C&ESAR 2014

Computer & **E**lectronics
Security **A**pplications
Rendez-vous

Détection et réaction face aux attaques informatiques

24-26 novembre 2014
Rennes - France

http://www.cesar-conference.org

Table des matières

Introduction C&ESAR 2014 : Détection et réaction face aux attaques informatiques	Hervé Debar - Yves Correc - Benoît Martin - Olivier Heen	2
De la cyberveille à la prévision des agressions	Thierry Berthier – Olivier Kempf	5
Behavioral Detection of Internet Frauds against Online Service Platforms	Nizar Kheir - Sok-Yen Loui - Ndeye-Seynabou Diop - François Olivier-Martin - Vincent Frey	23
Les attaques avancées : Constats réels et Tendances	Thibaud Signat	37
Utilisation de l'hétérogénéité des réseaux de capteurs sans fil pour accroître la résilience de la solution de détection d'intrusion	David Espesa - Nora Cuppens - Frédéric Cuppens - Philippe Le Parc	55
Cyber attacks in the guided transport domain	Christophe Gransart - Christian Pinedo - Marina Aguado - Marc Heddebaut - Eduardo Jacob - Igor Lopez - Marivi Higuero	69
Détection d'intrusion dans les systèmes industriels: Suricata et le cas de Modbus	David Diallo - Mathieu Feuillet	80
Architecture système sécurisée de sonde IDS réseau	Pierre Chifflier - Arnaud Fontaine	97
RoCaWeb - Reconstruction de spécifications pour la détection d'intrusion Web	Yacine Tamoudi - Djibrilla Amadou Kountché - Alain Ribault - Sylvain Gombault	111
Après l'attaque	Philippe Davadie	127
Comment accélérer la mise en mouvement des organisations au niveau Cyber Sécurité ou l'apport decisif d'une approche quantitative innovante	Gérard Gaudin	147
Catégorisation par objectifs de la visualization pour la sécurité	Christopher Humphries - Nicolas Prigent - Christophe Bidan - Frédéric Majorczyk	155
Dynamic Design of non Conflicting Responses against Simultaneous Attacks	Lea Samarji - Nora Cuppens-Boulahia - Frédéric Cuppens - Wael Kanoun - Samuel Dubus	171
Remediating Logical Attack Paths Using Information System Simulated Topologies	François-Xavier Aguessy - Lucie Gaspard - Olivier Bettan - Vania Conan	187
Vers une architecture « big-data » bio-inspirée pour la détection d'anomalie des SIEM	Véronique Legrand - Pierre Parrend - Pierre Collet - Stéphane Frénot - Marine Minier	205
Automatiser la construction de règles de corrélation : prérequis et processus	Erwan Godefroy - Eric Totel - Michel Hurfin - Frédéric Majorczyk - A. Maaroufi	223
Détecter et réagir face aux cyber-attaques – Retour d'expérience d'une expérimentation technico-opérationnelle multinationale	Jean-Luc Gibernon - Yann Pitault	233
Posture de sécurité dynamique	David Bizeul	245
S'inspirer d'autres domaines pour améliorer sa performance de traitement des incidents	Nicolas Loriot	259

C&ESAR 2014 : Détection et réaction face aux attaques informatiques

Pour sa 21ème édition, le thème des journées C&ESAR 2014 est la lutte contre les cyber-menaces, couvrant deux aspects spécifiques, les problèmes de la détection de ces cyber-menaces d'une part, et de la réaction d'autre part.

La détection, dite historiquement « détection d'intrusions », est un domaine de recherche actif depuis le début des années 1980, avec la publication du rapport de James Anderson et les travaux de Dorothy Denning dans le projet IDES. Les sondes de détection d'intrusion sont des technologies déployées opérationnellement depuis la fin des années 1990. Ces sondes font aujourd'hui partie de la panoplie des outils des professionnels de la sécurité. Les activités de recherche sur la détection se sont également fortement développées à ce jour, suivant les évolutions des systèmes et des réseaux : détection de code malveillant, détection des attaques dans les réseaux de capteurs, détection des attaques par déni de service, détection d'attaques contre des applications spécifiques, etc.

Dans la continuité du déploiement des sondes de détection d'intrusion sont apparues au début des années 2000 les plates-formes de gestion de la sécurité, puis les centres opérationnels de sécurité permettant d'externaliser la gestion des alertes issues des sondes de détection et la réaction associée. Les motivations pour le développement de ces plates-formes ont été à la fois technologiques et économiques. Du côté de la recherche et du développement technologique est apparu le besoin de traiter des volumes d'alertes très importants, dans un contexte ou les sondes émettent des alertes qui sont peu explicites individuellement. Du côté économique, la mise en évidence de la difficulté à opérer des sondes, a conduit à la conception de centres de sécurité spécialisés dans le traitement des alertes, et au développement d' une activité économique sur l'externalisation de la supervision de sécurité.

Depuis le milieu des années 2000, une activité de recherche se développe également autour de l'automatisation des contre-mesures. Malgré la difficulté de gestion manuelle des incidents de sécurité, cette activité est encore loin d'un déploiement opérationnel, à l'exception de la lutte contre les attaques par déni de service à grande échelle dans les réseaux. Cette expérience opérationnelle d'une vingtaine d'années permet de dresser le panorama suivant :

1. Les sondes permettent de détecter des attaques, mais il demeure toujours une part d'attaques non détectées, qui sont perçues comme les plus dangereuses. Il semble vraisemblable qu'il existe des attaques de grande ampleur dans les réseaux actuels, qui ne sont pas visibles à ce jour. Il est certain que ces attaques seront détectées, mais il est impossible aujourd'hui de savoir quand, ou de quelle manière.
2. Même lorsqu'elles sont détectées par les sondes, certaines attaques donnent lieu à compromission car les alertes ne sont pas ou mal traitées. Une bonne partie de la détection reste à ce jour le fait d'une manifestation externe, qui peut être liée à l'attaquant ou aux utilisateurs légitimes. La détection explicite des attaques par des sondes dédiées à cet usage reste mécanique. Du coup, la

perception d'une faiblesse des sondes persiste, alors qu'il s'agit plutôt d'une faiblesse du diagnostic.

3. Les opérationnels demeurent très réticents par rapport au concept de réaction automatisée, alors qu'ils continuent à se plaindre du nombre et de la faible qualité des alertes à traiter. Seul le domaine des attaques en déni de service par saturation ont donné lieu au déploiement de solutions opérationnelles pour lutter contre ces attaques, par la destruction du trafic correspondant.

Il apparait donc nécessaire de faire le point sur les technologies existantes pour détecter et réagir face aux cyber-menaces, et de proposer des usages et de nouveaux développements afin de les améliorer. Le programme de C&ESAR 2014 reprend donc les points de ce panorama, en parcourant les axes suivants :

1. La détection : malgré les travaux constants de R&D consacrés au sujet, l'émergence de nouvelles attaques et de nouveaux services fait qu'il est important de continuer à développer et valider de nouvelles sondes. Le programme s'intéressera à de nombreux aspects de la détection, couvrant différents types d'environnements et de cas d'usage.
2. Les SIEM : ces plates-formes devenant le cœur de nombreux centres opérationnels de sécurité, nous aborderons l'usage de différentes plates-formes au travers de cas concrets. Le programme inclut également une table ronde qui sera l'occasion de débattre sur l'efficacité de la détection.
3. La remédiation : en complément des sujets précédents orientés usage industriel, cette session couvrira des aspects de R&D liés à l'usage, au diagnostic et à la corrélation.
4. La réponse : finalement, nous aborderons le problème de la réaction, au travers de différents exemples, et en concluant par un débat sur la maturité et les contraintes de la réponse.

Les présidents du comité d'organisation et du comté de programme tiennent à remercier chaleureusement tous les acteurs qui ont encore une fois rendu possible notre rendez-vous annuel : les conférenciers, les membres des deux comités, les organisateurs, et tous nos fidèles partenaires sans qui cette manifestation ne pourrait avoir lieu. En leur nom, nous vous souhaitons une excellente conférence.

Hervé DEBAR (Télécom-SudParis), Président du comité de programme
Yves CORREC (ARCSI), co-Président du comité d'organisation
Benoît MARTIN (DGA-MI), co-Président du comité d'organisation
Olivier HEEN (Technicolor), Directeur de publication

Comité d'organisation

Yves CORREC, président	(ARCSI, France)
José ARAUJO	(ANSSI, France)
Boris BALACHEFF	(HP Labs, France)
Benoit MARTIN	(DGA-MI, min. Défense, France)
Florent CHABAUD	(DGSIC, min. Défense, France)
Olivier HEEN	(Technicolor, France)
Ludovic MÉ	(Supélec, France)
Ludovic PIETRE-CAMBACEDES	(EDF, France)
Éric WIATROWSKI	(Orange, France)

Comité de programme

Hervé DEBAR, président	(Télécom SudParis, France)
Marie-Thérèse ANDRE	(min. Défense, France)
Olivier BETTAN	(Thalès, France)
Guillaume BONFANTE	(LORIA, France)
Jérémy BUISSON	(IRISA – CREC, France)
Pierre CARON	(Orange Labs, France)
Yves CORREC	(ARCSI, France)
Frédéric CUPPENS	(Télécom Bretagne, France)
Marc DACIER	(SYMANTEC, France)
Gérard GAUDIN	(Club R2GS, France)
Christian GUERRINI	(SOGETI, France)
Nicolas GUILLERMIN	(min. Défense, France)
Olivier HEEN	(Technicolor, France)
Sébastien HEON	(CASSIDIAN, France)
Grégoire JACOB	(Lastline, USA)
Frédéric LE BASTARD	(La Poste, France)
Jean LENEUTRE	(Telecom ParisTech, France)
Ludovic MÉ	(Supélec, France)
Benjamin MORIN	(ANSSI, France)
Vincent NICOMETTE	(CNRS/LAAS, France)
Ludovic PIETRE-CAMBACEDES	(EDF, France)
Guillaume ROSSIGNOL	(BULL, France)
Jouni VIINIKKA	(6Cure, France)
Éric WIATROWSKI	(Orange, France)

De la cyberveille à la prévision des agressions

Conférence C&ESAR 2014

Thierry Berthier – Olivier Kempf

Abstract. En forte augmentation ces dernières années, les agressions informatiques perturbent et déforment le cyberespace, obligeant concepteurs et architectes à réinventer constamment la sécurité des systèmes d'information. Le Cloud Computing et le déferlement annoncé des objets connectés vont fournir de nouvelles opportunités d'agressions à la communauté du hacking qui sait parfaitement profiter, en temps réel, des progrès technologiques pour détecter puis exploiter leurs vulnérabilités. Cette communauté a tendance à se structurer et à fédérer ses forces dans l'objectif de mener à bien des cyberagressions de plus en plus nombreuses, organisées et lucratives.

On assiste à une « industrialisation » de la production des menaces sur le cyberespace. Celle-ci accompagne de façon cohérente l'augmentation exponentielle du volume mondial des données et la numérisation de toutes les sphères d'activités humaines. La cyberagression apparaît alors comme un effet collatéral, systémique et indissociable du développement du cyberespace. Si la détection des menaces constitue bien l'un des enjeux majeurs de la cybersécurité, la prévention fondée sur une veille généralisée des projections algorithmiques ouvertes ou traces numériques produites par les hackers semble également pouvoir répondre aux problématiques de sécurité. La projection algorithmique d'un individu sur un système selon un algorithme contient l'ensemble des données archivées lors de l'exécution de cet algorithme sur le système, décidée ou provoquée par l'individu. Elle permet de classer les données produites par un individu selon l'algorithme qui les a engendrées.

Nous montrons que dans certains cas, en exploitant le contenu des projections, il devient possible de « prévoir » une agression, sa cible et sa temporalité avec un indice de confiance raisonnable.

Keywords : prévision, cyberdéfense, hacking, analyse

1 Des agressions toujours plus nombreuses et mieux planifiées

1.1 Quelques chiffres

Selon une récente étude de l'éditeur de sécurité FireEye [1], le rythme des agressions avancées a doublé depuis un an. Les Menaces Persistantes Avancées (APT) sont des agressions ciblées, parfois organisées par des États et conçues pour endommager ou voler des informations sensibles. Elles sont souvent difficiles à identifier, peuvent échapper à la détection durant des mois et se décomposent en quatre phases :

- Une phase de reconnaissance qui se traduit par une investigation relative aux vulnérabilités de la cible, notamment des requêtes de domaines et des analyses des ports.
- Une phase d'entrée initiale qui recouvre l'ingénierie sociale, l'hameçonnage dirigé, l'analyse des comportements humains associés à la cible.
- Une phase d'élévation de privilège et de contrôle étendu qui intervient lorsque l'agresseur pénètre dans le périmètre réseau et qu'il tente d'acquérir des privilèges et un contrôle accru sur des systèmes critiques. Cette phase peut inclure l'installation d'outils de type « porte dérobée » afin de simplifier les prochains accès au réseau.
- Une phase d'exploitation continue qui, une fois le contrôle établi, permet à l'agresseur d'exporter des données sensibles de façon continue.

Les agressions APT qui procèdent par contournement des défenses logicielles et matérielles ont doublé entre 2012 et 2013 passant d'une APT dans le monde toutes les 3 secondes en 2012 à une APT toutes les 1.5 secondes en 2013.

L'expansion mondiale des cyberagressions est favorisée par la construction et la diffusion rapide de logiciels malveillants, et la banalisation des processus de mise en œuvre des agressions numériques. Toujours selon l'étude FireEye [1], de plus en plus de nations sont à l'origine de ces agressions : 206 pays en 2013 contre 184 en 2012. Les États-Unis, l'Allemagne, la Corée du Sud, la Chine, la Russie, le Royaume-Uni et les Pays Bas restent les leaders en termes de production d'agressions. Ces nations figurent également parmi les plus visées par les cyberagressions. En 2013 et 2014, les services publics, les secteurs de haute technologie et les services financiers apparaissent comme les cibles privilégiées des attaquants. L'exploitation des vulnérabilités du langage java, des navigateurs ou de "zero-day" engendre cinq fois plus d'agressions que les vulnérabilités liées au trafic mail.

Durant le second trimestre 2013, 24 millions de cyberagressions ont été enregistrées.

1.2 Les limites d'une défense périmétrique

Ces volumes croissants d'agressions obligent les responsables de la sécurité des systèmes d'information (RSSI) à mettre en œuvre des politiques de sécurité extrêmement réactives adaptées à des menaces toujours plus dynamiques, furtives et généralement très peu prévisibles. Des firewall « intelligents » capables de détecter la menace et l'activité malveillante commencent à être développés en fondant leurs analyses autonomes sur un processus d'apprentissage automatique des situations d'agression et sur des approches bayésiennes.

Si ces solutions strictement technologiques apportent un progrès réel en termes de sécurité active, elles ne prétendent pas « tout » détecter et sont par définition insensibles à des agressions furtives exploitant des failles « zeroday » ou construites précisément pour passer l'obstacle d'une détection automatisée de la menace. Le même problème limite d'ailleurs la construction d'antivirus polyvalents. Ces derniers, souvent fondés sur des bibliothèques de signatures, se révèlent insensibles à des codes

inédits (bien que rares parmi l'ensemble des virus), développés dans l'objectif de duper la protection logicielle. Le théorème de non décidabilité de la détection virale dû à Fred Cohen [2] affirme qu'il n'existe pas d'antivirus absolu détectant toutes les charges virales et toutes les menaces numériques. La détection absolue par un antivirus est donc une impossibilité purement mathématique. Ce résultat profond vient définitivement borner l'ambition des constructeurs de solutions universelles de protections logicielles et confirme formellement la forte complexité d'une automatisation de la détection des menaces. Les agressions utilisant le vecteur des vulnérabilités humaines et l'ingénierie sociale restent elles aussi, par définition, peu détectables avant qu'elles ne produisent leurs effets sur les systèmes.

Ces limitations techniques doivent donc nous interroger sur l'approche globale de la cybersécurité :

Comment compléter efficacement les dispositifs automatisés de détection des cybermenaces ?

Peut-on faire évoluer une stratégie de défense périmétrique essentiellement algorithmique et physique vers une stratégie composite associant une veille analytique humaine à une infrastructure logicielle-matérielle classique constituant ainsi une défense anticipative ? Pour cela, une solution consisterait à privilégier une surveillance systématique des pratiques numériques des auteurs d'agressions et développer des outils permettant l'analyse de leurs projections algorithmiques.

2 Une veille ciblée pour une meilleure évaluation du risque

2.1 Cellules et communautés de cellules

L'idée maîtresse de l'article réside dans la mise en place d'une veille systématique, la plus exhaustive possible, des espaces de communication utilisés quotidiennement par les cellules de hackers pour revendiquer des exploits passés ou pour préparer de nouvelles campagnes agressives.

En fournissant des données statistiques pertinentes, des tendances et des potentialités d'agressions, cette veille vient compléter les infrastructures algorithmiques et matérielles présentes sur les systèmes. Elle peut ainsi orienter ou augmenter la sensibilité et l'acuité des différents capteurs et sondes qui composent la couche défensive d'un système. Les données collectées par la veille sont issues des projections algorithmiques ouvertes, volontaires et systémiques créées par les attaquants [3]. Leurs formulations doivent être suffisamment précises pour pouvoir s'insérer efficacement dans une démarche prédictive-préventive et contribuer à repousser finalement l'agression.

Aujourd'hui, de nombreux hackers organisent leurs activités au sein de cellules polyvalentes ou au contraire, spécialisées dans un seul type d'agression (DDOS, hameçonnage, rançonnage, défacement..).

Composée d'un ou plusieurs individus actifs, la cellule constitue le premier niveau de structuration de l'entité agressive. Elle s'associe souvent à une communauté de

cellules liées entre elles par des intérêts communs, des croyances similaires, des convictions politiques ou religieuses partagées.

Un hacker peut appartenir à plusieurs cellules distinctes ou bien agir en solitaire au sein de sa propre cellule. Lors de campagnes d'envergure comme celles d' OPIsraël I et II orchestrées par Anonymous [4] en 2013 et 2014, de nombreuses cellules se sont agrégées autour d'un noyau actif pour former une communauté de cellules et mener des agressions coordonnées. Des centaines de sites web de l'Internet israélien ont été ciblés en soutien à la cause palestinienne.

2.2 Revendications et réputations

Les agresseurs organisés en cellules n'hésitent plus à revendiquer haut et fort chaque agression numérique victorieuse et contribuent ainsi à l'instauration d'un contexte mondial permanent de défi et de concurrence. Une structuration en cellules permet de rationaliser le coût de développement d'une campagne d'agressions ou de l'industrialiser en optimisant et mutualisant les moyens de sa mise en œuvre. Elle facilite ensuite l'émergence d'une réputation au sein des sphères du hacking car la montée en puissance et la reconnaissance d'un groupe constitué s'effectue beaucoup plus rapidement que celle d'un agresseur isolé. Cette mécanique réputationnelle minore la probabilité d'un hacker isolé assez habile pour construire seul une agression complexe de type APT.

La réputation d'une cellule s'établit également au regard des cibles qu'elle va définir : plus cette cible apparaît comme résistante ou représentative aux yeux de la communauté et plus l'agression sera valorisante et valorisée dans la compétition que se livrent les groupes de hackers. Une fois l'agression menée à bien, les phases de validation et de revendication viennent renforcer la réputation de la cellule et lui permettent de garnir son carnet de trophées numériques. La notoriété d'une cellule se mesure alors à son volume d' « hacktivité » annuelle et à la nature et qualité des cibles visées par ces agressions.

La réputation opère comme une mesure de compétence puis induit une hiérarchie au sein de la sphère mondiale du hacking. Réputation et hiérarchie facilitent la construction d'une communauté de cellules fédérées autour d'un projet commun d'agression. Nous nous trouvons ainsi en présence d'un processus réputationnel qui emprunte à trois secteurs distincts :

- Le sport : en effet, le sport introduit une compétition qui détermine un classement des participants. Développer sa notoriété permet de progresser dans le classement et donc d'obtenir d'autres gains (honneur ou ressources financières).
- La recherche : les chercheurs publient régulièrement leurs travaux afin certes de faire progresser la science (ici, le hacker tente de découvrir de nouvelles failles) mais aussi pour asseoir la renommée scientifique.
- L'économie : la réputation permet d'établir une cotation de marché qui permettra de fixer le prix de prestations annexes (le plus souvent cachées).

Ainsi, en faisant valoir sa réputation sur des actes semi-publics, on établit sa valeur (compétitrice, technique ou marchande) qui sera éventuellement exploitée dans la constitution d'équipes dédiées à des projets particuliers : soit pour des motifs politiques soit pour des motifs commerciaux en fonction des objectifs de chaque cellule de hackers. La cellule à faible notoriété sera honorée d'être acceptée et d'apporter sa contribution algorithmique à la communauté surtout si celle-ci est organisée autour d'un noyau de cellules à forte réputation. Cette contribution, si elle est pertinente, viendra renforcer l'image de la cellule débutante. Réciproquement, les cellules disposant d'une notoriété importante sauront déléguer en toute confiance une partie de l'activité de programmation et de préparation de l'opération à des cellules subordonnées (soit qu'elles soient spécialisées soit qu'il s'agisse de cellules innovantes qui désirent entrer dans le marché réputationnel). Elles pourront ensuite concentrer leurs efforts sur la supervision et la coordination de l'agression. Ces mécanismes de coopération profitent ainsi aux deux familles de cellules : ils facilitent et libèrent l'action des cellules fortes et augmentent la notoriété des cellules faibles.

2.3 L'exemple de zone-h pour le défacement

Un site comme « zone-h » [5] constitue une référence mondiale importante dans la revendication, la validation et l'archivage des cyberagressions. Il offre aux hackers de toute nationalité un espace d'enregistrement et d'homologation des agressions par défacement (le défacement désigne la modification non sollicitée d'une partie d'un site web à la suite d'un piratage). Avec plus de 90 000 utilisateurs inscrits, zone-h a référencé en 10 ans environ 10 millions de défacements homologués. Lorsqu'une cellule réalise un défacement, elle transmet immédiatement l'adresse (url) du site ciblé puis zone-h enregistre les pages modifiées sur un site miroir afin d'en conserver la trace. L'homologation de l'agression intervient sous réserve que zone-h ait vérifié la réalité du défacement.

Ce mécanisme d'enregistrement contribue pleinement à la construction et au renforcement de l'identité, de la popularité et de l'image d'une cellule. Les « faits d'armes » numériques des groupes de hackers y sont méticuleusement décrits et archivés sur un espace de stockage dédié qui contient les caractéristiques techniques et les messages de revendication ou de motivation de l'agression.

Ces dernières peuvent être d'origine politique, religieuse, ou relevant simplement d'un défi ludique.

Les cellules de hacking qui pratiquent le défacement effectuent souvent d'autres types d'agressions plus lucratives (DDOS, vol de données bancaires, rançonnage) qui ne donnent pas lieu à une revendication directe. Une cellule communique toujours sur ses défacements mais se montre en général beaucoup plus discrète sur ses autres agressions. Elle crée tout de même des projections algorithmiques ouvertes [3] lorsqu'il s'agit par exemple de mettre en vente une base de données captée durant l'agression ou de construire un botnet à l'échelle de la communauté de cellules.

Le défacement constitue bien l'activité « visible » de la cellule qui lui permet d'en tirer avantage en termes de notoriété. Plus celle-ci est importante, plus la cellule de-

vient visible et crédible sur le marché international du hacking. La réputation constitue un levier marketing de tout premier ordre lorsqu'il s'agit de mettre en valeur un savoir-faire, une compétence de programmation spécifique ou lorsque la cellule cherche simplement à vendre des informations volées ou des données bancaires.

L'étude attentive des bases de données fournies par zone-h révèle les comportements, les spécificités, les engagements et les « cultures » de chaque cellule référencée sur le site.

Elle permet d'établir une typologie, notamment des structures affichant une forte activité offensive.

Certaines cellules revendiquent ainsi plus de 2000 agressions par an, alors que d'autres (la majorité) en produisent très peu. Une telle répartition laisse penser qu'une loi de Pareto de la forme (90-10) détermine la distribution des agressions : les bases des agressions attribuées montrent en effet que « peu font beaucoup » c'est-à-dire que 90 % des agressions sont réalisées par 10 % des cellules actives référencées sur la base.

À partir de ces données, une étude statistique fine des cibles et des méthodes utilisées en agression peut apporter un éclairage pertinent sur les mécanismes à l'origine des agressions. Dans certaines circonstances, l'étude croisée des différentes projections algorithmiques ouvertes d'une cellule permet d'établir des hypothèses raisonnables et probables sur l'éventualité d'une future agression et sur ses cibles potentielles. Pour autant, il ne s'agit pas de systématiser la prévision mais plutôt d'estimer les probabilités d'activité d'une cellule à un instant donné, en fonction de son historique et de ses interactions avec d'autres cellules.

3 L'étude d'une communauté de 320 cellules actives

3.1 Méthodologie de l'étude

Nous avons isolé un groupe constitué de 320 cybercellules qui ont enregistré et validé des agressions par défacement sur le site zone-h durant la période avril-mai 2014. Certaines de ces cellules sont référencées sur zone-h depuis plusieurs années ; elles possèdent toutes un espace de communication et de stockage qu'elles utilisent pour échanger et faire connaître leurs trophées numériques. Par son fonctionnement simple et efficace, le site zone-h qui existe depuis dix ans, a su attirer de nombreux hackers. Sa charte déontologique est compatible avec un état d'esprit « hacker » qui possède ses propres codes et qui reste exigeant sur le référencement et l'homologation des agressions. Si la réputation fait clairement référence à la notoriété du hacker, l'homologation renvoie à l'efficacité et donc à la lutte pour la valeur (compétences techniques, valeur de marché). Il y a ainsi une logique économique sous-jacente à la démarche de la communauté du hacking.

Une approche plus polémologique met en lumière les trois sources de la guerre énoncées par Thucydide : l'honneur, la peur et la conquête des ressources (ou appropriation). Ici, deux sources sont identifiées : l'honneur (la réputation) mais aussi les ressources (la publicité permet de trouver des clients donc d'augmenter ses gains).

La réputation recouvre deux objectifs : soit celui de l'activisme autour d'un objectif donné (politique ou éthique), soit celui de la compétence technique.

Remarquons ici que ces processus d'agglomération de capacités renvoient à deux mécanismes typiques des sciences sociales : pour les politistes, il s'agit de la constitution d'alliances [11] ; pour les économistes, il s'agit de constitution de cartels. Dans les deux cas, l'objectif consiste à accroître sa puissance par agglomération de compétences pour modifier le rapport de forces dans le premier cas, pour développer son pouvoir de marché dans le second.

Le site zone-h a enregistré presque 10 millions d'agressions par défacement en 10 ans avec une augmentation sensible depuis 2010. zone-h a connu une baisse d'enregistrement en 2013 que l'on peut attribuer à l'ouverture simultanée de plusieurs sites proposant les mêmes services (phrack.org, attrition.org, blackhat.info, flashback.se, turkeynews.net spécialisé dans le défacement turc ou encore hackzone.ru, spécialisé dans le défacement russe).

Après avoir sélectionné le groupe des 320 cellules, nous avons construit une table contenant le nom de la cellule, le nombre total des agressions par défacement qu'elle a validées sur zone-h depuis la date de son enregistrement, le nombre de défacements obtenus sur IP unique et le nombre de défacements réalisés sur des IP groupées, puis, quand cela a été possible, l'origine géographique de la cellule et ses motivations affichées. Le groupe témoin des 320 cellules est à l'origine, à lui seul, de 1 091 996 agressions par défacements réalisées depuis le début des enregistrements de chaque cellule sur zone-h.

3.2 Un principe de Pareto dépassé

Le groupe des 320 cellules a produit 1 091 996 agressions par défacements (donnée valide au 5 mai 2014). Les deux cellules les plus actives (GhoST61 et Hmei7) sont à l'origine respectivement de 306 398 et 271 475 agressions, soit 577 873 défacements c'est-à-dire un peu plus de la moitié du total des agressions. Les 25 cellules les plus actives du groupe produisent à elles seules plus d'un million d'agressions, les 295 suivantes assurant le reliquat. La répartition des agressions par cellule montre que 90 % des défacements sont attribuables à moins de 10 % des cellules (table 1).

Le partage habituel du principe de Pareto (80 % - 20 %) est nettement dépassé. On retrouve d'ailleurs ce résultat sur des échantillons d'agressions issus des bases de données zone-h, plus resserrés et plus anciens (datant de 2012).

La cellule GhoST61 revendique son origine turque, alors que Hmei7 est une cellule « singleton », composée semble-t-il d'un seul jeune hacker indonésien. Compte tenu de sa forte activité, Hmei7 a fait l'objet de plusieurs publications d'articles et d'interviews sur les sites et forums consacrés au hacking (news.softpedia) [9]. Cette cellule a acquis rapidement une forte notoriété sur zone-h.

L'étude des liens et du contenu ouvert des espaces de communication des cellules montre que certaines d'entre elles, très actives, créent de nouvelles cellules périphériques pour réaliser des campagnes d'agressions spécifiques sans que ces dernières soient comptabilisées dans le score de la cellule principale. Cela explique en partie la

répartition de Pareto. Les cellules à faible activité sont parfois spécialisées dans un type de cible bien défini. Elles ne cherchent pas à augmenter leurs scores mais privilégient le choix de la cible. D'autres enfin sont enregistrées depuis quelques mois seulement et n'ont pas eu le temps d'accumuler les trophées.

Table 1. Les 10 cellules les plus actives (au 05 mai 2014)

Nom de cellule	Nombre de défacements	Sur IP unique	Défacements groupés	Origine
GhoST61	306398	23192	283206	Cellule turque
Hmei7	271475	122186	149289	Défaceur indonésien
d3b~X	93260	39521	53739	Défaceur Indonésien
Th3Sehzade	44728	185	44543	Cellule turque
HighTech	43787	13130	30657	Cellule brésilienne
ghost-dz	35541	6230	29311	Hacker algérien
ZoRRoKIN	29233	13971	15262	Cellule turque
AnonGhost	13582	1846	11736	Cellule de 32 hackers – leader mauritanien, Maroc, Malaisie, Indonésie, Tunisie, USA, Irlande
kwgdeface	12558	2251	10307	Kosova Warrior Group - Albanie
MagelangCyber	11864	4056	7808	Cellule Indonésie
Total	862426	226568	635858	

Les 10 cellules les plus actives du groupe témoin des 320 cellules représentent à elles seules 79 % des agressions référencées.

3.3 Une communauté de hacking

À l'intérieur de notre ensemble initial de 320 cellules, nous avons identifié un important sous-groupe pratiquant le hacking d'influence [10]. Ce type de hacking repose sur la volonté de diffuser un message militant associé à une cause bien définie par l'intermédiaire d'une cyberagression. Il conjugue donc une action dans la couche logique et une exploitation dans la couche sémantique [7].

Son auteur cherche toujours à modifier la perception commune d'une situation ou d'un ensemble d'événements dans un sens qui lui est profitable.

Le lien de cohésion de cette communauté est de nature politico-religieuse. Les cellules qui la composent revendiquent clairement sur leurs espaces de communication un « cybercombat musulman » porté contre des cibles américaines, européennes, asiatiques et russes. Ce sous-groupe utilise zone-h pour référencer, valider, et commenter ses agressions numériques. Il développe un hacking d'influence qui fédère le groupe et lui permet d'agréger de nouvelles cellules compatibles avec l'idéologie dominante.

Les cellules principales possèdent en général un compte twitter actif, un compte Facebook et parfois un site web dédié à la communication d'influence du groupe.

Les communautés peuvent organiser divers mécanismes de hacking : hacking d'influence, hacking d'opportunité, hacking d'intérêt.

Les mécanismes du hacking d'influence [6] reposent sur un ciblage restreint défini par « l'objectif » de l'attaquant. Celui-ci peut s'inscrire dans le contexte fortement contraint d'un conflit armé projeté sur le cyberespace, d'un hacktivisme politique ou religieux, d'une campagne de collecte d'informations ou d'une opération d'influence militaire ou civile. Les cibles sont alors choisies en fonction des intérêts de l'attaquant et selon une stratégie construite à l'avance. Ce sont les objectifs sémantiques qui vont permettre de construire la prévision de l'agression [12] , [13]. En examinant les motifs de l'agresseur, il sera parfois possible d'anticiper ses intentions pour en déduire l'ensemble des cibles probables. La phase suivante consiste à détecter les fragilités de la cible permettant une future agression. Ces fragilités sont recherchées sur les trois strates structurelles du cyberespace : les couches matérielle, logique et sémantique. Lorsque l'attaquant recherche des fragilités sur la strate humaine ou sémantique, il utilise les techniques d'ingénierie sociale et travaille directement sur la crédulité, la confiance et les biais cognitifs de l'opérateur humain. Une bonne connaissance des mécanismes comportementaux, de la psychologie des cibles humaines ou d'anthropologie facilite grandement le succès de cette première phase de l'agression. Le ciblage s'effectue d'abord sur la couche sémantique du cyberespace grâce à l'utilisation des projections algorithmiques des cibles. La recherche des fragilités sur les strates logique et matérielle mobilise une expertise purement technique (réseaux, systèmes, cryptographie-codage, développement, rétro-analyse...). Parfois, la cible est parfaitement identifiée mais l'agression n'a pas lieu car l'attaquant ne découvre pas de vulnérabilité raisonnablement exploitable. Il faut dissocier le hacking d'influence du hacking opportuniste.

Ce dernier se construit sur l'unique critère de présence et d'identification d'une faille de sécurité exploitable. Dans ce cas, le contenu sémantique de la cible importe peu, seules la vulnérabilité du support et sa notoriété déterminent l'agression.

L'espace des cibles d'un hacking opportuniste est particulièrement large puisqu'il n'est limité que par le niveau de sécurité protégeant le système. Un facteur aléatoire et une forme de sérendipité (ou sagacité accidentelle) peuvent également influencer l'agresseur. Enfin, on doit évoquer le hacking d'intérêt qui repose essentiellement sur le vol d'informations à la suite d'une commande, au profit d'un client. Il s'agit alors d'un hacking discret qui n'est pas divulgué sur les forums. Les projections algorithmiques des agresseurs sont dans ce cas toujours restreintes et optimisées.

L'étude du groupe témoin des 320 cellules montre que celles qui affichent une forte activité conjuguent souvent hacking d'influence et hacking opportuniste. Celui-ci est pratiqué pour élever le niveau du compteur d'agressions de la cellule et améliorer sa réputation. Le vol et la revente de données bancaires peuvent également motiver un hacking opportuniste qui opère alors comme un soutien financier du hacking d'influence.

Nous exposons dans la section suivante deux exemples de hacking d'influence issus des activités du groupe témoin et montrons que certaines des agressions apparaissent comme « prévisibles » au regard de l'historique des cellules et de leurs revendications.

Dans ce cas, il ne s'agit pas d'une prévision rétrospective qui n'aurait aucune utilité opérationnelle mais bien d'une hypothèse formulée avant l'agression, sur un ensemble de cibles probables et sur une période de réalisation possible. En particulier, nous avons formulé l'hypothèse d'une future campagne de cyberattaques OPIsraël II ciblant des intérêts numériques israéliens trois semaines avant la réalisation effective de l'attaque. Nous avons proposé une date d'exécution qui s'est révélée exacte. Enfin, nous avons identifié le site du Ministère de l'agriculture israélien comme une future cible hautement probable, ce qui a été confirmé lors de l'attaque.

4 Des exemples de communautés « hacktives »

4.1 Exemple 1 – Dr.SHA6H

La cellule Dr.SHA6H est créditée de 9396 agressions par défacement validées par zone-h dont 2518 sur IP uniques et 6878 sur IP groupées. Elle conjugue le hacking d'influence en affichant son appartenance à la « Free Syrian Army » et le hacking opportuniste.

Fig. 1. - Entête du compte twitter Dr. SHA6H

En avril 2014, Dr.SHA6H a revendiqué l'agression du site officiel de l'Unicef en Nouvelle-Zélande « Join the movement for children ». La première page du site a été augmentée d'une vidéo d'un enfant syrien gravement blessé par balle soigné sur la table d'un hôpital de fortune.

Fig. 2. - Défacement du site de l'Unicef par Dr. SHA6H

Un communiqué dénonçant la passivité des nations occidentales face au conflit syrien complétait la vidéo. Dans ce premier exemple, le hacking du site de l'Unicef n'était pas particulièrement prévisible mais seulement cohérent au regard de l'idéologie ambiante de la cellule Dr.SHA6H qui dénonce pêle-mêle Israël, l'Iran, l'Irak, la Russie, la Chine, les États-Unis, et tous les pays apportant, selon l'interprétation de la cellule, leur soutien au régime syrien.

4.2 Exemple 2 – OP Israël I et II

Le 7 avril 2013 débutait l'opération OPIsraël-I ciblant un grand nombre de sites web israéliens.

Une communauté de 23 cellules présentes sur zone-h avait pris part à cette agression coordonnée sans pourtant créer de lourds dégâts du côté de l'État hébreux. La plupart des cellules impliquées à l'époque figurent dans notre groupe témoin, elles se sont organisées autour d'un groupe leader qui a supervisé et dirigé l'opération. La communauté OP Israël-I était composée de :

AnonGhost, Anonymous, SEA, Afghan Cyber Army, Izz ad-Din al-Quassam Cyberfighter, Mauritania Hacker Team, AnonSec, ZHC, X-Blackerz INC, Xploiter, Team Hacking Argentino, Pakistian Haxors Crew (PHC), Egyptian Shell Team, Gaza Hacker Team, Fallaga, Indonesian Cyber Army, Indonesia Fighter Cyber, Anonymous Syria, Anonymous Malaysia, Anonymous Tunisia, Anonymous Algeria, Anonymous Jordan, Anonymous Lebanon.

L'une des cellules AnonGhost avait revendiqué les agressions de 478 sites web israéliens et promis de poursuivre la lutte pour le droit du peuple palestinien.

En début d'année 2014, le site opisraelbirthday.com est crée par AnonGhost. L'étude de son contenu croisée avec celle des espaces zone-h de chaque cellule et des comptes twitter des cellules principales du groupe laisse peu de doute sur la réalisation d'une OP Israël-II.

La date anniversaire du 7 avril 2014 apparaît comme cohérente pour déclencher la seconde opération. Remarquons que cette date ne peut être « anticipée » que par l'analyse et donc par des moyens (actuellement) non automatisés. L'espace des cibles est constitué a priori de l'ensemble très vaste des sites web israéliens mais l'étude des projections ouvertes de quelques cellules montre qu'elles ont détecté des vulnérabilités sur certains sites dont celui du Ministère de l'agriculture israélien. Sans jamais évoquer directement l'agression, les cellules échangent des informations qui, croisées entre elles, permettent de construire l'hypothèse d'une agression probable, le 7 avril 2014 sur un ensemble de sites dont celui du Ministère de l'agriculture israélien. Quelques semaines plus tard, OPIsraël-II a bien lieu à la date prévue et le site en question fait bien partie des cibles.

Cet exemple montre qu'une hypothèse peut être construite à la suite d'une veille des cellules actives et qu'elle peut faire l'objet d'un signalement préventif en direction des cibles potentielles identifiées.

Fig. 3. - Hacking du site du Ministère de l'agriculture israélien - OPIsraël II

Le cas d'OPIsraël n'est pas isolé. La cellule Anonghost, composée d'une trentaine de membres, a collaboré activement avec d'autres cellules sur les opérations OpUSA et OpPETROL. Là encore, certaines cibles étaient hautement probables bien avant que les agressions n'aient lieu. Les échanges et les métadonnées produites par ces cellules fournissaient des informations exploitables lors de la construction d'un en-

semble de cibles potentielles. L'Armée Syrienne Électronique (AES) [7], à l'origine de plusieurs centaines d'agressions par défacements, de captations de données et d'introductions de malwares contre des sites occidentaux, possède des comptes actifs sur les réseaux sociaux. L'analyse des projections algorithmiques de cette cellule permet d'identifier un ensemble de cibles probables notamment lorsqu'un média international publie un article défavorable ou que des noms de membres présumés de la AES sont divulgués.

5 Veilles et prévisions

Tâchons ici d'établir une méthode de prévision des agressions futures.

5.1 Une veille des cellules les plus actives

Le recensement des cellules actives constitue la première étape de la veille. Il s'effectue après avoir défini une mesure instantanée d'activité de la cellule, compatible avec l'ensemble des supports (forums, sites, comptes ouverts sur les réseaux sociaux) sur lesquels elle archive ses cyberagressions. Une fois identifiées à l'aide de cette mesure d'activité, les cellules principales font l'objet d'une surveillance systématique. Leurs projections algorithmiques sont agrégées et classées. Elles sont utilisées ensuite pour construire le graphe des relations entre cellules et pour déterminer leur « voisinage topologique relationnel ». Une cartographie précise, en temps réel, des collaborations, des rapprochements effectués lors d'une agression et des dépendances entre cellules, apporte de l'information exploitable durant la phase de prévision.

Il s'agit ensuite de déterminer leurs cibles privilégiées. Pour cela, on détermine leurs objectifs politiques identifiés surtout lorsqu'on a établi leur utilisation régulière de hacking d'influence. La prévision du hacking d'opportunité semble, par construction, non prévisible. Quant au hacking d'intérêt, il semble constituer une activité qui se déroule en dehors des places réputationnelles qui, telles zone-h, permettent d'établir la cote des différents groupes de hackers.

Il s'agit ensuite de déterminer les occurrences probables d'agression des cibles préalablement identifiées. Les concordances temporelles, anniversaires de campagnes d'agressions, dates de fusions ou de séparations de cellules doivent elles aussi être utilisées dans la recherche de la période la plus favorable à une future agression. De même, compte tenu des objectifs politiques des cellules, les grandes dates liées à la cible probable et associées au profil sémantique de la cellule d'agression peuvent être insérées dans les mécanismes de prévision. Ainsi, pour une cellule iranienne, l'anniversaire de la révolution de 1979 constitue un repère à ne pas négliger dans la construction d'une prévision. Il faut également détecter les contextes de coopération et de formation de communautés de cellules, motivées par un objectif ou un défi commun. Ces situations de concordance d'intérêts, durant lesquelles des cellules se

rapprochent ou se fédèrent autour d'une cellule leader, constituent souvent un signal fort de préparation d'une campagne d'agressions de grande envergure.

Le conflit ouvert (né de rivalités, défis ou concurrences) opposant deux cellules ou deux groupes de cellules donne lieu à des projections algorithmiques spécifiques révélatrices du conflit et engendre de l'information exploitable dans une construction de cibles potentielles. C'est typiquement le cas lorsqu'une confrontation armée « s'exporte » dans le cyberespace et engendre des campagnes de cyberagressions [8].

La veille exhaustive des cellules les plus actives nécessite une architecture informationnelle adaptée. Celle-ci doit être en mesure de réaliser une collecte en temps réel des projections algorithmiques des cellules puis de cartographier leurs connexions.

5.2 Vers la prévision de certaines cyberagressions

Opter pour une démarche de prévision, vise avant tout à développer une architecture permanente capable de fournir des cibles potentielles selon une certaine probabilité puis de donner l'alerte lorsqu'un seuil de confiance est atteint. Les phases de collecte et de structuration des données doivent permettre la construction de l'ensemble dynamique des cibles potentielles d'une cellule, à l'instant t. On associe à chaque cible figurant dans cet ensemble une probabilité de réalisation d'agression par la cellule durant une plage temporelle. La base des cibles et des probabilités est ensuite mise à jour en temps réel selon le rythme de collecte d'information et l'historique offensif de la cellule. La difficulté principale réside dans la méthode de calcul des probabilités d'agression. L'analyse bayésienne, l'apprentissage automatisé, les mesures de similarités de contenus offrent des approches efficaces mais l'analyse et la supervision humaine restent indispensables dans une architecture de calcul qui se veut optimale.

Nous proposons une architecture composite qui s'appuie en premier lieu sur l'analyse et l'approche systémique de la cellule de hacking. Cette analyse doit décrire les interactions de la cellule observée avec ses cibles passées et les relations qu'elle a noué avec d'autres cellules. C'est sur cette base de connaissances que s'appuiera ensuite le processus de déduction de cibles potentielles. La prise en compte des attributs, des qualités et des compétences de la cellule s'effectue à partir d'une analyse précise de son historique d'agression « ouvert » (fig.4). La phase de collecte de ces données est automatisable. Il faut ensuite établir les graphes des relations entre cellules :

- le graphe des coopérations et soutiens techniques,
- le graphe des alliances ou allégeances entre cellules,
- le graphe de compétition et défis entre cellules.

A partir de l'analyse des échanges ouverts entre cellules publiés sur les réseaux sociaux et les forums spécialisés, on doit identifier et extraire les contextes qui permettent de désigner un ennemi et les situations dans lesquelles la cellule cherche à venger une victime, à punir une cible à la suite d'une action ou d'une prise de position jugée inacceptable. La cellule souhaitera « rendre justice » à la suite d'une agression et la dénoncer publiquement par un défacement. Chaque contexte, potentiellement généra-

teur de cyberattaques, recevra un grade ou une cotation selon une échelle à définir. Les conjonctions de situations favorables à une future agression, permettant de désigner de façon indépendante une même cible, recueilleront un bonus de cotation. L'ensemble des cibles potentielles sera établi au regard des données précédentes. On évaluera en particulier la probabilité que la cellule agresse sa cible durant une période en fonction de la cotation des contextes. Le cas particulier de l'AES s'inscrit pleinement dans cette approche : la cellule AES produit suffisamment d'informations pour alimenter le mécanisme des contextes et permettre leur cotation. L'ensemble des cibles potentielles reste très large mais certaines d'entre elles bénéficient de contextes à forte cotation. Dans ce cas, la prévision peut donner de bons résultats. Inversement, lorsque la cellule produit très peu de données, la cotation devient plus délicate et la prévision plus hasardeuse.

Si l'étape d'évaluation des probabilités d'agression d'une cible par la cellule en fonction des cotations semble partiellement automatisable, celle de la cotation des contextes résulte pour l'instant de l'analyse humaine...

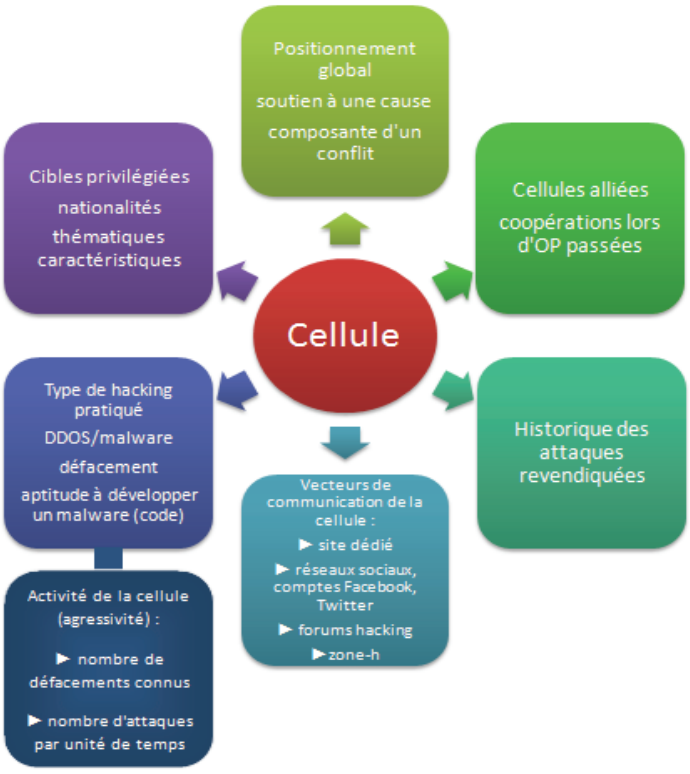


Fig. 4. - Attributs d'une cellule de hacking

Références et liens

[1] Fire Eye - http://www.fireeye.com/blog/?p=4776

[2] FILIOL E., *Les virus informatiques, théorie, pratique et applications*, deuxième édition, Collection IRIS, 2009, Springer – Verlag.

[3] BERTHIER T.- « Projections algorithmiques et cyberespace » R2IE – revue internationale d'intelligence économique – Vol 5-2 2013 pp. 179-195.

[4] http://opisraelbirthday.com/ et https://twitter.com/An0nGhost

[5] Zone-h : http://www.zone-h.org/archive

[6] VENTRE D., (Dir), *Cyberguerre et guerre de l'information*, Règles, stratégies, enjeux, Lavoisier, Paris, 2010.

[7] BERTHIER T. et KEMPF O. - « L'armée syrienne électronique : entre cyberagression et guerre de l'information » RDN – Revue de la défense nationale – dossier « Guerre de l'information » - mai 2014.

[8] KEMPF O., *Introduction à la cyberstratégie*, Economica, 2012

[9] Hmei7 - http://news.softpedia.com/news/Hackers-Around-the-World-Hmei7-Indonesian-Defacer-361176.shtml

et http://www.ehackingnews.com/2013/01/Indonesian-top-defacer-hmei7.html

[10] http://cyberland.centerblog.net/80-Le-hacking-influence

[11] KEMPF O., *Alliances et mésalliances dans le cyberespace*, Economica, Paris, à paraître novembre 2014.

[12] HUYGHE FB, KEMPF O. et MAZZUCCHI N., *Composantes politico-militaire, économique et sociétale d'une cyberstratégie française : agir dans la dimension sémantique du cyberespace*, Etude (IRIS) rendue au CSFRS, juin 2014.

[13] KEMPF 0., « Dimension informationnelle de la cyberstratégie » in HARBULOT Ch., *Manuel de l'intelligence économique*, PUF, 2ème édition à paraître décembre 2014.

Bibliographie partielle des auteurs

BERTHIER T. - « Projections algorithmiques et cyberespace » R2IE – revue internationale d'intelligence économique – Vol 5-2 2013 pp. 179-195.

BERTHIER T., *Cyberchronique – Décomposition systémique d'une cyberattaque, dissymétries et antifragilité*, Publications de la chaire de cyberstratégie CASTEX, janvier 2014

BERTHIER T., *Sur la valeur d'une donnée,* Publications Chaire de cyberdéfense Saint- *Cyr-Sogeti-Thales – mai 2014.*

BERTHIER T., *Concurrences et duels algorithmiques*, Revue de Défense Nationale, juin 2013

BERTHIER T., *Créons l'observatoire des évolutions algorithmiques*, Défense et Sécurité *Internationale, mai 2013.*

BERTHIER T. et KEMPF O. - « *L'armée syrienne électronique : entre cyberagression et guerre de l'information* » RDN – revue de la défense nationale – « Guerre de l'information » Vol. mai 2014.

KEMPF O., *Introduction à la cyberstratégi*e, Economica, 2012

KEMPF O., « Entreprise et cyberstratégie », *Nouvelle revue de géopolitique*, février 2013

KEMPF O., « Cyberstratégie à la française », *RIS* n° 87, septembre 2012

KEMPF O., « Cadre de recherche de la cyberstratégie », *Revue Défense Nationale*, juin 2012

KEMPF O. et GUARRIGUE A., « L'OTAN et le cyber », *Sécurité Globale, n° 19,* mai 2012

DOSSE S., KEMPF O., (dir), *Stratégies dans le cyberespace*, L'esprit du livre, septembre 2011.

KEMPF O. (dir), *Penser les réseaux*, L'Harmattan (2014)

KEMPF O., « Cyberterrorisme, un discours plus qu'une réalité », *Hérodote*,, n° 152-153, printemps 2014

KEMPF O. « Cyber et surprise stratégique », *Stratégiques* n° 106, printemps 2014.

KEMPF O. « Stratégies des réseaux : le cas des réseaux électriques intelligents », *Revue Géoéconomie*, n° 69, mars 2014.

KEMPF O. « Conséquences stratégiques de l'affaire Snowden », *L'Observatoire géostratégique de l'information, IRIS* mars 2014.

KEMPF O., « Cyberstratégie chinoise : du contrôle à l'expansion » (avec Vivien Fortat), *AGIR, revue de la société de stratégie*, Octobre 2013.

KEMPF O., « La cyberstratégie de l'Union Européenne », *Sécurité Globale* n° 24 (été 2013), p 25-40.

DOSSE S., KEMPF O., MALIS C. (dir), *Le cyberespace, Nouveau domaine de la pensée stratégique*, Economica, 2013.

KEMPF O., *Alliances et mésalliances dans le cyberespace*, Economica, Paris, à paraître novembre 2014.

HUYGHE FB, KEMPF O. et MAZZUCCHI N., *Composantes politico-militaire, économique et sociétale d'une cyberstratégie française : agir dans la dimension sémantique du cyberespace*, Etude (IRIS) rendue au CSFRS, juin 2014.

KEMPF 0., « Dimension informationnelle de la cyberstratégie » in HARBULOT Ch., *Manuel de l'intelligence économique*, PUF, 2ème édition à paraître décembre 2014.

Thierry Berthier est Maître de Conférences en mathématiques (université de Limoges), il mène des recherches sur les stratégies et projections algorithmiques, les situations de concurrences et de duels algorithmiques et leurs impacts sur le cyberespace.

Olivier Kempf est docteur en science politique, chercheur associé à l'IRIS, auteur de « Introduction à la cyberstratégie » (Economica, 2012)) et de « Alliances et mésalliances dans le cyberespace ». Ses recherches portent actuellement sur l'exploitation de la couche sémantique du cyberespace et sur la question des cyberalliances.

Behavioral Detection of Internet Frauds against Online Service Platforms

Nizar Kheir, Sok-Yen Loui, Ndeye-Seynabou Diop,
François Olivier-Martin and Vincent Frey

Orange Labs, France
{name.surname}@orange.com

Abstract. Over the past decade, Internet has become a lucrative market place that enables millions of business transactions daily. The widespread use of this network offered a unique platform to access all kinds of services, including trade, shopping, booking, but also fiscal and medical services. Unfortunately, this rapid evolution has also paved the way to an active underground economy where cyber-attacks and fraud are one of its major aspects. We address in this paper the problem of proactive detection of fraud attempts against online service platforms. We introduce a behavioral system that monitors user interactions with the target service in order to detect unusual access paths and associate them with ongoing fraud attempts.
Our system operates in two phases, including learning and detection. In the learning phase, it monitors user activity in order to build a behavioral model that captures all user interactions with the target system. Our model is represented with a weighted directed graph where nodes characterize elementary user actions and transitions characterize common user access paths. During detection, our system monitors service logs and uses the behavioral model in order to iteratively compute a suspicion score for each user. It detects and blocks a fraud attempt when the suspicion score for a user exceeds a pre-defined detection threshold. Our system is built on top of current security solutions. It implements a behavioral detection model that we demonstrate its efficiency against multiple real world fraud attempts.

Key words: Behavior graphs, log forensics, fraud detection
Category: *Specialized contribution*

1 Introduction

The recent years have witnessed a fast growing record of online frauds worldwide. In a recent interview to the Guardian, they are believed to have caused more than $100bn of annual losses to the global economy [1]. Despite increasing efforts by online services to strengthen their security, they still experience a massive amount of cyber-attacks, where up to 91% of organizations are being attacked at least once in 2013, according to Kaspersky [2]. In fact, online services are growing increasingly complex and inter-dependent. They are built upon and integrate heterogeneous technologies with different security and interoperability guarantees [3].

As a consequence, the security and accountability in these systems are confronted to multiple and oftentimes conflicting requirements. On the one hand, the static security-by-design approach cannot guarantee an acceptable level of security when used as a single line of defense [4]. On the second, security administrators are reluctant to enable proactive security mechanisms because of their prohibitive amount of false positives, which threatens to degrade the quality of service perceived by end users [5–7].

In the realm of big data technologies and the ability to process large amounts of security events, data analytics have acquired a significant importance in the recent years [8]. In fact, *data-driven* security was first implemented against banking frauds [9], but has further found applications in almost all types of online services. It enables administrators to monitor security incidents and detect unusual trends and malicious behaviors that are attributable to cyber-threats or fraud attempts.

Nonetheless, existing data analytics solutions mostly operate as post-mortem analysis tools. They only support generic data formats and correlation features. Administrators are still required to manually customize their own event management rules and to bridge the gap between *correlation* and *causation* [10]. This gap is the root cause of multiple misleading patterns that generate a significant amount of false positives. Furthermore, current data analytics solutions also do not integrate automated forensic and event correlation support that make possible to detect and *proactively* react against attacks in their early stages. Such automated mechanisms are still confronted to multiple challenges [11]. First of all, they do not embed automated management procedures that capture user interactions with the target system across the different components of the architecture. Besides, they do not support behavioral models that learn from user interactions with the system in order to build customizable and proactive detection models.

This paper leverages the use of data analytics procedures to build a behavioral graph that captures all user interactions with a target system. It further develops a comprehensive approach that uses this graph in order to characterize and detect fraud attempts against online services. The main underlying assumption of our system is that online services integrate predefined business models that lead to specific user interaction sequences with these services. Our system captures those sequences through a weighted directed graph where nodes characterize observed user interactions and edges characterize the transitions that follow each user interaction.

During detection, our system monitors user interactions with the online service. It constantly updates a suspicion score for each connected user. It identifies as suspicious fraudsters all users whose interactions with the service follow unusual access sequences in the graph, and so their suspicion score would exceed a pre-defined detection threshold. In order to achieve our goal, we will structure this paper as follows. Section 2 provides an overview of our approach and its applications through a real world deployment scenario. Section 3 describes the architecture and workflow of our system. Section 4 presents a practical use case. Section 5 discusses possible extensions to our system, and finally section 6 concludes.

2 System overview

2.1 Illustrative example

Online services are nowadays no longer built upon a single component or a web service. They integrate multiple connecting services that provide users with a unique customer experience. In this paper, we illustrate the problem through a real world example that we have been able to analyze, and that we refer to as S_{anon} (for anonymous service). It includes a provider of e-commerce solutions for online retailers of downloadable video games. Unfortunately, we cannot provide the exact identity of the online service because of a non-disclosure agreement. However, we believe the anonymized example that we describe in this paper illustrates the intricacies of our proposed detection system, and indeed can be generalized to every type of online services that are made accessible on the internet. In the example of service S_{anon}, only authenticated users who have valid accounts can connect and purchase video games to the service. User accounts can be either managed directly by the service, or by third party publishers who have an affiliate program with S_{anon}. Therefore, two user access sequences are supported by S_{anon}, according to whether user accounts are managed *directly by the service*, or by an *affiliate publisher*.

On the first hand, direct users of the service follow a straightforward access path where they are first redirected to a separate authentication service. The web service frontend authenticates users through a valid login/password combination. Successful authentication redirects users to an *authentication confirmed* status. They are thus granted access to an online shopping center where they can display and select their purchased video games. After a user has selected items to purchase, he can select and confirm his purchase options, and so the user confirmation would

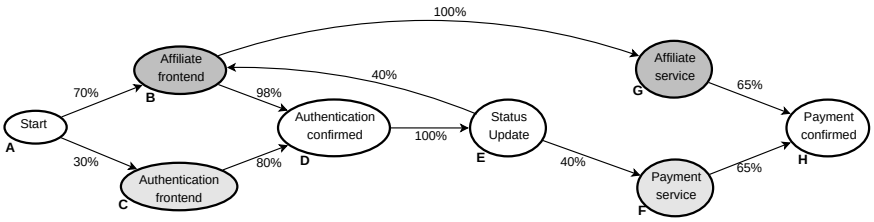


Fig. 1. Example of the Video Games online service

redirect users to a payment service where their purchased items are being validated.

On the other hand, users connecting to the service through an affiliate program are provided with valid authentication cookies by the affiliate program itself. They are redirected to a different web service frontend where their cookies are validated and so they are switched to an *authentication confirmed* status. Users of affiliate programs are then provided with the same access to the online shopping center, where they have the same privileges as for direct users. After they have confirmed their purchased items, they are redirected back to the authentication service that would synchronize the purchased items with the affiliate program before payment. Users are then redirected to the last step where their purchased items are being validated.

2.2 Overview of our approach

Our system tracks and represents the different interaction sequences that can be observed during a user connection to a service. It processes application and service logs in order to extract metrics and data, and to capture access paths that are available across the different components of the online service. In fact, the tracking of access paths in the service logs characterizes the nominal behavior of a service while it interacts with its remote users. Figure 1 illustrates the example of such a nominal behavior that is generated by our system during the learning phase, and as it processes service logs for S_{anon}.

As shown in the example of figure 1 and its associated log sequences in figure 2, our system represents the behavior of a service using a directed weighted graph. Graph nodes represent specific user activities or elementary service components, and edges represent sequences of user interactions. Unusual attacks or fraud attempts are associated with behaviors or access paths observed in the service logs, and that do not match

with only few or any of the nominal user behaviors represented in the graph. We implement our behavioral model as a directed weighted graph that captures all observed interactions between the users and the online service.

Our system operates in two phases, learning and detection. During the learning phase, it iteratively builds a directed weighted graph that captures all possible user interactions with the service. It computes the likelihood of a specific access path based on its occurrence frequency as observed in the learning phase. Note that our system uses unsupervised learning to build its behavioral detection model. Hence, it does not require a labeled set of malicious and benign user transactions. Indeed, it observes all user interactions with the service and iteratively computes an occurrence probability for each elementary user transaction, based on the number of times it has been observed during the learning phase. Our system identifies benign access paths in the behavioral graph when they have an overall occurrence probability that is higher than a predefined suspicion threshold. All paths that have an occurrence probability which is lower than this threshold, or sequences of interactions that occur during detection while they do not exist in the behavioral graph, are identified by our system as being malicious.

Our system automatically switches from learning to detection when the behavioral graph no longer evolves beyond a convergence threshold set by the administrator. An important property of our system is its ability to build reliable behavioral models even when the learning phase includes few fraud attempts. In fact, our system associates fraud attempts with inexistent events, or events that occur only few times during the learning phase, and so they would be assigned with a very low occurrence probability in the behavioral graph. Hence, as long as accidental fraud attempts are less likely to occur during the training phase than other benign user interactions, they will be assigned with lower occurrence likelihood and so they would lead to a higher suspicion score during detection.

During the detection phase, our system monitors *on the fly* the user interactions with the target service and matches them against available access paths in the graph. It iteratively updates the suspicion score for each user during his interaction with the service. It detects a fraud attempt when the suspicion score for a given user goes below a predefined detection threshold that is set as an input to our system.

```
// Direct user access path
[23:41:39]|<user-id>|<User-IP>|http://<service-name>/login.html?next=confirm
[23:41:42]|<user-id>|<User-IP>|http://<service-name>/login-confirm.html
[23:41:46]|<user-id>|<User-IP>|http://<service-name>/refresh?updatestatus
[23:42:12]|<user-id>|<User-IP>|http://<service-name>/payment.html
[23:42:12]|<user-id>|<User-IP>|http://<service-name>/payment-confirm.html

// Access path for affiliate users
[21:35:27]|<user-id>|<User-IP>|http://<sso-service>/authenticat.html?next=confirm
[21:35:29]|<user-id>|<User-IP>|http://<service-name>/login-confirm.html
[21:35:34]|<user-id>|<User-IP>|http://<service-name>/refresh?updatestatus
[21:37:42]|<user-id>|<User-IP>|http://<sso-service>/authenticat.html?next=payment
[21:37:49]|<user-id>|<User-IP>|http://<service-name>/payment.html
[21:37:58]|<user-id>|<User-IP>|http://<service-name>/payment-confirm.html
```

Fig. 2. Log sequences for the two available access paths

3 System description

This section introduces the behavioral graph that we build during the learning phase, and that captures the normal behavior of a service. It provides the formal concepts and the way we build this graph through monitoring of benign service activity. It also introduces the detection model, including the way we use the behavioral graph in order to compute a suspicion score for each user during runtime.

3.1 Building the behavioral graph

Graph definition: Our system represents user interactions with a service using a directed graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N}$ is the set of nodes, and $\mathcal{E}$ is the set of directed edges. During the learning phase, our module iteratively builds and updates the behavioral graph $\mathcal{G}$ as long as new users connect and interact with the service. The set $\mathcal{N}$ of graph nodes includes all couples (s_i, a_j), where $\mathcal{S} = (s_i)_{i\in[1..p]}$ is the set of all elementary services implemented by the system and $\mathcal{A} = (a_j)_{j\in[1..q]}$ is the set of all possible actions that may be implemented by these services. An edge $e_{1,2} = [(s_{i_1}, a_{j_1}), (s_{i_2}, a_{j_2})] \in \mathcal{E}$ is assigned an occurrence probability $p_{1,2}$. It specifies that users are redirected with probability $p_{1,2}$ from service s_{i_1} where they have performed action a_{j_1}, to service s_{i_2} where they can perform action a_{j_2}.

Graph Learning: In order to build the behavioral graph, our system monitors the user activity during a learning period of a dynamic length $\mathcal{L}$. It monitors user interactions with the service and iteratively updates the behavioral graph until this graph does no longer evolve beyond a given convergence threshold.

For each distinct user $(u_\alpha)_{\alpha \in [1..n]}$ who connects to the service during the learning phase, our system monitors all of its associated log sequences l^α. A user u_α is redirected from a graph node $n_1 = (s_{i_1}, a_{j_1})$ towards node $n_2 = (s_{i_2}, a_{j_2})$ in case where the log sequence l^α for user u_α includes two subsequent log instances $l^\alpha_{s_{i_1}}$ and $l^\alpha_{s_{i_2}}$ pertaining to actions a_{j_1} and a_{j_2} respectively. We compute the probability associated with the edge $e_{1,2}$ as $p_{1,2} = \frac{c_{1,2}}{\sum_i c_{i,1}}$. We introduce $c_{i,j}$ as a counter that is associated with edge $e_{i,j}$. During the learning phase, all counters are set to zero. A counter $c_{i,j}$ is incremented by one each time a new user is being redirected, during the learning period, from node n_i towards node n_j. In other terms, $p_{1,2}$ is evaluated as the number of times users are redirected from the node (s_{i_1}, a_{j_1}) towards the node (s_{i_2}, a_{j_2}), with respect to the total number of times users were observed at node (s_{i_1}, a_{j_1}). We consider that an edge $e_{i,j}$ is more likely to be used for benign service activity when it has a higher occurrence probability $p_{1,2}$ as observed by our system during the learning phase. Users who follow a low probability edge during detection would have a high suspicion score, and would be more likely to be detected as fraudsters by our system. We further describe in section 3.2 the way our system computes the suspicion score for each user and detects fraudsters during runtime. We shall also note that the number of times users are redirected *towards* a given node n_j does not necessarily correspond to the number of times users are redirected *from* this same node n_j towards other nodes in the graph. In fact, users may terminate their connection and close their sessions with the online service at any time, and therefore we may not have a single exit node in the behavioral graph. This condition is formally represented in our behavioral model as: $\sum_i c_{i,j} \neq \sum_k c_{j,k}$.

Graph convergence: Our system terminates the learning phase and switches to the detection phase when the behavioral graph does no longer evolve beyond a convergence threshold τ_{conv}. During the learning phase, our system monitors the graph evolution over time t. It dynamically computes a distance $d(\mathcal{G}_t, \mathcal{G}_{t-\delta})$ that captures the distance between two different instances of the behavioral graph $\mathcal{G}$. The distance d takes into account the fact of new benign access sequences being added to the graph. A recent graph instance $\mathcal{G}_t$ is similar (i.e. low values of d) to a previous one $\mathcal{G}_{t-\delta}$ only when the new connecting users follow the same benign paths as in the previous graph instance.

We define the distance function d as: $d = \sum P_b^t - \sum P_b^{t-\delta}$. It evaluates the occurrence probability of all benign access paths in the new graph instance $\mathcal{G}_t$, and that were considered as malicious in the previous graph

instance $\mathcal{G}_{t-\delta}$. The distance function d does not take into account the occurrence probability of other benign access paths that are common to both graph instances $\mathcal{G}_t$ and $\mathcal{G}_{t-\delta}$ because we consider these to be already identified as belonging to the benign service behavior. Therefore, we consider that the graph model $\mathcal{G}$ have indeed converged to a stable state when no more access paths are being added to the benign service behavior. Our system iteratively computes the distance function d over a sliding window δ that depends on the rate of new users connecting to the service. It switches to the detection phase when d goes below a convergence threshold that is defined by the administrator, and which is used as input to our system.

3.2 Detection model

During detection, our system operates online. It detects unusual access sequences when it has enough mismatch confidence with known benign access paths. Our system observes all log sequences associated with a single user. For each new log instance, it evaluates the probability that its associated node belongs to a benign access path. In our behavioral graph $\mathcal{G}$, the transition probability from a node n_i towards node n_j depends only on the current state of the system. Hence, for each new log instance $l_{s_j}^{\alpha_t}$ (t being the current observation time), our system evaluates the probability p_t^j for the associated node n_j to be part of a benign user access path. It detects a fraud attempt or an unusual access path in case p_t^j is lower than a static detection threshold τ.

The probability p_t^j characterizes the suspicion score for a given user at a given time t during runtime. It is iteratively evaluated by our system, at each new log instance, as follows:

$$p_t^j = \sum_{(n_i,n_j)\in\mathcal{E}}^{i} p_{t-1}^i \times p_{i,j}.$$

The suspicion score p_t^j only depends on the previous system states n_i which are directly connected to n_j through an edge $e_{i,j}$. The transition probability $p_{i,j}$ associated with the edge $e_{i,j}$ is provided by the graph. The probability p_{t-1}^i is also provided by the graph $\mathcal{G}$ in case where n_i is a root node (concrete examples are given in section 4). Otherwise it is recursively computed in the previous iteration $t-1$ using the same equation as for p_t^j. Hence, for each new log instance $l_{s_j}^{\alpha_t}$, our system computes the new transition probability p_t^j. It triggers a detection alert as soon as p_t^j goes below the detection threshold τ.

4 Experimental use case

This section experimentally validates our system through the example of the online service described in section 2.1. We first describe the behavioral graph in figure 1 and we discuss the way this graph captures the nominal behavior of the service S_{anon}. We further present three different attack scenarios and their associated artifacts in the service logs, and then we show how these attacks are successfully detected by our system.

4.1 Use case description and graph learning

We demonstrate the use of our system through the example of the online service described in section 2.1. As shown in figure 1, access to the online service is possible through two distinct access sequences, where direct users of the service connect to the authentication frontend, and affiliate users are redirected to the service through a third-party affiliate frontend. In order to build the behavioral model for service S_{anon}, we were provided with `15MB` of fully anonymized service logs that were collected during a period of three days, and that we used as input to our training phase. The format of these logs is illustrated in the excerpt of figure 2. Each log instance includes a timestamp, *anonymized* user identity and remote IP address, and the landing URL for the associated service. Note that the fact of these logs being fully and irreversibly anonymized does not limit the usage of our approach. Indeed, our system does not need to process the exact user identities at any time during the learning phase.

As shown by the behavioral model in figure 1, almost 30% of users connect directly to the service, while the remaining 70% are redirected through third-party affiliate programs. During the learning phase, 80% of users who connect to the authentication frontend have been successfully authenticated by the service. They were redirected to the authentication confirmed status, while the remaining 20% provided wrong credentials. On the other hand, affiliate users are authenticated by the affiliate program before they are redirected to the online service. Therefore, almost 98% of these users have been successfully redirected to the authentication confirmed status. For the remaining parts of the graph, up to 40% of authenticated users have selected online items and so they were redirected to the payment service. Indeed 65% of users who accessed the payment frontend have confirmed their purchased items and so they were redirected to the payment confirmed status. Note that affiliate programs manage their own payment service that redirects users to the same payment confirmed status when they have successfully confirmed their purchased items.

```
// Brute Force Password Guessing Attack
[10:32:27]|<victim-id>|<IP1>|http://<service-name>/login.html?next=confirm
[10:32:36]|<victim-id>|<IP2>|http://<service-name>/login.html?next=confirm
[10:32:44]|<victim-id>|<IP3>|http://<service-name>/login.html?next=confirm
[10:32:47]|<victim-id>|<IP4>|http://<service-name>/login.html?next=confirm

// Cross-Site Request Forgery Attack
[10:51:26]|<user-id>|<User-IP>|http://<service-name>/login.html?next=confirm
[10:51:28]|<user-id>|<User-IP>|http://<service-name>/login-confirm.html
[10:51:31]|<user-id>|<User-IP>|http://<service-name>/payment-confirm.html

// Affiliate Cookie hijack
[11:24:31]|<attacker-id>|<User-IP>|http://<sso-service>/authenticat.html?next=confirm
[11:24:34]|<attacker-id>|<User-IP>|http://<service-name>/login-confirm.html
[11:24:39]|<attacker-id>|<User-IP>|http://<service-name>/refresh?updatestatus
[11:24:44]|<attacker-id>|<User-IP>|http://<sso-service>/authenticat.html?next=payment
[11:25:01]|<victim-id>|<User-IP>|http://<service-name>/payment.html
[11:25:08]|<victim-id>|<User-IP>|http://<service-name>/payment-confirm.html
```

Fig. 3. Log sequences associated with common attack sequences

Last of all, we set the detection threshold of our system to 0.005. In other terms, our system detects a fraud attempt when the real-time suspicion score for a user goes below the threshold of 0.005.

4.2 Attack scenarios and detection

This section discusses multiple attack instances that we observed against the online service S_{anon}. It also demonstrates the use of our system during detection, as it successfully characterizes and detects the ongoing attacks.

Brute force password guessing attack: The first example includes a brute force password guessing attack. As in the first listing of figure 3, this attack is characterized through a number of login attempts towards the same user account. As shown in the behavioral graph of figure 1, users are redirected to the authentication confirmed status with a probability of 0.8. At $t_0 = 0$, all graph nodes have a null occurrence probability, except for the entry node A which has an initial probability $p_A^0 = 1$.

The first log instance notifies a login attempt against the victim account, and so the user is redirected to node C. After the first login attempt, the user is redirected to node D with a probability 0.8, or remains at the same node C with a probability $P_C = 1 - 0.8 = 0.2$. In case of a brute force attack, the user will be constantly redirected to the same node C until he provides valid login credentials. The probability p_C^n for a given user to be redirected n times to the same node C is equal to $p_C^n = 0.2 \times p_C^{n-1} + 0.3 \times p_A^{n-1}$.

For values of n greater than 1, the probability $p_A^{n-1} = 0$ and so p_C^n is recursively computed as $p_C^n = 0.2^{n-1} \times p_C^1 = 0.2^{n-1} \times 0.3$. In other

terms, p_C^n would be less than the threshold 0.005 for values of n greater or equal to 3. Hence, our system detects a brute force attack against a given user account as soon as the attacker attempts three invalid credentials associated all with the same user account.

Bypassing the payment service: The second example includes a Cross-Site Request Forgery (CSRF) attack [12] where a victim browser is confused into misusing a valid user identity on behalf of a remote attacker. In this example, the victim user is already authenticated to the service and so he would be automatically redirected to the authentication confirmed status. Through a CSRF attack, the victim user is redirected to the payment confirmed status without any access to the payment service. The log sequence that corresponds to this category of attacks is illustrated in the second listing of figure 3.

During detection, our system observes first a successful login attempt at t_1. The user is then redirected by the service at t_2 to the authentication confirmed status. Following the malicious URL access, the victim user who has been successfully authenticated to the service is redirected to the payment confirmed status (node H) at t_3.

At t_3, the probability p_H^3 is computed as $p_H^3 = 0.65 \times P_F^2 + 0.65 \times P_G^2$. The user has been authenticated at t_2 through the service authentication frontend (node C). The occurrence probability of the alternative path through node B is zero at t_2, as well as for the subsequent node G. On the other hand, the node F has also a null occurrence probability ($P_F^2 = 0$) because the user has never been redirected to node E. Hence, the node H also has a null occurrence probability at t_3. The probability $p_H^3 = 0$ is less than the suspicion threshold $\tau = 0.005$, and so our system would successfully detect the CSRF attack as soon as the victim user is redirected to the payment confirmed status.

Bypassing the affiliate payment service: The third example in this paper includes a cookie hijacking attack. The attacker uses either a malware or a cross-site scripting attack in order to hijack a valid cookie and to use this cookie as a way to bypass the payment service for the affiliate program at node G. In this attack scenario, the items that are purchased by the attacker would be paid by the victim user who owns the hijacked cookie. The log sequence for this category of attacks is illustrated in the third listing of figure 3.

As shown in the listing of figure 3, the user identifier that is used for access to the online service is different than the one that is used for

payment confirmation by the affiliate program. The last two log instances that are associated with the payment service would be thus handled by our system separately than the previous log instances. The occurrence probability of node G, which is associated with the first log instance that includes the victim identifier, is zero because no previous log instances for the victim identifier were seen by our system. Hence, our system detects a fraud attempt as soon as the attacker uses the hijacked cookie for payment confirmation at the affiliate program.

5 Discussion

5.1 Learning phase and detection coverage

Our system monitors user transactions with a service and represents them using a directed weighted graph. The performance of our system strongly depends on the coverage of the training phase and the number of user transactions being observed. Benign transactions that do not occur during the training phase would be mistakenly identified by our system as being malicious and so they would trigger false positives. Although this could be considered as a weakness to our system, it is common to all learning-based detection techniques. Indeed, during the learning phase, our system iteratively computes a coefficient of convergence $\mathcal{C}_{conv}$. It evaluates the level up to which our behavioral graph characterizes the online service to be protected. Our system automatically switches from learning to detection when $\mathcal{C}_{conv}$ does no longer exceed a predefined threshold τ_{conv}. As long as our behavioral model does no longer evolve beyond τ_{conv}, we consider it to be reliably characterizing all possible benign user transactions with the target service.

5.2 Learning phase and malicious behaviors

During the learning phase, our system observes user interactions with the service and integrates them as part of benign user activity. Malicious behaviors that accidently occur during the learning phase are represented in our system as benign user interactions. This is also a common limitation to all learning-based detection techniques. The quality of the ground truth labels has a great impact on the accuracy of our detection model. However, we shall note that our system detects malicious behaviors based on a suspicion score that is updated throughout the user interactions with the service. When only few malicious behaviors occur during the learning phase, they would be associated with low occurrence probabilities in our

behavioral model. Hence, they could still be detected by our system, as long as they are less likely to occur than other benign user interactions.

6 Conclusion

This paper presented a behavioral detection system that monitors user interactions with an online service and detects fraud attempts through log analysis and forensics. Our system compares user interactions with a service against a behavioral graph that captures all benign user interactions. It includes two separate phases, learning and detection. During the learning phase, our system passively monitors user interactions with the service in order to identify and characterize benign user interactions. During detection, it operates automatically in order to compute a suspicion score for each user and to detect suspect or fraudulent users. We presented in this paper our system and its implementation through the example of a real world online service. As shown in this paper, our system is able to detect attacks in their early stages, before any concrete damage is inflicted to the system.

References

1. McDonald, H.: Online fraud costs global economy 'many times more than $100bn'. The Guardian (October 2013)
2. Kaspersky: Global corporate it security risks. Kaspersky survey report (2013)
3. Debar, H., Kheir, N., Cuppens-Boulahia, N., Cuppens, F.: Service dependencies in information systems security. In: 5th international conference on Mathematical methods, models and architectures for computer network security. (2010)
4. Sood, A., Enbody, R.: Targeted cyberattacks: A superset of advanced persistent threats. In: IEEE security & privacy. (2013)
5. Cuppens, F., Ortalo, R.: Lambda: A language to model a database for detection of attacks. In: Thirs International workshop on Recent Advances in Intrusion Detection. (197-216)
6. Kheir, N., Cuppens-Boulahia, N., Cuppens, F., Debar, H.: A service dependency model for cost-sensitive intrusion response. In: 15th European Symposium on Research in Computer Security (ESORICS). (2010)
7. Debar, H., Dacier, M., Wespi, A.: Towards a taxonomy of intrusion-detection systems. In: Computer Networks. (1999) 805–822
8. Cardenas, A.A., Manadhata, P.K., Rajan, S.P.: Big data analytics for security. EEE Computer and Reliability Societies (2013)
9. Ngaia, E., Hub, Y., Wonga, Y., Chenb, Y., Sunb, X.: The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems **50** (2011) 559–569
10. McAfee, A., Brynjolfsson, E.: Big data: The management revolution. Harvard Business Review (2012)

11. Kruegel, C., F. Valeur, F., Vigna, G.: Intrusion detection and correlation: Challenges and solutions. In: Advances in Information Security. (2005)
12. Barth, A., Jackson, C., Mitchell, J.C.: Robust defenses for cross-site request forgery. In: ACM Conference on Computer and Communications Security (CCS). (2008)

Les attaques avancées : Constats réels et Tendances

Thibaud Signat
thibaud.signat@fireeye.com

FireEye

Résumé Cette présentation propose une vue d'ensemble des attaques ciblant les réseaux informatiques identifiées par FireEye en 2013. Il ne fait aucun doute à nos yeux que les activités décrites dans ce rapport ont été conçues pour servir une ou plusieurs des finalités suivantes : vol d'éléments de propriété intellectuelle, écoute des communications sensibles des autorités publiques, affaiblissement de la sécurité générale des sites en rapport avec la sécurité nationale. Les attaques avancées cataloguées en tant que menaces persistantes avancées dissimulent des activités en grande partie soutenues par des États, que ce soit directement ou indirectement.

Ce rapport repose sur les données fournies par le cloud FireEye Dynamic Threat Intelligence (DTI), qui propose des mesures des attaques communiquées par des clients FireEye du monde entier. Il démontre, preuves à l'appui, que les infections par des logiciels malveillants touchant les entreprises se multiplient à un rythme alarmant. Il montre également que les auteurs d'attaques parviennent sans mal à contourner les défenses traditionnelles telles que les pare-feux et les antivirus.

Key-workds : APT, ZéroDay, Attaques Ciblées, Exploit, CnC, cible, constat, tendances, secteur d'activité

1 Introduction

En 2013, les plates-formes de prévention des menaces FireEye ont mis au jour des millions d'incidents malveillants. Les chercheurs de FireEye s'appuient sur ceux-ci pour rechercher des menaces persistantes avancées, que nous définissons comme l'utilisation de techniques, tactiques et procédures distinctes par des États ou des organisations criminelles professionnelles, que ce soit directement ou indirectement. Ces attaques servent différentes finalités, du cyberespionnage à court terme à la subversion à long terme des réseaux ciblés.

L'année passée, les auteurs de cyberattaques sont restés à pied d'œuvre sans relâche. nous avons analysé près de 40 000 attaques avancées uniques chez ses clients, soit plus de 100 par jour en moyenne. Dans plus de 4 000 cas, il s'agissait de menaces persistantes avancées (plus de 11 menaces persistantes avancées uniques par jour en moyenne). Nous avons par ailleurs mis au jour près de 18 000 infections uniques par des logiciels malveillants dues à l'activité de menaces persistantes avancées (près de 50 par jour en moyenne).

Ces attaques observées ont pris de nombreuses formes différentes et émanaient de quasiment tous les pays et territoires de la planète. L'analyse de celles-ci ont permis de mettre en lumière plusieurs faits intéressants :

- Suivi de plus de 159 familles de logiciels malveillants distinctes associées à des menaces persistantes avancées .
- Confirmation de l'utilisation de certains outils de piratage disponibles publiquement par des menaces persistantes avancées, telles que Dark Comet, LV, Gh0stRAT et Poison Ivy.
- Découverte d'infrastructures de commande et de contrôle dans 206 pays et territoires, en hausse par rapport aux données de FireEye pour 2012 (184 pays, soit 81% des pays membres des Nations unies).
- Identification des pays hébergeant le plus de serveurs de commande et de contrôle, à savoir les États-Unis, l'Allemagne, la Corée du Sud, la Chine, les Pays-Bas, le Royaume-Uni et la Russie.

Sur la base des données recueillies, nous avons établi la liste des dix pays les plus souvent ciblés par des menaces persistantes avancées en 2013 : États-Unis, Corée du Sud, Canada, Japon, Royaume-Uni, Allemagne, Suisse, Taïwan, Arabie saoudite, Israël

Les menaces persistantes avancées ciblent des données de grande valeur, choisies avec soin, au sein de chaque secteur :

- Les menaces persistantes avancées ont ciblé plus de 20 secteurs, de l'aéronautique à la vente en gros.
- L'enseignement, la finance et les hautes technologies ont été, dans l'ensemble, les secteurs les plus ciblés.
- Les États-Unis, la Corée du Sud et le Canada ont enregistré le plus grand nombre de secteurs distincts ciblés.

D'après nos données, les secteurs suivants ont été la cible du plus grand nombre de familles de logiciels malveillants uniques : Organismes du secteur public (national), Services et consulting, Technologies, Services financiers, Télécommunications, Enseignement, Aéronautique, défense, Organismes du secteur public (régional et local),Chimie, Énergie

Quant aux vecteurs d'attaques, le Web et la messagerie électronique ont été les vecteurs les plus prolifiques, comme en témoignent les statistiques suivantes :

- Au total, nous avons analysé cinq fois plus d'alertes émanant du Web que de la messagerie électronique.
- Au niveau des pays individuels, nous avons enregistré trois fois plus d'alertes liées au Web qu'à la messagerie électronique.

Diverses raisons peuvent expliquer cet écart entre les attaques propagées par le Web et la messagerie électronique, notamment la sensibilisation accrue au problème du harponnage (spear phishing), l'essor des médias sociaux et la connexion continue au Web de certains internautes.

Les attaques zero-day constituent une arme essentielle de tout arsenal de menaces persistantes avancées, comme le prouvent les statistiques suivantes issues de cette année d'activité :

- FireEye a mis au jour 11 attaques zero-day en 2013.
- Au premier semestre 2013, Java a été la principale cible des attaques zero-day.
- Au deuxième semestre 2013, nous avons observé une explosion des exploits zero-day ciblant Internet Explorer utilisés dans le cadre d'attaques de type « watering hole ».
- Les groupes criminels sont aujourd'hui passés maître dans l'art de développer des exploits Java.
- Des menaces persistantes avancées ont pris pour cible des sites Web des administrations publiques américaines dans le cadre d'attaques de type « watering hole ».
- Les auteurs d'attaques mettent régulièrement au point de nouvelles solutions créatives pour contourner les environnements sandbox destinés à identifier les logiciels malveillants.

2 Conclusion

Lors de cette présentation, l'ensemble des informations les plus pertinentes de ces rapports seront mises en valeur, ceci également en fonction de différentes zones géographiques ainsi qu'en fonction de différents secteurs d'activité.

L'objectif de cette présentation est, de mettre en lumière certains constats quant aux attaques avancées, quant à leur proportion, origine, nature, répartition, etc.

REGIONAL ADVANCED

THREAT REPORT

Europe, Middle East and Africa 1H2014

Thibaud Signat
thibaud.signat@fireeye.com

1. Executive Summary

This FireEye Advanced Threat Report for EMEA provides an overview of the advanced persistent threats (APT) targeting computer networks that were discovered by FireEye during the first half of 2014 in EMEA.

Motivated by numerous objectives, threat actors are evolving the level of sophistication to steal personal data and business strategies, gain a competitive advantage or degrade operational reliability.

This report summarises first half of 2014 data gleaned from the FireEye Dynamic Threat Intelligence (DTI) cloud. Based on this information and insight, FireEye can report the following:

- Malware attacks—especially advanced targeted attacks—have nearly doubled in the first half of 2014
- The UK and Germany were the most targeted countries
- Government, financial services, telecommunications and energy were the most targeted verticals.

Disclaimer: This report only covers computer network attacks that targeted FireEye (anonymised) customers, sharing their metrics with FireEye – it is by no means an authoritative source for all APT attacks in EMEA and elsewhere in the world. In this dataset, we take reasonable precautions to filter out "test" network traffic as well as traffic indicative of manual intelligence sharing among our customer base within various closed security communities. We realise that some popular targeted threat actors' tools, techniques and procedures (TTP's) can be reused and repurposed by both cyber-criminals and nation-state threat actors alike. To address this issue, we employ conservative filters and crosschecks to reduce the likelihood of misidentification.

2. Definitions

Advanced Persistent Threat (APT): a distinct set of cyber tools, techniques, and procedures (TTPs) that are employed directly or indirectly by a nation-state or a sophisticated, professional criminal organisation for cyber espionage or the long-term subversion of adversary networks. Key qualifying APT characteristics include regular human interaction (i.e., not a scripted, automated attack), and the ability to extract sensitive information, over time, at will.

Callback: an unauthorised communication between a compromised victim computer and its attacker's command-and-control (C2) infrastructure.

Remote Access Tool (RAT): software that allows a computer user (for the purposes of this report, an attacker) to control a remote system as though he or she had physical access to that system. RATs offer numerous attractive features such as screen capture, file exfiltration, etc. Typically, an attacker installs the RAT on a target system via some other means such as spear phishing or exploiting a zero-day vulnerability, and the RAT then attempts to keep its existence hidden from the legitimate owner of the system.

Targeted Attack: a unique TTP-to-target pairing. Please note that APTs usually employ multiple TTPs and manage multiple targeted attacks at the same time.

Threat Actor: the nation-state or criminal organisation believed to be behind an APT. This could be a military unit, an intelligence agency, a contractor organisation, or a non-state actor with indirect state sponsorship.

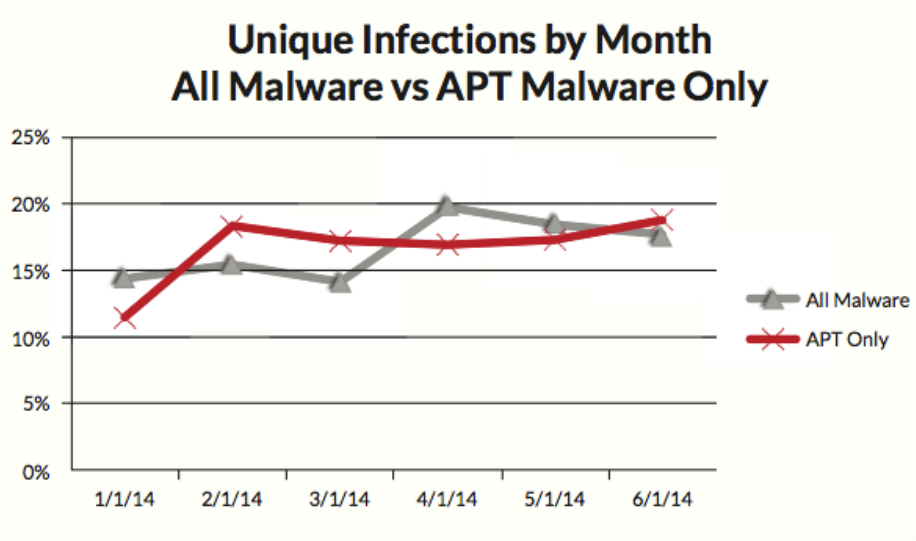
Tools, Techniques, and Procedures (TTPs): the characteristics specific to a threat actor in the cyber domain, usually referring to specific malware. As a caveat, it is important to remember that APTs normally employ multiple TTPs, and multiple APTs can also use the same TTPs. This dynamic frequently complicates cyber defence analysis.

Vertical: one of 20 distinct industry categories: Aerospace, Chemicals, Construction, E-Commerce, Education, Energy, Entertainment, Finance, Government, Healthcare, High-Tech, Insurance, Legal, Manufacturing, Other, Retail, Services, Telecom, Transportation, and Wholesalers.

3. Trend

Finding: Malware attacks—especially advanced targeted attacks—have nearly doubled in the first half of 2014. The number of unique infections has been growing steadily in EMEA. However if we focus on targeted attacks (that is, activity we've associated with targeted threat groups or malware known to be used by those groups), the number of unique infections almost doubled between January and June 2014.

Figure 1:
Unique Infections Trend



4. APT Detection

Country Analysis :

The UK and Germany are the most targeted countries. Let's first have a look at which countries that have been impacted by APT malware in EMEA.

The highest number of APT malware detected in EMEA in first half of 2014, by country, can be summarised:

1. United Kingdom (17%)

2. Germany (12%)

3. Saudi Arabia (10%)

4. Turkey (9%)

5. Switzerland (8%)

6. Italy (6%)

7. Qatar (5%)

8. France (4%)

9. Sweden (4%)

10. Spain (3%)

Figure 2:
APT Detection by Country

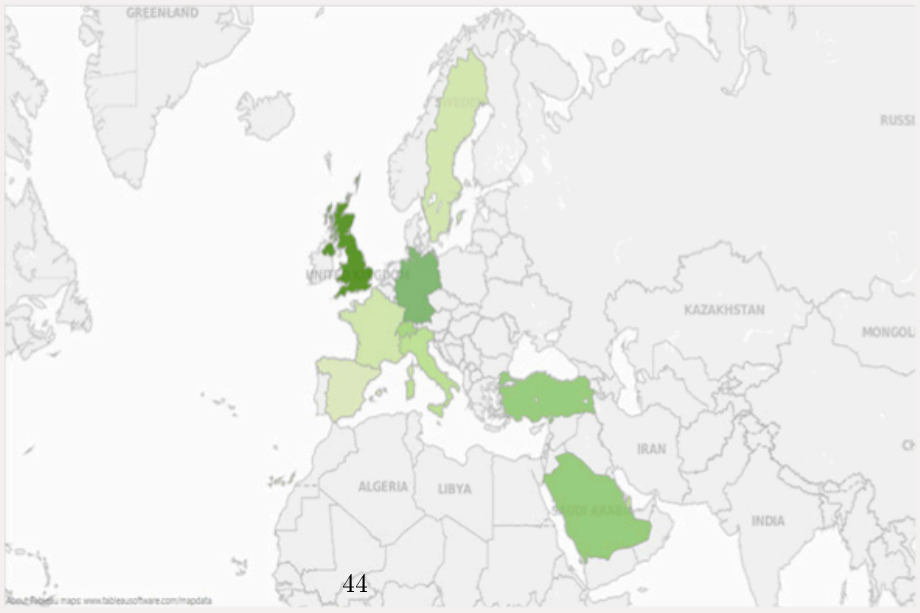
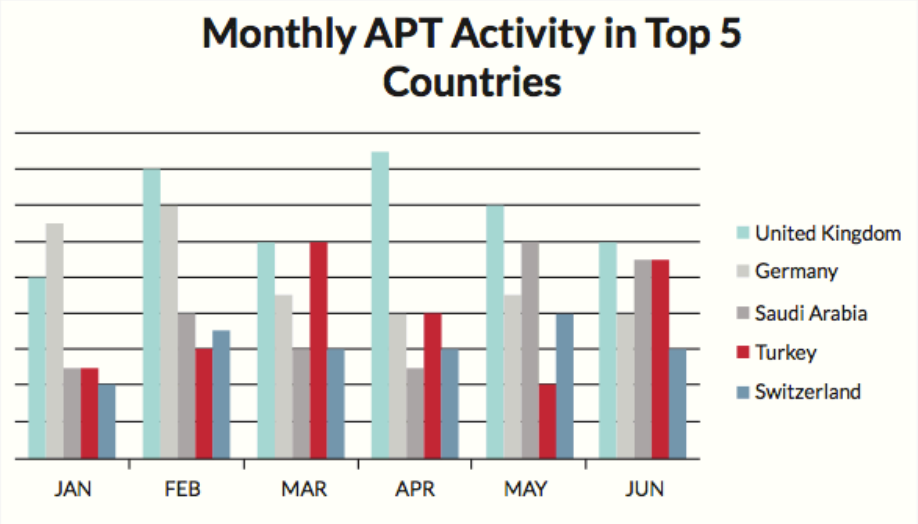
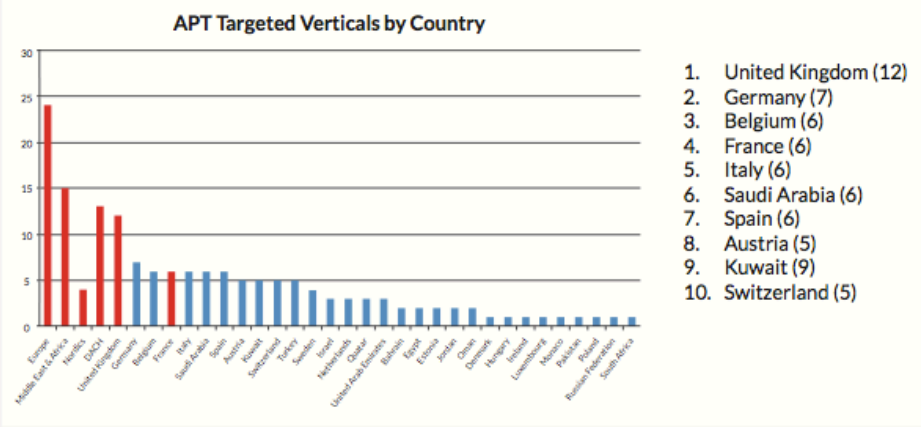


Figure 3: Monthly APT Activity in Top 5 EMEA countries



Comparing on an EMEA basis, we have identified that UK, Germany, Italy and France have the largest number of verticals targeted by APT.

Figure 4: Number of Verticals Hit By APT Malware by Country



We have also grouped specific EMEA sub regions:

- Europe (all European continent countries)
- Middle East & Africa (all Middle East and Africa countries)
- Nordics including Sweden, Denmark, Finland and the Baltic States
- DACH including Germany, Austria and Switzerland

Interestingly UK and DACH have similarities in the number of verticals targeted and also in terms of monthly activity highlighted in Figure 4. This suggests that a specific country is not being targeted but rather specific verticals.

Vertical Analysis

Government, financial services, telecommunications and energy were the most targeted verticals. The following figure presents APT activity, measured by number of alerts, by vertical.

Government, Financial Services and Telecom verticals represent more than 50% of total APT detections, and all are considered strategic industries.

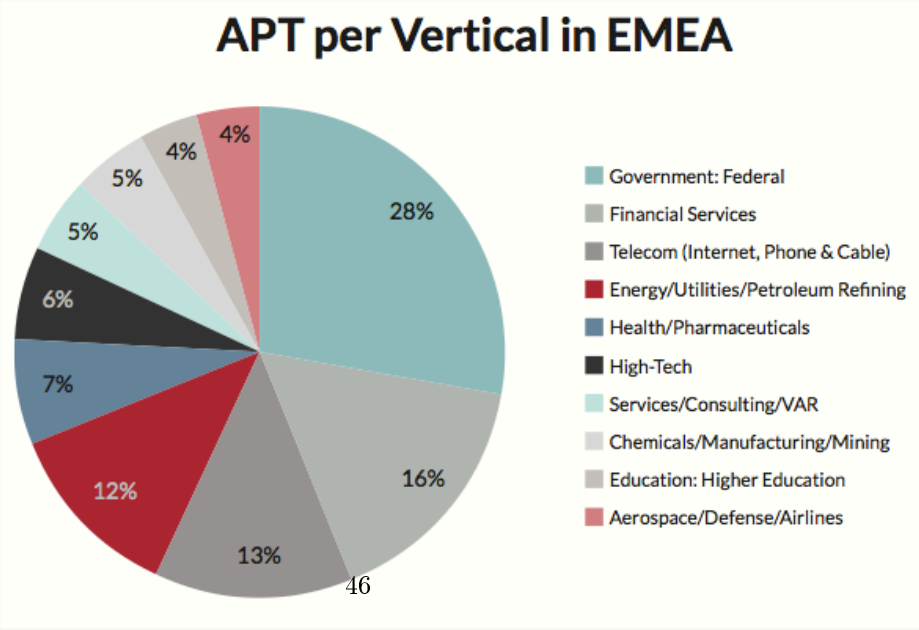
The following paragraph provides an in depth analysis of the top four verticals impacted.

Government:

Based on these findings we expect that government agencies and institutions will likely continue to face threats from financially motivated threat actors who are in search of personal or sensitive data. Central agencies and institutions that maintain citizens' data, like departments of revenue, are likely to be particularly at risk, due to the potentially valuable information stored on their networks. Local government entities may additionally face threats from cyber actors interested in testing their own skills or foreign government network defences.

Organisations in EMEA are almost certain to face cyber espionage risks from state-sponsored or state-associated threat actors working for or in association with nation-state governments. The Middle East in particular is one of the most politically volatile regions in the world and boasts some of the world's largest oil reserves, making it an area of strategic focus for many states outside the region. These countries almost certainly will employ cyber espionage capabilities to monitor their economic, political, and military interests, which will likely drive the further development of local cyber espionage efforts.

Figure 5: APT Malware Detections per Vertical



Agencies and institutions whose networks are connected to those of other local government entities also face potential risks from threat actors moving laterally from an initially compromised network. In a case study, we noted that the threat actors were able to move laterally from an initial compromise at a financial institution and gain access to the networks of other departments in the state. However, this group was also able to compromise the network of a local government outside of the original geography.

We suspect that a nation state actor may opt to target a local government network as opposed to that of a central government entity as the local network poses an easier and less complex target. Local governments likely lack the resources for stringent network security and monitoring, making them a technically easier target for threat actors. However, despite the relatively lax network security, local government networks also likely contain potentially valuable information for nation state threat actors, including insight into major industries operating within their jurisdictions, as well as personnel and financial data.

Financial Services:

FireEye suspects the large amount of activity in the sector is partly due to the diverse motivations of threat actors in the industry, to include (1) China-based APT actors seeking to support economic reforms and reach state goals, (2) financial threat actors seeking to financial gain through the direct theft of funds of the indirect theft of information to be sold, and (3) disruptive threat actors and hacktivists seeking to gain publicity, divert banks' attentions, or demonstrate a political motive. Any one of these threats would increase activity in an industry, but the presence of all three likely accounts for the large number of intrusions in the financial services industry.

Additionally, as financial advisors are often at the heart of the mergers and acquisition process, this is a sensitive time for organisations seeking to maintain some level of secrecy. It is also a potential strategic intelligence opportunity for threat actors seeking to collect valuable information and insights. FireEye has observed a number of APT groups target organisations during the mergers and acquisitions process.

We suspect that the threat actors conducted these operations in order to collect information that would prove advantageous during subsequent contract negotiations with the targeted organisations, as well as information collection for possible foreign government scrutiny and insight.

Candidate organisations with unidentified intrusions and unaudited networks pose a risk for any acquiring organisation. These risks include subsequent financial and reputational damage to parent organisations, and extending the possibility of spreading an existing compromise to the acquiring organisation's networks. Though for a different purpose, FireEye has observed APT groups target and compromise a target organisation's circle of providers, partners, and advisors as a means to leverage any bridged networks and gain access to the target organisation.

Energy:

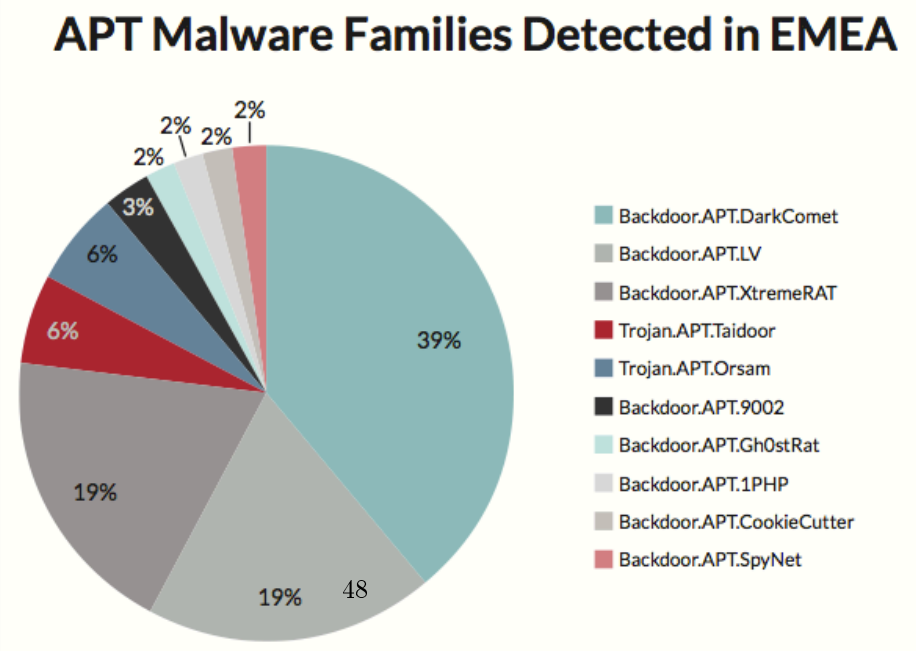
We have observed threat actors using HAVEX/ PEACEPIPE malware to try and compromise energy targets; Nordic energy companies and an EMEA state's national oil company have been recent targeted. We believe that multiple actors operating out of Russia are behind these campaigns.

APT Malware Families

The following graph represents the distribution of targeted malware families identified in the first half of 2014. The malware families are important to track from a risk perspective, as each family has different capabilities and risks to consider. This becomes significant when we can link specific malware use to threat actors or threat types, which aids in attribution and enables people to respond more effectively.

DarkComet, njRAT (LV), Taidoor, and XtremeRAT were the malware variants that FireEye appliances in the EMEA region detected the most frequently. DarkComet, njRAT, and XtremeRAT are all publicly available, easy-to-use RATs—njRAT is particularly popular in the region, and its author is based in Algeria. As such, they could be used by advanced attackers to "blend in", but they could also be employed by different types of threat actors with all manner of motives because of their ease of access and low barrier to entry. In addition, we have only observed suspected and confirmed China-based APT groups use Taidoor. The clients affected by this alert, all of which were in the federal government or energy industry verticals, track closely with the targeting pattern of the confirmed APT group, giving further credence to our belief that Chinese threat actors conduct cyberespionage against organisations in this region.

Figure 6:
APT families detected in EMEA



Rather than building custom malware and exposing valuable zero day exploits, many threat actors behind targeted attacks use publicly or commercially available remote access Trojans (RATs). This pre-built malware often has all the functionality needed to conduct cyber espionage and is controlled directly by the threat actor, who frequently possess the ability to adapt to network defences. As a result, the threat posed by these RATs should not be underestimated. On any given day detection by traditional security solutions for these well-known RATs varies widely – some may be well-known and detected quickly, others will remain undetected for months.

However, it is difficult to distinguish and correlate the activity of targeted threat actors based solely on their preference to use particular malware — especially freely available malware. From an analyst's perspective, it is unclear whether these actors choose to use this type of malware simply out of convenience or in a deliberate effort to blend in with traditional cybercrime groups, who also use these same tools.

DarkComet, for example, has been available for free since 2008. It is popular on a variety of underground forums and used by a wide range of actors for many purposes. (After reports indicated that DarkComet was used in connection with the conflict in Syria, the creator of DarkComet, DarkCoderSC, created a removal tool and ultimately quit developing the RAT).

Although publicly available RATs are used by a variety of operators with different intents, the activity of particular threat actors can still be tracked by clustering command and control server information as well as the information that is set by the operators in the builder. These technical indicators, combined with context of an incident (such as the timing, specificity and human activity) allow analysts to assess the targeted or non- targeted nature of the threat.

FireEye studied a sample 100 active CnC domains for njRAT (includes LV categorisation), XtremeRAT, njw0rm, h-worm, and DarkComet that threat actors used against our customers. Though advanced threat actors are also using these tools, we surmise that various individual hackers and hacking teams are largely conducting these activities for notoriety hacking, hacktivism, cybercrime, or hobby hacking, and not targeted data theft from an APT campaign. Domain resolutions for the 100 dynamic CnC fully qualified domain names (FQDNs) revealed more than 20,000 historical IP resolutions, suggesting that these actors use dynamic domains for connectivity via their local Internet service provider to their personal computers.

Nearly all the C2s domains used a dynamic domain name system, such as no-ip, dyndns, adultdns, zapto, sytes, servequake, myvnc, with a number of the FQDNs individually resolving to hundreds of IP addresses. The sample set of FQDNs resolved to more than 20,000 IP addresses in our historical data and possibly indicated the origins of the activity given the apparent direct use of local Internet service providers. For example, in one case involving more than 600 FQDN-IP resolutions, more than 500 of the IPs appeared to be Jordan-based IPs. Figure 1 below shows the primary and secondary countries1 for FQDN-IP resolutions. FireEye also found cases in which additional FQDNs simultaneously resolved to the same IPs as some of the identified C2 FQDNs.

The extent to which the use of DarkComet, LV and ExtremeRat in attacks that are "targeted", and not opportunistic, is unclear. They could be targeting an entire industry, simply capitalising on opportunities that arise.

Out of the other malware families detected in EMEA, several of them are confirmed to be in regular use by many different China-based APT groups. These include:

• Taidoor

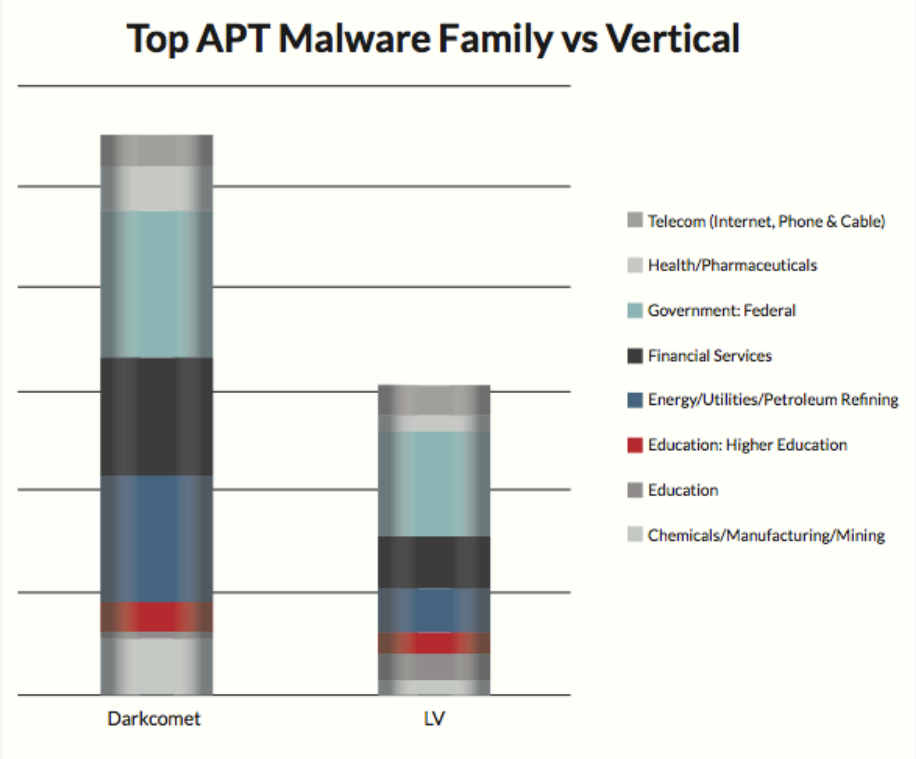
• Orsam aka DOM

• EXCHAIN

• 9002 aka HOMEUNIX

• COOKIECUTTER aka UPS

Figure 7:
Top APT malware family vs Vertical



Darkcomet and LV represent more than 50% of identified APT malware families detected in EMEA. Threat Actors are typically organising their attacks through campaigns that target very specific verticals. If we focus on these two APT families, we find that the Energy and Financial Services have been specifically targeted using the Darkcomet APT.

The following table presents the most popular APT families identified during this assessment:

Gh0stRat : Gh0stRAT is a malicious remote administration tool (RAT). Requiring little technical savvy to use, RATs offer unfettered access to compromised machines. They are deceptively simple—attackers can point and click their way through the target's network to steal data and intellectual property. But they are often delivered as key component of coordinated attacks that use previously unknown (zero-day) software flaws and clever social engineering. Features common to most Windows-based RATs include key logging, screen capturing, video capturing, file transfers, system administration, password theft, and traffic relaying.

9002 : Trojan.APT.9002 (aka HOMEUNIX) is a backdoor that was linked to an adversary that FireEye has named the Sunshop Group. In the past, FireEye has observed attackers leveraging vulnerabilities CVE-2013-0633 and CVE-2013-0634 to deliver this backdoor. More information can be found at: www.fireeye.com/blog/ technical/cyber-exploits/2013/02/lady-boyle-comes-to-town-with- a-new-exploit.html

ExtremeRAT : XtremeRAT is an openly available remote access tool (RAT). The author(s) is unknown, but they advertise the RAT for €350 via PayPal/Western Union. The tool is offered in three different languages; Portuguese, Spanish, and English, with several unique built-in features such as Windows 8 compatibility, opening/closing CD/DVD peripherals, hiding icons, pausing mouse movements, IRC chat functionality, in addition to other more commonly seen RAT capabilities such as key logging and file uploads/downloads. The developer also appears to be actively improving the RAT and offers free updates to paying customers through their website. The payload itself is UPX packed and coded mostly in Delphi, with communications to command and control (CnC) servers by default over port 81. XtremeRAT has previously been seen targeting international government institutions in the U.S., U.K., Turkey, Slovenia, Macedonia, New Zealand, Latvia, Palestine and Israel; most notably, Israeli Police computers were infected in October 2012, forcing the entire network offline for a brief period of time. The RAT is also popular among attackers based in the Middle East, commonly seen in attacks by Operation Molerats actors and also believed to be in use by the Syrian government. http://www.fireeye. com/blog/technical/2013/08/operation-molerats-middle-east- cyber-attacks-using-poison- ivy.html

Leouncia : Backdoor.APT.Leouncia is a powerful backdoor malware program. Leouncia's CnC payload decryption consists of two major phases. The first part is the formulation of a dynamic permutation table using a variable 128 bit key. This permutation table is further used to decrypt the actual payload. Leouncia hibernates itself for an extended period of time. This hibernation is controlled by a file named "readx". Once this command is received, Leouncia tries to read the "readx" file from the current directory. The file "readx" contains the activation date and time in 'FileTime' format like \HIGH DATE\LOW DATE. Leouncia constructs the system time from it and checks if the current date and time is ahead of or equal to this construct. If not it will hibernate itself until that time comes. Eventually, Leouncia enumerate the running process list, encrypts it, and sends it back to its CnC server. It receives dynamic data from the CnC

and writes it to a file specified by the attacker. It reads the attacker's specified file onto the target system and sends its contents back to the CnC. Then the attacker's specified process is applied to the infected system. The given pid (process id) terminates a running process and sends a list of all logical drives back to the attackers. A Windows command prompt is spawned and the attacker runs commands of choice. The attacker specify its commands in response to 'GET' requests and the backdoor component invokes these commands on the Windows command shell and sends the response back to the CnC in the form of a 'POST'.

SpyNet : SpyNet is a malicious remote administration tool (RAT). Requiring little technical savvy to use, RATs offer unfettered access to compromised machines. They are deceptively simple—attackers can point and click their way through the target's network to steal data and intellectual property. But they are often delivered as key component of coordinated attacks that use previously unknown (zero-day) software flaws and clever social engineering. Features common to most Windows-based RATs include key logging, screen capturing, video capturing, file transfers, system administration, password theft, and traffic relaying.

Cybercrime and Hacktivism:

Non-targeted cybercrime is a growing and serious risk to individuals and organisation in EMEA. As we mentioned before, the authors behind two popular remote access tools (RATs), njRAT and h-w0rm, likely reside in Kuwait and Algeria, respectively. Furthermore, most of the C2 domains associated with these malware are located in the Middle East and North Africa. While we have observed both tools used in targeted attacks against companies in the energy and telecommunications sector, they have also been used in run-of-the-mill phishing and cybercrime attacks as well. Cyber criminals will often harvest credential or financial information through logging keystrokes or grabbing credentials stored by a web browser. Though Microsoft recently seized more than 20 top-level domains associated with njRAT and h-w0rm botnets, we believe that regional hacktivists and cybercriminals will continue to rely on these tools, due to their ease-of-use and ability to escape detection by anti-virus security software.

n addition, we have observed local forums develop a cybercrime scene similar to what we have observed in China and Russia: forums with malware for sale and technical mentors who offer advice on evading anti-virus software and using dynamic domain hosting. This suggests growing expertise and specialization, which will likely result in more effective intrusions and cybercrime operations.

Hacktivism:

FireEye expects that high-profile organisations in the Middle East and North Africa, particularly government and military entities, face a high risk of targeting by hacktivists based inside and outside the region. Moreover, we expect that military and political conflicts will further escalate this risk.

Though hacktivists have targeted Israel in the past, an increasing amount of attacks targeted Israeli government agencies and media organisations during July, almost certainly due to increased violence between Israel and Hamas. Members of the Anonymous hacker collective, as part of a campaign dubbed #OpSaveGaza, announced they had taken more than one thousand Israeli websites offline over the course of July, including those belonging to the Bank of Israel, the Israeli Ministry of Justice, and Mossad.

The Syrian Electronic Army (SEA), a hacktivist group that formed during the 2011 Syrian protests, probably works with, if not entirely for, the Syrian government. They resolutely support President Assad and an alleged member of the group is the son of a powerful Syrian intelligence officer. The SEA has proven to be a prolific and public threat group—in one case FireEye's Mandiant Consulting Services team responded to, SEA threat actors targeted a media organisation with spear phishing emails and gained access to the company's Twitter feeds within 24 hours. The SEA has also targeted a variety of media organisations with Trojans such as njRAT and XtremeRAT, and has waded into other conflicts unrelated to the Syrian civil war: The SEA recently claimed responsibility for hacking the IDF's Twitter feed and posting false statements about missiles causing a leak at an Israeli nuclear facility.

Hacktivists have also targeted oil and gas companies in the region, to limited success: One group called "AnonGhost," who claim to be made up of Muslim hackers from around the world, threatened to attack oil companies in Kuwait, Saudi Arabia, and other countries it perceives to be acting in the interests of the United States and Israel. Their operation, however, resulted in little damage, taking only a few websites offline with DDOS. Nonetheless, as Internet access grows in MENA countries, we anticipate local hacktivist movements to grow in popularity and effectiveness.

5. Conclusion and Recommendations

The evidence highlighted in this report demonstrates that organisations in EMEA continue to be targets for advanced threats. The type of malware identified is consistent with what we see in other countries and verticals. Attackers are targeting high value organisations in (EMEA) and are making their way in. The high number of APT events suggests a large level of information theft.

We recommend the following:

1. Assume you and your organisation is a target and that your existing security controls can be bypassed

2. Establish a cyber-risk framework that enables the business with board level sponsorship

3. Establish an incident response/management service in a SOC/CIRT team to be able to

detect and react to an APT event quickly

4. Enhance your visibility with external threat intelligence to understand who might attack you and how to avoid the tools, techniques and procedures they use

5. Bring in the right technology that could identify an APT.

Utilisation de l'hétérogénéité des réseaux de capteurs sans fil pour accroître la résilience de la solution de détection d'intrusion

David Espes[a*], Nora Cuppens[b], Frédéric Cuppens[a], Philippe Le Parc[a]

a. IRT B-COM, LabSTICC – UMR CNRS 6285,
13 rue Claude Chappe, 35510 Cesson-Sévigné, France
b. Télécom-Bretagne, LabSTICC – UMR CNRS 6285,
2 rue de la Chataigneraie, 35576 Cesson-Sévigné, France
* david.espes@b-com.com

Catégorie : Spécialisée

Résumé. Les réseaux de capteurs sans fil sont généralement déployés dans des environnements hostiles mettant à rude épreuve ce type de réseau. De par l'aspect collaboratif de ce type de réseaux, des nœuds peuvent être compromis pouvant altérer drastiquement le comportement du réseau. Il est largement accepté que les systèmes de détection d'intrusion sont un des mécanismes les plus performants pour se prémunir contre la compromission des nœuds et les accès non autorisés. Nous mettons en avant dans ce papier la faiblesse des systèmes de détection d'intrusion collaboratifs se basant sur un mécanisme de consensus pour déceler la compromission ou non d'un nœud. Nous proposerons dans un deuxième temps une solution pour accroître la résilience de ces réseaux.

Mots clés : Réseaux de capteurs sans fil, Détection d'intrusion, Consensus

1 Introduction

Ces dernières années, l'émergence de réseaux de capteurs sans fil (RCSF) n'a cessé de croître. Ils couvrent un très large domaine d'activités [1,2] telles que le domaine militaire, de la santé, commercial et des habitations intelligentes. Dans l'ensemble de ces domaines, la sécurité des réseaux de capteurs sans fil est un véritable problème qui doit être pris en considération avant tout déploiement. La sécurité des RCSF fait face à un nombre très important de challenges qui n'a pour l'heure jamais été rencontré dans d'autres réseaux sans fils. Ces problèmes sont particulièrement présents à cause de trois phénomènes : l'environnement dans lequel opère ce type de réseau, les caractéristiques matérielles des nœuds et le comportement des nœuds entre eux [3-5]. L'environnement joue un rôle important dans les problèmes de sécurité rencontrés par les RCSF du fait que le support de transmission est un support à diffusion donc tout le monde peut écouter ou transmettre des messages. De même, les capteurs sont déployés dans un environnement hostile pouvant offrir un accès physique à l'équipement et une capacité à l'attaquant de prendre le contrôle d'un nœud. Les ca-

ractéristiques matérielles des nœuds sont également problématiques de par l'architecture de ces nœuds fortement contraintes d'un point de vue énergie (les nœuds fonctionnent sur batterie) mais également en puissance brute (la puissance processeur et la capacité en RAM est très faible). De fait, les mécanismes de sécurité actuels ne pourront pas être utilisés dans ces réseaux. Le dernier point, mais non des moindres, concerne le comportement du réseau. Le réseau est considéré comme autonome et auto-configurable donc les nœuds participent au relai des messages pour propager l'information. Ce comportement coopératif va permettre à certains nœuds de profiter de la confiance établie avec les autres nœuds pour avoir un comportement inopportun et pénaliser lourdement le fonctionnement global du réseau.

Les propriétés de sécurité conventionnelles telles que l'authentification, le chiffrement, la gestion des clefs sont une première barrière pour se prémunir contre certaines attaques. Cependant, ces solutions ne peuvent pas prémunir un tel réseau contre tous les types d'attaque. En effet, un attaquant est susceptible de compromettre un nœud et donc d'être perçu légitimement par l'ensemble des nœuds sains du réseau. Les mécanismes de détection d'intrusion peuvent donc être perçus comme une deuxième ligne de défense qui permet de déceler le comportement inapproprié d'un nœud [6,7].

Les systèmes de détection d'intrusion doivent être capables de différencier le comportement des nœuds malveillants de ceux qui sont sains. Dans les RCSF, le fonctionnement des systèmes de détection d'intrusion doit être adapté aux contraintes particulières de ces réseaux. En effet contrairement aux réseaux filaires, le système de détection d'intrusion ne peut pas être centralisé. Dans un système centralisé, le système de détection d'intrusion (IDS) s'exécute sur un équipement qui étudie le comportement des autres équipements du réseau ou l'ensemble des messages échangés pour détecter une attaque. Dans un tel cas, l'IDS est seul pour décider du comportement malveillant ou bienveillant d'un équipement. Dans les RCSF, les ressources des nœuds sont bien trop limitées pour pouvoir étudier/comparer le comportement de chaque nœud ou des messages échangés. L'IDS doit donc être distribué sur l'ensemble des nœuds du réseau pour former un système distribué [8-10].

Dans un tel système, chaque nœud possède un module local de détection d'attaque. Chaque nœud écoute le réseau afin de déceler le comportement déviant d'un nœud voisin. Lorsqu'une attaque est détectée, un nœud peut réagir de deux manières : réduction de la participation du nœud malveillant aux tâches communes en modifiant sa réputation ou isolation de la menace à l'aide d'un consensus (aussi appelé vote). Dans le cas du mécanisme de réputation [11,12], les nœuds vont pénaliser leurs voisins si ces derniers ne participent pas suffisamment à l'effort collectif. Un nœud malveillant qui effectuerait des attaques verrait sa popularité diminuer jusqu'à être isolé du réseau. Le mécanisme de réputation fait preuve d'une grande souplesse du fait qu'un nœud malveillant doit maîtriser les attaques commises sur le réseau afin de maintenir un niveau de confiance suffisant pour participer le plus longtemps possible au fonctionnement du réseau. A contrario, le mécanisme de réputation repose uniquement sur le module de détection d'un nœud. Chaque nœud se fait sa propre idée de la réputation de ses voisins en fonction des actions qu'il perçoit. Certaines attaques, telles que le WormHole, nécessitent d'avoir une connaissance plus large de l'environnement

pour les détecter (il est nécessaire au plus de connaître l'environnement voisin d'un nœud suspect). Le mécanisme de réputation est donc particulièrement sensible aux faux négatifs. En effet, certains nœuds ne percevront pas qu'un autre nœud est en train de commettre une attaque et maintiendront une réputation élevée pour ce nœud. Le mécanisme de consensus, contrairement au mécanisme de réputation, se veut beaucoup plus réactif puisqu'il permet d'isoler immédiatement la menace. Cependant, il nécessite l'approbation des nœuds voisins de l'occurrence de la menace (donc à 1 ou 2 sauts) avant de l'isoler. Lorsqu'un nœud détecte une attaque, il va diffuser une alerte dans son entourage. Tous les nœuds se trouvant dans le voisinage de la menace vont partager leurs votes en spécifiant si le nœud est malveillant ou non. Lorsque le nombre de vote allant dans la même direction (nœud malveillant ou non) dépasse un certain seuil, un consensus est trouvé et le nœud malveillant est isolé. Ce type de mécanisme est plus efficace en termes de détection que les mécanismes de réputation. En effet, l'environnement complet autour d'un nœud suspect est pris en compte. De fait, le nombre de faux négatifs est bien moindre par rapport au mécanisme de réputation. Par contre la décision n'étant pas prise seule, ce mécanisme est sensible à la compromission de nœuds participant au consensus. Il est donc essentiel de renforcer la résilience du mécanisme de consensus.

Le seuil peut être représenté de deux manières :

- Constante [13-15] : lorsque le nombre de vote, allant dans le même sens, dépasse une certaine valeur, le nœud est reconnu comme malveillant.
- Majorité des nœuds [16-19] : c'est le cas le plus courant. Lorsque la moitié des nœuds voisins du nœud attaquant vote à la majorité que ce nœud est malveillant, il se trouve donc isolé du réseau.

Le bon fonctionnement des types de consensus, énoncés précédemment, est dépendant du nombre de nœuds malveillant participant à ce consensus. En effet dans le cadre du consensus avec un seuil égal à une constante, le nombre d'attaquants participants ne doit pas dépasser ce seuil sinon le consensus peut être corrompu. Sinon, un nœud sain (respectivement malveillant) peut être annoncé malveillant (respectivement sain). Dans le cas du consensus à la majorité, le nombre de nœuds corrompus ne doit pas dépasser la moitié des nœuds. En effet, si f représente le nombre de nœuds malveillants présents dans le voisinage du nœud contrôlé, le nombre de nœuds voisins doit au moins être de $2f+1$ pour que le consensus ne soit pas corrompu. De fait avec de tels mécanismes de consensus, le nombre de nœuds compromis ne doit pas excéder les 50% de nœuds du réseau.

Nous démontrons, dans ce papier, qu'en utilisant l'hétérogénéité des RCSF on peut proposer un mécanisme de consensus qui tolère bien plus de 50% de nœuds corrompus présents dans le réseau. En effet, lorsqu'un attaquant trouve une faille de sécurité dans un nœud, il peut utiliser la même faille sur tous les nœuds ayant les mêmes caractéristiques logicielles et matérielles (même système d'exploitation, même logiciels installés, etc.). Afin de limiter l'impact de la redondance des failles sur les systèmes, nous montrons qu'il est préférable de réaliser un consensus entre groupes de nœuds disjoints. De fait si un type de nœud est extrêmement répandu dans le RCSF,

l'attaquant ne devra pas se limiter à corrompre un tel type mais à corrompre un ensemble de types différents. Sa tâche est donc largement complexifiée et la résilience du réseau largement accrue.

Le papier est structuré comme suit : la section 2 introduit le problème de consensus, et la partie 3 introduit des concepts pour améliorer la résilience des RCSF. Cette partie sera étendue dans la version longue de l'article pour proposer un mécanisme de consensus robuste pouvant supporter bien plus que 50% des nœuds corrompus. En section 4, nous présentons une solution au problème du consensus et en section 5 nous concluons sur le problème de consensus des mécanismes actuels.

2 Problème du consensus

Afin de déterminer si un nœud commet un impair, certains nœuds écoutent donc pendant une période de temps donnée les messages échangés par leurs voisins et les analysent afin de déceler une attaque. Un seul nœud ne peut-être suffisant pour détecter qu'une attaque a lieu. En effet, un nœud i peut être corrompu et annoncer qu'il vient de déceler qu'un nœud j vient de commettre une attaque alors que ce dernier n'est en rien compromis. La possible compromission de certains nœuds ne permet donc pas d'affirmer qu'une alerte remontée par un nœud soit réelle. Un consensus doit être trouvé par l'ensemble des nœuds voisins d'un nœud contrôlé.

2.1 Faiblesse du consensus

La probabilité de corrompre le consensus réside dans la probabilité de compromettre un ensemble suffisant de nœuds qui correspond à un seuil qui peut être représenté soit par la majorité soit par une constante c. L'ensemble des nœuds voisins d'un nœud x appartient à un ensemble V qui est composé d'éléments caractérisés par les mêmes composants matériels et logiciels. Chacun de ces éléments est disjoint avec les autres. Donc $V=\{V_1,\ldots,V_n\} \mid \forall i,j\ V_i \cap V_j = \varnothing$. Nous définissons deux opérateurs :

- $|X|$ est la cardinalité de l'ensemble X i.e., le nombre d'élément de X
- $N(X)$ est le nombre de nœuds de X i.e., la somme du nombre de nœuds de chaque élément

On suppose que la probabilité d'un attaquant à corrompre un nœud, en trouvant par exemple une faille de sécurité dans son système d'exploitation ou les applications qui le composent, est de p. Un attaquant qui est capable d'accéder à tous les nœuds possédant la même version de l'OS ou les mêmes applications, pourra compromettre ces derniers en utilisant la même faille de sécurité. De fait, pour tous les nœuds qui possèdent les mêmes caractéristiques matérielles et logicielles, la probabilité de corrompre l'ensemble des nœuds est la probabilité d'en corrompre un. La probabilité de corrompre un ensemble est indépendante de corrompre les autres.

Propriété 1 : $\forall\ V_i \in V$, la probabilité de corrompre un élément V_i est donné par $P_c(V_i)=p$

Un attaquant peut être classé en fonction de ces aptitudes. Dans cet article, nous considérons un attaquant fort qui peut donc écouter/communiquer à un instant t avec l'ensemble des nœuds du réseau. De fait, il peut avoir une grande mobilité de déplacement ou des équipements perfectionnés telles que des antennes directionnelles à gain élevé pour atteindre/écouter un plus grand nombre de nœuds. Son objectif est de biaiser le consensus ayant lieu et de faire en sorte d'inverser le comportement malveillant ou non d'un nœud. Nous définissons donc les possibilités de l'attaquant comme suit :

Définition 1 : un attaquant peut compromettre un nœud avec une probabilité p et peut communiquer ou écouter l'ensemble des nœuds du réseau sans contrainte particulière.

Pour inverser un consensus, l'attaquant doit corrompre un ensemble de nœuds suffisamment important et qui dépasse un certain seuil. Il est possible de définir un ensemble C qui représente des éléments de V dont le nombre de nœud dépasse ce seuil. La propriété définissant l'ensemble C est la suivante :

Propriété 2 : $C=\{C_1,\ldots,C_m\} \mid \forall i\ C_i \subset V \wedge \mathrm{N}(Ci) \geq \text{seuil}$

En fonction des propriétés 1 et 2, on peut déduire la probabilité de corrompre le consensus qui est donnée par la propriété suivante :

Propriété 3 : Soit $C=\{C_1,\ldots,C_m\}$, la probabilité de corrompre le consensus est la probabilité de corrompre au moins un élément de C donc :

$$P_{cc}(C) = \sum_{i=1}^{m} p^{|C_i|}(1-p)^{|V|-|C_i|}$$

En fonction de la propriété 3, on peut déterminer la probabilité $P_c(C_i)$ de corrompre un ensemble C_i de nœuds. Cette probabilité est énoncée par la propriété 4 :

Propriété 4 : $\forall C_i \in C,\ \exists C' = \{C'_1, \ldots, C'_o\} \subset C \mid \forall j \in \{1 \ldots o\}\ C_i \subset C'_j \wedge C_i \not\subset C \setminus C' \Rightarrow P_c(C_i) = \sum_{i=1}^{o} p^{|C'_i|}(1-p)^{|V|-|C'_i|}$

L'objectif de l'attaquant sera de trouver le plus petit sous-ensemble de nœuds qui satisfait le seuil de consensus. De fait, l'attaquant devra compromettre un sous-ensemble de nœuds avec la probabilité la plus grande. En fonction de la propriété 4, on peut en déduire la propriété suivante :

Propriété 5 : Soit $C=\{C_1,\ldots,C_m\}$, l'attaquant doit corrompre $C_i \mid P_c(C_i) = \max_{j=1,..,m} P_c(C_j)$

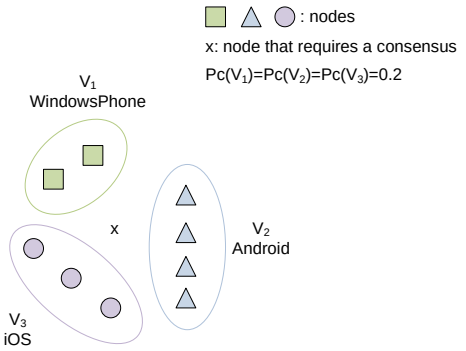


Fig. 1. Exemple d'ensemble de nœuds disjoints en fonction de leur système d'exploitation

Un exemple peut être donné en Figure 1. Trois ensembles distincts de nœuds sont voisins d'un même nœud *x*. Chaque ensemble est disjoint des autres de par leur système d'exploitation. Donc $V=\{V_1, V_2, V_3\}$. L'ensemble V_1 (respectivement V_2 et V_3) utilise le système d'exploitation WindowsPhone (respectivement Android et iOS). L'ensemble V_1 (respectivement V_2 et V_3) est composé de 2 nœuds (respectivement 4 et 3 nœuds). Le seuil pour le consensus est de 4. De fait, il nécessite qu'au moins 4 nœuds aient décelé une attaque pour annoncer que *x* soit corrompu. La probabilité de corrompre V_1, V_2 ou V_3 est la même et donc $\forall i \in \{1,\ldots,3\}$ $P_c(V_i)=0.2$. L'attaquant doit corrompre un élément de V dont le nombre de nœud est supérieur ou égal à 4. L'ensemble d'éléments qui peut dans ce cas être corrompu est $C=\{V_2, \{V_1, V_2\},\{V_1, V_3\},\{V_2, V_3\},\{V_1, V_2, V_3\}\}$.

En fonction de la propriété 3, la probabilité de corrompre le consensus est :

$$P_{cc}(C) = 0.2 \times (0.8)^2 + 3 \times 0.2^2 \times 0.8 + 0.2^3 = 0.232$$

En fonction de la propriété 4, on peut déduire la probabilité de corrompre chaque élément :

$$\mathrm{P_C}(\{\mathrm{V_1}, \mathrm{V_2}, \mathrm{V_3}\}) = 0.2^3 = 0.008$$
$$\mathrm{P_C}(\{\mathrm{V_1}, \mathrm{V_2}\}) = \mathrm{P_C}(\{\mathrm{V_1}, \mathrm{V_3}\}) = \mathrm{P_C}(\{\mathrm{V_2}, \mathrm{V_3}\}) = 0.2^2 \times 0.8 + 0.2^3 = 0.04$$
$$\mathrm{P_C}(\mathrm{V_2}) = 0.2 \times 0.8^2 + 2 \times 0.2^2 \times 0.8 + 0.2^3 = 0.2$$

De fait, l'exemple met bien en exergue que l'attaquant a tout intérêt à corrompre le plus petit sous-ensemble de nœud afin de modifier le consensus. Dans cet exemple, la probabilité de corrompre V_2 est bien plus grande que la probabilité de corrompre des ensembles plus importants.

2.2 Mise en veille des nœuds

Les équipements fonctionnant principalement sur batterie ces derniers doivent économiser au mieux leurs ressources. Pour cela, ils ne peuvent pas écouter en permanence le réseau ce qui réduirait drastiquement leur durée de fonctionnement. Les nœuds ont donc des phases d'éveil durant lesquelles ils participent au mécanisme de détection d'intrusion et des phases de sommeil durant lesquelles ils coupent leurs

émetteurs/récepteurs afin de ne plus écouter le réseau. Ces états successifs de sommeil et d'éveil vont réduire le nombre de nœuds corrompus que l'attaquant va pouvoir utiliser à un instant *t* pour faire pencher la balance de son côté lors d'un consensus. En outre, uniquement les nœuds corrompus en état d'éveil pourront participer au consensus. Soit *m* la probabilité d'éveil d'un nœud qui peut être représenté par la quantité de temps passé en éveil $t_{éveil}$ sur son temps de fonctionnement t_{total}. Cette probabilité est donnée par l'équation suivante :

$$m = \frac{t_{éveil}}{t_{total}} \quad (1)$$

A un instant *t*, on note $V_{éveil}$ les nœuds en éveil de l'ensemble *V*. Suivant l'équation 1, pour tout ensemble $V_i \in V$ on obtient en moyenne :

$$\left|V_{i_{éveil}}\right| = m|V_i| \quad (2)$$

En fonction de la propriété 3 et de l'équation 2, $C_{éveil}$ est inclus dans *C* donc on obtient l'équation :

$$P_{cc}(C) \geq P_{cc}(C_{éveil}) \quad (3)$$

On peut aisément déduire de l'équation 3, que la mise en veille des nœuds peut compliquer la tâche de l'attaquant. Tout de même, un attaquant fort est susceptible de modifier le fonctionnement d'un nœud l'empêchant de rentrer dans un état de sommeil. Ces nœuds supplémentaires participeront au consensus et faciliteront d'autant plus la compromission du consensus. On appellera ce type d'attaque, l'attaque du Somnambule.

3 Améliorer la résilience du consensus

Dans un réseau hétérogène, d'après la propriété 5, l'attaquant se focalisera à compromettre le plus petit élément pour corrompre le consensus. De fait, il pourra orienter la décision qu'un nœud compromis est en fait sain et inversement qu'un nœud sain soit compromis.

La solution de détection d'intrusion reposant sur le consensus pour déceler si une attaque a lieu ou non, doit prendre en compte l'hétérogénéité des réseaux afin d'accroître la résilience de l'infrastructure sous-jacente. Elle doit assurer au consensus une probabilité d'être corrompu la plus faible possible. Pour cela, la solution de détection d'intrusion devra s'assurer qu'une alerte soit remontée par le plus grand nombre de sous-ensemble de nœuds tel que mentionné par la propriété suivante :

Propriété 6 : Soit $C=\{C_1,\ldots,C_m\}$, la solution de détection d'intrusion doit s'assurer que l'attaquant doit corrompre $C_i \mid P_c(C_i) = \min_{j=1,..,m} P_c(C_j)$

Une solution de détection d'intrusion, basée sur la propriété 6, permet de réduire l'impact d'un grand nombre de nœuds identiques sur le résultat du consensus. Une

telle solution permet de réduire le taux de faux positifs et de faux négatifs grâce à une résilience accrue du mécanisme.

4 Solution proposée

La solution proposée permet de classer les nœuds d'un réseau en groupe de manière autonome afin de maximiser la résilience du consensus. Il est important de souligner que la méthode est totalement autonome et auto-organisée. Elle ne requiert l'aide d'aucune tierce partie pour grouper les nœuds et pour améliorer la résilience du consensus. De fait, la méthode reste rétro-compatible avec les solutions de détection d'intrusion existante puisqu'il suffit de changer le module de résolution du consensus pour que la méthode fonctionne. Les autres modules des solutions de détection ne sont en rien impactés.

4.1 Grouper les nœuds

L'administrateur du réseau ou l'autorité de certification définira la granularité des caractéristiques qui s'appliqueront au groupement. Par exemple, l'administrateur peut définir une granularité large telle que le système d'exploitation ou une granularité plus fine telles que le couple système d'exploitation et numéro de version.

Chaque nœud du réseau se verra fournir ses caractéristiques soit avant la phase de déploiement soit de manière périodique par le réseau. Ces caractéristiques pourront être signées par une entité de confiance (administrateur, collecteur…) afin d'en assurer l'intégrité. En cas de signature numérique, chaque nœud du réseau devra connaître la clef publique de l'entité de confiance.

Les caractéristiques de chaque nœud peuvent être représentées sous forme de message. Le détail d'un message d'annonce est représenté par la Figure 2.

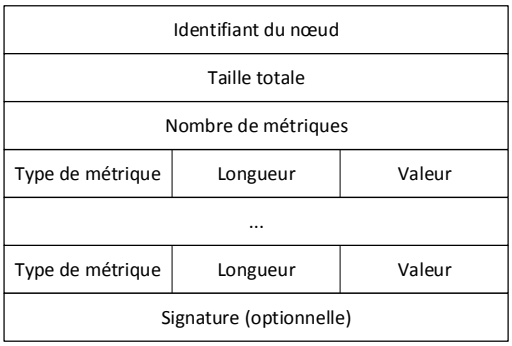


Fig. 2. Message d'annonce des caractéristiques d'un nœud

Le détail des champs d'entête du message est donné ci-dessous :

– Identifiant du nœud : il représente l'adresse du nœud qui émet le message ;

- Taille totale : elle représente la quantité d'information présente dans le message. Elle permet de déterminer la présence ou non du champ optionnel ;
- Nombre de métriques : il permet de définir combien de métriques sont utilisées pour grouper les nœuds ;
- Signature : elle est optionnelle. Elle permet de signer l'intégralité du message pour prouver que ce dernier est bien donné par l'autorité de confiance (administrateur, propriétaire du réseau, collecteur…). Un nœud malveillant ne pourra donc pas modifier ses caractéristiques pour changer de groupe. En effet, chaque nœud est susceptible de vérifier la conformité de ce message. Si cette option est activée, l'autorité de confiance doit générer un couple de clef privée/publique. La clef privée sert pour la génération de la signature et la clef publique est donnée aux nœuds avant leur déploiement. Le choix du mécanisme de chiffrement asymétrique (RSA, courbes elliptiques…) est laissé à la responsabilité de l'autorité de confiance.

Les champs suivants représentent les caractéristiques de chaque nœud :

- Type de métrique : elle prend la forme d'une caractéristique générale (système d'exploitation, numéro version, constructeur du processeur, type processeur…) ;
- Longueur : elle définit la taille du champ valeur. Elle permet de déterminer le début du prochain type de métrique ;
- Valeur : elle désigne la métrique qui est utilisée et qui caractérise le système d'exploitation. Par exemple si le type de métrique représente le système d'exploitation (respectivement le type de processeur), on peut retrouver la désignation commerciale du système (respectivement du processeur) utilisé par le nœud ;

Chaque nœud va propager périodiquement un message d'annonce à ses voisins à deux sauts en plus de son message de voisinage. Chaque nœud maintient une table de voisinage qui contient l'identifiant des nœuds voisins, ses caractéristiques et la table des nœuds voisins à un saut pour chaque nœud voisin. Chaque nœud à deux sauts pourra donc savoir quel nœud est voisin d'un autre et quelles sont ses caractéristiques. Il pourra ainsi déterminer le nombre de groupe en périphérie d'un de ses voisins.

4.2 Consensus de groupe

Le mécanisme de consensus fonctionne pour n'importe quelle architecture distribuée. Elle pourra aussi bien être opérationnelle pour des RCSF totalement autonomes ou au contraire des RCSF hiérarchisés.

Lorsqu'un nœud détecte un comportement suspect d'un nœud, il diffuse un message de vote dans son voisinage à deux sauts. Ce message de vote contient l'identifiant du nœud émetteur, l'identifiant du nœud suspect et le type du vote (nœud compromis ou non). Il peut, optionnellement, contenir le message d'annonce du nœud émetteur. A la réception d'un message de vote, le nœud regarde, dans sa table de voisinage, si le nœud suspect fait parti d'un de ses voisins. Dans le cas où ce n'est pas un nœud voisin, il supprime le message de vote. Par contre, s'il est présent dans le voisinage du nœud suspect, il diffuse à son tour un message de vote. Au bout d'un certain laps de

temps, l'ensemble des nœuds voisins du nœud suspect a reçu les votes de chacun. Ils connaissent, donc, les groupes qui ont votés la compromission d'un nœud.

Il se peut que deux nœuds appartenant aux mêmes groupes n'aient pas le même vote. En effet, la détection de certaines attaques, telles que les attaques par WormHole, est sensible à la position des nœuds. Il suffit d'avoir au moins un nœud dans le groupe dont le vote est positif sur la compromission du nœud suspect pour que le vote de la totalité du groupe soit positif. Cette spécificité de notre solution ajoute à la résilience du consensus par rapport aux mécanismes de consensus conventionnels. Cette particularité de notre mécanisme permet de diminuer le nombre de faux négatifs.

Afin de réduire le nombre d'informations transmises sur le réseau, un nœud, qui ne détecte pas de comportement anormal du nœud suspect, n'est pas obligé de transmettre un message de vote dont le type du vote est négatif.

Chaque nœud voisin du nœud suspect, après un certain temps d'attente, comptabilisera les votes reçus. En regardant sa table de voisinage, il est capable de déterminer le nombre de groupe total présent dans le voisinage direct du nœud suspect. Pour chaque vote reçu dont le type est positif et en consultant sa table des voisins, il peut connaître le groupe du nœud votant.

Après avoir déterminé le nombre de groupes total présent autour du nœud suspect et le nombre de groupe dont le vote est positif, chaque nœud peut calculer le consensus. Si le nombre de groupe dont le vote est positif dépasse un certain seuil, dans ce cas le nœud est déterminé comme compromis. Afin d'apporter le maximum de résilience au mécanisme du consensus, le seuil doit être égale à la moitié du nombre de groupe total présent dans le voisinage d'un nœud suspect.

Lorsqu'un nœud est détecté comme compromis, le collecteur ou l'administrateur du réseau peut être prévenu de sa compromission. Dans ce cas, une liste noire est maintenue qui contient les caractéristiques de chaque groupe compromis. Cette liste noire est diffusée périodiquement à l'ensemble des nœuds du réseau. De même, lorsqu'une faille est détectée par l'administrateur ou une autorité de confiance, la liste noire peut être enrichie des caractéristiques sensibles à la nouvelle faille de sécurité. L'ensemble des groupes présents dans la liste noire ne participeront plus au mécanisme de consensus. A tout moment, les groupes peuvent être réhabilités une fois les problèmes de sécurité corrigés. Une telle fonctionnalité permet d'accroître la réactivité à laquelle sont prises en compte les vulnérabilités détectées durant la durée de vie du réseau.

4.3 Avantages de la solution

Les avantages de la solution sont nombreux :

- Accroissement de la résilience du réseau : le fait de grouper les nœuds en fonction de leurs caractéristiques oblige l'attaquant à déployer des ressources supplémentaires pour changer le vote en sa faveur. En effet, il nécessite de trouver des failles de sécurité sur un ensemble de catégorie de nœud et non pour une seule catégorie.

Les mécanismes de consensus actuels se limitent à un maximum de 50 % de nœuds corrompus. Avec la solution proposée, ce seuil peut largement être dépassé.

- Economie d'énergie : la solution proposée est particulièrement efficace en termes d'économie d'énergie. Elle n'est pas sensible à des attaques par éveil forcé des nœuds. En effet, le nombre de nœuds corrompus n'entre plus en compte. Il est uniquement question du nombre de groupe dans le consensus. Il est donc possible d'avoir un grand nombre de nœuds en veille dans chaque groupe sans pénaliser la résilience du consensus.
- Réduction des faux positifs : chaque groupe peut exécuter des méthodes de détection d'attaque différentes. La détection de faux positifs par certaines méthodes peut donc être compensée par d'autres. En effet, le poids n'est plus sur chaque nœud mais sur le groupe. Chaque méthode de détection aura donc un poids équivalent dans la détection d'intrusion.
- Réactivité de la solution : la solution se veut particulièrement réactive en cas de détection d'une faille de sécurité sur un groupe de nœuds. Un élément de confiance (administrateur, autorité de certification…) peut propager une liste noire de caractéristiques de nœuds qui ne pourront pas participer à la détection d'intrusion. Le vote de tels groupes ne sera donc pas pris en compte dans le mécanisme de consensus. Un groupe peut être réhabilité lorsque la faille est corrigée sur l'ensemble des nœuds du groupe. La réhabilitation entraine que le groupe pourra, à nouveau, participer au mécanisme de consensus.

Suite à ces nombreux avantages, on peut facilement se rendre compte que l'hétérogénéité des réseaux joue un rôle majeur dans l'accroissement de la résilience du réseau. Une telle méthode de consensus offre des avantages considérables quelque soit le type de réseau. On pourrait étendre le concept à nombre de réseaux qui aujourd'hui utilisent une base sans-fil et dont les équipements ont des ressources limitées.

5 Conclusion

Ce papier met en avant que dans les mécanismes de consensus actuels si un grand nombre de nœuds possède les mêmes caractéristiques logicielles, il est suffisant pour l'attaquant de corrompre un de ces nœuds pour faire basculer le consensus en sa faveur. En effet, l'ensemble des nœuds ayant les mêmes caractéristiques logicielles possèdent les mêmes failles de sécurité. La probabilité de corrompre un ensemble identique de nœuds revient à trouver une faille de sécurité pour l'un de ces nœuds.

Nous mettons en exergue, dans ce papier, que l'hétérogénéité des nœuds est un atout important pour accroître la résilience des RCSF. En effet, en groupant les nœuds en fonction de leurs caractéristiques logicielles ou matérielles, la difficulté de corrompre le consensus est d'autant plus élevée. Nous proposons dans ce papier une solution qui permet d'accroître la résilience du mécanisme de consensus dans les RCSF. Notre proposition repose sur des messages d'annonce propagés à deux sauts. Chaque nœud connait donc les caractéristiques des nœuds se situant à deux sauts de lui. En associant

ce mécanisme au mécanisme de découverte de voisinage à deux sauts, les nœuds peuvent connaître l'ensemble des groupes présents dans le voisinage d'un de leurs voisins. Lors de la détection d'une attaque, les votes de chaque nœud voisin d'un nœud suspect permettent de réaliser un consensus. Un nœud est détecté comme corrompu si le nombre de groupes dont le vote est positif est supérieur à la moitié du nombre de groupes total présent dans le voisinage du nœud suspect. Notre proposition permet donc de repousser la barre des 50% de nœuds corrompus nécessaires pour attaquer le système.

Les perspectives à court terme sont nombreuses. Dans un premier temps, nous souhaitons réaliser des simulations afin de valider les avantages de notre mécanisme de consensus par rapport aux mécanismes conventionnels. Les simulations permettront de mettre en évidence si la consommation énergétique, due à la surcharge d'informations ajoutées par le protocole, se voit contenue grâce à une plus grande souplesse pour les états de veille et d'éveil des nœuds. Il est facile à imaginer que notre protocole surpassera aisément les protocoles existants de consensus car la surcharge d'informations reste très minime. En effet, seules les caractéristiques sont transmises en plus. A plus long terme, il pourrait être intéressant de tester, à échelle réelle, notre mécanisme de consensus en l'intégrant à des systèmes de détection d'intrusion déjà existants.

6 Références

1. K. Holger and W. Andreas. *Protocols and Architectures for Wireless Sensor Networks*. Wiley Press, 2005.
2. M. Ilyas and I. Mahgoub. *Handbook of sensor networks: Compact wireless and wired sensing systems*. CRC Press, 2005.
3. Y. Zhou, Y. Fang and Y. Zhang. *Securing Wireless Sensor Networks: A survey*. IEE Communications Survey, vol. 10, no. 3, pp. 6-28, 2008.
4. A.-S. K. Pathan, H.-W. Lee and C. S. Hong. *Security in Wireless Sensor Networks: Issues and Challenges*. In 8th International Conference on Advanced Communication Technology (ICACT'06), pp. 1043-1048, 2006.
5. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. *A Survey on Sensor Networks*. IEEE Communication Magazine, vol. 40, no. 8, pp.102-114, 2002.
6. H. Y. Lin and T. C. Chiang. *Intrusion Detection Mechanisms Based on Queuing Theory in Remote Distribution Sensor Networks*. Advanced Materials Research, 2010.
7. I. Onat and A. Miri. *An intrusion detection system for wireless sensor networks*. IEEE Wireless and Mobile Computing, Networking and Communication (WiMob'05), wol. 3, pp. 253-259, 2005.
8. A. Abduvaliyev, A.-S. K. Pathan, Z. Jianying, R. Roman and W. Wai-Choong. *On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks*. IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1223-1237, 2013.
9. S. Bhupinder and K. Kamaljit. *Wireless Sensor Network based: Design Principles & measuring performance of IDS*. International Journal of Computer Applications, no. 28, pp. 81-85, 2010.
10. N. A. Alrajeh, S. Khan and B. Shams. *Intrusion Detection Systems in Wireless Sensor Networks: A Review*. International Journal of Distributed Sensor Networks, vol. 2013, 2013.

11. M. Estiri and A. Khademzadeh. *A Game-theoretical Model For Intrusion Detection in Wireless Sensor Networks*. 23rd Canadian Conference on Electrical and Computer Engineering (CCECE), 2010.
12. Z. Zonghua, F. Naït-Abdesselam and L. Xiaodong. *RADAR: A ReputAtion-Based Scheme for Detecting Anomalous Nodes in WiReless Mesh Networks*. IEEE Wireless Communications and Networking Conference (WCNC'08), 2008.
13. R. A. Shaikh, H. Jameel, B. J. d'Auriol, L. Sungyoung, S. Young-Jae and L. Heejo. *Trusting Anomaly and Intrusion Claims for Cooperative Distributed Intrusion Detection Schemes of Wireless Sensor Networks*. IEEE Conference for Young Computer Scientists, 2008.
14. T. H. Hai and E. N. Huh. *Optimal Selection and Activation of Intrusion Detection Agents for Wireless Sensor Networks*. Proceedings of the Future Generation Communication and Networking, 2007.
15. M. Nouri and S. A. Aghdam. *Collaborative techniques for detecting wormhole attack in MANETs*. IEEE Conference on Research and Innovation in Information Systems (ICRIIS), 2011.
16. A. H. FathiNavid and, A. B. Aghababa. *A Protocol for Intrusion Detection Based on Learning Automata in Forwarding Packets for Distributed Wireless Sensor Networks*. IEEE Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012.
17. T. Manikandan and K. B. Sathyasheela. *Detection of malicious nodes in MANETs*. IEEE Conference on Communication Control and Computing Technologies (ICCCCT), 2010.
18. N. E. D. S. Eissa and G. I. Selim. *Cooperative Intrusion Detection Technique in Wireless Sensor Networks*. International Journal of Computers & Technology, vol. 13, no. 3, 2014.
19. H. Sedjelmaci and M. Feham. *Novel hybrid intrusion detection system for clustered wireless sensor network*. Journal of Computing Research Repository, 2011.

Cyber attacks in the guided_transport domain

Christophe Gransart[1], Christian Pinedo[2], Marina Aguado[2], Marc Heddebaut[1], Eduardo Jacob[2], Igor Lopez[2], and Marivi Higuero[2]

[1] Université Lille Nord de France,
French Institute of Science and Technology for Transport,
Development and Networks (IFSTTAR)
[2] University of the Basque Country UPV/EHU
christophe.gransart@ifsttar.fr, christian.pinedo@ehu.es,
marina.aguado@ehu.es, marc.heddebaut@ifsttar.fr, eduardo.jacob@ehu.es,
igor.lopez@ehu.es, marivi.higuero@ehu.es *

Abstract. Public guided transport systems like metros, trains or buses are using more and more wireless technologies. These technologies are used for different purposes: control/command, signaling, maintenance, passenger information system, ... Public guided transport systems are sensitive to cyber attacks. The intruder does not need to be inside the network, for instance on an IP network, to disturb the network.
This paper focuses on the Denial Of Service attack using some jammers to perturbate the wireless network at the physical level.
We first present the threat characterization and in a second step, we are presenting the architecture currently developed to monitor the threats and to take counter measures using a wireless resilient architecture.

Keywords: Public guided transport system, EMC, Cyber attack, Denial Of Service
Category: Specialized paper

1 Context & Introduction

Public guided transport is the enabling factor for sustainable mobility and a key strategy to promote a significant cut in Green-House Gas emissions. Consequently, efficiency and sustainability in transport-guided operation is one of the main challenges of our society. On the one hand, users demand access to ever faster, reliable and affordable modes of public transport. On the other, governments and public institutions must ensure the economic and environmental viability of these networks.

Communication services, in general, and more specifically, signalling systems, play a crucial role in this domain. These systems are responsible for an increasing number of safety-critical operations, and therefore the development of robust and reliable communication architectures to support theses signalling systems

* The present research work is supported by the European FP 7 SECRET project. The authors gratefully acknowledge the support provided by this institution.

and communication services has drawn the attention of industry and research community.

Moreover, the recent introduction of modern communication techniques has motivated a new challenge: the cyber attack risk. Even when safe operation is guaranteed by the signalling system, a denial-of-service cyber attack caused by a jammer clearly introduces the risk of a loss of service and, consequently, degradation in operation efficiency of the guided-transport service.

Our research is focused on providing a detailed threat characterization in the transport-guided domain under the point of view of different sources of cyber attacks. We also contribute with a modern communication architecture resilient reliable and affordable to "face" denial-of-service cyber attack caused by a jammer. And that the architecture is not limited to the specific jammer attacks considered, but it would be able to include protection against future jammer attacks[1].

This article is structured as follows. First, we carry out a threat characterization including detailed description of the most common jammers available in the wireless network that may impact on communication. We currently put the scope on the basic GSM system. Next, we detail our proposal: a resilient communication architecture able to face the previously characterized denial of service cyber attack and able to face other future jammers.

2 Threat characterization

For the kind of threat that we are studying in this project, the position of the jammer is a very sensitive information. We are studying the RX level (receive level) of the wireless link. Indeed, RX level is lower than TX level (Transmit level).

So, according to a preliminary study and tests on the field, we know that the reception is easier to perturbate than the emission which has more important power level.

2.1 Jammer on wireless network

According to the power of the jammer, the RX can be lightly perturbed or definitely stopped. For instance, in the railway domain, this could have some serious consequences on train operation. Indeed, in case of jamming, since no information can be exchanged with the railway radio block centre, this could prevent trains from starting from stations if a jammer is located on a platform, preventing radio data and voice exchanges between trains and ground. Moreover, when trains are moving, losing the train-to-track radiocommunication for more than 20 seconds could initiate train emergency braking if no fresh data is received from the ground and, no updated speed instructions are received [2]. We are currently studying one main scenario: the communication link is completely cut and then the public guided transport system is not able to communicate with the control center on the ground.

Table 1 from [3] shows that a jammer can impact the GSM communication. The "Loss of communication" means that the communication is lost after applying the disturbing signal. The "No connection" means that no GSM connection at all can be established when the jammer signal is applied.

Table 1. Jammer to GSM power ratio and consequences

P_GSM (dBm)	P_JAM (dBm)	BER_f1	BER
-38	-36	Loss of comm	No connection
-38	-37	12.6	No connection
-38	-38	5.6	No connection
-38	-40	2.1	14.7
-38	-42	0.8	7.8
-38	-44	0.2	2.7
-38	-46	0.1	2.7
-38	-48	0.006	02
-38	-50	0.02	0.04

2.2 Jammer position

We have studied two cases: when the jammer is close to the mobile and in the second case when the jammer is close to the base station (BTS). For each case, we focus on the reception level.

Jammer close to the mobile In this first case, we assume that the jammer is inside the vehicle. It is able to perturbate the signals received from the base station. If no message is received by the vehicle from the ground station during a certain delay, the vehicle automatically put itself in a safe mode and is, for instance, stopped on the line. This will have an impact on the global trafic on public transport system.

Jammer close to the BTS In this second case, the jammer is close to the the base station, so the jammer is able to perturbate all the incoming communications from all the vehicles into its cellular cell. The problem is symetric to the one presented for a jammer close to the mobile, so results shown in Table 1 remain valids.

3 General Architecture to solve the problem

The SECRET project [4] aims to design a resilient communication architecture that is capable of facing electromagnetic interferences in the public guided

transport domain. The communication architecture must be geographically distributed and must support mobile devices since most sensitive communications take part between vehicles and ground system.

In the public guided transport domain there are a lot of wireless technologies, which are sensitive to electromagnetic interferences. These technologies are used for different purposes such as train control systems or voice/data communications or Passenger Information System (PIS). The variety of wireless technologies complicates the detection of electromagnetic attacks. For instance, the frequency band may be different, which may require the deployment of new sensors; the characteristics of the interfering signals may also be different forcing to the use of different signal processing techniques to detect interferences, and so on. Furthermore, depending on the characteristics of the interference, the measures to overcome it may result useless and it might be necessary to employ alternative measures to reestablish communications.

Consequently, the SECRET project proposes a highly modular architecture to allow dealing with different kind of electromagnetic attacks and even to allow managing different recovery procedures in case of detection of attacks. The purpose is to design a generic architecture that can be completed with modules to detect and recover from specific kind of interferences.

The proposed resilient architecture consists of three main kinds of components as it is summarized in the figure 1. Acquisition Systems provide information about detected electromagnetic interferences. Reactive Systems provide measures that can be used in case of need. Finally, the Detection System is the core of the architecture and so, collects information about electromagnetic interferences from Acquisition Systems and makes use of the functionality provided by Reactive Systems in order to protect communications.

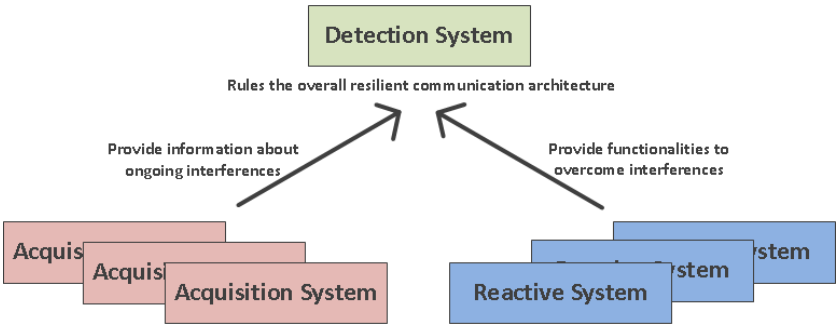


Fig. 1. Components of the resilient communication architecture.

In the following sections these three components are described in more detail.

3.1 Detection System

The Detection System is the core of the resilient architecture and the system that rules the overall resilient communication architecture. Its objective is to detect the interferences and react against them. In order to achieve this, the Detection System makes use of the other two components of the resilient communication architecture: Acquisition Systems and Reactive Systems.

The architecture of the Detection System, see figure 2, consists of multiple Health/Attack Managers deployed along the line of the public transport system and vehicles running the service, and one Central Health/Attack Manager located in the headquarters of the public transport operator. The role of Health/Attack Managers and Central Health/Attack Manager is quite different.

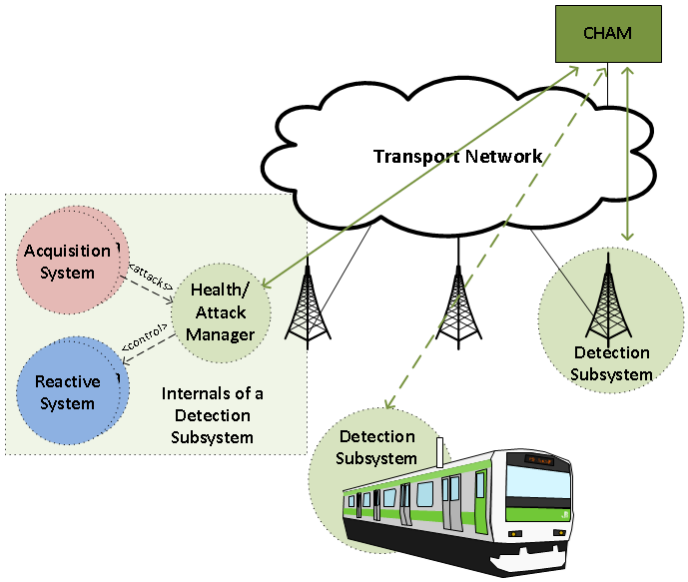


Fig. 2. Architecture of the Detection System applied on a railway domain.

Health/Attack Managers take care of the detection and reaction against electromagnetic interferences in a certain geographical area. The geographical area covered can be a moving vehicle or section of the line. Health/Attack Managers make use of locally available Acquisition Systems to detect attacks inside the area. In fact, the capabilities of attack detection depends on the functionality provided by the locally available Acquisition Systems. Secondly, the Health/-Attack Manager also uses the locally available Reactive Systems to harden the communications that may be affected by interferences. The sum of the local Health/Attack Manager and its locally available Acquisition Systems and Reactive System is called a detection subsystem.

Each detection subsystem operates autonomously by being managed by the local Health/Attack Manager. The configuration of the Health/Attack Manager consists on a series of conditions and actions to perform once matched the conditions. The typical actions to take once the attack conditions happen are the following ones: drop the detected interference, log the interference locally, report the interference to the Central Health/Attack Manager or perform a reactive action by using the functionality provided by one locally available Reactive System. This design allows each detection subsystem to react autonomously against electromagnetic interfaces even when the communication with the Central Health/-Attack Manager has been lost, which might happen.

On the other hand, the Central Health/Attack Manager connects with remote Health/Attack Managers to consolidate all the information regarding electromagnetic interferences obtained from multiple detection subsystems. Apart from this basic functionality, the Central Health/Attack Manager can perform more advanced actions such as remote configuration of Health/Attack Managers and remote notification about an ongoing interference to a Health/Attack Manager that brings closer to interfered area.

3.2 Acquisition Systems

Architectural presentation The acquisition system is composed of several components. Logical sensors are connected to the real physical sensors. The numerical signals can be from various kinds according to the Physical_sensor whom is connected. For Wi-Fi system, we catch for instance SNR. For GSM, we catch very low level information (I/Q parameters or EVM [5]). All the information get from the different sensors are pushed to the Acquisition_Sink. Then the Acquisition_Sink pushes the data to the Health_Attack_Manager that tries to infer if the mobile communication system is under an attack.

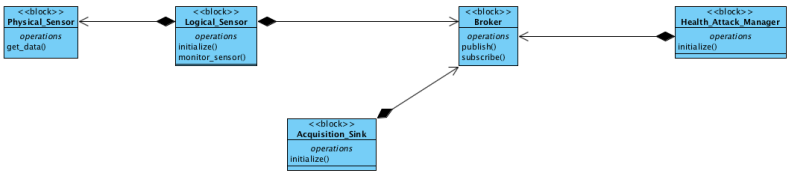


Fig. 3. Acquisition subsystem architecture.

Detection Algorithms Using our skills and experience on modeling different domains using ontologies [6] [7], we are modeling data generated from the sensors and we are defining the rules to know if a wireless network is under a jammer attack.

GSM perturbation description We are currently focusing on threat detection on the GSM wireless network. We are investigating two methods based on I/Q parameters and on a spectral analysis. We are presenting only the first one using very low level data extracted from the modem.

I/Q parameters The I/Q parameters method gets very low level data directly from the modem. The volume of data coming from the modem is around 46 Mbytes per seconds. The data are processed on a chip closed to the modem using algorithms using constellation radius and constellation standard deviation.

The figure 4 shows, in green, the standard GMSK constellation in a perturbation free environment. The black constellation is shifted from the green constellation position. Morevover, some points are closer to the center and some other are farer. The algorithms calculate the radius and the standard deviation to determine if the wireless network is inside a perturbed environment.

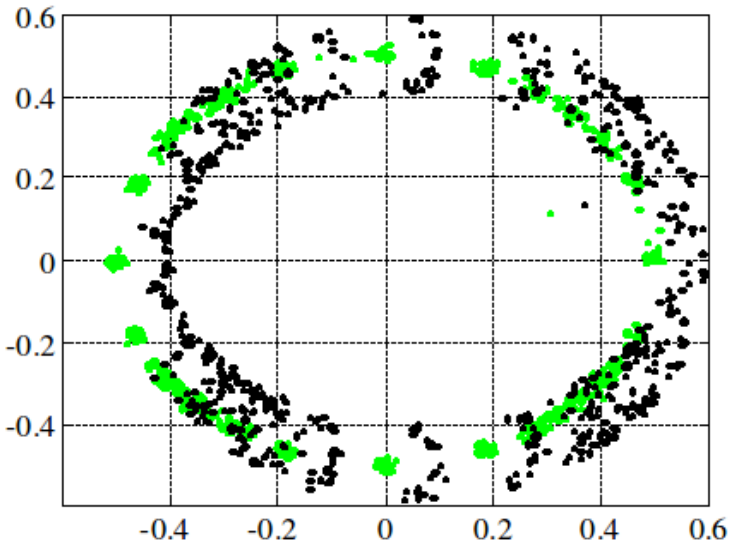


Fig. 4. GMSK constellation without perturbations (green) and under perturbations (black)

The following piece of code using the YAMI definition language (ydl)[8] shows the data that the Acquisiton_system gets from the modem. The Acquisition Subsystem is currently implemented using the Ada language with the Ravenscar profile. Ada was chosen because it is highly recommended in the railway domain to implement pieces of code that can be certified. The ontology engine with the rules is currently implemented using the Java language. Yami offers mapping for these two implementation languages.

```
package SECRET_Acquisition is
--
-- Physical position
--
    type Physical_Position is
        Latitude : Float;
        Longitude : Float;
        Time : Integer;
    end Physical_Position;

--
-- Generic Message
--
        type Current_State is
                Sensor_ID : String;
                Degree_Of_Attack : Integer;
                Detect_Delay : Float;
                Attack_Evolution : Float;
                Under_Attack : Boolean;
            Type_Of_Jammer : optional Integer; -- 5 or 6 categories
            Position : Physical_Position;
        end Current_State;

--
-- For GSM
--
    type I_Q_EVM is
        State : Current_State;
        Constellation_Radius : Float;
        Constellation_Standard_Deviation : Float;
        Constellation_Shape : String; -- two values "internal" or "external"
        EVM : Float_Array;
    end I_Q_EVM;
end SECRET_Acquisition;
```

These data are then pushed to the Healt_Attack_Manager that runs the ontology engine using rules to detect if the network is inside a perturbed wireless environment. The rules using the Protege tools are under evaluation.

Decision making The decision making is done using an ontology engine that applies rules to check the current state of the wireless link.

We chose an ontology creation tool using the Web Ontology Language (OWL), i.e. Protégé-2000[9]. Protégé-2000 was developed by Mark Musen's group at Stanford Medical Informatics [10]. In this environment, concepts are formalised as classes together with their several types of properties and the relations among them[11]. The so-called rules are created for the purpose of modeling require-

ments and certain behaviours of the system. When using the Protege tool, as in our case, the OWL is the W3C standard used to develop the ontology.

Figure 5 shows the representation of the data manipulated by the Detection System into the Protégé ontology tool.

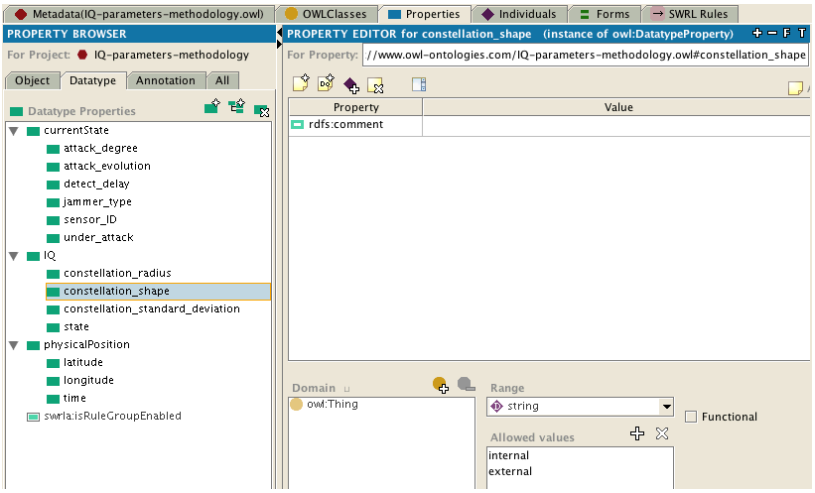


Fig. 5. Concepts represented into the ontology tool

The rules using the Protege tools are under evaluation. We are currenlty tuning the rules to find the good thresolds.

In case of a false positive detection, countermeasures could be initiated which are not appropriate. They could have for consequence a reduction of the efficiency of the railway system. For example, if an electromagnetic attack is detected but not real, limiting the speed of the train in the jammed area would reduce the overall line capacity significantly and for an extended period of time.

In case of a false negative detection, this could delay to start the requested countermeasures. Starting from what was mentioned previously in section 2.1, this could lead to emergency breaking of trains [2]. As a consequence, this will also reduce the overall line capacity quite significantly and for an extended period of time.

3.3 Reactive Systems

Reactive Systems provide local Health/Attack Managers with functionalities to try to dynamically overcome, or at least minimize, the impact of electromagnetic interferences. The simultaneously availability of several Reactive System allows the Health/Attack Manager to choose the best measure against an specific attack depending on its current configuration. Examples of a reactive system could be

a system that allows temporally increasing or decreasing the transmission power to face a sudden disturbance.

In the SECRET project we are working on the Multipath Communication System as an example of Reactive System. The Multipath Communication System provides a hardened communication path in order to protect IP communications between the mobile vehicle and ground system against electromagnetic attacks. This is achieved thanks to the availability of multiple wireless transceivers in the mobile vehicle that are managed directly by the Health/Attack Manager according to the detected attacks. The level of success of the system to face electromagnetic interferences will depend on the quantity and grade of decorrelation of the wireless transceivers available in the mobile vehicle. However, this is an implementation decision and even using a unique wireless technology could be useful to face electromagnetic attacks (i.e., employing different directional antennas) or for other purposes such as increasing resiliance against hardware or networking failures.

The Multipath Communication System, figure 6, consists of several managers deployed inside and outside detection subsystems. This architecture could be applied for trains, buses or metros. These devices manages all the outgoing communication interfaces of the detection subsystem and thanks to the use of a multipath protocol are able to use advanced traffic policies. In fact, these managers are a kind of proxies that translate traditional IP protocols used by legacy IP devices into a multipath protocol that it is used among managers. For this project we have analysed several multipath protocols and we have selected Multipath TCP (MPTCP)[12] as the basis for the multipath protocol of the system. Regarding the communication transceivers used by managers, any wireless or wired technology can be used as long as it is IP capable (feasible wireless technologies: WiFi, WiMAX, LTE, . . .).

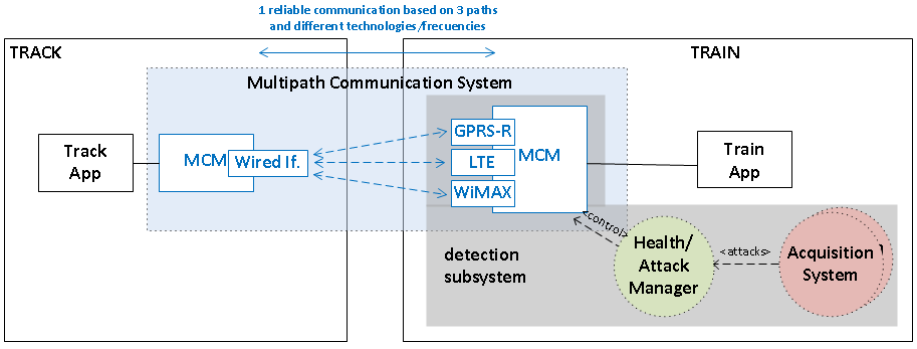


Fig. 6. Architecture of the Multipath Communication System applied on a railway domain.

However, there is a big difference among functionalities of managers. On the one hand, managers of the detection subsystems can be managed directly by the

local Health/Attack Manager. So, the Health/Attack Manager can choose the interface and traffic policy of the manager dynamically according to the current requirements. For example, if there is an attack that affects one communication transceiver, the Health/Attack Manager can force the Multipath Communication Manager to use another alternative transceiver. Furthermore, this switch between interfaces or traffic policies can be performed dynamically and without reseting ongoing connections.

On the other hand, managers outside detection subsystems only provides a termination point of the multipath protocol before the communication ends in a device that does not support directly the multipath protocol. These managers will be generally deployed in locations where a detection subsystem is not required, for example, in a data center where all the connections are wired, which are not affected by electromagnetic interferences.

4 Conclusion

Denial Of Service attack can be done at the physical level of a wireless network. None of them is currently able to protect itself against this kind of threat. Counter measures can be established to switch from one wireless communication technology to another one which is not impacted by the DOS. In this project, we have characterized different kinds of attacks and we are proposing an architecture to be able to determine if we are under attack and if we can find a fallback solution to maintain the communication between the different distributed components to maintain the service availability.

References

1. M. Heddebaut, S. Mili, D. Sodoyer, E. Jacob, M. Aguado, C. Pinedo Zamallo, I. Lopez, and V. Deniau. Towards a resilient railway communication network against electromagnetic attacks. In *TRA 2014 Transport Research Arena Conference*, Paris La Défense, April 2014.
2. REE. Le réseau GSM-R de RFF. *Revue de l'Electricité et de l'Electronique -REE*, 3(12).
3. Mili Souheir, David Sodoyer, V. Deniau, Marc Heddebaut, Henry Philippe, and Flavio Canavero. Recognition process of jamming signals superimposed on gsm-r radiocommunications. In *EMC Europe 2013*, page 6p, Bruges, Belgium, 2013.
4. WWW site Project SECRET http://www.secret-project.eu/.
5. S. Mili, V. Deniau, D. Sodoyer, and M. Heddebaut. Detection of railway signalling jamming signals using the evm method. In *AMEREM 2014 Symposium*, Albukerque, USA, 27 July – 01 Aug. 2014.
6. Olimpia Hoinaru, Georges Mariano, and Christophe Gransart. Ontology for complex railway systems application to ERTMS/ETCS system. In *FM-RAIL-BOK Workshop in SEFM'2013 11th International Conference on Software Engineering and Formal Methods*, page 6p, Espagne, January 2013.
7. Olimpia Hoinaru, Christophe Gransart, Georges Mariano, and Etienne Lemaire. An ontology for the ertms/etcs. In *Transport Research Arena 2014, Paris*, page 10p, MAY 2014.

8. Maciej Sobczak. *Programming Distributed Systems with YAMI4*. Self published book, 2013.
9. Protege website http://protege.stanford.edu/.
10. Noy N. and Musen M. The prompt suite : Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, pages 983 – 1024, 2003.
11. Natalya F. Noy and Deborah L. Mcguinness. Ontology development 101: A guide to creating your first ontology. Online, 2001.
12. A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824 (Experimental), January 2013.

Détection d'intrusion dans les systèmes industriels: Suricata et le cas de Modbus

David Diallo[12] et Mathieu Feuillet[1]

[1] Agence nationale de la sécurité des systèmes d'information
prenom.nom@ssi.gouv.fr
[2] École supérieure d'informatique, électronique, automatique

Résumé Les systèmes industriels offrent des caractéristiques très propices à la détection d'intrusion : une définition très précise du fonctionnement et une évolution lente du système d'information. Dans ce papier, nous proposons de détailler les règles de détection possibles dans un système industriel sur un exemple : le protocole Modbus. Ces règles ont été implémentées dans l'IDS Suricata et mises en œuvre sur un système réel.

Mots clés: IDS, SCADA, Modbus, Suricata, Systèmes industriels.

1 Introduction

Les systèmes industriels ou ICS [3] sont des systèmes d'information ayant pour finalité de piloter des procédés industriels au moyen de capteurs et d'actionneurs. Ils ont la caractéristique de pouvoir engendrer des actions physiques, contrairement aux systèmes d'information classiques. Les premières architectures étaient constituées de technologies spécifiques, souvent propriétaires, et étaient relativement isolées des autres systèmes d'information de l'entreprise. Aujourd'hui, notamment pour des raisons de coûts, les systèmes industriels utilisent de plus en plus des technologies développées pour des systèmes d'information classiques.

En contre-partie, les systèmes industriels deviennent vulnérables aux mêmes attaques que les systèmes d'information classiques [6]. Enfin, les systèmes industriels sont de plus en plus connectés avec les réseaux de gestion des entreprises, ce qui les exposent à des attaques à distance [7].

Les incidents se multipliant, une véritable prise de conscience sur la nécessité de sécuriser les systèmes industriels a émergé. De nombreuses initiatives apparaissent en ce sens. Ainsi, l'ANSSI a publié des guides de sensibilisation [1,2], de classification des installations [4] et de mesures à appliquer pour les sécuriser [3].

Les systèmes industriels ont des spécificités telles que les mesures usuelles de sécurité des systèmes d'information ne peuvent pas être appliquées systématiquement. Par exemple, il existe souvent des contraintes fortes de temps de réponse pour certains échanges, parfois de l'ordre de quelques dizaines de micro-secondes, qui rendent difficile l'usage de cryptographie pour assurer l'intégrité des données.

3. *Industrial Control Systems.*

De même, l'usage d'un simple équipement de filtrage peut introduire parfois une latence inacceptable.

Par ailleurs, la modification ou la mise à jour d'un système industriel ne peut pas se faire de manière aussi souple qu'un système d'information classique. Nombre de ceux-ci fonctionnent en continu et ne sont arrêtés que pour des opérations planifiées de maintenance. Ainsi, en cas d'apparition d'une vulnérabilité sur un équipement, il ne sera pas possible de mettre à jour celui-ci rapidement et le système industriel restera vulnérable.

Enfin, s'il est tout à fait envisageable d'intégrer l'ensemble des mesures de défense en profondeur préconisées dans le guide [3] sur les systèmes industriels à venir, ceci n'est pas forcément vrai pour les systèmes déjà existants. Or, la durée de vie typique d'un tel système est de plusieurs décennies et il faut trouver d'autres solutions pour atteindre un niveau de sécurité acceptable dans ces cas-là.

Pour toutes ces situations, le développement de systèmes de détection d'intrusion offre des caractéristiques très intéressantes car il permet d'augmenter les capacités de surveillance sans perturber le fonctionnement et permet de réagir rapidement en cas d'évolution de la menace.

Afin d'évaluer la pertinence d'un tel développement, une preuve de concept a été réalisée en implémentant un moteur de détection complet pour le protocole Modbus à l'intérieur de l'IDS [4] Suricata [15]. Ce préprocesseur est en cours d'intégration dans le projet. L'objectif n'est pas de détecter seulement les violations protocolaires ou les tentatives d'exploitation de vulnérabilités connues mais de proposer un jeu de règles permettant de décrire le comportement admissible du système industriel.

Les systèmes industriels disposent déjà d'une supervision, souvent appelée système SCADA [5]. Cette supervision récupère des informations au travers de requêtes aux différents éléments du système (automates, entrées/sorties) afin d'informer l'opérateur sur l'état du système industriel. Cette supervision peut être utilisée pour détecter certains comportements anormaux du procédé. Ceci est une approche complémentaire de l'IDS qui va observer le trafic sans interagir avec le système industriel. Une réflexion doit être menée sur l'articulation de ces deux systèmes de supervision.

Dans la section 2, un état de l'art succinct des différentes techniques de détection d'intrusion est proposé. Dans la section 3, des éléments de réflexion sont présentés sur les architectures de détection et sur l'articulation possible entre système SCADA et IDS. Dans la section 4, des détails sur le protocole Modbus, sont présentés ainsi que l'implémentation et des exemples d'utilisation pour détecter des évènements anormaux dans le système industriel. Dans la section 5, les résultats de tests de performance effectués sont donnés. Quelques perspectives sont présentées dans la section 6.

---

4. *Intrusion Detection System.*

5. *Supervisory Control And Data Acquisition.* Cet acronyme est souvent utilisé pour désigner les systèmes industriels dans leur ensemble.

2 Détection d'intrusion pour les systèmes industriels

La détection d'intrusion consiste à déceler toute tentative d'atteinte à l'intégrité, à la disponibilité ou éventuellement à la confidentialité du système industriel. Les systèmes de détection d'intrusion ou IDS se décomposent en deux grandes familles : les systèmes de détection sur les équipements ou HIDS [6] et les systèmes de détection sur le réseau ou NIDS [7]. Un état de l'art détaillé est présenté dans [9]. Nous nous concentrons ici sur la détection d'intrusion en analysant le trafic du réseau.

Comme expliqué dans [8], l'efficacité d'un IDS peut se mesurer à l'aide de trois métriques : le taux de faux positifs appelé *pertinence*, le taux de faux négatifs appelé *complétude* et le débit maximum que l'IDS peut analyser appelé *performance*. La logique des IDS peut de plus être classée en deux grandes catégories : l'approche comportementale qui cherche à représenter le fonctionnement normal du système d'information et à vérifier que le système s'y conforme et l'approche par scénarios qui consiste à inventorier une liste d'attaques possibles sur le système et à mettre en place des règles pour les détecter.

Aujourd'hui, la plupart des IDS commerciaux utilisent une approche par scénarios avec une base de signatures de vulnérabilités connues. L'IDS ne conserve pas d'état à proprement parler et chaque requête ou réponse est analysée de manière indépendante. Ainsi Emergency Threats [10] fournit une base de signatures d'attaques connues contre les systèmes industriels pour les IDS Snort [16] et Suricata [15]. Cette technique offre souvent des performances intéressantes et une bonne pertinence mais une mauvaise complétude, notamment du fait de son incapacité à détecter une vulnérabilité inconnue.

De par leur comportement parfaitement défini et leur évolution lente, les systèmes industriels se prêtent extrêmement bien à l'approche comportementale. Dans les articles [5] et [14], les auteurs définissent différents types de règles pour le cas du protocole Modbus. Ces règles permettent de vérifier les conformités protocolaire et comportementale du système.

Par ailleurs, des méthodes avancées proposent d'analyser l'état du système industriel et de vérifier que celui-ci ne s'approche pas d'un état dangereux (voir [9]). Cette approche peut paraitre séduisante mais relève de la surveillance du processus qui incombe au SCADA. Il convient de prendre garde à ce que l'IDS ne cherche pas à remplacer celui-ci.

En se basant sur les règles proposées dans les articles [5] et [14] complétées avec de nouvelles règles, nous avons implémenté la détection d'intrusion sur le protocole Modbus à l'intérieur de l'IDS Suricata [15]. L'objectif était d'analyser la complexité d'écriture d'un jeu de règles pour arriver à un niveau de pertinence et de complétude acceptable. Par ailleurs, la performance d'un IDS dépend fortement du nombre de règles. Des tests de performance ont été réalisés sur un cas réel. Cependant, en premier lieu, il est important de mener une réflexion

6. *Host Intrusion Detection System.*
7. *Network Intrusion Detection System.*

sur le positionnement des sondes de détection dans l'architecture du système industriel.

3 Architectures de détection

3.1 Éléments sur la menace

Afin de bien positionner les sondes, il convient d'évoquer rapidement les menaces considérées pour un système industriel tel que rencontré actuellement en pratique. Pour la suite de cette section, nous considérerons une architecture représentative telle que schématisée sur la figure 1. Sur cette dernière, on retrouve une architecture classique avec une séparation entre le réseau de supervision sur lequel se trouvent les clients SCADA, les stations d'ingénierie, les serveurs d'historique, etc. et le réseau de procédé sur lequel se trouve également le serveur SCADA ainsi que les automates et les IHM permettant de contrôler le procédé au plus près des installations.

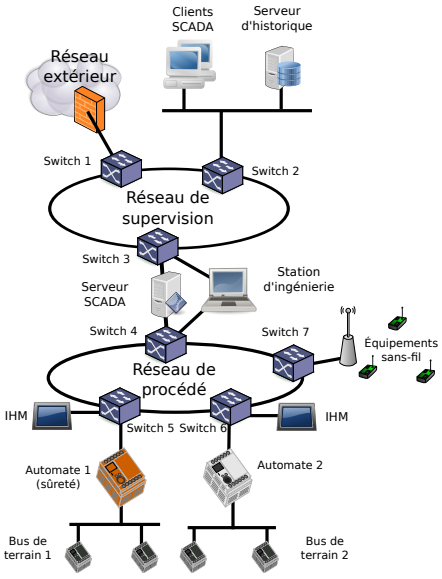


FIGURE 1: Architecture typique d'un système industriel.

Nous considérons ici que l'objectif de l'attaquant est de porter atteinte à l'intégrité ou à la disponibilité du procédé piloté et contrôlé par le système in-

dustriel. Nous supposons également que l'attaquant n'a pas accès physiquement aux bus de terrain avec les capteurs et les actionneurs [8].

Dans ces conditions, les cibles de l'attaquant seront les automates programmables industriels car, dans la plupart des systèmes industriels, ils pilotent le procédé industriel et l'on peut perturber le fonctionnement du procédé industriel

— en modifiant leur mode de fonctionnement (STOP/MARCHE) ;
— en modifiant leur programme automate ou leur *firmware* ;
— en le mettant hors service en exploitant une vulnérabilité.

Pour porter atteinte aux automates, deux pistes essentielles doivent être considérées :

— une corruption lors d'une phase de maintenance ;
— une attaque par le réseau (depuis l'extérieur, par corruption d'un équipement tiers, etc.).

Dans le cas d'une attaque par le réseau, plusieurs scénarios sont possibles. Le cas le plus probable est celui de l'attaquant arrivant par l'extérieur. Dans ce cas, il devra compromettre plusieurs équipements pour atteindre les automates : la passerelle d'interconnexion si elle existe, un équipement dans le réseau de supervision puis le serveur SCADA qui est en coupure entre le réseau de supervision et celui de procédé. Un cas un peu moins probable est celui de l'attaquant à portée du réseau sans-fil qui peut alors tenter de le compromettre. Il sera alors directement sur le réseau de procédé. Enfin, le cas le moins probable est celui de l'attaquant ayant accès physiquement au réseau de procédé, en mesure de connecter directement un équipement pirate sur celui-ci.

3.2 Les rôles de l'IDS et du système SCADA

Comme évoqué dans la section 2, les systèmes industriels disposent déjà d'un système de supervision du procédé : le système SCADA. Celui-ci collecte des informations récupérées auprès des automates, stocke un historique plus ou moins important et présente ces informations à l'opérateur pour qu'il puisse prendre des décisions. Dans certains cas, le système SCADA récupère directement les informations depuis les équipements de terrain. Il est donc en théorie tout à fait envisageable d'avoir un système SCADA dédié à la cybersécurité. L'utilisation des mêmes outils que ceux auxquels les opérateurs sont habitués permet de simplifier leur formation et accélère l'appropriation qu'ils peuvent en faire.

De par ses fonctions usuelles, un « système SCADA de cybersécurité » serait le meilleur candidat pour surveiller les déviations du procédé industriel, remonter les alarmes en cas de défaillance d'un équipement de terrain. Si l'on souhaite s'assurer du comportement normal de l'automate, il peut être envisageable pour le SCADA de vérifier la cohérence des informations collectées sur le terrain par rapport au comportement attendu de l'automate. Ainsi, une usure prématurée d'équipements due à des ordres dangereux de l'automate pourrait être repérée en analysant les informations reçues directement des capteurs au travers d'une

8. On peut noter que dans ce cas là, l'attaquant pourrait également directement porter atteinte au procédé industriel, limitant l'intérêt d'une attaque informatique

passerelle de conversion connectant le bus de terrain au réseau de procédé, par exemple. En particulier, le SCADA de cybersécurité est donc sans doute le plus adapté pour détecter une compromission d'équipement lors d'une phase de maintenance car elle ne peut être détectée que par la modification du comportement de l'équipement [9]

Cependant, le système SCADA de cybersécurité n'ayant pas accès au trafic reçu par les différents équipements présents sur le réseau mais uniquement aux informations qu'il récupère, il ne sera jamais à même de détecter un certain nombre d'attaques. Par exemple, une modification d'un programme automate ne sera pas forcément décelable si l'automate n'envoie pas de notification, une exploitation d'une vulnérabilité peut également ne pas être décelable. Enfin, ce système SCADA ne pourra jamais détecter des modifications des flux entre équipements. Par exemple, la disparition d'un flux entre deux automates ou des requêtes illégitimes depuis des machines vers des automates ne pourront pas être détectées. C'est sur ces aspects que l'IDS va pouvoir apporter des informations complémentaires.

Enfin, l'IDS et le système SCADA de cybersécurité ayant des points de vue différents sur le système industriel, cela peut permettre un recoupement des informations facilitant l'analyse d'un incident. À titre d'exemple, un système SCADA peut remonter une alarme en cas de détection d'une indisponibilité d'un automate. Celle-ci peut être inexpliquée et ne pas laisser d'informations exploitables dans les journaux d'évènements. Si en parallèle, l'IDS peut signaler du trafic illégitime, le diagnostic sera plus rapide.

3.3 Positionnement des sondes

Le placement des sondes est une question cruciale pour assurer la pertinence du système de détection dans sa globalité. Cependant, la multiplication des sondes augmente le coût, la quantité de travail nécessaire pour la configuration et la difficulté d'exploitation des alertes. Nous proposons ici quelques éléments de réflexion sur l'architecture et notamment sur les emplacements des sondes par ordre d'importance. L'architecture globale de détection est représentée sur la figure 2.

Il ressort des éléments de réflexion sur les menaces considérées qu'une sonde doit être placée au niveau de la passerelle d'interconnexion entre le système industriel et le réseau extérieur (Sonde S1 sur la figure 2). Il s'agit d'une sonde classique qui pourrait être gérée par les équipes des services informatiques généraux (par exemple les exploitants de l'informatique de gestion).

En second lieu, une sonde doit être placée de façon à analyser le trafic en provenance du système SCADA et de la station d'ingénierie et à destination des automates (Sonde S2 sur la figure 2). Dans le cas où un cloisonnement logique est effectué et où les automates n'ont pas besoin de communiquer entre eux, une telle sonde est en mesure d'analyser l'intégralité du trafic à destination

9. On pourrait également envisager un contrôle d'intégrité après chaque opération de maintenance mais peu d'équipements offrent cette possibilité aujourd'hui.

des automates. Cependant, une telle sonde ne sera pas en capacité de détecter une compromission du SCADA ou de la station d'ingénierie si l'attaquant utilise ensuite des actions légitimes pour perturber le fonctionnement du procédé industriel.

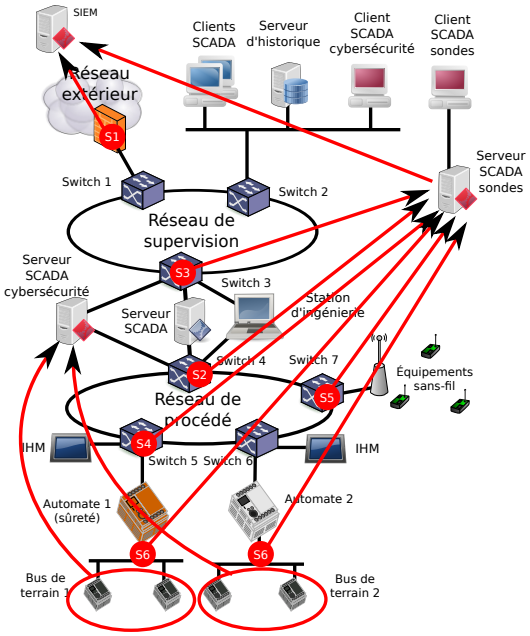


FIGURE 2: Architecture de détection.

En troisième lieu, la station d'ingénierie et le serveur SCADA étant des vecteurs naturels pour attaquer les automates, il est opportun de mettre une sonde au niveau du commutateur 3 pour détecter tout trafic anormal à destination de ces équipements (Sonde S3 sur la figure 2).

Enfin, lorsque nous sommes en présence de liens sans-fil ou d'automates particulièrement sensibles comme des automates de sûreté (systèmes instrumentés de sûreté), il est intéressant de placer une sonde au plus près des équipements pour surveiller le trafic qui y entre ou en sort (Sondes S4 et S5 sur la figure 2).

Enfin, dans le cas où certains actionneurs sont plus critiques et nécessitent une surveillance particulière lorsque l'on craint la compromission d'un équipement sur le bas de terrain, on peut mettre une sonde qui analyse le trafic sur ce bus (Sondes S6 sur la figure 2).

3.4 Traitement centralisé ou distribué

Une fois que les emplacements des sondes sont choisis, il faut réfléchir à la façon de traiter le trafic capté. Deux systèmes sont possibles. Dans le cas d'un traitement centralisé, les sondes ne sont pas des IDS à proprement parler mais effectuent un traitement minimal sur le trafic avant de tout faire remonter à un IDS centralisé qui va effectuer cette analyse. Cette architecture présente l'intérêt de permettre la corrélation entre les différents points de capture sur le réseau. Elle facilite la maintenance des sondes qui sont alors très simples. Cependant, le passage à l'échelle sur des installations de grande taille n'est pas forcément évident. En effet, les petites sondes vont générer un trafic équivalent au trafic qu'elles analysent et le réseau des sondes devra être dimensionné pour supporter l'intégralité d'un trafic généré. De manière similaire, l'IDS centralisé devra également être dimensionné pour traiter l'intégralité du trafic généré par le système industriel.

La deuxième approche possible consiste à ce que les sondes soient des IDS complets analysant le trafic localement et ne remontant que les alertes. Le trafic résultant de ce type d'analyse est beaucoup plus faible dès lors que les règles de détection sont choisies correctement. Le deuxième avantage d'un tel système est le passage à l'échelle automatique car le nombre de sondes croît en même temps que la taille de l'installation. Si un tel système ne permet pas la même souplesse que l'approche centralisée en matière de corrélation entre les différentes sources de trafic capté, elle permet cependant la corrélation entre des alertes remontées par différents IDS. En revanche, il faut s'assurer qu'une sonde avec des caractéristiques techniques similaires à celles de pare-feu ou VPN industriels d'entrée de gamme peut analyser le trafic généré par des équipements de terrain comme les automates.

Dans les deux cas, afin de ne pas perturber le système industriel surveillé, l'ensemble des sondes sont placées dans un réseau dédié. Que le système soit centralisé ou distribué, les alertes doivent être présentées à un opérateur. Les SCADA étant déjà munis d'une supervision du procédé, il parait pertinent d'utiliser le même genre d'interface en créant un superviseur Sondes pour permettre aux opérateurs de s'approprier ce système plus rapidement. Les opérateurs pourraient alors faire le rapprochement entre les évènements de sécurité remontés par le superviseur Sondes ou le système SCADA Cybersécurité (cf section 3.2) avec les informations fournies par le système SCADA du procédé.

Par exemple, l'opérateur peut observer un plantage inexpliqué d'un équipement ; le signalement de trafic non-conforme sur le système SCADA Sondes permettra un diagnostic plus rapide et accélèrera la réaction. La seule exception étant la sonde S1 dont les alertes ne correspondent pas forcément à des évènements métier et qui ne sont donc pas forcément pertinentes pour les opérateurs.

Dans la suite de ce papier, nous allons présenter la preuve de concept qui a été effectuée en se basant sur un choix d'architecture distribuée. Le protocole choisi pouvant servir à la communication entre le système SCADA et les automates ou entre les automates et leurs entrées/sorties, cette preuve de concept peut s'appliquer aux sondes S2, S4, S5 et S6 représentées sur la figure 2.

4 Intégration de Modbus dans Suricata

4.1 Le protocole Modbus

Modbus est initialement un protocole série publié par Modicon en 1979 et ses spécifications [13] sont ouvertes. Depuis 1999, ce protocole a été porté sur TCP/IP [11]. Dans la suite, nous utiliserons le terme Modbus uniquement pour parler de cette variante.

Ce protocole fonctionne sur le modèle du client (aussi appelé maître) et du serveur (aussi appelé esclave). Le client envoie des requêtes auxquelles le serveur répond. Aucune connexion ne peut être initiée par le serveur. À réception de la requête, le serveur la traite avant de renvoyer une réponse au client. La réponse donne le résultat de la requête et, le cas échéant, les données demandées.

Chaque paquet TCP contient un ADU [10] avec une structure comme représentée sur la figure 3. L'en-tête MBAP [11] contient un identifiant de transaction pour pouvoir associer une réponse à la requête correspondante, un champ `Protocole` correspondant à la version du protocole Modbus, la longueur du PDU [12] Modbus contenu dans l'ADU [13] ainsi qu'un champ pour les passerelles Modbus (appelé `Unité`).

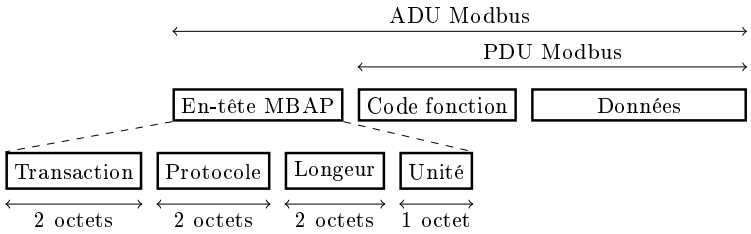


FIGURE 3: Structure d'un ADU Modbus et de l'en-tête MBAP.

Le PDU, quant à lui, contient un code identifiant une fonction et les données associées. Les fonctions sont réparties en trois catégories :

— les fonctions publiques, définies dans la spécification [13] ;
— les fonctions utilisateur, destinées à des besoins internes ;
— les fonctions réservées, utilisées par des fournisseurs pour leurs produits mais qui ne sont pas documentées.

L'essentiel des fonctions publiques ont pour objet de permettre au client de lire ou d'écrire dans des registres présents sur le serveur.

10. La norme n'interdit pas de mettre plusieurs *Application Data Unit* (ADU) dans un paquet mais le déconseille fortement.
11. *Modbus Application Protocol.*
12. *Protocol Data Unit.*
13. Pour être tout à fait exact, il s'agit de la longueur du PDU plus un octet.

À titre d'exemple, la fonction `Read Coils` permet de lire des bits (appelés *coils* en Modbus) consécutifs. Comme indiqué sur la figure 4, lors de la requête, les données sont l'adresse de départ suivie de la quantité de bits à lire. Lors de la réponse, les données associées sont le nombre d'octets renvoyés, suivi des bits demandés avec éventuellement du bourrage pour compléter le dernier octet.

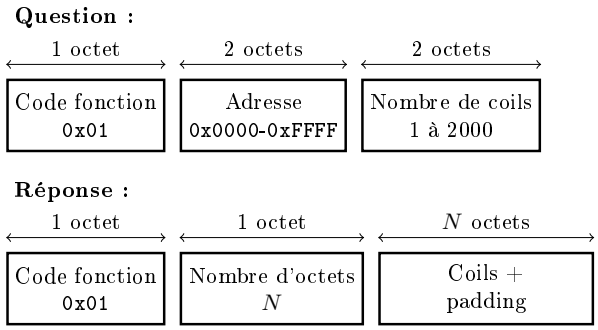


FIGURE 4: Requête et réponse de la fonction `Read Coils`.

La fonction symétrique est la fonction `Write Multiple Coils`. Comme indiqué sur la figure 5, dans la requête, le client spécifie l'adresse de départ, le nombre de bits et leurs valeurs. Dans la réponse, le serveur redonne l'adresse de départ et le nombre de bits.

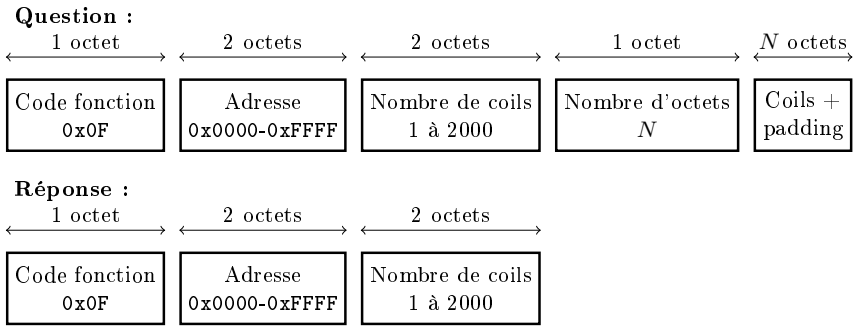


FIGURE 5: Requête et réponse de la fonction `Write Multiple Coils`.

Il existe également des fonctions de diagnostic, utiles pour la maintenance, mais qui ne permettent ni d'accéder ni de modifier le contenu des registres.

4.2 Conformité au protocole Modbus

À partir de la spécification de Modbus [13], il est possible d'établir des règles pour détecter des trames non conformes au protocole. Dans la suite, nous donnons quelques exemples pour illustrer les moyens disponibles pour vérifier la conformité des échanges à la norme.

Grâce à l'analyse de l'en-tête et à partir de la valeur des champs ou de la relation existante entre un ou plusieurs champs de la trame, il est possible de détecter un comportement anormal ou illicite.

En effet, le champ `Protocole` de la figure 3 représente la version du protocole, et conformément à la spécification [13], la version actuelle du protocole Modbus est 0. Toute valeur différente n'est donc pas possible et constitue une violation de la norme.

Le champ `Unité` est utilisé pour le routage inter-systèmes pour communiquer au travers d'une passerelle et sert à adresser des serveurs Modbus connectés en lien série. Dans le protocole Modbus sur lien série [12], les adresses comprises entre 248 et 254 ne sont pas possibles et constituent aussi une violation du protocole.

Au delà de l'en-tête, il est également possible de vérifier la cohérence entre des champs pour lesquels il existe des relations implicites ou explicites.

Par exemple, dans le cas d'une réponse à une requête `Read Coils` comme présentée sur la figure 4, le champ `Nombre d'octets` est forcément compris entre 1 et 250 car le nombre de coils est compris entre 1 et 2000 selon la norme [13]. Mais il est également possible de vérifier la relation implicite avec le champ `Longueur` de l'en-tête MBAP qui doit valoir $3 + N$ où N est la valeur du champ `Nombre d'octets` de la même trame.

Le protocole définit des échanges (appelés transactions) bidirectionnels et, à quelques exceptions près, une requête adressée à un serveur génère une réponse. Comme explicité au 4.1, aucune connexion ne peut être initiée par le serveur, donc il est possible de détecter une requête initiée par un serveur, une requête sans réponse ou une réponse orpheline à l'aide du champ `Transaction` qui doit être identique dans la requête et la réponse.

La cohérence entre la requête et la réponse peut être vérifiée plus en profondeur. En effet, le code fonction dans les deux trames doit être identique si la requête s'est exécutée correctement [14]. Dans le cas d'une transaction `Read Coils`, le champ `Nombre d'octets` de la réponse vaut $\lceil N/8 \rceil$ où N est la valeur du champ `Nombre de coils` de la requête associée.

L'ensemble de ces vérifications protocolaires ont été implémentées dans le préprocesseur Modbus pour Suricata et peuvent être utilisées de manière transparente à l'aide du jeu de règles fourni dans Suricata. Un exemple détectant les erreurs sur le champ longueur est donné dans la figure 6.

14. Dans le cas contraire, une exception doit être renvoyée d'après la spécification.

```
alert modbus any any -> any any
    (app-layer-event:modbus.invalid_length;
     msg:"Modbus invalid Length"; sid:5; rev:1;)
```

FIGURE 6: Règle de détection des longueurs invalides.

4.3 Vérification de conformité du processus industriel

Comme expliqué à la section 2, les systèmes industriels ont un comportement extrêmement bien défini qu'il est possible de décrire au niveau protocolaire afin que l'IDS puisse valider la conformité du trafic par rapport au comportement attendu. Cependant, la mise en place de règles fines pour un IDS nécessite d'avoir une cartographie qui n'a pas encore été établie pour de nombreux systèmes. Dans ces cas là, il est malgré tout possible de définir des comportements dangereux qui permettent de couvrir beaucoup de menaces. Par exemple, la règle donnée dans la figure 7 permet de détecter l'utilisation de fonctions réservées.

```
alert modbus any any -> any any
    (msg:"Diagnostic functions not allowed";
     modbus.function: reserved; sid:31; rev:1; )
```

FIGURE 7: Règle de détection des fonctions réservées.

Pour les systèmes industriels critiques, une des premières mesures préconisées par l'ANSSI est l'établissement d'une cartographie du système [4]. Elle peut être utilisée pour écrire des règles de détection de plus en plus précises. A l'aide de la matrice de flux, il est possible de distinguer les clients des serveurs et de spécifier la liste des serveurs que chaque client a le droit de contacter. À partir de là, tout échange entre deux machines non identifiées comme un couple client/serveur légitime sera anormal. Les règles associées sont données sur la figure 8.

```
alert modbus $MODBUS_CLIENT any -> !$MODBUS_SERVER
    any (msg:"Invalid Server"; sid:20; rev:1;)
alert modbus !$MODBUS_CLIENT any -> $MODBUS_SERVER
    any (msg:"Invalid Client"; sid:21; rev:1;)
```

FIGURE 8: Détection d'une communication illégitime.

Pour aller plus loin, pour chaque couple de client/serveur, il est possible de définir la liste des fonctions autorisées. Ainsi, en fonctionnement normal, l'usage de fonctions de diagnostic est considéré comme illégitime et doit générer des alertes. De même, certains clients peuvent ne pas avoir le droit d'écrire et l'usage de fonctions telles que `Write Multiple Coils` devra alors générer une alerte.

Enfin, pour chaque fonction autorisée, il est possible de spécifier les comportements autorisés. Ainsi, il est possible de définir des plages d'adresses accessibles en lecture ou en écriture. De plus, certaines valeurs peuvent avoir des bornes (par exemple, la vitesse maximum d'un moteur ou deux coils ne pouvant valoir 1 en même temps) et toute tentative d'écriture violant ces règles doit générer une alerte.

```
alert modbus any any -> $MODBUS_SERVER any
     (msg:"Holding regs write at address 8603 value >2000 not allowed";
      modbus.access: write holding, address 8603, value >2000;
      sid:52; rev:1; )
```

FIGURE 9: Détection de valeurs interdites à une adresse donnée.

5 Tests de performance

Comme expliqué à la section 2, trois critères sont utilisés pour évaluer la qualité d'un IDS pour les systèmes industriels. Parmi ceux-ci, se trouve la performance qui est la capacité de l'IDS à analyser le trafic réseau sans perte de paquet. L'objectif de cette section est de présenter des tests de performance qui ont été effectués sur Suricata.

5.1 Banc d'essai

Les tests de performances sont réalisés sur un système embarqué, Mirabox [15]. Le prix de ce boîtier et de ses accessoires est compris entre 100 et 200 euros. S'il n'est pas compatible avec un usage en milieu industriel, ses caractéristiques techniques et ses dimensions sont proches de celles des équipements industriels de sécurité tels que les pare-feu d'entrée de gamme.

Pour les tests, une Debian unstable de septembre 2014 avec un noyau Linux 3.15.8 a été installée. La version 2.0.3 de Suricata a été utilisée avec les développements du préprocesseur Modbus réalisés.

Le même trafic a été utilisé pour l'ensemble des tests et rejoué à différents débits en fonction des besoins. Il a été généré grâce à la machine virtuelle *Target Service* [16], de la société Digital Bond, qui simule un serveur Modbus et un client réalisé avec l'extension modLib [17] de Scapy. Il est composé de requêtes et réponses de la fonction `Write Single Register`.

Afin d'évaluer les performances de Suricata, différents scénarios ont été joués avec les règles suivantes :

15. `https://www.globalscaletechnologies.com`.
16. `http://www.digitalbond.com/tools/scada-honeynet/`.
17. `https://www.scadaforce.com/modLib.py`.

— une règle pour détecter l'usage de la fonction `Write Single Register`, générant ainsi une alerte pour chaque transaction, en effectuant une recherche de motif sans utiliser le préprocesseur Modbus ;
— une règle similaire à la précédente en utilisant le préprocesseur Modbus ;
— les règles permettant d'activer l'ensemble des vérifications de conformité au protocole Modbus ;
— les règles permettant d'activer l'ensemble des vérifications de conformité au protocole Modbus et cent règles identiques utilisant le préprocesseur Modbus mais ne générant pas d'alerte.

L'utilisation de cent règles identiques permet d'évaluer l'influence de leur nombre sur les performances de Suricata car il n'optimise pas les règles redondantes ; toute règle déclarée dans le fichier est évaluée.

5.2 Résultats

La première expérience vise à évaluer l'influence de l'utilisation du préprocesseur Modbus sur les performances. À cette fin, trois tests sont menés.

Le premier test consiste à utiliser la version officielle de Suricata sans le préprocesseur Modbus. La règle de détection de `Write Single register` par recherche de motif est utilisée. La courbe bleue de la figure 10a montre que l'équipement est capable d'analyser plus de 35 000 paquets par seconde sans perte.

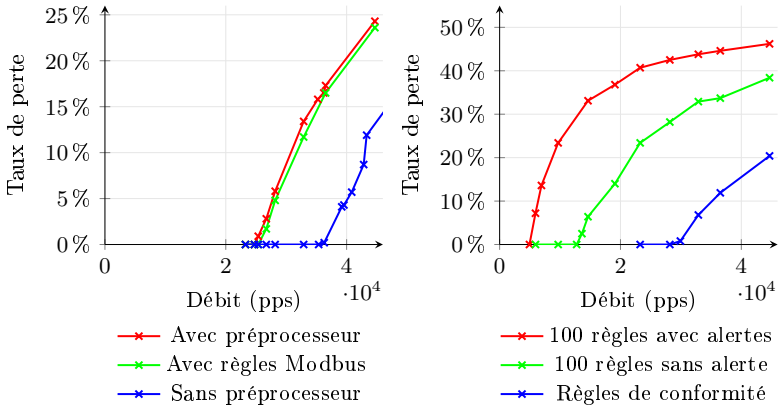
Pour le second test, le préprocesseur Modbus est présent mais la même règle de détection a été utilisée. La courbe rouge de la même figure montre que la capacité de traitement chute alors à 25 000 paquets par seconde soit environ 30 % de moins. Cette différence s'explique par le fait que, même si aucune règle Modbus n'a été spécifiée, le préprocesseur analyse les paquets transmis sur le port de Modbus.

Enfin, le dernier test vise à évaluer Suricata avec le préprocesseur et la règle équivalente utilisant le préprocesseur Modbus. La courbe verte montre que les résultats sont légèrement meilleurs que dans le test précédent. L'évaluation d'une règle avec les mots-clés Modbus est légèrement moins coûteuse que la règle équivalente avec les options génériques.

On peut noter que la génération d'une alerte pour chaque requête grève fortement les performances et que dans un cas d'usage plus réaliste avec peu d'alertes, la capacité de traitement est bien meilleure comme le montre l'expérience suivante qui cherche à caréctériser l'influence du nombre d'alertes générées et du nombre de règles.

Le premier test est effectué avec le préprocesseur Modbus et les règles de vérification de la conformité protocolaire. Le trafic étant conforme, aucune alerte n'est générée pendant ce test. La courbe bleue sur la figure 10b montre que la capacité de traitement dans ce cas est légèrement inférieure à 30 000 paquets par seconde.

Pour le deuxième test, cent règles sont ajoutées en plus des règles de conformité mais aucune alerte n'est générée. La capacité de traitement chute alors à environ 13 000 paquets par seconde.



(a) Performance du préprocesseur.

(b) Performance des règles.

FIGURE 10: Évaluation de performances du préprocesseur Modbus.

Enfin, pour le dernier test, les cent règles sont remplacées par cent autres qui génèrent toutes une alerte pour chaque transaction. La capacité de traitement chute alors à 5 000 paquets par seconde. Ce scénario est un cas très défavorable car la situation où chaque transaction génère une alerte pour chaque règle est très irréaliste.

Si l'on imagine un scénario avec de la sûreté de fonctionnement, les échanges typiques nécessitent une transaction toutes les 4 ms ; ce qui correspond à un débit de 500 paquets par seconde. Dans le cas du pire scénario, ce boîtier avec Suricata est dimensionné pour, au plus, une dizaine de couples d'équipements. Pour le cas d'un scénario plus réaliste avec une centaine de règles et peu d'alertes, ce boîtier est alors en capacité de surveiller plus d'une vingtaine de couples d'équipements. Ceci valide la faisabilité d'un tel système avec des équipements peu onéreux.

6 Conclusion

Dans cet article, nous avons pu explorer les possibilités offertes par la détection d'intrusion dans les systèmes industriels. Les caractéristiques de ces derniers font que cette mesure de sécurité peut donner d'excellents résultats alors que le caractère non-intrusif des IDS facilite leur déploiement et leur usage.

En guise de preuve de concept, un préprocesseur complet pour le protocole Modbus a été développé au sein de l'IDS Suricata. Son intégration prochaine dans le projet le rendra accessible au plus grand nombre pour une expérimentation voire un usage opérationnel. Les tests de performance menés montrent la faisabilité de sondes de détection avec du matériel bon marché.

Cependant, avant d'arriver à un produit pleinement opérationnel, de nombreux travaux restent à faire. Tout d'abord, Modbus ne représente qu'une frac-

tion des protocoles utilisés sur ce marché et il est indispensable d'effectuer le même travail pour d'autres tels que Profinet ou EtherNet/IP. Cela devra également être fait pour des protocoles propriétaires tels que S7 ou UMAS qui sont utilisés pour mettre à jour les *firmwares* et les programmes des automates.

Afin, d'avoir la supervision la plus précise possible, il est nécessaire d'avoir une cartographie très détaillée du système industriel supervisé. Pour faciliter cette tâche, il parait indispensable de développer des outils assistant les experts dès la phase de conception du système et tout au long de son existence. Si la cartographie est suffisamment précise, un tel outil peut également générer automatiquement les règles de détection des IDS.

Références

1. ANSSI : La cybersécurité des systèmes industriels : Cas d'étude (juin 2012)
2. ANSSI : La cybersécurité des systèmes industriels : Maitriser la SSI pour les systèmes industriels (juin 2012)
3. ANSSI : La cybersécurité des systèmes industriels : Mesures détaillées (janvier 2014)
4. ANSSI : La cybersécurité des systèmes industriels : Méthode de classification et mesures principales (janvier 2014)
5. Cheung, S. et al : Using model-based intrusion detection for SCADA network. In : Proc. SCADA Security Symposium. pp. 1–12 (2007)
6. Clusif : Enjeux de sécurité des infrastructures scada. `http://www.clusif.asso.fr/fr/production/ouvrages/pdf/CLUSIF-SCADA-Intro.pdf` (avril 2008)
7. Clusif : Cybercrime : évolution des cibles. `http://www.clusif.asso.fr/fr/production/ouvrages/pdf/CLUSIF-20120126-Cybercrime-Nouvelles-Cibles-UCL.pdf` (janvier 2012)
8. Debar, H., Marc, D., Andreas, W. : A revised taxonomy for intrusion detection systems. Annals of Telecommunications 55, 361–378 (2000)
9. Demongeot, T. : Détection d'intrusion pour les systèmes industriels. In : C&ESAR (2013)
10. Emergency Threats : Emergency threats SCADA ICS. `http://www.emergingthreats.net/2011/08/29/energysec-and-the-oisf-announce-new-scada-research/` (Consulté en mai 2014)
11. Modbus Foundation : MODBUS messaging on TCP/IP implementation guide v1.0b (octobre 2006)
12. Modbus Foundation : MODBUS over serial line specifiations and implementation guide v1.02 (décembre 2006)
13. Modbus Foundation : MODBUS application protocol specification v1.1b3 (avril 2012)
14. Morris, Thomas et al. : Deterministic intrusion detection rules for modbus protocols. In : 46th Hawaii International Conference on System Science. pp. 1773–1781 (2013)
15. OISF : Suricata, open source IDS. `www.suricata-ids.org` (Consulté en mai 2014)
16. Sourcefire : Snort, open source IDS. `www.snort.org` (Consulté en mai 2014)

Architecture système sécurisée de sonde IDS réseau

Pierre Chifflier and Arnaud Fontaine

Agence Nationale de la Sécurité des Systèmes d'Information
{pierre.chifflier,arnaud.fontaine}@ssi.gouv.fr

Résumé Les systèmes de détection d'intrusion réseau (NIDS [1]) sont largement utilisés pour effectuer la supervision de sécurité. Ces systèmes sont cependant eux-mêmes peu sécurisés, alors qu'ils sont par nature très exposés aux attaques.

Nous proposons une architecture de référence pour les sondes hébergeant des logiciels IDS [2]. Cette architecture renforce la sécurité des logiciels IDS et permet de limiter les conséquences d'une attaque sur la sonde. Elle permet de définir des nouveaux flux d'information pour traiter les alertes collectées.

Un prototype d'implémentation, adapté à la détection sur des réseaux industriels, est réalisé pour démontrer la faisabilité et analyser l'efficacité des protections et leur impact sur les performances.

Keywords : architecture, IDS, sécurité, réseau, détection

1 Introduction

En complément des mesures de protection appliquées sur un réseau, il est classique de déployer un système de supervision. Ce système a pour but d'observer en temps réel les communications afin de détecter des tentatives de compromission des équipements connectés au réseau supervisé, mais également les compromissions réussies de ces équipements.

De part leur fonction et leur exposition, les sondes IDS réseau constituent une cible potentielle pour les attaquants : elles embarquent une grande quantité de décodeurs protocolaires, sont présentes sur la plupart des réseaux, et perçoivent l'ensemble du trafic.

Les décodeurs protocolaires au cœur des logiciels IDS ont un rôle très sensible en sécurité car leurs implémentations sont fréquemment sujettes à l'introduction de vulnérabilités [2,3]. En effet, les protocoles à décoder sont d'une part de plus en plus complexes, et d'autre part les spécifications qui les accompagnent sont parfois incomplètes ou erronées, lorsqu'elles existent. Le plus souvent écrits dans des langages bas niveaux tels que le C pour des raisons de performances, ces décodeurs sont alors potentiellement vulnérables à de nombreuses classes de problèmes : débordements mémoire, injection de code, corruption mémoire, double

1. *Network Intrusion Detection System*
2. *Intrusion Detection System*

libération, *etc.* Bien que ces problèmes soient connus et récurrents, aucun changement notable n'est constaté sur le choix des langages [10] employés par les logiciels IDS.

En ciblant directement les sondes, un attaquant peut rendre aveugle la supervision de sécurité d'un réseau. Au sein d'un même système de supervision, les sondes sont généralement identiques : une vulnérabilité exploitable sur l'une d'entre elles est *de facto* reproductible sur toutes les autres, ce qui peut conduire à l'indisponibilité (ou la compromission) de l'ensemble du système de supervision de sécurité.

Lorsque le logiciel IDS d'une sonde s'exécute avec des privilèges importants, un attaquant qui réussit à exploiter une vulnérabilité avec succès peut prendre le contrôle de la sonde. Si les sondes sont inter-connectées et gérées de manière centralisées, l'attaquant peut alors remonter de la sonde vers la machine d'administration, puis rebondir vers d'autres machines. Le réseau de supervision de sécurité se retrouve alors dans la situation singulière où il risque de devenir le réseau permettant de compromettre l'ensemble des machines supervisées.

Enfin, les alertes correspondant aux événements détectés sont souvent remontées sur le même réseau que le réseau surveillé. La confidentialité et l'intégrité des événements ne sont donc pas toujours assurées.

Face à ces constatations, il nous est apparu essentiel d'assurer la sécurité des éléments de détection réseau, condition indispensable pour avoir confiance dans le système de supervision de sécurité. L'objectif de cet article est de proposer une architecture de sonde IDS réseau, en s'appuyant sur un matériel muni de fonctions de sécurité, permettant d'en améliorer la sécurité intrinsèque, indépendamment du logiciel IDS utilisé. En effet, la modification du logiciel IDS peut s'avérer complexe à réaliser, en particulier sans introduire de vulnérabilité, mais rend surtout difficile sa maintenance en conditions opérationnelle et de sécurité.

Cet article comporte trois parties. Nous détaillons tout d'abord dans la section 2 les bases systèmes nécessaires à la conception d'une architecture système sécurisée de sonde IDS réseau. Dans la section 3, nous expliquons ensuite comment ces protections permettent de définir une architecture robuste, basée sur l'isolation des rôles et sur des flux d'informations unidirectionnels. Enfin, nous expliquons dans la section 4 comment nous avons conçu un prototype de sonde légère mettant en œuvre cette architecture et quelques résultats expérimentaux permettant de valider cette architecture.

2 Protections système

La fonction principale d'un IDS étant d'analyser du trafic réseau, on peut considérer que tout le trafic qu'il perçoit est potentiellement malveillant. L'objectif des protections présentées dans cette section est de s'assurer que l'attaquant ne puisse pas désactiver la fonction IDS, corrompre le journal d'alertes, prendre le contrôle de la machine hébergeant la fonction IDS, voire remonter dans le réseau de supervision. Il ne doit pas non plus pouvoir faire fuir de l'information depuis l'IDS, par exemple les règles de l'IDS qui pourraient être sensibles.

L'idée générale est d'appliquer au maximum les principes de défense en profondeur et de moindre privilèges. Les protections apportées peuvent être classées en deux catégories : celles qui renforcent la sécurité des logiciels s'exécutant sur la sonde présentées dans la section 2.1, et celles qui renforcent l'isolation des composants du système présentées dans la section 2.2. La section 2.3 présente les éléments matériels nécessaires au bon fonctionnement de l'architecture proposée, notamment au regard de l'impact sur les performances.

2.1 Système d'exploitation

Afin de protéger n'importe quel logiciel IDS qui s'exécute sur la sonde, la sécurité de l'architecture proposée s'articule autour de deux axes : la configuration du système, et le durcissement du noyau.

Le travail initial consiste à appliquer les principes essentiels de sécurisation [5] : réduire la surface d'attaque, supprimer les applications et services inutiles, réduire les permissions et les privilèges des processus importants, et recompiler les applications et services avec les options de durcissement de `gcc`. Pour empêcher la persistance d'une attaque réussie, et assurer que chaque démarrage se fait dans un état « propre », l'environnement d'exécution des processus est figé : toute modification (configuration, fichiers exécutés ou installés, accès aux périphériques) est interdite en montant toutes les partitions en lecture seule, à l'exception de celle contenant des données variables (telles que les journaux) qui doivent persister entre les redémarrages.

Bien qu'essentiels, ces éléments de configuration ne sont pas suffisants. Des modifications plus profondes s'avère en effet nécessaires, comme l'application des *patches* PaX et grsecurity [7] au noyau. Sans entrer dans le détail des bénéfices apportés par ces modifications [4], il convient toutefois de citer les propriétés les plus pertinentes dans le cadre de l'élaboration d'une sonde durcie :

- espace d'adressage aléatoire (ASLR [3]) ;
- vérification de débordement lors de copies de données ;
- interdiction d'exécuter du code situé en espace utilisateur depuis l'espace noyau ;
- restrictions sur les programmes utilisateurs : suppression de la visibilité des programmes des autres utilisateurs, interdiction de créer des pages en écriture et exécution, allocations de mémoire forcée à des adresses aléatoires, interdiction du *debugging* d'un processus ;
- protections additionnelles des conteneurs : réduction systématique des *capabilities*, restrictions additionnelles appliquées aux environnements confinés (*chroots*) ;
- montage définitif d'une partition en lecture seule (sans possibilité de refaire un montage avec écriture) ;
- *Trusted Path Execution*, qui permet de s'assurer qu'un utilisateur ne peut pas exécuter de programmes situés dans un répertoire qui n'appartient pas à *root*.

3. *Address Space Layout Randomization*

L'application de ces *patches* est complétée par une configuration minimale du noyau, qui supprime les éléments inutiles (USB, accélération graphique, *etc.*) voire dangereux (chargement des modules, accès direct à la mémoire, *etc.*).

Il est important de noter que ces mesures ne modifient pas le comportement de l'IDS, ou ses vulnérabilités intrinsèques. Si les techniques classiques telles que l'exécution sur la pile seront détectées et bloquées, certaines autres restent applicables, comme le ROP [4] [12] par exemple.

Afin de diminuer autant que possible les conséquences d'une compromission, nous proposons d'utiliser, en plus des protections décrites ci-dessus, un mécanisme de cloisonnement.

2.2 Cloisonnement

Le cloisonnement permet d'ajouter une isolation supplémentaire entre les différentes fonctions à accomplir dans la sonde, et ainsi de maîtriser plus finement les communications entre ces différents composants logiciels. De cette manière, chaque processus isolé ne voit que l'environnement matériel et logiciel qui lui est strictement nécessaire, et voit ses possibilités de modification du système réduites.

Le cloisonnement est utilisé pour créer un flux de traitement des données, et des séparations entre les rôles des utilisateurs et des processus. Ces aspects seront décrits en détail dans la section 3.

Il existe différentes méthodes pour mettre ce mécanisme en œuvre : la virtualisation, ou le contrôle d'accès obligatoire.

La virtualisation permet de créer une isolation, soit par l'utilisation de fonctions matérielles (virtualisation assistée), soit logicielles (traduction d'instructions ou interprétation). Elle peut être complète, par la virtualisation d'un systèmes d'exploitation et des processus (Qemu ou Xen par exemple), ou légère, en partageant le noyau mais en créant des conteneurs pour isoler des processus. Parmi les mécanismes de virtualisation légère, on trouve trouve notamment LXC [5], VServer, OpenVZ, les « jails » de FreeBSD et les « zones » de Solaris.

Le contrôle d'accès obligatoire, ou MAC [6], est une autre possibilité pour restreindre et isoler chaque processus. En utilisant un mécanisme tel que SELinux [6], RBAC [7] (intégré à grsecurity), ou encore AppArmor [1], il est possible de contrôler finement les actions de chaque processus, et d'appliquer une politique de sécurité dédiée. Les processus ainsi isolés ont une vision limitée du système, et ne peuvent pas appeler des méthodes ou du code qui n'ont pas été explicitement autorisés.

Chacun de ces mécanismes permet d'aboutir au même résultat fonctionnel. Les différences se font surtout sur les performances, la simplicité, et sur les ressources nécessaires : la virtualisation matérielle demande plus de ressources que

4. *Return Oriented Programming*
5. *LinuX Containers*
6. *Mandatory Access Control*
7. *Role Based Access Control*

la virtualisation légère et les MAC, mais permet d'isoler les noyaux du socle et des conteneurs. En contrepartie, la virtualisation légère et les MAC s'avèrent globalement plus performants que la virtualisation matérielle.

2.3 Impact sur le matériel

L'attaquant pouvant manipuler les données du réseau surveillé, il apparaît comme évident de disposer au minimum de deux réseaux physiques distincts : l'un dédié à l'administration et à la remontée d'alertes, l'autre dédié à l'acquisition du trafic du réseau surveillé. Un troisième réseau serait idéal pour pouvoir séparer l'administration et la remontée d'alertes, mais en pratique, ces deux réseaux peuvent coexister sur un même réseau physique si ceux-ci sont isolés de manière logique, par exemple, par des VLAN ou des connexions chiffrées différentes. Deux interfaces réseau au moins sont donc nécessaires.

Certaines plates-formes apportent des fonctions matérielles qui peuvent être exploitées pour améliorer la sécurité du système d'exploitation et des applications exécutées. C'est le cas par exemple du bit NX [8], appelé XN sur ARM, qui permet de s'assurer qu'une page mémoire ne peut pas être projetée avec à la fois les droits d'écriture et d'exécution.

D'autres éléments peuvent être ajoutés, tels qu'une source d'entropie matérielle, élément utile pour l'utilisation de la cryptographie, ou encore un TPM [9], qui permettra d'assurer l'intégrité du code exécuté sur la plate-forme depuis le démarrage. La fonction de génération d'aléa d'un TPM peut également être avantageusement utilisée pour fournir une source d'aléa supplémentaire au système d'exploitation.

3 Flux d'information entre conteneurs

Par défaut, un IDS réseau est un processus qui demande des privilèges élevés pour pouvoir lire tous les paquets depuis le réseau. Sur un système classique, ce processus s'exécute avec les mêmes privilèges que l'administrateur du système, ce qui n'est pas souhaitable.

Nous définissons des rôles distincts :

— l'*administrateur du système* met à jour les logiciels ;

— l'*administrateur de l'IDS* met à jour les règles de l'IDS ;

— l'*auditeur* accède aux alertes remontées par l'IDS.

Nous nous appuyons sur l'architecture présentée précédemment pour créer des conteneurs correspondants à chacun de ces rôles. Chaque conteneur n'a qu'une vue en lecture seule sur le système de fichiers, à l'exception des répertoires où il doit pouvoir écrire des données. Ces répertoires peuvent être utilisés pour transférer des fichiers entre un conteneur et le socle (ou un autre conteneur) : le conteneur source écrit le fichier, et le destinataire surveille le répertoire (périodiquement, ou par notifications) pour pouvoir traiter le fichier.

8. *No eXecute*
9. *Trusted Platform Module*

3.1 Détection

Nous décrivons ici le cas où le socle doit héberger soit plusieurs logiciels IDS tels que Suricata [11], Bro [8] ou encore Snort [13], soit des instances différentes du même IDS, avec des paramétrages distincts.

L'architecture proposée est décrite dans la figure 1. Chaque instance d'IDS est isolée dans un conteneur dédié. La sécurité de ces conteneurs repose sur les mécanismes suivants :

1. tous les systèmes de fichiers des conteneurs sont montés en lecture seule. Si le conteneur a besoin d'écrire des données temporaires, un point de montage en mémoire uniquement (*tmpfs*) est utilisé (contenu non exécutable, effacé à chaque démarrage du conteneur) ;
2. les conteneurs n'ont accès à aucun périphérique de l'hôte, en particulier les cartes réseau ou l'affichage ;
3. les protections systèmes décrites dans la section 2 doivent également être appliquées dans les conteneurs ;
4. les données de configuration et les règles des IDS sont exposées depuis le socle par un mécanisme de type montage par recouvrement (*bind-mount*), qui s'assure que le conteneur en charge d'exécuter un IDS ne peut en aucun cas écrire dans ces données ;
5. les protections apportées sont complémentaires des mesures qui peuvent être proposées dans un IDS, telles que la réduction de privilèges (ou de *capabilities*), l'utilisation d'un utilisateur non privilégié, ou encore la limitation d'utilisation de ressources système.

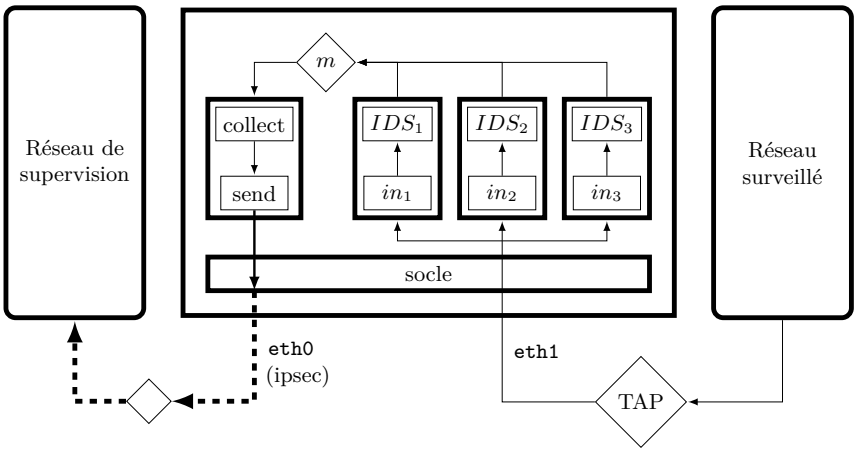


Figure 1. Architecture du système de détection.

Pour pouvoir être perçues depuis chaque conteneur qui héberge une instance de logiciel IDS, les données du réseau doivent être collectées par le socle, puis copiées dans chacun des conteneurs i, en direction de leurs mécanismes d'acquisition respectifs in_i. Plusieurs options sont possibles, en fonction du type de cloisonnement, et des capacités d'acquisition de chaque IDS_i : *socket unix* exposée *via* un autre *bind-mount*, connexion réseau *via* interface virtuelle privée, utilisation d'une file *NFLOG*, répertoire de données, *etc.*

Ce mécanisme doit respecter, au sens fonctionnel, les propriétés d'une diode : les communications doivent être unidirectionnelles depuis le socle vers le(s) conteneur(s), et ne doivent pas exposer d'élément du socle qui pourrait être compromis depuis ce(s) conteneur(s).

L'impossibilité d'envoyer des données sur le réseau depuis les conteneurs permet de limiter les éventuelles fuites d'information mais aussi la compromission d'autres équipements connectés au réseau surveillé depuis la sonde elle-même.

Après le traitement des données réseau, les alertes ou informations émises par chaque IDS_i doivent être collectées/agrégées avant de pouvoir être remontées vers un point de collecte sur le réseau de supervision. Pour cela, on propose également la mise en place d'une « diode » exposée par le socle dans chaque conteneur hébergeant un IDS, afin qu'il puisse y écrire ses journaux d'alertes et d'informations. Généralement, un fichier de journalisation pour chaque conteneur i suffit, ce qui présente l'avantage de pouvoir utiliser les attributs de fichiers (commande `chattr`) pour le marquer en mode ajout uniquement, ce qui garantit l'interdiction de modification ou de suppression.

Le socle doit également disposer d'un mécanisme de collecte m, dont le but est de collecter et d'agréger les informations remontées par tous les IDS. Ces données seront ensuite envoyées à un autre conteneur, dont le but est de gérer la remontée d'alertes. Cette transmission doit impliquer le moins de manipulations possibles pour s'assurer que le socle ne puisse pas être compromis ; idéalement, une simple recopie est donc préférable.

3.2 Remontée d'alertes

Une fois les informations collectées par le socle, elles doivent être mises en forme et envoyées vers le réseau de supervision de sécurité. La mise en forme implique la manipulation des données, c'est pourquoi elle est isolée dans un conteneur spécifique soumis aux mêmes mécanismes de durcissement que les conteneurs hébergeant les IDS.

L'exfiltration des événements depuis la sonde doit respecter des contraintes de confidentialité et d'intégrité. Il faut donc s'assurer qu'aucune donnée ne puisse être envoyée en clair, ou être modifiée pendant l'envoi sur le réseau de supervision. Pour cela, il est nécessaire de chiffrer puis signer les données à transmettre. La solution la plus efficace pour transmettre en temps réel les alertes est de mettre en place un réseau privé virtuel (VPN), par exemple de type *IPsec*, entre le socle et le réseau de supervision. Une communication doit également être mise en place entre le conteneur de collecte et le socle. Pour cela, un moyen simple est d'utiliser une interface privée virtuelle entre les deux. Le conteneur envoie

simplement ses données vers le réseau de supervision sans avoir à se soucier de leur protection ; la politique de routage du socle s'assure alors que toute donnée sortante est protégée.

La confidentialité et l'intégrité des données sont assurées par le VPN. La séparation en conteneurs permet de faire en sorte que seul le socle ait accès aux éléments secrets nécessaires à l'établissement du VPN, garantissant ainsi la protection des éléments secrets nécessaires l'établissement du VPN.

3.3 Mises à jour

La mise à jour des logiciels est un processus crucial pour le maintien en condition de sécurité. Cependant, comme le système (socle et conteneurs) n'est accessible qu'en lecture seule, il est impossible de modifier les fichiers. Nous proposons un mécanisme de démarrage spécifique qui permet de résoudre ce problème.

Le mécanisme de mise à jour sécurisée est le suivant : l'administrateur du système dépose les paquets de mise à jour dans le répertoire qui lui est accessible dans son conteneur dédié. Le socle copie ces fichiers dans une zone qui lui est réservée, soit par une tâche planifiée, soit par une action déclenchée. Il faut ensuite attendre le prochain redémarrage.

Le plus tôt possible au démarrage, avant que les différents conteneurs ne soient démarrés, et que les interfaces réseau ne soient configurées et opérationnelles, il existe une période pendant laquelle le système peut sans risque ne pas être verrouillé en écriture. Durant cette période, le système vérifie si il existe des paquets à mettre à jour dans la zone réservée. Si c'est le cas, il vérifie leur signature cryptographique par rapport à une chaîne de confiance. Si l'ensemble des paquets présents sont correctement authentifiés, alors les mises à jour sont appliquées. Lorsque les mises à jour ont été appliquées, le démarrage « normal » du système peut reprendre : les systèmes de fichiers sont verrouillés en écriture (de manière irréversible, *cf* section 2.1), les autres actions de durcissement sont mises en place, les interfaces réseau sont configurées et rendues opérationnelles, et enfin les différents conteneurs peuvent être instanciés.

4 Preuve de concept d'une sonde IDS légère

Généralement, les besoins de surveillance réseau peuvent être classés en deux grandes catégories :

— les réseaux à fort trafic, de type bureautique par exemple, pour lesquels les sondes déployées doivent disposer de ressources conséquentes, mais pour lesquelles le déploiement se fait sans trop de contraintes ;
— les réseaux répartis, qui nécessitent de multiples points d'analyse et disposent de fortes contraintes sur l'encombrement (impossible de déployer une baie de serveurs), la consommation électrique, et le nombre important d'éléments à déployer. C'est le cas des réseaux de systèmes industriels.

Bien que les principes abordés dans les sections précédentes sont valables dans ces deux cas de figure, nous avons choisi de réaliser une sonde IDS réseau légère dont le but est de surveiller des communications au sein de réseaux industriels. L'objectif est de réaliser un prototype fonctionnel, en implémentant l'architecture présentée, sur du matériel léger, peu encombrant, et peu coûteux.

4.1 Choix matériel et logiciel

Nous avons choisi un matériel de type embarqué, pour lequel plusieurs solutions à base de processeur ARM sont disponibles sous forme de set-top box. Ces solutions conviennent en terme d'encombrement, et peuvent être facilement déployées. De plus, elles ont l'avantage de pouvoir n'utiliser aucun élément mécanique (disque dur ou ventilateur), ce qui augmente considérablement leur durée de vie et diminue leur consommation.

De nombreuses plates-formes de ce type existent sur le marché, telles que RaspberryPi, GuruPlug, cartes FreeScale (SabreLite, Nitrogen, ...) ou encore MiraBox. Étant données les contraintes exposées dans la section 2.3, nous avons retenu la MiraBox qui dispose d'un CPU ARMv7 cadencé à 1,2 GHz, d'un Go de RAM et de deux interfaces réseau gigabit.

Pour le socle, une Debian *unstable* avec un noyau Linux 3.16.3 modifié pour intégrer grsecurity a été installée. Pour cloisonner les conteneurs, nous avons retenu la virtualisation légère LXC intégrée au noyau Linux. Chaque conteneur est basé sur une Debian *unstable* réduite à son strict minimum.

Un seul conteneur IDS est instancié, exécutant la version 2.0.3 de Suricata [11], modifié pour utiliser les règles et le préprocesseur Modbus spécifiques aux systèmes industriels [9].

Les flux de remontée d'alertes et d'administration de la sonde se font sur la même interface physique, en les isolant dans des VPN IPsec séparés.

La seconde interface est dédiée à l'acquisition. Cette interface n'est pas directement exposée dans le conteneur IDS. Un *TAP* logiciel est mis en place à l'aide d'un *bridge* et d'une paire d'interfaces virtuelles *veth* dont une extrémité est présentée à l'IDS. Des règles *ebtables* sont ajoutées dans le socle pour garantir l'unidirectionnalité des flux.

Le mécanisme de diode utilisé pour remonter les alertes est un tube nommé de type *FIFO*[10] entre le conteneur IDS et le conteneur de collecte de journaux.

4.2 Performances

Les tests ont pour but de mesurer l'impact des mesures de durcissement sur les performances. À cet effet, nous nous sommes basés sur les travaux de [9], pour disposer d'une sonde témoin (non durcie) et d'une capture de 200 000 paquets de trafic réel d'un système industriel. Cependant, les règles de l'IDS déclenchent une alerte pour chaque transaction Modbus. En pratique, sur la capture utilisée, cela équivaut à une alerte tous les deux paquets : l'objectif n'est pas de représenter

10. First In First Out

un volume d'alertes réaliste, mais de se placer dans le cas le plus défavorable pour évaluer les performances.

Les figures 2 et 3 présentent les résultats obtenus en rejouant la capture à différents débits avec la sonde témoin et la sonde durcie, respectivement.

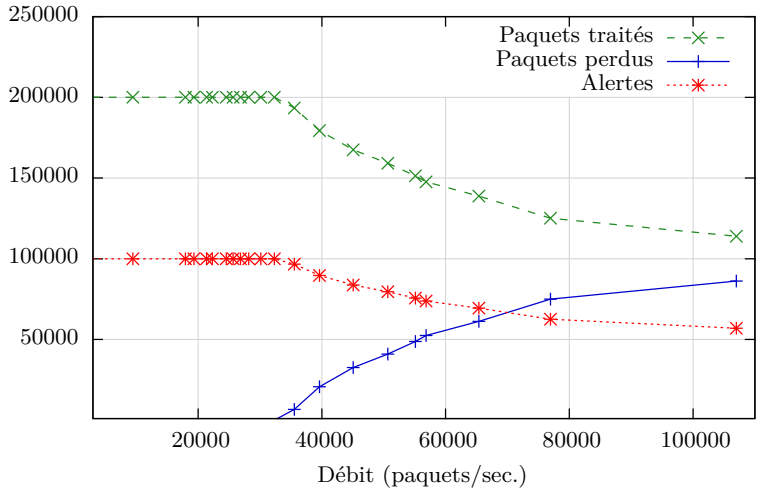


Figure 2. Résultats sur la sonde IDS témoin.

En observant le nombre de paquets perdus, on constate que l'impact global des mesures de durcissement sur les performances est d'environ 33%. En effet, tous les paquets sont correctement traités jusqu'à un débit de 32 500 paquets par seconde pour la sonde témoin (générant 16 500 alertes par seconde), alors que la sonde durcie peut traiter jusqu'à environ 21 500 paquets par seconde (générant 11 250 alertes par seconde).

Des mesures supplémentaires (non présentées ici) ont permis de déterminer que la mesure de durcissement la plus coûteuse en performance est l'utilisation du *TAP* logiciel (environ 60% du coût total). Viennent ensuite la diode de collecte d'alertes (environ 15%), les modifications apportées par PaX dans le patch grsecurity (environ 13%) et enfin la virtualisation légère LXC (environ 12%).

4.3 Discussion

Durant les tests de performances, la principale limitation constatée est la saturation des ressources CPU, alors qu'une marge était encore disponible pour les ressources mémoire, disque et réseau. En dehors de quelques réglages, aucune optimisation particulière de l'IDS n'a été réalisée.

Le *TAP* logiciel a un impact important sur les performances. Il est cependant facile et peu coûteux de remplacer ce *TAP* logiciel par un *TAP* matériel qui offre

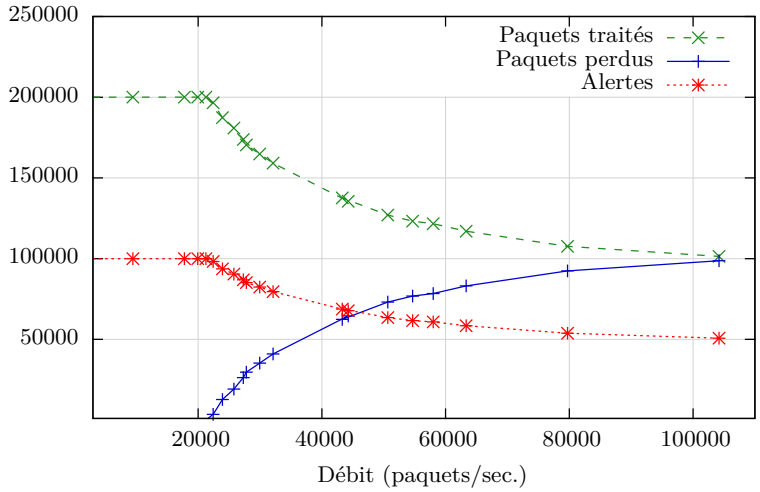


Figure 3. Résultats sur la sonde IDS durcie.

tout autant voire plus de garantie en terme de sécurité que le *TAP* logiciel, mais dont l'impact sur les performances est nul.

Le matériel choisi (MiraBox) ne dispose que d'un seul CPU, qui de plus est mono-cœur. Cette limitation est particulièrement importante ici, car l'IDS choisi (Suricata) et la virtualisation légère (LXC) sont des composants qui bénéficieraient particulièrement d'une augmentation des ressources CPU.

Dans l'ensemble, la preuve de concept a permis de valider qu'un matériel non spécialisé, peu coûteux (moins de 150€), supporte la mise en place d'une architecture sécurisée pour une sonde IDS. Si le nombre d'alertes générées dans notre expérience est artificiellement élevé, il permet toutefois de valider le fait que cette preuve de concept de sonde IDS durcie peut effectivement traiter un débit équivalent à plusieurs dizaines d'équipements [9] sans perdre aucune alerte.

5 Conclusion

Les sondes réseau, et plus généralement les logiciels IDS, sont les briques essentielles des systèmes de supervision de sécurité. La qualité de ces composants se résume traditionnellement à la sensibilité et à la précision de leur processus de détection de comportements inhabituels ou malveillants. Cependant, de part leur fonction et leur exposition, la sécurité intrinsèque de ces équipements s'avère cruciale. En effet, la compromission d'un IDS par un attaquant lui donnerait non seulement une position stratégique pour observer tout le trafic du réseau surveillé, mais lui permettrait également de compromettre d'autres équipements de ce réseau sans être détecté.

L'architecture logicielle de sonde IDS que nous proposons intègre les principes de défense en profondeur à l'état de l'art en matière de sécurité logicielle : durcissement du socle logiciel pour prévenir des exploitations « triviales », principe de moindre privilèges pour les utilisateurs et processus, différenciation des rôles d'administrateur système, d'administrateur IDS et d'auditeur, cloisonnement par virtualisation légère des différents composants logiciels (IDS, collecte, journalisation, mises à jour), flux unidirectionnels de collecte d'alertes, remontée des alertes sur un réseau de supervision dédié protégé en confidentialité et en intégrité.

Il faut cependant rappeler que l'architecture proposée ne modifie pas la qualité de la détection, ni les causes premières des vulnérabilités, qui restent liées à chacun des logiciels IDS utilisés. Sur ce point, le durcissement des logiciels IDS, voire l'écriture de parties critiques (en particulier les décodeurs protocolaires) dans des langages sécurisés restent des actions complémentaires indispensables.

Afin de montrer que l'architecture durcie que nous présentons dans ce papier est réalisable et viable, nous avons choisi de monter une preuve de concept de sonde IDS légère, particulièrement adaptée aux réseaux industriels. Nous avons conduit différents tests sur cette sonde durcie en utilisant une capture de trafic réel d'un système industriel dans le but de mesurer l'impact des mesures de durcissement. Les résultats obtenus montre un impact significatif des mesures de durcissement sur les performances. Néanmoins, nous nous sommes délibérement placés dans des conditions défavorables tant sur le nombre d'alertes générées que sur les choix d'architectures matérielle et logicielle. Nous pouvons donc raisonnablement statuer sur le fait que cette mesure constitue une borne maximale du coût engendré par les mesures de durcissement que nous préconisons dans notre architecture.

Bien qu'appliquée à la conception d'une sonde IDS légère, l'architecture que nous proposons reste entièrement valable pour la conception de sonde disposant de plus de ressources et capables de traiter des volumes de données dépassant le gigabit. De plus, sur du matériel de type serveur, la majorité des mécanismes de durcissement que nous proposons d'utiliser tirent partie de propriétés du matériel sous-jacent. L'impact sur les performances engendré par ces protections s'en trouvera alors fortement réduit puisque qu'ils ne sont plus implantés de manière logicielle, comme dans notre preuve de concept.

Références

1. AppArmor, Application Armor Linux security module. `http://wiki.apparmor.net/`.
2. Exemples de vulnérabilités critiques dans le logiciel Snort. `http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=snort`. Consulté en Juin 2014.
3. Exemples de vulnérabilités dans les décodeurs du logiciel WireShark. `http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=wireshark`. Consulté en Juin 2014.
4. PaX/grsecurity features. `http://grsecurity.net/features.php`.

5. Recommandations de sécurité relatives à un système GNU/Linux. http://www.ssi.gouv.fr/fr/guides-et-bonnes-pratiques/recommandations-et-guides/securite-du-poste-de-travail-et-des-serveurs/recommandations-de-securite-relatives-a-un-systeme-gnu-linux.html.
6. SELinux, Security-Enhanced Linux. http://selinuxproject.org/.
7. B. Spengler *et al.* PaX/grsecurity. http://grsecurity.net/.
8. Bro community. The Bro Network Security Monitor. http://www.bro.org/.
9. D. Diallo and M. Feuillet. Détection d'intrusion dans les systèmes industriels : Suricata et le cas de Modbus. In *C&ESAR*, 2014.
10. E. Jaeger and O. Levillain. Mind You Language(s). In *1st LangSec workshop of IEEE Security & Privacy*, May 2014.
11. Open Information Security Foundation (OISF). Suricata, Open Source IDS / IPS / NSM engine. http://suricata-ids.org/.
12. H. Shacham, E. Buchanan, R. Roemer, and S. Savage. Return-Oriented Programming : Exploits Without Code Injection. In *Black Hat USA*, 2008.
13. Sourcefire. Snort, Open Source network intrusion prevention and detection system. http://www.snort.org/.

RoCaWeb

Reconstruction de spécifications pour la détection d'intrusion Web

Yacine Tamoudi(1), Djibrilla Amadou Kountché(2), Alain Ribault(1),

Sylvain Gombault(2)

(1)Kereval ; 4 Rue Hélène Boucher, Z.A. Bellevue, 35235 Thorigné-Fouillard – France ;
{yacine.tamoudi,alain.ribault}@kereval.com
(2)Institut Mines-Telecom ; Telecom Bretagne ; IRISA/D2/OCIF RSM,
Université européenne de Bretagne, France
{djibrilla.amadoukountche,sylvain.gombault}@telecom-bretagne.eu

Résumé. L'analyse comportementale basée sur les spécifications est une stratégie de détection d'intrusion ayant de nombreux atouts reconnus par rapport à l'analyse à partir d'une base de signatures pourtant largement plus utilisée. Nous avions montré le bénéfice de cette approche dans le projet DALI[1] en imaginant le Shield [1]. Cet outil crée de manière semi-automatisée une liste blanche de règles décrivant le comportement normal de l'application via des contraintes. Toute requête client non conforme à ces contraintes sera identifiée et lèvera une alerte. Le but du projet RoCaWeb est de construire ces contraintes à partir de flux réseau sain, permettant de spécifier les requêtes autorisées d'une application web, en considérant toujours le serveur comme une boîte noire (donc sans accès au code source de l'application web). L'ensemble des contraintes est donc construit en utilisant des méthodes d'analyse des protocoles web et des techniques de génération automatique de règles à base d'expression régulières.

Mots clés: détection d'intrusion web, reconstruction de spécifications, alignement de séquences

1 Introduction

La détection d'intrusion est au cœur du processus de sécurisation des systèmes d'information (SI). Elle est complémentaire aux mesures de sécurité préventives dont l'efficacité est limitée par nature. Le référentiel général de sécurité [3] publié par l'ANSSI insiste sur l'importance de l'utilisation d'IDS (Intrusion Detection System) dans une architecture de sécurité. La détection doit être la plus précise, rapide et efficace possible pour déclencher au plus vite la réaction : c'est-à-dire les traitements

[1] DALI (Design and Assessment of application Level Intrusion detection systems) réf : ANR ARPEGE 2008.

prévus en aval. Par conséquent, il est primordial de comprendre au mieux la logique métier du service à protéger et d'adapter les processus de détection. Dans le cas des applications web, les mesures de sécurité préventives et la détection d'attaques sont rendues difficiles par plusieurs facteurs :

- La multiplicité des technologies web : elles varient de par leur langage, la plate-forme, et l'implémentation particulière qui sont parfois combinés en un ensemble hétérogène. On constate aussi la présence d'anciennes applications développées avec des technologies dépassées et dont les vulnérabilités sont connues.
- L'évolution rapide des applications : la majorité des applications sont développées pour offrir le plus de fonctionnalités le plus rapidement possible.
- La facilité d'exploitation des vulnérabilités en raison de la disponibilité d'outils d'attaques performants sur Internet.
- Le partage du code entre le serveur et le navigateur : la majorité du code s'exécute sur le serveur, mais le navigateur a la capacité de réaliser des traitements complexes avec l'interprétation du HTML, du JavaScript ou des protocoles comme web socket.

Un pare-feu traditionnel (ou dit de niveau paquet) est conçu pour avoir une action sur les couches réseaux (adresse IP, numéro de port) et ne protège pas contre une attaque de niveau applicatif transitant à l'intérieur d'un flux qu'il autorise. C'est le rôle d'outils de protection de type WAF (Web Application Firewall). Du point de vue du réseau, ce type d'outil a le rôle actif d'un mandataire inverse (reverse proxy) web et relaie les flux web vers le serveur. ModSecurity[2] est un logiciel gratuit de type WAF très répandu : il dispose d'un puissant moteur de règles pouvant appliquer une règle à plusieurs endroits de la transaction web. La plupart des entreprises utilisent la base de règles issue du projet OWASP ModSecurity Core Rule Set (CRS)[3] (règles décrivant des attaques connues et génériques. Par défaut, ModSecurity laisse passer les flux ne correspondant à aucune attaque, donc cette base a besoin d'être mise à jour régulièrement pour prendre en compte de nouvelles attaques.

Un autre moyen de sécuriser une application web est d'instrumenter son code avec un HIDS (Host IDS) dédié, en déterminant des invariants, comme cela a été fait par Ludinard et al. [4] et Felmetsger et al. [5].

L'approche innovante que nous présentons, RoCaWeb, permet de combiner reverse proxy, apprentissage d'une liste blanche de contraintes à partir d'un flux réseau sain, en considérant le serveur comme une boite noire.

Ce document s'articule de la façon suivante : le chapitre deux présente un rapide état de l'art dans le contexte de la détection et prévention d'intrusion web, auquel s'ajoute des méthodes de génération automatique d'expression régulière. Dans le chapitre trois nous décrivons RoCaWeb. La méthodologie d'évaluation de l'outil et les premiers résultats obtenus seront exposés dans le chapitre quatre. Enfin nous conclurons sur les perspectives de notre approche.

---

[2] ModSecurity : an open-source web application firewall: http://www.modsecurity.org

[3] OWASP ModSecurity Core Rule Set Project : https://www.owasp.org/index.php/Categorie:OWASP_ModSecurity_Core_Rule_Set_Project

2 Etat de l'art

De façon générale, un système de détection d'intrusion repose sur une stratégie de détection choisie parmi deux possibles [2]. La première est dite basée sur la connaissance (*Knowledge-based*) d'attaques type ciblant les biens sensibles du SI. Ces IDS possèdent une base de signature décrivant les attaques connues sous forme de liste noire.

La seconde est dite basée sur le comportement (*Behavioral-based*) et s'appuie la capacité à modéliser et valider le comportement voulu du système à protéger. Ces techniques font l'hypothèse qu'une intrusion peut être détectée en observant une déviation par rapport au comportement normal du système ou de l'utilisateur. L'avantage de cette approche est qu'elle permet de détecter des attaques non répertoriées et elles peuvent contribuer à faire découvrir des anomalies par rapport au modèle.

2.1 Détection d'intrusion basée sur la spécification

Cette stratégie de détection appartient à l'approche comportementale. Nous l'avons mise en œuvre avec succès dans le cadre du Shield [1]. Ce reverse proxy a été conçu pour renforcer la sécurité des applications web contre des attaques exploitant la faiblesse du contrôle de leurs entrées. Une application web peut avoir plusieurs paramètres d'entrées qui lui sont envoyés soit sous forme de formulaire, soit sous forme d'URL. Dans le cadre des formulaires, le code HTML les paramètres d'entrées sont localisés dans les pages HTML : ce sont des champs dont le contenu a été rempli par un utilisateur. Le Shield a deux phases de fonctionnement : une première construit ses contraintes en analysant la syntaxe HTML des réponses venant du serveur web. Il obtient ses pages en parcourant de manière semi-automatique l'ensemble du site. Cette phase génère un ensemble de contraintes correspondant aux champs d'un formulaire placées dans un agent de contrôle.. Lors de la deuxième phase, cet agent compare les requêtes en provenance des clients et les laisse passer si les contraintes sont respectées et par défaut une requête est bloquée. Pour palier les limites de cet apprentissage, nous avions enrichi notre outil avec un éditeur manuel de contraintes, permettant d'ajouter des règles contenant des expressions régulières dans des contraintes associées à des champs de la requête web. Au final, le Shield a donné de bons résultats dans sa capacité à détecter les attaques web, mais avec un effort important d'éditions de contraintes manuellement [1].

Reconstruction de spécifications à partir du trafic observé.

La reconstitution de modèle à partir du trafic observé peut être utilisée pour la détection d'intrusion web par spécification. Le comportement normal du système est déterminé à partir du trafic réseau observé. C'est à partir de ce comportement que sont générées les contraintes ou règles.

Netzob[4] est un exemple de projet consacré au retro engineering de protocoles réseaux et implémente plusieurs techniques de génération automatique de modèles.

2.2 Génération automatique de règles

Nous distinguons différentes formes de spécifications des règles : i) les dictionnaires ; ii) les automates à états ; iii) les expressions régulières ; iv) des seuils, relations, etc. Dans la section suivante, nous allons nous focaliser sur les expressions régulières et présenter des algorithmes qui permettent de les générer de façon automatique.

Problème de la génération automatique des expressions régulières.

Le problème consiste à générer une expression régulière qui correspond au « mieux » aux données et qui doit aussi être suffisamment générale pour accepter les données du même type avec des valeurs différentes. Cependant, il y a un grand nombre d'expressions régulières possibles correspondant à un ensemble de données et cela pose une limite à l'utilisation des expressions régulières pour la sécurité positive. Une expression régulière correspond à un automate à états finis. Il y eu plusieurs travaux sur la détermination de l'automate et de l'expression régulière qui est un problème difficile. Henning Fernau a proposé des algorithmes permettant de générer une expression régulière en se basant sur les arbres de suffixes et les automates à états [6]. *Conditional random field* (CRF) est une famille de méthodes initialement utilisée pour taguer les éléments d'une phrase avec les identifiants de groupes comme sujet, verbe, complément. En considérant, *X* comme l'ensemble des données observées et *Y* comme l'alphabet des expressions régulières, CRF peut être utilisé pour générer une expression régulière qui correspond aux données [7]. ReLIE est un algorithme proposé par Yunyao Li et al dans [8] pour la génération automatique d'expressions régulières à partir de documents textes et dans le but de servir à rechercher d'informations.

Méthodes basées sur l'alignement de séquences.

L'alignement de séquences permet de déterminer entre plusieurs chaines de caractères le maximum de caractères à la même position par insertion d'un caractère de bourrage. Cette méthode trouve son origine dans la bio-informatique pour : i) observer des patterns de conservation (ou de variabilité) ; ii) trouver des motifs communs présents dans les deux séquences d'ADN par exemple; iii) déterminer si les deux séquences ont évoluées à partir de la même séquence ; iv) trouver quelles séquences de la base de données sont similaires à une séquence de référence.

Par exemple, un alignement possible des chaînes « Kereval » et « Telecom-Bretagne » est donné par :

[4] Netzob, Reverse Engineering Communication Protocols. http://www.netzob.org/

```
T e l e c o m - B r e t a g n e
K e r e $ $ $ $ $ $ $ v a l $ $
```

Le caractère de bourrage est $. Pour un alignement de deux séquences, on constate qu'il existe quatre types de colonne : i) l'insertion : le caractère de bourrage a été inséré dans la première chaîne ; ii) la suppression : le caractère de bourrage a été inséré dans la seconde chaîne ; iii) la correspondance : les deux caractères sont identiques dans la même colonne ; iv) la substitution : les deux caractères ne sont pas identiques mais sont différents du caractère de bourrage.

Ces notions sont rassemblées dans une matrice des scores qui servira à déterminer l'alignement optimal entre les deux séquences. La *matrice des scores* indique le score entre les deux sous-chaînes des deux séquences à un instant de la progression de l'alignement. Elle se base sur le *score* et le *gap*. Le score est la fonction coût d'un alignement. Il est déterminé en se basant sur une fonction de similarité entre les caractères des deux séquences. Quant au gap, il indique la similarité entre un caractère et le caractère de gap. C'est valeur négative, le plus souvent, qui permet de pénaliser les insertions et les suppressions.

La matrice des scores pour une chaine s de longueur n et une chaine t de longueur m est une matrice de taille *(n+1)*(m+1)* et qui est remplie de la façon suivante [9]:

- la première ligne et la première colonne sont remplies par i*gap et j*gap ;
- pour toute autre cellule M(i,j) la valeur est déterminée par les trois cellules environnantes :

$$M(i,j)=\begin{cases} M(i-1,j)+g & \text{Insertion} \\ M(i,j-1)+g & \text{Suppression} \\ M(i-1,j-1)+d(x,y) & \text{Match/Mismatch} \end{cases}$$

Fig. 1. Fonction de remplissage d'une cellule de la matrice des scores

- Plusieurs algorithmes de programmation dynamique ont été proposés et exploitent la matrice des scores pour déterminer l'alignement optimal. Ces algorithmes peuvent être regroupés en trois catégories [9] :
- L'*alignement global* consiste à déterminer pour les deux chaînes, sur toute leur longueur, l'alignement qui produit le score maximal. C'est un problème d'optimisation dont la fonction objective dépend des paramètres décrits précédemment. L'algorithme de Needleman-Wunsch a été un des premiers à proposer un alignement global et trouve la solution optimale.
- L'*alignement local* réalise un alignement optimal des deux chaines par rapport à des sous-chaines. Ainsi, l'alignement ne s'effectue plus sur toute la longueur. L'algorithme de Smith-Waterman en est un exemple.
- L'*alignement multi-séquence* détermine un alignement consensuel entre toutes les séquences.

Génération des règles après l'alignement.

Cette phase consiste à partir des séquences alignées à déterminer une expression régulière. L'alignement de séquences a été déjà utilisé pour générer des signatures (expressions régulières) dans le cadre de la sécurité négative [12,13,14].. Nous avons mis en œuvre les stratégies suivantes : i) notre algorithme d'alignement multi-séquence combiné à notre méthode de génération d'expressions régulières; ii) Combinaison de ces deux approches avec le clustering.

Nous allons décrire ces deux points dans la section consacrée au module d'apprentissage de RoCaWeb.

3 Description de RoCaWeb

RoCaWeb comprend deux composants principaux :

- L'agent placé en amont du serveur web à protéger en tant que reverse-proxy. Il reçoit puis analyse le trafic échangé entre un client et un serveur pour détecter de potentielles attaques. Il utilise pour la validation un profil de sécurité comprenant les règles sur les entrées sorties.
- Le gestionnaire ayant pour but la création, la modification et la visualisation et le déploiement du profil de sécurité.

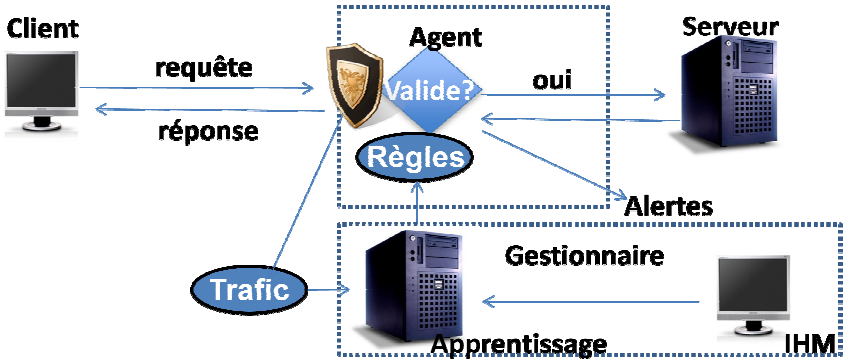


Fig. 2. Architecture de RoCaWeb

3.1 Agent

L'agent est un reverse-proxy chargé de réceptionner les messages provenant d'un client et de les faire suivre au serveur web. Pour ce faire il extrait les informations applicatives pour ensuite appliquer les contraintes issues d'un profil de sécurité.

Ces contraintes sont exprimées sous la forme d'expressions régulières considérées comme une liste blanche. Pour l'instant les contraintes ne concernent que les paramètres GET et POST et l'entête http, mais il est prévu que le module de validation

soit par la suite étendue pour appliquer des analyseurs syntaxiques de formats de données comme XML, HTML, JSON.

3.2 Gestionnaire

L'innovation du projet RoCaWeb ne se situe pas tant dans la validation du trafic par l'agent mais dans la gestion du profil de sécurité utilisé par celui-ci.

Dans le cas de la protection web, le protocole incontournable est HTTP. Les entrées sorties de l'application sont assimilées aux requêtes et réponses HTTP. En particulier les requêtes POST et GET contiennent des paramètres qu'un attaquant peut modifier et qui sont lus par l'application. C'est la technique d'attaque la plus courante et qui occupe la première place du top 10 OWASP : l'attaque par injection.

Dans notre approche, une requête est modélisée par un identifiant, l'URL, la méthode et la liste des paramètres associés.

Comme nous l'avons vu dans d'état de l'art, les contraintes peuvent être modélisées de plusieurs façons : la méthode que nous avons choisie de présenter dans ce document est l'expression de contraintes sous forme d'expressions régulières. Notre approche ayant pour but l'élaboration de spécifications en interaction avec l'utilisateur, il est important que la méthode de validation reste compréhensible et facilement modifiable.

La formalisation de règles sous forme d'expressions régulières peut être extrêmement coûteuse pour gérer la protection d'un site complet, celui-ci pouvant comprendre un nombre important de points d'injection potentiels.

Cette complexité est adressée de deux façons par notre outil :

- par la création automatique de règles à partir de trafic sain (phase d'apprentissage)
- par la visualisation des règles pour une gestion aisée du profil de sécurité (phase de modification et de validation des règles).

Apprentissage.

La source de données actuellement utilisée est le PDML (Packet Details Markup Language) comprenant les messages HTTP à destination du server. Le PDML est un format de sortie de l'utilitaire de capture réseau Wireshark.

Apprentissage par typage.

Cet algorithme associe aux paramètres un « type de donnée ». Les types sont représentés par leur expression régulière associée et sont classés du moins restrictif au plus restrictif. Le type le plus restrictif validant les données d'apprentissage est retenu pour la création de la règle. Les types utilisés sont par exemple :

- Alphanumérique : [0-9a-zA-Z]+
- Email : *^[_a-zA-Z0-9-]+(\\.[_a-zA-Z0-9-]+)*@[a-zA-Z0-9-]+(\\.[a-zA-Z0-9-]+)*\\.(([0-9]{1,3})|([a-zA-Z]{2,3})|(aero|coop|info|museum|name))$*
- Base64 : *^(?:[A-Za-z0-9+/]{4})*(?:[A-Za-z0-9+/]{2}==|[A-Za-z0-9+/]{3}=)?$*
- Chiffres *: [0-9]+*

- Texte libre : *\\s*[\\x09\\x20-\\xff]*\\s**
- Etc.

Pour affiner les règles créées, un autre algorithme associant des bornes aux règles a été implémenté. Les longueurs minimales et maximales des données de l'ensemble d'apprentissage sont intégrées à la formation de la règle.

Par exemple, à partir du paramètre *link_id* et de la liste de valeurs suivante *[206648, 206648, 206648, 206652, 205808, 206731]* :

- la règle obtenue par typage est : *[0-9]+*
- la règle obtenue par typage borné est *: [0-9]{6,6}*

Apprentissage par alignement.

Le module d'apprentissage effectue un prétraitement sur les données. Il applique l'alignement de séquences et génère les expressions régulières. Il est schématisé sur la figure suivante :

Fig. 3. Description des étapes du module d'apprentissage

Le prétraitement consiste à éliminer les doublons, les données manquantes et de les normaliser.

Alignement multi-séquence.

L'inconvénient d'aligner les chaînes deux à deux puis de générer une règle, de considérer cette règle comme une chaîne puis d'itérer, est que, d'une part on perd de l'information sur le consensus entre les champs fixes dans toutes les chaines de caractères et, d'autre part on réaligne l'expression régulière avec les autres chaînes ce qui constitue une limite en terme de performance. L'avantage de l'alignement multi-séquence est que nous générons une seule règle pour un ensemble de données (voir le clustering) à un instant de l'apprentissage et nous avons ainsi la garantie que les parties fixes ont été isolées dans toutes les chaînes. Lorsqu'un ensemble de chaines ne comporte pas de partie fixe (comme par exemple les mots de passe), nous déterminons des longueurs minimale et maximale pour caractériser cet ensemble.

Nous avons mis au point *notre propre algorithme* d'alignement multi-séquences .et cette méthode peut être combinée avec tout algorithme d'alignement de paire de séquences.

```
Entrée : un ensemble de chaînes de caractère
Sortie : un ensemble de chaines alignées
    1.Déterminer n*(n-1)/2 alignements de toutes les séquences
       et choisir les deux chaines ayant le meilleur aligne-
       ment.
    2.Insérer ces deux chaines dans un nouvel ensemble et les
       supprimer de l'ensemble de départ
    3.Tant que l'ensemble de départ n'est pas vide faire
       a. Supprimer un élément de l'ensemble
       b. Aligner cet élément à l'ensemble des chaînes déjà
          alignées puis choisir le meilleur alignement
       c. Si la chaine incluse dans l'ensemble des chaines ali-
          gnées a été modifié par le nouvel alignement alors
            i. Remplacer l'ancienne valeur de la chaine par la
                 nouvelle
       d. Inclure le résultat de l'alignement dans l'ensemble
       e.  Réajuster le contenu de l'ensemble en modifiant les
          chaines de caractères pour qu'elle ait la même lon-
          gueur.
```

Plusieurs algorithmes d'alignement multi-séquence ont été mis au point. Dans cette version de notre système, nous avons choisi de développer notre propre algorithme mais, nous envisageons de le comparer à plusieurs autres comme T-Coffee, Clustal, etc. Nous allons maintenant décrire la génération de règles.

Génération de règles.

L'algorithme de génération des règles est décrit par :

```
Entrée : Les chaines alignées ayant la même longueur.
Sortie : L'expression régulière correspondante.
  1.Pour chaque colonne faire
        a.  déterminer les caractères et leur fréquence
        b.  Si cette colonne ne comporte qu'un seul caractère
            alors c'est un caractère fixe.
        c.  Tant que le caractère suivant est fixe continuer
              i.  Si le caractère n'est pas fixe alors
                    1. Créer un ensemble et inclure tous les
                       caractères observés entre ces deux po-
                       sitions fixes
                    2. déterminer la longueur minimale et
                       maximale
                    3. ajouter à la règle les caractères or-
                       données et l'intervalle ([ensemble or-
                       donné des caractères]{min, max}
  2.Retourner la règle.
```

Clustering.

Nous avons utilisé le clustering hiérarchique [10] ascendant pour déterminer des classes avant de générer les règles. Dans cette version, nous avons utilisé une distance d'édition et la méthode « *complete link* ». Nous proposons deux stratégies : i) utiliser l'ensemble du dendrogramme ; ii) choisir un nombre de classe.

La première approche, consiste à déterminer toutes les classes possibles en partant d'une seule classe (comportant toutes les valeurs) à n classes par valeurs. L'intérêt de cette approche est de fournir à l'expert la possibilité de visualiser les classes et aussi de choisir celles qu'il veut prendre en compte. Par contre, cette approche est limitée lorsque la taille des données est importante. La seconde approche spécifie directement un nombre de classes à trouver et détermine les règles pour ces classes. Par la suite la règle finale peut être une combinaison des règles ou une arborescence.

Nous allons illustrer le fonctionnement du module d'apprentissage dans la partie évaluation.

Meta apprentissage.

La diversité des données sur le Web augmente la difficulté de l'apprentissage. On y retrouve des variables internes à l'application, des adresses email, des mots de passes ainsi que toute donnée potentiellement utilisable par un être humain.

D'après la loi de conservation pour la performance de généralisation [11], il est naturel qu'il n'existe pas de méthode d'apprentissage permettant de créer des règles pour tout type de donnée. Par exemple, une création d'expression régulière par alignement sur un champ de mots de passe n'est, bien sur, pas la méthode qui convergera le plus rapidement en fonction des données fournies. Par conséquent pour un algorithme donné, le nombre de faux positif des règles brutes sera donc très élevé sur certains paramètres. Or le taux de faux positif est très handicapant pour une application réelle.

Pour répondre à ce problème, la validation croisée des données a été choisi pour pouvoir choisir au mieux l'algorithme à utiliser en fonction des données associées à un paramètre. L'algorithme choisi est la *K-fold* validation. Les données d'apprentissage sont découpées en *K* parties et chaque combinaison de *K-1* clusters est utilisée pour générer une règle. Le cluster restant sert ensuite à valider les règles générées. Il est donc possible, à partir des données saines d'estimer le taux de faux positif final.

Visualisation.

Le profil obtenu peut être visualisé dans l'interface graphique. Les contraintes permettent de caractériser la valeur des paramètres transmis dans des messages HTTP. Elles sont donc identifiables par l'URL associée à ces requêtes et réponses HTTP permettant d'agir avec l'application à protéger. Il est alors possible de les classer dans une structure en arbre.

Cette représentation a plusieurs utilités : elle permet à l'utilisateur d'évaluer la structure du profil et de la comparer à la structure du site réel pour pouvoir estimer le taux de couverture de son profil et combler les manques si besoin est. Elle permet d'accéder aux règles pour modifier les expressions régulières de façon aisée. Il est

aussi possible de gérer les données d'apprentissage, de choisir les algorithmes utilisés et d'en modifier les paramètres. Des indicateurs obtenus lors de la création des règles comme la taille de l'échantillon d'apprentissage ou le taux de correspondance de la validation croisée, permettent d'estimer le degré de confiance que peut avoir l'utilisateur dans les règles créées. L'interface doit aussi permettre d'agir sur l'agent ainsi que le déploiement des règles.

4 Evaluation

Dans cette partie, nous allons illustrer, sur quelques exemples, le fonctionnement de nos méthodes et leurs performances.

4.1 Illustration de l'alignement et de la génération de règles

Le tableau ci-dessous donne les descriptions de trois bases d'apprentissage : de pays se terminant par « -stan », de mot de passe et d'un exemple courant en alignement multi-séquence.

Nom de pays	Mots de passe	Garfield
Afghanistan *Pakistan* *Kazakhstan* *Kurdistan* *Kirghizistan* *Ouzbékistan* *Tadjikistan* *Turkménistan*	J>K;Goa4g dS2T}<:Cz X^2N(eby1 *xMeWq3I$ GU#h4Pu(l =S!3xu\fg R*3CvVB!}	GARFIELD THE LAST FAT CAT GARFIELD THE FAST CAT GARFIELD THE VERY FAST CAT THE FAT CAT
n=9	n=7	n=4

Table 1. Les bases d'apprentissage

En utilisant les paramètres suivant : *gap = -10*, match = 1, *mismatch = -1*, le *caractère de bourrage* = « ¤ », le résultat obtenu avec un alignement multi-séquence est :

¤Afghanistan ¤Ouzbékistan Turkménistan ¤¤¤¤Pakistan ¤¤¤¤Dagestan ¤¤Kazakhstan ¤¤¤Kurdistan ¤Tadjikistan Kirghizistan	GU#h4Pu(l X^2N(eby1 dS2T}<:Cz =S!3xu\fg J>K;Goa4g R*3CvVB!} xMeWq3I$	GARFIELD THE VERY FAST CAT GARFIELD THE ¤¤¤¤¤FAST CAT ¤¤¤¤¤¤¤¤¤THE ¤¤¤¤¤FA¤T CAT GARFIELD THE LAST FA¤T CAT

[ADKOPTabdefghijkmnruzé]{4,8}stan	[!#\$\(*1234:;\<=\>BCGIJKMNPRSTUVWX\\\^abdefghloquvxyz\}]{9}	[ADEFGILR]{0,9}THE [AELRSTVY]{0,5}FA[S]{0,1}T CAT

Table 2. Résultats de l'alignement multi-séquence

Une autre approche que nous avons utilisée, est la combinaison du clustering avec l'alignement multi-séquence. Le processus est continué jusqu'à atteindre la fin du dendrogramme.

Notre méthode permet de rapidement créer des règles à partir d'un ensemble d'entrée. Le cas des mots de passe illustre le fait que lorsque de parties communes n'existent pas, notre méthode détermine des bornes et l'ensemble de caractères. L'intérêt du clustering est de permettre de découper l'ensemble d'apprentissage et d'affiner les règles pour les besoins de l'expert. En gardant les mêmes paramètres et en prenant la base des pays se terminant en « stan » nous obtenons le résultat suivant :

Clusters	Expressions régulières	Observations
[Tadjikistan Turkménistan Kurdistan Pakistan Kazakhstan Ouzbékistan Afghanistan Dagestan Kirghizistan]	[ADKOPTabdefghijk-mnruzé]{4,8}stan	Dans ce cas, tous les éléments sont dans la même classe.
[Tadjikistan, Turkménistan, Ouzbékistan, Afghanistan, Kirghizistan] [Kurdistan, Pakistan, Kazakhstan, Dagestan]	[AKOTabdfghijkmnruzé]{6,7}istan [DKPadeghikruz]{4,6}stan	Ce clustering comporte deux classes. Et pour chacune une règle est générée.

Table 3. Résultats de la combinaison du clustering et de l'alignement multi-séquences

4.2 Evaluation des performances de détection d'intrusion

Nous allons maintenant comparer les performances des algorithmes de reconstruction de spécifications sur un cas réel.

Environnement.

Pour l'expérimentation une version intermédiaire de RoCaWeb a été utilisée. Cette version comprenait les modules d'apprentissage suivants :

- Typage : simple ou borné,
- Alignement multi séquence,
- Validation croisée *K-fold* avec *k=10* et *g=1* puis avec *k=10* et *g=0,8*.

Le site utilisé pour le test est un site PHP d'intranet d'entreprise. Les points d'injection considérés dans cette expérimentation ont été limités aux paramètres des messages HTTP GET et POST. Environ 500 paramètres potentiellement injectables ont été pris en compte lors de l'expérimentation. Ceux-ci sont de types variables avec comme valeur par exemple :

- OK = « Valider » *: caractères alphanumériques*
- t = « 1 » : *chiffres*
- desc =« Repr%E9sentation+de+la+matrice+des+comp%E9tences » : *texte libre*

Deux types de données ont été utilisés : le trafic réel provenant du site de gestion d'entreprise. On considère que les données sont saines. Elles ont été récoltées sur une période de 5 semaines : semaine *1 : 1389* requêtes dynamiques ; semaine *2 : 1790* requêtes ; semaine *3 : 651* requêtes ; semaine *4 : 1351* requêtes et semaine *5 : 1356* requêtes. Les quatre premières semaines ont été utilisées pour l'apprentissage et la cinquième pour le test.

Les données d'attaques provenant de l'outil d'automatisation d'injection SQL SQLMAP[5]. L'outil à été configuré pour couvrir un maximum de point d'injection de l'application.

Les critères pris en compte pour l'évaluation seront le *taux de faux positifs* et le *taux de faux négatifs*. Ces informations permettent d'évaluer la pertinence d'un algorithme de décision et de le représenter sur une courbe ROC. Le taux de faux positif représente le taux de messages sains injustement bloqués, ou le taux de fausses alarmes. Le taux de faux négatif est le taux de messages malsains non détectés. Les variables manipulées lors de l'expérimentation ont été : les algorithmes d'apprentissage la quantité de données (semaines) utilisées pour l'apprentissage.

Résultats.

Les performances ont été mesurées en incluant progressivement les données des semaines capturées pour l'apprentissage : points S4 (données de la semaine 4), point S3,4 : données des semaines 3 et 4 ; S 2,3,4 : données des semaines 2, 3, 4 et S1,2,3,4 : données des semaines 1, 2,3 et 4.

[5] SQLMAP, https://github.com/sqlmapproject/sqlmap

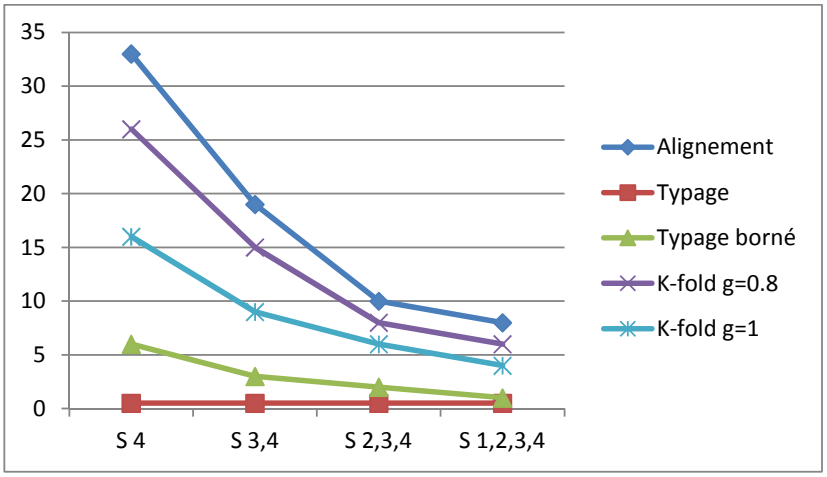


Fig. 4. Evolution du taux de faux positif

On constate que le taux de faux positif diminue avec la quantité de données fournies aux algorithmes. Le nombre de fausses alertes de l'algorithme par alignement reste le plus fort mais se réduit progressivement. Le taux de faux positif de l'algorithme de typage simple reste constant et quasi nul. La *K-fold* validation permet d'avoir un intermédiaire entre l'alignement et le typage et le taux de faux positif de cette méthode varie selon le taux de correspondance utilisé.

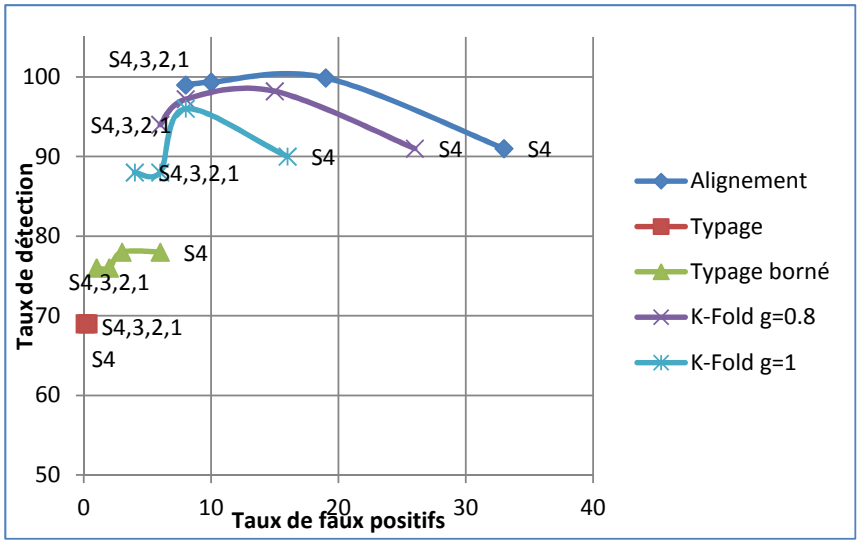


Fig. 5. Représentation ROC

Les taux de faux positif et de faux négatif sont ensuite représentés sur une courbe ROC. Chaque courbe représente l'amélioration des performances des algorithmes avec l'ajout progressif de données. On constate que le typage classique est quasi constant : les performances ne s'améliorent pas avec l'ajout d'information. Le taux de faux positif est faible et le taux de faux négatif fort.

Pour les autres méthodes on observe une amélioration du taux de faux négatif entre la semaine 4 seule et les semaines 3 et 4. La baisse initiale de ce taux est du à l'incomplétude des données initiales : la semaine 4 ne couvrant pas suffisamment l'application, des parties du site ne sont pas couverte par les règles. Ce type de défaut peut être géré au niveau de la visualisation.

Le taux de faux négatif augmente ensuite légèrement pour l'algorithme d'alignement avec l'utilisation de toutes les données car les expressions régulières se généralisent et deviennent ainsi plus flexibles. Enfin le taux de faux négatif des méthodes de validation croisées augmente plus brusquement pour les méthodes de validation croisée dans les mêmes circonstances car la complexité des données impacte la sélection automatique de la méthode de typage.

La performance en charge n'a pas été prise en compte. L'accent a donc été porté sur la mesure de la pertinence de ces expressions régulières par rapport aux données d'apprentissage. Les algorithmes d'apprentissage n'ont pas pour but de fournir des règles immédiatement déployables mais d'aider l'utilisateur à créer son profil de sécurité. Néanmoins les performances sont ici testées sans modifications des règles. Le taux de faux positif mesure donc ici le nombre de règles que l'utilisateur doit éditer.

5 Conclusion et perspectives

Le projet RoCaWeb explore une approche peu courante de la protection Web : la sécurité par la spécification. Celle-ci permet de protéger une application à partir du comportement attendu de l'application : c'est l'utilisateur qui apporte sa connaissance métier pour construire un profil de sécurité, ce qui lui permet de garder le contrôle sur les actions effectuées par l'utilitaire. Cette approche peut non seulement de prévenir les attaques mais aussi les violations de la logique métier qui peuvent, intentionnellement ou non, nuire au bon fonctionnement de l'application. Cette création de profil est considérablement accélérée grâce à la reconstruction des contraintes à partir d'enregistrement de trafic sain. Ce pré-profil peut être ensuite facilement compris et modifié par l'utilisateur grâce à une visualisation pertinente.

Cette vision se rapproche, par sa méthodologie, du principe du test basé modèle. Cette « sécurité basée modèle » présente des résultats encourageant au cas d'application du web malgré le dynamisme des applications.

RoCaWeb est un projet en cours et il est prévu de compléter l'approche en suivant différents axes. Les contraintes prises en compte actuellement sont exprimées sous la forme d'expression régulière. Cette forme de contrainte pourrait être étendue en utilisant un langage avec un plus grand pouvoir d'expression. La validation des entrées peut aussi être améliorée en ne considérant pas les paramètres individuellement mais par les relations qu'ils peuvent avoir entre eux.

Les travaux présentés dans ce document ne traitent pas de la validation des sorties de l'application et du parcours utilisateur. Les sorties peuvent aussi être caractérisées grâce à un apprentissage à partir de données saines. Les contraintes construites concernent alors les structures de données et le traitement des ressources liées par exemple. La validation du parcours utilisateur peut se faire par génération d'automate et suivi de la session utilisateur.

RoCaWeb est réalisé dans le cadre d'un projet RAPID d'innovation duale et nous tenons à remercier Constant Chartier et Fréderic Majorczyk nos interlocuteurs à la DGA-MI pour leur conseils. Nous souhaitons aussi remercier le personnel de Kereval, ainsi que les étudiants de Telecom Bretagne qui ont contribué au projet.

6 Références

1. Mouelhi Tejeddine, Le Traon Yves, Abgrall Erwan, Gombault Sylvain, Baudry Benoit. Tailored shielding and bypass testing of web applications. International Conference on Software Testing, Verification and Validation, 21-25 march 2011, Berlin, Germany, 2011.
2. Debar, Hervé, Marc Dacier, and Andreas Wespi. "A revised taxonomy for intrusion detection systems." Annales Des Telecommunications Juillet–Aout 2000, Volume 55, Issue 7-8, pp 361-378.
3. Référentiel general de sécurité V2.0. http://www.ssi.gouv.fr/IMG/pdf/rgs_v2.pdf
4. Ludinard, Romaric, Loic Le Hennaff, and Eric Totel. "RRABIDS, un système de détection d'intrusion pour les applications Ruby on Rails." *Actes du Symposium 2011 sur la Sécurité des Technologies de l'Information et des Communications.* 2011.
5. Felmetsger, Viktoria, et al. "Toward automated detection of logic vulnerabilities in web applications." *USENIX Security Symposium*. 2010.
6. Fernau, H. (2009). Algorithms for learning regular expressions from positive data. *Information and Computation*, *207*, 521-541.
7. Sutton, C., & McCallum, A. (2006). An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, 93-128.
8. Li, Y., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., & Jagadish, H. V. (2008, October). Regular expression learning for information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 21-30). Association for Computational Linguistics.
9. Gusfield, D. (1997). *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge University Press.
10. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.
11. A conservation law for generalization performance - Schaffer – 1994
12. Tang, Y., Xiao, B., & Lu, X. (2009). Using a bioinformatics approach to generate accurate exploit-based signatures for polymorphic worms. *computers & security*, *28*(8), 827-842.
13. Li, Z., Sanghi, M., Chen, Y., Kao, M. Y., & Chavez, B. (2006, May). Hamsa: Fast signature generation for zero-day polymorphic worms with provable attack resilience. In *Security and Privacy, 2006 IEEE Symposium on* (pp. 15-pp). IEEE.
14. Newsome, J., Karp, B., & Song, D. (2006, January). Paragraph: Thwarting signature learning by training maliciously. In *Recent advances in intrusion detection* (pp. 81-105). Springer Berlin Heidelberg.

Après l'attaque

Philippe Davadie

Gendarmerie Nationale

Résumé *L'indispensable préparation des entreprises aux attaques informatiques pour sérieuse qu'elle aura été, ne peut suffire à les repousser. La réaction immédiate et à terme à une attaque doit donc être connue de l'entreprise, le simple retour en ordre de marche ne pouvant suffire, une réitération de l'attaque étant possible. L'entreprise doit d'abord différencier l'attaque de la négligence. Si elle cherche à obtenir réparation, l'embryonnaire voie de l'assurance ainsi que la voie judiciaire, civile ou pénale sont envisageables. Chacune a ses partisans, l'intérêt de la voie pénale étant d'obtenir la condamnation du coupable. La voie pénale n'est possible qu'en cas de faits prévus et réprimés par le code pénal, la tentative étant parfois punissable. Le dépôt de plainte doit s'effectuer dans les meilleurs délais pour éviter la prescription. Enfin, une coopération étroite de l'entreprise avec les enquêteurs augmente les chances de réussite de l'enquête.*

Key words: attaque, négligence, infraction, tentative, Code pénal, plainte, prescription

évoquer la cybercriminalité consiste le plus souvent à décrire les mesures de protection indispensables, les infractions possibles et le verdict du procès. La littérature se focalise d'ailleurs sur ces sujets, oubliant la phase primordiale de réaction à l'attaque. L'objet de cet article est de participer à combler ce manque.

Malgré ses efforts de protection, l'entreprise sera attaquée. Focalisée sur sa production, elle laisse ainsi un temps d'avance aux attaquants, concentrés sur la lutte informatique. Le succès de l'attaque est donc fort probable. Pour autant, anticiper l'attaque est utile car cela permet de réduire les dommages à venir. Mais cette préparation, pour indispensable qu'elle soit, n'empêchera pas tout passage à l'acte. Pour préparer et anticiper le succès de l'attaque, l'entreprise dispose d'une large palette d'actions, étant entendu qu'une *légitime défense* informatique, aussi séduisante que l'idée paraisse, ne peut tenir juridiquement.

Précisons aussi qu'une négligence ou une erreur de manipulation de l'un de ses employés peut causer le dysfonctionnement de l'une des informatiques de l'entreprise. Par commodité, le terme générique d'*incident* sera utilisé dans les lignes qui suivent pour englober tant l'attaque informatique que l'erreur ou la négligence.

L'entreprise doit réagir à tout *incident*. Prendre des premières mesures, dites d'urgence, lui permettra parfois de trancher le dilemme attaque ou négligence. De là découleront les mesures prises à plus long terme, tel le fait de porter plainte. Le souci de préserver ses droits doit accompagner ces mesures car une attaque peut

passer pour une négligence. Et pour éviter la réitération de cet *incident*, elle devra aussi en limiter la propagation, tant à l'intérieur de l'entreprise qu'à l'extérieur.

Enfin, elle pourra demander réparation du préjudice subi grâce aux assurances ou la justice (civile ou pénale), celle-ci impliquant l'état, seul à disposer des pouvoirs pour mener une enquête judiciaire et condamner les coupables. De plus, impliqué dans la guerre économique aux côtés des entreprises, il peut user de réponses non judiciaires, comme l'a rappelé le ministre de la Défense dans le discours d'ouverture du colloque sur la cyberdéfense à Rennes le 3 juin 2013 : *Cette capacité de réponse fera en premier lieu appel à l'ensemble des moyens diplomatiques, juridiques ou policiers, sans s'interdire l'emploi gradué de moyens relevant du ministère de la défense, si les intérêts stratégiques nationaux sont menacés.*

1 Réagir

Connaître les causes d'un dysfonctionnement informatique impose de se demander si une attaque a eu lieu ou si un employé a commis une négligence ou une erreur. Ce distinguo peut être difficile à établir car l'entreprise semblera d'abord confrontée à une *panne*. Elle voudra donc réparer l'informatique visée, ce qui peut nuire à la collecte des preuves en cas d'attaque informatique avérée.

Cette remise en marche a néanmoins des vertus, car elle permet d'évaluer sommairement la cause de la panne : informatique, mécanique, voire inexpliquée. Cette analyse et les mesures prises pour revenir à un état de marche normal sont des éléments importants pour connaître ce qui s'est vraiment passé.

1.1 Négligence ou attaque

Aborder le distinguo négligence ou attaque, pose la question de l'acte délinquant et donc de sa qualification pénale. Rappelons qu'il faut trois éléments pour constituer une infraction : l'élément *légal* (la loi qui prévoit et réprime l'infraction), l'élément *matériel* (les actes exécutés par l'auteur) et l'élément *moral* (l'intention coupable ou la faute d'un auteur conscient de ses actes). Que l'un de ces éléments fasse défaut et l'infraction ne peut être caractérisée. Mais ce défaut ne signifie pas nécessairement que l'entreprise est face à une négligence. Ce peut être une attaque qui, par manque d'un seul élément constitutif, ne sera pas réprimée pénalement. Si l'élément légal et l'élément matériel semblent faciles à établir, il est plus malaisé d'établir l'*intention coupable* qui peut parfois se confondre avec le mobile de l'acte, ces notions étant parfois proches. En première approche, on pourrait estimer qu'une intention est coupable dès lors que l'acte est commis *en pleine conscience et de propos délibéré*. L'auteur des faits devait être en pleine possession de ses moyens intellectuels et animé de la volonté de faire ce qu'il réalisait : son jugement ne devait pas être altéré ni sa volonté contrainte. Le prouver s'avère ardu, à moins que l'auteur des faits avoue son intention. Cette difficulté à prouver l'intention coupable est caractéristique de la *délinquance en col blanc* pour laquelle l'élément moral est parfois confondu avec l'élément matériel.

Au sein de l'entreprise, certains éléments peuvent disculper un employé de cette intention. Toutes les entreprises affirment disposer d'une politique de sécurité connue par chaque employé. Mais chacun a-t-il bénéficié d'une formation adaptée à son poste et à ses enjeux de sécurité ? Parmi les faits s'opposant à cette parfaite connaissance nous pouvons noter :

- une méconnaissance de l'importance du SI de l'entreprise et du caractère indispensable de son bon fonctionnement quotidien ;
- la possession d'appareils informatiques personnels (ordiphones, tablettes et PC) considérée comme une preuve de connaissance des enjeux de sécurité alors qu'elle n'est souvent qu'une nécessité ou un sacrifice à la modernité ;
- une méconnaissance généralisée de la sécurité informatique, trop souvent présentée comme une affaire d'experts ;
- une présentation de la sécurité trop théorique et pas assez axée sur ce que chacun doit accomplir pour se protéger et protéger son entreprise ;
- une sécurité informatique présentée uniquement sous son aspect contraignant (car à contre-courant du confort d'utilisation) sans en montrer les gains ;
- une prise de connaissance parcellaire de la politique de sécurité car bien souvent la lecture rapide et l'émargement d'un document en tiennent lieu ;
- enfin, l'absence de respect de la SSI par les employés ayant le plus de responsabilités, ce qui pousse les autres à faire de même.

Si plusieurs de ces éléments font partie du quotidien de l'entreprise, il est fort probable que l'un de ses employés se rende un jour coupable d'un acte préjudiciable à l'entreprise. La négligence sera alors plus à blâmer que la volonté de nuire, et l'établissement de l'intention coupable n'ira pas de soi. En outre, le personnel de l'entreprise en utilise les ressources à des fins personnelles, ce qui n'est pas grave car une certaine tolérance existe toujours. Ce faisant, il fragilise l'entreprise par des connexions à des sites non sécurisés ou à des réseaux sociaux pour lesquels les alertes de sécurité abondent. Ces défauts de sécurité permettent souvent de procéder à une ingénierie sociale assez aisée, et donc de s'introduire dans le SI de l'entreprise.

Le manque de formation à l'utilisation des différents matériels et logiciels peut aussi occasionner des actes préjudiciables à l'entreprise. Prenons le cas de l'informatique périmétrique [1] : sa gestion quotidienne est souvent confiée à des vigiles et non des informaticiens. L'exploitation de cette informatique par une société tierce ajoute à la complexité de discrimination négligence/malveillance car l'entreprise n'a qu'une faible influence sur la qualification de ces sous-traitants, tant pour leur compétence en matière de sécurité que d'informatique. Compter alors sur les exploitants du système pour administrer correctement le réseau sur lequel se

1. Par informatique périmétrique nous entendons l'ensemble des capteurs et logiciels qui permettent la détection de toute transgression, par une personne ou une chose, d'un périmètre donné, quel qu'en soit le sens, entrée ou sortie, l'identification du transgresseur, et qui aident à déterminer si ce *bris de clôture* est une entrée ou une intrusion, celle-ci étant autorisée, celle-là n'étant pas souhaitée.

trouvent les capteurs (caméras, lecteurs de badge, etc.), détecter des bugs ou des intrusions est illusoire. N'oublions pas que le manque de vigilance des employés envers le matériel qui leur est confié (badges, ordinateurs, téléphones portables, etc.) est aussi responsable d'atteintes involontaires envers l'entreprise. Il se traduit par des pertes ou des vols qui sont de formidables opportunités pour les attaquants de toute sorte. Une étude menée en 2008 par l'institut *Ponémon* a montré que, chaque semaine, plus de 3000 ordinateurs portables professionnels étaient perdus dans les huit principaux aéroports européens. D'après la même étude, plus de la moitié des ordinateurs égarés contiennent des données confidentielles non cryptées. On ne peut cependant occulter le savoir-faire et le professionnalisme de voleurs chevronnés qui feront que la victime déclarera plutôt une perte inexplicable qu'un vol dont elle n'a pas la moindre preuve.

Deux autres phénomènes méritent également d'être mentionnés dans le dilemme attaque/négligence. Il s'agit du cas de l'AVAP (Apportez Votre Appareil Personnel ou BYOD en anglais) et celui du contrôle des licences logicielles.

L'AVAP concerne l'informatique personnelle souvent développée sans prendre en compte sa sécurité car la rapidité de lancement de nouveaux produits est primordiale pour devancer la concurrence. En se développant, l'entreprise a dû composer avec la volonté de ses employés de disposer du même type de matériel que le leur. Confrontée à un coût trop important pour elle, elle a donc accepté que ses employés utilisent leurs appareils personnels. Le développement de ce phénomène est paradoxal à plusieurs titres :

- la protection de la vie privée a pris de l'importance ces derniers temps (Prism le montre) mais un appareil personnel connecté au réseau de l'entreprise peut voir ses données consultées par les administrateurs dudit réseau ;
- chacun veut disposer de davantage de temps libre mais la vie professionnelle tend à s'insérer dans la vie privée (des logiciels proposent de renvoyer les images vidéo de l'informatique périmétrique sur les ordiphones personnels) ;
- l'entreprise offre davantage de possibilités techniques à ses employés tout en restreignant leurs accès (aux réseaux notamment) et leurs droits.

La question plus critique de la compatibilité des politiques de sécurité informatique de l'entreprise et de celles de chaque employé ne peut manquer d'être posée. Comment garantir un niveau de sécurité nécessaire, voire suffisant, en accueillant sur son réseau des matériels sur lesquels on n'a aucune prise ? Imposer des configurations et contraintes de sécurité, restreindre les connexions est possible pour les matériels de l'entreprise, mais impossible, à moins d'un effort lourd et permanent, pour les appareils personnels des employés. L'hétérogénéité et la fréquence de changement de ce parc informatique personnel rend ce problème encore plus aiguë.

Mettre en rapport l'AVAP et la sécurité informatique de l'entreprise amène à se poser la question de sa pertinence. S'il est un souci supplémentaire pour les responsables de la sécurité du SI, est-il utile ? Les gains espérés valent-ils l'introduction de nouvelles vulnérabilités ? Selon une étude de l'institut Ponémon de 2013, 70 % des entreprises françcaises n'ont pas défini de politique relative à l'AVAP. 73 %

de celles qui en ont défini une estiment qu'elle doit s'appliquer strictement, 16 % consentent des exceptions pour les cadres. Seulement 21 % des entreprises ont informé leur personnel des risques que cette pratique pouvait entraîner pour l'entreprise et eux-mêmes. à ces constats, le directeur de l'ANSSI répond de manière très claire dans son discours de Monaco en 2012 : *Dans une entreprise : non on ne travaille pas avec son terminal privé, non on ne connecte pas un terminal contrôlé par un tiers, non on n'installe pas le dernier joujou à la mode.* Le rapport coût efficacité de cette pratique doit donc être pris en compte, car il fait courir le risque que l'entreprise passe du *bring you own device* au *bring your own disease*.

Toute entreprise utilise des logiciels développés par d'autres qu'elle-même en versant une redevance à l'éditeur. Conformément à la logique et aux dispositions du contrat, l'éditeur peut venir vérifier *in situ* qu'aucune copie illégale n'est installée et qu'il est rétribué selon l'usage fait de ses logiciels. Cependant, ces opérations peuvent donner lieu à des actes qui, s'ils ne peuvent être qualifiés de malveillants, laissent perplexe. Lors de l'audit, l'auditeur peut accéder à l'intégralité du système d'information de l'entreprise auditée. Cette politique de contrôle est d'ailleurs souvent qualifiée d'agressive, parce qu'il existe avant même le début des opérations un déséquilibre entre l'auditeur et l'audité. Que l'éditeur effectue ou délègue l'audit, les opérations sont intrusives et permettent de connaître les arcanes du système d'information de l'entreprise. Une maladresse ou une malveillance aura donc des répercussions importantes sur le SI audité. Pour s'en prémunir ou du moins les limiter, l'audité doit s'assurer que la clause d'audit est claire et limitée, c'est-à-dire qu'elle prévoit le périmètre de l'audit et l'engagement que ces opérations ne perturberont pas l'activité de l'entreprise. Il peut aussi demander que l'auditeur prenne par écrit la responsabilité des conséquences des commandes passées sur les machines de l'entreprise auditée. La probabilité d'acceptation étant faible, l'entreprise auditée pourra alors ne faire face qu'à un audit déclaratif qui préservera l'intégrité de son SI.

En cas de problème durant l'audit, l'audité peut poursuivre l'auditeur au motif qu'il a outrepassé ses droits : Oracle a été condamnée par le tribunal d'Amsterdam en 2010 dans le cadre d'une action en contrefaçcon qu'elle avait intentée contre son client (Philips). Le tribunal a en effet estimé que la demande formulée par Oracle envers son client était trop vague, trop large et que durant la phase de collecte de données, elle s'était livrée à une « *fishing expedition* ».

1.2 Réactions pratiques

L'incident constaté, il faut réagir au plus vite et de manière appropriée. Rechercher et analyser les causes de l'incident s'impose. Si l'entreprise ne dispose pas de compétences internes suffisantes, elle peut faire appel à des partenaires de confiance. L'article 15 du décret 2010-112 [2] prévoit qu'un organisme habilité par l'ANSSI délivre à certains prestataires une *qualification qui atteste de la conformité des services à un niveau de sécurité défini par le référentiel général de sécurité.*

2. Décret n° 2010-112 du 2 février 2010.

Bien que ne visant que les autorités administratives, ce décret permet de faire émerger des prestataires rigoureux auxquels les entreprises pourront faire appel.

La recherche des causes de l'incident est très importante, surtout si l'entreprise se rend compte *a posteriori* qu'il s'est agi d'une attaque. L'étude des opérations effectuées et de leur résultat permettra de déduire tout ou partie du mode opératoire de l'attaque. Si l'entreprise pense avoir été attaquée, il vaut mieux que ces opérations techniques (mesures prises et résultats) soient authentifiées par un huissier, afin de ne pas être contestées lors de leur production devant un tribunal. Il ne certifiera pas la pertinence des opérations, mais juste leur nature (commande passée), leur enchaînement chronologique ainsi que les conditions dans lesquelles elles auront été effectuées (par qui, où, sur quel support sont stockés les résultats, etc.). Devant l'abondance de la littérature technique relative à l'exécution de ces opérations d'investigation, il n'est pas nécessaire de les détailler.

Le but des premières opérations est de déterminer ce qui est en *panne* ou contaminé par un code malveillant afin de circonscrire le périmètre de l'intervention. à ces opérations peuvent se superposer celles qui ont pour objectif de déterminer l'enchaînement des faits qui a conduit à la *panne*. Remettre l'informatique attaquée en ordre de marche ne suffit pas. Il faut éviter qu'une telle mésaventure se reproduise. L'enchaînement des faits identifié, il faut réunir l'ensemble des protagonistes de l'incident (auteurs de la négligence, équipes d'alerte et de réaction, responsables techniques et fonctionnels) pour procéder à une analyse à froid de l'incident. Cet exercice, inspiré des *retex* (RETour d'EXpérience) militaires, n'a pas pour but de désigner des coupables, mais de mieux comprendre ce qui a mené à l'incident et l'incident lui-même pour y ajuster les réponses de l'entreprise. Afin qu'il se déroule dans les meilleures conditions, il doit être animé par un *bureau de la confiance* pour que tout soit dit. Selon le degré de confiance régnant dans l'entreprise, il peut être effectué sous forme de table ronde unique ou de rencontres séparées. Si un tel bureau n'existe pas, la conduite du *retex* peut être confiée à un tiers de confiance. Notons cependant que cette séance doit être conduite avec un panel d'experts assez large pour que les déclarations de l'un ou de l'autre cherchant à se disculper ne créent pas de confusion. La réactivité de ce bureau en cas d'incident indiquera son efficacité. S'il est efficace et que personne ne vient y avouer de négligence lorsque l'entreprise doit batailler dans le cyberespace, il est alors vraisemblable qu'elle affronte une attaque. Une fois les entretiens terminés, ce même bureau devra rédiger des recommandations pertinentes puisque fondées sur les déclarations des protagonistes de l'affaire. Elles seront tout d'abord soumises aux responsables de la sécurité. Puis, si ces derniers l'estiment judicieux, ils diffuseront à toutes les personnes concernées par la réitération d'un tel incident les éléments suivants :

- failles exploitées ;
- mode opératoire choisi par l'auteur ;
- conséquences pour l'entreprise ;
- nouvelles mesures à prendre.

En sachant ce qui s'est passé, les employés seront davantage impliqués dans la sécurité et la pérennité de leur outil de travail que si la direction leur diffusait de manière abrupte une liste de nouvelles mesures à prendre.

1.3 éviter la propagation

Remettre l'informatique en ordre de marche, éviter la réitération de l'incident par la diffusion des éléments *supra* ne saurait suffire. Ne pas diffuser ces éléments aux victimes potentielles laisse l'avantage à l'attaquant, un mode opératoire secret pouvant être réutilisé contre une autre cible. N'oublions pas qu'un attaquant, comme toute autre personne, cherche à *rentabiliser* ce qu'il a élaboré. Il le réutilisera jusqu'à en trouver un meilleur ou estimer qu'il n'est plus efficace.

Il n'est pas question de dévoiler à tous, tant le mode opératoire de l'attaque que les détails de l'organisation interne de l'entreprise, une large diffusion de ces éléments revenant à prévenir l'attaquant qu'il a été détecté et à exposer ses failles. Il est néanmoins judicieux de l'exposer à des partenaires de confiance : comment a-t-il procédé, par où s'est-il introduit dans le réseau de l'entreprise, comment a-t-il camouflé son intrusion, quel cheminement a-t-il suivi, quel était son but estimé, quelles erreurs a-t-il commises et comment a-t-il été décelé ? En agissant ainsi, l'entreprise met en garde les cibles potentielles contre ce type d'attaque, charge à elles de s'en prémunir. Ces éléments doivent être diffusés tant aux sous-traitants qu'aux co-traitants, car leur sécurité participe à celle de l'entreprise pour laquelle ils travaillent.

Cette alerte peut être utilement transmise à des experts (CERT *Computer Emergency Response Team*, CSIRT *Computer Security Incident Response Team*, ANSSI et ses relais locaux OZSSI *Observatoire Zonal de la SSI*) et au réseau consulaire (CCI *Chambre de Commerce et d'Industrie*). L'intérêt de la transmission aux experts est d'obtenir une analyse précise des faits puis des recommandations pertinentes et à jour. Quant au réseau consulaire, il peut diffuser l'alerte de manière rapide et descendante : la gendarmerie s'appuie actuellement sur les CCI pour transmettre aux commerces des alertes spécifiques (braquages, etc.). Les forces de sécurité méritent également d'être avisées. Les mettre au courant est utile car elles ont aussi un rôle de prévention et de conseil. Cela devient indispensable lorsque l'attaque est avérée et que l'entreprise veut porter plainte. Dans ce dernier cas, il faut les prévenir des opérations techniques que l'entreprise compte réaliser afin d'éviter qu'un acte effectué en toute bonne foi nuise à l'enquête.

évoquer la plainte amène tout naturellement à se poser la question des modalités de réparation des dommages subis par l'entreprise.

2 Obtenir réparation

Les premières réactions qui permettent de limiter les dégâts occasionnés par un incident touchant l'une des informatiques de l'entreprise sont cruciales. Indispensables, elles ne peuvent cependant suffire : une fois l'incident réglé, l'entreprise

doit définir sa ligne de conduite. Estime-t-elle que cela fait partie des mauvais moments inéluctables ou qu'une réparation lui est due ? Si elle estime qu'une réparation lui est due, deux voies s'offrent à elle : celle de l'assurance et celle de l'action judiciaire.

2.1 La voie de l'assurance

L'indemnisation par l'assurance est de plus en plus débattue. Nouveauté pour la profession en France, elle pose plusieurs questions. L'assureur n'est pas qu'un simple payeur : il doit évaluer les risques, prodiguer des conseils de sécurité, estimer le montant des dégâts pour fixer la prime d'assurance. Ce n'est qu'en fin de processus que vient l'indemnisation, une fois le dommage constaté. Avant de détailler chacun de ces points, précisons que les actuelles assurances couvrent très peu, voire pas du tout, le risque cyber. Seules certaines de ses conséquences peuvent être couvertes par la responsabilité civile classique. Autre point à savoir, les polices d'assurance dédiées aux risques cyber couvrent généralement les :

- reconstitutions des données à partir de leurs sauvegardes ;
- frais d'investigation et de décontamination suite à l'action d'un maliciel ;
- pertes d'exploitation et frais complémentaires consécutifs ;
- frais de notification ;
- frais d'expertise et de procédure (pour prouver sa bonne foi) ;
- pertes d'image et d'e-réputation.

L'offre ne couvre donc pas encore tout l'environnement d'un cyber-incident. Terrain à explorer, il n'est cependant pas certain que les offres futures correspondent à la demande des entreprises.

L'évaluation des risques C'est la première question posée. Comment un assureur peut-il sérieusement évaluer les risques d'attaque informatique contre une entreprise ? De plus, les experts sont peu nombreux, assez circonspects quand il s'agit de recenser de manière exhaustive les risques et parfois d'avis opposés. Quels assureurs disposent de suffisamment d'experts pour évaluer ces risques ? Qui est en mesure d'affirmer qu'une attaque informatique ne peut avoir de répercussions sur tel domaine de l'entreprise ? Pour accomplir cette mission il faut d'importants moyens (dont des experts des domaines concernés) qui ne peuvent être déployés par tous les cabinets d'assurance au profit de toutes les entreprises. Seules les grandes entreprises pourront s'offrir une telle évaluation, les PME estimant vraisemblablement son coût prohibitif. Mais encore faut-il que les entreprises connaissent les risques auxquels elles s'exposent après une cyber attaque. Arrêt ou retard de la production, dégradation de sa qualité, accidents du travail causés par le dysfonctionnement d'un robot, la palette est vaste. Quel assureur les couvrira tous ? L'entreprise devra alors se focaliser sur les plus probables et les plus handicapants ou destructeurs.

Les conseils, les dommages couverts et la fixation de la prime Quels experts prodigueront quels conseils de sécurité ? Sur quelle base ? Dans un premier temps, l'utilisation des guides pratiques et techniques de l'ANSSI ou du CERT-FR est vraisemblable. D'autres ingrédients classiques de sécurité sont également utiles pour réduire les chances de succès d'une attaque ou en minimiser les effets, tous n'étant pas cités ici :

- cartographier informatiques et réseaux, y compris leurs points d'entrée ;
- établir et tester des plans de continuité et de reprise d'activité ;
- définir une PSSI appliquée à toute l'entreprise donc à tous ses employés ;
- sauvegarde régulière des données et configurations éprouvées des machines ;
- mise en place d'une politique de supervision et de remontée d'incidents.

Ces éléments purement techniques et préventifs, indispensables à la sécurisation de l'entreprise, ne peuvent suffire à la rendre inviolable. Aucun expert ni cabinet de conseil ne certifiera l'efficacité totale de ses logiciels ou recommandations. Les contrats d'assurance seront fondés sur des auto évaluations, avec tous les risques de refus de paiement dus aux erreurs de déclarations que ce système comporte. Considérons cette étape résolue, il reste à fixer les dommages couverts et le montant de la prime d'assurance. La difficulté réside dans le fait qu'une attaque informatique se caractérise aussi par l'ambiguïté du dommage. Sur quelles bases en inclure certains et en exclure d'autres ? Comment évaluer une prime en cas d'atteinte à la réputation de l'entreprise, conséquence possible d'une attaque informatique ? L'absence de définition par l'entreprise des risques qu'elle encourt (cf. *supra*) nuit à une fixation réaliste des dommages couverts et de la prime d'assurance.

L'évaluation des dommages et le paiement des indemnités Une fois l'incident consommé, l'assureur doit estimer le montant des dommages. Exercice difficile, car est-il certain que l'attaque est finie ? Que se passera-t-il si l'attaque comporte plusieurs phases espacées d'un délai ? De plus, des dommages apparents peuvent dissimuler des dommages réels, bien plus profonds. Cette question du constat tardif des dommages soulève celle de leur identification. Soit une entreprise dont l'informatique périmétrique est piratée pour permettre l'accès à la salle informatique afin de copier l'intégralité des fichiers de R&D. Quels dommages l'assureur prendra-t-il en compte dans l'indemnisation de l'entreprise ? Soit une attaque visant plusieurs entreprises simultanément. Le risque systémique ne peut être négligé : l'assureur pourra-t-il dédommager tous ses clients ? Cette difficulté peut amener les assureurs à ne pas démarcher tous les clients potentiels par crainte d'une telle calamité.

Les questions en suspens Outre les questions que nous venons d'évoquer, d'autres points, tout aussi probables, demeurent en suspens.

Dans des domaines très techniques tels le transport ferroviaire, les experts judiciaires sont d'anciens employés des sociétés de chemin de fer. Quelle que soit leur probité personnelle, un doute quant à leur impartialité ne peut jamais être absent surtout en cas de litige. Cet endorecrutement étant fort probable, comment

d'éventuels conflits d'intérêts (entre leur attachement à leur ancien employeur et leur expertise chez un de ses concurrents par exemple) seront-ils évités ?

L'ambiguïté caractérise l'attaque informatique. Sur quoi l'expert de l'assureur se basera-t-il pour accorder ou refuser l'indemnisation ? Sera-t-il certain qu'une cyber attaque n'est pas un incident déguisé en attaque à des fins lucratives ?

2.2 La voie judiciaire

Choisir la voie de l'assurance n'interdit pas de choisir la voie judiciaire. Deux possibilités s'ouvrent à l'entreprise : agir au civil ou au pénal. Ces actions ne sont pas exclusives l'une de l'autre, puisque selon l'article 3 du Code de procédure pénale *l'action civile peut être exercée en même temps que l'action publique et devant la même juridiction*, étant entendu que l'article 4 précise qu'*il est sursis au jugement de cette action [civile] tant qu'il n'a pas été prononcé définitivement sur l'action publique lorsque celle-ci a été mise en mouvement.* Ce qui signifie que la décision du juge civil peut être prononcée avant la décision pénale, sauf lorsqu'une action civile est engagée en réparation du préjudice direct résultant de l'infraction pour laquelle le juge pénal est saisi. Le choix reste libre, sachant qu'une fois la voie civile choisie, il n'est plus possible de porter l'affaire au pénal (art. 5).

La question de la voie à choisir est fondamentale : opter pour le civil interdit tout retour en arrière. Selon certains juristes, ce point ne constitue nullement un problème car l'action civile peut être plus rapide que l'action pénale. Afin d'aider l'entreprise à décider si la voie pénale est opportune, une aide à la décision sur l'opportunité de la plainte est disponible sur le site internet de l'auteur [3].

Arrêtons-nous d'abord sur la voie civile.

Elle est à privilégier lorsque la victime souhaite obtenir uniquement réparation du préjudice subi. La différence entre les deux procès (civil et pénal) est que « le procès civil permet d'obtenir réparation du préjudice subi. Le procès pénal permet en plus d'obtenir la condamnation du coupable. » Les contentieux civils ont donc pour objet des rapports d'obligation entre personnes, quelle que soit la nature de ces rapports : privés, commerciaux, contractuels, etc.

Pour initier la procédure, il faut que la partie lésée ait un intérêt légitime à agir (art. 31 du Code de procédure civile) : elle doit pouvoir arguer que la partie qu'elle attaque a lésé un de ses intérêts. L'intérêt à agir doit être né et actuel et ne peut donc être hypothétique. Cependant, la jurisprudence admet les actions lorsque l'intérêt à agir est futur.

L'objet du litige est ce que revendiquent les parties en présence (dommages et intérêts, droit d'auteur, etc.). Il ne peut être modifié arbitrairement en cours de procédure (art. 4 du Code de procédure civile). La partie lésée doit prouver l'existence du préjudice avancé en fournissant des preuves au tribunal et à la partie adverse (art. 132 du Code de procédure civile). La fourniture de la preuve est libre. Une partie peut demander au juge la production d'une pièce détenue par un tiers

3. http ://informatiques-orphelines.fr/ page Guides et auto-tests.

(qui peut être la partie adverse). Si le juge estime qu'elle est utile à la procédure, il peut en ordonner la délivrance ou la production (art. 139). Mais cette demande doit préciser les faits dont elle entend rapporter la preuve (art. 222).

Afin de prouver les faits reprochés à l'auteur présumé, le plaignant peut faire appel à des agents de recherche privés (ARP). Encadrés par la Loi, astreints à une formation, leur profession est réglementée par l'état et chaque agent doit se voir attribuer un agrément par le CNAPS (Conseil National des Activités Privées de Sécurité) pour exercer. Leurs enquêtes font l'objet d'un rapport présentable à la justice qui, sous réserve de l'article 1353 du Code civil, sera pris en compte.

Le litige peut être réglé de différentes manières. Par la voie amiable (transaction, conciliation, médiation) si les parties s'accordent ou par la voie contentieuse en cas de désaccord. En matière civile, la phase d'instruction se nomme *mise en état*. Elle a pour but qu'un magistrat vérifie périodiquement que les parties procèdent à l'échange de leurs pièces et conclusions. La durée du procès n'est pas fixe, le prononcé du jugement peut ne pas être immédiat et être renvoyé (art. 450 du Code de procédure civile).

3 Est-ce punissable ? La voie pénale

Une fois l'*incident* qualifié d'attaque ou de négligence, l'entreprise a un aperçcu de l'élément moral de l'infraction. Si la négligence est écartée, il est indispensable de savoir si cette attaque (ou tentative) est pénalement répréhensible. Il est aussi utile de savoir si l'attaque subie se distingue du *bruit de fond* des attaques informatiques qui peuvent saturer les capacités de défense de l'entreprise mais dont la probabilité de succès est plutôt faible.

3.1 Le *bruit de fond* des attaques informatiques

Le développement du cyberespace et des maliciels a conduit à un accroissement du nombre d'attaques. Un marché de logiciels d'attaque s'est développé, conduisant à la multiplication des attaques lancées a priori, sans grande préparation. L'effet d'aubaine et la négligence des cibles sont de plus grands facteurs de succès que le sérieux de la préparation. Ces attaques constituent le *bruit de fond* des attaques informatiques : attaques de faible intensité lancées régulièrement, sans cible précise, avec des moyens peu performants au vu de l'état de l'art, qui réussissent si les cibles visées ont négligé la mise en Å“uvre des mesures de sécurité élémentaires. Comme l'entreprise doit distinguer les attaques des négligences, il lui faut distinguer les attaques du *bruit de fond*. Ce travail est ardu et s'apparente à celui du sous-marinier qui détecte un bruit ou un silence suspect au milieu du bruit d'un banc de crevettes.

Constituent ce *bruit de fond* les tentatives d'escroquerie en tout genre (*phishing*, faux courriels de détresse, spam, etc.), les *scans a priori* des sites internet des entreprises, ainsi que les tentatives d'intrusion dans les fonctions externes de l'entreprise (portail, service de courriel, etc.) souvent conduites par des robots.

Le succès de ces tentatives est généralement limité, voire nul, car les méthodes utilisées sont connues. Cependant, elles saturent les défenses de l'entreprise et dispersent l'attention de ses cyber-gardiens. étant des infractions à la loi pénale, ces attaques sont cependant punissables.

Il peut néanmoins être possible de discriminer une véritable attaque du *bruit de fond* si on prend la peine de définir une activité moyenne de l'entreprise, d'un employé-type. Cette définition ainsi que sa caractérisation (volume moyen de données échangées, horaires d'activité, etc.) permet d'automatiser les recherches et d'être alerté selon des critères discutables, mais qui présentent l'avantage d'éviter de chercher à l'aveuglette. Cette définition est nécessaire, car s'attaquer à tout ce qui compose le bruit de fond constitue une perte de temps pour l'entreprise. De cette manière, les équipes spécialisées peuvent se concentrer sur les véritables attaques.

Mais, de même qu'il existe des avions furtifs, se pose la question des attaques furtives, à savoir celles que l'attaquant camoufle dans le *bruit de fond*. L'épineuse question de l'obus et de la cuirasse trouve ici son adaptation cybernétique.

3.2 Quelles infractions sont punissables ?

Une fois certain de la réalité de l'attaque il convient de la qualifier précisément afin de savoir si elle est prévue et réprimée par un article du Code pénal. Les attaques informatiques possibles contre l'entreprise sont évoquées dans les livres II (crimes et délits contre les personnes) et III (crimes et délits contre les biens) du Code pénal.

Trois articles du Code pénal sont particulièrement utiles pour les atteintes aux personnes.

La protection de l'image individuelle est prévue par l'article 226-1 qui précise que la captation des paroles ou des images d'une personne, sans son consentement, est punie d'un an d'emprisonnement et de 45000 € d'amende. Cet article peut être invoqué contre un attaquant qui s'introduirait dans le réseau de l'entreprise et mettrait en marche les webcam des ordinateurs à l'insu de leurs utilisateurs ou capterait les images des caméras de surveillance, puis les utiliserait soit pour les divulguer, soit pour exercer des pressions sur les personnes filmées.

La porosité des frontières entre vie publique et vie privée est un fait de plus en plus prégnant qui se remarque également par la forte croissance des réseaux sociaux utilisés pour donner des nouvelles tant d'ordre privé que professionnel. Or, les alertes de sécurité relatives à ces réseaux sont nombreuses, notamment en ce qui concerne les vols d'identifiants, piratages de comptes, etc. Celui qui fait usage de l'identité d'un tiers en vue de troubler sa tranquillité ou celle d'autrui, ou de porter atteinte à son honneur ou à sa considération, risque un an d'emprisonnement et 15 000 € d'amende selon l'article 226-4-1.

Enfin, les correspondances électroniques sont protégées par l'article 226-15 qui réprime leur interception, détournement, utilisation ou divulgation.

Si ces articles protègent l'entreprise et ses employés, il ne faut cependant pas oublier que l'entreprise a des obligations, notamment lorsqu'elle met en œuvre des

traitements de données à caractère personnel. Le Code pénal l'oblige à respecter les prescriptions de la loi informatique et liberté (art. 226-16 et 17), interdit la collecte de données personnelles par des moyens frauduleux illicites ou déloyaux (art. 226-18) ou lorsque la personne s'y oppose (art. 226-19), impose de ne pas conserver ces données au-delà du délai légal (art. 226-20), et interdit tant de les détourner des fins pour lesquelles elles ont été collectées (art. 226-21) que de les divulguer (art. 226-22). Toutes ces infractions sont punies de 5 ans d'emprisonnement et de 300000 € d'amende, les personnes morales pouvant également être tenues pour responsables de ces faits.

Les atteintes aux personnes étant les plus sensationnelles, elles trouvent un fort écho dans la presse grand public. Toutefois, les atteintes aux biens peuvent être lourdes de conséquences, certaines d'entre elles figurant au livre III du Code pénal.

L'article 323-1 réprime l'accès ou le maintien frauduleux dans un système de traitement automatisé de données, quand bien même celles-ci ne sont ni modifiées ni supprimées, quel que soit le moyen utilisé pour le faire : intrusion directe par un collaborateur de l'entreprise ou une personne qui lui est extérieure, rebond, etc. Bien qu'il n'existe pas de définition légale du terme système de traitement automatisé de données, sa compréhension par les tribunaux est assez large pour englober le système d'information de l'entreprise et ses composants. Sont considérés comme des systèmes de traitement automatisé de données :

- les réseaux de télécommunications ;
- le réseau des cartes bancaires (Trib. Corr. Paris, 25 février 2000) ;
- un disque dur (Cour d'appel de Douai, 7 oct. 1992) ;
- un radiotéléphone (Cour d'appel de Paris, 18 nov. 1992) ;
- un ordinateur isolé.

Cette large définition met l'entreprise à l'abri des intrusions dans tout ou partie de son SI, que ces composants soient ou non reliés au réseau de l'entreprise.

L'article 323-2 réprime l'entrave au fonctionnement d'un système de traitement automatisé de données. Chaque réseau spécifique de l'entreprise devrait donc être considéré par les tribunaux comme un systèmes de traitement automatisé des données, et les altérations de leur fonctionnement seront réprimées par la loi.

L'article 323-3 réprime l'introduction frauduleuse de données dans un système de traitement automatisé de données, la suppression et la modification des données qu'il contient, ce qui protège l'intégrité des données du SI de l'entreprise.

Le Code pénal réprime donc tout ce qui perturbe la disponibilité et l'intégrité des données intégrées dans le SI de l'entreprise, mais leur confidentialité n'est pas expressément visée. Ce n'est pas une difficulté, car les juges estimeront sûrement qu'une telle divulgation de données est consécutive à une intrusion, prévue et réprimée par l'article 323-1. Seuls les puristes pourront regretter une telle absence.

La protection du SI de l'entreprise et de ses composants est effective grâce à ces articles. Cependant, le Code pénal dissuade d'agir d'une manière que certains qualifient abusivement de légitime défense. Prétextant l'invoquer, l'entreprise peut tomber sous le coup de l'article 323-3-1 : *le fait, sans motif légitime, d'importer, de*

détenir, d'offrir, de céder ou de mettre à disposition un équipement, un instrument, un programme informatique ou toute donnée conÃ§us ou spécialement adaptés pour commettre une ou plusieurs des infractions prévues par les articles 323-1 à 323-3 est puni des peines prévues respectivement pour l'infraction elle-même ou pour l'infraction la plus sévèrement réprimée.

De plus, pour le type d'infractions que nous venons d'étudier, l'article 323-6 prévoit que les personnes morales peuvent être condamnées.

3.3 La tentative est-elle également punissable ?

Il peut arriver que l'entreprise déjoue l'attaque prévue. Se réjouir de cette heureuse issue n'est pas une fin en soi car, dans certains cas, la tentative de commission de l'infraction est punissable, ce qui autorise l'entreprise à porter plainte. Définie par l'article 121-5 du Code pénal, la tentative est une infraction manquée contre la volonté de son auteur, soit qu'elle ait été contrée par la victime, soit qu'un grain de sable ait empêché son achèvement. Le *commencement d'exécution* signifie que l'auteur de l'attaque a dépassé le stade des actes préparatoires (installation, démarrage des logiciels, connexion au réseau, etc.). La jurisprudence le définit comme étant *l'acte ou les actes tendant directement à la consommation de l'infraction*. Il n'y a donc pas de désistement volontaire de la part de l'auteur (telle que pourrait l'être une déconnexion volontaire du réseau avant de lancer l'attaque) qui empêche toute poursuite devant les tribunaux. Dès lors qu'il y a tentative, son auteur est considéré comme auteur de l'infraction (art. 121-4) et passible des mêmes sanctions que l'auteur d'une infraction. Encore faut-il que la tentative de commission de l'infraction considérée soit expressément considérée comme punissable, ce qui n'est pas le cas par défaut. Pour les délits évoqués dans cet exposé, le Code pénal prévoit formellement la punition de leur tentative : l'article 226-5 réprime les tentatives d'atteintes aux personnes visées par les articles 226-1 à 226-4-1, et l'article 323-7 la tentative de commission des infractions visées par les articles 323-1 à 323-3-1. Pour ces deux types d'infractions, la responsabilité pénale des personnes morales est engagée.

3.4 L'indispensable sécurisation préalable

Au vu des articles cités, l'entreprise peut s'estimer suffisamment protégée par la loi. Cependant, des jurisprudences posent comme préalable à son indemnisation la sécurisation de son système d'information. Tel est le sens de la décision du TGI de Créteil du 23 avril 2013 relaxant une personne accusée de s'être introduite frauduleusement dans un système de traitement automatisé de données, de s'y être maintenue et d'y avoir soustrait des données, au motif que l'accusé avait récupéré *l'ensemble des documents sans aucun procédé de type hacking*.

4 La plainte

Une fois l'entreprise persuadée que les faits (ou tentatives) dont elle a été victime sont prévus et réprimés par le Code pénal, elle peut porter plainte pour que leur auteur soit puni et obtenir réparation des dommages subis. Elle peut aussi estimer trop risqué de porter plainte : déposer plainte est révélateur de faiblesses ou de défaillances dans la sécurité de l'entreprise et par là même préjudiciable à son image de marque, surtout si les faits viennent à être rendus publics.

4.1 L'intérêt de la plainte

Cette publicité des faits est possible, quand bien même la volonté du plaignant est de les conserver discrets le plus longtemps possible. Si les parties prenantes à la plainte (en nombre limité : victime, enquêteurs, magistrats) sont tenues ou ont intérêt au secret de l'enquête, on ne peut exclure une indiscrétion de l'une d'elles au cours d'une banale conversation devant une tierce personne. Ni sous-estimer le fait que dans le cadre de l'intérêt qu'une personne extérieure à l'enquête porte à l'entreprise, son attention soit attirée par le comportement de ses interlocuteurs qui fuiraient de façcon inhabituelle ou singulière des sujets précis. Il est logique que l'entreprise voie dans la publicité de la plainte un risque pour son image, car cette publicité prouverait qu'elle n'a pas su remédier à toutes ses vulnérabilités ou qu'elle n'a pas été en mesure de contrer correctement l'attaque dont elle a fait l'objet. Même si aucune entreprise ne peut s'enorgueillir d'être invulnérable, remarquons que dans la guerre de l'information actuelle, une telle révélation peut constituer non seulement un aveu de faiblesse, mais aussi le point de départ d'une attaque informationnelle qui sera difficilement contenue voire contrée.

Les fuites d'information ne sont cependant pas systématiques, et se focaliser sur elles aurait pour conséquence de tétaniser les entreprises qui n'oseraient plus jamais porter plainte. Cela encouragerait *ipso facto* les attaquants en leur donnant le sentiment qu'ils bénéficient d'une impunité. S'abstenir de porter plainte a pour effet de ne pas mettre l'action publique en mouvement. L'état ne déploie alors pas ses moyens pour rechercher et identifier l'auteur. Au vu des ambiguïtés de l'attaque informatique[4], on pourrait estimer que ce n'est pas grave, la probabilité de trouver l'auteur des faits étant minime. Mais faible probabilité ne signifie pas certitude de l'échec, et l'état améliore ses moyens de lutte contre toutes les formes de cyberdélinquance. Il est de l'intérêt des entreprises de rompre cet état de fait qui s'apparente à la loi du silence. Ainsi, l'état pourra identifier les auteurs, analyser et diffuser leur mode opératoire pour en éviter toute réitération. L'intérêt de l'entreprise est de mener une véritable analyse de risque à long terme avant d'aller ou non porter plainte. Le long terme s'explique par le fait qu'un gain immédiat (préservation de l'image par l'absence de plainte) peut être compensé par une perte à long terme (récidive fructueuse de l'attaque selon un mode

4. Ambiguïté de la source, des moyens, des dommages, de la finalité.

plus élaboré). Cette analyse de risque doit être rapide pour des questions de délai que nous détaillerons *infra*. Dans le cas où les faits sont publics, il est utile que l'entreprise porte plainte, une abstention de sa part pouvant être interprétée comme une preuve de désintérêt envers sa sécurité ou une totale méconnaissance des préjudices possibles.

4.2 Contre qui porter plainte ?

La plainte peut viser soit une personne identifiée, soit personne si l'entreprise n'a pu identifier formellement l'auteur des faits (plainte contre X). En tout état de cause, l'absence d'auteur identifié ne doit pas faire obstacle au dépôt de plainte.

La personne visée par la plainte n'est pas obligatoirement l'auteur de l'infraction. Si l'idéal est d'identifier puis dénoncer formellement l'auteur des faits, cela ne correspond pas toujours à la réalité et l'entreprise peut n'avoir identifié que des *seconds couteaux*. Cette absence d'identification n'est pas importante, car c'est aux enquêteurs qu'il appartiendra d'établir formellement les faits et d'identifier les auteurs, puis aux magistrats de déterminer les responsabilités de chacun. Plusieurs personnes peuvent avoir pris part aux faits délictueux :

- l'auteur matériel commet les actes de l'infraction considérée ;
- le coauteur participe matériellement à l'action aux côtés de l'auteur principal et encourt donc les mêmes peines que l'auteur principal de l'infraction ;
- le complice fournit aide ou assistance dans la préparation ou la consommation d'une infraction (l'instigateur est considéré comme complice).

Il convient d'agir avec prudence lorsqu'on vise une personne dans une plainte, car dans le cas où l'absence d'éléments viendrait infirmer les accusations portées, la personne visée pourrait s'estimer victime d'une dénonciation calomnieuse et porter plainte sur le fondement de l'article 226-10 du Code pénal. Les personnes morales peuvent aussi être déclarées pénalement responsables de cette infraction.

4.3 Quels faits viser dans la plainte ?

La plainte n'étant pas une simple demande d'assistance de l'état par le biais de ses enquêteurs et des magistrats sur le fondement d'un fait plus ou moins précis, il est nécessaire que l'entreprise explique, au moins succinctement, dans quelle mesure l'attaquant lui a porté préjudice. Il faut que les faits soient pénalement répréhensibles, sinon la plainte sera sans effet. Elle peut viser une attaque délibérée. Elle peut aussi, si les conséquences de l'acte reproché ont eu pour résultat le décès ou l'incapacité temporaire de travail d'un employé viser une faute d'imprudence, de négligence ou de manquement à une obligation de sécurité telle que le prévoit l'article 121-3 du Code pénal. Dans ce cas également, les personnes morales peuvent être tenues responsables de ce type d'infractions.

Pour les intrusions, altérations, modifications, etc. dans des systèmes de traitement automatisés de données la tentative est punissable. L'entreprise est alors fondée à porter plainte pour des tentatives de commission de ces infractions. Il

n'est nullement besoin d'estimer un préjudice pour porter plainte. L'infraction peut être constituée sans qu'il y ait préjudice comme c'est le cas des faits visés par l'article 323-1 du Code pénal. Dans cet article, le seul fait de s'introduire dans le système de traitement automatisé des données est constitutif de l'infraction.

4.4 Les délais pour porter plainte

Une fois l'attaque constatée, il est nécessaire de porter rapidement plainte pour deux raisons principales.

La première est que la rapidité du dépôt de plainte permet de sauvegarder les logs de connexion. Les proxys internet les conservent pendant environ 24 h, cette durée étant de 3 mois pour les logs d'IP chez beaucoup de FAI. Selon la Directive Européenne 2006/24/CE du 15 mars 2006, les données doivent être conservées pendant une période allant de six mois à deux ans, le comité des libertés civiles justice et affaires intérieures du parlement européen ayant pour sa part recommandé de limiter la durée de conservation des données à un an. Pour ajouter un peu de complexité à cette affaire, la France demande que *les données à caractère personnel doivent être conservées pendant une durée qui n'excède pas la durée nécessaire aux finalités pour lesquelles elles sont collectées et traitées.* Face à cette incertitude, il est alors prudent de porter plainte au plus vite afin d'éviter un dépérissement des traces conservées.

La rapidité est aussi de mise pour éviter la prescription de l'action publique. Les infractions citées dans cet article constituent des délits prescrits trois ans après leur date de commission (article 8 du Code de procédure pénale), ce qui signifie que trois ans après la date de commission de l'infraction, l'action publique ne peut être mise en mouvement. Cette situation peut paraître paradoxale, car une des caractéristiques maintes fois évoquée du cyber espace est son ambiguïté. Comment peut-on connaître avec certitude la date de commission d'une infraction dans un espace où l'ambiguïté règne ? La solution idéale serait alors de faire débuter la prescription, pour les infractions réalisées à l'aide ou via le cyberespace, non pas à la date de leur commission mais à la date à laquelle elles sont révélées, comme c'est le cas en matière d'abus de biens sociaux (Crim., arrêt du 10 août 1981). à côté de cette solution qui provoquerait sûrement une levée massive de boucliers, on peut estimer que la prescription débute, dans le cas de l'entrave au fonctionnement d'un système de traitement automatisé de données (prévue par l'article 323-2) au jour de la dernière entrave constatée. De même, en cas de suppression ou de modification frauduleuse de données (art. 323-3), elle débuterait le jour de la dernière modification. Ces propositions, qui ne constituent pas un point de doctrine, doivent être validées par une juridiction et n'ont pas l'autorité de la jurisprudence.

4.5 Les précautions

Pour que la plainte soit recevable et produise son effet, quelques précautions doivent être prises.

Il faut que les faits ne soient pas prescrits et que l'affaire n'ait pas été déjà jugée. Ainsi, une entreprise déboutée ne peut de nouveau porter plainte à propos des faits pour lesquels elle a déjà agi, vainement, en justice.

L'entreprise ne doit pas être complice des faits qu'elle reproche à son agresseur. Aux termes de l'article 121-7 du Code pénal, *est complice d'un crime ou d'un délit la personne qui sciemment, par aide ou assistance, en a facilité la préparation ou la consommation. Est également complice la personne qui par don, promesse, menace, ordre, abus d'autorité ou de pouvoir aura provoqué à une infraction ou donné des instructions pour la commettre.* Cette question de la complicité peut aussi se poser en d'autres termes : en ne portant pas plainte, l'entreprise est-elle complice de l'attaquant ? Non, puisque la complicité s'estime au moment de la commission de l'infraction. De plus, l'article 434-1 qui pénalise la non-dénonciation de crimes en cours n'est pas applicable dans les cas qui nous intéressent, puisque nous traitons ici des délits et non des crimes.

L'entreprise ne doit pas avoir incité l'un de ses employés à commettre la faute visée par la plainte, et il lui est conseillé de s'abstenir de tenter de résoudre d'abord l'enquête par elle-même. Pour ce dernier cas de figure, l'exemple calamiteux de la récente affaire d'espionnage chez Renault devrait avoir des vertus pédagogiques, même si l'exemple ne concerne pas l'informatique. Enfin, pour laisser aux enquêteurs des chances de réussir leur enquête, il ne faut pas contacter les auteurs identifiés pour les avertir des intentions de l'entreprise, et conserver les preuves de l'attaque.

4.6 Les protagonistes de la plainte

Dans les cas où le préjudice vise l'entreprise, c'est à elle de porter plainte par le biais d'un de ses employés dûment mandaté. Cependant, dans les cas où l'intimité de la vie d'autrui est l'objet d'une attaque (art. 226-1 et 226-2 du Code pénal), c'est à *la victime, [...] son représentant légal ou [...] ses ayants droit* de le faire en application de l'article 226-6 du même code.

La plainte peut être déposée de différentes manières, sans qu'il y ait de gage de plus grande efficacité de l'une ou de l'autre. Elle peut être déposée dans toute brigade de gendarmerie ou commissariat de police, selon les termes de l'article 15-3 du Code de procédure pénale : *La police judiciaire est tenue de recevoir les plaintes déposées par les victimes d'infractions à la loi pénale et de les transmettre, le cas échéant, au service ou à l'unité de police judiciaire territorialement compétent.* Le terme de *police judiciaire* est pris dans son sens générique tel que défini par l'article 15 du Code de procédure pénale, et non dans celui restreint de service de la police nationale. Il va de soi que l'entretien de bonnes relations entre les enquêteurs et l'entreprise est un gage de rapidité de la réception de la plainte et de sa qualité. Si l'entreprise choisit cette solution, il faut qu'elle garde en tête le fait que la personne à laquelle elle s'adressera sera très vraisemblablement un généraliste peu au fait de l'activité de l'entreprise. Le risque existe qu'en donnant moult détails techniques, l'enquêteur décroche et qu'un dialogue de sourds s'instaure. Il est

donc recommandé de décrire les faits d'une manière généraliste, quitte à ce que la personne recevant la plainte la transmette à un spécialiste qui pourra, cette fois, établir un dialogue plus technique et donc plus fructueux avec le représentant de l'entreprise.

Elle peut aussi être adressée par lettre au procureur de la République qui, selon l'article 40 du Code de procédure pén ale, *reÃ§oit les plaintes et les dénonciations et apprécie la suite à leur donner*. Il peut engager des poursuites, une procédure alternative aux poursuites ou classer sans suite (art. 40-1), le plaignant étant tenu au courant des suites données à sa plainte (art. 40-2). Le classement sans suite peut être contesté par le plaignant en formant un recours devant le procureur général de la cour d'appel dont dépend le procureur de la République (article 40-3). Là aussi, le magistrat qui recevra sa lettre ne sera ni un spécialiste de l'informatique ni de l'entreprise, et l'abondance de détails techniques pertinents pour le rédacteur, sera obscur ou déstabilisant pour le lecteur. Enfin, on peut aussi se constituer partie civile par l'envoi d'une lettre au juge d'instruction, (art. 85). Là encore, les détails techniques ne doivent pas déstabiliser le lecteur.

Pour faciliter ces démarches, un *guide pratique du dépôt de plainte* est disponible sur le site de l'auteur [5]. évoquons la confidentialité de la plainte. Même si le Code de procédure pénale prévoit que l'instruction est secrète (art. 11), aucun mode de dépôt de plainte ne garantit la confidentialité de l'affaire, l'actualité l'illustre. Par conséquent, cette question doit être exposée clairement aux enquêteurs ou aux magistrats, en leur expliquant en quoi la publicité de l'affaire serait nuisible à l'entreprise, afin que les mesures de préservation du secret soient les plus efficaces possible.

La plainte ayant été déposée, un nouveau volet s'ouvre, celui de l'enquête judiciaire. Complexe mais obéissant à un formalisme strict, elle peut aussi avoir des incidences à l'étranger, la cybercriminalité étant souvent transfrontalière.

Après l'enquête vient la phase du procès pénal qui suit aussi un formalisme strict et que conclut le verdict. La sévérité des peines encourues et prononcées peut susciter un débat.

Ces deux phases de la lutte contre la cybercriminalité, hors du propos de cet article, méritent cependant des développements propres.

Références

1. Davadie, Philippe *L'entreprise, nouveaux défis cyber*. Economica, 2014.
2. Freyssinet, Eric *La cybercriminalité en mouvement*. Hermès/Lavoisier, 2012
3. Quemener, M., Charpenel, Y *Cybercriminalité, droit pénal appliqué*. Economica, 2010

5. http ://informatiques-orphelines.fr/ page Guides et auto-tests.

Comment accélérer la mise en mouvement des organisations au niveau Cyber Sécurité ou l'apport decisif d'une approche quantitative innovante

Gérard Gaudin

Abstract. Au vu des défis en sécurité auxquels sont confrontées aujourd'hui les entreprises à l'ère du numérique, la Cyber sécurité ne peut plus être la seule affaire des spécialistes. Une mobilisation globale au sein des entreprises devient de plus en plus nécessaire, et elle doit s'appuyer sur 2 axes : une Cyber sécurité plus opérationnelle et à l'efficacité mesurée quantitativement, une implication forte du management et de chaque utilisateur pour faire de la sécurité l'affaire de tous dans le cadre d'une force collective homogène focalisée sur un seul et même objectif. Ce papier décrit les moyens permettant à de grandes entreprises européennes d'avancer aujourd'hui concrètement dans ces 2 directions.

Keywords: Cyber Défense, indicateurs, Comex, "Team building"

1. Introduction

Malgré des initiatives tous azimuts depuis quelques années, la mise en mouvement des organisations et entreprises au niveau Cyber Défense et Sécurité Opérationnelle reste très insuffisante, les résultats constatés étant bien maigres et les frustrations souvent perceptibles au sein de la communauté SSI dans de nombreux pays. Mais de nouvelles démarches très innovantes de nature à modifier cette situation sont apparues récemment, démarches qui sont en cours de mise en œuvre au sein de certains membres de la communauté Européenne de Clubs R2GS. Ces démarches sont organisées autour des 2 axes principaux que sont :

- D'une part l'usage de référentiels standardisés de classification d'événements de sécurité et d'indicateurs opérationnels associés disposant de chiffres statistiques d'état de l'art et permettant un benchmarking précis du niveau de sécurité,
- D'autre part l'instauration d'un meilleur dialogue avec le management via une sensibilisation efficace préalable des COMEX.

Ce 2ème axe peut par ailleurs largement bénéficier de l'approche quantitative novatrice apportée par le 1er axe.

2. 1er axe : évaluation de l'efficacité des mesures de sécurité, d'une approche qualitative à une approche quantitative en cyber sécurité

À ce jour dans le domaine de la cyber sécurité, l'énergie portée sur la mise en conformité réglementaire et technique a relégué à une échéance ultérieure l'évaluation de l'efficacité concrète des mesures de sécurité destinées à lutter contre les cyber menaces. Cette situation, atypique dans le monde des entreprises qui s'évaluent en permanence sous tous les angles, apparait aujourd'hui de plus en plus anormale si l'on songe à ce qui se passe dans d'autres disciplines du management, telles que la gestion, la qualité, la communication, la sûreté et la sécurité physique pour ne citer que quelques exemples. Cependant, pour rectifier cette situation, nous avons besoin de points de repère adaptés et précis, reliés d'une part aux risques IT des principaux

métiers de l'entreprise, et d'autre part aux référentiels IT généraux (tels que ISO 27002, Cobit DS5, NIST SP 800-53, US CAG Consensus Audit Guidelines).

Le but est de donner au management de l'entreprise un degré d'assurance élevé à travers l'évaluation continue de son niveau de sécurité réel, objectif qui va devenir de plus en plus nécessaire dans un contexte où :

- Le recours accru à des ressources IT externes ou personnelles génère de nouveaux défis,
- Les menaces cyber augmentent et évoluent constamment avec les aspects humains devenant cruciaux (car impliqués directement ou indirectement dans 70 % des incidents de sécurité liés à la malveillance),
- La plupart des attaques sont le fait de groupes organisés et motivés ayant des objectifs précis.

Dans cet environnement non stabilisé et de plus en plus complexe, le principal enjeu pour les organisations est de conserver la maîtrise de leurs systèmes d'information. Et le meilleur et seul moyen d'y répondre est de positionner sur le système d'information des outils de confiance, qui vont en quelque sorte « stabiliser une matière devenue friable » et agir comme des « barres de fer dans un béton armé ». Il est possible de lister les 3 principaux : gestion des identités et des accès (employés et partenaires), infrastructure à clés publiques (PKI), indicateurs opérationnels mesurant les comportements déviants et les écarts par rapport aux risques résiduels acceptés.

Concernant le 3ème moyen, il s'agit de mettre en œuvre des outils de détection et de mesure pouvant permettre de répondre aux questions suivantes :

- Les pratiques de l'organisation sont-elles conformes à sa politique de sécurité ou à son SMSI (Système de Management du Système d'Information) et avec quel niveau d'application ?
- Les mesures et les outils de sécurité existants sont-ils efficaces ?

Ces 2 sujets peuvent être traités de manière harmonisée et commune en s'appuyant sur un cadre de référence (constitué au moins d'un modèle de classification des événements de sécurité et d'un ensemble complet d'indicateurs opérationnels associés) ayant plusieurs caractéristiques :

- Attention égale portée aux incidents de sécurité et aux vulnérabilités et non-conformités,
- Approche de cyber sécurité centrée sur les événements de sécurité (accompagnant le movement de fond international actuel vers une surveillance et un contrôle continus),
- Correspondance claire avec les risques IT de l'organisation (via des profils de risques synthétiques par métier) et avec les points de repère des SMSI.

La standardisation à ce niveau est essentielle car un tel ensemble d'indicateurs (typiquement au nombre de 80 à 100) doit être partagé et validé par l'expérience terrain indispensable en matière de sécurité opérationnelle ; il doit par ailleurs faire l'objet d'une base commune publiée sur une large échelle pour rendre possible et stimuler l'élaboration de chiffres statistiques d'état de l'art au sein de la profession. De tels nouveaux standards peuvent permettre aux organisations de manière beaucoup plus aisée de :

- Se benchmarker elles-mêmes par rapport à ces chiffres d'état de l'art (éventuellement disponibles et accessibles à terme via de larges bases de données publiques ou privées) et d'évaluer précisément leur niveau de sécurité,
- Identifier tous les types d'événements de sécurité et les catégoriser de manière claire et compréhensible par tous les acteurs concernés, et être capable demain de les notifier précisément et sans ambiguité aux organismes officiels (pour

tous types d'intrusions et pour les informations associées – Cf. LPM et loi sur la protection des données à caractère personnel),
- Fournir aux compagnies d'assurance des chiffres nouveaux les autorisant à élaborer des polices d'assurance des cyber risques adaptées.

Plusieurs initiatives ou projets (standardisation ou recherche ou étatique ou à dominante utilisateur) ont été lancées visant à proposer des points de repère dans ce domaine, tels que :

- CYSPA (European Cyber Security Protection Alliance), alliance initiée par 17 organisations fondatrices afin d'accroître la capacité de l'industrie à se protéger des cyber attaques, et afin de soutenir la mise au point au niveau Européen de réglementations améliorant le niveau global de protection,
- CAPITAL (complement du projet CYSPA), une collaboration entre 9 organismes de recherche conduite par l'association EOS (European Organization for Security) et visant à coordonner les efforts de recherche Européens en cyber sécurité avec l'objectif de répondre à un programme "Research & Innovation" complet,
- EC3 (European Cyber Crime Centre) créé à Europol pour constituer le point focal dans la lutte contre le cyber crime en Europe, en accélérant les réactions en cas d'attaques sur Internet,
- Le service StaySafeOnline (permettant aux utilisateurs de notifier les incidents et ainsi de mieux connaître les cybermenaces), mis en oeuvre pau la US NCSA (US National Cybersecurity Alliance), dont la mission est d'éduquer et ainsi donner la capacité à une société numérique d'utiliser Internet de manière sûre à la maison, au travail et à l'école, en protégeant la technologie que les individus utilisent, les réseaux auxquels ils se connectent et nos biens numériques partagés,
- Le service Hackmageddon.com pour collecter les incidents de sécurité (uniquement sur une base volontaire) et pour offrir une image du cyber crime, de l'hacktivisme, de la cyber guerre et du cyber espionnage,
- ETSI ISG ISI (Industry Specification Group Information Security Indicators), une initiative récente au sein de l'ETSI (European Telecommunications Standard Institute) avec la création à l'automne 2011 d'une entité de standardisation dédiée.

L'initiative ETSI ISG ISI a été lancée avec l'objectif de répondre aux questions et défis mentionnés précédemment. Les 1ères spécifications ISG ISI ont été publiées en 2013 (GS ISI-002 concernant un modèle de classification, GS ISI-001-1 et GS ISI-001-2 concernant un ensemble d'indicateurs et leurs divers usages, GS ISI-004 concernant les principales métho-des de mise en œuvre de stratégies de détection)1, et elles sont aujourd'hui utilisées par plus de 50 grandes organisations et entreprises en Europe appartenant à tous les secteurs d'activités économiques, notamment ceux rassemblés au sein du réseau d'associations Club R2GS (France, Royaume-Uni, Allemagne, Italie et Luxembourg), une communauté d'utilisateurs spécialisée dans les domaines Cyber Défense et SIEM (Security Information and Event Management). L'entité ISG ISI a aussi noué des relations étroites avec les entités ISO JTC1 SC27 (IT Security techniques) et ITU-T SG17 Q4 (Cybersecurity) à travers des liaisons officielles. Et le monde de la standardisation internationale a reconnu qu'un manque important a été comblé par ces nouveaux standards très prometteurs.

Ces avancées substantielles et plusieurs années d'expérience européenne d'implémentations d'une telle démarche organisée autour d'indicateurs de sécurité opérationnels ont démontré que :

[1] See Wikipedia page:
https://en.wikipedia.org/wiki/Information_security_indicators

- Les progrès en cyber sécurité peuvent être accélérés grâce à la solidité de la démarche et à son alignement avec les diverses préoccupations du management :
    a. Commissaire aux comptes et auditeur (Niveau d'assurance amélioré)
    b. Direction d'activités (Meilleure conscience des risques et enjeux IT majeurs)
    c. Direction des opérations et de la production IT (Mieux utiliser et mieux mettre en perspective les apports des différents outils techniques et des équipes opérationnelles en SSI, créant ainsi plus de valeur)
    d. Direction de l'ingénierie IT (Évaluer en profondeur les possibilités des outils et solutions SIEM et VDS dans le cadre des appels d'offres)
    e. Management général et RSSI (Mesurer précisément l'amélioration des comportements utilisateurs en matière d'hygiène informatique dans un environnement de plus en plus menaçant, et se benchmarker sur la base de chiffres fiables)
    f. Ressources humaines et management (Mesurer l'attachement réel et la loyauté envers l'entreprise)

- Les échanges au sein du monde de la sécurité peuvent être notablement accentués (au-delà de ceux habituellement rencontrés au sein des associations existantes dans le domaine) :
    a. À travers la collecte et le partage d'expérience entre experts sur les méthodes de surveillance et de détection et les "use cases" associés pour les types majeurs d'incidents, de vulnérabilités et de non-conformités
    b. À travers demain des notifications légales plus aisées des incidents de sécurité aux autorités officielles.

3. 2ème axe : vers l'implication quotidienne réelle de l'ensemble des employés dans la lutte de l'entreprise contre les cyber-menaces

Cette vision quantitative du niveau de protection de l'entreprise et les progrès qu'elle permet s'avèrent cependant aujourd'hui très insuffisants si la démarche proposée reste l'affaire des seuls spécialistes sécurité et IT. A l'appui de ces dires, citons 2 chiffres clés qui illustrent à eux seuls les limites des approches trop exclusivement techniques et la nécessité d'étendre la démarche à l'ensemble des employés et du management de l'entreprise :

- 70 % de l'ensemble des incidents de sécurité sont rendus possible par l'exploitation de vulnérabilités logicielles critiques ou de non-conformités techniques ou comportementales de base (par exemple, les mots de passe faibles cités dans le dernier rapport Data Breach Investigation Report de la société Verizon),
- 70 % de l'ensemble des incidents de sécurité (d'origine interne ou externe) mettent en cause un comportement humain négligent, déviant ou malveillant au sein de l'entreprise (par exemple, la navigation sur Internet).

Nous entrons ici dans le domaine de la sensibilisation à la sécurité des utilisateurs, considéré comme un sujet en soi depuis plusieurs années au sein de la profession sécurité, mais qui a malheureusement à ce jour que trop faiblement répondu aux attentes et espoirs des entreprises. Car il est extrêmement rare de voir les comportements habituels des utilisateurs dans la société numérique s'infléchir sensiblement et se transformer en maillon fort dans la chaîne sécurité. Tout est

cependant loin d'avoir été exploré, et de nouvelles pistes prenant beaucoup plus en compte les aspects psychologiques sont aujourd'hui en cours de mise en œuvre.

Un changement radical de paradigme avec une refonte complète de la démarche Cyber sécurité est en réalité nécessaire. Ses composantes essentielles peuvent se résumer de la façon suivante :

- Impliquer le COMEX de l'entreprise en le faisant aborder les cyber risques comme les autres risques majeurs auxquels il a à faire face, afin de rendre l'ensemble du management de l'entreprise plus concerné,
- Sensibiliser et impliquer réellement ce management à la SSI en jouant sur des leviers nouveaux de motivation personnelle et en tentant de répondre à la question fondamentale du pourquoi (« Pourquoi devrais-je me préoccuper de Cyber sécurité ? »), avant celles du comment et du quoi,
- Montrer une volonté de changement de la part des responsables SSI pour mieux prendre en compte les besoins business et ceux relatifs à la facilité d'usage, et pour simplifier autant que faire se peut les processus IT associés.

Pour ce qui est de la **1ère composante**, les tendances actuelles en matière de numérisation accélérée de l'économie, de montée des risques en ligne avec les impacts et les coûts croissants des incidents (notamment en matière de réputation), de prise de conscience au niveau du monde politique et des dirigeants d'entreprise, font qu'il devient plus indispensable pour un RSSI de s'adresser à un COMEX pour aborder ces questions. Ce qui importe dans ce nouveau dialogue est de présenter clairement les enjeux (de préférence en termes quantitatifs et en s'appuyant en particulier sur les indicateurs évoqués précédemment) et les scénarii de maîtrise possible des cyber risques identifiés, avec une mise en lumière claire des coups de pouce attendus de la part des dirigeants. Et la façon de communiquer à ce sujet, exercice encore très inhabituel pour la plupart des RSSIs et en général effectué en temps très contraint, revêt un caractère primordial sous peine de voir se refermer une porte tout juste ouverte. Il y a des effets concrets possibles et souhaitables :

- Inscription de la cyber sécurité aux agendas des dirigeants et managers,
- Implication des dirigeants dans la valorisation des employés les plus participatifs en remontées d'incidents ou en suggestions d'améliorations des mesures existantes,
- Mise dans les mains des managers par leurs patrons de méthodes nouvelles pour les aider à associer leurs collaborateurs à la prise de conscience des enjeux et des changements nécessaires de comportements au quotidien (Team building) et pour les pousser à s'engager véritablement (Forme d'approche beaucoup plus « bottom-up » alignée sur les risques propres aux activités de l'équipe concernée).

Au même titre que la qualité promue au rang de cause d'entreprise il y a maintenant 25 ans, il n'y pas d'autre chemin pour faire de la cyber sécurité l'affaire de tous. Face à des groupes d'attaquants organisés et motivés, les entreprises pourront de moins en moins se permettre de mettre en place des défenses a minima et de ne pas faire appel à leur force collective. Des exemples récents ont montré ce qu'il pourrait en coûter à l'avenir aux entreprises et à leurs dirigeants jusqu'au plus haut niveau en cas de manquements par trop manifestes en la matière.

Pour ce qui est de la **2ème composante**, force est de constater le besoin patent de mise au point d'un « cocktail » gagnant de nature à mobiliser efficacement les utilisateurs, en comprenant mieux les ressorts psychologiques pouvant déclencher de réelles envies de modification de leurs comportements. Il existe essentiellement 4 leviers de motivation possibles :

- Augmenter sa réputation et sa visibilité au sein de l'entreprise en participant par exemple au développement d'une cyber communauté active sur les questions de Cyber défense,

- Profiter de l'acquisition de compétences et de pratiques nouvelles pour en faire bénéficier sa sphère familiale et privée et mieux se protéger à la maison,
- Se prémunir par ses attitudes contre des attaques ou des négligences pouvant affecter gravement ses propres activités,
- Améliorer à terme son employabilité en mettant en avant des compétences qui seront demain dans un monde numérique ouvert aussi importantes que la maîtrise de l'anglais ou de la bureautique.

Il s'agit de se comporter en professionnel maîtrisant mieux son environnement numérique et plus conscient de son rôle dans le collectif. Si ces motivations sont déclenchées et entretenues par l'approche de « Team building » évoquée précédemment, certains outils de sensibilisation et de formation actuels (notamment les plus avancés comme les « serious games ») peuvent alors pleinement remplir leurs objectifs de développement des pratiques d'hygiène informatique. Et la construction d'une cyber communauté interne à l'entreprise devient alors envisageable en encourageant les participants à partager et à renforcer ainsi l'adhésion aux programmes de formation et finalement à améliorer la résilience du système d'information de l'entreprise. Cette communauté doit être fortement animée par l'apport d'information nouvelle régulière sur les incidents apparaissant au sein de la profession, sur les progrès technologiques les plus récents et sur les avancées les plus marquantes. Un appui sur des réseaux sociaux d'entreprise est souhaitable. Ajoutons qu'un des effets de cette transformation est la mise en lumière du rôle clé et la valorisation résultante des experts et administrateurs sécurité (notamment SOC et CERT), les faisant considérer de ce fait comme un capital humain essentiel à sauvegarder plutôt que n'appartenant qu'à un simple centre de coûts. Leur motivation accrue est par ailleurs la garantie d'une bien meilleure application du "back-to-basics" dont il est si souvent fait mention au sein de la profession avec malheureusement encore souvent trop peu de résultats.

Enfin, un ensemble de 30 indicateurs (parmi les 94 disponibles dans le standard ETSI GS ISI-001) permet à l'entreprise de mesurer les progrès en matière de comportement des utilisateurs et de se benchmarker très précisément par rapport à l'état de l'art des meilleurs.

Pour ce qui est de la **3ème composante**, il s'agit de capitaliser sur le mouvement enclenché et les desideratas exprimés par le management en matière de solutions de sécurité plus adaptées aux exigences business. Plus précisément, il s'agit de solliciter des rencontres entre responsables cyber sécurité, responsables IT, assistants en maîtrise d'ouvrage en informatique applicative et managers pour œuvrer ensemble à une optimisation et une simplification des processus IT. On peut aborder dans ce contexte les divers types de « mauvaise sécurité » sous l'angle des conséquences autant que des coûts, le travail à effectuer consistant à identifier les plus courants, les plus pénalisants (pour l'agilité, l'usage, la crédibilité et l'image de la sécurité), et les plus coûteux. Les principaux types envisageables sont les suivants :

- Conséquences et coûts des processus et technologies inopérants ou peu efficients (engendrant des incidents résiduels non désirés et subis),
- Conséquences et coûts des processus et technologies nuisibles ou handicapants (pour le business) ou excessifs et non indispensables,
- Conséquences et coûts du savoir et des compétences clés non exploités.

Il faut cependant garder à l'esprit que « trop de sécurité » peut certes nuire fortement, mais que « pas assez de sécurité » peut quant à elle tuer une activité. Globalement, il est ainsi question ici de donner une image renouvelée et plus positive de la sécurité afin de la rendre plus acceptable aux yeux de tous. Et une fois de plus dans cette 3ème composante, il s'agit au moins autant d'actes forts de communication, d'animation et de management que de technique et d'expertise sécurité.

4. Conclusion

Au vu des défis en sécurité auxquels sont confrontées aujourd'hui les entreprises à l'ère du numérique, la Cyber défense ne peut plus être la seule affaire des spécialistes. Une mobilisation globale au sein des entreprises devient de plus en plus nécessaire, et elle doit s'appuyer sur 2 axes :

- Une Cyber sécurité plus opérationnelle et à l'efficacité mesurée quantitativement,
- Une implication forte du management et de chaque utilisateur pour faire de la sécurité l'affaire de tous dans le cadre d'une force collective homogène focalisée sur un seul et même objectif.

Des entreprises avancent aujourd'hui dans ces directions en mettant en œuvre les grandes lignes évoquées ci-dessus. La très bonne acceptation par les dirigeants et par le management en général de ces nouvelles orientations laissent beaucoup de motifs d'espoir quant à la capacité des entreprises à relever le défi auquel elles sont de plus en plus confrontées.

Catégorisation par objectifs de la visualization pour la sécurité

C. Humphries, N. Prigent, C. Bidan, and F. Majorczyk

[1] Inria, Équipe Cidre prenom.nom@inria.fr
[2] Supélec, Équipe Cidre prenom.nom@supelec.fr
[3] Supélec, Équipe Cidre prenom.nom@supelec.fr
[4] DGA-MI prenom.nom@intradef.gouv.fr

La visualisation est désormais une fonctionnalité fréquente dans les outils de sécurité (Fig.1) qui a été appliquée sur de nombreux types de données : événements réseaux, données système, analyse statique de binaires, analyse de la structure de *malware*, par exemple.

Dans le cadre de cet article, nous nous intéressons à la visualisation dans le domaine de la securite, et plus spécifiquement, la securite des reseaux. Nous nous concentrons donc sur les evenements reseau, et non pas sur l'analyse statique de binaires, ni sur la structure de malwares.

Dans son état actuel, la visualisation pour la sécurité des systèmes d'information est plus souvent le résultat de l'application expérimentale de techniques de visualisation venant d'autres domaines sur des données de sécurité montrant des problèmes similaires. Par exemple, une visualisation capable de bien afficher une hiérarchie sera utilisée pour les systèmes de fichiers et les adresses ; les coordonnées parallèles [5] et les nuages de points seront utilisés pour faire de la corrélation d'événements réseau ; les *sparklines* afficheront efficacement des métriques et les graphes de nœuds seront souvent utilisés pour les réseaux, à la fois physiques et sociaux.

Les exemples les plus marquants en visualisation pour la sécurité inspirés par d'autres domaines viennent des outils d'analyse biologique. Ainsi, Circos, utilisé initialement pour l'analyse de données génomiques, a été adapté pour la conscience de la situation [LAMF05,FA07] et l'analyse de communications par email. Nous pouvons également citer les *hive plots* [hiv13], utilisés en tant qu'alternative au graphe de noeuds pour représenter les transferts de protéines dans les bactéries, et qui ont été ré-appliquées pour la visualisation de calculs en mémoire distribuée [EW12].

En étudiant les différents outils de visualisation pour la sécurité, nous avons identifié trois catégories dépendant de l'objectif visé. En premier lieu, les outils de visualisation pour la **supervision** des serveurs ou des réseaux ont pour objectif de surveiller certaines métriques du système d'information en vue de détecter au plus tôt des anomalies. En second lieu, les outils de **fouille visuelle** permettent

5. Un graphe par coordonnées parallèles affiche un axe par champ de données, puis lie les variables par rapport à ces axes.

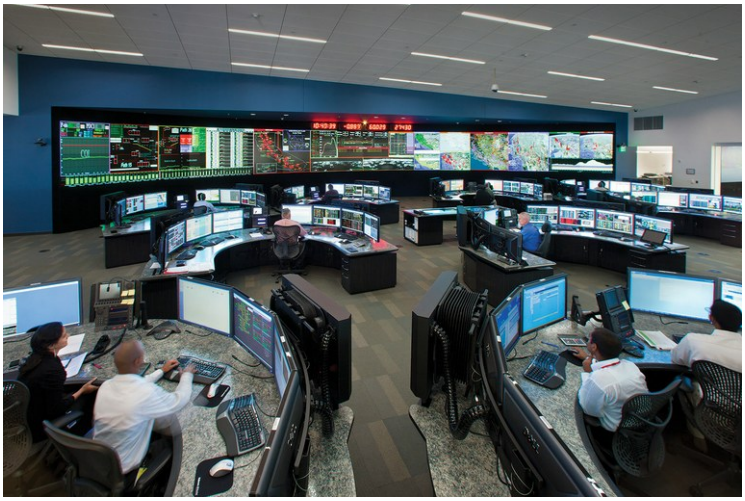


Figure 1. Le centre de controle du “California Independent System Operator", qui gère 80% de l'énergie de l'état.

d'explorer et d'analyser les données de sécurité pour expliquer les anomalies et identifier les scénarios d'attaque, ou encore localiser des intrusions qui auraient été manquées. Enfin, les outils de visualisation utilisés pour établir un **rapport** facilitent la compréhension des événements et de leurs implications. Ces trois catégories ne sont bien évidemment pas disjointes, certains outils de visualisation pouvant avoir plusieurs objectifs.

Cet article présente un etat de l'art des outils de visualisation pour la securite des systemes d'information et une classification des outils basee sur les objectifs. La section 1 présente les outils utilisés pour la surveillance de réseaux ou de systèmes. La section 2 présente les outils relatifs à la fouille visuelle de données liées à la sécurité. Les outils les moins nombreux sont ceux dédiés à la rédaction de rapport visuel ; ils sont présentés dans la section 3. Finalement, nous discutons notre choix de classification et concluons.

1 Visualisation pour la surveillance

La surveillance de systèmes est une des utilisations de la visualisation la plus commune en sécurité. Il s'agit de s'assurer que les systèmes fonctionnent de façon nominale. Dans ce but, la visualisation est utilisée pour détecter des changements alarmants ou des signes clairs d'intrusion. La bonne conception de l'outil de visualisation, notamment dans le choix des données et de leur représentation, permet de capturer un groupe spécifique de motifs. Les tableaux de bords sont des représentations adaptées pour observer des tendances ou repérer des valeurs

spécifiques. Cependant, des configurations visuelles plus complexes sont nécessaires pour capturer des motifs plus evasifs et des corrélations suspectes. De ce fait, les outils de surveillance sont généralement conçus pour résoudre un problème en particulier. Ils sont bien adaptés à la réalisation de cette tâche, mais ne sont généralement pas flexibles. Nous classons les outils de visualisation pour la supervision en trois catégories distinctes suivant leur objectif : la corrélation rapide, la décision rapide et le passage à l'échelle.

1.1 Corrélation rapide

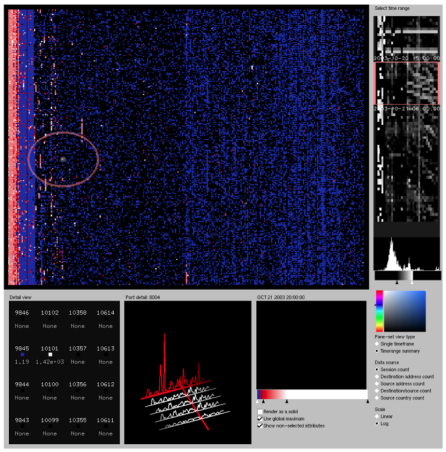


Figure 2. PortVis [MKL04], un example de visualisation en nuage de points.

Les outils de corrélation rapide permettent la recherche de motifs et de signes d'anomalies en faisant des corrélations ou en reconnaissant des motifs visuels.

Snortview [KO04] permet une corrélation visuelle simple pour gérer les faux positifs émis par une sonde de détection d'intrusions Snort [Sno13]. Les alertes sont tracées suivant leur date et leur adresse source ; leur type et leur priorité sont également affichés. Cette représentation a pour objectif de corréler la fréquence et le nombre d'alertes pour déterminer les vrais positifs dans le flot d'alertes émises par la sonde. Ceux-ci se démarquent des tendances et deviennent ainsi plus visibles, ce qui permet de se concentrer sur des nouveaux motifs d'attaques.

Colombe et al. proposent une méthode de définition de profils visuels statistiques [CS04] similaires en apparence aux visualisations à vue « galaxie », et dans lesquels les alertes sont des rangées entières de pixels, codés par couleur selon le temps. Chaque colonne est construite suivant des étiquettes associées aux alertes. Cette méthode permet de comparer les propriétés des alertes et de détecter des alertes inhabituelles ou inconnues.

PortVis [MKL04] est un autre outil qui utilise des visualisations en nuage de points. Différents événements de sécurité sont représentés par des pixels suivant le temps. Deux vues globales sont utilisées : la première utilise un axe de temps général (Fig. 2), l'autre un axe de temps avec une échelle horaire. Ensemble, ces vues permettent une détection d'événements périodiques. Des vues détaillées plus petites affichent les ports concernés et un graphe de leur historique avec

des paramètres de gradients de couleurs. La combinaison de ces multiples vues globales et détaillées de façon synchronisée permet à l'outil d'agir comme un filtre multi-modal ainsi qu'une exploration en profondeur.

Utilisant une visualisation en carte de points, IDSRainstorm [AL05,AC06] a été conçu pour afficher un grand nombre d'alertes (une journée entière d'alertes sur le réseau de GeorgiaTech). Une première visualisation représente par des points les alertes ; la couleur du point permet de déterminer la sévérité. L'axe vertical permet de déterminer à quel groupe d'adresses sont reliées les alertes alors que l'axe horizontal est un axe de temps représentant une journée. Un pixel agrège plusieurs alertes et adresses et représente l'alerte la plus sévère dans ce groupe pour cette période de temps. Pour inspecter une zone d'intérêt particulière, une seconde visualisation fournit une vue agrandie de la zone sélectionnée. Ce mode plus détaillé affiche chaque alerte sans les grouper par adresse, et affiche les connexions à partir d'adresses externes pour indiquer les alertes déclenchées par des hôtes externes.

IPMatrix [Koi05] propose également une visualisation basée sur deux nuages de points en coordination : le premier nuage pour les espaces A et B d'adressage au « niveau internet » et le second pour les espaces C et D au « niveau local ». Chaque nuage affiche les attaques détectées selon l'espace d'adressage en tant que pixels colorés suivant un code correspondant au type d'attaque. Ces points sont tracés sur une grille de cases d'agrégation affichant le nombre d'attaques par bloc d'adresses. Des histogrammes accompagnent chaque vue pour assister et ameliorer la lisibilite des chiffres des attaques. Une version tridimensionnelle de l'outil est également disponible : elle empile les representations des espaces et utilise des cartes de hauteur plutot que les cases d'agregation pour afficher les densites des attaques.

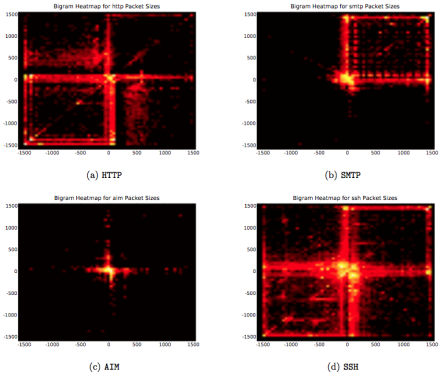


Figure 3. Quatre protocoles représentés avec des motifs visuels formés par des visualisations par nuages de points.

Avec l'objectif de détecter les scans, Irwin et al. utilisent aussi des nuages de points en 3D [IVR]. Leur outil affiche les connexions d'une adresses source à une adresse et un port de destination. Les connexions disparaissent avec le temps, ce qui permet de voir lesquelles sont les plus récentes. En utilisant cette combinaison de techniques, des millions de connexions peuvent être représentées en quasi temps-réel, ce qui permet de discerner visuellement les motifs de scan.

Dans le but de créer des visualisations reconnaissables caractérisant les protocoles réseau, Wright et al. construisent des motifs visuels [WMM03] en utilisant des nuages de points. De manière similaire aux cartes de chaleur, ils tracent la taille des packets sur un axe temporel (Fig. 3), et réussissent à montrer que des protocoles tels que SSH et HTTP sont caractérisés par des motifs très différents avec cette technique. De la même manière, se basant sur l'hypothèse que les protocoles au niveau internet ont des comportements temporels et des tailles de packets significatifs, Lian et al. utilisent des cartes de chaleur pour eux aussi fabriquer des motifs visuels [LMM10] pour différents protocoles. Chaque représentation trace un chemin basé sur la taille du paquet et le signe de sa direction : positif pour client vers serveur, négatif dans l'autre direction.

Contrairement aux outils présentés précédemment, VisFlowConnect [YA04] utilise des coordonnées parallèles pour faire apparaitre des corrélations en combinant la visualisation avec un curseur pour le temps et des filtres pour l'exploration de flux réseau. Cette visualisation facilite la corrélation multidimensionnelle de flux réseaux. Il faut noter cependant que la découverte de motifs intéressants dépend fortement de l'ordre d'arrangement des dimensions dans la représentation.

Visual Firewall [LC05] utilise une combinaison de plusieurs fenêtres de visualisation pour la détection et la reconnaissance de motifs en relation avec les pare-feu dans le but d'aider à leur configuration. Une vue en temps-réel des échanges permet d'afficher les paquets allant d'adresses externes à des ports locaux. Les paquets qui traversent le pare-feu génèrent un code de statut, alors que les paquets rejetés rebondissent. Une visualisation a coordonnees paralleles affiche l'historique de ces échanges et permet leur correlation avec le temps. Une seconde visualisation permet d'affiche des graphes de mesures sur les flux entrants et sortants. Enfin, une derniere visualisation a coordonnees paralleles permet de relier les connexions entre des sous-reseaux distants avec des types d'alertes, un axe vertical permettant d'indiquer la date de l'alerte et sa severite à l'aide de la coloration du point.

Muelder et al. [MMB05] proposent une méthodologie de visualisation ayant deux vues synchonisées pour déterminer des motifs dans le trafic réseau. Un graphe global affiche les relations entre les nœuds du réseau. La visualisation secondaire plus détaillée propose une représentation par traces de scan par rapport à deux espaces d'adressage, colorés suivant la date du scan, et combinés avec des « scalograms » (c.-à-d. des histogrammes mis à l'échelle) pour faciliter la comparaison de motifs. Cette configuration fournit un cycle rapide entre la vue globale et les vues détaillées pour comparer différentes motifs réseau typiques.

Comme nous le montrent les outils présentés précédemment, les nuages de points et les coordonnées parallèles ont été très utilisés pour permettre une corrélation rapide entre des événements. Quand des motifs et des signes d'anomalies sont détectés, l'inspection rapide des données est facilitée pour permettre une compréhension rapide de la situation.

1.2 Décision rapide

Avec l'objectif d'améliorer la connaissance de la situation, Livnat et al. présentent VisAlert [LAM+05,LAMF05,FA07], un système de visualisation radiale innovant qui vise à répondre rapidement à trois questions : que s'est-il passé, quand et où ? Des alertes sont tracées sur une tranche colorée radiale selon le type et se déplacent vers l'extérieur avec l'âge. Pour localiser l'alerte, l'espace à l'intérieur des anneaux héberge une visualisation spatiale ou organisationnelle à laquelle chaque alerte est liée. Par exemple, quand on visualise des intrusions réseau, cette visualisation centrale peut être une représentation en graphe de nœuds du réseau. On peut noter que cet outil a été utilisé dans d'autres domaine tels que la gestion des désastres et les appels aux urgences.

Pour fournir un point de départ aux analystes réseau, Overflow [GBT+09] propose une composition de trois visualisations. La première est une visualisation radiale affichant les différents éléments du réseau. Des lignes montrent les communications entrantes et sortantes entre. La seconde visualisation affiche la hiérarchie réseau détaillée pour l'élément sélectionné en utilisant un *tree map*, avec le même code couleur que l'affichage simplifié en anneaux. La dernière visualisation montre les groupes d'adresses IP pour chaque élément réseau.

Pour améliorer la réactivité et multiplier les options disponibles pour les opérateurs de sécurité, Hertzog propose une nouvelle stratégie de construction de visualisations [Her06]. La première étape est de réduire le nombre de données aux plus importantes pour réduire la charge de stockage, et ensuite de regrouper les données. Par exemple, les applications utilisées pour naviguer sur l'Internet peuvent être agrégées. Pour illustrer cette stratégie, deux visualisations sont présentées. La première est une visualisation interactive en coordonnées parallèles affichant les connexions d'un utilisateur par source, application, port et destination. Plusieurs nœuds sont regroupés pour simplifier le graphe, et des chemins spécifiques peuvent être isolés de façon interactive. La deuxième visualisation utilise un tracé bidimensionnel de l'utilisation de l'application selon le temps. Des segments d'utilisation sont colorés quand ils correspondent à des alertes. Des histogrammes affichent le niveau de ces alertes et fournissent des détails à la demande pour chaque segment.

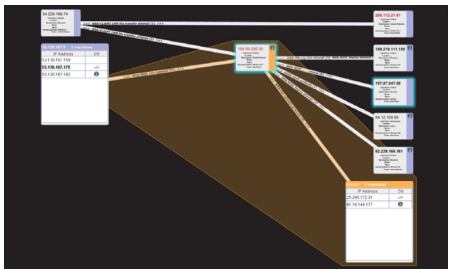


Figure 4. Représentation par cartes d'hôtes dans NIMBLE [RER+10].

Les administrateurs de systèmes ont parfois besoin de prendre des décisions rapidement. Rasmussen et al. proposent l'outil NIMBLE [RER+10] qui repose sur une visualisation en graphe de nœuds qui fait des recommandations basées

sur un apprentissage automatique. Des cartes sont affichées et reliées suivant le déclenchement d'alertes. Chaque carte représente un ou plusieurs hôtes, avec plus ou moins de détails, et les liens apportent ensuite une description sur le type de l'alerte concernant les deux noeuds (Fig. 4). Les explications sont affichées dans une liste à côté, et lors de la sélection d'un des éléments de la liste, les nœuds et les connexions concernés sont mis en évidence.

Pour permettre d'améliorer encore la connaissance de la situation, les outils de visualisation permettant la décision rapide dans la surveillance intègre souvent de nombreuses sources de données. Ils peuvent ainsi représenter de nombreux aspects de systèmes entiers d'hôtes et de processus. Ils utilisent souvent des graphes de nœuds et diminuent ainsi la distance mentale référent-référé. Ces solutions ont besoin de maintenir un état fonctionnel pour gérer des quantités de données grandissantes en temps-réel.

1.3 Passage à l'échelle

La visualisation en temps-réel est d'habitude seulement utilisée pour des ensembles de données filtrées. Pour améliorer la visualisation de données réseau en temps-réel, Daniel et al. proposent deux visualisations [DBWW10]. Le premier, nommé CLIQUE, est basé sur le projet LiveRac [6] et fournit une représentation tabulaire de données avec des colonnes pour chaque service et des rangées pour les utilisateurs. Chaque cellule contient initialement une visualisation de type *sparkline* qui peut être agrandie pour afficher plus de détails. La deuxième visualisation est un tracé radial similaire à un affichage radar conçu pour les grands affichages à haute résolution qui représente les flux réseau pour une période spécifique de temps. Chaque flux est représenté par un pixel coloré placé à un angle correspondant à sa date et avec une position radiale paramétrable.

Kintzel et al. [KFM11] proposent quatre visualisations pour surveiller des grands nombres d'hôtes, dans un ordre qui permet un filtrage et une focalisation progressifs. La première représentation basique utilise un graphe de type *sparkline* ou en barres affichant l'activité de chaque hôte sur 24 heures. La seconde, appelée ClockView, représente chaque hôte par un graphe radial, similaire à une horloge, affichant de nouveau l'activité sur 24 heures, mais en tant que *glyphe* permettant ainsi de comparer les différents hôtes. Pour la perception de la structure réseau, un graphe des communications peut être superposé à cette vue. Il est possible de se focaliser sur certains hôtes : la vue globale est alors remplacée par d'autres visualisations plus adaptées. L'activité est affichée en utilisant une matrice de pixels avec une granularité plus fine sur le temps, et les graphes « horloge » sont affichés en cordonnées parallèles pour permettre des corrélations. Une vue encore plus détaillée affiche une matrice de ports pour examiner les interactions entre deux machines spécifiques.

6. LiveRac est un outil de visualisation utilisé pour explorer des données à dimensions nombreuses pour un grand nombre d'hôtes.

Se basant sur la représentation en graphe de nœuds, Pearlman et al. utilisent des *glyphes* composés [PR] pour représenter les services en fonctionnement. Ici chaque noeud représente les services qu'il héberge avec un diagramme radial hierarchique. Ces services sont affichés de façon proportionelle, avec des anneaux externes représentant l'état le plus récent, ce qui permet de voir un historique à court-terme. Les hôtes simples ont des représentations simples, et peuvent être représentés en plus petit, ce qui permet de mettre des hôtes plus importants en évidence.

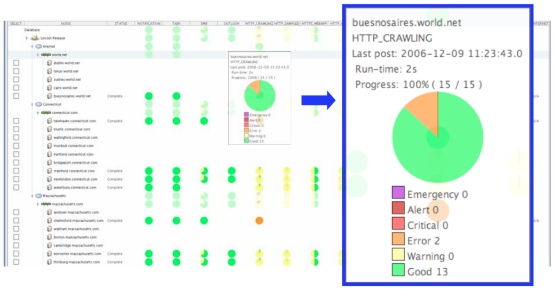


Figure 5. Utilisation de grilles de diagrammes circulaires pour visualiser des bancs d'essai de services [YFB+07].

Dans l'objectif de gérer et de surveiller des grands bancs d'essai d'hôtes qui hébergent de multiples services, Yu et al. proposent un outil de visualisation de bancs d'essai [YFB+] avec des arbres pour afficher la hiérarchie, des vues en graphe de nœuds et en matrices pour afficher les flux réseau, des chronologies pour afficher l'activité utilisateur, mais aussi une visualisation d'état d'appareils qui utilise des grilles de diagrammes circulaires multiples. Chaque diagramme représente l'état actuel du service et les proportions d'alertes, et fournit des détails sur demande.

Les outils de surveillance font une utilisation significative de visualisations permettant de percevoir des motifs (nuages de points, coordonnées parallèles), synchronisées et adaptées à la parallélisation et qui évoluent souvent en temps-réel.

2 Visualisation pour l'analyse

L'analyse de données est une étape nécessaire quand les outils de surveillance ont échoué ou quand aucune explication n'est disponible pour une anomalie. Les outils de visualisation adaptés à l'analyse permettent aux opérateurs de mieux comprendre les situations et les processus qui y ont mené. Analyser des données implique l'exploration de plusieurs configurations et plusieurs types de visualisations pour voir lesquels afficheront des résultats. L'analyse de données est générale guidée par un objectif spécifique tel que la recherche de tentatives d'intrusions répétées sur un système donné, ou encore la recherche de motifs ou signes d'activités potentiellement malveillantes

Les outils de visualisation adaptés à l'analyse utilisent des cycles de recherche similaires à ceux utilisés pour la surveillance avec cependant plus de contrôles dans la profondeur de recherche.

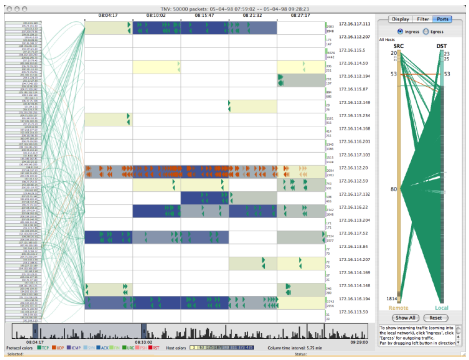


Figure 6. The Network Visualiser (TNV) [GL05] vise à offrir une image plus générale de captures de paquets réseau.

Des outils tels que « The Analyst's Notebook » [Ana] fournissent des visualisations adaptées à l'analyse, en utilisant des cartes pour localiser des données et un graphe de nœuds pour visualiser les liens entre ces données. En utilisant ces outils, un opérateur adoptera un cycle de raffinement entre chaque outil, en utilisant chaque outil comme un filtre pour les autres jusqu'à ce que l'information essentielle soit trouvée.

NVisionIP [LAL04] offre plusieurs vues pour explorer des données NetFlow. Au niveau de détails le plus large, une vue « galaxie » affiche les flux par sous-réseau et par hôte ; ces flux sont colorés suivant des groupes paramétrables. Quand une activité suspecte est visible, l'opérateur peut zoomer sur plusieurs vues plus petites qui comparent différents hôtes, puis sur une vue machine plus spécifique. Cette approche en profondeur est typique d'un outil de visualisation pour l'analyse. Deux articles complémentaires décrivent des extensions pour l'outil. Le premier [LSYN05] présente la capacité à enregistrer des chemins d'exploration visuelle et créer des règles correspondant aux motifs quand de nouvelles attaques sont découvertes. Le deuxième [Yur06] décrit la possibilité de comparer différents fichiers logs avec une vue des différences, des graphes linéaires pour représenter de densité de données, et des tracés de formes cherchent à mettre à profit les principes de perception *Gestalt*[7] afin de faciliter la reconnaissance de motifs dans les fichiers logs.

Au lieu de proposer des visualisations séquentielles pour explorer des données d'attaques Sybil dans des réseaux WiFi, Harrison et al. [HLW10] proposent un outil utilisant de multiples vues coordonnées. Chaque vue change de façon synchrone par rapport aux autres et agit comme un filtre pour l'interaction utilisateur. Pour l'analyse spatiale, la première vue propose un arrangement en graphe de nœuds du réseau. Pour l'analyse temporelle, un histogramme temporel affiche tous les événements sur une période donnée. La dernière vue est un nuage de points, pour visualiser des dimensions configurables à partir d'une analyse spectrale de données. En utilisant ces trois vues, l'utilisateur suit un processus de filtrage incrémental et trouve graduellement des points d'intérêt.

BGP Eye [Ran03] vise à aider la détection d'anomalies au niveau de BGP (Border Gateway Protocol) en utilisant quatre visualisations. La première visualisation est un graphe de nœuds qui affiche des événements déclenchés par des systèmes

7. La psychologie Gestallt décrit notre capacité à percevoir des liens et des groupes de formes dans une image avant de se concentrer sur celles-ci individuellement.

autonomes (AS) spécifiques. La deuxième utilise un arrangement spatial basé sur la distance du chemin avec les nœuds sources en bas, des nœuds puits en haut et tous les noeuds intermédiaires placés suivant leur distance. La troisième visualisation utilise un diagramme circulaire pour afficher les connexions entre les routeurs d'observation dans un anneau interne et les routeurs paires dans un anneau externe. La dernière vue est une association complexe de deux visualisations planaires : la première affiche les routeurs concernés, la seconde les statuts préfixés correspondants. Ceci permet de voir quels routeurs ont récemment vus des changements de chemins de routage.

D'autres outils de visualisation pour l'analyse ont adopté une approche différente de celle des cycles de recherche. Leur approche est basée sur une narration simple qui permet de montrer la progression des événements de façon détaillée.

TNV [GL05] vise à offrir une adaptation visuelle de l'outil d'inspection de paquets Wireshark pour obtenir une image plus générale de captures de paquets réseau. Une matrice centrale de visualisation affiche les paquets, annotés par direction et colorés par type de protocol. De chaque côté, les adresses sont listées et liées aux différents flux de paquets. À droite, une visualisation par coordonnées parallèles affiche les ports utilisés, et en dessous, un histogramme affiche l'évolution du trafic réseau qui permet le filtrage. Les filtres suivent une progression horizontale des paquets, ce qui rend la compréhension du trafic réseau plus facile.

Avec une approche similaire de circulation de gauche à droite, Portall [FMN05] permet à l'utilisateur d'explorer les processus client et serveur et la manière dont ils communiquent entre eux. Les clients sont listés sur la gauche, les serveurs sur la droite, et les liens entre processus sont représentées par un graphe de nœuds, affichés en tant que boites avec les détails du processus, notamment un histogramme d'activité de connexion.

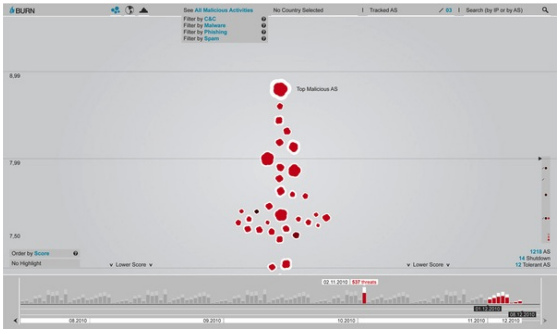


Figure 7.
BURN [RCDM+11] affiche et ordonne des systèmes autonomes par activité malveillante, en faisant une utilisation poussée d'animation et de transitions.

Roveta et al. propose BURN (*Baring Unknown Rogue Networks*) [RCDM+11] qui prend en entrée des alertes concernant des systèmes autonomes (AS). Cet outil utilise des techniques d'animation et des effets de transparence afin de faciliter

l'identification des comportements irréguliers (Fig 7). Les AS sont représentés par des bulles colorées qui sont plus ou moins animés et irrégulières en fonction du nombre d'alertes associées à chaque AS. Des *sparklines* permettent l'inspection rapide des activités et la détection de changements de comportement. En bas de la visualisation, un histogramme montre l'activité globale sur le temps et permet un filtrage temporel. Deux autres vues, une carte globale et une matrice d'activité permettent à la fois une inspection géographique de l'activité et de ses changements.

Comparés aux outils de surveillance, les outils d'analyse offrent une meilleure flexibilité, souvent sans point de départ ou configuration prédéfinis. Lors de la conception de ces outils, une réflexion profonde est nécessaire pour définir les possibilités de cycles d'exploration de données. L'ajout de transitions et d'animations aide à désigner les zones importantes et les modifications de données.

3 La visualisation pour le rapport

Dans certains cas, montrer le résultat visuel lui même peut suffire. Par exemple, quand une alerte est déclenchée par VisAlert, une capture de l'état de la vue pourra rapporter toute l'information nécessaire pour comprendre l'alerte. Quand des anomalies et des alertes sont trouvées pendant la surveillance, ou quand des scénarios d'attaques sont découverts en utilisant des outils d'analyse visuelle, il peut être difficile de communiquer l'idée dans sa globalité. Pour comprendre un scénario complet ou pour expliquer le processus entier d'exploration qui a mené un opérateur à trouver un motif, de nouveaux outils sont nécessaires qui aident à progressivement prendre des notes et construire un rapport compréhensible.

Les auteurs de FlowTag [LC06] partent du postulat que les analyses longues et compliquées de données réseau produisent souvent de mauvais rapports. Leur outil fournit du filtrage simple et de la corrélation par coordonnées parallèles mais surtout la capacité d'étiqueter des éléments intéressants et reliés (Fig 8). Cela facilite la gestion du processus d'analyse mais aussi le partage avec les autres et transforme ce processus en une tâche collaborative. Des données étiquetées facilitent aussi la production de rapports, avec la possibilité de regrouper les données liées.

Pour représenter l'organisation générale de réseaux et le charactère atteignable des différents éléments les composant, Williams et al. utilisent un *tree map* pour représenter les graphes d'attaques combinées [WLI07]. Avec cette technique, des chemins et des scénarios d'attaque potentiels peuvent être représentés de façon plus descriptive.

Alors qu'il pourrait être utilisé en tant qu'outil de surveillance pour l'activité malveillante et les alertes à échelle globale, EMBER [YRB10] est un outil de visualisation dont un des objectifs est la détection d'activités malveillantes et la visualisation d'alertes à grande échelle. Grâce à une configuration fixée de

visualisations classiques, il permet également de communiquer des fréquences d'attaque.

Pour mieux comprendre des attaques complexes comportant de multiples étapes, des visualisations complexes voire des combinaisons de celles-ci sont nécessaires. Yelizarov et al. présentent un outil [YG09] pour visualiser et détecter ces attaques. Positionné dans l'espace, des cylindres représentent des événements, colorés par type et dont la taille correspond à la sévérité. Ces cylindres sont connectés pour montrer les événements liés et séquentiels. Les événements suivent le même axe de temps horizontal sur plusieurs rangées. Ces cylindres sont ensuite reliés à un autre graphe, soit en coordonnées parallèles, soit un nuage de points, qui montre l'adresse source de l'événement.

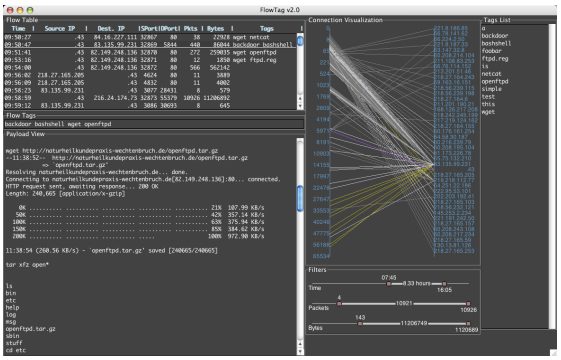


Figure 8. Flowtag [LC06] ajoute la possibilité de faire des annotations pendant ses analyses pour faciliter la production de rapports et leur partage.

En ajoutant aux outils de surveillance et d'analyse des possibilités de prise de note ou de collecte d'une grande quantité de données, les outils pour le rapport facilitent la communication des événements de sécurité qui ont eu lieu.

4 Discussion et conclusion

À notre connaissance, trois autres études ont proposé des taxonomies des outils de visualisation pour la sécurité [Kas06,SSG12,FDCN06]. La première étude [Kas06] classe ces outils suivant le type de données considérées en source (données brutes ou données de plus haut niveau provenant d'IDS, séparées en IDS par signature et IDS comportementaux) et suivant la caractéristique abstraite ou concrète de la visualisation choisie. Les différentes tâches que doivent remplir les outils de visualisation sont évoquées : détecter les activités malveillantes, déterminer les faux positifs, entraîner les IDS par apprentissage. Les deux premières tâches rentrent dans notre catégorie d'outils dédiés à la supervision. La troisième est externe à notre classification. La seconde étude [SSG12] est centrée sur la sécurité réseau et propose une classification par cas d'utilisation : surveillance d'hôtes ou de serveurs, surveillance des connexions entre le réseau interne et le réseau

externe, surveillance de l'activité sur les ports TCP, détection de motifs d'attaques, surveillance des comportements de routage. Les auteurs analysent pour une trentaine d'outils les différentes techniques de visualisation mises en œuvre et les sources de données considérées. Enfin, la troisième étude [FDCN06] propose une classification en deux dimensions des outils de visualisation pour la sécurité réseau. Les différents outils sont classés en fonction de la complexité de la visualisation (du mode texte aux représentations complexes) et de la taille du système représenté (d'une machine à un grand réseau).

Aucune de ces trois autres classifications ne met réellement en avant les objectifs des outils de communications. En ce sens, notre classification se distingue des autres, puisqu'elle repose précisément sur les trois grands objectifs d'un outil de visualisation : la supervision, la fouille visuelle et l'aide à la communication.

Les outils de visualisation en sécurité sont conçus soit pour surveiller des systèmes, soit pour faire des fouilles exploratoires de données de sécurité, soit pour rapporter des résultats. Pour atteindre leurs objectifs, ces outils utilisent différentes approches, en implémentant des combinations caractéristiques de visualisations et d'expériences utilisateur. La conception d'un outil de visualisation va ainsi dépendre à la fois des données d'entrées et de ses objectifs de plus haut niveau. Une majorité d'outils de visualisation pour la sécurité se sont concentrés sur la problématique de la surveillance de systèmes alors que peu d'entre eux se sont intéréssés à l'analyse ou l'aide à la communication. Nous pensons qu'il est important pour les concepteurs d'outils de visualisation de porter leurs efforts sur ces deux objectifs, notamment face à la quantité de données émises par les systèmes d'information de nos jours et à la difficulté de communiquer sur les évènements de sécurité.

Références

AC06. K Abdullah and J A Copeland. Tool Update : High Alarm Count Issues in IDS RainStorm. In *Proc. of VizSEC'06*, pages 129–136, 2006.

AL05. K Abdullah and C Lee. IDS RainStorm : Visualizing IDS Alarms. In *Proc. of VizSEC'05*, pages 1–10, 2005.

Ana. Assisted Analysis. Analyst ' s Notebook 8 Increase the depth of intelligence for effective resource utilization .

CS04. J B Colombe and G Stephens. Statistical Profiling and Visualization for Detection of Malicious Insider Attacks on Computer Networks. In *Proc. of VizSEC/DMSEC'04*, pages 138–142, 2004.

DBWW10. M Daniel, S Bohn, A Wynne, and A William. Real-Time Visualization of Network Behaviors for Situational Awareness. In *Proc. of VizSEC'10*, pages 79–90, 2010.

EW12. S Engle and S Whalen. Visualizing distributed memory computations with hive plots. In *Proc. of VizSEC '12*, pages 56–63, 2012.

FA07. S Foresti and J Agutter. VisAlert : From Idea to Product. In *Proc. of VizSEC'07*, pages 159–174, 2007.

FDCN06. Glenn A Fink, Vyas Duggirala, Ricardo Correa, and Chris North. Bridging the Host-Network Divide : Survey, Taxonomy, and Solution. In *Proc. of LISA '06 : 20th Large Installation System Administration Conference*, pages 247–262, 2006.

FMN05. G A Fink, P Muessig, and C North. Visual Correlation of Host Processes and Network Traffic. In *Proc. of VizSEC'05*, pages 11–19, 2005.

GBT+09. J Glanfield, S Brooks, T Taylor, D Paterson, C Smith, C Gates, and J Mchugh. OverFlow : An Overview Visualization for Network Analysis. In *Proc. of VizSEC'09*, pages 11–19, 2009.

GL05. J R Goodall and W G Lutters. Preserving the Big Picture : Visual Network Traffic Analysis with TNV. In *Proc. of VizSEC'05*, pages 47–54, 2005.

Her06. P Hertzog. Visualizations to Improve Reactivity Towards Security Incidents Inside Corporate Networks. In *Proc. of VizSEC'06*, pages 95–101, 2006.

hiv13. Hive Plots - Linear Layout for Network Visualization - Visually Interpreting Network Structure and Content Made Possible. Technical report, April 2013.

HLW10. L Harrison, A Lu, and W Wang. Interactive Detection of Network Anomalies via Coordinated Multiple Views. In *Proeedings of VizSEC'10*, pages 91–101, 2010.

IVR. B Irwin and J Van Riel. Using InetVis to Evaluate Snort and Bro Scan.

Kas06. R R Kasemri. *A Survey, Taxonomy, and Analysis of Network Security Visualization Techniques.* PhD thesis, 2006.

KFM11. C Kintzel, J Fuchs, and F Mansmann. Monitoring Large IP Spaces with ClockView. In *Proc. of VizSEC'11*, pages 2 :1–2 :10, 2011.

KO04. H Koike and K Ohno. SnortView : Visualization System of Snort Logs. In *Proc. of VizSEC/DMSEC'04*, pages 143–147, 2004.

Koi05. H Koike. Visualizing Cyber Attacks using IP Matrix. In *Proc. of VizSEC'05*, pages 91–98, 2005.

LAL04. K Lakkaraju, E S Ave, and A J Lee. NVisionIP : NetFlow Visualizations of System State for Security Situational Awareness. In *Proc. of VizSEC/DMSEC'04*, pages 65–72, 2004.

LAM+05. Y Livnat, J Agutter, S Moon, R F Erbacher, and S Foresti. A Visualization Paradigm for Network Intrusion Detection. In *Proc. of the Information Assurance Workshop (IAW'05)*, pages 92–99, 2005.

LAMF05. Y Livnat, J Agutter, S Moon, and S Foresti. Visual correlation for situational awareness. In *IEEE Symposium on Information Visualization (INFOVIS'05)*, pages 95–102, 2005.

LC05. C P Lee and J A Copeland. Visual Firewall : Real-time Network Security Monitor Workshop on Visualization for Computer Security. In *Proc. of VizSEC'05*, pages 129–136, 2005.

LC06. C P Lee and J A Copeland. FlowTag : A Collaborative Attack-Analysis, Reporting, and Sharing Tool for Security Researchers. In *Proc. of VizSEC'06*, pages 103–108, 2006.

LMM10. W Lian, F Monrose, and J Mchugh. Traffic Classification Using Visual Motifs : An Empirical Evaluation. 2010.

LSYN05. K Lakkaraju, A Slagell, W Yurcik, and S North. Closing-the-Loop in NVisionIP : Integrating Discovery and Search in Security Visualizations. In *Proc. of VizSEC'05*, pages 75–82, 2005.

MKL04. J Mcpherson, P Krystosk, and L Livermore. PortVis : A Tool for Port-Based Detection of Security Events. In *Proc. of VizSEC/DMSEC'04*, pages 73–81, 2004.

MMB05. C Muelder, K L Ma, and T Bartoletti. A Visualization Methodology for Characterization of Network Scans Workshop on Visualization for Computer Security. In *Proc. of VizSEC'05*, pages 29–38, 2005.

PR. J Pearlman and P Rheingans. Visualizing Network Security Events Using Compound Glyphs from a Service-Oriented.

Ran03. S Ranjan. BGP Eye : A New Visualization Tool for Real-time Detection and Analysis of BGP Anomalies. pages 81–90, 2003.

RCDM+11. F Roveta, G Caviglia, L Di Mario, S Zanero, F Maggi, and P Ciuccarelli. BURN : Baring Unknown Rogue Networks. In *Proc. of VizSEC'11*, pages 6 :1–6 :10, 2011.

RER+10. J Rasmussen, K Ehrlich, S Ross, S Kirk, D Gruen, and J Patterson. Nimble Cybersecurity Incident Management through Visualization and Defensible Recommendations. In *Proc. of VizSEC'10*, pages 102–113, 2010.

Sno13. Snort. Technical report, January 2013.

SSG12. H Shiravi, A Shiravi, and A A Ghorbani. A survey of visualization systems for network security. *IEEE Transactions on Visualization and Computer Graphics*, 18(8) :1313–1329, August 2012.

WLI07. L Williams, R Lippmann, and K Ingols. An Interactive Attack Graph Cascade and Reachability Display. In *Proc. of VizSEC'07*, pages 221–236, 2007.

WMM03. C V Wright, F Monrose, and G M Masson. Using Visual Motifs to Classify Encrypted Traffic. pages 41–50, 2003.

YA04. X Yin and E S Ave. VisFlowConnect : NetFlow Visualizations of Link Relationships for Security Situational Awareness Categories and Subject Descriptors. In *Proc. of VizSEC/DMSEC'04*, pages 26–34, 2004.

YFB+. T H Yu, B W Fuller, J H Bannick, L M Rossey, and R K Cunningham. Integrated Environment Management for Information Operations Testbeds. pages 67–83.

YFB+07. Tamara Yu, Benjamin Fuller, John Bannick, Lee Rossey, and Robert Cunningham. Integrated Environment Management for Information Operations Testbeds LARIAT Provides High-Fidelity Network Emulation and User Simulation. pages 1–25, 2007.

YG09. A Yelizarov and D Gamayunov. Visualization of Complex Attacks and State of Attacked Network. In *Proc. of VizSEC'09*, pages 1–9, 2009.

YRB10. T Yu, J Riordan, and S Boyer. EMBER : A Global Perspective on Extreme Malicious Categories and Subject Descriptors. 2010.

Yur06. W Yurcik. Tool Update : NVisionIP Improvements (Difference View, Sparklines, and Shapes). In *Proc. of VizSEC'06*, pages 65–66, 2006.

Dynamic Design of non Conflicting Responses against Simultaneous Attacks

Léa Samarji[1,2], Nora Cuppens-Boulahia[2], Frédéric Cuppens[2], Wael Kanoun[1], and Samuel Dubus[1]

[1] Alcatel-Lucent Bell Labs, Villarceaux, route de Villejust, 91620 Nozay, France
[2] Télécom Bretagne, rue de la Chataigneraie, 35510 Cesson-Sévigné, France

Abstract. ICT systems are frequently targeted by simultaneous attacks, which causes deterioration in system's performance and induces great damage to assets. Existing response systems still handle attacks independently, suffering thereby efficiency issues against coordinated attacks (e.g. DDoS), and potential conflicts between parallel responses. Consequently, we propose in this paper a new response model against simultaneous threats, defined as a sequence of non conflicting parallel actions. Our response is dynamically designed based on a new definition of applicability-aware logic anticorrelation, and modeled using Situation Calculus (SC) logic language. Our model is been applied on a multi-service network use case in order to investigate its efficiency.

Keywords: Prevention, Response, Simultaneous Attacks, Situation Calculus

1 Introduction

Modern attack tools are rapidly evolving to become more powerful and sophisticated. Networks and information systems are frequently targeted by simultaneous attacks, which causes deterioration in system's performance and induces great damage to physical assets. Simultaneous attacks are attacks launched against a system at the same time and by different attack entities. Each attack entity have an attack objective to reach in the system, and can be composed of either a single independent attacker, or a group of coordinated attackers (GCA). When the attack entity is a GCA, the system risks to suffer coordinated attacks [ZLK10]. Unfortunately, existing response systems proposals [TK02] [SBW07] [KCBCD10a] only handle the case of individual attackers, and responses are predesigned to react against each ongoing attack scenario as if it was the only scenario running in the system. Moreover, the majority of automated intrusion response systems rely on mapping attacks to predefined responses. This approach allows a system administrator to deal with intrusions faster. However it lacks flexibility as "things do not always turn out the way we planned". Actually, the following problems can occur owing to the above mentioned limitations:

– Until now, when a security expert predesigns one or several possibilities of responses against a threat, the different responses are statically prioritized. The prioritization is either based on the expert knowledge, or on a

comparison between the Return On Response Investment (RORI) indexes [KCBCD10b] [GGBDJ14] corresponding to the different responses. RORI indexes are usually assessed in offline before running the system. However, in a reactive phase, when a system is effectively threatened by different simultaneous threats, and that one of these threats becomes risky, the more prior response predesigned against this threat may not be the most efficient one in the current situation. Actually, this response may have unexpected effects on the other ongoing threats. Consequently, the different responses should be dynamically designed based on the current situation, and the one leveraging the highest RORI calculated in real time, should be activated.

– When multiple ongoing threats get risky, and the system is designed to automatically launch multiple responses against these threats, some responses may be conflicted (an example of conflicting responses is provided in Section 5. In [CCBB+08], the different types of conflict between responses are described. Current solutions [CCBB+08] to avoid situations of conflict propose to perform in offline a static assignment of priority over conflicting responses. However, conflicts between responses can strongly depend on the current system's state and the dynamic allocation of resources. Hence conflicts between actions should be dynamically considered. Additionally, priorities between responses should be dynamically assigned, because they depend on: (1) the dynamic risk of each of the ongoing threats, (2) the risk evolution of a threat regarding the activation of a response corresponding to another threat, and (3) the indefeasibility of one of the conflicting responses, if any. In order to consider these factors, responses against the different threats should be dynamically designed considering the potential conflicts in the current state.
– When predesigning responses, an expert may assign similar responses for different threats. However, if these threats appear simultaneously in the system, the same response will be activated and deployed multiple times corresponding to the number of the threats. As an example, consider an Oracle Java machine (supporting version Java SE 6u75, 7u60, or 8u5). This latter is suffering from two vulnerabilities (e.g. CVE-2014-4209[1] and CVE-2014-4268[2]) allowing remote attackers to affect confidentiality and integrity, either via vectors related to JMX, or via unknown vectors related to Swing. Consider that two independent threats aiming at exploiting these vulnerabilities were simultaneously detected in the system. In order to prevent these threats from reaching their goal, two responses are predesigned by an expert: Response 1 consists in installing a first patch against the first CVE and then restart the server for the patch to be effective, and Response 2 also consists in installing a second patch for the second CVE, and then similarly restart the server. Consider that the server is running a profit-making service for the system and restarting it puts a considerable time, inducing, thereby, losses to the system. When these two responses are launched in parallel, the restart action may be triggered two times, each corresponding to an installed patch,

[1] http://cvedetails.com/cve/2014-4209
[2] http://cvedetails.com/cve/2014-4268

unless both patchs put the same installation time, and restart actions are triggered at the same time. Consequently, the system will incur two times losses related to restarting time. Consequently, in order to avoid re-executing actions making part of simultaneous responses and uselessly consuming resources, responses should be intelligently designed on the fly against all the concerned threats. For instance, in this case, an efficient design would be to first install both patches and then trigger a single restart action once the installation of patches is finished.

In this paper we overcome all these problems by proposing a new framework that dynamically designs response candidates for simultaneous attack threats. Hence, we first introduce a new response scheme. Our response is thus described as a sequence of non conflicting parallel actions, allowing thereby an execution in parallel or in sequence of different actions handling all the risky threats. Our response is dynamically designed based on a new definition of a applicability-aware logic anticorrelation approach [CAB+06], and modeled through an efficient logic language, the Situation Calculus (SC) [MH69] [Rei01].

The paper is organized as follows: Section 2 presents a Simultaneous Attacks Graph (SAG) which forecasts simultaneous attack scenarios for the detected attackers in the system. SAG is an input for our framework, since it allows to identify risky threats for which we have to design a response. Section 3 introduces our new response scheme, and explains how to dynamically design a response based on a new definition of anticorrelation. Section 4 introduces the Situation Calculus language, and shows how to model our dynamic response with SC. Section 5 presents a use case example to demonstrate the effectiveness of our dynamic response design model in avoiding conflicting actions. Section 6 discusses related work, and finally Section 7 concludes our work.

2 Simultaneous Attacks Graphs

[SCCB+13] differs from other works on attack modeling [CO00] [AC06] [BB01] [TL00], by proposing a formal description of actions that correspond to all types of attacks (individual, coordinated and simultaneous ones). An algorithm was also proposed to generate Attack Graphs (SAG) corresponding to Simultaneous attacks scenarios. Hence, given a system's state and a set of suspicious attackers, the algorithm is able to generate multiple SAGs corresponding each, to a combination of scenarios predicted for those attackers. Figures 1 and 2 are two examples of generated graphs for a telephony operator threatened by fourteen suspicious attackers at time $t0$. In each graph, attackers may be assembled differently into groups, and may have a different end goal. Consequently, each graph represents only one combination of potential scenarios that can be simultaneously performed in the future by suspicious attackers. In the SAG of Figure 1, the algorithm assembles attackers into 3 groups of coordinated attackers $\{a1, a2, a3, a4, a5, a6, a7, a8, a9, a10\}$, $\{a11, a12\}$, and $\{a13, a14\}$. For the first group, a scenario targeting the operator's reputation on its principal services (e.g.

QoS) is predicted by performing a Distributed Denial of Service (DDoS) on one of its SIP servers. For the second group, a scenario that aims at inducing losses for an operator's client is predicted by hijacking its account and performing a toll fraud. For the third group, a scenario that aims at inducing losses for an operator's political client is predicted by recording its conversation and using it for black mailing. In another example, in Figure 2, attackers $a11$ and $a12$ are not coordinated, and each one has it own end goal. Notice that an attack entity (e.g. $a11$) can block another one (e.g. 12) from progressing. This can occur due to a simultaneous access to unshareable resources in the system. Attackers in this case are called concurrent.

In [SCBC+14], a new framework was proposed to properly assess the Likelihood of Individual, Coordinated, and Concurrent Attack Scenarios. The method is based on a coordination aware-*Game Theoric* approach to derive an *Attack Likelihood* equation. This latter is referred to *NIST* in the way it considers: the attack's complexity, the attackers' motivation, and the existence of potential responses. An algorithm was also proposed to assess the *Scenario Likelihood* of each attack scenario in a given SAG, considering the concurrency between attackers. Consequently, the SAG containing scenarios that are the most likely to happen (i.e. scenarios with high likelihood) in the system can be chosen.

The most likely SAG is an input for our framework. Based on the attack scenarios predicted for the simultaneous attack entities, our framework will dynamically design and co-simulate a response efficient against the set of scenarios that are tagged as risky within the SAG.

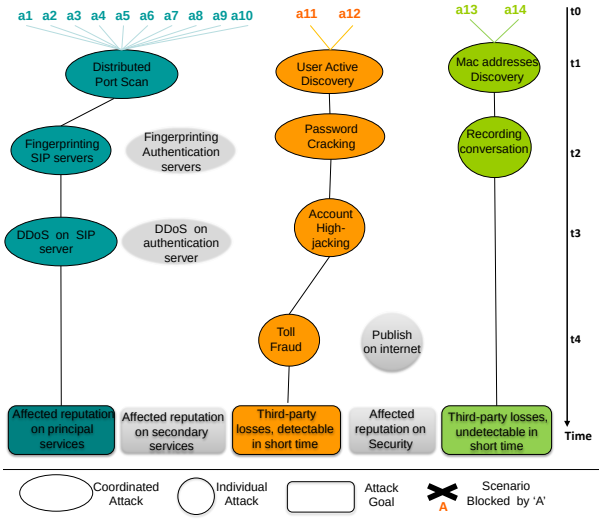


Fig. 1. Example 1 of a VoIP SAG.

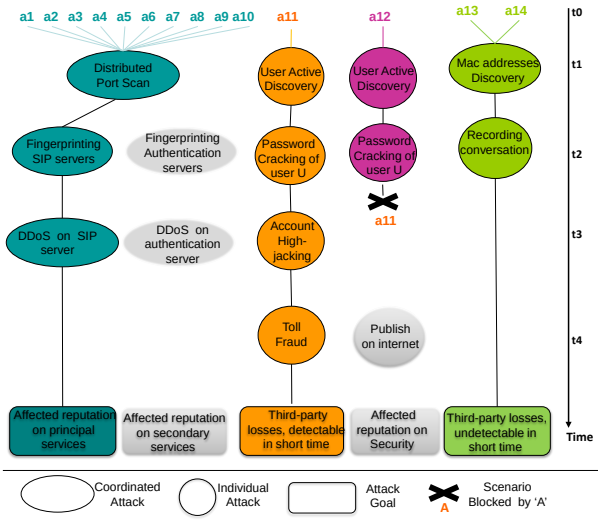


Fig. 2. Example 2 of a VoIP SAG.

3 Formal Description of a Response against Simultaneous Risky Threats

Let $[[a_1^1, a_2^1, ..., a_M^1]; [a_1^2, a_2^2, ..., a_M^2]; ...; [a_1^N, a_2^N, ..., a_M^N]]$ be the sequence of simultaneous attacks forecasted in SAG for the M different attack entities present in the system, with a_j^i being the action that is going to be executed in the i^{th} place by the attack entity j. Symbol $'$,$'$ represents parallelism, and symbol $'$;$'$ represents sequencing. Note that a can also be no operation if no action is predicted for the entity. Consider that attack entities $1, 2, ..., K$ are considered as risky and attack entities $K+1, ..., M$ are considered as not risky. Hence, the following is the sequence of Risky Simultaneous Attacks ($RiskySAS$):
$RiskySAS = [[a_1^1, ..., a_K^1]; [a_1^2, ..., a_K^2]; ...; [a_1^N, ..., a_K^N]]$.

R is considered a response against $RiskySAS$, if R is able, while maintaining the system in an operational state, to either prevent or delay entities $1, 2, ..., K$ from reaching their attack objectives. In order to respond to simultaneous threat, the response should not be limited to a single elementary action, we thus consider a response as a complex action, and we define it as a set of partially ordered elementary actions. In other words, a response may consist of non conflicting system's actions activated in parallel and of system's actions activated in sequence. Contrarily to an attack action, a system's action is an action triggered/executed by the system. $openSession(Src\_ip, Src\_port, Dest\_ip, Dest\_port)$, $restart(Server)$, $installPatch(patchID, machineIP)$, $deployProtocol(StrongAuthentication, Server)$, are examples of system's actions.

We introduce, in the following, a generic action scheme for a response R described as a sequence of length x of parallel system's actions:
$R = [[r_1^1, ..., r_{l1}^1]; [r_1^2, ..., r_{l2}^2]; ...; [r_1^x, ..., r_{lx}^x]]$
with r_i^k being one of the lk system's actions executed in the k^{th} place (i.e. k^{th} time step) , and $\forall k/\ 1 < k < x$, $\forall i \neq j\ /\ 1 < i, j < lk$, r_i^k and r_j^k are not conflicting (i.e. meaning that parallel actions should be compatible together for a parallel execution).

In order to design a response on the fly against a $RiskySAS$, a dynamic anticorrelation logic approach should be applied. Unfortunately, existing work on anticorrelation is limited to an anticorrelation definition unaware of the applicability of actions, and is thereby inefficient for dynamic use. Moreover, the existing definition of anticorrelation is limited to the case where a response consists of a single elementary action. We thus propose in the following sections an applicability-aware anticorrelation definition adapted to our response scheme.

3.1 Applicability-aware Anticorrelation in Logic Programming

Anticorrelation in logic programming was defined in [CAB+06] as follows:

Definition 1. *Let r and a be respectively logic descriptions of a system's action and an attack action. $postr$ is the set of predicates of post-conditions of r and $prea$ is the set of predicates of pre-condition of a. r and a are anti-correlated if*

the following condition is satisfied:
$anticorrelated(r, a) \leftrightarrow \exists Pr, Pa/(Pr \in postr \wedge \neg Pa \in prea) \wedge (Pr, Pa$ *are unifiable).*

For example, consider $passwordCrack(Attacker_1, U, Serv)$ a cracking attack of user U's password through server $Serv$. A precondition of this attack is to have $Attacker_1$ having network access to $Serv$. Let $discard(Attacker_1, Serv)$ be a system's action consisting in disconnecting $Attacker_1$ from the network of $Serv$. Consequently, we have:
$anticorrelated(discard(Attacker_1, Serv), passwordCrack(Attacker_1, U, Serv))$.

Unfortunately, Definition 1 does not consider the relevance and the applicability of the system's action. In other words, the possibility to execute the system's action in the current state of the system is not taken into consideration while reasoning on anticorrelated actions. Actually, if we reconsider the latest example, the $discard(Attacker_1, Serv)$ action, may not be possible for execution in the current state because the data base containing allowed ip addresses for connection to $Serv$ is exclusively opened by another module in the system. Hence, response system should wait until the data base is released, to be able to execute its action. Consequently, the applicability of an action in a system's state S should be considered while designing responses based on logic anticorrelation. Therefore, we propose an applicability-aware anticorrelation definition as follows:

Definition 2. *Let r and a be respectively logic descriptions of a system's action and an attack action. Let S be the current state/situation of the system, and $poss(r, S)$ a predicate meaning that it is possible to execute r in state S. r and a are anti-correlated in state S if the following condition is satisfied:*
$anticorrelated(r, a, S) \leftrightarrow poss(r, S) \wedge anticorrelated(r, a)$.
$poss(r, S) \leftrightarrow \forall P \in prer,\ P$ *is fulfilled in* S

3.2 Applicability-aware Anticorrelation for Complex Actions

Unfortunately, Definition 2 is limited to the case where a response consists of a single elementary action. However, a system may sometimes need to coordinate multiple elementary actions in order to react against an attack a, especially when a is a coordinated attack [SCCB+13]. As an example, consider a server $Serv$ able to handle up to K connection requests per second. Consider that 25 users are registered to this server, and each one can send up to $K/20$ connection requests per second to the server. Thus, the server can handle up to 20 users deploying their entire bandwidth in sending requests. Consider also that 22 users were been infected with an external bot, and they are coordinately flooding (i.e. executing a DDoS) against $Serv$. The set of compromised users is thus considered as a GCA. This latter was able to perform a DDoS because $\mid GCA \mid > 20$. In order to respond to DDoS attack, the system should discard in parallel at least three of the infected users to reduce the receiving flow below the threshold of $Serv$. Thus, it is the resulting effect of the three *discard* actions which is opposite to

the precondition of DDoS ($|\ GCA\ |> 20$). The following is a logic expression of the combined effect of the three elementary actions.
$discarded(User_{10}) \wedge discarded(User_{11}) \wedge discarded(User_{12}) \rightarrow |\ GCA\ | < 20$.
In this case, it is the set of parallel actions $[discard(User_{10}), discard(User_{11}), discard(User_{12})]$ which is anticorrelated with $DDoS(GCA, Serv)$.

Therefore, we propose a new definition of applicability-aware anticorrelation between a set of coordinated elementary system's actions and an attack, as follows:

Definition 3. *Let* $rcoordinated = [r_1, r_2, ..., r_C]$ *be a set of parallel system's actions, and* a *an attack action.* $postrk$ *is the set of predicates of post-conditions of* r_k *and* $prea$ *is the set of predicates of pre-condition of* a*.* $rcoordinated$ *and* a *are anti-correlated if the following condition is satisfied:*
$anticorrelated([r_1, r_2, ..., r_C], a, S) \leftrightarrow$
$poss([r_1, r_2, ..., r_C], S)\ \wedge anticorrelated([r_1, r_2, ..., r_C], a)$.
with:
$anticorrelated([r_1, r_2, ..., r_C], a) \leftrightarrow \exists Pr\ /(\forall postr_k, \exists Pr_k \in postr_k),\ (Pr_1 \wedge Pr_2 \wedge ... \wedge Pr_C \rightarrow Pr)\ \wedge \exists Pa\ /(\ \neg Pa \in prea\ \wedge\ Pr, Pa$ *are unifiable).*
and
$poss([r_1, r_2, ..., r_C], S) \leftrightarrow \forall 1 < i < C, poss(r_i, S) \wedge \forall j \neq i / 1 < j < C, \neg conflict(r_i, r_j)$.

$\neg conflict(r_i, r_j)$ means that r_i and r_j are not conflicted (i.e. they can be executed simultaneously). The semantic definition of *conflict* will be given in Section 4.3. Furthermore, in some cases, a system's action r or a set of parallel system's actions $rcoordinated$ may be logically anticorrelated with a given attack action, but it is not applicable in the current state S. Consequently, the system may need to sequentially execute a set of 'applicability enabling' actions in order to make r or $rcoordinated$ applicable. For instance, consider that an external attacker has gain remote access to a machine M in a system, and that he/she has installed a bot on M in order to have total control of it making him/her able to disrupt other connected machines or services in the system. In order to prevent the attacker from inducing damage in the system, a security patch, able to patch the vulnerability which enabled the attacker to gain a remote access from the beginning, must be installed. However, in general, security patches can not be installed on infected machines before reformatting these machines to eliminate potential existing bots. Besides, reformatting a machine requires, first, a backup for important files on it. Consequently, in this case, installing the security patch corresponds to r because it is logically anticorrelated with the remote access attack. And, 'applicability enabling' actions are $reformatHarddrive(M)$ and $backupFiles(M, BackupServer)$. The sequence of actions required to block the attacker is, thus, the following:
$R = [\ [backupFiles(M, BackupServer)]; [reformatHarddrive(M)];$
$install(SecurityPatch, M)]\ ]$.

Therefore, we propose a new definition of applicability-aware anticorrelation between a complex action (i.e. a sequence of parallel system's actions) and an attack action, as follows:

Definition 4. *Let* $R* = [[r_1^1, ..., r_{l1}^1]; [r_1^2, ..., r_{l2}^2]; ...; [r_1^c, ..., r_{lc}^c]]$ *be a complex action, and a an attack action. Let* S_{i+1} *be the state of the system after the execution of* $[r_1^i, ..., r_{li}^i]$ *in state* S_i. $R*$ *and a are anticorrelated in state* S_0 *if the following condition is satisfied:*
$anticorrelated([[r_1^1, ..., r_{l1}^1]; [r_1^2, ..., r_{l2}^2]; ...; [r_1^c, ..., r_{lc}^c]], a, S_0) \leftrightarrow$
$anticorrelated([r_1^c, ..., r_{lc}^c], a) \wedge poss([r_1^1, ..., r_{l1}^1], S_0) \wedge poss([r_1^2, ..., r_{l2}^2], S_1) \wedge .... \wedge$
$poss([r_1^c, ..., r_{lc}^c], S_{c-1})$.

Based on Definition 4, we now propose a definition of applicability-aware anticorrelation between a complex action R, and a $RiskySAS$ as follows:

Definition 5. *Let* $R = [[r_1^1, ..., r_{l1}^1]; [r_1^2, ..., r_{l2}^2]; ...; [r_1^x, ..., r_{lx}^x]]$ *be a complex action, and RiskySAS a set of risky simultaneous attack scenarios, such that* $RiskySAS = [[a_1^1, ..., a_K^1]; [a_1^2, ..., a_K^2]; ...; [a_1^N, ..., a_K^N]]$, *with* a_j^i *being an attack performed by attack entity* j. R *and RiskySAS are anti-correlated if the following condition is satisfied:*
$anticorrelated(R, RiskySAS, S) \leftrightarrow \forall entity(j),\ \exists R* \in R\ , \exists a \in [a_j^1; a_j^2; ...; a_j^N]/$
$anticorrelated(R*, a, S)$.

Thus, R is anticorrelated with $RiskySAS$ if for each attack sequence corresponding to an attack entity, we can find a complex action $R*$ within R which is anticorrelated and applicable with at least one of the attacks that the entity will execute throughout her sequence.

3.3 Response Definition

A complex action R is considered as a dynamic response against $RiskySAS$ in a state S, if R is applicable in S and anticorrelated with $RiskySAS$, and no nominal constraint is violated at the end of R's execution. Nominal constraints are those related to critical system assets that should not be violated, in order to grantee a minimum operating state (i.e. service continuity) in the system. Consequently, R should include 'operability' actions, responsible for maintaining the service, whenever nominal constraints risks to be violated by the system's actions composing R. We, thus, define a response R against a $RiskySAS$ as follows:

Definition 6. *Let* $R = [[r_1^1, ..., r_{l1}^1]; [r_1^2, ..., r_{l2}^2]; ...; [r_1^x, ..., r_{lx}^x]]$ *be a complex action, and RiskySAS a set of risky simultaneous attack scenarios. Let* S_x *be the state of the system after the execution of R. In other words,* S_x *is the state of the system when* $[r_1^x, ..., r_{lx}^x]$ *is executed. And, let min_constraints be the set of nominal constraints. R is a response against RiskySAS in state S if the following condition is fulfilled:*
$response(R, RiskySAS, S) \leftrightarrow anticorrelated(R, RiskySAS, S) \wedge \forall P_{operational} \in$
$min\_constraints, P_{operational}$ *is fulfilled in* S_x.

For example, consider a system threatened simultaneously by two risky threats T1 and T2. T1 aim at over-flooding server S1, and T2 aim at hijacking a legitimate user's account U throughout a machine M infected with a bot. The

following sequence of parallel actions corresponds, thus, to a response against T1 and T2.
$R = [\ [shareLoad(S1, S2), disconnect(M)]; [backupFiles(M, BackupServer)];$
$[reformatHarddrive(M)]; [install(SecurityPatch, M)]; [connect(M)]\ ].$
In a first step, sharing load is settled between $S1$ and another server $S2$ in order to prevent $S1$ from being overcharged by T1. In parallel, M is disconnected from the network, and files on M are backed up in order to reformat the machine and install a security patch in further steps. By this, the bot on M is removed and the vulnerability is patched. In this example, $shareLoad(S1, S2)$, and $install(SecurityPatch, M)$ are two system's actions anticorrelated respectively with T1 and T2. $disconnect(M)$, $reformatHarddrive(M)$ and $backupFiles(M, BackupServer)$ are 'applicability enabling' actions for $install(SecurityPatch, M)$. $connect(M)$ is an 'operability' action. Note that without this latter, the response against T2 will not be effective. Another possibility of response can be the following:
$R = [\ [prioritize(S1, PremiumUsers), deploy(StrongAuthentication, AuthServer)\ ].$
Here, in order to prevent the denial of service of $S1$, this latter is configured to prioritize premium users over normal ones. In parallel, a strong authentication is deployed on the authentication server in order to prevent cracking the password of all users including U. In this case R do not include 'applicability enabling' actions, nor 'operability' actions.

In order to design our responses based on logic anticorrelation, system's and attacks' actions should be modeled using the same logic language. Additionally, the modeling language should follow a pre/post condition approach for actions description. In [SCCB+13] different modeling languages were compared: LAMBDA [CO00] [AC06], STRIPS [BB01], JIGSAW [TL00], and Situation Calculus [MH69] [Rei01], and the latter turns to be the most adapted language to describe all types of attacks (individual, coordinated and simultaneous ones). We, thus, investigate in the next section the adaptability of SC in modeling anticorrelation and responses as we have defined them.

4 Modeling Responses with Situation Calculus (SC)

4.1 Basics of the Situation Calculus

Situation Calculus [MH69] [Rei01] is a dialect of first order logic, with second order-logic terms for representing dynamic change. It basically consists of:

- Situations: a situation is a first-order term denoting a sequence of actions. It represents the system's state, and the action's history (i.e. sequence) from an initial empty action sequence $S0$.
- Fluents and Predicates: the world is described in terms of static predicates and fluents. Static predicates do not change, no matter what actions are taken. Whereas fluents are predicates that can vary over time, and thus must take situations as arguments. For example, $Server(Serv)$ is a static predicate meaning that $Serv$ is a server. While, $network\_access(M1, S2, s)$

is a fluent meaning that machine $M1$ has a network access to server $S2$ in situation s. Additionally, Fluents can be either relational, or functional. Relational fluents return boolean values, e.g. $is\_on(Serv, s)$, while functional fluents return a non boolean value, e.g. $received\_flow(Serv, s) = 500$.
- Actions: consist of a function symbol and its arguments. For example, $reboot(Server1)$ is the action of rebooting $Server1$. In order to reason about the effects of an action, we need to be able to refer to the situation that results from the execution of this action. This is done using the do function. $do(a, s)$ denotes the situation that results from doing action a in situation s.

SC also provides essential axioms to represent dynamic changes:

- Action precondition axioms: for each action a, there is a predicate $Poss(a, s)$ that states if it is possible for action a to be executed in situation s.
- Successor state axioms: there is one for each fluent F. It characterizes the conditions under which a fluent $F(x, do(a, s))$ changes from situation s to situation $do(a, s)$.

4.2 Modeling System's Actions using Situation Calculus

SC provides semantics for pre/postconditions that are not provided by other languages. The precondition of action a is represented by $Poss(do(a, s))$. Whereas the postcondition is represented by the function $do(a, s)$ which denotes the fluents that change after applying action a to situation s. SC provides an expressive framework for encoding actions whose effects are functions of the state in which they are executed. Therefore, SC answers our need in (1) offering the possibility to dynamically design a response whose requirements and effects depend crucially on the system's state, and (2) modeling operational and defense actions following a pre/post condition approach. The following is an SC description of an operational action which consists in forcing a server $S1$ in sharing the load with another one $S2$ when overcharged.
shareLoad$(Server_x, Server_y)$
Poss$(shareLoad(Server_x, Server_y), S) \leftrightarrow$
$overcharged(Server_x, S) \ \wedge is\_on(Server_y) \ \wedge runningService(Server_x, S) = runningService(Server_y, S)$.
do$(shareLoad(Server_x, Server_y), S) = S' \rightarrow \neg overcharged(Server_x, S')$.

4.3 Modeling Concurrent actions with Situation Calculus

In [Rei96] and [Pin94], SC ontology was expanded to handle concurrency. A new sort *concurrent* is added. Every *concurrent* variable c is a set of concurrent simple actions a. In our case IA and CA are simple actions. The binary function $do(c, s)$ returns a situation term that results from the application of concurrent actions c in situation s. The item $Poss(a, s)$ is also extended to handle concurrent actions. Consequently, $Poss(c, s)$ means that concurrent actions set c is possible in situation s. Additionally, in a simultaneous actions context, some actions can

not be performed concurrently. This is due to incompatibility between actions in terms of resources that each action uses. For instance, if action a_1 needs a resource for its execution, and another action a_2 needs the same resource, then the set of concurrent actions $c = \{a_1, a_2\}$ can not be executed unless this resource can be shared. As a solution, Pinto [Pin94] proposed to add a finer level of granularity by appealing to the notion of resource: $xres(a, r)$ means that action a requires the exclusive use of the resource r, and $sres(a, r)$ means that action a requires the use of the resource r for its execution, but r can be shared. Finally, $poss(c, s)$ makes use of a conflict predicate $conflict$ as a precondition in order to test compatibility between actions:

$$conflict(c) \leftrightarrow \exists a_1, a_2 \in c, \exists r \mid [(xres(a_1, r) \wedge xres(a_2, r)) \vee (xres(a_1, r) \wedge sres(a_2, r)) \vee (sres(a_1, r) \wedge xres(a_2, r))]$$
$$Poss(c, s) \leftrightarrow [\forall a \in c, Poss(a, s)] \wedge \neg conflict(c)$$

Concurrent SC is thus efficient for avoiding conflicting actions while dynamically designing a response. In Section 5, an example of conflicting actions is addressed, and the notion of resource of concurrent SC is appealed to overcome the problem.

4.4 Modeling Anticorrelation with Situation Calculus

Let r and a be respectively a SC description of a system's action, and an attack action. Anticorrelation between r and a presented in Definition 2 is expressed in SC as follows:
$anticorrelated(r, a, S) \leftrightarrow poss(r, S) \wedge \neg poss(a, do(r, S))$.

Let $rconcurrent = [r_1, r_2, ..., r_L]$ be a set of parallel system's actions, and a an attack action. Anticorrelation between $rconcurrent$ and a, as presented in Definition 3, can be expressed in concurrent SC as follows:
$anticorrelated([r_1, r_2, ..., r_L], a, S) \leftrightarrow$
$poss([r_1, r_2, ..., r_L], S) \wedge \neg poss(a, do([r_1, r_2, ..., r_L], S))$.

Now, let $R* = [[r_1^1, ..., r_{l1}^1]; [r_1^2, ..., r_{l2}^2]; ...; [r_1^c, ..., r_{lc}^c]]$ be a complex action, and a an attack action. Anticorrelation between $R*$ and a, as presented in Definition 4, can be expressed in concurrent SC as follows:
$anticorrelated([[r_1^1, ..., r_{l1}^1]; [r_1^2, ..., r_{l2}^2]; ...; [r_1^c, ..., r_{lc}^c]], a, S) \leftrightarrow poss([r_1, r_2, ..., r_L], S)$
$\wedge\, poss([r_1^2, ..., r_{l2}^2], do([r_1, r_2, ..., r_L], S)) \wedge ... \wedge$
$poss([r_1^c, ..., r_{lc}^c], do([r_1^{c-1}, ..., r_{lc}^{c-1}], (....(do([r_1, r_2, ..., r_L], S)...)) \wedge$
$\neg poss(a, do([r_1^c, ..., r_{lc}^c], (do([r_1^{c-1}, ..., r_{lc}^{c-1}], (....(do([r_1, r_2, ..., r_L], S)...)))$.

Consequently, concurrent SC is adapted to model anticorrelation of a complex action against a $RiskySAS$ as we defined it in Definition 5.

Finally, a response (see Definition 6) can be modeled in SC as follows:
$response(R, RiskSAS, S) \leftrightarrow anticorrelated(R, RiskSAS, S) \wedge \forall P_{operational} \in min\_constaints, P_{operational}(do(R, S)) == True$.

5 Use Case: Multi-Service ICT infrastructure

As a use case example for our framework, we consider the following:

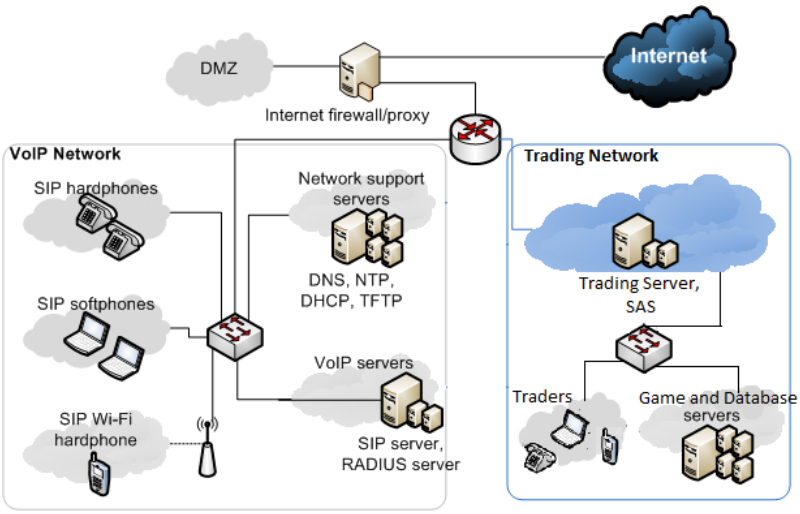


Fig. 3. A system running VoIP and Trading services.

Consider a system running a VoIP service, and a Trading service as in Figure 3. For clients subscribed to VoIP, a password based authentication using PAP (password authentication protocol) is considered and handled by a RADIUS server. While for clients subscribed to Trading service, a strong authentication (e.g. multi-factor authentication, Digest access authentication, etc.) is adopted and handled by a Strong Authentication Server (SAS). We also consider that SAS is suffering a Zero day vulnerability (e.g. the OpenSSL vulnerability [3] discovered recently).

At time t, two different risky threats T1 and T2 lead respectively by a compromised machine in the VoIP network $Attacker_1$, and a malicious trader $Trader_A$ were forecasted.

In T1, a password cracking attack is previewed on a VoIP user 'U' account, with the following attack sequence:
$T1 = [scan(Attacker_1, U); passwordCrack(Attacker_1, U, RADIUS);$
$accountHighjack(Attacker_1, U, RADIUS); tollFraud(Attacker_1, U)]$.

In T2, $Trader_A$ will try to exploit the vulnerability of SAS and prevent other traders from connecting to the trading service, as follows:
$T1 = [scan(Trader_A, SAS); scanVulnerability(Trader_A, SAS, OpenSSL);$

[3] http://www.zdnet.com/heartbleed-serious-openssl-zero-day-vulnerability-revealed-7000028166/

$exploitVulnerability(Trader_A, SAS, OpenSSL)]$

Consider the following abstract actions modeled in SC for the system:
$transferData(Server_A, Server_B)$.
$install(Server, Patch)$.
$requestChangePassword(Server, Client)$.
$discard(Service, Client)$.

Where, $Server$, $Service$, $Client$ and $Patch$, are abstract variables corresponding respectively to the servers (e.g. VoIP server, SAS, RADIUS, Trading server, etc.), the services (e.g. VoIP and Trading), the clients (VoIP users and traders), and the possible patchs that can be installed on servers.

In order to prevent T1, a solution would be to adapt the strong authentication to the VoIP service. To do so, the database containing information (passwords, accounts, etc.) about VoIP clients should be transferred to server SAS, performing strong authentication. Thus, $r_1 = transferData(RADIUS, SAS)$ which is anticorrelated with $passwordCrack(Attacker_1, U, RADIUS)$ can be chosen. Another solution would be to notify U to change his password before that $Attacker_1$ highjacks his account. Thus, action $r_3 = requestChangePassword(VoIPserver, U)$ which is anticorrelated to $accountHighjack(Attacker_1, U, RADIUS)$ can be also chosen.

In order to prevent T2, a solution would be to disconnect SAS in order to install security patches or a new software version (e.g. OpenSSL 1.0.1g) to patch the existing vulnerability. Thus, action $r_2 = install(SAS, SecurityPatch)$ which is anticorrelated to $scanVulnerability(Trader_A, SAS, OpenSSL)$ can be chosen. Another solution would be to discard or blacklist the malicious trader for a while. Thus, $r_4 = discard(Trading, Trader_A)$ which is anticorrelated to $scan(Trader_A, SAS)$ can also be chosen.

A naive solution to respond to both threats would be to choose any combination $[r_i, r_j]$, with i an even number and j an odd number. However, r_1 and r_2 are conflicting actions. Actually, installing the security patch requires disconnecting SAS from the network, whereas transfering data to SAS requires this latter to stay online. Consequently, our framework prevents the dynamic design of these two actions in parallel, by appealing the notion of resource of concurrent SC:
$xres(r_2, SAS)$
$sres(r_1, SAS)$
Thus, $conflict([r_1, r_2])$ returns true, and $Poss([r_1, r_2], S)$ returns false.

Therefore, our framework dynamically chooses non conflicting actions to design different responses possibilities preventing the system from simultaneous threats T1 and T2:

Responses with parallel actions:
$Response1 = [transferData(RADIUS, SAS), discard(Trading, Trader_A)]$.
$Response2 = [requestChangePassword(VoIPserver, U), install(SAS, SecurityPatch)]$.

$Response3 = [requestChangePassword(VoIPserver, U), discard(Trading, Trader_A)]$.

Responses with sequential actions:
$Response4 = [[install(SAS, SecurityPatch)]; [transferData(RADIUS, SAS)]]$.
$Response5 = [[transferData(RADIUS, SAS)]; [discard(Trading, Trader_A)]]$.
$Response6 = [[discard(Trading, Trader_A)]; [transferData(RADIUS, SAS)]]$.
$Response7 = [[requestChangePassword(VoIPserver, U)]; [install(SAS, SecurityPatch)]]$.
$Response8 = [[install(SAS, SecurityPatch)]; [requestChangePassword(VoIPserver, U)]]$.
$Response9 = [[discard(Trading, Trader_A))]; [requestChangePassword(VoIPserver, U)]]$.
$Response10 = [[requestChangePassword(VoIPserver, U)]; [discard(Trading, Trader_A))]]$.

Note that, due to lack of space, we did not consider 'applicability enabling', nor 'operability' actions in this example. We wanted, here, to highlight the effectivness of our framework in handling conflict between anticorrelation actions.

6 Related Work

In [KCBCD10a], authors presented a risk-aware framework for activating and deactivating policy-based responses. However, an expert is needed to specify the response policy and enforce, in advance, a set of response rules for every specific identified threat context. Therefore, conflicts between responses, and the possibility to launch a common response for multiple threats, are not taken into consideration.

In [CCBB+08], the different types of conflict between responses are described, and a solution to avoid situations of conflict was proposed. This latter consists in performing, in offline, a static assignment of priorities over conflicting responses. However, conflicts between responses may strongly depend on the current system's state and the dynamic allocation of resources. Besides, conflict should be dynamically handled considering the dynamic risk of each of the ongoing threats, (2) the risk evolution of a threat regarding the activation of a response corresponding to another threat, and (3) the indefeasibility of one of the conflicting responses, if any.

In [GGBDJ14] , authors introduced a well structured approach to evaluate a Return On Response Investment (RORI) index for all possible combinations of security measures that can be launched against simultaneous threats. Until now, RORI indexes are assessed in offline before running the system. However, in offline, one can not consider all the possible states that a system can have in real time. Consequently, one can either miss an efficient combination of responses or can be surprised by an unexpected bad effect that a predesigned combination of responses has on the system. Consequently, the different combination of responses should better be dynamically designed based on the real time system's state, and the one leveraging the highest RORI, calculated in real time, can be then activated.

7 Conclusion

In this paper, we proposed a new formal description of a response against simultaneous threats, as a sequence of non conflicting parallel actions. Our response is dynamically designed based on a new definition of applicability-aware logic anticorrelation, and modeled using Situation Calculus logic language. This latter is very efficient to describe conflicts between parallel actions by appealing the notion of resource.

As a future work, we intend to assess the Risk Mitigation of each response possibility in order to choose the most effective one. To do so, each response possibility should be co-simulated, considering the system's state and the currently existing simultaneous attack entities. When co-simulating a response, the attack scenarios (1) blocked, (2) became less risky or, in the worst case, (3) appeared after the response simulation, are generated in order to be considered in the risk mitigation calculus process.

References

AC06. Fabien Autrel and Frédéric Cuppens. Crim: un module de corrélation d'alertes et de réaction aux attaques. *Annales des Télécommunications*, 61(9-10):1172–1192, 2006.

BB01. Craig Boutilier and Ronen I. Brafman. Partial-order planning with concurrent interacting actions. *J. Artif. Int. Res.*, 14(1):105–136, April 2001.

CAB+06. Frédéric CUPPENS, Fabien AUTREL, Yacine BOUZIDA, Joaquin GARCIA, Sylvain GOMBAULT, and Thierry SANS. Anti-correlation as a criterion to select appropriate counter-measures in an intrusion detection network, 2006.

CCBB+08. F. Cuppens, N. Cuppens-Boulahia, Y. Bouzida, W. Kanoun, and A. Croissant. Expression and deployment of reaction policies. In *Signal Image Technology and Internet Based Systems, 2008. SITIS '08. IEEE International Conference on*, pages 118–127, Nov 2008.

CO00. Frédéric Cuppens and Rodolphe Ortalo. Lambda: A language to model a database for detection of attacks. In *Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection*, RAID '00, pages 197–216, London, UK, UK, 2000. Springer-Verlag.

GGBDJ14. Gustavo Gonzalez Granadillo, Malek Belhaouane, Herv Debar, and Grgoire Jacob. Rori-based countermeasure selection using the orbac formalism. *International Journal of Information Security*, 13(1):63–79, 2014.

KCBCD10a. Wael Kanoun, Nora Cuppens-Boulahia, Frédéric Cuppens, and Samuel Dubus. Risk-aware framework for activating and deactivating policy-based response. *Network and System Security, International Conference on*, 0:207–215, 2010.

KCBCD10b. Nizar Kheir, Nora Cuppens-Boulahia, Frdric Cuppens, and Herv Debar. A service dependency model for cost-sensitive intrusion response. In *Computer Security ESORICS 2010*, volume 6345 of *Lecture Notes in Computer Science*, pages 626–642. Springer Berlin Heidelberg, 2010.

MH69. J. Mccarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4, 1969.

Pin94. Javier Andrs Pinto. Temporal reasoning in the situation calculus, 1994.

Rei96. Raymond Reiter. Natural actions, concurrency and continuous time in the situation calculus. In Luigia Carlucci Aiello, Jon Doyle, and Stuart C. Shapiro, editors, *KR*, pages 2–13. Morgan Kaufmann, 1996.

Rei01. Raymond Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. The MIT Press, Massachusetts, MA, illustrated edition edition, 2001.

SBW07. Natalia Stakhanova, Samik Basu, and Johnny Wong. A cost-sensitive model for preemptive intrusion response systems. In *Proceedings of the 21st International Conference on Advanced Networking and Applications*, AINA '07, pages 428–435, Washington, DC, USA, 2007. IEEE Computer Society.

SCBC$^+$14. Léa Samarji, Nora Cuppens-Boulahia, Frédéric Cuppens, Wael Kanoun, Serge Papillon, and Samuel Dubus. [to appear] coordination and concurrency aware likelihood assessment of simultaneous attacks. In *Security and Privacy in Communication Networks*. 2014.

SCCB$^+$13. Layal Samarji, Frédéric Cuppens, Nora Cuppens-Boulahia, Wael Kanoun, and Samuel Dubus. Situation calculus and graph based defensive modeling of simultaneous attacks. In *Cyberspace Safety and Security*, volume 8300 of *Lecture Notes in Computer Science*, pages 132–150. Springer International Publishing, 2013.

TK02. Thomas Toth and Christopher Kruegel. Evaluating the impact of automated intrusion response mechanisms, 2002.

TL00. Steven J. Templeton and Karl Levitt. A requires/provides model for computer attacks. In *Proceedings of the 2000 workshop on New security paradigms*, NSPW '00, pages 31–38, New York, NY, USA, 2000. ACM.

ZLK10. Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, pages 124–140, 2010.

Remediating Logical Attack Paths Using Information System Simulated Topologies

Category: Specialized

François-Xavier Aguessy[1,2], Lucie Gaspard[1], Olivier Bettan[1], and Vania Conan[1]
francois-xavier.aguessy@thalesgroup.com

[1] Thales Group, 4 avenue des Louvresses, 92622 Gennevilliers, France
[2] Telecom SudParis, 9 rue Charles Fourier, 91011 Evry, France

Abstract. With the increase of attacks and Information Systems getting ever more complex, security operators need tools to help them protecting critical assets. An attack graph is a model to assess the level of security of an Information System, but it can be used to compute actions that mitigate the modeled threats. In this paper we present a method to remediate the most relevant attack paths extracted from a logical attack graph. In order to help an operator to choose between several remediation candidates, we rank them according to a cost of remediation combining operational and impact costs. We implement this method using MulVAL attack graphs and several publicly available sets of data.

Keywords: logical attack paths, remediation candidates, MulVAL attack graph, simulated topology, remediation costs, remediation database.

1 Introduction

Due to the increase in the number and complexity of attacks, any Information System (IS) is vulnerable. Accurately assessing the risk is necessary but can be difficult. An attack graph is a risk analysis model regrouping all the paths an attacker may follow. It is composed of nodes, representing the actions possible for an attacker. Nodes are linked together with edges, representing dependencies between these actions. The main attack paths can be extracted from this graph in accordance to the IS priorities. In this paper we do not take into account the probability of occurrence of the attack path nor the damages it can do on the IS, but rather focus on the computation of remediations to an attack path.

A remediation aims at protecting the IS against an attack path by preventing its fulfillment. The nodes of an attack path not having any incoming edge are the starting points of attacks and are called *preconditions*. It is possible to offer remediation actions to apply on preconditions to mitigate the vulnerabilities and secure the targeted assets. In this paper, the definition of remediation encompass a patch, a firewall rule or an Intrusion Prevention System rule. Other more complex remediations such as access control or security policy management will

not be investigated here. The computation of filtering remediations has required the use of an accurate simulation of the IS network topology. To be usable in an operational environment, remediation candidates have to be ranked according to their operational and impact costs depending on the monitored IS. The assessment of the impact cost requires both the simulation of network flows and a functional model describing the normal behavior of the IS, in order to determine which nominal services may be disturbed by the deployment of a remediation.

The main contributions of this paper are (1) the design of a remediation process correcting the relevant attack paths rather than the whole attack graph generally too complex, (2) the method to remediate these paths based on the correction of attack preconditions, ranking remediation candidates according to a cost function considering both operational and impact costs, (3) the build of a generic remediation database assigning vulnerabilities with their remediations.

This paper is organized as follows: in Section 2, we briefly describe the state of the art. Section 3 formally defines the attack path and its main components. Section 4 explains how to compute remediations for an attack path. Section 5 describes the ranking of remediation candidates according to their cost. Section 6 details an implementation of this method with the MulVAL attack graph engine we use for the experiments of Section 7. Finally, in Section 8 we compare our model with the related work, before concluding on our work.

2 State of the art

2.1 Topological and functional models of the Information System

An efficient security analysis needs an accurate topological model of the system studied. Recent developments have shown that models of an IS can be created automatically from network scans [16] [10]. It can also be created by importing the configuration of network devices as it is done in some commercial solutions [28]. The model should be as accurate as possible to be exploitable. Nevertheless, its completeness is not guaranteed by the available technology.

The functional model of an IS is complementary to the topological one. It contains the dependencies between components of the IS and can be used to improve the automation of the deployment of remediations. In [33], Toth and Kruegel present a dependency model that allow to determine the impact of remediations on the whole system. In [14], Kheir et al. propose a framework modeling dependencies which handles both confidentiality, integrity and availability.

2.2 Attack graphs and attack paths

Attack graphs are a model regrouping all the steps an attacker may follow in an IS during an attack. It has been first introduced by Phillips and Swiler in [25]. It has been widely used ever since, thus many heterogeneous models are now behind the name *attack graph*. Generally, vertices (also called nodes in the literature) represent opportunities in an IS or actions that can be done by an attacker,

and edges (also called arcs in the literature) represent the dependency relations between the opportunities and/or actions. An attack graph can be built using information about the potential exploits that can be carried out on a network and using existing vulnerabilities databases [19] [20]. A summary of the state of the art on the early papers about attack graphs (from 2002 to 2005) has been done by Lippmann and Ingols in [17]; a more recent by Kordy et al. in [15].

MulVAL, The Multi-host, Multi-stage Vulnerability Analysis Language tool is an open-source attack graph engine created by Ou et al. [24]. It uses Datalog, a logic programming language, in order to generate an attack graph in which nodes are related to each other with logical relations (OR or AND). The other two main attack graph engines are commercial products: Cauldron [11] (originally presented by Jajodia et al. in [12]) and Artz's NetSPA [3].

An attack graph contains targets and multiple paths to reach them. Such attack paths can be extracted from the graph using different algorithms, according to the needs. Swiler et al. describe in [31] shortest paths algorithms used to find the most likely or lowest cost attack paths. In [26], Sawilla and Ou present a generalization of Google's Page-Rank algorithm to identify the most critical attack nodes. An attack path extraction and scoring function has also been presented by Bettan et al. in PoSecCo, a FP7 European Research project [4].

2.3 Selecting remediations

Remediations to an attack can be regrouped in three types: corrective, active and passive. The first one is the correction of the exploitable vulnerability. This is generally implemented by patch management software. The technology is quite mature (several tools exist, vendors regularly propose patches for their software) but it suffers from limitations detailed by Cavusoglu et al. in [5]. The most important is that patch deployment still requires human intervention: each patch must be tested on all platforms to prevent conflicts or regressions before being applied. The active remediations regroup those that prevent the exploitation of a vulnerability that still exists after the deployment. This is for example the case of simple filtering by a firewall or an Intrusion Prevention System (IPS). An IPS blocks flows that has been flagged as abnormal thanks to a signature or due to its statistical behavior [30]. More generally, an Intrusion Response System (IRS) is a system that provides other types of responses to a detection. This is a currently active research topic and many papers treat this subject, as summarized by Shameli-Sendi et al. in [27]. Finally, the passive remediations regroup the detection of the exploitation of a vulnerability and its report. This is a last resort but is widely used, as it can help security operators to know what happened in their IS. A system that only provides passive responses (alerts, reports, logs...) is generally called Intrusion Detection System (IDS) [34].

Some papers propose techniques to compute or select remediations using attack graphs. In [6], Cuppens et al. describe how the LAMBDA language, that has been used to model attacks, can also be used to model counter-measures and using the concept of anti-correlation allow to compute the appropriate counter-measures for an attack. In [35], Wang et al. base their analysis on the initial

conditions of an attack graph to compute all hardening options for a network. This approach has been improved by Albanese et al. in [2] with a near-optimal algorithm more efficient and cost-sensitive. In [23], Noel and Jajodia describe several methods to prioritize the deployment of patches depending on an attack graph. In [9], Ingols et al. explain how they represent three types of counter-measures (Firewall rules, IPSs and proxy firewall) in the NetSPA attack graph. Attack graphs are also used by Noel and Jajodia [22] to compute the optimal locations to deploy IDSs in an IS: they allow to minimize the cost of sensors while keeping a complete coverage of potential attack paths.

2.4 Ranking remediations

There are sometimes several remediations that could be deployed for a detection. Thus, they should be ranked to select the most appropriate one. Generally, this evaluation contains an estimation of the impact of a remediation and thus rely on a functional model as presented above. The ranking method presented by Toth and Kruegel in [33] first uses an algorithm that evaluates the impact of the remediations. Then, it advises to select the remediation with minimal negative effect on legitimate users. In [14], Kheir et al. present how to compute an index called RORI which represents a return on investment of a remediation. It is based on a dependency graph where are propagated the levels of Confidentiality, Integrity and Availability of assets. This index can then be used to rank remediations. An other parameter that can be taken into account when selecting a remediation is the impact of such counter-measure against the success likelihood of the attack. This kind of approach has been presented by Kanoun et al. in [13], where they implement a model based on dynamic Markov Models to assess the success likelihood of attacks. This model allow to select the most appropriate counter-measures to prevent an attacker from reaching its objectives.

Selecting a remediation among many brings challenges to overcome because it needs the knowledge of many parameters (costs parameters, functional and topological models) and how to combine them. It also requires assumptions regarding the coverage of the remediations on the attack.

3 Attack paths and preconditions

3.1 Attack path representation

Definition 1. *A **logical attack graph** G is a directed AND-OR graph represented by $G(V, A)$ where:*

- *V is a set of vertices that describe logical facts. Each vertex could be an* AND *(respectively an* OR*) vertex, meaning that this vertex needs the* conjunction *(resp. the* disjunction*) of its incoming arcs to be true.*
- *A is a set of directed arcs that represent a logical dependency from the child vertex to the parent one.*

In an attack graph as defined above, it is possible to choose attack targets. These are the vertices describing important final steps for an attacker. Based on these targets can be built attack paths using a bottom-up approach from the target to the upper preconditions of the attack graph. They can be ranked according to their impact and probability of occurrence, but this is a full-fledged subject that has been, for example, described in [4] or [26].

Definition 2. *An **attack path** is an acyclic and* logically valid *subgraph of an attack graph with one* target *and several* preconditions*.*

Definition 3. *The **target** of an attack path is the vertex whose outdegree is 0,* $\deg^+(v) = 0$ *(no outgoing arcs).*

Definition 4. *A **precondition** in an attack path is a vertex whose indegree is 0,* $\deg^-(v) = 0$ *(no incoming arcs).*

Definition 5. *A subgraph S of an attack graph G is **logically valid** if S contains at least one vertex and for each vertex $v \in S$, $v \in G$ and if* $\deg^-_G(v) > 0$ *(more than one incoming arc in G):*

- *if v is an AND, all the parents and incoming arcs of v in G are in S,*
- *if v is an OR, at least one parent of v and its incoming arc in G is in S.*

An attack path may have several intermediate goals but has only one main goal: the target of the attack path. It contains one, several or all the possible paths in the attack graph allowing to compromise this target.

3.2 Remediations can be applied only on preconditions

Proposing remediations to an attack path is searching the means to prevent the attacks and protect its target. An attack path is a logical graph: a fact is true if and only if the conjunction or disjunction of its parents is also true. As a precondition does not have any parent, it is not deducted from any other vertex. So, they are the first conditions from which all other vertices are deducted and thus the only vertices where can be applied remediations. This is the basic assumption on which we will build our remediation method.

Sufficient preconditions The attack path contains one target that should be protected by the remediations. As the attack path is an AND-OR graph, it is possible to compute all conjunctions of to-be-remediated preconditions, sufficient to protect the target. This logical expression SP can be represented with a set of disjunctions containing conjunctions as following:

$$SP = \bigvee_i SP_i = \bigvee_i \bigwedge_j SP_{i,j} \tag{1}$$

where $\bigvee$ is logical OR, $\bigwedge$ is logical AND, SP_i is a conjunction of preconditions sufficient to protect the target (i indexing the conjunction of preconditions) and $SP_{i,j}$ is a precondition to remediate (j indexing the preconditions).

Fig. 1. Recursive algorithm computing the conjunctions of sufficient preconditions

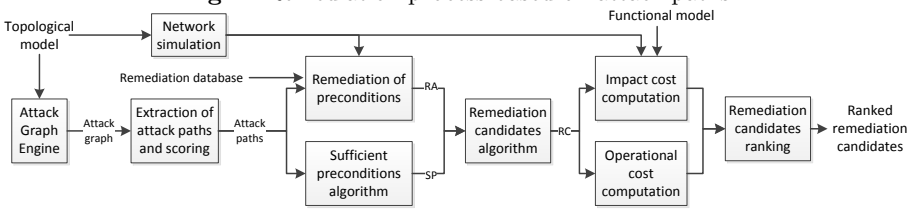
```
1: function computeSP(Vertex v)        ▷ Returns the list SP to delete the vertex v
2:     if v is a precondition then                                          ▷ v has no parent
3:         return [[v]]                                                     ▷ it is the only precondition
4:     else if v is an AND then
5:         res ←[[]]
6:         for each parent p of v do
7:             res ← res;computeSP(p)
8:         end for
9:         return res                                                       ▷ ⋁_{p∈{parents of v}} computeSP(p)
10:    else if v is an OR then
11:        res ←computeSP(first parent of v)
12:        for each other parents p of v do
13:            res ←conjunctionOfSets(res,computeSP(p))
14:        end for
15:        return res                                                       ▷ ⋀_{p∈{parents of v}} computeSP(p)
16:    end if
17: end function
18: function conjunctionOfSets(A, B)   ▷ Makes the conjunction of sets A and B
19:    result ← [[]]                   ▷ A and B are Or/And sets: A = ⋁_i A_i, A_i = ⋀_k A_{i,k}
20:    for i = 1 to size(B) do
21:        buildingResult ← A
22:        for j = 1 to size(buildingResult) do
23:            buildingResult[j] ← buildingResult[j]; B[i]
24:        end for
25:        result ← result; buildingResult
26:    end for
27:    return result                                                        ▷ (⋁_i A_i) ∧ (⋁_j B_j) = ⋁_{i,j}(A_i ∧ B_j)
28: end function
```

In fact, computing SP is identical to find all the conjunctions of preconditions sufficient to delete the target vertex according to the AND/OR formalism.

Definition 6. *A conjunction of precondition SP_i is **sufficient** to delete the target t of an attack path AP if the deletion of each precondition $SP_{i,j} \in SP_i$ and its propagation in AP implies that henceforth $t \notin AP$.*

The recursive algorithm that computes SP can be found in Figure 1. It should be called on the target of the attack path and will go up recursively along the arcs. All conjunctions of preconditions computed with this algorithm allow to prevent the access to the target, if remediated. Of course all preconditions can not be remediated. If this is the case in a conjunction, the entire set of preconditions will not be usable to successfully protect the target of the attack path.

Fig. 2. Remediation process based on attack paths



4 Remediation of an attack path

A diagram summarizing the remediation process can be found in Figure 2.

4.1 Remediate a precondition

A precondition contains a logical fact describing what can be used by an attacker. We detail real preconditions and their remediations in Subsection 6.3, but will first describe two general methodologies that can be applied to preconditions.

Simple remediations to preconditions The first case appearing when remediating a precondition is a simple remediation that can be applied to negate this precondition. This usually corresponds to the first type of remediation presented in the state of the art. You need a database that makes the link between the precondition (eg: the vulnerability) and how you can remediate it (eg: a patch).

Remediations using the network topology Some remediations require more advanced techniques. This is the case of those which try to prevent the *exploitation* of a vulnerability. It corresponds to the second type of remediation of the state of the art. Computing such remediations requires an accurate knowledge of the flows exchanged on the network and thus need to simulate the network topology.

4.2 Remediation candidates for an attack path

Each remediation potentially contains several elementary actions. More formally, for each attack path, we have succeeded to compute:

- A disjunction of conjunctions of **sufficient preconditions** to remediate, in order to protect the target of the attack path: SP
- A disjunction of conjunctions of **remediation actions** sufficient to prevent a precondition p:

$$RA(p) = \bigvee_i RA_i(p) = \bigvee_i \bigwedge_j RA_{i,j}(p) \tag{2}$$

  where $RA_i(p)$ is a conjunction of remediation actions allowing to prevent the precondition p (i indexing each conjunction) and $RA_{i,j}(p)$ is a remediation action (j indexing each action of the conjunction) each remediation action $RA_{i,j}$ is constituted of the tuple (action to apply, machine to deploy it).

Fig. 3. Algorithm computing the remediation candidates

```
1: function computeCandidates(SP, RA)   ▷ Returns all remediation candidates protecting an attack path.
2:     res ←[[]]
3:     for SP_i in SP do
4:         res ← res ; computeRemedsToPreconds(SP_i, 1, RA)
5:     end for
6:     return res
7: end function
8: function computeRemedsToPreconds(SP_i, j, RA)   ▷ Returns all conjunctions of actions allowing to remediate the preconditions of SP_i starting from j.
9:     SP_{i,j} ← SP_i[j]                          ▷ j^th precondition to remediate
10:    RA_j ← RA[SP_{i,j}]                         ▷ Remediations of j^th precond
11:    if empty(RA_j) then                         ▷ j^th precondition can not be remediated
12:        return [[]]
13:    else
14:        if j=size(SP_i) then                    ▷ Terminaison of recursion
15:            return RA_j
16:        else
17:            RA_{j+1..n} ←computeRemedsToPreconds(SP_i, j + 1, RA)
18:            return conjunctionOfSets(RA_j, RA_{j+1..n})
19:        end if
20:    end if
21: end function
```

We need to combine SP and RA to compute a disjunction of **remediation candidates** containing the actions that allow to protect the target:

$$RC = \bigvee_i RC_i = \bigvee_i \bigwedge_j RC_{i,j} \tag{3}$$

where RC_i is a remediation candidate (i indexing the number of candidates) and $RC_{i,j}$ is a remediation action (j indexing the number of actions in the candidate).

An algorithm computing such remediation candidates is shown in Figure 3.

5 Costs of remediations

The last essential point for our remediation method is to estimate the cost of a candidate. This will help an operator to choose between several candidates remediating the same attack path. We have identified two principal sources of cost that must be considered: the operational and the impact costs.

5.1 Operational cost

The first important cost for an operator deploying a remediation is the operational cost (OC). It represents the difficulty to implement the remediation on

the assets and to maintain it. For each remediation action $RC_{i,j}$ that should be applied, we identified four categories in which this cost can be split.

1. **Remediation cost (RC)**: This is the cost of the input necessary to apply the remediation, for example, the price of a patch or of a signature.
2. **Deployment costs (DC)**: This is the cost representing the workload to apply the remediation on the concerned machine.
3. **Test costs (TC)**: This is the cost to test that all important features of the Information System are still working as expected, after the deployment.
4. **Maintenance costs (MC)**: This is the cost per year of the maintenance induced by the remediation. It reflects for example the increase of CPU, memory, storage and treatment of logs that will be induced.

These elements can be expressed in a monetary unit and we detail in Subsection 6.5 how to compute them. The operational cost of a remediation action is the sum of all these elements as shown in Equation 4.

$$OC(RC_{i,j}) = RC + DC + TC + MC \tag{4}$$

To simplify the estimation of the operational costs of a candidate containing several remediation actions, we made the assumption that these actions are independent. This assumption has been introduced and justified by Gonzalez-Granadillo et al. in [8] as Axiom 1. This is moreover the most common case, as the remediation actions are usually deployed on different machines or are of different types. With this assumption, the operational cost of a candidate is the sum of the operational costs of its actions, as shown in Equation 5.

$$OC(RC_i) = \sum_j OC(RC_{i,j}) \tag{5}$$

5.2 Impact cost

We propose here a basic impact cost (IC) function that measures the loss due to the unavailability of services after the deployment of a remediation. This function uses a list of (1) dependencies of business applications toward services, (2) services toward network accesses and (3) interdependencies between services. In addition to this are added cost values related to the temporary and permanent unavailability (UC) of business applications.

Thanks to those parameters, we can compute the cost of unavailability of all business applications before (on the real system) and after (on a simulated system) deploying a candidate RC_i, by checking recursively that the services dependencies are verified as shown in Equation 6.

$$IC(RC_i) = \sum_{ba \in businessApp} isImpactedBy(RC_i, ba) * UC(ba) \tag{6}$$

The impact cost is certainly the most important part to take into account when deploying a remediation but it is perhaps also the most difficult to quantify,

as it is hard to estimate the cost if a business application is unavailable and to know which applications will be disrupted by a remediation candidate. It is therefore very important to provide security operators with indications about such a cost, to help them choose at best the remediation to deploy.

5.3 Ranking remediation candidates

These costs allow us to attach a global cost C to a candidate as shown below:

$$C(RC_i) = OC(RC_i) + IC(RC_i) \tag{7}$$

The candidate cost function can be considered as a ranking function taking as input an unsorted set of remediation candidates and that outputs a set of the same candidates sorted according to their cost. This allow a security operator to select one of the candidates that has the lowest cost.

As the remediation candidates cost function is only used to compare candidates with each other, even if the cost parameters are not assigned exactly, it does not change significantly the order between them. Thus, the details of the cost models parameters are not required, but only need to represent a tendency, in order to conserve the ranking between candidates. This assumption has also already been justified by Gonzalez-Granadillo et al. in [8].

6 Implementation

6.1 Simulation of the network topology

As we need a simulated network topology representing the target IS, we created a network simulator that accurately reproduces simple network behaviors of hosts: we are able to simulate exchanges between hosts, calculate routes, test if a packet can pass firewall rules, etc. We designed a pivot file in which we put the topological information needed by the simulator. We also created connectors to automatically build such a file. The first connector we built was a python server that gathered the topological information collected by agents deployed on Linux machines into the pivot format. The second connector was built for the European Research Project PoSecCo [1], where we had an ontology containing the network topology. We thus implemented a connector that was querying in the ontology for the information needed.

6.2 Generation of attack paths

We use the open source attack graph engine MulVAL [24]. It requires three types of inputs: topological, filtering and vulnerabilities information. We combine our simulated topology with a vulnerability scan (of Nessus [32]) by merging the information about services and their vulnerabilities extracted from the scanner report into our topology. MulVAL inputs are stored in a file using the Datalog language. It outputs an XML file containing the logical attack graph computed thanks to its engine from which the attack paths are extracted.

Table 1. Main MulVAL preconditions and their remediations

Preconditions	Description	Possible remediations
hacl(src, dst, port, portocol)	The host *src* has access to *dst* on *port* using *protocol*	Deploy a firewall rule
vulExists(host, vulID, program)	*program* on *host* has a vulnerability *vulID*	Apply a patch or deploy a Snort rule
networkServiceInfo(host, program, protocol, port, user)	*program* on *host* launched as *user* open *port* using *protocol*	Stop this network service
hasAccount(user, host, account)	*user* has *account* on *host*	Disable this account

6.3 Preconditions in MulVAL and their remediations

The main preconditions proposed by MulVAL to model attacks and their remediations can be seen in Table 1. We will focus here only on three relevant types of remediation for an enterprise: applying a patch, deploying a rule on an IPS (preventing *vulExists()*) and deploying a firewall rule (preventing *hacl()*).

Application of a patch In order to propose the right patch to a vulnerability, we use the parameter in the fact of the precondition *vulExists* containing the identifier of a vulnerability, generally a CVE (Common Vulnerabilities and Exposures) [20]. We use this identifier to look for known patches in the remediation database we describe in Subsection 6.4.

Deployment of a firewall rule To compute the firewall rule that should be deployed, we use all the parameters of the fact *hacl(src, dst, port, protocol)*. This precondition explains the network access the attacker needs for his attack. So, it should be negated by the rule to deploy, which should have the following form:

DROP FROM src TO dst:port USING protocol

It can be generated according to the type of firewall aimed. For example, we propose an automatic generation of iptables [18] firewall rules.

The last problem we need to deal with for the firewall rules proposal is where it should be deployed. We use here the topology simulation presented in Section 4.1 to determine the route followed by packets between *src* and *dest:port*. We then deduce on which machine the firewall rule can be deployed.

Deployment of an IPS rule The last type or remediation we will detail is the deployment of IPS rules for Snort [29] which prevent the exploitation of a vulnerability. For each *vulExists* related to an *hacl*, we can know (1) The Snort rules that may exist to prevent the exploitation of the vulnerability by searching its identifier in the remediation database presented in Subsection 6.4, and (2) the network routes that may be used by the attacker to exploit this vulnerability, by using the simulation of the network and a deduction process similar to the calculation of the firewall rules. On each route, we must have an IPS host where we can deploy the rule. Otherwise, the remediation is not possible. The rules we propose here must be used with Snort in inline mode and they begin with the *drop* keyword, meaning that we use it as an IPS.

6.4 Filling the remediation database

One challenge of the proposition of remediation is the ability to build automatically a remediation database. We will describe here how we overcome it.

Database model We use a relational model stored in a SQLite file. We choose to use a model similar to the one used in the National Vulnerability Database [21] to represent vulnerabilities. Then, we added two tables corresponding to the types of remediations. In order to have a N-to-N association with the vulnerabilities, we also add a join table for each kind of remediation.

Patches We used the NVD [21] to find the links toward patches that correct vulnerabilities. Among the attributes related to a CVE, a *reference* can point to a website describing how to patch the vulnerability. So, we parse the dumps of the NVD, extract the links toward patches and store it in the database. Around 20% of the CVE have a "PATCH" reference attached.

Snort rules In the standard format of a Snort rule, there is an option "reference" which often contains a CVE. In the freely available database of rules provided by Sourcefire [29], nearly 50% of the rules are related to a CVE.

6.5 Providing the costs parameters

Operational cost Operational costs depends highly on the company and on the remediation. So, we choose in our prototype to assign parameters per types of remediation. Generally the difference of operational cost between remediations of the same type is low, but it may be also possible to add the cost parameters into the remediation database, in order to be able to attach to each remediation specific operational cost parameters.

Impact cost The description of dependencies used for impact cost is also totally dependent on the IS and has to be provided by the security operator. To describe these dependencies, we use an XML file in which the dependencies relations are described according to a dependency graph.

7 Experiments and results

In order to validate the whole remediation method described in this paper, we applied it on several test topologies. We implemented it on the use-cases of the European Research Project PoSeCo [4]. But before detailing this test-bed and our experiments on it, we will present a simpler scenario implementing the main concepts. We will end with a discussion about the complexity of our approach.

7.1 Simple experiment scenario

Network topology and attack scenario The first scenario we implement rely on a topology that we deployed on virtual machines. It contains 5 Linux-based hosts: a web server, a database server, an administration machine inside

a LAN, a firewall that protects the LAN and the servers from the Internet, and the attacker's machine that is on the Internet. We configure the firewall in such a way that the web server is the only service that is accessible from the Internet and the LAN. The web server needs the database server to work properly and has a full access to it. The enterprise has two business applications using the IT: an Extranet which is rarely used and an Intranet which is used for all employees and is thus much more critical. The database server contains also some confidential information that the company wants to protect.

When the web server is exploited, the attacker can access the database server and with an other exploit can try to gain access to all the data it contains. This is the attack path that will be described in the rest of this scenario.

Generation of the attack path and proposition of remediations To collect the topological information, we use the python agents described in Subsection 6.1. We generate MulVAL inputs and launch the attack graph engine, then extract the attack paths and select the one presented above. It chains the exploitation of two vulnerabilities: the first one CVE-2004-1315 is on the web server, the second one, CVE-2012-3951, is on the database server.

We use our prototype to visualize the attack path to correct and the remediation candidates. We present here the four most relevant candidates, ranked by cost. We also explain for each candidate why its cost is low, medium or high.

1. The first candidate is the deployment on the firewall of the Snort rule sid:12610 that allows to block the exploitation of the first vulnerability. This remediation has a lot of advantages, because it doesn't interrupt any genuine service, is not too much expensive (deployment can be nearly fully automated), and blocks successfully the attack. This candidate has no impact cost, a low operational cost and thus a very low global cost, that is why this is the first one to be proposed.
2. The second one is the proposal of a patch to the first vulnerability. As the first one, it doesn't have any impact on normal service, but has much more operational cost, because deploying a patch need human intervention. So, this candidate has a low global cost and thus is the second one to be proposed.
3. The third one is a firewall rule that blocks all the traffic from the Internet to the web server on http port. It has a low operational cost, because it can be automatized, but has a medium impact because it cuts the access from the Internet to the web server, even if it keeps all the accesses from the LAN. So this candidate has a medium global cost and thus is the third to be proposed.
4. The last one is a firewall rule that blocks all the traffic to the web server on http port. It has also a low operational cost, but has a huge impact because it cuts also all the accesses for the employees of the LAN to the web server. So this candidate has a high global cost and thus is the last one to be proposed.

7.2 Results on PoSecCo's testbed

We will now present the results of this method applied on the testbed of the FP7 European Research Project PoSecCo [1].

The main use case in which the PoSecCo prototypes have been tested has two main business services: a broadcaster Internet distribution and a corporate streaming service. These services have several security requirements and run on a testbed that has been deployed during the project, on which prototypes have been tested. It contains around twenty machines (some are representing server farms) and eight routers. All the topological information needed for our prototype are collected from an ontology and the attack paths extracted are ranked according to their impact on security requirements.

On the twenty machines and eight routers, there are more than a thousand vulnerabilities in total. It was chosen for this project that there will be one attack path per target, gathering the relevant ways to compromise it. Thus, after establishing a list of five hosts to protect in priority, the operator has five attack paths to assess and correct. These attack paths contain between thirty and hundreds of nodes, the possible remediations are computed in a few seconds. Due to project limitations, only two types of remediations are proposed: patches and firewall rules. For each attack path, many candidates are proposed (up to a hundred), and are ranked according to their operational and impact cost. The first candidates (lower cost) offer the best compromise between efficiency and cost and should be the best option for a security operator.

During the project, the prototype implementing this approach was presented to end-users that compared their risk analysis and its remediation, in anticipation of a change in the testbed topology, with and without the prototype. The end-users, independent of the project, concluded that the scenario using this methodology was much more efficient: it reduces the analysis from four hours to twenty two minutes and could reduce the number of people needed for this task from between three and six to only one. The result of this evaluation can be found in PoSecCo's Deliverable 1.7, in scenario SP06 [7].

7.3 Complexity

What must be well understood before talking about the complexity of our algorithms is that in this paper, we propose remediations to attack *paths* and not to a whole attack *graph*, we thus have smaller complexity issues. Indeed, an attack graph is usually a large graph whereas, an attack path is smaller, because it focuses only on the very impactive or the most likely ways to access a target.

The complexity of the algorithm computing the candidates is not very impactive, because 1) it is linear in the number of conjunctions of precondition and 2) the number of remediations for one precondition is generally low. The algorithm computing SP depends highly on the structure of the input attack path, especially on the number of parents of each vertex. In the best case (each vertex has only one parent) this algorithm is linear in the number of vertices. In the worst case (each vertex has several parents), the complexity is exponential in the number of parents of OR vertices. This is the factor that most influence the complexity. We made simulations on non-realistic graphs with different varying parameters (number of parents, OR nodes, AND nodes, preconditions...) to validate these results. Nevertheless, in practice on several real use cases, we found

that the number of parents for OR vertices in attack paths is generally low: an average of 1.7 per OR vertex in attack paths produced by MulVAL. This can be simply explained knowing that, in an attack path, as explained above, we only have few different possibilities to compromise a target, alternatives creating disjunctions in the attack path. It implies that this methodology generally scales well, if the attack paths treated are properly generated.

8 Related Work

The papers which describe the closest approach to our work are [35] and [2]. In [35], Wang et al. base their analysis on the preconditions of an attack graph to compute ways to prevent attacks. But even when they evocate the cost to choose one remediation solution rather than another, they do not present a cost function to sort candidates as we did in this paper. In [2], Albanese et al. extend [35] mainly by adding a cost model, similar to the one we present here, and by improving the complexity of the algorithm to compute candidates. But what distinguish our approach from both these ones is that we do not compute remediations to an attack *graph* but to attack *paths*, meaning that our algorithms are working with smaller inputs. We are convinced that it is much more sound and efficient to correct only the paths that are significant rather than reasoning on the global attack graph. This was assessed on realistic use-cases.

What is also original in our approach is that our remediation computation is generic. In [23] and [22], Noel and Jajodia propose various types of remediations to predefined types of attacks modeled by attack graphs. The method we present here is more generic. Indeed, the expressiveness of logical attack graphs allows the modeling of every attack described with AND/OR conditions and our method applies to all of them, without the limitations identified in the related work. Dealing with new attacks only imply to define remediation for potential new kinds of preconditions. These remediations can be simple or may require network topology simulation, as the ones presented in this paper.

Furthermore, several databases that contain remediations exist. However, each database is dedicated to a type of remediation. For example, the NVD contains information about patches [21] and Snort databases contain only Snort rules [29]. In this paper, we design a remediation database and fill it using several online available sets of data. This database contains different kinds of remediations and can be extended to provide new types of remediations.

9 Conclusion

We present in this paper a method describing how to compute remediations for scored attack paths extracted from an attack graph. Attack graphs have been widely used for assessing the security level of an Information System, we chose instead to use them in order to propose solutions to enhance this security level, by computing remediations preventing attack paths in an Information System. Using scored attack paths extracted from an attack graph allows us to remediate

only the very likely or impacting paths that lead to main assets which is much more efficient than remediating the global attack graph.

We have stated that the only vertices on which we compute remediations, within a logical attack path, are the preconditions. We have implemented algorithms to cluster these nodes into conjunctions of sufficient preconditions to be remediated, in order to protect the target of an attack path. Then, after explaining how to compute remediation actions to prevent a precondition, we detailed their combination with the sufficient conjunctions of preconditions to determine the candidates. As the operator has to choose one remediation among several candidates providing the same remediation objective, we assign to each remediation a global cost combining operational and impact costs. To calculate topological remediations to certain preconditions and to assess the effects of remediations on the system, we have designed a simulated network topology.

Limitations of the logical model used for modeling attack graphs were detailed during this study since this model is deterministic and not dynamic. Thus, it has to be extended into a quantitative model to represent dynamic attacks and to model them more accurately. However, this logical model has the advantage to be efficiently generated and processed and is well suited to model potential attacks. Future work will study the computation of more complex remediations with the development of a more accurate cost function. A prerequisite will be a better knowledge of the IS through new mining techniques.

References

1. Posecco, `http://www.posecco.eu`
2. Albanese, M., Jajodia, S., Noel, S.: Time-efficient and cost-effective network hardening using attack graphs. In: Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on. pp. 1–12. IEEE (2012)
3. Artz, M.L.: NetSPA: A Network Security Planning Architecture. Ph.D. thesis, Massachusetts Institute of Technology (2002)
4. Bettan, O., Ponta, S., Musaraj, K., Casalino, M.: D4.8 - prototype: Standardized audit interface. Tech. rep., PoSecCo European Project from the 7th Framework (project no. 257129) (2012)
5. Cavusoglu, H., Cavusoglu, H., Zhang, J.: Security patch management: Share the burden or share the damage? Management Science 54(4), 657–670 (Apr 2008)
6. Cuppens, F., Autrel, F., Bouzida, Y., Garcia, J., Gombault, S., Sans, T.: Anti-correlation as a criterion to select appropriate counter-measures in an intrusion detection framework. In: Annales des télécommunications. vol. 61, pp. 197–217. Springer (2006)
7. Demetz, L., Maier, R., Manhart, M., Plate, H., Fitz, M.: D1.7 - final project evaluation. Tech. rep., PoSecCo European Project from the 7th Framework (project no. 257129) (2013)
8. Granadillo, G.G., Jacob, G., Debar, H., Coppolino, L.: Combination approach to select optimal countermeasures based on the rori index. In: Second International Conference on Innovative Computing Technology. pp. 38–45. IEEE (2012)
9. Ingols, K., Chu, M., Lippmann, R., Webster, S., Boyer, S.: Modeling modern network attacks and countermeasures using attack graphs. In: Annual Computer Security Applications Conference. pp. 117–126. IEEE (2009)

10. Jajodia, S., Noel, S.: Advanced cyber attack modeling analysis and visualization. Tech. rep., DTIC Document (2010)
11. Jajodia, S., Noel, S., Kalapa, P., Albanese, M., Williams, J.: Cauldron mission-centric cyber situational awareness with defense in depth. In: Military Communications Conference. pp. 1339–1344 (2011)
12. Jajodia, S., Noel, S., O'Berry, B.: Topological analysis of network attack vulnerability. Managing Cyber Threats pp. 247–266 (2005)
13. Kanoun, W., Dubus, S., Papillon, S., Cuppens-Boulahia, N., Cuppens, F.: Towards dynamic risk management: Success likelihood of ongoing attacks. Bell Labs Technical Journal 17(3), 61–78 (2012)
14. Kheir, N., Debar, H., Cuppens-Boulahia, N., Cuppens, F., Viinikka, J.: Cost evaluation for intrusion response using dependency graphs. In: Network and Service Security. pp. 1–6. IEEE (2009)
15. Kordy, B., Piètre-Cambacédès, L., Schweitzer, P.: Dag-based attack and defense modeling: Don't miss the forest for the attack trees. CoRR (Mar 2013)
16. Lagadec, P.: Visualisation et analyse de risque dynamique pour la cyber-défense. SSTIC (2010)
17. Lippmann, R.P., Ingols, K.W.: An annotated review of past papers on attack graphs. Tech. rep., DTIC Document (2005)
18. Netfilter: iptables, `http://www.netfilter.org/projects/iptables/index.html`
19. NIST: Capec, common attack pattern enumeration and classification, `http://capec.mitre.org/`
20. NIST: Cve, common vulnerabilities and exposures, `https://cve.mitre.org/`
21. NIST: Nvd, national vulnerability database, `https://nvd.nist.gov/`
22. Noel, S., Jajodia, S.: Optimal ids sensor placement and alert prioritization using attack graphs - springer. Journal of Network and Systems Management (2008)
23. Noel, S., Jajodia, S.: Proactive intrusion prevention and response via attack graphs. Tech. rep., Addison-Wesley Professional (2009)
24. Ou, X., Govindavajhala, S., Appel, A.W.: Mulval: A logic-based network security analyzer. In: Proceedings of the 14th conference on USENIX Security Symposium-Volume 14. pp. 8–8. USENIX Association (2005)
25. Phillips, C., Swiler, L.P.: A graph-based system for network-vulnerability analysis. In: the 1998 workshop. pp. 71–79. ACM Press, New York, New York, USA (1998)
26. Sawilla, R.E., Ou, X.: Identifying critical attack assets in dependency attack graphs. Springer (2008)
27. Shameli-Sendi, A., Ezzati-Jivan, N., Jabbarifar, M.: Intrusion response systems: survey and taxonomy. SIGMOD (2012)
28. Skybox security, i.: Skybox, `http://www.skyboxsecurity.com/`
29. Sourcefire: Snort, `http://www.snort.org/`
30. Stakhanova, N., Basu, S., Wong, J.: A taxonomy of intrusion response systems. International Journal of Information and Computer Security 1(1), 169–184 (2007)
31. Swiler, L.P., Phillips, C., Ellis, D., Chakerian, S.: Computer-attack graph generation tool. In: DARPA Information Survivability Conference and Exposition. pp. 307–321. IEEE (2001)
32. Tenable: Nessus, `http://www.tenable.com/products/nessus`
33. Toth, T., Kruegel, C.: Evaluating the impact of automated intrusion response mechanisms. In: CSAC. pp. 301–310. IEEE (2002)
34. Tucker, C.J., Furnell, S.M., Ghita, BV, Brooke, P.J.: A new taxonomy for comparing intrusion detection systems. Internet Research 17(1), 88–98 (2007)
35. Wang, L., Noel, S., Jajodia, S.: Minimum-cost network hardening using attack graphs. Computer Communications 29(18), 3812–3824 (2006)

Vers une architecture « big-data » bio-inspirée pour la détection d'anomalie des SIEM

Véronique Legrand[1,3], Pierre Parrend[2,3], Pierre Collet[2], Stéphane Frénot[1], Marine Minier[1]

[1] CITI, Laboratoire, INSA de Lyon, Villeurbanne, France.
{veronique.legrand, stéphane.frenot,marine.minier}@insa-lyon.fr
[2]iCUBE, Laboratoire UMR7357, Strasbourg, France.
{pierre.parrend, pierre.collet}@unistra.fr
[3]ECAM Strasbourg-Europe, Schiltigheim, France.
pierre.parrend@ecam-strasbourg.eu
[4] INTRINSEC, www.intrinsec.com, Nanterre, France.
{veronique.legrand }@intrinsec.com

Abstract. Pour assurer la surveillance des systèmes d'information (SI), les systèmes de gestions des évènements et informations de sécurité actuels (SIEM) reposent sur un modèle en trois phases qui traite les évènements émis par les sondes de surveillance (logs). Or, de nombreux évènements issus d'attaques complexes ciblées ne présentent pas de structure simple, si bien que la définition d'une signature d'attaque demeure une question difficile. Le recours à la connaissance des experts est donc indispensable. On s'intéresse ici à la modélisation qui permettra de détecter, d'identifier et de caractériser des attaques complexes. Cette modélisation constitue l'un des principaux problèmes qui entrave le développement de SIEM. Cet article présente une analyse des enjeux des SIEM, des techniques d'apprentissage de la connaissance experte et de la fouille de données associée. Il identifie les voies qui méritent d'être plus explorées comme les ontologies et les treillis de connaissances afin d'abstraire les raisonnements afin de guider les experts dans leurs actions.

Keywords. SIEM, sécurité, connaissances, architecture cognitive, détection d'anomalie, sécurité, détection d'anomalie.

1 Introduction

Ces dernières années, toutes les sphères de l'Internet, qu'elles soient marchandes, industrielles, privées ou même publiques ont accru considérablement leur dépendance aux technologies et leur exposition aux failles de sécurité. Afin d'assurer la surveillance de l'ensemble de leurs activités d'échange, elles ont dû déployer une multitude de sondes, chacune adaptée à un type de système et à la nature des anomalies surveillées émettant une multitude d'alertes destinées aux analystes. Chaque jour, le volume d'alertes à analyser ne cesse de croitre de façon inquiétante. Selon les analystes, ce phénomène sera encore appelé à s'amplifier, en 2016, 40% des entreprises - contre moins de 3 % en 2011[1] - devra analyser activement au moins 10 téraoctets d'alertes de sécurité. On assimile d'ores et déjà ces systèmes de surveillance à des systèmes « big-data » devant traiter des flux riches en volume, en variété, en vitesse et à grande variabilité.

Les attaquants ont eux aussi tiré parti des flux « big-data » pour affiner considérablement leur phase d'ingénierie sociale, l'un des facteur-clé de leur réussite puisqu'ils peuvent développer des attaques ciblées, en contexte, distribuées dans le temps et dans l'espace pour se rendre moins perceptibles. De telles attaques sont devenues redoutables [1][6][8]. Pour détecter et analyser de telles attaques à l'échelle globale, un SIEM apporte une réponse globale en centralisant l'ensemble des alertes émises par les sondes.

Sous l'angle de la gestion d'un SI, un SIEM est un système de gestion des informations de sécurité et des évènements (« logs »), il constitue à ce titre un outil d'aide à la décision pour accompagner le travail de l'analyste de sécurité face aux menaces qui pèsent sur le SI. Un SIEM incorpore plusieurs fonctions de surveillance, mais pour accomplir la fonction de détection d'anomalie, un SIEM incorpore comme une sonde des connaissances à l'échelle globale, sous forme de modèles ou de signatures.

Dans cet article, nous présentons tout d'abord en section 2 un état de l'art des travaux actuels sur la modélisation de la connaissance pour les SIEM, puis, nous décrivons en section 3 notre modèle «Knowledge and Information Logs-based System ».

2 Travaux sur la modélisation des connaissances pour un SIEM

La plupart des travaux de recherche pour les SIEM se focalise encore aujourd'hui sur la fonction de détection. Pour accomplir une telle fonction, les SIEM s'appuient sur l'analyse globale des journaux issus de tout horizon, c'est en effet par ces journaux que parviennent les anomalies mais également les informations de contexte précieuses pour raisonner en situation. Dans cette section, nous présentons le principe général d'un SIEM, ensuite, nous expliquons le rôle joué par le modèle de connaissances pour l'analyse, puis, nous abordons un état des techniques de modélisation des connais-

[1] Neil MacDonald, Information Security Is Becoming a Big Data, Analytics Problem, Gartner whitepaper, 03/2012

sances appliquées aux SIEM en décrivant comment les experts de sécurité modélisent et transmettent à la machine leurs connaissances sur les modes opératoires d'attaques.

2.1 Principe général de la détection d'anomalie pour un SIEM

Un SIEM (**Fig. 1**) suit un processus en plusieurs phases ; en entrée, il collecte et analyse les informations et évènements de sécurité sous forme de « logs », et, en sortie, il adapte et applique sur les systèmes défaillants des actions correctives ciblées.

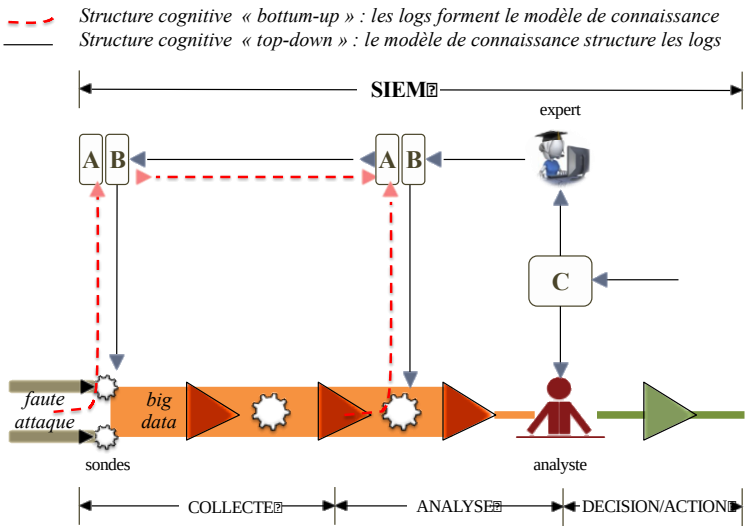


Fig. 1. *Processus général de surveillance d'un SIEM*

(A): règles et connaissances apprises par IA, (B) : règles et connaissances de conception ou de configuration : règles de firewall, (C) : règles et connaissances préalables externes : normes, expériences,…

Au niveau de la phase de collecte, les « logs » des sondes de surveillance sont recueillis et centralisés au sein de la base du SIEM. Cette phase est généralement bien automatisée par les SIEM du marché, mais elle mériterait d'être précédée d'une phase plus stratégique de défense en profondeur déployant judicieusement les points de détection, ce qui étendrait le champ d'observation des analystes et augmenterait la variété des « logs ».

Au niveau de la phase d'analyse, une suite d'opérations plus ou moins automatisées est conduite par les analystes de sécurité dans le but de détecter des anomalies à partir des « logs ». Afin de répondre au mieux et en un temps très court, les analystes peuvent compléter leurs opérations de détection par des investigations. Nous présentons les SIEM en abordant successivement les flux, le moteur d'analyse et les règles.

2.1.1 Analyse des flux de « logs »

Les flux d'informations et d'évènements de sécurité proviennent des sondes de surveillance, leurs caractéristiques peuvent les faire assimiler à des flux « big-data ». Tout d'abord, ils apparaissent de façon massive et à l'échelle des entreprises ou des opérateurs, ces volumes sont de l'ordre de plusieurs milliers d'évènements par seconde (eps). A titre d'exemple, le challenge VAST [14] a estimé à plus de 800 000 « logs » par jour le volume moyen d'une interconnexion faite d'une ferme de serveurs web, de messagerie, DNS et d'authentification. La variété des « logs » s'avère très élevée, de l'ordre de 100 à 200 par jour selon les formats, langages et codes des sources les ayant générés.

Chaque « log » porte un message dont le sens contribue à deux niveaux de décision : l'analyse de l'incident, l'action corrective, de façon plus ou moins efficace selon :

- la nature de la source qui l'a émis, les messages sont: i) plutôt génériques si la sonde s'intègre à des composants systèmes du SI, (messages disque, mémoire (**Fig.3-b**), « syslog »), ii) plutôt spécifiques si la sonde est une fonction spécifique (message d'intrusion fourni par un mécanisme d'authentification (**Fig.2**)),
- le niveau d'information sur l'action à appliquer, les messages sont : i) des alertes et renferment des éléments liés à des fautes pour organiser une «réaction rapide », les fautes peuvent être de différente nature : de gestion (« service stoppé »), d'usage (« échec ou succès de l'ouverture d'une session » (**Fig.2,3-b**)) ou même illicite, comme une attaque, enfreignant des règles de sécurité ou comme une vulnérabilité ouvrant à des attaques (« contournement d'un système d'authentification »), ii) des informations et renferment des éléments liés à des changements d'état ou de configurations (**Fig.3-a**) - la « création ou la modification d'un fichier », il s'insère généralement dans une « activité post-mortem », par exemple, journalisés pour des opérations d'investigation.

Par ailleurs, un message repose sur une structure en champs variables dont on distinguera deux niveaux (**Fig.2**) :

- fortement structuré (rose) pour spécifier les caractéristiques spatio-temporelles de l'activité : l'horodatage (1,2), l'auteur (4,5), la localisation de la cible (6),
- faiblement structuré (blanche), indépendant des conditions précédentes pour décrire la nature de l'action à l'aide d'un langage naturel (3). Par exemple, le champ « Classification Text » ((**Fig.2-3**) du protocole IDMEF [4] constitue une variable faiblement structurée.

Fig. 2. *Message de type « login » extrait d'une sonde Radius*

Ainsi, de par leur volume, leur vélocité et leur variété, les « logs » requièrent des espaces de stockage et des capacités de traitement considérables, néanmoins, les messages qu'ils portent constituent des éléments essentiels de l'analyse de sécurité.

2.1.2 Moteur d'analyse pour la détection

Au sein de l'informatique décisionnelle [27], le moteur d'analyse du SIEM traite des journaux d'évènements en trois étapes : Normalisation, Agrégation, Corrélation (NAC) afin de traiter de grands volumes de données hétérogènes[1][8][15].

La normalisation a pour but de transformer une donnée non structurée en une donnée structurée. La nature hétérogène des évènements de sécurité a entrainé le besoin d'un standard : l'IETF a retenu IDMEF [4] et IODEF [3] alors que l'ISO a proposé CIM [21]. Même si de par son modèle « objet », CIM est plus puissant, son usage reste limité et IDMEF s'est imposé comme le standard de fait, de par sa simplicité et sa capacité d'expression. Néanmoins, la plupart des SIEM du marché normalise des champs de variables fortement structurées. Le champ IDMEF, descriptif de l'action, est faiblement structuré, il s'intitule le champ « Classifucation Text » [4]. L'agrégation regroupe plusieurs évènements, qui une fois normalisés, présenteraient des similarités sur un ou plusieurs champs afin de réduire le volume à analyser. Enfin, la corrélation regroupe des évènements selon leur appartenance à un même scenario, plan ou profil, qu'ils soient d'usage ou d'attaque.

La fonction en sortie du processus NAC permet d'informer de la situation du SI en proposant à l'analyste une alerte d'agrégation ou de corrélation ; celle-ci doit guider l'expert à mieux cibler son intervention. Mais en situation de surveillance, face à plusieurs téraoctets d'informations, les activités devront être simplifiées en représentant l'anomalie à l'aide d'une information simple et efficace. Les notions « d'indicateur » ou de « visualisation graphique en mode radar » suggèrent toutes deux des représentations simplifiées des connaissances, les SIEM préfèrent synthétiser les résultats sous forme de tableaux de bord pour la sécurité [2]. Face à la détection d'attaques ciblées, cette question devient un véritable défi pour l'environnement « big-data » et en particulier, la représentation des résultats issus de l'analyse.

Ainsi, la présence massive de « logs » constitue-t-elle un avantage pour l'analyste : plus il y aura d'alertes pour décrire des activités, plus il y aura d'indices pour décrire un incident et l'avérer, néanmoins, le succès de ces étapes repose aujourd'hui sur sa capacité à relier ces « logs » et leurs variables.

2.1.3 Principe des règles pour les SIEM du marché

Les SIEM du marché, propriétaires ou « open source », emploient généralement des connaissances de type « signature » ou « à règles ». Cette approche engage l'entreprise qui le déploie dans un mécanisme lourd en temps et en ressources. En effet, pour exprimer et transmettre ces règles, les experts combinent leurs connaissances de programmation et de sécurité, ce qui accroit les temps de développement.

Un SIEM opère à la façon d'un système expert [1][15][17] en deux phases :

- la phase préalable s'appuie sur un moteur d'apprentissage ou d'acquisition. Cette phase consiste à modéliser sous forme de règles les connaissances des experts de sécurité humains en vue de les acquérir ou de les apprendre. Elles concernent les comportements des systèmes, des usagers ou des attaquants vis-à-vis de la sécurité, des liens entre « logs » résultant d'actions ou de séquences d'actions. Le modèle de type A ou/et B (**Fig. 1**) sera décrit en 2.2 et 2.3 ci-dessous
- la phase en situation s'appuie sur un moteur d'analyse. En sortie, cette phase détecte des valeurs déviantes. Pour ce faire, le moteur d'analyse applique les règles acquises (modèle A ou B) au préalable pour rechercher les liens susceptibles d'exister avec d'autres occurrences de « logs ». Ces événements sont réunis sous forme d'une séquence pouvant être unitaire en testant des règles d'agrégation et/ou de corrélation à chaque étape : si la valeur de la séquence correspond à un modèle,

un indice spécifie le processus appliqué (**Fig.4**) : par exemple, **a** pour l'agrégation (lignes `1` et 2) ou **c** pour la corrélation (lignes `5,6` et `8`).

Les moteurs d'analyse des SIEM rencontrent aujourd'hui des difficultés dans la qualité de leurs analyses qui se mesure généralement par le taux de faux-positifs et/ou faux-négatifs [1][6]. Certaines attaques détectées par les SIEM du marché peuvent générer jusqu'à 8 alertes sur 10, faux-positives, c'est-à-dire d'alertes non avérées. Cette situation étonnante ne met pas en cause les savoirs des experts mais la précision du modèle de connaissances transmis au SIEM pour l'analyse. Les produits industriels emploient des règles fondées sur les expressions régulières formellement définies par les experts portant sur des règles explicites. On constate de ce fait un décalage entre le mode d'acquisition des règles par les SIEM du marché et les outils de d'apprentissage proposés par les outils scientifiques.

2.2 Modèle de connaissances pour l'analyse

Les connaissances des analystes sont énormes, et, proviennent de sources et savoirs variés : des expériences et savoir-faire des experts de sécurité, réseau, applicatif, ou, encore des sources externes (C) d'ordre divers - juridique (durée de garde des journaux), normatif (recommandations, bonnes pratiques : contrôles ISO 27001[22]) ou même d'ordre statistique issu d'observatoires officiels comme les CERT/MITRE [7] diffusant à l'échelle mondiale des bulletins de vulnérabilités. Cette partie présente les connaissances en abordant successivement les flux, les règles et en décrivant les aspects particuliers d'une attaque complexe.

2.2.1 Analyse des flux d'évènements

L'analyste mène une analyse de sécurité appelant différentes capacités en vue de détecter puis d'avérer une attaque.

Tout d'abord, pour détecter, l'analyste collecte des indices, les identifie et les localise pour déceler au plus tôt un potentiel comportement attaquant. Pour y parvenir, il étudie les « logs » collectés au préalable. Pour comprendre leur sens, chaque message renferme des caractéristiques intéressantes comme des états systèmes qui établissent les circonstances, les adresses qui localisent les faits ou des activités inhabituelles qui suscitent son attention. Mais, l'étape de détection n'est pas triviale, par exemple, en vue de détecter s'il s'agit de l'incident « l'utilisateur n'a pas ouvert sa session ? » (**Fig.3**), l'analyste devra compléter avec sa propre connaissance : « Kerberos ne peut ouvrir une session sans le succès de la pré-authentification » ou d'autres « logs ».

Puis, pour avérer un incident, l'analyste devra souvent rechercher les liens entre différents indices. Il mène une enquête au travers des « logs » pour caractériser l'incident et relier les indices pour comprendre et reconstituer les comportements.

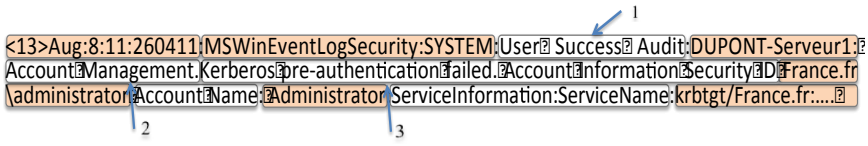


Fig. 3. *Message de type « login » extrait d'une sonde du système MSWindows*

Le « log » décrit deux actions : a) « Le système supervise les activités des utilisateurs» (1,2), b) « la pré-authentification Kerberos a échoué »(3).

Pour la suite, nous retenons qu'un mode opératoire est une suite d'actions reliées entre elles en vue de l'atteinte d'un même objectif [30][42]. Nous admettons également que certains « logs » résultent de l'observation d'une action (**Fig.2**) et d'autres, de plusieurs (**Fig.3**). En somme, nous émettons l'hypothèse que les « logs » peuvent être reliés par des règles similaires à celles qui des actions qui les ont générés.

2.2.2 Principe des règles du modèle de connaissance

Le niveau fortement structuré présente une syntaxe dont les règles de construction sont connues : une adresse « mail » - prenom.nom@domaine, les champs renferment suffisamment de sens pour prendre une décision comme le montre la règle :

REGLE 0 : « Si DomaineName <>HomeDomain , alors, « :Login denied:»

En revanche, le niveau faiblement structuré présente une description sous forme d'un texte en langage naturel ce qui pose le problème de la nature des outils utilisés pour en extraire le sens en vue d'automatiser cette décision.

Les règles se programment, au préalable, et sont élaborées par les experts humains afin de traduire leurs connaissances en un langage formel. Il s'agit pour les experts 2) d'identifier les variables et les règles associées, 3) de les codifier en vue de leur acquisition ou de leur apprentissage par les machines. [1][13][17] regroupent ces connaissances sous deux catégories principales (**Fig. 1**) :

- les connaissances apprises de type (A) : variables et règles sont extraites plus ou moins automatiquement par des outils de type statistique de l'IA : une heuristique anti-virus par exemple,
- les connaissances acquises de type (B) : variables et règles sont programmées par les experts lors de la conception ou de la configuration : signatures, règles de firewall,….

Dans les exemples (**Tab.1**), les règles 1,2,3 se suivent, 4 est indépendante. Les règles extraient d'une part des valeurs de variables, celles-ci peuvent être de nature spatiale (adresse IP, USERNAME) ou temporelle (FT) ou autre : ID, C,TYPE et extraire également des résultats : le nombre d'occurrences de valeur de champ (OccNb). Les règles 1,2,3 s'appuient sur une structure au format IDMEF qui exprime en plus de ces variables un champ particulier nommé « classification text » (CT) et responsable de la description de l'événement. Néanmoins, les règles d'agrégation et de corrélation appliquées au champ CT risquent de ne pas fonctionner pour certaines alertes car ces mécanismes comparent les chaines textuelles. Dans notre exemple, en supposant le CT de N : « login failed » et celui de N' : « login unsuccessful », on peut conclure que ces « logs » ne seront ni agrégés ni corrélés alors qu'ils représentent le concept d'« une tentative d'authentification non réussie ». Ce cas apparait souvent, par exemple, pour exprimer ce concept, on trouve plus de 200 expressions variant selon la marque et la nature des sondes de surveillance (CISCO, Microsoft, Linux,…).

REGLE 1 : «étant donné un événement unique ID : N :, lire les champs de description de l'action : CT : & le nom de l'utilisateur : USERNAME : pour une fenêtre temporelle FT : t : secondes, compter le nombre d'occurrences : OccNb pour CT : login failed : & USERNAME : X : »

REGLE 2 : « Si OccNb > 3, alors, générer une alerte TYPE : Suspicious Brute Force Attack Failed :»

REGLE 3 : «étant donné un évènement ID : N':, lire les champs description : CT : & nom de l'utilisateur : USERNAME :, pour une fenêtre temporelle FT : t : secondes, Si les valeurs CT sont : login success : & d'USERNAME : X : Si OccNb > 3 Alors générer alerte TYPE : Suspicious BruteForce Attack Success :
REGLE 4 : «lister toutes les sources adresses IP: 192.168.1.0:»

Table 1. Exemple de règles d'agrégation (1,4) et de corrélation (2,3)

2.2.3 Problème des attaques complexes

Lorsqu'il s'agit d'une attaque complexe, certaines règles ne sont pas évidentes. En effet, l'attaquant prend soin d'effacer ses traces et s'il ne le peut, de les disperser dans le temps et dans l'espace. Ainsi, pour détecter et avérer ce genre d'attaque, faut-il en tout premier lieu que les liens entre actions aient généré des « logs ». Il faut également que les connaissances de l'analyste soient suffisamment précises pour établir des corrélations entre les « logs ». Ainsi, un événement n'apporte-t-il à l'analyste qu'une infime partie d'un comportement ou d'une situation, d'autant qu'une attaque complexe pourra reposer sur plusieurs « logs » probablement conformes.

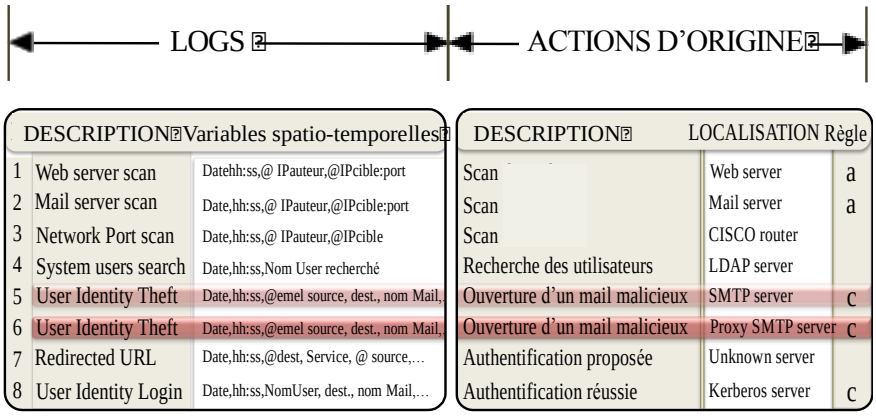


Fig. 4. *Processus de détection d'anomalie : action/« logs »*

A l'image du détective, l'analyste établit des faits puis applique des règles apprises dans le but de les relier et d'élever en continu sa connaissance. L'exemple (Fig.4) d'une attaque ciblée de type Cross Site Scripting représente 8 étapes pour exploiter une vulnérabilité et voler l'identité de l'utilisateur se connectant sur une page Web piratée au préalable. On suppose que chaque action a généré un « log ». A droite, les actions successives et à gauche les « logs » correspondant. Les « logs » sont structurés en vue d'exprimer un auteur, (1,2,5,6,8 pour un « attaquant », 7 pour un usager et 3 ,4 probablement non reliés et incertains. Pour relier les « logs », le rôle de l'analyste reste central mais la présence d'outils d'apprentissage seront des atouts. Mais, dans cette optique, les SIEM rencontrent des difficultés aux deux niveaux : la collecte et l'analyse.

Tout d'abord, la collecte ne peut assurer la complétude des « logs ». Ce processus est confronté aux limites techniques des machines. Pour qu'ils puissent être analysés, les « logs » doivent présenter une sous-séquence finie du flux. Celui-ci ne cesse de croitre, si bien qu'à l'intérieur de cette fenêtre, certains évènements peuvent manquer alors qu'ils sont essentiels pour donner du sens à d'autres. De même, certains évènements pourraient être masqués du fait de leurs cooccurrences massives.

En outre, l'analyse ne peut garantir la pertinence des « logs ». Certains « logs », seuls ou combinés entre eux, sont conçus pour alerter d'un dysfonctionnement mais ils ne le sont pas pour alerter de fautes de sécurité. De plus, un événement de sécurité peut être une feinte de l'attaquant. Il en est de même pour garantir la pertinence des règles de réduction et de détection décrites au préalable par les experts mais qui ne modélisent généralement pas des concepts mais des valeurs spécifiques au contexte. Ces règles, appliquées sur les « logs », sont contournables, peu généralisables.

2.3 Apprentissage des connaissances pour l'analyse

Face à ces difficultés, les outils les mieux placés pour assister l'analyste sont les outils capables d'apprendre des connaissances à haut niveau d'abstraction et en continu.

Traditionnellement, les techniques d'apprentissage des connaissances se distinguent selon le niveau d'intervention de l'expert dans la définition de la structure. On évoque le mode supervisé pour désigner les structures apprises sous le contrôle de l'expert assisté ou non d'un outil et le mode non supervisé pour désigner les outils capables de mener l'acquisition de connaissances de façon quasi-automatique. Dans cette partie, nous présentons tout d'abord la notion de structure des connaissances appliquées à la détection d'anomalies puis les méthodes d'enrichissement par apprentissage.

2.3.1 Structure d'apprentissage en contexte

La structure d'un message d'évènement (**Fig.1**) présente deux types de données : le type (A) fortement contextuel [1][40] en relation avec les données spatiotemporelles de l'action, on peut citer par exemple l'horodatage, la localisation par l'adresse IO, etc. puis le type (B), quasi indépendant de ces éléments du contexte décrivant la nature de l'action.

La plupart des outils de l'AA propose des structures fondées sur l'extraction de données contextuelles de type (A). Ces dernières années, de nombreuses publications ont été proposées sur le protocole NetFLow [20][41] qui regroupe des données de type (A). Dans ce cas, les sondes émettant des messages structurés de type Netflow, Les outils du « big-data analysis » (BDA) [1][40] reposent sur des outils statistiques dégageant les tendances et la structure sous-jacente d'une grande masse de données [29]. Dans cette même lignée, de nombreux travaux proposent de transmettre directement à l'analyste les résultats issus des sondes de surveillance. Par exemple, dans l'industrie, certaines solutions offrent une représentation graphique « radar ». Une autre catégorie détermine des critères spatio-temporels de type (A) : on trouve sur ce registre les travaux sur la détection d'anomalie comportementale en contexte [18]. La construction de profils « attaquants » représente un problème ardu souligné par de nombreux auteurs [17], bon nombre d'entre eux expliquent que la difficulté n'est pas les outils mais la faiblesse des modèles de connaissance insuffisamment expressifs pour décrire une attaque. Face à ces méthodes d'apprentissage statistiques, les attaquants peuvent

introduire des biais et réorienter les résultats vers des déductions erronées ou générer un événement rare pour absorber l'attention de l'analyste. L'apprentissage statistique [1] reste donc efficace sur de grands volumes de données, face aux dénis de service distribués mais ne résout pas la détection d'attaques ciblées.

Par ailleurs, comme le présente le schéma de la détection d'anomalie (**Fig.4**) l'expert de sécurité mène son analyse en contexte en extrayant les informations de type (B), il travaille ensuite à un haut niveau d'abstraction en reliant les alertes en fonction de ce qu'il connaît. Ce n'est qu'une fois ces liens mis en évidence qu'il en valide la faisabilité à l'aide de données de type (A). Par exemple, dans le contexte IDMEF, le champ « classification text » est une donnée de type (B) : l'expert caractérise le type de message « login failed » en l'associant à « login success ». Au moment de conclure, il valide les adresses des usagers.

Le travail sur le champ CT semble prometteur. A notre connaissance, il n'existe pas de modèle d'apprentissage reposant sur le champ « classification text », ces travaux consistent à traiter la sémantique de l'alerte en employant les outils pour élaborer une structure d'apprentissage.

2.3.2 Enrichissement par apprentissage

Nous distinguerons les modes d'enrichissement selon la façon dont la structure est apportée ou mise à jour :

1. « bottom-up » (B-u): la structure apportée à la donnée dépend de la propre structure des données en entrée,
2. « top-down » (T-d): la structure apportée à la donnée provient d'une structure externe, plus riche comme l'expert humain, la machine ou leur coopération.

2.3.2.1 Apprentissage « T-d » pour l'enrichissement des structures

Le mode supervisé permet de définir des structures d'apprentissage des connaissances à l'aide de règles définies par l'expert humain. Ces règles constituent aujourd'hui le modèle de connaissance le plus précis pour des problèmes bien définis. Lors de l'apprentissage, les experts humains créent deux jeux de données, le premier regroupant les données d'un comportement « normal », le second celles « d'attaque ». L'échantillonnage associe un label à chaque donnée. Ce mode d'apprentissage encore appelé « classification » permet de créer au préalable un « classifieur » qui sera ensuite destiné à marquer chaque nouvelle entrée. La plupart des travaux sur les méthodes supervisées traite de la précision aux frontières du normal et de l'anormal des jeux de données, sur les techniques pour créer les classifieurs. Les règles d'association, « rules-based » ou « à signature » [16][26]permet de déterminer des classifieurs statistiques et comportementaux où la structure est planifiée au préalable. Afin d'affiner la détection, [32] propose la notion de seuil par laquelle un événement est normal ou pas. Dans les données d'apprentissage anormales, les cas « d'attaque » sont rares en comparaison des cas normaux, l'identification de catégories précises et représentatives devient un problème difficile qui l'apparente à la construction de modèles prédictifs. Pour améliorer cette étape, certains auteurs [11] proposent d'élaborer un jeu de données artificiel (synthétique) « attaque » en injectant des attaques. [24] propose une autre technique intéressante fondée sur le parcours d'arbre de décision pour élaborer un modèle de connaissance afin qu'il devienne le plus généralisé possible tout en conservant ses caractéristiques spécifiques : de cette façon, par induction, [24] permet de généraliser les cas rares afin d'en améliorer la discrimination. D'autres

techniques plus connues traitent des problématiques de l'échantillonnage des données de l'apprentissage, la mise à jour post-apprentissage, la structure de la donnée en sortie trouvent traditionnellement une réponse par l'emploi des classifieurs. La littérature autour des classifieurs est vaste, on peut citer les travaux sur les « k-nearest » ou encore le « Support Vector Machine » (SVM) [16]. On peut également citer deux autres techniques comme les réseaux de neurones [26]. Ces travaux cherchent généralement à décider par régression de la règle d'appartenance d'une donnée vis-à-vis d'une classe.

En dehors des approches par règles, la recherche autour des méthodes supervisées s'est peu développée, bien que par le passé, elles aient été amplement déployées par le monde industriel. Ces techniques sont généralement les plus précises mais réputées couteuses en moyens et en temps et affaiblies face à de nouvelles attaques car aucune méthode d'échantillonnage ne peut garantir un jeu de donnée pur, exempt en totalité d'« attaques ». Ce dernier point ne semble pas concerner la forte variabilité des flux big-data. L'emploi de méthodes supervisées pour accomplir les étapes NAC semble adapté car le besoin de précision est essentiel dans la détection d'attaques complexes. La mise à jour des classifieur restera néanmoins un point important dans nos choix.

2.3.2.2 Apprentissage « B-u» pour la détermination des structures

Le mode non supervisé permet de travailler la masse de données en entrée et d'en extraire « en aveugle » des dimensions statistiquement élevées : la structure d'apprentissage se construit de façon implicite à partir de la donnée et est « bottom-up ». Le principe consiste à inférer un modèle de référence en cluster à partir d'éléments décelés dans les évènements [1][40]. [36] propose de constituer les clusters à partir d'une SVM en désignant un événement comme un point dont la densité constituera le critère pour décider qu'une zone dense est normale et éparse est anormale. [41] propose un mécanisme d'agrégation et de corrélation par le calcul de similarité entre plusieurs champs de la structure de la donnée, un champ de la donnée pouvant également devenir un cluster par apprentissage non supervisé. Ce mode est le plus intéressant dans le cas de nombreuses incertitudes puisqu'il peut déceler des structures sous-jacentes directement dans les données brutes. Ces techniques souffrent cependant de nombreuses imprécisions, leur adoption reste encore aujourd'hui modeste car ces outils conduisent à une énorme quantité de faux-positifs et faux-négatifs sans rendre compte de leur évolution en contexte, ni de l'alerte destinée à l'analyste. En outre, la phase complexe de normalisation reste une question (clustering) [25][33].

L'approche NAC (Normalisation-Agrégation-Corrélation) est devenue l'approche classique des SIEM pour favoriser l'apprentissage automatique de la connaissance. Parmi les travaux les plus connus, on peut regrouper ceux qui tentent de structurer les données en entrées (les journaux) [1][3]15][38] ou d'automatiser l'apprentissage des règles d'agrégation [15][41] ou de corrélation [17][18][25][31][33][36][38][39][40].

3 KILS, un modèle de connaissance à boucle

A l'heure actuelle, l'exploitation des SIEM reste la meilleure réponse pour aborder le processus de la détection d'intrusion. Néanmoins, nous avons relevé plusieurs points qui élèvent de façon inquiétante le taux de faux-positifs et de faux-négatifs et mettent en échec les différentes techniques NAC et BDA des SIEM [1][15][40] :

1. La difficulté d'incorporer de nouvelles variables nuit à la détection de nouvelles attaques. Avec les BDA, les variables sont extraites à l'aide d'outils statistiques à partir des « logs » mais ne prévoient pas d'interactions pour les valider avec l'analyste et son outil ; avec les NAC, chaque nouvelle connaissance est élaborée par les experts entrainant des temps de conception considérables.
2. L'abondance de flux « big-data » réduit la fenêtre de collecte et retire par ce biais du champ de l'analyse des « logs » pertinents. Face à cela, les règles de réduction des NAC sont les mieux placées mais limitées par leurs capacités de traitement générant de forts taux d'erreurs. Les BDA résistent mieux pour réduire des flux massifs mais souffrent d'un apprentissage « Bottum-up » de leurs règles.
3. La difficulté de modéliser des séquences d'attaques dispersées dans le temps et dans l'espace met en échec les règles de détection des modèles de connaissances des NAC. Ces règles, programmées par les experts de façon trop spécifique au contexte, se contournent par les attaquants à qui il suffit de modifier une valeur de variable pour ne pas être détectés (adresse IP, forme de mail,…).

Pourtant, les évènements sont les manifestations de l'environnement, leur collecte massive peut élever les connaissances des analystes pour améliorer les modes opératoires des attaquants. Selon la littérature [10], les enjeux du futur appellent des architectures de surveillance capables d'apporter une meilleure maîtrise des architectures de sécurité des Cloud et de celles incorporant des objets connectés, c'est pourquoi, face à cela, de nouveaux modèles de surveillance se devaient d'être proposés. Dans cette partie, nous présentons le modèle KILS (Knowledge and Information Logs-based System) pour:

- répondre à ces trois enjeux par une approche cognitive à multiples niveaux,
- apporter une architecture compatible en mesure de supporter cette approche.

3.1 Une approche cognitive à multiples niveaux

Pour répondre à ces défis, KILS apporte un modèle à multiples niveaux. KILS propose : 1) une structure cognitive, 2) un langage, 3) la combinaison de plusieurs algorithmes bio-inspirés.

3.1.1 Structure cognitive en boucle pour de nouvelles connaissances

L'objectif de la structure cognitive est d'enrichir les SIEM classiques des atouts des deux modèles B-u/T-d en utilisant les structures semblables à celles des humains.

Afin d'organiser leurs connaissances et de les appeler efficacement en situation d'analyse, les humains possèdent une structure en réseaux sémantiques, appelés ainsi parce qu'ils extraient le sens des faits qu'ils analysent et que leur structure repose sur des concepts - les nœuds du réseau - reliés par une ou plusieurs relations - les arcs. A l'aide d'une telle structure, les humains raisonnent face à la multitude de signaux variés de leur environnement. Ils adaptent également leurs raisonnements face à des signaux non connus. Selon [42], dans cette optique, ils classent leurs concepts et les ordonnent en fonction de relations choisies qu'ils évaluent en fonction de préférences.

Leur structure cognitive s'établit ainsi sur des ordres partiels en treillis, et lorsqu'un signal non connu apparait, ils explorent cette structure et comparent les concepts qu'ils ont appris. S'ils conviennent, ils décident ensuite d'une action dont ils évaluent

les effets au travers d'une boucle de contrôle dont le résultat en retour permet d'améliorer leurs connaissances. Dans notre contexte, les concepts pour détecter au plus tôt les signes de l'environnement, analyser une situation connue ou en imaginer de nouvelles, proviennent des différentes connaissances que nous avons décrites :

1) connaissances fortement contextuelles et apprises à partir des « logs » par un mécanisme « B-u » où l'événement fournit la structure et de nouvelles connaissances,

2) connaissances peu contextuelles et apprises à partir de règles expertes abstraites par un mécanisme «T-d » où la structure (le code) vient modéliser l'événement.

KILS (Fig.5) combine ces connaissances au travers des deux modèles.

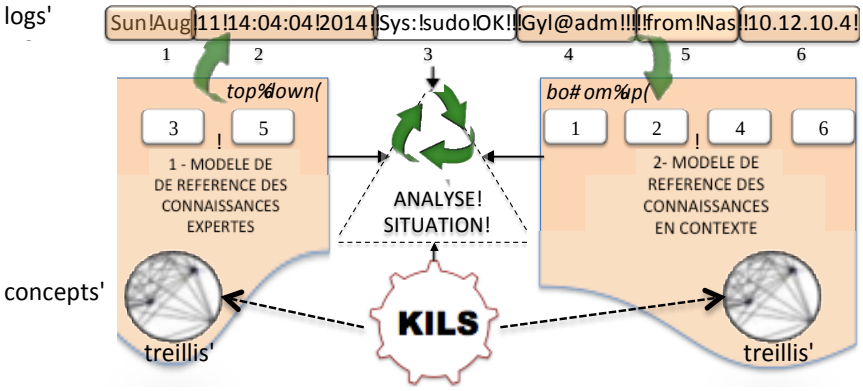


Fig. 5. *Modèle de connaissance KILS et boucle cognitive T-D et B-U*

Lors de l'apparition de signaux (logs), KILS les codifie en les comparant aux concepts qu'il connaît puis les classe pour ensuite les associer aux réseaux sémantiques de sa structure cognitive. Enfin, comme nous l'expliquons en § 3.1.3, s'il s'agit de nouveaux concepts, un mécanisme avancé de raisonnement intervient pour explorer ces réseaux sémantiques et insérer de nouvelles connaissances.

De cette façon, KILS favorise la coopération de ces deux modèles à partir de sa structure cognitive fondée sur des réseaux sémantiques organisés en concepts et relations.

3.1.2 Langage pour élever la connaissance de l'analyse

Le bruit lié aux flux massifs de « logs » nuit à l'application des règles et réduit la fenêtre de traitement des « logs ». Une approche intéressante consiste à codifier les « logs » par des concepts de telle sorte que l'on ne raisonne que sur ce sous-ensemble de concepts, réduit et dont la pertinence sera préservée.

Pour codifier le sens des « logs », l'expert utilise un langage. Pour couvrir tous les cas, exprimer le sens de chaque événement par des concepts appropriés, un langage doit être exhaustif ; dans notre cas, le domaine de la sécurité implique des concepts et propriétés de sécurité (DIC). Pour recueillir le consensus des experts, la définition des concepts doit être non ambiguë. Pour être exhaustive, la modélisation doit être généralisable et non ambiguë pour être spécifiable [28] [34].

Sous cet angle, l'idée est d'identifier un concept présent tant dans les « logs » mais également présent dans les connaissances des experts : au sens de Davidson et Such-

man [30][37], l'action est ce concept au cœur aussi bien de l'expression d'une politique de sécurité qui s'énonce par une action de configuration (« fermer un port ») de celle de l'alerte qui en dénonce la violation (« port ouvert »). Ainsi, de l'action à la réaction, tous les évènements pourront-ils se codifier par un même concept : l'action. Néanmoins, ce concept reste trop vague, il doit spécifier les circonstances de son déroulement. KILS spécifie le modèle de l'action à l'aide de cinq concepts de base : le mouvement, la cible, le résultat et l'auteur du mouvement, ce dernier étant mu par un objectif et un résultat [8].

Les ontologies [28] sont efficaces pour exprimer des classes de concepts et faire varier leur degré d'abstraction et de spécialisation. Le moteur d'analyse sémantique de KILS repose sur une ontologie centrale organisée en cinq concepts où chacun est spécifié à l'aide d'ontologies secondaires organisées en concepts spécifiques ou attributs ; ces attributs représentent les buts et finalités de l'action ainsi que des propriétés de sécurité : disponibilité, intégrité, confidentialité. Le rôle du moteur sémantique est d'extraire le sens des « logs » en identifiant les concepts qu'ils portent puis codifier les concepts de l'action. Pour mener ce processus d'extraction automatiquement, KILS emploie des algorithmes de « labellisation automatique du Machine Learning»[1][40].

3.1.3 Algorithmes bio-inspirés pour les corrélations

Un analyste opère de nombreuses corrélations en explorant au sein de ses connaissances la meilleure façon de rechercher des alertes manquées, de corréler des séquences de concepts, d'enrichir sa capacité de détection d'attaques nouvelles et complexes. KILS fait appel à plusieurs raisonnements bio-inspirés afin d'explorer sa structure cognitive dans les mêmes objectifs.

L'un des intérêts de l'exploration de réseaux sémantiques sera de résoudre le problème des règles d'expert inapplicables en présence de flux massifs. KILS fait appel aux deux modes de raisonnement B-u et T-d. Le premier mode de parcours est lié au fonctionnement intrinsèque de KILS qui raisonne en priorité sur les concepts. Dans cet exemple, la règle de type 2 (**Tab.**1) s'appliquerait chaque fois que le nombre de « logs » de type « login failed » dépasse le seuil de 3. Or, avec ce principe, KILS génère une alerte à chaque « login failed » : ce qui crée un faux-positif.

```
REGLE 2 : « Si OccNb > 3, alors, générer une alerte TYPE : Suspicious
Brute Force Attack Failed :»
```

Pour éviter ce biais, le second mode de parcours de KILS emploie un filtre d'extraction des éléments de contexte, dans notre exemple, il s'applique pour la variable : USERNAME ; un tel filtre permet d'avérer à partir d'expressions régulières les erreurs de login d'un même utilisateur. Ainsi, les expressions régulières, initialement utilisé par les NAC, inopérant sur des flux massifs reprend tout son sens appliqué à une classe restreinte de possibilités : dans notre cas, la classe « login failed ».

Un autre problème est d'avérer les alertes détectées, KILS utilise une classification des évènements qui traduits dans le langage KILS sont considérés comme rares et n'apparaissant qu'en certaines circonstances caractérisées au préalable. Par exemple, la classe « configuration détruite » n'appartient pas à une activité normale de l'usager mais à celle d'un administrateur. Une action d'administration dans les « logs » viendra avérer ce fait comme normal, sinon une attaque sera avérée. Ce genre de connaissances est décrit sur de nombreux supports. Les bulletins de sécurité du

CERT/MITRE [7] (détaillés en annexe) en sont un exemple avec plus de 64 000 vulnérabilités identifiées et avérées décrivant les modes opératoires des attaquants. Leur apprentissage reste néanmoins une difficulté [31]. Pour ce faire, le moteur d'analyse de KILS modélise et traduit ces scenarii en concepts à partir du langage KILS [1][3][15][17][18][25][31][33][36][38][39][40][41]. Le moteur d'analyse extrait la sémantique à partir d'une ontologie de l'action.

Pour appréhender la variété de relations causales, hiérarchiques, temporelles, le moteur d'analyse sémantique de KILS utilise des modèles variés qu'il met ensuite à jour. Il existe dans la littérature une grande variété d'algorithmes pour modéliser ces relations entre les actions et leurs concepts. KILS incorpore plusieurs algorithmes pour détecter des séquences d'attaques étant donné la valeur d'une ou plusieurs variables : markoviens [9][34] ou bayésiens [1]. Les markoviens présentent l'avantage de détecter d'éventuelles absences dans les séquences pour résoudre le problème des alertes manquées et consolider la fenêtre d'analyse de KILS.

Une fois ces nouvelles connaissances acquises par le biais de ces outils, KILS peut détecter de nouvelles anomalies en explorant sa structure cognitive à partir de données contextuelles et d'un algorithme d'optimisation par colonie de fourmis ayant été adapté à la détection d'anomalies par exploration de structures complexes[5].

3.2 Une architecture pour fédérer des connaissances H-M

Les technologies NAC et BDA des SIEM proposent une chaine de traitement linéaire de type B-U (**Fig.1**) : en entrée les « logs » et en sortie le tableau de bord. On constate que l'analyste se situe en dehors de la boucle de connaissances et n'est pas en mesure d'échanger avec son outil, d'évaluer et comprendre les flux qu'il analyse. Selon [28], les systèmes les mieux placés pour aborder la description de règles complexes reposent sur des systèmes à bases de connaissances [28] où celles-ci sont organisées selon leurs rôles dans le processus et se modélisent à l'aide d'une logique descriptive : par exemple, les ontologies. Selon [12], les systèmes de surveillance doivent incorporer l'humain dans la boucle de connaissance, [12] propose une architecture de surveillance en niveaux qui se distinguent selon leur rôle dans le processus et selon le niveau de leur automatisation. Pour [28][42], les humains utilisent leurs connaissances à partir d'une véritable architecture, face à des flux « big-data », ils organisent également leurs connaissances en plusieurs niveaux. Chaque niveau joue en rôle selon l'objectif souhaité : la vitesse et le volume de traitement, la précision. [35] propose montre également l'importance pour les humains de maintenir une boucle cognitive avec l'environnement. En nous inspirant des travaux de ces auteurs, KILS propose un système à base de connaissances en niveaux :

i) « monde réel » : les diverses activités informatiques ont lieu, les événements sont surveillés et des alertes sont générées par des capteurs,
ii) « stockage » : les alertes (logs) sont stockées et gérées par l'IT, il s'agit des entrepôts de données,
iii) « inférences » : une variété d'outils d'inférence permet de transformer les "logs" en informations pour établir une décision éclairée, on trouve les moteurs d'analyse de type NAC [1][38][40].
iv) « connaissances expertes » : les mécanismes de surveillance reposent sur plusieurs fonctions : détecter, mesurer, classer, localiser, analyser et choisir

les actions pour nettoyer, confiner, améliorer. Afin de guider les outils d'inférence, la connaissance experte représente les situations de surveillance.

Le niveau (iv) repose sur l'emploi d'ontologies métiers qui favorisent également les coopérations entre des centres de compétence ou de décision. Elles présentent l'intérêt de formaliser les connaissances en réseaux sémantiques, des connaissances H-M dans le sens où elles sont compréhensibles de machine à humain, humain à machine, machine à machine et humain à humain.

4 Conclusion

Les capacités d'analyse des SIEM sont les plus adaptées pour appréhender une réponse à l'échelle globale face aux attaques complexes. Nous avons vu que la situation actuelle des SIEM se heurte à de nombreuses erreurs dans la détection d'anomalie. Nous avons présenté les atouts des outils de l'IA. Du fait d'un contexte hostile et dynamique, ces outils doivent permettre la mise à jour de leur modèle de connaissance en fonction du contexte. Nous avons présenté l'importance de l'élaboration d'un modèle de connaissance cognitif, en contexte, multi-échelle et capable d'inférer des structures de connaissances complexes et multi-valuées. La modélisation de grandes quantités d'informations permet la classification des actions et des situations qui pourront être visualisées sur un écran de type radar à partir de l'extraction du sens des évènements. Ce travail ouvre la perspective d'un nouveau type de détection d'anomalies adapté à l'analyse globale des systèmes informatiques complexes.

Références

1. Chandola,V., Banerjee, A., Kumar, V. :Anomaly detection - A survey. ACM Computing Surveys, 41(3) : 15, 2009.
2. C.Llorens, L.Levier, D.Valois : Tableaux de bord de la sécurité réseau, 2ème édition. 3ème édition. Eyrolles, ISBN2-212-12821-5, 562 pages, août 2010.
3. Danyliw, R., Mejier, J.,Demchenko, Y. : The Incident Object Description Exchange Format', RFC 5070, (2007).
4. Debar, H., Curry, D., and Feinsten, B. : Intrusion Detection Message Exchange Format. RFC 4765,(2007).
5. G. Valigiani, Y. Jamont, R. Biojout, E. Lutton, P. Collet: Experimenting with a Real-Size Man-Hill to Optimize Pedagogical Paths. H. Haddad et al., Eds., Symposium on Applied Computing, ACM, Santa Fe, New Mexico, 2005
6. Divya, S.: A Survey on Various Security Threats and Classification of Malware Attacks, Vulnerabilities and Detection Techniques. International Journal of Computer Science & Applications (TIJCSA), 2(04), (2013).
7. Waltermire, D., Quinn, S., Scarfone, K., & Halbardier, A. : The technical specification for the Security Content Automation Protocol (SCAP): SCAP version 1.2. NIST Special Publication, 800, 126. (2011).
8. Legrand, V.: Confiance et risque pour engager un échange en milieu hostile. Thèse de doctorat, INSA-Lyon, soutenue le 19/6/2013, (2013).
9. Pietre-Cambacedes, M. Bouissou, "Modeling safety and security interdependencies with BDMP (Boolean logic Driven Markov Processes)," Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC 2010), Istanbul, Turkia, pp. 2852-2861, October 2010
10. Taleb, N.N.: The Black Swan:: The Impact of the Highly Improbable Fragility. Random House LLC, (2010).

11. Abe, N., Zadrozny, B., & Langford, J. : Outlier detection by active learning. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 504-509). ACM. (2006, August).
12. Basseville, M., Cordier, M-O, : Surveillance et diagnostic de systèmes dynamiques : approche complémentaire du traitement de signal et de l'intelligence artificielle. Rapport INRIA, N°2861,(1996).
13. Celeda, P., Novotny, J., Minarik,P., Rehak,M., Pechoucek,M. : Camnep : Agent-based network intrusion detection system (short paper). In AAMAS'08 (Autonomous Agents and MultiAgent Systems), (2008).
14. Cook, K., Grinstein, G., Whiting, M., Cooper, M., Havig, P., Liggett, K., ... & Paul, C. L. «:VAST Challenge 2012: Visual analytics for big data. In Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on (pp. 251-255). IEEE.
15. Dua, S., and Du,X. : Data mining and machine learning in cybersecurity. Taylor & Francis Group, (2011).
16. Duffield, N., Haffner, P., Krishnamurthy, B., Ringberg, H. : Rule-based anomaly detection on IP flows. In Proceedings of IEEE INFOCOM, Rio de Janeiro, Brazil, April 2009. (2009).
17. Estevez-Tapiador, J.M., Garcia-Teodoro, P. and Diaz-Verdejo, J.E. : Anomaly detection methods in wired networks: A survey and taxonomy. Computer Communications, v27, 1569-1584, (2004).
18. Forrest, S., D'haeseleer, D., Helman, P.: An immunological approach to change detection : Algorithms, analysis and implications. In Proceedings of the 1996 IEEE Symp. on Security and Privacy. IEEE Computer Society, 110.(1996).
19. Forrest, S.: Emergent computation: Self-organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks, proceedings of the ninth annual CNLS Conference", in Emergent computation, (pp.1-11), Cambridge, Ma, MIT Press. (1990).
20. Introduction to Cisco IOS : NetFlow - a technical overview. /http://www.cisco.com/en/US/prod/collateral/iosswrel/ ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.htmlS. Retrieved June 3, (2012).
21. ISO, "Common management Information Protocol specification(cmip). International Organization for Standardization, International Standard 9596-1, (1991).
22. ISO : Technologies de l'information - Techniques de sécurité - Systèmes de management de la sécurité de l'information – Exigences. ISO/IEC 27001 : 2005, Publiée, 90.92 (2008-10-15), TC/SC : JTC 1/SC 27. (2005).
23. Ji, Z., Dasgupta, D. : Real-valued negative selection algorithm with variable-sized detectors. In : Genetic and Evolutionary Computation GECCO-2004 Part I.Vol.3102 of LNCS. Seattle WA USA pringer-Verlag, p. 287-298, (2004).
24. Joshi, M. and Kumar, V. : CREDOS: Classification using Ripple Down Structure (A Case for Rare Classes). Proceedings of the 2004 SIAM International Conference on Data Mining. 321-332.(2004).
25. Julish, K. :Clustering Intrusion Detection Alarms to Support Root Cause Analysis. ACM Transactions on Information and System Security, 6(4) : 443-471, (2003).
26. Kruegel, C., Vigna, G.: Anomaly detection of web-based attacks. In Proceedings of the 10th ACM conference on Computer and communications security(CCS'03), pp. 251-261,(2003).
27. Labraty, J.F, Puidupin, A. : Information, cognition et décision : le cas du projet CORTIM. Chapitre de « Management, systèmes d'information et connaissances tacites » par Lesca, N., 283 p., - R : 004 LES - Hermès - 2007288 p., ISBN 978-2-7462-1487-3, (2007).
28. Le Ber, F., Lieber, J, Napoli, A. : Systèmes à base de connaissances. In : Encyclopédie de l'informatique et des systèmes d'information, éd. par Akoka, J.,et Comyn-Wattiau, I., pp. 1197-1208.,Vuibert, (2007).
29. Li, B., Springer, J., Bebis, G., & Hadi Gunes, M. : A survey of network flow applications. Journal of Network and Computer Applications, 36(2), 567-581.(2013).
30. Livet, P. : La notion d'événement chez Whitehead et Davidson. Noesi, N°13, (2008).
31. Ning, P.,Cui, Y.,Reeves,S.R.: Constructing attack scenarios through correlation of intrusion alerts.In ACM Conference on Computer and Communications Security, p.245-254, (2002).

32. Otey, M., Parthasarathy, S., Ghoting, A., Li, G., Narravula, S., & Panda, D. : Towards nic-based intrusion detection. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 723-728). ACM. (2003, August).
33. Qin, X., Lee, W.: Causal discovery-based alert correlation. 21th Annual Computer Security Applications Conference (ACSAC 2005), Tucson, December (2005).
34. Russell S., Norvig P.: Artificial intelligence : a modern approach. 2nd Edition. Prentice Hall, (2003).
35. Samsonovich, A.: Toward A Unified Catalog of Implemented Cognitive Architectures. In Proceedings of the 2010 conference on Biologically Inspired Cognitive Architectures 2010, pages 195-244, IOS Press. (2010).
36. Shon, T., Moon, K.: A hybrid machine learning approach to network anomaly detection. Inf. Sci., vol. 177, pp. 3799-3821, September 2007 (2007).
37. Suchman, L. : Plans and Situated Actions. Cambridge University Press. (1983).
38. Tan, P. N., & Steinbach, M. : Kumar. Introduction to Data Mining. (2006).
39. Tricaud, S., « Corrélation de journaux avec le framework Prelude », tutoJRES (2008).
40. UA, Sumeet et DU, Xian. : Data mining and machine learning in cybersecurity, CRC press,(2011).
41. Valdes, A., and K. Skinner, K. : Probabilistic alert correlation. In Proceedings of Recent Advances in Intrusion Detection (RAID), pages 54-68, (2001).
42. Weil-Barais, A. : L'homme cognitif. 1ère édition Quadrige, Paris : Presses Universitaires de France, (2005).

5 Annexe

Overview

(a)

The login method in LoginModule implementations in Apache Geronimo 2.0 does not throw FailedLoginException for failed logins, which allows remote attackers to bypass authentication requirements, deploy arbitrary modules, and gain administrative access by sending a blank username and password with the command line deployer in the deployment module.

(/a)

Impact

CVSS Severity (version 2.0):

CVSS v2 Base Score: 10.0 (HIGH) (AV:N/AC:L/Au:N/C:C/I:C/A:C) (legend)

Impact Subscore: 10.0

Exploitability Subscore: 10.0

CVSS Version 2 Metrics:

Access Vector: Network exploitable

Access Complexity: Low

Authentication: Not required to exploit

Impact Type: Allows unauthorized disclosure of information; Allows unauthorized modification; Allows disruption of service

Table 2. Extrait d'un bulletin du CERT décrivant (a)(/a) un mode opératoire

Automatiser la construction de règles de corrélation : prérequis et processus

E. Godefroy[1 2 3], E. Totel[3], M. Hurfin[2], F. Majorczyk[1], and A. Maaroufi[3]

[1] DGA-MI, Bruz, France frederic.majorczyk@dga.defense.gouv.fr
[2] Inria, Rennes, France michel.hurfin@inria.fr
[3] Supélec, Rennes, France erwan.godefroy@supelec.fr eric.totel@supelec.fr

Résumé Les systèmes d'entreprise sont aujourd'hui composés de plusieurs dizaines, centaines ou milliers d'entités communiquant potentiellement avec des machines externes inconnues. Dans ces systèmes de nombreux détecteurs, sondes et IDS sont déployés et inondent les systèmes de supervision de messages et d'alertes. La problématique d'un administrateur en charge de la supervision est alors de détecter des motifs d'attaques contre le système au sein de ce flot de notifications. Pour cela, il dispose d'outils de corrélation permettant d'identifier des scénarios complexes à partir de ces notifications de bas niveau. Cependant, la spécification de ces scénarios demande d'avoir au préalable construit les règles de corrélation adéquates. Ce papier se focalise sur une méthode de génération de règles de corrélation et des prérequis nécessaires à cette opération. Il évalue ensuite le travail requis pour obtenir de telles règles dans le cas d'un processus de génération automatisé.

Keywords: corrélation d'alertes explicite, scenario d'attaque, base de connaissances, taxonomie des attaques

1 Introduction

L'un des problèmes liés à la supervision des systèmes complexes réside dans le flot continu de notifications (messages, logs, alertes) générées par différents éléments de surveillance (sondes, IDS). Ces notifications sont généralement dans des formats variés et qui dépendent de la sonde (Syslog, CSV, IDMEF, formats spécifiques ...). Des systèmes de corrélation ont été conçus pour pouvoir traiter cette masse d'information [1]. Ces systèmes utilisent des relations connues entre des éléments apparaissant dans le flux d'information pour réduire le nombre d'alertes remontées et générer des méta-alertes résumant des informations sur des événements importants. On s'intéresse ici aux méta-alertes liées à la reconnaissance de la réalisation d'un scénario d'attaque spécifique au sein du système (reconnaissance de scénario explicite). Cependant, pour que cette détection puisse avoir lieu, il est nécessaire de réaliser plusieurs travaux préliminaires. Un expert doit tout d'abord spécifier les scénarios d'attaques redoutés pour le système surveillé. Cette spécification suppose une parfaite connaissance des enchainements potentiels des actions d'un attaquant. Ensuite, l'expert doit déterminer la manière

dont ces actions se manifesteront au sein du système cible. Il est donc nécessaire de maîtriser le système à protéger (et plus particulièrement les moyens de supervision mis en place). En effet, pour chaque sonde ou IDS, l'expert doit connaître i) les éléments surveillés, ii) les événements détectables, et iii) le format des messages ou alertes levées. En pratique, la connaissance de toutes ces caractéristiques demande beaucoup de temps et peu d'experts disposent de toutes les connaissances requises, ce qui rend difficile la construction de règles de corrélation complètes et correctes. En outre, même en supposant que des règles de corrélation soient correctes à un instant donné, à chaque évolution du système (ajout d'un nœud, nouveau plan d'adressage, modification de la supervision), les règles de corrélation deviennent potentiellement obsolètes (incomplètes, générant des faux positifs ...).

Nous présentons ici les éléments indispensables à la réalisation de telles règles de corrélation. Parmi ces éléments, on trouve tout d'abord le scénario d'attaque redouté. Ce scénario doit être spécifié de manière à inclure toutes les informations indispensables à la déduction de la manière dont chaque étape élémentaire peut être vue par différents éléments de détection. Il est également souhaitable de disposer d'une représentation la plus générique possible (qui ne dépende pas fortement des caractéristiques spécifiques d'un système). Ces contraintes sont prises en compte dans une structure appelée arbre d'actions obtenue à partir de l'enrichissement d'un arbre d'attaque spécifique à un scénario d'attaque donné. La qualité des règles de corrélation dépend non seulement de la qualité du scénario d'attaque décrit, mais également de la manière dont le système surveillé peut détecter la réalisation du scénario. Pour cela, il est nécessaire de prendre en compte les spécificités du système en instanciant le scénario sur le système cible afin d'identifier les acteurs concernés ainsi que les sondes potentiellement capables de détecter chaque action. Ce processus exige d'avoir la capacité de disposer des informations sur la topologie et la cartographie du système pour pouvoir déterminer les éléments internes potentiellement impliqués dans le scénario. Il est également indispensable de connaître les différentes sondes et IDS déployés au sein du système ainsi que leurs capacités de détection. En outre, il faut également maîtriser les formats et les contenus possibles des messages et des alertes générées par ces systèmes de surveillance.

Dans un second temps, nous évaluons à la fois le travail requis pour construire les règles de corrélation mais également la complexité de ces dernières. Cette évaluation est réalisée à partir d'un prototype automatisant un certain nombre d'étapes de création des règles de corrélation.

2 Procesus de conception des règles

Nous décrivons ici les prérequis nécessaires au processus de génération des règles de corrélation, puis les grandes étapes de ce processus. Étant donné un système à protéger et un scénario d'attaque, ce processus permet d'obtenir la règle de corrélation correspondant à l'instanciation du scénario d'attaque sur le système concerné. À partir de données d'entrées suffisamment précises, ce

processus peut être automatisé. Ces données initiales sont constituées par 1) une spécification du scénario d'attaque 2) une spécification du système.

2.1 Prérequis

Un arbre d'attaque constitue le point de départ de la description d'un scénario d'attaque. Cette structure a été retenue car c'est un outil largement utilisé en analyse de risques, plus lisible que les graphes d'attaques (généralement construits automatiquement par des outils qui produisent des structures difficilement lisibles [2], [3], [4]). Pour chaque sous objectif de l'arbre d'attaque, on précise les actions nécessaires à sa réalisation. Cependant, cette structure est trop informelle pour permettre de lever toutes les ambigüités d'interprétation. On souhaite également disposer d'un scénario relativement indépendant du système. Il est donc nécessaire de définir une représentation générique d'un fait observable. Cette représentation consiste à associer des paramètres correspondant à tous les attributs potentiellement observables par tous les types de sondes (réseau, système, applicatives). En fonction de la supervision mise en place, une partie seulement de ces paramètres sera réellement contenue dans les messages des sondes. Plus de précisions sont données dans [5]. En plus des attributs observables, il est nécessaire de préciser la sémantique de chaque action. Cela consiste à attribuer un nom (issu d'une taxonomie) qualifiant le type d'action. Ce nom permet ensuite de discriminer les sondes ou IDS capables de détecter l'action et également de sélectionner les messages potentiellement générés. Les taxonomies CEE et CAPEC semblent les plus adaptées pour nommer respectivement des actions standards (exécution, écriture, connexion) et malveillantes (injection, DOS, buffer overflow). Dans le cas d'un IDS à signature comme SNORT, les alertes contiennent un champ optionnel indiquant le type d'attaque (misc, web-attack, attempted-admin). Ces catégories sont à la fois ambigües et non normalisées. De plus, chaque IDS utilise sa propre taxonomie pour caractériser les types d'attaques ou d'événements reconnus. L'utilisation d'une taxonomie commune permet de réaliser des équivalences entre ces différents vocabulaires. Cette étape manuelle est identifiée par le chiffre *1* sur la figure 1.

Ensuite, le système cible doit être modélisé et comprendre tous les éléments nécessaires à la construction des règles de corrélation. Les éléments nécessaires sont décrits dans la section 3.

2.2 Étapes automatiques

Les étapes 2 à 4 de la figure 1 ont pour caractéristique commune de modifier l'arbre qu'ils reçoivent comme donnée en entrée. L'étape 2 de la figure 1 consiste à instancier le scénario d'attaque générique sur le système spécifié dans la base de connaissances. À la fin de cette étape, toutes les machines participant au scénario d'attaque sont identifiées. L'étape 3 détermine les observateurs (sondes, IDS) capables de détecter chacune des actions. L'étape 4 dérive les différents messages (logs, alertes) qui seront levées par les observateurs sélectionnés au moment de la détection. L'étape finale (5) construit les règles de corrélation

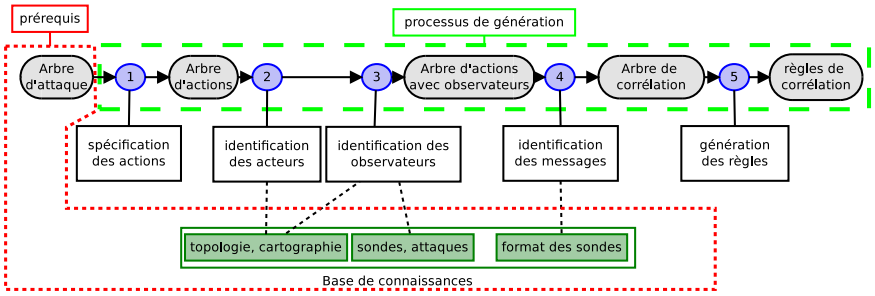


Figure 1. Transformation d'un arbre d'attaque en arbre de corrélation : étapes et structures

lisibles par un corrélateur donné. Elle consiste en une traduction de la structure d'arbre de corrélation générique vers une syntaxe spécifique.

3 La base de connaissances

Un modèle contenant les différentes informations du système cible est incontournable pour obtenir des règles de corrélation précises. Différents travaux ont été proposés pour modéliser des systèmes (totalement ou en partie). Les modèles tels que [6] ou M4D4 [7] apportent une partie des éléments nécessaires. Il existe également des outils de gestion et d'inventaire de parc informatique (OCS inventory, GLPI, OpenNMS) qui permettent de renseigner des informations sur les machines présentes, mais leurs capacités de modélisation sont généralement limitées à un inventaire partiel (pas de prise en compte des processus, services réseaux, utilisateurs, pas de modélisation de la supervision ...).

Dans notre approche, nous partons de la base M4D4 que nous étendons pour ajouter des éléments de modélisations nécessaires au processus. On se propose de décrire les différentes parties de cette base de connaissances et les éléments qui y sont modélisés. Tous les éléments présents dans la base ont pour but de servir de support à la réalisation d'un des trois objectifs suivants : 1) déterminer les éléments du système jouant un rôle dans le scénario d'attaque (machines directement ou indirectement impliquées) 2) déterminer l'observabilité de différentes actions concernant une ou plusieurs machines 3) déduire le contenu des messages qui seront générés par les sondes au moment de la détection.

3.1 Topologie

Les éléments centraux de la base de connaissances sont les différents nœuds du système. Ces derniers peuvent représenter tout types de machines (serveurs, routeurs, clients). Chaque nœud dispose d'une ou plusieurs adresses IP associées

à un sous-réseau particulier. Les interactions entre sous-réseaux sont modélisées par des routeurs (qui sont des nœuds particuliers) disposant de tables de routages. Ces routeurs peuvent également jouer le rôle de pare-feu et ainsi limiter les flux entre différents sous-réseaux. Ces informations topologiques sont nécessaires pour déterminer les connectivités entre les différents nœuds ainsi que les adresses IP des machines concernées par un scénario.

3.2 Cartographie

Cette partie apporte les informations sur les configurations logicielles de chaque machine (OS, logiciels et processus exécutés, informations sur les services réseaux fonctionnels). Ces informations jouent un rôle pour la sélection des machines dans le cadre d'une action de l'attaquant. Une action peut en effet viser un logiciel ou un type de service spécifique et présent uniquement sur une partie des machines du réseau. Cela permet également de connaître les numéros de ports ainsi que les noms des utilisateurs et des processus concernés par une action.

3.3 Sondes

Les sondes sont définies par 1) leurs visibilités 2) le contenu de leurs messages. Les notions de visibilité étendent celles introduites dans [7]. La visibilité topologique correspond aux capacités de détection d'une sonde en fonction de sa position dans le système.

La visibilité opérationnelle caractérise le type d'action qu'un observateur est capable de détecter pour une configuration donnée. Dans notre approche, cette visibilité est liée directement au nom de l'action spécifiée. Dans le cas d'un IDS utilisant des signatures, chaque signature est associée à une classe d'attaque spécifique. Dans le cas d'un IDS ou d'une sonde pouvant détecter directement certaines actions (appels systèmes, envoi de paquets réseau), on lie la sonde directement aux noms d'actions détectables. Une sonde est ainsi liée à une configuration de détection qui lui permet de détecter une ou plusieurs actions (actions standards ou attaques).

La visibilité des sondes n'est pas suffisante pour construire une règle de corrélation. Il est nécessaire de disposer des informations caractéristiques du message généré par la sonde lors de la détection. Ces caractéristiques incluent le nom du format du message (utilisé pour décoder le message), une indication sur le type de message (identifiant de signature dans le cas d'alertes déclenchées par la reconnaissance d'une signature) et tous les champs présents dans les messages (IP, port, user, ...). Ces champs correspondent à un sous-ensemble des champs décrits dans l'action.

3.4 Attaques

La base de connaissances contient l'ensemble des actions malveillantes (Attaques) possibles sur le système. Ces actions sont issues de la taxonomie CAPEC

et sont organisées sous forme hiérarchique (certaines attaques sont des cas particulier d'attaques plus génériques). Cette structure est importante pour permettre de lier des types d'attaques génériques à des attaques spécifiques. Lorsqu'un IDS utilise une signature (ou un modèle comportemental) capable de détecter un type d'attaque identifié, on associe cette signature à cette attaque dans la base de biens. L'intérêt est de pouvoir par la suite sélectionner l'ensemble des sondes capables de détecter une attaque spécialisant l'attaque générique utilisée dans la spécification d'une action. De plus, CAPEC est simplement utilisé comme base de données initiale d'attaques et il est possible d'ajouter des attaques spécifiques au système à protéger.

3.5 Classes d'équivalences

Dans le cas où un sous-réseau contient un ensemble de machines qui partagent les mêmes configurations (même systèmes d'exploitation, même configuration logicielle, système de supervision commun ...) et qui peuvent ainsi toutes participer de manière interchangeable à une étape d'un scénario d'attaque, il est possible de définir une classe d'équivalence. Cette classe d'équivalence est caractérisée par la configuration commune des nœuds ainsi qu'un ensemble d'adresses IP incluant toutes les machines qui partagent cette configuration. L'intérêt est qu'au moment de la génération des règles, une seule instance prenant en compte toutes les machines d'une même classe d'équivalence sera générée, réduisant significativement la taille de la règle de corrélation.

4 Évaluation

Un prototype automatisant les étapes 2 à 5 de la figure 1 a été réalisé. Notre objectif est de montrer que la méthode est applicable à des systèmes réels et que l'utilisation du processus de génération décrit dans cet article apporte un gain en termes de simplicité dans le cas d'évolutions du système et de complexité d'écriture des règles de corrélation.

Le système utilisé pour cette évaluation est représenté à la figure 2. Cette configuration est identifiée par *Ref*. Pour évaluer le coût d'une modification de la configuration, nous définissons la configuration *Mod* correspondant au système initial auquel on ajoute une sonde Snort dans le sous-réseau *dmz* et un serveur dans cette même zone. Dans chaque sous réseau, tous les serveurs ou clients ont une configuration logicielle identique. Ainsi, il est possible d'ignorer cette caractéristique ou d'utiliser des classes d'équivalences. Le but de cette évaluation est donc également de mesurer l'intérêt d'utiliser ces classes d'équivalence. À partir des deux systèmes *Ref* et *Mod*, on identifie quatre cas différents (Ref-PH, Ref-CE, Mod-PH, Mod-CE) avec PH pour "Pas d'Hypothèse" et CE pour "avec des Classes d'Équivalence" selon la modélisation choisie pour représenter un ensemble de machines équivalentes.

L'estimation de la quantité de travail requise par un expert pour construire la base de connaissances est calculée à partir du nombre de faits à renseigner dans

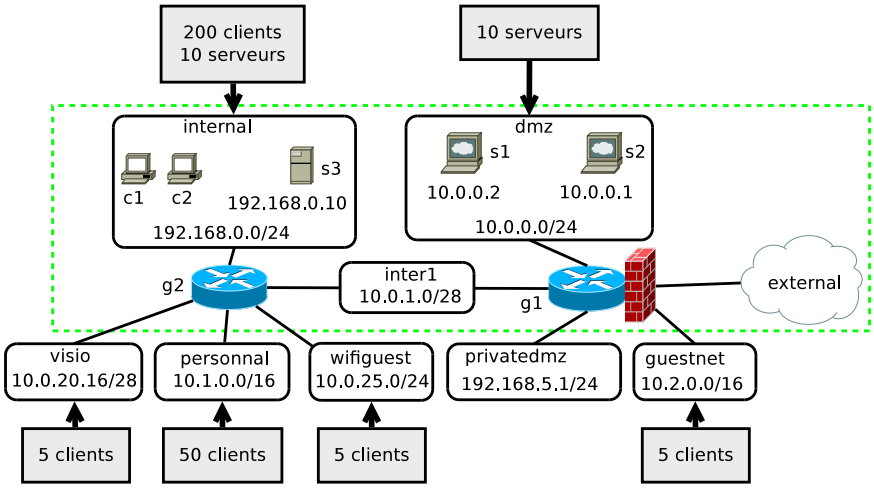


Figure 2. Système de référence pour l'évaluation

Configuration	Topologie	Cartographie	Sondes	Attaques
Ref-PH / Ref-CE	900/68	81/6	68/63	114/114
Mod-PH / Mod-CE	906/68	83/6	76/71	114/114

Table 1. Nombre de faits dans la base de connaissances

la base de connaissances. Le tableau 1 exprime le nombre de faits pour chacun des quatre cas d'étude. Les faits sont classifiés dans quatre catégories selon la partie du système décrite : la topologie, la cartographie, les informations sur les sondes et les attaques. L'utilisation de classes d'équivalence réduit de manière significative la taille de la base. L'ajout d'un nouveau serveur et d'une sonde demande trois nouveaux faits dans la base de connaissances (la référence du nœud, son adresse et son sous-réseau). Dans le cas de la modélisation utilisant des classes d'équivalence, le nombre de faits topologiques reste identique (on modifie simplement la plage d'adresse IP pour qu'elle inclue le nouveau serveur).

Le tableau 2 donne les tailles de l'arbre de corrélation pour différentes configurations du système étant donné des scénarios d'attaque de taille croissante. La dernière ligne du tableau présente les résultats pour un scénario de huit actions faisant intervenir les clients du réseau interne de la figure 2, ce qui explique la taille importante de l'arbre de corrélation lorque les classes d'équivalence ne sont pas utilisées. Ces résultats montrent d'une part que l'approche fonctionne dans le cas d'un système réel et d'autre part qu'une petite modification du système est plus simple à prendre en compte en mettant à jour la base de connaissances et en regénérant les règles qu'en modifiant manuellement les règles. En effet,

Nombre d'actions (feuilles)	Configuration	Messages	Operateurs
2 actions	Ref-PH / Ref-CE	30/3	21/2
	Mod-PH / Mod-CE	66/6	56/5
5 actions	Ref-PH / Ref-CE	44/6	51/5
	Mod-PH / Mod-CE	90/10	100/9
8 actions	Ref-PH / Ref-CE	11816/20	10551/13
	Mod-PH / Mod-CE	22472/30	21201/21

Table 2. Complexité de la règle de corrélation (nombre de messages et d'opérateurs)

dans le cas de notre modification, la taille de la règle de corrélation augmente d'un facteur deux.

5 Travaux connexes

En grande partie, les travaux qui portent sur la corrélation explicite se concentrent surtout sur l'efficacité des algorithmes en termes de faux positifs et de faux négatifs et s'attardent peu sur les problématiques de création de règles de corrélations cohérentes et complètes. Cependant, on peut comparer certaines techniques mises en œuvre dans notre approche à celles utilisées dans d'autres travaux du domaine.

En premier lieu, la prise en compte d'une partie de l'environnement d'exécution est réalisée dans certains travaux ([8], [9]). Cet environnement inclut généralement les services actifs, les vulnérabilités connues, ainsi que la connectivité entre les machines du système. Ces éléments sont alors utilisés pour vérifier la pertinence des alertes levées lors de la détection.

Ensuite, différentes approches ont été proposées pour modéliser les alertes levées par les IDS. Dans [8], des types d'alertes sont définis mais ces types ne reposent pas sur l'utilisation d'une taxonomie existante. L'approche [9] propose une solution permettant d'associer les alertes générées par l'IDS Snort à des étapes d'un scénario d'attaque (modélisé par un graphe d'attaque). Ceci est rendu possible par la réalisation d'une association manuelle entre les signatures Snort et les identifiants de vulnérabilités Nessus.

Enfin, certaines approches ([2], [10]) reposent sur la représentation de scénarios d'attaques à partir de graphes d'attaques générés automatiquement. Cependant, ces graphes prennent en général en compte uniquement les chemins d'attaque qui exploitent des vulnérabilités connues et identifiées du système, alors que notre approche permet de s'abstraire de cette information. De notre point de vue, notre approche est complémentaire car l'arbre d'attaque initial peut être éventuellement obtenu à partir des informations extraites d'un graphe d'attaque.

6 Conclusion

Cet article décrit les prérequis nécessaires à la génération de règles de corrélation forrtement liées au système surveillé. L'approche s'appuie sur l'existence d'une base de connaissances regroupant les informations liées au système et sur un scénario d'attaque spécifié dans un langage d'actions. Une fois ces prérequis remplis, la suite du processus peut être automatisée. Pour prouver que cette approche est réalisable, nous avons créé un prototype permettant d'évaluer notre démarche. Dans cette évaluation, nous montrons qu'une petite modification dans le système est plus simple à prendre en compte dans la base de connaissances qu'une modification directe des règles de corrélation.

Références

1. Valeur, F. : Real-Time Intrusion Detection Alert Correlation. PhD thesis, University of California (2006)
2. Jajodia, S., Noel, S. : Topological vulnerability analysis : A powerful new approach for network attack prevention, detection, and response. Indian Statistical Institute Monograph Series (2007)
3. Ritchey, R.W., Ammann, P. : Using model checking to analyze network vulnerabilities. In : Proceedings of the 2000 IEEE Symposium on Security and Privacy. (May 2000)
4. Ou, X., Boyer, W.F., McQueen, M.A. : A scalable approach to attack graph generation. In : Proceedings of the 13th ACM conference on Computer and communications security, ACM (2006) 336–345
5. Godefroy, E., Totel, E., Hurfin, M., Majorczyk, F. : Génération automatique de règles de corrélation pour la détection d'attaques complexes. In : SARSSI. (2014)
6. Granadillo, G.G., Mustapha, Y.B., Hachem, N., Debar, H. : An ontology-based model for siem environments. In : ICGS3 '11 : 7th Int. Conf. in Global Security, Safety and Sustainability. Volume 99. (2012) 148–155
7. Morin, B., Mé, L., Debar, H., Duccassé, M. : M4d4 : a logical framework to support alert correlation in intrusion detection. Information Fusion **10**(4) (2009) 285–299
8. Xu, D., Ning, P. : Alert correlation through triggering events and common resources. In : ACSAC. (2004)
9. Noel, S., Robertson, E., Jajodia, S. : Correlating intrusion events and building attack scenarios through attack graph distances. In : ACSAC. (2004) 350–359
10. Roschke, S., Cheng, F., Meinel, C. : A new alert correlation algorithm based on attack graph. In : Computational Intelligence in Security for Information Systems. (2011)

Détecter et réagir face aux cyber-attaques – Retour d'expérience d'une expérimentation technico-opérationnelle multinationale

A l'occasion de l'exercice multinational « Combined Endeavor 2014 », réalisé sous la responsabilité de l'USEUCOM (commandement américain en Europe), les 40 nations impliquées ont en l'occasion de tester la résistance de leurs systèmes d'information face à des cyber-attaques grandeur nature.

« Combined Endeavor 2014 » a au lieu en Allemagne en aout et septembre. Chaque nation a mis en œuvre ses propres moyens techniques de détection et de protection face à des attaques qui n'étaient pas à l'avance connues des participants.

Le retour d'expérience issu de l'exercice est d'une grande richesse, tant par les enseignements apportés sur nos propres capacités de détection et de réaction, par les enseignements apportés par l'observation des procédures et solutions des autres nations, et également par la prise en compte de la dimension multinationale hétérogène de cet exercice.

Les chapitres qui suivent présentent les objectifs de l'édition 2014 de l'exercice Combined Endeavor, les attaques qui ont été mises en œuvre, et les principaux enseignements d'intérêt général qui en ont été retirés.

1 Présentation de l'exercice « Combined Endeavor »

1.1 Introduction

Combined Endeavor (CE) est un exercice technico-opérationnel organisé par le commandement des forces américaines en Europe (US EUCOM). Initialement mis en place pour acculturer les nations du PfP (Partnership for Peace) au sortir de la guerre froide, CE est devenu une institution dans le périmètre des exercices opérationnels internationaux et a pris de l'ampleur grâce au partage des activités avec les exercices OTAN CWIX et SteadFast Cobalt.

Ce sont près de 2500 personnels issus de 40 pays différents qui participent à cet exercice. Cette année, Combined Endeavor s'est déroulé sur la base de l'US Army à Grafenwoehr (Allemagne), entre le 25 août et le 12 septembre.

Le but de cet exercice qui a lieu chaque année depuis 1995 sous le patronage et la direction du quartier général des Forces armées américaines en Europe est de préparer les opérationnels de nations différentes à travailler en coalition pour, à partir d'éléments divers, créer une vision opérationnelle commune. Cela passe par la coordination entre eux les systèmes de communication et d'information des États qui participent aux opérations de maintien de la paix. Lors de cet exercice sont réalisés la

planification, le test et la documentation d'interopérabilité entre les participants dans le domaine des systèmes de communication et des systèmes d'information.

Les résultats obtenus doivent permettre, lors d'engagements pour le maintien de la paix et aussi dans le cadre de la «Sécurité par la coopération» de répondre rapidement et avec une bonne préparation aux exigences requises par une communication et des transmissions efficaces.

Cette année, la France a déployé une SIA BOX lors de cet exercice ; celle-ci embarquait :

- un socle de services d'interopérabilité et facilité: AD, DNS, NTP, Courriel, SSO
- un module de médiation MIG en version HuB NFFI
- les solutions de supervision GSYS (Système) et GSEC (Sécurité)
- un serveur SFTP

1.2 Scénarios CE14

Une fois les configurations effectuées, des scénarios sont déroulés afin de mettre en œuvre les systèmes de commandement.

Trois niveaux de scénarios ont été déroulés :

- Assistance humanitaire

- Operations de stabilisation

- Operations de combats

L’action et les scénarios qui en découlent, se passent dans la corne de l’Afrique :

Organisation du déploiement :

L’exercice est divisé en « Mission Network » interconnectés entre eux.

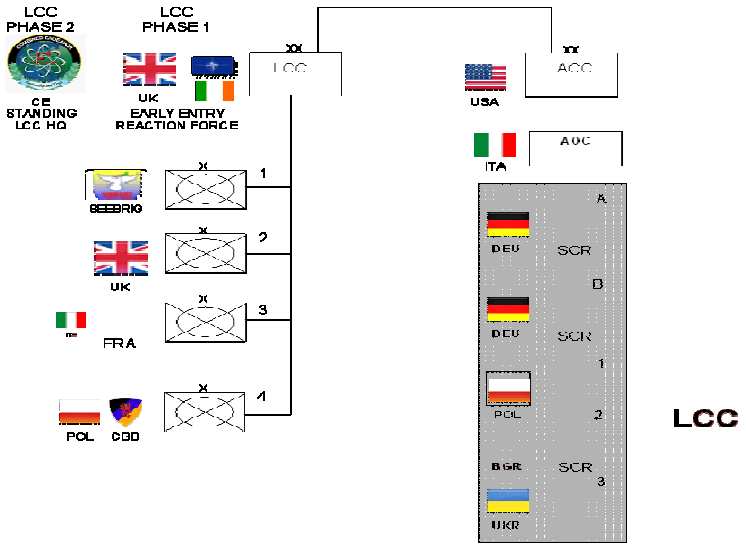
Au sein de chaque « Mission Network », les pays sont organisés sous la forme d’un ordre de bataille dans lequel une brigade a pour subordonnés plusieurs bataillons.

Au niveau supérieur, sont situés les 3 « Command Components » Air, Mer, Terre (ACC, MCC et LCC).

Cérémonie d’ouverture

Répartition des nations

Organisation



LCC

Figure ***Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.****-1 : Organisation Générale Logistics Coordination Centre*

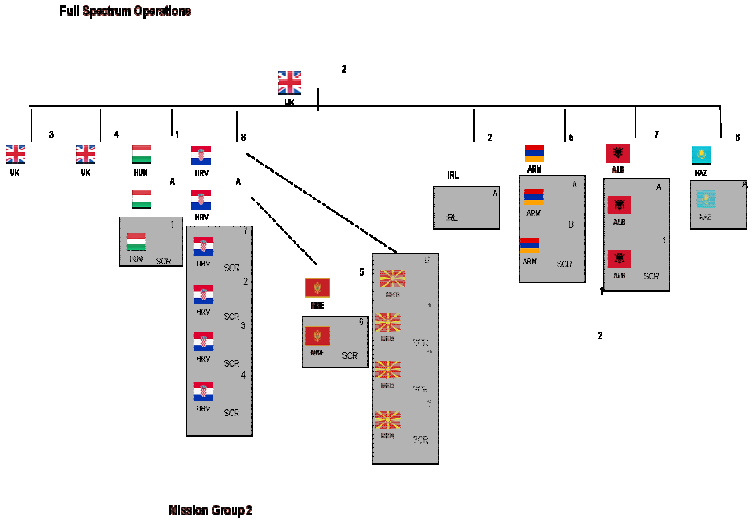


Figure -2 : Organisation Mission Group 2

Mission Network MSN3

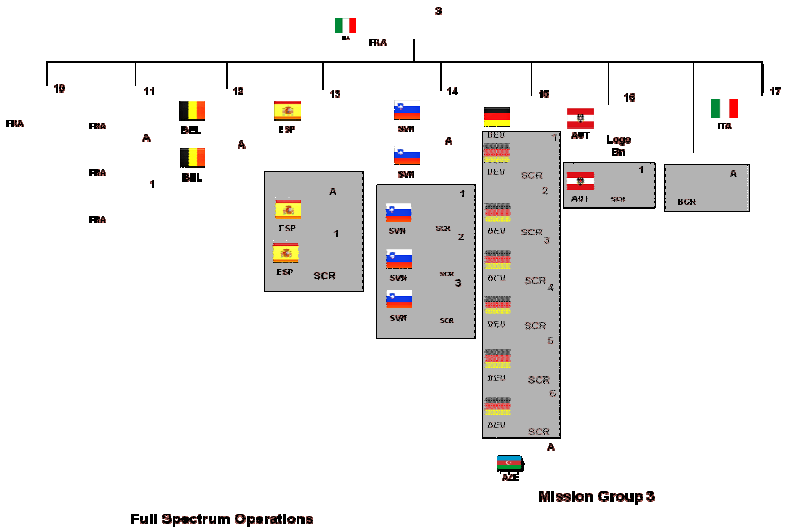


Figure -3 : Organisation Mission Group 3

Pour ce « MSN3», la France a tenu le rôle de brigade et donc aussi celui de « Lead Nation ». Dans ce cadre, outre l'interconnexion avec les autres MSN et le CJTF, elle doit offrir aux nations de MSN3 un ensemble de services (pont de visioconférence, DNS, NTP, HUB NFFI, PKI …).

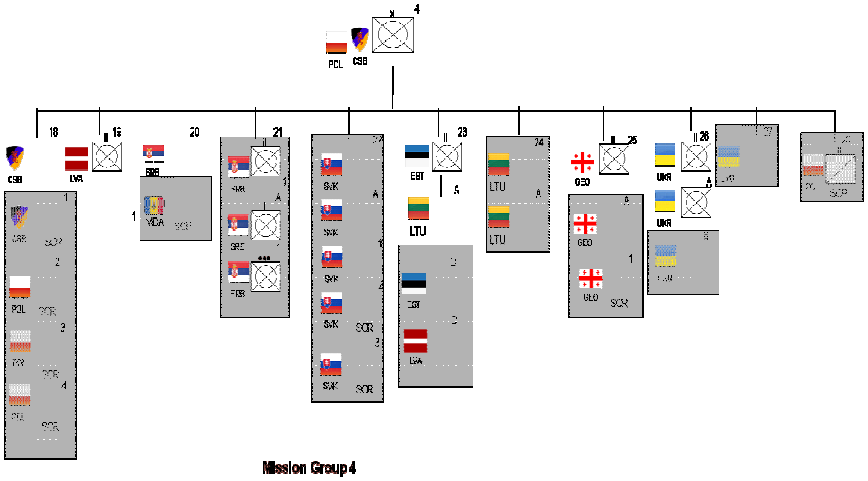


Figure -4 : Organisation Mission Group 4

2 Objectifs « CYBER » de l’exercice « Combined Endeavor »

Les objectifs de l’exercice en matière de cyber-attaques étaient les suivants :

- Tester la capacité des nations à détecter, réagir, reporter et remédier aux attaques
- Pousser les nations à collaborer et se coordonner lors de cyber-attaques de grande ampleur
- Permettre l’élaboration ainsi que la mise en œuvre d’une stratégie d'audit informatique fondée sur les risques en conformité en fonction des normes en vigueur ainsi que l'alignement avec les pratiques généralement acceptées
- Planifier des audits spécifiques afin de déterminer si les systèmes d'information sont protégés, contrôlés
- Signaler les conclusions de l'audit et faire des recommandations aux parties prenantes lors de la communication des résultats afin d’apporter les changements nécessaires
- Effectuer un suivi ou préparer des rapports d'état pour s'assurer que les mesures appropriées ont été prises en temps opportun
- Évaluer le plan de reprise après sinistre de l'organisation afin de déterminer si elle permet la récupération des capacités de traitement de l'information en cas de catastrophe
- Évaluer la conception, la mise en œuvre et le suivi du système et des contrôles de sécurité logiques pour vérifier la confidentialité, l'intégrité et la disponibilité des informations

3 Mise en œuvre des attaques

3.1 Les scénarios d’attaque

Une équipe d’attaque « Cyber Inject Team » a préparé des scénarios d’attaque dont les grandes lignes étaient connues et diffusées aux participants. Toutefois, les événements précis n’étaient pas connus des participants pour maintenir un bon niveau de réalisme.

Dans les grandes lignes, les scénarios d’attaque étaient les suivants :

- Networks Scanning
- Identity Spoofing (IP Address Spoofing)
- Denial of Service Attack
- Man in the Middle Attack
- Spear Phishing Emails
- Malicious Attachments
- Malicious websites
- Password Auditing
- Unauthorized Devices

Les attaques qui ont été employées sont représentatives des attaques classiquement observées par ailleurs et susceptibles d'être réellement rencontrées lors des opérations extérieures.

3.2 Les méthodes et outils utilisés par les attaquants

Les attaquants disposaient d'un ensemble d'outils, dont en particulier :

- McAfee Vulnerability Manager
- Nessus Vulnerability Scanner
- NMAP (Network Mapping)
- HP Webinspect (Web Servers)
- Hydra (Password Auditing)
- Metasploit (Password Auditing feature)

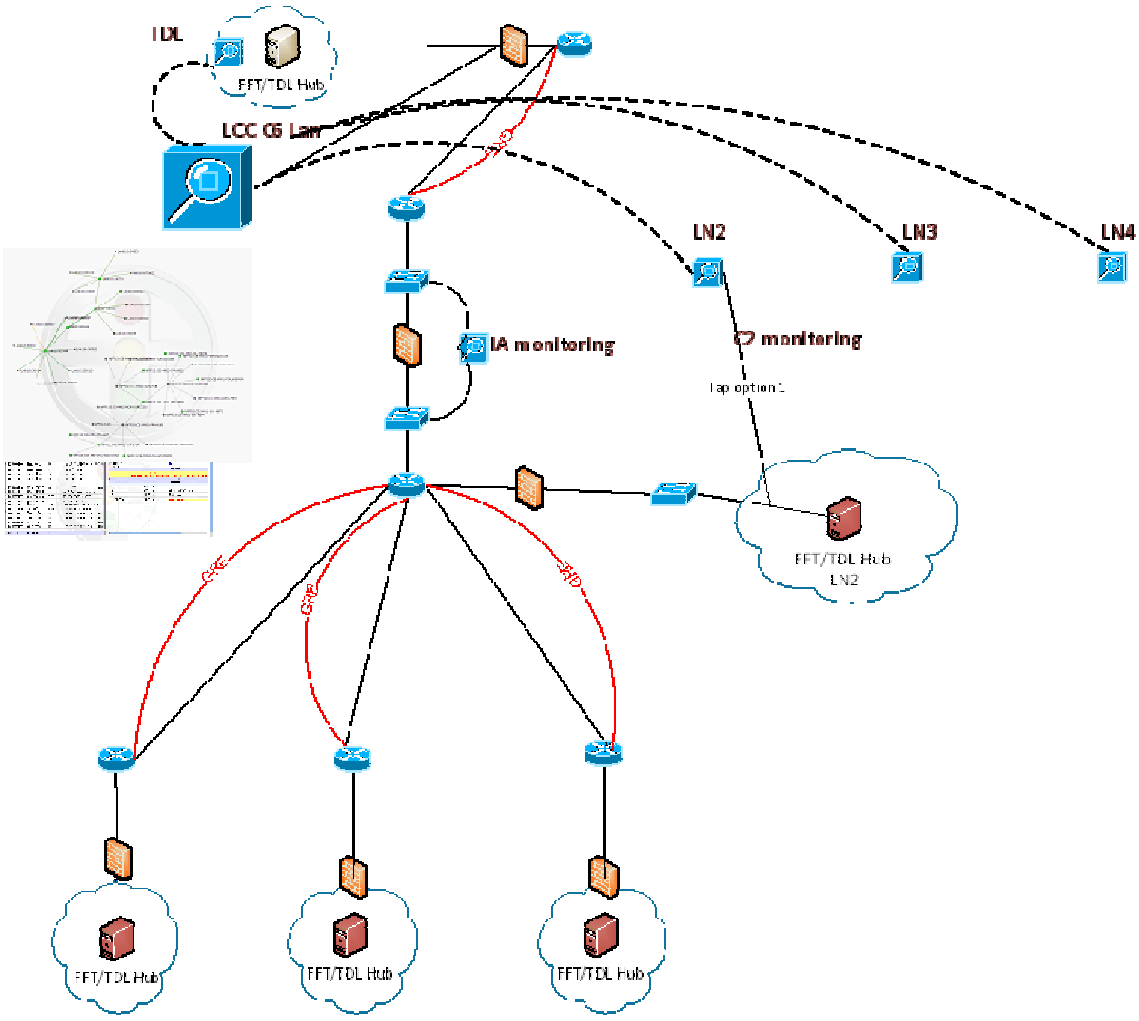
Pour renforcer la coopération interalliés en matière de cybersécurité multilatérale et afin d'accroître la capacité des pays à réagir conjointement et efficacement aux incidents cybernétiques, les étapes suivantes sont mise en œuvre :

- Étape 1 – Un atelier cyberdéfense fournissait sensibilisation et familiarisation aux personnels clés et au personnel de soutien dans les domaines de l'évaluation de la vulnérabilité, de gestion des incidents, demande de changement des processus, audit de sécurité et de gestion des correctifs
- Étape 2 – Des processus de validation réseau qui assurait la conformité avec les documents JMEIs (Joining Membership & Exit Instructions) de préconisations fournies. Les membres des équipes Cyber Inject, sur une base libre ont ainsi aidé les pays à répondre aux exigences d'accréditation et de validation
- Étape 3 – L'équipe Cyber a évalué l'efficacité des phases 1 et 2

3.3 Les moyens de protection contre les attaques

Chaque Nation doit mettre en place ses propres moyens de prévention, la France a mis en place plusieurs pare feu (dont deux dans la SIA BOX afin de protéger l'accès aux services fournis) mais également un SIEM (Prelude au travers de la solution GSEC) afin de garantir une visibilité sur le statut du réseau.

L'OTAN a déployé et utilisé également une surveillance du trafic C2 (Command and Control Services) pour permettre un suivi en temps réel de la disponibilité des services FFI / TDL entre les Mission Network selon le concept décrit dans la figure suivante.



L’outil Net Inspector est distribué et les Lead Nation, dont la France, ont fourni un serveur dédié pour la surveillance.

3.4 Éléments de retour d'expérience de l'exercice

L'exercice Cyber Combined Endeavor s'est ainsi déroulé suivant trois phases :

- **Phase I** - Workshop avec la mise en place des équipes, du 26 au29 aout 2014
- **Phase II** – Scanner de vulnérabilité et tests d'intrusion *(sur la base du volontariat), du* 1er au 06 septembre 2014 (Blue Team) : 15 nations ont participé
  - ***8 Networks Scanned*** En utilisant une adresse IP usurpée provenant d'une source chinoise connue
  - ***19 Networks Scanned*** En utilisant une adresse IP usurpée provenant d'une source iranienne connue
  - ***2 Phishing Campaigns*** Approximativement 500 emails envoyés
  - ***3 Brute Force Attacks***
- **Phase III** – Campagne Cyber, du 08 au 12 septembre 2014 (Red Team)
  - Networks Scanning
  - Vulnerability Scans
  - Spear Phishing Emails
  - Malicious Attachments
  - Malicious websites
  - Password Auditing
  - Attaques de type "Man in the middle"
  - Équipements non autorisés

A l'issue de l'exercice, des retours d'expérience d'intérêt ont pu être tirés dans différents domaines. Au-delà de l'état des lieux du niveau de vulnérabilité des systèmes opérationnels face à différentes cybermenaces actuelles, certaines considérations d'intérêt général peuvent être apportées.

Pour commencer, il faut noter que la partie « mise en œuvre » de l'exercice a été réalisée sur une période très courte, en à peine quelques jours. Le contexte est donc représentatif d'un déploiement en opération extérieure, et non d'un déploiement pérenne en métropole. La contrainte temporelle forte qui a pesé sur le déploiement a conduit à la présence de vulnérabilités que l'on ne rencontre plus sur des déploiements stabilisés :

- présence de mots de passe par défaut
- présence de plusieurs services non autorisés par les recommandations de l'exercice (TELNET)
- comptes par défaut et mots de passe à blanc sur plusieurs équipements (Tandbergs, téléphones IP)
- certains équipements permettaient un accès sans authentification à la configuration
- des versions de logiciels non mises à jour, présentant donc des failles de sécurité, ont été détectées sur des serveurs web (sujet à des attaques de type DDoS)

- de nombreuses vulnérabilités liées à des certificats avec des signatures de type digital « self signed » ont été détectées, laissant ainsi le système vulnérable face à des attaques de type « man in the middle »

Il est pourtant indispensable que la sécurité des SI déployés lors des opérations extérieures soit en place dès le début du déploiement. Des outils automatisant le déploiement de la sécurité et la vérification de la conformité doivent être employés, voire développés.

Sur le plan organisationnel, la collaboration entre les personnels du Security Operations Center et les personnels en charge du fonctionnement des réseaux doit être un point vigilance, en particulier lors de l'étape de conception des réseaux. L'optimisation de l'articulation des actions menées par ces personnels doit absolument être recherchée.

Les rapports de tests montrent que les capacités qu'ont les nations à résister aux attaques sont très variables, certaines nations ayant pu détecter ou bloquer des attaques, alors que d'autres n'ont pas su faire de même.

Dans un déploiement en coalition, les nations apportant leurs SIC sont susceptibles de subir les mêmes attaques. Par ailleurs, les différents SIC sont largement ouverts et interconnectés, ce qui augmente largement la surface d'attaque.

Le niveau de sécurité d'un déploiement étant celui de la nation la plus « faible », il est essentiel de permettre l'échange entre les nations des éléments permettant de constituer une Cyber Common Operational Picture (CCOP), qui a pour objectif de donner une représentation globale synthétique de la tenue de situation CYBER d'une opération militaire.

La défense collective de l'OTAN face aux cybermenaces implique donc la coopération dans le domaine CYBER par des échanges :

- sur les événements de sécurité rencontrés par chacune des nations
- sur les attaques détectées

De manière très globale, le niveau de sécurité a encore besoin de progresser :

- des backdoors installées sur les systèmes ont été identifiées
- il s'est avéré que certaines nations ont vu leurs serveurs tomber sous les attaques qui ont été menées

Pour terminer, l'exercice a également permis, comme pour les éditions précédentes de Combined Endeavor, de progresser sur les axes suivants :

- réflexions concernant les outils de gestion des risques à utiliser au niveau d'un poste de commandement
- réflexions concernant les outils de supervision de la sécurité
- des procédures manquantes ont pu être identifiées, par exemple dans le domaine de la maintenance des réseaux
- réflexions sur des outils de simulation permettant de tester la supervision de la sécurité

Posture de sécurité dynamique

David Bizeul

david.bizeul@cassidian.com

Abstract. Ce document propose une mise en relation des signaux externes à une organisation avec ses capacités opérationnelles internes. Cette démarche associe des approches souvent isolée permettant tout simplement de conditionner les moyens opérationnels de sécurité à une réalité de menace perceptible en externe.

Keywords: Posture de sécurité / sécurité adaptative / threat intelligence / indicateur de menace / réaction automatisée

1 Introduction

Le mode silo en sécurité s'effondre de plus en plus. Historiquement lié au développement de la sécurité, il a démontré ses limites sur le plan de la valeur ajoutée face aux attaques.

Traditionnellement la sécurité s'est développée en silo. Mise en place par des acteurs qui privilégiaient la pureté technique au détriment de l'intérêt de l'entreprise, cautionnée par des distributions de solutions clé en main fonctionnant en autonome plutôt que de s'interfacer à un écosystème global sécurité, le terrain était propice à ce développement en silo.

Mais les choses commencent à changer. De plus en plus, la fonction sécurité doit présenter un bilan de chaque sujet qu'elle mène et travailler un business plan pour chaque projet. C'est bien là un signe de maturité qui l'oblige à percevoir le bénéfice de chaque action de la sécurité dans l'organisation.

Les médias ont également mené à une prise de conscience à grande échelle ou tout du moins au niveau stratégique de l'entreprise. « A quoi bon allouer tous ces budgets sécurité si je ne suis pas à l'abri de toutes ces cyber attaques dont tout le monde parle ! »

Les fonctions même d'acteur de la sécurité se sont énormément professionnalisées. On trouve maintenant des métiers nouveaux directement importés des Etats-Unis tels que « security evangelist », « reverse engineer », « threat analyst ». Loin d'être inutiles, ces fonctions sont celles d'un modèle qui a besoin de convaincre, de connecter les menaces avec des cibles, de décortiquer la réalité des attaques. Bref une sécurité qui apporte une valeur ajoutée.

Si l'organisation de la sécurité évolue progressivement vers des modèles sains dans les organisations matures, c'est un peu moins le cas pour les solutions techniques de sécurité. Elles sont encore trop statiques et s'interfacent difficilement avec les composantes endogènes et exogènes du périmètre de l'entreprise.

Face à ce constat, nous présentons différentes pistes permettant de dynamiser une sécurité opérationnelle pour atteindre un but ultime : une posture de sécurité dynamique prenant en compte l'écosystème dans sa globalité.

2 Connaissance exogène

Rien n'est indissociable de son environnement. C'est vrai pour un individu, pour une entreprise et même pour un code malveillant. Face à cet état de fait, il apparaît nécessaire pour tout élément ayant atteint un certain degré d'interdépendance de comprendre son environnement. Cette évidence se traduit pourtant dans une prise de conscience assez récente pour la sécurité et qui se caractérise notamment par une dynamique, celle de la « Threat Intelligence »

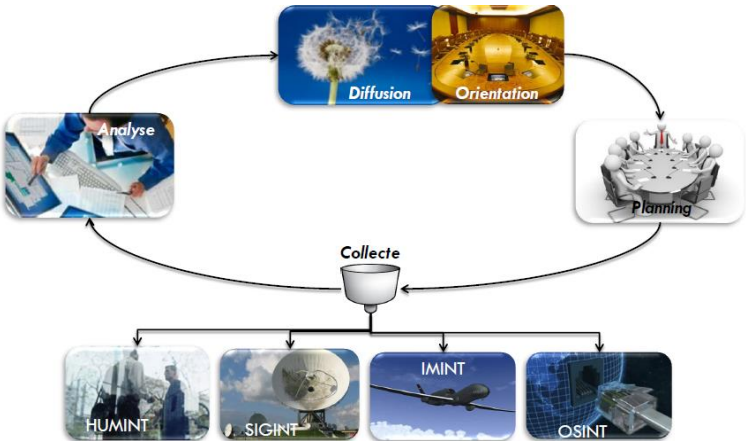
2.1 Threat Intelligence

De quoi parle-t-on ? La Threat Intelligence est l'activité consistant à collecter de la matière opérationnelle pouvant être directement actionnable. Cette matière peut provenir de différentes sources d'information :

- L'humain via des connexions directes de personne à personne
- La collecte d'informations directement accessibles sur Internet
- L'interception de signaux

Mais la collecte n'est pas une fin en soi. Cette démarche se traduit au sein d'un processus récursif passant par les phases suivantes :

- collecte
- analyse
- diffusion/stockage de l'information
- prise de décision



De manière très concrète dans les activités sécurité, cette démarche de Threat Intelligence va permettre de caractériser et de comprendre le fonctionnement des attaques telles qu'elles existent aujourd'hui.

2.2 Marqueurs opérationnels

La caractérisation des attaques peut se traduire sous différentes formes :

- des points de communications
- des caractéristiques de communications
- des modes opératoires spécifiques d'attaquants
- des traces laissées sur les systèmes

Chacune de ces caractéristiques peut être subdivisée en éléments distincts. Ainsi, pour les points de communication, on peut lister par exemple :

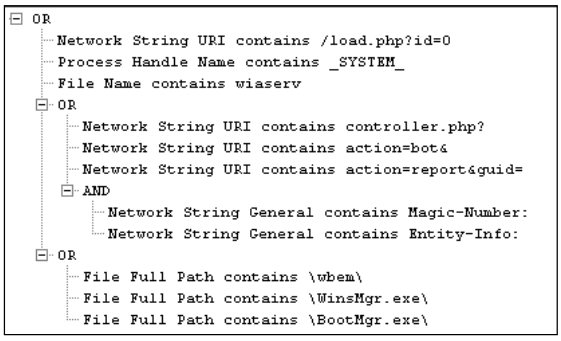
- les adresses IP
- les noms de domaine
- les enregistrements DNS
- les hébergeurs
- les AS (autonomous systems)
- les pays

Comment structurer toutes ces informations ? Des formats existent aujourd'hui adaptés à différents enjeux et plus ou moins compatibles entre eux. Le lecteur pourra se documenter sur plusieurs d'entre eux (SNORT [1], OpenIOC [2], YARA [3], IODEF [4], STIX [5]).

---

1 http://www.snort.org
2 http://www.openioc.org
3 http://plusvic.github.io/yara/
4 http://www.ietf.org/rfc/rfc5070.txt
5 https://stix.mitre.org/

A titre d'exemple, le format OpenIOC permet de décrire un comportement anormal sur un poste de travail par le biais de règles logiques :



2.3 Plateformes d'échange

Certains marqueurs sont récupérables directement sur Internet via une démarche de collecte d'information en source ouverte mais il existe également des plateformes d'échange ayant des objectifs collaboratifs ou lucratifs telles que :

- VERIS[6]
- Mandiant/FireEye[7]
- CIF[8]
- MISP[9]
- ThreatConnect[10]

Bien entendu, chaque plateforme favorise un format d'échange particulier ce qui rend les choses particulièrement complexes lorsqu'il s'agit de penser à la consolidation.

2.4 Indicateur de menace

Quel que soit le format de marqueur retenu et quelle que soit la plateforme d'échange utilisée, la question essentielle reste : « que faire de ces données ».

---

6 http://veriscommunity.net/
7 https://community.fireeye.com/welcome
8 https://code.google.com/p/collective-intelligence-framework/
9 https://github.com/MISP/MISP
10 http://www.threatconnect.com/

Un indicateur de menace a pour objectif de ressentir la proximité de la menace en répondant aux questions suivantes :

- cette menace est-elle généralisée ?
- cette menace cible-t-elle mon activité ?
- quel impact cette menace aurait-elle sur mon système d'information ?

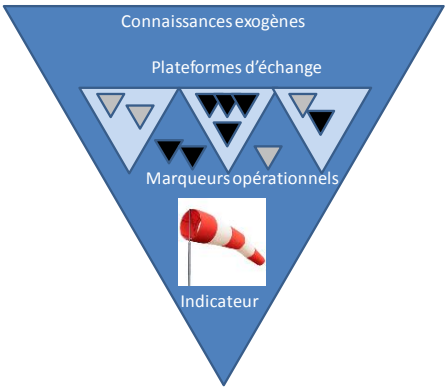
L'indicateur de menace doit révéler une information mesurée indiquant un danger. Cette définition implique différentes notions: quantité, qualité, fusion, évolution.

L'approche quantitative consiste à niveler la menace selon des niveaux. Un indicateur doit en effet posséder une certaine granularité pour permettre à l'analyste d'apporter des nuances à ces niveaux d'alerte : dans la pratique, la mise en place d'un indicateur de menace se base sur un score propre à la source étudiée (par exemple un nombre d'attaques observé sur une période donnée) qui est dans un second temps rapporté à un niveau de menace (1 à 4) calculé selon un barème préétabli.

L'approche qualitative permet de créer un contexte précis pour les indicateurs. Ainsi, il est fondamental de préciser ce contexte lors de la communication de la valeur de l'indicateur.

Lorsqu'un ensemble d'indicateurs a été identifié et/ou généré, il convient de fusionner ces informations en un indicateur unique pour permettre une représentation simplifiée et facilement exploitable. L'avantage d'avoir au préalable qualifié chaque indicateur résidera alors dans la possibilité de présenter un indicateur de menace sous différents angles (par exemple, par motivation d'attaque : cyber sabotage, cyber espionnage, cyber crime, hacktivisme).

Une fois cet ensemble d'indicateurs établi et leur fusion mise en place, il convient de suivre l'évolution de la menace dans le temps. Un paramètre essentiel à mettre en place et propre à chaque indicateur est le temps de latence de la menace. En effet, si pour un indicateur prédéfini la question ne se pose pas, pour un indicateur calculé, il convient de définir le temps pendant lequel la menace est effective, puis de décrémenter la valeur de l'impact dans le temps.



2.5 Micro-indicateurs et ontologie

L'Indicateur de Menace donne une indication sur l'évolution de la menace mise en œuvre par l'attaquant et pesant sur la cible : il convient donc de pouvoir décrire précisément cet écosystème. Pour cela, il est nécessaire d'élaborer une ontologie permettant d'obtenir un indicateur final conditionné selon les différents points de vue (qualité) pouvant être décrit par cette ontologie. On distingue plusieurs catégories principales :

- global (qualités liées à l'environnement global)
- attaque (qualités liées exclusivement à l'attaque)
- attaquant (qualités liées exclusivement à l'attaquant)
- cible (qualités liées exclusivement à la cible)

Pour chacune de ces catégories, il est donc nécessaire de mettre en place un ensemble de micro-indicateurs qui viendront alimenter un indicateur de menace final fiable et complet.

L'indicateur de menace global, défini selon un nombre de dimensions propre à son ontologie, sera techniquement exportable vers un système de supervision.

3 Connaissance endogène

Connaître parfaitement son système d'information relève de la gageure dès lors que la taille de l'organisation dépasse la TPME. De nombreuses zones d'ombre informatiques existent. Des projets naissent et meurent parfois de façon isolée. Les priorités d'entreprise oublient parfois d'associer la sécurité aux évolutions de l'IT.

3.1 Cartographie d'entreprise

Vouloir se connaître est un vœu que tout le monde peut exprimer. Pouvoir le faire est plus complexe. Cela requiert que l'organisation soit suffisamment claire pour savoir définir la raison d'être de chaque partie du SI associé. Cela impose une vraie connaissance historique de l'entreprise et une proximité avec les organes métier.

La capitalisation de cette cartographie peut se faire en de multiples schémas et documents. D'une manière plus structurée ITIL a apporté le concept de CMDB permettant de pouvoir faire vivre une cartographie d'entreprise.

3.2 Architecture

Tout comme un corps humain, chaque organe du Système d'Information a une raison d'être. Les règles d'architecture doivent être claires, simples et suivies dans le temps pour s'assurer que chaque projet de l'entreprise est pensé dans le cadre d'une architecture globale cohérente (ndlr, certains parleront ici d'urbanisation). Le principe essentiel derrière cette évidence est celui de la fédération des fonctionnalités auprès de composants dédiés aussi souvent que possible.

3.3 Tests d'intrusion

Chaque test d'intrusion effectué au sein d'une organisation contribue à mieux cerner les vulnérabilités et à maîtriser les risques. La réelle pertinence de la démarche dépendra toutefois de la latitude qui sera laissée à l'auditeur, l'objectif étant bien de savoir si tel scénario de risque peut se concrétiser dans l'entreprise et non de savoir si tel scénario de risque peut se concrétiser à travers la nouvelle porte d'entrée blindée nouvellement installée.

3.4 Scan de vulnérabilités

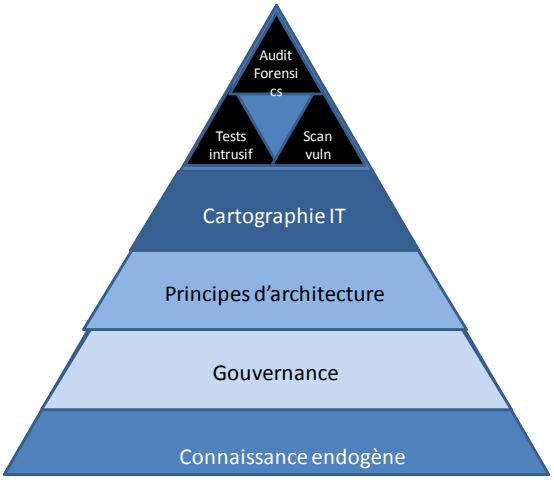
Les solutions pour détecter la présence de vulnérabilités permettent à la fois de rendre plus visible les investissements sur le patch management, d'améliorer la réactivité de ce même patch management en cas de menace avérée et surtout de prioriser les actions en cas d'incident concrétisé par une vulnérabilité déjà connue au préalable

3.5 Audit forensics d'entreprise

Le forensics a changé d'échelle. Il est maintenant fait à la taille du SI de l'entreprise et il est tout à fait envisageable de définir les comportements nominaux de l'entreprise à travers des solutions résidentes et qui surveillent et enregistrent les activités des machines en temps réel.

3.6 Principes et gouvernance

Mise à jour de cartographie, principe d'architecture, identification de problème… Tout ceci devient illusoire si une gouvernance forte n'est pas en place. Les règles doivent être édictées, transmises, expliquées et accompagnées pour être suivies d'effet. Tout ceci a bien évidemment un coût qui se traduira en accompagnement de la sécurité dans les projets.

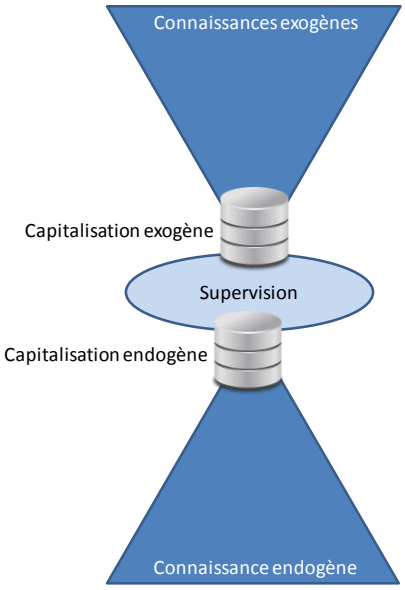


4 Détection

Idéalement, chaque organisation sera dotée d'une capacité de détection lui permettant de percevoir tout signal de risque de provenance interne comme externe.

4.1 Supervision

Avoir capitalisé la connaissance exogène et la connaissance endogène ne sert à rien si elle n'est pas mise à profit. Cela passe tout naturellement par un processus de supervision et une équipe dédiée qui pourra être en interface de ces deux mondes, le monde interne à l'entreprise permettant de la caractériser, et le monde externe permettant de connaître les dangers.



Dans cette organisation, la supervision est clairement le point d'articulation entre la connaissance de l'entreprise et la connaissance de l'écosystème

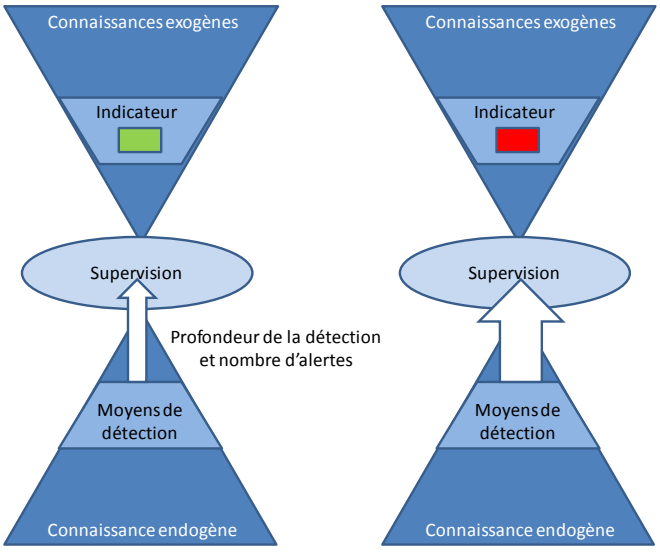
4.2 Moyens opérationnels de détection

Plusieurs composants d'infrastructure ou composants systèmes peuvent être particulièrement utiles pour apporter des éléments caractéristiques d'activités anormales :

- les équipements réseau capables de traiter les flux de communication
- les composants logiciels capables de suivre l'activité des systèmes de fichier

4.3 Posture de détection

La posture de détection est basée sur un postulat très simple : augmenter la vigilance lorsque le danger augmente. Dans notre approche, l'indicateur de menace exogène perçoit ce danger. En réaction à ces stimuli, les moyens de détection peuvent être affinés pour remonter plus d'information, la vigilance humaine et les processus du centre de supervision peuvent être calibrés. Dans ce contexte, tout est présent pour faire évoluer la posture de sécurité comme indiquée sur les schémas ci-dessous.



5 Réaction automatisée et semi-automatisée

La détection ne sert à rien si elle n'est pas suivie d'effet. Et toute cette réflexion est bien entendu conditionnée par un objectif de performance dans la réaction. Mais comment juger cette performance ?

5.1 Stratégie de réaction

Et si notre approche SSI classique n'était-elle pas déjà armée pour nous aider à répondre à ce challenge ? L'analyse de risque permet en effet déjà d'identifier les biens sensibles et de qualifier le risque associé à des scénarios concrets.

Il suffit d'appliquer ces démarches d'analyse de risque, non plus sous un angle de protection, mais en réfléchissant à l'impact de la réaction sur le fonctionnement opérationnel.

5.2 Principe de la réaction automatique

L'orchestration de la réponse à un incident repose sur des solutions qui se placent en complément des SIEM et autres instruments d'analyse de la sécurité des SI (ndlr : On parle parfois de cockpits de sécurité, d'hyperviseurs, voire d'orchestrateurs). Un cockpit de sécurité a pour vocation de donner un état des lieux synthétique de la situation de sécurité des SI et des réseaux. Pour ce faire, il rassemble l'ensemble des données collectées et analysées par les autres solutions de sécurité (SIEM, IDS, etc.) et

dresse des tableaux de bord récapitulatifs. Il propose un ensemble de vues dont les données affichées sont mises en relation avec toutes celles collectées. Ainsi, un cockpit de sécurité permet de faire de l'investigation poussée sur des types de données très différentes pourvu qu'elles aient un point commun (exemples : une adresse IP, un lieu, une heure). Il est capable d'évaluer le niveau de menace sur des entités de haut niveau comme les activités métiers (ex : facturation, contrôle de trafic ferroviaire) dans le secteur civil ou les missions dans le secteur militaire. On trouve ces solutions présentes dans les SOC et dans les centres de commandement car ils revêtent une importance stratégique du fait de leur capacité à présenter des vues cartographiques ou topologiques en relation avec des problèmes de sécurité.

Outre l'aspect présentation de l'état de sécurité et investigation, ces solutions embarquent également des capacités de recommandation de contre-mesures adaptées à la situation découverte, tenant compte d'un ensemble de paramètres contextuels (exemple : valeur des cibles visées par une attaque).

5.3 Principe de l'orchestration

L'orchestration de la réponse à un incident repose sur la mise en place d'une séquence d'actions répondant à plusieurs enjeux : analyser sa nature, contenir les effets du problème, rassembler des preuves, et appliquer une procédure d'éradication maîtrisée.

Un orchestrateur est donc un point central dans le dispositif de réponse. Afin de pouvoir mettre en place une approche d'orchestration, il faut pouvoir disposer à tout moment des données suivantes :

- L'inventaire des équipements du réseau
- Les services opérationnels avec leurs dépendances
- Les dispositifs de sécurité
- Les alarmes et incidents de sécurité
- L'état de sécurité en cours
- Les contextes d'attaques connus
- Les plans de réaction s'appliquant aux contextes d'attaques

La connaissance de la situation permet de localiser le problème et d'en mesurer l'impact, donc de déterminer l'urgence à traiter ce problème. L'orchestration doit être contextuelle à cette situation. Cela signifie d'une part que les contextes d'incidents doivent être définis, de même que la procédure en cas d'incidents inconnus. La prise en compte du contexte opérationnel est également un facteur à prendre en compte. Les actions quant à elles doivent être spécifiques aux éléments touchés et aux intervenants à impliquer. Il faut donc que l'orchestrateur soit en mesure de spécialiser les actions à entreprendre en fonction de la situation en cours. Cela suppose d'une part une définition d'actions génériques prenant en paramètres des informations fournies dans les alarmes ou présentes dans la base de connaissance de l'orchestrateur.

L'orchestration implique également des contraintes pour garantir l'application de la réponse en conformité avec les processus organisationnels. L'orchestrateur est ainsi un composant capable de gérer l'ordonnancement des actions de la réponse, notamment par la maîtrise des statuts des actions entreprises et la prise en compte des informations obtenues.

D'un point de vue humain, l'orchestrateur est un organe du centre de supervision. L'objectif est d'orienter la décision à prendre et les actions adaptées à la réponse. Certaines phases d'analyse, faisant partie de la réponse, amènent d'autres profils tels que des analystes ou des experts à intervenir. La composante de récupération et de diffusion d'information est donc essentielle, que ce soit des messages stipulant une demande d'aide ou des transferts de fichiers pour investigation.

5.4 Vers l'automatisation des réactions

Réfléchir à l'automatisation conduit nécessairement aux questions suivantes : comment s'assurer de la pertinence des actions entreprises ? N'y a-t-il pas de danger à laisser agir un système pouvant bloquer des services vitaux ? C'est pourquoi on peut écarter d'emblée une automatisation complète de la réponse. En fait, il faut plutôt l'envisager sur certaines phases. Les actions récurrentes et sans danger, comme l'affectation d'un opérateur au traitement du problème et la notification aux possibles intervenants sont de bons exemples. On peut aussi envisager la modification de la dangerosité perçue en fonction de certains critères vérifiés grâce aux données à disposition (cf. la liste plus haut) : absence de vulnérabilité exploitable à distance, faible sensibilité du bien touché. Plus dangereux car cela nécessite d'être attentif à ne pas laisser ce genre de règle persister : fermer automatiquement un incident lié à un faux-positif régulier (dû par exemple à un problème de configuration). Un aspect de la réponse intéressant à automatiser se rapporte à la partie recherche et collecte de preuves, sans oublier l'analyse de fichiers suspects si on dispose d'analyseurs dédiés.

Avec des outils intégrés, il est en effet possible de réduire les temps de réponse en automatisant les séquences ne nécessitant pas d'arbitrage humain.

Cela amène donc à la conclusion que l'application d'un plan de réaction doit pouvoir combiner des actions automatiques, semi-automatiques et sur demande. N'oublions pas non plus qu'une procédure textuelle est bien souvent présentée, pour laquelle on ne pourra rien automatiser.

5.5 Anticipation des effets d'une réaction

C'est un aspect peu abordé généralement au niveau de l'aide à la décision car les effets d'une réaction sont estimés en phase préparatoire. Néanmoins, les situations opérationnelles peuvent être plus variées qu'imaginées lors de ces phases amont. En

cas de situation particulière, il peut être bon de vérifier les impacts opérationnels d'une réaction. Prenons le cas d'une maintenance, du déploiement de correctifs par exemple, il convient de connaître les effets sur la QoS d'une telle mesure (quel est le nombre d'éléments à mettre à niveau, quels seront les éléments d'infrastructure traversés, quel est la taille du correctif, etc.). Ceci ne veut pas dire que les moyens seront à disposition de l'équipe sécurité mais qu'un processus existe entre les différents intervenants (sécurité et réseau). Il est bien sûr difficile d'automatiser ce type de réponse car il y a des actions inter-équipes, avec un temps d'évaluation assez long.

5.6 Adaptation de la posture de détection

Comme expliqué plus haut, la posture de détection doit s'adapter au contexte de menace (connaissance exogène) mais aussi au contexte opérationnel (connaissance endogène). En outre ce changement peut s'opérer comme une réaction à de nouvelles menaces ou à la mise à exécution de certaines menaces connues. Ces facteurs sont donc à prendre en compte par les moyens de réaction dès lors qu'ils ont pour vocation de modifier la stratégie de détection. Par exemple, il va s'agir de déployer des sondes IDS sur des secteurs non couverts ou plus simplement de les reconfigurer en ajoutant de nouvelles règles.

Prenons le cas d'une alerte indiquant qu'un groupe d'hacktivistes a l'intention de perpétrer des actions à l'encontre de grandes organisations, sans les nommer particulièrement. Les actions de renseignement (Threat Intelligence) peuvent conduire à identifier la charge utile déployable par ce groupe. La réaction à mener sera alors la mise à jour des NIDS pour tenir compte de cette information et alerter avec un niveau élevé tout paquet contenant cette information. Cette phase liée à l'adaptation de la détection peut être automatisée, dans la mesure où ce processus a été intégré en tant que plan de réaction prédéfini. Une seconde phase sera de vérifier que des plans de lutte existent si une attaque par ces hacktivistes ciblée sur l'organisation survient. Il s'agira alors de déterminer à quelle ontologie de menace on se réfère, quel est le mode opératoire présumé et comment l'organisation sera touchée (cibles techniques, effets opérationnels, business, réputation). Dans la majorité des cas, on se retrouvera dans une situation de réponse connue. Restera alors à définir les bons déclencheurs de la réponse. N'oublions pas l'aspect contextuel et opérationnel qui pourra faire varier cette réponse.

Enfin la modification de la posture de détection est parfois temporaire. Les outils doivent donc permettre l'application d'opérations planifiées pour répondre à des crises, gérant notamment le retour à la normale.

S’inspirer d’autres domaines pour améliorer sa performance de traitement des incidents

Nicolas Loriot- Airbus Defence & Space, Cybersecurity

nicolas.loriot@cassidian.com

Abstract. Adapter sa stratégie de détection à sa capacité réelle de traitement à la différence de rechercher la perfection technique qui est contre productive. S'inspirer de la sureté de fonctionnement et des personnes pour calibrer sa stratégie de réaction.

Keywords: SOC, outils, facteur humain, efficacité, stratégie de détection, stratégie de réaction, ergonomie, présentation des informations

1 Introduction

La performance d’un SOC [1]ne se limite pas à la performance des outils qui sont à sa disposition. L’efficacité d’un SOC se mesure sur les outils mais aussi sur sa capacité à interagir avec les autres services de votre société, et cela tout au long de la chaine : de la détection jusqu’à la résolution. Les personnes contribuent donc fortement à cette performance.

La performance de traitement implique donc autant la performance de détection que celle de réaction. Les outils modernes de détection peuvent générer beaucoup d’informations et donc noyer d’informations les opérateurs du SOC. Il convient donc d'aménager sa stratégie de détection afin qu’elle soit adaptée à l’homme derrière l'écran. Pour ce faire, nous allons aborder les deux thèmes suivants :

- Adapter sa stratégie de détection technique à la capacité réelle des équipes de supervision en place. Eviter le piège de la mise en œuvre d'une stratégie la plus parfaite techniquement
- Supporter la réaction en outillant stratégiquement cette dernière. Il s'agit ici du support pour réduire la charge de travail des opérateurs et ainsi pouvoir augmenter la pertinence et vitesse de réaction

Ce papier a vocation, au travers d’analogies sur d’autres secteurs d’activité et de retours d’expérience, de rappeler que le facteur humain est essentiel dans la capacité de détection. Ce facteur humain est souvent relégué au second plan face aux évaluations de performances des outils disponibles. Il est habituel de croiser des appels d’offres où les performances techniques des outils sont des critères prépondérants. L’exploitabilité de ces derniers est souvent relégué au second plan. Les sondes, corré-

[1] Security Operation Center : Centre Opérationnel de Sécurité

lateurs ou toutes autres technologies utilisées dans les SOC ne restent que des outils. Et comme leurs noms l'indiquent, ils ne sont là que pour outiller un travail afin de faciliter une mission. Il faut donc savoir adapter ces derniers à votre mission et ne pas tomber dans le piège exclusif d'adapter votre mission à vos outils. Remettons donc les outils au service de l'opérateur et pas l'inverse.

2 Adapter sa stratégie de détection à sa capacité réelle de traitement … et si le sport, une usine, le transport ou la médecine nous donnaient des pistes de réponses

2.1 Accepter l'imperfection et vivre (autrement) avec

L'objectif d'une stratégie de détection est souvent d'être la plus performante possible d'un point de vue technique. Bien que tout le monde accepte que la détection à 100% ne soit pas techniquement possible, une grande majorité cherche à s'approcher des 99%. Cette quête du Graal mène, par la même occasion, à un niveau de performance globale du SOC plus faible que si un objectif raisonnable avait été fixé.

Les phrases de type, « j'achète du faux positif pour traiter plus tard en forensic » le cas échéant ou « je veux tout voir » restent assez courantes. Malheureusement cette stratégie ne peut que rarement être adaptée aux moyens – humains, outils et financiers - en place. Imaginez un parc constitué de milliers de postes de travail sur lesquels la journalisation de toutes les actions aurait été activée sur les machines. Votre réseau sera chargé de messages de journalisation et vous devrez avoir une infrastructure de collecte importante (performance de messages acceptés par seconde et volume de stockage).

Et pourtant, afin d'assurer une organisation (outils, processus et personnes) qui fonctionne, il faut savoir rester raisonnable. Il faut donc accepter qu'une partie des attaques passe à travers les mailles du filet ou que certaines notifications ne se feront pas à la seconde.

2.2 Gérer son effort pour durer

Etre raisonnable à la manière d'un sportif qui gère sa course d'endurance. La gestion stratégique de la course est au moins aussi importante que la performance intrinsèque du sportif.

Dans certains milieux, il est d'usage de dire qu'un sportif atteint son apogée au début de la trentaine. Non pas qu'il soit le plus performant physiquement à 30 ans, il l'est effectivement plus à 20-25 ans, mais il connait mieux son corps et sait gérer une course grâce à son expérience pour tirer le meilleur parti de son physique en fonction de son déroulement. Un jeune, qui a plus de potentiel physique, aura souvent tendance à vouloir trop donner en début de course ou à suivre trop vite une accélération, il s'épuisera et se fera doubler dans la dernière montée. La gestion de l'effort est primordiale.

De la même manière, un pilote automobile adapte sa voiture à la piste sur laquelle il va rouler : rapport de boite, suspension, pneu … Il cherchera un compromis performance / facilité de conduite pour tenir sa course d'endurance et se garder du confort afin ne pas devoir toujours rouler à 100% lorsque la fatigue se rajoutera. Cette gestion lui évitera probablement la sortie de route fatale ou tout du moins contre performante. Pour gagner sa course, il doit avant tout la finir. Encore une fois, la gestion de l'effort est primordiale.

En fonction de l'objectif de la course : sprint de 15min, course de 2h, course de 24h, le pilote ajustera son véhicule. Il se permettra des réglages plus incisifs (plus fatiguant tant physiquement que psychologiquement) sur un sprint que sur une course de 24h où il devra ménager sa monture et son physique pour durer.

On peut, d'une certaine manière, rapprocher le réglage de sa voiture aux réglages des outils, et la capacité physique des pilotes à la capacité de l'équipe SOC. La stratégie de réglage, doit donc être adaptée pour tirer le meilleur en fonction de la durée (mode SOC en support à réponse à incident / course sprint et mode SOC nominal / course 24h).

2.3 S'adapter à l'équipe et non à l'individuel

Lors d'un sprint le pilote est seul. Il règle le véhicule à sa guise. Lors d'une course de 24h, il roule en équipe. La voiture doit donc être réglée pour que tous les pilotes soient à l'aise à son volant. De ce fait, un compromis devra être fait. Il est fort probable que certains pilotes auraient été plus rapides s'ils avaient pu régler le véhicule individuellement.

Pour avoir une bonne équipe, il faut la faire durer pour qu'elle apprenne son système et sache travailler ensemble. La faire fonctionner à 110%, c'est user les hommes comme on use les mécaniques et risquer la casse (démission, démotivation car l'objectif est inatteignable …) donc être au final moins performant.

Il faut également se rappeler que la performance d'une équipe sportive n'est pas liée uniquement à la performance des personnes ou du matériel dont elle dispose, mais surtout aux capacités de ses individus à travailler ensemble. L'objectif est d'atteindre un esprit de groupe, une osmose, un régime de croisière.

On peut voir l'organisation idéale d'une équipe SOC de la même manière. Il en ressort souvent que d'utiliser à 100% un maillon de la chaine est souvent contre-productif (à part en usage ponctuel - sprint).

De la même manière, sur une chaine d'assemblage d'une usine, une équipe qui produit moins vite ou plus vite que le régime de croisière ralentit la cadence de la chaine car il va générer des dysfonctionnements. L'optimum de la chaine n'est pas l'optimum de chaque maillon indépendant.

Il faut donc prendre du recul et adapter les outils de son SOC à ses équipiers.

2.4 Le chef d'équipe doit adapter le tempo

Le bon objectif du responsable de la stratégie de détection est de tirer le meilleur profit, non pas des outils, mais de l'organisation mise en place (outils, processus et

personnes). Des exemples dans le sport ou dans l'industrie sont courants pour nous montrer que la perfection unitaire est souvent l'ennemi du résultat global.

Acceptant ce constat, il parait raisonnable de dire qu'il ne faut pas chercher à l'atteindre (est-elle possible en détection des attaques ?) et qu'il faut ajuster cette dernière à la capacité de notre équipe.

Des attaques passeront et devront être traitées d'une autre manière (Réponse à Incident). Acceptons ce fait et définissons la stratégie de détection par rapport à nos moyens (outils, processus et personnes).

S'appuyant sur le principe des chaines d'assemblage d'usine, la bonne stratégie pour atteindre une performance globale revient souvent à adapter son système à la vitesse de croisière du maillon le plus lent. Ensuite, on essaiera d'optimiser le maillon le plus lent pour augmenter progressivement la cadence de la chaine. Une étape après l'autre. Le rodage est primordial avant la montée en cadence.

Il faut donc identifier la capacité de traitement de son équipe pour les incidents dit « courants » et ajuster initialement ses outils sur cette capacité. Les autres incidents seront traités dans un mode d'exception (support de niveau 3, réponse à incident).

2.5 Présenter de la bonne manière …

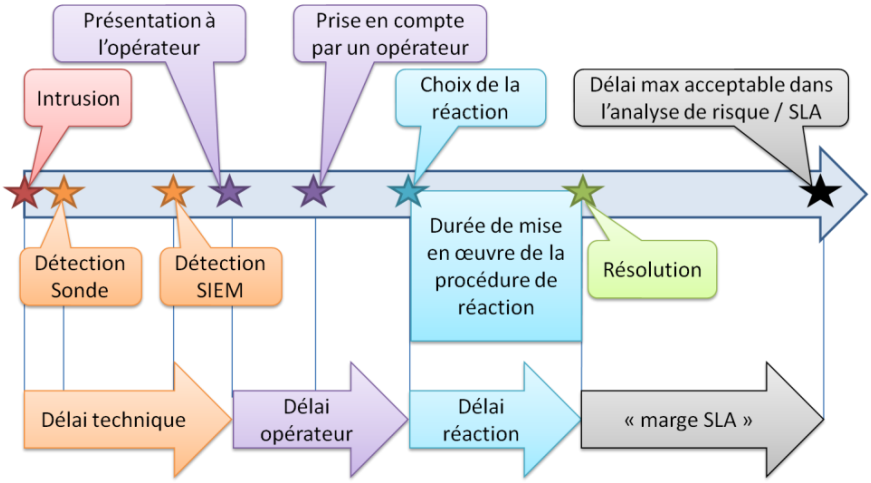
Sachant cela, l'ergonomie va prendre une place importante dans la stratégie de présentation des incidents. Par exemple, les premiers avions n'avaient que quelques voyants ou jauges pour informer le pilote de la situation. A cette époque le pilote devait connaitre par cœur ses derniers.

Avec l'avionique moderne, les cockpits des avions ont évolué et présentent un grand nombre d'informations disponibles. Il est humainement peu envisageable que le pilote retienne, sans exception, plusieurs centaines de messages. L'ergonomie des cockpits d'avion a donc pris une place importante afin de présenter les informations aux pilotes et copilotes.

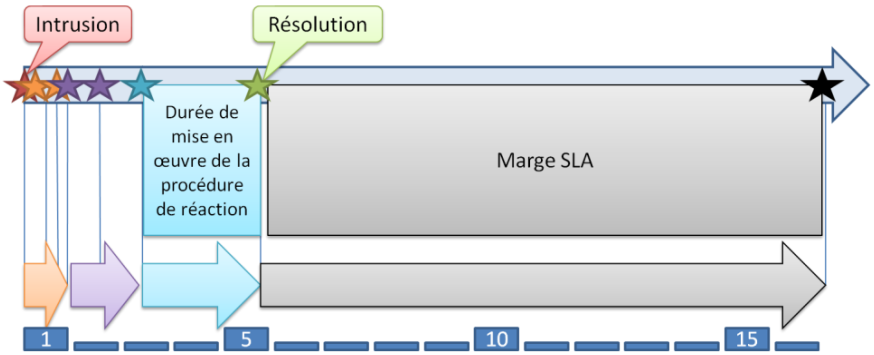
http://www.flickr.com/photos/83823904@N00/64156219/ (CC-BY-2.0)

Regroupement des messages par familles (regroupement de premier niveau) et priorité (incident commande de vol par rapport à une panne sur des équipements de confort). Les affichages s'adaptent alors aux informations à présenter.

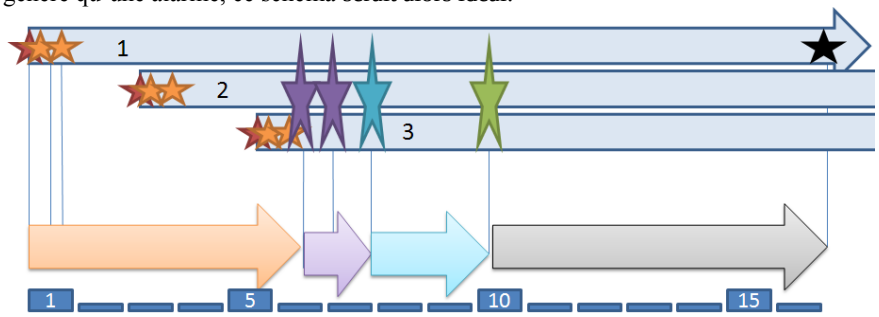
2.6 … au bon moment



La présentation à la « milliseconde » à la suite d'un incident n'a quasiment aucun intérêt si elle ne vous permet pas de bloquer l'attaque. L'intervention humaine va introduire plusieurs minutes dans la réaction. Posez-vous la question suivante : présenter cette information au bout de 5min, 1h ou même une demi-journée aurait-ce changé la donne pour autant ?



Dans la chaine de traitement suivante, la résolution a été réalisée en moins du tiers du temps qui a été accordé pour ce type d'incident. Si l'intrusion est unique et ne génère qu'une alarme, ce schéma serait alors idéal.



Cependant, si votre système détecte plusieurs intrusions du même type assez rapprochées et générant à chaque fois une nouvelle alarme (à cause de règles trop précises par exemple), vos équipes vont devoir regarder 3 alarmes. Si les alarmes 2 et 3 tombent sous la tutelle du même opérateur, le surcoût de traitement sera limité car il se rendra compte de la similitude. Si les alarmes 1, 2 et 3 tombent sur des opérateurs distincts, alors, chaque opérateur va dépenser un temps d'analyse et ne traitera pas d'autres incidents durant ce laps de temps.

L'exemple peut être appliqué à la surveillance des requêtes DNS en erreur qui permettent, parfois, d'identifier la présence d'un malware dans notre système d'information. Allez-vous analyser dès la première requête en erreur ? Ou bien toutes les heures, 6h, jours ou semaines ? L'analyse horaire sera plus coûteuse au final que d'analyser la liste complète de manière hebdomadaire. Posez-vous la question du délai raisonnable durant lequel vous acceptez d'être infecté sans être notifié. Après tout, certains d'entre nous acceptent d'attendre le soir ou le weekend pour aller voir son médecin quand ils ont certains symptômes. Dans d'autres cas, les symptômes sont beaucoup plus forts et vous poussent à aller aux urgences. Identifiez et distinguez les vraies urgences des symptômes qui peuvent attendre.

Optimiser, c'est accepter de perdre (un peu) à un endroit pour gagner beaucoup à un autre. Il vous appartient de régler ces temps de notification, en fonction de l'activité habituelle de votre système par rapport à la taille de votre équipe. Cependant, ne sous-estimez pas que votre système doit en garder un peu sous le pied pour les jours de crise.

Si votre équipe est capable de traiter une vingtaine d'incidents par jour, il faut alors calibrer votre système de détection pour qu'il reste proche de ce chiffre en incidents qui seront à traiter manuellement. Pour améliorer cette performance, la réaction automatique peut être envisagée. Sinon, il faudra demander du renfort pour digérer ce pic.

Il est possible de segmenter alors les incidents en plusieurs catégories afin d'adapter sa stratégie de notification. Il faut donc adapter ses notifications en relation avec la fréquence d'occurrence des phénomènes et le risque de laisser longtemps cet incident en cours sur votre système.

- Le temps réel strict: l'incident doit être contré avant qu'il se termine et donc à sa détection. C'est donc une mesure à appliquer au niveau des équipements (firewall/proxy qui bloque un flux, IPS[2] qui réinitialise une session …). L'intervention humaine n'est alors pas compatible avec la continuité du service. Elle a finalement peu de place dans la présentation à un opérateur pour action. - Vous noterez, dans cet exemple, que la réaction automatique est donc déjà dans votre système d'information. - C'est une notification pour information (la contre mesure étant déjà appliquée). Il faut donc se poser la question de l'utilité d'informer l'opérateur (par exemple : un virus courant détecté sur un poste seul connecté à internet et nettoyé par l'antivirus) sur un évènement unitaire de ce type. Cette même information croisée avec d'autres données ou une occurrence plus élevée de ce même évènement peut, elle, prendre du sens pour l'opérateur.
- Le temps réel mou à temps contraint (contraint par un PCA[3]/PRA[4] ou par SLA par ex.) : l'incident doit être présenté dans des délais permettant de mettre en œuvre le plan de PCA/PRA associé à l'impact métier identifié. Si le PCA/PRA vous accorde 1 journée pour effectuer une action que vous savez faire en 1h, alors la notification sous la demi-journée est suffisante. Ce temps de notification peut être mis à profit pour agréger plus d'informations. Bien sûr, si votre équipe est dans une phase de creux de charge, vous serez intéressé de lui notifier au plus tôt. Mais avant de définir une stratégie au millimètre, il faut viser le nominal [cf. paragraphe 2.6]. Rien ne vous empêche d'avoir un mécanisme pour forcer la notification plus tôt si la bande passante de votre équipe le permet.
- Temps différé : Le rapport devient votre allié. Cette technique permet d'accumuler les évènements qui sont répétitifs (paquets bloqués par un pare-feu, url en erreur, requêtes DNS en erreur ...) mais qui ne nécessitent pas un traitement urgent (ou qui ne sont pas humainement traitables en temps réel) et dont l'analyse est quasiment aussi coûteuse avec une dizaine d'entrées qu'avec plusieurs milliers. Vous pouvez également faire rentrer dans cette catégorie les outils d'analyse par représentation

[2] Intrusion Prevention System
[3] Plan de Continuité d'Activité
[4] Plan de Reprise d'Activité

graphique, les outils de type « big data analytics [5]» … toute la famille d'outils ayant comme objectif de rechercher les signaux faibles. Ces derniers sont taillés en effet pour travailler sur de longues durées afin de présenter des éléments pertinents. Les outils de recherche de signaux faibles nécessitent la présence d'analyste chevronnés pour analyser leurs contenus.

3 Réactions, s'inspirer des domaines de la sureté de fonctionnement et des personnes pour calibrer sa stratégie

3.1 Mesurer l'impact sur la continuité de service de l'attaque … et de sa parade

Ce chapitre considère que vos analyses seront faites sur la base des analyses faites avec les équipes métiers. Limiter ces analyses aux équipes IT ne permettra pas d'identifier des plans de réactions automatiques qui n'impacteront pas les métiers. En effet, une réaction qui semble valable d'un point de vue infrastructure peut avoir un impact négatif sur un service métier.

Les réactions automatiques, entre enjeu opérationnel (pour digérer la charge de travail) et craintes de mise en œuvre (vais-je tout casser ?) … et si la sureté de fonctionnement nous donnait des pistes de bonnes pratiques pour caler notre stratégie ?

Comme évoqué au chapitre précédent, la réaction automatique est déjà présente dans nos systèmes : antivirus, pare-feu, IPS, proxy filtrant sont des exemples parmi d'autres. Cependant, ces derniers travaillent avec peu de contexte et agissent, la plupart du temps, très localement...

L'enjeu est donc de prendre du recul afin de pouvoir mettre en place quelques scénarios de réponses au niveau système.

La sureté de fonctionnement des systèmes (industriels) amène à concevoir des systèmes permettant de continuer à rendre la mission malgré les pannes. Ces systèmes sont alors conçus (ou reçoivent des évolutions) pour être tolérants à la double panne (FS[6]/FS), triple panne (FS/FS/FS) ou plus. Il faut donc tirer profit des systèmes redondés pour définir sa stratégie de réaction. Il faut donc distinguer :

- Les actions réversibles
- Les actions irréversibles
- Les actions qui ne diminuent pas la capacité opérationnelle mais entament les jokers (par exemple une réaction qui coupe un élément d'un système tolérant à la triple panne)
- Les actions qui diminuent la capacité opérationnelle (légère baisse de performance acceptable)
- Les actions qui impactent la mission en cours

[5] Outils d'analyses de grand volume de données

[6] Fail Safe : le système reste sûr en cas de panne

Les pannes peuvent être causées par des origines humaines ou techniques. Il convient alors de prendre en compte dans les études de sureté de fonctionnement, autant le facteur humain (qui est faillible) que technique. Et si nous appliquons ces mêmes principes pour définir nos stratégies de réactions : l'humain doit être pris en compte dans le processus de réaction et dans les causes d'erreurs.

3.2 Préserver l'opérateur du stress, facteur d'erreur … ne pas (toujours) tout lui dire pour qu'il soit plus performant

Le stress est un élément important qui peut fortement perturber le comportement d'une personne.

Par exemple, dans certaines situations, les autorités décident de ne pas communiquer un évènement afin d'éviter les mouvements de panique des foules qui peuvent mener à des pertes humaines. Dans certains cas, il peut donc être judicieux de ne pas notifier votre niveau 1 mais directement votre niveau 2 voire niveau 3 qui doivent être moins pris par les tâches récurrentes et donc avoir plus de recul.

3.3 La formation et l'entrainement améliorent la performance sous stress, mais ne sont pas l'arme absolue contre les erreurs

Par exemple, une grande majorité des conducteurs ne sait pas gérer une situation d'urgence (évitement, dérapage …) et arrive à l'accident. Souvent ils portent le regard sur ce qu'il faut éviter (la voiture, le mur) alors qu'il faut au contraire porter le regard là où l'on veut s'échapper. Le cerveau et votre attitude se mobilisent autour de ce que vous regardez. Il faut donc apprendre aux personnes à regarder les solutions et non le problème. La formation du conducteur permet d'atténuer le stress mais en aucun cas ne permet de rendre une personne 100% insensible à ce dernier.

Il faut donc partir du postulat qu'une personne non formée fera beaucoup d'erreurs sous stress … et qu'une personne formée en fera aussi, mais un peu moins.

3.4 Supporter la réaction et notifier de votre support

L'industrie automobile a mis en place des systèmes pour aider l'humain à gérer ces situations (ABS[7], ESP[8], AFU[9], radar de distance/freinage d'urgence automatique …).

Ces derniers ont été pensés soit pour améliorer votre capacité, soit pour compenser votre manque de capacité (par ex. l'AFU car les personnes n'appuyaient pas assez sur la pédale de frein en cas d'urgence, ce qui n'est plus un souci depuis que l'ABS existe), soit encore pour remplacer l'humain en cas de non réaction (freinage automatique anticollision). Dans la plupart des cas, la notification est importante afin que l'opérateur puisse adapter sa posture en fonction de ce qui se passe.

---

[7] Système empêchant le blocage des roues au freinage

[8] Electronic Stability Program : système régulant les pertes d'adhérences

[9] Aide au Freinage d'Urgence : amplificateur de freinage

Par exemple, l'ABS signale dans le ressenti de la pédale de frein (vibrations) qu'il est en action. Cela permet de vous rendre compte que vous roulez peut-être trop vite sous la pluie et qu'il faudrait ralentir la cadence.

Pour l'opérateur, cela peut lui montrer qu'un phénomène est en cours et qu'il doit probablement augmenter sa vigilance sans pour autant avoir à réagir en l'état.

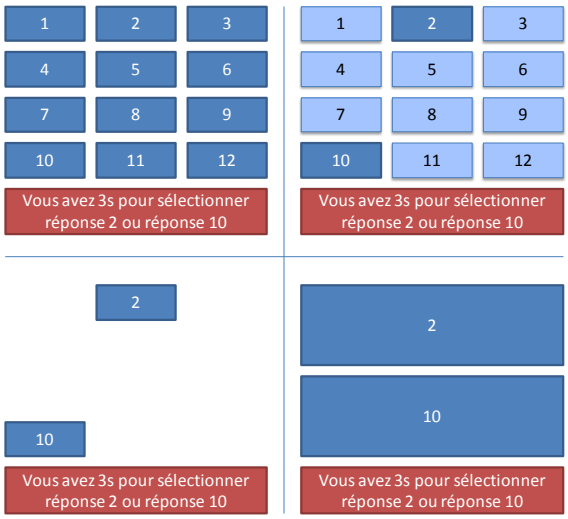
3.5 La réaction automatique pour préparer la réaction …

L'automobile a aussi mis en place des systèmes permettant de préparer un évènement fortement probable. Par exemple, les radars de distance détectent un obstacle et rapprochent les plaquettes des disques, ferment les ouvrants (vitres, toit ouvrant) et serrent les ceintures afin de préparer la suite (anticiper). Ces mesures n'ont aucun danger pour le conducteur et n'altèrent pas la fonction première du véhicule.

Une vision possible sur les systèmes d'information est d'anticiper les phases de collecte de données suite à la détection d'un incident (dump mémoire, dump disque dur, archivage de journaux ou de pcap …). Ces stratégies de réactions automatiques ne sont pas intrusives sur l'activité métier, mais permettent de simplifier et faire gagner du temps à vos équipes d'analystes.

3.6 Préserver l'opérateur du stress, facteur d'erreur … lui simplifier la vie, limiter les choix

Limiter les propositions de réactions à celles qui ont du sens… et ne présenter à l'opérateur que ces dernières en accès direct afin d'éviter les erreurs de décisions sous stress.



Imaginez un conducteur de train avec la procédure d'urgence suivante : pour stopper sa locomotive : basculer l'interrupteur 879 vers la droite ! La probabilité qu'il se trompe est proche de 100% (situation de stress + action complexe à exécuter). Lui présenter un bouton poussoir rouge est plus performant pour s'assurer de sa bonne réaction (qui n'est pas fiable à 100% à cause du facteur stress).

Parfois il est intéressant de mettre en avant une information disponible dans la représentation courante pour la mettre plus en avant. Par exemple, la bascule sur la réserve de carburant notifiée par un voyant/son/message pour « sortir de l'ordinaire » de la jauge d'essence.

La réaction automatique sur votre système de surveillance peut donc aussi être :

- Escalader automatiquement un incident impactant une zone ou un personnel V.I.P de votre entreprise,
- Présenter automatiquement des fiches de réactions associées au contexte de l'incident,
- Montrer les autres incidents similaires en cours sur votre système,
- Mettre en valeur un nouvel incident entrant quand il est de niveau plus élevé que ceux en cours de traitement par vos opérateurs. Cela permettra aux opérateurs de se rendre compte qu'il y a peut être un nouveau cas plus prioritaire qui vient d'arriver.

3.7 Adapter sa stratégie de réponse en fonction de l'impact

Et si notre bonne vieille sécurité des systèmes d'information qui inclut une analyse de risque, n'était-elle pas déjà armée pour nous aider à répondre à ce défi ? N'a-t-elle pas déjà des processus et méthodes pour identifier les biens sensibles et estimer le risque (tests d'intrusion, EBIOS, ISO27xxx, Mehari, …) ?

Il suffit d'appliquer, ces démarches d'analyse de risque, non plus dans une vocation unique de protection, mais en réfléchissant à l'impact de la réaction sur le fonctionnement opérationnel. Considérez la réaction (et ses changements associés) comme une nouvelle attaque ou vecteur d'attaque à analyser.

- Si l'impact est quasi nul ou faible mais peut être inversé rapidement (par un choix d'annulation rapide), la réaction automatique fait sens. Avertir l'opérateur n'est pas forcément obligatoire car cela peut le distraire de sa mission première (un simple virus bloqué par exemple).
- Si l'on approche des limites acceptables en cas de bon fonctionnement, la réaction automatique avec une notification a du sens. Dans ce cas l'opérateur doit être conscient qu'il a perdu ses jokers ou qu'un phénomène de masse est en cours (par exemple un virus présent sur une grande partie du parc ce qui signifie qu'il y a eu une propagation et que cela nécessite une investigation).
- Si l'impact d'une réaction altère le bon fonctionnement du système d'information, alors la décision doit être présentée à l'opérateur, voire passée dans un circuit de validation. L'automatisation garde son intérêt pour dérouler le plan après approbation de ce dernier.

4 Synthèse

4.1 Performance de supervision

Pour avoir une équipe performante, il ne faut pas la noyer d'informations.

- Pensez, dès la supervision, à la modélisation métier afin de pouvoir notifier les responsables métier en temps voulu,
- Commencez avec une stratégie modérée (en fréquence de notification et en précision des incidents) et améliorez cette stratégie quand votre équipe a pris le rythme, pas avant,
- Soignez la présentation des informations à vos opérateurs,
- Adapter les fréquences de notification à la capacité de traitement de votre équipe et à l'urgence associée aux alarmes,
- Dissociez notamment les alarmes à présenter en temps court et celles qui peuvent être présentées dans un rapport planifié
- Vous ne verrez pas tout, acceptez-le et basez aussi votre stratégie sur l'après : activité de réponse à incident.

4.2 Réactions

Pour bien protéger son système et prendre les bonnes décisions, il faut avant tout le connaitre.

- Cartographiez votre système d'information et les risques associés,
- Faites votre analyse de risque avec les métiers,
- Formez vos équipes pour être plus efficace en cas de crise : simulations régulières, etc.
- N'attendez pas le zéro faute de votre équipe, apprenez à vivre avec les erreurs,
- Préparez des plans de réaction qui identifient l'impact de l'attaque et de sa parade sur la continuité de service et qui seront
  - soit manuels,
  - soit lancés suite à une confirmation humaine (simple ou multiple),
  - soit automatique en fonction de votre analyse de risque,
- Limitez les options de réactions proposées au niveau 1 pour le préserver du stress,
- Notifiez les métiers que vous êtes en train de réagir quand ils sont impactés.

Remerciements à V. Bodson et R. de Besombes pour leur relecture.