Correlating Intrusion Alerts into Attack Scenarios based on Improved Evolving Self-Organizing Maps

Yun Xiao and Chongzhao Han
Integrated Automation Institute,
School of Electronic & Information Engineering,
University of Xi'an Jiaotong, Xi'an, Shaanxi, 710049, P.R. China

Summary
Traditional intrusion detection systems (IDSs) focus on low-level attacks and anomalies, and raise alerts independently, though there may be logical connections between them. In this paper, a method of correlating intrusion alerts into attack scenarios based on the improved evolving self-organizing map (IESOM) was proposed. IESOM gives a rational formula to calculate the initial values of connection strengths instead of assigning some experiential or tentative constants as connection strength values in ESOM. IESOM is an evolving extension of the self-organizing map (SOM) model, which allows for an evolvable network structure and very fast incremental learning. System of correlating intrusion alerts into attack scenarios based on IESOM has four functions of filtering, aggregation, condensing and combination, and the visual attack scenarios are given as the output of the system. The results on LLS DDOS1.0 and real-word dataset B prove that our method is useful and effective.

Key words:
Intrusion alert, correlation, attack scenarios, improved evolving self-organizing map

1 Introduction

Security and privacy are the growing concerns in the open distributed software systems due to Internet's rapid growth and the desire to conduct business over it safely. This desire has led to the advent of many security architectures and protocols, which deals with authentication, cryptography, and authorization to avoid a possible intrusion. There has been significant work in the field of intrusion detection that comes into picture after an attack. But traditional intrusion detection systems still have some weaknesses [1]:

1. Flooding. Intrusion detection systems provide a large number of alerts to the operator, who then has difficulties coping with the load.
2. Context. Attacks are likely to generate multiple related alerts, which make it difficult for the operator to logically group related alerts.

Since intrusion detection systems generally focus on low-level attacks and anomalies, and raise alerts independently, though there may be logical connections between them, network administrators are often overwhelmed by large volumes of alerts. This has motivated recent work in alert aggregation, to reduce administrator workload and provide higher-level situational awareness. In fact, intrusion alerts correlation is the key to the two challenges: it consists in reducing and interpreting multiple intrusion alerts such that new meanings are assigned to these alerts. Various approaches have been proposed to correlate intrusion alerts. Debar and Wespi proposed an aggregation and correlation algorithm to acquire intrusion detection alerts and relate them together to expose a more condensed view of the security issues raised by intrusion detection systems [1]. In [2], the approach to causal correlation is to define logical rules that relate generic intrusion events through preconditions/ postconditions. The approach in [2] does include merging of identical alerts, and the alert merging is accomplished through clustering correlation, a form of correlation that has been described by other authors, e.g., [3][4][5]. In this paper, a new clustering method, improved evolving self-organizing maps (IESOM), was proposed to aggregate multiple alerts. Then the clustering results are condensed and united to attack scenarios. The advantages from correlating intrusion alerts are as following:

1. Condensing alerts. Repeated alerts can be saved only one, so the number of alerts is reduced, witch exposes a condensed view of the security issues raised by IDSs.
2. Clear attack scenarios. The operator can understand related alerts easily form these clear attack scenarios,

The remainder of this paper is organized as follows. Section 2 describes the theory and algorithm of IESOM. The structure and processes of correlating alerts into attack scenarios are delineated in section 3. Two experiments that illustrate the properties of our system are showed in section 4. In the end, conclusions and future work are given in section 5.

2 Improved Evolving Self-organizing Maps

A self-organizing map (SOM), or Kohonen network is a form of competitive learning artificial neural network [6].

Manuscript revised June 21 , 2006.

SOM needs determinate the class label and has a shortcoming of slow convergence, which makes it not adaptive for the data that is large and hard to determine the class label. Based on SOM, an algorithm called evolving self-organizing map (ESOM) was proposed [7]. ESOM adopts soft modification to prototype nodes in neighbourhood, and is a computational model for on-line pattern learning. But when we use ESOM, we find that it is difficult to choose reasonable value of initial connection strengths between the winner and other nodes, which can affect the structure and the cluster precision. So we improved the algorithm, and we called the new algorithm as improved evolving self-organizing map (IESOM). IESOM gives a rational formula to calculate the initial values of connection strengths instead of assigning some experiential or tentative constants as connection strength values in ESOM. IESOM also gives the more intuitionistic definition of neighbourhood. IESOM is an evolving extension of the SOM model, allowing for a growing network structure and very fast incremental learning.

2.1 The learning rule of IESOM

IESOM is starting from a null network, and new prototypes are gradually allocated when new data samples cannot be matched well into existing prototypes. Assume the current input is $\boldsymbol{x}$, and the existing prototype node is $\boldsymbol{W}_i$ ($i = 1, \cdots, N$). If

$$d_g = \min(\|\boldsymbol{W}_i - \boldsymbol{x}\|) \geq r, i \in [1, N] \quad (1)$$

where r is a growing threshold, and subscript g denotes the node which has minimum distance, then a new node is inserted as

$$\boldsymbol{W}_{N+1} = \boldsymbol{x} \quad (2)$$

The growing threshold r affects the size of neural network, and the smaller r is, the more number of nodes are and the more complex the network structure is. If the new input matches well to some prototypes, that means $d_g < r$, the activation on the prototype nodes is defined as

$$\alpha_i(\boldsymbol{x}) = \exp(-\|\boldsymbol{x} - \boldsymbol{W}_i\|^2 / r^2), i \in [1, N] \quad (3)$$

It indicates the proximity of the current input to weight vector $\boldsymbol{W}_i$.

The strength of connections between the winner g and other nodes indicates the closeness of nodes, and is defined as

$$S_i(0) = \exp(-\|\boldsymbol{W}_g - \boldsymbol{W}_i\|^2 / r^2), i \in [1, N] \quad (4)$$

The neighbourhood of node g indicates the region with g as its center and including some nodes. It is defined as

$$N_g(t) = \{i \,|\, S_i(t) > \eta\}, \forall i \in [1, N] \quad (5)$$

where η is connection threshold.

The neighbourhood function is defined within the neighbourhood as

$$h_{i,g}(\boldsymbol{x}) = \alpha_i(\boldsymbol{x}) / \sum_k \alpha_k(\boldsymbol{x}) \quad (6)$$

Such a definition of neighbourhood saves the rigid topological constraint in SOM, and the time-consuming neighbourhood ranking in neural gas.

Hence, we have the learning rule of IESOM as

$$\Delta \boldsymbol{W}_i = \begin{cases} \varepsilon(t) \dfrac{\alpha_i(\boldsymbol{x})}{\sum_k \alpha_k(\boldsymbol{x})} (\boldsymbol{x} - \boldsymbol{W}_i), i \in N_g(t) \\ 0, \qquad\qquad i \notin N_g(t) \end{cases} \quad (7)$$

where $\varepsilon(t)$ is the learning rate. The connection strength is updated as following:

$$S_i(t+1) = \beta S_i(t) + (1-\beta)\alpha_i(\boldsymbol{x})\alpha_g(\boldsymbol{x}) \quad (8)$$

where β is a forgetting constant.

2.2 Algorithm summary

The learning algorithm of IESOM is summarized in following steps:

Step 0. Determine the initial value of learning rate, the maximal iteration number, the growing threshold r and the connection threshold η.

Step 1. Input a new data vector $\boldsymbol{x}$. If there are no prototype nodes, go to step 3; otherwise go to step 2.

Step 2. Look for a winner among the prototype. If the minimum distance is smaller than the threshold r, go to step4; otherwise go to step 3.

Step 3. Create a new node in network representing the input according to Eq. 2.

Step 4. Modify the winner, its neighbors and their connections with Eq. 5, Eq. 7 and Eq. 8.

Step 5. After $t_{\max}$ steps of learning time, assign the label of $\boldsymbol{x}$ to the winner.

Step 6. Go back to step 1 (until no more data are available).

2.3 Parameters Setting

The forgetting constant β is usually set as 0.8. The growing threshold r affects the size of neural network, and it is usually set as the same quantitative level of $\|\boldsymbol{W}_i - \boldsymbol{x}\|$. The connection threshold η affects the size of the neighbourhood $N_g(t)$, and the radius of $N_g(t)$ can even be more than half the diameter of the network with the reasonable η. $\varepsilon(t)$ should start with a value that is close to unity, thereafter decreasing monotonically. ε_i =0.9 may be a reasonable choice.

3 Correlating Intrusion Alerts into Attack Scenarios

3.1 The structure of correlating intrusion alerts into attack scenarios

Intrusion alert is denoted as A={time, attack, source_ip, target_ip, source_port, target_port}. The fundamental functions of intrusion alert correlation involve the filtering, aggregation, condensing and combination. These terms are defined as follows:

Filtering. $[A, p(A) \in H] \Rightarrow \varnothing$. Delete the intrusion alerts that have a value $p(A)$ belonging to the invalid value set H. For example, an alert with an invalid time stamp must be deleted.

Aggregation. $[A_1, A_2, \cdots, A_n] \Rightarrow A_c$. These similar intrusion alerts are grouped together with IESOM.

Condensing. $[A_{cj}, A_{cj}, \cdots, A_{cj}] \Rightarrow A_{cj}$. For each cluster, these repeated alerts only reserve one.

Combination. $[A_{cm}, T, A_{cn}] \Rightarrow A_I$. Intrusion events that belong to the same attack pattern are united to an attack scenario according to the order of their time T. An attack pattern, which consists of multiple and multistage attacks, describes a big and integrated attack process caused by a certain hacker. Attack patterns can be extracted from plentiful attack data. In this paper we just use the extracted attack patterns and don't discuss how they were extracted. Now we have some attack patterns such as Scan-BufferOverflow-Action, Unicode, DDOS, Worm and so on. The structure of correlating intrusion alerts into attack scenarios system is showed in figure 1.

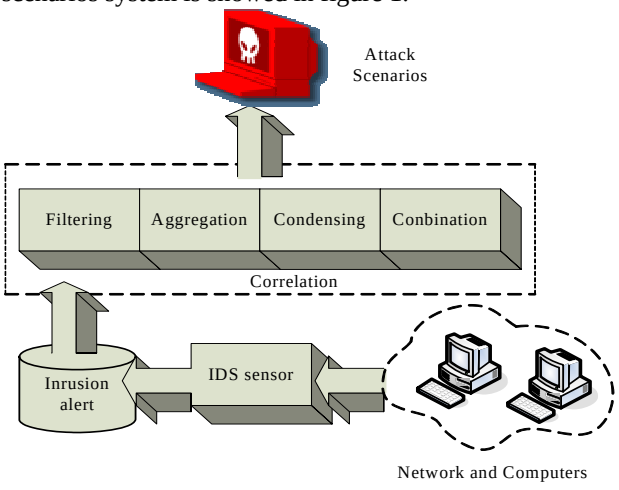

Fig. 1 The structure of correlating intrusion alerts into attack scenarios system

4 Implementation of correlating intrusion alerts into attack scenarios

In this section, we report the two experiments we performed to evaluate the effectiveness of the proposed method. The first experiment was done on LLS DDOS1.0 of the 2000 DARPA intrusion detection scenario specific datasets [8]. LLS DDOS1.0 contains a series of attacks in which an attack probes, breaks in, installs the components necessary to launch a DDOS attack, and actually launches a DDOS attack against an off-site server. Each dataset includes the network traffic collected from both the DMZ and the inside part of the evaluation network. We utilized TCPREPLAY to replay the selected network traffic in an isolated network monitors by our network sensor XJTU-sensor. TCPREPLAY is a utility to replay packets that were previously captured with tcpdump program to a live network. XJTU-sensor is a good network sensor developed by our security research team. Intrusion alerts in LLS DDOS1.0 are showed in table 1.

Table 1: Intrusion alerts in LLS DDOS1.0

Alerts' name	*Alerts' number*
FTP_Syst	1
Email_Almail_Overflow	2
Sadmind_Ping	3
TelnetTerminaltype	126
Email_Debug	2
Mstram_Zombie	6
Sadmind_Amslverify_Overflow	14
SSH_Detected	4
TelnetXdisplay	1
HTTP_Java	8
Stream_Dos	1
Port_Scan	1
HTTP_Cisco_Catalyst_Exec	2
TelnetEnvAll	1
SNMP_Suspicious_Get	1
RIPExpire	1
RIPAdd	1
HTTP_Shells	15
HTTP_Cookie	37

We clustered these alerts with IESOM clustering algorithm, and condensed the clustering result, so we gained attack scenarios according to the DDOS attack pattern. The attack scenario from LLS DDOS1.0 is showed in figure 2. These correlated intrusion alerts can be divided into five stages horizontally. The first stage

consists of three Sadmind_Ping events, which the attacker used to find out the vulnerable Sadmind services. This intrusion alerts in the first stage are from source IP address 202.077.162.213, and target IP addresses 17.016.112.110, 172.016.115.020, and 172.016.112.050, respectively. The second stage consists of some Sadmind_Amslverify_Overflow alerts. The attacker tried some different stack pointers and commands in Sadmind_Amslverify_Overflow attacks for each victim host until one attempt succeeded. All the above three hosts were successfully broken into. The third stage consists of some Rsh alerts that are hyper-alerts including some alerts about telnet service. The attacker installed and started the mstream daemon and master programs. The fourth stage consists of intrusion alerts corresponding to the communications between the mstream master and daemon programs. Finally, the last stage consists of a DDOS alert.

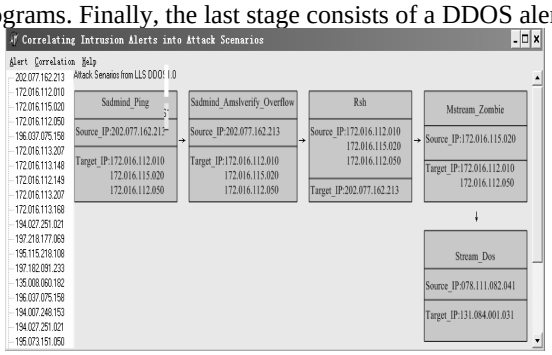

Fig. 2. The attack scenario from LLS DDOS1.0

The second experiment was on the real-word dataset, called dataset B, namely the intrusion alerts monitored by our network sensor, called XJTU-sensor, in the centre of northwest network of CERNET over the period of a week. We do not claim that it is representative for all real-word sets, but it is an example of a real dataset. Hence, we used this dataset as a second validation of our method. Dataset B has 11629 intrusion alerts that over 95% are scan alerts, which shows that intrusion detection system generally focus on low-level alerts. We used our method on dataset B and obtained 346 groups of clustering results. We condensed the clustering results and gained 31 groups of correlated intrusion events according to the attack patterns. Owing to privacy issues only one attack scenario is showed in figure 3. The hacker form IP address 202.114.245.196 first tried to scan the victim host with IP address 202.200.046.065, then broke into the victim host by some FTP_Buffer_Overflow attacks. At last the hacker downloaded a hacker tool to the victim host from the host with IP address 202.117.129.016.

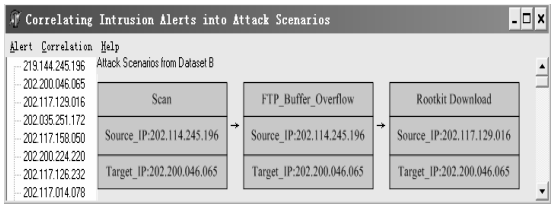

Fig. 3. Part of attack scenarios from dataset B

5 Conclusion and future work

This paper presented a method for correlating intrusion alerts into attack scenarios based on the improved evolving self-organizing map. IESOM gives a rational formula to calculate the initial values of connection strengths instead of assigning some experiential or tentative constants as connection strength values in ESOM. IESOM is an evolving extension of the self-organizing map (SOM) model, allowing for an evolvable network structure and very fast incremental learning. System of correlating intrusion alerts into attack scenarios based on IESOM has four functions of filtering, aggregation, condensing and combination, and the visual attack scenarios are given as the output of the system. The results on LLS DDOS1.0 and real-word dataset B prove that our method is useful and effective to correlate intrusion alerts into attack scenarios. Our next work is to collect new-style attack data to evaluate our method ulteriorly. So it can do better in the Integrate Network Guard System Net-Keeper.

Acknowledgments

This paper was partly supported by a grant from the National High Technology Research and Development Program of China (2004AA1Z2280) and a grant from the Major State Basic Research Development Program of China (2001CB309403). The authors thank Professor Qinghua Zheng for his good advices.

References

[1] Debar H, Wespi A. Aggregation and Correlation of Intrusion Detection Alerts. Proceedings of Fourth International Symposium on Recent Advance in Intrusion Detection, UC Davis, 2001, 85-103.

[2] Cuppens F, Miege A. Alert Correlation in a Cooperative Intrusion Detection Framework. Proceeding of the 2002 IEEE symposium on security and Privacy, Oakland, 2002, 202-215.

[3] YS. Wu, B. Foo, Y. Mei & S. Bagchi. Collaborative Intrusion Detection System (CIDS): A Framework for Accurate and Efficient IDS. Proceedings of the 19th Annual Computer Security Applications Conference, Las Vegas, Nevada, December 2003, 234-244.

[4] Valdes A, Skinner K. Probabilistic Alert Correlation. Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection, Davis, California, 2001, 54-68.
[5] Ning P, Cui,Y, Reeves, D. S . Analyzing intensive intrusion alerts via correlation. Proceeding of the 5th International Symposium on Recent Advance in Intrusion Detection (RAID 2002). Zurich, Switzerland. 2002, 74-94.
[6] Kohonen T. Self-organized formation of topographically correct feature maps. Biological Cybernetics, vol. 43(1), 1982, 59-69.
[7] Deng D, Kasabov N. On-line pattern analysis by evolving self-organizing maps. Neurocomputing, vol. 51, 2003, 87-103.
[8] http://www.ll.mit.edu/IST/ideval/data/2000/LLS_DDOS_1.0.html

Yun Xiao is a PH.D. student in the school of electronic & information engineering at Xi'an Jiaotong University. She received the B.S. and M.S. degrees in Control Theory and Control Engineering from Xi'an University of Science and Technology in 2000 and 2003, respectively. Her research interests are in network security and information fusion.

Chongzhao Han received the B.S. and M.S. degrees from Xi'an Jiaotong University and Graduate University of Chinese Academy of Science in 1968 and 1981, respectively. He is a Ph.D. Advisor in the school of electronic & information engineering at Xi'an Jiaotong University. His research interests are in stochastic control, nonlinear theory and information fusion.