Reducing the Correlation Processing Time by Using a Novel Intrusion Alert Correlation Model

Huwaida Tagelsir Ibrahim Elshoush
Computer Science Department
Faculty of Mathematical Sciences
University of Khartoum - Sudan
Email: htelshoush@uofk.edu

***Abstract*—Alert correlation analyzes the alerts from one or more Collaborative Intrusion Detection Systems (CIDSs) to produce a concise overview of security-related activity on the network. The correlation process consists of multiple components, each responsible for a different aspect of the overall correlation goal. The sequential order of the correlation components affects the correlation process performance. Furthermore, the total time needed for the whole process depends on the number of processed alerts in each component. This paper presents an innovative alert correlation framework that minimizes the number of processed alerts on each component and thus reducing the correlation processing time. By reordering the components, the introduced correlation model reduces the number of processed alerts as early as possible by discarding the irrelevant, unreal and false alerts in the early phases of the correlation process. A new component, *shushing the alerts*, is added to deal with the unrelated and false positive alerts. A modified algorithm for fusing the alerts is outlined. The intruders' intention is grouped into attack scenarios and thus used to detect future attacks. DARPA 2000 intrusion detection scenario specific datasets and a testbed network were used to evaluate the innovative alert correlation model. Comparisons with a previous correlation system were performed. The results of processing these datasets and recognizing the attack patterns demonstrated the potential of the improved correlation model and gave favorable results.**

Keywords — Alert Correlation, Alert Reduction, Intrusion Detection Systems, False Alarm Rate

I. Introduction

Intrusion Detection Systems (IDSs) play an essential role in minimizing the damage caused by different attacks. On the other hand, many of the weaknesses in traditional IDSs are due to the lack of collaborations among different detection mechanisms, and between intrusion detection and other network management operations and security mechanisms. Therefore, a Collaborative Intrusion Detection System (CIDS) architecture is introduced. In particular, a Collaborative Intelligent Intrusion Detection System (CIIDS) is proposed to include both misuse- and anomaly-based techniques, since it is concluded from recent research that the performance of an individual detection engine is rarely satisfactory. Employing multiple IDSs and other security systems gives a better view of the monitored network. They may cooperate to complement each other's coverage. Even when different detection methods are used, they analyze each other's alerts and reduce false positive alerts [1][2]. It has been proven by many researchers that collaborative approaches are more powerful and give better performance over individual approaches. In fact, the use of complementary IDSs is a promising technique as each IDS implements different detection scheme, algorithms and signatures and therefore gives a more exact and complete view of suspicious network events. Specifically, two main challenges in current CIDSs research are highlighted: CIDS architectures and alert correlation algorithms. *Alert correlation* in CIDSs will be more challenging. Deploying multiple IDSs might generate a huge number of alerts, where many are redundant, irrelevant and false positive alerts. Hence, data reduction, such as alert aggregation, alert filtering and false alert reduction, without losing valuable information is essential. The focus in this research is on correlation of Collaborative Intelligent Intrusion Detection System (CIIDS) alerts.

Automation of alert management and analysis is crucial because of the huge number of alarms, the false positives and the irrelevant alarms and to study the cause of these alarms. Thus, there is a need of alert correlation, which is a process that contains multiple components with the purpose of analyzing alerts and providing high-level insight view on the security state of the network surveillance [1][3][4]. Correlation aims to relate a group of alerts to build a big picture of the attacks, hence can be used to trace an attack to its source.

The core of this process consists of components that implement specific function, which operate on different spatial and temporal properties [5].

The correlation components are effective in achieving alert reduction and abstraction. Research shows that the effectiveness of each component depends heavily on the nature of the data set analyzed [5]. Moreover, the performance of the correlation process is significantly influenced by the topology of the network, the characteristics of the attack, and the available meta-data [6].

Since alerts can refer to different kinds of attacks at different levels of granularity, the correlation process cannot treat all

This paper is substantially extended from a preliminary version titled "An Innovative Framework for Collaborative Intrusion Alert Correlation" submitted to the Science and Information Conference 2014, August 27-29, 2014, London, UK.

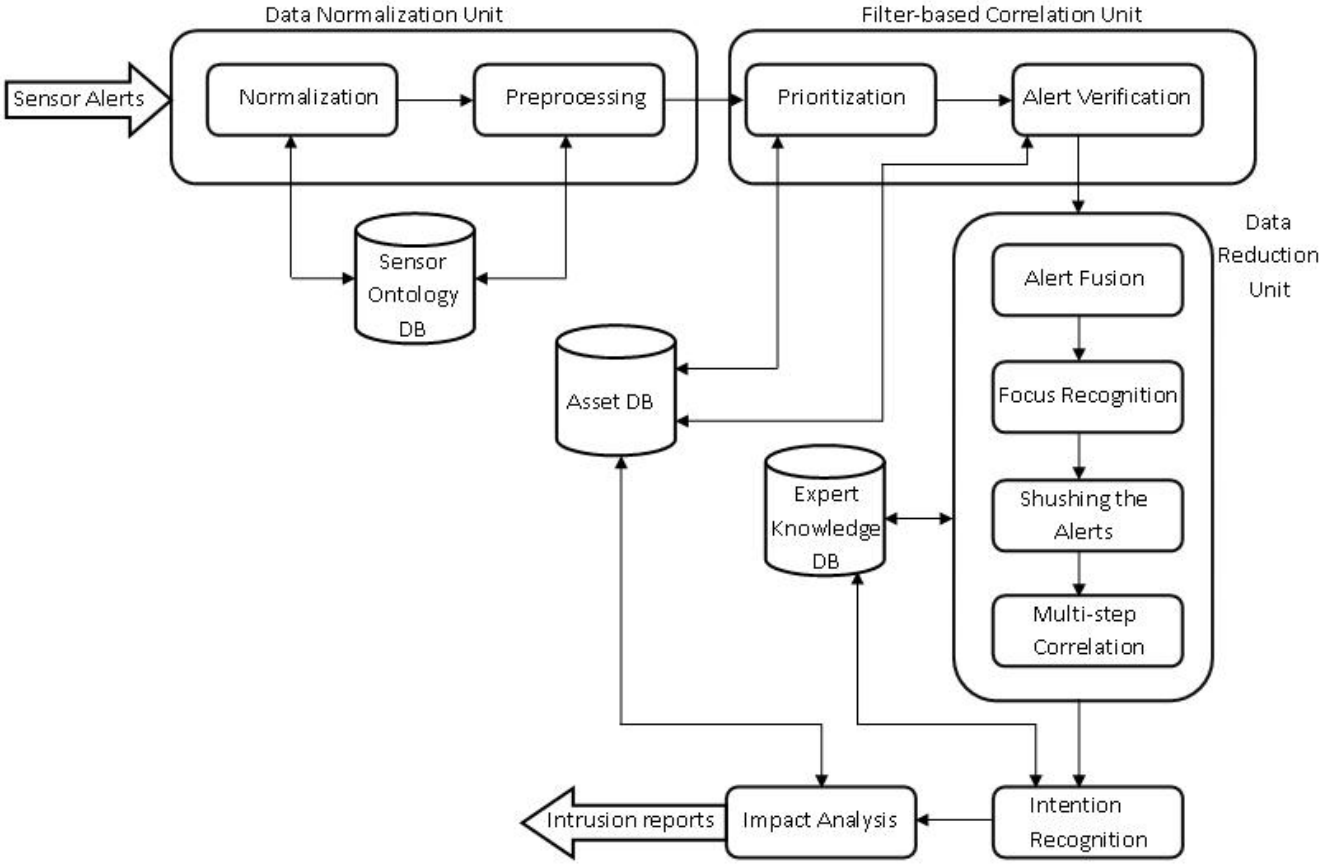

Fig. 1: The Innovative Correlation Model [7].

alerts equally. Instead, it is necessary to have a set of components that focus on different aspects of the overall correlation task. Some components, see Fig.1, e.g. those at the initial and second units, implement general functionality applicable to all alerts, independent of their type. Other components (e.g. in the third unit) are responsible for performing specific correlation tasks that cannot be generalized for arbitrary alerts, but for certain class of alerts.

Thus, one cannot, in general, determine a ranking among components with respect to their effectiveness. Each component can contribute to the overall analysis. Therefore, the most complete set of components should be used [5].

An innovative framework focuses on reordering the correlation components such that redundant, irrelevant and false alerts are reduced as early as possible thus reducing the number of processed alerts to enhance the performance. The unrelated alerts that are not correlated are dealt with in a separate component, *shushing the alerts*. Hence, the overall effectiveness of the correlation process is improved.

II. Overview of Improved Alert Correlation Framework

The proposed architecture, see Fig. 1, is composed of ten components: *normalization*, *preprocessing*, *prioritization*, *alert verification*, *alert fusion*, *focus recognition*, *shushing the alerts*, *multi-step correlation*, *intention recognition*, and *impact analysis* [7].

In the *normalization* component, alerts that are generated by multiple IDSs are collected and stored in a database before they are modeled and converted into a standard format called Intrusion Detection Message Exchange Format (IDMEF) [8]. Then data *preprocessing* is required in order to clean the data, do feature extraction and selection, and finally deal with any incomplete or missing data [9][10][11][12].

The *filter-based correlation* unit either assigns a priority to each alert or identifies irrelevant alerts. Thus, alerts are ranked based on their severity level in order to distinguish between the high and low risks alerts depending on information in the asset DB. In the *alert verification* component, alerts are checked to find out the verifiable alerts, false positives and unverifiable alerts.

Redundant alerts are fused based on similarity functions [2] in the *alert fusion* component in the *data reduction unit*. This component combines a series of alerts that refer to attacks launched by one attacker against a single target. It removes duplicates created by the independent detection of the same attack by different sensors, and also correlates alerts that are caused by an attacker who tests different exploits against a certain program or that runs the same exploit multiple times to guess correct values for certain parameters (e.g., the offsets

and memory addresses for a buffer overflow) [5][6][17].

The alert fusion component method, see algorithm 1 below, keeps a sliding timewindow of alerts. The alerts within the timewindow are stored in a time-ordered queue. When a new alert arrives, it is compared to the alerts in the queue, starting with the alert with the earliest timestamp.

A *fusion* match is found if all overlapping attributes are equal and the new alert is produced by a different sensor. Here variable *fuse* in the algorithm is set. The timestamp of the meta-alert is assigned the earlier of the sub-alerts times. The later timestamp simply indicates a delayed detection by the other sensor.

On the other hand, attack *threads* are constructed by merging alerts with equivalent source and target attributes that occur in a certain temporal proximity but the alerts need not be produced by different sensors. Hence, variable *thread* is set, see algorithm *Alert Fusion* below. The timestamp of the meta-alert is assigned the earlier of the two start-times and the later of the two end-times.

Function *fuse-merge(alert1, alert2)* in algorithm 1 assigns the different attributes to the meta-alert constructed depending on whether fuse or thread resulted.

The value of the time window should be a good trade-off between a small value, which would cause several attack threads to go undetected, and a larger value, which would slow down the system by requiring the component to keep a large number of alerts in the queue.

Following is algorithm 1, the alert fusion component method.

Algorithm 1:

Alert Fusion

```
Parameter window-size, fuse-window, thread-window
Global alert-queue, fuse, thread

fuse ← false
thread ← false

fuse(alert)
al ← get a:alert with lowest start-time from alert-queue where
if alert.analyzer ∩ a.analyzer is empty and all overlapping attributes except start-
time, end-time, analyzer, alertid are equal then
    fuse
    window-size = fuse-window
else
    if alert.victimhosts = a.victimhosts and alert.attackerhosts = a.attackerhosts
    then
        thread
        window-size = thread-window
    end if
end if
if al¬null then
    replace al in alert-queue with fuse-merge(alert, al)
else
    add alert to alert-queue
    remove all a:alert from alert-queue where
    a.start-time < (alert.start-time - window-size)
    pass removed alerts to next correlation component
end if

fuse-merge(alert1, alert2)
r ← new alert
r.alertid ← get unique-id()
r.start-time ← min(alert1.start-time, alert2.start-time)
r.reference ← (alert1.alertid ∪ alert2.alertid)
if fuse then
    r.end-time ← min(alert1.end-time, alert2.end-time)
    for each attr:attribute except start-time, end-time, reference, alertid do
        r.attr ← alert1.attr ∪ alert2.attr
    end for
    fuse ← false
else
    if thread then
        r.end-time ← max(alert1.end-time, alert2.end-time)
        r.analyzer = alert1.analyzer ∪ alert2.analyzer
        thread ← false
    end if
end if
if alert1.name = alert2.name then
    r.name ← alert1.name
else
    r.name ← "Attack Thread"
end if
for each attr:attribute except start-time, end-time, reference, analyzer, alertid do
    if alert1.attr = alert2.attr then
        r.attr ← alert1.attr
    else
        r.attr ← null
    end if
end for
return r
```

end

In the *focus recognition* component, alerts are aggregated then classified using feature similarity. Unrelated and false alerts tend to be random and will not correlate, hence uncorrelated alerts are removed by *shushing the alerts* component. Lastly, *multi-step correlation*, is expected to achieve substantial improvement in the abstraction level and data reduction [13]. In this component, priori information of the network topology, known scenarios, etc are provided by the expert knowledge DB; hence high level patterns are specified.

In the *intention recognition* component, relevant behavior is grouped into attack scenarios to extract attack strategy and plan recognition.

In the final component, *impact analysis*, the asset DB is consulted to determine all services that are dependent on a specific target. The heartbeat monitor checks whether all dependent services are still operational. If any service is failed, this information can be added to the alert as a likely consequence of the attack [5].

The asset DB stores information about installed network services, dependencies among services, and their importance to the overall operation of a network installation. So the DB does not represent an absolute measure of the importance of any asset, but rather reflect the subjective view of a security administrator. It is updated if there is any new information from the impact analysis or prioritization components.

The *knowledge DB* is a complete repository including all the necessary information about attacks, vulnerabilities, and the topological and configuration information about the protected networks and hosts.

III. Implementation and Experimental Results

In order to evaluate the improved alert correlation model performance, a collection of 10 experiments on DARPA 2000 scenarios datasets and a testbed network dataset have been carried out. Each component's function of the innovative alert correlation framework is explained in details, together with the implementation of the model based on the improved

TABLE I: Impact of Preprocessing Component on LLDOS Scenarios

	DMZ 1.0	Inside 1.0	DMZ 2.0.2	Inside 2.0.2
Input alerts	891	922	430	494
Output alerts	886	922	425	489

framework. In the implementation, Microsoft SQL Server 2005 was used as the relational database to store the alert datasets, the intermediate data, and the analysis results of each component as well as the correlated alerts. Programs written in C#, Microsoft Visual Studio 2010, were created to implement the correlation components' functionalities. The alert log files generated by RealSecure IDS of the DARPA simulation network is used [14] in eight experiments, which are explained in the next section.

A. Experiments on DARPA 2000 Datasets

DARPA 2000 [15] is a well-known IDS evaluation dataset created by the MIT Lincoln Laboratory. It consists of two multistage attack scenarios, namely Lincoln Laboratory DoS Data Sets Scenario (LLDOS) 1.0 and LLDOS 2.0.2. Each scenario includes network traffic collected from both the Demilitarized Zone (DMZ) and the inside part of the evaluation network. Eight experiments were performed, four on the improved model and four on the comprehensive approach model.

1) Data Normalization Unit:

- **Normalization**
  The alerts were already normalized, and in IDMEF standard format [8].
- **Preprocessing**
  In both scenarios, there are 45 features, of which only 7 features were extracted, namely EventID, timesec, SrcIPAddress, DestPort, DestIPAddress, OrigEventName, and SrcPort. The date attribute was represented in date/time format, and was then converted to time in seconds (represented as timesec). 5 alerts, representing incomplete data, were removed in all datasets, except for the inside segment of scenario 1.0., see Table I.

2) Filter-based Correlation Unit:

The primary goal is to reduce the number of alerts to be correlated by eliminating false, irrelevant and low risk alerts. False alerts need to be handled at an early stage as they will have negative impact on the correlation result, and moreover the number of processed alerts will be greatly reduced.

- **Prioritization**
  The ranking/priority of alerts of LLDOS scenarios from [16] is used. Thus low risk alerts are discarded, and only the medium and high risk alerts are sent to the next component. Table II shows the implementation results.

- **Alert Verification**
  This requires that the protected assets be available for real-time verification of the actual exposure of the system and/or that a detailed model of the installed network services be available. Unfortunately, this information is not available for the data set analyzed and there is no sufficient information found about the asset DB, so this component could not be implemented.

TABLE II: Impact of Prioritization Component on LLDOS Scenarios

	DMZ 1.0	Inside 1.0	DMZ 2.0.2	Inside 2.0.2
Input alerts	886	922	425	489
Output alerts	188	167	54	71
Reduction Rate	78.78%	81.89%	87.29%	85.48%

TABLE III: Impact of Alert Fusion Component on LLDOS Scenarios

	DMZ 1.0	Inside 1.0	DMZ 2.0.2	Inside 2.0.2
Input alerts	188	167	54	71
Output alerts	92	110	34	45
Reduction Rate	51.06%	34.13%	37.04%	36.62%

3) Data Reduction Unit:

Similar alerts are fused and thus data is reduced by eliminating data redundancies, and irrelevant, false and unreal alarms using alert correlation. False alerts are usually less likely to be correlated using alert correlation.

- **Alert Fusion**
  There were two sensors in DARPA data sets, but all the alerts generated by one of the sensors contained null and incomplete values and thus were removed by the preprocessing component. Thus, there were no fusion in the data set used as all traffic injected into this component were seen by one sensor, but there were thread reconstruction. Table III shows the results of the implementation.

- **Focus Recognition**
  This component has the task of identifying hosts that are either the source or the target of a substantial number of attacks. This is used to identify Denial-of-Service (DoS) attacks or port scanning attempts. It aggregates the alerts associated with single hosts attacking multiple victims (called a one-to-many scenario) and single victims that are targeted by multiple attackers (called a many-to-one scenario).
  The one-to-many scenario has two tunable parameters: the size of the timeout, which is used for the initial window size, and the minimum number of alerts for a meta-alert to be generated. On the other hand, the many-to-one scenario has three tunable parameters: the first two are the same as for the one-to-many scenario. The third parameter is the number of meta-alerts required before a many-to-one alert is labeled as a denial-of-service attack [5][6][17].
  In the carried out experiments, the minimum number of alerts in a meta-alert was two. We first applied one-

TABLE IV: Impact of Focus Recognition Component (one-to-many)

	DMZ 1.0	Inside 1.0	DMZ 2.0.2	Inside 2.0.2
Input alerts	92	110	34	45
Output alerts	42	57	23	27
Reduction Rate	56.52%	48.18%	32.35%	40%

TABLE V: Impact of Focus Recognition Component (many-to-one)

	DMZ 1.0	Inside 1.0	DMZ 2.0.2	Inside 2.0.2
Input alerts	42	57	23	27
Output alerts	31	28	5	24
Reduction Rate	26.19%	50.88%	78.26%	11.11%

TABLE VI: Impact of Shushing the alerts Component on LLDOS Scenarios

	DMZ 1.0	Inside 1.0	DMZ 2.0.2	Inside 2.0.2
Input alerts	31	28	5	24
Output alerts	6	7	3	5
Reduction Rate	80.65%	75%	40%	79.17%

TABLE VII: Impact of Multi-step Correlation Component on LLDOS Scenarios

	DMZ 1.0	Inside 1.0	DMZ 2.0.2	Inside 2.0.2
Input alerts	6	7	3	5
Output alerts	6	5	2	4
Reduction Rate	0%	28.57%	33.33%	20%

to-many focus recognition on DARPA datasets, then followed by many-to-one focus recognition. Some horizontal scan and multi-scan attacks were observed. Tables IV and V show the reduction rates. DMZ in scenario 2.0.2 shows a great RR as is expected being a multistage attack scenario.

- **Shushing the Alerts**
  As shown in [18], alert correlation can be used to differentiate between false and true alerts. False alerts and unreal alarms tend to be more random than actual alerts, and are less likely to be correlated. Thus, based on this founding, we intentionally removed the alerts that are not correlated in the alert fusion and focus recognition, resulting in Table VI, which shows great reduction rates.
- **Multi-step Correlation**
  The goal of this component is to identify high-level attack patterns that are composed of several individual attacks. The high-level patterns are usually specified using some form of expert knowledge [2][5][17][19].
  Relying on the information in [20], attack patterns are identified, and used to implement this component resulting in Table VII.

4) Intention Recognition:
Intention or plan recognition is the process of inferring the goals of an intruder by observing his/her actions [21]. It deduces strategies and objectives of attackers based on attack scenarios that are output by correlation systems. Failed attacks can be useful to know so to be avoided in the future. Using alert correlation, the intruders' relevant behavior can be grouped into attack scenarios, and later on, their attack strategy or plan can be extracted and fed back to update the expert knowledge DB.

Inadequate information of attack strategies or plans of intruders in the data set used hindered the implementation of this component.

5) Impact Analysis:
This component contextualizes the alerts with respect to a specific target network. It combines the alerts from the previous correlation components with data from an asset DB and a number of heartbeat monitors to determine the impact of the detected attacks on the operation of the monitored network and on the assets that are targeted by the attacker. Thus, it requires a precise modeling of the relationships among assets in a protected network and constant health monitoring of those assets. Hence, insufficient information of asset DB of LLDOS scenarios deters the implementation.

B. Experiments on Testbed Network

Two Snort 2.9.3.1 IDSs were installed in a Linux machine, and a Windows XP machine. Attacks were launched remotely and the alert log files were analyzed. Two experiments were performed. Tables VIII and IX show the reduction rates using the improved approach were 96.1% compared to 94.81% using the comprehensive approach.

C. Summary of Experiments on DARPA 2000 Scenarios

Tables X and XI displays the number of processed alerts and the total reduction rates using the novel approach and the Comprehensive approach respectively on each of the LLDOS scenarios 1.0 and 2.0.2.

Table XII presents a summary of the total alert reduction for each dataset. Fig. 2 illustrates the effect of the improved correlation model on LLDOS 1.0 and 2.0.2 scenarios. There is a substantial drop in the number of alerts in the priority component for all datasets. Since the processing time is proportional to the number of processed alerts, hence both Fig. 2 and 3 assured the affirmation of the better performance of the novel improved model over the Comprehensive approach.

1) Comparison of the Performance of the Improved Model with the Comprehensive Approach on LLDOS Scenario 1.0:
Tables XIII and XIV show the number of processed alerts in each component for scenario 1.0 for the improved model compared to the Comprehensive approach discussed in [5]. Since the processing time is proportional to the number of processed alerts, hence Fig. 4 shows that the improved model gives better results.

TABLE VIII: No. of Processed Alerts and the total RR using Improved Model for the Testbed Network

	Prepr.	Prio.	Fus.	1:M	M:1	Shush.	Multi	total
No. of Alerts	77	44	16	9	6	5	3	159
Reduction Rate	0%	42.86%	63.63%	43.75%	33.33%	16.67%	40%	96.1%

TABLE IX: No. of Processed Alerts and the total RR using Comprehensive Approach for the Testbed Network

	Prepr.	Fus.	1:M	M:1	Multi	Prio.	total
No. of Alerts	77	40	15	13	12	4	161
Reduction Rate	0%	48.05%	62.5%	13.33%	7.69%	66.67%	94.81%

TABLE X: No. of Processed Alerts and the Total Reduction Rates (RR) using Novel Model for LLDOS Scenarios

	Prepr.	Prio.	Fus.	1:M	M:1	Shush.	Multi	# of Processed Alerts	RR
DMZ 1.0	886	188	92	42	31	6	6	1251	99.33.%
Inside 1.0	922	167	110	57	28	7	5	1296	99.46%
DMZ 2.0.2	425	54	34	23	5	3	2	546	99.53%
Inside 2.0.2	489	71	45	27	24	5	4	665	99.19%

TABLE XI: No. of Processed Alerts and the Total Reduction Rates (RR) using Comprehensive Approach for LLDOS Scenarios

	Prepr.	Fus.	1:M	M:1	Multi.	Prio.	# of Processed Alerts	RR
DMZ 1.0	886	619	208	175	118	13	2019	98.54%
Inside 1.0	922	622	193	151	107	16	2011	98.26%
DMZ 2.0.2	425	241	63	46	44	5	824	98.84%
Inside 2.0.2	489	276	71	44	33	7	920	98.57%

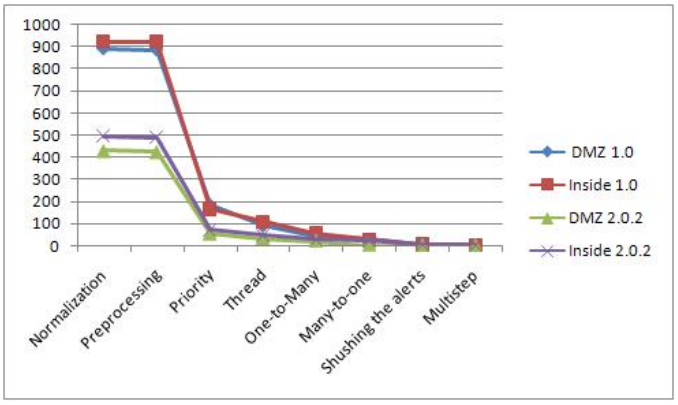

Fig. 2: Effect of Improved Correlation Model on LLDOS Scenarios 1.0 and 2.0.2.

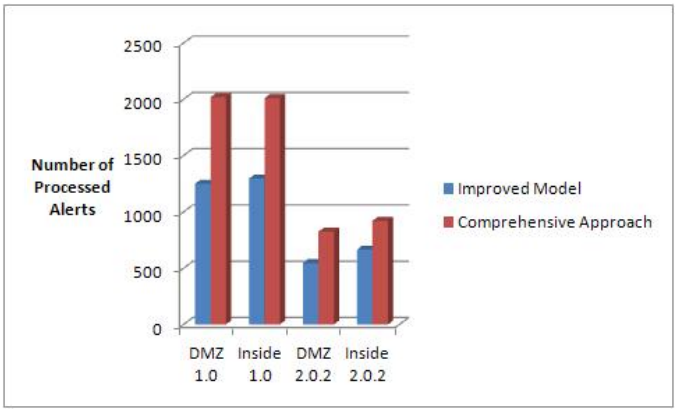

Fig. 3: Comparison of Processing Time of Improved Correlation Model and Comprehensive Approach on LLDOS Scenarios 1.0. and 2.0.2

TABLE XII: Total Alert Reduction for the Improved Model

	DMZ 1.0	Inside 1.0	DMZ 2.0.2	Inside 2.0.2
Input alerts	891	922	430	494
Output alerts	6	5	2	4
Reduction Rate	99.33.%	99.46%	99.53%	99.19%

TABLE XIII: No. of Processed Alerts using Improved Model for Scenario 1.0

	Prepr	Prio	Fus	1:M	M:1	Shush	Multi	total
DMZ	886	188	92	42	31	6	6	1251
Inside	922	167	110	57	28	7	5	1296

TABLE XIV: No. of Processed Alerts using Comprehensive Approach for Scenario 1.0

	Prepr.	Fus.	1:M	M:1	Multi.	Prio.	total
DMZ	886	619	208	175	118	13	2019
Inside	922	622	193	151	107	16	2011

2) Comparison of the Performance of the Improved Model with the Comprehensive Approach on LLDOS Scenario 2.0.2: Tables XV and XVI show the number of processed alerts in each component for scenario 2.0.2 for the improved model and the Comprehensive approach [5] respectively. The graph

TABLE XV: No. of Processed Alerts using Improved Model for Scenario 2.0.2

	Prepr	Prio	Fus	1:M	M:1	Shush	Multi	total
DMZ	425	54	34	23	5	3	2	546
Inside	489	71	45	27	24	5	4	665

TABLE XVI: No. of Processed Alerts using Comprehensive Approach for Scenario 2.0.2

	Prepr.	Fus.	1:M	M:1	Multi.	Prio.	total
DMZ	425	241	63	46	44	5	824
Inside	489	276	71	44	33	7	920

of Fig. 5 assured the affirmation of the better performance of the innovative improved model.

D. Sumnmary of Experiments on Testbed Network

1) Testing the Improved model on a Testbed Network:

The functionality of the improved alert correlation model was validated by processing this dataset and hence verifying that the attack patterns were recognized.

2) Comparing the Improved Model with the Comprehensive Approach using the Testbed Network:

Fig. 6 shows that the total number of processed alerts using the improved approach is 159 compared to 161 using the comprehensive approach respectively. The improved approach shows better results, although slight difference, but confirms the effectiveness of the improved alert correlation model.

IV. Related Work

Valeur et al in [5] presented a complete comprehensive set of components. Their experiments demonstrated that the effectiveness of each component is dependent on the data sets being analyzed, and each component can contribute to the overall performance.

Yu et al. presents a collaborative architecture for multiple IDSs to detect real-time network intrusions.The architecture is composed of three parts: Collaborative Alert Aggregation, Knowledge-based Alert Evaluation and Alert Correlation to cluster and merge alerts from multiple IDS products to achieve an indirect collaboration among them [22].

Also Depren et al. proposed a novel IIDS architecture utilizing both anomaly and misuse detection approaches, together with a decision support system to combine their results [23]. In the same year, Zhang et al. suggested a distributed IDS based on Clustering with unlabeled data [24]. Later in that year, Katti et al. presented the first wide-scale study of correlated attacks, and their results showed that collaborating IDSs need to exchange alert information in realtime [25].

Sadoddin and Ghorbani showed an overall view of the applied techniques which have been used for different aspects of correlation. The techniques were presented in the context of a comprehensive correlation framework. As high-level comparison between techniques, either competitive or

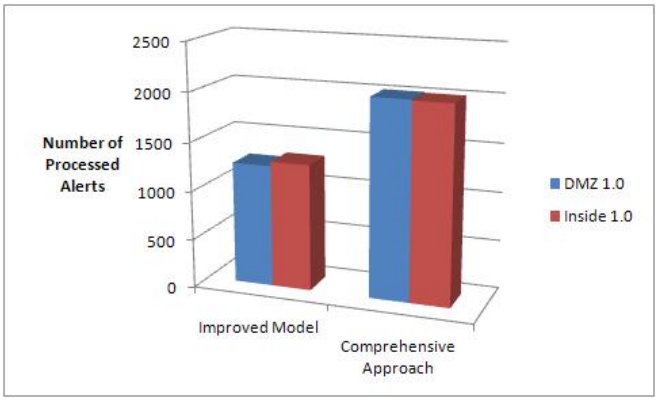

Fig. 4: Comparison of Processing Time of Improved Correlation Model and Comprehensive Approach on LLDOS Scenario 1.0.

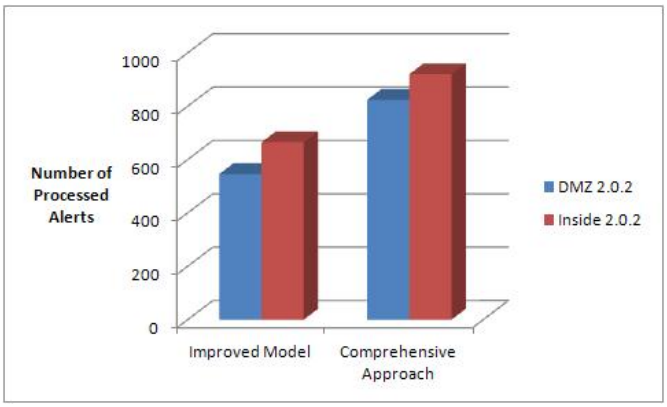

Fig. 5: Comparison of Processing Time of Improved Correlation Model and Comprehensive Approach on LLDOS Scenario 2.0.2.

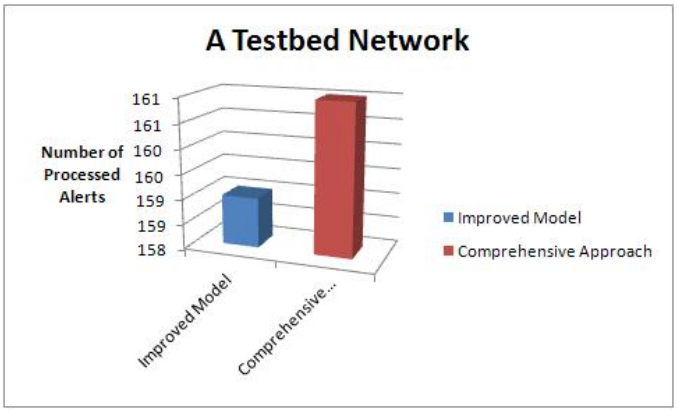

Fig. 6: Comparison of Processing Time of Improved Correlation Model and Comprehensive Approach on the Testbed Network

complementary to each, the pros and cons of the techniques were described from their point of view [26].

From the analysis in [13], researchers propose an improved solution for an alert correlation technique based on six capabilities criteria identified which are capabilities to perform alert reduction, alert clustering, identify multi-step attacks, reduce false alert, and to detect known and unknown attacks.

In February 2009, Zhou et al. proposed a decentralized, multi-dimensional alert correlation algorithm for CIDSs. A two-stage algorithm, implemented in a fully distributed CIDS, first clusters alerts locally at each IDS, before reporting significant alert patterns to a global correlation stage [3]. Later in the same year, Zhou et al. proposed a decentralized, multi-dimensional alert correlation algorithm for CIDSs. A two-stage algorithm, implemented in a fully distributed CIDS, first clusters alerts locally at each IDS, before reporting significant alert patterns to a global correlation stage. They summarized the current research directions in detecting coordinated attacks using CIDSs. In particular, two main challenges in CIDS research: CIDS architectures and alert correlation algorithms are highlighted and analyzed [1].

Sadoddin and Ghorbani proposed a framework for real-time alert correlation which incorporates novel techniques for aggregating alerts into structured patterns and incremental mining of frequent structured patterns [27].

In [28], Taha et al presented an agent-based alert correlation model. A learning agent learns the nature of dataset to select which components to be used and in which order. They proved that their method achieved minimum alerts to be processed on each component, depending on the dataset, and minimum time for correlation process. Their method differs from ours, in that they have learning agent, and we specify an order of the components which gives better performance by processing less number of alerts, hence minimum correlation time as only the high risk alerts are processed.

Ghorbani et al in [21] showed an overall view of the applied techniques which have been used for different components of an alert correlation framework.

Meinel et al in [29] identified the data storage and processing algorithms to be the most important factors influencing the performance of clustering and correlation. They proposed and implemented the utilization of memory-supported algorithms and a column-oriented DB for correlation and clustering in an extensible IDS correlation platform.

In October 2011, an alert correlation architecture is proposed by Amiri et al. Their architecture consists of four important components namely: log management, alert correlation, incident response and knowledge base system. The proposed architecture uses anomaly-based analysis in the alert correlation component. They reviewed and compared different techniques for alert correlation. Their study finally proposes that a hybrid model of multiple techniques leads to better performance of alert correlation engine [30].

Early the following year, in April 2012, Njogu et al. proposed a comprehensive approach to address the shortcomings of the vulnerability based alert management approaches. They proposed a fast and efficient approach that improves the quality of alerts as well as reduce the volumes of redundant alerts generated by signature based IDSs. Their approach has several components that are presented in three stages: Stage 1 involves alert pre-processing, correlation of alerts against the meta alert history and verification of alerts against Enhanced Vulnerability Assessment (EVA) data; Stage 2 involves classification of alerts based on their alert metrics; and Stage 3 involves correlation of alerts in order to reduce the redundant and isolated alerts as well discover the causal relationships in alerts [31].

In the same month, Soleimani and Ghorbani took a different view and consider alert correlation as the problem of inferring an intruder's actions as alert patterns that are constructed progressively. Their work is based on a multi-layer episode mining and filtering algorithm. A decision-tree-based method is used for learning specifications of each attack pattern and detecting them in alert streams. They also used a Correlation Weight Matrix (CWM) for encoding correlation strength between attack types in the attack scenarios. One of the distinguishing features of their proposed technique is detecting novel multi-step attack scenarios, using a rule prediction method. The results have shown that their approach can effectively discover known and unknown attack strategies with high accuracy. They actually achieved more than 90% reduction in the number of discovered patterns while more than 95% of final patterns were actual patterns. Furthermore, their rule prediction capability showed a precise forecasting ability in guessing future alerts [32].

In July 2012, Amaral et al. presented an automated alarm correlation system composed of three layers, which obtains raw alarms and presents to network administrator a wide view of the scenario affected by the volume anomaly. In the preprocessing layer, the alarm compression is performed using their spatial and temporal attributes, which are reduced into a unique alarm named Device Level Alarm (DLA). The correlation layer aims to infer the anomaly propagation path and its origin and destination using DLAs and network topology information. The presentation layer provides the visualization of the path and network elements affected by the anomaly propagation. Moreover, the Anomaly Propagation View (APV) is presented, which is a graphical tool developed to provide a wide visualization of the network status [33].

Lately in September of the same year, Mohamed et al. constructed a holistic solution that is able to reduce the number of alerts to be processed and at the same time produced a high quality attack scenarios that are meaningful to the administrators in a timely manner. Their proposed framework and the novel clustering method, architectured solely with the intention of reducing the amount of alerts generated by IDS. The clustering method was tested against two datasets; a globally used dataset, DARPA and a live dataset from a cyber attack monitoring unit that uses Snort engine to capture the alerts [34].

V. Conclusion and Future Work

This paper presents an innovative alert correlation framework based on a Collaborative Intelligent Intrusion Detection System (CIIDS) architecture. Alert correlation analyzes the alerts and aims to relate different alerts to build a big picture of the attack, thus giving a high-level view of the security status. The innovative framework attempts to minimize the number of processed alerts on each component and thus minimizing the correlation processing time. Hence, the correlation model components are reordered in such a way that achieves better performance by processing less number of alerts. It removes irrelevant, unreal and false alerts in the early phases of the correlation by reordering the components. Uncorrelated alerts are also dealt with in a new component, *shushing the alerts* in order to discard irrelevant and false positives. Any alert that is not correlated after being processed by a number of components is deliberately removed. An algorithm for this new component is presented. The performance is improved after the attention is focused on correlating higher severity alerts. High level patterns are specified in the multi-step component. The impact of the attack on the network assets and services is also investigated. Thus by diverting more resources to deal with high risk/priority alerts to be correlated, the effectiveness of alert correlation is significantly improved.

Further experiments and comparisons with different datasets and a real network dataset will be investigated. Several research directions exist. The generation of datasets to be utilized in evaluations is essential. Furthermore, there is a need for a mission model and its relationship to the network assets, and also a health monitoring system to determine the impact of the attacks on the network. Another research direction is the investigation of applying soft computing techniques to enhance the correlation process.

Acknowledgment

The author would like to extend her appreciation to Professor Izzeldin Mohammed Osman for providing very helpful suggestions and valuable comments. In addition, the author would like to thank the computer science department, Faculty of Mathematical Sciences, the University of Khartoum for their continuous support.

References

[1] C. V. Zhou, C. Leckie and S. Karunasekera, *A survey of coordinated attacks and collaborative intrusion detection.* Elsevier Ltd., Computer Security, pp. 1–17, June 2009.

[2] H. T. Elshoush and I. M. Osman, *Alert Correlation in Collaborative Intelligent Intrusion Detection Systems - A Survey.* Elsevier Ltd., Journal of Applied Soft Computing, **11**, Issue No. 7, 4349–4365, October 2011.

[3] C. V. Zhou, C. Leckie and S. Karunasekera, *Decentralized multidimensional alert correlation for collaborative intrusion detection.* Elsevier Ltd., Journal of Network and Computer Applications, **32**, 1106–1123, Feb. 2009.

[4] R. Bye, S. A. Camtepe and S. Albayrak, *Collaborative Intrusion Detection Framework: Characteristics, Adversarial Opportunities and Countermeasures*, August 2010.

[5] F. Valeur, G. Vigna, C. Kruegel and R. Kemmerer, *A Comprehensive Approach to Intrusion Detection Alert Correlation.* IEEE Transactions on Dependable and Secure Computing, the IEEE Computer Society, **1**, Issue No. 3, 146–169, July-September, 2004.

[6] F. Valeur, *Real-time ID Alert Correlation.* PhD Thesis, June 2006.

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

[8] H. Debar, D. Curry and B. Feinstein, *The Intrusion Detection Message Exchange Format (IDMEF).*, 2007. http://www.ietf.org/rfc/rfc4765.txt.

[9] A. Zainal1, M. A. Maarof1 and S. M. Shamsuddin, *Features Selection Using Rough-PSO in Anomaly Intrusion Detection*, 2007.

[10] A. Zainal, M. A. Maarof and S. M. Shamsuddin, *Feature Selection Using Rough Set in Intrusion Detection.*

[11] F. Amiri, M. M. R. Yousefi, C. Lucas and A. Shakery, *Improved Feature Selection for Intrusion Detection System.* Elsevier Ltd., Journal of Network and Computer Applications, Jan. 2011.

[12] J. J. Davis and A. J. Clark, *Data Preprocessing for Anomaly-based Network Intrusion Detection: A Review.* Elsevier Ltd., Journal of Computer and Security, 353–375, October 2011.

[13] R. Yusof, S. R. Selamat and S. Sahib, *Intrusion Alert Correlation Technique Analysis for Heterogeneous Log.* International Journal of Computer Science and Network Security (IJCSNS), **8** Issue No. 9, 132–138, Sept. 2008.

[14] P. Ning, *TIAA: A Toolkit for Intrusion Alert Analysis*, 2007. http://discovery.csc.ncsu.edu/software/correlator/.

[15] MIT Lincoln Laboratory 2000 DARPA Intrusion Detection Scenario Specific Datasets. http://www.ll.mit.edu/index.html.

[16] M. M. Siraj, M. A. Maarof and S. Z. M. Hashim, *Intelligent Alert Clustering Model for Network Intrusion Analysis.* Published by ICSRS Publication, Int. J. Advance. Soft Comput. Appl., **1**, Issue No. 1, July 2009, ISSN 2074-8523.

[17] C. Kruegel, F. Valeur and G. Vigna, *Intrusion Detection and Correlation - Challenges and Solutions.* Springer, a textbook, 2005.

[18] P. Ning,, Y. Cui and D. S. Reeves, *Constructing Attack Scenarios through Correlation of Intrusion Alerts.* In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington D.C., pp. 245–254, November 2002.

[19] P. Ning, Y. Cui and D. S. Reeves, *Analyzing Intensive Intrusion Alerts Via Correlation.* In Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002), LNCS 2516, Zurich, Switzerland, pp. 74–94, October 2002.

[20] Y. Cui, *A Toolkit for Intrusion Alerts Correlation Based on Prerequistes and Consequences of Attacks.* MSc. Thesis, Dec. 2002.

[21] A. A. Ghorbani, W. Lu and M. Tavallaee, *Network Intrusion Detection and Prevention: Concepts and Techniques.* Springer, a textbook, 2010.

[22] J. Yu, Y. V. R. Reddy, S. Selliah, S. Reddy, V. Bharadwaj, and S. Kankanahalli, *Trinetr: An architecture for collaborative intrusion detection and knowledge-based alert evaluation.* Elsevier Ltd. Advanced Engineering Informatics, Vol. 19:pp. 93–101, May 2005.

[23] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, *An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks.* Elsevier Ltd. Expert Systems with Applications, Vol. 29:pp. 713–722, May 2005.

[24] Y. F. Zhang, Z. Y. Xiong and X. Q. Wang, *Distributed intrusion detection based on clustring*. Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, August 2005.

[25] S. Katti, B. Krishnamurthy and D. Katabi, *Collaborating against common enemies.* USENIX Association. Internet Measurement Conference, pages 365–378, October 2005.

[26] R. Sadoddin and A. Ghorbani, *Alert correlation survey: Framework and techniques.* ACM, Markham, Ontario, Canada, Oct-Nov. 2006.

[27] R. Sadoddin and A. A. Ghorbani *An incremental frequent structure mining framework for real-time alert correlation.* Elsevier Ltd. Computers Security, Vol. 28:pp. 153–173, 2009.

[28] A. E. Taha, I. Abdel Ghaffar, A. M. Bahaa Eldin and H. M. K. Mahdi, *Agent Based Correlation Model For Intrusion Detection Alerts.* IEEE Computer Society, May 2010.

[29] S. Roschke, F. Cheng and C. Meinel, *A Flexible and Efficient Alert Correlation Platform for Distributed IDS.* Fourth International Conference on Network and System Security, 2010.
[30] F. Amiri, H. Gharaee and A. R. Enayati, *A complete operational architecture of alert correlation.* International Conference on Computational Aspects of Social Networks (CASoN), October 2011.
[31] H. W. Njogu, L. Jiawei, J. N. Kiere and D. Hanyurwimfura, *A comprehensive vulnerability based alert management approach for large networks.* Elsevier Ltd. Future Generation Computer Systems, April 2012.
[32] M. Soleimani and A.A. Ghorbani, *Multi-layer episode filtering for the multi-step attack detection.* Computer Communications, doi:10.1016/j.comcom.2012.04.001, April 2012.
[33] A. A. Amaral, B. B. Zarpelao, L. de S. Mendes, J. J. P. C. Rodrigues and M. L. Proenca Junior, *Inference of network anomaly propagation using spatio-temporal correlation.* Journal of Network and Computer Applications, July 2012.
[34] A. B. Mohamed, N. B. Idris and B. Shanmugum, *An operational framework for alert correlation using a novel clustering approach.* International Journal of Computer Applications (0975 8887), Volume 54, Issue No.12, September 2012.