Empirical Analysis and validation of security Alerts filtering Techniques

Abstract:

System administrators cope with security incidents through a variety of monitors, such as intrusion detection systems, event logs, security information and event management systems. Monitors generate large volumes of alerts that overwhelm the operations team and make forensics time-consuming. Filtering is a consolidated technique to reduce the amount of alerts. In spite of the number of filtering proposals, few studies have addressed the validation of filtering results in real production datasets.

Existing system:

Security datasets have been addressed so far through a variety of techniques. The work describes the transformation steps and analysis workflow, which are required to develop security situation awareness from the raw data. For example, correlation aims to group data pertaining to the same incident. This research capitalizes on machine-learning-based approaches - such as neural networks, decision trees and Bayesian inference - and aims to build aggregate data points from the raw alerts. Forensics aims to gather evidence from the data with the aim of understanding entry points, progression and impact of incidents. We are aware of few forensics research contributions that disclose measurements based on naturally occurring attacks.

Disadvantages:

1. These applications security levels have to be improved.
2. Accurate data will not be maintained.

Proposed system:

This paper analyzes a number of state-of-the-art filtering techniques that are used to address security datasets. We use 14 months of alerts generated in a SaaS Cloud. Our analysis aims to measure and compare the reduction of the alerts volume obtained by the filters. The analysis

highlights pros and cons of each filter and provides insights into the practical implications of filtering as affected by the characteristics of a dataset. We complement the analysis with a method to validate the output of a filter in absence of ground truth, i.e., the knowledge of the incidents occurred in the system at the time the alerts were generated. The analysis addresses blacklist, conceptual clustering and bytes techniques, and our filtering proposal based on term weighting.

Advantages:

1. This application improves the security of the datasets.
2. Application performance will be improved.

SYSTEM REQUIREMENTS

H/W System Configuration:-

- Processor - Pentium –III
- RAM - 256 MB (min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

S/W System Configuration:-

- Operating System : Windows95/98/2000/XP
- Application Server : Tomcat5.0/6.X
- Front End : HTML, Jsp
- Scripts : JavaScript.
- Server side Script : Java Server Pages.
- Database : MySQL 5.0
- Database Connectivity : JDBC