J. Basic. Appl. Sci. Res., 3(6)272-279, 2013
© 2013, TextRoad Publication
ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Enhancement Intrusion Detection using Alert Correlation in Co-operative Intrusion Detection Systems

Ali Ahmadian Ramaki[1], Reza Ebrahimi Atani[2], Reza Kazemi Iman Abadi[3], Mohsen Tavaghoe[4]

[1,2]Dep. of Computer Engineering, University of Guilan, Rasht, Iran
[3,4]Dep. of Information Technology, University of Guilan, Rasht, Iran

ABSTRACT

Increasing growth on employing computer networks services on the one hand and networks intrusion on the other hand have caused Intrusion Detection Systems (IDSs) to become a critical research subject in the area of computer systems security. To establish security in computer systems other administrations such as IDSs are required as well as firewalls and other intrusion prevention policies so as to be capable of detecting and dealing with intruders in case of breaking in through firewalls, antivirus and other security tools. The number of alerts generated by IDS, in some of cases, escalates over 2000 messages a day. A tremendous volume of alerts coupled with their low quality makes it challenging for a system administrator to handle intrusions in timely manner. It is hardly possible for systems security managers to handle such distributed alerts in order to increase their quality and convey a comprehensible report on current security state to security analyzer. One of the important approaches to handle such inefficiency is the employment of correlation of raw generated alerts by the system security sensors including IDSs. Such process aims at reduction of generated alerts as well as extraction of attacks scenario in CIDS environment. In this paper, we apply a probabilistic correlation algorithm that is works based on similarity between alerts on three standard data sets. The results indicate that the incoming alerts significantly reduced by this algorithm in rate of 99.96% on Treasure Hunt data set.
KEYWORDS: Alert correlation, alert fusion, alert reduction, CIDS environment, probabilistic correlation.

1. INTRODUCTION

In today's world computers and computer networks connected to the Internet play a big role in communications and data transmissions. At the same time, by accessing the important data of certain organizations or people, intrusion, offence or even disruption of systems routines the self-seekers are engaged with interfering with computer systems. Hackers, crackers and intruders are people who break into systems and put their security in danger. Since from a technical point of view it is practically infeasible to assemble computers (hardware or software) with no flaws in security, intrusion detection is followed with a serious attention in computer security researches.

Recently IDSs are noticeably used in computer systems to raise security. IDSs are employed to help system security administrators in order to detect intrusion and attack. An IDS goal is not attack prevention but detection of intrusions and security violations in the system or computer network and informing the system administrator [1].

In the systems, IDSs are generally applied alongside with firewalls and act as security supplementary and follow three principal tasks below:

- Supervision and evaluation
- Detection
- Reaction

IDSs, in case of encountering security events flaws, generate an alert of current state description for high-level administrators. As mentioned before, with respect to huge size of such raw alerts and their low quality as well as large quantity of out-of-place messages, security administrators probably face difficulties to handle these alerts. Correlation of generated alerts is an approach to processing such huge number of generated alerts. IDSs fall into several categories from three perspectives of detection method, architecture and how to detect intrusion. Types of intrusion detection methods include anomaly and misuse detection. There are several architectures for IDSs which are usually classified into three groups of Host-based Intrusion Detection System (HIDS), Network-based Intrusion Detection System (NIDS) and Distributed Intrusion Detection System (DIDS).

IDSs play a significant role in minimizing damages caused by various computer attacks. These systems by checking a system or computer network users' behaviors, search for intrusion indications and inform the system administrator by a certain alert message when encountering a suspicious or dangerous behavior. Correlation of alerts is a process during which alert generated by an IDS or more in the network platform is analyzed in order to achieve a brief, high-level viewpoint from probable intrusion attempts. Correlator conveys a high-level to the system administrator alert instead of generating hundreds of low-level, discrete ones by checking generated alerts and discovering their logical relations. in the next sections we

Corresponding author: Reza Ebrahimi Atani, Department of Computer Engineering, The University of Guilan, Rasht, Iran.
Email address: rebrahimi@guilan.ac.ir.

present a probabilistic alert correlation algorithm based on similarity of alerts. The end of this work is reducing the number of incoming alerts from different nodes of CIDS environment based on alert fusion and getting a concise picture of network security situation.

The rest of paper is organized as follows: at first, in the section II we study the architecture of distributed IDSs. Intrusion detection parallelization in distributed IDSs as CIDS is later discussed in the section III. The role of alert correlation in distributed environment for intrusion detection and its framework to employ them is then illustrated respectively in the sections IV and V. Correlation algorithm in co-operative environment or CIDS is discussed in section VI. In the section VII the evaluation process is described and in the section VIII the conclusion is drawn.

2. Intrusion Detection in Distributed Environment

The word intrusion is generally applied for a collection of illegal acts which puts authenticity or confidentiality of a source or accessing it at risk. Intrusions are categorized in two groups of internal and external. External intrusions are those which are made by legal or illegal individuals from the outside and internal intrusions are fulfilled by legal individuals inside the system.

Intruders usually enjoy software deficiencies, password cracking, wiretapping network traffic and flaws in network establishment, services or network computers so as to break into systems and computers.

To tackle intruders into systems and computers, several methods well-known as intrusion detection methods are offered which oversee occurred events in a system or computer network. Detection methods used in IDSs drop in two categories:

- Anomaly detection methods
- Signature-based misuse detection methods

With everyday growing network bandwidth and relentless need for processing of data in mind, IDSs in the network platform inevitably require developing commensurate with them. Intensive network-driven architectures are no longer well-qualified for today networks. Such approaches are susceptible to drops in throughput and trashing because of a unique serving center when they try to detect multi-phase attacks and maintain connections and protocols communications status quos. Intrusion detection algorithms are bound with a number of rules which are increasing fast. Creating distributed IDSs require schemes for network architecture, suitable software for distributed performances and network traffic division between parallel components. For distributed IDSs performances, two significant techniques are considered: 1) Traffic Division and 2) Load Moderation. Traffic division driven approaches mostly function on data stream, security policies and IDS structure basis towards goals listed below:

- Packets belonging to each of likely attackers reach a sensor.
- Performance and efficiency are maintained according to network speed and bandwidth.
- System accommodation to different situation.

Load moderation driven approaches count an appropriate amount of load for each sensor in a time slot such that system capacity is employed in an efficient way. Weight load is measured up in either of methods below:

1. Using load divider: the device is positioned in the network entry and whole traffic must pass through it. Consequently, it should have high performance not to make the network traffic bottleneck.
2. Each of sensors using several load divider algorithms and performing particular computations recognizes sensors overweighed by extra load and reduces their input load by a series of computations. This operation is fulfilled through several techniques such as premature filtering; which some of packets are processed in the load divider. The other technique is using a central sensor which receives messages from other sensors and recognizes extra loads on sensors, distributed or multi-phase attacks on the network. After this step, the node causes packets stream in the sensors to be dynamically adjusted only by sending control commands. These algorithms have high complicacy but noticeable performance if installed successfully. Distributed IDSs with ineffective sensors and load distribution is depicted in Fig.1.

3. Intrusion Detection Parallelization in CIDS

In general, intrusion detection parallelization for distributed systems is established in four levels: packets, protocol communication, security rules and IDS components in co-operative intrusion detection systems (CIDS) that its figure has been shown in Fig.2.

1. Packet level: in this approach a load divider, distributes loads between sensors in a circulation policy so that load balance is properly established. However, we need a component to set up stability and keep information in various streams and different protocol communications and hold complete information about every communication. Such component is called information analyzer. This component receives all data it requires by preprocessors. In such system architecture, every sensor must communicate in a proper and safe way with the analyzer. It also requires a precise installation; because it may easily serve as performance bottleneck.
2. Protocol communication level: Load distribution , in the divider, is fulfilled in a way that packets belonging to each communication reach a particular sensor, consequently, tables and connections related to each stream is demanded in sensors. In this structure, we need a component called network analyzer in order to detect multi-phase attacks and attacks on multiple communications. The so-called component must have the ability of analysis of events pertaining to different streams and creating connection between such events. This technique guarantees no fair load division.

3. Rules level: existing load in IDS is distributed between sensors. A component called traffic duplicator in the beginning of the course transfers a duplicate of the traffic to each packet and consequently, sensors are able to apply definite rules on packets. In case of proper division, rules between sensors and application of rules from a class and category to a particular sensor, detection is fulfilled appropriately and a load balance is reached. Nevertheless, each sensor must perform processing and removal of redundant packets which is followed by waste of resources.

4. Components level: IDS structural components are situated in an amount of sensors. These sensors operate autonomously and are equipped with processing components. For instance, two basic operations performed in IDSs are packets decoding, preprocessing and multi-algorithm matching. Let's say, the first sensor receives and then decodes physical layer, assembles segment packets and consequently categorizes them according to transport and network layers protocol. The second sensor performs preprocessing and repairing of transport layer streams and the next sensor applies rules and discovers attack patterns. Sending alert to database and external communications are of the fourth sensor tasks.

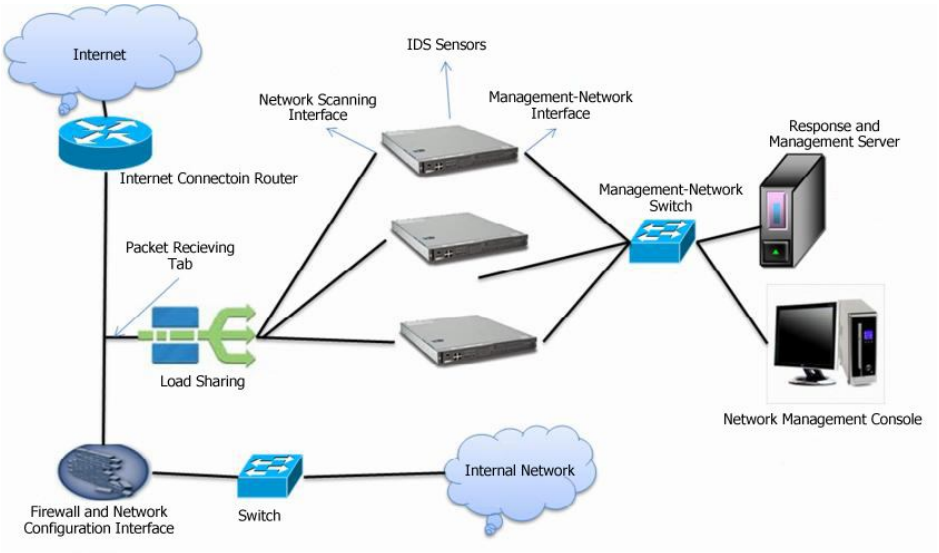

Fig. 1. Distributed intrusion detection architecture

4. Alert Correlation in Distributed Environment

IDSs play a significant role in minimizing damages caused by various computer attacks. These systems by checking a system or computer network users' behaviors, search for intrusion indications and inform the system administrator by a certain alert message when encountering a suspicious or dangerous behavior. Researches performed over the past years reveal that for some reasons e.g. large number of generated alerts and their following problems, impossibility of system security state detection and alerts low-levelness, the performance of IDSs operating individually is seldom satisfactory. Therefore, the correlation in IDSs is discussed. The correlation between these raw alerts contributes to compressing and reducing duplicate data, thus decreasing false alert rate and increasing the sensitivity of the system.

Correlation of alerts is a process during which alert generated by an IDS or more in the network platform is analyzed in order to achieve a brief, high-level viewpoint from probable intrusion attempts. Correlator conveys a high-level to the system administrator alert instead of generating hundreds of low-level, discrete ones by checking generated alerts and discovering their logical relations. Correlator usually proceeds by deleting duplicate alerts, false alerts, prioritizing alerts and discovering logical relations between alerts. Though correlation of alerts is a one-step process, the analysis is performed by several components each of which follows a special objective.

From issues and challenges giving motivation for more study and research on correlation of alerts toward intrusion detection in the network platform we may list:

- The number of generated alerts in an IDS may reach tens alerts per second. Regarding large amount of these alerts, the system administrator probably has no clear idea on system current security state. So analyzing such raw multitude alerts manually is a time-consuming process requiring many other activities. One of main solutions is the correlation of alerts in order to decrease their number and extraction of attacks scenarios.
- Some of systems do not have the necessary performance during online correlation so that by fulfilling the process online, the system performance escalates and eventually leads to a reduction in illegal intrusions.
- One of the other much important issues in this area is extraction of invaders attacks scenarios by using these weak generated alerts so that the process of correlation of alerts fulfills it by reducing their number, level of similitude and closeness of generated alerts and classifying them as larger group of attacks.

Alerts carry very useful information security teams and correlation of alerts is a beneficial process toward network security deficient events detection. The advantages include:

- Deleting manually extraction of current information of the network by administrators.
- Fixing real-time event security
- Some security events may be naturally downplayed but such low-quality alerts are parts of a main attack.
- One of the other much important issues in this area is extraction of invaders attacks scenarios by using these weak generated alerts so that the process of correlation of alerts fulfills it by reducing their number, level of similitude and closeness of generated alerts and classifying them as larger group of attacks.
- Detecting threats and events instantly and achieving a general viewpoint on system security state in every moment.

As we proceed, a general perspective on the correlation of the alert system is offered, components are enumerated and the responsibility of each of them is described in detail.

5. The Main Framework and Correlation Components

As is shown in the Fig .2, a perfect system based on correlation of alerts includes main components for normalization, pre-processing, aggregation, correlation, false alert reduction, attack strategy analysis and prioritization. As follows we describe the main task of each component for correlation of raw alerts generated and collected by supervising sensors.

- ***Normalization***: the component supplying data receives the data required for the system in an online or offline method from network threats and delivers to the normalizing component. In order to normalize and integrate the format of received alerts from sensors, we use XML format for structuring the data. In this format, some information is stored in XML for alerts from sensors such as dynamic fields like time, date, user, port, source and destination IP address and etc.
- ***Pre-processing***: normalized alerts have standard names in a certain format which are recognizable by other correlation process components. Other pre-processing components may be required since some of sensors delete some couple of fields such as start and end time, source and destination which are required by other correlating components [2]. The main task of pre-processing component is providing alerts with missing fields which are necessary for other correlating components.
- ***Aggregation***: Similar events are grouped together and the way of attacks occurrence at a certain time interval is studied.
- ***Correlation***: by combining three tasks of normalization, pre-processing and aggregation, the performance of this task escalates. There are several techniques for correlation which are addressed in the next section. The key step for selecting a method for correlation process is to consider nature of environment followed by more ability for reception of alerts, trace of tracks, preparation logs with simple entities and trace of events with such entities. The quality of correlation step also depends on level of spontaneity of tools.
- ***False Alert Reduction***: the aim of this component is to distinguish between false positive true positive alerts. Different sensors have their own pros and cons in various attacks detection and this is a famous bottleneck for low-level sensors to generate lots of false positive alerts [2].
- ***Attack strategy analysis***: the aim of attack strategy analysis is to comprehend the real intentions of invaders. The input of this component is correlated low-level alerts. The motivation for such analysis is that correlated low-level alerts probably represent the complete strategy of planned attack by invaders and is because of alerts loss by sensors. This issue leads us to attacks scenarios requiring higher-level correlation. Prediction of attacks next steps for suitable reaction against them and spontaneous response toward prevention from next damages are totally important and useful [1, 5].
- ***Prioritization***: the aim of prioritization of alerts according to severity and fitting operation against every attack type. The component of prioritization of alerts must consider types of alerts as well as other information. Security policies, network topologies, network services vulnerability analysis and installed applications and profiles evaluation are the rest of effective measures for prioritization of alerts.

The aim of correlation component is to discover relations between alerts in order to reconstruct the attack scenarios according to isolated alerts [10]. Though the correlation may not have any certain efficiency in alerts reduction, the responsibility of the correlation of alerts component in providing a high-level outlook on real attacks is considerably crucial. The performed works in the area of alerts correlation are suggestive of the four classifications below to fulfill the process and analyze the normalized and aggregated raw alerts from previous steps [6, 7, 8, 9]:

- ***Scenario-based correlation***: in this class, the relation between alerts is recognized according to scenarios. In the other words, if alerts are able to form an attack by combination they are connected.
- ***Rule-based correlation***: in this class, the relation between alerts is selected as a rule. In the other words, alert A connects to alert B if the first alert to the second one is a prerequisite - post condition relation.
- ***Statistical correlation***: the methods of this class connect two alerts if the alerts ate statistically connected.
- ***Temporal correlation***: in such correlation, two alerts are connected according to temporal relations.

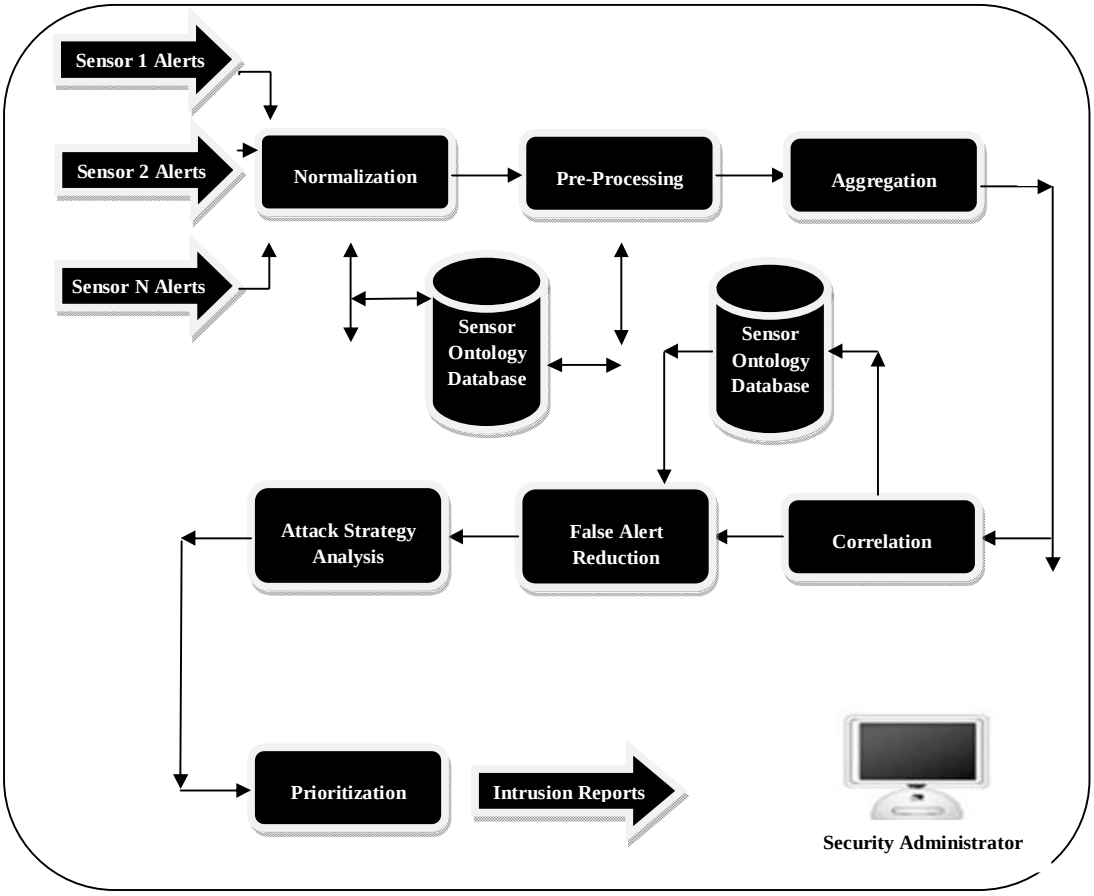

Fig. 2. The main components of correlation process

6. Correlation Algorithm in Co-operative Environment

Co-operative Intrusion Detection System (CIDS) architecture for security events detection is depicted in the Fig.3 in which every sensor is equipped with intrusion detection components that generated alerts of every sensor forwarded to a centralized correlation unit to correlation process [11].

The algorithm introduced in the Fig. 4 illustrates as how to correlate alerts generated by distributed sensors and assemble them in order to detect security events and multi-phase attacks. Symbols d_i, P_{ij}, r_i, n_{ij},srcPrt ,dstPrt and ℓ respectively represent the number of distributed elements in distributed environment, logged suspicious behavior in an entry of lattice, raw alert received from element i, a number of raw alerts for the pattern P_i reported by IDS with number d_i, source port address, destination port address and a number of elements in an attack scenario detection. This algorithm, in different time slots Δ, has a local threshold σ_l and a general threshold $\sigma_g$for each security event so that this parameter for every security event is stored in local database belonging to each element. Distributed elements collect raw alerts locally and deliver to the main system. The main subsystem correlates raw alerts that it receives from each element and then filters valueless and inappropriate ones. It finds the destination of a security event detected and examines the general status. Then it forms the network behavior for each security event received from the previous stage and stores the patterns.

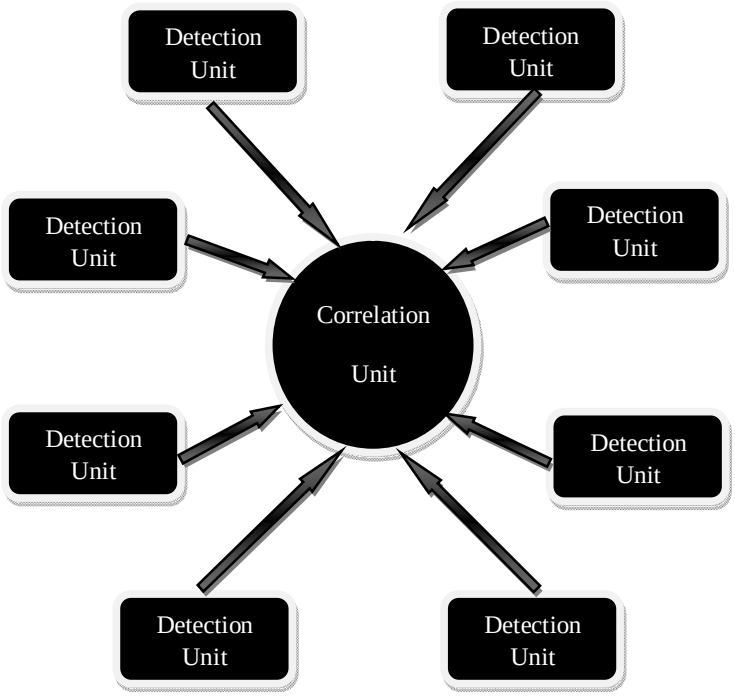

Fig. 3. Centralized CIDS architecture

```
1  Detection system di
2  //σl is the local correlation threshold, σg is the global
                        correlation threshold
3  for each time interval Δ do
4    correlate raw alert ri locally
5    // LAi: local alert report on di
6    LAi ← correlate-and-filter (ri, σl)
7    for each pij∈ LAi do
8      // look up the destination node for pij
9      dt ← look up (srcIP of pij)
10     subscribe (pij, nij, di) on dt
11   end for
12 end for
13 //Subscription and notification process on dt
14 while message received do
15   if message == subscribe (pij, nij, di) then
16     //re-construct the index lattice
17     lattice ← add-to-lattice (pij, nij, di)
18     if last subscription within this time interval then
19       //Subt: subscription pattern list on dt
20       Subt ← correlate-and-filter (lattice, σg)
21       for each pij∈ Subt
22         for each dk that has subscribed to pij do
23           notify (dk, pij, |pij|, |ℓ|)
24         end for
25       end for
26     end if
27     else if message == notify (dk, pij, |pij|, |ℓ|) then
28       pij is confirmed as an attack instance
29     end if
30 end while
```

Fig. 4. General operation of CIDS

For every network in the desired time slot, with regard to creating threshold and detecting similitude of received network detection is sent. CIDS algorithm is shown in Fig. 4.

7. Experimental Results

In the following, the applied algorithm for alert correlation that is similarity based is described. This algorithm works based on alert fusion. Then, the algorithm is implemented on tree standard data sets. The effect of alert correlation is explained by alert features similarity method. With the help of that, the volume of the alerts is reduced[12].

In the mentioned centralized method, for every input alert, the attributes stated in Table 1, are logged. By applying IDMEF format for all receiving alerts, similarity based algorithm is used on theme. Here by a concise picture of network security is achieved for network administrators.

Table 1. The alert log attributes

Feature	Feature	Feature
messageId	attackClass	destPort
creationTime	srcIp	protocolType
detectTime	srcPort	portList
analyzerName	destIp	protocolType
startTime	endTime	sensorNode

The applied correlation algorithm for input alerts is of the probabilistic type. This correlation does alert fusion based on attributes similarity of two different alerts. By using the values recorded for each alert features, with equation 1, the similarity rate of two different alerts is calculated. If this rate is greater than a minimum similarity, two alerts are fused and a meta alert is obtained. The similarity of two desired alerts is a number between 0 and 1.

$$SIM(X, Y) = \frac{\sum_j E_j SIM(X_j, Y_j)}{\sum_j E_j} \qquad (1)$$

X= Candidate meta alert for matching
Y= New alert
j= index of the alert features
E_j= Expectation of similarity for feature j
X_j, Y_j= Values for feature j in alerts X and Y, respectively.

Each input alert from every IDS component in CIDS environment is compared with a previous meta alert list that is gathered. Based on this similarity criterion, the new meta alert is created. If the similarity rate between new incoming alert and meta alerts, the new alert is stored alongside the other meta alerts as a new meta alert. The results of applying the correlation algorithm to three main standard data sets (MIT/LL 2000 [13], CTV [14], and Treasure Hunt [15]) are depicted in Table 2. The results indicate that the incoming alerts significantly reduced in rate of 99.96% on Treasure Hunt data set.

Table 2. Reduction ratio of alerts based on correlation algorithm

	MIT/LL 2000	CTV	Treasure Hunt
Input alerts	36,631	215,113	2,808,595
Output alerts	7,985	142,822	1,080
Reduction Ratio	79.22%	43.53%	99.96%

7. Conclusion

In this paper, similarity based correlation of alerts in co-operative intrusion detection system is offered for understanding the security situation of a protected network and applied to three main datasets. Generated alerts in computer network environments are detected and logged by security establishment components over the network such as IDSs. After normalization of the data related to alerts and pre-processing operation on these raw alerts data, these alerts are correlated according to correlation techniques towards quality increase of generated raw alerts, attack strategy characteristics with an attack pattern, a proper alert for attack detection of invaders and reaching a general outlook from system security state. After the correlation of alerts, the summarized and high-quality alerts demonstrate the system security state truly and convey the probable alerts to the system high-level security administrator(s). Correlating process of raw alerts that generated by distributed IDSs is an efficient task that improves the ability of security events detection. In distributed environment correlation method selection is very important that depends on network data streams. In this case, the results indicate that the incoming alerts significantly reduced in rate of 99.96% on Treasure Hunt data set.

REFERENCES

1. Ning, P., Xu, D., Healey, C.G. and Amant, R.A., Building Attack Scenario through Integration of Complementary Alert Correlation Method, In the Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS'04), p. 97-111, 2004.

2. Valeur, F., Vigna, G., Kruegel, C., and Kemmerer, R.A., A Comprehensive Approach to Intrusion Detection Alert Correlation, IEEE Transactions on Dependable and Secure Computing, p. 146-169, July 2004, 1(3).

3. Manganaris, S., Cheristensen, M., Zarkle, D. and Hermiz, K., A Data Mining Analysis of RTID Alarms, The International Journal of Computer and Telecommunications Networking, p. 571-577, October 2000, 34(4).

4. Kendall, K., A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, Proceedings DARPA Information Survivability Conference and Exposition (DISCEX'99), p. 12-26, 1999.

5. Pietraszek, T., Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection, In the Proceedings of 7th International Symposium, RAID 2004, p. 102-124, Sophia Antipolis, France, 2004.

6. Sadoddin, R., Ghorbani, A., Alert Correlation Survey: Framework and Techniques, In the Proceedings of the 2006 International Conference on Privacy, Security and Trust, NY, USA, 2006.

7. Cuppens, F. and Ortalo, R., "LAMBDA: a Language to Model a Database for Detection of Attacks", In the Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection, p. 197-216, London, UK, Springer-Verlog, 2000.

8. Cuppens, F. and Miege, A., Alert Correlation in a Cooperative Intrusion Detection Framework, In the Proceedings of the 2002 IEEE Symposium on Security and Privacy, p. 202-215, Washington, DC, USA, IEEE Computer Society, 2002.

9. Qin, X., A Probabilistic-Based Framework for INFOSEC Alert Correlation, PHD Thesis, Georgia, Institute of Technology, 2005.

10. Qin, X. and Lee, W., Statistical Causality of INFOSEC Alert Data, In the Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID'03), p. 73-93, 2003.

11. Zhou, C.V., Leckie, C. and Karunasekera, S., Decentralized Multi-dimensional Alert Correlation for Collaborative Intrusion Detection, Journal of Network and Computer Application, p. 1106-1123, September 2009, 32(5).

12. Valdes, A. and Skinner, K., Probabilistic Alert Correlation, In the Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID'00), p. 54-68, London, UK, 2001.

13. MIT Lincoln Laboratory, Lincoln Lab Data Sets, http://www.ll.mit.edu/IST/ideval/data/data_index.html, 2000.

14. Haines, J., Ryder, D.K., Tinnel, L. and Taylor, S., Validation of Sensor Alert Correlators, IEEE Security and Privacy Magazine, pp. 46-56, January 2003, 1(1).

15. UCSB Reliable Software Group, Collection of Intrusion Detection Datasets, http://www.cs.ucsb.edu/rsg/datasets/, 2004.