*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 8, Issue 5 (Jan. - Feb. 2013), PP 63-69*
*www.iosrjournals.org*

# Generation of Meta Alerts by Aggregating Intrusion Alerts

Vishwesh.N[1], Rakesh Kumar.D[2], Anil Kumar.M[3], Mamatha.G[4]
*[1]Department of CNIS, School of IT JNTUH, India*
*[2](Department of SE, School of IT JNTUH, India*
*Assistant Professor*
*[3](Department of CSE, Nalla MallaReddy College, India*
*Assistant Professor*
*[4](Department of CSE, Aurora Engineering College, India*

***Abstract:*** *Online intrusion detection systems play an important role in protecting IT systems. Tools like Snort, firewall also detect intrusions. Such intrusion detection systems provide feedback in the form of alerts. However, the number of alerts is more in number and often security personnel are confused with such voluminous messages. This makes them difficult to take decision immediately. They take time to analyze the alerts and come to a conclusion for directions for taking actions. The security risk estimation and resolving the security problem depends on quick understanding of alerts. The bulk of alerts given by low level intrusion detection systems make it time consuming to arrive at decisions. To overcome this problem the alerts provided by low level detection systems can be programmatically aggregated and summarized alerts can be given to security personnel so as to enable them to draw conclusions quickly and take required actions. We propose a new technique for the purpose of online alert aggregation based on dynamic, probabilistic model. The solution is based on maximum likelihood approach which is a data stream version. The empirical results revealed that the proposed solution is effective and useful.*

***Index Terms*** *– Online intrusion detection, data streaming, probabilistic model, alert aggregation.*

## I. Introduction

Information security is important in IT systems. As emergence of innovative technologies in the arena of computing and ITC and the involvement of networks like Internet, security threats are increasing in a rapid pace. There are many techniques to prevent such attacks. They include authentication, authorization, cryptographic techniques like encryption, decryption; usage of virtual private networks and Intrusion Detection Systems (IDSs). Most of the IDS are capable of detecting attacks made by adversaries and defend the security of IT systems. The detection system is independent systems or also distributed collaborated systems. It may work in different kinds of networks including Wireless Sensor Networks (WSNs). They are of two types again. They are network-based intrusion detection systems and host – based intrusion detection systems. They generally use techniques pertaining to misuse and anomaly detection while detecting intrusions [1]. The intrusion detection systems are indispensable in the view of ensuring security to IT systems. The intruders are people with malicious intensions. Their aim is to break security of IT systems for monetary and other gains. The effective IDS which run in a network can prevent such threats. IDS can detect various kinds of attacks such as buffer overflow, SQL injection, DoS (Denial of Service) and so on. There are tools readily available to detect intrusions. The tools include Snort, Firewalls etc. These tools continuously monitor the systems for ensuring fool proof security. They work on the network flows of TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) and detection actions which are suspicious. They can verify attack instances of various kinds known to them. Each IDS can have different capabilities and the collaboration of IDSs in a distributed environment is quite possible for improving efficiency. Especially in WSN, it is essential and that way energy consumption of the network can be reduced thereby improving the life span of the network. IDS generally detect attack types and takes appropriate actions. In the process of detection the IDSs provide many alerts including false alerts. The alerts might have different features such as false positives and true positives. They log the findings so as to enable security personnel to take steps required to ensure that the IT system remain secure and the sensitive conversations between parties are protected from insider and outsider attacks.

When flood of alerts are created by IDSs in order to prompt security administrators the happening in the network, it is not easy to interpret each and every alert and come to a conclusion about the risk, severity of risk and the protection measures. Moreover security personnel may take wrong decisions due to false positives in the alerts and their inability to correctly interpret the bulk of alerts raised by the system. This is the motivation behind this paper. This paper aims at aggregating the flood of security alerts and provides concise feedback to security personnel so as to enable them to take actions quickly. The solution to this problem is to have an IDS that is perfectly situation-aware [2] and considers filtering of alerts and also aggregating alerts in such a way that the final alert (s) given to security personnel is concise, simple and straightforward in taking steps to mitigate risk or avoid it altogether. This can be achieved by clustering related attack instances. Without losing important alerts, aggregation of the alerts is to be done carefully. Missing meta-alerts is avoided and having to some degree of false or redundant meta-alerts is accepted. This kind of problem in IT systems has been around for many years. The solutions that came up so far are focusing on sorting alert messages based on destination, source, attack type etc. The IP addresses used in the output might be false because of spoofing attacks. Our approach has some distinct properties.

- It uses probabilistic methods and is a generative modeling approach [3]. It also assumes attack instances as random processes that produce security alerts. These are modeled using approximative maximum likelihood. It also detects the attack details like its start time and ending time.
- The proposed system uses a data stream approach [4]. It does mean that the alerts that have been observed are processed only for few times. This makes it suitable for online and also under strict constraints it can be used.

## II. Related Work

IDS are widely used in IT systems. They are reviewed by many researchers. Most of them are very effective and work with highest accuracy. In spite of this, the current IDSs have many problems. Lot of effort has been put in the past to overcome these problems. Many researchers analyzed existing IDEs and stated various problems of IDS. One such problem they identified is that IDS produces large number of security alerts that can make the job of security administrator difficult to take decisions quickly due to the confusing and conflicting alerts out of the flood of them. The researchers also provided directions for future work [5]. All IDS are having the provision of producing security alerts as and when required. Many approaches came into existence to solve those problems. However, [6] came with a comprehensive solution for alert correlation. One of the steps followed by [6] in correlation is to reconstruct attack thread. It is also known as attack instance recognition. It has not used any clustering algorithms but simple sorting is used. The results of the sorting are presented in a temporal window. It has duplicates of alerts as well. This duplicates problem has been prevented in [7] which is mostly similar to [6]. Thus it provides more concise way of alert presentation. This kind of approach is also used in [8] where clustering is used for the same purpose. Alert clustering approach is used by [9] based on the similarity of attack occurrences. It considers certain time and any two instances of attack are considered similar when both of them occur in a specific time window besides the exact similarity of their destination and source. As they use low level IDS, these detections systems may not work effectively in real time applications as they use imperfect classifiers. In [10] also alert correlation approach is used with the help of an operator known as weighted attribute wise similarity which determines whether to combine two given alerts or not. This approach and similar approaches provided in [11] and [12] are having a drawback which is the necessity of providing many parameters to the system. Same disadvantage is found with [13] as well where no guidance is provided in order to obtain good values. Attribute – wise similarity measures along with parameters given by user is used in [14]. As it also involves in sorting alerts in ascending or descending order based on the source and destination, it degenerates similarity measure. Various approaches are proposed in [15] for combining alerts. The first approach groups related alerts based on IP addresses. The second approach and third approach use some data mining techniques known as supervised learning techniques. It also used decision trees, radial basis function networks, multilayer perceptions and least square error approach in order to determine whether to combine a new alert with existing ones or not. To achieve this labeled training data is used as part of supervised learning that makes it difficult when attack instances are different.

In the field of intrusion scenario detection as presented in [16], [17] and [18] many similar tetchiness are used to making alert correlation. Out them very important procedure for scenario detection is in [19]. Base on an algorithm by name CURE, offline clustering solution is proposed in [20]. The solution makes use of numeric attributes only. Setting of clusters also manual process here. Though it has drawbacks it has significant advantage as it is supporting manual input from an expert. The problem with this approach is that the security expert who gives knowledge of domain expertise must be having knowledge about current attack instances. Another clustering solution is proposed in [11]. This proposal is closely similar to our approach. Its clustering method is known as “link based clustering”. It focuses on the reasons or meta data about the alerts generated by IDS. Only root causes are considered here. There is a problem of ignoring alerts that form into smaller clusters. The main difference between the [11] and our approach is that the [11] supports only offline intrusion detection. It depends on the historical traces present in the log files. However, our approach supports both offline and also online intrusion detection mechanism that makes it unique from existing IDSs. The alert clustering approach in [12] is also having good feature that reduces the number of false positives. This is also based on [11] in case of alert clustering. The approach presented in [22] is different completely. It makes use of reconstruction error of AA-NN (Auto Associator Neural Network) to differentiate alerts. Its approach is that it considers all alerts are same if they have same reconstruction error and put them into the same cluster. And this works in online and offline scenarios. The training requirements for AA-NN are training phase and also an offline training phase.

## III. Online Alert Aggregation Technique

Based on the probabilistic model of the current situation, a novel online alert aggregation technique is presented here. Many algorithms are proposed in order to achieve the goal of the system. The aim of the system is to effectively aggregate online alerts generated by IDSs. The architecture of the proposed system is as shown in fig. 1.

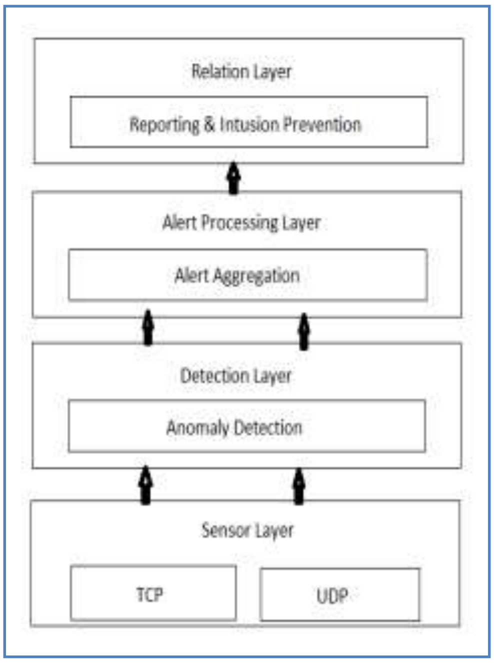


Fig. 1: Architecture of proposed system

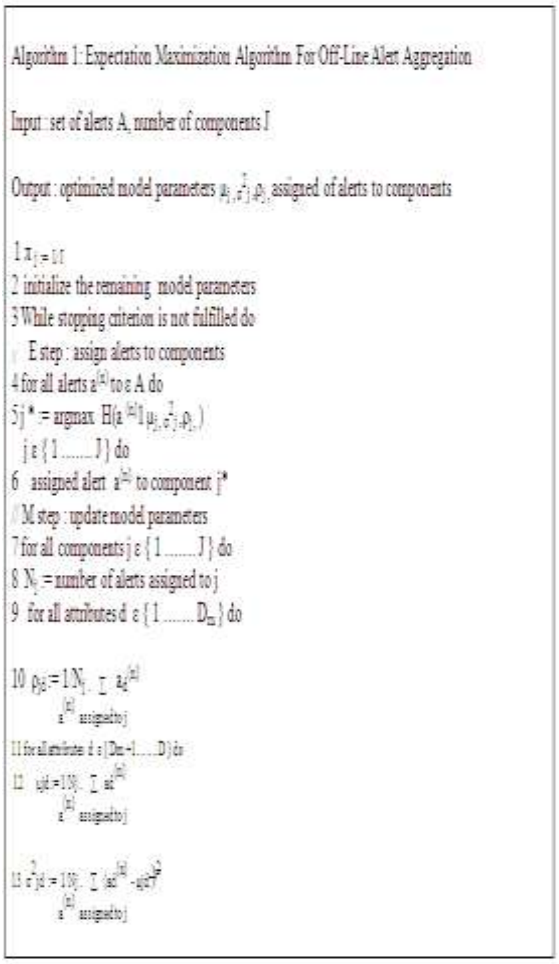
As can be seen in fig. 1, there are many layers namely sensor layer, detection layer, alert processing layer and reaction layer. The sensor layer is responsible to generate TCP or UDP traffic that is given to the next layer for detection. The detection layer is responsible to detect intrusion based on misuse and anomaly detection. The generated flood of alerts is given to the next layer known as alert processing layer. The alert processing layer makes use of the proposed probabilistic technique in order to aggregate alerts. The aggregated alerts are given to reaction layer which provides meaningful reports to security personnel besides taking prevention measures.

## III.i. Offline Alert Aggregation

Envisage that various attacks are made on the TCP or UDP traffic and the flood of generated alerts labeled with false positives, false negatives etc. This logged information can be analyzed and the alert aggregation can be done offline. However, the following are the problematic situations with respect to alert aggregation.

- Non recognition of false alerts and wrong assignment of them to clusters.
- Genuine alerts are assigned to clusters wrongly.
- The splitting of clusters is done wrongly.
- Many clusters are clubbed into one in a wrong way.

The offline alert aggregation algorithm known as expectation maximization is presented in fig. 2.



```
Algorithm 1: Expectation Maximization Algorithm For Off-Line Alert Aggregation

Input : set of alerts A, number of components J

Output : optimized model parameters $\mu_j, \sigma^2_j, \rho_j$, assigned of alerts to components

1 $\pi_j = 1/J$
2 initialize the remaining model parameters
3 While stopping criterion is not fulfilled do
   // E step : assign alerts to components
4 for all alerts $a^{(n)}$ to ε A do
5 $j^* := \text{argmax } H(a^{(n)} | \mu_j, \sigma^2_j, \rho_j)$
    j ε { 1 ....... J } do
6   assigned alert $a^{(n)}$ to component $j^*$
   // M step : update model parameters
7 for all components j ε { 1 ....... J } do
8 $N_j$ := number of alerts assigned to j
9  for all attributes d ε { 1 ....... $D_m$ } do

10 $\rho_{jd} := 1/N_j \cdot \sum_{a^{(n)} \text{ assigned to } j} a_d^{(n)}$
11 for all attributes d ε {Dm+1.....D} do
12   $\mu_{jd} = 1/N_j \cdot \sum_{a^{(n)} \text{ assigned to } j} a_d^{(n)}$

13 $\sigma^2_{jd} = 1/N_j \cdot \sum_{a^{(n)} \text{ assigned to } j} (a_d^{(n)} - \mu_{jd})^2$
```

Fig. 2: Expectation Maximization Algorithm for Offline Aggregation

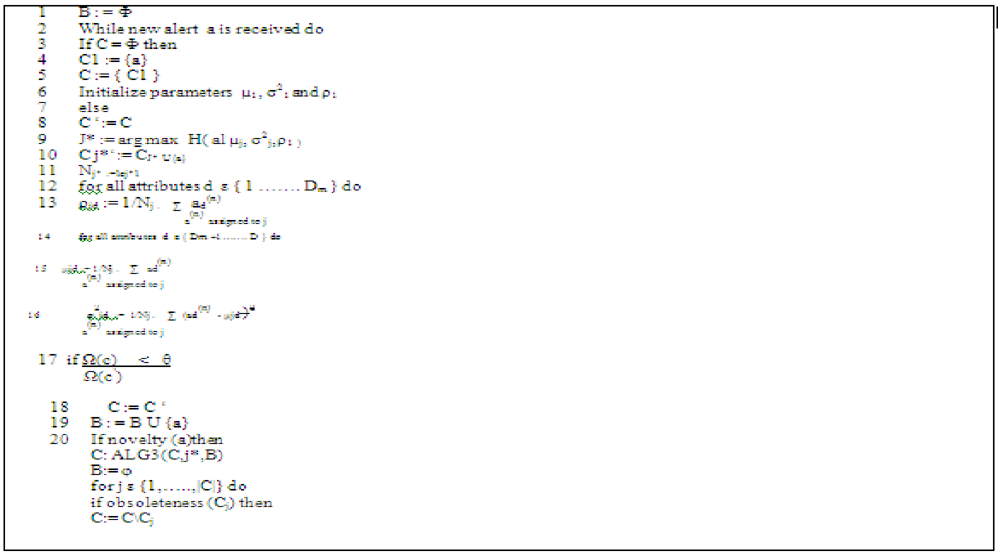
As can be seen in fig. 2, the algorithm performs steps like initialization of model parameters, hard assignment of alerts to components, stopping criterion, and fixed mixing coefficients. Getting good initial values is the aim of initialization of model parameters. The second step adds alerts to components gradually. The third step ensures that there is a condition that helps in stopping the process. Wide range of possible cluster sizes is a problem in expectation maximization. Coefficients help EM algorithms to optimize the process of offline alert aggregation.

## III.ii. Data Stream Alert Aggregation

Offline alert aggregation can be extended to make it online alert aggregation. This process is described here. To achieve these IDS should have the following.

- Component Adaption: alerts associated with attack instances are to be identified and assigned to respective clusters besides using component parameters.
- Component Creation: new attack instances are to be created and component parameters are to be set accordingly.
- Component Detection: the completion of identification of attack instances is to be detected and such components are to be deleted from the model.

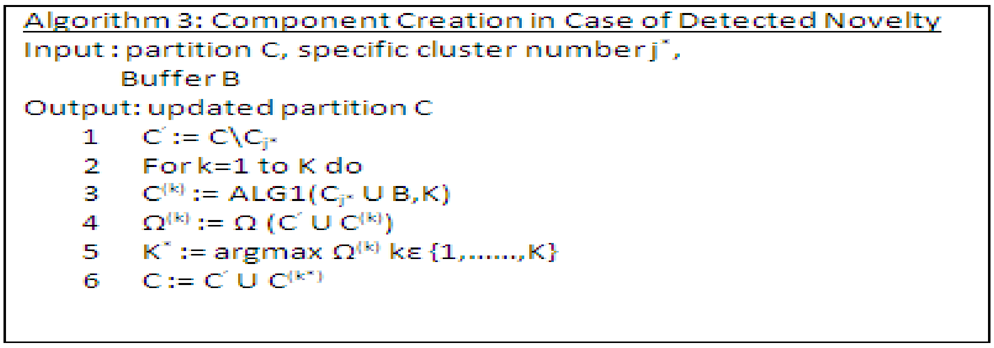
The online alert aggregation algorithm is presented in fig. 3.



```
1    B := Φ
2    While new alert a is received do
3    If C = Φ then
4    C1 := {a}
5    C := { C1 }
6    Initialize parameters $\mu_1$, $\sigma^2_1$ and $\rho_1$
7    else
8    C' := C
9    J* := arg max H( a| $\mu_j$, $\sigma^2_j$, $\rho_1$ )
10   Cj*' := $C_{j^*}$ ∪ {a}
11   $N_{j^*}$ := $N_{j^*}$+1
12   for all attributes d ε { 1 ....... $D_m$ } do
13   $\rho_{jd}$ := 1/$N_j$ . Σ $a_d^{(n)}$
                       $a^{(n)}$ assigned to j
14   for all attributes d ε { Dm+1 ....... D } do
15   $\mu_{jd}$ := 1/$N_j$ . Σ $a_d^{(n)}$
          $a^{(n)}$ assigned to j
16   $\sigma^2_{jd}$ := 1/$N_j$ . Σ ($a_d^{(n)}$ - $\mu_{jd}$)$^2$
          $a^{(n)}$ assigned to j
17 if $\frac{\Omega(c) \quad < \quad \theta}{\Omega(c')}$
18     C := C'
19   B := B U {a}
20   If novelty (a)then
     C: ALG3(C,j*,B)
     B:= φ
     for j ε {1,......,|C|} do
     if obsoleteness ($C_j$) then
     C:= C\$C_j$
```

Fig. 3: Online Alert Aggregation Algorithm

In case of detected novelty, component creation is done using the algorithm shown in fig. 4.This algorithm takes partition, cluster number, and buffer as input and generates updated patterns as output.



```
Algorithm 3: Component Creation in Case of Detected Novelty
Input : partition C, specific cluster number j*,
        Buffer B
Output: updated partition C
    1    C' := C\$C_{j^*}$
    2    For k=1 to K do
    3    $C^{(k)}$ := ALG1($C_{j^*}$ U B,K)
    4    $\Omega^{(k)}$ := Ω (C' U $C^{(k)}$)
    5    K* := argmax $\Omega^{(k)}$ kε {1,......,K}
    6    C := C' U $C^{(k^*)}$
```

Fig. 4: Algorithm for component creation in case of detected novelty

## IV. IMPLEMENTATION and RESULTS

We have implemented a custom simulator for online intrusion alert aggregation using Java programming language. The software used to implement this is Eclipse, JDK 1.6, and JME. The system was run in Windows XP OS. The implementation has GUI developed using SWING API of Java programming language. For attack simulation, IDS and alert aggregation simulation user interfaces were built. The UI screen for attack simulation is as shown in fig. 5.

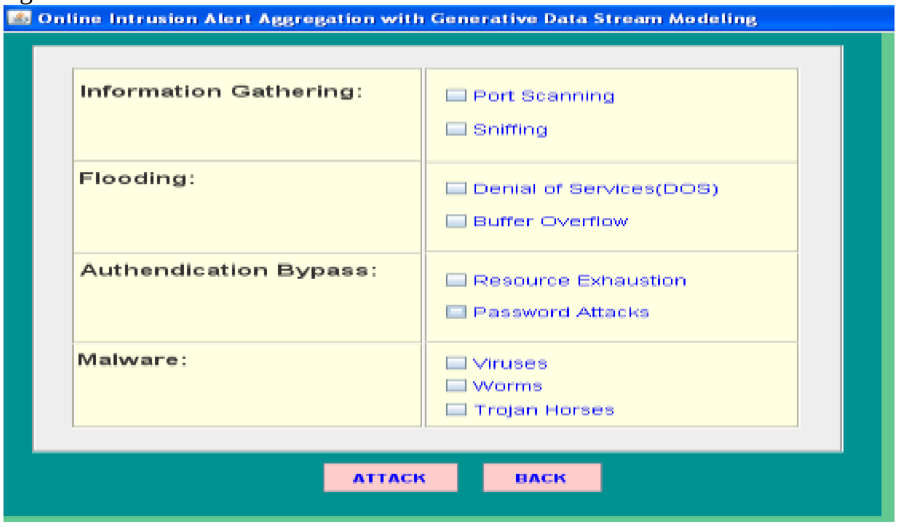


Fig. 5: Various Security Attacks

As can be seen in fig. 5, provision is given for simulating various kinds of attacks grouped into malware, authentication bypass, flooding and information gathering. The malware attacks include viruses, worms, and Trojan horses. Authentication bypass attacks include resource exhaustion and password attacks. The information gathering attacks include port scanning and sniffing. The alerts aggregation is shown in another GUI form as shown in fig. 6.

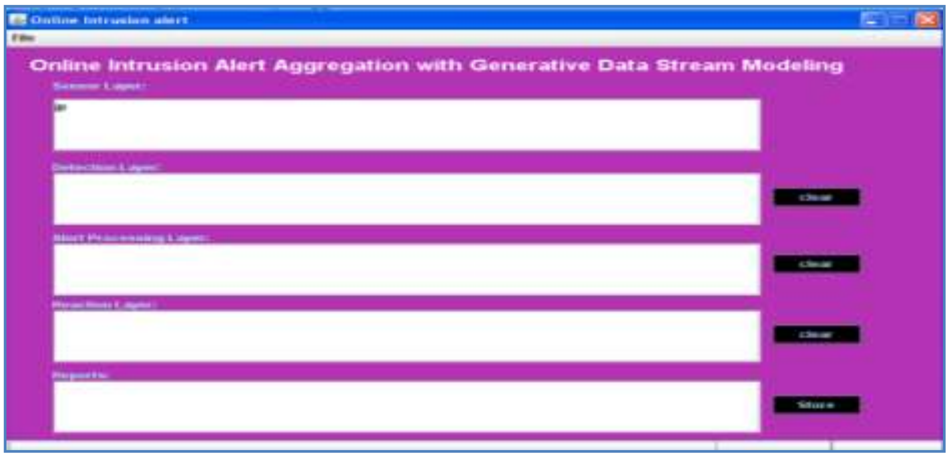


Fig. 6: Alert Aggregation Simulation

As can be seen in fig. 6, for each and every layer presented in architecture diagram (fig. 1), there is a place for aggregated alert messages. The layers include sensor layer, detection layer, alert processing layer, reaction layer and at the bottom a text area is found for showing reports. When attack is made the attack related message is shown as given in fig. 7.

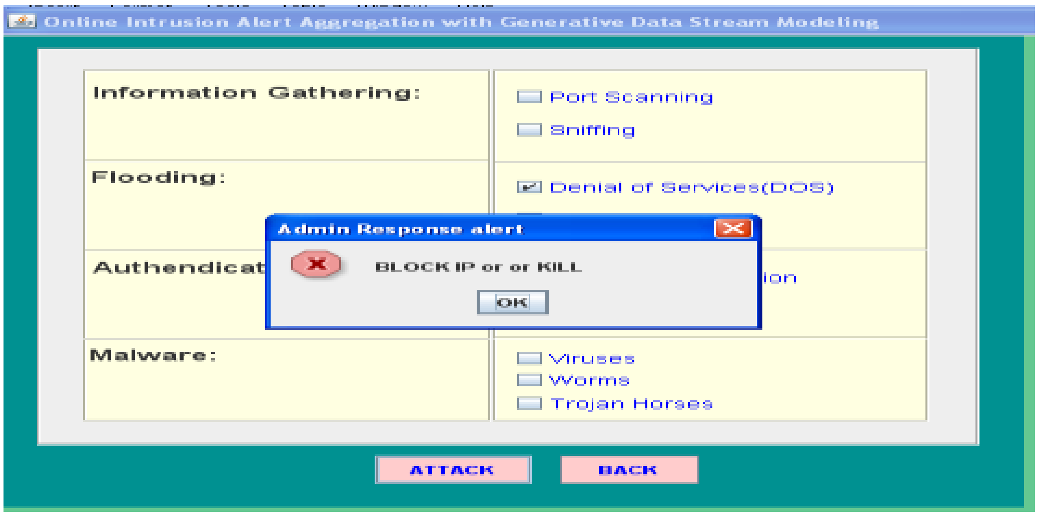


Fig. 7: Shown response of the system when an attack is made

## V. Conclusion

The proposed approach for intrusion detection and alert aggregation has been implemented using a custom simulator that shows the process of intrusion detection and also aggregation of alerts to obtain meaningful and summarized alerts that help in taking decisions quickly. The proposed prototype application supports simulation of various kinds of attacks like port scanning, sniffing, and buffer overflow, denial of service, resource exhaustion, password attacks, viruses, worms, and Trojan horses. The experimental results revealed that the simulation study of the online intrusion detection alert aggregation is effect and useful when implemented in real time applications. It can be further improved by considering some more security attacks.

## References

[1] S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," Technical Report 99-15, Dept. of Computer Eng., Chalmers Univ. Of Technology, 2000.

[2] M.R. Endsley, "Theoretical Underpinnings of Situation Aware- ness: A Critical Review," Situation Awareness Analysis and Measurement, M.R. Endsley and D.J. Garland, eds., chapter 1, pp. 3-32, Lawrence Erlbaum Assoc., 2000.

[3] C.M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[4] M.R. Henzinger, P. Raghavan, and S. Rajagopalan, Computing on Data Streams. Am. Math. Soc., 1999.
[5] A. Allen, "Intrusion Detection Systems: Perspective," Technical Report DPRO-95367, Gartner, Inc., 2003.
[6] F. Valeur, G. Vigna, C. Krugel, and R.A. Kemmerer, "A Comprehensive Approach to Intrusion Detection Alert Correla- tion," IEEE Trans. Dependable and Secure Computing, vol. 1, no. 3, pp. 146-169, July-Sept. 2004.
[7] H. Debar and A. Wespi, "Aggregation and Correlation of Intrusion-Detection Alerts," Recent Advances in Intrusion Detection, W. Lee, L. Me, and A. Wespi, eds., pp. 85-103, Springer, 2001.
[8] D. Li, Z. Li, and J. Ma, "Processing Intrusion Detection Alerts in Large-Scale Network," Proc. Int'l Symp. Electronic Commerce and Security, pp. 545-548, 2008.
[9] F. Cuppens, "Managing Alerts in a Multi-Intrusion Detection Environment," Proc. 17th Ann. Computer Security Applications Conf. (ACSAC '01), pp. 22-31, 2001.
[10] A. Valdes and K. Skinner, "Probabilistic Alert Correlation," Recent Advances in Intrusion Detection, W. Lee, L. Me, and A. Wespi, eds. pp. 54-68, Springer, 2001.
[11] K. Julisch, "Using Root Cause Analysis to Handle Intrusion ¨ Detection Alarms," PhD dissertation, Universitat Dortmund, 2003. 294 IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING,
[12] T. Pietraszek, "Alert Classification to Reduce False Positives in¨ Intrusion Detection," PhD dissertation, Universitat Freiburg, 2006.
[13] F. Autrel and F. Cuppens, "Using an Intrusion Detection Alert Similarity Operator to Aggregate and Fuse Alerts," Proc. Fourth Conf. Security and Network Architectures, pp. 312-322, 2005.
[14] G. Giacinto, R. Perdisci, and F. Roli, "Alarm Clustering for Intrusion Detection Systems in Computer Networks," Machine Learning and Data Mining in Pattern Recognition, P. Perner and A. Imiya, eds. pp. 184-193, Springer, 2005.
[15] O. Dain and R. Cunningham, "Fusing a Heterogeneous Alert Stream into Scenarios," Proc. 2001 ACM Workshop Data Mining for Security Applications, pp. 1-13, 2001.
[16] P. Ning, Y. Cui, D.S. Reeves, and D. Xu, "Techniques and Tools for Analyzing Intrusion Alerts," ACM Trans. Information Systems Security, vol. 7, no. 2, pp. 274-318, 2004.
[17] F. Cuppens and R. Ortalo, "LAMBDA: A Language to Model a Database for Detection of Attacks," Recent Advances in Intrusion Detection, H. Debar, L. Me, and S.F. Wu, eds. pp. 197-216, Springer, 2000.
[18] S.T. Eckmann, G. Vigna, and R.A. Kemmerer, "STATL: An Attack Language for State-Based Intrusion Detection," J. Computer Security, vol. 10, nos. 1/2, pp. 71-103, 2002.
[19] A. Hofmann, "Alarmaggregation und Interessantheitsbewertung in einem dezentralisierten Angriffserkennungsystem," PhD dis-¨ sertation, Universitat Passau, under review.
[20] M.S. Shin, H. Moon, K.H. Ryu, K. Kim, and J. Kim, "Applying Data Mining Techniques to Analyze Alert Data," Web Technologies and Applications, X. Zhou, Y. Zhang, and M.E. Orlowska, eds. pp. 193-200, Springer, 2003.
[21] R. Smith, N. Japkowicz, M. Dondo, and P. Mason, "Using Unsupervised Learning for Network Alert Correlation," Advances in Artificial Intelligence, R. Goebel, J. Siekmann, and W. Wahlster, eds. pp. 308-319, Springer, 2008.

## AUTHORS

**Vishwesh Nagamalla**, Recently Completed M.Tech (Computer Networks & Information Security) from School of IT JNTUH, B.Tech(CSE) from Kamala Institute of Technology &Science, His research interests are Information Security, Wireless &Mobile Computing, Computer Networks & Cloud Computing.

**Rakesh Kumar Donthi**, Recently Completed M.Tech (Software Engineering) from School of IT JNTUH, B.Tech(IT) from Aurora Scientific Technological Research Academy, His research interests are Computer Networks, Operating System, DBMS & Cloud Computing

**Anil Kumar Mula**, working as an Assistant Professor in Nalla Malla Reddy Engineering College, Hyderabad. Completed M.Tech (Software Engineering) from School of IT JNTUH, B.Tech (CSE) from Kamala Institute of Technology &Science. His research interests are Information Security & Software Engineering

**Gudipati Mamatha**, working as an Assistant Professor in Bhongir,Nalgonda. Completed M.Tech (CSE) & B.Tech (CSE) from JNTU Hyderabad. Her research interests are DBMS & Operating Systems