Investigation of the Effectiveness of Alert Correlation Methods in a Policy-Based Security Framework

Bartłomiej Balcerk¹, Piotr Dragan¹, Bogdan Trawiński², and Marcin Wojtkiewicz¹

¹ Wroclaw Centre for Networking and Supercomputing, Plac Grunwaldzki 9, 50-377 Wroclaw, Poland
² Wroclaw University of Technology, Institute of Applied Informatics, Wybrzeże S. Wyspiańskiego 27, 50-370 Wroclaw, Poland
{bartlomiej.balcerk,piotr.dragan,bogdan.trawinski,marcin.wojtkiewicz}@pwr.wroc.pl

Abstract. The investigation of the effectiveness of alert correlation methods implemented in the Alert Correlation Module was presented in the paper. The module was developed within the POSITIF project (Policy-based Security Tools and Framework) funded by the European Commission. Three effectiveness metrics were applied: reduction coefficient, average ancestor count in alert trees and percentage of meta-alerts in outgoing alerts. Research was conducted using a test environment comprising 14 computers with IDS installed. Network traffic was monitored for 40 days, and during this time the IDSs generated 211 251 alerts. The module correlated and recognized as dangerous 6994 of them therefore achieving a level of reduction equal to 96.69 percent, which could be regarded as a good result.

Keywords: intrusion detection, alert correlation, alert merging.

1 Introduction

Alert correlation is an essential process in a network security system because it recognizes similar alerts and merges them assuring the work of network administrators will be more effective. Several alert correlation approaches were proposed. Cuppens and Miege [1] distinguished three phases in this process: alert clustering, alert merging and alert correlation. IDMEF (Intrusion Detection Message Exchange Format) was used as the format of input data and the attacks were described by means of pre- and post-conditions using the LAMBDa language. Valdes and Skiner [10] proposed a mathematical approach determining minimal values of feature similarities of alerts to be merged. Xu and Ning [12] introduced the notion of the "triggering event" focusing first of all on the causes generating alerts. Our implementation has been inspired by the approach proposed by Valeur et al. [11], and similar methods were described by Debar and Wespi [4]. All of the above mentioned methods are based on the similarity between alert
attributes and our approach also belongs to this class. Research reported in the present paper was carried out within the POSITIF (Policy-based Security Tools and Framework) project [5,8] funded by the European Commission.

2 Outline of Alert Correlation Methods

The alert correlation module is composed of a set of procedures which can be arranged in many ways. Some procedures process data of an alert and the others implement correlation methods within individual filters. Each filter is characterized by a time window i.e. maximal time of alert correlation and a reduction flag determining whether the ancestors of alerts being correlated should be deleted when the time window for a given alert expires. All alert correlation algorithms use a time window to determine the maximal time an alert can stay in a queue of a given filter. But not all alerts wait in queues for a whole window length. If a new alert occurs in a queue with a time tag of window length greater than other alerts then the latter is moved to the queue of the next filter. Such a mechanism assures better effectiveness and lower memory consumption, which is especially important during archival file processing. Five types of filters were implemented: Fusion, One2One, Network-Host, One2Many and Many2One. The Fusion filter is a simple and very useful one. Its task is to merge two or more alerts referring to the same event but generated by different IDSs. It should be located at the beginning of the correlation process. The One2One filter associates alerts from the same source and targeted to the same machine. The main task of the Network-Host filter is to correlate alerts generated by the network IDSs with alerts produced by the host IDSs concerning the same network node. In turn One2Many and Many2One filters are designed for detecting repeated alerts which are similar with respect to source or destination address.

3 Investigation of Alert Correlation Effectiveness

3.1 The Architecture of the Testbed

To conduct experiments a test bed was constructed comprising 14 computers. Four of them were unshielded and the other ten created a local area network and were protected from the Internet by means of a firewall. Each of the first four computers was equipped with a Snort functioning as IDS, hepteneths - honeypot, and the Prelude LML log analyzer. This group of computers logged alerts to a Prelude system which could communicate with the correlation module over a BEEP protocol. Other computers operating within the local network had no IDS, but the network traffic from the switch binding them was redirected to a computer with a Snort and second Prelude system, which was also connected to the correlation module. The test bed was connected to the Internet and exposed to actual network traffic. The traffic was monitored for 40 days from January 23th through March 3rd 2007. During this time all IDSs generated a total of 211 251 alerts.