Journal of Network and Computer Applications 75 (2016) 156–168

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Towards scalable security analysis using multi-layered security models

Jin B. Hong*, Dong Seong Kim

Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand

ARTICLE INFO

Article history:
Received 13 September 2015
Received in revised form
29 February 2016
Accepted 26 August 2016
Available online 31 August 2016

Keywords:
Attack graphs
Attack trees
Complexity analysis
Scalability
Security analysis
Security model

ABSTRACT

Security models, such as an attack graph (AG), are widely adopted to assess the security of networked systems, such as utilizing various security metrics and providing a cost-effective network hardening solution. There are various methods of generating these models, but the scalability problem exists for single-layered graph-based security models when analyzing all possible attack paths. To address this problem, we propose to use a multi-layer hierarchical attack representation model (HARM) that models various components in the networked system in different layers to reduce the computational complexity. First, we formulate key questions that need to be answered to assess the scalability of security models. Second, we formally define the multi-layer HARM. Last, we conduct experiments to show the scalability of security models. Our experimental results show that using the HARM can improve the scalability of assessing the security of the networked system significantly in comparison to the single-layered security models in various network scenarios.

1. Introduction

Cyber criminals can compromise networked systems by exploiting vulnerabilities, and such events impose a critical socio-economic impact on enterprises and individuals. An attack surface describes vulnerabilities that cyber criminals can exploit to penetrate through the networked system (Manadhata and Wing, 2011), and so it is of paramount importance to secure the networked system by minimizing the attack surface (e.g., patching vulnerabilities).

Security models, or also known as attack representation models (ARMs), are well-defined means of analyzing the security of networked systems in efforts to enhance the fundamental framework for network security (Schumacher and Ghosh, 1997; Lippmann and Ingols, 2005; Kordy et al., 2013). These models can be used to analyze vulnerabilities in the networked system, and provide solutions to effectively manage them (e.g., network hardening) (Ammann et al., 2002; Dewri et al., 2007; Saini et al., 2008; Dawkins and Hale, 2004). However, analyzing all possible attack paths using single-layered graph-based ARMs has a scalability problem (e.g., an attack graph (AG), Sheyner et al., 2002). This is an emerging problem as network systems are becoming large, such as the Cloud (Popovic et al., 2010; Mell and Grance, 2011; Sood, 2012).

Two main approaches, namely structural modifications (Ou et al., 2006; Ingols et al., 2006; Xie et al., 2009) and heuristic methods (Homer et al., 2008; Poolsappasit et al., 2012; Chen et al., 2010), are proposed and used (separately or combined) to improve the scalability of ARMs. However, networked systems are becoming larger and highly dynamic (e.g., Cloud networks). Consequently, structural modifications solutions still suffer the scalability problem when the size of the networked system becomes very large (Lippmann and Ingols, 2005; Noel and Jajodia, 2004; Hong and Kim, 2012). Also, heuristic methods are model-centric (i.e., only applicable to a subset of security models). As a result, users may require multiple implementations of different security models to be able to analyze various security metrics (e.g., a security model may not support all security metrics required). Moreover, it becomes difficult to validate the result of security analysis when other security models do not provide the same function. Therefore, there is a need for security modeling and analysis techniques to deal with the scalability problem to compute all possible attack paths to analyze the security of networked systems.

We propose to use a multi-layered hierarchical attack representation model (HARM) to improve the scalability problem (Hong and Kim, 2012), and we analyzed the computational complexities of the HARM in each phase of an ARM lifecycle (which is described in Hong and Kim, 2013c) to compare the theoretical performances of different functionalities. The ARM lifecycle consists of five phases (pre-processing, generation, representation, evaluation, and modification), which are required steps in

* Corresponding author.
E-mail addresses: jho102@uclive.ac.nz (J.B. Hong), dongseong.kim@canterbury.ac.nz (D.S. Kim).

http://dx.doi.org/10.1016/j.jnca.2016.08.024
1084-8045/© 2016 Elsevier Ltd. All rights reserved.

analyzing the security of the networked system. More details can be found in Hong and Kim (2013c). Our previous study in Hong and Kim (2012) showed that the HARM has better or equal computational complexities in comparison to the simplified AG, as well as the performance when taking into account different network topologies and varying number of vulnerabilities (as shown in Hong and Kim, 2013a). In this paper, we extend our previous work in Hong and Kim (2013a) to answer the following questions: (i) Can we improve the performance using more layers in the HARM? (ii) What is the performance of ARMs with respect to more complex network topologies? To answer these questions, the network density is also taken into account in the analysis. The network density describes the average number of edges between hosts, such that a high density value means there are many host pairs connected (e.g., a fully connected network topology), and a low density value means there are not many host pairs connected (e.g., a star network topology). The contributions of this paper are:

- A formal definition of h-layered HARM (h-HARM) using AGs.
- Scalability analysis and comparison of h-HARM, an AG and Two-Layered AG (TLAG) with respect to complex network topologies.
- Investigating the effects of the network density for generations and evaluations of ARMs.

The rest of the paper is as organized as follows. In Section 2, related work is introduced, and the overview of the HARM is shown in Section 3. In Section 4, we answer some of the key questions when assessing the scalability of ARMs using complexity analysis. In Section 5, we conduct security analysis using various ARMs, and experimental results are presented in Section 6. Discussion on our findings is presented in Section 7, and we conclude our paper in Section 8.

2. Related work

Security is an ongoing problem for enterprises and individuals (Schumacher and Ghosh, 1997), because analyzing the security posture is a difficult task for networked systems with many hosts and vulnerabilities. To improve our understanding of this problem, ARMs are developed and used with an intensive focus on improving the usability and functionalities. Graph-based ARMs are one of the most widely adopted security modeling techniques (Kordy et al., 2013), because they are user-friendly and provide various metrics as well as security analysis methods that were developed during the past decade. One of the first introduced graph-based ARMs by Phillips and Swiler (1998) is an attack graph (AG) that maps all possible attack paths in a given networked system, and Sheyner et al. (2002) formally defined the AG. However, computing all possible attack paths yields an exponential computational complexity, so many researchers presented more efficient methods of generating and evaluating the AG. Either improvements to the full AG using heuristic methods (e.g., graph simplification and clustering) (Noel and Jajodia, 2005; Sawilla and Skillicorn, 2012; Mehta et al., 2006; Hong and Kim, 2013b) or a new graph-based ARM structures (Ou et al., 2006; Ingols et al., 2006; Xie et al., 2009) based on logical references to the networked system properties are proposed to address the scalability problem. However, as the networked system becomes larger and highly dynamic (e.g., a Cloud network, Mell and Grance, 2011; Sood, 2012), these existing solutions still suffer from the scalability problem.

Ou et al. (2006) proposed a logical attack graph (LAG) that can be generated with a polynomial computational complexity, but they did not consider analyzing the complexity of security evaluation. Moreover, their experiment assumed that each host has the same vulnerabilities (i.e., a homogeneous networked system). Ingols et al. (2006) proposed a predictive graph and a multiple prerequisite graph (MPG). They reported that the scalability of the MPG has the size complexity of O(n log n), where n is the number of hosts in the networked system (i.e., almost linear with respect to the size of the networked system). MPG used graph simplification prior to evaluating the security of networked systems. Also, the number of vulnerabilities was fixed in their experiment (e.g., fixed with 10 vulnerabilities). Xie et al. (2009) used a two-layer attack graph (TLAG), where the upper layer captured the host reachability and the lower layer captured the vulnerability information. This is very similar to 2-HARM (i.e., 2 layered HARM), but the difference is that the lower layer information (i.e., vulnerability models) are stored in each edge between nodes in the upper layer in the TLAG (i.e., construct the vulnerability attack graph between host pairs). In contrast, the lower layer models have a one-to-one relationship with the upper layer nodes in the HARM. As a result, less memory space is required for the HARM than the TLAG. If we assume that the same methods are used for both HARM and TLAG, then the generation and evaluation times of the HARM will be better as there is less number of lower layer models in the HARM than the TLAG. However, authors did not conduct scalability analysis, and the vulnerability information was not given. In a similar way, Machida et al. (2013) proposed an automated composition of a hierarchical stochastic model from SysML to analyze the system availability. It demonstrates the scalability improvement while maintaining the accuracy of the analysis, given the model is decomposable. However, we focus not only on the availability modeling and analysis, but a broad field of security analysis using various security metrics.

Heuristic methods avoid computing all possible attack paths (e.g., graph simplification, Chen et al., 2010; Homer et al., 2008; Ingols et al., 2009; Noel and Jajodia, 2005; Sawilla and Skillicorn, 2012, and approximation algorithms, Abadi and Jalili, 2010; Islam and Wang, 2008; Mehta et al., 2006; Hong and Kim, 2013b; Hong et al., 2014). However, a major drawback of these methods is the lack of reusability of them due to their model-centric properties (i.e., methods proposed specific to certain security models and their properties). For example, a probabilistic approach to evaluating the security of the networked system proposed by Wang et al. (2013) requires to use a *dependency* AG. However, the method proposed specifically requires the use of *dependency* AG and it cannot be used by other security models directly. There is also a potential loss of security information in the evaluation phase due to taking into account only the subset of security information in the analysis. In this paper, we focus on improving the scalability of ARMs to evaluate all possible attack paths using the HARM that does not depend on the model properties, as well as without any loss of security information in the evaluation phase. A set of key questions that should be considered to assess the scalability of ARMs is given in Section 4, and we try to answer these questions with respect to some of the most recent work on structural modification solutions.

3. Overview of the HARM

The main idea of the HARM is to model the system components onto multiple layers in the model. By doing so, we can improve the scalability of utilizing security models, especially for large sized networked systems. For example, a small network shown in Fig. 1 has one malicious host (H_0) and two legitimate hosts (H_1 and H_2). The goal of the attacker H_0 is to compromise the root privilege of H_2.

We can create an AG to model the attack scenario of the given network as shown in Fig. 2. Although with a very small number of