Collection Mechanism and Reduction of IDS Alert

Karim Hashim Al-Saedi
National Advanced IPv6 Centre
Universiti Sains Malaysia
11800 USM, Penang, Malaysia
The University of
Mustansiriyah, Iraq

Sureswaran Ramadass
National Advanced IPv6 Centre
Universiti Sains Malaysia
11800 USM, Penang, Malaysia

Ammar ALmomani
National Advanced IPv6 Centre
Universiti Sains Malaysia
11800 USM, Penang, Malaysia

Selvakumar Manickam
National Advanced IPv6 Centre
Universiti Sains Malaysia
11800 USM, Penang, Malaysia

Wafaa A.H. Ali Alsalihy
CEMIS, Department of
Information System
Nizwa University, Oman

ABSTRACT
Numerous techniques and approaches are used to address the threats that are faced by computer networks today's. Some of these reactive approaches involve Intrusion Detection System (IDS), malware data mining and network monitoring. Numerous false positive alerts are generated by the IDS, contributing negatively to system complexity and performance. In this paper, we present a new framework called collection mechanism and reduction of IDS alert framework (CMRAF) to remove duplicate IDS alerts and reduce the amount of false alerts. CMRAF is based on two models. The first model develops a mechanism to save IDS alerts, extract the standard features as intrusion detection message exchange format, and save them in DB file (CSV-type). The second model consists of three phases. The first phase removes redundant alerts, the second phase reduces false alerts based on threshold time value, and the last phase reduces false alerts based on rules with threshold common vulnerabilities and exposure value. We applied CMRAF on two environments: the Darpa 1999 and the NAv6 network center data sets. The result obtained from the experiment on Darpa 1999 data set recorded an 92% alert reduction rate, whereas that on the NAv6 data set recorded an 84% alert reduction rate. From the results, CMRAF was able to scale back a massive quantity of redundant alerts and effectively reduces false alerts.

General Terms
Network Security

Keywords
False positive, Reduction alert, Network security, IDS, Aggregation alert.

1. INTRODUCTION
Recently, networks have been considerably used in different fields of interest. Despite the facilities they offered, they are not free of danger. For instance, they contain diffused and myriad threats and different types of malicious programs. Such programs have an effect on the efficiency of networks when transmitting data, urging researchers to improve and develop techniques that can block such threats. A definite example is the development of the intrusion detection system (IDS). This system has been designed to provide computer systems with extra protection from thousands of alerts that the system receives per day by providing a security analyst to verify each alert that depends on an aggregation criterion. Several systems [1-3], techniques, and methods have been designed in this field to reduce the number of threats by aggregating or correlating them to understand the working mechanisms of such threats. Internet and other network usages have become essential and extensive. Correspondingly, threats and intrusion activities have become wider and smarter.

IDS trigger a large volume of alerts by detecting these intrusions. Analysts spend much effort to analyze these alerts to determine the cause, relationship among alerts, and other features of the intrusions. The large number of alerts triggered by IDS causes problems during the analysis process. Several studies have been conducted to help analysts study these alert databases [3].The present paper proposed a new framework to address the problem of false positives in IDS based on two models. The first model develops a mechanism to save IDS alerts, extract the standard features as intrusion detection message exchange format (IDMEF), and save them in a DB file (CSV-type). The second model removes the duplicated IDS alerts and reduces the amount of false alerts.

2. RELATED WORK
Several researchers e.g.,[4-8] have studied and analyzed these alerts and their respective properties in order to reduce the proportion of false alerts as well as the discrimination between the real and the false alerts. In particular, an alert consists of several features, and these must be taken into account and closely analyzed so that they can explore efficient methods and techniques to reduce the amount of false alerts. Alharby and Imai [5] have proposed a similar concept, wherein frequent behaviors are observed in additional time. They found that accurate formation can represent a normal alert pattern; thus, an unexpected burst of the sequence alerts could be flagged when it is impossible to see this sequence appear as a suspicious activity. Few limitations have been mitigated by constructing a systematic model. This facilitates the use of historical alerts pattern by utilizing sequential pattern extraction; in turn, this helps the system understand future alerts. The newly extracted sequence pattern is similar with the sequential pattern extracted using the proposed system. In particular, normal behaviors are represented by this extracted pattern. When the process contains several similarities, there is a high possibility of having normal behavior. A new technique based on data mining has been

proposed by Al-Mamory and Zhang, [9] with the aim of reducing false positive alerts. The main idea of this technique is that alerts are gathered into a group of clusters where a generalized alert is created from each cluster by this technique. The root causes are then converted to a filter so that the future alert could be reduced. In that proposed technique, generalization and the concept of the nearest neighboring are taken into account.

Based on the observed network's background knowledge, a new measurement is used in order to calculate the distances among alert feature values, resulting in an 82% reduction of total alerts. Julisch [10] has proposed an approach based on alert clustering. The main aim of this approach is to perform an analysis for the root cause. Julisch assumed that triggering alerts are caused by the root cause. For instance, the HTTP server that has a broken TCP/IP stack might fragment IP alters. The IDSs operates on the basic assumption that alerts are triggered daily, and that 90% of these alerts are triggered by some root causes. According to Julisch, clusters of alerts are identified and similar alerts are included in every cluster. The same root cause corresponds to these alerts. These features are generated into a generalization hierarchy with various levels of alert features. This ensures the significance of the dissimilarity measurements of the clustering analysis. The shortest path between two points is the structure concept of their method, which aims to perform calculations for the average dissimilarity between an alert cluster and a generalized alert.

In order to reduce IDS alerts, an alert cluster has been introduced by Njogu and Jiawei [11] based on the principle that a strong mechanism is required to reduce the false alerts. The similarities of verified alerts are computed by this mechanism based on the distance among the new alert features. Supporting evidence (Vulnerability data) and both clustering techniques are used by this approach so that a strong alert cluster can be effectively constructed. The objective of this mechanism is to ensure that the unnecessary alert load is reduced and the transmitted quality of alerts to analysts is enhanced. A statistical causality analysis correlation approach was proposed by [12]. This approach was based on statistical analysis and time series to develop attack scenarios. The authors proposed a clustering technique to aggregate the alerts to be represented as one hyper alert in each cluster based on time intervals. The objective of their approach was to reduce the amount of alerts and obtain alert prioritization to identify the important alerts. The drawback of this approach is also its incapacity to remove redundant alerts and its inflexibility to choose the alert features.

The present research proposed a new approach called collection mechanism and reduction of IDS alert framework (CMRAF), which depends on the IDS alert database obtained from the network by leveraging IDS Snort. This framework is based on two models. The first model operates mainly in two phases: the first phase collects alerts in the form of a text file and converts them to a CSV-type file, and the second phase extracts the features that will be used by the second model. The second model reduces the false alerts based on three main phases. The first phase reduces the false positive alerts by removing duplicate alerts. The second phase also reduces duplicate alerts based on threshold time value, and the final phase reduces the alerts depending on the rules and value of common vulnerabilities and exposure (CVE).

3. METHODOLOGY

Computer intrusions have become an increasingly serious problem in the past few years. IDS is an integral component of an in-depth architecture that provides a complete computer network security defense. It monitors packets for evidence of intrusive behaviors. An alarm is raised once an intrusive event is detected, providing the security analyst the opportunity to react promptly against such behavior. However, it provides an unmanageable number of alerts with 99% being false positives [3]. CMRAF is a significant framework that removes duplicate IDS alerts and reduces the amount of false alerts. This framework has two main components: Traffic Data Retrieval and Collection Mechanism model and Reduction IDS Alert Process model (RAPM). Fig. 1 shows the CMRAF architecture.

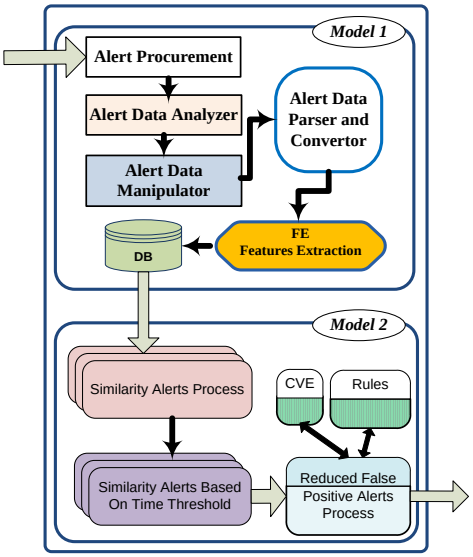

Fig. 1 shows the architecture of the RAPM.

3.1 Traffic Data Retrieval and Collection Mechanism model (Model 1)

An IDS generates alerts when suspicious traffic is detected based on redefining rules. Then, these alerts are stored in a file to be used by the system administrator. There are many types of IDS used, including ACARM, Bro IDS, Suricata, Prelude, and so on. One of the most common detectors is the IDS Snort, a packet sniffer that monitors network traffic in real time and audits each packet closely to detect a suspicious payload [13].

IDS Snort is an open-source IDS created by Roesch. It is a very flexible and feasible software system that can be used with different types of databases, such as MySql, Oracle, and so on. This software has an attack-detection engine and a port scanner. The latter helps warn or respond to any type of previously identified attacks. IDS Snort can provide two types of alerts: fast mode and full mode. The system presents option to the user to choose the required type of available alerts.The Traffic Data Retrieval and Collection Mechanism model has three main components: PRE-KNOWLEDGE, feature extraction of IDS alerts, and CVE, as shown in Fig. 1.

3.1.1- PRE-KNOWLEDGE

Pre-Knowledge determines the different data formats and helps exchange and share information of interest to both the intrusion detection and response systems, as well as to the

management systems that might be included. Pre-Knowledge has two main components: *Procurement of IDS Alerts* and *Field Reduction and Data Standardization*.

1- Procurement of Alerts

Procurement of IDS Alerts is the first component of Pre-Knowledge that receives IDS alerts from IDS and saves the said alerts into one text file. Figure 2 clearly shows part of the IDS Snort alert text file.

```
[**] [1:384:5] ICMP PING [**]
[Classification: Misc activity] [Priority: 3]
06/12-19:57:08.574994 192.168.1.5 -> 192.168.1.1
ICMP TTL:32 TOS:0x0 ID:1693 IpLen:20 DgmLen:50
Type:8  Code:0  ID:8   Seq:24841  ECHO
[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
06/12-19:57:08.588236 192.168.1.1 -> 192.168.1.5
ICMP TTL:255 TOS:0x0 ID:1693 IpLen:20 DgmLen:50
Type:0  Code:0  ID:8  Seq:24841  ECHO REPLY

[**] [1:384:5] ICMP PING [**]
[Classification: Misc activity] [Priority: 3]
```

Fig. 2 Section from text file IDS snort alert

2-Field Reduction of IDS Alerts and Data Standardization of IDS Alerts

Field Reduction and Data Standardization of IDS Alerts is the second component of Pre-Knowledge, which is responsible for extracting the standard features from the IDS alert file after the first component has performed its function. This consists of three subcomponents, namely, *Alert Data Analyzer, Alert Data Manipulator,* and *Alert Data Parser and Converter.*

A- ***Alert Data Analyzer***

Alert Data Analyzer checks whether or not the format of the IDS alert file conforms to the specification. If the alert file is an invalid format file, then the analyzer returns an error message and quits; otherwise, the file is to the next subcomponent called Alert Data Manipulator.

B- Alert Data Manipulator

The Alert Data Manipulator is responsible for checking the IDS alert features. This subcomponent fills any missing feature with a default value, although a missing feature does not usually occur. Nonetheless, when the Internet Control Message Protocol (ICMP) is used, the "port" features can be replaced by P-9 or any other possible offset values. The value is chosen because it does not have a port with the same substitute values. Figure 5 shows an example of alerts.

C- Alert Data Parser and Converter

The Alert Data Parser and Converter is considered a significant subcomponent of this model. It extracts features from the IDS alert file and saves the file in a DB file format (CSV-type) that the software will later use. Because it is highly flexible, the Alert Data Parser and Converter can load features of the processed alerts in a table that consists of a number of rows and columns. The table can be constructed based on user selection from a number of existing features. Figure 3 shows an example of alerts.

```
Alert1 Before:
  [**] [1:408:5] ICMP Echo Reply [**]
  [Classification:Misc activity] [Priority: 3]
  06/12-21:57:07.142376 192.168.1.1 -> 192.168.1.5
  ICMP TTL:255 TOS:0x0 ID:1705 IpLen:20 DgmLen:50
  Type:0  Code:0  ID:8  Seq:24841  ECHO REPLY
Alert1 After:
  1:408:5, ICMP Echo Reply,
  Classification: Misc activity, Priority: 3,
  06/12-21:57:07.142376, 192.168.1.1, P-9, 192.168.1.5, P-9,
  ICMP, 255, 0x0, 1705, 20, 50,  0, 0, 8, 24841
```

Fig. 3 Example for alerts

3.1.2-Feature Extraction of IDS Alerts

Feature Extraction of IDS Alerts is used to determine the most effective feature in the alert. The system uses information gain ratio algorithms, which function based on the extraction of similarities between sets of alerts. Afterwards, the algorithms provide the highest weight to the most effective features, based on the class of alerts belonging to the information gain ratio algorithm. This is expressed in the following equations:

$$GainR(X,C) = gain(X,C) / split\_info(C) \quad (1)$$

Where, GainR(X,C) represents the gain ratio of the feature x frequency in class C.

$$Split\_Info\ (C) = -\sum_i (|c_i|/c) \log |c_i|/c \quad (2)$$

Where $\mathbf{c_i}$ and $|\mathbf{c_i}|$ refer to the frequency of feature *X* in class C, *I* is the subclass of *C* and the number of features in $\mathbf{c_i}$ respectively. Table (1) shows the result of the information gain ratio of the IDS alert features.

Table (1) information gain ratio on IDS alerts

Ranked	position	feature name
2.0759	12	Portd
2.0759	9	Ips
1.9529	7	Time
1.9038	13	Protocol
1.4476	4	Classification
1.3613	11	Ipd
1.2943	10	Ports
1.0797	5	Priority
0.7858	2	G:S:RID
0.5197	15	TOS
0.3167	14	TTL
0.0441	6	Date
0	17	IpLen
0	1	Alert_ID
0	16	ID
0	18	DgmLen

The FE of the IDS alert features depends on the results applied on the information gain ratio algorithm, and are selected from feature gains with high weights.

3.1.3- CVE

CVE is the term used to refer to security threats and consists of two types, namely, vulnerabilities and exposures. Vulnerability refers to a computer, server, or network that is responsible for generating a definite and identifiable security risk in a particular context. Exposure refers to a security-related situation, event, or fact that might present vulnerability to a number of people. The MITRE Corporation developed the CVE to facilitate data-sharing process among diverse interests in security-related fields. CVE is a process of surfing for information using either security-related databases or the Internet. Such process is a collaboration of products from experts and representatives coming from different security-related organizations throughout the world. Figure 4 shows An example of CVE Information details for one type.

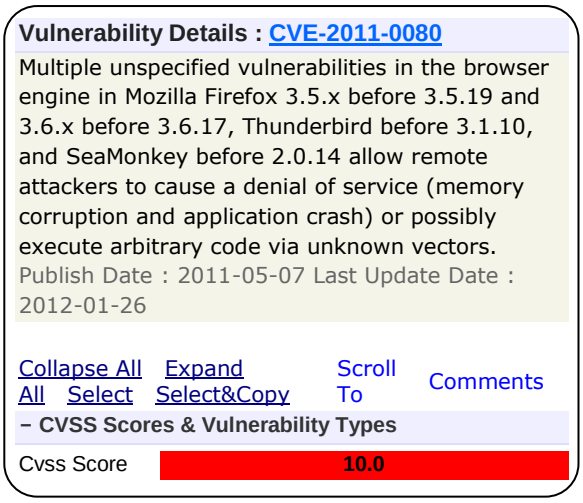

Fig. 4 An example of CVE Information details for one type

Once the items in the CVE are generated, the items are given names based on two criteria: (1) the year of the formal inclusion of each item; and (2) the order of each item in the list for the given year. To illustrate further, consider CVE-2008-0080, which indicates a specific buffer overflow in the WebDAV Mini-Redirector in Microsoft Windows XP SP2, Server 2003 SP1 and SP2, and Vista. This condition may allow remote attackers to execute arbitrary codes via a crafted WebDAV response. The given item is thus added in 2008 and then given the sequence number 80 for that year [14, 15]. In each IDS alert, features that refer to the CVE reference are present and are relied upon in issuing the IDS alert. When a return occurs to the CVE reference, the value of the score weakness or vulnerability is found. The values of the CVE contain a system that is constantly up-to-date and available for all users, making it trustworthy.

3.2 Reduction IDS Alert Processes Model (RAPM)

The objectives of developing the IDS are: (1) to monitor the activities of a given environment, and (2) to decide whether the activities are malicious or normal, depending on the integrity of the system and on the confidentiality and the information resources availability [3, 16]. The following issues should be considered when building the IDS: (1) data collection, (2) data preprocessing, (3) intrusion recognition, (4) reporting, and (5) the act of responding. Among the five issues, intrusion recognition is considered the most vital. The mechanism of Audit data compared with the detection models helps describe the patterns of the intrusive behavior. Intrusion efforts help in identifying both successful and unsuccessful intrusion attempts.

Constructing models automatically from the data is not an easy or trivial task, particularly when several Intrusion Detection (ID) problems are present. Intrusion attempts can be represented based on the following: (1) huge network traffic volumes, (2) highly imbalanced data distribution, (3) difficulty in realizing decision boundaries between normal and abnormal behaviors, and (4) the requirement of continuously adapting to a constantly changing environment.

The problems reveal that current IDS techniques are still unsatisfactory due to a number of the following limitations [3, 17, 18]:

- Detection of only specific types of attacks,
- Inspection of thousands of alerts per day, 99% of which are false positive alerts such that events are erroneously classified as attacks [3, 19], and
- Generation of an huge amounts of alerts

These limitations result in a more error-prone and time-consuming security implementation of IDS. Moreover, such limitations complicate the tasks of Internet security officers who must identify and learn about different Internet threats or attacks.

Input: ***DB_Alerts***
Output: ***Update DB-Alerts***
1) Set the Threshold_Time, Threshold_CVE
2) Read DB_Alerts as A [f_1, f_2…, f_n]
3) For I=1; I<=Alert_No;
4) { j=i+1
5) Do
6) If A_i [f_1, f_2…, f_n] = A_j [f_1, f_2…, f_n] Then
7) Remove A_j from DB_Alerts
8) j=j+1
9) Else
10) If A_i [f_{Time}<= Threshold_Time, f_2…, f_n] = A_j [f_{Time}<= Threshold_Time, f_2…, f_n] Then
12) Remove A_j from DB_Alerts
13) j=j+1
14) Else
15) j=j+1
16) Until j=Alert_No
17) Next I
18) }
19) For I=1; I<=Alert_No;
20) {
21) If A_i [(F_{RFC} =R_1 or R_2… or R_n) and (f_{CVE}<= Threshold_Time)] Then
22) Remove A_i from DB_Alerts
23) Next I
24) }

Fig. 5 New Reduction Alert algorithm (NRA)

The RAPM is significant in removing duplicated IDS alerts and in reducing the amount of false alerts. This model has three components, namely, Similarity Alerts Process, Similarity Alerts Based on Time Threshold, and Reduced False Positive Alerts Process. Fig. 1 shows the architecture of the RAPM. The PARM is built over a New reduction alert (NRA) algorithm is proposed in order to remove the redundant alerts, and to reduce the amount of false positive alerts. Figure 5 shows the proposed algorithm.

3.2.1- Similarity Alerts Process

The Similarity Alert Process is responsible for the removal of redundant alerts based on the similarity of alert features. This subcomponent reads the first alert together with the next alert and makes a comparison between the features of the two alerts. If similarity exists, the corresponding alert is deleted to retain only one alert; otherwise, the process proceeds to the next alert, and so on. Consequently, a database with no duplicate alerts is obtained.

3.2.2 - Similarity Alerts Based on Time Threshold

The Similarity Alert Based on Time Threshold reduces redundant alerts. In particular, this threshold is adopted to help the end user select a value. Here, a default value of 137 ms is provided, which represents the minimum time of the thread of the W32.BlueCode.Worm to obtain Web requests [20, 21. In this component, a comparison between alert features within the threshold value is made. If similarities exist, the corresponding alert is deleted to retain only one alert; otherwise, the process proceeds to the next alert, and so on. Finally, a database that has no duplicate alert is obtained.

3.2.2 - Reduced False Positive Alerts Process

The Reduced False Positive Process removes false positive alerts, and is based on two main principles: (1) the rules prepared for this purpose, and (2) the CVE value that represents the threshold value. This sub-component ensures reliability and accuracy, and minimizes the amount of alerts, because the process is based on the threshold value and rules. Table 2 presents these rules. Five types of alerts are considered as false positives (*INFO web bug 1x1 gif attempt, ICMP Destination Unreachable Port Unreachable, ICMP Echo Reply, ICMP PING,* and *CHAT IRC message*). IDS Alert Snort determines these alerts, which use the same protocol, namely, ICMP (Tjhai et al., 2010).

Table 2 The rules used in the NRA algorithm

Rule No.	Rules
Rule 1	If RFC=INFO web bug 1x1 gif attempt and CVE Score < = thr. then delete
Rule 2	If RFC=ICMP Destination Unreachable Port Unreachable and CVE Score < = thr. then delete
Rule 3	If RFC=ICMP Echo Reply and CVE Score < = thr. then delete
Rule 4	If RFC=ICMP PING and CVE Score < = thr. then delete
Rule 5	If RFC=CHAT IRC message and CVE Score < = thr. then delete

The alerts are checked according to the alert feature RFC with a CVE value through the application of the rules presented in Table 2. If a match is found, the alert is deemed as a false alert and the alert is excluded. An indication that no false alert exists means that the rule is not applicable; thus, the next alert is selected. The process is repeated until all alerts in the database are checked. An example is presented below.

1:408:5, ICMP Echo Reply, Classification: Misc activity, Priority: 3, 06/12-21:57:07.142376, 192.168.1.1, P-9, 192.168.1.5, P-9, ICMP, 255, 0x0, 1705, 20, 50, 0, 0, 8, 24841. CVE-2004-0790

Based on these alert features: FRC = ICMP Echo Reply. The classification value of CVE is five because the reference of this alert is CVE-2004-0790; thus, Rule 5 is applied on this alert. Depending on the similarity, this alert is excluded from the alert database. In this model, the threshold CVE value is entered into the analyzer system, although a default value of six is provided. This default threshold value is selected, because it is based on the average value of the CVE score provided in the Web site [23].

4. EXPERIMENTAL REDUCTION RESULTS AND ANALYSIS

The proposed approach was implemented on an Intel Core Duo E6750 at 2.66 GHz system with 3 GB memory (RAM), and a Windows 7 Professional operating system. We implemented the proposed framework by leveraging Matlab version R2010a. Six experiments were conducted to prove our objectives, explained in details in Section 4.2. The next section explains the data sets utilized in our system.

4.1 Datasets.

To check the validity of the proposed framework, our checks were evaluated based on two environments. The first one was the Darpa 1999 data set, a standard data set known by all authors in this field. We used the data set for this purpose and compared it with that of another author within the same area of the same data set. The second was the Nav6 data set; it is a collection of real data sets from the server in our Nav6 center within the period from June 6 to June 10, 2012. The number of experiments was three. The next sections discuss the experiments conducted.

4.2 Accomplishment experiments

The data sets in our check leverage IDS Snort 2.9 [20]. To generate an acceptable alert set, we used IDS Snort 2.9 in our experiments, which has the flexibility of providing alerts in a flat file [9]. Afterward, we conducted a preprocess to remove the symbols and convert the file to a CSV file on a table format to simplify its application in any system, including our proposed system.

Experiment 1: Three days from the DARPA 1999 datasets are used to compare the results of the Reduction of IDS Alert Process module with other approaches, such as those of [11, 19]. This experiment is tested on Thursday of the fourth week, Thursday of the fifth week, and Friday of the fifth week. This test dataset has been prepared for use by the Reduction of IDS Alert Process module. The sizes of these data are 287, 599 and 925 MB, respectively. Table 3 presents the parameters used in this experiment

Table 3: Parameters used in Experiment1

Parameters	Setting
Dataset	This experiment used a three days from DARPA 1999 dataset, which consist of 7,390 alerts.
Time Threshold	The time threshold used < 137ms
CVE Threshold	The value of CVE threshold used < 6

Table 4: The results of the 3days of the DARPA 1999 dataset

Days	Input Alerts	Output Alerts	Reduction Rate %
Thursday 4th week	**1728**	**313**	**81.89%**
Thursday 5th week	**4120**	**375**	**90.90%**
Friday 5th week	**1542**	**451**	**70.75%**

Table 4 presents the results of the alerts obtained before and after the implementation of the proposed module. This module reduced the amount of alerts with an average alert reduction rate of 84.58% [(81.89%+90.90%+70.75%)/3]. Figure 6 presents the results of Experiment 1 using three days of the DARPA 1999 dataset, and Figure 7 illustrates the reduction rate.

Table 5 Comparisons between the proposed approach and other approaches

Approaches	Thursday 4th week	Thursday 5th week	Friday 5th week	Total
Perdisci et al., 2006	80.30%	51.10%	62.40%	58.90%
Njou and Jiawei, 2010	79.90%	74.50%	79.80%	78.00%
Our Approach Proposed	81.89%	90.90%	70.75%	84.58%

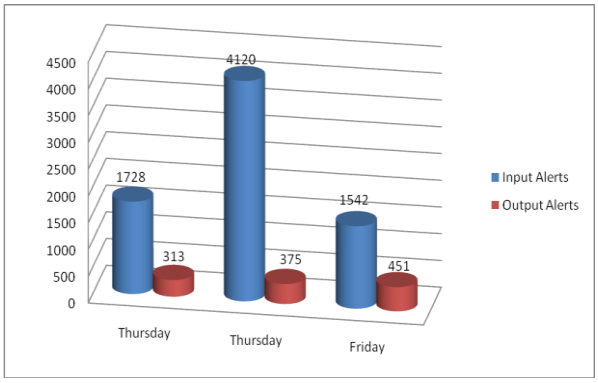

Figure 6: The results of Experiment 1 using three days of the DARPA 1999 dataset

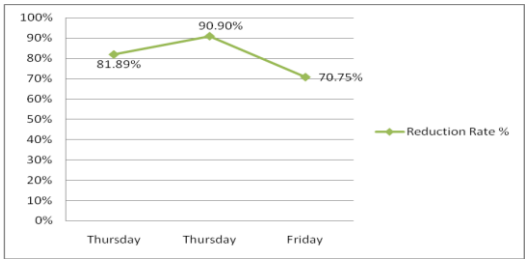

Figure 7 The reduction rate for three days of the DARPA 1999 dataset

On Thursday of the fourth week, a reduction rate ratio of 81.89% is obtained (Table 3). This ratio is obtained because of several alerts that contain high ratios of the RFC feature and are similar with the rules mentioned in Table 2. In addition, redundancy occurs in the alerts.

The results obtained on Thursday of the fifth week are higher than the results obtained on Thursday of the fourth week, because rate ratios are again higher in the former than in the latter. However, on Friday of the fifth week, the obtained results of the reduction rate ratio are 70.75%. This day contains a small proportion similar to the rules and few redundancies, which affect the rate reduction by reducing it to less than two days (Thursday of the fourth week and Thursday of the fifth week). Comparing these results with those obtained by previous researchers [11, 19], we note that the results of the proposed module are significantly better than previous results (Table 5).

Table 5 shows that the obtained results of this module are significantly better than those of other available modules [11]. However, for Friday of the fifth week, the rate ratio reached 79.80%; this ratio is higher than the proposed module that reached 70.75%, whereas the rate ratios of the three modules are low. The main reason is that on Friday of the fifth week, few redundant alerts occurred. Moreover, the false alarm rate on this day is low compared with the rest of the days. This module also depends on rules, including the rule that the CVE that has reached a value that is greater than its threshold. The researcher depends on three features, whereas this module uses more than these features to obtain highly accurate results.

Experiment 2: This experiment id conducted over a period of five weeks (the first, second, third, fourth, and fifth week) of the DARPA 1999 dataset, which contains the prepared traffic for the test set in the reduction of the IDS alert module. The sizes of these data are 1733, 1385, 1832, 1430 and 2582 MB, respectively. Table 6 presents the parameters used in this experiment. Table 7 illustrates the obtained results using the Reduction of IDS Alert Process module.

Table 6 Parameters used in Experiment 2

Parameters	Setting
Dataset	This experiment used a five weeks of DARPA 1999 dataset, which consist of 62785 alerts.
Time Threshold	The time threshold used < 137ms
CVE Threshold	The value of CVE threshold used < 6

Table 7 The results of the DARPA 1999 for five weeks using the proposed module

Week	Amount Alerts Before Reduction	Amount Alerts After Reduction	Reduction Rate
First week	6129	66	98.92%
Second week	24010	2016	91.60%
Third week	6304	64	98.98%
Fourth week	8048	810	89.94%
Fifth week	18294	1669	90.88%
Total	62785	4625	92.63%

In particular, two weeks obtain extremely high percentages in the reduction rate (Table 7). Those weeks (the first week and third week) and the fourth week obtain inputs not higher than those in other modules despite of the number of alerts in the proposed module. Based on the information provided by the DARPA Web site, the DARPA 1999 dataset is classified into

two groups based on the labels of attacks. The first group includes the first and third weeks, which are classified under the free attack label. The second group is classified as the attack label. Table 8 presents the reduced rate based on the attack label.

Table 8 The classification of the reduced rate based on the attack label

Week	Amount Alerts Before Reduction	Amount Alerts After Reduction	Reduced Rate
First & Third (Free attack label)	12433	130	98.95%
Second& Fourth& Fifth (Attack label)	50352	4726	90.61%
Total Weeks	62785	4856	92.27%

The first group (free attack label) means that no attack is contained in those datasets. They are pure datasets through which the rate ratio of the reduction is 98.90%. This ratio includes an average error rate not exceeding a ratio of 1.10%. This result provides a clear and robust indication of the proposed method and the credibility to reduce the false alerts.
The second group (attack label) means that attacks are contained in those datasets. The obtained results of the three weeks indicated a rate ratio reduction of 90.61%. This obtained ratio is good, as the overall rate of the false alerts is 99% [3]. Thus, the error rate of the proposed method does not exceed 6.73%. Figure 5.4 presents the results for the entire duration of the DARPA 1999 dataset usage. The total number of alerts before and after reduction amount to 62785 and 4999, respectively. The rate reduction of the alert is 92.03%. Figures 8 and 9 present more details for the alerts in the DARPA 1999 dataset before and after reduction, respectively

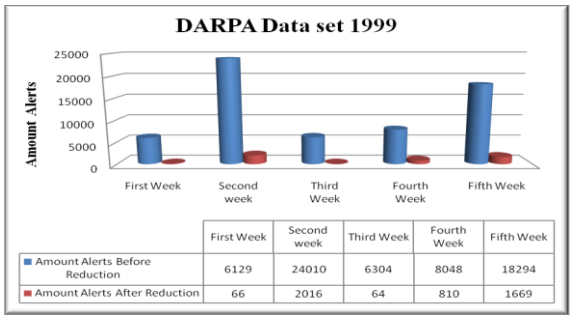

Figure 8 Alert reduction for five weeks of using the DARPA 1999 dataset

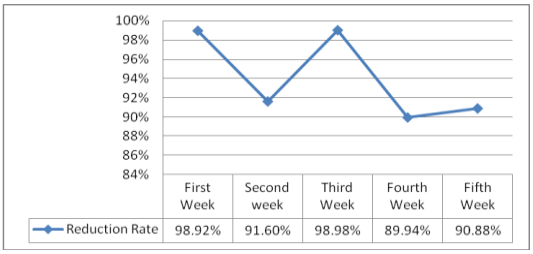

Figure 9 Reduction rate for five weeks of using the DARPA 1999 dataset

Comparing these results with those obtained by [25], [26] and [9], we find that they all used the DARPA 1999 dataset for five weeks, and the results of the proposed module are significantly better than those they have reported (Table 9).

Table 9 Comparisons between the proposed module and other approaches

Approaches	Datasets period	Input alerts	Reduction rate
Pietraszek, 2006	5 weeks	59812	60%
Jie Ma et al., 2008	5 weeks	-	90%
Al-Mamory et al., 2010	5 weeks	233615	70%
Our Approach Proposed	5 weeks	62785	92.27%

Table 9 shows that all researchers used the same dataset, but the obtained results of this module are given below.
Experiment 3: The second dataset used in this experiment for the proposed module is the NAv6 2012 dataset, which is collected as a real dataset from the NAv6 Center during two weeks from 14 May to 23 May. It has a size of 3.348 MB.
Table 10 illustrates the parameters used in this experiment. Table 11 presents the obtained results using the Reduction of IDS Alert Process module.

Table 10 Parameters used in Experiment 3

Parameters	Setting
Dataset	This experiment used a two weeks of NAv6 2012 dataset, which consist of 22685 alerts.
Time Threshold	The time threshold used < 137ms
CVE Threshold	The value of CVE threshold used < 6

Table 11 Results of the NAv6 2012 dataset for two weeks using the proposed module

Date	Amount Alerts Before Reduction	Amount Alerts After Reduction	Reduction Rate
First week 14-18 May-2012	14143	935	93.39%
Second week 19-23 May-2012	8542	889	89.59%
Total	22685	1824	91.95%

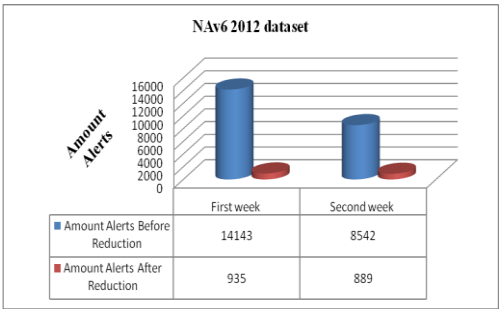

Fig. 10 Alert reduction for two weeks using the NAv6 2012 dataset

5. CONCLUSION

The most important problem confronting IDS is the generation of a large number of false alerts, which poses difficulties to network administrators when analyzing and tracking attacks. In this paper, we have proposed CMRAF, a significant framework for removing duplicate IDS alerts and reducing the number of false alerts. This framework was based on two models. The first model developed a mechanism to save IDS alerts, extract the standard features, and save them in DB file (CSV-type). The second model involved three phases. The first phase removes redundant alerts, the second phase reduces false alerts based on threshold time value, and the last phase reduces false alert based on rules of threshold CVE value.

The IDS alerts generated are analyzed in terms of the features contained in the alert. The information algorithm is used to determine the weights of these features. In addition, the features are selected in a scientific manner and with high flexibility rather than randomly without scientific support. Therefore, some researchers such as [10, 8, and 9], considered these features accordingly to reduce the large amount of false positive alerts and remove the duplicates achieved through the Reducing the IDS Alert Process module. This model is based on the NR algorithm, which removes the duplicate alerts and reduces the false positive alerts. This algorithm obtained a ratio rate of 92.3%, which is considered a good rate compared with the previous results; moreover, the truth rate of false alerts reached 99% [3].

6. ACKNOWLEDGMENTS

This research is supported by National Advanced IPv6 Centre of Excellence (NAV6) UNIVERSITI SAINS MALAYSIA (USM).

7. REFERENCES

[1] Fredrik Valeur Vigna, G. ; Kruegel, C. ; Kemmerer, R.A. ; "Comprehensive approach to intrusion detection alert correlation", Dependable and Secure Computing, IEEE Transactions on, Sept. 2004, Volume: 1 , Issue: 3 Page(s): 146 – 169, DOI: 10.1109/TDSC.2004.21

[2] Federico Maggi, Matteo Matteucci, and Stefano Zanero. "Reducing false positives in anomaly detectors through fuzzy alert aggregation", Elsevier, Information Fusion 10 (2009) 300-311. doi:10.1016/j.inffus.2009.01.004

[3] Elshoush and Osman, 2011, "Alert correlation in collaborative intelligent intrusion detection systems - A survey", (2011) Applied Soft Computing Journal, 11 (7), pp. 4349-4365.doi: 10.1016/j.asoc.2010.12.004

[4] Spathoulas, G. and Katsikas, S. (2010). Reducing False Positives in Intrusion Detection Systems, Computers & Security, 29, pp. 35–44. (Cited on pages 22, 34, 57 and 73.)

[5] Alharby, H. Imai, (2005). "IDS False Alarm Reduction Using Continuous and Discontinuous Patterns". Proceedings of ACNS 2005, Springer, Heidelberg, pp. 192-205.

[6] Viinikka, J., Debar, H., M´e, L., Lehikoinen, A., and Tarvainen, M. (2009). Processing intrusion detection alert aggregates with time series modeling. Information Fusion, 10, pp. 312–324.

[7] Jan, N., Shun-Chieh, L., Shian-Shyong, T., Nancy P., and Lin, A. (2009). Decision support system for constructing an alert classification model, Expert Systems with Applications, 36, pp. 11145-11155.

[8] Autrel, F. and Cuppens. (2005). Using an intrusion detection alert similarity operator to aggregate and fuse alerts. the 4th Conference on Security and Network Architecture Bat sur Mer, France.

[9] Al-Mamory, S. O. and Zhang, H. (2009). Intrusion detection alarms reduction using root cause analysis and clustering, Computer Communications, 32 , pp. 419-430.

[10] Julisch, K. (2003). "Using root cause analysis to handle intrusion detection alarms", PhD dissertation, University of Dortmund.

[11] Niogu and Jiawei, 2010, H. W. Njogu, L. Jiawei, "Using alert cluster to reduce IDS alerts", The third IEEE International Conference on Computer Science and Information Technology,China,9-11 July,2010, pp.467-471. DOI : 10.1109/ICCSIT.2010.5563925.

[12] Qin and Lee, 2005 Wenke Lee and Xinzhou Qin, "Statistical Causality Analysis of Infosec Alert Data" Massive Computing, 2005, Volume 5, Part II, 101-127, DOI: 10.1007/0-387-24230-9_4

[13] Ignacio Porres Ruiz and María del Mar Fernández de Ramón, "An Evaluation of current IDS", Master thesis in Information Coding, Department of Electrical Engineering, at Linköping Institute of Technology, Sweden, 2008

[14] Techtarget, 2011, http: // searchfinancialsecurity.techtarget.com / definition/ Common-Vulnerabilities-and-Exposures

[15] Mitre, 2012, http://cve.mitre.org/about/

[16] Hoang, X. D., Hu, J., Bertok, P. (2009). A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference. Journal of Network and Computer Applications, 32, pp. 1219-1228.

[17] Pietro, R.D., Mancini, L.V. (2008) "Intrusion detection systems " Handbook of Advances in Information Security S. Jajodia (Series editor) Springer ISBN 978-0-387-77265-3, e-ISBN: 978-0-387-77266-0. Volume 38, 65-92, DOI: 10.1007/978-0-387-77265-3_4

[18] Xu, D., Ning, P. "Correlation analysis of intrusion alerts", (2008) Intrusion Detection Systems, Advances in Information Security, 38, pp. 65-92. R. Di Pietro, L.V. Mancini (Eds.) Springer ISBN 978-0-387-77265-3. DOI :10.1007/978-0-387-77265-3_4

[19] Perdisci, R., Giacinto, G., Roli, F. (2006),"Alarm clustering for intrusion detection systems in computer networks", (2006) Engineering Applications of Artificial Intelligence, 19 (4), pp. 429-438. Doi: 10.1016/j.engappai.2006.01.003

[20] Yatagai, T., Keio, U., Yokohama, I. T., and Sasase, I. (2007). Detection of http-get flood attack based on analaysis of page access behavior. Proc. PACRIM 07, pp. 232-235. doi: 10.1109/PACRIM.2007.4313218.

[21] Wei-Zhou, L. and Shun-Zheng, Y. (2006). An HTTP flooding detection method based on browser behavior. Proc. of 2006 International Conference on Computational Intelligence and Security, Guangzhou, 2, pp. 1151–1154.

[22] Tjhai G. C. (2011). "Anomaly-Based Correlation Of Ids Alarms", PhD thesis, The University of Plymouth, UK.

[23] CVE Details, 2012, http://www.cvedetails.com/index.php.

[24] DARPA 1999 dataset https://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1999data.html

[25] Pietraszek. (2006). Alert classification to reduce false positives in intrusion detection, Ph.D. dissertation, Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Germany.

[26] Jie Ma, Zhi-tang L., and Hong-wu Zhang (2009). "An Fusion Model for Network Threat Identification and Risk Assessment". Artificial Intelligence and Computational Intelligence, AICI'09. International Conference, pp. 314-318. DOI 10.1109/AICI.2009.487.