International Journal of Security and Its Applications
Vol.8, No.3 (2014), pp. 393-406
http://dx.doi.org/10.14257/ijsia.2014.8.3.38

Improving the management of IDS alerts

Tu Hoang Nguyen[1,2], JiaWei Luo*[1] and Humphrey Waita Njogu[3]

[1]College of Information Science and Engineering, Hunan University, Changsha , 410082, P.R. China
[2]Centre for Informatics and Foreign Language, Hanoi University of Industry, Hanoi, Vietnam
[3]Kenya Institute for Public Policy Research and Analysis (KIPPRA), Nairobi, Kenya
hoangtu8081@yahoo.com, luojiawei@hnu.edu.cn, hnjogu@yahoo.com

Abstract

Intrusion Detection Systems (IDSs) play very crucial role in minimizing the damage caused by different computer attacks. In fact, most IDSs are capable of detecting many attacks, but often appear problematic because of triggering huge number of non-interesting alerts which diminish the value and urgency of interesting alerts. The analysts who review the alerts rarely look at the voluminous alerts until a sign is reported by other security means because it is laborious and challenging task to identify the interesting alerts. This has led to the emergence of many approaches to manage the overwhelming number of alerts. The existing approaches suffer from several limitations. This paper conducts a comprehensive study and evaluation of the key approaches that aim to manage the huge number of alerts in order to identify some research gaps that will objectively motivate researchers to come up with better approaches. At the end of the review, this paper suggests a strategy that can be exploited in order to improve the quality of final alerts.

Keywords: *Intrusion detection systems, Alert classification, Alert correlation, Knowledge based alert filtering*

1. Introduction

The ever rising of cybercrime on the global has seen a great demand for Intrusion Detection Systems (IDSs). Unfortunately, there is no single system with the ability to tackle all the network security concerns. As a result, organizations are increasingly looking for additional security technologies to counter risk and vulnerability that other security tools fail to address. IDSs are commonly used to complement other security tools in detecting network intrusions [1, 2]. Recently, IDSs have gained wide acceptance as very necessary and valuable investment in securing computer networks. They inspect traffic looking for suspicious behaviors and generate alerts or alarms to analysts so that appropriate actions could be taken.

IDSs provide an extra layer of defence to computer networks by gathering and analysing information in a network in order to identify possible security breaches. If an intrusion is detected, IDS generates a warning called an alert or alarm. Generally, there are two broad of classes of IDSs: signature based and anomaly based. The signature based IDSs generally recognize patterns of attack. This IDS essentially contains attack descriptions or signatures and match them against the audit data stream, looking for evidence of known attacks. A signature -based IDS works similar to anti-virus software. It employs a signature database of well-known attacks, and a successful match with

ISSN: 1738-9976 IJSIA

current input raises an alert. Signatures generally target widely used applications or systems for which security vulnerabilities are widely advertised. Anomaly based IDS usually looks for deviations from normal usage behavior in order to identify abnormal behaviour. Generally, the anomaly detection techniques rely on models of the normal behavior of a computer system. The anomaly based IDSs may focus on the users, the applications, or the network.

Despite several successes linked to IDSs, they are plagued by several issues making the art of accurately detecting intrusions far from perfect [3, 4]. Among the issues that contribute to poor performance of IDSs are: production of overwhelming number of alerts and high number of false positive alerts. It is estimated that an IDS may generate tens of thousands alerts per day. The vast imbalance between interesting alerts and non-interesting alerts has undoubtedly undermined the performance of IDSs [5, 6]. The interesting alerts are always buried under heaps of non-interesting alerts. Practically, no IDS can completely eliminate all non-interesting alerts [7, 8]. The issue of managing the huge volumes of alerts undermine the performance of IDSs. With limited expertise and experience, the analysts are presented with the overwhelming number of alerts and expected to evaluate them in order to identify the important and urgent alerts that are buried under tons of non-interesting alerts. As a result, the important alerts might be misclassified, misinterpreted, delayed or ignored [9].

According to Georgios *et al.*, [10], over the last few years, the research in intrusion detection has focused on the post processing of alerts in order to identify and separate interesting alerts. According to Pietraszek [9], identifying and separating interesting alerts has always been challenged by several issues such as IDSs use general signatures that hardly capture all variations of known attacks hence difficult to differentiate legitimate activities from the illegitimates ones.

The main focus of this paper is to carry out a comprehensive study of the existing works that address the management of overwhelming number of alerts generated by signature based IDS in order to identify some research gaps that need to be addressed in future in order to come up with better approaches. This paper reviews all the key works that aim to deliver alerts of high quality to the analysts. To achieve this goal, this paper considers three major techniques of alert management namely: Alert verification, Alert classification and Alert correlation. At the end of the review, the paper suggests a strategy that can be exploited in order to improve the quality of final alerts.

The rest of this paper is organized as follows. Section 2 describes the structure and the nature of alerts as well as an overview of alert management. The comprehensive review of major works of alert management and the main challenges affecting the performance of the existing systems is presented in Section 3. Section 4 describes the proposed strategy to improve the management of alerts. Finally, the paper is concluded in Section 5.

2. Alert Management Concepts

2.1. Struct and Natures Alerts

An IDS produces an alert or alarm immediately after detecting a malicious activity. Usually, an alert contains several features including: Sensor Id; Alert Id; Date and Time the alert occurred; Rule generating the alert; Source IP Address; Destination IP Address; Source Port; Destination Port; Protocol; Priority; Xref and Classification [11].

IDSs are popularly known to indiscriminately generate voluminous number of different alerts with a slight deviation from normal event or behavior [9]. Often, the raw alerts are a mixture of non-redundant, similar, unrelated, relevant, non-relevant, true alerts, false alerts, frequent, non-frequent, interesting, non-interesting, severe and non-severe alerts. According to Kruegel and Robertson [3], when a sensor outputs an alert, there are three possibilities: The sensor has correctly identified a successful attack. This alert is most likely relevant (true alert or interesting alert); The sensor has correctly identified an attack, but the attack failed to meet its objectives (non-relevant true alert or non-interesting alert); and the sensor has incorrectly identified an event as an attack giving a false alert (non- interesting alert). Figure 1 illustrates the process of alert management. An IDS produces an alert after detecting a malicious activity and then alerts are forwarded to the analysts for analysis [9]. Usually, the analysts use their knowledge to distinguish true alerts from false alerts, a task that could be frustrating and time consuming when dealing with huge volumes of alerts. After identifying the true alerts, the analysts may investigate the intrusion, identify and fix the problem causing the alert, and or modify IDS signatures to filter out the non-interesting alerts.

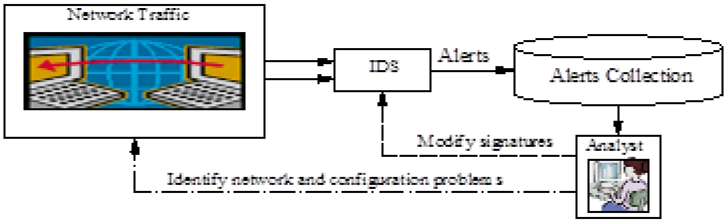


Figure 1. Overview of Alert Management

3. Existing Works

The research on alert management is gaining popularity and particularly on post processing of alerts for the last 15 years. In this section, we present the related work in three main categories Alert classification, Alert correlation and Knowledge based alert filtering.

3.1. Alert Classification

Alert Classification is one of the practical techniques studied for many years to manage alerts in order to handle the alert load. We have studied alert classification approaches as illustrated in this section. Generally, the classification methods label alerts as true or false alerts.

Pietraszek [9] propose an adaptive Learner for Alert Classification (ALAC) framework for reducing false positive. Central to the proposed approach is a classification engine that accepts the main attributes of an alert along with some background knowledge as input and estimates the probability of the alert to a true positive one. This approach depends on human intervention because the analyst must either recognize root causes of alert clusters, or label past alerts in order to train a classifier. Also the analyst must either recognize root causes of alert clusters, or label past alerts in order to train a classifier. Our experience with the alert management shows that alert classification is important but can further be strengthened by alert verification in order to differentiate between successful and failed intrusion attempts especially when dealing with signature based IDS.

Vaarandi [12] proposes a data mining based real time classification method for distinguishing important alerts from frequently occurring false alerts and events of low importance. This approach is motivated by the observations that: Most alerts are triggered by only a few signatures; If a signature has triggered many alerts over longer periods of time, it is also likely to do so in the near future; and relatively small numbers of very frequent alert pattern can be found in alerts and these alerts describe the majority of alerts that will appear in future. These alerts are well known to analysts who often review IDS alerts logs regularly.

An interesting approach is presented by Viinikka *et al.*, [13] that aggregates alerts into an alert sequence and is based on two assumptions: (i) normal system behaviours in alert flow can be identified by the regularities and smooth changes in the alert intensity. (ii) the normal behaviour is not observable at an individual alert but at alert sequence. The proposed approach models the regularities of the alert flows from the normal behaviours and used the created model to filter out the irrelevant or low impact alarms from the alarm log. Even though the approach has shown better performance, the approach presents a risk of modelling abnormal behaviours into a normal behaviour model if no abrupt changes in the alert intensity caused by true alerts have been detected.

A decision support system is proposed by Jan *et al.*, [14] that constructs alert classification behaviour patterns for on-line network behavior monitoring. It has three kinds of alert classification rule classes: normal behaviour, intrusion behavior and suspicious behaviour classification rule classes. Each class consists of a fixed number of classification rules. The proposed approach is able to run an on-line monitoring and enables the domain experts to quickly and accurately discover suspicious behaviour patterns. From the experimental results, the approach reduces the workload of on-line alert analysis for the analysts by up to 80% of false alerts. Nevertheless, the approach suffers from several common limitations such as the classification rules need to be frequently refined thus requiring high computational overhead and sufficient domain knowledge from the experts. In addition, the labeled data (rules) are not readily available in most cases and thus with a very large volume of network data encountered, it is undoubtedly expensive to manually classify them.

We noted that most of the classification methods faired well when classifying alerts however they have several weaknesses. One of the weaknesses is the human expertise involved when training the alert classifiers. The analysts need to have experience and knowledge when determining whether a suspicious alert is true or false. For example, as seen in the alert based decision support system proposed in reference [14], it may require the analysts to have experience when evaluating the suspicious alerts. Tian *et al.*, [15] propose an adaptive classifier which requires human expertise to format frequent item sets and association rules to generate a knowledge base of custom made filtering that automatically discard false alerts. Secondly, most of the alert classification approaches lack the mechanisms for verifying the validity of the raised alarms. As we are going to see in the next sub section, alert verification is advocated as an important tool to reduce noise in the alert system. Actually, the noisy alerts degrade the quality of IDS performance. To model a better alert classifier, there is need to incorporate the alert verification mechanisms to reduce unnecessary alert load before alerts are classified. In addition, most of the alert classification based approaches fail to fully discover the causal relationships between alerts. To fully discover alerts causal relationships and give more reliable results, there is need to include the concept of alert correlation when process the alerts.

3.2. Alert Correlation

The goal of any alert management system is to reduce the unnecessary alerts as much as possible. Alert correlation is the process of interpreting multiple alerts such that new meanings are assigned to these alerts. The correlation process tries to convert low level alerts to high level alerts. Generally, each step of a given attack can be identified by an IDS. The analysts need to obtain information about the detected attack based on the correlation of alerts related to different steps rather than on each single alert.

Julisch [16] presents a clustering technique for grouping all the alerts sharing the same root causes. Central to this technique is the generalization hierarchies which decompose the attributes of the alerts from the most general values to the most specific ones. These generalization hierarchies are later used for measuring the distance between alerts in clusters. The idea of generalization hierarchies seems promising for the purpose of aggregating but has the drawback of assuming a predetermined size for all clusters.

Another interesting work is proposed by Valdes and Skinner [17] to group alerts based on their overall similarity. The overall similarity of two alerts is defined based on their similarity on the corresponding features. This approach provides a basic probability model for measuring similarities between alerts. Although this method seems to effectively reduce a number of false alerts, it cannot fully discover the causal relationship between related alerts especially from multi-step intrusions. A closely related approach is proposed by Ning *et al.*, [18]. These authors present an intrusion-alert correlator based on the observation that most attacks consist of several related stages, with the early stages preparing for the later ones. Hyper-alert correlation graphs are used to represent correlated alerts in an intuitive way. Njogu *et al.*, [19] present an approach to verify, classify and prioritize alerts based on post processing of alerts. Central to this approach is the computation of alert metrics in order to further describe alerts when they are being evaluated. The approach synergizes the alert verification and alert prioritization techniques to build an effective alert management approach. The approach is fast, efficient and yields superior results. However, this approach does not reduce the redundant alerts after alert verification. A related work is presented by the same authors [20] propose a clustering approach to reduce unnecessary alerts. The approach uses vulnerability data to compute for alert metrics. It determines the similarities of alerts based on their alert metrics and the alerts that show similarity are grouped together in one cluster. The approach has extended the aforementioned approach by implementing a better architecture of building a dynamic vulnerability data that draws information from three sources *i.e.*, network resource data, scan reports from multiple vulnerability scanners and known vulnerability database.

Al-Mamory and Zhang [21] summarizes that most of the alert correlation approaches do not make full use of the information that is available on the network under consideration. In fact, the general alert correlation approaches rely on the information on the alerts which may not be comprehensive and reliable to understand the nature of attack and may lead to poor correlation results. Correlating alerts that refer to failed attacks can easily result in the detection of whole attack scenarios that are nonexistent. Thus, it is useful to integrate the network context information in alert correlation in order to identify the exact level of threat that the protected systems are facing. In the next section, we review approaches that use additional knowledge to improve the quality of alerts.

3.3. Knowledge Base Alert Filtering

The traditional IDSs often tend to generate general alerts that are not network specific because of being unaware of the network context they monitor [22]. The use of additional data such as vulnerability data is recommended to improve the quality of alerts [3, 22]. In fact verifying alerts by vulnerability assessment data helps to rule out whether an intrusion is real or not hence boosts the confidence of the analysts on alerts. The alert verification technique reduces noise in the alert system because noisy alerts degrade the quality of IDS performance. The alert quality is crucial when determining interestingness of alerts and should be associated with alerts in earliest stages of alert processing. Studies contained in [17, 21] note that working on quality of alerts is the most practical way of addressing issues of alert management and the degree of IDS performance is closely related to the quality of alerts. However, according to Colajanni *et al.*, [6], it is not possible to correct some known vulnerabilities before damaging network intrusions can take place in short time because of several reasons contained in this work. In this sub section, we describe some of the knowledge based novel approaches that improve the quality of alerts.

Gula [1] illustrates how the vulnerability information can elicit high quality alerts from huge alerts that are primarily false alerts. The author points out that a particular true intrusion targets a particular vulnerability and therefore correlating the alert with vulnerability could reveal whether the vulnerability is exploited or not. This work illustrates a variety of the approaches and theories that can be used to correlate alerts with vulnerability data. The work focuses on alerts before they are validated but not after alert verification. Similar work is presented by Porras *et al.*, [23]. The authors present vulnerability based alert management approach in the framework of the M-Correlator project where alerts are filtered and clustered according to the knowledge of the network architecture and known vulnerabilities. Closely related to the aforementioned approach is the work of Kruegel and Robertson [3]. The authors propose an alert verification system that improves the false positive rate of IDSs. However, the approach lacks useful additional information on host architecture, software and hardware.

A collaborative and systematic framework is proposed by Liu *et al.*, [24] in order to correlate alerts from multiple IDSs by integrating vulnerability information. This approach applies contextual information to distinguish between successful and failed intrusion attempts. After the verification process, the alerts are assigned confidence and the corresponding actions are triggered based on that confidence. The confidence values are: 0 for false alert while 1 is for true alert. Although the approach has some merits, it has several shortcomings. The scheme does not give details on the procedure used to validate the alerts and does not include the details of how alerts are transformed into meta alerts.

Similarly, Colajanni *et al.*, [6] present a scheme to filter innocuous attacks by taking advantage of the correlation between the IDS alerts and detailed information concerning the protected information systems. Some of the core units of the scheme are filtering and ranking units. The authors extended this work by proposing a distributed architecture [25] to provide security analysts with selective and early warnings. One of its core components is the alert ranking unit that correlates alerts with vulnerability assessment data. According to this approach, alerts are ranked based on match or mismatch of alerts between the alert and the vulnerability assessment data. We considered this type of ranking to be limiting because of two reasons: First, it relies on match or mismatch of only one feature (softwares or application) to make the decision whether an alert is critical or non-critical. Secondly, it does not show the degree of relevance of alerts hence it does not offer much help to the analyst. Further the two aforementioned approaches do not reduce the redundant and isolated alerts contained in the validated alerts.

Massicotte *et al.*, [26] propose a possible alert and vulnerability integration approach based on combination of Snort, Nessus and Bugtraq databases. The approach uses reference numbers or identifiers found on Snort alerts. The reference numbers link to Common Vulnerability Exposure (CVE). This approach is not effective because not all alerts have reference numbers. Neelakantan *et al.*, [27] propose a similar but a better method comprising of three stages: (i) find vulnerabilities in the network under consideration, (ii) create a database of vulnerabilities with reference numbers and (iii) then select signature corresponding to identify vulnerabilities. However, this approach requires reconfiguration of the signatures leading to downtime of IDS engine. And it only handles alerts with reference numbers thus lowering detection rates. Other interesting works are contained in [28, 29].

We noted that knowledge based approaches in the literature are able to validate alerts successfully. However, validating alerts cannot guarantee alerts of high quality. There is little attention given to the issue of reducing the huge number of redundant and isolated alerts contained in the validated alerts. The knowledge based approachesmay produce validated alerts containing massive number of redundant and isolated alerts from the same intrusion event and those carried out in different stages. Therefore, the knowledge based approaches need to be integrated with other techniques such as alert classification, alert reduction and prioritization thereby combining their respective advantages in order to deliver better results.

3.4. Challenging Issues and Recommendations

The detailed reviews presented in this paper demonstrate several problems or limitations of the existing works that have been carried out in past researches. The existing works suggest solutions that might overcome many of these limitations associated with management of huge volumes of alerts. After thorough and comprehensive review of the existing approaches of alert management proposed by different researchers, we made the following key observations which could motivate for better approaches for alert management:

i. As noted earlier in the literature, IDSs are usually isolated and not fully integrated into the network under consideration. As a result, they generate huge volumes of raw alerts that are a mixture of all sorts: non severe, redundant, unverified, and false alerts. Most of these alerts are not network specific useful alerts. The nature of alerts often makes analysts lose confidence in them. The validity and interestingness of an alert is not fully reflected by the values explicitly contained on its attributes. We feel that there is need to verify the validity of alerts in order to link them to known vulnerabilities. Therefore there is need touse additional vulnerability assessment data to improve the accuracy of alerts hence elicitingfinal alerts of high quality.

ii. We noted that the act of alert verification may not guarantee final alerts of high quality. In fact most of the alert verification based approaches focus much of the attention on alerts before they are verified. It is evident in the literature that little attention is given to alerts after alert verification. Therefore, we feel that future alert management approaches should also focus on alerts after alert verification because they may contain huge number of redundant and isolated alerts that need to be reduced.

iii. Most of the existing approaches use one technique to process alerts. For example, alert classification approaches only label alerts either as true or false and do not perform other equally important tasks such as prioritizing the alerts. We feel that there is need to combine several techniques when processing alerts in order to offer novel solutions.

iv. Although several efforts have been done regarding the collaboration of IDS sensors, there is need to further improve on this aspect especially when dealing with sensors of different IDS products. A better collaboration would help to reduce the unnecessary alerts significantly.

v. We noted that analysts who review the alerts regularly are in better position to identify and tell the sources (sensors) that are well known for generating high numbers of true alerts as well as sources (sensors) that are well known for generating high numbers of false alerts. They can easily notice unusual or inconsistencies in the number of alerts generated by sensors. This is possible because alerts are usually linked to their sources (sensors). Thus, the reliability of sensors generating alerts is important when processing the alerts and should be incorporated in the future alert management approaches.

vi. Lastly, it is evident in the literature that knowledge based approaches suffer from issue of having incomplete details about the vulnerabilities such as reference ids that uniquely identify different vulnerabilities. This problem worsens when dealing with alerts generated by different IDS products because each product uses its own set of reference ids which may be different with other IDS products. Although some efforts have been made to have uniform vulnerability data such as CVE, we feel there is need to incorporate reference ids from different IDS products and vulnerability assessment data in a better and more organized manner.

4. Proposed Strategy

In this section, we describe the proposed strategy in order to address the challenges of alert management. The proposed solution has three major components: Alert verification module, alert classification module and alert aggregator module. The proposed solution collects raw alerts produced by different IDS products using the alert pre-processor unit. The function of the pre-processor is to collect raw alerts and pre-process the alerts. Alert pre-processing involves converting the raw alerts into Intrusion Detection Message Exchange Format (IDMEF) and the extraction of important features of alerts. The pre-processed alerts are forwarded to the alert verification, alert classification and alert aggregation modules in that order. The final output is in form of meta alerts that are forwarded to the analyst. Figure 2 illustrates the proposed strategy.

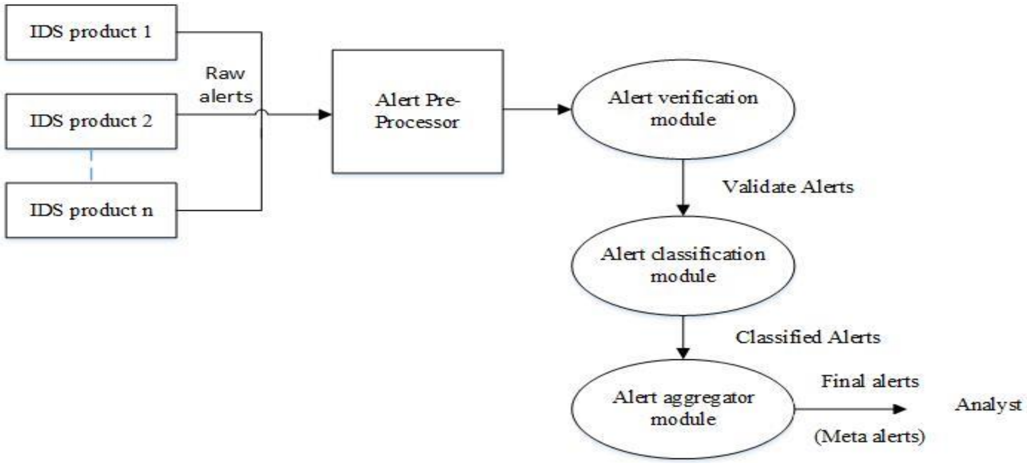
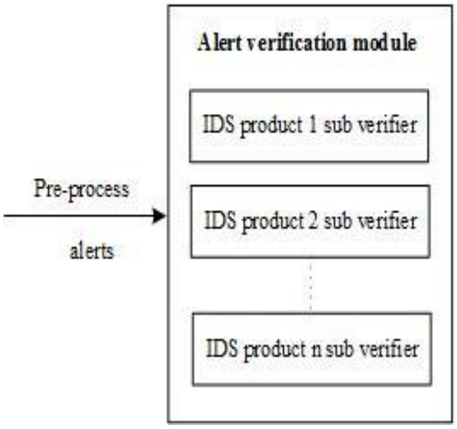


Figure 2. Proposed Strategy Framework

4.1. Alert Verification Module

Generally, signature based IDSs are run with their default configurations (signature database) and are not fully integrated with network resources. That is, IDSs do not check the relevance of an intrusion to the local network context. As a result, they generate huge volumes of raw alerts, majority of which are irrelevant alerts and are not useful in the context of the network.

The alert verification module receives the pre-processed alerts from the alert pre-processor. This module has several sub verifiers dedicated to alerts produced by different IDS products in order to simplify the process of alert verification when handling alerts of multiple IDS products as shown in Figure 3. The main function of the alert verification module is to improve the accuracy of alerts (produced by different IDS products) by validating the alerts with vulnerability assessment data. The alert verification module will validate alerts by measuring their similarity with respect to the network in order to determine the alert relevance score as illustrated in Figure 4. This will involve measuring match or mismatch of features between alert and vulnerability in vulnerability assessment data since both alert and vulnerability assessment data have comparable features. The higher the number of matches (alert relevance score), the higher the likelihood of a successful intrusion is. The alert relevance score will help to separate false alerts from true alerts. We will perform a series of regrouping of alerts on the scale with the goal of searching for the best thresholds for the true and false alerts.


Alert verification module
Pre-process
alerts
IDS product 1 sub verifier
IDS product 2 sub verifier
IDS product n sub verifier

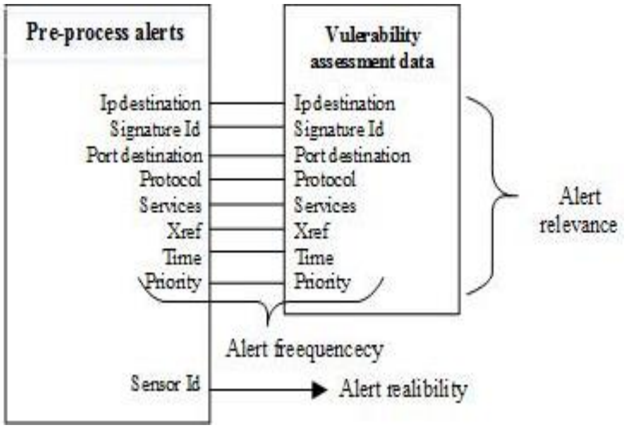


Figure 3. Alert Verification Module **Figure 4. Alert Verification**

We will design and implement a comprehensive vulnerability assessment data using the threat profile generator. The generator draws data from four sources, that is: scan reports produced by different vulnerability scanners and popular vulnerability databases such as CVE [30] and OSVDB [31], reference details from different IDS products, reports from vulnerability scanners and the details of network resources of a given network. We plan to include the aspect of how different IDS products reference the vulnerability into the vulnerability assessment data in order to have a more accurate and comprehensive vulnerabilities as Figure 5.

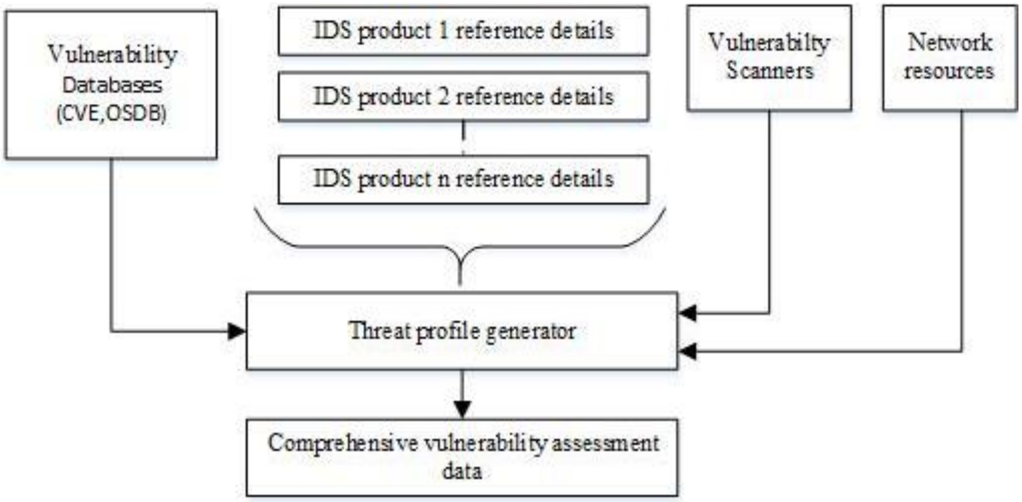

Vulnerability Databases (CVE,OSDB)
IDS product 1 reference details
IDS product 2 reference details
IDS product n reference details
Vulnerabilty Scanners
Network resources
Threat profile generator
Comprehensive vulnerability assessment data

Figure 5. Construction of Comprehensive Vulnerability Assessment Data

In order to improve the semantic of alerts, the alert verification module will include additional information to alerts such as alert relevance and alert reliability. The alert relevance is based on similarity of alerts and their corresponding vulnerabilities while the alert reliability is based on several factors regarding the IDS product (such as version of the signature, opinion of the analyst, location of IDS product).

4.2. Alert Classification Module

The alert classification based approaches generally label alerts as true or false alerts and sometimes it is difficult to reliably identify true alerts from these false alerts. As a result, it may not be comprehensive approach to classify alerts hence offering limited options to the analyst. Therefore the analysts need to have experience and knowledge when determining whether a suspicious alert is true or false. Our experience with the alert management shows that alert classification is important but can further be strengthened by alert verification in order to differentiate between successful and failed intrusion attempts especially when dealing with signature based IDS.

In our work alerts are verified prior to being classified to ensure high quality of alerts and we plan to classify alerts according to their levels of interestingness. In addition, the addition information of alerts such as relevance and reliability of alerts are taken into consideration when classifying alerts. The alert classification module has several sub classifiers for different IDS products in order to simplify the process of alert classification when handling alerts of multiple IDS products as illustrated in Figure 6.

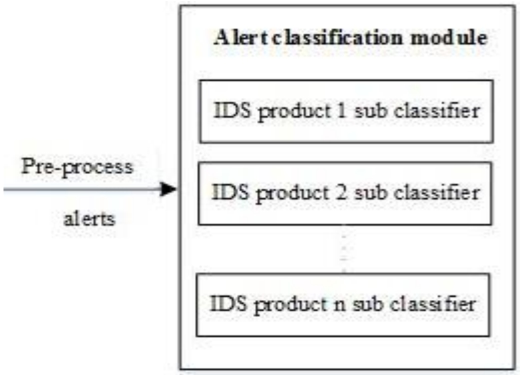

Alert classification module
IDS product 1 sub classifier
IDS product 2 sub classifier
IDS product n sub classifier
Pre-process
alerts

Figure 6. Alert Classification Module

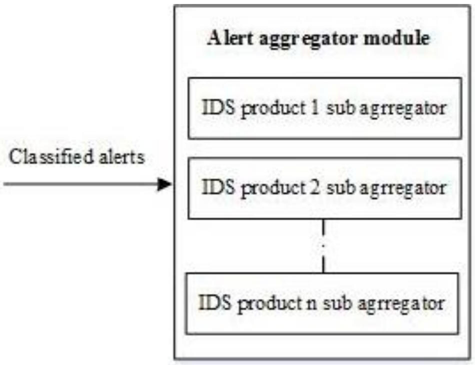

Alert aggregator module
IDS product 1 sub agrregator
IDS product 2 sub agrregator
IDS product n sub agrregator
Classified alerts

Figure 7. Alert Aggregator Module

4.3. Alert Aggregator Module

The trend of the multi-step intrusions is on rise leading to unmanageable redundant alerts. A single intrusion can generate several alerts with common features. Analysis of single redundant alerts provides partial information on the attack hence not valuable to uncover the real patterns of the attacks. There is need to aggregate the individual redundant and isolated alerts representing every step of attack to have a big picture of attacks for the different IDS products. The goal of this aggregator is to reduce the volumes of redundant and isolated alerts belonging to the same attack activity within a particular time window for the different IDS products. The proposed alert aggregation module has several sub aggregators for different IDS products in order to simplify the process of alert aggregation when handling alerts of multiple IDS products as illustrated in Figure 7.

4.4. Evaluation of the Proposed Solution

In order to validate the proposed solution, we plan to use a real-life computer network composed of different IDS products. We will evaluate the approach and consider factors such asspeed, effectiveness, ease-of-use, CPU and memory usage, and scalability. We will employ metrics such as accuracy, detection rates and alert reduction rates in order to evaluate the performance of the proposed approach.

5. Conclusion

From the literature, it is evident that an extensive research is going on in the field of alert management and most of the approaches appear promising to address the issue of managing huge volumes of alerts generated by signature based IDSs. However, there are numerous issues that continue to negatively affect the performance of the existing works. This calls for an urgent need to have a better approach that improves the quality of the final alerts. This paper has reviewed several key approaches on alert management and has compared these approaches by identifying their strengths and weaknesses. This study is particularly useful to researchers in future when designing alert management systems in order to improve the quality of alerts. Finally, we have included a strategy that can be exploited in order to improve the management of alerts.

Acknowledgements

This work is supported by Hunan Provincial Natural Science Foundation of China (Grant No. 13JJ2017).

References

[1] R. Gula, “Correlating ids alerts with vulnerability information”, Tenable Network Security, Revision 4, Technical Report, **(2011)**.

[2] C. Thomas and N. Balakrishnan, “Improvement in intrusion detection with advances in sensor fusion”, IEEE Transactions on Information Forensics and Security, vol. 4, no. 4, **(2009)**, pp. 542-550.

[3] W. Robertson and W. K. Robertson, “Alert verification determining the success of intrusion attempts”, The Proceedings of the Detection of Intrusions and Malware and Vulnerability Assessment, Dortmund, Germany, **(2004)** July 6-7, pp. 25-38.

[4] N. Hubballi, S. Biswas and S. Nandi, “Network specific false alarm reduction in intrusion detection system”, Security and Communication Networks, John Wiley and Son, **(2011)**, vol. 4, no. 11, pp. 1339-1349.

[5] T. Chyssler, S. Burschka, M. Semling, T. Lingvall and K. Burbeck, “Alarm Reduction and Correlation in Intrusion Detection Systems”, Proceedings of the International Workshops on Enabling Technologies, Infrastructures for Collaborative Enterprises, **(2004)**, pp. 229-234.

[6] M. Colajanni, D. Gozzi and M. Marchetti, “Selective alerts for the run-time protection of distributed systems”, The Proceeding of the Ninth International Conference on Data Mining, Protection, Detection and other Security Technologies, Cadiz, Spain, **(2008)** May.

[7] D. J. Chaboya, R. A. Raines, R. O. Baldwin and B. E. Mullins, “Network intrusion detection: Automated and manual methods prone to attack and evasion”, IEEE Security and Privacy, vol. 4, no. 6, **(2006)**, pp. 36–43.

[8] 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”, Advanced Engineering Informatics, vol. 19, no. 2, **(2005)**, pp. 93-101.

[9] T. Pietraszek, “Using adaptive alert classification to reduce false positives in intrusion detection”, The Proceedings of the symposium on Recent Advances in Intrusion Detection (RAID’04), Sophia Antipolis, France, **(2004)** September 15-17, pp. 102-124.

[10] G. P**.** Spathoulas and S. K**.** Katsikas, “Reducing False Positives in Intrusion Detection System”, Computer and Security, vol. 29, no. 1, **(2010)**, pp. 35-44.

[11] Snort, http:// www.snort.org.

[12] R. Vaarandi, “Real-time classification of IDS alerts with data mining techniques”, The Proceedings of the IEEE MILCOM Conference, **(2009)** October 19-21, pp. 1786-1792.

[13] J. Viinikka, H. Debar, L. Me, A. Lehikoinen and M. Tarvainen, “Processing intrusion detection alert aggregates with time series modeling”, Information Fusion, vol. 10, no. 4, **(2009)**, pp. 312–324.

[14] N. Y. Jan, S. C. Lin, S. S. Tseng and N. P. Lin, “A decision support system for constructing an alert classification model”, Expert Systems with Applications, vol. 36, no. 8, **(2009)**, pp. 11145-11155.

[15] Z. Tian, W. Zhang, J. Ye, X. Yu and H. Zhang, “Reduction of false positives in Intrusion Detection via Adaptive alert classifier”, Proceedings of the 2008 IEEE International Conference on Information and Automation, Zhangjiajie, China, **(2008)** June 20-23.

[16] K. Julisch, “Clustering intrusion detection alerts to support root cause analysis”, ACM Transactions on Information and System Security, vol. 6, no. 4, **(2003)**, pp. 443-471.

[17] A. Valdes and K. Skinner, “Probabilistic alert correlation”, The Fourth International Symposium on Recent Advances in Intrusion Detection RAID, Lecture Notes in Computer Science, Springer, UK, vol. 3224, **(2001)**, pp. 54-68.

[18] P. Ning and D. S.Reeves, Y. Cui, “Correlating Alerts Using Prerequisites of Intrusions”, Technical Report TR-2001-13, North Carolina State University, **(2001)**.

[19] H. W. Njogu, L. J. Wei, J. N. Kiere and T. Soungalo, “An efficient approach to manage IDS alerts”, International Journal of Digital Content Technology and its Applications (JDCTA), vol. 5, no. 11, **(2011)** , pp. 35-43.

[20] H. W. Njogu and L. J. Wei, “Using alert cluster to reduce IDS alerts”, The third IEEE International Conference on Computer Science and Information Technology, **(2011)**, pp. 467-471.

[21] S. O. Almam-Mory and H. Zhang, “Introduction Detection Alerts Reduction Using Root Cause Analysis and Clustering”, Computer Communications, vol. 32, no. 2, **(2009)**, pp. 419-430.

[22] B. Morin, L. Me, H. Debar and M. Ducasse, “A logic-based model to support alert correlation in intrusion detection”, Information Fusion, vol. 10, **(2009)**, pp. 285-299.

[23] P. A. Porras, M. W. Fong and A. Valdes, “A mission impact based approach to INFOSEC alarm correlation”, In The Fifth International Symposium on Recent Advances in Intrusion Detection (RAID 2002), Lecture Notes in Computer Science, Switzerland, Springer, vol. 2516, **(2002)**, pp. 95-114.

[24] X. Liu, D. B. Xiao and X. Peng, “Towards a collaborative and systematic approach to alert verification”, Journal of software, vol. 3, no. 3, **(2008)**, pp. 77-84.

[25] D. Bolzoni, B. Crispo and S. Etalle, “ATLANTIDES: An Architecture for Alert Verification in Network Intrusion Detection Systems”, Proceedings of the 21st Large Installation System Administration Conference, **(2007)**, pp. 141-152.

[26] F. Massicotte, M. Couture, L. Briand and Y. Labiche, “Context based intrusion detection using snort, nessus and bugtraq databases”, Proceedings of the Annual Conference on Privacy, Security and Trust, **(2005)**, pp. 1-12.

[27] S. Neelakantan and S. Rao, “A threat-aware signature based intrusion detection approach for obtaining network specific useful alarms”, The Third International Conference on Internet Monitoring and Protection,Romania, **(2008)**, pp. 80-85.

[28] D. Hong, “Network Intrusion Detection Algorithm Using Modified Support Vector Machine”, Advances in information Sciences and Service Sciences(AISS), vol. 4, no. 19, **(2012)** October.

[29] G. Gao, G. Miao, J. Sun and Y. Han, “Improved Semi-supervised Fuzzy Clustering Algorithm and Application in Effective Intrusion Detection System”, vol. 5, no. 4, **(2013)**.

[30] Common Vulnerability and Exposures (CVE), http://www.cve.mitre.org/about.

[31] OSVDB, http://osvdb.org.

Authors

Tu Hoang Nguyen, received his M.Sc in Computer Science from College of Information Science and Engineering,Hunan University, Changsha, Republic of China. After completing his Masters program, he got the distinguished student scholarship to for a PhD program in Computer Science at Hunan University. His research interest include Intrusion detection and prevention, vulnerability analysis, network security, bioinformatics, and data mining.

Jiawei Luo, is a full professor and vice dean at the College of Information Science and Engineering, Hunan University, Changsha, Republic of China. She holds Ph.D, M.Sc. and B.Sc. degrees in Computer Science. Her research interests include data mining, network security and bio informatics. She has a vast experience in implementing national projects on Bioinformatics. She has authored many research articles in leading international journals.

Humphrey Waita Njogu, is currently an IT Security Expert working in a government agency in Kenya. Hereceived his Ph.D and M.Sc in Computer Science (Security) from Hunan University, China. He received his B.Sc. degrees in Information Science from Moi University, Kenya. He holds several IT professional certifications in Cisco, Oracle, Comptia, Linux and Microsoft. His research interests include most aspects of IT security, with an emphasis on network security, Intrusion detection and prevention, vulnerability analysis, data mining, Green Computing, Cloud computing security and Social media security. He has a vast experience in implementing several IT security projects. He has authored many IT security research articles in leading ISI/SCI top indexed international journals and conferences.