HAL
open science

Détection des intrusions et aide à la décision

David Pierrot, Nouria Harbi, Jérôme Darmont

▶ **To cite this version:**

David Pierrot, Nouria Harbi, Jérôme Darmont. Détection des intrusions et aide à la décision. 12e Conférence sur les Avancées des Systèmes Décisionnels (ASD 2018), May 2018, Marrakech, Maroc. hal-01761914

HAL Id: hal-01761914

https://hal.science/hal-01761914v1

Submitted on 20 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Détection des intrusions et aide à la décision

David PIERROT*, Nouria HARBI**, Jérôme DARMONT**

*Université de Lyon, ERIC EA 3083
5 avenue Pierre Mendès France F69676 Bron Cedex France
david.pierrot1@univ-lyon2.fr
**{nouria.harbi,jerome.darmont}@univ-lyon2.fr

Résumé. Les conséquences d'une intrusion dans un système d'information peuvent s'avéver problématiques pour l'existence d'une entreprise ou d'une organisation. Les impacts sont synonymes d'une perte financière, d'image de marque et de sérieux. La détection d'une intrusion n'est pas une finalité en soit, la réduction du delta détection-réaction est devenue prioritaire. Nous proposons une méthode prenant en compte les aspects techniques par l'utilisation d'une méthode hybride de Data mining mais aussi les aspects fonctionnels. L'addition de ces deux aspects permet d'obtenir une vision générale sur l'hygiène du système d'information mais aussi une orientation sur la surveillance et les corrections à apporter.

1 Introduction

La détection d'intrusions est devenue une priorité pour garantir le maintien opérationnel d'un système d'information. Au cours des dernières décennies, de nombreuses approches ont été créées pour détecter les intrusions. Il existe deux catégories principales de détection d'intrusions : les abus et les anomalies. La détection des abus est basée sur des modèles spécifiques identifiés par des signatures. Cette méthode souffre de signatures manquantes pour les attaques inconnues. L'inconvénient principal qui est déjà largement démontré repose sur les faux positifs. Cette problématique était déjà évoquée lors de la conférence Black Hat en 2006 (Zambon et Bolzoni, 2006). Il n'est donc pas étonnant de retrouver cette difficulté en 2018 comme le soulignent parfaitement H.Rais et T.Mehmood. (2018)

La seconde méthode de détection basée sur les anomalies est obtenue à partir d'un modèle de comportement normal : une dérivation significative générera une alerte.

De nombreuses approches ont été développées pour améliorer la détection des anomalies à l'aide de méthodes de statistiques ou d'exploration de données (data mining). Cependant, la plupart des approches nécessitent de capturer le trafic réseau avec des outils spécifiques. Ceci peut avoir un impact sur la performance globale du réseau. La détection d'une intrusion est certainement possible, mais sa prise en compte et son traitement restent nébuleux. Les aspects de sécurité fonctionnelle tels que la gestion des risques ne sont jamais exploités. Ce point peut avoir un impact sur une prise de décision rapide faute de connaissance de l'actif visé. Par conséquent, les IDS existantes et les travaux de recherche sont assez difficiles à mettre en

œuvre sans un effort significatif et soutenu. L'objectif de cet article est : 1) d'analyser et d'expliquer l'état actuel des pratiques de détection d'intrusions ; 2) de discuter du travail que nous avons effectué pour faciliter la visualisation du flux de données du système d'information et la détection d'intrusions / d'attaques. Dans la continuité du document de David Pierrot (2016), nous proposons d'ajouter la mise en œuvre de la gestion des risques et l'utilisation des résultats obtenus lors d'un audit de sécurité. Notre principale contribution est l'extraction et l'analyse de journaux de type Firewall en combinant des méthodes de data mining pour détecter automatiquement les dérivations de comportements et les abus. Nous intégrerons le résultat d'une analyse de risque afin de cibler les actifs les plus sensibles visés par un comportement anormal. Notre travail tente de prouver qu'il est possible sans sonde de détection ou base de signatures de détecter des intrusions et d'appliquer une méthode de prise en charge.

2 Étude de l'existant

Le Data mining offre diverses solutions avantageuses pour réaliser la détection et l'analyse des intrusions. Trois types d'approches d'exploration de données sont possibles comme le précisent Deepa et Kavitha (2012). L'apprentissage supervisé utilise un ensemble de données labellisées pour effectuer des prédictions (classification ou régression). La labellisation des données peut être assistée d'un expert. L'apprentissage non supervisé est appliqué sur des ensembles de données non étiquetés pour découvrir des similitudes. Enfin, l'apprentissage semi-supervisé est une combinaison de techniques d'apprentissage sur les données étiquetées et non étiquetées.

Il est également possible d'utiliser des méthodes hybrides combinant un apprentissage supervisé et non supervisé.

Pour tester les différentes méthodes d'apprentissage, des "benchmarks" (jeux de données réels ou synthétiques) ont été proposés.

KDD99 est sans doute le plus cité et utilisé (University of California, Irvine, 1999) et ceci avec une existence de 20 années. Il est construit à partir de sept semaines de trafic réseau capturé avec TCPdump (Garcia, 2017). Il fournit des données étiquetées pour quatre types d'attaques :

— DOS : déni de service ;
— R2L : accès non autorisé depuis une machine distante ;
— U2R : accès non autorisé aux privilèges du super-utilisateur local (root) ;
— Probe (sonde) : surveillance.

NSL-KDD (`http://www.unb.ca/cic/research/datasets/nsl.html`) est similaire à KDD99. Il est dénué de doublons et avec un nombre d'enregistrements limité. Ainsi, les méthodes d'apprentissage peuvent être entièrement utilisées dans des délais raisonnables (Elkhadir et al., 2016). Enfin, le jeu de données ORNL (`https://www.ornl.gov`) propose une capture de réseau à partir de l'IDS open-source Snort qui présente les mêmes attaques que KDD99 (Cisco, 2017). Ces trois ensembles de données sont basés sur la bibliothèque libpcap généralement utilisée pour les captures du trafic réseau (Garcia, 2017). L'acquisition de données reste donc similaire pour ces derniers. Nous pouvons définir cette acquisition comme symétrique, c'est à dire que la volumétrie de capture et de sauvegarde correspond à la quantité des flux entrants et sortants.

2.1 Détection des intrusions par l'apprentissage supervisé

Les réseaux de neurones artificiels (ANN : Analysis Neural Networks) peuvent être utilisés pour détecter les cyber-menaces (Bognar, 2016). Le thème central de cette approche est d'apprendre et de modéliser le comportement en imitant le cerveau humain. Ces expériences ont été menées sur KDD99. Les 41 premiers attributs sont considérés comme des valeurs d'entrée exogènes. Le 49 ^{ième} attribut est la valeur labellisée, simplifiée avec la valeur 1 en cas d'attaque, et 0 pour les activités normales. L'algorithme de rétropropagation (backpropagation) de Levenberg-Marquardt (Levenberg, 1944; Marquardt, 1963) est utilisé comme méthode d'apprentissage supervisé avec dix "layer" cachés. Il peut prédire les attaques et les comportements normaux avec seulement 3% de faux positifs. Cependant, les ANN présentent des inconvénients intrinsèques tels qu'une lenteur de convergence et la nécessité d'une importante quantité de données d'apprentissage. De plus, des problèmes de sur-apprentissage doivent être traités.

Les arbres de décision (DT) peuvent également être utilisés pour la détection d'intrusions. L'algorithme J48 (une extension de C4.5) a été appliqué sur le jeu de données ORNL. Bien que les arbres de décision obtiennent une meilleure précision que les autres méthodes d'apprentissage supervisées telles que Naive Bayes ou Support Vector Machines (Gupta et al., 2016), les principaux inconvénients sont la consommation de puissance de calcul et de mémoire.

2.2 Détection des intrusions par l'apprentissage non-supervisé

La détection d'intrusions basée sur des méthodes de data mining non-supervisées permettent une automatisation complète. Ces dernières n'exigent pas d'intervention d'expert. Lee et al. (1999) ont travaillé sur la classification, du méta-apprentissage et des règles d'association à partir de données d'audit. L'objectif est de calculer des modèles qui capturent avec précision le comportement des intrusions et des activités normales selon une fréquence temporelle. De telles méthodes ont tendance à générer beaucoup de règles d'association et donc a augmenter exponentiellement le niveau de complexité du système. Un modèle améliorant la précision de détection des intrusions a été obtenu en utilisant l'algorithme DBSCAN (Ajboye et al., 2015). Il s'agit d'identifier des points au sein d'une classe (cluster). DBSCAN nécessite deux paramètres, le rayon et les points requis minimum à l'intérieur d'une classe, et ceci pour déterminer sa taille. Le résultat fournit des régions denses. Si certaines instances n'appartiennent à aucune classe, elles sont considérées comme aberrantes. La détermination du rayon ainsi que le nombre de points reste difficile à estimer Cet algorithme nécessite de fortes ressources de mémoire et de calcul.

2.3 Détection des intrusions par une méthode d'apprentissage hybride

L'utilisation des méthodes data mining uniques donne des résultats limités (Tanpure et al., 2016). Ainsi, des approches hybrides ont été proposées ces dernières années. Il s'agit en fait de mixer les deux méthodes d'apprentissage (supervisée et non supervisée). Une IDS a été développée sur un ensemble de méthodes utilisant l'algorithme K-means, les réseaux de neurones flous (FNN) et les arbre de décisions (C4.5). K-means attribue une valeur (38 type d'attaques de KDD99) au cluster avec les centroïdes les plus proches. FNN apprend les paramètres avec une rétropropagation plus rapide que ANN pour obtenir la classification dans les catégories d'attaques KDD99. Enfin, C4.5 utilise la nouvelle sortie d'étiquette de FNN pour classer les

données entrantes et détecter les attaques réseau (Meghana et Dhamdhere, 2015). Les principaux inconvénients de cette approche résident à nouveau dans le besoin d'une grande puissance de calcul et dans la détermination du paramètre k classe de K-means. L'IDS APMINING repose sur un algorithme de règles d'association appliqué à une collection de connexions sans attaque de KDD99 vers des données de trafic entrant (Emna Bahri, 2013). Le résultat fournit deux profils : normal et anormal. Ensuite, la labellisation est affinée avec une méthode supervisée pour obtenir une classification finale (normale ou attaque). Cependant, le processus de génération des règles d'association est relativement long avec une consommation mémoire importante. Pour renforcer l'exploration de données sur KDD99, un moteur d'analyse basé sur quatre parties utilise une IDS, le jeu de données KDD99, la classification Naive Bayes et le K-means (Tanpure et al., 2016). Les données des sondes aident à améliorer la base de données KDD99. Si des données (flux réseau) sont déjà présentes dans la base de données, K-means est utilisé pour déterminer s')il s'agit d'une activité malveillante ou normale. En cas d'activité malveillante, un message d'alerte est envoyé. Si les données sont nouvelles, K-means est également sollicité. Le résultat est envoyé à un classificateur Naive Bayes pour analyser les relations potentielles. Ce principe de fonctionnement est intéressant car il tient compte à la fois du comportement et des relations. Cependant, il est difficile d'ajouter une nouvelle attaque dans la base de données KDD99 sans expertise. Chaque paquet réseau doit être analysé et comparé avec les contenus existants.

Une autre méthode hybride mise au point par Sunita et al. (2016) consiste à utiliser l'ANN et le Fuzzy C-Means (FCM). L'ANN et la rétropropagation sont utilisés pour l'apprentissage supervisé et FCM pour la formation des classes. Le trafic réseau entrant est classé par l'AAN avec six neurones et quatre "layer" cachés. Le résultat est envoyé à la méthode FCM qui est configurée avec cinq classes correspondant à KDD99 (quatre types d'attaques et une de trafic normal). Ce système résout les problèmes relatifs à la classification, mais il n'intègre pas de nouvelles attaques. Ce point est considéré comme fastidieux et chronophage. De plus, il requiert également de l'expertise.

L'utilisation de l'analyse des composants principaux du noyau (KPCA) avec le K-plus proche voisin (KNN) pour la détection d'intrusion a été étudiée par Elkhadir et al. (2016). Cette approche démontre la supériorité du KPCA sur l'analyse du composant principal (ACP). KPCA est une forme non linéaire de l'ACP et effectue une réduction des dimensions. Le résultat du KPCA (taux de détection) est ensuite envoyé à KNN. Enfin, les données sont classées en attaque ou en connexion normale. Le choix de KNN est discutable, puisque cette méthode est coûteuse en calcul et que les données volumineuses sont souvent la norme dans les captures de paquets réseau.

Nous pouvons noter que toutes les études existantes sont basées sur des outils de capture de paquets. Comme nous l'avons déjà souligné, les outils de capture de paquets ont besoin d'expertise et d'une capacité d'absorption de trafic réseau selon la quantité de flux. Afin de reproduire les différents travaux existants, il est obligatoire d'utiliser les outils de capture de paquets et l'ensemble de données comme KDD99 ou NSL-KKD. Dans ces deux ensembles de données, R2L (accès non autorisé) et U2R (accès non autorisé aux privilèges du super utilisateur local) peuvent être considérés comme incomplets ou même hors de propos. Les outils de capture de paquets sont généralement aveugles devant une connexion cryptographiquement sécurisée entre deux hôtes. Les connexions sécurisées rendent difficile l'analyse de l'escalade des privilèges, comme des bugs ou des shellshock (exécuter des commandes arbitraires pour

obtenir un accès non autorisé à un serveur).

3 Approche combinée pour la détection des intrusions

3.1 Motivations et propositions

La gestion de sécurité étant relativement coûteuse, il convient que cette dernière soit abordable par tous et de limiter l'intervention des experts. La détection des anomalies ou des intrusions doit être suffisamment compréhensible afin d'automatiser au maximum les actions en découlant.

Notre étude portera sur quatre phases qui couvrent un spectre relativement large. Ces phases se décomposent de la façon suivante :

— Phase 1 : "Monitoring et visualisation" des données réseau.
— Phase 2 : Analyse des comportements et alertes, phase qui s'appuiera sur des méthodes de Data Mining.
— Phase 3 : "Scoring" des risques et phase d'évaluation.
— Phase 4 : Détermination d'un plan d'actions.

Nous avons opté pour une phase monitoring/visualisation, qui est une approche conventionnelle et une évaluation du risque déterminée par l'analyse du comportement. Enfin, le plan d'action permet d'arrêter toute action anormale. La nouveauté réside principalement dans la petite quantité d'informations nécessaires et la définition des classes de comportement sans une détermination en apriori.

Les travaux présentés dans le précédent chapitre sont basés sur des flux provenant de jeux de données. De ce fait, nous ne pouvons reproduire des résultats similaires car, par défaut, les variables étudiées (durant la phase 1) sont inférieures au nombre de variables utilisées par KDD99 (41 variables). Il est donc judicieux d'utiliser certaines méthodes de Data Mining sur des événements de type "Firewall" pour la détection des anomalies. Le challenge repose sur la possibilité, à partir d'un équipement de filtrage qui par sa nature ne peut délivrer autant d'information qu'une sonde, d'identifier les intrusions.

3.2 Les séquences d'une intrusion

Avant de présenter nos travaux, nous souhaitons décrire brièvement l'organisation d'une attaque. La méthode présentée ci-dessous est dite "éthique", elle est basée sur les cinq étapes suivantes.

1. La reconnaissance, basée sur une recherche d'informations à partir d'Internet.
2. Balayage réseau (inventaire des services et ports, des systèmes d'exploitation et des versions logiciels serveurs utilisés).
3. Obtenir l'accès (exploitation des vulnérabilités et obtenir un accès).
4. Maintenir l'accès (rendre l'accès permanent).
5. Couvrir les traces (effacer et réduire les traces).

La première étape est difficilement détectable car dépendante de la vie numérique d'une entreprise ou de son personnel. Nos travaux seront axés sur les étapes deux et trois. Toute attaque informatique commence généralement par une prise de renseignements (phase 2).

Il existe des méthodes plus détaillées comme la Cyber Kill Chain (Yadav et Mallari, 2016) qui aborde la notion de pivot et implicitement d'APT [1]. Dans les faits, les actions importantes restent similaires.

3.3 Phase 1 : Monitoring et visualisation

Cette phase est utilisée pour fournir une visualisation complète des données aux utilisateurs concernés (ingénieurs de sécurité, analystes de réseau, responsables de la sécurité de l'information). Les captures ou sondes étant relativement lourdes en déploiement, nous avons opté pour l'utilisation des journaux issus d'un Firewall. La mission d'un Firewall est de filtrer le trafic réseau selon une politique basée sur le flux autorisé dans un réseau sur son origine, sa destination et les services souhaités (Al-Shaer et Hamed, 2003).

Grâce à sa position, un Firewall offre une visibilité complète et selon les points suivants :

— Adresse IP source, adresse IP de destination.
— Port de destination et protocole.
— Date à laquelle le Firewall a appliqué une règle de filtrage.
— Numéro de règle de filtrage (ID) et actions du Firewall (acceptées ou rejetées)

Les journaux d'accès sont envoyés à un serveur syslog-ng (Balabit, 2016). Par la suite, un traitement de découpage des différents champs est réalisé via des expressions régulières (PCRE). Le script Afterglow (Marty, 2013) et la librairie Graphviz (ATT, 2016) sont utilisés pour créer des graphiques.

3.4 Phase 2 : Analyse des comportements et alertes

La phase 2 permet l'analyse des données et la détection d'un comportement anormal. Nous exploitons les données décrites dans la section 3.3. Nous effectuons une transformation afin de réduire les modalités. A titre d'exemple, la variable de port de destination dispose de 65535 modalités. Nous choisissons de regrouper les variables dans les trois catégories suivantes :

— les ports inférieurs à 1024 acceptés et refusés ;
— ports allant de 1024 à 65535 acceptés et refusés ;
— ports d'administration (portadm), activité sur les ports d'administration d'un actif (acceptés et refusés).

Nous réalisons un agrégat des actions réalisées par les adresses IP source. Ceci permet de considérer le nombre total de transactions effectuées par la même adresse IP source et le nombre de flux rejetés (action refusée) et autorisés (action autorisée) par le Firewall.

Nous avons demandé l'avis de cinq experts pour déterminer si le comportement observé pouvait être défini comme à risque (étiquetage des agrégats proposés dans le tableau 1).

L'apprentissage supervisé nous donne la possibilité de construire un estimateur qui peut prédire le risque à partir des adresses IP source. L'analyse des experts nous donne une image de la politique de sécurité sous la forme d'un ensemble de données d'apprentissage. Pour obtenir la meilleure précision de détection, nous avons testé différents algorithmes d'apprentissage supervisé en utilisant la validation croisée (10 fold cross validation). Le meilleur résultat a été obtenu en utilisant l'algorithme Random Forest (Table 2).

1. Advanced Persistent Threat : Menace Permanente Avancé

TAB. 1 – *Données agrégées selon l'IP source*

	Variable name	Description
1	ipsrc	Adresse IP source
2	nbr	Nombre d'occurence de l'adresse IP source est presente
	cnbripdst	Nombre de d'adresse IP différentes contactées
	cnportdst	Nombre de ports de destination contactés
	permit	Nombre de d'occurrence autorisées par le Firewall
	deny	Nombre de d'occurrence rejetées par le Firewall
3	inf1024permit	Nombre de port <1024 autorisés par le Firewall
	sup1024permit	Nombre de port $\geq$ 1024 autorisés par le Firewall
	adminpermit	Nombre de ports d'administration (21, 22, 23, 3389, 3306) accepté par le Firewall
	inf1024deny	Nombre de ports <1024 rejetés par le Firewall
	sup1024deny	Nombre de ports $\geq$ 1024 rejetés par le Firewall
	admindeny	Nombre de ports d'administration (21, 22, 23, 3389, 3306) rejetés par le Firewall
4	risk	Variable à prédire et étiquetée par l'expert

TAB. 2 – *Comparaison des taux d'erreur en apprentissage supervisé*

Algorithme	Taux d'erreurs (%)
CART	9.09
C4.5	11.3
C5.0	8.5
Random forest	8.2
Boosting (Adaboost)	14.3

Durant cette phase et comme dans la plupart des travaux, nous nous concentrons uniquement sur l'adresse IP source. Or, il existe des systèmes d'anonymisation comme "Tor" ou l'utilisation de VPN anonymes. Malgré cela, notre premier cycle d'analyse nous informe que le système d'information (IP de destination) est analysé en vue d'une potentielle intrusion.

En utilisant deux méthodes d'apprentissage différentes, il devient possible d'avoir une vue d'ensemble sur les dérivations de comportements ainsi que sur les actifs visés (Meesala et Xavier, 2015). La première étape consiste à créer un jeu de données et d'extraire les flux reçus sur chaque serveur. L'étape suivante repose sur l'identification des comportements différents. Nous voulons regrouper le comportement d'IP source en fonction de l'adresse IP de destination. Cette analyse de classe s'articule autour du concept de placement d'un ensemble d'objets dans le même groupe ou classe. En suivant cette méthodologie, il est possible d'identifier et de vérifier tout écart de comportement.

Le principal problème avec les méthodes de clustering est de déterminer le nombre de classes à utiliser. Pour trouver la meilleure méthode déterminant le bon nombre de classes, nous utilisons la validation interne. Nous avons testé différents algorithmes et après avoir analysé les résultats, nous avons opté pour le Partitioning Around Medoids (PAM) décrit par Lamiaa et Manal (2013). Cette méthode a fourni le meilleur résultat par sa rapidité et l'ensemble des définitions de k classe. La figure 1 montre les résultats obtenus par rapport aux méthodes utilisant k classe en apriori. Le temps de calcul pour la détermination des classes pour 53 serveurs a été de 0.353 secondes. Ce résultat est utilisé en tant que référentiel. En d'autres termes,

chaque serveur (actif) dispose d'un nombre de classes de comportement et toute dérivation sera considérée comme anormale. Nous avons réalisé différents tests d'activité malveillante et nous avons pu constater une diminution des classes sur les actifs visés. Ceci confirme le principe d'attaque. Pour confirmer le résultat obtenu, nous avons utilisé le coefficient de variation (CV). Il permet d'obtenir un score Breunig (2001) et il est possible de se focaliser sur l'actif subissant le comportement le plus déviant.

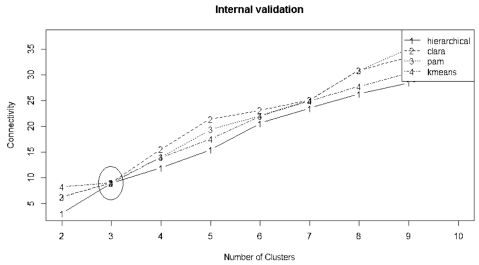

FIG. 1 – *Validation interne pour un serveur*

3.5 Phase 3 : Scoring du risque et évaluation

Il existe plusieurs méthodes d'analyse de risques. Dans les faits, peu importe la méthode comme le précise la norme ISO 27001 (ISO 27001 :2013 chapitre 9.1 point b) du moment que cette dernière soit documentée et reproductible. Nous utilisons la méthode EBIOS qui traite de l'analyse contextuelle en fonction de la dépendance du système d'information (DCSSI, 2003). Il ne s'agit pas de décrire une méthode d'évaluation des risques, mais de savoir comment utiliser le résultat obtenu et de l'inclure dans notre cadre de recherche. La figure 2 montre un graphique réalisé à partir d'une analyse de risques EBIOS. Il a été décidé que l'acceptation du risque résiduel est fixée à un niveau inférieure à 6.

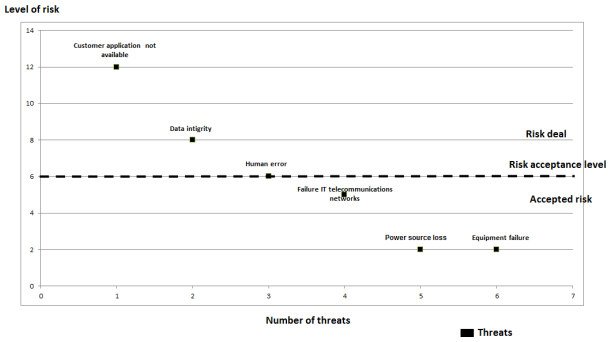

FIG. 2 – *Exemple graphique d'une analyse de risques*

Nous pouvons utiliser le niveau d'acceptation du risque de chaque actif comme indicateur de gravité. La détection du nouveau comportement à partir d'une adresse IP source identifiée comme à risque (phase 2 : Random Forest et PAM) pourra être priorisée (score du risque phase 3). Avec ces trois différentes informations, nous construisons un indicateur d'attaque (IOA : violation de la politique de sécurité phase 2, dérivation du comportement phase 2).

3.6 Phase 4 : Détermination du plan d'actions

A l'aide de la phase 3, nous savons exactement quels actifs protéger selon le "scoring" du risque. Il est souvent recommandé de réaliser une évaluation des vulnérabilités et des tests de pénétration.

Les audits visent à indiquer la gravité des vulnérabilités. Une vulnérabilité est qualifiée de faible, moyenne, élevée ou critique selon un catalogue de menaces de sécurité connues. Au cours de cette phase, nous utilisons le référentiel construit dans les phases 2 (classe IP de destination) et 3 (évaluation des risques). Nous implémentons les quatre qualifications d'audit qui correspondent à la gravité de la vulnérabilité sur chaque actif (IP de destination). Pour ce faire, nous utilisons le scanner de sécurité OpenVas largement utilisé par la communauté de la sécurité. Le rapport d'audit est exporté au format XML et intégré dans notre référentiel. De plus, à la suite du rapport, les CVE (vulnérabilités et expositions communes) sont listées. Ces dernières sont une liste de vulnérabilités avec les actions à effectuer pour corriger les faiblesses de sécurité établies. A l'issue de cette phase, nous obtenons le résultat suivant :

— serveur (IP de destination) nombre de classes ;
— numéro de niveau de risque ;
— nombre de vulnérabilités élevées trouvées ;
— nombre de vulnérabilités moyennes trouvées ;
— nombre de vulnérabilités faibles trouvées.

La notation des risques et le plan d'actions doivent permettre d'identifier et de prioriser une série d'actions prédéterminées.

4 Preuve de concept (POC)

4.1 Cas d'utilisation

Nous avons travaillé sur l'architecture d'une entreprise du domaine de la santé constituée de plusieurs dizaines d'employés. Notre analyse se concentre sur trois réseaux interconnectés au sein d'un réseau étendu (WAN), et protégé par un équipement de filtrage. Ainsi, nous avons pu créer plusieurs jeux de données incluant des attaques dans une échelle temporelle relativement longue (d'une à quarante huit heures).

4.2 Résultats

L'exécution de la Phase 1 permet de visualiser l'activité du réseau avec un outil graphique, avec une meilleure compréhension que l'examen des données brutes. Nous estimons que les résultats peuvent être considérés comme satisfaisants. Nous proposons une interface simple dans laquelle les flux de données réseau peuvent être évalués (diagnostic ou pertinence de la politique de filtrage). Nous avons intégré les journaux Firewall vers un conteneur Syslog-Ng et nous avons appliqué l'apprentissage supervisé de Random Forest. Le résultat fournit une variable "risque" avec 2 modalités (Oui et Non). Nous pouvons détecter une activité malveillante (Table 3, ligne 3) : le flux autorisé de 0,7 % pourrait être défini comme un marqueur d'une intrusion. Lors des tests de méthodes non supervisées, l'ACP et le regroupement agglomératif hiérarchique ont fourni une bonne visualisation graphique. Par conséquent, nous avons décidé d'intégrer ces fonctions dans notre outil D113. De plus, l'évaluation visuelle de la technique

TAB. 3 – *Aggregate flow with risk analysis result*

Sum	Action denied	Action allowed	Inf 1024	Sup 1024	Adm ports	Risk
16	0.0	100.0	0.0	0.0	0.0	No
12	0.0	100.0	0.0	0.0	0.0	No
3296	99.3	**0.7**	61.9	38	0.1	**Yes**
36	100.0	0.0	0.0	100	0.0	Yes

de la tendance (VAT), pour identifier visuellement le nombre de classes (Bezdek et Hathaway, 2002) nous donne une confirmation graphique du nombre de classes. Les trois graphiques (PCA, VAT, Dendrogram) montrent l'existence de 3 classes. PAMK donne les mêmes résultats sans aucune interprétation humaine.

4.3 Retour d'experts

Le tableau 4 montre le retour des experts sur l'utilité de notre système, suivant l'approche de Ghoniem et al. (2014). Nous avons demandé à cinq experts (deux ingénieurs de sécurité E1 et E2, un chef de la sécurité des informations E3, un consultant en sécurité E4 et un analyste de réseau E5) d'évaluer sur une note comprise entre 0 et 5 les quatre phases de notre approche. De leur point de vue, l'outil D113 offre la possibilité d'accéder à tous les flux rejetés, y compris le balayage des ports et les attaques par force brute. La représentation des flux acceptés donne un aperçu global intéressant.

TAB. 4 – *Overview of expert feedback*

Question about the usefulness of	E1	E2	E3	E4	E5
Visualization (Phase 1)	5	4	3	4	4
Policy derivation (Phase 2)	5	4	4	4	5
Behavior derivation (Phase 2)	5	5	5	4	5
Risk management (Phase 3)	3	4	4	2	3
Action plan (Phase 4)	3	4	3	4	3

La combinaison de l'apprentissage supervisé et non supervisé ainsi que la notation des risques permet d'identifier les serveurs affectés et leurs importances dans le système d'information. L'utilité des Phases 3 et 4 n'a pas été remise en question, mais nécessite un suivi régulier et un travail supplémentaire.

5 Conclusion et perspectives

Nous avons proposé IDS mixant les apprentissages supervisés et non supervisés. En utilisant ceci, nous sommes capables de détecter les violations de la politique de sécurité et les dérivations de comportement. L'ajout de l'analyse des risques et de l'audit de vulnérabilité permettent de nous concentrer sur les serveurs les plus sensibles avec les vulnérabilités clairement identifiées. Par conséquent, la connaissance complète du système d'information est moindre. Selon les experts, la manière de présenter l'information (IP source à risque, modification du comportement sur l'IP de destination et notation des risques) doit être plus facile à comprendre. De plus, les attaques orientées WEB doivent être prises en compte. Les algorithmes Bagging, K-mean++ doivent être testés pour améliorer les méthodes d'apprentissage.

Références

Ajboye, A. et al. (2015). Anomaly Detection in Dataset for Improved Model Accuracy Using DBSCAN Clustering Algorithm. pp. 39–46.

Al-Shaer, E. et H. Hamed (2003). Firewall policy advisor for anomaly detection and rules. In *International Symposium on Integrated Network Management*, Volume 118, pp. 17–30.

ATT (Last accessed September 10, 2016). Graphviz – Graph Visualization Software. http://www.graphviz.org.

Balabit (Last accessed September 03, 2016). Reliable, scalable, secure central log management. https://www.balabit.com/log-management.

Bezdek, J. C. et R. J. Hathaway (2002). VAT: A Tool for Visual Assessment of (Cluster) Tendency. In *International Joint Conference on Neural Networks*, pp. 2225–2230.

Bognar, E. (2016). Data mining in cyber threat analysis neural networks for intrusion detection. Volume 15, pp. 187–197.

Breunig, R. (2001). An almost unbiased estimator of the coefficient of variation. *Economics Letters*, 15–19.

Cisco (2017). Snort. https://www.snort.org.

David Pierrot, Nouria Harbi, J. D. (2016). Hybrid intrusion detection in information systems. In *3rd International Conference on Information Science and Security (ICISS 16), Pattaya, Thailand*, pp. 27–32.

DCSSI (2003). The EBIOS method – Expression of Needs and Identification of Security Objectives. https://www.ssi.gouv.fr/archive/en/confidence/documents/methods/ebiosv2-methode-plaquette-2003-09-01_en.pdf.

Deepa, A. J. et V. Kavitha (2012). A comprehenseive survey on approaches to intrusion detection system. *International Conference on Modelling Optimization and Computing 38*, 2063 – 2069.

Elkhadir, Z. et al. (2016). Intrusion Detection System Using PCA and Kernel PCA Methods. *International Journal of Computer Science 43*, 72–79.

Emna Bahri, N. H. (2013). Real detection intrusion using supervised and unsupervised learning. pp. 321–326.

Garcia, L. M. (2017). TCPdump and Libpcap. http://www.tcpdump.org.

Ghoniem, M. et al. (2014). VAFLE: Visual Analytics of Firewall Log Events. In *Visualization and Data Analysis*.

Gupta, M. et al. (2016). Intrusion Detection Using Decision Tree Based Data Mining Technique. *International Journal for Research in Applied Science and Engineering Technology 4*, 24–28.

H.Rais et T.Mehmood. (2018). Dynamic ant colony system with three level update feature selection for intrusion detection, injs. *nternational Journal of Network Security, 20*, 184–192.

Lamiaa, F. et H. Manal (2013). Using Modified Partitioning Around Medoids Clustering Technique in Mobile Network Planning. *International Journal of Computer Science Issues 9*,

10–25.

Lee, W. et al. (1999). A data mining framework for building intrusion detection model. In *IEEE Symposium on Security and Privacy*, pp. 120–132.

Levenberg, K. (1944). *A method for the solution of certain problems in least squares*, Volume 2.

Marquardt, D. (1963). *An algorithm for least-squares estimation of nonlinear parameters*, Volume 11.

Marty, R. (2013). AfterGlow. http://afterglow.sourceforge.net.

Meesala, S. et B. Xavier (2015). A Hybrid Intrusion Detection System Based on C5.0 Decision Tree and One-Class SVM. *International Journal of Current Engineering and Technology 5*, 59–70.

Meghana, S. et V. Dhamdhere (2015). Hybrid approach for Intrusion Detection Using Data Mining. *International Journal of Innovative Research in Science, Engineering and Technology 4*, 5588–5595.

Nguyen, H. et al. (2011). An efficient Local Region and Clustering-Based Ensemble System for Intrusion Detection. In *15th International Database Engineering and Applications Symposium*, Volume 185–191.

Sunita, S. et al. (2016). A Hybrid approach of Intrusion Detection using ANN and FCM. *European Journal Advances in Engineering and Technology 3*, 6–14.

Tanpure, S. et al. (2016). Intrusion detection system in data mining using hybrid approach. *National Conference on Advances in Computing, Communication and Networking 5*, 18–21.

University of California, Irvine (1999). KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Yadav, S. et D. Mallari (2016). Technical Aspects of Cyber Kill Chain. *Communications in Computer and Information Science 536*, 438–452.

Zambon, E. et D. Bolzoni (2006). Network intrusion detection systems.

Summary

The consequences of an intrusion into an information system can be problematic for the existence of a company or an organization. The impacts are synonymous with financial loss, brand loss and seriousness. The detection of an intrusion is not an end in itself, the reduction of the delta detection-reaction has become a priority. We propose a method dealing with the technical aspects by the use of a hybrid method of data mining but also the functional aspects. The addition of these two aspects makes it possible to obtain a general vision on the hygiene of the information system but also an orientation on the monitoring and the corrections to be made.