Corrélation d'événements et découverte de scénarios d'attaque multi-étapes

Charles Xosanavongsa[2], Eric Totel[1] and Nizar Kheir[2]

[1]CentraleSupelec, Rennes, France, prénom.nom@centralesupelec.fr
[2]Thales Services, Palaiseau, France, prénom.nom@thalesgroup.com

***Résumé*—Les réseaux informatiques constituent la clé des infrastructures de technologies de l'information. Il est donc indispensable d'assurer leur sécurité avec la mise en place de moyens de prévention ainsi que de sondes de supervision. Cependant, les sondes génèrent énormément d'événements dont la grande majorité correspond à une utilisation bénigne et légitime du système. L'analyste de sécurité se retrouve facilement submergé par leur quantité et peut difficilement différencier les événements nécessitant une attention particulière, c'est-à-dire ceux liés à une attaque, des autres. Notre objectif est de réduire le gap sémantique, ainsi que la quantité d'information que l'analyste doit traiter, en identifiant les ensembles d'événements significatifs de symptômes d'attaques ayant réellement eu lieu. Plus précisément, nous nous intéressons à la reconstruction de scénario d'attaque à travers l'identification des actions de l'attaquant, déduites à partir des événements remontés par les sondes de supervision, et leur enchaînement logique et temporel.**

***Index Terms*—Détection d'intrusion, scénario d'attaque, corrélation d'événements, corrélation d'alertes**

I. Introduction

Malgré tous les moyens de prévention mis en place pour protéger un système informatique, il est raisonnable de faire l'hypothèse qu'un attaquant motivé trouvera toujours un moyen de s'y introduire pour mener à bien des objectifs tels que l'espionnage ou le sabotage. La mise en place de mécanismes de supervision de sécurité est donc indispensable. Cette supervision requiert le déploiement de nombreuses sondes ayant différentes capacités de détections et limitations. Ces sondes rassemblent des informations issues de plusieurs types de sources différentes en fonction du type de supervision défini :

— Réseau (NIDS, Network Analyzers, Packet Captures)
— Système (HIDS, Syscalls, ...)
— Applicatif (Applications Web, Firewall, DNS, ...)

De plus, il est important de noter que les sondes spécialisées dans la détection d'intrusion (IDS) se divisent en 2 catégories : la détection par signature, qui ne permet de détecter que les attaques connues, et la détection d'anomalies, qui permet de détecter tout comportement déviant vis-à-vis d'un comportement de référence. Dans les faits, les sondes génèrent énormément d'événements, dont la grande majorité correspond à une utilisation bénigne et légitime du système, difficiles à comprendre et à traiter telles quelles du fait de leur faible niveau sémantique. L'analyste de sécurité se retrouve facilement submergé par leur quantité et peut difficilement différencier les événements nécessitant une attention particulière, c'est-à-dire ceux liés à une attaque, des autres.

Notre objectif est de réduire le gap sémantique, ainsi que la quantité d'information que l'analyste doit traiter, en identifiant les ensembles d'événements significatifs de symptômes d'attaques ayant réellement eu lieu. Ces ensembles d'événements permettent par la suite de déduire les actions effectuées par l'attaquant, autrement dit, de reconstruire un scénario d'attaque multi-étapes. Les difficultés de cette démarche reposent dans le fait que les traces de l'attaquant sont dispersées dans le temps, dispersées dans plusieurs machines, projetées sur différents types de logs et potentiellement effacées par l'attaquant. Nous commençerons par exposer un rapide état de l'art dans la section suivante puis nous présenterons brièvement l'approche que l'on envisage de développer en section III.

II. Approches Existantes

A. Approches basées sur des connaissances d'expert

La plupart des précédents travaux portant sur la corrélation d'événements sont des approches statiques. Elles reposent sur l'utilisation d'une base de connaissance décrivant le système supervisé [1]. Cette base est souvent composée de la topologie réseau, la cartographie des logiciels déployés, une base de vulnérabilités ainsi que les capacités de détection d'intrusions du système. Partant de cette base de connaissance, deux stratégies sont utilisées :

1- Déduction automatique d'un graphe d'attaque de l'environnement supervisé à partir de la base de connaissance et génération des règles de corrélation associées [2]. Du point de vue de l'attaquant, un système informatique peut se représenter comme un réseau de vulnérabilités ayant la bonne propriété d'être inter-dépendantes, l'exploitation d'une vulnérabilité donnant les pré-requis pour en exploiter une autre sur le même système ou bien un autre. Un graphe d'attaque modélise les relations entre les différentes vulnérabilités connues des systèmes composant l'environnement. Chacun des chemins du graphe constitue une séquence d'actions de l'attaquant à travers l'exploitation des différentes vulnérabilités. Un chemin peut alors servir de scénario d'attaque pour la génération d'une règle de corrélation d'alertes.

2 - Description manuelle d'un scénario d'attaque redouté sous forme d'une règle de corrélation d'événements, à l'aide d'un langage de description d'attaque [3] [4], ces événements étant les symptômes du scénario vis-à-vis du système supervisé. Les règles décrites permettent ensuite de configurer des

moteurs de corrélation. Ces derniers effectuent une analyse, en temps réel, du flux d'événements et lèvent une alerte lorsqu'une règle de corrélation est vérifiée. Dans la pratique, l'écriture de règles de corrélation, permettant de reconnaître des scénarios d'attaque redoutés, est un problème très difficile. Pour cela, l'analyste de sécurité doit adopter deux points de vue : celui de l'attaquant et celui du défenseur. En plus d'imaginer un scénario d'attaque, il doit décrire précisément ses symptômes, autrement dit les alertes et événements, générés par les sondes de supervision, résultant des actions de l'attaquant dans le système supervisé. Pour cela, il doit donc avoir une connaissance fine des moyens de détection du système supervisé. Le principal avantage de cette méthode est que l'analyste sait exactement comment traiter une alerte issue de la vérification d'une telle règle de corrélation. De récents travaux [5] permettent de simplifier le processus de génération de règles en découplant les connaissances de l'environnement supervisé, telles que la topologie, les vulnérabilités et les moyens de détection, et les connaissances liées à la description de scénarios d'attaque redoutés. L'analyste commence par renseigner la base de connaissance de l'environnement supervisé. Il décrit ensuite un scénario d'attaque générique, indépendant de l'environnement supervisé, à l'aide d'un langage d'actions proposant des opérateurs logiques tels que *or*, *and* et *sequence*. Pour finir, cette description générique est automatiquement instanciée en une ou plusieurs règles de corrélation d'événements à l'aide des informations contenues dans la base de connaissance.

B. Approches basées sur l'analyse de données

Contrairement au type d'approches précédent, qui utilise les connaissances d'un expert pour décrire les événements à détecter et corréler, les approches data driven partent de l'ensemble des événements pour essayer d'en extraire de l'information. Elles reposent grandement sur les avancées du domaine du Machine Learning [6]. Encore une fois, la majorité des méthodes retrouvées dans la littérature se concentrent sur l'analyse des alertes issues d'IDS par signature. Elles sont généralement composées de deux modules. Un premier module analyse l'historique des alertes pour mettre en évidence les stratégies des attaquants. Le second module a pour objectif la reconnaissance, généralement en temps réel, des stratégies découvertes par le premier module en analysant le flux d'alertes. Ces méthodes ne permettent pas de détecter les nouveaux modes opératoires des attaquants.

Les travaux proposant de corréler différents types d'événements tels que les événements système, réseau et applicatif, sont plus rares. Les auteurs de [7] proposent une approche originale du problème en modélisant la détection de scénario d'attaque comme un problème de découverte de communautés dans un graphe. Ils s'inspirent des réseaux sociaux où les individus partageant les mêmes centres d'intérêts forment une communauté. En considérant un événement comme un individu, les événements générés par une même activité, bénigne ou malveillante, seraient enclins à être proches les uns des autres et donc à former une communauté. Plus précisément, un lien est établi pour chaque couple d'événements capturés pendant une fenêtre temporelle donnée. Un algorithme d'apprentissage supervisé associe ensuite un poids à chacun des liens. Le poids d'un lien est élevé si les deux événements qu'il relie sont tous les deux liés à une attaque ou bien à une action bénigne. Il est faible s'il relie un événement lié à une attaque et un autre lié à des actions bénignes. Des algorithmes de découverte de communautés peuvent ensuite être utilisés sur le graphe pondéré obtenu. Cette méthode est prometteuse. Cependant, elle ne permet pas d'effectuer de la détection en temps réel et ne semble pas pouvoir passer à l'échelle pour le moment. De plus, les expérimentations effectuées n'ont été menées que sur une unique machine et sur une courte fenêtre temporelle de l'ordre de quelques heures.

C. Approches basées sur la supervision des flux d'information

Ces méthodes reposent sur la supervision des flux d'information entre les processus et les objets (fichiers ou sockets) d'un système. Cette supervision permet la construction de graphes de flux d'information permettant à un analyste de remonter la chaîne causale de l'état d'un processus ou objet [8]. Accompagnée d'une politique de contrôle des flux d'information, la supervision permet également d'effectuer de la détection d'intrusion. Une alerte est levée lorsqu'un flux d'information enfreint une règle de la politique de contrôle [9]. L'utilisation du graphe de flux d'information permet alors de retrouver les processus et objets causalement liés à cette alerte et ainsi de découvrir son origine et l'impact de l'attaque sur le système. Malheureusement, la collecte des flux d'informations est coûteuse et la majorité des méthodes de l'état de l'art ne permettent pas de corréler des événements issus de différentes machines. De plus, la mise en place de telles politiques de contrôle des flux nécessite de contraindre l'utilisateur en lui empêchant par exemple de télécharger et exécuter des applications issues d'internet.

III. Méthode Envisagée

Chacune des actions effectuées par un utilisateur peut être observée par différentes sondes produisant des événements dispersés dans plusieurs types de logs. Cependant, peu de travaux s'intéressent à l'identification des événements hétérogènes engendrés par une même action pour tirer parti des différents niveaux d'information qu'ils apportent. Ce problème se traduit en la recherche des liens de causalité entre ces événements. Dans le cadre d'une attaque, les traces des différentes actions sont également projetées sur tous ces différents types de logs. Notre objectif est de pouvoir retrouver tous les événements liés à l'attaque pour aider l'analyste de sécurité à comprendre son contexte et ses impacts ainsi que lui permettre de caractériser l'attaque sous forme d'une règle de corrélation d'événements.

Dans la section II-C, nous avons présenté des approches basées sur la construction de graphes de flux d'information. Ces dernières établissent des liens de causalité entre les processus, fichiers et sockets au niveau système en déduisant les flux d'information à partir des logs syscalls. Malheureusement, il n'est pas toujours aisé d'établir un lien de causalité entre des événements non issus des logs syscalls. Prenons l'exemple d'une requête HTTP telle que "GET https ://www.google.com

HTTP/1.1". Si le nom de domaine n'est pas déjà résolu, la requête HTTP est précédée d'une requête DNS pour obtenir l'adresse IP de l'hôte à contacter. Les deux requêtes HTTP et DNS sont donc causalement liées. Une des méthodes, présentée dans la section II-B, permettant d'inférer le lien de causalité entre les événements est de comparer leurs attributs [7]. Dans notre exemple, les deux requêtes ont toutes les deux des attributs de valeur "www.google.com" et "x.x.x.x" pour l'ip du serveur à contacter. Ces égalités nous permettent d'établir un lien qui, en l'occurence, représente une causalité. Cependant, l'établissement d'un tel lien ne permet pas d'assurer la causalité.

Notre idée est de combiner ces deux modèles, flux d'information et inférence de causalité, pour tirer parti de la diversité des logs disponibles. Partant des logs syscalls, nous déduisons dans un premier temps le graphe de flux d'information. Ensuite, nous cherchons à enrichir le graphe en établissant le lien de causalité entre les événements des autres logs et les noeuds du graphe de flux d'information à l'aide de la comparaison de leurs attributs. Une fois ce graphe établi, notre objectif est de pouvoir retrouver les événements liés à l'attaque. Pour cela, nous utilisons les événements suspects, levés par les IDS par exemple, comme point de départ pour parcourir le graphe. Nous espérons ainsi retrouver les origines de l'attaque, différents niveaux d'information apportés par la diversité des logs, ainsi que ses potentiels impacts.

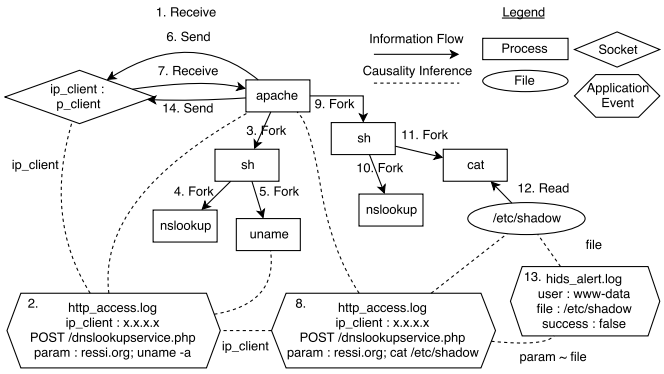

FIGURE 1: Injection de commande dans un script PHP

L'exemple suivant illustre notre approche. Nous souhaitons superviser une application web proposant le service nslookup. Pour cela, nous utilisons le mécanisme d'audit du kernel Linux pour journaliser les syscalls nous permettant de déduire des flux d'information tels que open, read, write, socket et ses dérivés et fork. Un HIDS surveille l'accès du fichier /etc/shadow et lève une alerte lorsqu'un utilisateur non privilégié tente de l'ouvrir. Nous activons également un module permettant au serveur HTTP apache de journaliser les requêtes effectuées ainsi que leurs paramètres. Peu de temps après la mise en ligne du service, une alerte est levée par le HIDS. Cet événement déclenche la recherche des événements causalement liés et nous permet de découvrir le scénario d'attaque illustré par le graphe de la Figure 1. L'événement suspect n°13, correspondand à notre alerte, a été déclenché par le n°12. La déduction des flux d'information nous permet de remonter jusqu'au serveur apache. L'événement n°8, dont les liens de causalité ont été inférés en comparant les attributs, nous renseigne la page vulnérable, la requête effectuée par l'attaquant ainsi que son adresse IP. Cette dernière permet d'établir le lien entre les deux requêtes n°8 et n°2, qui a son tour nous permet de découvrir les événements liés à la première action de l'attaquant. Le constat est finalement le suivant : un utilisateur averti effectue une requête sur la page dnslookupservice.php et tente, par la même occasion, une injection de commande pour voir si elle y est vulnérable. Voyant le résultat de sa commande, il décide de s'attaquer directement au fichier /etc/shadow dans l'objectif d'obtenir la liste des utilisateurs ainsi que les hash de leur mot de passe. L'utilisateur d'apache, www-data, n'ayant pas les droits de lecture sur le fichier /etc/shadow, les conséquences de l'attaque ne sont pas importantes.

IV. CONCLUSION

Ce document présente l'orientation actuelle de nos recherches : la découverte de scénarios d'attaque multi-étapes à l'aide de la corrélation d'événements issus de logs hétérogènes. La section II propose une classification des différentes méthodes de l'état de l'art. Le section III expose les principales idées de notre démarche. Dans un premier temps, nous cherchons à établir les liens de causalité entre les événements hétérogènes. Pour cela, nous utilisons deux modèles : la déduction des flux d'information au niveau des syscalls ainsi que l'inférence de causalité en comparant les valeurs des attributs des événements. Une fois établis, ces liens nous permettent de retrouver tous les événements causalement liés à un événement suspect. Nous espérons ainsi aider l'analyste de sécurité à déterminer si cet ensemble d'événements correspond à un scénario d'attaque.

RÉFÉRENCES

[1] B. Morin, L. Mé, H. Debar, and M. Ducassé, "M4d4 : A logic-based model to support alert correlation in intrusion detection," *Information Fusion*, vol. 10, no. 4, pp. 285–299, 2009.

[2] S. Noel, E. Harley, K. H. Tam, and G. Gyor, "Big-Data Architecture for Cyber Attack Graphs," 2014.

[3] J. Goubault-Larrecq and J. Olivain, "A smell of orchids," in *International Workshop on Runtime Verification*. Springer, 2008, pp. 1–20.

[4] E. Totel, B. Vivinis, and L. Mé, "A language driven intrusion detection system for event and alert correlation," in *IFIP International Information Security Conference*. Springer, 2004, pp. 209–224.

[5] E. Godefroy, E. Totel, M. Hurfin, and F. Majorczyk, "Generation and assessment of correlation rules to detect complex attack scenarios," in *2015 IEEE Conference on Communications and Network Security (CNS)*, Sep. 2015, pp. 707–708.

[6] B. Zhu, "Alert Correlation for Extracting Attack Strategies," *International Journal of Network Security*, 2005.

[7] K. Pei, Z. Gu, B. Saltaformaggio, S. Ma, F. Wang, Z. Zhang, L. Si, X. Zhang, and D. Xu, "HERCULE : attack story reconstruction via community discovery on correlated log graph." ACM Press, pp. 583–595.

[8] A. Gehani and D. Tariq, "SPADE : support for provenance auditing in distributed environments," in *Proceedings of the 13th International Middleware Conference*. Springer-Verlag New York, Inc., 2012, pp. 101–120.

[9] M. N. Hossain, S. M. Milajerdi, J. Wang, B. Eshete, R. Gjomemo, R. Sekar, S. Stoller, and V. Venkatakrishnan, "SLEUTH : Real-time attack scenario reconstruction from COTS audit data," in *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, 2017, pp. 487–504.