iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VIII **Month of publication:** August 2021

DOI: https://doi.org/10.22214/ijraset.2021.37789

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Multilevel Intrusion Alert Post-processing for the Elimination of False Positives

Riyad AM[1], Shamsudeen E[2], Mohamed Jamshad K[3]
[1,3]*Assistant Professor, Department of Computer Science, EMEA College of Arts and Science, Kondotty, Kerala, India*
[2]*Assistant Professor, Department of Computer Application, EMEA College of Arts and Science, Kondotty, Kerala, India*

Abstract: ***Intrusion detection systems are the last line of defence in the network security domain. Improving the performance of intrusion detection systems always increase false positives. This is a serious problem in the field of intrusion detection. In order to overcome this issue to a great extend, we propose a multi level post processing of intrusion alerts eliminating false positives produced by various intrusion detection systems in the network. For this purpose, the alerts are normalized first. Then, a preliminary alert filtration phase prioritize the alerts and removes irrelevant alerts. The higher priority alerts are then aggregated to fewer numbers of hyper alerts. In the final phase, alert correlation is done and alert correlation graph is constructed for finding the causal relationship among the alerts which further eliminates false positives. Experiments were conducted on LLDOS 1.0 dataset for verifying the approach and measuring the accuracy.***

Keywords: Intrusion detection system, alert prioritization, alert aggregation, alert correlation, LLDOS 1.0 dataset, alert correlation graph.

I. INTRODUCTION

Digital devices and networks are very necessary resources used by everybody in the day to day life for accomplishing ones tasks. The security of various network resources is a big concern in this era since extremely critical activities are carried out through computer networks. There are a wide range of security tools used for protection mechanisms in the computer networks and resources in which intrusion detection systems (IDS) play a vital role. IDS detect various events and generates huge amount of relevant as well as irrelevant alerts. A great number of irrelevant or false positive alerts create noise which cause great difficulty in processing the alerts to get meaningful results for further actions. IDS itself have various methods for reducing false positive alarms. Still, false positives exist as the IDS are not capable for getting a bigger picture of the attack scenarios for obtaining better inference. Hence, post processing of alerts generated by the IDS is the next level of eliminating false positives unidentified by the IDS to filter out. Usually, IDS obtains input from various sensors and hosts in the network. This generates a huge amount of low level alerts. Hence, a preliminary low level alert filtering is needed since the whole system can contain various numbers of IDSs deployed in the network in a distributed fashion producing enormous duplicate alerts for the same event. After eliminating the low priority alerts, the alerts are aggregated to reduce the number of alerts by merging group of similar alerts to a hyper alert. After the aggregation process, the hyper alert correlation is done for finding real attacks and eliminating others for further reduction of false positives. Various methods are available for correlating alerts. Usually various probabilistic approaches are carried which doesn't need any prior knowledge. But, these approaches fail to find the real causal relationship among the alerts. We propose an alert correlation approach that finds various stages of attacks where previous event prepares for the later ones. We can obtain high performance with the support of knowledge base. Alert correlation graphs are constructed to identify the attacks with their stages. Finding the causal relationships between the various events will help to identify various stages of attacks which will even help to predict the upcoming attack stages helping the system to take precautionary measures. The remaining sections of this paper are organized as follows. Section 2 discusses the literatures reviewed for the accomplishment of the proposed approach. Section 3 explains preliminary alert prioritization and filtering phase. Section 4 describes the higher level alert processing for false positive elimination. Experimental results are depicted in the section 5 and the paper is concluded in the section 6.

II. LITERATURE SURVEY

Network security is accomplished by providing availability, integrity and confidentiality of critical information. The idea of intrusion detection was first introduced by James P Anderson in his technical report on intrusion detection systems [1]. The intrusion detection systems are basically categorized into signature based and anomaly based systems. Signature based IDS detects the intrusions by comparing with previously known intrusion detection signatures. They are also known as misuse detection. On the other hand, anomaly based IDS searches for the deviations from the normal activities [2].

The IDS are also categorized as network based and host based IDS where network based IDS captures network traffic for finding intrusions while host based IDS collects information from host security log files to find intrusions [3]. Intrusion detection systems can be implemented in the form of standalone system as well as multiple IDS modules distributed over the network [4]. Distributed IDS is capable of finding distributed and collaborated attacks which are difficult to identify with single IDS [5]. Performance of IDS is judged by its attack detection rate and false positive elimination capability. False positives are the normal events labelled by IDS as intrusions. It is a real painstaking activity to eliminate false positives since any effort on increasing the performance of detection will eventually increase the chance of false positives. The increase in false positive rate is a serious issue as even a high performance IDS will be practically unusable due to the high rate of false positives. The false positives can be minimized in the IDS level where the IDS eliminates false positive during the detection phase itself or in the post processing level where the IDS alerts already generated is further evaluated for eliminating false positives [6].

Post processing of intrusion alerts is comparatively a new field. There are various techniques used for eliminating false positives during the post processing of intrusion alerts. In this, normalization is a technique where the alert data is transformed to a standardized format for post processing activities [7, 8]. It also eases the seamless communication between various alert processing components. Prioritization is another technique through which the alerts are rated and prioritized since the importance of each alert varies [9, 10]. Alert aggregation is another alert processing technique through which tremendous reduction in the number of alerts is performed by aggregating similar alerts into a single hyper or meta-alert [11, 12]. This reduces the processing effort of the following post processing activities significantly. Alert correlation is the most promising method among alert post processing methods [13, 14, 15, 16]. The idea behind the correlation of alerts is to find the causal relationship among the alerts. The algorithm is usually applied on aggregated hyper alerts. In correlation, the related alerts are verified for identifying possible attack scenarios. This is a fantastic approach which can identify more bigger and collaborated attacks like distributed denial of service (DDOS) attack which is virtually impossible for a standalone IDS to detect. Alert correlation graphs are used to correlate and find multi stage attacks [17, 18, 19]. Through alert correlation, it is also possible to eliminate scenarios which are not attacks eventually eliminating false positives. There are similarity based, statistical based and knowledge based correlation techniques. Correlation with prerequisite and consequence knowledge is a promising approach which can provide very high accuracy due to the support of knowledge base [20, 21, 22].

III.PRELIMINARY ALERT POST-PROCESSING AND FILTERING PHASE

Maximum elimination of false positive alerts in an exhaustive fashion is carried out here as part of preparing quality alerts for further processing. A multilevel false positive elimination approach is conducted in which extensive reduction of alarms is done which in turn reduces the false positives. Here, in the preliminary alert filtering stage, a huge amount of unwanted alerts generated by various intrusion detection systems are filtered out. Low level alerts generated from various sensors are first normalized for further processing. This is the normalizing stage. The normalized alerts with various details are then taken for preliminary prioritization in order to filter out irrelevant alerts. The multilevel false alarm reduction layout is depicted below in the figure 1.

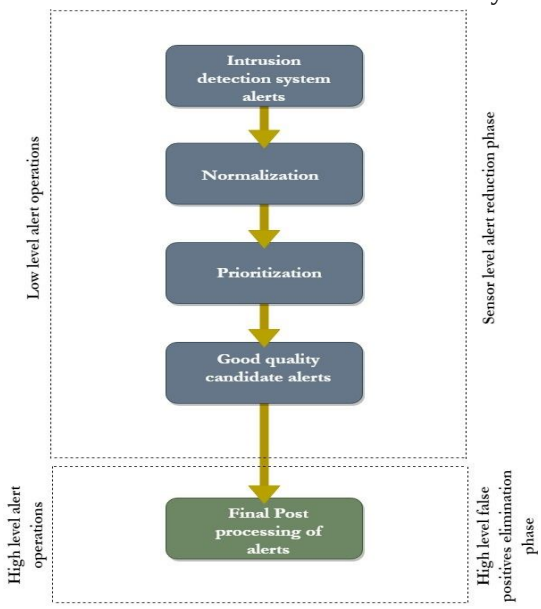


Fig. 1 Preliminary low level alert prioritization phase forwarding only relevant alerts to the next phase

A. Normalization of Alerts

First of all, the alerts are collected from various IDSs in the network which contains features such as alert number, sensor number, event number, time stamp, attack type, IP addresses of source and destination machines and port numbers of source and destination machines.

The normalization component transforms the raw alerts from the IDS to a standardized format. Here, in our system, we use Intrusion Detection Message Exchange Format (IDMEF) [23] which enables easy sharing between various IDS agents and coordinator agents in a collaborated distributed framework. It is a standardized input for various false elimination modules. It is an object oriented model for alert data.

B. Prioritization of low level alerts

The normalized alerts are utilized for further filtering activities. Basically two information bases are used for the preliminary prioritizing phase such as vulnerability information database and network and host configuration database. With the virtue of these databases a priority value is given for each alert which facilitate the filtering mechanism accordingly. The preliminary alert filtering and prioritization phase is depicted in the figure 2 below.

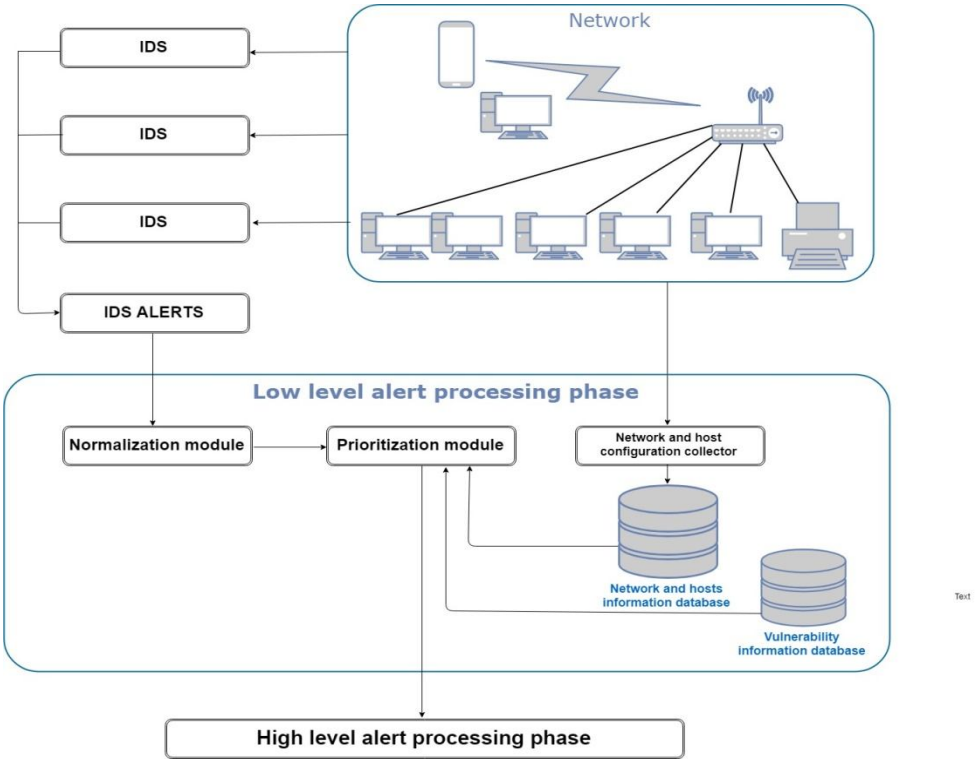


Fig. 2 Low level alert prioritization module

C. Vulnerability Information Database

The vulnerability information database is formed mainly using National Vulnerability Database (NVD) [24] which is primarily based on CVE (Common Vulnerabilities and Exposures) [25]. These databases are up to date. Vulnerability databases are freely available for download. The databases are updated regularly for the best performance of the relying processes. With the knowledge from the Vulnerability Information Database, the network and host information database is verified which is discussed below.

D. Network and Host Information Database

Network and host configuration collector gathers all the details of the network and the devices installed in the network and stores in the network and hosts information database. It contains the information about the devices, the ports they listen and details of various software installed including the operating systems. Details of the vulnerabilities and exploits in the network and hosts are stored in this database. It also monitors the live hosts. Whenever an alert is generated by IDS, an initial filtering process is carried out for checking the relevancy of the alert produced. Here, the destination IP, destination port and the vulnerabilities in the target device software and services are checked from the respective database to identify whether it is an attack or not.

This strategy is highly promising one and will filter out a huge number of irrelevant alerts since most of the false positive alerts are formed due to the context unawareness of the system. This will help to reduce a big amount of noise during the post processing activities done in the next phase for finding the false positives. An attack targeting a host which is not alive or non-existing/irrelevant port or software service can easily be discarded without further scrutiny. Attack on apache in a device running another web server is an example. An alert with an external source IP trying to access the internal resource will be considered more than an internal source IP accessing an internal device. The prioritizing module will allot priorities to the alerts according to their relevance. This will allot an alert with a priority from 0 to 1, where 0 represents a complete false positive and 1 represents a complete true alert. The priority number between 0 and 1 will reflect the amount of relevance of the alert. We set a priority threshold of 0.4 and above as a relevant alert that is forwarded for further verification to higher level aggregation and correlation modules. A pseudo code of the alert priority algorithm is depicted below.

```
Algorithm: AlertPriority
Input:   IDS alert
Output: Alert with priority P, where 0 ≤ p ≤ 1
Begin
Alert features, AF[ ] = AlertDecode (AlertNumber)
Aggregated vulnerability,
AV = VulnerabilityCalc(AF[ ])
(AV is the average of host, port and software services vulnerability and non existence of any of them will set the priority to 0)
If AV≥0.4
    Consider the alert as a candidate of true positive.
Else
    Mark the alert as false positive
End
```

Fig. 3 Prioritizing all low level alerts.

E. Experiment conducted with LLDOS 1.0 dataset

The dataset used here for experiment is DARPA 2000 LLDOS 1.0 inside and LLDOS 1.0 dmz dataset [26]. The result of the experiment conducted is shown in the table 1. It is observed from the result that more than half of the irrelevant alerts of inside and dmz datasets are filtered to reduce the noise for further investigations.

TABLE 1
Table showing the rate of low level alerts eliminated in the preliminary phase

	Dataset	Total number of alerts	Number of filtered alerts with priority <0.4	Number of relevant alerts with priority ≥0.4	Percentage of alert reduction
LLDOS 1.0	Inside	1326	678	648	51.13%
	Dmz	1285	682	603	53.07%

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

IV. HIGH LEVEL ALERT POST-PROCESSING PHASE

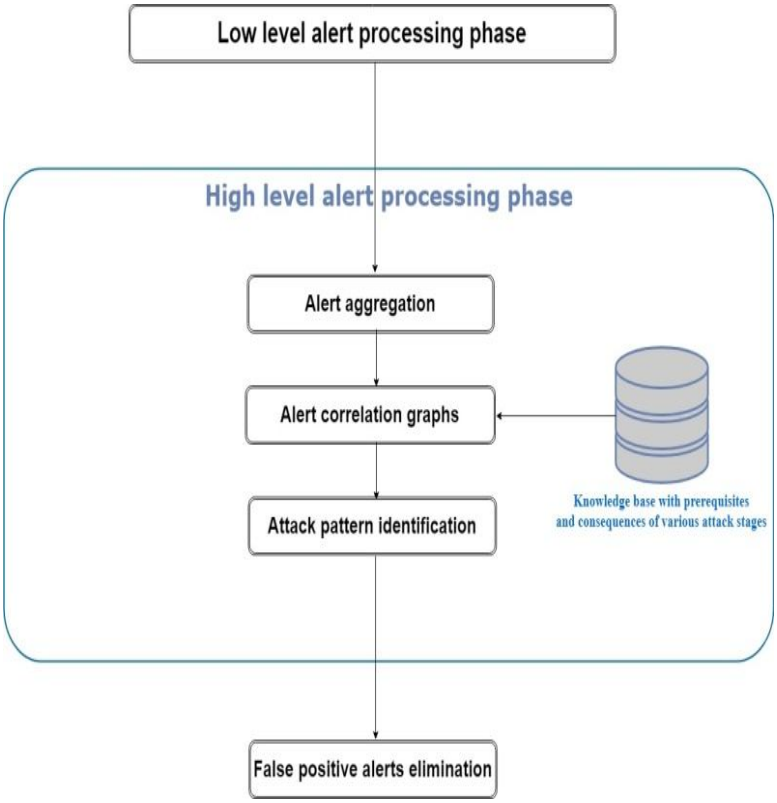


Fig. 4 High level alert aggregation and correlation module

The figure 4 shows the high level alert processing phase. Here, the highly prioritized low level alerts are taken to the high level aggregation and correlation phase. This is the phase were maximum false positive alerts are eliminated which were unidentified in the preliminary phase.

A. Aggregation of pre-processed Alerts

After removing the irrelevant alerts in the preliminary phase, maximum false positive alert elimination is carried out using aggregation and correlation components. Since the system is arranged in a distributed manner for identifying attacks in a coordinated fashion, obviously there will be multiple alerts generated for a single event. Duplicate alerts will be there due to the spatial and temporal constraints in the network. Redundant alerts are irrelevant ones and hence false positives. So, aggregation of the similar alerts to a single hyper alert is done. Aggregation can be done based on similarities between relevant features of alerts in a particular time interval. For this purpose source IP, destination IP, alert number and time stamp is used to aggregate similar attack types. Pseudo code of the alert aggregation algorithm is depicted below.

Algorithm: AlertAggregation

```
Input:    Array of individual alerts A[ ]
Output: Array of aggregated alerts AA[ ]
Begin
Create an array AT[ ] of distinct attack types
for atk = 0 to AT[ ].size (where “atk” is the index in AT[ ])
    Create an individual array IA[ ] from A[ ] with all     alerts for  the attack type in AT[atk]
    set all IA[ ].reduntant = false
    ca = 0(where index “ca” is the candidate alert index in IA[ ] and ‘n’ is length of IA[ ])
    {
```

```
while(IA[ca].redundant = false)
{
nca = ca + 1, (where "nca" is the index of next candidate alert for comparison)
while (nca < IA[ ].size)
    {
    if (TimeDiff(IA[ca].timestamp, IA[nca].timestamp) < threshold AND IA[ca].sourceIP=IA[nca].sourceIP AND
      IA[ca].destIP=IA[nca].destIP)
      IA[ca].timestamp = IA[nca].timestamp
      IA[ca].redundant=false
      IA[nca].redundant=true
    end if
    nca = nca +1
    }
ca = ca + 1
}
for i = 0 to IA[ ].size
if (IA[i].redundant = false)
   AA[ ].add(IA[i]) (Add to the aggregated alerts array)
end if
next atk
```

Fig. 5 Alert aggregation for generating hyper alerts.

In the above alert aggregation algorithm, each distinct attack types are identified first. Then each attack type is taken into consideration one at a time. Array of alerts of a particular attack is compared each other with the aggregation criteria given below.

1) Alerts should be within a time threshold.
2) Source IP address should be the same.
3) Destination IP address should be the same.

All the other alerts are discarded whose redundant value is true and a representative alert is taken as the hyper alert. This process is continued for obtaining all hyper alerts for various attack types. If we increase the time threshold, the degree of aggregation is increased. But there is a possibility of omitting some distinct attack events. Hence, the time threshold is fixed accordingly.

The LLDOS 1.0 scenario specific dataset [26] was used to experiment the effectiveness of the algorithm and the following results were found which is depicted in the table 2. From the table it can be found that a huge number of alerts are aggregated to lesser number of hyper alerts. 70.99% of LLDOS 1.0 inside dataset and 74.79% of LLDOS 1.0 dmz dataset was reduced and hence eliminating the false positive alarms considerably.

TABLE 2
Result of aggregation showing the alert reduction

Dataset	Number of alerts	Number of hyper alerts	% of aggregation
LLDOS 1.0 Inside	648	188	70.99%
LLDOS 1.0 DMZ	603	152	74.79%

B. Hyper Alert Correlation for Identifying Attack Graphs

Pseudo code of the hyper alert correlation graph algorithm is depicted below.

Algorithm: HyperAlertCorrelation

Input: Array of hyper alerts

Output: Hyper alert correlation graph HACG(V,E), a connected direct acyclic graph where V is the vertex which is a set of hyper alerts and E is the set of edges where each edge is an edge between two vertices if first vertex prepare for the next.

Begin

HACG(V, E) = ø

for i = 0 to n (where 'n' is the number of hyper alerts ha)

if (ha_i.timestamp – ha_{i+1}.timestamp < threshold) AND (ha_i.timestamp < ha_{i+1}.timestamp)

if prepare(ha_i.consequence, ha_{i+1}.prerequisit) = true

HACG ←HACG ∪ HACG (ha_i,ha_{i+1})

end if

end if

next i

return

Fig. 6 Hyper alert correlation graph generation.

In the hyper alert correlation graph algorithm above, array of hyper alerts are given as input. A hyper alert correlation graph is constructed for identifying the causal relationship among hyper alerts. The hyper alerts are compared to each other if the time difference between the hyper alerts comes under a predefined threshold value and the event time of first alert is less than the second one. A knowledgebase of prerequisites and consequences of various types of attacks are compared with the respective hyper alerts to find whether there is a causal relationship. A hyper alert prepares for another one if consequence of previous attack is matched with the prerequisite of current hyper alert. If this satisfies, the alerts are added to the correlation graph.

V. IMPLEMENTATION AND RESULTS

The figure 7 shows the experimental layout where nodes in the network are connected with switches. The hosts in the network are configured with core i7 3.4Ghz with 8GB RAM. Operating system used was 64 bit Windows 8.1. The network was connected to the outside network through router and firewall. We use IDS in the form of mobile agents which use the ensemble classification technique [27]. The multiple sensors collect the data and stored in the databases where mobile analysis agents in JADE environment [28, 29] analyses the events. The alerts produced were normalized and prioritized in the preliminary filtering stage discussed before. The quality alerts with higher priorities are aggregated using the alert aggregation algorithm already explained. These hyper alerts were served to the correlation process for generating correlated attack graphs. Alert post processing programs were written in java and SQL Server database was used for storage purpose. In order to experiment LLDOS 1.0, Tcpreplay 2.3.2 [30] was used as the replay tool to feed the sensors in the network which are eventually processed by various phases ending with generation of correlated attack graphs. The results obtained were highly promising and discussed in subsequent sections. The experiment was conducted with the architecture depicted in the figure 7 given below.

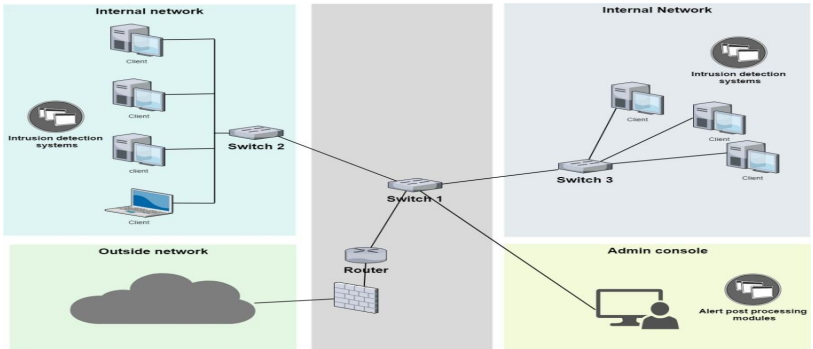


Fig.7 Experimental environment.

A. Dataset used for Implementing the Approach

The LLDOS 1.0 scenario specific dataset was used which was created by MIT Lincoln Labs [26]. Various alert processing methods were carried out using this dataset in different post processing stages. Using this dataset will facilitate comparing our work with various other works. The dataset contains attack with multiple steps.

The steps are as follows,

1) In the initial stage, attacker scans the whole network to obtain the active hosts.
2) The attacker use ping for sadmint exploit on the hosts found in the previous step.
3) Now, the attacker uses the sadmint vulnerability to attain the root privilege.
4) After getting the root access, the attacker installs the DDOS malware in the compromised hosts and makes the machine DDOS master.
5) The master machine launches the DDOS attack.

B. Final attack patterns identified from the LLDOS 1.0 inside data set

The attack graphs discovered for LLDOS 1.0 inside dataset is shown in the figure 8. There are two attack patterns identified from the hyper alert correlation graph.

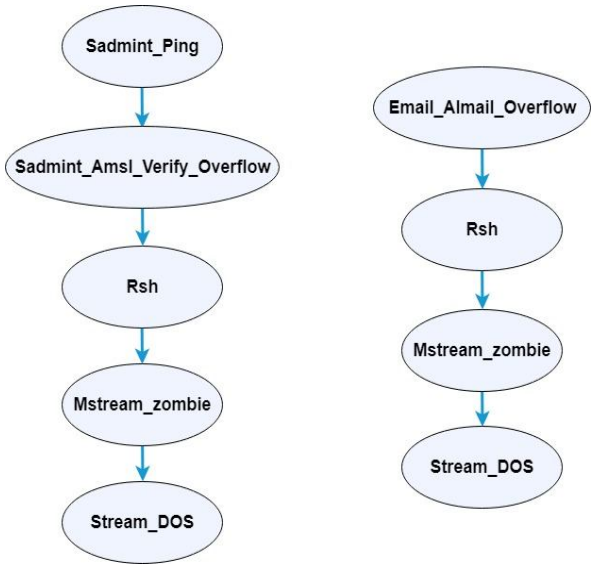


Fig.8 Attack graphs identified from LLDOS 1.0 inside dataset.

C. Experiment with inside and dmz traffic of LLDOS 1.0 datasets showing detection rate and false positive rates

The details are furnished in the table 3 below. From the experiment conducted on LLDOS 1.0 inside and dmz datasets, the following results were obtained. In the inside and dmz data set, very high detection rate is maintained such as 95% and 95.5% respectively even with correlation process that eliminates false positives. Similarly, the false positive rate was remarkably decreased from 69.15% to nil in inside dataset and 43.42% to 1.15% in dmz dataset.

TABLE 3. Detection rate and false positive rate of the proposed approach

Dataset	Method	Number of attacks	Number of detected attacks	Detection rate	Number of alerts	Number of true alerts	False positive rate
LLDOS 1.0 Inside	IDS only	60	57	95%	188	58	69.15%
	IDS with Correlation	60	57	95%	58	58	0
LLDOS 1.0 DMZ	IDS only	89	85	95.5%	152	86	43.42%
	IDS with Correlation	89	85	95.5%	87	86	1.15%

D. Correlation Accuracy for LLDOS 1.0 Dataset

In order to test the accuracy of the approach, two measures such as completeness and soundness described by ning et al [31] was used. Completeness denotes the capability of the approach to correctly correlate the related alerts. Soundness refers to the ability of the approach to accurately correlate alerts. The values for these measures are calculated with the formulae given below.

$$R_c = \frac{\text{Number of correcly correlated alerts}}{\text{Number of related alerts}} \quad (1)$$

$$R_s = \frac{\text{Number of correcly correlated alerts}}{\text{Number of correlated alerts}} \quad (2)$$

TABLE 4
Measurement of Completeness and soundness

	LLDOS 1.0 Inside	LLDOS 1.0 Dmz
Number of related alerts	59	89
Number of correlated alerts	58	87
Number of correctly correlated alerts	58	86
Completeness, R*c*	96.61%	96.63%
Soundness, R*s*	100%	98.85%

The completeness and soundness of the approach is shown in the table 4 above. The values obtained prove the effectiveness of the approach.

E. Comparison of Completeness and Soundness Between various works Implemented with LLDOS 1.0 dataset

The performance of the approaches has been measured with the help of completeness and soundness. The values obtained with our approach are compared with other approaches that have used the LLDOS 1.0 dataset and calculated the completeness and soundness measures. The comparison done is depicted in the table 5 below showing the accuracy of our approach and others with respect to inside and dmz dataset. The results are highly promising.

TABLE 5
Comparison of completeness and soundness between various approaches

Works	Completeness %		Soundness %	
	LLDOS 1.0 Inside	LLDOS 1.0 Dmz	LLDOS 1.0 Inside	LLDOS 1.0 Dmz
Ning et al.[32]	93.18%	94.74%	93.18%	94.74%
Safa et al.[33]	83.3%	89.7%	100%	100%
Bateni et al.[34]	94.1%	-	95%	-
Our work	96.61%	96.63%	100%	98.85%

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

F. Comparison of detection rates between various works implemented with LLDOS 1.0 dataset with respect to detection rate

The detection rate of our system when compared with others is depicted in the table 6 given below. Minimizing the false positives while maintaining the detection rate is a challenging task. From the table it can be seen that our system outperforms others with respect to the detection rate while keeping the false positives near to nil.

TABLE 6
Comparison of the proposed work with other approaches with respect to detection rate

	Ning et al.[32]	Safa et al.[20]	Yu et al.[35]	Alserhani et al.[36]	Our work
Detection rate for LLDOS 1.0 Insider	60%	65%	93.8%	92%	95%
Detection rate for LLDOS 1.0 Dmz	56.18%	64.05%	93.8%	92%	95.5%

G. Comparison between various works implemented with LLDOS 1.0 dataset with respect to false positive rate

The table 7 shown below compares our approach with various other approaches who implemented LLDOS 1.0 dataset. It can be seen that the rate of false positive is almost nil with our approach showing that our primary objective is accomplished and the method can be implemented without the chaos generated by the false alarms.

TABLE 7
Comparison of the proposed work with other approaches with respect to false positive rate

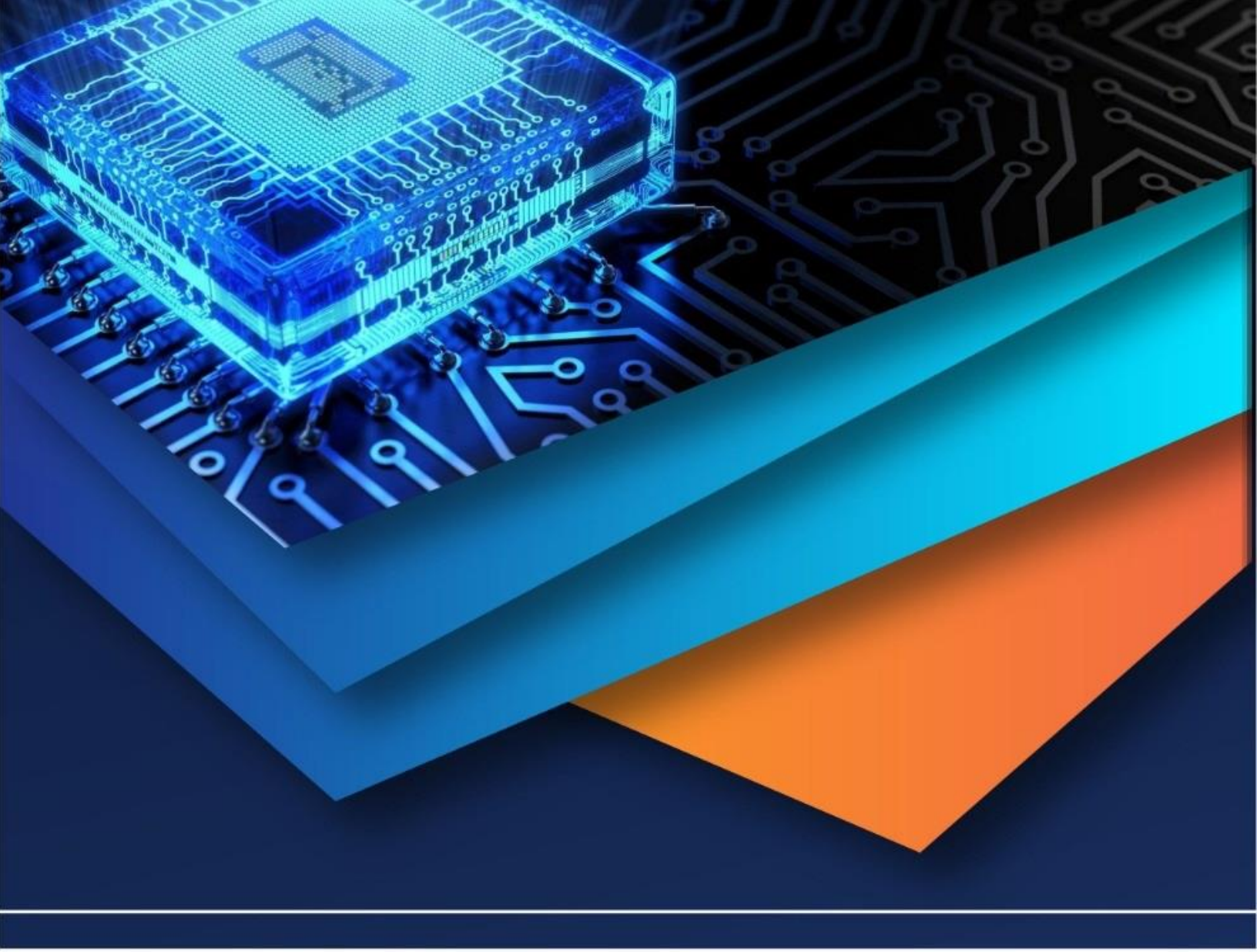
	Ning et al.[32]	Alserhani et al.[36]	Belaton et al.[37]	Yu et al.[38]	Our work
False Positive rate for LLDOS 1.0 Insider	6.82%	8.1%	35.57%	0	0
False Positive rate for LLDOS 1.0 Dmz	5.26%	8.25%	32.44%	20%	1.15%

VI. CONCLUSIONS

There is a huge difficulty in implementing and deploying the current intrusion detection systems due to high rate of false positives alarms. Even though intrusion detection systems are capable of eliminating false alarms up to an extent, there exits false positives creating headache for security administrators. For the maximum elimination of the false positives, post processing of intrusion alerts is done. We proposed here a multilevel false positive elimination framework. Intrusion alerts are normalized to a standard format. Preliminary prioritization of alerts is done for filtering a huge number of irrelevant alerts. An aggregation algorithm is proposed for aggregating the alerts to fewer hyper alerts. Finally, an alert correlation graph is constructed to find the causal relationship among these hyper alerts. LLDOS 1.0 dataset was used for implementing various stages of the alert reduction and false positive elimination mechanism. From the results it was found that the approach is highly promising in eliminating maximum false positives with superior detection rate and very high accuracy when compared with the existing methods.

REFERENCES

[1] J.P Anderson. “Computer security threat monitoring and surveillance”, Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, April 1980.
[2] H Debar, M Dacier, and A Wespi, “Towards a Taxonomy of Intrusion-Detection Systems”, International Journal for Computer and Telecommunications Networking, vol. 31, no. 9, pp. 805–822, 1999.
[3] Liao, Hung-Jen, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. "Intrusion detection system: A comprehensive review." Journal of Network and Computer Applications 36, no. 1, 2013.
[4] R. Gopalakrishna and E.H. Spafford, “A Framework for Distributed Intrusion Detection using Interest Driven Cooperating Agents”, In Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection, Davis, CA, USA,2001.
[5] Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer. “Taxonomy and survey of collaborative intrusion detection.” ACM Comput. surveys 2015.
[6] Ali Ahmadian Ramaki, Abbas Rasoolzadegan, and Abbas Ghaemi Bafghi. 2018, “A Systematic Mapping Study on Intrusion Alert Analysis in Intrusion Detection Systems”, ACM Comput. Surv. 51, 3, Article 55, June 2018.
[7] Zuech, R., Khoshgoftaar, T. M., & Wald, R, “Intrusion detection and big heterogeneous data: a survey”, Journal of Big Data, 2(3), 1-41, 2015.
[8] Chengpo, H. Houkuan, and T. Shengfeng, “A survey of intrusion-detection alert aggregation and correlation techniques”, Journal of Computer Research and Development, 2006.
[9] Sadoddin, and A. Ghorbani, “Alert correlation survey: Framework and techniques”, In Proceedings of the 2006 International Conference on Privacy, Security, and Trust: Bridge the Gap Between PST Technologies and Business Services ACM, New York, NY 2006.
[10] Yusof, S. R. Selamat, and S. Sahib, “Intrusion alert correlation technique analysis for heterogeneous log”, Int. J. Comput. Sci. Network security, 2008.
[11] H. Debar and A. Wespi, “Aggregation and Correlation of Intrusion-Detection Alerts”, In Proceedings of the 4th International Symposium, Recent Advances in Intrusion Detection (RAID) 2001, Springer-Verlag Lecture Notes in Computer Science, October 2001.
[12] Tu Hoang Nguyen, Jiawei Luo and Humphrey Waita Njogu, “An efficient approach to reduce alerts generated by multiple IDS products”, Int. J. Network Mgmt 2014; 24: 153–180, March.
[13] Yu Beng, S. Ramadass, S. Manickam, and T. Soo Fun, “A survey of intrusion alert correlation and its design considerations”, IETE Tech, 2014.
[14] Salah, G. Maciá-Fernández, and J. E. DíAz-Verdejo, “A model-based survey of alert correlation techniques”, Comput. Netw., vol 57 issue 5, 2013.
[15] Spathoulas and S. K. Katsikas, “Enhancing IDS performance through comprehensive alert post-processing”, Comput. Secur. 37, 176–196, 2013.
[16] Spathoulas and S. Katsikas, “Methods for post-processing of alerts in intrusion detection: A survey”, Int. J. Info. Secur. Sci, 2013.
[17] Zhang, J. Li, X. Chen, and L. Fan, “Building network attack graph for alert causal correlation”, Comput. Secur. 27, 5, 188–196, 2008.
[18] Haas, S., & Fischer, M., “GAC: Graph-Based Alert Correlation for the Detection of Distributed Multi-Step Attacks”, Proceedings of the 33rd Annual ACM Symposium on Applied Computing - SAC, 2018.
[19] Zhang, D., Qian, K., Zhang, P., Mao, S., & Wu, H, “Alert correlation analysis based on attack path graph”, 2017 IEEE Conference on Energy Internet and Energy System Integration, 2017.
[20] Al-Mamory, S. O., and Zhang, H, “IDS alerts correlation using grammar-based approach”, Journal in Computer Virology, 2009.
[21] Zali Z, Hashemi MR, Saidi H, “Real-time intrusion detection alert correlation and attack scenario extraction based on the prerequisite-consequence approach”, The ISC International Journal of Information Security, 2013.
[22] Ali Ahmadian Ramaki, Abbas Rasoolzadegan, “Causal knowledge analysis for detecting and modeling multi-step attacks”, Security Comm. Networks, 2016.
[23] D.Curry and H. Debar, “Intrusion Detection Message Exchange Format Data Model and extensible Markup Language (XML) Document Type Definition,” Internet draft, 2002.
[24] https://nvd.nist.gov/vuln/data-feeds
[25] https://cve.mitre.org/data/downloads/index.html
[26] MIT Lincoln Laboratory, DARPA 2000 intrusion detection scenario specific data sets, 2000, https://www.ll.mit.edu/r-d/datasets.
[27] A.M Riyad, M.S Irfan Ahmed, "An Ensemble Classification Approach for Intrusion Detection", International Journal of Computer Applications, vol. 80, pp. 37-42, October 2013.
[28] JADE Board, 2005, “JADE Security Add-On GUIDE”, Administrator's guide of the Security add-on, Version 28-February-2005, JADE 3.3, Copyright (C) 2004, TILAB.
[29] http://jade.tilab.com/download/jade/license/jade-download/
[30] A Turner, Tcpreplay: Pcap editing and replay tools for *nix 2010, http://tcpreplay.appneta.com/wiki/tcpreplay-man.html
[31] Peng Ning, Yun CUI, Douglas Reeves and Dingbang XU, “Techniques and Tools for analyzing intrusion alerts”, ACM Transactions on Information and Security, 2004.
[32] Peng Ning, Yun Cui, and Douglas S. Reeves, “Constructing Attack Scenarios through Correlation of Intrusion Alerts”, In Proc. 9th ACM Conference on Computer & Communications Security, 2002
[33] Safaa O. Al-Mamory, Hongli ZHANG, “Building Scenario Graph Using Clustering”, in proceeding of international conference on convergence information technology (ICCIT 2007), Korea, 2007.
[34] Mehdi Bateni, Ahmad Baraani, “An Architecture for Alert Correlation Inspired By a Comprehensive Model of Human Immune System”, I.J. Computer Network and Information Security, 2014.
[35] D. Yu, and D. Frinche, “Improving the quality of alerts and predicting intruder’s next goal with Hidden Colored Petri-Net,” Computer Networks, vol. 51, 2007.
[36] Faeiz Alserhani and Monis Akhlaq et al, “MARS: Multi Stage Attack Recognition System”, In Proc. of the International Conf. on Advanced Information Networking and Applications (AINA), Perth, 2010.
[37] B. Belaton, N.A. Bakar, A. Samsudin, “False Positives Reducetion Via Intrusion Alert Quality Framework”, 13th IEEE International Conference on Communication, 2005
[38] D. Yu and D. A. Frincke, “A novel framework for alert correlation and understanding”, In ACNS, Springer-Verlag Berlin Heidelberg, 2004.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY