JOURNAL OF INFORMATION SECURITY AND APPLICATIONS 29 (2016) 27–56

Available online at www.sciencedirect.com

ScienceDirect

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

A taxonomy for attack graph generation and usage in network security

Kerem Kaynar *

German Turkish Advanced Research Center (GT-ARC), TU Berlin, Ernst Reuter Platz 7, 10587 Berlin, Germany

ARTICLE INFO

Article history:
Available online 23 March 2016

Keywords:
Vulnerability
Full attack graph
Reachability analysis
Exploit
Weakness

ABSTRACT

Attack graphs model possible paths that a potential attacker can use to intrude into a target network. They can be used in determining both proactive and reactive security measures. Attack graph generation is a process that includes vulnerability information processing, collecting network topology and application information, determining reachability conditions among network hosts, and applying the core graph building algorithm. This article introduces a classification scheme for a systematical study of the methods applied in each phase of the attack graph generation process, including the usage of attack graphs for network security. The related works in the literature are stated based on the proposed classification scheme and contributive ideas about potential challenges and open issues for attack graph generation and usage are provided.

1. Introduction

Ever increasing utilization of computer networks in various areas of public and private sector amplifies the need to find mechanisms for securing data stored and transferred over the networks. Network security administrators employ specific proactive and reactive defense measures to ensure the confidentiality, integrity and availability of the network users' data. Extracting possible paths that an attacker can use to intrude into a target network is one of the most important activities in determining both proactive and reactive defense measures. It can be used in situational assessment in terms of network security, recognition of ongoing attack scenarios and prediction of future attacks.

An attack graph represents possible ways via which a potential attacker can intrude into the target network by exploiting a series of vulnerabilities on various network hosts and gaining certain privileges at each step. In a typical attack graph, the nodes represent the privileges gained by the attacker on the network hosts and the edges represent the software vulnerability exploits employed by the attacker to gain these privileges. The attacker may need to have a set of privileges on certain hosts in order to exploit a specific vulnerability on a network host. After successfully exploiting a vulnerability on the host, the attacker gains additional privileges on it and either continues attacking other hosts from this host or tries to elevate her privileges on this host using additional vulnerabilities.

The computation of an attack graph requires the computation of the reachability conditions among the network hosts by considering all network protocol layers, modeling attacks and attack paths, and devising an efficient method to compute possibly huge number of attack paths. This computation process should be systematically described in order to provide opportunities to the researchers for improvement in specific areas of attack graph generation in a structured manner. The systematic description should clearly cover the overall scientific landscape in attack graph generation and shed light to the researchers on every aspect on it. According to us, the usage of attack graphs should also be included by the systematic description in order to motivate the researchers to make the necessary refinements to their attack graph structures and

* *Corresponding author.* Tel.: +4915785300687; fax: +493031474003.
E-mail address: kerem.kaynar@dai-labor.de.
http://dx.doi.org/10.1016/j.jisa.2016.02.001

generation methods by solidifying their aims of usage of attack graphs at the beginning.

This article focuses on a systematic study of the literature related to attack graphs in network security in order to derive a taxonomy for the methods applied in attack graph generation and usage. The different methods proposed in the literature for handling basic problems arising in attack graph generation and usage are abstracted into categories defined by the proposed taxonomy. In this respect, an individual past work in the literature may have introduced more than one method each of which is related to different category. This situation results in putting the corresponding work into more than one category. For instance, if a past work proposes two different methods, one for attack modeling and the other for attack graph core building phase, then we relate each method of this work with a different category, one for each phase. Namely, we put the methods defined in the past works into categories by specifying its proposing work.

The systematic study of the proposed methods in the literature is performed by starting with the basic problems leading to the development of these methods. The next section opens the topic by presenting background information on attack graphs and identifying and discussing the basic problems that are encountered during the attack graph generation process. These problems shed light to the formation of the proposed taxonomy for classification of the methods employed during different phases of the attack graph generation process.

The proposed taxonomy is detailed in Section 3. The usage of attack graphs for network security is also categorized and exemplified by pointing to the past related works in Section 4. Section 5 provides a tabled categorization of the past works according to the proposed taxonomy, which can be used for quick reference. The description of the proposed taxonomy and the exemplification of the corresponding classification criteria are facilitated and streamlined by grouping the past works according to the laboratory or corporate working on the topic of attack graph generation and usage. The groups are as follows:

- Center for Secure Information Systems, George Mason University (G. M. U. Center for Secure Information Systems, 2015)
- MIT Lincoln Laboratory (M. Lincoln Laboratory, 2015)
- Computer Science Department, Carnegie Mellon University (C. M. U. Computer Science Department, 2015)
- Concordia Institute for Information Systems Engineering, Concordia University (C. U. Concordia Institute for Information Systems Engineering, 2015)
- Sandia National Laboratories, Albuquerque (A. Sandia National Laboratories, 2015)
- Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation (S. P. I. f. I. Laboratory of Computer Security Problems, 2015)
- LAAS-CNRS, France (L. for Analysis, F. Architecture of Systems, 2015)
- Core Security Corporate, Buenos Aires, Argentina (A. Core Security Corporate, 2015)
- Department of Electrical and Computer Engineering, University of Illinois (D. of Electrical, U. o. I. Computer Engineering, 2015)
- The Ruhr Institute for Software Technology University of Duisburg-Essen (T. R. I. f. S. T. U. o. D.-E. PALUNO, 2015) and Istituto di Informatica e Telematica—Consiglio Nazionale delle Ricerche (T. I. of Informatics, T. of CNR, 2015)

In each of the above three sections, the works of the groups are described according to the proposed classification criteria and cited. If the works of a group have no contribution according to a specific classification criterion, they are not mentioned in the description of this criterion.

We present the shortcomings of the current state-of-the-art methods and the opportunities for further research in the area of attack graph generation and usage in Section 6. Section 7 concludes the paper by summarizing the proposed taxonomy and describing the drawn conclusions.

2. Background on attack graph generation/usage and basic problems

The attack graph generation process is usually driven by a set of *initial privileges* that the attacker is assumed to posses at the beginning. The eventual target/leaf nodes of a possible attack graph are represented by the *goal privileges* that the attacker aims to gain at the end. A full attack graph tries to identify all possible attack paths from the initial privileges to the goal privileges, while a partial attack graph shows a portion of these possible attack paths (not necessarily all).

An attack graph correlates the vulnerability exploits that can be employed by a potential attacker on the network hosts and shows the evolution of multi-step attacks followed by the attacker. It may be dynamic, i.e., its nodes and edges can be updated, when new products are installed or existing products are uninstalled on the target network hosts. In such cases, new vulnerabilities may be added to the hosts or existing vulnerabilities may be removed. An attack graph may also contain vulnerability exploits as nodes instead of edges or contain nodes representing facts other than the privileges gained on the hosts or the vulnerability exploits. An example may be an attack graph containing information asset usages as its nodes. The usage of an information asset on a host may lead to specific privileges gained on the host or on any host indirectly reachable via this host. An example of such an information asset can be cookie files managed by a web browser on a specific network host.

The configuration of the installed software on the target network and the relationships among them determine the contents of the attack graphs produced for the network. A portion of an example attack graph for an example small network is shown in Fig. 2. The example network is shown in Fig. 1. The firewalls in the example network contain simple allowance rules. (The IP addresses in the figure are artificial.)

The format of the attack graph shown in Fig. 2 is custom designed and serves just as an example format. The example attack graph is composed of four types of nodes:

1. Privilege nodes indicating attacker privileges that can be obtained on the software installed on the network hosts with specific IP addresses,
2. Nodes indicating vulnerability exploits that can be applied by an attacker on the installed software,