Cooperative Architecture for Signature Based Alarm System

Dr.V.Dhanakoti and D.Meenaakshi
Department of Computer Science and Engineering
SRM Valliammai Engineering College
Chennai
INDIA

Abstract: - Existing system security and architecture is vulnerable to withhold the offensiveness of different system risks. An advanced security framework in needed, to obtain better security in various layers of the network. This article represents a collective design for several Intrusion Recognition Systems (IRS) identifying intrusions during authentication. The recognition is efficient and effective, prepared by collective smart agents, appropriate knowledge base and blending several recognition sensors. The design has three divisions namely: Collective Alarm Indexing, Signature Based Alarm Estimation and Alarm Association. This design sinks the alarm by correlating outcomes of many sensors to produce condensed views. This decreases false positives as host system, system information from the estimation technique and combine measures based on reasonable relations, together produces Universal alarm intelligence Database. This design is above the intrusion recognition layer for post recognition alarm scrutiny and precaution actions. This paper discloses the design executed to obtain the results.

Key-Words:- System security; Intrusion recognition; Alarm; Smart agents; Collective work.

1. Introduction

System performance has become a critical factor in the digital world. With the new inventions in Collective work practices, system based association within persons and clusters spectacularly increase its efficiency. However, the Internet Era, the widespread dependence on system and Collective technology makes PC attacks a lot more devastating than ever before.

The distributed nature of Collective work application can be more vulnerable than traditional individual applications. For instance, intruders disguise as remote trusted entities and send an executable file to the victim. The victim executes it as a shared application, thereby exploiting Collective work application. Security systems inevitably become an essential part of Collective work applications as the latter's success is determined by the former. However, the daunting risk of system assaults specifically the increasing amount of distributed, synchronized malicious assaults (such as DDoS assaults), the collaboration among dissimilar security systems, security information and the collaboration among skilled becomes crucial to win this war against spiteful system hackers. Now-a-days Collective work techniques play a vital role in designing, developing and deploying security systems, gadgets and policies.

In response to confronts from malicious system assaults, a promising access (such as Computer and system forensics) to deter intruders is to identify, record and analyze assaults. This will help to prosecute spiteful assaulters using the evidence of the above process. This newly emerging discipline is becoming more significant as the society has recognized the facts of system assaults. It involves capturing, recording and analyzing system incidents in order to find out the stock of security assaults or additional setback events. It prevents hackers from assaulting a system as the chance of their discovery. The above evidence obtained will be useful in prosecuting the hackers or in planning the counter measures.

Intrusion Recognition System (IRS) is an essential part of computer and system examining. It is a security gadget or a system deployed to observe system and host performance including data flows, information accesses etc.

Furthermore it discovers suspicious performance and capture relevant evidence for future use. Its main purpose is to identify authentic time, ongoing intrusions and warn system administrators through an alarm so that necessary actions could be taken to stop the intrusions or to identify the damage, if the assault has already done.

At present there are two basic accesses to recognition of an intrusion [1]. The first access, called the anomaly recognition (also known as performance recognition). Anomaly recognition is to define and distinguish the correct stagnant form and the adequate vibrant activities of the system. It also identifies illegal changes or unfair performance. The second access, called the misuse recognition (also known as signature recognition) involves distinguishing known ways to break into a system.

All known penetration technique is typically depicted as a prototype. The misuse recognition system looks for explicit prototypes. The prototype may be a fixed bit string such as a precise virus bit string inclusion. Instead, the prototype may describe a suspect set or sequence of actions.

An intrusion recognition system (IRS) has been built using both the accesses: the anomaly recognition and the misuse recognition. Some of the IRS systems including EMERALD [2] and DIRS [3] explore hybrid access. In recent years, intrusion recognition products are widely deployed to gain acceptance as a worthwhile investment. But neither of these two recognition systems is quite satisfactory. For example, the anomaly recognition systems often produce too many false positives due to deviation from the normal performance, which does not correspond to the occurrence of an assault. Besides, the critical limit is hard to define. The misuse recognition systems can only identify those intrusions depicted in their signature repository and fails to capture new or vaguely modified intrusions.

In addition to the above weaknesses, IRS products are subjected to other problems, such as alarm flooding, too many false positives and false pessimistic alarms, isolated alarms against a series of assaults, blindness to system and hosts which it is observing.

Many of the above weaknesses in traditional IRS products are due to the lack of various collaborations among others, it includes: (a) dissimilar recognition systems, (b) intrusion recognition and system management operations, (c) recognition with other security systems.

In order to overcome the deficiency, a design to enable collaboration among several intrusion recognition systems using distributed smart agents and Collective work techniques was proposed. The ultimate goal of the anticipated collective design is to make intrusion recognition more accurate, competent and easier to use by system administrators. Wherever suitable, some of these processes may even be automated. The solution to the understanding of this goal is the use of a collective design that uses distributed smart agents and relevant information knowledge bases. In the anticipated design, dissimilar IRS products, the protected hosts and system asset information are integrated as a holistic security system. This is done through the deployment of a distributed system of autonomous smart agents. They interact via information swap and make decisions in a cooperative and synchronized manner. The design is placed as a layer above intrusion recognition system for post recognition alarm scrutiny and security actions.

The rest of this paper is organized as follows: In Section 2, different types of Collaborative work is analyzed. Section 3 represents proposed architecture and design of Collective elements such as Collective alarm indexing; Signature based alarm estimation and alarm association. Execution and Results are presented in Section 4. Section 5 concludes the paper.

2. Related Work

Researchers explored the benefits of collaboration among dissimilar IRS products by understanding the restrictions of single recognition systems. The main objective of the IRS cooperation is to reduce the number of alarms produced by correlating dissimilar IRS outputs and discard false alarms. By threading several alarms produced by related assaults, cooperating IRS modules will provide a Universal view of intrusion performance.

The first IRS collaboration research was initiated in the IDES [4] project and then refined in the EMERALD [2] project. Currently there are number of ongoing projects in the area of alarm indexing and the false positive decline.

Honeywell is developing Argus, a qualitative Bayesian estimation technology to combine results from several intrusion recognition systems [5]. Cuppens [6,7] is developing an intrusion recognition indexing and association module (MIRADOR) using snort and e-trust. A skilled-system based access for similarity formulation is used in their work. SRI international [8] is using a probability-based access to characteristic similarity recognition. Another access is the Tivoli Enterprise suggested by Debar and Wespi [9]. Whose association technique uses a significant system to state what types of alarms may pursue a given alarm category. Collaborative trust aware intelligent intrusion detection in VANETs was developed by Kumara and Chilamkurtib [10].

A generic Collective IRS and scrutiny design for several IRS products with smart agents and Signature based alarm estimation was developed. The proposed access differs from the above mentioned in the following aspects. First, the associations of two dissimilar types were categorized: (a) information asset association and (b) the alarm association. Information asset association uses the system and host hardware and software information to measure the alarm priority and the likelihood of the success of the assault. The alarm association associates dissimilar alarms based on the logical relations among them to provide a Universal vision of the effects of intrusion. Secondly, a generic collective design for several diversified IRS and alarm estimation was designed. This design sits as a layer above IRS products. Third, based on the precise alarms, suitable security solutions collected from several vulnerability screening corporations and organizations will be provided with the alarm. This assists security administration in taking necessary actions. Although this design uses smart agents and relevant knowledge base to provide a layer above intrusion recognition, it is novel in using agents to identify malicious intrusions.

Distributed Intrusion Recognition System (DIRS) [3,11] uses agents to aggregate audit intelligences from hosts. The design has a host administrator, an observing process or collection of procedures. Autonomous Agents for Intrusion Recognition (AAFID) [12] is a distributed anomaly recognition system that employs autonomous agents at the lowest level for data collection and scrutiny. At the superior levels, other agent entities are used to obtain a Universal view of performance. Helmer et al. [13] use mobile trivial agents in intrusion recognition. These agents travel amid screened systems to obtain, classify and associate information to identify suspicious performance. All the above researches use agents to some extent to collect information for the purpose of recognition. Therefore, the information is often retrieved from audit archives.

Many agent based recognition accesses such as DIRS, are used minimally. The Proposed design is a layer above intrusion recognition. The information is mainly used for the alarm estimation and security decision making. Agent technologies such as agent swap information languages and services for the collaboration among distributed smart agents were used. This design was named as Cooperative Layer Design. The implication is that a third eye is the system developed to watch larger than a computer system.

3. The Cooperative Layer Design

As shown in Fig.1, the collective design has three main elements: (a) Collective alarm indexing, (b) Signature based alarm estimation and (c) Alarm association. Each of these three parts is depicted in the following sections.

3.1. Collective Alarm Indexing

Collective alarm indexing is mainly aimed at mitigating alarm flooding. Meanwhile, a loosely coupled collaboration among several IRS products is also achieved using smart agents and advanced methods. This element provides three functions: (i) alarm pre-developing, (ii) alarm grouping and (iii) collective alarm merging. These functions are provided by the following distributed agents.

3.1.1. IRMEF Agent

To aggregate alarms from several IRS products with dissimilar output formats, first the element needs to convert the diversified formats into a unified customary representation. The format that was chosen is the Intrusion Recognition Message Exchange Format (IRMEF) [14] is emerging as an industry customary.

The conversion of heterogeneous alarm output into the customary IRMEF format is performed by IRMEF agent. This agent independently runs the entity, which collects alarms emanated from an IRS product and converts them into the customary IRMEF format. In the proposed design, a single IRMEF agent is developed and deployed close to that of IRS product. The IRMEF agent is then dedicated to those IRS products and converts its output into IRMEF format. Therefore, an IRMEF agent can be executed in any language and deployed in any system atmosphere to get customary IRMEF format as output. Whenever a new IRS is integrated into the system, there is need to provide a connection to IRMEF agent to serve it. Therefore, the design is extremely scalable and can evolve with advances in IRS systems. IRMEF agents store all their outputs in the IRMEF database for later scrutiny.

3.1.2. Grouping Agent

As IRMEF agent store alarms in the IRMEF database, another agent called Grouping agent, combines the alarms into dissimilar groups on basis of target, time and classification. Each group is then used by a ‘representative alarm’ or Meta alarm to represent that group. Grouping eliminates duplicate alarms and combines them from the occurrence of the same assault. After grouping, the alarms can be significantly concentrated. Groups are then used in alarm merging process by a Merging Agent.

3.1.3. Merging Agent

Currently, the collaboration among several IRS products in Cooperative layer design may be distinguished as loose and transparent. The Cooperative layer design is designed as a layer above IRS products. In this design, IRS products are integrated as black boxes. The collaboration is then achieved by this upper layer. In practice, when IRS vendors develop their own IRS products, vendors seldom have the collaboration with others in mind.

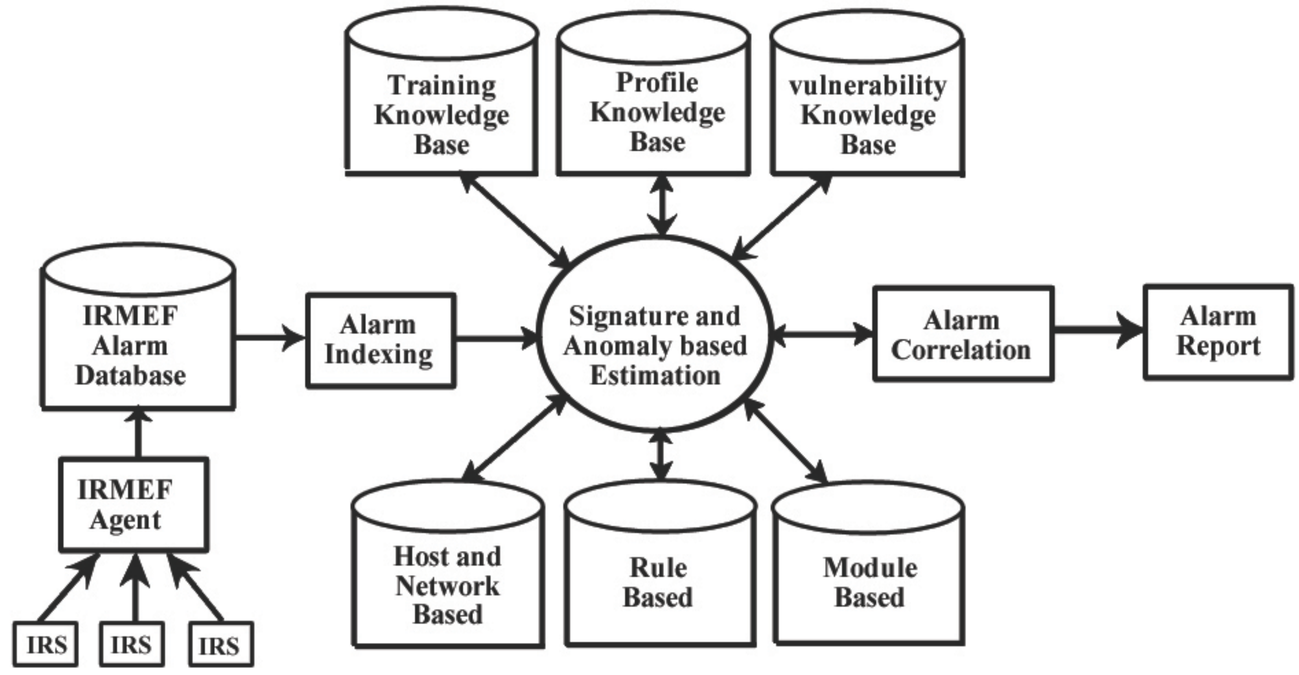

Fig.1. A Collective Design for Several Intrusion Recognition System

Consequently, the alarm formats are heterogeneous and recognition systems are proprietary and not released to public. Therefore, direct collaboration among several IRS products from dissimilar vendors is very difficult and may not be feasible at all. The collaboration among several IRS products is done indirectly by the alarm merging agent in the Cooperative layer design. The scrutiny of alarms from several IRS products results in useful information which could be gathered collectively and resolution of conflicts is carried out by this kind of alarm merging agent. This agent is also used to merge the alarms from dissimilar IRS products into blended alarms. Alarm merging agent is extremely smart because it uses a voting method to solve conflicts when IRS products have dissimilar ‘voices’. The method works as follows: if the number of IRS exposing alarms is greater than the number of non-dynamic IRS systems when an event occurs, the voting method will notify that an assault has occurred and produce a synthetic alarm. When two IRS deployed, a priority based voting method is chosen. When conflict arises, the method will produce a synthesized alarm. This happens only when the priority levels of the produced alarm pass the associated priority screening levels. The priority of the blended alarm will still be concentrated, one level lower evaluated to its unique one. After the alarm merging process, clean and synthesized alarms with detailed information from all of the dynamic IRS systems are sent to the Signature based estimation element for further scrutiny.

3.2. Alarm estimation

One of the main weaknesses of existing IRS products is that it often produces too many false positive alarms or system immune alarms. False positive alarms correspond to false alarms produced against normal or vaguely skewed performance. System immune alarms are authentic assaults but not harmful to the targets.

In practice, it should be treated with dissimilar priority evaluated to authentic harmful assaults. But existing IRS products often are unable to sort out these two kinds of alarms and even allocate critical priorities to these alarms. For illustration, in the laboratory, it was observed that an IRS system send out an alarm with top most severity in response to an exploiting and vulnerable assault. Whereas the assault was a well known Windows 2000 operating system even though the IRS was observing a Linux operating system.

System security operators will be distracted by these impossible-to-succeed assaults or low priority probes. Top volume of false positive alarms will make the system security operator miss the authentic spiteful assaults that are more likely to succeed. There are several reasons why existing IRS products are prone to produce excessive false positive alarms or system immune alarms. The poor quality of signature sets is one. Similarly the difficulty to define precisely the boundary amid abnormal and normal performance is another. But a significant reason is the fact that current IRS systems are often unaware of the system context it is protecting. The systems often are not fully integrated into the system atmosphere in which it is running on. The system only serves as an add-on security product rather being an integral part of the whole security system. Fundamentally, this situation is the result of a lack of collaboration among IRS products with other system or system management entities. That is to say, there is a lack of information and information sharing and exchanging amid them.

IRS products are isolated from the rest of the system. This isolation amid the IRS products and the system security atmosphere results in the top speed of false positives and system immune alarms. Common techniques suggested by IRS vendors and used in practice to solve the above problem are alarm sorting or to manually tune up the signature sets in misuse based recognition systems. This reconfigures the limits in abnormal recognition systems. But these techniques only push the burden on to system administrators' shoulders, but do not solve the problem. Furthermore, alarm sorting and signature tuning will require the system security operators to understand the IRS signatures thoroughly. Sorters are also difficult to maintain. For illustration, dissimilar sorters will be needed for dissimilar IRS products, since each IRS product has its unique signature sets or recognition methods. The anticipated solution to the above problem is the use of Signature based alarm estimation. Hence it was decided to use autonomous smart agents to fully integrate IRS systems into the system atmosphere.

The information exchanging and information sharing among these smart agents and extensive knowledge about existing vulnerabilities in protected system and prospective assault techniques enable us too competently and effectively appraise produced alarms. In practice, most of the existing IRS products especially commercial products employ signature based recognition accesses. This is because signature based access is much easier to execute and successful in practice.

Furthermore, many IRS products, such as Snort [15], refer their signatures to customary CVE [16], Bugtraq [17] or CERT [18] etc. exploits. Meanwhile, these security observing organizations such as CVE, Bugtraq and CERT broadly list all the vulnerable systems and application information infected by this assault. Consequently, this assault and the infected system information equivalent can be used to appraise the assault severity and its likelihood of success in Cloud Computing [19].

The function of this Signature based estimation is to eliminate these kinds of system immune or false positive alarms and at the same time enabling the system security operators to concentrate on authentic frightening alarms. In this context, the 'knowledge' implies the integration of the system topology configuration, the hardware gadgets, the operating systems and the application information (combined with well known vulnerability information) which runs into the risk management and scrutiny system. In addition, 'knowledge' refers to the knowledge about the protected system and ongoing intrusion techniques. An information asset containing all the above hosts and system information is used in the estimation process to sort out false positives and rank the severity of assaults.

The vulnerability knowledge base (Fig.1) stores all the known exploits and system vulnerability information together with the equivalent security solutions. The functions provided by this Signature based estimation are similar to alarm sorting and signature tuning. But it significantly relieves the heavy burden on system administrators and is more competent and easier to configure and maintain. Fig. 2 shows the design of the estimation element and illustrates the alarm estimation process in a typical scenario. In the estimation process, the following solution elements are involved:

3.2.1. Host Agents

A smart host agent is installed on each host in the observed system. It collects low level details of the host's system and configuration information. Host agents are autonomous, smart and aware of the system atmosphere in which it is running. Host agents do not communicate with each other directly. Instead, it communicates with a director agent.

3.2.2. Director Agent

Director agent is synchronized with host agents. The swap information and collaboration amid host agents and director agent is vital to the whole estimation element. Host information collected from host agents are first sent to the director agent. The director agent then stores the information in suitable places in the system and host asset knowledge base. Meanwhile, if any information needed by an estimation process found missing in the knowledge base, director agent will request the equivalent host agent to collect that information and sends it back.

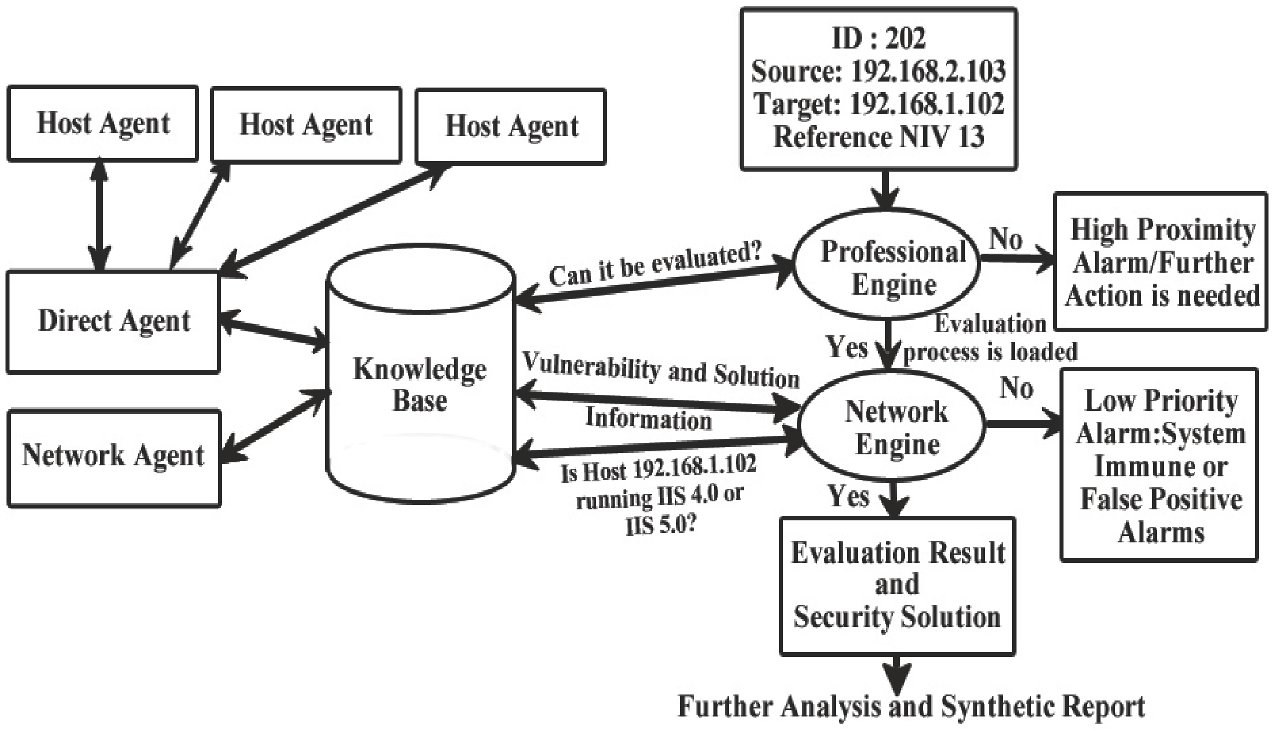
ID : 202
Source: 192.168.2.103
Target: 192.168.1.102
Reference NIV 13
Host Agent
Host Agent
Host Agent
Direct Agent
Network Agent
Knowledge Base
Can it be evaluated?
Professional Engine
No
High Proximity Alarm/Further Action is needed
Yes
Evaluation process is loaded
Vulnerability and Solution Information
Network Engine
No
Low Priority Alarm:System Immune or False Positive Alarms
Is Host 192.168.1.102 running IIS 4.0 or IIS 5.0?
Yes
Evaluation Result and Security Solution
Further Analysis and Synthetic Report

Fig.2. Knowledge Based Alarm Estimation using a Typical Circumstance

3.2.3. The Coordination System

To enable coordination among distributed agents, Collective work and new cluster techniques was executed. For illustration, a common swap information language facilitates information swap and information sharing. Agent Swap information Language (ACL) provides agents with a means to swap information and knowledge. In the design of swap information amid director agent and host agents, the system does not use the customary ACL, such as KQML [20] or FIPA ACL [21]. Instead, a design was developed to execute a simple ACL language or procedure to be used in Cooperative Layer Design. A fully specified ACL such as KQML or FIPA ACL was not chosen because the swap information amid director agent and host agents is simple, mainly of the query answer type. The domain ontology is the same, but the syntax of the information in the procedure is adapted from KQML. For illustration, simple information sent from a host agent to the director agent can be encoded as:

(Inform Director
:sender hostagent A
:content (Linux OS 3)
:content (information)
:receiver director)

In this simple information, the host agent A is telling the director that the host is observing and running a Linux operating system with version 3. A query from the director agent to the host agent can be encoded as:

(Enquiry Agent
:sent Director
:content (PATCH LEVEL)
:receiver hostagent A)

To facilitate swap information amid director agent and host agents, an infrastructure of services was needed for naming, registration and other services such as encryption. Naming services are used to provide corresponding amid host agent name and IP address. Registration service is used by host agent to specify and collect the system information. This information is then used by director agent to determine which information to query to which agent. Encryption service is very significant in the swap information among agents because the information being swapped contains sensitive information, which should be prevented from unauthorized access. In addition, the information should also be signed by using the host agent's secret solution to verify its integrity.

The swap information or coordination model amid the director agent and the host agent is a 'push–pull' model. The host agent pushes system atmosphere information to the director under three circumstances: (1) after each reboot of the host, host agents automatically start to collect system information and send it to the director agent. (2) After a user defined renew time period, the host agents look for renewed system information and send it to the director agent. (3) Upon arrival of the director agent's request, the requested information is sent. The director pulls information from host agents under two situations: (1) Upon the arrival of a request from an estimation process. (2) After a certain user defined updating time period. This model corresponds to the dominant 'request reply' swap information paradigm larger than the System. Besides host agents and director agents, there is a special agent, which is a system agent deployed for each system segment.

3.2.4. Network Agent

Network agent manages an internal topology map of the protected system. Network agent identifies the available assets on the system, their existing state (up or down), IP address to hostname equivalent, open ports and the application behind those ports, dynamic TCP and UDP system services and hardware information etc. Network agent runs at periods to maintain a renewed topology map of the protected system. The Network agent does not communicate with director agent directly. The host information and system information collected by host agents and Network agents are all stored in a system and hosts asset knowledge base. This knowledge refers to the knowledge about the protected system.

3.2.5. Knowledge Bases

This knowledge base contains vibrant system and host asset information. The information is used to compare with vulnerability need information to measure alarms and provide suitable security solutions for authentic harmful assaults. In addition to knowledge about assaults there is a need to know about Vulnerability. The knowledge base contains known vulnerabilities and the known penetration accesses exploiting these vulnerabilities. These are identified by the security observing organizations and their equivalent security solutions. Vulnerabilities are categorized and sorted according to CVE, Bugtraq and CERT vulnerability references. For each vulnerability reference, there is an equivalent security solution to facilitate security decision making.

3.2.6. Skilled System Engine

In the alarm estimation framework, the estimation processes are executed by a skilled system engine. After indexing process, IRS alarms are asserted as 'facts' into the knowledge base. Suitable estimation process is then activated by the assertion of new alarms. A skilled system is build into the framework due to the following reasons: (1) IRS products should never be used individually. It should be used closely coupled with security policy, contingency plan and other defense systems. In realism, security policy and contingency plan are often executed with predefined security rules. Therefore, after the estimation process, certain defense actions activated by these rules can be executed by the skilled system to deter intrusions. (2) By representing alarms as facts in knowledge base, the system can query the knowledge base to find associations amid facts (alarms). This is the primary purpose of the third element Alarm Association in the Cooperative Layer Design. In addition, predefined defense rules can take actions based on the contents of one or more facts (alarms) leading to automation of some defense processes.

3.2.7. Alarm Estimation

Fig.2 depicts the alarm estimation process in a typical scenario. After the collective alarm indexing process, IRS alarms are asserted as 'facts' into the knowledge base. After that, an estimation process is freighted and executed. The estimation process first queries the vulnerability knowledge base to see whether the NIV-13 vulnerability information is available or not. If not, the alarm is marked as a topper priority alarm. Otherwise, the vulnerability information including security solution suggestions concerning NIV-13 is retrieved from the knowledge base.

The estimation process then identifies the target and uses system topology information to roughly measure the alarm. If the target's Operating System is in the vulnerability list, the estimation process then goes to the next step. Otherwise, the process marks the alarm as a lower priority alarm. In this example, the estimation process goes to next step to retrieve target's host information. The estimation process then compares the target information against the vulnerability. A relevance score is calculated based on the counterparts of the information. The relevance score ranges from 0 to 1 with 1 meaning a perfect equivalent and 0 meaning no equivalent at all. Finally the estimation process presents the measured result as well as suggested security solution. This is equivalent to the assault of system administrator to facilitate decision making.

3.3. Alarm Correlation

The main purpose of this element is to find out the logical correlation among the alarms. Assaulters are likely to launch a series of assaults against their targets. Existing IRS systems can only produce isolated alarms based on each step it identifies. Smart hackers are more likely to disguise their authentic purpose by launching many other minor assaults.

Alarm correlation element is used to associate alarms based on logical associations among the alarms. This function will provide the system security operator with a great insight into where the initial assaults came from and where it actually ends up. This function can also be used to find prototypes among series of assaults. After the alarm correlation, a top-level intelligence providing an overall view of the assaults will be presented to the system security operators.

The Alarm correlation element has other functionalities. By suitable correlation of the alarms from dissimilar IRS systems, the occurrence of a certain assault was verifiable. For illustration, a system based IRS identifies a suspicious remote buffer overflow assault. This gets over shell access to a server machine. But due to its restriction, it does not know what is really going on inside that host after that. Meanwhile, a host-based IRS system deployed inside the same server identifies a suspicious shell process producing an alarm. Therefore by correlating these alarms from the two dissimilar IRS products, the system security operator can further confirm that some remote shell access assault is in progress. Furthermore, since each IRS product has its own blind spots, an association can help to remove some of the false pessimistic.

There are many factors to consider when evaluating IDSs such as speed, cost, effectiveness, ease-of-use, CPU and memory usage, and scalability. The ease-of-use includes user interface, interoperability with other products, reporting capabilities, and investigation capabilities.

Table 1 New Confusion Matrix

Class	Predicted Negative Class (Normal)	Predicted Positive Class (Assault)
Actual Negative Class (Normal)	True Negative-TN	False Positive-FP
Actual Positive Class (Assault)	False Negative-FN	True Positive-TP

The effectiveness of an ID is evaluated by its ability to make correct predictions. According to the real nature of a given event compared to the prediction from the IDS, Four possible outcomes are shown in Table 1, known as the confusion matrix. True negatives (TN) as well as true positives (TP) correspond to a correct operation of the IDS; that is, events are successfully labeled as normal and assaults, respectively. False positives (FP) refer to normal events being predicted as assaults; false negatives (FN) are assault events incorrectly predicted as normal events by Wu et al [22].

A high FP rate will seriously affect the performance of the system being detected. A high FN rate will leave the system vulnerable to intrusions. So, both FP and FN rates should be minimized, together with maximizing TP and TN rates by Mansour et al [23].

Equations (1) - (6), based on the confusion matrix, Table 1, show a numerical evaluation that applies the following measures to quantify the performance of IDSs by Wu et al [22]:

$$\text{TrueNegativeRate(TNR)}=\text{TN}/(\text{TN}+\text{FP})= \text{Specificity}; \quad (1)$$

$$\text{TruePositiveRate(TPR)} = \text{TP}/ (\text{TP} + \text{FN}) = \text{DR or Sensitivity}; \quad (2)$$

$$\text{FalseAlarmRate(FAR)}=\text{FP}/(\text{TN}+\text{FP})= 1-\text{Specificity}; \quad (3)$$

$$\text{FalseNegativeRate(FNR)}=\text{FN}/(\text{TP}+\text{FN})= 1-\text{Sensitivity}; \quad (4)$$

$$\text{Accuracy} = (\text{TN} + \text{TP})/ (\text{TN} + \text{TP} + \text{FN} + \text{FP}) \quad (5)$$

$$\text{Precision} = \text{TP}/ (\text{TP} + \text{FP}) \quad (6)$$

Thus, two metrics are to be used to evaluate the proposed CIDS performance, namely, the intrusion DR and FAR.

4. Execution and Observations

As of the writing of this paper, the collective alarm indexing element and the Signature based estimation element of the design was executed in the proposed system. Two dissimilar systems were used based in IRS products: Snort [15] and Prelude [24]. These two IRSs were installed in a Redhat Linux system to observe a subnet with heterogeneous operating systems and dissimilar configurations.

The alarm pre-developing, alarm grouping and alarm merging were all executed in Perl. For each IRS, a Perl script was printed. This Perl script served as an IRMEF agent to that IRS sensor, converting its alarms into the customary IRMEF format and storing alarms into a relational IRMEF database. Another Perl script was printed as a grouping agent to combine the alarms into dissimilar collections based on the blending of the basis, target, time and the classification information.

The merging agent was also printed using Perl script, which includes the voting method. Host, director and network agents were also executed in Perl. To enable swap information among agents, processes were needed to be provided to parse incoming information's, compose information's for transport and channel them through the system using some lower level system procedures.

In the system, execution was carried out by as shared Perl libraries. A Jess Skilled Engine [25] was deployed for the estimation process. Two dissimilar estimation processes was designed and executed, single for UNIX (Linux) and the other for Windows. All estimation processes were executed as Java classes and freighted on demand to measure the equivalent alarms. Finally, a PHP front end was provided as a web portal to view the alarms, system and system information. The validation was done using an assault database for one month.

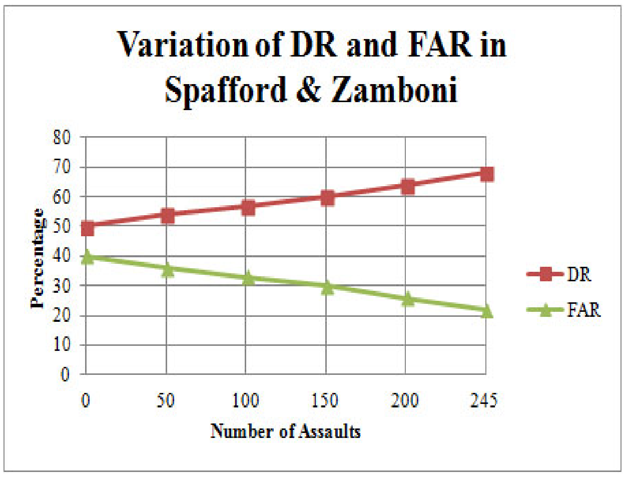

Fig.3. Variations of DR and FAR using Spafford and Zamboni method

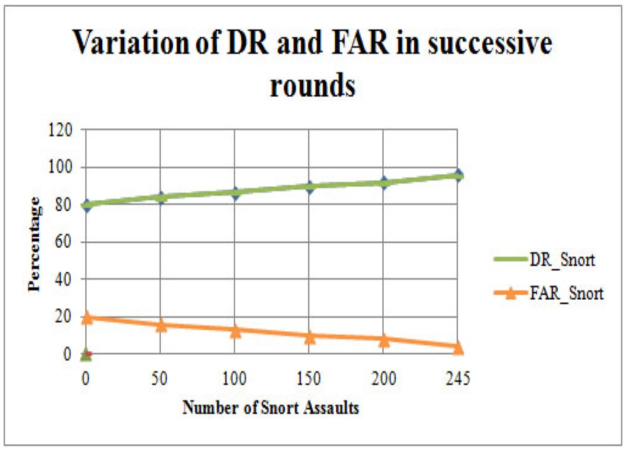

Fig.4. Variations of DR and FAR in Snort

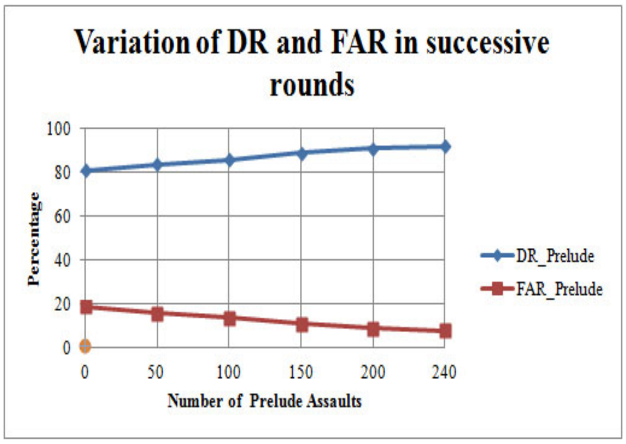

Fig.5. Variations of DR and FAR in Prelude

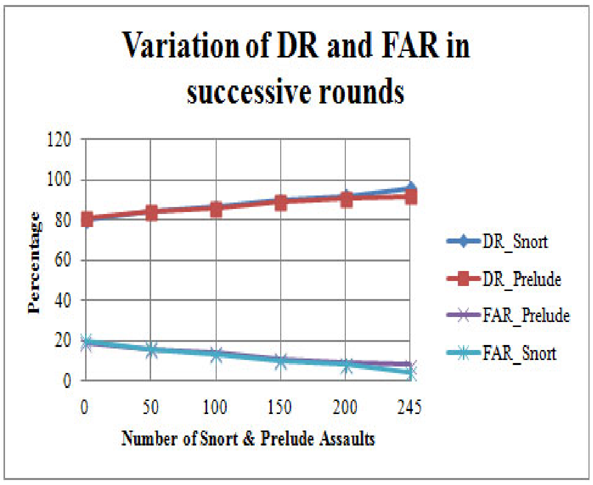

Fig.6. Variations of DR and FAR in Snort and Prelude

During that Observation, Snort produced 1553 alarms and Prelude produced 1405 alarms. Of these 485 assaults, Snort identified 245 assaults and Prelude identified 240 assaults. 15 assaults were missed by both of them. Thus, two metrics are to be used to evaluate the anticipated collective design for several intrusion recognition systems performance, namely, the intrusion DR (Detection Rate) and FAR (False Alarm Rate) [26, 27 and 28].

The indexing element produced 485 indexing. Thereby the system was able to detect 95% of the assaults. The voting method in the merging process successfully removed 45 false positives of Snort and 36 false positives of Prelude. Deliberately assaults were designed in order to produce system immune alarms and false positives. Fig.3 presents the DR and FAR of assaults using Spafford and Zamboni method. The system was only able to capture 70% of the assaults and the FAR was 30% in Spafford and Zamboni method without the use of the Collective IRS. Fig.4 presents DR and FAR of assaults in Snort tool. Fig.5 presents DR and FAR of assaults in Prelude tool. Fig.6 presents DR and FAR of assaults in Snort and Prelude tools. Hence the proposed Collective IRS performance was able to obtain above 90% of DR and which is better than Spafford and Zamboni agent based method.

For illustration, one of the Linux hosts with several IIS buffer overflow was assaulted and for a sample, one of the Window 2008 machines installed containing the necessary hot fix with an assault exploiting certain vulnerability in Service Pack 2. But during the Observation, all of the assaults were identified by Snort and Prelude and some of the produced alarms were marked with the top most severity. The Signature based estimation process successfully measured them as system immune alarms.

Furthermore, some exploits were commenced, which will certainly make the targets vulnerable. This time the alarm estimation process not only marked them as top priority alarms, but also suggested us the equivalent suitable security solutions. So far, the test was conducted in authentic atmospheres. Presentation of the alarm indexing element and the alarm estimation element was measured in an open and authentic atmosphere.

5. Conclusion and Future Work

In this paper, the need for collective intrusion recognition, collaboration among intrusion recognition and other system management systems were emphasized. A collective design was created and named as Collective Design to synchronize several intrusion recognition systems using various distributed smart agents. Relevant information from knowledge base and Collective work techniques were used to group and merge alarms from several IRS products to achieve an indirect collaboration among them. By combining the strengths of the heterogeneous IRS products, alarm association can also be used to verify the occurrence of certain assaults and eliminate some false pessimistic.

The collective intrusion recognition and knowledge based alarm estimation design presented in Cooperative Layer Design can be widely applied to the design of integrated intrusion recognition products and related system security engineering applications. This will be more efficient and effective in terms of identifying authentic spiteful assaults. As of this writing, the system was executed with a set of assaults and the results thus obtained were quite satisfactory.

References:

[1] A.K. Jones and R.S. Sielken, Computer system intrusion detection: a survey, *Technical report, Computer Science Department, University of Virginia;* 2000.

[2] P.A. Porras, P.G. Neumann, EMERALD: event monitoring enabling responses to anomalous live disturbances, *In: Proceedings of 19th national information system security conference (NISSC). Advanced Engineering Informatics,* Vol.19, 2005, pp. 93–101.

[3] S.R. Snapp, J. Brentano, G.V. Dias, T.L. Goan, L.T. Heberlein, C. Ho et al, DIDS (distributed intrusion detection system)—motivation, architecture, and an early prototype, *In: Proceedings of the 14th national computer security conference, 1991.*

[4] T. Lunt, IDES: An intelligent system for detecting intruders, *Computer security, threats and countermeasures,* 1990.

[5] C.W. Geib, Goldman RP, Probabilistic plan recognition for hostile agents, *In: Proceedings of the FLAIRS, 2001.*

[6] F. Cuppens, Managing alerts in a multi-intrusion detection environment, *In: Proceedings of 17th annual computer security applications conference (ACSAC).* New-Orleans, 2001.

[7] F. Cuppens and A. Miege, Alert correlation in a cooperative intrusion detection framework, *In: Proceedings of the IEEE symposium on security and privacy,* 2002.

[8] A. Valdes and K. Skinner, Probabilistic alert correlation, *Fourth international workshop on the recent advances in intrusion detection,* 2001.

[9] H. Debar and A. Wespi, The intrusion detection console correlation mechanism, *Fourth international workshop on the recent advances in intrusion detection,* 2001.

[10] N. Kumara and N. Chilamkurtib, Collaborative trust aware intelligent intrusion detection in VANETs, *Computers & Electrical Engineering,* Vol.40, No.6, 2014, pp.1981-1996.

[11] S. Salah, G.M. Fernandez and J.E. Diaz Verdejo, A model based survey of alert correlation techniques, *Computer Networks,* Vol.57, No.5, 2013, pp.1289-1317.

[12] EH. Spafford and D. Zamboni, Intrusion detection using autonomous agents, *Comput Networks,* Vol.34, No.4, 2010, pp.547-570.

[13] G. Helmer, JSK. Wong, V. Honava, L. Miller and Y. Wang, Lightweight agents for intrusion detection, *J Syst Software,* Vol.67, 2003, pp.109–22.

[14] Intrusion detection message exchange message format (IDMEF). Last Accessed On: 18 Dec 2014.Available:http://www.ietf.org/rfc/rfc4765.txt.

[15] SNORT. Last Accessed On: 18 Dec 2014. Available: http://www.snort.org.

[16] Common vulnerabilities and exposures (CVE), The MITRE Corporation. Last Accessed On: 18 Dec 2014. Available: http://www.cve.mitre.org.

[17] BUGTRAQ, Security focus online. Last Accessed On: 18 Dec 2014. Available: http://www.securityfocus.com/archive/1.

[18] CERT. Last Accessed On: 18 Dec 2014. Available: http://www.cert.org.

[19] G. Andreadisa, G. Fourtounisb and K.D. Bouzakisa, Collaborative design in the era of cloud computing, *Advances in Engineering Software,* Vol.81, 2015, pp.66-72.

[20] ARPA Knowledge Sharing Initiative, Specification of the KQML agent communication language. ARPA Knowledge Sharing Initiative, External Interfaces Working

Group, *CIKM '94 Proceedings of the third international conference on Information and knowledge management*,1994, pp. 456-463.

[21] FIPA, Foundation for Intelligent Physical Agents. Last Accessed On: 18 Dec 2014. Available: http://www.fipa.org.

[22] S.X. Wu and W. Banzhaf, The Use of Computational Intelligence in Intrusion Detection Systems:A Review. *Applied Soft Computing*, vol.10, No.1, 2010, pp.1-35.

[23] N. Mansour, I.C. Maya and A. Faour, Filtering intrusion detection alarms, *Cluster Computing*, vol.13, No.1, 2010, pp.19-29.

[24] PRELUDE. Last Accessed On: 18 Dec 2014. Available: http://www.prelude-ids.org.

[25] JESS. Last Accessed On: 18 Dec 2014. Available: http://herzberg.ca.sandia.gov/jess.

[26] K. Hwang, M. Cai, Y. Chen and M. Qin, Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes, *IEEE Transactions on Dependable and Secure Computing*, Vol.4, No.1, 2007, pp.41-55.

[27] P. Mohana Priya, V. Akilandeswari and S. Mercy Shalinie, Detecting DRDoS attack by Log File based IP pairing mechanism, *WSEAS TRANSACTIONS on COMPUTERS*, Vol.13, 2014, pp.538-548.

[28] C. Fung, J. Zhang, I. Aib and R. Boutaba, Trust Management and Admission Control for Host Based Collaborative Intrusion Detection, *Journal of Network and Systems Management*, Vol.19, No.2, 2011, pp.257-277.