Système de détection des intrusions distribué pour les systèmes industriels

Estelle Hotellier, Franck Sicard, Julien Francq, Stéphane Mocanu

Système de détection des intrusions distribué pour les systèmes industriels

Estelle Hotellier*†, Franck Sicard*, Julien Francq* and Stéphane Mocanu†
*Naval Cyber Laboratory, Naval Group
83190 Ollioules, France
{estelle.hotellier, franck.sicard, julien.francq}@naval-group.com
†Laboratoire d'Informatique de Grenoble
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, 38000 Grenoble, France
{estelle.hotellier, stephane.mocanu}@inria.fr

***Résumé*—Ce papier présente un travail en cours concernant une méthodologie distribuée de détection des intrusions pour les systèmes industriels (ICS). L'approche concerne les ICS complexes, c'est-à-dire des systèmes hiérarchiques et distribués reposant sur des boucles de contrôle locales. Dans le cadre de ces travaux, nous nous intéressons aux attaques basées sur la connaissance des processus industriels qui visent à compromettre la mission de l'ICS.**

Nous adoptons une méthode de détection comportementale basée sur des spécifications de sécurité du système industriel. Les spécifications sont traduites en propriétés de sûreté qui représentent alors des exigences de cybersécurité et peuvent être exploitées par notre système de détection des intrusions (IDS). Notre approche se base sur l'IDS open source Zeek, pour lequel nous avons développé des extensions afin d'étendre ses capacités de détection pour les systèmes industriels. À l'aide de cet outil, nous souhaitons développer une architecture de détection distribuée capable de traiter des captures de trafic réseau hétérogènes provenant de différentes sources. Le but de ce papier est de présenter notre méthodologie de détection et les pistes de recherches concernant son aspect distribué.

Un système de contrôle commande industriel est un réseau de composants physiques et numériques qui interagissent pour l'exécution d'une tâche en milieu industriel [1]. Présents dans de nombreux secteurs (énergie, transports, médical, manufacturier, etc), ils remplissent des tâches essentielles dans nos sociétés. Ces systèmes sont plus que jamais la cible de cyberattaques qui perturbent la partie physique du processus industriel allant parfois jusqu'à la détruire. De telles attaques peuvent avoir des conséquences désastreuses comme le témoigne quelques exemples récents (en avril 2022, le malware Industroyer2 a été déployé dans un poste électrique du réseau de distribution électrique Ukrainien [2] par exemple).

Ces récentes actualités soulignent la nécessité de déployer des solutions de cybersécurité pour les systèmes industriels. À l'origine, ces systèmes fonctionnaient de manière isolée et ne devaient faire face qu'à des risques accidentels. Mais les progrès des dernières décennies ont apporté de nouvelles technologies et toujours plus d'interconnexions entre les équipements. Par conséquent, les systèmes industriels doivent désormais faire face à des risques de cyberattaques.

En raison des contraintes strictes liées aux temps de réponse des infrastructures industrielles et des ressources informatiques limitées, les dispositifs de sécurité classiques (antivirus, pare-feu, etc) sont difficiles à déployer dans les systèmes industriels. L'approche que nous développons est un IDS comportemental par spécifications, basé sur des captures de trafic réseau, présentant un monitoring décentralisé.

Dans la section suivante, nous présentons les caractéristiques du système considéré. La section II décrit le modèle de menace et la typologie d'attaque associée. Dans la section III, nous détaillons notre approche de détection. Dans la section IV, nous discutons de l'implémentation avec un cas d'étude défini et l'utilisation de Zeek dans notre framework. La section V résume l'état de l'art. Nous concluons et présentons les travaux futurs dans la section VI.

I. Caracteristiques des systèmes industriels

Dans notre approche, nous considérons des systèmes industriels complexes, c'est-à-dire avec un contrôle hiérarchique et distribué. Cela signifie que plusieurs objectifs de contrôle coexistent afin d'imposer un comportement global correct du système. Le système global est composé de sous-processus, contrôlés localement. Nous appelons *boucles locales*, les blocs de construction élémentaires de tels systèmes ; ces boucles locales sont généralement composées d'un contrôleur local, de capteurs et d'actionneurs. Ces boucles locales permettent d'atteindre des objectifs de contrôle de plus bas niveau, comme le contrôle de la vitesse ou la position d'un moteur par exemple. Les objectifs de plus haut niveau, comme le suivi de trajectoire, sont assurés par des contrôleurs de plus haut niveau qui se chargent de synchroniser les boucles locales. Ces contrôleurs de niveau supérieur sont généralement des automates programmables industriels (API). Le niveau le plus élevé est le système de contrôle et d'acquisition de données qui permet la coordination entre les niveaux inférieurs et permet une vision d'ensemble des données du processus pour les opérateurs humains.

Considérons maintenant un système comportant plusieurs API affectés à différentes tâches industrielles, eux-mêmes synchronisant différentes boucles locales. Il existe alors des objectifs de contrôles globaux (de haut niveau) permettant d'ordonnancer les différentes tâches industrielles. Ainsi, dans les systèmes industriels complexes, des objectifs de contrôles locaux coexistent avec des objectifs de contrôle globaux. C'est

typiquement le type d'architecture que l"on trouve dans une chaîne de production par exemple : les moyens de production ont des objectifs lors de leurs activités individuelles, mais puisqu'ils partagent des ressources communes (les pièces à produire notamment), ils doivent également suivre des objectifs de synchronisation.

II. Modèle de la menace

Nous nous intéressons aux attaques basées sur la connaissance des processus industriels (« Process-Aware » [3], [4]). Ces attaques nécessitent une connaissance approfondie du système physique et utilisent des trames légitimes dont la syntaxe respecte parfaitement les spécifications des protocoles de communication. Voici quelques exemples : envoyer des commandes utilisant des ordres légitimes hors de leur contexte, forcer des valeurs hors des seuils autorisés pour des variables de processus, inverser des commandes valides, perturber la logique de commande.

Nous définissons le modèle de la menace par rapport à la matrice "MITRE ATT&CK for ICS" [5] qui fournit les tactiques pertinentes pour notre étude. En l'occurence dans notre cas, il s'agit des tactiques : "Collection", "Command & Control", "Inhibit Response Function", "Impair Process Control" et "Impact". Les autres tactiques ne concernent pas ce travail car nous considérons que la menace est déjà dans le système.

III. Approche de détection

La particularité de notre approche est d'utiliser des spécifications de sécurité du système pour caractériser son comportement normal. Ci-dessous nous décrivons la construction du modèle dans un premier temps puis son fonctionnement à l'exécution.

A. Construction du modèle

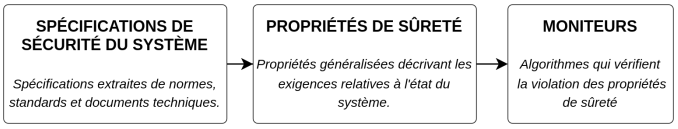

Figure 1. Construction du modèle

La Fig 1 représente les étapes de la construction de notre modèle de détection. L'objectif de notre approche est de lier les propriétés de sûreté et de sécurité. Selon la norme IEC 60050, le terme *sécurité* (*safety* en anglais) est « l'absence de risque inacceptable » [6] alors que *sûreté* (*security* en anglais) considère les menaces potentielles dues à des actions malveillantes [7]. Le point de départ de notre approche sont les spécifications de sécurité du processus physique qui ne peuvent pas être exploitées directement à des fins de détection. Ces spécifications sont rapportées au domaine de la cybersécurité en introduisant une signification au niveau du réseau, on parle alors de propriétés de sûreté. La propriété de sûreté peut être traduite dans un formalisme exploitable pour la détection ce qui permet l'implémentation d'un moniteur.

B. Runtime Monitoring

Lors de l'exécution d'un système, les moniteurs générés à l'étape de construction du modèle évaluent les propriétés de sûreté précédemment établies. Cette approche est inspirée du *runtime monitoring*, c'est-à-dire le fait d'observer et évaluer le comportement d'un système pendant son exécution [8]. En plus de la propriété à vérifier, un moniteur prend en entrée le trafic réseau du système. Enfin, la violation d'une propriété lève une alerte.

IV. Implementation

L'enjeu principal de notre système de détection est son aspect distribué. Nous considérons un système industriel complexe dans lequel toutes les données ne peuvent pas être centralisées. Pour adresser cette problématique, nous présentons dans un premier temps notre étude de cas ; nous discuterons ensuite notre architecture de détection qui est basée en partie sur Zeek.

A. Etude de cas

Nous considérons un atelier de production de découpe de pièces métalliques. La chaîne que nous considérons est constituée de deux machines : (i) un bras articulé, assurant l'approvisionnement des pièces et (ii) un robot de découpe, permettant de tailler les pièces métalliques suivant un programme défini.

Au niveau individuel, chacune des machines possède des spécifications de sécurité et des propriétés de sûreté qui en découlent. Ainsi, des moniteurs peuvent être implémentés pour des propriétés de sûreté locales. Au niveau global, les activités des deux machines sont liées puisque la première machine approvisionne la seconde en pièces. Il est alors nécessaire d'implémenter des propriétés de sûreté globales.

Pour implémenter à la fois des moniteurs locaux et globaux, un grand nombre d'évènements (au sens d'informations contenues dans le trafic réseau) doivent être surveillés. Pour des raisons de scalabilité, il est nécessaire d'adopter une approche distribuée et de ne pas centraliser tous les évènements. Plusieurs challenges sont alors à relever :

- Traitement de données provenant de réseaux hétérogènes, de sources différentes ;
- Répartition des évènements à surveiller au sein de l'approche distribuée.

B. Adapter Zeek aux systèmes industriels

Nous nous basons sur l'IDS open-source Zeek v4.1.0-dev.704 [1]. Les avantages de l'utilisation de Zeek sont nombreux : outil supporté par une large communauté, possibilité de développer des règles de détection très sophistiquées, framework orienté vers la détection comportementale avec des modules permettant des analyses temporelles, possibilité de déploiement distribué.

Pour nos besoins de détection, nous avons étendu les capacités de détection de Zeek aux bus de terrain. En effet,

1. https://www.zeek.org

Zeek est par défaut limité aux protocoles basés sur Ethernet. Nous avons donc développé des Workers additionnels pour permettre la capture de trafic réseau sur des bus de terrain. Les Workers sont des agents qui interceptent les trames et filtrent le trafic réseau. D'une part, nous avons développé un Worker pour le bus de terrain CAN. Ce Worker est un module directement déployé dans le framework de Zeek ; il est basé sur la librairie SocketCAN [2]. D'autre part, nous avons développé un Worker autonome afin de permettre la capture de trafic sur le bus de terrain Modbus RTU. Le Worker Modbus RTU est développé en C et C++, le Worker CAN est développé en langage natif Zeek.

L'architecture de notre contribution est illustrée Fig. 2 (seuls les Workers additionnels sont représentrés). Les Workers transmettent les trames à un Broker qui centralise et distribue les messages aux outils de détection. Les scripts de détection contiennent les moniteurs et ils s'abonnent à certains types de messages. Les scripts de détection sont distribués au sein de l'architecture du système observé et leur emplacement dépend des propriétés de sûreté à vérifier (locales ou globales).

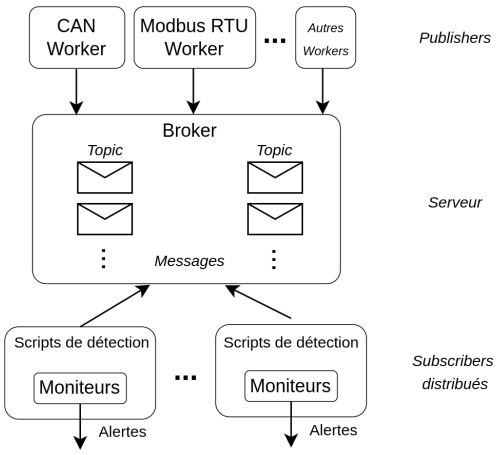

FIGURE 2. Architecture de la contribution

V. ETAT DE L'ART

La majorité des IDS comportementaux dans la littérature se concentrent sur l'analyse du trafic réseau, sans considérer la connaissance des processus physiques [9]–[11]. Ces approches sont souvent inefficaces contre les attaques Process-Aware.

Parmi les études qui se concentrent sur les attaques Process-Aware, l'approche de [12] est similaire à la nôtre dans la méthodologie de construction des moniteurs et dans l'utilisation de l'outil de détection Zeek. Cependant, la détection est limitée aux programmes séquentiels et nécessite une connaissance des programmes des automates. Une autre approche complémentaire est celle de [13]. Les spécifications des systèmes sont également utilisées afin d'extraire les propriétés de sûreté pour les variables discrètes et continues. Les auteurs caractérisent entièrement les états critiques et l'état actuel du système est évalué par sa distance à un état critique. L'approche a été adaptée pour les systèmes discrets dans [14].

2. https://www.kernel.org/doc/html/latest/networking/can.html

VI. CONCLUSION ET PERSPECTIVES

Nous avons présenté un IDS distribué, basé sur des spécifications pour les systèmes industriels complexes. Le système considéré est déployé autour de plusieurs boucles locales assurant des fonctions variées et étant parfois correlées. La construction du modèle repose sur les spécifications de sécurité du système. Le comportement du système est évalué pendant son execution à l'aide de moniteurs et les propriétés de sûreté observées peuvent être locales ou globales. D'un point de vue pratique, l'implémentation repose sur Zeek et est étendue pour le support des bus de terrain, l'analyse et la surveillance des événements.

L'étude de l'aspect distribué de notre approche de détection est à l'heure actuelle un travail en cours. Des expériences doivent être menées afin d'évaluer les performances de notre méthodologie.

RÉFÉRENCES

[1] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, and S. Lightman, "Guide to Operational Technology (OT) Security," NIST Special Publication 800-82 Rev. 3, Tech. Rep., 2022, [Online] https://doi.org/10.6028/NIST.SP.800-82r3.ipd, last accessed Jan. 2023.

[2] "INDUSTROYER.V2 : Old Malware Learns New Tricks," Mandiant, Technical Report, 2022, [Online] https://www.mandiant.com/resources/blog/industroyer-v2-old-malware-new-tricks, last accessed Jan. 2023.

[3] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. Nai Fovino, and A. Trombetta, "A multidimensional critical state analysis for detecting intrusions in SCADA systems," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, 2011.

[4] J. Nivethan and M. Papa, "A SCADA Intrusion Detection Framework that Incorporates Process Semantics," in *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, 2016, pp. 1–5.

[5] MITRE | ATT&CK® for Industrial Control Systems, 2021, [Online] https://attack.mitre.org/matrices/ics/, last accessed Jan. 2023.

[6] "IEC60050 Vocabulaire électrotechnique international - Partie 351 : Technologie de commande et de régulation," Industrial Electrotechnical Commission, Technical Specification, 2014.

[7] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," in *Reliability Engineering & System Safety*, vol. 139, 2015.

[8] E. B. et al., "Specification-based monitoring of cyber-physical systems : A survey on theory, tools and applications," in *Springer International Publishing AG, Lectures on Runtime Verification*, vol. 10457, 2018.

[9] N. Goldenberg and A. Wool, "Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems," in *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, 2013.

[10] M. Caselli, E. Zambon, and F. Kargl, "Sequence-aware Intrusion Detection in Industrial Control Systems," in *Proc. ACM Workshop CPSS*, 2015.

[11] B. Ferling, J. Chromik, M. Caselli, and A. Remke, "Intrusion detection for sequence-based attacks with reduced traffic models," in *Measurement, Modelling and Evaluation of Computing Systems*. Springer International Publishing, 2018.

[12] O. Koucham, S. Mocanu, G. Hiet, J.-M. Thiriet, and F. Majorczyk, "Efficient Mining of Temporal Safety Properties for Intrusion Detection in Industrial Control Systems," in *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, 2018.

[13] Nai Fovino, I. and Coletta, A. and Carcano, A, and Masera, M., "Critical State-Based Filtering System for Securing SCADA Network Protocols," in *IEEE Transactions on Industrial Electronics*, 2012.

[14] F. Sicard, É. Zamaï, and J.-M. Flaus, "An approach based on behavioral models and critical states distance notion for improving cybersecurity of industrial control systems," in *Reliability Engineering and System Safety*, vol. 188. Elsevier, 2019.