OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Online Intrusion Alert Aggregation with Generative Data Stream Modeling

Kothawale Ganesh S.[1], Borhade Sushama R.[2], Prof. B. Raviprasad[3]

[1] *Student, M.Tech(CSE), R R S College of Engineering & Technology, Muthangi (Vill), Patancheru (Mdl), Andhra Pradesh, India , Pin- 502 300.*

[2] *AAEMF's College of Engineering and Management Studies Koregaon Bhima, Tal- Shirur, Dist- Pune, Maharashtra, India PIN- 412216*

[3] *Guide, R R S College of Engineering & Technology, Muthangi (Vill), Patancheru (Mdl),Andhra Pradesh, India,*
Pin- 502 300.

Abstract: *Online intrusion alert aggregation with generative data stream modeling is a approach which uses generative modeling. It also use a method called as probabilistic methods. It can be assume that instances of an attack is similar as a process may be a random process which is producing alerts. This paper aims at collecting and modeling these attacks on some similar parameters, so that attack from beginning to completion can be identified. This collected and modeled alerts is given to security personnel to estimate conclusion and take relative action. With some data sets, we show that it is easy to deduct number of alerts and count of missing meta alerts is also extremely low.*

Also we demonstrate that generation of meta alerts having delay of only few seconds even after first alert is produced already.

Keywords: *online intrusion detection system, data stream, alert aggregation, IDS, offline alert aggregation, online alert aggregation etc.*

I. Introduction

In general, IT system is having huge number of information. This information is always confidential. Providing security to information is essential task in information technology system. To provide information security, emergence of new technologies which are innovative should be happened.

Intrusion Detection System plays an very important role in information security. It can be a device or a software application which is capable to detect outside intrusion as well as monitors inside activities such as unauthorized access. It detects suspicious actions by evaluating TCP/IP connections or LOG files. The working of this IDS is such a way that when it finds some action which is suspicious action then it produces alerts immediately. This alert contains information about source IP address, destination IP address, and possible type of attack. This possible type of attack consist of buffer overflow, denial of service, SQL injection etc. This alert processing is done at very low level of IDS. So it may be possible that single attack instance can have thousands of alerts. It becomes drawback of existing IDS.There are two types of IDS.

1.1 NIDS: NIDS is nothing but Network Based IDS. This IDS is an independent platform. It analyze the traffic on internet. It also monitors many hosts. Network based IDS access network by network tap, network switch, network router etc. In network based IDS sensors are placed, which identifies network traffic and analyze the content. Snort is the example of Network based IDS.

1.2 HIDS: HIDS is Host Based Intrusion Detection System. This IDS is may be dependent or independent platform. Agents are placed in Host based IDS. This agent in Host based IDS analyze log files, system calls and any other activities. Sensors are consists of agents. OSSEC is the example of Host based IDS.

II. Related Work

Existing IDS are having very high accuracy to detect the attacks, but still they have some drawbacks such as alerts are produces at very low level of IDS, thousands of alerts may produce for single attack instances, confusions may happened due to large number of alerts produced in taking appropriate actions while attack is done and so on. Many scientist or publication have done their work to remove these drawbacks. They have provided some direction to do the future enhancement in IDS.

The most suitable way to apply the correlation between different alerts is done in[6]. In this paper reconstruction of alert thread is done. The alerts which are produced by IDS can be aggregated by using some fixed length window. But it can produce duplicates, which should be eliminated for proper working of IDS. So elimination of these types of duplicates is done in[7] by clustering the alerts online as well as offline. First offline algorithm has been developed to eliminates the duplicates and this offline algorithm had been extended for online algorithm. The situation of current attack is done in[8]. The cluster is used to group same attacks is done in [9]. Instead of using alert clustering, another way to correlate alerts is done in [10]. In this paper the process of combining two alerts is done on the basis of weighted, attribute wise similar operator. But from [11] and [12] this way has one disadvantage that large number of parameters are needed to be set.[13] has same disadvantage as[10]. To overcome this disadvantage [14] uses another clustering algorithm that uses user defied parameters. It uses strict sorting based on source and target i.e. destination IP addresses and ports in alerts. [22] uses fully different and unique way for clustering, AA-NN i.e. auto associator neural network's error is reconstructed and it helps to analyze different alerts. Alerts which produces same reconstruction error are grouped or placed into same cluster. The major advantage of this approach is it can be applied to offline as well as online. Offline training is required to do first of all and that can be extended to online training of AA-NN.

III. Online Intrusion Alert Aggregation Technique

In this section, we will discuss our new alert aggregation approach. As we have already stated that it is probabilistic model of current situation of different types of attacks. First of all we start with architecture of our system. The architecture is consists of the diagram showing detailed view and description about the layers in detail. Then we will describe about the process of generation of alerts and the alert format i.e. what are the contents of alerts. After that we discuss about the clustering algorithm for offline alert aggregation and how to extend it to apply it online. At last we prepare result. Analyze it to produce remark for generation of meta alerts. Whatever meta alerts has been produced we will send it to users registered mobile.

3.1 Architecture:

The following figure shows the architecture of proposed system.

3.1.1 Sensor Layer: It is low level layer which acts as an interface between the network and host (agent reside). It captures raw data from both i.e. from network and host, filters it and takes out essential data to create an event. Sensor layer consists of sensors which captures traffic on network

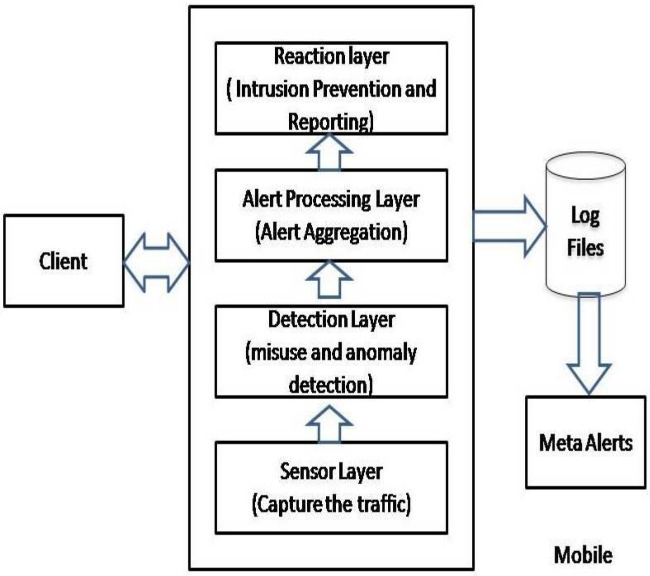


Fig 1. Layered Model of Proposed System

.

3.1.2 Detection Layer: This layer consists of different types of detectors e.g. Support Vector Machines, Snort. It looks for misuse detection and anomaly detection. If it finds suspicious behaviors, it create alerts and forward to next layer of our proposed architecture i.e. alert processing layer.

3.1.3 Alert Processing Layer: Whatever alerts has been received from detection layers that alerts are processed at this layer in such a way that meta alerts is generated. This generation of meta alerts is done on the basis of attack instance information which includes source and destination IP address and possible type of attack.

3.1.4 Reaction Layer: It is something like Intrusion Prevention System which prevents intrusions. Relative and appropriate action is taken for meta alerts produced by alert processing layer.

3.2 Alert Generation and Format:

We have discussed the functions of each layer in proposed system. Now we will discuss process of generation of alerts in detail. Sensors in sensor layer captures traffic over the internet. It also decides attributes as an input for detector in detection layer. This attribute can be used for differentiation of attack instances. They may be dependent or independent. Attributes generated by the detectors are source IP address, target IP address, and possible type of attack which includes denial of service, buffer overflow and SQL injection etc.The format for alert(A) is as follows:

It has N number of attributes. Out of these N attributes let us suppose that N_m are categorical and remaining i.e. N_{m+1} are continuous.

$A = (A_1, \ldots\ldots\ldots, A_{Nm}, A_{Nm+1}, \ldots\ldots\ldots\ldots, A_N)$

Where, $A_1, \ldots\ldots\ldots, A_{Nm}$ are categorical attributes and

$A_{Nm+1}, \ldots\ldots\ldots\ldots, A_N$ are continuous attributes.

3.3 Offline Alert Aggregation:

In this section we develop offline alert aggregation which will be extended to data streaming for online alert aggregation. We can show that different attacks are done on TCP/UDP traffic. Some alerts are false positive and some alerts are false negative. All information then get analyzed and finally offline alert can be generated. But they have some drawbacks.

1. Some of false alerts are not identified and they may get assigned to cluster.
2. Wrong assignment of true alerts to cluster may happened.
3. Splitting of cluster may be wrongly done.
4. Many different clusters may get combined wrongly into one single cluster.

```
Algorithm  :  Offline alert aggregation
Input     : set of alerts (A),
              number of components C
Output     : $\mu_{c,}$  $\sigma_c^2$ , $\rho_c$ parameters
       Assignment of alerts to components.
1.   $\Pi c = 1/C$
2.   Initiate $\sigma_c^2$ , $\rho_c$
3.   While stopping not done do
     // E step : assign alerts to components
4.   For all alerts $A^{(p)}$ ε A do
     C* := argmax H( $a^{(p)}$ |  $\mu_{c,}$  $\sigma_c^2$ , $\rho_c$ )
5.    c ε {1,…….,C}
6.   Assign alert $a^{(p)}$ to C*
     // M step : updating of model parameters.
7.   For all component c ε {1,…….,C} do
8.   $N_c$ := No. of alerts assigned to C
9.   For all attributes n  ε {1,…….,$N_m$} do
10.  $\rho_{cn}$ : = 1 / $N_c$    □  $a_n^{(p)}$
11.  for all attribute n  ε { $N_{m+1}$,………..,  N } do
12.  $\mu_{cn}$ : = 1 / $N_c$    □  $a_n^{(p)}$
13.  $\sigma_{cn}^2$ : = 1 / $N_c$    □  ( $a_n^{(p)}$ - $\mu_{cn}$)$^2$
```

We can conclude from above algorithm that this algorithm performs steps like initialization of model parameters, assignment of alerts to components, updating of model parameters stopping process, coefficient mixing. Next it adds alerts to components slowly.

3.4 Online Alert Aggregation:

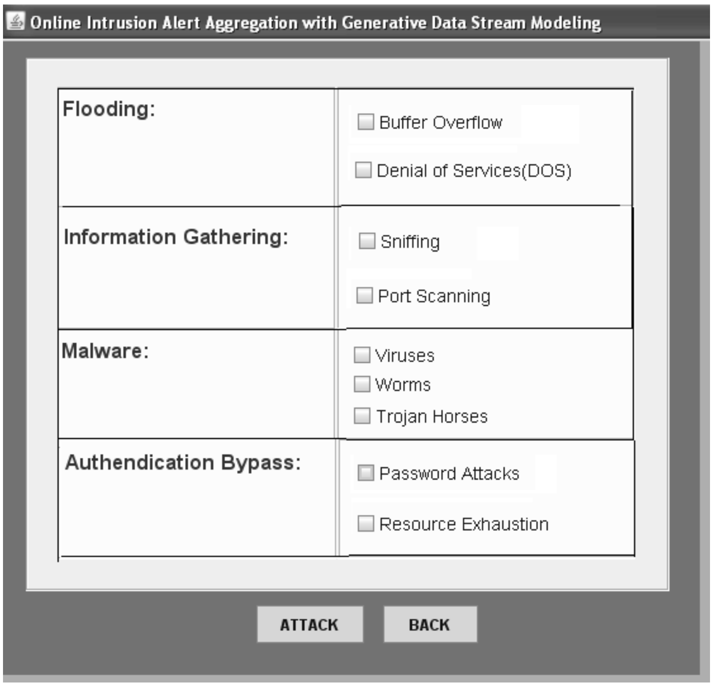
The above algorithm is extended to perform online alert aggregation. For this IDS should have component adaption, component creation and component detection. In component adaption attack instances must be identified and should get assigned to proper cluster. In component creation new attack should created and parameters should set. In component detection attack instances should be detected.

```
Algorithm: online alert aggregation
Input       : buffer B, Partition P, cluster number j
Output      : $\mu_{j,}$  $\sigma_j^2$ , $\rho_j$ parameters
       Assignment of alerts to components.
1.   B : = Φ
2.   While new alert do
```

```
If P : = Φ then
P₁ : = { a}
P : = { P₁ }
Initiate parameters like $\mu_,\ \sigma^2, \rho$
Else
P' : = P
J* : = argmax $H(\mu_{j,}\ \sigma_j^2, \rho_j)$
$\rho_j^* : = \rho_j$ U { a }
$O_j : = | C_j^* |$
For all attributes n  ε {1,…….,$N_m$} do
$\rho_{jn} : = 1 / O_{j(n)}$  □  $a_n^{(p)}$
For all attributes n  ε { $N_{m+1}$,……….., N } do
$\mu_{j\,n} : = 1 / O_{j(n)}$  □  $a_n^{(p)}$
$\sigma_{jn}^2 : = 1 / O_{j(n)}$  □  $(a_n^{(p)} - \mu_{j\,n})^2$
if  Ω( p) < □
P : =  P'
B : = B U {a}
If novelty ( a ) then
P : ALG3 ( C, J*, B)
B : = Φ
For j ε { 1, . . . . . .  | C|} do
If obsoleteness ( Pj) then
P : = P/ Pj
```

IV. Implementation And Results

We have implemented custom simulator by using java programming language. System requirement to do the implementation is JDK 1.6, Eclipse or Netbeans, JME. The operating system used to do the implementation is Windows XP. We have developed graphical user interface by using swing application programming interface. Following are some user interface of attack simulation.



Online Intrusion Alert Aggregation with Generative Data Stream Modeling

Flooding:
□ Buffer Overflow
□ Denial of Services(DOS)

Information Gathering:
□ Sniffing
□ Port Scanning

Malware:
□ Viruses
□ Worms
□ Trojan Horses

Authendication Bypass:
□ Password Attacks
□ Resource Exhaustion

Figure 2. Different Types of Attacks

As shown in figure different attacks can be simulated into information gathering, authentication failure, malwares, and flooding of data. Following is the GUI for alert aggregation

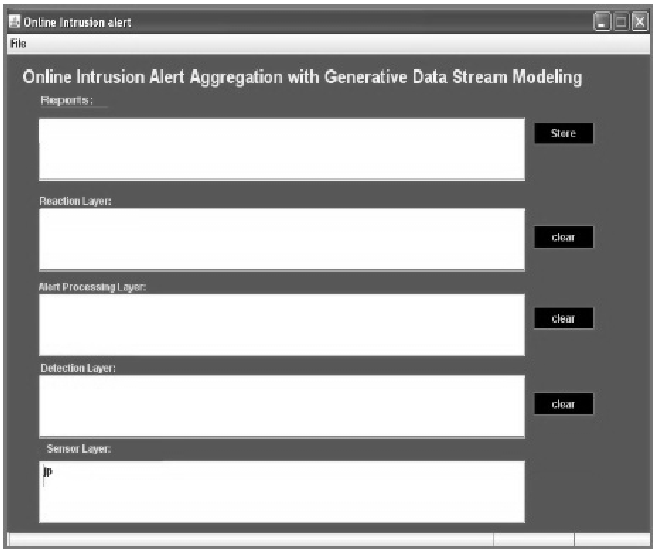
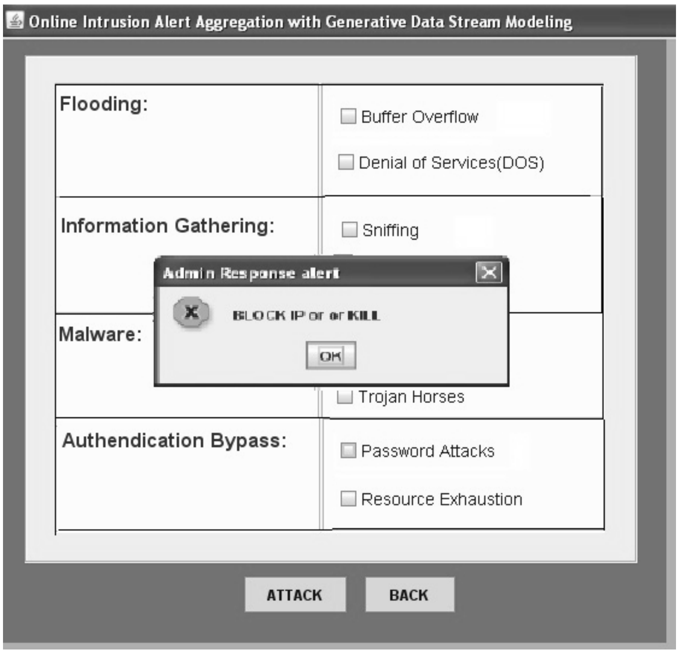


Figure 3 Simulation of Alerts

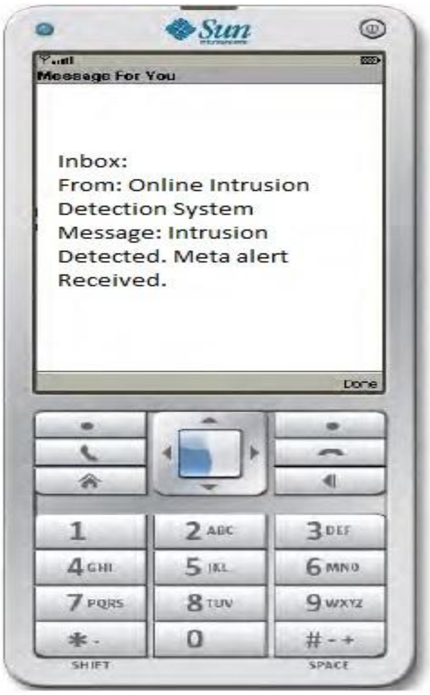
As shown in above figure there is separate space for each and every layers aggregation messages. When attack is done the relevant or appropriate action or message is displayed as shown in figure

Alerts can be send to users registered mobile as shown in figure.



BLOCK IP or or KILL

Figure 4. Response when attack is done



Inbox:
From: Online Intrusion Detection System
Message: Intrusion Detected. Meta alert Received.

Figure 5. GUI of Mobile Alert

V. Conclusion

The proposed way for online alert aggregation generation has been implemented and it found that meta alerts can be generated. Missing false positive rate gets reduced as it uses property of data streaming i.e. it executes a few times only. The experimental result shows that it is very effective and helpful when it gets implemented in real time application. Also IDS accuracy gets increased. More alerts can be detected but compare to number of attacks detected very few false positive alerts gets introduced. So online intrusion alert aggregation with data streaming system is extremely efficient in information technology field to provide security to information.

REFERENCES

[1] Alexander Hofmann, Bernhard Sick, "Online Intrusion Alert Aggregation with Generative Data Stream Modeling", IEEE transaction on dependable and secure computing, vol 8 No. 2 March-April 2011.

[2] M. Hanock, K. Srinivas, A. Yaganteeswarudu, "Online Intrusion Alert Aggregation with Generative Data Stream Modeling",International Journal of Electronics and computer Science Engineering, ISSN-2277-1956

[3] S. Mangesh kumar, K. Mohan, G. Kadirvelu, S. Muruganandam, "Online Intrusion Alert Aggregation Through Mobile", International journal of advancement in Research and Technology, volume 1, issue3, August-2012

[4] Ravindra Bhat,"Intrusion Detection System with Data Stream Modeling using Conditional Privilages", International journal of computer science and technology, vol.3 no.7july 2012 ISSN:2299-3345.

[5] Rupali Shewale, Yugandhar Pandey, Maheshkumar A. Sali, " Distributed Intrusion Alert Aggregation with Data Stream Modeling", International journal of electronics,communication and soft computing science and engineering, ISSN:2277-9477 March-2012.

[6] V. SrujanarReddy , G. Dileep Kumar, "Online and Offline Intrusion alert aggregation" , International Journal of Computer Scirence and Communication Networks, vol.2(4), 520-525 ISSN:2249-5789

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

[8] M.R. Endsley, "Theoretical Underpinnings of Situation Awareness: 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.

[9] C.M. Bishop, Pattern Recognitin and Machine Learning. Springer, 2006.

[10] M.R. Henzinger, P. Raghavan, and S. Rajagopalan, Computing on Data Streams. Am. Math. Soc., 1999.

[11] A. Allen, "Intrusion Detection Systems: Perspective," Technical Report DPRO-95367, Gartner, Inc., 2003.

[12] F. Valeur, G. Vigna, C. Kru¨ gel, and R.A. Kemmerer, "A Comprehensive Approach to Intrusion Detection Alert Correlation," IEEE Trans. Dependable and Secure Computing, vol. 1, no. 3, pp. 146-169, July-Sept. 2004.

[13] 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.

[14] 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.

[15] F. Cuppens, "Managing Alerts in a Multi-Intrusion Detection Environment," Proc. 17th Ann. Computer Security Applications Conf. (ACSAC '01), pp. 22-31, 2001.

[16] 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.

[17] K. Julisch, "Using Root Cause Analysis to Handle Intrusion Detection Alarms," PhD dissertation, Universita¨ t Dortmund, 2003.

[18] T. Pietraszek, "Alert Classification to Reduce False Positives inIntrusion Detection," PhD dissertation, Universita¨ t Freiburg, 2006.

[19] F. Autrel and F. Cuppens, "Using an Intrusion Detection AlertSimilarity Operator to Aggregate and Fuse Alerts," Proc. Fourth Conf. Security and Network Architectures, pp. 312-322, 2005.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] A. Hofmann, "Alarmaggregation und Interessantheitsbewertung in einem dezentralisierten Angriffserkennungsystem," PhD dissertation, Universita¨t Passau, under review.

[26] 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.

[27] 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.

[28] 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. 308-31 , Springer, 2008.

[29] 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.

[30] M. Roesch, "Snort—Lightweight Intusion Detection for Networks," Proc. 13th USENIX Conf. System Administration (LISA '99), pp. 229-238, 1999.

[31] 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.