Intrusion Detection System with Correlation Engine and Vulnerability Assessment

D.W.Y.O.Waidyarathna[1], W.V.A.C.Nayantha[2], W.M.T.C.Wijesinghe[3], Kavinga Yapa Abeywardena[4]
Department of Information System Engineering
Sri Lanka Institute of Information Technology
Kandy Road, Malabe, Sri Lanka

***Abstract*—The proposed Intrusion Detection System (IDS) which is implemented with modern technologies to address certain prevailing problems in existing intrusion detection systems' is capable of giving an advanced output to the security analyst. Even though the network of an organization has been secured internally as well as externally the intruders find ways to penetrate the network. With the system that is proposed activities of those intruders can be identified with a higher probability even if managed to bypass security controls of the network. The goal of this project is to give a reliable output to the system users where all the alerts are more accurate and correlated using HIDS alerts and NIDS alerts which is similar to the modern SIEM concept. The system will perform as a centralized IDS by getting inputs from both HIDS and NIDS which gives data regarding the activities of hosts and network traffic. With those implementations, the system is capable of monitoring host activities, monitoring network traffic with existing tools and give a correlated output which is more accurate, advanced and reliable prioritizing the possible attacks by using machine learning techniques and rule-based correlation techniques. With all these capabilities final product is a fully automated Intrusion Detection System which gives correlated alerts as outputs with a less rate of false positives compared to the existing systems.**

Keywords*—Intrusion detection system (IDS); intrusion detection message exchange format (IDMEF); network intrusion detection system (NIDS); host intrusion detection system (HIDS); security information and event management (siem); correlation; machine learning

I. Introduction

With ever growing technological solutions, computer systems and computer networks play a major role in the world. Today's enterprises, businesses, and organizations have mainly automated their old manual systems with the computerized solutions. As a result, while doing the processing, a lot of data is generated and consequently stored within these systems. Almost all devices of an organization are network aware and multiple devices are connected to the outside world via technologies such as virtual private networks and internet allowing outsiders to connect into the internal network of the organization. All these external users and processes, connected devices, networks and systems, security has become a major requirement in the field of information technology.

With these interconnected systems, there is a persistent risk for the organization as the probability of an attacker penetrating the network or sending a malicious payload to a system is comparatively high. Physical security is the first step of securing a network, however for a skilled intruder finding a small weak point to enter the internal network is not a challenge. Due to this concern physical security is merely not enough, hence internal security mechanisms are constantly applied and reviewed for network and system protection. Access Control Systems, Security Incident & Event Management Systems, Intrusion Prevention Systems, Intrusion Detection System and many more concepts are out there which are deployed against the intrusions done by various types of intruders. In this research, authors have focused on addressing prevailing drawbacks that exist in current Intrusion Detection Systems.

An intrusion detection system (IDS) is a device or software application that monitors a network or systems for malicious activity or policy violations. Any malicious activity or violation of a policy is typically reported either to an administrator or collected centrally using a Security Information and Event Management System (SIEM) [1]. Intrusion Detection Systems are mainly can be categorized as Host based Intrusion Detection Systems and Network Intrusion Detection Systems. A host-based intrusion detection system (HIDS) is an intrusion detection system that is capable of monitoring and analyzing the internals of a computing system such as user, operating system and application activities [2]. A network intrusion detection system (NIDS) monitors traffic on a network looking for suspicious activity, which could be an attack or unauthorized activity [3].

Another way of categorizing is according to the detection method, one of the most common way is signature-based detection which looks for known patterns of malicious network packets and malicious activities, referred to as signatures [4]. The other option available can be described as the anomaly-based detection which detects any abnormal behavior varies from the normal/legitimate traffic and operations [5]. Considering the above-mentioned detection methods, there are commonly studied problems especially with the signature-based models. For an example, most of the times systems tend to give false negatives as there is no way of identifying newly constructed attacks due to limited detecting capability with the known type of attacks/patterns. These newly constructed attacks are commonly referred to as zero day exploits. There is no way of detecting new attacks with signature based models [4]. The behavioral based models always try to generate alerts by identifying malicious traffic and operations even though it might belong to legitimate

events and network traffic. As a result, it creates a higher false positive rate [5]. Therefore, presently there is no perfect solution in the information security industry with respect to intrusion detection.

Even though security engineers cannot fully rely on the existing intrusion detection systems, the number of new types of attacks and the amount of attacks developed in the world hasn't been decreasing. Instead, zero day attack rate has increased considerably during the past few years. In order to address this problem, the security domain requires an advanced intrusion detection system that all types of organizations can afford without spending over the top. This solution should be capable of detecting intrusions with a higher rate of accuracy while maintaining a lower rate of false positives. Modern developments in the field of computing should be utilized to achieve this task.

II. Research Gap and Research Problem

Traditional Intrusion Detection Systems are mostly built in a way that they identify attacks by using 'signatures' and 'anomalies'. Even though some researchers have published papers and ideas regarding applying machine learning techniques for existing Intrusion Detection Systems, any of those systems are not performing well in a way that it can stand against to the current cyberwar [6]. When we analyzed the existing systems we could discover the main problems that prevailing the existing IDSs. Those limitations can be listed as,

- A high rate of false positives and false negatives
- Separate NIDS and Separate HIDS
- No correlation of NIDS and HIDS alerts, using both rule-based and machine learning techniques
- Lack of integrated vulnerability management
- Small-scale businesses can't afford existing IDS systems due to the high cost of implementation

With these limitations, it has been difficult to achieve the core security components through the Intrusion Detection Systems. When these systems give a higher rate of false positives, security analysts cannot depend on those results. They have to manually do the process that was done by the IDSs by diving deeper into the system and analyzing raw logs [8]. It consumes a lot of time as well as it makes the intrusion detections system useless when the task is redone manually by a person.

The Intrusion Detection Systems always act separately as Host Intrusion Detection Systems or Network Intrusion Detection Systems [6]. However, with modern threats and their undiscoverable quality, if we can merge these two systems into one then the output shall be more effective. This can be defined as the main objective of this proposed solution. The proposed system attempts to correlate the alerts received from both HIDS and NIDS which are received in a common format, processed in an optimized environment and presented in a meaningful manner.

As far as the IDS functionality is concerned it is important to understand why and how these attacks are managing to bypass the existing solutions. Usually for each attack there exist major known vulnerability of a system i.e. the real cause of an attack is some critical vulnerability of the systems. However, most of the current Intrusion Detection Systems does not address the vulnerability management area [7]. Hence those IDSs are not equipped with any integrated vulnerability management/assessment tool. This also should be identified as a major concern with existing solutions.

III. Methodology

The goal of this research was to develop a system which is mainly capable of detecting intruder activities with a higher rate of accurate alerts by minimizing the number of false positives using both rule-based and machine learning techniques for detecting the intrusion activities.

A. Detailed Flow

Brief workflow of the proposed system has been depicted below in Fig. 1, and it emphasizes the overall architecture and placement of the solution in an existing network.

B. Standardization of Alerts

1) IDMEF: The Intrusion Detection Massage Exchange Format known as IDMEF is focusing on defining data formats and exchange procedures for information sharing, which is a crucial factor for Security Incident and Event Management Systems. To define a standard representation of alerts, IDMEF is using XML based data models and this identical representation enables interoperability among different devices or systems [9]. The data from different devices are also allowed to be stored accurately by a standardized log format. The main purpose of the research is also focused on improving security by combining Snort NIDS System with OSSEC HIDS System. Many of past researches have used the IDMEF, as a protocol for exchanging intrusion detection massages which are being standardized by the IETF [13]. Intrusion Detection Message Exchange Format is basically an object-oriented depiction of generated alert data by Intrusion Detection Systems. Two types of implementation for IDMEF was proposed by Intrusion Detection Working Group (IDWG) [14]. One method is implemented using the Structure of Management Information (SMI) and the other is using XML.

C. Optimization of NIDS

1) Snort as NIDS: Snort is the de facto standard for network-based intrusion detection. This network intrusion detection system is an open source and rule-driven language, which fuses the benefits of protocol, signature, and anomaly-based inspection methods. As this method is rule-based, it is generally a misuse detection system but apart from that, it has some anomaly recognition capabilities. It supports logging events to either log files or a database.

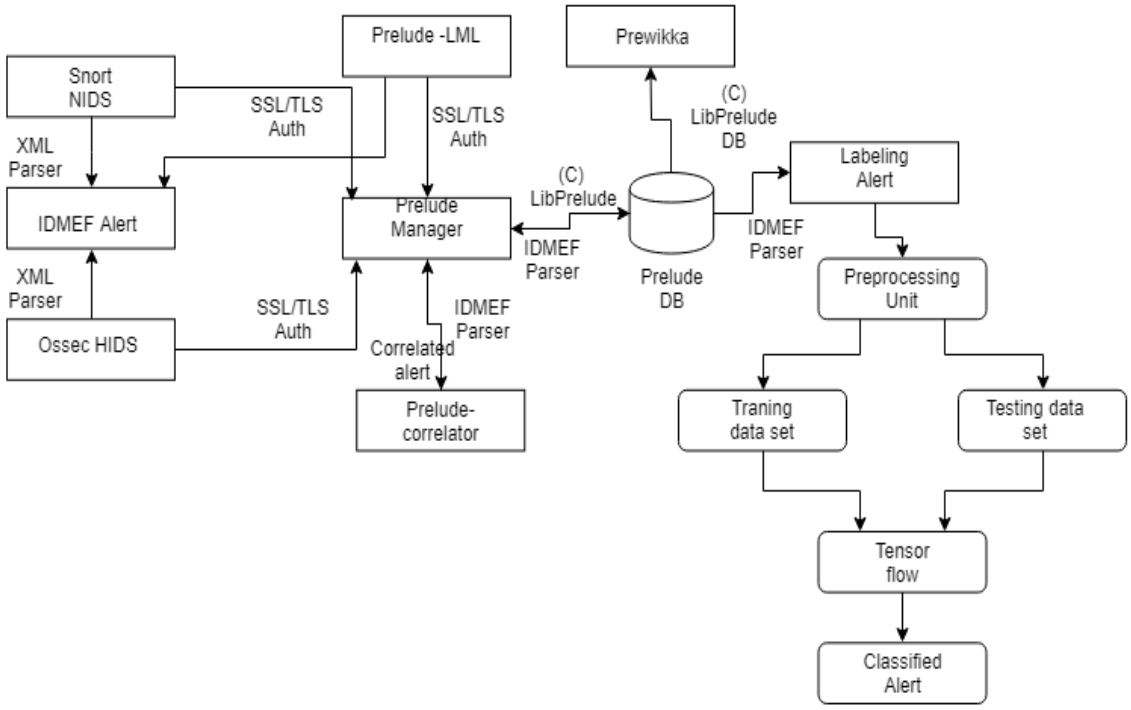

Fig. 1. Overall Diagram of the IDS.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<idmef:IDMEF_Message xmlns:idmef="http://iana.org/idmef version =1.0">
 <idmef:Alert messageid="1">
  <idmef:Analyzer analyzerid="527">
   <idmef:Node category="8" />
  </idmef:Analyzer>
 </idmef:Alert>
 <idmef:CreateTime ntpstamp="06/16-11:22:28.515219" />
 <idmef:Source>
  <idmef:Node>
   <idmef:Address category="ipv4-addr">
    <idmef:address>0.0.0.0</idmef:address>
   </idmef:Address>
  </idmef:Node>
  <idmef:Service>
   <idmef:priority> 2</idmef:priority>
   <idmef:protocol>IGMP</idmef:protocol>
  </idmef:Service>
 </idmef:Source>
 <idmef:Target>
  <idmef:Node>
   <idmef:Address category="ipv4-addr">
    <idmef:address>224.0.0.22</idmef:address>
   </idmef:Address>
  </idmef:Node>
  <idmef:Service>
   <idmef:protocol>IGMP</idmef:protocol>
  </idmef:Service>
 </idmef:Target>
 <idmef:Classification text=" Potentially Bad Traffic" />
</idmef:IDMEF_Message>
```

Fig. 2. Snort Rules.

NIDS is frequently placed between the edge firewall and a back-esnd firewall that protects the internal network from the publicly accessible network in between, called the DMZ or perimeter network or screened subnet [2].

Snort is configured to operate on NIDS mode. Whereas in Network IDS mode, Snort executes actual analysis to determine malicious traffic, based on that alerts are generated. To conduct testing DARPA 1998 datasets were downloaded from MIT Lincoln Labs website. Furthermore, the dataset comprises of replicated network traffic embedded with marked attacks. Snort was configured in the Network Intrusion Detection System to use this dataset. Example IDMEF messages obtained from a Snort alert file is shown below in Fig. 2.

D. Optimization of HIDS

1) OSSec as HIDS: OSSEC is a host-based intrusion detection system. It has a powerful log analysis, registry monitoring, and integrity checking and rootkit detection engine. It is a signature-based IDS, which detects intrusions based on rules [12]. The following Fig. 3 depicts the OSSEC architecture.

The system is using OSSEC agents to collect logs in all the hosts and pass into the OSSEC server for the analysis process. In the server side, received logs are decoded in two phases which are pre-decoding and decoding as in the Fig. 3 depicted above [15]. Pre-decoding phase is used to extract static information like time, hostname, log message and date from received events. Non-static information like event ID, source

and destination IP addresses are extracted in the decoding phase [11]. After that decoded data are passed to analyzing phase. Analyzing is performed through rule matching, based on predefined signatures this extracted information are matched. If malicious patterns are detected, it will be stored these data in the database in IDMEF format for correlation and alerting purposes.

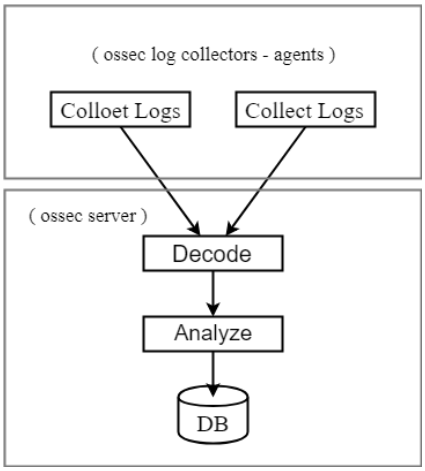

Fig. 3. OSSec Architecture.

E. Alert Correlation of HIDS and NIDS

Deployment of one sensor and then consequent alert generation to the network administrator is the most basic setup of the NIDS. The bigger network setups frequently need to use more than one sensor to cover the entire network. Presuming that, all of these sensors used in the above mentioned basic setup, this will result in a large number of alerts being directed to the administrator. The most common solution for this problem is to deploy a database in a central server to store all the generated messages sent by sensors. Then the central server can send status reports to the administrator. This is an exhaustive process which causes excessive information generation. This is where the concept of alert correlation fits in. By correlating the alerts coming from different sensors, information can be merged together to reduce the volume of information. It can also help in detecting attacks which are going to be missed otherwise. When distributed attacks are needed to identify several nodes of different subnets, sensors distributed across the network can be used.

Prelude is a Security Incident and Event Management (SIEM) system which allows us to achieve correlation capabilities easily. It has a rule-based correlation engine known as Prelude Correlator. Prelude also has a relatively user-friendly interface called Prewikka to present the analyzed data. Hence Prelude is quite useful to group events, identify unique alerts and to identify which of the flagged events have been caught by connected sensors or Prelude itself. The Correlator of Prelude has limited correlation abilities since it has only a limited number of rulesets. The researchers intended to detect various types of advanced attacks using this approach. Authors intended to extend the correlation functionalities of Prelude Correlator by writing a set of custom rules. The correlation process of Prelude SIEM is displayed below in Fig. 4.

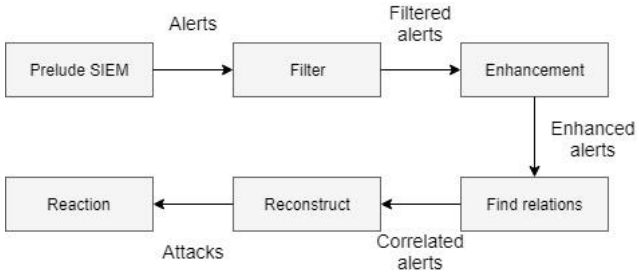

Fig. 4. Correlation Path.

In the proposed system vulnerability assessment tool will be integrated into the main system. This will be implemented via an API. Vulnerability Scanner will be called as a service when the system admin wants to run a vulnerability assessment. The vulnerabilities will be listed down according to the Common Vulnerability Scoring System (CVSS) [7]. Then the user will be notified which vulnerabilities should be addressed first and patches should be applied.

IV. Implementation

The important contributions of this work were to the preliminary execution of mandatory decoders, the integration of Prelude components, detection modules of Snort and OSSEC sensors, Prelude configurations for Snort and OSSEC analyzers, building classification algorithm, and the monitoring events within the overall IDS framework.

When malicious packets reach the perimeter, packets are captured and immediately sent to the Snort IDS preprocessor for inspection. By detecting the attack, the alerts produce from Snort IDS reach Prelude instantly, provided the particular sensor is already TLS authenticated.

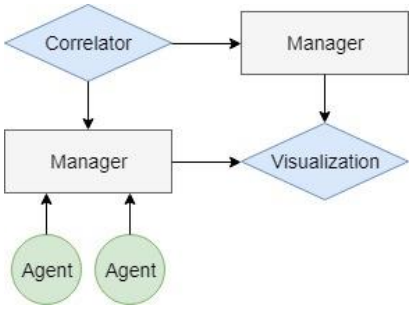

Fig. 5. Prelude Architecture.

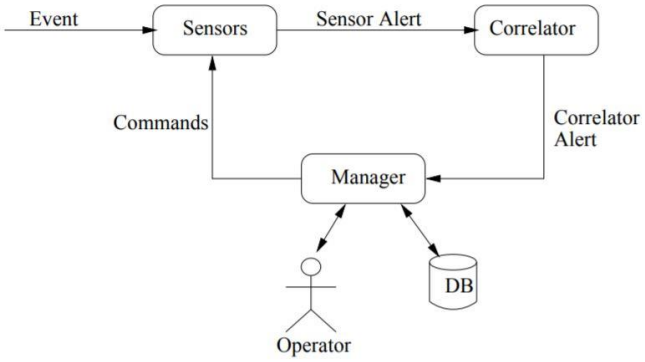

Fig. 6. Prelude Detailed View.

Prelude offers a great flexibility by combining a broad range of security tools under one powerful monitoring system. By correlating received alerts from other monitoring equipment such as Snort, OSSEC etc., it is possible to reduce the false positive alerts generated. Below figures, Fig. 5 & Fig. 6, illustrate the overall architecture of the Prelude SIEM.

All events have been normalized into the Intrusion Detection Message Exchange Format (IDMEF) by Prelude. Events from different devices are allowed to be stored in a structured format by normalization and more importantly, it allows all events collected, to be stored in the same database in the same format. It also makes the stored events well organized in order to maintain all the processed data. It is important to note that it does not need multiple storage devices to achieve this [3].

By reducing overheads and increasing efficiency Prelude helps to reduce security cost as well. Prelude SIEM records the events and filters them to eliminate non-threatening alerts, as well as to see whether if threats are connected via correlation.

Through Prelude web interface Prewikka, real-time event monitoring can be obtained. Manual reviewing of logs can easily result in a missed attack in the past. The real-time event analyzing allows, decrease in response time to the possible incidents. In an organization, it also allows seeing up to date activities easily on their entire network. Most recent events are showing automatically when the web page is set to refresh.

Libprelude gives an API (Application Programming Interface) which allows communicating with the Prelude subsystems for third-party software [10]. If any disturbance takes place between any of the components in the system, libprelude also makes sure that re-transmission of data is performed. Any device acting as a manager or a sensor, the libprelude package is required. As well, converting the logs into Preludes binary IDMEF format also entail. It ensures management servers and sensors use secure transmissions such as Secure Socket Layer/Transport Layer Security (SSL/TLS) to communicate.

For storing IDMEF alerts in the database, libpreludedb is the library that supplies an abstraction layer, and this library makes easier management of the database [10]. It allows the user to access the database without depending on the log format by hiding the inner workings. The hosting machine should require installing libpreludedb, in order to use the Prewikka web interface.

To analyze various different types of logs, Prelude requires Prelude LML (Log Management Library) component. The Prelude LML log analyzer determines whether activity within the logs is malicious by using a set of rules, and it is comparable to the way Snort uses the rules file to analyze packets. The rules files of Prelude LML attempt to match data within the log files instead of network packets.

The component which allows the correlation of events between various Prelude Management Servers is Prelude Correlator. Prelude Correlator is a python rule-based correlation engine and has the ability to connect and fetch alerts from a remote Prelude Manager Server. The users are allowed writing correlation rules using the Python programming language by Prelude Correlator. A correlation message is generated, once the streams of events match a correlation rule.

The web-based GUI (Graphical User Interface) for Prelude is Prewikka interface [10]. When a user logged into Prewikka, the Alerts tab act as the default page. Under the classification column, event summary is listed, by clicking it, a user can view more information about a particular event. This link will display all of the events that match a particular event description, source, target, and sensor. A user can view the actual event detail by clicking on the Threats tab. The Agents tab gives a detailed analysis of the agents which has reported to the Prelude Manager. Then the sensor is currently offline or online will be displayed on the system. The heartbeat analysis and heartbeat listing will appear when clicking on the Heartbeats tab and also it shows a list of recent heartbeats received by Prelude Manager Server. Heartbeat can be defined as a simple message which indicates that the agent is reporting, running and sending messages, and also it ensures whether the device is properly working, although it is not generating alerts.

A. Alert Classification using Machine Learning

The main purpose of the Intrusion Detection System is to differentiate between normal events and attacks as discussed earlier. The common situation about the generation of high false alarms is caused by the most of Intrusion Detection Systems. The research proved that the IDS System is more efficient when it holds a fewer number of false negatives and false positives. The use of machine learning techniques is a one way to deal with this problem, and the machine learning can be used to differentiate between attacks and false alarms [1].

In the proposed system, the format dissimilarity of alerts come from different sensors overcome by using the IDMEF (Intrusion Detection Message Exchange Format) format. Analyzed IDMEF alerts obtained from prelude API, classified into false alarms and true attacks using machine learning techniques [9]. Thereafter alerts from various Intrusion Detection Systems is gathered and the process as follows,

- Obtain collected alerts from common IDMEF format
- Labeling of alerts
- Constructing the dataset
- Modeling of the supervised machine learning algorithm classifier
- Classification of alerts into false alarm or true attacks using the below mentioned machine learning technique

Algorithm for labeling alerts was executed in python, and the labeled alert file is used for categorization, which was attempted using a supervised machine learning algorithm K-Nearest Neighbors (K-NN) classifier. The proposed solution was implemented using the Tensorflow framework to build the algorithm classifier. According to the Fig. 7 below, the KNN classifier has the best accuracy rate [16]. Hence the classification process has adopted the KNN classifier.

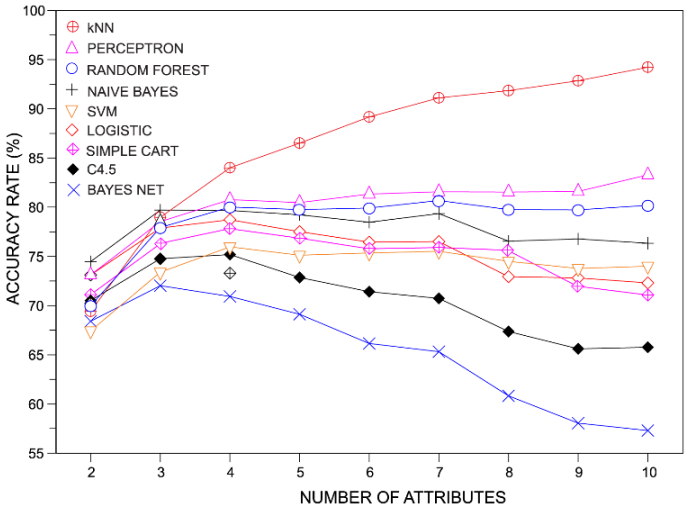

Fig. 7. Different Classifiers Accuracy Rate Comparison Graph.

V. Results & Discussion

Traditional IDSs available today has its own relative weaknesses and strengths. While one solution may be strong at host-based intrusion detection, the other solution may be strong at network-based intrusion detection. The organizations are highly concerned about their network and system performance; hence they use multiple IDSs from various vendors as they do not wish to take a chance with security. Different IDSs generate alert events in different formats, as well as use different protocols. If the outputs alerts are not integrated properly, false positive rates may increase hence interrupting the legitimate performance of a system or a network. False alarms caused by the large volume of IDSs is intolerable to the administrators as it delays the smooth functioning of an organization. It is necessary to decrease the excessive of false alarms to reduce the operational cost and excel in the reliability of a security system. Hence, this research was conducted intending to advance a procedure to obtain alerts from different sensors and standardizes them into IDMEF.

Rule-based architecture and machine learning techniques were used to compare security events. These methods analyze alerts generated from various sensors, which are normalized and combined into meta-alerts, then it used to classify true alerts or false alarms.

VI. Future Works & Conclusion

The research intended to introduce an advanced machine learning and rule-based, HIDS and NIDS correlated intrusion detection system. The system gives an optimized and reliable output which creates a fewer false positive rate compared to the past researches and existing IDS solutions. Further research can be conducted in developing an advanced intrusion detection system using the proposed approach. There are various open source IDS tools which can further be integrated with the proposed architecture to compare findings to fin the best possible combination. The overall objective is to achieve a more successful result in order to persevere against the modern types of attacks, which cannot be discovered by the traditional standalone Intrusion Detection Systems.

Acknowledgment

We would like to acknowledge the Sri Lanka Institute of Information Technology for providing us the necessary lab facilities and the test beds to conduct our research activities. We would also like to acknowledge everyone who continuously supported during the period of our research.

References

[1] A B. Athira, V. Pathari, “Standardisation and Classification of Alerts Generated by Intrusion Detection Systems”, IJCI, International Journal on Cybernetics & Informatics, Vol 5 Issue 2, 2016.

[2] Johansson Daniel, Andersson Par, “Intrusion Detection Systems with Correlation Capabilities”

[3] Yasm Curt, “Prelude as a Hybrid IDS Framework”, March, 2009

[4] Kumar Vinod, Sangwan Prakash Om, “Signature Based Intrusion Detection System Using SNORT”, IJCAIT, International Journal of Computer Applications & Information Technology, Vol. I, Issue III, November 2012.

[5] Singh Deepak Kumar, Gupta Jitendra Kumar, “An approach for Anomaly based Intrusion detection System using SNORT“, IJSER, International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September 2013.

[6] S, Vijayarani, and Maria Sylviaa S. “Intrusion Detection System – A Study”, IJSPTM, International Journal of Security, Privacy and Trust Management ,Vol 4, Issue 1, pp. 31–44, February 2015.

[7] Yang Guangming, Chen Dongming, Xu Jian, Zhu Zhiliang, “Research of Intrusion Detection System Based on Vulnerability Scanner”, ICACC, Advanced Computer Control, March 2010.

[8] Chakraborty Nilotpal, “Intrusion Detection System and Intrusion Prevention System: A Comparative Study”, IJCBR, International Journal of Computing and Business Research , Volume 4 Issue 2, May 2013.

[9] Kothari Pravin, “Intrusion Detection Interoperability and Standardization”, February, 2002.

[10] TIMOFTE Jack, “Intrusion Detection using Open Source Tools”, Revista Informatica Economică nr.2(46), pp. 75-79, 2008.

[11] A T. Oğuz, R. Maraş, “Host-Based Intrusion Detection Systems Ossec Open Source HIDS”

[12] Hwang Oun Seong, “APT Detection with Host-Based Intrusion Detection System and Intelligent Systems”, FCTA, Future Computational Technologies and Applications, 2017.

[13] Argyroudis, Patroklos, Paraskakis Iraklis, “Distributed Intrusion Detection using Mobile Agents”, 2002.

[14] Carey Nathan, Clark Andrew, Mohay George, “IDS Interoperability and Correlation Using IDMEF and Commodity Systems”, pp. 6-22, 2002.

[15] Hay Andrew, Bray Rory, Cid Daniel, Northcutt Stephen, “OSSEC Host-Based Intrusion Detection Guide”, Syngress, 2008.

[16] Diego Raphael, Cesar Henrique, Francisco Aparecido, "A Systematic Comparison of Supervised Classifiers", PLOS ONE, Public Library of Science, Vol 9, Issue 4, pp. 1-14, April 2014.