Applying CEP to problems of real-time data analysis in distributed IDS.

Tigran Tsaturyan,
Bauman Moscow State Technical University
a915200@yandex.ru

Abstract

Nowadays developers and researchers apply different approaches from the traditional rule based solutions to data mining or pattern search in DIDS. To implement their inventions into life, performance question is still open as DIDS is designed to operate in real-time with millions of packets per second. In this paper, we pay close attention to data systems used and will offer our draft of the data processing system to be used at implementation including all best from old and new approaches, designed especially to data mining applications.
Academic supervisor: Yuri Gapanyuk, gapyu@bmstu.ru

1.Introduction

Distributed Intrusion detection system (DIDS) is software or device designed for detection malicious activity in several inspected objects. Every DIDS require data to perform its analysis. The faster data delivery system is, the quicker decision will be made and attacker locked. Usually such systems are built as client-server architecture, where server consist of main engine, logic, rules, logs, etc. and a client (also called sensor), that collects data or perform basic/extended analysis. [3]

The more functions perform the client; the more is the requirement for a hardware. On the other hand, making the sensor to perform only data collection will result in extreme network loading as a copy of every in/out packet will be sent to the server. As combined approach here is used, so designed system should be able to function according to both possible options and act not only as a communication platform, but the processing solution itself twisted with IDS.

2. Related Work

Question of data-processing architecture in DIDS is relatively new, probably as data-mining IDS are still in development and modern security products are either closed in documentation or are limited. There are attempts [2] to use MySQL database as data-storage core, for example, but solution offered does not practice any intelligent techniques, but rather is a good practical guide to set up DIDS easy. Researchers [4] used CEP (Complex Event Processing) as basement for network scans detection system with positive results. Complex Event processing (CEP) is a method of tracking and analyzing (processing) streams of information combined from several sources about things that happen and deriving a conclusion from them. Due to CEP limitation (described lately here) these researchers had to employ extra data warehouse. Some researchers [10] simply use sensors as a sniffer to redirect in and out going traffic with only native or implemented lists, arrays, and other datasets in memory. Distributed Prelude-IDS [11] work as with MySQL as well as with PostgreSQL, storing however only results and perform "analysis on the fly". Experiments, based on Intelligent Techniques such as Genetic Algorithms, Neural Networks, Data Mining were carried on relatively little amount of traffic with some possible delays, however the desire to apply it in real time transformed the software for real-time intelligence.

3 Requirements

To build the framework for the DIDS to operate successfully, we must meet main requirement for such systems, i.e. being fast, reliable and fit to possible tasks.

3.1 Understanding Data Processing in DIDS

To understand what kind information and queries, we are facing with, there is need to understand the attacks themselves and used identification approaches.

Basically attacks can be classified into three groups:

- Network scanning
- Denial of service attack
- Different human attacks (exploits, errors in code, vulnerabilities, etc.)

Network scanning is a way to understand network structure, its hosts and open ports. Such hacking technique can give an answer to questions what is the network, how many hosts are there, what OS they are running, what is opened or locked. Let TCP SYN be an example [please, refer to appropriate documentation of such attack]: the scanner sends SYN packet to open connection to the target and waits for response. If SYN-ACK packet is received, so scanner suppose such port open and host up. If RST-ACK packet is received than scanner conclude port to be closed. If nothing is

received, so port can be filtered or host down. The scanner can be run on one host, or the scanner can be several hosts. To identify the scanning attack, IDS (DIDS), for example, create a list of enquires to different ports from one source, sum them and compare with some threshold [4].

Denial of Service attack (DoS) is an attack to block or overload the target machine and make it inaccessible to users. DoS is easily identified, but not so easy to prevent. The approach for attack recognition is similar to network scanning with difference that now we have to consider packet size or go deeper to application layers and understand requests. Very often attack is performed by the group of machines (bot net) and to distinguish real user from malicious computer necessitate some analysis. Moreover, we need to mention here that DoS significantly increases system traffic, what means that the data-processing system should be able also to withstand an attack and not be the narrow bottle.

Different human attacks are designed for targeting errors in code (vulnerabilities) or wrongly configured software. Basic rule approach is used here, where each packet is searched in existing signature database or some anomaly detection. Anomaly detection require to have a vast database of good packets and examples of wrong one to function properly [5]. For example, K-means clustering algorithm gives satisfactory results with its simplicity [5].

Realization of these approaches requires a solution to transfer fast data between sensors and server and a storage system in server to keep suspicious packets and relevant data. The process of evaluation information as it arrives is called Real-time intelligence.

3.2 Performance

When it comes to solution, some real data is needed. For example, ordinary communication speeds are 1-40Gbs depending on an application. In case of distributed system, roughly, this figure is multiplied by s times, where s equals to number of sensors. Such approximation shows that there is no opportunity to store all data. However, statistics, IP addresses, ports, type and other important and crucial to our application information need to be. Data can be hashed to store minimum required information. It's hard to predict real numbers as they are very seriously application specific, but in any case, system should be able to handle millions of packets per second, what is the average load of middle size network.

3.3 Transferring data

Transferred data from client to server consist of informational messages, messages for statistics (IP's, ports, etc.) and packets themselves. Ideally, a new protocol should be designed here based on existing efforts, however, we see employing Binary XML as a possible solution, allowing storage of binary data.

4 Existing and Our Architectures

Researchers in [6] showed architecture (please, refer to figure 1) of data mining IDS that is not distributed. Also the performance of such system was not given.

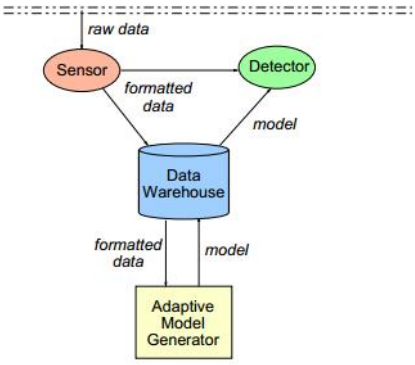

Figure 1. Design of Data Mining base IDS [6].

Assuming that millions packets move through such system build on relational database, we understand that it will probably:

1) Be too slow (at least 2 x millions INSERT per second, 2 x millions SELECT per second)
2) Not very scalable.
3) Possible problems with parallel processing, as simultaneous INSERT (and it's very likely to occur) may face that table is locked thus bringing delays.

Based on current IDS design, our requirements stated above and different approaches and demand of researchers in [1-7], we decided to consider next structure:

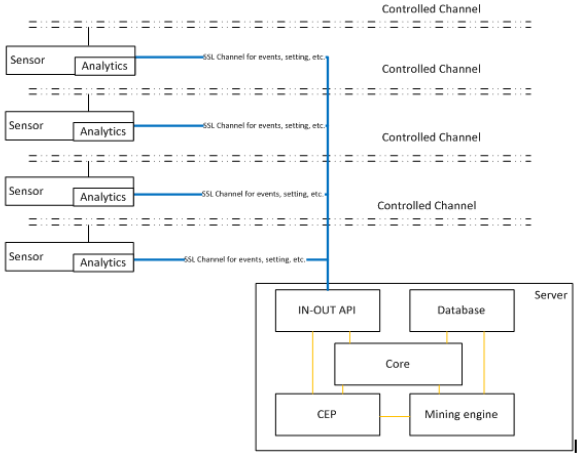

Figure 2. Our DIDS architecture based on CEP

Sensor is responsible for capturing data. It can be installed on the gate or dedicated server linked with gate via hub to collect every in- or out- going packets. For example, detection of network scans or DoS attacks requires getting IPs and ports information. Sensor collects this data based on packed headers, groups it and sends back to the server. Search for dependencies or patterns are performed on the server.

5 CEP role

Due to high volume of information, it should be processed in real-time or with minimum delay, with instant decisions and limited backing for future investigation. Some papers [9] however suggest that so called off-line processing (processing after communications, in off-line mode) give benefits of more accurate recognition and usually is employed along with real-time analysis.

We suppose that complex event processing fits all required conditions. CEP is designed like a database turned upside-down. We load rules there and put all information through it to get results. Such approach supposed to be much faster than traditional SQL at least because CEP (for example, Esper) will extract data only once, while in traditional SQL every query will search over and over this data again. Researchers [4] succeeded in using Esper for DIDS to detect scans, so we may conclude that it is also possible to extend task and make universal solution.

CEP perfectly fit searching for network scans or DoS application; however, using it for data mining looks not fully clear for us and further research is needed. We can assume that combining SQL database and brief results from CEP can significantly narrow search area and improve results. On the other hand, existing rules (for example from Snort ®) can be easily rewritten for CEP application, as Snort rule-based detection approach looks similar to real-time event processing.

Authors [9] give a list of data mining approaches that they infer from different research groups:

A) Feature selection.

Feature selection is selection of important parts of data and reject non-important. It can significantly limit available options what surely benefit machine learning. For example, this can be % of same service to same host, % on same host to same service or average duration / all services. Such markers can be calculated in real-time and researchers [4] showed basic examples that can be extended. The CEP problem is that it does not offer data storage. Researchers [4] used a mechanism of time windows (this is roughly the time when SEP store all your information, and after it perform search and results appear). Their test platform used value of 10 seconds [4]. Calculation of average (10Gbit/sec, 5 networks) shows that at least 62.5 Gb of RAM is needed and to solve problem with data storage they used global lists where they stored IP addresses that require future inspection. To store our values, we recommend using in-memory traditional SQL database because of vast number of parameters that need to be inspected (this can be at least 30) and quick data changes.

B) Machine learning.

Machine learning mainly consist of classification into good or malicious traffic or Clustering techniques. To apply most of methods full chain of events is needed. Network communication is time process and packets storage should be implemented. Database from Figure 3 acts as a packet storage, where the task of CEP in such process is primary filtering in order not to overload the database and searching for chains, with additional filtering performed at the core.

C) Statistical techniques

Here hidden Markov Models are used for example. Such IDS are off-line systems work on existing database of packets and are quite complex [9]. With same quality as other approaches, we see no reason to assume that they might be used in scalable, high performance DIDS, but our designed system allows performing such deep and complex analysis on the database of packets.

6 Packet operations

All operations are performed on sequences of undependable chains of packets (events). We can infer next basic operations:

- Sub data extraction
- Filtering
- Sequences searching
- Text comparison
- Full text search

Everything else (Data Mining, Patterns Search, etc.) is based on these packet operations, as at least “sub data extraction” operation can return the packet itself.

6.1 Analysis in real-time

At previous sections was described CEP function to perform basic packet analysis and limiting packets stored to the database. For example, a vast part of traffic can occupy file transactions (download, upload) that can be regarded as safe with ξ probability. ξ threshold value is determined by many factors such as network aim, application tasks, DIDS responsibilities and can be as fixed or dynamically changing. Function F (also determined) is applied to packet and ξ_{packet}= F(mixed packet data), where packet data are our packet information, like port, protocol, flags, direction and many others. Dynamic rules engine create rules that detect safe packets and delete them from flow to database insert.

6.2 Scalability

“Esper exceeds over 500 000 event/s on a dual CPU 2GHz Intel based hardware, with engine latency below 3 microseconds average (below 10us with more than 99% predictability) on a VWAP benchmark with 1000 statements registered in the system - this tops at 70 Mbit/s at 85% CPU usage. Esper also demonstrates linear scalability from 100 000 to 500 000 event/s on this hardware, with consistent results across different statements.” [8].

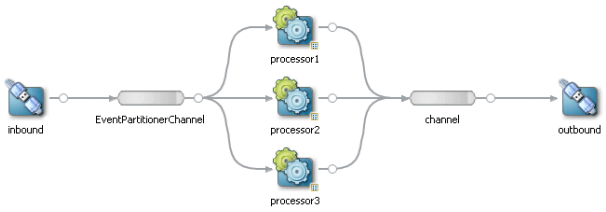

Figure 3. CEP application scalability.

Scalability of CEP based applications shows extreme flexibility and inter-independence, see figure 3. Compared to databases, where database files has to be shared, such approach shows that it is convenient and elegant way to apply parallel filtering that leads to further transmission latency fall.

7 Conclusions and Future Work

This paper present CEP based methodology united with relational SQL Database as a platform for designing different types and prototypes of distributed intrusion detection system. Despite that no line of code was presented here, we considered and examined different approaches (as real and working, as well research and planned systems) for inside IDS structure and showed how it can benefit from event orientated system. We can clearly see that DIDS processing idea and avenue mainly relies on event aggregation and including event processing component benefit the system and facilitate development. The warehouse and data-processing system is a crucial factor for DIDS performance and success. At future, we plan to implement a platform designed for transferring crucial data and events. We will use several networks with Esper&Storm to transfer traffic, parallel it and feed to several IDS for next inspection.

References

[1] Steven R. Snapp, James Brentano *DIDS (Distributed Intrusion Detection System) − Motivation, Architecture, and An Early Prototype*. In Proceedings of the 14th National Computer Security Conference

[2] Michael P. Brennan *Using Snort For a Distributed Intrusion Detection System*, SANS Institute

[3] Rogier Spoor. *A Distributed Intrusion Detection System based on passive sensors*, SURFnet, 2005

[4] Leonardo Aniello, Giorgia Lodi and Roberto Baldoni. *Inter-Domain Stealthy Port Scan Detection through Complex Event Processing*, Proceedings of the 13th European Workshop on Dependable Computing

[5] Stefano Zanero, (pdf document), 2014 http://www.blackhat.com/presentations/bh-dc-07/Zanero/Presentation/bh-dc-07-Zanero.pdf

[6] Wenke Lee, Salvatore J. Stolfo, *Real Time Data Mining-based Intrusion Detection*, In DARPA Information Survivability Conference and Exposition II, 2003

[7] Huy Anh Nguyen and Deokjai Choi, *Application of Data Mining to Network Intrusion Detection: Classifier Selection Model*, 11th Asia-Pacific Network Operations and Management Symposium, 2010

[8] Esper website: http://esper.codehaus.org/tutorials/faq_esper/faq.html#how-does-it-work-overview

[9] Theodoros Lappas and Konstantinos Pelechrinis, *"Data Mining Techniques for (Network) Intrusion Detection Systems"*, (pdf document), May 2010, online http://www.slideshare.net/Tommy96/data-mining-techniques-for-network-intrusion-detection-systems

[10] Mohamad Eid, *A New Mobile Agent-Based Intrusion Detection System Using Distributed Sensors*, (pdf document) http://www.academia.edu/2884731/A_new_mobile_agent-based_intrusion_detection_system_using_distributed_sensors

[11] Prelude-IDS web site: https://www.prelude-ids.org/