A Survey on IDS Alerts Classification Techniques

Shashikant Upadhyay
PG Scholar
VNSFOE,
Neelbud, Bhopal, India

Rajni Ranjan Singh
Assistant Professor
VNSFOE,
Neelbud, Bhopal, India

ABSTRACT

Intrusion detection can be defined as the method of identifying malicious activities that target a network and its resources. The main use of intrusion detection systems (IDS) is to detect attacks against information systems and networks. A main difficulty in the field of intrusion detection is the organization of alerts. Normally IDS's produced numerous alerts, which cannot provide a clear idea to the analyst about what type of alert occur, which type of alert is generated etc. because of the huge number of alerts generated by these systems. One solution of this problem is classifying the alerts. During this paper, we try to represent an overview of IDS alerts classification techniques.

Keywords

Alert Correlation, Classification technique, Intrusion Detection system, Cyber Attack.

1. INTRODUCTION

World Wide Web (WWW) plays vital role in today routine life. WWW is employed in number of applications such as social networking, business, education, shopping etc. Due to this risk of Network systems connected to the web becoming targets of intrusions by cyber criminals. Cyber crooks violence systems to improvement unauthorized access to information, waste information or to decrease the obtainability of information to authorized handlers. This results in vast economic losses to companies also down their goodwill to clients. Intrusion prevention methods such as user authentication (e.g. using a password or biometrics), information protection (like encryption), avoiding programming mistakes and firewalls have been used to protect network systems. But, unluckily these intrusion avoidance techniques alone are not enough. Because of design and programming flaws in operating systems, protocols and application programs number of faults of the system might be unrecognized. That's why, we require a scheme to discover intrusions as soon as possible and take suitable actions [1]

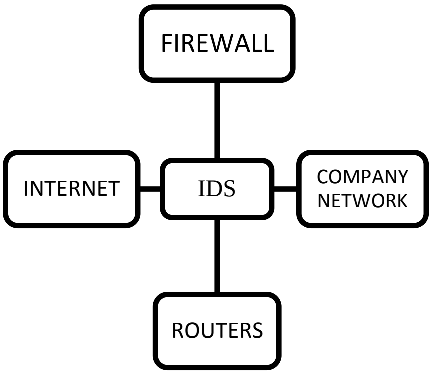

Fig.1 Intrusion detection systems Model

The reminder of the paper is as follows: Section 2 explains attack categories, intrusion detection techniques and general working theory of intrusion detection systems. In section 3 we give details of Alert Processing and Alert Correlation Model of intrusion detection system. Section 4 discuss about Requirement for post-processing of intrusion detection alerts. Section 5 presents detailed analysis of early important work on post-processing of intrusion detection alerts. The final conclusion is mentioned in section 6.

2. CYBER ATTACK CATEGORIES

2.1. Attack Categories

According to the classification proposed by Kendall [2], attacks can be categorized into subsequent four groups:

2.1.1. Denial of Service (DoS)

The Denial-of-Service (DoS) or Distributed Denial-of-Service (DDoS) is an attack wherein the attacker attempts to formulate a computer resource too busy or too full to respond to its intended users. Prototypes of these attacks contain Teardrop, Smurf Back, Neptune, Ping of death Land etc. [2]

2.1.2. User to Root (U2L)

A User to Root (U2R) is an attack that tries to get super user access to the system. Attacker gets super user access via flaw in application software or operating system. Attacker opens out with access to a public user account on the system and is capable to exploit some weakness to gain root access to the Most common attack during this category of attack could be a buffer overflow attack. Other attacks include Perl, Ps Loadmodule, and Xterm etc. [2]

2.1.3. Remote to User (Local) (R2L)

A Remote to local (R2L) is an attack where the attacker efforts to get illegal access from a remote machine into super user account of the victim system. In this type of attack, the attacker sends packets to a machine over a network then exploits some vulnerability to achieve local access as a user of that device. Samples of remote to user attack are Dictionary Guest, Imap, Phf and Ftp_write etc. [2]

2.1.4. Probing

Probing is an attack where the attacker checks a network of computers to accumulate information or discover known flaws. The attacker who recognizes which machines and services are accessible on network can use this information to look for flaws. Attacker uses this information to plot upcoming attacks. Here a variety of tools presented for probe attack which can be used by even a much untrained attacker. Examples of these attacks are Mscan, Nmap, Saint, Ipsweep, Satan etc. [3]

2.2. Intrusion Detection Techniques

Intrusion detection is the way of discovering actions that attempts to compromise the secrecy, honesty and accessibility of a resource. Established on analysis approach intrusion detection system could be classified as follow [1] [4]:

Anomaly Detection: - it is a main tool for finding fraud, network intrusion, and other few events that may have great significance but are hard to find.

Misuse Detection: - it is an approach in detecting attacks. In misuse detection approach, we explain the abnormal system response at first, and then explain any other response, as normal response.

2.2.1 Type of IDS based on Deployment

Host-Based IDS: - The host-based intrusion detection system (HIDS) is an intrusion detection system that checks and examines the internals of a computing system in addition to the network packets on its network interfaces.

Network-Based IDS: - The network-based intrusion detection system (NIDS) is applied to monitor and study network traffic to secure a system from network-based threats.

2.3 Some other Terms

2.3.1 Signatures

Signature is understanding form which we appear for inside a data packet. A signature is identifying to detect one or multiple types of attacks. For example, the event of “scripts/admin” in a packet going to our web server may show an intruder activity. [5]

2.3.2 Alerts

Alerts are any type of user report of an intruder activity. Once IDS finds an intruder, it has to inform the security controller about these suspicious activities by generating alerts. These alerts have many form of as logging to a console, automatic open windows, sending e-mail etc. [5]

2.3.3 Logs

The log messages are usually unbroken on file. Normally Snort keeps these messages under /var/log/snort directory. However, the situation of log messages is modified victimization the command switch once starting Snort. Log messages saved either in text or binary format. [5]

2.3.4 False Alarms

False alarms are alerting made thanks to a clue that\'s not associate intruder activity. As an example, misconfigured internal hosts could generally broadcast messages that trigger a rule leading to the generation of a false alert. [5]

2.3.5 Sensor

The machine on that associate intrusion detection system is running is additionally known as the sensing element within the works as a result of it\'s accustomed “sense” the network. [5]

2.4 Working Principle of Intrusion Detection Systems

Following four phases are proposed for general working of IDS by authors of [4].

2.4.1. Data Gathering

In data gathering useful data is collected for intrusion detection. In Network-Based Intrusion Detection network traffic is collected with the help of sniffer software like TCPDUMP. On behalf of host-based intrusion detection data for example disk usage, process activity, memory usage and system calls are collected. Commands like netstas, ps and strace will be used for this purpose.

2.4.2. Attribute selection

The organized data are substantially larger and can\'t be used as it is, therefore set of this information is selected by making feature vectors that contain only necessary data required for intrusion detection. In network primarily based intrusion detection, it may be IP packet header data which have source and destination IP addresses packet length, layer four protocol type and different flags. In host-based intrusion detection it contains user name, login time and date, amount of sessions and range of opened files.

2.4.3. Analysis

The collected data are analysed in this step to determine whether the data are anomalous or not. This is the main research area where many methods have been proposed and used to detect intrusion.

2.4.4. Action

IDS alerts the system administrator that an attack has occurred using numerous methods like e-mail, alarm icons imagining techniques. IDS can similarly break or manage attack by closing network ports or killing processes.

3. ALERT PROCESS

Alerts are any kind of user warning of an unknown person activity. Once an IDS finds this type of activity, it\'s to inform the security superintendent about these using alerts. Alerts can be in the type of sending email, automatic windows, etc. An event is a low level entity that is analyzed by the IDS, whereas an alert is produced by the IDS to notify parties of interesting events.

A single event can cause several alerts (that is a problem) mainly in a networked IDS environment, and a single alert can define a set or sequence of events [6]. Each alert is doubtful but an event is not necessarily doubtful. An alarm is the user interface mechanism by which a user manages an alert [7].

The organization of alert processing techniques is shown in Fig. 2. They can be classified into dual categories:

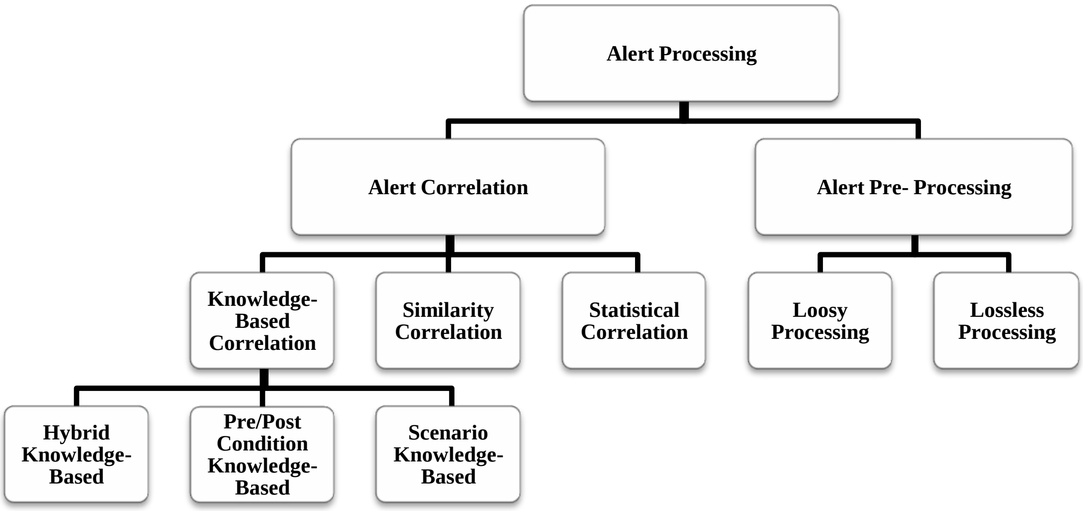

Fig.2 Organization of alert processing techniques

Alert Pre-Processing and Alert Correlation. The Alert Pre-Processing works at the network layer whereas the attacker is at the application layer, so we need a method of alerts in the application layer that is Alert Correlation. The following two subsections define these two categories in more details.

3.1 Alert Pre-Processing

This type of the alert processing goes to moderate the effect of false alerts and to make the next step (i.e. correlation process) more correct. There are various procedures in this class of processing, all of which try to eliminate the noise from the stream of alerts and make it more meaningful. These methods either lose some information (because it focuses on some events that occurred or not) or don't lose any information (because it uses additional knowledge). It is stated in the following two subgroups.

3.1.1 Loosy Pre-Processing

The main techniques of this type of processing are the alert prioritization and alert aggregation, both of them try to reduce alert flooding and they are always used as components in the systems. Alert prioritization is performed to evaluate the relative importance of alerts produced through the sensors. This scheme has to take into reasons the security policy and the security requirements of the site where the correlation system is deployed [8]. The importance of an alert can depend on many features such as Cause/effect criticality, Attack criticality, and Alert confidence. Therefore, selecting of alerts aids in substantial reduction of alert volume [9].

3.1.2 Lossless Pre-Processing

Sometimes this type of technique is called filters and it generally uses rules to filter the alerts. These rules are built either by experts or by automated programs. It is designed to eliminate the false alerts that make the correlation process incorrect. In this paragraph, we examine three of these methods that are Alert Verification, Root Cause Discovery and Machine Learning.

Julisch [10] presents methodology focuses on locating the main reasons for large groups of alarms, which generally correspond to issues within the computing infrastructure that leads to many false positives (with the potential exception of large-scale automated attacks). It doesn't search for little, stealthy attacks in the alarm logs, however aims to reduce the noise in the raw alarms to make it simpler to identify real attacks in the resulting analysis. This method is used to remove the false alerts generated from misconfigured equipment. But, in the small networks this method is useless because it is easy to configure all equipment in the network. However in the large networks it is a hard task to configure all equipment well so this method will be useful. Also, the written filters must be held in reserve secret because the attacker may use it to avoid detection.

3.2 Alert Correlation

Alert correlation is a very important technique for managing larger the quantity of intrusion alerts that are raised by heterogeneous Intrusion Detection Systems (IDSs). Alert correlation algorithms may be divided into three classes supported their characteristics:

1) Similarity-based, 2) Knowledge-based and 3) Statistical-based [11].

The similarity-based and statistical-based algorithms need less context data and that they are able to correlate only supported similarities between alert features and learned data from previous steps whereas knowledge-based algorithms completely perform based on alert meanings. It's to be familiar that this categorization isn't completely precise and a few algorithms are on the sting between two classes. Thus, distribution and algorithmic to a class is based on the actual fact that the algorithm has the foremost similarity to that one.

3.2.1 Similarity-based Correlation

Three main subclasses are supposed for these types of processes. The first subclass is based on defining very simple rules for expressing relations between alerts. The second subclass is presented with the goal of identifying basic drawbacks in the network structure. The third subdirectory includes processes which produce comparison factors using models based on machine learning. In the following subsections, different researches in each subcategory will be described.

3.2.2 Knowledge-based Correlation

The most common methods in the field of correlation are the knowledge-based methods. They can be separated into three groups as shown in Fig. 2. They differ in the required type of knowledge. The following three subgroups will present them in some detail.

3.2.2.1 Scenario Knowledge-base Correlation

This class requires attack scenario knowledge such as the work of Dain et al. [12]. They use an alert clustering scheme that fuses alerts into scenarios using a "probabilistic method." Here, situations are created as they occur, i.e., whenever a new alert is received it is compared with the current existing scenarios and then assigned to the scenario that yields the highest probability score. If the score falls below a threshold, it starts its own scenario. This testing is done in a time proportional to the number of candidate scenarios.

In the same way to Valdes et al. [13] work, this process maintains a continuously updated list of alert groups called scenarios. The task of an alert to the scenario is final and irreversible. But, unlikely, in which the similarity is calculated based on set-valued attributes, in this method the probability score is a function of a new alert and only the last alert in the existing scenario.

3.2.2.2 Pre/Post Conditions Knowledge-base Correlation.

Ning et al. [14] suggest an alert correlation model created on the essential observation represent most intrusions include of various phases, with the early phase preparing for the later ones. The connection model is based upon two features of intrusions that are, Prerequisites (the necessary conditions for an intrusion to be successful) and Consequences (the possible outcomes of an intrusion). With knowledge of prerequisites and consequences, the correlation model can classify related alerts by discovering causal connections between them, i.e., with matching the implications of previous alerts with conditions of later ones.

The approach proposed by Cuppens and Miège in [15] also uses pre/post-conditions. In addition, it includes a number of phases including alert grouping, alert inclusion, and intention recognition. In the first two phases, alerts are clustered and merged using a parallel function. The purpose detection phase is referenced in their model, but has not been implemented. An interesting aspect of this approach is the attempt to generate correlation rules automatically. While it may seem appealing, this technique could generate a number of spurious correlation rules that, instead of reducing the number of alerts and increasing the abstraction level of the reports, could introduce the correlation of alerts that are close or similar by pure chance, in this way increasing the noise in the alert stream.

3.2.2.2 Hybrid Knowledge-Base Correlation

This form of correlation techniques tries to use most of the available information to leverage correlation reliability. An interesting and active method was proposed by Lingyu et al. [16] to correlate alerts and hypothesize the missed ones. The information used in this method is weaknesses, their needs, and network connectivity. The first step in this method is to build attack graph AG from the previous information. At that point another strategy, specifically queue graph QG, was proposed to correlate alerts in real time depending on AG and exploits. They show that this method can process alerts faster than an IDS can report them.

3.2.3 Similarity-based Correlation

The basic idea of these procedures is that relevant attacks have similar statistical attributes and a proper classification can be found by detecting these similarities. These types of procedures store causal relationships between different incidents and analyses their occurred frequencies in the system education period using previous data statistical analysis and then attack steps are generated. After learning these relationships and being confirmed by the supervisor, this learning is utilized for correlating distinct attack stages. Pure statistical procedures do not have any prior knowledge on attack scenarios. But the scientific results indicate that using these procedures is possible only in very specific domains in which domain attributes are taken into account in designing procedures and then, high fault rate exist. Furthermore, merging data using this procedure is impossible if the previous sensors provide incomplete or abnormal information. This type is also divided into three subsections. The first subsection's goal is to detect alerts which are regularly repeated and finding their repetition pattern. The purpose of the second subsection is estimating causal relationships between alerts, predicting the next alert occurrence, and detecting attacks and the third subsection's goal are merging reliability with completely similar alerts.

3.3 Alert Correlation Model

Fig. 3 is composed of ten components: normalization, pre-processing, prioritization, alert confirmation, alert mixture, focus recognition, shushing the alerts, multi-step correlation, intention recognition, and impact analysis [17].

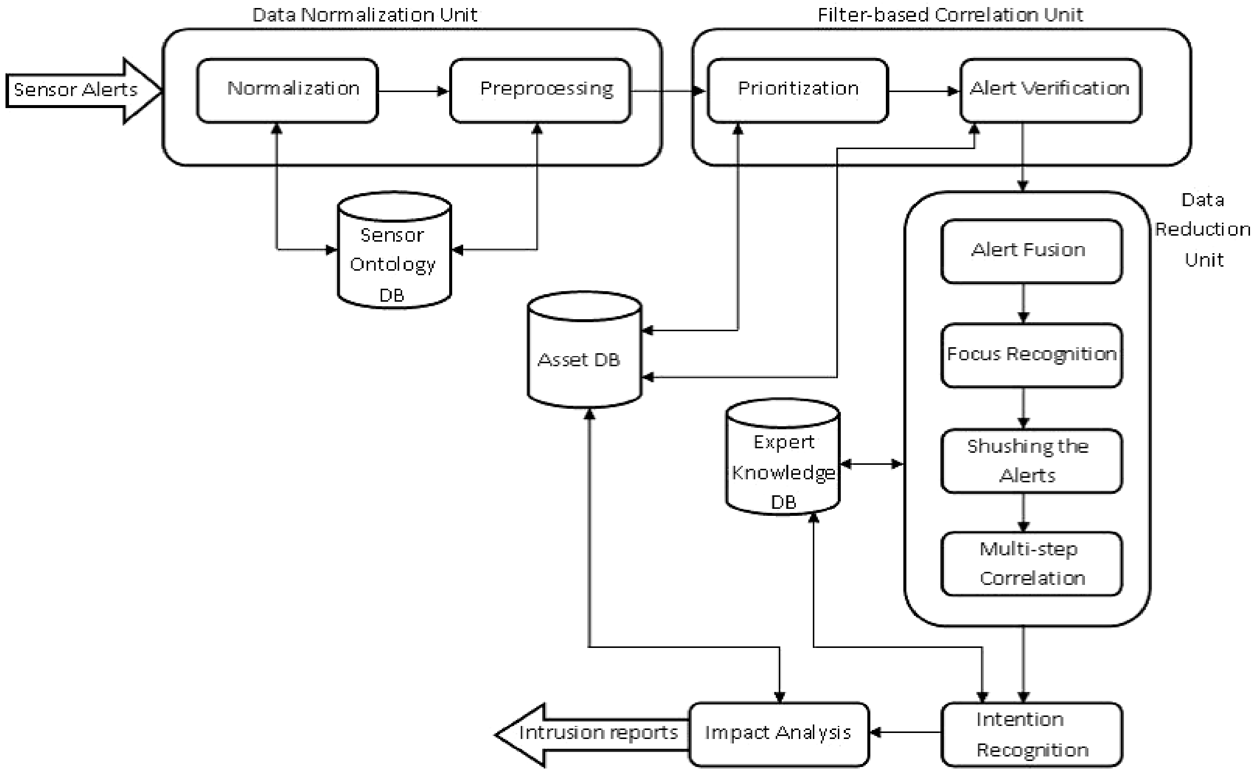

Fig.3 Alert Correlation Model

4. THE REQUIREMENT FOR POST-PROCESSING OF INTRUSION DETECTION ALERTS

Once intrusion detection research developments and the first real world implementations of such systems were installed, the problem of the low quality of the resulting alert-set became marked. In theory intrusions were identified at a high rate, but the alert sets created by the intrusion detection systems were incorrect for use in an environment where instant reaction is critical. The main aspects of this problem of low quality of alerts are:

• **High volume:** Generally, intrusion detection systems protect complex systems and networks, whose utilization is relatively high; thus, high volumes of data (network traffic or system calls) are examined for possible malicious activity.

This produces large volumes of alerts. In the majority of the cases it is impossible for the analyst to read a real world alert-set in an alert by alert fashion, as alerts are created at a rate much higher than the rate in which she can read them.

• **High false positives rate**: Aside from being huge in volume a real world alert-set comprises essentially of false alarms, i.e. alarms that don't compare to true interruption occurrences. This commonly happens because intrusion detection systems attempt to get high detection rates (percentage of true intrusions detected), so their sensitivity is set at relatively high levels.

• **Low level of alerts:** Alerts relate to low level events in a system (e.g. to an IP packet or to a system call). Tried intrusions are higher level events and they generally produce multiple different alerts. This change in the level between events and alerts makes it tough to infer useful information when reading an alert set. In order to reduce these deficiencies, several researchers have employed methods of post-processing intrusion detection alerts. These methods fall into three main methods; reduction of false positives, clustering, and alternative representation techniques.

5. EARLY IMPORTANT WORK ON POST-PROCESSING OF INTRUSION DETECTION ALERTS

In the last two decades some serious research work on post-processing alerts has been conducted quality. In this Sector the most important research efforts in the field are presented.

5.1 Defining Alerts Similarity

In 2001 Valdes and Skinner proposed using probabilistic similarity between alerts as a means to post-process those [18]. In this approach alerts for which there is a relevant match are aggregated. For each different alert attribute an appropriate similarity function is defined. Additionally, an expected similarity value is calculated, which in practice is a weight that is later used to calculate the overall similarity. A minimum match specification is also incorporated, that unconditionally rejects a match if any feature similarity is lower than the minimum specified value. For every new alert, the similarity to all current meta alerts is computed taking into account attribute similarities along with the corresponding expected similarities. The alert is then merged with the best matching meta alert, as long as their similarity is above a threshold value. The concept of combining the results of similarity functions for each attribute of alerts, to calculate an overall similarity has influenced the work of other researchers [19], [20], and [21].

5.2. Discriminating between Aggregation and Correlation

At about the same time, Debar and Wespi [22] presented the first analytical descriptions of alert aggregation and correlation procedures. They highlight the most important problems in intrusion detection alert-sets as:

- Intrusion detection systems offer the operator with a huge number of alerts; the operator then has problems coping with the load.
- Attacks are likely to produce multiple related alerts and it is not easy for operators to logically assemble them.
- Intrusion detection systems are likely to generate many false alerts, false positives or false negatives.
- Intrusion detection system architectures, at the time, made it difficult to achieve large-scale deployment.

5.3 Improving the Alert Quality

Bakar et al [23] Major contributions have been made towards improving the quality of alerts in order improve the identification of intrusions in computer networks. As noted by many researchers, improving the quality alerts contribute to reduction of huge voluminous alerts that security Analysts have to evaluate when identifying true alerts.

5.4 Use of Supporting Evidence

To improve the quality of alerts according to Kruegel [24], several approaches use supporting evidence such as vulnerability assessment data, logs and alert contexts. The additional information provides the basis of comparing or matching the alerts [25]. Approaches based on this principle of using additional information help to manage alerts in a better orderly way. Thus the additional information provides solid evidence and indicator of what is happening in the network. It is the surest practical way of ensuring the Security Analysts are dealing with the necessary alerts. Without additional information, it is not possible to decide with certainty whether an alert is true or false.

5.5 Alert Clustering

According to Pietraszek [26], there are many different data mining techniques for cluster analysis and the suitability of the different methods strongly depends on the area of application and its properties. Pietraszek work [26] is based on the modified Attribute-Oriented Induction described by Julisch [27]. He made the following interesting observations:

- Large groups of alerts have a common root cause.
- Few of these root causes account for a large volume of alerts.
- These root causes are relatively persistent over time.

6. CONCLUSIONS

This paper has presented a survey of the various IDS Alerts Classification Techniques. There are many conclusions are drawn from this survey like Alert Pre-Processing techniques use to eliminate the noise from the stream of alerts and make it more meaningful. But it suffers some problems. Alert prioritization and alert aggregation, equally try to decrease alert flooding and they are always used as elements in the systems. Alert aggregation is mainly aimed at mitigating alert flooding. However, the generated alert may not hold information such as arrival time of each alert that was initially known before aggregation. From another point of view, we can call the Knowledge-based correlation as a misuse correlation because this type of correlation matches the alerts with a prior knowledge and search for fixed patterns of alerts (like misuse IDSs). The challenge is to achieve high alert classification rate and reduce false alarm rate. During the survey, we also find some points that can be further explored in future, such as finding some effective security solutions and protecting the alert classification based Intrusion Detection System (IDS). Keeping that in view here, we have prepared an effort to review the familiar alert processing methods. Comparison of several methods is made for demonstration the strength and weakness of these methods. We hope this survey will be useful for researchers to carry forward research on system security for designs of a correlation system that not only will have recognized strengths but also overcome the problems.

7. REFERENCES

[1] Lee, W., & Stolfo, S. (1998), “Data mining approaches for intrusion detection,” In Paper presented at the proceedings of the seventh USENIX security symposium (SECURITY’98). San Antonio, TX.

[2] K. Kendall, “A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems,” Massachusetts Institute of Technology Master's Thesis, 1998.

[3] Mohammad Sazzadul Hoque1, Md. Abdul Mukit2 and Md. Abu Naser Bikas3,” An Implementation of Intrusion Detection System Using Genetic Algorithm”, International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012.

[4] Mittal, Mitali, Alisha Khan, and Chetan Agrawal. "A Study of Different Intrusion Detection and Prevension System" International Journal of Scientific & Engineering Research, Volume 4, Issue 8, August-2013. pp. 1526-1531

[5] Intrusion detection FAQ at http://www.sans.org/newlook/resources/IDFAQ/ID_FAQ.html

[6] Hätälä A., Särs C., Addams-Moring R., Virtanen T., Event data exchange and intrusion alert correlation in heterogeneous networks, Proceedings: 8th Colloquium for Information Systems Security Education West Point, NY, 2004, pp. 84-92.

[7] Sun B., Wu K., Pooch U. W., Alert aggregation in mobile ad hoc networks, ACM WiSE’03, San Diego, California, USA, Sep. 2003.

[8] Valeur F., Vigna G., Kruegel C., Kemmerer R. A., Comprehensive approach to intrusion detection alert correlation, IEEE Transactions on Dependable and Secure Computing 1(3), 2004,pp. 146-169.

[9] Siraj A., A unified alert fusion model for intelligent analysis of sensor data in an intrusion detection environment, Ph.D. thesis, Mississippi State University, Aug 2006.

[10] Julisch K., Mining alarm clusters to improve alarm handling efficiency, Proceedings: 17th Annual Computer Security Applications Conference (ACSAC'01), New Orleans, LA, Dec 2001.

[11] Al-Mamory, S. O., Zhang, H.: A survey on IDS alerts processing techniques. In: Proceeding of the 6th WSEAS International Conference on Information Security and Privacy (ISP), pp. 69-78 (2007).

[12] Dain O. M., Cunningham R. K., Building scenarios from a heterogeneous alert stream, IEEE Transactions on Systems Man and Cybernetics, 2002.

[13] Valdes A., Skinner K., Probabilistic alert correlation, Proceedings: Recent Advances in Intrusion Detection, LNCS 2212, 2001, pp. 54-68.

[14] Ning P., Reeves D., Cui Y., Correlating alerts using prerequisites of intrusions, technical report TR-2001-13, Department of Computer Science, North Carolina State University,2001.

[15] Cuppens F., Miège A., Alert correlation in a cooperative intrusion detection framework, Proc. IEEE Symposium, Security and Privacy, May 2002.

[16] Wang L., Liu A., Jajodia S., Using attack graphs for correlating, hypothesizing, and predicting intrusion alerts, Computer Communications 29, Apr 2006, pp. 2917- 2933.

[17] Elshoush, H. T., Osman, I. M.: An Improved Framework for Intrusion Alert Correlation. Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering 2012, WCE 2012, 4-6 July, 2012, London, U.K., pp. 518-523.

[18] A. Valdes and K. Skinner, “Probabilistic alert correlation,” in Recent Advances in Intrusion Detection (RAID 2001), ser. Lecture Notes in Computer Science, no. 2212. Springer-Verlag, 2001.

[19] H. Ren, N. Stakhanova, and A. Ghorbani, “An online adaptive approach to alert correlation,” in Detection of Intrusions and Malware, and Vulnerability Assessment, ser. Lecture Notes in Computer Science, C. Kreibich and M. Jahnke, Eds. Springer Berlin Heidelberg, 2010, vol. 6201, pp. 153–172.

[20] K. Julisch, “Clustering intrusion detection alarms to support root cause analysis,” ACM Trans. Inf. Syst. Secur., vol. 6, no. 4, pp. 443–471, Nov. 2003.

[21] S. Lee, B. Chung, H. Kim, Y. Lee, C. Park, and H. Yoon, “Real-time analysis of intrusion detection alerts via correlation,” Computers & Security, vol. 25, no. 3, pp. 169 – 183, 2006.

[22] H. Debar and A. Wespi, “Aggregation and correlation of intrusion-detection alerts,” in Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection, ser. RAID ’00, 2001, pp. 85–103.

[23] Najwa A. Bakar, Bahari Belaton (2005). Towards Implementing Intrusion Detection Alert Quality Framework. Proceedings of the first International Conference on Distributed Framework for Multimedia Applications 2005. IEEE.

[24] Kruegel C., Roberstson W., Vigna G. Using Alert Verification to Identify Successful Intrusion Attempts. In: PIK 27 (2004).

[25] Xuejiao Liu, Debao Xiao, Xi Peng. Towards a Collaborative and Systematic Approach to Alert Verification. In: Journal of Software, Vol. 3, No. 9, December 2008. pg. 77-84.

[26] Pietraszek T. (2004). Using adaptive alert classification to reduce false positives in intrusion detection. Recent advances in Intrusion detection (RAID2004).In: Lecture notes in computer Science.vol.3324. Sophia Antipolis, France: Springer-Verlag; 2004 pg.102-124.

[27] Julisch Klaus, Dacier Marc. Mining intrusion detection alarms for actionable knowledge. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and Data mining. Alberta, Canada: Edmonton; 2002.pg.366-375.