A Temporal Logic Based Framework for Intrusion Detection

Zanolin Lorenzo

August 31, 2023

Abstract

The purpose of this paper is to introduce MONID, which is a framework created for system intrusion detection. This framework uses EAGLE, a rich and effectively monitorable logic, to express intrusion patterns using temporal logic formulas; EAGLE's ability to include data values and parameterized recursive equations makes it possible to represent security threats that include complex temporal event sequences and attacks with intrinsically statistical signatures succinctly. This tool can be used in offline and real-time scenarios. The implementation uses an algorithm for online monitoring that matches descriptions of the lack of an assault with indications of system execution; an alarm is set off whenever the standard is broken.

Contents

1 Introduction

Even with all of the advances in computer security research, totally safe computer systems remain a long way off. Almost every large and complicated computer system nowadays contains vulnerabilities. Intrusion detection means maintaining constant surveillance on a system in order to detect any misuse of these weak areas as soon as feasible so that they can be repaired.

There are three Intrusion Detection System approaches, according to the literature: *signature-based*[4], *anomaly-based*[6] and *hybrid*[5]. The first approach aims to identify patterns and match them with known signs of intrusions relying on a database of previous intrusions. If activity within the network matches the "signature" of an attack or breach from the database, the detection system creates an alert. This approach has a low false-alarm rate, but it requires us to know the patterns of security attacks in advance and previously unknown attacks would go undetected. In contrast, *anomaly-based* is capable to detect new attacks since an alarm is raised if an observed behavior deviates significantly from pre-learned normal behavior. Finally, a *hybrid system* combines the best of both worlds by looking at patterns and one-off events, a Hybrid Intrusion Detection system can flag new and existing intrusion strategies.

In this paper we will focus on *signature-based* approach using temporal logic; we use EAGLE[3, 1] to specify a system's attack-safe behavior. EAGLE allows recursively built temporal formulas, parameterizable by both logical formulas and data expressions, across three primitive modalities: "next", "previous", and "concatenation". The logic allows us to describe temporal patterns involving reasoning about data-values observed in individual events. Unlike LTL, EAGLE allowa us to design attacks with fundamentally statistical characteristics; password guessing attacks and ICMP-flood denial of service attacks are two examples. We'll see how the implementation, MONID, examines the event stream to see if the monitored formula (which is used to hold the pertinent summary of the system) is being broken.

In Section 2 we will analyze the basics of EAGLE. In Section 3 we will analyze MONID framework and some evalutations on the performances. In Section 4 we will see some attacks specifications. In section 5 we will conclude.

2 EAGLE, a Temporal Monitoring Logic

According to [2], EAGLE offers a succinct but powerful set of primitives, essentially supporting recursive parameterized equations, with a minimal/maximal fix-point semantics together with three temporal operators: next-time ($\bigcirc$), previous-time ($\odot$), and concatenation ($\cdot$).

2.1 Basics and syntax

In EAGLE recursion definitions are supported; in the current framework we can build the following definitions:

$$\underline{\text{max}}\ \text{Always}(\underline{\text{Form}}\ F) = F \wedge \bigcirc \text{Always}(F)$$
$$\underline{\text{min}}\ \text{Sometime}(\underline{\text{Form}}\ F) = F \vee \bigcirc \text{Sometime}(F)$$

As we can see, rules are parameterized, which means that we will be able to write EAGLE formulas such as Always(Sometime($x > 0$)). Also, the Always operator is defined as maximal solution of the equation $X = F \wedge \bigcirc X$ and the Sometime operator represents the minimal solution to the equation $X = F \vee \bigcirc X$.

We will use it to create formulas that describe the absence of some attacks and given a trace of events σ generated from the logs we will check whether

References

[1] Howard Barringer et al. "EAGLE can do Efficient LTL Monitoring". In: *Fossacs Ortacas*. 2003.

[2] Howard Barringer et al. "Program monitoring with LTL in EAGLE". In: *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.* IEEE. 2004, p. 264.

[3] Howard Barringer et al. "Rule-based runtime verification". In: *Verification, Model Checking, and Abstract Interpretation: 5th International Conference, VMCAI 2004 Venice, Italy, January 11-13, 2004 Proceedings 5*. Springer. 2004, pp. 44–57.

[4] Wei Gao and Thomas H Morris. "On cyber attacks and signature based intrusion detection for modbus based industrial control systems". In: *Journal of Digital Forensics, Security and Law* 9.1 (2014), p. 3.

[5] Akash Garg and Prachi Maheshwari. "A hybrid intrusion detection system: A review". In: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. IEEE. 2016, pp. 1–5.

[6] VVRPV Jyothsna, Rama Prasad, and K Munivara Prasad. "A review of anomaly based intrusion detection systems". In: *International Journal of Computer Applications* 28.7 (2011), pp. 26–35.