Computers and Electrical Engineering 38 (2012) 1062–1072

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

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

Automatic network intrusion detection: Current techniques and open issues ☆

Carlos A. Catania [a,*], Carlos García Garino [a,b]

[a] ITIC, Universidad Nacional de Cuyo, Mendoza, Argentina
[b] Facultad de Ingeniería, Universidad Nacional de Cuyo, Mendoza, Argentina

ARTICLE INFO

Article history:
Available online 14 June 2012

ABSTRACT

Automatic network intrusion detection has been an important research topic for the last 20 years. In that time, approaches based on signatures describing intrusive behavior have become the de-facto industry standard. Alternatively, other novel techniques have been used for improving automation of the intrusion detection process. In this regard, statistical methods, machine learning and data mining techniques have been proposed arguing higher automation capabilities than signature-based approaches. However, the majority of these novel techniques have never been deployed on real-life scenarios. The fact is that signature-based still is the most widely used strategy for automatic intrusion detection. In the present article we survey the most relevant works in the field of automatic network intrusion detection. In contrast to previous surveys, our analysis considers several features required for truly deploying each one of the reviewed approaches. This wider perspective can help us to identify the possible causes behind the lack of acceptance of novel techniques by network security experts.

1. Introduction

A network intrusion detection system (NIDS) is the software tool that automates the network intrusion detection process. From an architectural point of view a NIDS can be analyzed from several angles (i.e. traffic capture process, system location, appropriate measures selection, among others). However, from a more simplified point of view, intrusion detection can be seen just as a classification problem in which a given network traffic event is assigned as *normal* or *intrusive*.

In the past 20 years, several techniques have been proposed to address the embedded classification problem inside NIDS. Perhaps the most successful approach has been the one based on pattern signatures describing known attacks behavior [1]. Under this approach, a malicious event is detected when some monitored event matches against a signature pattern. Despite signature-based NIDS are considered the *de facto* standard, they face the problem of needing a new set of signature patterns each time a new attack emerges. In addition, signatures describing such attacks have to be written by experts, which are not always available. In other words, the signature-based approach has failed in providing the level of automation required by security staff members.

Alternatively, techniques including statistical methods, machine learning and data mining methods have been proposed as a way of dealing with some of the issues regarding signature based-approaches. Such techniques aim at facilitating the work of the network security staff, providing a higher automation in the intrusion detection process along with good detection capabilities. Despite the success in obtaining high accuracy levels, most of these techniques have actually not been

☆ Reviews processed and proposed for publication to Editor-in-Chief by Guest Editor Dr. Gregorio Martinez.
* Corresponding author. Tel.: +54 02614291000.
E-mail addresses: ccatania@itu.uncu.edu.ar (C.A. Catania), cgarcia@itu.uncu.edu.ar (C.G. Garino).

0045-7906/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compeleceng.2012.05.013

deployed in real-life scenarios. This situation suggests that accuracy is not the only goal in the pursuit of automatic intrusion detection.

The present work reviews the most relevant network intrusion detection techniques for wired networks, putting special emphasis on the embedded classification problem. However, in opposition to previous surveys on this field, analysis is performed considering not only accuracy results but also other features required for implementing the discussed techniques in real-life scenarios.

The rest of this work is organized as follows: Section 2 provides background information about the intrusion detection problem, including attack definitions, a taxonomy and a simplified NIDS architecture. Then, in Section 3, the most relevant approaches applied to intrusion detection are reviewed and compared based on the taxonomy along with common measures related to NIDS. Section 4 remarks the remaining open issues, which aim to explain why all except the signature-based approach are not being deployed on current networks. Finally, concluding remarks are provided in Section 5.

2. Background

Before discussing the most relevant approaches to NIDS, we proceed to describe the fundamental elements inside the intrusion detection problem.

2.1. Attack definition and classification

A computer *attack* can be defined as the intelligence of evading or evading attempt of computer security policies, acceptable use policies, or standard security practices. In the security research community, the terms *attack* and *intrusion* are often used with the same meaning.

In the past years, there have been several attempts to build taxonomies aimed at classifying attacks. One of the most accepted taxonomy is the one proposed by Kendall [2], in which attacks can be classified into four categories:

Probing: Attacks oriented to gather information about the system, for further intrusion. These attacks include network traffic sniffing and port/address scanning.
Denial of Service (DoS): Attacks attempting to diminish or totally interrupt the use of a system or a service to their legitimate users.
User to Root (U2R): Attacks that aim to gain superuser access to the system by means of exploiting vulnerabilities in operating systems or software applications. The attacker has a valid account in the system.
Remote to Local (R2L): Attacks oriented to gain local access from outside the network.

A broad attack taxonomy is presented by Lazarevic et al. in [3], in which a new category is added for programs that replicate on host machines or propagate through the network. This new category includes programs such as *viruses*, *worms* and *trojan horses*.

2.2. A simple NIDS architecture

In general, from an architectonic point of view, a NIDS is based on the following modules:

Traffic Data Acquisition: This module is used in the data collection phase. In the case of a NIDS, the source of the data are raw network frames or information from upper protocol layers (i.e. IP or UDP protocols).
Traffic Features Generator: This module is responsible for extracting a set of selected traffic features from captured traffic. Network traffic features can be classified in *low-level* features and *high-level* features. A *low-level* feature can be directly extracted from captured traffic (e.g. IP header). Whereas a *high-level* feature consists of traffic information deduced from captured traffic by a subsequent process. Features can be also classified according to the network traffic source used for generating them. *Packet* features are those directly obtained from network raw packets headers. *Flow* refers to features containing aggregated information related to network connections. Finally, *Payload* stands for those features obtained from packet payload.
Incident Detector: This module processes the data generated by the *Traffic Features Generator* module to identify intrusive activities. Traditionally, network intrusion detection methodologies have been classified into two broad categories [4]: *misuse detection* (matches the input data against a definition of an attack) and *anomaly detection*(based on a definition of normal behavior of the target system). No matter the detection methodology implemented by the *Incident Detector*, once a malicious event has been detected, an alert will be raised and sent to the *Response Management* module.
Traffic Model Generator: This module contains the reference data used by the *Incident Detector* to compare with. The source of information of the *Traffic Model Generator* could come from human knowledge or from some automatic knowledge acquisition procedures.
Response Management: Once an alert is received, this module has the responsibility to initiate actions in response of a possible intrusion.