Définition Formelle de la Relation de Dépendance Causale entre Événements Journalisés

Charles XOSANAVONGSA*†, Eric TOTEL* et Olivier BETTAN†
*CentraleSupélec, Inria, Univ Rennes, CNRS, IRISA, prénom.nom@centralesupelec.fr
†Thales Six GTS France, prénom.nom@thalesgroup.com

***Abstract*—Malgré tous les moyens de prévention mis en place, un attaquant motivé trouvera toujours le moyen d'infiltrer un réseau informatique. Il est donc indispensable de superviser le système sous l'angle de la sécurité informatique. Partant d'un événement suspect, un analyste de sécurité peut commencer son investigation en identifiant les événements liés à l'événement d'intérêt. Dans l'objectif de retrouver les traces des actions de l'attaquant, il cherche alors à retrouver les liens de dépendance causale entre les événement journalisés. Dans les faits, ce type de lien n'est pas simple à définir et découvrir. Le travail présenté dans cet article a pour objectif de définir formellement ce qu'est la relation de dépendance causale entre les événements journalisés, ces événements pouvant être de type différent.**

I. Introduction

Du fait de leur environnement compétitif, les entreprises sont sujettes à l'espionnage industriel, le sabotage ou encore le vol de données. Le nombre de rapports de sécurité détaillant les brèches subies par les entreprises ne fait qu'augmenter. Le constat est le suivant : un attaquant motivé finit par réussir à s'infiltrer dans un réseau malgré les moyens de prévention mis en place. La supervision en sécurité du système est donc indispensable pour pouvoir observer toute action effectuée par l'attaquant. Pour cela, plusieurs sondes et systèmes de détection d'intrusion (IDS) sont déployés par l'administrateur. Partant d'un événement suspect, un analyste de sécurité peut commencer son investigation en identifiant les groupes d'événements qui sont liés à l'événement[1] d'intérêt. Dans l'objectif de retrouver les traces des actions de l'attaquant, il cherche alors à retrouver les liens de dépendance causale entre les événements. Dans les faits, ce lien n'est pas simple à découvrir et à définir. La littérature manque d'une définition formelle de la sémantique de ces liens et donc de la description d'une relation entre événements journalisés. Dans cet article, nous proposons une définition claire de la relation de dépendance causale entre événements. L'objectif est de définir un modèle formel permettant, dans l'idéal, d'unifier les travaux existants portant sur la notion de dépendance causale entre événements hétérogènes, c'est-à-dire, des événements issus de différentes couches d'abstraction (réseau, système d'exploitation, application). La relation entre événements que nous définissons permet la découverte de tous les événements pouvant être considérés comme étant la cause ou l'effet d'un événement d'intérêt, telle qu'une alerte par exemple. Cet article est organisé de la manière suivante : la Section II présente progressivement le modèle. Dans la Section III, nous présentons de manière succincte différentes implémentations permettant de calculer des parties du modèle. Finalement, la Section IV conclut l'article et donne une ouverture sur les travaux à venir.

[1]Dans cet article, un « événement » correspond à toute information journalisée par une application, une sonde ou un IDS. Lorsqu'un événement est produit par un IDS, il peut aussi être appelé « alerte ».

II. Définition d'une Relation de Dépendance Causale entre Événements

Dans un premier temps, nous commençons par introduire progressivement la notion d'Action Contextuelle ainsi que sa relation de dépendance causale. Nous décrivons ensuite les liens qui unissent actions contextuelles et événements journalisés. Nous pourrons alors définir la notion de dépendance causale entre événements journalisés.

A. Introduction de la Relation de Dépendance Causale entre Actions Contextuelles

1) Définition d'une Action Contextuelle: Une action contextuelle est composée de : (1) l'action effectuée par un objet, de manière similaire au modèle de Lamport [1], et comprennent les actions effectuées par les objets actifs (processus, réseau) d'un système distribué; et (2) la valeur du contexte de l'objet au moment où l'action a été effectuée, autrement dit, l'état de l'objet, au sens du modèle défini par d'Ausbourg [2]. Formellement, une action contextuelle est un couple $(a,(o,t))$, où a est l'action effectuée et (o,t) l'état de o au temps t. Pour résoudre le problème de dépendance entre objets de types différents, nous devons distinguer les objets en deux catégories : les objets actifs, qui effectuent des actions (tel que les processus ou le réseau), et les objets passifs (comme les conteneurs d'information tels que les fichiers, sockets, ...). Un objet actif est supposé effectuer des actions pouvant être liées aux contextes de l'objet. Pour les objets passifs, seul leur contexte peut être observé.

Définition 1: L'ensemble des actions produites par un objet actif ou passif est $ObjectActions(o) = \{a_i\} \cup \{\emptyset\}$ avec a_i les actions pouvant être effectuées par o et $\emptyset$ l'absence d'action.
Par exemple, un processus p invoque des appels système via l'API dédiée pour requêter un service au kernel. Nous considérons alors que les invocations d'appels système sont des actions de $ObjectActions(p)$. Nous pouvons maintenant introduire formellement la notion d'action contextuelle :

Définition 2: Une *Action Contextuelle* est un couple $(a,(o,t))$ où $a \in ObjectActions(o)$ et (o,t) est l'état de l'objet o au temps t.

Étant donné deux actions a et b produites par un processus donné telles que $a \prec b$, « $\prec$ » étant la relation *happens-before* [1], Lamport suppose que b est causalement dépendant de a. Nous voulons définir un modèle plus précis permettant de casser la relation de causalité entre a et b lors de l'évolution d'un objet, c'est-à-dire, si l'état de cet objet est indépendant de ses états précédents au sens de d'Ausbourg. En pratique, de nombreux services ne gardent pas en mémoire les différentes séquences d'exécution. Ceci implique que l'exécution d'un processus donné peut être divisée en intervalles de temps où les exécutions sont partiellement ou complètement indépendantes l'une de l'autre. Dans notre modèle, un tel intervalle dans l'exécution d'un processus est appelé *Session* :

Définition 3: Étant donné un objet o, une session $Session_n(o)$ est une séquence d'actions contextuelles $(a_i, (o, t_i))$ où $a_i \in ObjectActions(o)$ et $Session_n(o) = \{(a_i, (o, t_i)) \ / \ (o, t_i) \rightarrow (o, t_{i+1})$ et $(o, t_{end_{n-1}}) \not\rightarrow (o, t_{start_n})$ et $(o, t_{end_n}) \not\rightarrow (o, t_{start_{n+1}})\}$. t_{start_n} est l'horodatage de la première action contextuelle de $Session_n(o)$, t_{end_n} est l'horodatage de la dernière action contextuelle de $Session_n(o)$ et « $\rightarrow$ » correspond à la relation de dépendance causale définie par d'Ausbourg [2].

La notion de session n'est pas nouvelle. En particulier, les actions délimitant les sessions au sein des processus ayant une longue durée de vie, tels que les services par exemple, ont fait l'objet d'une étude approfondie pour permettre de diminuer le nombre de fausses dépendances causales entre les actions [3].

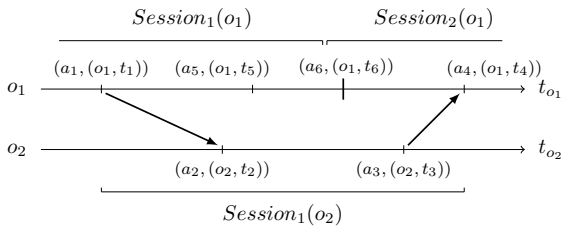

Fig. 1. Dépendances causales entre actions contextuelles dans différentes sessions.

Une exécution d'un objet o est l'union de toutes $Sessions(o)$, comme illustré par la Fig. 1. Dans cet exemple, l'action a_6 commence une nouvelle session. Ainsi, (o_1, t_6) est indépendant de l'état précédent (o_1, t_5). En pratique, de telles actions peuvent être identifiées grâce à une connaissance experte par exemple. La notion de sessions s'applique à tout type d'objet. Exemples : un fichier peut être effacé; pour un processus Apache, deux requêtes consécutives sont indépendantes. Dans les faits, une action qui commence une nouvelle session peut être produite par une application ou encore le système d'exploitation.

2) Définition de la Relation de Dépendance Causale entre Actions Contextuelles: Le concept d'action contextuelle prend en compte les actions effectuées par les objets et leurs états lors des actions. Ceci nous permet de profiter des deux modèles pour définir une relation de dépendance causale entre différents types d'actions. Cette nouvelle relation s'appelle *Dépendance Causale entre Actions Contextuelles*, dénotée « $\mapsto$ », et est définie sur l'ensemble de toutes les actions contextuelles produites par tous les objets du système.

Définition 4: Étant donné deux actions contextuelles $(a_1, (o_1, t_1))$ et $(a_2, (o_2, t_2))$, $(a_2, (o_2, t_2))$ est causalement dépendante de $(a_1, (o_1, t_1))$, dénoté $(a_1, (o_1, t_1)) \mapsto (a_2, (o_2, t_2))$, lorsque :

1) o_1 et o_2 sont le même objet o, $\exists$ n tel que $(a_1, (o, t_1)) \in Session_n(o)$, $(a_2, (o, t_2)) \in Session_n(o)$ et $t_1 < t_2$;
2) ou $o_1 \neq o_2$, $(o_1, t_1) \rightarrow (o_2, t_2)$, c'est-à-dire que les deux états sont causalement dépendants au sens de d'Ausbourg. Autrement dit, il existe un flux d'information de l'état (o_1, t_1) vers l'état (o_2, t_2);
3) ou $o_1 \neq o_2$, l'action a_1 correspond à l'envoi d'un message m et l'action a_2 correspond à la réception de m, ce qui signifie que $a_1 \prec a_2$ en utilisant la relation *happened-before* de Lamport;
4) ou $\exists$ $(c, (o, t))$ tel que $(a_1, (o_1, t_1)) \mapsto (c, (o, t))$ et $(c, (o, t)) \mapsto (a_2, (o_2, t_2))$.

La Fig. 1 illustre l'utilisation de notre modèle. La relation « $\mapsto$ » étant transitive, nous avons par exemple $(a_1, (o_1, t_1)) \mapsto (a_4, (o_1, t_4))$ même si ces deux actions contextuelles appartiennent à deux sessions différentes. Il est important de noter que les deux objets possèdent leur propre horloge, t_{o_1} et t_{o_2}, et que notre modèle ne nécessite pas qu'elles soient synchronisées.

B. Des Actions Contextuelles vers les Événements Contextuels et les Événements Journalisés

En pratique, les actions peuvent être journalisées ou non. En effet, les événements journalisés sont produits par des applications ou des sondes (au niveau du système d'exploitation ou du réseau).

1) Définition des Événements Contextuels: Comme défini dans [4], un événement est défini comme étant « une action identifiable ayant lieu sur un dispositif et étant enregistrée comme une entrée de journal ». Il est important de noter qu'une action peut ne pas être journalisée. Dans ce cas, elle n'est observée par aucune sonde. Certaines actions peuvent donc être manquées par l'analyste de sécurité. De plus, plusieurs sondes peuvent être déployées au sein du système supervisé. Ainsi, une action peut également être observée par plusieurs sondes de type différent. Les événements correspondant à une même action sont alors répartis sur différents fichiers de journalisation.

Étant donné l'ensemble des événements journalisés du système, dénoté $\mathbb{E}$, chacun des événements est produit à un temps donné, c'est-à-dire au moment de son horodatage, en observant une action contextuelle effectuée par un objet.

Définition 5: Un *Événement Contextuel* est un triplet (e, o, t_e) où $e \in \mathbb{E}$, o représente l'objet observé et t_e est l'horodatage de l'événement e.

D'après la définition 2, l'action a d'une action contextuelle donnée $(a, (o, t_a))$ peut représenter une action réelle ou l'absence d'action. Par conséquent, a peut ne pas être observable. On peut ainsi étendre la définition précédente en

introduisant l'événement contextuel $(\emptyset, o, t_a)$ correspondant à l'absence d'observation de a au temps t_a. Nous pouvons maintenant introduire la fonction *Obs* :

Définition 6: Étant donné une action $a \in ObjectActions(o)$ arrivant au temps t_a, l'observation d'une action contextuelle est $Obs\big((a,(o,t_a))\big) = \{(e_i, o, t_{e_i})\} \cup \{(\emptyset, o, t_a)\}$ où $e_i \in \mathbb{E}$ est l'observation de a; $(\emptyset, o, t_a)$ correspond à l'absence d'observation de a et donc à l'absence d'événement.

2) Définition de la Relation de Dépendance Causale entre Événements Contextuels: L'objectif de ce modèle est de définir la relation de dépendance causale entre événements. Pour cela, nous devons tout d'abord définir la relation de dépendance causale entre événements contextuels dénotée « $\rightharpoonup$ ».

Définition 7: Étant donné deux événements contextuels (e_1, o_1, t_{e_i}) et (e_2, o_2, t_{e_j}), (e_2, o_2, t_{e_j}) est causalement dépendant de (e_1, o_1, t_{e_i}), dénoté $(e_1, o_1, t_{e_i}) \rightharpoonup (e_2, o_2, t_{e_j})$, si et seulement si il existe deux actions contextuelles $(a_1, (o_1, t_1))$ et $(a_2, (o_2, t_2))$ telles que $(a_1, (o_1, t_1)) \mapsto (a_2, (o_2, t_2))$ et $(e_1, o_1, t_{e_i}) \in Obs\big((a_1, (o_1, t_1))\big)$ et $(e_2, o_2, t_{e_j}) \in Obs\big((a_2, (o_2, t_2))\big)$.

3) Définition de la Dépendance Causale entre Événements: Nous arrivons maintenant au résultat attendu du modèle, c'est-à-dire, la définition de la dépendance causale entre événements dénotée « $\triangleright$ ».

Définition 8: Étant donné deux événements e_1 et e_2, e_2 est causalement dépendant de e_1, dénoté $e_1 \triangleright e_2$, si et seulement si $(e_1, o_1, t_{e_1}) \rightharpoonup (e_2, o_2, t_{e_2})$ où o_1 et o_2 sont respectivement les objets observés et t_1 et t_2 sont les horodatages des événements.

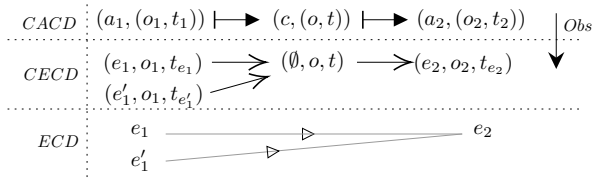

Fig. 2. Tableau récapitulatif des 3 relations définies dans le modèle.

Le tableau représenté par la figure 2 résume les trois relations définies dans notre modèle.

4) Graphes de Cause et de Dépendance pour les Événements: Les relations « $\mapsto$ », « $\rightharpoonup$ » et « $\triangleright$ » définissent respectivement des ordres partiels sur les ensembles des actions contextuelles, des événements contextuels et des événements. Elles sont de plus transitives. Cette propriété nous permet de construire le *Graphe de Cause*, $cause(e) = \{e'/e' \triangleright e\}$, et le *Graphe de Dépendance*, $dep(e) = \{e'/e \triangleright e'\}$, d'un événement d'intérêt donné e. Ces deux graphes représentent tous les événements qui, respectivement, contribuent ou dépendent de l'événement donné.

III. L'Implémentation Idéale du Modèle

Les méthodes de calcul de dépendances causales permettent d'observer et journaliser les actions effectuées par les objets actifs du système supervisé. Les travaux existants permettent d'observer le système sous un point de vue donné. Ils ne donnent donc qu'une information partielle de ce qu'il s'est réellement passé sur le système. Il est donc important de noter que les travaux existants ne permettent de calculer qu'une approximation du modèle. Cependant, l'implémentation la plus avancée de notre modèle consisterait en la fusion des technologies suivantes :

- Observation des flux d'information entre les objets du kernel tel que dans [5] par exemple;
- Méthodes d'instrumentation des applications au niveau du code source (application en elle même ou librairies telle que la *libc*) ou au niveau du binaire tel que dans [3] par exemple;
- Méthodes de capture et de rejeu des états du système ou des objets tel que dans [6] par exemple;
- Systèmes d'échange de messages issus du domaine de recherche des systèmes distribués.

IV. Conclusion

L'objectif de cet article est de proposer une unification des relations de causalité définies entre les entités actives, passives et les événements journalisés du système. Plus précisément, nous définissons formellement la notion de dépendance causale entre les événements journalisés et les alertes produites par des processus distribués. Inspiré des relations de dépendance causale de Lamport et de d'Ausbourg, nous définissons la notion d'action contextuelle et la relation de dépendance causale associée. Ceci nous permet d'introduire progressivement la notion de dépendance causale entre événements journalisés. À notre connaissance, aucun système ne fournit toutes les notions requises pour calculer notre modèle. Seules des parties de ces notions sont disponibles dans une même implémentation donnée. Faute d'espace, nous ne présentons pas les détails de notre implémentation actuelle du modèle. Cette première implémentation décrit une approche partant des événements journalisés pour calculer une approximation du modèle. Les travaux à venir vont consister en la proposition du complémentaire de l'approche actuelle. C'est-à-dire, instrumenter un système d'exploitation et certaines applications pour pouvoir facilement calculer et retrouver les dépendances causales, au sens de notre modèle définit dans cet article, entre les événements journalisés.

References

[1] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Commun. ACM*, vol. 21, no. 7, pp. 558–565, Jul. 1978.

[2] B. d'Ausbourg, "Implementing secure dependencies over a network by designing a distributed security subsystem," in *European Symposium on Research in Computer Security*. Springer, 1994, pp. 247–266.

[3] K. H. Lee, X. Zhang, and D. Xu, "High Accuracy Attack Provenance via Binary-based Execution Partition," in *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*, Feb. 2013.

[4] European Commission. (2010) Standard on logging and monitoring.

[5] Y. Liu, M. Zhang, D. Li, K. Jee, Z. Li, Z. Wu, J. Rhee, and P. Mittal, "Towards a timely causality analysis for enterprise security," in *Proc. of the 25th Network and Distributed System Security Symposium (NDSS)*, 2018.

[6] Y. Ji, S. Lee, E. Downing, W. Wang, M. Fazzini, T. Kim, A. Orso, and W. Lee, "Rain: Refinable attack investigation with on-demand inter-process information flow tracking," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 377–390.