Filtrage et vérification de flux métiers dans les systèmes industriels

Maxime Puys, Marie-Laure Potet et Jean-Louis Roch

Univ. Grenoble Alpes, VERIMAG, F-38000 Grenoble, France
CNRS, VERIMAG, F-38000 Grenoble, France
Prénom.Nom@imag.fr *

Résumé

De plus en plus d'attaques informatiques contre les systèmes industriels sont présentées par les médias. Ces systèmes tendent à devenir géographiquement distribués et à communiquer via des réseaux vulnérables tels qu'Internet. Régissant de nos jours des domaines tels que la production et la distribution d'énergie, l'assainissement des eaux ou le nucléaire, la sécurité des systèmes industriels devient une priorité pour les gouvernements. L'une des difficultés de la sécurisation des infrastructures industrielles est la conciliation des propriétés de sécurité avec les attendus métiers en terme de flux. Pour ce faire, nous regardons comment filtrer les messages en tenant compte des aspects métiers. Ensuite, nous nous intéressons à la vérification formelle des propriétés des protocoles de communication industriels. Enfin nous proposons une approche *Model-Based Testing* permettant de générer des attaques informatiques contre des systèmes industriels.

1 Introduction

De plus en plus d'attaques informatiques contre les systèmes industriels sont présentées par les médias. Ces systèmes tendent à devenir géographiquement distribués et à communiquer via des réseaux vulnérables tels qu'Internet. Depuis son apparition, l'informatique industrielle a toujours été physiquement isolée du reste du monde. Les attaques nécessitaient d'être présent sur le site et étaient peu probables. L'industrie a donc préféré se focaliser sur la protection contre les risques naturels et les erreurs de manipulation, appelée *sûreté* [20, 21, 15]. La particularité de la sécurité informatique face à la sûreté est la volonté de

*Ce travail a été partiellement financé par le LabEx PERSYVAL-Lab (ANR–11-LABX-0025) et le projet Programme Investissement d'Avenir FSN AAP Sécurité Numérique n° 3 ARAMIS (P3342-146798).

nuire de l'attaquant, doublée de sa capacité à réfléchir et à apprendre de ses erreurs, suivant ainsi l'état de l'art de la sécurité informatique. Aujourd'hui, ces systèmes régissent aussi bien la production et la distribution d'énergie, l'assainissement des eaux ou le nucléaire. Il apparaît que leur protection contre les attaques informatiques devient une priorité des agences gouvernementales avec notamment la sécurisation des opérateurs d'importance vitale [8], détaillée dans le Livre blanc sur la défense et la sécurité nationale (édition de 2008 [11] puis de 2013 [12]).

L'une des difficultés de la sécurisation des infrastructures industrielles est la conciliation des propriétés de sécurité avec les attendus métiers en terme de flux, parfois en adéquation mais aussi parfois antagonistes [20]. Pour ce faire, la section 2 détaille trois axes : (i) le filtrage des messages tenant compte des aspects métiers, (ii) la vérification formelle des propriétés des protocoles de communication industriels, (iii) un approche *Model-Based Testing* pour décrire les attaques informatiques contre les systèmes industriels en fonction de paramètres tels que les objectifs, les capacités des attaquants et leur position dans une architecture réseau. Enfin la section 3 conclut.

2 Contributions

Cette section propose trois axes pour la sécurisation des infrastructures industrielles, s'intéressant à différents niveaux d'abstraction de sécurité et qui devraient être intégrés dans une approche de bout en bout.

2.1 Filtrage des communications

Un premier axe de notre travail se déroule dans le cadre du projet ARAMIS [2] qui vise à proposer un dispositif permettant de cloisonner physiquement les réseaux et de filtrer les échanges pour rejeter tout flux identifié comme non autorisé, donc potentiellement malveillant. Cette fonctionnalité distingue les filtres [23, 13, 1, 22] des systèmes de détection d'intrusion [14, 19, 18, 24]. La spécificité de ce projet est de fournir simultanément ces deux fonctionnalités dans un même dispositif robuste, dont l'ajout n'impacte pas le système existant. La majeure partie des communications se résument à des requêtes envoyées par des clients vers des systèmes SCADA (système de contrôle et d'acquisition de données à grande échelle). Cependant, les protocoles plus récents proposent des services plus complexes tels que la découverte interactive des composants du serveur ou la demande d'historique des commandes. Afin de palier les attaques portant sur la couche réseaux, tous les messages sont interprétés puis transformés dans une langage intermédiaire. Cette conversion des protocoles de communication en un format commun permet d'effectuer un filtrage applicatif en prenant en compte les besoins liés au corps de métier du système industriel à protéger. Ainsi, ce filtrage peut prendre plusieurs formes incluant :

— La vérification des identités des clients et serveurs communiquant entre eux,

— La vérification des permissions d'accès des clients aux différentes variables des serveurs,
— La vérification des contenus des messages eux-mêmes avec la possibilité de garder en mémoire l'état de certaines variables du serveur afin détecter des messages rendu illicites par le contexte des messages précédents. Ce type de filtrage peut par exemple empêcher l'ouverture d'un disjoncteur déjà ouvert.

Comme tout filtrage local, il ne garantit qu'une décision dépendant de sa connaissance de l'état du système. Enfin, les systèmes industriels acceptant une latence maximale de l'ordre du centième de seconde, le filtrage doit être pensé pour une efficacité maximale. Cela passe par l'utilisation de structures de données possédant une faible complexité d'accès ou de recherche tels que des tables de hashage ou des B-Arbres [4]. De plus, des analyses de pire temps d'exécution devraient assurer le respect des contraintes temporelles.

2.2 Vérification formelle de protocoles industriels

Il apparaît depuis plusieurs décennies que la preuve formelle de la sécurité des protocoles de communication est un enjeu majeur. Un exemple connu est *miTLS* [5], une implémentation prouvée du protocole TLS [9] servant par exemple au paiement en ligne. Il existe plusieurs outils permettant de modéliser des protocoles afin d'en tester la sécurité tels que : AVISPA [3], Tamarin [17], Scyther [7], ProVerif [6]. Ces outils considèrent un intrus dit de Dolev-Yao [10] qui contrôle le réseau, espionne, stoppe, forge, modifie, entrelace ou rejoue des messages en utilisant la connaissance des messages qu'il a appris précédemment. Il est alors possible de spécifier des propriétés telles que l'*intégrité d'origine* (le plus souvent assurée par des mécanismes d'authentification) et la *confidentialité*. La première propriété signifie qu'un participant est convaincu qu'il parle avec un participant bien identifié. La confidentialité assure qu'un agent (y compris l'intrus) n'accède qu'aux messages qui lui sont destinés. Enfin, les outils se basent sur l'hypothèse du chiffrement parfait, selon laquelle il n'est pas possible de déchiffrer un message sans la chef de chiffrement ou d'usurper une signature.

Les protocoles de communication des systèmes industriels tels que OPC-UA [16] (futur standard des communications industrielles) n'ont jamais été vérifiés de cette façon. Nous proposons donc de tester formellement la sécurité de ces protocoles et notamment les sous-protocoles *OpenSecureChannel* et *CreateSession*, intervenant dans le *handshake* d'OPC-UA, à l'aide de l'outil ProVerif. Enfin, la difficulté d'appliquer ce type d'approche aux protocoles industriels repose sur le fait qu'ils s'intéressent plus à la *disponibilité* et à l'*intégrité des messages* qu'à l'*intégrité d'origine* et la *confidentialité*. L'expression de ces propriétés dans les langages des outils est donc un défi important.

2.3 Modèles d'attaques

Enfin, nous nous intéressons à la mise en place d'un modèle générique permettant de produire des scénarios d'attaques informatiques contre les systèmes

industriels. Cette phase d'analyse vise à être incluse dans une approche globale allant de la modélisation du système à la production automatique des paquets réseau implémentant et testant les attaques identifiées. Cette approche *Model-Based Testing* aura pour objectif de vérifier si les attaques trouvées par l'analyse de la modélisation sont effectivement jouables sur une plate-forme, voir de quantifier leur plausibilité. La figure 1 illustre la méthodologie que nous voulons développer.

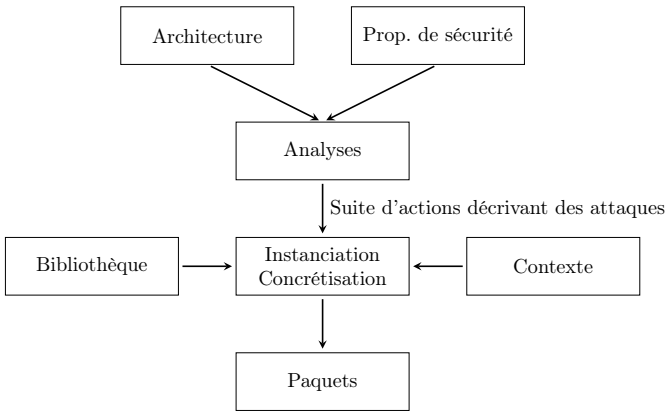

FIGURE 1 – Méthodologie globale

En partant d'une modélisation de l'infrastructure du système industriel et des protections pouvant être présentes sur celui-ci (ex. : des protocoles de communication sécurisés), nous sommes en mesure de produire des suites d'actions à réaliser par un attaquant pour atteindre un objectif. Ces actions sont ensuite concrétisées en paquets réseaux à l'aide d'une bibliothèque décrivant comment implémenter les vecteurs pour chaque protocole (ex. : comment modifier un paquet OPC-UA, ou comment outrepasser l'authentification d'un serveur SCADA). Enfin, ces paquets sont instanciés, soit de manière aléatoire, soit en fonction de la logique applicative de la plate-forme.

3 Conclusion

En conclusion, cet article explique le filtrage au sein du dispositif ARAMIS. Il aborde ensuite l'application d'outils de vérification de protocoles cryptographiques aux protocoles de communication industriels. Enfin il propose une approche *Model-Based Testing* permettant de jouer des attaques obtenues à l'aide de l'analyse d'une modélisation du système. Ces trois axes, présentés dans la figure 2 sont complémentaires. En effet, l'approche modèles d'attaques est une approche globale dépendant de différents paramètres incluant notamment les propriétés de sécurité offertes par les protocoles de communication. Or cette approche ne peut être pertinente que si ces propriétés sont prouvées. Enfin, le filtre ARAMIS étant développé dans le cadre d'un projet industriel, il est alors

possible de tester sa capacité à bloquer des attaques à l'aide de notre approche de modélisation.

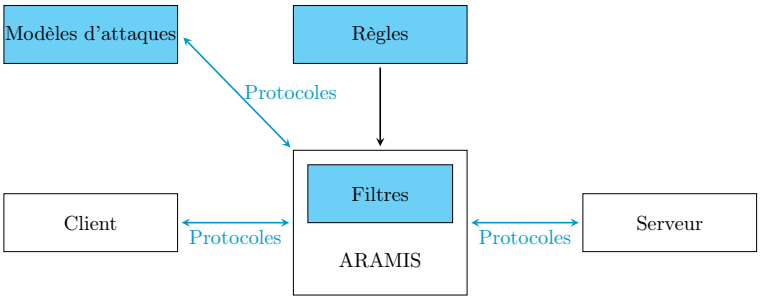

FIGURE 2 – Articulation des axes de recherche

Références

[1] Deny ALL : rweb4. Agence nationale de la sécurité des systèmes d'information, juin 2013.

[2] Projet ARAMIS : Architecture robuste pour les automates et matériels des infrastructures sensibles (2014-2017). http ://aramis.minalogic.net/.

[3] A. ARMANDO, D. BASIN, Y. BOICHUT, Y. CHEVALIER, L. COMPAGNA, J. CUELLAR, P. H. DRIELSMA, P.-C. HEÁM, O. KOUCHNARENKO, J. MANTOVANI, S. MÖDERSHEIM, D. von OHEIMB, Michael R., J. SANTIAGO, M. TURUANI, L. VIGANÒ et L. VIGNERON : The AVISPA tool for the automated validation of internet security protocols and applications. *In Proc. of CAV'2005*, LNCS 3576, pages 281–285. 2005.

[4] Rudolf BAYER : Binary b-trees for virtual memory. *In Proceedings of the 1971 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control*, pages 219–235. ACM, 1971.

[5] K. BHARGAVAN, C. FOURNET, M. KOHLWEISS, A. PIRONTI et P. STRUB : Implementing tls with verified cryptographic security. *In Security and Privacy (SP), 2013 IEEE Symposium on*, pages 445–459, mai 2013.

[6] Bruno BLANCHET : An efficient cryptographic protocol verifier based on prolog rules. *In Proceedings of the 14th IEEE Workshop on Computer Security Foundations*, CSFW '01, pages 82–, Washington, DC, USA, 2001. IEEE Computer Society.

[7] C.J.F. CREMERS : The Scyther Tool : Verification, falsification, and analysis of security protocols. *In Computer Aided Verification, 20th International Conference, CAV*, volume 5123/2008 de *LNCS*, pages 414–418. Springer, 2008.

[8] Code de la DÉFENSE : Article r1332-2 - protection des installations d'importance vitale, avril 2007.

[9] T. Dierks et E. Rescorla : The transport layer security (tls) protocol, version 1.2. IETF RFC 5246, août 2008.

[10] D. Dolev et Andrew C. Yao : On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, mars 1981.

[11] France. Commission du Livre blanc sur la défense et la sécurité nationale : *Livre blanc (Le)*. Odile Jacob, 2008.

[12] France. Commission du Livre blanc sur la défense et la sécurité nationale : *Livre blanc (Le)*. Documentation française, 2013.

[13] STONESOFT France : Fonctionnalités de pare-feu de stonegate firewall/vpn 5.2.4 build 8069. Agence nationale de la sécurité des systèmes d'information, décembre 2011.

[14] STONESOFT France : Ips stonegate - version 5.4.1. Agence nationale de la sécurité des systèmes d'information, février 2013.

[15] IAEA International Nuclear Safety Group INSAG : Insag-10 defence in depth in nuclear safety. Rapport technique, Report STI/PUB/1013, 1996.

[16] Wolfgang Mahnke, Stefan-Helmut Leitner et Matthias Damm : *OPC unified architecture*. Springer Science & Business Media, 2009.

[17] Simon Meier, Benedikt Schmidt, Cas Cremers et David Basin : The tamarin prover for the symbolic analysis of security protocols. *In* Natasha Sharygina et Helmut Veith, éditeurs : *Computer Aided Verification*, volume 8044 de *Lecture Notes in Computer Science*, pages 696–701. Springer Berlin Heidelberg, 2013.

[18] OISF : Suricata : Open source ids / ips / nsm engine. http ://suricata-ids.org/, avril 2016.

[19] Vern Paxson : Bro : a system for detecting network intruders in real-time. *Computer networks*, 31(23):2435–2463, 1999.

[20] Ludovic Pièvre-Cambacédès : *The relationships between safety and security*. Theses, Télécom ParisTech, novembre 2010.

[21] Ludovic Pièvre-Cambacédès et Pascal Sitbon : An analysis of two new directions in control system perimeter security. *In Proceedings of the 3rd SCADA Security Scientific Symposium (S4)*, pages 4.1–4.30, Miami, USA, janvier 2009.

[22] SECLAB-FR : Dz-network. Agence nationale de la sécurité des systèmes d'information, juin 2014.

[23] EDF R&D Département SINETICS : Dispositif d'échange sécurisé d'informations sans interconnexion réseau (desiir) v1.0. Agence nationale de la sécurité des systèmes d'information, avril 2010.

[24] Snort Team : Snort : Open source network intrusion prevention system. https ://www.snort.org, avril 2016.