Vers un jumeau numérique pour la sécurité des systèmes d'informations

Manuel Poisson
Amossys, CentraleSupélec, CNRS, Inria, IRISA
manuel.poisson@irisa.fr

Sébastien Kilian
CentraleSupélec, CNRS, Inria, IRISA
sebastien.kilian@centralesupelec.fr

Valérie Viet Triem Tong
CentraleSupélec, CNRS, Inria, IRISA
Valerie.VietTriemTong@centralesupelec.fr

Gilles Guette
IMT Atlantique
gilles.guette@imt-atlantique.fr

Frédéric Guihéry
Amossys
fguihery@amossys.fr

Damien Crémilleux
Amossys
dcremilleux@amossys.fr

***Résumé*—Les systèmes d'information (SI) doivent être protégés contre des menaces croissantes, et l'évaluation de leur niveau de sécurité reste un défi crucial. Ce travail propose un processus de création d'un jumeau numérique, un environnement virtuel reproduisant les chemins d'attaque possibles d'un SI analysé, comme outil pour tester la sécurité sans perturber le système en production. En premier lieu, un modèle de données relationnel structuré est présenté pour capturer les caractéristiques critiques du SI. À partir de ces données, des graphes d'attaque sont générés pour modéliser les chemins potentiels qu'un attaquant pourrait exploiter. Ces graphes permettent de sélectionner uniquement les données pertinentes pour le déploiement du jumeau numérique. Enfin, un agent d'attaque est intégré au jumeau pour valider ces chemins et simuler des scénarios d'attaque dynamiques.**

***Mots clé*—sécurité, jumeau numérique, systèmes d'information, modèle de données, graphe d'attaque**

I. Introduction

L'évaluation de la sécurité d'un système d'information (SI) est possible par la réalisation d'un test d'intrusion afin d'identifier les vulnérabilités de ce système comme le feraient des attaquants. Cependant, ces tests présentent un défi majeur : comment évaluer les vulnérabilités sans compromettre le système en production ? Pour éviter tout risque sur le SI, ces tests devraient de préférence être réalisés sur un jumeau numérique, identique au SI. Pour que les tests effectués sur un jumeau numérique soient pertinents, il faut pouvoir s'assurer de la similarité du jumeau par rapport à la réalité du SI. Les travaux précédents s'intéressant à la notion de jumeau numérique montrent que cette recherche de similarité peut porter sur différents aspects et être plus ou moins approfondit (Section II). Dans le contexte d'un jumeau numérique pour l'évaluation du niveau de sécurité d'un système d'information, le besoin n'est pas d'avoir une copie parfaite du SI original. Ce papier présente un moyen de s'assurer que les chemins d'attaque présents dans un jumeau numérique sont également présents dans le SI original. Ainsi, un jumeau numérique permettra de mettre en évidence des faiblesses dans le système d'information qu'il copie.

Pour atteindre cet objectif, nous proposons de suivre le processus présenté dans la Figure 1. D'abord, nous présentons un modèle de données qui permet d'avoir une vue structurée et cohérente des composants du SI pouvant être impliqués dans des chemins d'attaque (Section III). Nous détaillons ensuite comment ce modèle de données sert à générer des graphes d'attaques représentant les chemins qu'un attaquant pourrait suivre pour se propager dans le SI (Section IV). Enfin, nous proposons comment ces chemins devraient être testés dans le jumeau numérique (Section V).

II. État de l'art

Un jumeau numérique peut être défini comme "une réplique logicielle qui capture un aspect d'un système physique" [1]. Ce terme peut désigner une modélisation structurée d'un système réel (sous forme de graphe par exemple), grâce à laquelle il est possible de raisonner. Par exemple, cela peut servir à déduire des contrôles de sécurité à appliquer sur un système [2]. La suite de ce papier s'intéresse aux jumeaux numériques plus réalistes qu'une simple modélisation théorique. Nous désignons par jumeau numérique un ensemble de machines virtuelles. Sur chacune, il est possible d'interagir en se connectant, localement ou à distance, avec différents utilisateurs.

L'utilisation d'un jumeau numérique afin de tester et analyser un environnement de production a été d'abord étudiée dans le cadre d'environnements industriels et systèmes cyber-physiques tels que les SCADAs [3], [4]. D'autres études cherchent à cloner un environnement réseau [5], [6]. La similarité entre le réseau initial et son jumeau est ensuite évaluée. Les auteurs d'INSALATA [5] évaluent la similarité entre la topologie du réseau initial et la topologie du jumeau en comparant la sortie des outils `ping` et `traceroute`. Ils indiquent aussi qu'une inspection manuelle est effectuée, sans plus de détails. Dans le papier [6], l'évaluation consiste à capturer les paquets émis par un système industriel et par son jumeau numérique lors de l'exécution des mêmes actions. Cela permet de voir si les deux systèmes produisent les mêmes paquets quand ils sont soumis aux mêmes entrées. En l'occurrence, la comparaison porte sur moins de soixante paquets, ce qui est peu comparé à la quantité de données transitant sur un véritable réseau.

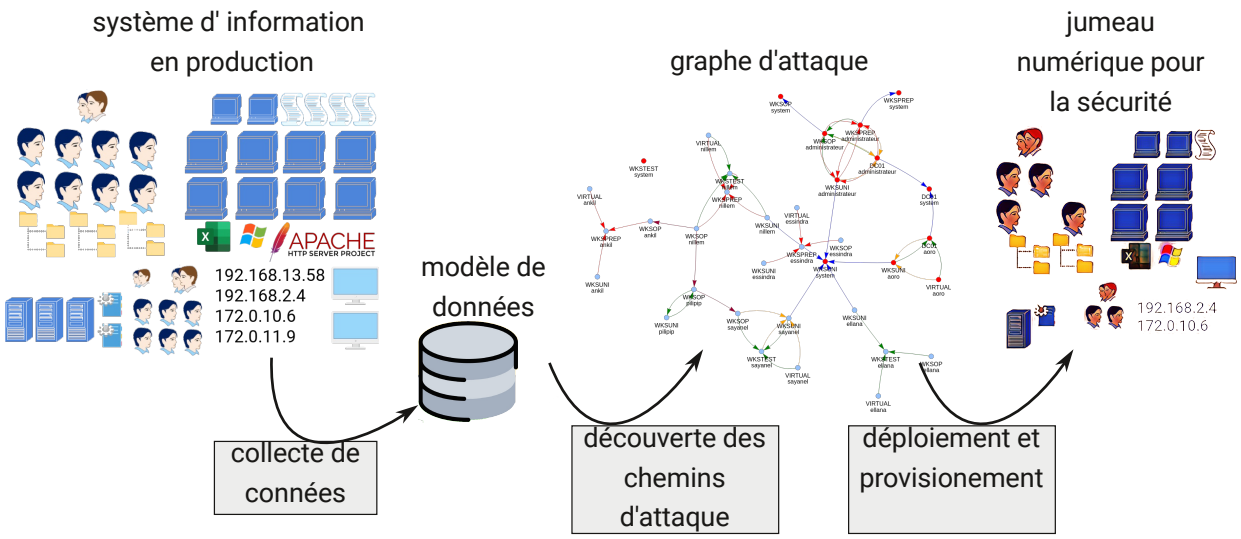

FIGURE 1. Construction d'un jumeau numérique pour la sécurité d'un SI.

Les travaux présentés dans les sections suivantes portent sur la reproduction de caractéristiques réseau et système d'un système d'information (SI) typique d'une entreprise.

III. MODÈLE DE DONNÉES

Dans le but de construire un jumeau numérique d'un SI à des fins de sécurité, il est nécessaire de collecter des informations sur le SI. Afin de pouvoir raisonner sur ces informations, elles sont ensuite organisées dans un modèle de données tel que décrit dans cette section.

Le modèle de données permet de stocker la connaissance du SI et est construit pour répondre à deux critères. Tout d'abord, les informations qu'il contient sont pertinentes pour l'identification de chemins d'attaques. Ensuite, ces données sont stockées de manière structurée pour assurer leur cohérence. Concrètement, le modèle de données prend la forme d'une base de données relationnelle PostgreSQL avec des clés étrangères qui forcent les dépendances entre certaines entrées et des contraintes d'intégrité. Cette base de données contient 33 tables représentant chacune un type de caractéristique du SI et 54 clés étrangères liant ces tables.

Parmi ces caractéristiques, nous avons choisi de représenter, entre autres, les fichiers, les machines et les utilisateurs. Il y a aussi des tables pour les groupes d'utilisateurs, les services exécutés et logiciels installés. Les attributs de ces tables spécifient, par exemple, le type et la version du système d'exploitation d'une machine. Les propriétés de cohérence et d'intégrité de la base expriment des propriétés du SI telles que *"un fichier est nécessairement lié à une machine et à un chemin qui le localise dans la machine"* ou *"pour une même machine et un même chemin, il n'y a qu'un seul fichier"*.

La connaissance de l'existence de chemins d'attaque dans le SI dépend de trois choses : la connaissance des vulnérabilités du SI, les procédures d'attaques supposées maîtrisées de l'attaquant et l'état du SI (tel que sa configuration et les logiciels qui y sont installés). Par exemple, il peut y avoir un chemin d'attaque dans le SI qui permet de faire un mouvement latéral (exécution de code sur une machine distante) en exploitant la vulnérabilité identifiée par la CVE-2024-7593 liée au logiciel Ivanti vTM version 22.3. Pour être conscient de ce chemin d'attaque dans le SI il faut à la fois savoir que le logiciel Ivanti vTM version 22.3 est exécuté sur une machine du SI et en même temps avoir connaissance de la CVE qui lui est associée et de son impact. La connaissance des vulnérabilités et des procédures d'attaque guide le choix des données à collecter dans le SI pour en déduire les chemins d'attaque. La Figure 2 représente, dans le cadre vert, un exemple d'ensemble de caractéristiques d'un SI. On fait l'hypothèse que savoir que l'image `landscape.png` est utilisée pour le fond d'écran ne permet pas de déduire un chemin d'attaque. Par conséquent, cette information est exclue des données présentes dans le modèle de données. Inversement, on suppose que les services, les fichiers qu'ils exécutent, les droits sur ces fichiers et les membres des groupes d'utilisateurs peuvent être impliqués dans la découverte de chemins d'attaque. Ces informations sont donc collectées et stockées dans le modèle de données.

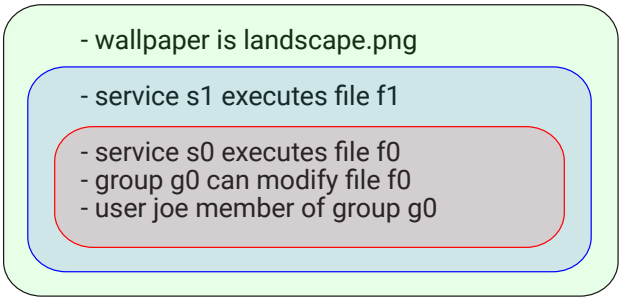

FIGURE 2. Différentes granularités des données dans le SI et le jumeau numérique. vert : données dans le SI bleu : données dans le modèle de données, rouge : données dans le jumeau numérique

IV. Construction du graphe d'attaque

Les chemins d'attaques peuvent être représentés dans un graphe dont un exemple est présenté dans la Figure 1. Les nœuds du graphe représentent des positions d'attaque définie comme des couples (m, u) indiquant que l'utilisateur u peut exécuter des commandes sur la machine m. Les arcs représentent la possibilité d'exécuter une procédure d'attaque permettant un mouvement latéral ou vertical, associé respectivement à la prise de contrôle d'une nouvelle machine ou d'un nouvel utilisateur. Un chemin d'attaque est un chemin partant d'une position d'attaque supposée initiale vers une position d'attaque dite finale dans un tel graphe comme proposé dans [7]–[9]. La Figure 3 illustre un mouvement vertical allant de la position d'attaque `uni,joe` à `uni,system` en exploitant une procédure d'attaque étiquetée `serviceExeModif`.

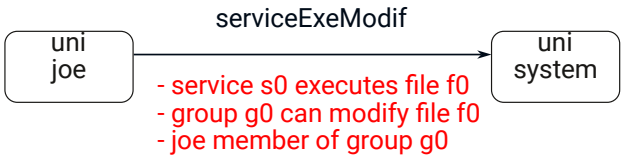

Figure 3. Exemple de chemin avec 1 procédure d'attaque (mouvement vertical).

Toute procédure d'attaque ne peut être exécutée que si un ensemble de préconditions est vérifié. Par exemple, `serviceExeModif` est possible si un fichier exécuté par un service peut être modifié par un utilisateur. Construire le graphe d'attaque revient à vérifier si les données collectées dans le SI en production vérifient ces conditions. Les arcs du graphe indiquent non seulement quelles procédures d'attaque peuvent être exécutées, mais aussi pour quelles raisons elles peuvent l'être. Dans l'exemple Figure 2, les données collectées dans le SI et stockées dans le modèle de données (en bleu) indiquent que le service `s0` exécute le fichier `f0` qui peut être modifié par les membres du groupe `g0` dont fait partie l'utilisateur `Joe`. Ces données sont associées à l'arc `serviceExeModif` sur la Figure 3 car elles constituent les préconditions qui justifient la possibilité d'exécuter cette procédure d'attaque. Certaines données collectées ne sont finalement impliquées dans aucun chemin d'attaque, étant donné l'état actuel de la connaissance des vulnérabilités. C'est le cas du service `s1` dans l'exemple de la Figure 2. Avec l'objectif de reproduire dans le jumeau uniquement les chemins d'attaque présents dans le SI, ce sont uniquement les données impliquées dans un chemin d'attaque (en rouge sur la Figure 2) qui sont répliquées lors du déploiement du jumeau numérique.

V. Jumeau numérique et exploitation des chemins d'attaque

La création du jumeau numérique passe par le déploiement et la configuration de machines virtuelles (VMs) de façon à produire un environnement présentant les caractéristiques identifiées dans le graphe d'attaque.

Afin de vérifier la présence de chemins dans le jumeau et de vérifier la faisabilité de ces chemins, un agent d'attaque est utilisé. Grâce aux procédures qu'il intègre, l'agent d'attaque peut récupérer de lui-même des informations sur le système et, si les conditions sont remplies, effectuer l'exploit pour gagner des connaissances sur le système ou se propager.

Les informations que possède l'agent d'attaque sont structurées sous forme de graphe de connaissances, encapsulant diverses stratégies d'attaque, les vulnérabilités du système et leurs interrelations. Dans ce cadre, les exploits sont formalisés par des règles d'inférence sur le graphe de connaissances. L'agent peut ainsi générer dynamiquement des séquences d'attaques en fonction de la configuration du système et des vulnérabilités détectées. Par exemple sur la Figure 3, si l'agent d'attaque est présent sur la machine `uni,Joe`, une fois qu'il a dans son graphe de connaissances les informations sur les préconditions de l'arc `serviceExeModif`, il peut exécuter la procédure d'attaque et se déplacer vers la machine `uni,system`.

VI. Avancement, travaux futurs et conclusion

La construction d'un jumeau numérique d'un SI est un travail en cours. À ce jour, nous avons mis en place un SI relativement simple, avec trois machines virtuelles, dans lequel il est possible d'exécuter quelques chemins d'attaque. Connaissant exactement les caractéristiques de ce SI, nous les avons reportées dans un fichier sous forme de texte. Cela s'assimile à une collecte de données manuelle en interrogeant, par exemple, un administrateur du système. Un script lit ensuite ce fichier texte et peuple automatiquement notre modèle de données en conséquence. Puis, nous déployons, à la main pour le moment, trois VMs vierges, avec les mêmes systèmes d'exploitation que les machines du SI original. Nous générons automatiquement des playbooks Ansible pour configurer ces VMs suivant les informations contenues dans le modèle de données. Après l'exécution automatique des playbooks, nous obtenons un jumeau numérique dans lequel nous avons pu vérifier, manuellement, qu'il reproduit effectivement un des chemins d'attaque du SI original.

Nous sommes donc capable de produire un jumeau numérique d'un SI qui contient des chemins d'attaque présents dans le SI original. Cependant, certaines étapes du processus sont encore manuelles et en cours d'automatisation. En particulier, cela concerne la collecte des données du SI original et le déploiement des VMs. La découverte des chemins d'attaque basée sur le modèle de données n'est pas complètement aboutie. Il reste notamment à bien définir les données impliquées pour chaque procédure d'attaque.

Références

[1] M. Grieves, "Virtually intelligent product systems : Digital and physical twins," in Complex Systems Engineering : Theory and Practice, 07 2019, pp. 175–200.

[2] E. Hadar, D. Kravchenko, and A. Basovskiy, "Cyber Digital Twin Simulator for Automatic Gathering and Prioritization of Security Controls' Requirements," in 2020 IEEE 28th International Requirements Engineering Conference (RE), Aug. 2020, pp. 250–259, iSSN : 2332-6441. [Online]. Available : https ://ieeexplore.ieee.org/document/9218140

[3] M. Dietz, M. Vielberth, and G. Pernul, "Integrating digital twin security simulations in the security operations center," in Proceedings of the 15th International Conference on Availability, Reliability and Security, ser. ARES '20. Association for Computing Machinery, pp. 1–9. [Online]. Available : https ://dl.acm.org/doi/10.1145/3407023.3407039

[4] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in CPS-based production systems," vol. 11, pp. 939–948. [Online]. Available : https ://www.sciencedirect.com/science/article/pii/S2351978917304067

[5] N. Herold, M. Wachs, M. Dorfhuber, C. Rudolf, S. Liebald, and G. Carle, "Achieving reproducible network environments with INSALATA," in Security of Networks and Services in an All-Connected World, D. Tuncer, R. Koch, R. Badonnel, and B. Stiller, Eds. Springer International Publishing, pp. 30–44.

[6] M. Eckhart and A. Ekelhart, "Towards security-aware virtual environments for digital twins," in Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, ser. CPSS '18. Association for Computing Machinery, pp. 61–72. [Online]. Available : https ://dl.acm.org/doi/10.1145/3198458.3198464

[7] M. Poisson, V. Viet Triem Tong, G. Guette, E. Abgrall, F. Guihéry, and D. Crémilleux, "Unveiling stealth attack paths in Windows Environments using AWARE," in CSNet 2023, Montreal, Canada.

[8] B. Pierre-Victor, V. T. T. Valérie, A. Erwan, G. Gilles, and P. Guillaume, "Formalisation de scénario d'attaque adaptée au déploiement d'architectures volontairement vulnérables," in RESSI, Chambon-sur-Lac, France, May 2022.

[9] A. Berady, M. Jaume, V. Viet Triem Tong, and G. Guette, "PWNJUTSU : A dataset and a semantics-driven approach to retrace attack campaigns," IEEE Transactions on Network and Service Management, pp. 1–13, 2022. [Online]. Available : https ://hal.inria.fr/hal-03694719