] Giovanni Vigna, Steven T. Eckmann, and Richard A. Kemmerer. The stat tool suite. In *Proceedings of DISCEX 2000*, Hilton Head, South Carolina, January 2000. IEEE Computer Society Press. URL : `http://www.cs.ucsb.edu/~rsg/pub/2000_vigna_eckmann_kemmerer_discex00.ps.gz`.

[Vii03] Jouni Viinikka. Statistical Analysis Techniques to Find Trends in Alert Information. Master's thesis, Helsinki University of Technology, Espoo, Finland, October 2003.

[VJK03] Giovanni Vigna, Erland Jonsson, and Christopher Kruegel, editors. *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID 2003)*, volume 2820 of *Lecture Notes in Computer Science*, Heidelberg, Germany, 2003. Springer–Verlag.

[VS00] Alfonso Valdes and Keith Skinner. Adaptive, Model-Based Monitoring for Cyber Attack Detection. In Debar et al. [DMW00], pages 80–93.

[VS01] Alfonso Valdes and Keith Skinner. Probabilistic Alert Correlation. In Lee et al. [LMW01].

[Wel67] Peter D. Welch. The use of fast fourier transform for the estimation of power spectra : A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2) :70–73, June 1967.

[WMR03] Xiaoyong Wu, Vinay A. Mahadik, and Douglas S. Reeves. A summary of detection of denial-of-qos attacks on diffserv networks. In *DARPA Information Survivability Conference and Exposition (DISCEX'03)*, pages 277–282, Washington, DC, USA, 2003. IEEE Computer Society.

[WS04] Ke Wang and Salvatore J. Stolfo. Anomalous payload-based network intrusion detection. In Erland Jonsson, Alfonso Valdes, and Magnus Almgren, editors, *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004)*, number 3224 in Lecture Notes in Computer Science, pages 203–222, Berlin, Heidelberg, Germany 2004. Springer–Verlag.

[WVD02] Andreas Wespi, Giovanni Vigna, and Luca Deri, editors. *Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002)*, volume 2516 of *Lecture Notes in Computer Science*, Heidelberg, Germany, 2002. Springer–Verlag.

[YBC02] Nong Ye, Connie Borror, and Yebin Chang. EWMA Techniques for Computer Intrusion Detection Through Anomalous Changes In Event Intensity. *Quality and Reliability Engineering International*, 18 :443–451, 2002.

[YECV02] Nong Ye, Syed Masum Emran, Qiang Chen, and Sean Vilbert. Multivariate Statistical Analysis of Audit Trails for Host-Based Intusion Detection. *IEEE Transactions on Computers*, 51(7) :810–820, July 2002.

[YVC03] Nong Ye, Sean Vilbert, and Qiang Chen. Computer Intrusion Detection Through EWMA for Autocorrelated and Uncorrelated Data. *IEEE Transactions on Reliability*, 52(1) :75–82, March 2003.

[ZGTG05] Cliff C. Zou, Weibo Gong, Don Towsley, and Lixin Gao. The monitoring and early detection of internet worms. *IEEE/ACM Transaction on Networking*, 13(5) :961–974, October 2005. URL : `http://www.cs.ucf.edu/~czou/research/earlyDetectionJournal.pdf`.