Volume 3, Issue 7, July 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Intrusion Detection System with Meta Alert Generation

Kathula Ajith Kumar*
M.Tech, Software Engg. KITS WARANGAL
India

M.S.B.Prudhviraj
Asst.Prof.(CSE) KITS WARANGAL
India

Abstract: Security plays an important role in IT systems. Intrusion detection systems can be used to ensure security in a network. The existing IDSs (Intrusion Detection Systems) such as Firewall, Snort provide huge number of alerts as they monitor the network flows. Since the number of alerts isplenty, the network administrator might be confused to know exact problem. This will delay indecision making in the presenceof any security threats. As it takes more time to understand the alerts when they are more number, the network administrator needs to spend some time to make effective decisions. In this paper, we proposed a framework which aggregates alerts and generates few Meta alerts. These Meta alerts can be understood by the network personnel quickly and take decisions immediately. A data stream version of maximum likelihood approach is used in the framework. The experimental results revealed that the framework is very useful and can be used in the real world networks.

Index Terms – IDS, online intrusion detection, probabilistic model, online intrusion detection, alert aggregation.

I. INTRODUCTION

Security is essential in IT systems. In all kinds of networks, it is important. As the networks are spreading and related technologies are growing, the threats of various kinds are also growing rapidly. The attacks over networks can be prevented using some techniques namely encryption, decryption, authentication, authorization and so on. VPN (Virtual Private Network) technology and IDS (IntrusionDetection System) can also be used to protect networks. Most of the existingIDS such as Snort are capable of detecting intrusions when hackers try to intrude into the IT networks. The detection systems might work independently or in a distributed environment. The IDS can be used in various kinds of networks such as MANET (Mobile Ad Hoc Networks), WSNs (Wireless Sensor Networks) and so on. The IDS is of two types. They are known as host based and network based. They are meant for anomaly detection, misuse detection and detection of intrusions [1]. Keeping the security requirement in mind, the IDSs are mandatory. The intruders are also known as hackers or adversaries. They does it for monetary gains or otherwise. The presence of IDS can help protect network from intrusions. It also prevents attacks like SQL Injection, buffer overflow, denial of service and so on. The tools like Snort continuously monitor networks that make use of protocols such as UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). Each IDS can have its own capabilities and some of the IDSs may work in collaboration with other IDS instances in a network. They can detect intrusions and take necessary steps. They generate alerts which are logged into some designated file. Thus they can protect the network from insider and outsider attacks. The problem with existing IDS is that it provides flood of alerts that are to be intercepted by network personnel and make decisions. The bulk of alerts may cause security personnel get confused and it may result in taking wrong decisions also. For this reason it is important to have the ability to aggregate the alerts and generate more meaningful Meta alerts that will help network administrators to make decisions immediately.

The aim of this paper is to aggregate the flood of alerts to generate Meta alerts so as to help security personnel take decisions quickly. This needs IDS which is area of situations [2] and filters the alerts to generate Meta alerts.

The aggregation is achieved by using alert instances and grouping them. At the same time it is important to avoid missing Meta alerts. Thus the problem of flooding of alerts can be avoided as the proposed IDS can give clear and concise alerts to the user. The approach used in this paper has the following properties.

- It makes use of probabilistic methods based on a model known as generative modeling [3]. It considers attack instances and aggregates them.
- The data streaming approach [4] is used by the proposed system which is suitable for online intrusion with alert aggregation.

II. RELATED WORK

An ID is a tool that can help protecting IT systems. Lot of research went on IDS. Many IDSs present in the real world now are effective and provide accurate detections. However, researchers found various problems with IDS. One suchproblem with IDS is its characteristic to produce flood of alerts making the job of network administrator difficult. Future work directions are also given by researcher in the light of problems with IDS [5]. Many approaches came into IDS with respect to the alert

correlation. In [6] comprehensivesolution is attempted for alert correlation. Attack instance recognition is a concept used here. No clustering algorithms are used to achieve this. The results are aggregated in a temporal window. There are alert duplication problems as explored in [7] where a solution is presented to aggregate alerts in order to provide concise message to end users. In [9] an approach known as alert clustering is used to group similar attack instances. In [10] an approach for alert correlation is used based on weighted attributes for finding similarity. This approach and other approaches proposed in [11] and [12] suffer from various drawbacks such as the need for parameters. Almost same problem is found in [13]. In [14] also user has to give some parameters for IDS towork effectively. In [15] various approaches are presented to prepare concise alerts. One of the approaches group alerts based on IP while the others follow some supervised learning methods. They used labeled training data to achieve this. Many similar techniques were presented in [16], [17] and [18] out of which [16] is important. Offline clustering solutions are also made as presented in [20] based on an algorithm named "CURE". The problem with this is that it considers only numeric attributes and it needs clustering to be done manually. However, its advantage is that it support domain expert input that will improve performance. Again the success depends on the expertise of the domain expert.

In [11] there is another clustering approach provided which is closely similar to the approach we followed in this paper. Its approach is known as link based clustering. It focuses on Meta alert generation. In this approach only root causes are taken as important messages. The difference between our approach and [11] is that our approach supports online and offline intrusion detection while [11] supports only offline intrusion detection. In [12] also has feature that reduces false positives. This approach uses alert clustering as used in [11]. In [22] a different approach is used. It is based on AA-NN (Auto Assocator Neural Network) for alert differentiation. It considers alerts same based on the reconstruction error similarity. This also works in both offline and online scenarios. Our approach in this paper is presented in the next section.

III. ONLINE ALERT AGGREGATION TECHNIQUE

We present a novel alert aggregate approach in this paper based on the probabilistic model. Towards this goal many algorithms are proposed. The aim of the new approach is to aggregate alerts effectively and producesMeta alerts so as to enable network administrators take decisions faster. The new approach followed in this paper is presented in fig. 1.

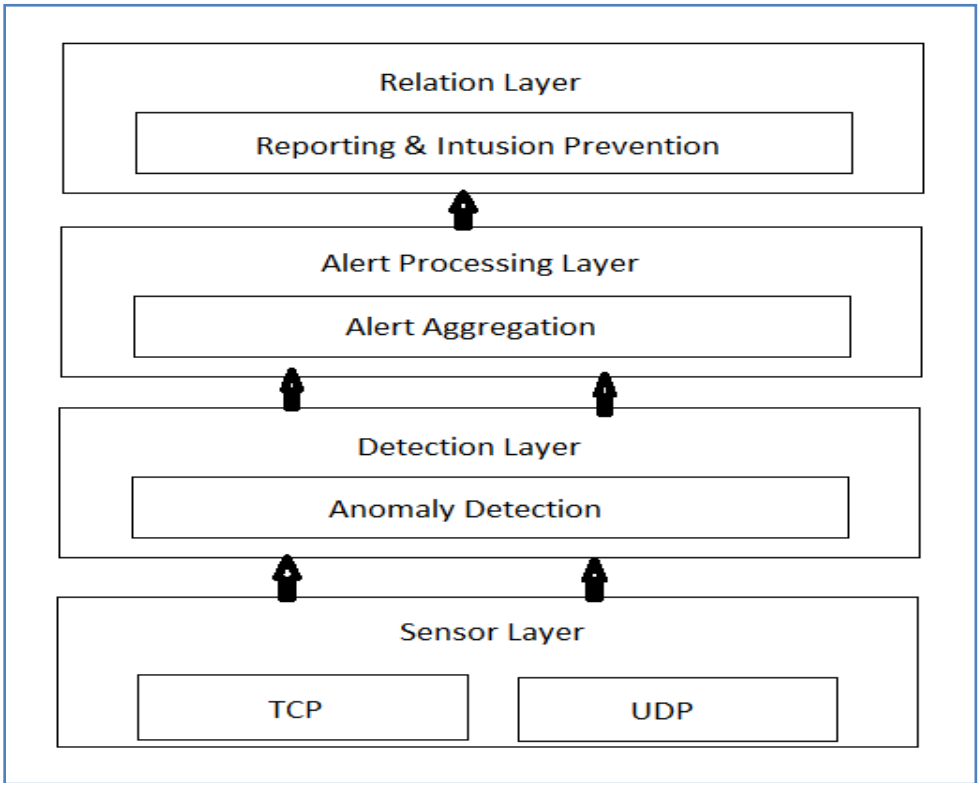

Fig. 1- Overview of Proposed Approach

As seen in fig. 1, it is evident that the proposed system is made up of many alters. The layers include reaction layer, processing layer, detection layer and sensor layer. The sensor layer produced UDP and TCP traffic. The detection layer detects intrusion based on the anomaly or misuse detection. This layer sends the generated alerts to processing layer which will aggregate alerts based on probability theory. Then the reaction layer takes the aggregate alerts and produce meaningful reports to network personnel.

IV. OFFLINE ALERT AGGREATION

Assuming that UDP and TCP traffic is attacked, the floods of alerts are labeled false negatives and false positives. Such information is logged and used for alert aggregation to be taken place offline. The following situations are challenging for alert aggregation.

- Inability to recognize false alerts and adding them to incorrect clusters.
- Adding genuine alerts to wrong clusters.
- Wrong splitting of alerts.
- Wrong aggregation of alerts.

The alert aggregation algorithm which works offline is presented in fig. 2. The algorithm is based on expectation maximization.

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

Input : set of alerts A, number of components J

Output : optimized model parameters $\mu_{i,}\sigma^2_{i,}\rho_{i,}$ assigned of alerts to components

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

$10\ \rho_{jd} := 1/N_i \cdot \sum_{a^{(a)} \text{ assigned to } j} a_d^{(a)}$

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

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

Fig. 2: Algorithm for Offline Alert Aggregation

As shown in fig. 2, there are some steps followed by algorithm. They include parameter initialization, alerts to components assignment, checking stopping criteria, working with coefficients. Initialization is aimed at obtaining correct initial values. Afterwards alerts are added to components. Then a condition is verified for stopping process. Cluster sizes possible are found based on expectation maximization. Coefficients are used to help in the optimization process

.

V. DATA STREAM ALERT AGGREGATION

Offline alert aggregation is improved further in order to make it work for online alert aggregation as well. It needs the following steps to be followed.

1. Component Adaption
2. Component Creation
3. Component Detection

The three steps are carried out as per the intended work they are supposed to do. The work of them is intuitive and self explanatory. The final step is meant for detecting the components thus helping in alert aggregation. The algorithm for the same is presented in fig. 3.

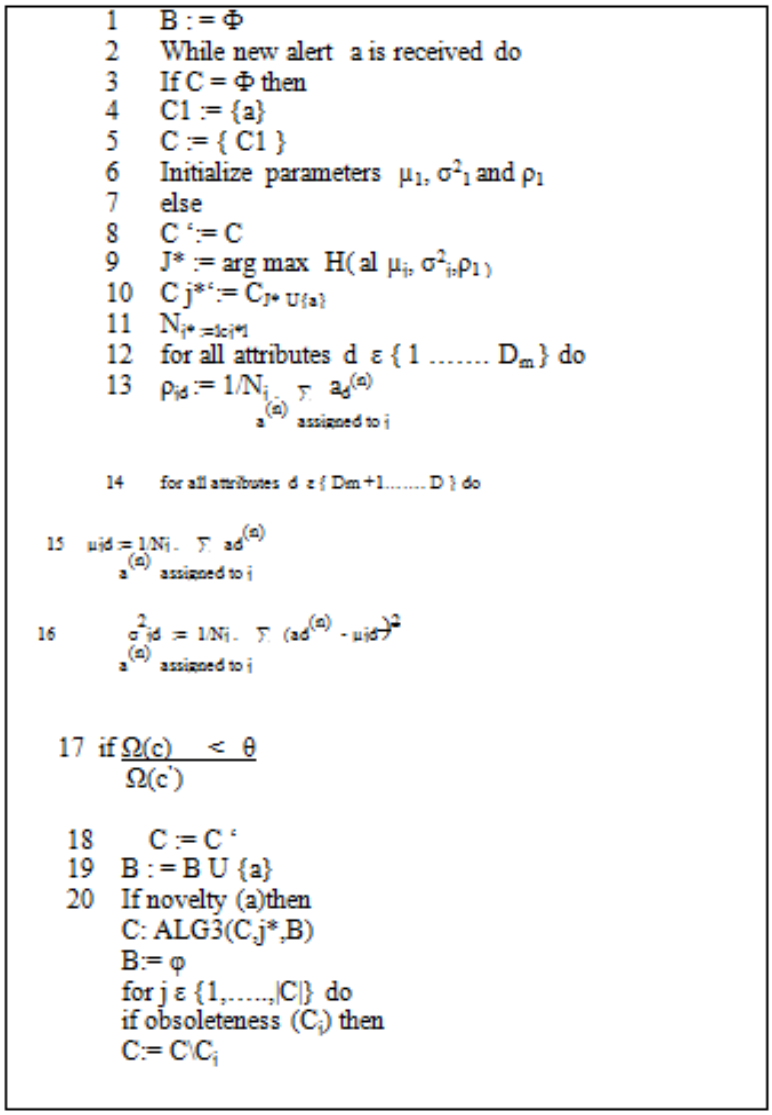

Fig. 3: Algorithm for online alert aggregation

As seen in fig. 4, component creation is done. The inputs to the algorithm include buffer, cluster number and partition and the output is updated patterns.

Algorithm 3: Component Creation in Case of Detected Novelty

Input : partition C, specific cluster number j^*,
Buffer B
Output: updated partition C

1 $C' := C\backslash C_{j*}$
2 For k=1 to K do
3 $C^{(k)} := ALG1(C_{j*} \cup B,K)$
4 $\Omega^{(k)} := \Omega (C' \cup C^{(k)})$
5 $K^* := \text{argmax } \Omega^{(k)}$ kε {1,……,K}
6 $C := C' \cup C^{(k*)}$

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

VI. IMPLEMENTATION AND RESULTS

The IDS has been developed as customer simulator in Java programming language. The software used includes NetBeans, JDK 1.6, and JME. Windows XP OS which runs in a PC with 4 GB of RAM and Core 2 duo processor. The implementation is done with GUI using AWT and SWING API of Java. UI is built for attack simulation and other user interfaces. Fig. 5 shows UI for attack simulation.

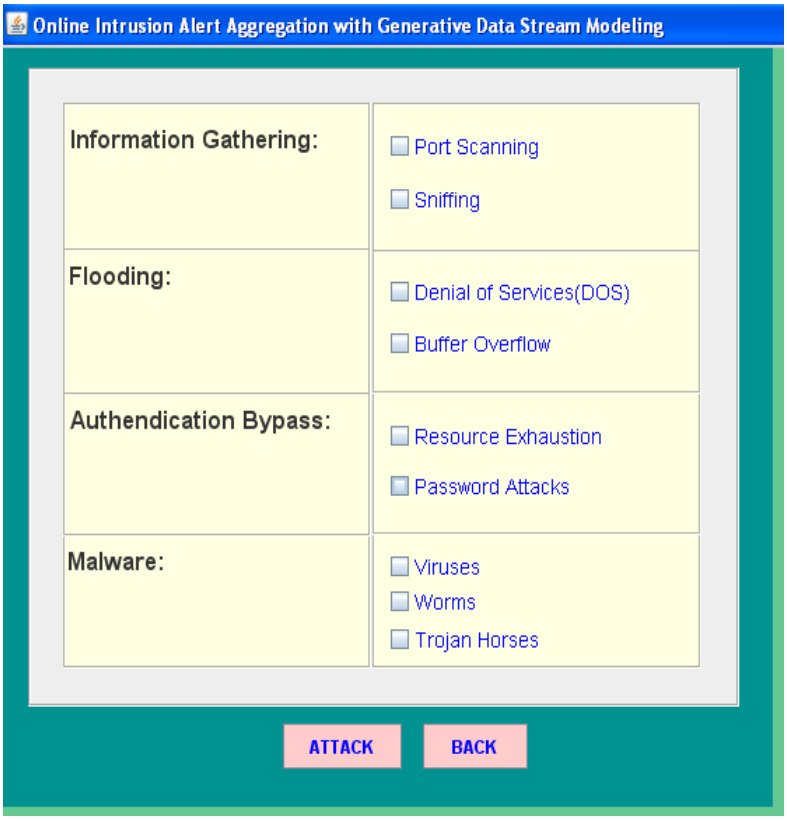

Fig. 5: UI for Security Attacks

As seen in fig. 5, UI is provided to simulate various kinds of attacks. The attacks include Trojan horses, worms, viruses, flooding, authentication bypass, and malware. The port scanning and sniffing come under information gathering attacks. Fig. 6 shows UI for alerts aggregation.

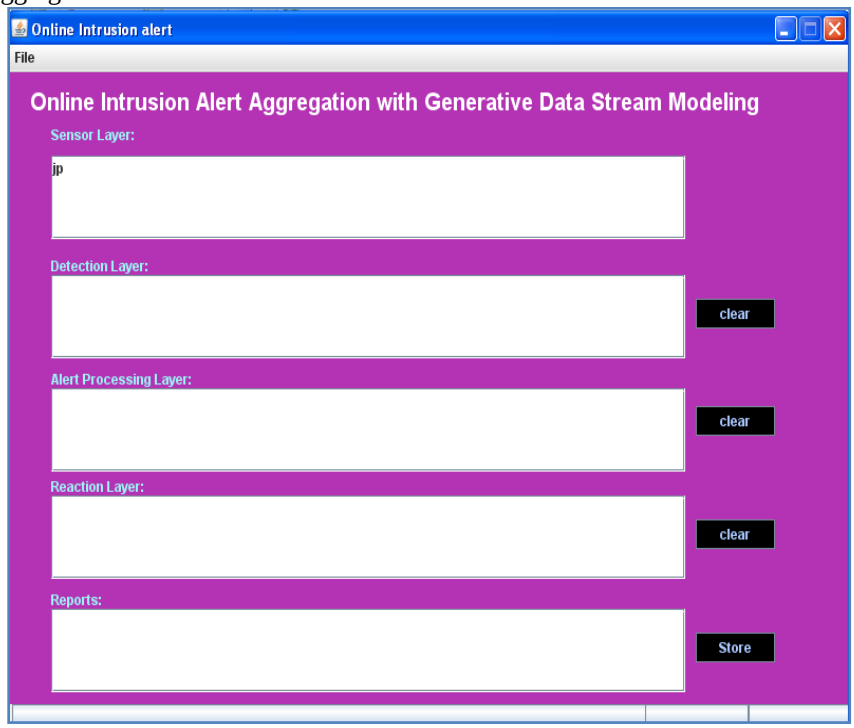

Fig. 6: UI for alert aggregation

As shown in fig. 6, as per the architecture in fig. 1, there is provision in the UI for alert aggregations. The layers include sensor layer to reaction layer. When attacks are made the alerts are shown as in fig. 7.

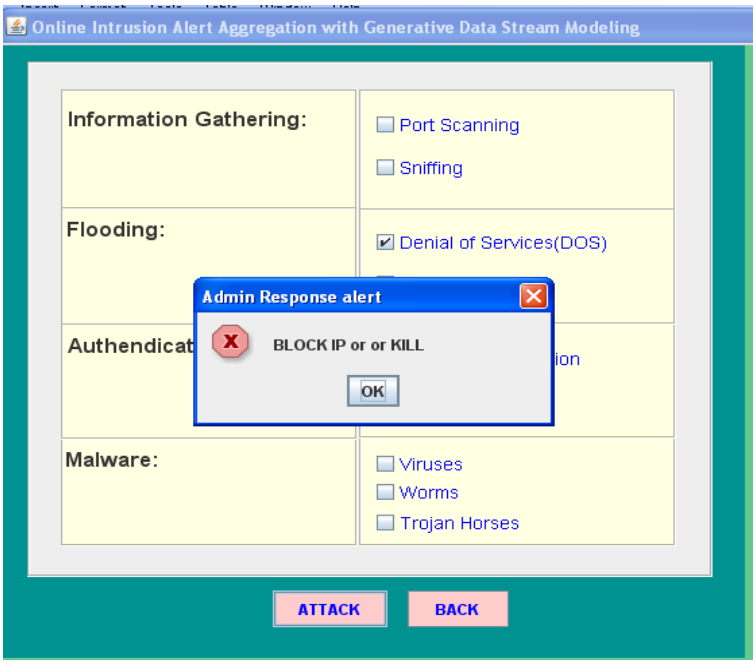

Fig. 7 – Shows generated Meta alert

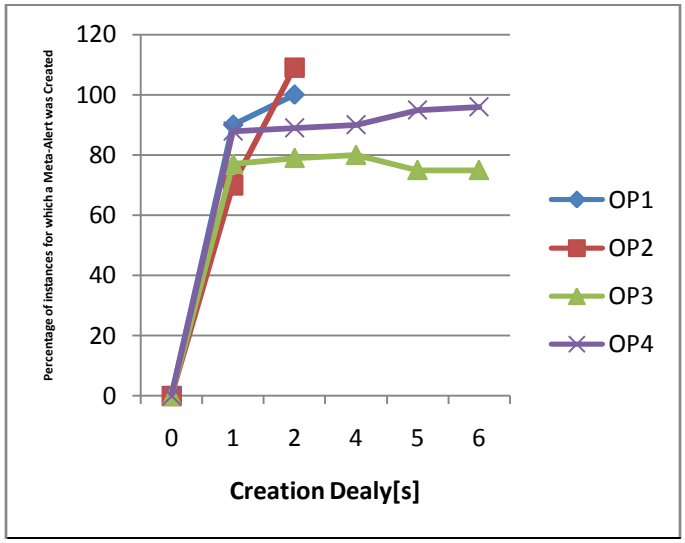

Fig. 8 –Meta alerts vs. creation delay

As can be seen in fig. 8, thecreation delay in seconds is represented in horizontal axis while the vertical axis represents percentage of instance for whichMeta alert was created.

VII. CONCLUSION

The proposed IDS works well for intrusion detection and also generatesMeta alerts by aggregating many similar alerts. The Meta alerts give accurate attack information that helps the network personnel to take decisions faster. We have built a prototype application that demonstrates the effectiveness of the proposed approach. The attacks it can simulate include Trojan horses, worms, viruses, password attacks, resource exhaustion, denial of service and buffer overflow besides sniffing and port scanning attacks. The empirical results revealed that the prototype is useful for both online and offline aggregation of alerts of IDS.

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 IntrusionDetection 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] J. Song, H. Ohba, H. Takakura, Y. Okabe, K. Ohira, and Y. Kwon, "A Comprehensive Approach to Detect Unknown Attacks via Intrusion Detection Alerts," Advances in Computer Science—ASIAN 2007, Computer and Network Security, I. Cervesato, ed., pp. 247-253, Springer, 2008.
[22] 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.
[23] A. Hofmann, D. Fisch, and B. Sick, "Identifying Attack Instances by Alert Clustering," Proc. IEEE Three-Rivers Workshop Soft Computing in Industrial Applications (SMCia '07), pp. 25-31, 2007.
[24] M. Roesch, "Snort—Lightweight Intrusion Detection for Net- works," Proc. 13th USENIX Conf. System Administration (LISA '99), pp. 229- 38, 1999.
[25] O. Buchtala, W. Grass, A. Hofmann, and B. Sick, "A Distributed Intrusion Detection Architecture with Organic Behavior," Proc. First CRIS Int'l Workshop Critical Information Infrastructures (CIIW '05), pp. 47-56, 2005.
[26] D. Fisch, A. Hofmann, V. Hornik, I. Dedinski, and B. Sick, "A Framework for Large-Scale Simulation of Collaborative Intrusion Detection," Proc. IEEE Conf. Soft Computing in Industrial Applica- tions (SMCia '08), pp. 125-130, 2008.
[27] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, second ed. Wiley Interscience, 2001.
[28] IANA, "Port Numbers," http://www.iana.org/assignments/ port- numbers, May 2009.
[29] Y. Rekhter, B. Moskowitz, D. Karrenberg, and G. de Groot, "RFC 1597—Address Allocation for Private Internets," http://www. faqs.org/rfcs/rfc1597.html, Mar. 1994.
[30] J. Postel, "RFC 790—Assigned numbers," http://www.faqs.org/ rfcs/rfc790.html, Sept. 1981.

[31] O. Buchtala, A. Hofmann, and B. Sick, "Fast and Efficient Training of RBF Networks," Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003, O. Kaynak, E. Alpaydin, E. Oja, and L. Xu, eds., pp. 43-51, Springer, 2003.
[32] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines, K.R. Kendall, D. McClung, D. Weber, S.E. Webster, D. Wyschogrod, R.K. Cunningham, and M.A. Zissman, "Evaluating Intrusion Detec- tion Systems: The 1998 DARPA Offline Intrusion Detection Evaluation," Proc. DARPA Information Survivability Conf. And Exposition (DISCEX), vol. 2, pp. 12-26, 2000.
[33] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On Clustering Validation Techniques," J. Intelligent Information Systems, vol. 17, nos. 2/3, pp. 107-145, 2001.
[34] J.C. Dunn, "Well Separated Clusters and Optimal Fuzzy Parti- tions," J. Cybernetics, vol. 4, pp. 95-104, 1974. VOL. 8, NO. 2, MARCH-APRIL 2011
[35] D.L. Davies and D.W. Bouldin, "A Cluster Separation Measure," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 1, no. 2, pp. 224-227, Apr. 1979.
[36] M. Halkidi and M. Vazirgiannis, "Clustering Validity Assessment Using Multi Representatives," Proc. SETN Conf., vol. 2, pp. 237- 249, 2002.
[37] A. Hofmann, I. Dedinski, B. Sick, and H. de Meer, "A Novelty- Driven Approach to Intrusion Alert Correlation Based on Distributed Hash Tables," Proc. 12th IEEE Symp. Computers and Comm. (ISCC '07), pp. 71-78, 2007.
[38] F. Provost and T. Fawcett, "Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions," Proc. Third Int'l Conf. Knowledge Discovery and Data Mining (KDD '97), pp. 43-48, 1997.
[39] J. McHugh, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory," ACM Trans. Information and System Security, vol. 3, no. 4, pp. 262-294, 2000.
[40] M.V. Mahoney and P.K. Chan, "An Analysis of the 1999 DARPA/ Lincoln Laboratory Evaluation Data for Network Anomaly Detection," Recent Advances in Intrusion Detection , G. Vigna, ¨ E. Jonsson, and C. Krugel, eds., pp. 220-237, Springer, 2003.
[41] A. Hofmann, D. Fisch, and B. Sick, "Improving Intrusion Detection Training Data by Network Traffic Variation," Proc. IEEE Three-Rivers Workshop Soft Computing in Industrial Applica- tions, pp. 25-31, 2007.
[42] Sourcefire, Inc., http://www.snort.org/, 2009.
[43] CISCO Systems, Inc., "Cisco PIX Firewall System Log Messages, Version 6.3," http://www.cisco.com/en/US/docs/security/pix/ pix63/system/message/pixemsgs.html, 2009.
[44] Organic Computing, R.P. Wurtz, ed. Springer, 2008.

AUTHORS BIOGRAPHY

KathulaAjith Kumar: is pursuing M.Tech in Software Engg.from Kakatiya Institute Of Technology. CompletedB.Techfrom JNTUHKondagattu in 2011.His interested areas are data mining, database administration**.**

M.S.B.Phridviraj: received the B.Tech degree in Computer science and engineering from JNTU University, Hyderabad, in 2001,M.Tech degree in Computer science and Engineering from VTU university , Belgaum in 2006 and currently doing research work towards the PhD degree in Computer Science and Engineering. He is majoring in computer science and is familiar with the data mining and Stream mining area. His research interests include data mining, Software engineering, Networks, Artificial Intelligence

