Génération systématique de scénarios d'attaques contre des systèmes industriels

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

VERIMAG, Univ. Grenoble Alpes / Grenoble-INP, France
prénom.nom@imag.fr ⋆

Résumé Les systèmes industriels (SCADA) sont la cible d'attaques informatiques depuis Stuxnet [4] en 2010. De part leur interaction avec le mode physique, leur protection est devenue une priorité pour les agences gouvernementales. Dans cet article, nous proposons une approche de modélisation d'attaquants dans un système industriel incluant la production automatique de scénarios d'attaques. Cette approche se focalise sur les capacités de l'attaquant et ses objectifs en fonction des protocoles de communication auxquels il fait face. La description de l'approche est illustrée à l'aide d'un exemple.

1 Introduction

Les systèmes industriels aussi appelés SCADA (Supervisory Control And Data Acquisition) sont la cible de nombreuses attaques informatiques depuis Stuxnet [4] in 2010. De part leur interaction avec le monde physique, ces systèmes peuvent représenter une réelle menace pour leur environnement. Suite à une récente augmentation de la fréquence des attaques, la protection des installations industrielles est devenue une priorité des agences gouvernementales. Ces systèmes diffèrent également de l'informatique de gestion du fait de leur très longue durée de vie, de leur difficulté à appliquer des correctifs et des protocoles souvent spécifiques utilisés.

État de l'art : La modélisation et la génération de scénarios d'attaques sont essentielles à la sécurité des systèmes industriels. En 2013, la norme IEC 62443-3-3 [2] détaille de façon très précise une méthode d'analyse de la sécurité informatique des installations industrielles. En 2015, Conchon *et al.* [1] proposent une approche se basant sur EBIOS. Toujours en 2015, Kriaa *et al.* [3] décrivent S-CUBE, une approche de modélisation de systèmes industriels faisant le pas entre la sécurité et la sûreté de fonctionnement. Leur article analyse un grand nombre d'approches de production de scénarios d'attaques et conclut notamment qu'aucune n'est automatisée ni facilement automatisable.

⋆. 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).

Contributions : Nous proposons dans cet article une approche de modélisation d'attaquants basée à la fois sur l'infrastructure et les possibilités d'attaques dans un système industriel. Cette approche se base sur les composants d'une infrastructure et les canaux de communication entre ses composants [6]. Elle vise la production automatique de scénarios d'attaques. Nous nous focalisons sur l'attaquant en modélisant sa position dans l'infrastructure, ses objectifs d'attaques et les protocoles qu'il peut utiliser en tenant compte de leurs propriétés de sécurité. Enfin, à l'aide d'une approche systématique, nous générons l'ensemble des scénarios d'attaques pour lesquels un attaquant est capable de réaliser l'un de ses objectifs face à un protocole donné.

Plan : La section 2 détaille la modélisation des attaquants et la production des scénarios d'attaques à l'aide d'un un exemple. Ensuite, la section 3 décrit comment nous souhaitons inclure cette approche dans une approche *Model-Based Testing* plus globale. Enfin la section 4 conclut.

2 Approche de génération des scénarios d'attaques

Dans cette section, nous détaillons le formalisme utilisé dans notre modélisation des attaquants et comment exploiter cette modélisation pour générer des scénarios d'attaques. Cette approche est double. Elle propose dans un premier temps d'étudier les objectifs des attaquants et comment ils pourraient les réaliser (approche descendante) avant de les mettre dans un second temps face aux protections apportées par les protocoles de communication (approche ascendante).

2.1 Approche descendante

Nous commençons par définir l'ensemble des attaquants $\mathcal{A}$ et l'ensemble des objectifs d'attaques $\mathcal{O}$. Ces ensembles sont liés par la relation $\mathcal{R}_{Obj}$ entre un attaquant a et un objectif o tel que $\mathcal{R}_{Obj} \subseteq \mathcal{A} \times \mathcal{O}$ si a cherche à atteindre o fait partie de notre analyse de risque. Il va donc de soit que cette relation, fournie par le concepteur du modèle doit tenir compte de la position des attaquants dans l'architecture globale du système (ex. : un attaquant ne peut pas avoir pour objectif la modification d'un message s'il n'y a jamais accès).

——— Exemple : ———

Nous considérons l'infrastructure de communication en figure 1 où les attaquants (en couleur) sont $\mathcal{A} = \{Client_A, Routeur_A\}$ (un client et un routeur compromis) et les objectifs d'attaques (tirés de recommandations gouvernementales) considérés, sont $\mathcal{O} = \{VolId, ContAuth, Alte, Alte_C\}$ avec :

— $VolId$ = Vol d'identifiants,

— $ContAuth$ = Contournement d'authentification,
— $Alte$ = Altération d'un message,
— $Alte_C$ = Altération ciblée d'un message.

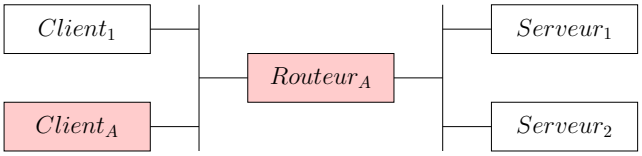

FIGURE 1: Exemple d'infrastructure

Les attaquants $\mathcal{A}$ et les objectifs $\mathcal{O}$ sont liés par la relation $\mathcal{R}_{Obj}$ définie en Table 1, où un ✓ signifie que l'objectif o est retenu pour l'attaquant a.

$\mathcal{R}_{Obj}$	$VolId$	$ContAuth$	$Alte$	$Alte_C$
$Client_A$		✓		
$Routeur_A$	✓	✓	✓	✓

TABLE 1: Exemple d'objectifs retenus pour chaque attaquant

Nous définissons ensuite l'ensemble des vecteurs d'attaques $\mathcal{V}$ et $Real \subseteq \mathcal{O} \times \mathbb{P}(\mathcal{V})$, la fonction entre un objectif et l'ensemble des parties de $\mathcal{V}$ (ie. les combinaisons de vecteurs d'attaques). Ainsi, $Real(o)$ décrit la réalisation d'un objectif o à l'aide de vecteurs de $\mathcal{V}$. Un vecteur peut être vu comme les techniques, tactiques ou procédés pouvant servir à la réalisation d'une attaque [1].

Exemple :

Nous considérons ici les vecteurs $\mathcal{V} = \{Lire, Usurp, Mod, Rej\}$ avec :
— $Lire$ = Lecture (et compréhension) d'un message
— $Usurp$ = Usurpation d'une identité
— Mod = Modification d'un message
— Rej = Rejeu d'un message

La fonction $Real$ décrivant comment réaliser les objectifs d'attaque à l'aide des vecteurs d'attaques peut par exemple être :
— $Real(VolId) = \{\{Lire\}\}$
— $Real(ContAuth) = \{\{Usurp\}, \{Rej\}\}$
— $Real(Alte) = \{\{Mod\}\}$
— $Real(Alte_C) = \{\{Lire, Mod\}\}$

En particulier, $ContAuth$ peut être réalisé en usurpant une identité **ou** en rejouant un message d'authentification (ex. : l'envoi d'un mot de passe). Tandis que $Alte_C$ nécessite **à la fois** la capacité de modifier un message et d'en comprendre le contenu.

Ainsi, nous sommes en mesure de représenter les attaquants, leurs objectifs et comment ces objectifs peuvent être réalisés au moyen de vecteurs d'attaques. La section suivante décrit comment une analyse des propriétés de sécurité offertes par les protocoles vérifie si ces objectifs sont atteignables.

2.2 Approche ascendante

Dans un second temps, nous définissons l'ensemble des configurations des protocoles $\mathcal{P}$ considérés pour l'analyse. Dans la suite de cet article nous considérerons chaque configuration comme un protocole différent. $Vect \subseteq \mathcal{P} \times \mathbb{P}(\mathcal{V})$ est l'ensemble des vecteurs d'attaques de $\mathcal{V}$ accessibles à l'attaquant pour chaque protocole (ie. leurs faiblesses exploitables).

Exemple :

Pour les protocoles $\mathcal{C} = \{$MODBUS, FTP, FTP$_{Auth}$, OPC-UA$_{None}$, OPC-UA$_{Sign}$, OPC-UA$_{SignEnc}$ $\}$, les capacités d'attaques $Vect$ sont définies en table 2, où un ✓ signifie que le vecteur d'attaque est accessible à l'attaquant pour ce protocole.

$Vect$	$Lire$	$Usurp$	Mod	Rej
MODBUS	✓	✓	✓	✓
FTP	✓	✓	✓	✓
FTP$_{Auth}$	✓		✓	✓
OPC-UA$_{None}$	✓	✓	✓	✓
OPC-UA$_{Sign}$	✓			
OPC-UA$_{SignEnc}$				

TABLE 2: Exemple de capacités d'attaques retenus pour chaque protocole

Les protocoles MODBUS, FTP et OPC-UA$_{None}$ ne garantissent aucune sécurité et permettent donc tous les vecteurs d'attaques. Le protocole FTP$_{Auth}$ ajoute une authentification à l'aide d'un mot de passe empêchant l'usurpation d'identité. Les protocoles OPC-UA$_{Sign}$ et OPC-UA$_{SignEnc}$ apportent des signatures cryptographiques et de l'estampillage aux messages, empêchant ainsi leur usurpation, modification ou rejeu. Enfin OPC-UA$_{SignEnc}$ garantit également la confidentialité des communications.

Il est alors possible de déterminer si un objectif o est réalisable à l'aide de l'ensemble des vecteurs d'attaques pour un protocole p en vérifiant si : $\exists e \in \mathcal{R}_{Real}(o) \mid e \subseteq Vect(p)$. Alors, l'ensemble des scénarios d'attaques $\mathcal{S}_{a,p}$ pour un attaquant a et un protocole p est alors défini par :

$$\mathcal{S}_{a,p} = \{(o, e) \mid o \in \mathcal{O} \wedge e \subseteq \mathcal{R}_{Real}(o) \wedge e \subseteq Vect(p) \wedge (a, o) \in \mathcal{R}_{Obj}\}$$

L'ensemble des scénarios d'attaques à considérer dans le cadre d'une campagne de test est alors l'ensemble des $\mathcal{S}_{a,p}, \forall a \in \mathcal{A}$ et pour tous les protocoles de $\mathcal{P}$ que pourraient utiliser chaque attaquants.

Exemple :

Dans notre exemple, si l'on suppose que $Client_A$ communique via FTP$_{Auth}$:

$$\mathcal{S}_{Client_A,\text{FTP}_{Auth}} = \{(ContAuth, Rej)\}$$

Cela s'explique par le fait que seul l'objectif $ContAuth$ est considéré pour $Client_A$ (table 1) et qu'il est réalisable avec au moins l'un des vecteurs d'attaques $Usurp$ ou Rej dont le dernier est offert par FTP$_{Auth}$. De même pour $Routeur_A$ qui est face à la fois à OPC-UA$_{None}$ et FTP$_{Auth}$:

$$\mathcal{S}_{Routeur_A,\text{OPC-UA}_{None}} = \{(VolId, Lire), (ContAuth, Usurp), (ContAuth, Rej), (Alte, Mod), (Alte_C, (Lire, Mod))\}$$

$$\mathcal{S}_{Routeur_A,\text{FTP}_{Auth}} = \{(VolId, Lire), (ContAuth, Rej), (Alte, Mod), (Alte_C, (Lire, Mod))\}$$

3 Méthodologie globale

Notre objectif est d'utiliser cette phase d'analyse dans une approche globale allant de la modélisation du système à la production automatique des paquets réseau implémentant et testant les attaques identifiées. Nous proposons donc une approche *Model-Based Testing* dans l'objectif de vérifier si les attaques trouvées par l'approche sont effectivement jouables sur une plate-forme, voire de quantifier leur plausibilité. La figure 2 illustre la méthodologie que nous voulons développer. L'approche part d'une architecture représentant les composants du systèmes, les canaux de communication et les protocoles (similaire à la figure 1) ; croisée avec des propriétés de sécurité spécifiant les objectifs des attaquants. Ensuite, l'analyse présentée en section 2 permet d'obtenir les vecteurs d'attaques à utiliser par des attaquants pour violer les propriétés de sécurité. Ces vecteurs sont ensuite concrétisés en paquets réseaux à l'aide d'une bibliothèque décrivant comment implémenter les vecteurs pour chaque protocole (ex. : comment modifier un paquet OPC-UA). Enfin, ces paquets sont instanciés, soit de manière aléatoire, soit en fonction de la logique applicative de la plate-forme. Cette approche pourrait se généraliser aux systèmes d'information, mais elle exploite néanmoins des propriétés souvent présentes dans les systèmes industriels telles que l'absence de réseaux dynamiques, simplifiant la représentation de l'infrastructure.

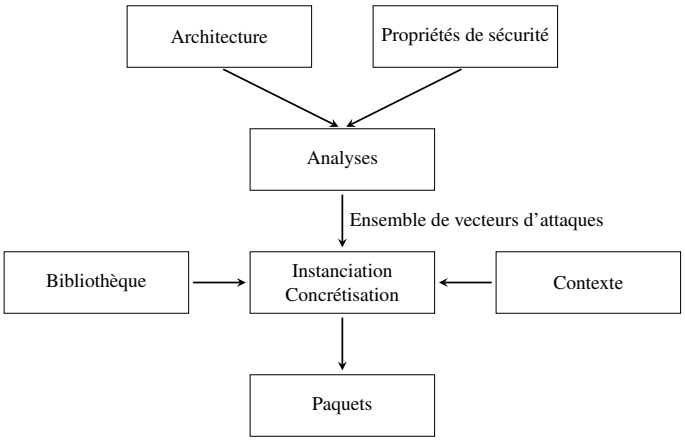

FIGURE 2: Méthodologie globale

4 Conclusion

En conclusion, nous proposons une approche de modélisation d'attaquants basée à la fois sur l'infrastructure d'un système industriel et des possibilités d'attaques contre celui-ci. Des scénarios d'attaques montrent comment un attaquant peut exploiter les faiblesses d'un protocole de communication pour satisfaire ses objectifs. Nous nous focalisons ici sur les possibilités de forger des attaques. A terme, nous souhaitons produire des attaques sur le fonctionnement du système, tenant donc compte du contenu des messages. L'analyse en elle-même pourrait être améliorée en considérant des attaques en plusieurs étapes. Par exemple, nous avons énoncé en section 2.2 que la configuration FTP$_{Auth}$ ne permet pas le vecteur d'attaque $Usurp$ car elle utilise une authentification par mot de passe. Cependant, le vecteur d'attaque $Lire$ pourrait révéler ce mot de passe et ainsi donner l'accès en un second temps au vecteur $Usurp$. Ces attaques en plusieurs étapes, tenant alors compte de l'ordre des actions à exécuter, pourraient être décrites sous forme d'arbre d'attaques [5] (représentation classique pour modéliser des attaquants). Un autre axe pourrait être la prise en compte de plusieurs attaquants coopérant pour des objectifs communs.

Références

1. Sylvain Conchon and Jean Caire. Expression des besoins et identification des objectifs de résilience. *C&esar'15*, 2015.
2. ISA-62443-3-3. Security for industrial automation and control systems, part 3-3 : System security requirements and security levels, 2013.

3. S Kriaa, M Bouissou, and Y Laarouchi. A model based approach for SCADA safety and security joint modelling : S-Cube. In *IET System Safety and Cyber Security*. IET Digital Library, 2015.
4. Ralph Langner. Stuxnet : Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE*, 9(3) :49–51, 2011.
5. Bruce Schneier. Attack trees. *Dr. Dobb's journal*, 24(12) :21–29, 1999.
6. Theodore J Williams. *A Reference Model for Computer Integrated Manufacturing (CIM) : A Description from the Viewpoint of Industrial Automation : Prepared by CIM Reference Model Committee International Purdue Workshop on Industrial Computer Systems*. Instrument Society of America, 1991.