Journal of Information Security and Applications 34 (2017) 153–165

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Detection of attacks based on known vulnerabilities in industrial networked systems

Manuel Cheminod*, Luca Durante, Lucia Seno, Adriano Valenzano

CNR-IEIIT, c.so Duca degli Abruzzi 24, Torino I-10129, Italy

ARTICLE INFO

Article history:
Available online 27 July 2016

Keywords:
Industrial distributed systems
Software vulnerabilities
Attack paths
Automated analysis

ABSTRACT

Vulnerabilities in software and hardware components can be exploited by attackers to cause damages through the cyberspace. Nowadays, this problem also affects a large number of industrial networked systems (INS) and experts are well aware that suitable prevention/detection techniques and countermeasures have to be developed, taking into account INS characteristics and peculiarities. The exposure of a large and complex system to attacks carried out by exploiting well-selected sequences of vulnerabilities can be hard to evaluate, but this is a fundamental step to prevent potential menaces in both the system design and operation phases. This paper deals with an innovative technique, which is able to compute all attack patterns leveraging known vulnerabilities present in an industrial system. The proposed approach is based on the extension of a twofold model, which was successfully developed for verifying the implementation of access control policies in INS. Our solution enables the development of an automated software analyser that can help with the design and maintenance of INS when their security is considered.

1. Introduction

Cyber-security has become a major issue of concern in most application areas which involve cyber-physical systems either directly or indirectly (Cheminod et al., 2013; Granzer et al., 2010; Knowles et al., 2015; Lin et al., 2009; Shakshuki et al., 2013). Among the others, industrial networked systems (INS), such as those used for automated production plants and process control, are now exposed to the same menaces and attacks, carried out by hackers and saboteurs, as those experienced by many non-industrial Internet users every day. Major reasons for this are a higher degree of openness, which has progressively been introduced in INS with respect to the proprietary solutions widely adopted until some years ago, and the ever increasing use in industrial scenarios of advanced information and communication technologies (ICT), such as wireless communications and off-the-shelf hardware (h/w) and software (s/w) components, that were originally conceived for boosting general-purpose and/or consumer applications.

Electronic devices and s/w products are notoriously known for not being perfect, at least from the security point of view, and are frequently subject to patches and updates when bugs and weaknesses are discovered. Unfortunately, this good practice, which is now routine in many networked systems, can be rarely adopted in INS, because of their well-known peculiarities (Cheminod et al., 2013; Knowles et al., 2015). Vulnerabilities, which unavoidably affect IT components, can be leveraged by malicious users to carry out attacks and cause damages to the target systems in the broader sense. Vulnerabilities are not the same as attacks, but rather a means to make possible attacks successful. This may occur when vulnerabilities are exploitable so that attackers can take advantage of them. However, a system can tolerate the presence of non-exploitable vulnerabilities because they cannot be leveraged to carry out attacks, and this aspect should also be considered in many INS where changes to the h/w and/or s/w are hardly possible or simply not financially convenient. In this case, in fact, the presence of vulnerabilities can be tolerated until they are proven not to be exploitable for causing harms/damages to the people, the system and the environment.

Unfortunately, weaknesses that are not exploitable on their own can also give rise to serious security threats and enable attacks when they are suitably combined in the same network. In this case, in fact, attackers can leverage well-selected patterns of known vulnerabilities to carry out sequences of steps that eventually result in reaching their malicious goals. Therefore, detecting such a kind of attack paths before their possible exploitation is of utmost importance to prevent unwanted consequences for the system.

* Corresponding author. CNR-IEIIT, c.so Duca degli Abruzzi 24, Torino I-10129, Italy. Fax: +39 011 0905429.
E-mail addresses: manuel.cheminod@ieiit.cnr.it (M. Cheminod), luca.durante@ieiit.cnr.it (L. Durante), lucia.seno@ieiit.cnr.it (L. Seno), adriano.valenzano@ieiit.cnr.it (A. Valenzano).

http://dx.doi.org/10.1016/j.jisa.2016.06.003
2214-2126/

Software vulnerabilities, in particular, are of predominant practical interest because they are frequently included in products and are easily replicated, so that they are being collected and sorted in publicly-available databases (MITRE et al.; NIST et al.; OSVDB; Symantec) since several years. Vulnerability databases are maintained and updated as new possible security breaches are discovered, and some information repositories, which are particularly oriented to industrial ICT components and supervision, control and data acquisition (SCADA) systems (Siemens AG; U.S. Department of Homeland Security and Industrial Control Systems Cyber Emergency Response Team (ICS-CERT)), have also been started and are regularly updated.

Our past investigations on the automated analysis and detection of attacks based on sequences ("chains" in Cheminod et al., 2009; Maggi et al., 2008) of known vulnerabilities have enabled us to pioneer researches in this area and to evaluate pros and cons of the original solution described in Cheminod et al. (2009) and Maggi et al. (2008).

More recently, we have presented a new technique (Cheminod et al., 2015; Cibrario Bertolotti et al., 2015), based on a twofold innovative model, for computer-aided security analyses (and, in particular, for checking the implementation of access control policies as described in Cheminod et al. 2014, 2015a) by means of automatic software tools, which has been conceived bearing in mind the typical requirements and peculiarities of INS. The proposed approach and underlying formal model can be successfully extended to the study of attacks leveraging sequences of vulnerabilities as shown in Cheminod et al. (2015b). This paper focuses on the extension of the technique introduced in Cheminod et al. (2015) and Cibrario Bertolotti et al. (2015) to enable the semi-automated analysis of vulnerability patterns, while circumventing some practical limitations of our previous approach in Cheminod et al. (2015b).

The following benefits are obtained by putting together vulnerability analysis and verification of access control policies:

1. Just one model is needed to carry out both analyses. In fact, even if the effort for building the model is seldom stressed by authors in the literature, it represents a major obstacle to make a theoretical framework applicable in practice. Moreover, as far as we know, the two kinds of analysis have always been dealt with separately until now, thus requiring different models and system descriptions, even if many system elements in the model (e.g., the network topology) are needed for both. This is why, for instance, Section 2, discussing the current state of the art, takes into account each kind of analysis separately.
2. What-if analyses are enabled, for instance, to evaluate either the effects of an employee identity theft or the behaviour of a disgruntled employee. Actually, this becomes quite natural in our unified framework, where the attacker is managed as whatever user interacting with the system. In other words, it is easy to study how employees can exploit vulnerabilities to enlarge their set of feasible actions, despite their initial allowed capabilities.
3. Physical access and system topology are of utmost importance in INS and cyber-physical systems and our model takes into account users' movements inside the system. This enables the description of attacks which leverage physical accesses to the system resources, and results in the security analysis of more realistic scenarios.

The reader should be warned that the proposed methodology, as the one in Cheminod et al. (2015) and Cibrario Bertolotti et al. (2015), is not applicable only to INS but can be also profitably employed for detecting chain of exploitable vulnerabilities in general-purpose networked systems too. By contrast, while many tools, which rely on different techniques, are already available for the latter, some characteristics of INS (e.g., heterogeneity of h/w and s/w components, proprietary/special purpose communication protocols, peculiar application s/w, 24/7 availability requirements) often prevent them from borrowing existing solutions (unfortunately based on unacceptable assumptions) and justify the development of ad-hoc formalism and techniques (Cheminod et al., 2013).

The paper is structured as follows: Section 2 discusses related works on both vulnerability analysis and access control policy management, while Section 3 briefly recalls the main characteristics of the analysis framework that are needed to understand the remaining part of the paper. Section 4 deals with the extensions and changes that enable the description of vulnerabilities, their inclusion in the formal system model and the computation of all their possible exploitable sequences. Section 5 presents a small realistic example built to show how the analysis can be carried out in practice, while some conclusions are drawn in Section 6. Appendix A describes those differences with respect to the formalism presented in Cibrario Bertolotti et al. (2015), which were specifically introduced to make the model more comprehensive and to deal with the exploitation of vulnerabilities.

2. Related works

Vulnerability analysis and access control policy management are major research topics for the security of networked computer systems. As far as we know, they have always been addressed separately, and this paper is an attempt to put them together. The main reason of such an approach can be found in the development of the system model. Actually, in our experience, this step is often a critical aspect for any analysis and assessment framework, since an adequate system description can be a cumbersome and complex task, and the same is true for keeping the model up to date and aligned with the real system. A single unified model able to satisfy requirements of different kinds of analysis, besides making things easier, can save effort and time. Scientific papers, published in the past, focus on either policy management or vulnerability exploitation, so we deal with them separately in the following. More detailed discussions of the relevant literature can be found in the "Related Works" sections of Cheminod et al. (2009) and Cheminod et al. (2015) respectively.

2.1. Vulnerability analysis

Vulnerability analysis, also known as vulnerability assessment, was introduced in Phillips and Painton Swiler (1998); Ritchey and Ammann (2000); and Sheyner et al. (2002). All those works were aimed at computing an attack graph in which each vertex represents a system state, whereas each edge describes a possible state transition determined by an attacker action. The attack graph generation in Phillips and Painton Swiler (1998); Ritchey and Ammann (2000); and Sheyner et al. (2002) has exponential computational complexity, thus leading to intractability when dealing with reasonably sized real systems. By means of the *monotonicity* hypothesis (i.e., the attacker never discards any of the acquired capabilities), introduced in Ammann et al. (2002) and, for instance, followed by Cheminod et al. (2009) and Jajodia et al. (2005), the computational complexity becomes polynomial. Some tools such as TVA (Jajodia et al., 2005), NetSPA (Lippmann et al., 2006), MulVAL (Ou et al., 2005) and Skybox (Skybox) rely on this assumption. Monotonicity is a reasonable hypothesis, as it is conservative, i.e., it guarantees that the worst case is never missed.

More recent works introduce enhancements to the above techniques along different axes. Kotenko and Chechulin (2013) suggest an *anytime* approach based on a set of algorithms that provide results with different timing and precision, and support the system manager in a near real-time fashion, paying particular attention to the (semi)automatic feed of (parts of) the system model (e.g. vul-